

한국스마트정보교육원

42기 1팀 프로젝트 완료 보고서

프로젝트명
AI 쌍방향 홈 트레이닝 시스템

팀 원	최보라
	최민희
	이영호
	김재환
	이민식
	진수경
작성일	2022년 4월 21일

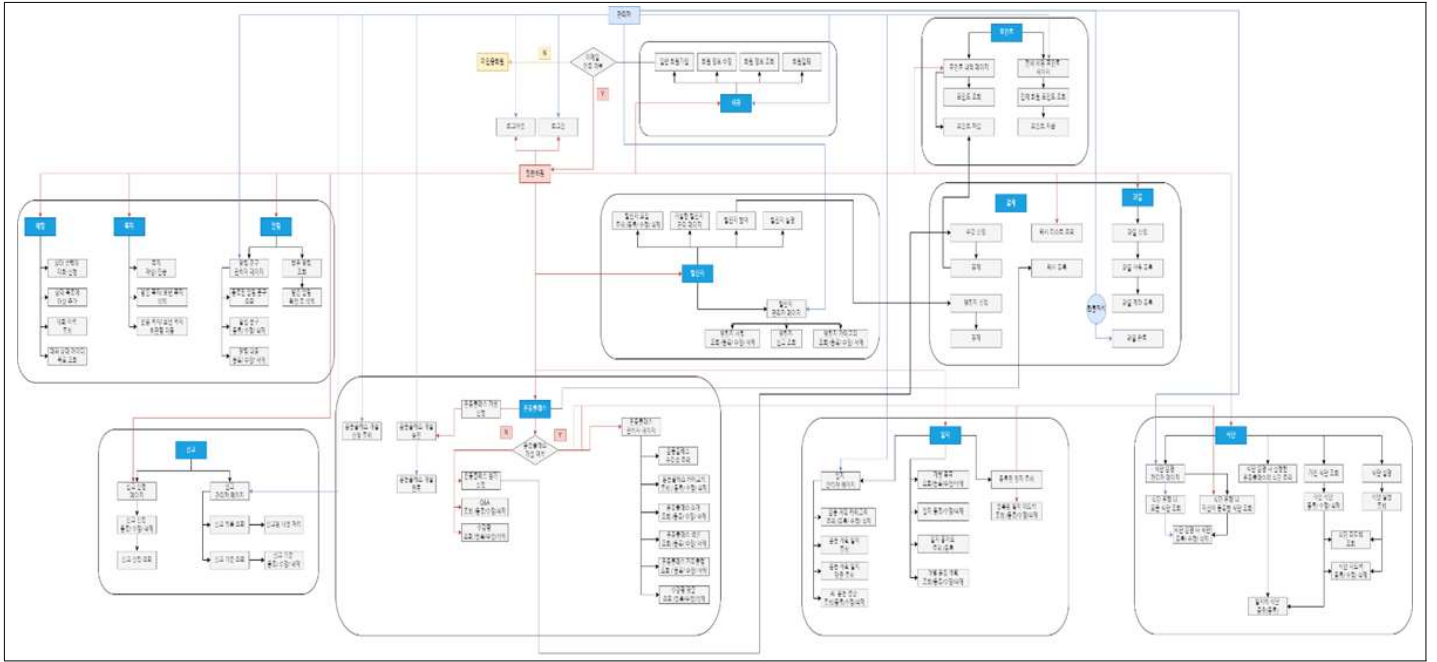
호스팅	http://ddoggang12.cafe24.com
GitHub	https://github.com/codeco33/aihometraining

목차
1 . 프로젝트 기본 정보
1.1 시스템 흐름도
2 . 요구사항 정의서
2.1 기능적 요구사항
2.2 상세 기능 정의
3 . 업무 분담(담당 업무)
4 . 설계 사양서
4.1 ERD
4.2 테이블 구조
4.3 네이밍 규칙
4.4 디렉토리 구조
4.5 패키지 구조
5 . 개발환경
6. 어플리케이션
6.1 어플리케이션 프로시저
6.2 서버 및 기타 설정
6.3 주요 기능별 정리

1. 프로젝트 기본 정보

<p>프로젝트명</p>	<h2>AI 쌍방향 홈 트레이닝 시스템</h2>	
<p>개발 기간</p>	<p>■ 2021년 12월 01일 - 2022년 04월 21일 (142일)</p>	
	<p>2021년 12월 01일 ~ 2021년 12월 31일 (31일)</p>	<ul style="list-style-type: none"> ● 프로젝트 주제 선정 및 사전조사
	<p>2022년 01월 01일 ~ 2022년 03월 06일 (65일)</p>	<ul style="list-style-type: none"> ● 시스템 구조도 작성 ● 설계자료 취합, 수정 및 보완 ● ERD 작업 및 메서드, 패키지 정리
	<p>2022년 03월 07일 ~ 2022년 03월 09일 (3일)</p>	<ul style="list-style-type: none"> ● 서버 설정 ● 개발환경 설정 ● Hosting 설정
	<p>2022년 03월 10일 ~ 2022년 04월 15일 (37일)</p>	<ul style="list-style-type: none"> ● 기능 구현(개인 분담 프로그래밍)
	<p>2022년 04월 16일 ~ 2022년 04월 19일 (4일)</p>	<ul style="list-style-type: none"> ● 테스트, 오류 수정 및 보완 ● 미완성 기능 구현
	<p>2022년 04월 20일 ~ 2022년 04월 21일 (2일)</p>	<ul style="list-style-type: none"> ● 완료 보고서 작성
<p>개발 목적</p>	<p>운동 목표 달성에 어려움을 느끼는 사람들의 목표 달성을 도움을 준다. 운동 목표 달성을 위한 객관적인 지표를 제공한다. 합리적인 수익 관계를 형성한다.</p> <p>프로젝트는 다음과 같은 효과를 기대한다.</p> <p>사용자 : 비대면으로 체계적인 운동 계획 수립과 실천을 할 수 있다. 트레이너 : 비대면으로 서비스 제공 가능, 경력 관리, 수익 창출할 수 있다. 관리자 : 공간 제약이 없고, 초기 자본 절약이 가능하다. 목표 달성률 상승으로 인한 이용자 만족도가 증가한다. 퍼스널 트레이닝 시장의 활성화를 유도한다.</p>	
<p>개발 방식</p>	<pre> graph TD A[계획] --> B[요구분석] B --> C[설계] C --> D[중요설정 우선 코딩, 테스트] D --> E[우선순위 코딩, 테스트] E --> F[출시] F --> G[낮은 순위 코딩, 테스트] G --> B G --> C G --> D G --> E G --> F G --> A </pre>	

1.1 시스템 구조도



2. 요구사항 정의서

2.1 기능적 요구사항

전체 기능	관리자	일반 회원	미인증 회원
회원 관리			
회원가입	회원가입	회원가입	회원가입
회원 정보 수정	회원 정보 수정	회원 정보 수정	
회원 탈퇴	회원 탈퇴	회원 탈퇴	
아이디 찾기	아이디 찾기	아이디 찾기	
비밀번호 찾기	비밀번호 찾기	비밀번호 찾기	
전체 회원 조회	전체 회원 조회		
권한별 검색	권한별 검색	권한별 검색	
회원 개인 추가 정보 입력	회원 개인 추가 정보 입력	회원 개인 추가 정보 입력	
회원 개인 추가 정보 수정	회원 개인 추가 정보 수정	회원 개인 추가 정보 수정	
회원 개인 추가 정보 삭제	회원 개인 추가 정보 삭제	회원 개인 추가 정보 삭제	
회원 정보 보기	회원 정보 보기	회원 정보 보기	
회원 결제 조회	회원 결제 조회	회원 결제 조회	
회원 환불 관리	회원 환불 관리	회원 환불 관리	
로그인 관리			
각자 권한 로그인	각자 권한 로그인	각자 권한 로그인	
로그아웃	로그아웃	로그아웃	
운동 클래스 등록			
운동 클래스 개설 신청	운동 클래스 개설 신청	운동 클래스 개설 신청	
운동 클래스 개설 신청 현황 조회	운동 클래스 개설 신청 현황 조회	운동 클래스 개설 신청 현황 조회	
운동 클래스 개설 승인	운동 클래스 개설 승인		
미승인 사유 메시지 조회	미승인 사유 메시지 조회	미승인 사유 메시지 조회	
미승인 사유 메시지 등록	미승인 사유 메시지 등록		
미승인 사유 메시지 수정	미승인 사유 메시지 수정		
미승인 사유 메시지 삭제	미승인 사유 메시지 삭제		

운동 클래스 카테고리 관리			
운동 클래스 카테고리 조회	운동 클래스 카테고리 조회	운동 클래스 카테고리 조회	운동 클래스 카테고리 조회
운동 클래스 카테고리 대 등록	운동 클래스 카테고리 대 등록		
운동 클래스 카테고리 대 수정	운동 클래스 카테고리 대 수정		
운동 클래스 카테고리 대 삭제	운동 클래스 카테고리 대 삭제		
운동 클래스 카테고리 중 등록	운동 클래스 카테고리 중 등록		
운동 클래스 카테고리 중 수정	운동 클래스 카테고리 중 수정		
운동 클래스 카테고리 중 삭제	운동 클래스 카테고리 중 삭제		
운동 클래스 카테고리 소 등록	운동 클래스 카테고리 소 등록		
운동 클래스 카테고리 소 수정	운동 클래스 카테고리 소 수정		
운동 클래스 카테고리 소 삭제	운동 클래스 카테고리 소 삭제		

승인된 운동 클래스 검색

승인된 운동 클래스 조회	승인된 운동 클래스 조회	승인된 운동 클래스 조회	승인된 운동 클래스 조회
승인된 운동 클래스 조건별 검색	승인된 운동 클래스 조건별 검색	승인된 운동 클래스 조건별 검색	승인된 운동 클래스 조건별 검색

운동 클래스 관리 - 일반

클래스 소개 조회	클래스 소개 조회	클래스 소개 조회	클래스 소개 조회
클래스 소개 등록	클래스 소개 등록	클래스 소개 등록	
클래스 소개 수정	클래스 소개 수정	클래스 소개 수정	
클래스 소개 삭제	클래스 소개 삭제	클래스 소개 삭제	
공지사항 등록	공지사항 등록	공지사항 등록	
공지사항 수정	공지사항 수정	공지사항 수정	
공지사항 삭제	공지사항 삭제	공지사항 삭제	
공지사항 조회	공지사항 조회	공지사항 조회	공지사항 조회
운동 클래스 Q&A 질문 조회	운동 클래스 Q&A 질문 조회	운동 클래스 Q&A 질문 조회	운동 클래스 Q&A 질문 조회
운동 클래스 Q&A 질문 등록	운동 클래스 Q&A 질문 등록	운동 클래스 Q&A 질문 등록	
운동 클래스 Q&A 질문 수정	운동 클래스 Q&A 질문 수정	운동 클래스 Q&A 질문 수정	
운동 클래스 Q&A 질문 삭제	운동 클래스 Q&A 질문 삭제	운동 클래스 Q&A 질문 삭제	
운동 클래스 Q&A 답변 조회	운동 클래스 Q&A 답변 조회	운동 클래스 Q&A 답변 조회	운동 클래스 Q&A 답변 조회
운동 클래스 Q&A 답변 등록	운동 클래스 Q&A 답변 등록	운동 클래스 Q&A 답변 등록	
운동 클래스 Q&A 답변 수정	운동 클래스 Q&A 답변 수정	운동 클래스 Q&A 답변	

수정 운동 클래스 Q&A 답변 삭제	운동 클래스 Q&A 답변 삭제	수정 운동 클래스 Q&A 답변 삭제	
수강평 조회	수강평 조회	수강평 조회	수강평 조회
수강평 등록	수강평 등록	수강평 등록	
수강평 수정	수강평 수정	수강평 수정	
수강평 삭제	수강평 삭제	수강평 삭제	
수강평 댓글 조회	수강평 댓글 조회	수강평 댓글 조회	수강평 댓글 조회
수강평 댓글 등록	수강평 댓글 등록	수강평 댓글 등록	
수강평 댓글 수정	수강평 댓글 수정	수강평 댓글 수정	
수강평 댓글 삭제	수강평 댓글 삭제	수강평 댓글 삭제	
운동 클래스 식단 조회	운동 클래스 식단 조회	운동 클래스 식단 조회	
수강생 조회(결제 완료된 수강생 조회)	수강생 조회(결제 완료된 수강생 조회)	수강생 조회(결제 완료된 수강생 조회)	
수강생 조건별 검색	수강생 조건별 검색	수강생 조건별 검색	

운동 클래스 - 섹션 및 커리큘럼

운동 클래스 내 섹션 조회	운동 클래스 내 섹션 조회	운동 클래스 내 섹션 조회	운동 클래스 내 섹션 조회
운동 클래스 내 섹션 등록	운동 클래스 내 섹션 등록	운동 클래스 내 섹션 등록	
운동 클래스 내 섹션 수정	운동 클래스 내 섹션 수정	운동 클래스 내 섹션 수정	
운동 클래스 내 섹션 삭제	운동 클래스 내 섹션 삭제	운동 클래스 내 섹션 삭제	
운동 클래스의 섹션 내 커리큘럼 등록	운동 클래스의 섹션 내 커리큘럼 등록	운동 클래스의 섹션 내 커리큘럼 등록	
운동 클래스의 섹션 내 커리큘럼 수정	운동 클래스의 섹션 내 커리큘럼 수정	운동 클래스의 섹션 내 커리큘럼 수정	
운동 클래스의 섹션 내 커리큘럼 삭제	운동 클래스의 섹션 내 커리큘럼 삭제	운동 클래스의 섹션 내 커리큘럼 삭제	
운동 클래스의 섹션 내 커리큘럼 조회	운동 클래스의 섹션 내 커리큘럼 조회	운동 클래스의 섹션 내 커리큘럼 조회	운동 클래스의 섹션 내 커리큘럼 조회

운동 클래스 - 과제

운동 클래스 내 커리큘럼 내 과제 조회	운동 클래스 내 커리큘럼 내 과제 조회	운동 클래스 내 커리큘럼 내 과제 조회	운동 클래스 내 커리큘럼 내 과제 조회
운동 클래스 내 커리큘럼 내 과제 등록	운동 클래스 내 커리큘럼 내 과제 등록	운동 클래스 내 커리큘럼 내 과제 등록	
운동 클래스 내 커리큘럼 내 과제 수정	운동 클래스 내 커리큘럼 내 과제 수정	운동 클래스 내 커리큘럼 내 과제 수정	
운동 클래스 내 커리큘럼 내 과제 삭제	운동 클래스 내 커리큘럼 내 과제 삭제	운동 클래스 내 커리큘럼 내 과제 삭제	
운동 클래스 내 커리큘럼 내 과제 수행 조회	운동 클래스 내 커리큘럼 내 과제 수행 조회	운동 클래스 내 커리큘럼 내 과제 수행 조회	운동 클래스 내 커리큘럼 내 과제 수행 조회
운동 클래스 내 커리큘럼 내 과제 수행 등록	운동 클래스 내 커리큘럼 내 과제 수행 등록	운동 클래스 내 커리큘럼 내 과제 수행	

운동 클래스 내 커리큘럼 내 과제 수행 수정	운동 클래스 내 커리큘럼 내 과제 수행 수정	등록 운동 클래스 내 커리큘럼 내 과제 수행 수정	
운동 클래스 내 커리큘럼 내 과제 수행 삭제	운동 클래스 내 커리큘럼 내 과제 수행 삭제	운동 클래스 내 커리큘럼 내 과제 수행 삭제	

식단 - 식품 영양성분

식품영양성분 - 개별음식/재료 조회	식품영양성분 - 개별음식/재료 조회	식품영양성분 - 개별음식/재료 조회	
식품영양성분 - 개별음식/재료 등록	식품영양성분 - 개별음식/재료 등록		
식품영양성분 - 개별음식/재료 수정	식품영양성분 - 개별음식/재료 수정		
식품영양성분 - 개별음식/재료 삭제	식품영양성분 - 개별음식/재료 삭제		

식단 - 식단 은행

식단 은행 식단 조회	식단 은행 식단 조회	식단 은행 식단 조회	
식단 은행 식단 등록	식단 은행 식단 등록	식단 은행 식단 등록	
식단 은행 식단 수정	식단 은행 식단 수정	식단 은행 식단 수정	
식단 은행 식단 삭제	식단 은행 식단 삭제	식단 은행 식단 삭제	
식단 은행 한끼 음식 종류 조회	식단 은행 한끼 음식 종류 조회	식단 은행 한끼 음식 종류 조회	
식단 은행 한끼 음식 종류 등록	식단 은행 한끼 음식 종류 등록	식단 은행 한끼 음식 종류 등록	
식단 은행 한끼 음식 종류 수정	식단 은행 한끼 음식 종류 수정	식단 은행 한끼 음식 종류 수정	
식단 은행 한끼 음식 종류 삭제	식단 은행 한끼 음식 종류 삭제	식단 은행 한끼 음식 종류 삭제	
식단 은행 한끼 음식 내 식품영양 성분 조회	식단 은행 한끼 음식 내 식품영양 성분 조회	식단 은행 한끼 음식 내 식품영양 성분 조회	
식단 은행 한끼 음식 내 식품영양 성분 등록	식단 은행 한끼 음식 내 식품영양 성분 등록	식단 은행 한끼 음식 내 식품영양 성분 등록	
식단 은행 한끼 음식 내 식품영양 성분 수정	식단 은행 한끼 음식 내 식품영양 성분 수정	식단 은행 한끼 음식 내 식품영양 성분 수정	
식단 은행 한끼 음식 내 식품영양 성분 삭제	식단 은행 한끼 음식 내 식품영양 성분 삭제	식단 은행 한끼 음식 내 식품영양 성분 삭제	

식단 - 개인 계획

식단 계획 날짜별 조회	식단 계획 날짜별 조회	식단 계획 날짜별 조회	
식단 계획 시간별 조회	식단 계획 시간별 조회	식단 계획 시간별 조회	
식단에 등록할 음식 목록 조회	식단에 등록할 음식 목록 조회	식단에 등록할 음식 목록 조회	
식단에 등록할 음식 목록 조건 검색	식단에 등록할 음식 목록 조건 검색	식단에 등록할 음식 목록 조건 검색	
식단 계획 등록		식단 계획 등록	

식단 계획 수정	식단 계획 수정	식단 계획 수정	
식단 계획 삭제	식단 계획 삭제	식단 계획 삭제	
식단 은행 조회	식단 은행 조회	식단 은행 조회	

식단 - 개인 실행

식단 실행 조회	식단 실행 조회	식단 실행 조회	
식단 실행 등록		식단 실행 등록	
식단 실행 수정	식단 실행 수정	식단 실행 수정	
식단 실행 삭제	식단 실행 삭제	식단 실행 삭제	

식단 - 영양 성분 표시

실행한 영양 성분 조회	실행한 영양 성분 조회	실행한 영양 성분 조회	
목표한 영양 성분 조회	목표한 영양 성분 조회	목표한 영양 성분 조회	

관심 목록

관심 운동 클래스 조회		관심 운동 클래스 조회	
관심 운동 클래스 등록		관심 운동 클래스 등록	
관심 운동 클래스 수정		관심 운동 클래스 수정	
관심 운동 클래스 삭제	관심 운동 클래스 삭제	관심 운동 클래스 삭제	

결제 - 수강신청

결제 방식 조회	결제 방식 조회	결제 방식 조회	
결제 방식 등록	결제 방식 등록		
결제 방식 수정	결제 방식 수정		
결제 방식 삭제	결제 방식 삭제		
결제(수강신청) 조회	결제(수강신청) 조회	결제(수강신청) 조회	
결제(수강신청) 등록		결제(수강신청) 등록	
결제(수강신청) 취소		결제(수강신청) 취소	

결제 - 환불

환불 기준 조회	환불 기준 조회	환불 기준 조회	
환불 기준 등록	환불 기준 등록		
환불 기준 수정	환불 기준 수정		
환불 기준 삭제	환불 기준 삭제		
환불 신청 조회	환불 신청 조회	환불 신청 조회	
환불 신청 등록		환불 신청 등록	

환불 신청 취소		환불 신청 취소	
환불 신청 승인	환불 신청 승인		
환불 계좌 조회	환불 계좌 조회	환불 계좌 조회	
환불 계좌 등록		환불 계좌 등록	
환불 계좌 수정		환불 계좌 수정	
환불 계좌 삭제	환불 계좌 삭제	환불 계좌 삭제	
권한별 환불 사유 조회	권한별 환불 사유 조회	권한별 환불 사유 조회	

챌린지 - 일반

챌린지 카테고리 조회	챌린지 카테고리 조회	챌린지 카테고리 조회	
챌린지 카테고리 등록	챌린지 카테고리 등록		
챌린지 카테고리 수정	챌린지 카테고리 수정		
챌린지 카테고리 삭제	챌린지 카테고리 삭제		
챌린지 세팅 옵션 조회	챌린지 세팅 옵션 조회	챌린지 세팅 옵션 조회	
챌린지 세팅 옵션 등록	챌린지 세팅 옵션 등록		
챌린지 세팅 옵션 수정	챌린지 세팅 옵션 수정		
챌린지 세팅 옵션 삭제	챌린지 세팅 옵션 삭제		
챌린지 신고 조회	챌린지 신고 조회		
챌린지 신고 등록		챌린지 신고 등록	
챌린지 포인트 지급	챌린지 포인트 지급		

챌린지 - 등록

챌린지 모집 조회	챌린지 모집 조회	챌린지 모집 조회	
챌린지 모집 등록		챌린지 모집 등록	
챌린지 모집 수정		챌린지 모집 수정	
챌린지 모집 삭제		챌린지 모집 삭제	
챌린지 모집 계획 조회	챌린지 모집 계획 조회	챌린지 모집 계획 조회	
챌린지 모집 계획 등록		챌린지 모집 계획 등록	
챌린지 모집 계획 수정		챌린지 모집 계획 수정	
챌린지 모집 계획 삭제		챌린지 모집 계획 삭제	

챌린지 - 실행

챌린지 참여자 조회	챌린지 참여자 조회	챌린지 참여자 조회	
------------	------------	------------	--

챌린지 참여 신청		챌린지 참여 신청	
챌린지 참여 결제		챌린지 참여 결제	
챌린지 실행 조회	챌린지 실행 조회	챌린지 실행 조회	
챌린지 실행 등록		챌린지 실행 등록	

AI 운동 기본 영상

AI 운동 기본 영상 조회	AI 운동 기본 영상 조회	AI 운동 기본 영상 조회	
AI 운동 기본 영상 등록	AI 운동 기본 영상 등록		
AI 운동 기본 영상 수정	AI 운동 기본 영상 수정		
AI 운동 기본 영상 삭제	AI 운동 기본 영상 삭제		

운동 목표 및 계획

운동 계획 카테고리 조회	운동 계획 카테고리 조회	운동 계획 카테고리 조회	
운동 계획 카테고리 등록	운동 계획 카테고리 등록		
운동 계획 카테고리 수정	운동 계획 카테고리 수정		
운동 계획 카테고리 삭제	운동 계획 카테고리 삭제		
개별 운동 목표 조회	개별 운동 목표 조회	개별 운동 목표 조회	
개별 운동 목표 등록		개별 운동 목표 등록	
개별 운동 목표 수정		개별 운동 목표 수정	
개별 운동 목표 삭제		개별 운동 목표 삭제	
개별 운동 목표 내 계획 조회	개별 운동 목표 내 계획 조회	개별 운동 목표 내 계획 조회	
개별 운동 목표 내 계획 등록		개별 운동 목표 내 계획 등록	
개별 운동 목표 내 계획 수정		개별 운동 목표 내 계획 수정	
개별 운동 목표 내 계획 삭제		개별 운동 목표 내 계획 삭제	

일지

일지 조회	일지 조회	일지 조회	
일지 등록		일지 등록	
일지 수정		일지 수정	
일지 삭제		일지 삭제	
일지 댓글(피드백) 조회	일지 댓글(피드백) 조회	일지 댓글(피드백) 조회	
일지 댓글(피드백) 등록		일지 댓글(피드백) 등록	
일지 댓글(피드백) 수정		일지 댓글(피드백) 수정	

일지 댓글(피드백) 삭제		일지 댓글(피드백) 삭제	
일지 댓글(피드백) 답변 조회	일지 댓글(피드백) 답변 조회	일지 댓글(피드백) 답변 조회	
일지 댓글(피드백) 답변 등록		일지 댓글(피드백) 답변 등록	
일지 댓글(피드백) 답변 수정		일지 댓글(피드백) 답변 수정	
일지 댓글(피드백) 답변 삭제		일지 댓글(피드백) 답변 삭제	
일지 좋아요 수 조회	일지 좋아요 수 조회	일지 좋아요 수 조회	
일지 좋아요 등록		일지 좋아요 등록	
일지 좋아요 취소		일지 좋아요 취소	

포인트

포인트 지급기준 조회	포인트 지급기준 조회		
포인트 지급기준 등록	포인트 지급기준 등록		
포인트 지급기준 수정	포인트 지급기준 수정		
포인트 지급기준 삭제	포인트 지급기준 삭제		
포인트 발생 조회	포인트 발생 조회	포인트 발생 조회	
포인트 발생 등록	포인트 발생 등록		
포인트 발생 수정	포인트 발생 수정		
포인트 발생 삭제	포인트 발생 삭제		
포인트 내역 조회	포인트 내역 조회	포인트 내역 조회	
포인트 내역 검색	포인트 내역 검색	포인트 내역 검색	

사이트 내 신고

신고 기준 조회	신고 기준 조회	신고 기준 조회	신고 기준 조회
신고 기준 등록	신고 기준 등록		
신고 기준 수정	신고 기준 수정		
신고 기준 삭제	신고 기준 삭제		
신고 신청 조회	신고 신청 조회	신고 신청 조회	
신고 신청 등록	신고 신청 등록	신고 신청 등록	
신고 신청 수정	신고 신청 수정	신고 신청 수정	
신고 신청 삭제	신고 신청 삭제	신고 신청 삭제	

2.2 상세 기능 정의

1) 회원 관리

No.	요구사항명	요구사항 상세 (처리 process)
1.1	회원 가입	회원 가입 가능
1.2	회원 정보 수정	관리자 또는 본인의 경우 수정 가능
1.3	회원 탈퇴	회원 탈퇴 가능
1.4	비밀번호 찾기	비밀번호 찾기 가능
1.5	전체 회원 조회	관리자의 경우 가능
1.6	권한별 조회	권한별로 조회할 수 있는 정보 조회 가능
1.7	회원 개인 추가 정보 등록	회원 개인 추가 정보 등록 가능
1.8	회원 개인 추가 정보 수정	회원 개인 추가 정보 수정 가능
1.9	회원 개인 추가 정보 삭제	회원 개인 추가 정보 삭제 가능
1.10	회원 정보 조회	관리자 또는 본인의 경우 회원 정보 조회 가능
1.11	회원 결제 조회	관리자 또는 본인의 경우 회원 결제 정보 조회 가능
1.12	회원 환불 관리	관리자 또는 본인의 경우 회원 환불 정보 조회 가능

2) 로그인 관리

No.	요구사항명	요구사항 상세 (처리 process)
2.1	권한별 로그인	권한별 로그인 가능
2.2	로그아웃	로그아웃 가능

3) 운동 클래스 카테고리 관리

No.	요구사항명	요구사항 상세 (처리 process)
3.1	운동 클래스 카테고리 조회	운동 클래스 카테고리 조회 가능
3.2	운동 클래스 카테고리 대 등록	관리자의 경우 운동 클래스 카테고리 대 등록 가능
3.3	운동 클래스 카테고리 대 수정	관리자의 경우 운동 클래스 카테고리 대 수정 가능
3.4	운동 클래스 카테고리 대 삭제	관리자의 경우 운동 클래스 카테고리 대 삭제 가능
3.5	운동 클래스 카테고리 중 등록	관리자의 경우 운동 클래스 카테고리 중 등록 가능
3.6	운동 클래스 카테고리 중 수정	관리자의 경우 운동 클래스 카테고리 중 수정 가능
3.7	운동 클래스 카테고리 중 삭제	관리자의 경우 운동 클래스 카테고리 중 삭제 가능
3.8	운동 클래스 카테고리 소 등록	관리자의 경우 운동 클래스 카테고리 소 등록 가능
3.9	운동 클래스 카테고리 소 수정	관리자의 경우 운동 클래스 카테고리 소 수정 가능
3.10	운동 클래스 카테고리 소 삭제	관리자의 경우 운동 클래스 카테고리 소 삭제 가능

4) 신고 관리

No.	요구사항명	요구사항 상세 (처리 process)
4.1	신고 기준 조회	신고 기준 조회 가능
4.2	신고 기준 등록	관리자의 경우 신고 기준 등록 가능
4.3	신고 기준 수정	관리자의 경우 신고 기준 수정 가능
4.4	신고 기준 삭제	관리자의 경우 신고 기준 삭제 가능
4.5	신고 신청 조회	관리자일 경우 전체 신고 목록 조회 본인의 경우 본인의 신청 내역 조회 가능
4.6	신고 신청 등록	신고 신청 등록 가능
4.7	신고 신청 수정	신고 신청 수정 가능
4.8	신고 신청 삭제	신고 신청 삭제 가능
4.9	신고 사유 리스트 등록	관리자일 경우 신고 사유 등록 가능
4.10	신고 사유 리스트 수정	관리자일 경우 신고 사유 수정 가능
4.11	신고 사유 리스트 삭제	관리자일 경우 신고 사유 삭제 가능
4.12	신고 사유 리스트 조회	관리자일 경우 신고 사유 조회 가능 (신고 사유명, 정지일수, 사용여부 출력)
4.13	회원 신고 조회	관리자일 경우 전체 신고 목록 조회 본인의 경우 본인의 신청 내역 조회 가능
4.14	신고 처리	관리자일 경우 신청 처리 가능

5) 클래스조회/검색

No.	요구사항명	요구사항 상세 (처리 Process)
5.1	승인된 운동 클래스 조회	모든 사용자가 볼 수 있는 승인된 클래스 리스트
5.2	승인된 운동 클래스 조건별 검색	모든 사용자가 조건별 검색 할 수 있는 승인된 클래스 리스트

6) 클래스 관리

No.	요구사항명	요구사항 상세 (처리 process)
6.1	클래스 소개 조회	클래스 상세내용의 클래스 소개란
6.2	클래스 소개 등록	클래스 상세내용의 클래스 소개등록
6.3	클래스 소개 수정	클래스 상세내용의 클래스 소개수정
6.4	클래스 소개 삭제	클래스 상세내용의 클래스 소개삭제
6.5	공지사항 등록	클래스 상세내용의 클래스 공지사항등록
6.6	공지사항 수정	클래스 상세내용의 클래스 공지사항수정
6.7	공지사항 삭제	클래스 상세내용의 클래스 공지사항삭제
6.8	공지사항 조회	클래스 상세내용의 클래스 공지사항란
6.9	운동 클래스 Q&A 질문 조회	클래스 상세내용의 클래스 질문란
6.10	운동 클래스 Q&A 질문 등록	클래스 상세내용의 클래스 질문등록
6.11	운동 클래스 Q&A 질문 수정	클래스 상세내용의 클래스 질문수정
6.12	운동 클래스 Q&A 질문 삭제	클래스 상세내용의 클래스 질문삭제
6.13	운동 클래스 Q&A 답변 조회	클래스 상세내용의 클래스 답변란
6.14	운동 클래스 Q&A 답변 등록	클래스 상세내용의 클래스 답변등록

6.15	운동 클래스 Q&A 답변 수정	클래스 상세내용의 클래스 답변수정
6.16	운동 클래스 Q&A 답변 삭제	클래스 상세내용의 클래스 답변삭제
6.17	수강평 조회	클래스 상세내용의 클래스 수강평란
6.18	수강평 등록	클래스 상세내용의 클래스 수강평등록
6.19	수강평 수정	클래스 상세내용의 클래스 수강평수정
6.20	수강평 삭제	클래스 상세내용의 클래스 수강평삭제
6.21	수강평 댓글 조회	클래스 상세내용의 클래스 수강평댓글란
6.22	수강평 댓글 등록	클래스 상세내용의 클래스 수강평댓글등록
6.23	수강평 댓글 수정	클래스 상세내용의 클래스 수강평댓글수정
6.24	수강평 댓글 삭제	클래스 상세내용의 클래스 수강평댓글삭제
6.25	운동 클래스 식단 조회	클래스 상세내용의 클래스 식단조회
6.26	수강생 조회(결제 완료된 수강생 조회)	클래스를 수강하는 사용자 조회
6.27	수강생 조건별 검색	클래스를 수강하는 사용자 조건별조회

7) 클래스 섹션/커리큘럼 관리

No.	요구사항명	요구사항 상세 (처리 process)
7.1	운동 클래스 내 섹션 조회	클래스 상세내용의 섹션란
7.2	운동 클래스 내 섹션 등록	클래스 상세내용의 섹션등록
7.3	운동 클래스 내 섹션 수정	클래스 상세내용의 섹션수정
7.4	운동 클래스 내 섹션 삭제	클래스 상세내용의 섹션삭제
7.5	운동 클래스의 섹션 내 커리큘럼 등록	클래스 상세내용의 섹션 내 커리큘럼 등록
7.6	운동 클래스의 섹션 내 커리큘럼 수정	클래스 상세내용의 섹션 내 커리큘럼 수정
7.7	운동 클래스의 섹션 내 커리큘럼 삭제	클래스 상세내용의 섹션 내 커리큘럼 삭제
7.8	운동 클래스의 섹션 내 커리큘럼 조회	클래스 상세내용의 섹션 내 커리큘럼란

8) 클래스 과제 관리

No.	요구사항명	요구사항 상세 (처리 process)
8.1	운동 클래스 내 커리큘럼 내 과제 조회	클래스를 개설한 사용자가 수강하는 사용자에게 주는 과제
8.2	운동 클래스 내 커리큘럼 내 과제 등록	클래스를 개설한 사용자가 수강하는 사용자에게 주는 과제등록
8.3	운동 클래스 내 커리큘럼 내 과제 수정	클래스를 개설한 사용자가 수강하는 사용자에게 주는 과제수정
8.4	운동 클래스 내 커리큘럼 내 과제 삭제	클래스를 개설한 사용자가 수강하는 사용자에게 주는 과제삭제
8.5	운동 클래스 내 커리큘럼 내 과제 수행 조회	클래스를 개설한 사용자가 수강하는 사용자에게 주는 과제 수행
8.6	운동 클래스 내 커리큘럼 내 과제 수행 등록	클래스를 개설한 사용자가 수강하는 사용자에게 주는 과제 수행등록
8.7	운동 클래스 내 커리큘럼 내	클래스를 개설한 사용자가 수강하는 사용자에게 주는 과제 수행수정

8.8	과제 수행 수정 운동 클래스 내 커리큘럼 내 과제 수행 삭제	클래스를 개설한 사용자가 수강하는 사용자에게 주는 과제 수행삭제
-----	---	-------------------------------------

9) 챌린지

No.	요구사항명	요구사항 상세 (처리 process)
9.1	챌린지 카테고리 등록	관리자 챌린지 카테고리 등록 가능
9.2	챌린지 카테고리 수정	관리자 수정 가능
9.3	챌린지 카테고리 삭제	관리자 삭제 가능
9.4	챌린지 카테고리 조회	전체 카테고리 조회
9.5	챌린지 세팅 등록	관리자 등록 가능
9.6	챌린지 세팅 수정	관리자 수정 가능
9.7	챌린지 세팅 삭제	관리자 삭제 가능
9.8	챌린지 세팅 조회	전체 세팅 목록 조회
9.9	모집 챌린지 조회	관리자 or 일반회원일 경우 조회 가능
9.10	모집 챌린지 상세 조회	관리자 or 일반회원일 경우 해당 챌린지의 상세 정보 조회 가능
9.11	진행 챌린지 조회	관리자 경우 조회 가능
9.12	진행 챌린지 상세 조회	관리자 경우 해당 챌린지 상세 정보 조회 가능
9.13	챌린지 포인트 관리	관리자일 경우 포인트 내역 조회 가능
9.14	챌린지 포인트 상세 조회	관리자일 경우 해당 챌린지의 지급예정 포인트, 진행률 조회 가능
9.15	챌린지 모집 목록 조회	관리자 or 일반회원일 경우 조회 가능
9.16	챌린지 모집 등록	일반회원일 경우 조회 가능
9.17	챌린지 모집 상세 조회	일반 회원일 경우 해당 챌린지의 상세 정보 조회 가능
9.18	참가 챌린지 조회	참가 챌린지가 있는 경우 목록 조회 가능
9.19	참가 챌린지 상세 조회	참가 챌린지의 상세 정보 조회 가능(참여회원과 진행률도 조회 가능)
9.20	참가 챌린지 인증 등록	참가 챌린지에 대한 인증을 등록 가능

10) AI 운동 기본 영상

No.	요구사항명	요구사항 상세 (처리 process)
10.1	AI 기본 영상 등록	관리자일 경우 영상 등록
10.2	AI 기본 영상 수정	관리자일 경우 영상 제목 및 영상 수정 가능
10.3	AI 기본 영상 삭제	관리자일 경우 영상 삭제 가능
10.4	AI 기본 영상 조회	관리자 또는 일반 회원의 경우 가능

11) 운동 목표 및 계획

No.	요구사항명	요구사항 상세 (처리 process)
11.1	운동 계획 카테고리 등록	관리자일 경우 운동 계획 카테고리명 등록
11.2	운동 계획 카테고리 수정	관리자일 경우 운동 계획 카테고리명 수정 가능
11.3	운동 계획 카테고리 삭제	관리자일 경우 운동 계획 카테고리 삭제 가능
11.4	운동 계획 카테고리 조회	관리자 또는 일반 회원의 경우 가능
11.5	개별 운동 목표 등록	일반 회원일 경우 목표 내용, 수강 클래스, 목표 기간 등록
11.6	개별 운동 목표 수정	일반 회원일 경우 목표 내용, 수강 클래스, 목표 기간 수정 가능
11.7	개별 운동 목표 삭제	일반 회원일 경우 삭제 가능

11.8	개별 운동 목표 조회	관리자 또는 일반 회원의 경우 가능
11.9	개별 운동 목표 내 계획 등록	일반 회원일 경우 개인 목표에 대한 계획 및 과제, 챌린지 계획 등록
11.10	개별 운동 목표 내 계획 수정	일반 회원일 경우 개인 목표에 대한 계획 및 과제, 챌린지 계획 수정 가능
11.11	개별 운동 목표 내 계획 삭제	일반 회원일 경우 삭제 가능
11.12	개별 운동 목표 내 계획 조회	관리자 또는 일반 회원의 경우 가능

12) 일지

No.	요구사항명	요구사항 상세 (처리 process)
12.1	일지 등록	일반 회원일 경우 제목, 공개 범위, 운동 클래스 카테고리, 인증사진, 내용 등록
12.2	일지 수정	일반 회원일 경우 제목, 공개 범위, 운동 클래스 카테고리, 인증사진, 내용 수정 가능
12.3	일지 삭제	일반 회원일 경우 삭제 가능
12.4	일지 조회	관리자 또는 일반 회원일 경우 가능
12.5	일지 댓글(피드백) 등록	일반 회원(클래스 개설한 회원)만 등록
12.6	일지 댓글(피드백) 수정	일반 회원(클래스 개설한 회원)만 수정 가능
12.7	일지 댓글(피드백) 삭제	일반 회원(클래스 개설한 회원)만 삭제 가능
12.8	일지 댓글(피드백) 조회	일반 회원(클래스 개설한 회원 또는 해당하는 클래스 수강생) 가능
12.9	일지 댓글(피드백) 답변 등록	일반 회원(클래스 수강생) 등록
12.10	일지 댓글(피드백) 답변 수정	일반 회원(클래스 수강생) 수정 가능
12.11	일지 댓글(피드백) 답변 삭제	일반 회원(클래스 수강생) 삭제 가능
12.12	일지 댓글(피드백) 답변 조회	일반 회원(클래스 개설한 회원 또는 해당하는 클래스 수강생) 가능
12.13	일지 좋아요 등록	관리자 또는 일반 회원 등록
12.14	일지 좋아요 수 조회	관리자 또는 일반 회원 가능
12.15	일지 좋아요 취소	관리자 또는 일반 회원 취소 가능

13) 식단은행

No.	요구사항명	요구사항 상세 (처리 process)
13.1	식단은행 등록	관리자의 식단 은행 등록
13.2	식단은행 수정	관리자의 식단 은행 수정
13.3	식단은행 삭제	관리자의 식단 은행 삭제
13.4	식단은행 조회	관리자의 식단 은행 조회
13.5	식단은행 상세 등록	관리자의 식단 은행 상세 등록
13.6	식단은행 상세 수정	관리자의 식단 은행 상세 등록
13.7	식단은행 상세 삭제	관리자의 식단 은행 상세 등록
13.8	식단은행 상세 조회	관리자의 식단 은행 상세 등록

14) 수강 신청

No.	요구사항명	요구사항 상세 (처리 process)
1.1	관심 목록 등록	운동클래스 관심 목록 등록 가능
1.2	수강 신청	운동클래스 수강 신청 가능
1.3	수강 신청 조회	수강 신청 내역 조회 가능
1.4	수강 운동클래스 조회	수강중인 운동클래스 조회 가능

1.5	수강생 조회	운동클래스 개설자의 경우 수강중인 회원 조회 가능
-----	--------	-----------------------------

15) 결제

No.	요구사항명	요구사항 상세 (처리 process)
2.1	운동클래스 수강 결제	운동클래스 수강 신청 관련 결제 가능
2.2	챌린지 참여 결제	챌린지 참여 관련 결제 가능
2.3	회원 결제 내역 조회	회원별 결제 내역 조회 가능
2.4	결제 내역 조회	관리자의 경우 사이트 내 결제 내역 조회 기능
2.5	결제 상세 내역 조회	결제 별 상세 내역 조회 가능

16) 환불

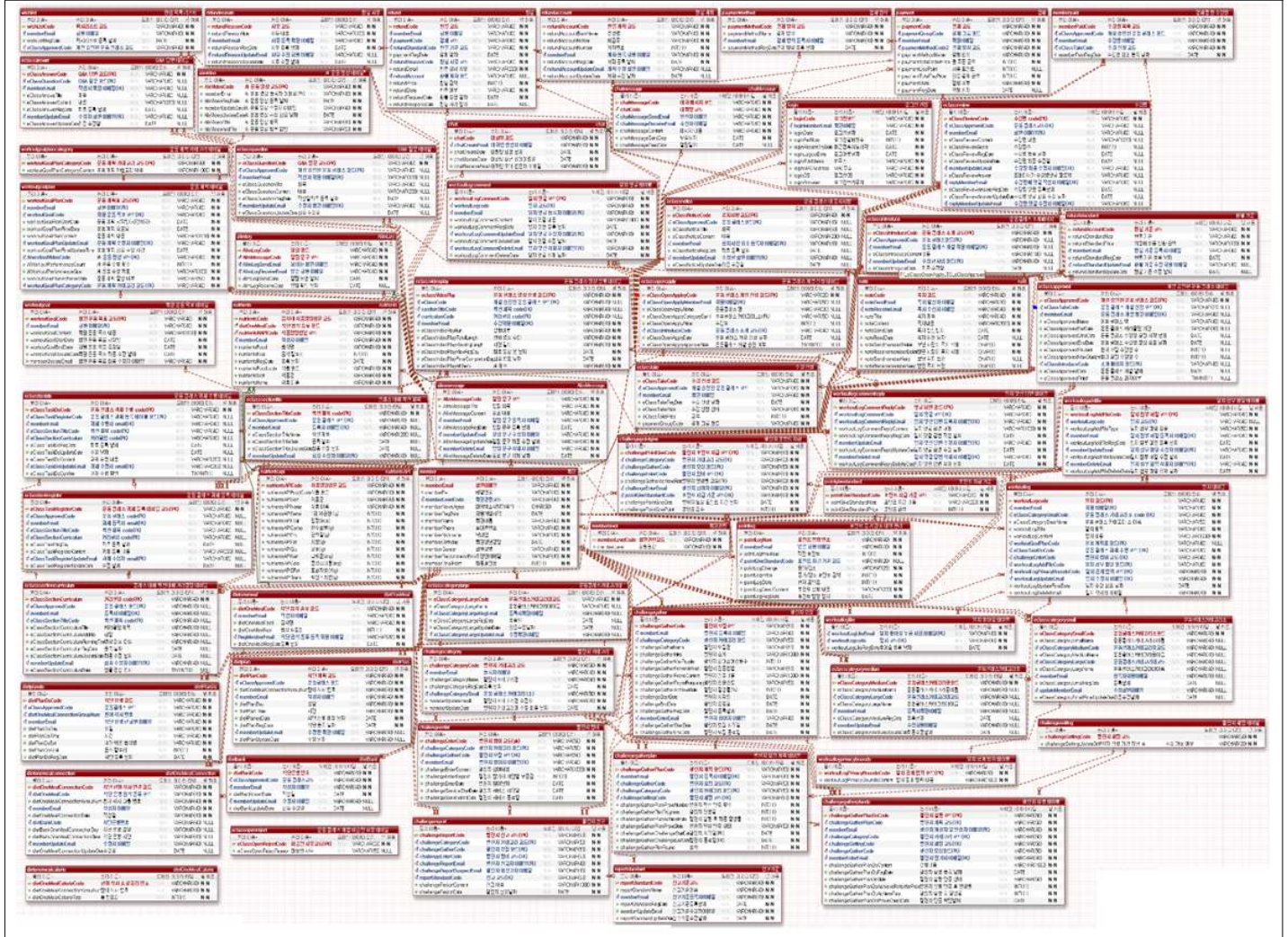
No.	요구사항명	요구사항 상세 (처리 process)
1.1	결제 환불 승인	환불 기준에 따른 환불 승인 처리 가능
1.2	환불 신청	회원 결제별 환불 신청 가능
1.3	환불 계좌 조회	회원 환불 신청별 환불 계좌 조회 가능
1.4	환불 계좌 등록	회원 환불 신청별 환불 계좌 등록 가능
1.5	환불 계좌 수정	회원 환불 신청별 환불 계좌 수정 가능
1.6	환불 기준 조회	관리자의 경우 환불 기준 조회 가능
1.7	환불 기준 등록	관리자의 경우 환불 기준 등록 가능
1.8	환불 기준 수정	관리자의 경우 환불 기준 수정 가능
1.9	환불 기준 삭제	관리자의 경우 환불 기준 삭제 가능

3. 업무분담(담당 업무)

이름	담당 업무	업무 내용
최보라	팀장	회의 진행, 일정 관리, 업무 조율
	Hosting	URL 연결 작업, 배포 업무 수행
	시스템구조도 작성	프로세스 흐름도를 도표로 정리
	테이블 정의서 작성	각 테이블의 컬럼 정보 정리
	챌린지 관리	챌린지 CRUD 및 화면 구현
	메인 화면	메인 화면 구현
이민식	팀원	회의 진행, 일정 관리, 업무 조율
	Hosting	프로젝트 호스팅 계정관리, URL 연결 작업, 배포 업무 수행
	기능정의서 작성	권한에 따른 기능을 정리
	테이블 정의서 작성	각 테이블의 컬럼 정보 정리
	회원 관리	회원 CRUD 및 화면 구현
	운동 클래스 카테고리 관리	운동 클래스 카테고리 CRUD 및 화면 구현
	신고 관리	신고, 신고 기준 CRUD 및 화면 구현
김재환	운동 클래스 관리	운동 클래스의 등록 신청 및 수정, 삭제
	운동 클래스 과제	운동 클래스 수강시 사용자에게 부여할 수 있는 과제 등록 유무
	운동 클래스 과제 수행	과제 부여받은 사용자의 수행 등록 및 수정, 삭제
	클래스 승인 및 미승인	클래스 등록 신청 리스트 관리 및 승인, 미승인(사유) 여부 미승인 사유 관리페이지
진수경	팀원	회의 진행, 일정 관리, 업무 조율
	Hosting	프로젝트 호스팅 계정관리, URL 연결 작업, 배포 업무 수행
	기능정의서 작성	권한에 따른 기능을 정리
	테이블 정의서 작성	각 테이블의 컬럼 정보 정리
	수강신청	수강신청 CRUD 처리 및 화면구현
	결제	결제 CRUD 처리 및 화면구현
	환불	환불 CRUD 처리 및 화면구현
	포인트	포인트 지급, 포인트 지급 기준 CRUD 처리 및 화면 구현
최민희	Hosting	프로젝트 호스팅 계정관리, URL 연결 작업, 배포 업무 수행
	DB(객체)목록 작성	사용할 논리, 물리적 DB 정리
	팀 프로젝트 문서 관리	관리 및 문서 작성
	일지 관리	일지 CRUD 및 화면구현
이영호	팀원	회의 진행, 일정 관리, 업무 조율
	개별 식단 기능	개별 식단을 계획하고 관리한다.
	식단 은행 기능	여러 사람이 쓸 수 있는 식단 은행을 생성하고 관리한다.
	식단 영양성분 관리기능	식단 영양 성분을 표시해주고, 관리한다.

4. 설계사양서

4.1 ERD



4.2 테이블 구조

회원(member)

회원 이메일 (PK)	memberEmail	VARCHAR(40)	NOT NULL
비밀번호	memberPw	VARCHAR(40)	NOT NULL
권한 코드 (FK)	memberLevelCode	VARCHAR(40)	NOT NULL
마케팅 소식 받기 동의	memberNewsAgree	VARCHAR(40)	NOT NULL
회원 가입 시각	memberRegDate	DATE	NOT NULL
회원 이름	memberName	VARCHAR(40)	NOT NULL
휴대폰 번호	memberPhone	VARCHAR(40)	NOT NULL
닉네임	memberNickname	VARCHAR(40)	NOT NULL

회원 생년월일	memberBirthday	DATE	NOT NULL
회원 성별	memberGender	VARCHAR(40)	NOT NULL
추천인 이메일	memberRecommendEmail	VARCHAR(40)	NOT NULL
최종 포인트	memberFinalPoint	int(11)	NOT NULL
회원 권한(memberLevel)			
회원 권한 코드 (PK)	memberLevelCode	VARCHAR(40)	NOT NULL
회원 권한 이름	memberLevelName	VARCHAR(40)	NOT NULL
회원 권한 등록자 이메일(FK)	memberLevelRegEmail	VARCHAR(40)	NOT NULL
회원 권한 등록 날짜	memberLevelRegDate	DATE	NOT NULL
로그인(login)			
로그인 코드(PK)	loginCode	VARCHAR(40)	NOT NULL
회원 이메일(FK)	loginMemberEmail	VARCHAR(40)	NOT NULL
로그인 날짜	loginDate	DATE	NOT NULL
로그인 실패 횟수	loginFailNum	int(11)	NOT NULL
최근 접속 시도 시각	loginRecentTryDate	DATE	NOT NULL
로그아웃 날짜	loginLogoutDate	DATE	NOT NULL
IP주소	loginIPAddress	VARCHAR(40)	NOT NULL
MAC주소	loginMACAddress	VARCHAR(40)	NOT NULL
로그인 OS	loginOS	VARCHAR(40)	NOT NULL
로그인 브라우저	loginBrowser	VARCHAR(40)	NOT NULL
운동 클래스 카테고리 대(eClassCategoryLarge)			
운동 클래스 카테고리 대 코드 (PK)	eClassCategoryLargeCode	VARCHAR(40)	NOT NULL
운동 클래스 카테고리 대 이름	eClassCategoryLargeName	VARCHAR(40)	NOT NULL
등록자 회원 이메일(FK)	eClassCategoryLargeReg	VARCHAR(40)	NOT NULL

	Email		
등록 날짜	eClassCategoryLargeRegDate	DATE	NOT NULL
최종 수정 날짜	eClassCategoryLargeUpdateDate	DATE	NULL
수정 회원 이메일 (FK)	eClassCategoryLargeUpdateEmail	VARCHAR(40)	NULL
운동 클래스 카테고리 중(eClassCategoryMedium)			
운동 클래스 카테고리 중 코드 (PK)	eClassCategoryMediumCode	VARCHAR(40)	NOT NULL
운동 클래스 카테고리 중 이름	eClassCategoryMediumName	VARCHAR(40)	NOT NULL
운동 클래스 카테고리 대 코드 (FK)	eClassCategoryLargeCode	VARCHAR(40)	NOT NULL
운동 클래스 카테고리 대 이름	eClassCategoryLargeName	VARCHAR(40)	NOT NULL
등록자 회원 이메일 (FK)	memberEmail	VARCHAR(40)	NOT NULL
등록 날짜	eClassCategoryMediumRegDate	DATE	NOT NULL
수정 회원 이메일 (FK)	memberUpdateEmail	VARCHAR(40)	NULL
최종 수정 날짜	eClassCategoryMediumUpdateDate	DATE	NULL
운동 클래스 카테고리 소(eClassCategorySmall)			
운동 클래스 카테고리 소 코드 (PK)	eClassCategorySmallCode	VARCHAR(40)	NOT NULL
운동 클래스 카테고리 소 이름	eClassCategorySmallName	VARCHAR(40)	NOT NULL
운동 클래스 카테고리 중 코드 (FK)	eClassCategoryMediumCode	VARCHAR(40)	NOT NULL
운동 클래스 카테고리 중 이름	eClassCategoryMediumName	VARCHAR(40)	NOT NULL
운동 클래스 카테고리 대 코드 (FK)	eClassCategoryLargeCode	DATE	NOT NULL
운동 클래스 카테고리 대 이름	eClassCategoryLargeName	VARCHAR(40)	NOT NULL
등록자 회원 이메일 (FK)	memberEmail	VARCHAR(40)	NOT NULL
등록 날짜	eClassCategoryMediumRegDate	DATE	NOT NULL
수정 회원 이메일 (FK)	memberUpdateEmail	VARCHAR(40)	NULL
최종 수정 날짜	eClassCategoryMediumUpdateDate	DATE	NULL
신고 기준(reportStandard)			

신고 기준 코드(PK)	reportStandardCode	VARCHAR(40)	NOT NULL
신고 기준 이름	reportStandardName	VARCHAR(40)	NOT NULL
신고 기준 등록자 이메일(FK)	memberEmail(FK)	VARCHAR(40)	NOT NULL
신고 기준 등록 날짜	reportStandardRegDate	DATE	NOT NULL
신고기준수정자이메일	memberUpdateEmail	VARCHAR(40)	NOT NULL
신고기준수정날짜	reportStandardUpdateDate	DATE	NOT NULL
신고(report)			
신고 코드	reportCode	VARCHAR(40)	NOT NULL
신고자 이메일	reportMemberEmail	VARCHAR(40)	NOT NULL
신고자 이름	reportMemberName	VARCHAR(40)	NOT NULL
신고 대상 종류	reportTargetType	VARCHAR(40)	NOT NULL
신고 대상	reportTarget	VARCHAR(40)	NOT NULL
신고 기준 코드	reportStandardCode	VARCHAR(40)	NOT NULL
신고 사유	reportReason	VARCHAR(40)	NOT NULL
신고자 IP	reportIP	VARCHAR(40)	NOT NULL
신고 등록 날짜	reportRegDate	DATE	NOT NULL
신고 처리 상태	reportProcessState	VARCHAR(40)	NOT NULL
신고 처리 최초 등록자 이메일	reportProcessRegMemberEmail	VARCHAR(40)	NULL
신고 처리 최초 등록 날짜	reportProcessDate	DATE	NULL
신고 처리 수정자 이메일	reportProcessUpdateMemberEmail	VARCHAR(40)	NULL
신고 처리 수정 날짜	reportProcessUpdateDate	DATE	NULL
승인된 클래스 테이블(eclassapproved)			
운동 클래스 코드(PK)	eClassApprovedCode	VARCHAR(40)	NOT NULL

운동 클래스 개설 회원 이메일(FK)	memberEmail	VARCHAR(40)	NULL
운동 클래스 카테고리(FK)	eClassCategorySmallCode	VARCHAR(40)	NULL
운동클래스 소개코드(FK)	eClassIntroduceCode	VARCHAR(40)	NULL
운동클래스 섹션코드(FK)	eClassSectionTitleCode	VARCHAR(40)	NULL
운동클래스 커리큘럼코드(FK)	eClassSectionCurriculumCode	VARCHAR(40)	NULL
운동클래스 질문코드(FK)	eClassQuestionCode	VARCHAR(40)	NULL
운동클래스 답변코드(FK)	eClassAnswerCode	VARCHAR(40)	NULL
운동 클래스 가격	eClassApprovedPrice	INT	NULL
운동 클래스 명	eClassApprovedName	VARCHAR(40)	NULL
운동 클래스 커리큘럼 기간	eClassApprovedSetDate	INT	NULL
운동 클래스 수강생 모집 시작 날짜	eClassApprovedStratDate	DATE	NULL
운동 클래스 수강생 모집 종료 날짜	eClassApprovedEndDate	DATE	NULL
현재 모집 수강생 수	eClassApprovedStudent	INT	NULL
최대 모집 수강생 수	eClassApprovedMaxStudent	INT	NULL
운동 클래스 개설 날짜	eClassApprovedDate	DATE	NULL
클래스의 상태 여부	eClassApprovedState	TINYINT	NULL

클래스 소개(eclassintroduce)

운동 클래스 소개 코드(PK)	eClassIntroduceCode	VARCHAR(40)	NOT NULL
운동 클래스 개설 회원 이메일(FK)	memberEmail	VARCHAR(40)	NULL
내용	eClassIntroduceContent	VARCHAR(100)	NULL
수정한 사람 코드(FK)	memberUpdateEmail	VARCHAR(40)	NULL
최종 수정일	eClassIntroducetDate	DATE	NULL
운동클래스 소개 이미지	eClassIntroducelmng	VARCHAR(40)	NULL

클래스 섹션(eclassectiontitle)

섹션 제목 code(PK)	eClassSectionTitleCode	VARCHAR(40)	NOT NULL
등록자 이메일 (FK)	memberEmail	VARCHAR(40)	NULL
섹션 제목	eClassSectionTitleName	VARCHAR(40)	NULL
등록 날짜	eClassSectionTitleDate	DATE	NULL
최종 수정 날짜	eClassSectionTitleUpdateDate	DATE	NULL
최종 수정자 이메일(FK)	memberUpdateEmail	VARCHAR(40)	NULL

클래스 커리큘럼(eclassectioncurriculum)

커리큘럼 code(PK)	eClassSectionCurriculumCode	VARCHAR(40)	NOT NULL
등록자 이메일(FK)	memberEmail	VARCHAR(40)	NULL
섹션 제목 code(FK)	eClassSectionTitleCode	VARCHAR(40)	NULL
커리큘럼 제목	eClassSectionCurriculumTitle	VARCHAR(40)	NULL
파일	eClassSectionCurriculumAddFile	VARCHAR(40)	NULL
영상 러닝 타임	eClassSectionCurriculumRunningTime	VARCHAR(40)	NULL
등록 날짜	eClassSectionCurriculumRegDate	DATE	NULL
최종 수정 날짜	eClassSectionCurriculumUpdateDate	DATE	NULL
최종 수정자 이메일(FK)	memberUpdateEmail	INT	NULL
샘플 영상 표시	eClassSectionCurriculumState	TINYINT	NULL

클래스 Q&A 질문(eclassquestion)

Q&A 질문 코드(PK)	eClassQuestionCode	VARCHAR(40)	NOT NULL
작성자 회원 이메일(FK)	memberEmail	VARCHAR(40)	NULL
제목	eClassQuestionTitle	VARCHAR(40)	NULL
작성일최초 등록 날짜	eClassQuestionRegDate	DATE	NULL
수정자 회원 이메일(FK)	memberUpdateEmail	VARCHAR(40)	NULL
최종 수정일	eClassQuestionUpdatedDate	DATE	NULL

	ate		
클래스 Q&A 답변(eclassanswer)			
Q&A 답변 코드(PK)	eClassAnswerCode	VARCHAR(40)	NOT NULL
Q&A 질문 코드(FK)	eClassQuestionCode	VARCHAR(40)	NULL
작성자 회원 이메일(FK)	memberEmail	VARCHAR(40)	NULL
제목	eClassAnswTitle	VARCHAR(40)	NULL
최초 등록 날짜	eClassAnswerRegDate	DATE	NULL
수정자 회원 이메일(FK)	memberUpdateEmail	VARCHAR(40)	NULL
최종 수정일	eClassAnswerUpdateDate	DATE	NULL
클래스 과제 등록(eclasstaskregister)			
운동 클래스 과제 등록 테이블 코드(PK)	eClassTaskRegisterCode	VARCHAR(40)	NOT NULL
운동 클래스 code(FK)	eClassApprovedCode	VARCHAR(40)	NULL
과제 등록자 email(FK)	memberEmail	VARCHAR(40)	NULL
섹션 제목 code(FK)	eClassSectionTitleCode	VARCHAR(40)	NULL
커리큘럼 code(FK)	eClassSectionCurriculum	VARCHAR(40)	NULL
최초 등록 날짜	eClassSectionCurriculum	DATE	NULL
과제 등록 내용	eClassTaskRegisterContent	VARCHAR(40)	NULL
과제 수정자 email(FK)	eClassTaskRegisterUpdateEmail	VARCHAR(40)	NULL
수정 날짜	eClassTaskRegisterUpdateDate	DATE	NULL
클래스 과제 수행(eclasstaskdo)			
운동 클래스 과제 수행 code(PK)	eClassTaskDoCode	VARCHAR(40)	NOT NULL
운동 클래스 과제 등록 테이블 코드(FK)	eClassTaskRegisterCode	VARCHAR(40)	NULL
과제 수행자 email(FK)	memberEmail	VARCHAR(40)	NULL
섹션 제목 code(FK)	eClassSectionTitleCode	VARCHAR(40)	NULL
커리큘럼 code(FK)	eClassSectionCurriculum	VARCHAR(40)	NULL

최초 등록 날짜	eClassTaskDoRegDate	DATE	NULL
수정 날짜	eClassTaskDoUpdateDate	DATE	NULL
과제 수행 내용	eClassTaskDoContent	VARCHAR(40)	NULL
과제 수정자 email(FK)	eClassTaskDoUpdateEmail	VARCHAR(40)	NULL
과제 수행 확인	eClassTaskDoConfirm	TINYINT	NULL
클래스 미승인 사유(eclassopenreject)			
미승인 사유코드(PK)	eClassOpenRejectCode	VARCHAR(40)	NOT NULL
미승인 사유	eClassOpenRejectReason	VARCHAR(40)	NULL
클래스 공지사항(eclassnotice)			
공지사항 코드(PK)	eClassNoticeCode	VARCHAR(40)	NOT NULL
운동 클래스코드(FK)	eClassApprovedCode	VARCHAR(40)	NULL
제목	eClassNoticeTitle	VARCHAR(40)	NULL
내용	eClassNoticeContent	VARCHAR(40)	NULL
공지사항 최초 등록자 이메일(FK)	memberEmail	VARCHAR(40)	NULL
최초 등록 날짜	eClassNoticeRegDate	DATE	NULL
수정한 회원 이메일(FK)	memberUpdateEmail	VARCHAR(40)	NULL
최종 수정일	eClassNoticeUpdateDate	DATE	NULL
클래스 수강평(eclassreview)			
수강평 code(PK)	eClassReviewCode	VARCHAR(40)	NOT NULL
운동 클래스코드(FK)	eClassApprovedCode	VARCHAR(40)	NULL
회원 이메일(FK)	memberEmail	VARCHAR(40)	NULL
수강평 내용	eClassReviewContent	VARCHAR(40)	NULL
수강점수	eClassReviewScore	VARCHAR(40)	NULL
수강평 등록 날짜	eClassReviewRegDate	VARCHAR(40)	NULL

수강평 최종 수정일	eClassReviewUpdateDate	VARCHAR(40)	NULL
수강평 최종 수정자 이메일(FK)	memberUpdateEmail	VARCHAR(40)	NULL
트레이너가 수강평댓글 피드백	eClassReviewAnswer	INT	NULL
수강평에 댓글 작성자 이메일(FK)	replyMemberEmail	VARCHAR(40)	NULL
수강평 댓글 등록날짜	eClassReviewAnswerRegDate	INT	NULL
수강평 댓글 최종 수정 날짜	eClassReviewAnswerUpdateDate	DATE	NULL
수강평 댓글 수정자 이메일(FK)	replyMemberUpdateEmail	DATE	NULL

결제(payment)

결제 코드(PK)	paymentCode	VARCHAR(40)	NOT NULL
결제 그룹 코드	paymentGroupCode	VARCHAR(40)	NOT NULL
회원이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
결제 방식 코드 (FK)	paymentMethodCode	VARCHAR(40)	NOT NULL
결제 방식	paymentMethodName	VARCHAR(40)	NOT NULL
총 주문 금액	paymentTotalOrderPrice	INT	NOT NULL
사용 포인트	paymentUsePoint	INT	NULL
최종 결제 금액	paymentTotalPayPrice	INT	NOT NULL
결제 여부	paymentState	VARCHAR	NOT NULL
결제 일자	paymentRegDate	DATETIME	NOT NULL

결제방식(paymentmethod)

결제 방식 코드 (PK)	paymentMethodCode	VARCHAR	NOT NULL
결제 방식	paymentMethodName	VARCHAR	NOT NULL
결제 방식 등록자 이메일(FK)	memberEmail	VARCHAR	NOT NULL
결제 방식 등록 날짜	paymentMethodRegDate	DATETIME	NOT NULL

위시리스트(wishlist)

위시리스트 코드(PK)	wishListCode	VARCHAR(40)	NOT NULL
회원 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
위시리스트 등록 날짜	wishListRegDate	DATETIME	NOT NULL
개설 승인된 운동 클래스 코드(FK)	eClassApprovedCode	VARCHAR(40)	NOT NULL
환불(refund)			
환불 코드(PK)	refundCode	VARCHAR(40)	NOT NULL
회원 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
결제 코드(FK)	paymentCode	VARCHAR(40)	NOT NULL
환불 기준 코드(FK)	refundStandardCode	VARCHAR(40)	NOT NULL
결제 일자	paymentRegDate	DATETIME	NOT NULL
환불 사유 코드(FK)	refundReasonCode	VARCHAR(40)	NOT NULL
환불 상세 사유	refundDetail	VARCHAR(40)	NOT NULL
환불 계좌 코드(FK)	refundAccount	VARCHAR(40)	NULL
환불 금액	refundPrice	INT	NOT NULL
환불 여부	refundState	VARCHAR(40)	NOT NULL
환불 요청 일자	refundRequestDate	DATETIME	NOT NULL
환불 처리 일자	refundRespondDate	DATETIME	NULL
환불기준(refundstandard)			
환불 기준 코드(PK)	refundAccountCode	VARCHAR(40)	NOT NULL
환불 기준	refundStandardNote	VARCHAR(40)	NOT NULL
기준에 따른 반환 금액	refundStandardPrice	VARCHAR(40)	NOT NULL
환불 기준 등록자 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
환불 기준 등록 날짜	refundStandardRegDate	DATETIME	NOT NULL

환불 기준 수정 회원 이메일(FK)	refundStandardUpdateEmail	VARCHAR(40)	NULL
환불 기준 수정 날짜	refundStandardUpdateDate	DATETIME	NULL
환불계좌(refundaccount)			
환불 계좌 코드(PK)	refundAccountCode	VARCHAR(40)	NOT NULL
은행명	refundAccountBankName	VARCHAR(40)	NOT NULL
예금주	refundAccountHolder	VARCHAR(40)	NOT NULL
계좌번호	refundAccountNumber	INT	NOT NULL
계좌 등록 회원 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
계좌 등록 날짜	refundAccountRegDate	DATETIME	NOT NULL
계좌 수정 회원 이메일(FK)	refundAccountUpdateEmail	VARCHAR(40)	NULL
계좌 수정 날짜	refundAccountUpdateDate	DATETIME	NULL
환불사유(refundreason)			
사유 코드(PK)	refundReasonCode	VARCHAR(40)	NOT NULL
사유 내용	refundReasonNote	VARCHAR(40)	NOT NULL
사유 등록 회원 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
사유 등록 날짜	refundReasonRegDate	DATETIME	NOT NULL
사유 수정 회원 이메일(FK)	refundReasonUpdateEmail	VARCHAR(40)	NULL
사유 수정 날짜	refundReasonUpdateDate	DATETIME	NULL
포인트 지급 기준(pointgivestandard)			
사유 코드(PK)	refundReasonCode	VARCHAR(40)	NOT NULL
사유 내용	refundReasonNote	VARCHAR(40)	NOT NULL
사유 등록 회원 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
포인트 내역(pointlog)			
포인트 발생 번호(PK)	pointLogNum	VARCHAR(40)	NOT NULL

받은 회원 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
직전 포인트	pointLogPrevious	INT	NOT NULL
포인트 지급 기준 코드(FK)	pointGiveStandardCode	VARCHAR(40)	NULL
증가/감소	pointLogChange	VARCHAR(40)	NOT NULL
증가/감소 포인트 금액	pointLogPrice	INT	NOT NULL
현재 포인트	pointLogState	INT	NOT NULL
포인트 상세 내용	pointLogDescContent	VARCHAR(200)	NOT NULL
포인트 발생 일자	pointLogRegDate	DATETIME	NOT NULL

음식내 식품영양성분(nutrients)

음식내 식품영양성분 코드(PK)	nutrientsCode	VARCHAR(40)	NOT NULL
작성자 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
음식명	nutrientsFood	VARCHAR(40)	NOT NULL
식단 음식 종류 코드(FK)	dietOneMealCode	VARCHAR(40)	NOT NULL
음식 칼로리	nutrientsKcal	VARCHAR(40)	NOT NULL
등록 날짜	nutrientsRegDate	DATE(11)	NOT NULL
식품영양성분 코드(FK)	nutrientsNAPICode	VARCHAR(40)	NOT NULL
식품 코드	nutrientsFoodcode	VARCHAR(40)	NOT NULL
식품군	nutrientsSort	VARCHAR(40)	NOT NULL
식품 이름	nutrientsName	VARCHAR(40)	NOT NULL

한끼 식사 식단 연결 테이블(dietOneMealConnection)

식단 한끼 식사 연결 코드(Pk)	dietOneMealConnection Code	VARCHAR(40)	NOT NULL
식단 음식 종류 코드(FK)	dietOneMealCode	VARCHAR(40)	NOT NULL
한끼 식사 그룹 번호	dietOneMealConnection GroupNum	VARCHAR(40)	NOT NULL
작성자 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL

작성일	dietOneMealConnectionDate	VARCHAR(40)	NOT NULL
식단은행번호(FK)	dietBankCode	VARCHAR(40)	NOT NULL
식단 은행 요일	dietBankOneMealConnectionDay	VARCHAR(40)	NOT NULL
식단 은행 시간	dietBankOneMealConnectionTime	VARCHAR(40)	NOT NULL
수정자 이메일(FK)	memberUpdateEmail	VARCHAR(40)	NULL
수정일	dietOneMealConnectionUpdateDate	VARCHAR(40)	NULL

식단 은행 등록 테이블(dietBank)

식단은행 번호	dietBankCode	VARCHAR(40)	NOT NULL
운동 클래스코드(FK)	eClassApprovedCode	VARCHAR(40)	NULL
작성자 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
작성일	dietBankInsertDate	VARCHAR(40)	NOT NULL
수정자 이메일(FK)	memberUpdateEmail	VARCHAR(40)	NULL
최종 수정일	dietBankUpdateDate	VARCHAR(40)	NULL

한끼 식사 총 칼로리 테이블(dietOneMealCalorie)

한끼 식사 총 칼로리 번호(PK)	dietOneMealCalorieCode	VARCHAR(40)	NOT NULL
한끼 식사 그룹 번호	dietOneMealConnectionGroupNum	VARCHAR(40)	NOT NULL
총 칼로리	dietOneMealCalorieTotal	VARCHAR(40)	NOT NULL

식단 계획 테이블(dietPlan)

식단 계획 코드(PK)	dietPlanCode	VARCHAR(40)	NOT NULL
운동클래스 코드(FK)	eClassApprovedCode	VARCHAR(40)	NOT NULL
한끼 식사 그룹 번호	dietOneMealConnectionGroupNum	VARCHAR(40)	NOT NULL
작성자 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
요일	dietPlanDay	VARCHAR(40)	NOT NULL
시간	dietPlanTime	VARCHAR(40)	NOT NULL

식단 실행 예정 날짜	dietPlannedDate	DATE(11)	NULL
식단 등록 날짜	dietPlanRegDate	DATE(11)	NULL
수정 날짜	dietPlanUpdateDate	DATE(11)	NULL
수정한 회원 이메일 (FK)	memberUpdateEmail	VARCHAR(40)	NULL
식단 계획 실행여부	dietPlanDo	INT(2)	NULL

식단 계획 테이블(dietPlanDo)

식단 실행 코드(PK)	dietPlanDoCode	VARCHAR(40)	NOT NULL
운동클래스 코드(FK)	eClassApprovedCode	VARCHAR(40)	NOT NULL
한끼 식사 그룹 번호	dietOneMealConnectionGroupNum	VARCHAR(40)	NULL
식단 실행한 회원 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
요일	dietPlanDoDay	DATE(11)	NOT NULL
시간	dietPlanDoTime	VARCHAR(40)	NOT NULL
내가 먹은 음식명	dietPlanDoEat	VARCHAR(40)	NOT NULL
음식 칼로리	dietPlanDoKcal	VARCHAR(40)	NOT NULL
식단 등록 날짜	dietPlanDoRegDate	DATE(11)	NOT NULL

식품영양성분 테이블(nutrientsAPI)

식품영양성분 코드(PK)	nutrientsAPICode	VARCHAR(40)	NOT NULL
식품 코드	nutrientsAPIFoodCode	VARCHAR(40)	NOT NULL
식품군	nutrientsAPISort	VARCHAR(40)	NOT NULL
식품 이름	nutrientsAPIName	VARCHAR(40)	NOT NULL
1회 제공량 (g)	nutrientsAPIServe	INT(10)	NOT NULL
열량(kcal)	nutrientsAPIKcal	INT(10)	NOT NULL
탄수화물(g)	nutrientsAPICarbo	INT(10)	NOT NULL
단백질(g)	nutrientsAPIPro	INT(10)	NOT NULL

지방(g)	nutrientsAPIFat	INT(10)	NOT NULL
당류(g)	nutrientsAPIGlu	INT(10)	NOT NULL
나트륨(mg)	nutrientsAPINatr	INT(10)	NOT NULL
콜레스테롤(mg)	nutrientsAPICole	INT(10)	NOT NULL
포화지방산(g)	nutrientsAPISFat	INT(10)	NOT NULL
트랜스지방(g)	nutrientsAPITrans	INT(10)	NOT NULL
챌린지 카테고리(challengecategory)			
챌린지 카테고리 코드 (PK)	challengeCategoryCode	VARCHAR(50)	NOT NULL
등록자 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
챌린지 카테고리명	challengeCategoryName	VARCHAR(50)	NOT NULL
등록 날짜	challengeCategoryRegDate	DATE	NOT NULL
운동 클래스 카테고리 (소)(FK)	challengeCategorySmall	VARCHAR(50)	NOT NULL
챌린지 카테고리명 수정자	memberUpdateEmail	VARCHAR(40)	NOT NULL
챌린지 카테고리명 수정 최종 날짜	memberUpdateDate	DATE (100)	NOT NULL
챌린지 모집(challengegather)			
챌린지 모집코드(PK)	challengeGatherCode	VARCHAR(50)	NOT NULL
챌린지 등록자 이메일 (FK)	memberEmail	VARCHAR(50)	NOT NULL
챌린지 카테고리 코드 (FK)	challengeCategoryCode	VARCHAR(50)	NOT NULL
챌린지 모집명	challengeGatherName	VARCHAR(50)	NOT NULL
챌린지 소개	challengeGatherIntro	VARCHAR(50)	NOT NULL
챌린지 최대참여인원수	challengeGatherMaxPeople	INT(10)	NOT NULL
챌린지 인증방법	challengeGatherProveMethod	VARCHAR(50)	NOT NULL
챌린지 인증내용	challengeGatherProveContent	VARCHAR(50)	NOT NULL
챌린지 인증빈도	challengeGatherProveFr	VARCHAR(50)	NOT NULL

	equency		
챌린지 달성률(%)	challengeGatherAchieveRate	INT(10)	NOT NULL
챌린지 시작일	challengeStartDate	DATE	NOT NULL
챌린지 종료일	challengeEndDate	DATE	NOT NULL
챌린지 등록날짜	challengeGatherRegDate	DATE	NOT NULL
챌린지 참여자 이메일(FK)	memberEnterEmail	VARCHAR(50)	NOT NULL
챌린지 모집 시작일	challengeGatherStartDate	DATE	NOT NULL
챌린지 모집 종료일	challengeGatherEndDate	DATE	NOT NULL

챌린지 모집 계획(challengegatherplan)

챌린지 계획 코드(PK)	challengeGatherPlanCode	VARCHAR(50)	NOT NULL
챌린지 등록자 이메일(FK)	memberEmail	VARCHAR(50)	NOT NULL
챌린지 모집 코드(FK)	challengeGatherCode	VARCHAR(50)	NOT NULL
챌린지 카테고리 코드(FK)	challengeCategoryCode	VARCHAR(50)	NOT NULL
챌린지 세팅 코드(FK)	challengeSettingCode	VARCHAR(50)	NOT NULL
챌린지 필수 인증 횟수	challengeGatherPlanProveNumber	INT(10)	NOT NULL
챌린지 진행률	challengeGatherPlanProgress	INT(10)	NOT NULL
챌린지 실행 후 최종 달성률	challengeGatherPlanAchieveRate	INT(10)	NOT NULL
챌린지 달성 인증 상태	challengeGatherPlanProveState	VARCHAR(50)	NOT NULL
챌린지 시작일	challengeGatherChallengeStartDate	DATE	NOT NULL
챌린지 종료일	challengeGatherChallengeLastDate	DATE	NOT NULL
회차	challengeGatherPlanRound	INT(10)	NOT NULL

챌린지 참여(challengeenter)

챌린지 참여 코드(PK)	challengeEnterCode	VARCHAR(50)	NOT NULL
챌린지 카테고리 코드(FK)	challengeCategoryCode	VARCHAR(50)	NOT NULL
챌린지 모집 코드(FK)	challengeGatherCode	VARCHAR(50)	NOT NULL

챌린지 참여자 이메일 (FK)	memberEmail	VARCHAR(50)	NOT NULL
챌린지 참여내용	challengeEnterContent	VARCHAR(50)	NOT NULL
챌린지 참가자 개인별 보증금	challengeEnterDeposit	INT(10)	NOT NULL
챌린지 참여날짜	challengeEnterDate	DATE	NOT NULL
챌린지 서비스 시작일	challengeServiceStartDate	DATE	NOT NULL
챌린지 서비스 종료일	challengeServiceEndDate	DATE	NOT NULL
챌린지 실행(challengegatherplando)			
챌린지 실행 코드(PK)	challengeGatherPlanDoCode	VARCHAR(50)	NOT NULL
챌린지 계획 코드(FK)	challengeGatherPlanCode	VARCHAR(50)	NOT NULL
챌린지 개설자 및 인증자 이메일(FK)	memberEmail	VARCHAR(50)	NOT NULL
챌린지 카테고리 코드(FK)	challengeCategoryCode	VARCHAR(50)	NOT NULL
챌린지 세팅 코드(FK)	challengeSettingCode	VARCHAR(50)	NOT NULL
챌린지 모집코드(FK)	challengeGatherCode	VARCHAR(50)	NOT NULL
챌린지 참가자 이메일 (FK)	memberEnterEmail	VARCHAR(50)	NOT NULL
실행내용	challengeGatherPlanDoContent	VARCHAR(50)	NOT NULL
챌린지 실행 등록 날짜	challengeGatherPlanDoRegDate	DATE	NOT NULL
챌린지 실행 인증 상태	challengeGatherPlanDoState	VARCHAR(50)	NOT NULL
챌린지 실행 인증 후 달성률	challengeGatherPlanDoAchieveRateAfterProve	INT(10)	NOT NULL
챌린지 실행 후 달성률	challengeGatherPlanDoAchieveRate	INT(10)	NOT NULL
챌린지 인증 확인날짜	challengeGatherPlanDoProveCheckDate	DATE	NOT NULL
챌린지 세팅(challengeSetting)			
챌린지 세팅 코드(PK)	challengeSettingCode	VARCHAR(50)	NOT NULL
챌린지 진행 기간 지난 후 수정 가능 여부	challengeSettingUpdateDo	VARCHAR(20)	NOT NULL
챌린지 포인트 지급(challengePointGive)			

챌린지 포인트 지급 코드(PK)	challengePointGiveCode	VARCHAR(50)	NOT NULL
챌린지 카테고리 코드(FK)	challengeCategoryCode	VARCHAR(50)	NOT NULL
챌린지 모집 코드(FK)	challengeGatherCode	VARCHAR(50)	NOT NULL
챌린지 참여 코드(FK)	challengeEnterCode	VARCHAR(50)	NOT NULL
챌린지 달성률 코드(FK)	challengeGatherAchieveRate	VARCHAR(50)	NOT NULL
챌린지 참여자 이메일(FK)	challengeEnterEmail	VARCHAR(40)	NOT NULL
포인트 지급 기준 코드(FK)	pointGiveStandardCode	VARCHAR(50)	NOT NULL
챌린지 보상 포인트 지급 날짜	challengePointGiveDate	DATE	NOT NULL
포인트 점수	challengePointGivePoint	INT(11)	NOT NULL
챌린지 신고(challengeReport)			
챌린지 신고 코드(PK)	challengeReportCode	VARCHAR(50)	NOT NULL
챌린지 카테고리 코드(FK)	challengeCategoryCode	VARCHAR(50)	NOT NULL
챌린지 모집 코드(FK)	challengeGatherCode	VARCHAR(50)	NOT NULL
챌린지 참여 코드(FK)	challengeEnterCode	VARCHAR(50)	NOT NULL
챌린지 신고자 이메일(FK)	challengeReportEmail	VARCHAR(50)	NOT NULL
챌린지 피신고자 이메일(FK)	challengeReportSuspectEmail	VARCHAR(40)	NOT NULL
신고 코드(FK)	reportStandardCode	VARCHAR(40)	NOT NULL
신고 내용	challengeReportContent	VARCHAR(50)	NOT NULL
챌린지 신고날짜	challengeReportDate	DATE	NOT NULL
AI 운동 영상(aivideo)			
AI 운동 영상 코드(PK)	fileIdx	VARCHAR(40)	NOT NULL
AI 운동 영상 등록자 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
AI 운동 영상 등록 날짜	AIVideoRegDate	DATE	NOT NULL
AI 운동 영상 수정자 이메일	memberUpdateEmail	VARCHAR(40)	NOT NULL
AI 운동 영상 수정 최종	AIVideoUpdateDate	DATE	NOT NULL

날짜			
AI 운동 영상 제목	AIVideoTitle	VARCHAR(40)	NOT NULL
회원 운동 목표 테이블(workoutgoal)			
회원 운동 목표 코드 (PK)	workoutGoalCode	VARCHAR(40)	NOT NULL
회원 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
회원 운동 목표 내용	workoutGoalContent	VARCHAR(1000)	NOT NULL
회원 운동 목표 시작일	workoutGoalStartDate	DATE	NOT NULL
회원 운동 목표 종료일	workoutGoalEndDate	DATE	NOT NULL
회원 운동 목표 최종 수정 날짜	workoutGoalUpdateDate	DATE	NULL
회원 운동 목표 최종 수정자 이메일	memberUpdateEmail	VARCHAR(40)	NULL
운동 계획 테이블(workoutgoalplan)			
운동 계획표 코드(PK)	workoutGoalPlanCode	VARCHAR(40)	NOT NULL
회원 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
회원 운동 목표 코드 (FK)	workoutGoalCode	VARCHAR(40)	NOT NULL
운동 계획 시작일	workoutGoalPlanStartDate	DATE	NOT NULL
운동 계획 종료일	workoutGoalPlanFinalDate	DATE	NOT NULL
운동 계획 내용	workoutGoalPlanContent	VARCHAR(1000)	NOT NULL
운동 계획 수정자 이메일(FK)	workoutGoalPlanUpdateEmail	VARCHAR(40)	NULL
운동 계획 최종 수정일	workoutGoalPlanFinalUpdateTime	DATE	NULL
AI 운동 영상 코드(FK)	fileIdx	VARCHAR(50)	NULL
AI 운동 수행 횟수	AIWorkoutPerformanceCount	INT(11)	NULL
AI 운동 수행 목표	AIWorkoutPerformanceGoal	VARCHAR(100)	NULL
운동 계획 달성 여부	workoutGoalPlanAchieveState	VARCHAR(50)	NULL
운동 계획 카테고리 코드 (FK)	workoutGoalPlanCategoryCode	VARCHAR(40)	NOT NULL
운동 계획 카테고리 테이블(workoutgoalplancategory)			
운동 계획 카테고리 코드 (PK)	workoutGoalPlanCategoryCode	VARCHAR(40)	NOT NULL
운동 계획 카테고리 내용	workoutGoalPlanCategoryContent	VARCHAR(40)	NOT NULL
일지 테이블(workoutlog)			
일지 코드(PK)	workoutLogCode	VARCHAR(40)	NOT NULL

회원 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
일지 좋아요 카운트	workoutLogLikeCount	INT(10)	0
운동 클래스 카테고리 소 code (FK)	eClassCategorySmallCode	VARCHAR(40)	NULL
운동 클래스 카테고리 소 이름	eClassCategorySmallName	VARCHAR(50)	NULL
일지 제목	workoutLogTitle	VARCHAR(40)	NOT NULL
일지 내용	workoutLogContent	VARCHAR(1000)	NOT NULL
운동 계획표 코드(FK)	workoutGoalPlanCode	VARCHAR(40)	NULL
운동 클래스 과제 수행 코드 (FK)	eClassTaskDoCode	VARCHAR(40)	NULL
챌린지 참여 코드 (FK)	challengeEnterCode	VARCHAR(40)	NULL
일지 첨부 파일 코드 (FK)	fileIdx	VARCHAR(50)	NULL
일지 공개 범위 코드 (FK)	workoutLogPrivacyBoundsCode	VARCHAR(40)	NOT NULL
일지 수정자 이메일 (FK)	workoutLogUpdateEmail	VARCHAR(40)	NULL
일지 수정 최종 날짜	workoutLogUpdateFinalDate	DATE	NULL
일지 삭제자 이메일	workoutLogDeleteEmail	VARCHAR(40)	NULL

일지 댓글 테이블(workoutlogcomment)

일지 댓글 코드(PK)	workoutLogCommentCode	VARCHAR(40)	NOT NULL
일지 코드(FK)	workoutLogcode	VARCHAR(40)	NOT NULL
일지 댓글 등록자 이메일(FK)	memberEmail	VARCHAR(40)	NOT NULL
일지 댓글 내용	workoutLogCommentContent	VARCHAR(1000)	NOT NULL
일지 댓글 등록 날짜	workoutLogCommentRegDate	DATE	NOT NULL
일지 댓글 수정자 이메일(FK)	workoutLogCommentUpdateEmail	VARCHAR(40)	NULL
일지 댓글 수정 날짜	workoutLogCommentUpdateDate	DATE	NULL
일지 댓글 삭제자 이메일(FK)	workoutLogCommentDeleteEmail	VARCHAR(40)	NULL
일지 댓글 삭제 날짜	workoutLogCommentDeleteDate	DATE	NULL

일지 댓글 답변 테이블(workoutlogcommentreply)

댓글 답변 코드 (PK)	workoutLogCommentReplyCode	VARCHAR(40)	NOT NULL
일지 댓글 코드(FK)	workoutLogCommentCode	VARCHAR(40)	NOT NULL

일지 댓글 답변 등록자 이메일(FK)	workoutLogCommentRe plyEmail	VARCHAR(40)	NOT NULL
일지 댓글 답변 내용	workoutLogCommentRe plyContent	VARCHAR(1000)	NOT NULL
일지 댓글 답변 작성 날 짜	workoutLogCommentRe plyRegDate	DATE	NOT NULL
일지 댓글 답변 수정자 이메일(FK)	memberUpdateEmail	VARCHAR(40)	NULL
일지 댓글 답변 수정 날 짜	workoutLogCommentRe plyUpdateDate	DATE	NULL
일지 댓글 답변 삭제자 이메일(FK)	memberDeleteEmail	VARCHAR(40)	NULL
일지 좋아요 테이블(workoutloglike)			
일지 좋아요 번호(PK)	workoutLogLikeNum	VARCHAR(50)	NOT NULL
일지 좋아요 누른 사람 이메일(FK)	workoutLogLikeEmail	VARCHAR(40)	NOT NULL
일지 코드(FK)	workoutLogCode	VARCHAR(40)	NOT NULL
좋아요 등록 날짜	workoutLogLikeRegDate	DATE	NOT NULL
일지 공개 범위 테이블(workoutlogprivacybounds)			
일지 공개 범위 코드 (PK)	workoutLogPrivacyBoun dsCode	VARCHAR(40)	NOT NULL
일지 공개 범위 내용	workoutLogPrivacyBoun dsContent	VARCHAR(1000)	NOT NULL
첨부 파일 테이블(tb_file)			
파일idx	fileIdx	VARCHAR(50)	NOT NULL
파일오리지날명	fileOriginalName	VARCHAR(150)	NOT NULL
새로운파일명	fileNewName	VARCHAR(150)	NOT NULL
파일경로	filePath	VARCHAR(150)	NOT NULL
파일사이즈	fileSize	INT(11)	NOT NULL
첨부 파일_2 테이블(tb_file_control)			
순번	no	VARCHAR(50)	NOT NULL
참조 코드	referenceCode	VARCHAR(40)	NULL
파일idx	fileIdx	VARCHAR(40)	NULL

4.3 네이밍 규칙

<p>공통 규칙</p>	<ol style="list-style-type: none"> 1. 모든 이름은 a-z, A-Z, 0-9의 글자로 조합한다. 2. 프로그램 내에서 사용하는 모든 이름은 64자를 넘지 않는 것으로 한다. 3. 정확한 의미 전달을 위해 약어의 사용은 피한다. 4. 특수기호는 사용하지 않는 것을 원칙으로 한다. 5. 약어 사용이 의미 전달에 도움이 되는 경우에는 표준용어사전에 포함되어있는 약속된 약어를 사용할 수 있으나, 그렇지 않은 경우에는 단어의 spelling을 모두 적는 것으로 한다. 																																		
<p>파일명 규칙</p>	<p>JAVA</p>	<p>대소문자를 사용하며 단어의 시작부분은 대문자로 표기한다. (파스칼 표기법) 단어는 표준용어사전을 따른다. 명사 또는 동사 + 명사의 혼합된 형태로 이름을 구성한다. 첫 번째 문자는 항상 대문자를 사용한다. 클래스 유형에 따라 다음 표를 참고하여 구성한다.</p>																																	
	<p>mapper</p>	<p>Dao (java파일)의 명명 규칙을 따른다. (ex: Dao MemberMapper -> mapper MemberMapper) 파일의 확장자는 .xml로 한다.</p>																																	
	<p>html</p>	<p>대소문자를 사용하여 단어의 시작부분은 소문자로 표기하고 뒤에 오는 단어 앞자리 한 글자를 대문자를 사용한다. (카멜 표기법) 파일의 확장자는 .html로 한다.</p>																																	
	<p>css</p>	<p>대소문자를 사용하여 단어의 시작부분은 소문자로 표기하고 뒤에 오는 단어 앞자리 한 글자를 대문자를 사용한다. (카멜 표기법) 파일의 확장자는 .css로 한다.</p>																																	
	<p>javaScript</p>	<p>대소문자를 사용하여 단어의 시작부분은 소문자로 표기하고 뒤에 오는 단어 앞자리 한 글자를 대문자를 사용한다. (카멜 표기법) 파일의 확장자는 .js로 한다. 오픈소스 자바스크립트의 경우, 최초 배포된 이름 그대로 유지한다.</p>																																	
<p>Method 규칙</p>	<ol style="list-style-type: none"> 1. 동사 혹은 동사 + 명사로 작성하며, 혼합된 형태로 이름을 구성할 때에는 각 단어의 첫 글자는 대문자로 작성한다. (카멜표기법) 2. 상황에 따라 다음 표를 참고하여 일반적인 Prefix를 적용하며, 나머지는 필요에 따른 적절한 이름을 사용한다. 																																		
	<table border="1" data-bbox="469 1249 1366 1615"> <thead> <tr> <th>기능유형</th> <th>대표용어</th> <th>기능유형</th> <th>대표용어</th> </tr> </thead> <tbody> <tr> <td>get</td> <td>조회</td> <td>save</td> <td>저장</td> </tr> <tr> <td>insert</td> <td>등록</td> <td>call</td> <td>호출</td> </tr> <tr> <td>update</td> <td>수정</td> <td>send</td> <td>송신</td> </tr> <tr> <td>delete</td> <td>삭제</td> <td>receive</td> <td>수신</td> </tr> <tr> <td>confirm</td> <td>확인</td> <td>transfer</td> <td>송수신</td> </tr> <tr> <td>cancel</td> <td>취소</td> <td>check</td> <td>검토</td> </tr> <tr> <td>calc</td> <td>계산</td> <td>is</td> <td>유무(여부)확인</td> </tr> </tbody> </table>			기능유형	대표용어	기능유형	대표용어	get	조회	save	저장	insert	등록	call	호출	update	수정	send	송신	delete	삭제	receive	수신	confirm	확인	transfer	송수신	cancel	취소	check	검토	calc	계산	is	유무(여부)확인
기능유형	대표용어	기능유형	대표용어																																
get	조회	save	저장																																
insert	등록	call	호출																																
update	수정	send	송신																																
delete	삭제	receive	수신																																
confirm	확인	transfer	송수신																																
cancel	취소	check	검토																																
calc	계산	is	유무(여부)확인																																
<p>변수명 규칙</p>	<ol style="list-style-type: none"> 1. 대소문자를 사용하여 변수명의 시작부분을 소문자로 표기한다. 2. 변수명에는 기호나 특수문자를 사용하지 않는다. 3. 변수명은 의미 있는 단어를 사용하도록 한다.(표준 용어사전 참고) 4. 변수에 해당하는 컬럼이 테이블에 존재하는 경우에는 컬럼명을 기준으로 camel 표기법을 준수하여 사용한다. 5. 변수명은 중복되는 이름이 나오지 않도록 하여 일련번호를 붙여 사용하는 경우가 없도록 한다. 6. 상수명은 공통코드를 제외한 변수명 전체를 대문자로 하고, 단어 연결에는 '_'를 사용한다. 																																		
<p>DB</p>	<ol style="list-style-type: none"> 1. DB의 테이블 및 칼럼의 모든 명칭은 영문자(소문자)로 시작한다. 2. 테이블명칭은 해당 주요기능을 표현하며, 칼럼 또한 칼럼의 속성이 표현될 수 있는 명칭을 사용한다. 3. 의미를 알 수 있는 약어는 사용하지 않으며 되도록 풀네임을 사용한다. 																																		

4.4 디렉토리 구조

view	member	addInfo	회원 추가 정보 등록 화면	
		addMember	회원 가입 화면	
		login	관리자 권한 로그인 화면	
		login2	일반 회원 권한 로그인 화면	
		login3	미인증 회원 권한 로그인 화면	
		loginHistory	로그인 내역 조회 화면	
		memberDetail	회원 정보 조회 화면	
		memberLevelList	회원 권한 조회 화면	
		memberList	회원 목록 조회 화면	
		modifyMember	회원 정보 수정 화면	
		mypage	마이페이지 화면	
		removeMember	회원 탈퇴 화면	
		wishList	위시리스트 화면	
		eClass	eClassConfig	eClassCategoryInsert
			eClassCategoryUpdate	운동 클래스 카테고리 수정 화면
			eClassCategoryDelete	운동 클래스 카테고리 삭제 화면
			eClassCategoryList	운동 클래스 카테고리 조회 화면
		report	addReport	신고 신청 화면
			modifyReport	신고 수정 화면
			removeReport	신고 취소 화면
			reportList	신고 목록 조회 화면
			addReportStandard	신고 기준 등록 화면
			modifyReportStandard	신고 기준 수정 화면
			removeReportStandard	신고 기준 삭제 화면
			reportStandardList	신고 목록 조회 화면
	reportProcess		신고 처리 화면	
	eClass	eClassApplyadminList	개설 승인된 운동 클래스 리스트 (관리자)	
		eClassApproved	개설 승인된 운동 클래스 상세내용 (사용자)	
		eClassApprovedList	개설 승인된 운동 클래스 리스트	
		eClassOpenAppleyComplete	개설 신청 완료 페이지	
		eClassOpenAppleyForm	클래스 개설 신청 폼	
		eClassOpenAppleyFormUpdate	클래스 개설 신청 수정폼	
		eClassOpenApplyadminList	개설 신청한 운동 클래스 리스트 (관리자)	
		myEClassApplyList	사용자가 신청한 미승인된 클래스 리스트	
view	eClass	eClassTake	수강 신청 화면	
		myeClassList	사용자 수강 클래스 목록 조회 화면	
		mySignUpForClassList	사용자 수강 신청 내역 조회 화면	
		studentList	운동클래스 수강생 목록 조회 화면	
	payment	mypaymentList	사용자 결제 내역 조회 화면	
		payment	결제 화면	
		paymentDetail	결제 상세 내역 조회 화면	
		paymentList	관리자 결제 내역 조회 화면	
	refund	myrefundList	사용자 환불 내역 조회 화면	
		refundDetail	환불 상세 내역 조회 화면	
		refundList	관리자 환불 내역 조회 화면	
		refundStandard	환불 기준 조회 및 설정 화면	
	point	givePoint	포인트 지급 화면	
		mypointList	사용자 포인트 내역 조회 화면	
		pointList	관리자 포인트 내역 조회 화면	
		pointStandard	포인트 지급 기준 화면	
		dietMyList	사용자 개인 식단 페이지	
	diet	AjaxTable	dietBankListAdmin	식단 은행 조회 페이지
			updateDietBankList	식단은행 상세사항 관리 페이지
DietBankConnMealListAjax			식단은행 상세사항 관리 조회 화면1	
DietBankConnMealListDayAjax			식단은행 상세사항 관리 조회 화면2	
DietUserMealplannedListAjax			날짜별 사용자 개인 식단 조회	
DietUserMealListAjax			추가할 음식조회 페이지	
DietMyListSelectBankDetailAjax			식단 은행 상세내용 조회페이지	
dietMyListSelectBankAjax			식단은행 조회페이지	

		DietmainMealSortAjax	식단 대분류	
		DietmainMealSortDAjax	식단 상세분류	
challenge	challengeadmin		모집 챌린지 관리자 메인 화면	
	challengeConfig	challengeConfigList	챌린지 관리 설정 화면 (관리자) - 카테고리 조회, 세팅 조회, 삭제는 바로 가능	
		challengeCategoryInsert	챌린지 카테고리 입력 폼 (관리자)	
		challengeCategoryUpdate	챌린지 카테고리 수정 폼 (관리자)	
		challengeSettingInsert	챌린지 세팅 입력 폼 (관리자)	
		challengeSettingUpdate	챌린지 세팅 수정 폼 (관리자)	
		challengeReportList	챌린지 신고 관리 화면 (관리자) - 챌린지 신고 된 전체리스트 및 검색	
		challengeList.html	챌린지 관리 화면 (관리자) 챌린지 전체 목록 조회 (달성률 100인 챌린지 포인트 지급 가능)	
		challengeListDetail	선택한 챌린지에 세부 정보 보기 모달 화면(참가중인 회원목록 확인 가능)	
		challengeReportDetail	선택한 신고 정보 세부 내용 보기 모달 화면	
		challengeGather	challengeGatherList	챌린지 모집 화면 - 모집 중인 챌린지 전체 리스트 조회 및 검색
	challengeGatherInsert		챌린지 모집 입력 폼 (일반회원/관리자) - 모집 기본 내용, 모집 계획 입력하는 폼	
	challengeGatherAdmin		개설 챌린지 관리 화면 (챌린지 개설자) - 개설한 챌린지 조회	
	challengeGatherUpdate		챌린지 모집 수정 폼	
	challengeGaterInMember		개설 챌린지 참여 관리 화면 (챌린지 개설자) - 개설된 챌린지 별 참가 회원 조회, 달성률 조회, 인증 관리(모달로 인증내용 확인가능/ 인증 가능)	
	challengeGaterInMemberDetail		챌린지 참여자의 인증 내용 상세보기 모달 화면	
	challengeGatherListDetail		선택한 모집 챌린지 상세보기 모달화면, 참가신청도 할수 있음	
	challengeGatherAdminDetail		선택한 개설 챌린지 상세보기 모달 화면	
	challengeEnter	challengeEnterPaymem	챌린지 참가를 위한 결제 폼	
		challengeEnterList	참가 챌린지 화면 - 참가 하고 있는 챌린지 전체 리스트 조회 및 검색	
		challengeEnterDetail	선택한 참가 중인 챌린지 세부 정보 확인 화면 - 실행 관리, 참가한 회원 조회	
		challengePlanDoInsert	챌린지 실행하기 등록 화면(인증) - 챌린지 실행 내용 입력	
	workoutLog	workoutLogUser	workoutLogMain	일지 목록 조회
			workoutGoalList	운동 목표 목록 조회
			workoutGoalInsert	운동 목표 등록
			workoutGoalUpdate	운동 목표 수정
			workoutGoalDelete	운동 목표 삭제
			workoutGoalPlanList	운동 계획 목록 조회
workoutGoalPlanInsert			운동 계획 등록	
workoutGoalPlanUpdate			운동 계획 수정	
workoutGoalPlanDelete			운동 계획 삭제	
workoutLogInsert			일지 등록	
workoutLogUpdate		일지 수정		
workoutLogDelete		일지 삭제		
workoutLogConfig		workoutCategoryList	운동 계획 카테고리 목록 조회	
		workoutCategoryInsert	운동 계획 카테고리 등록	
		workoutCategoryUpdate	운동 계획 카테고리 수정	
		workoutCategoryDelete	운동 계획 카테고리 삭제	
		workoutAIVideoList	AI 운동 영상 목록 조회	
		workoutAIVideoInsert	AI 운동 영상 등록	
		workoutAIVideoUpdate	AI 운동 영상 수정	
		workoutAIVideoDelete	AI 운동 영상 삭제	

4.6 패키지 구조

패키지명	클래스명	메서드명	설명		
aihometraining.team.controller	MemberController	addMember	회원 가입		
		addInfo	회원 추가 정보 입력		
		login	관리자 권한 로그인		
		login2	일반 회원 권한 로그인		
		login3	미인증 회원 권한 로그인		
		isEmailCheck	이메일 중복 체크		
		logout	로그아웃		
	MemberAdminController	getLoginHistory	로그인 이력 조회		
		getMemberList	회원 목록 조회		
		memberLevelList	회원 권한 목록 조회		
	EClassCategoryController	eClassCategoryInsert	운동 클래스 카테고리 추가 등록		
		eClassCategoryList	운동 클래스 카테고리 목록 조회		
ReportController	reportList	신고 목록 조회			
	reportStandardList	신고 기준 목록 조회			
aihometraining.team.config	MemberController	addMember	회원 가입		
		addInfo	회원 추가 정보 입력		
		login	관리자 권한 로그인		
		login2	일반 회원 권한 로그인		
		login3	미인증 회원 권한 로그인		
		isEmailCheck	이메일 중복 체크		
		logout	로그아웃		
	MemberAdminController	getLoginHistory	로그인 이력 조회		
		getMemberList	회원 목록 조회		
		memberLevelList	회원 권한 목록 조회		
	EClassCategoryController	eClassCategoryInsert	운동 클래스 카테고리 추가 등록		
		eClassCategoryList	운동 클래스 카테고리 목록 조회		
ReportController	reportList	신고 목록 조회			
	reportStandardList	신고 기준 목록 조회			
aihometraining.team.mapper	EClassMapper	eClassCategoryLargeList	AJAX를 이용한 카테고리 조회		
		eClassCategoryMediumList	AJAX를 이용한 카테고리 조회		
		eClassCategoryLarge	AJAX를 이용한 카테고리 조회		
		eClassCategoryMedium	AJAX를 이용한 카테고리 조회		
		eClassOpenApplyList	신청한클래스조회		
		EClassIntroduceInsert	클래스소개등록처리		
		EClassIntroduceUpdate	클래스소개수정처리		
		EClassSectionTitleInsert	클래스섹션등록처리		
		EClassSectionTitleUpdate	클래스섹션수정처리		
		EClassSectionCurriculumInsert	클래스커리큘럼등록처리		
		EClassSectionCurriculumUpdate	클래스커리큘럼수정처리		
		EClassQuestionInsert	클래스 질문 등록 처리		
		EClassQuestionUpdate	클래스 질문 수정 처리		
		EClassAnswerInsert	클래스답변등록처리		
		EClassAnswerUpdate	클래스답변수정처리		
		EClassPriceInsert	클래스세부설정등록처리		
		EClassPriceUpdate	클래스세부설정수정처리		
		eClassApprovedByCode	클래스수정화면		
		aihometraining.team.eClassController	eClassController	eClassApprovedList	개설승인된리스트
				OpenApplyForm	운동클래스신청폼
EClassLarge	카테고리 조회				
EClassMedium	카테고리 조회				
EClassOpenApplyFormInsert	개설 신청 처리				
EClassOpenApplyFormUpdate	개설 신청 수정 화면				
EClassApplyFormUpdate	개설 신청 수정 처리				
complete	신청 완료 화면				

		eClassApproved	개설 승인된 클래스 상세내용		
		eClassOpenApplyadminList	개설 신청한 리스트(관리자)		
		eClassApplyadminList	개설 승인된 리스트(관리자)		
		MyApplyList	사용자가 신청한 클래스 개설 현황		
		studentList	수강생목록		
aihometraining.team.eclass Service	eClassService	eClassCategoryLargeList	카테고리 조회		
		eClassCategoryMediumList	카테고리 조회		
		MyApplyList	사용자가 신청한 클래스 개설 현황		
		EClassIntroduceInsert	클래스 소개 등록처리		
		EClassIntroduceUpdate	클래스 소개 수정처리		
		EClassSectionTitleInsert	클래스 섹션 등록처리		
		EClassSectionTitleUpdate	클래스 섹션 수정처리		
		EClassSectionCurriculumInsert	클래스 커리큘럼 등록처리		
		EClassSectionCurriculumUpdate	클래스 커리큘럼 수정처리		
		EClassQuestionInsert	클래스 질문 등록처리		
		EClassQuestionUpdate	클래스 질문 수정처리		
		EClassAnswerInsert	클래스 답변 등록처리		
		EClassAnswerUpdate	클래스 답변 수정처리		
		EClassPriceInsert	클래스 세부사항 등록처리		
		EClassPriceUpdate	클래스 세부사항 수정처리		
		aihometraining.team.contr oller	MypageController	eClassApprovedByCode	개설 신청한 클래스 수정화면
				mypaymentList	사용자 결제 내역 조회
myrefundList	사용자 환불 내역 조회				
mypointList	사용자 포인트 내역 조회				
paymentDetail	사용자 결제 상세 내역 조회				
refundDetail	사용자 환불 상세 내역 조회				
myClassList	사용자 수강 클래스 조회				
mySignUpForClassList	사용자 수강 신청 내역 조회				
PaymentController	wishList		위시리스트 조회		
	wishList		위시리스트 삭제		
	eClassTake		수강신청 화면		
	eClassTake		수강신청 처리		
	payment		결제 처리		
	payment		결제 화면		
	memberPhoneCheck		휴대폰 번호 인증		
PaymentAdmin Controller	paymentList		결제 내역 조회		
	paymentDetail		결제 상세 내역 조회		
RefundController	refundList		환불 내역 조회		
	refundStandard		환불 상세 내역 조회		
PointController	givePoint		포인트 지급		
	pointList		포인트 내역 조회		
	pointStandard	포인트 지급 기준 조회			
aihometraining.team.diet.c ontroller	DietUserController	selectDietMyList	개인 식단 페이지 이동, 음식 대분류 조회, 개인 식단 계획 조회		
		selectDietMealList	Ajax 대분류 선택 후 음식 조회(user)		
		insertUserDietPlan	Ajax 개인 식단 계획에 음식 삽입		
		selectChangedDietPlanList	Ajax 식단 계획 화면 새로고침		
		deleteUserDietPlan	Ajax 식단 계획 삭제		
		updateUserDietPlan	Ajax 식단 계획 페이지에서 실행 눌렀을 때, 혹은 취소 눌렀을 때 update		
		updateProgress	AjaxUser식단표progress-bar		
	selectDietBankListByUser	Ajaxuserdiet에서식단은행블러오기			
DietService	selectDietBankListDetail	Ajax user diet에서 식단은행 디테일 블러오기			

		insertDietBankList2	신규식단은행생성
		getDietBankListAdmin	식단은행 불러오기(관리자)
		deleteDietBankList	식단은행 삭제
		selectDietBankListNewCode	삽입시 db 신규 PK생성
		insertDietOnemealConnection	식단은행 개별 식단 생성
		selectBankDay	식단 은행 시간 목록 조회
		MealDayList	식단 은행 날짜 목록 조회
		deleteDietOneMealConnection	식단은행 개별 식단 삭제
		updateDietBank	식단 은행 상세페이지 업데이트
		insertDietBankListDetail	Ajaxuser식단에dietBank내용삽입
		selectBankDay	식단은행 시간값 날짜값 자동 생성
		insertUserDietPlan	User DietPlan 추가버튼 눌렀을 때 insert
		selectUserDietPlan	UserDietPlan 페이지 select
		deleteUserDietPlan	UserDietPlan 에서 삭제
		updateUserDietPlan	UserDietPlan에서 식단 실행, 취소 후 Update
		UserdietPlanList	영양소 프로그래스바를 위한 total 영양소 계산기
		DietController	insertDietOneMealConnection All
	getDietBankListAdmin		식단은행관리자페이지조회
	insertDietBankList		식단은행 생성
	insertDietBankList2		식단 은행 생성, 생성된 페이지(수정페이지)로 이동
	updateDietBankList		식단은행 식단 수정
	deleteDietBankList		식단 은행 삭제
	selectDietMealLevelList		Ajax 식단음식대분류조회
	selectDietMealListDetail		Ajax 음식소분류조회
	selectDietMealList		Ajax 대분류선택후음식조회
	dietOneMealConnection	Ajax 식단은행에서추가버튼insert, 내역update, 변화내용select	
	deleteDietOneMealConnection	Ajax 식단은행에서제거버튼누를때delete, 내역update, 변화내용select	
	DietMapper	getDietBankListAdmin	식단 은행 식단 화면에 조회
		selectDietBankListNewCode	식단 생성시 DB에 넣어줄 새 코드 작성
		selectDietMealLevelList	Ajax 식단 음식 대분류 조회
		selectDietMealListDetail	Ajax 음식 소분류 조회
		selectDietMealList	Ajax 대분류 선택 후 음식 조회
		updateDietBankList	식단 은행 식단 화면에 조회
selectDietBankListNewCode		식단 은행 새 코드 불러오기	
getDietBankListAdmin		식단 은행 리스트 불러오기	
getDietBankListAdminByC		식단 은행 리스트 코드로 불러오기	
selectDietMealLevelList		음식 검색 대분류	
selectDietMealListDetail		음식 검색 소분류	
selectDietNutrientList		음식 리스트 검색	
insertDietOneMealConnection		요일별 식단 삽입	
selectDietOneMealConnection ByBankCode		요일별 식단 조회	
deleteDietOneMealConnection		요일별 식단 삭제	
insertUserDietPlan		User 식단 계획 생성	
selectUserDietPlan	User 식단 계획 조회		
deleteUserDietPlan	User 식단 계획 삭제		
updateUserDietPlan	User 식단 계획 실행 및 취소 (update)		

aihometraining.team.challengeConfig.controller	ChallengeConfigController	challengeCategoryInsert	챌린지 카테고리 등록 처리
	ChallengeConfigController	challengeConfigList	챌린지 관리 설정 화면 : 챌린지 카테고리 , 세팅 목록 Data view 반환
	ChallengeConfigController	challengeCategoryUpdate	챌린지 카테고리 수정처리
	ChallengeConfigController	challengeCategoryDelete	챌린지 카테고리 삭제처리
	ChallengeConfigController	challengeSettingInsert	챌린지 세팅 등록 처리
	ChallengeConfigController	chllengeSettingUpdate	챌린지 세팅 수정처리
	ChallengeConfigController	deleteChallengeSetting	챌린지 세팅 삭제처리
	ChallengeConfigController	getChallengeReportList	챌린지 신고 관리 화면 : 챌린지 신고 목록 Data view 반환 - click값 전달 : 신고 세부 내용 보기 모달
	ChallengeConfigController	challengeList	모집 챌린지 목록 화면 : 챌린지 전체 목록 Data view 반환 -click값 전달 : 챌린지에 세부 정보 보기 모달
	ChallengeConfigController	geReportttListDetailByCode	(모달)챌린지 신고 관리 화면에서 전달받은 값으로 해당 신고 세부 정보 Data view 반환
	ChallengeConfigController	getListDetailByCode	(모달)모집 챌린지 목록 화면에서 전달받은 값으로 해당 챌린지 세부 정보 Data view 반환
	ChallengeConfigController	challengeListIng	진행 챌린지 목록 화면 : 진행 챌린지 전체 목록 Data view 반환 -click값 전달 : 챌린지에 세부 정보 보기 모달
	ChallengeConfigController	challengePoint	챌린지 포인트 목록 화면: 챌린지 포인트전체 목록 Data view 반환 -click값 전달 : 챌린지에 세부 정보 보기 모달, 포인트 지급페이지로 이동
	ChallengeConfigService	getPointDetailByCode	(모달)모집 챌린지 목록 화면에서 전달받은 값으로 해당 챌린지 포인트 정보 Data view 반환
ChallengeConfigService	challengeCategoryInsert	챌린지 카테고리 등록 처리	
aihometraining.team.challengeConfig.service	ChallengeConfigService	challengeConfigList	챌린지 관리 설정 화면 : 챌린지 카테고리 , 세팅 목록 Data view 반환
	ChallengeConfigService	challengeCategoryUpdate	챌린지 카테고리 수정처리
	ChallengeConfigService	challengeCategoryDelete	챌린지 카테고리 삭제처리
	ChallengeConfigService	challengeSettingInsert	챌린지 세팅 등록 처리
	ChallengeConfigService	chllengeSettingUpdate	챌린지 세팅 수정처리
	ChallengeConfigService	deleteChallengeSetting	챌린지 세팅 삭제처리
	ChallengeConfigService	getChallengeReportList	챌린지 신고 관리 화면 : 챌린지 신고 목록 Data view 반환 - click값 전달 : 신고 세부 내용 보기 모달
	ChallengeConfigService	getChallengeList	챌린지 관리 화면 : 챌린지 전체 목록 Data view 반환 -click값 전달 : 챌린지에 세부 정보 보기 모달 - buttonClick 값 전달 : 해당 챌린지에 참여한 참가자에게 포인트 지급
	ChallengeConfigService	getChallengeReportDetail	(모달)챌린지 신고 관리 화면에서 전달받은 값으로 해당 신고 세부

			정보 Data view 반환
	ChallengeConfigService	getChallengeListDetail	(모달)챌린지 관리 화면에서 전달 받은 값으로 해당 챌린지 세부 정보 Data view 반환
aihometraining.team.challengeConfig.mapper	ChallengeConfigMapper	insertChallengeCategory	챌린지 카테고리 등록 처리
	ChallengeConfigMapper	getChallengeConfigList	챌린지 관리 설정 화면 : 챌린지 카테고리 , 세팅 목록 Data view 반환
	ChallengeConfigMapper	updateChallengeCategory	챌린지 카테고리 수정처리
	ChallengeConfigMapper	deleteChallengeCategory	챌린지 카테고리 삭제처리
	ChallengeConfigMapper	InsertChallengeSetting	챌린지 세팅 등록 처리
	ChallengeConfigMapper	updateChallengeSetting	챌린지 세팅 수정처리
	ChallengeConfigMapper	deleteChallengeSetting	챌린지 세팅 삭제처리
	ChallengeConfigMapper	getChallengeReportList	챌린지 신고 관리 화면 : 챌린지 신고 목록 Data view 반환 - click값 전달 : 신고 세부 내용 보기 모달
	ChallengeConfigMapper	getChallengeList	챌린지 관리 화면 : 챌린지 전체 목록 Data view 반환 -click값 전달 : 챌린지에 세부 정보 보기 모달 - buttonClick 값 전달 : 해당 챌린지에 참여한 참가자에게 포인트 지급
	ChallengeConfigMapper	getChallengeReportDetail	(모달)챌린지 신고 관리 화면에서 전달받은 값으로 해당 신고 세부 정보 Data view 반환
ChallengeConfigMapper	getChallengeListDetail	(모달)챌린지 관리 화면에서 전달 받은 값으로 해당 챌린지 세부 정보 Data view 반환	
aihometraining.team.challengeGather.controller	ChallengeGatherController	challengeGatherInsert	챌린지 모집과 모집 계획 등록 처리
	ChallengeGatherController	challengeGather	챌린지 모집 목록 조회 요청 시 모집 중인 목록 Data View 반환 (메인 화면 이동시에도 모집중인 목록 data view 반환) -click값 전달 : 모집 챌린지 세부 정보 보기 모달
	ChallengeGatherController	getGatherDetail	(모달) 선택한 모집 챌린지 세부 조회 요청 시 해당 챌린지의 세부 정보 data view 반환
	ChallengeGatherController	updateChallengeGather	챌린지 모집 수정처리
	ChallengeGatherController	deleteChallengeGather	챌린지 모집 삭제처리
	ChallengeGatherController	getChallengeGatherAdmin	개설 챌린지 관리 화면 이동시 개설한 챌린지 목록 Data View로 반환 -click값 전달 : 모집 챌린지 세부 정보 보기 모달
	ChallengeGatherController	getChallengeGatherInMember	개설 챌린지 참여 관리 화면 이동시 개설된 챌린지 별 참가 회원 목록, 달성률 조회, 인증 관리 Data View로 반환 - radioCheck 값 전달 : 챌린지 참여자의 인증내용 상세보기 모달
	ChallengeGatherController	getChallengeGaterInMemberDetail	(모달) 참여 관리 화면에서 전달 받은 값으로 해당 회원의 인증 내용 상세보기 Data View 반환

	ChallengeGatherController	getChallengeGatherListDetail	(모달) 모집 목록 화면에서 전달 받은 값으로 해당 챌린지의 세부 정보 Data View로 반환
	ChallengeGatherController	getChallengeGatherAdminDetail	(모달) 개설 챌린지 목록 화면에서 전달 받은 값으로 해당 챌린지의 세부 정보 Data View로 반환
aihometraining.team.challengeGather.service	ChallengeGatherService	insertChallengeGather	챌린지 모집과 모집 계획 등록 처리
	ChallengeGatherService	getChallengeGatherList	챌린지 모집 목록 조회 요청 시 모집 중인 목록 Data View 반환 (메인 화면 이동시에도 모집중인 목록 data view 반환) -click값 전달 : 모집 챌린지 세부 정보 보기 모달
	ChallengeGatherService	updateChallengeGather	챌린지 모집 수정처리
	ChallengeGatherService	deleteChallengeGather	챌린지 모집 삭제처리
	ChallengeGatherService	getChallengeOngoingList	챌린지 목록 조회 요청 시 진행 중인 챌린지 목록 data view 반환 -click값 전달 : 진행되고 있는 챌린지 세부 정보 보기 모달(필요한가?)
	ChallengeGatherService	getChallengeOngoingDetail	(모달) 선택한 진행되고 있는 챌린지 세부 조회 요청 시 해당 챌린지의 세부 정보 data view 반환(필요한가?)
	ChallengeGatherService	getChallengeGatherAdmin	개설 챌린지 관리 화면 이동시 개설한 챌린지 목록 Data View로 반환 -click값 전달 : 모집 챌린지 세부 정보 보기 모달
	ChallengeGatherService	getChallengeGatherInMember	개설 챌린지 참여 관리 화면 이동시 참여한 회원 목록 Data View로 반환 - radioCheck 값 전달 : 챌린지 참여자의 인증내용 상세보기 모달
	ChallengeGatherService	getChallengeGaterInMemberDetail	(모달) 참여 관리 화면에서 전달 받은 값으로 해당 회원의 세부 정보 Data View 반환
	ChallengeGatherService	getChallengeGatherListDetail	(모달) 모집 목록 화면에서 전달 받은 값으로 해당 챌린지의 세부 정보 Data View로 반환
	ChallengeGatherService	getChallengeGatherAdminDetail	(모달) 개설 챌린지 목록 화면에서 전달 받은 값으로 해당 챌린지의 세부 정보 Data View로 반환
aihometraining.team.challengeGather.mapper	ChallengeGatherMapper	getChallengeGatherList	챌린지 모집 목록 조회 요청 시 모집 중인 목록 Data View 반환 (메인 화면 이동시에도 모집중인 목록 data view 반환) -click값 전달 : 모집 챌린지 세부 정보 보기 모달
	ChallengeGatherMapper	updateChallengeGather	챌린지 모집 수정처리
	ChallengeGatherMapper	deleteChallengeGather	챌린지 모집 삭제처리
	ChallengeGatherMapper	getChallengeOngoingList	챌린지 목록 조회 요청 시 진행

			중인 챌린지 목록 data view 반환 -click값 전달 : 진행되고 있는 챌린지 세부 정보 보기 모달(필요한가?)
	ChallengeGatherMapper	getChallengeOngoingDetail	(모달) 선택한 진행되고 있는 챌린지 세부 조회 요청 시 해당 챌린지의 세부 정보 data view 반환(필요한가?)
	ChallengeGatherMapper	getChallengeGatherAdmin	개설 챌린지 관리 화면 이동시 개설한 챌린지 목록 Data View로 반환 -click값 전달 : 모집 챌린지 세부 정보 보기 모달
	ChallengeGatherMapper	getChallengeGatherInMember	개설 챌린지 참여 관리 화면 이동시 참여한 회원 목록 Data View로 반환 - radioCheck 값 전달 : 챌린지 참여자의 인증내용 상세보기 모달
	ChallengeGatherMapper	getChallengeGaterInMemberDetail	(모달) 참여 관리 화면에서 전달 받은 값으로 해당 회원의 세부 정보 Data View 반환
	ChallengeGatherMapper	getChallengeGatherListDetail	(모달) 모집 목록 화면에서 전달 받은 값으로 해당 챌린지의 세부 정보 Data View로 반환
	ChallengeGatherMapper	getChallengeGatherAdminDetail	(모달) 개설 챌린지 목록 화면에서 전달 받은 값으로 해당 챌린지의 세부 정보 Data View로 반환
aihometraining.team.challengeEnter.controller	ChallengeEnterController	challengeEnterList	참가한 챌린지 목록 요청시 참가한 챌린지 전체 목록 data view 반환
	ChallengeEnterController	challengePlanDoInsert	챌린지 실행 내용 등록 처리
	ChallengeEnterController	challengeEnterDetail	선택한 참가 챌린지 세부 정보 화면으로 이동시 해당 챌린지의 세부 정보 data view 반환
	ChallengeEnterController	getChallengeEnterMember	선택한 참가 챌린지 세부 정보 화면으로 이동시 해당 챌린지의 회원 목록 data view 반환
aihometraining.team.challengeEnter.service	ChallengeEnterService	getChallengeEnterList	참가한 챌린지 목록 요청시 참가한 챌린지 전체 목록 data view 반환
	ChallengeEnterService	insertChallengePlanDo	챌린지 실행 내용 등록 처리
	ChallengeEnterService	getChallengeEnterListDetail	선택한 참가 챌린지 세부 정보 화면으로 이동시 해당 챌린지의 세부 정보 data view 반환
	ChallengeEnterService	getChallengeEnterMember	선택한 참가 챌린지 세부 정보 화면으로 이동시 해당 챌린지의 회원 목록 data view 반환
aihometraining.team.challengeEnter.mapper	ChallengeEnterMapper	getChallengeEnterList	참가한 챌린지 목록 요청시 참가한 챌린지 전체 목록 data view 반환
	ChallengeEnterMapper	insertChallengePlanDo	챌린지 실행 내용 등록 처리
	ChallengeEnterMapper	getChallengeEnterListDetail	선택한 참가 챌린지 세부 정보 화면으로 이동시 해당 챌린지의 세부 정보 data view 반환
	ChallengeEnterMapper	getChallengeEnterMember	선택한 참가 챌린지 세부 정보 화

			면으로 이동시 해당 챌린지의 회원 목록 data view 반환
aihometraining.team.controller	MainController	challengeadmin	모집챌린지개설자관리자화면
aihometraining.team.dto	ChallengeCategory		DTO
	ChallengeEnter		DTO
	ChallengeGather		DTO
	ChallengeGatherPlan		DTO
	ChallengeGatherPlanDo		DTO
	ChallengePointGive		DTO
aihometraining.team.workoutlog.controller	WorkoutLogController.java	workoutLogMain	일지 목록 조회
		workoutGoalList	운동 목표 목록 조회
		workoutGoalInsert	운동 목표 등록
		workoutGoalUpdate	운동 목표 수정
		workoutGoalDelete	운동 목표 삭제
		workoutGoalPlanList	운동 계획 목록 조회
		workoutGoalPlanInsert	운동 계획 등록
		workoutGoalPlanUpdate	운동 계획 수정
		workoutGoalPlanDelete	운동 계획 삭제
		workoutLogInsert	일지 등록
		workoutLogUpdate	일지 수정
		workoutLogDelete	일지 삭제
		workoutLogLikeCountUpdate	일지 좋아요(Ajax)
		workoutLogListFeedback	일지상세화면 피드백 위치로 화면 전환
	workoutLogBest	인기 일지 화면	
	geteClassCategoryMedium	운동 계획 카테고리 대-중 분류 (Ajax)	
	geteClassCategorySmall	운동 계획 카테고리 중-소 분류 (Ajax)	
	getworkoutLogPrivacyboundsList	일지 수정 시 공개범위 선택(Ajax)	
	WorkoutLogConfigController.java	workoutCategoryList	운동 계획 카테고리 목록 조회
		workoutCategoryInsert	운동 계획 카테고리 등록
workoutCategoryUpdate		운동 계획 카테고리 수정	
workoutCategoryUpdateModal		운동 계획 카테고리 수정처리 (Ajax)	
workoutCategoryDelete		운동 계획 카테고리 삭제	
workoutAIVideoList		AI 운동 영상 목록 조회	
workoutAIVideoInsert		AI 운동 영상 등록	
workoutAIVideoUpdate		AI 운동 영상 수정	
workoutAIVideoDelete	AI 운동 영상 삭제		
aihometraining.team.workoutlog.service	WorkoutLogUserService.java	getworkoutLogList	일지 목록 조회
		getworkoutGoalList	운동 목표 목록
		workoutGoalInsert	운동 목표 등록
		workoutGoalUpdate	운동 목표 수정
		workoutGoalDelete	운동 목표 삭제
		getWorkoutGoalPlanList	운동 계획 목록
		workoutGoalPlanInsert	운동 계획 등록
		workoutGoalPlanUpdate	운동 계획 수정
		workoutGoalPlanDelete	운동 계획 삭제
		workoutLogInsert	일지 등록
		workoutLogUpdate	일지 수정
		workoutLogDelete	일지 삭제
		workoutLogLikeInsert	일지 좋아요 등록
		workoutLogLikeCountUpdate	일지 좋아요 카운트
	geteClassCategoryLargeList	운동 클래스 카테고리 large 목록 조회	
	WorkoutLogConfigService.java	getWorkoutLogCategoryList	운동 계획 카테고리 목록
		workoutGoalCategoryInsert	운동 계획 카테고리 등록
workoutGoalCategoryUpdate		운동 계획 카테고리 수정	

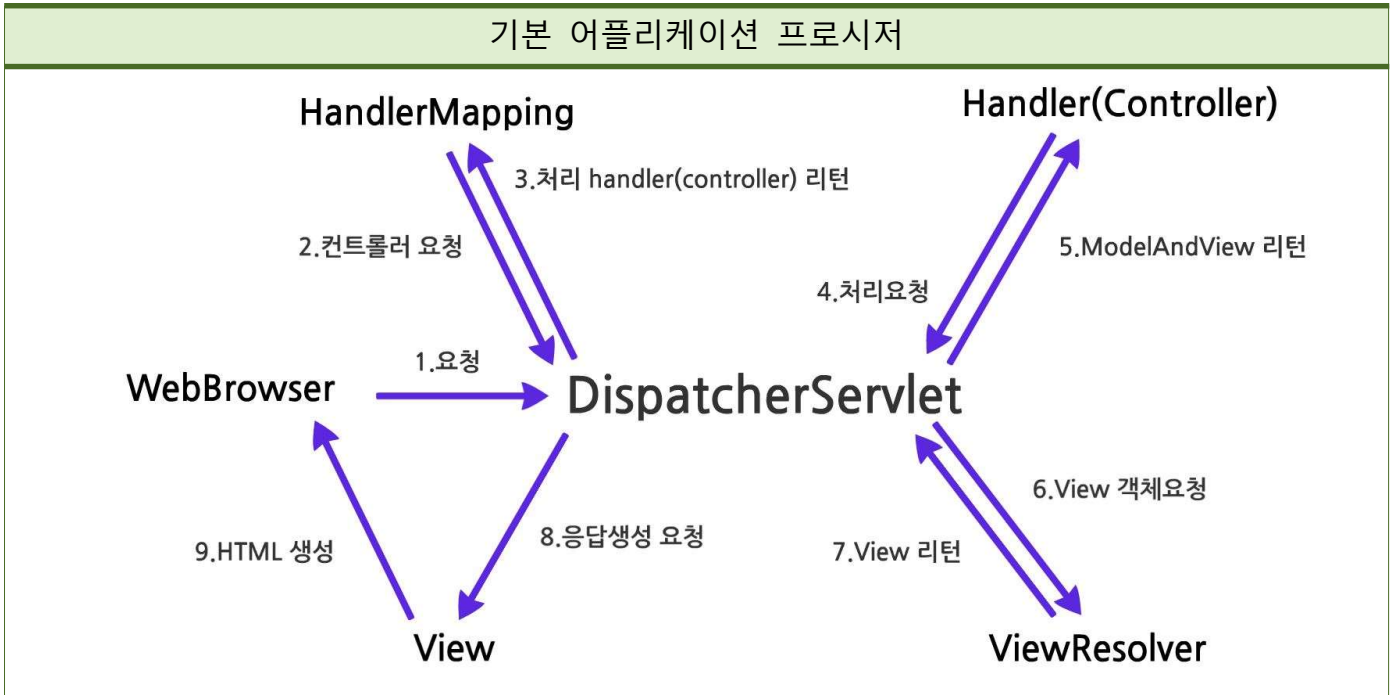
		workoutGoalCategoryDelete	운동 계획 카테고리 삭제
		getWorkoutAIVideoList	AI 운동 영상 목록
		workoutAIVideoInsert	AI 운동 영상 등록
		workoutAIVideoUpdate	AI 운동 영상 수정
		workoutAIVideoDelete	AI 운동 영상 삭제
aihometraining.team.workoutlogconfig.mapper	WorkoutLogUserMapper.java	workoutLogMain	일지 목록 조회
		workoutGoalList	운동 목표 목록 조회
		workoutGoalInsert	운동 목표 등록
		workoutGoalUpdate	운동 목표 수정
		workoutGoalDelete	운동 목표 삭제
		workoutGoalPlanList	운동 계획 목록 조회
		workoutGoalPlanInsert	운동 계획 등록
		workoutGoalPlanUpdate	운동 계획 수정
		workoutGoalPlanDelete	운동 계획 삭제
		workoutLogInsert	일지 등록
		workoutLogUpdate	일지 수정
		workoutLogDelete	일지 삭제
		workoutLogListFeedback	일지상세화면 피드백 위치로 화면 전환
		workoutLogBest	인기 일지 화면
	geteClassCategoryLargeList	운동 계획 카테고리 대 목록 조회	
	WorkoutLogConfigMapper.java	workoutCategoryList	운동 계획 카테고리 목록 조회
		workoutCategoryInsert	운동 계획 카테고리 등록
		workoutCategoryUpdate	운동 계획 카테고리 수정
		workoutCategoryDelete	운동 계획 카테고리 삭제
		workoutAIVideoList	AI 운동 영상 목록 조회
workoutAIVideoInsert		AI 운동 영상 등록	
workoutAIVideoUpdate		AI 운동 영상 수정	
workoutAIVideoDelete		AI 운동 영상 삭제	
aihometraining.team.dto	AIVideo.java		DTO
	WorkoutGoal.java		DTO
	WorkoutGoalPlan.java		DTO
	WorkoutLog.java		DTO
	WorkoutLogCategory.java		DTO
	WorkoutLogLike.java		DTO
	WorkoutLogPrivacybounds.java		DTO

5. 개발 / 서비스 환경

*	구 분	개발 환경		서비스 환경	
S / W	OS	Windows 10		Windows 10	
	웹브라우저	Chrome		Chrome 최적화	
	WAS	apache-tomcat-9.0.27		apache-tomcat-9.0.27	
	개발언어	서버	Java		-
		클라이언트	JavaScript HTML5 CSS thymeleaf jQuery		
	Framework	bootstrap 4 / SpringFramework / Mybatis 3.0		클라이언트	
	DBMS	MySQL5.5.32		MariaDB 10.1	
Tools	JDK Eclipse Spring Tool Suite		OpenJDK 8.x		

6. 어플리케이션

6.1 어플리케이션 프로시저



6.2 서버 및 기타 설정

pom.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.4</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>aihometraining</groupId>
  <artifactId>teamproject</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>teamproject</name>
  <description>teamproject project for Spring Boot</description>
  <properties>
    <java.version>1.8</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jdbc </artifactId>
  
```

```

</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
    <groupId>org.mybatis.spring.boot</groupId>
    <artifactId>mybatis-spring-boot-starter</artifactId>
    <version>2.2.2</version>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
</dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-tomcat</artifactId>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/nz.net.ultraq.thymeleaf/thymeleaf-layout-dialect
-->
<dependency>
    <groupId>nz.net.ultraq.thymeleaf</groupId>
    <artifactId>thymeleaf-layout-dialect</artifactId>
</dependency>
<
    !
-
https://mvnrepository.com/artifact/org.bgee.log4jdbc-log4j2/log4jdbc-log4j2-jdbc4.1 -->
<dependency>
    <groupId>org.bgee.log4jdbc-log4j2</groupId>
    <artifactId>log4jdbc-log4j2-jdbc4.1</artifactId>

```

```

        <version>1.16</version>
    </dependency>

</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>

```

application.properties

```

# 서버 포트 설정
server.port=80

# thymeleaf cache 설정여부 false
spring.thymeleaf.cache=false

#JSESSION 비활성화
server.servlet.session.tracking-modes=cookie
server.servlet.session.cookie.http-only=true

# db 연결정보 jdbc datasource 설정
#spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
#spring.datasource.url=jdbc:mysql://localhost:3306/ksmart42db?serverTimezone=UTC&characterEncoding=UTF8
#spring.datasource.username=*****
#spring.datasource.password=*****

# db 연결정보 log4jdbc datasource 설정
spring.datasource.driver-class-name=net.sf.log4jdbc.sql.jdbcapi.DriverSpy
spring.datasource.url=jdbc:log4jdbc:mysql://ddoggang12.cafe24.com:3306/ddoggang12?serverTimezone=UTC&characterEncoding=UTF8
spring.datasource.username=*****
spring.datasource.password=*****

# mybatis 설정
# mapper xml 파일 위치 경로 설정 ex:)classpath: == src/main/resources/
mybatis.mapper-locations=classpath:mapper/**/*.xml
# mapper dto 설정
mybatis.type-aliases-package=aihometraining.team.dto

```



```
# log 설정파일 위치
```

```
logging.config=classpath:logback-spring.xml
```

```
# spring banner 파일 경로 설정
```

```
spring.banner.location=classpath:templates/banner.txt
```

6.3 주요 기능별 정리

6.3.1 회원가입

회원 가입

이메일

example@email.com

이메일 중복 체크

비밀번호

영문, 숫자, 특수 문자를 혼합하여 8~10자리로 입력해주세요.

비밀번호 확인

위 비밀번호와 동일하게 입력해주세요.

회원 가입

회원 가입

회원 가입이 가능한 화면

■ JavaScript

```
<script src="https://cdn.iampart.kr/js/iampart.payment-1.1.5.js"
type="text/javascript"></script>
<script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>
<th:block layout:fragment="customBodyScript">
  <script type="text/javascript">
    //$(document.ready(function()
    $(function){
      function validationCheck(data){
```

```

        if(typeof data == 'undefined' || data == '' || data == null){
            return false;
        }
        return true;
    }

    $('#emailCheck').click(function(){
        var memberEmail = $('input[name="memberEmail"]').val();
        console.log(memberEmail);
        var vCheck = validationCheck(memberEmail);
        console.log(vCheck);
        if(vCheck){
            $.ajax({
                url: '/emailCheck'
                ,type: 'POST'
                ,data: {'memberEmail' : memberEmail}
                ,dataType: 'json'
                ,success: function(data){
                    console.log(data);
                    //화면에서 맨 처음 이메일을 제외한 모든 입
                    //이메일 중복이 발생되지 않은 경우에 회원
                    //가입을 할 수 있다. 입력하는 요소는 활성화
                    if(data){
                        $('#addMemberForm
                        input').prop('disabled', false);
                        $('#addMemberForm
                        select').prop('disabled', false);
                        $('#addMemberForm
                        button').prop('disabled', false);
                        Swal.fire("사용 가능한 이메일입니
                        다.")
                    }else{
                        //이메일 중복이 발생하는 경우에 회
                        //원 가입을 할 수 없다. 입력하는 요소는 비활성화
                        $('#addMemberForm
                        input').not('input[name="memberEmail"]').prop('disabled', true);
                        $('#addMemberForm
                        button').not('#emailCheck').prop('disabled', true);
                        $('#addMemberForm
                        select').prop('disabled', true);
                        /* var message = '<input value="0|
                        $('#emailInput').append(message);
                        Swal.fire("이미 가입된 이메일입니
                        다.")
                    }
                }
            });
        }

        $('#addMemberBtn').click(function(){
            var isSubmit =true;
            var inputObj = $('#addMemberForm').find('input');
            /* for(var i=0; i < inputObj.length; i++){
                inputObj[i]
                return false;
            } */
            $.each(inputObj, function(){
                var inputData = $(this).val();

```

```

        var result = validationCheck(inputData);

        //사용자가 입력하지 않은 경우
        if(!result){
            alert('필수 항목을 확인해주세요.');
```

\$(this).focus();

isSubmit =false;

return false;

}

});

console.log('유효 여부 폼 전송할건지'+ isSubmit);

if(isSubmit) \$('#addMemberForm').submit();

});

\$.validator.addMethod(

"regex",

function(value, element, regexp) {

var re =new RegExp(regexp);

var res = re.test(value);

//console.log(res, value, regexp, re)

return res;

},

"영문, 숫자, 특수 문자를 혼합하여 8~10자리로 입력해주세요."

);

\$('#addMemberForm').validate({

rules: {

memberEmail: {

required: true,

email: true

},

memberPw: {

required: true,

regex: '^(?=.*[a-zA-Z])(?=.*[0-9])(?=.*[!@%\$^*]).{8,10}\$'

},

memberPwCheck: {

required: true,

equalTo: '#password1'

}

},

messages: {

memberEmail: {

required: "이메일을 입력해주세요"

},

memberPw: {

required: "비밀번호를 입력해주세요."

},

memberPwCheck: {

required: "비밀번호 확인을 입력해주세요."

}

}

});

if(\$('#password1').val() == ""){

\$('#p1').val("영문, 숫자, 특수 문자를 혼합하여 8~10자리로 입력해주세요.");

}

</script>

■ Controller

```
/**
 * 이메일 중복 체크
 * emailCheck ajax
 * @RequestParam(value = "memberEmail") == request.getParameter("memberEmail");
 */
@PostMapping("/emailCheck")
@ResponseBody
public boolean isEmailCheck(@RequestParam(value = "memberEmail") String memberEmail) {
    boolean emailCheck = false;
    log.info("이메일 중복 체크 클릭 시 요청 받은 memberEmail의 값: {}", memberEmail);

    boolean result = memberMapper.isEmailCheck(memberEmail);
    if(result) emailCheck = true;

    log.info("이메일 중복 체크 여부 : {}", result);
    return emailCheck;
}

/* 회원가입 폼 */
@GetMapping("/addMember")
public String addMember(Model model) {

    model.addAttribute("title", "회원가입");

    return "member/addMember";
}

/**
 * 회원 가입 처리
 * /addMember method 방식이 다르기 때문에 주소를 중복해서 사용 가능
 * @param member(회원가입 폼 전송시 요소의 name 속성의 값과 dto의 멤버변수의 이름과 같
으면 자동으로 바인딩하는 객체)
 * 커맨드객체
 * @return redirect: -> request.sendRedirect("")
 */
@PostMapping("/addMember")
public String addMember(Member member, RedirectAttributes reAttr) {
    log.info("회원가입폼에서 입력받은 데이터: {}", member);

    memberService.addMember(member);

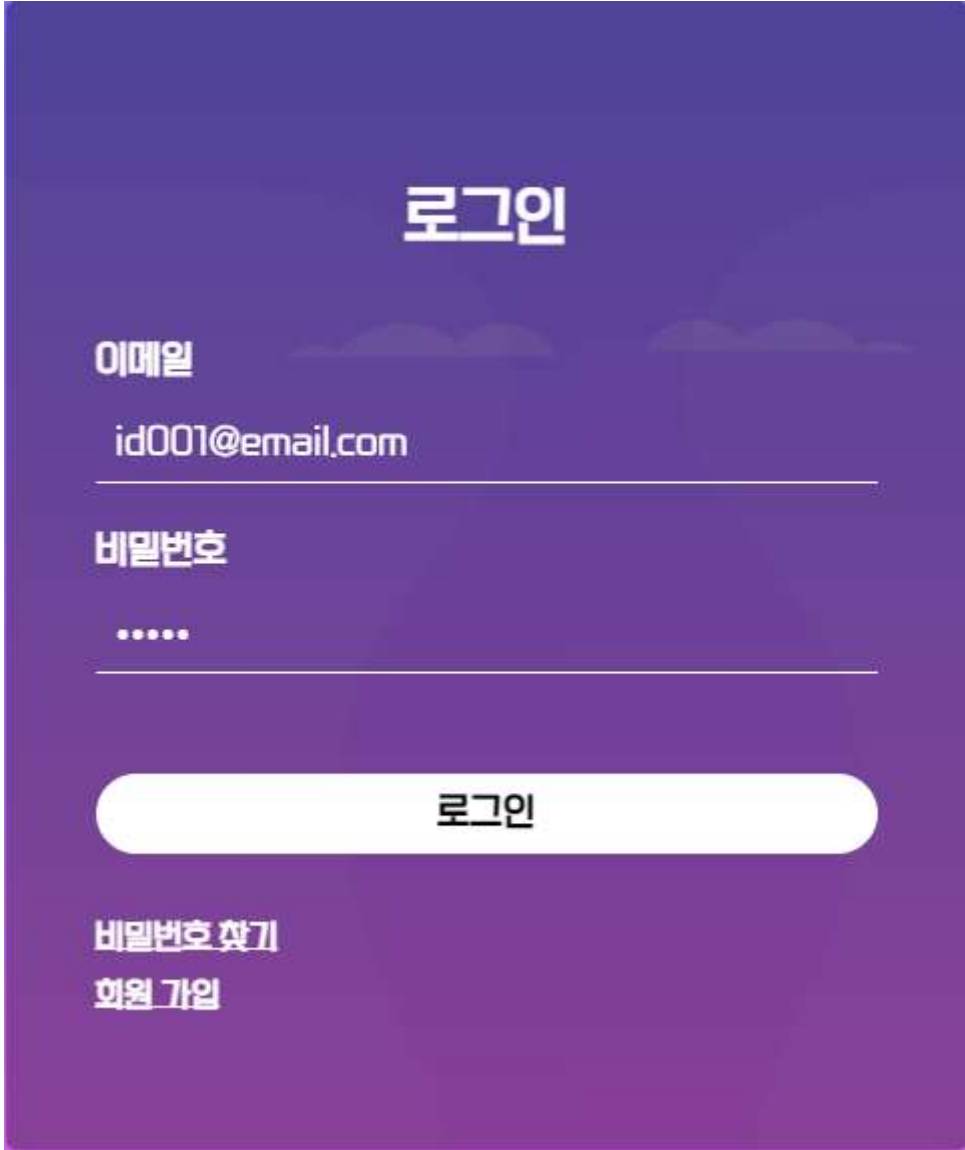
    reAttr.addAttribute("memberEmail", member.getMemberEmail());

    return "redirect:/addInfo";
}
```

■ Service

```
//회원가입
public int addMember(Member member) {
    member.setMemberLevelCode("levelCode003")
    int result = memberMapper.addMember(member)
    return result
}
```

6.3.2 로그인



로그인

로그인이 가능한 화면

■ JavaScript

```
<script src="https://cdn.iampart.kr/js/iampart.payment-1.1.5.js" type="text/javascript"></script>
<script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>
<th:block layout:fragment="customBodyScript">
  <script type="text/javascript">
    $(function(){
      function validationCheck(data){
        if(typeof data =='undefined' || data =='' || data ==null){
          return false;
        }
        return true;
      }
      $('#loginBtn').click(function(){
```

```

        var inputObj = $('#loginForm').find('input');
        var memberEmail = $('input[name="memberEmail"]').val();
        var
        =/^[0-9a-zA-Z]([-_]?[0-9a-zA-Z])*@[0-9a-zA-Z]([-_]?[0-9a-zA-Z]).*[a-zA-Z]{2,3}$/i; // 검증
        에 사용할 정규식 변수 regExp에 저장
        if (memberEmail.match(regExp) !=null) {
            $('#loginForm').submit();

        } else {
            Swal.fire("이메일을 확인해주세요.");
        }

    });

</script>

```

■ Controller

```

/* 로그인 */
/* 관리자 로그인 */
@GetMapping("/login")
public String login(Model model
                    ,@RequestParam(value="result", required =false) String
result) {

    model.addAttribute("title", "회원 로그인");

    if(result !=null) model.addAttribute("result", result);

    return "member/login";
}

/* 관리자 로그인 처리 */
@PostMapping("/login")
public String login(Member member, HttpSession session, RedirectAttributes reAttr) {
    String memberEmail = member.getMemberEmail();
    String memberPw = member.getMemberPw();

    Member checkMember = memberMapper.getMemberInfoByEmail(memberEmail);
    log.info("{} ", checkMember);

    if(checkMember !=null && checkMember.getMemberPw() !=null &&
memberPw.equals(checkMember.getMemberPw())) {
        String sessionLevelCode = checkMember.getMemberLevelCode();
        String sessionName = checkMember.getMemberName();

        String getMaker = challengeGatherService.getChallengeMaker(memberEmail);
        if(getMaker ==null || getMaker == "") {
            getMaker = "권한없음";
        } else {
            getMaker = "권한있음";
        }

        session.setAttribute("SEMAIL", memberEmail);
        session.setAttribute("SLEVEL", sessionLevelCode);
        session.setAttribute("SNAME", sessionName);
        session.setAttribute("SCHALLENGE", getMaker);

        log.info("로그인 성공");
    }
}

```

```

        return "redirect:/";
    }

    reAttr.addAttribute("result", "등록된 회원이 없습니다.");

    return "redirect:/login";
}

```

■ Interceptor

```

@Component
public class LoginInterceptor implements HandlerInterceptor{

    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object
handler)
        throws Exception {

        HttpSession session = request.getSession();
        String requestURI = request.getRequestURI();

        String sessionEmail = (String) session.getAttribute("SEMAIL");

        //인증한 일반 회원은 이 주소로 못 들어가도록 설정
        List<String> authUser = new ArrayList<String>();
        authUser.add("/admin/memberList");
        authUser.add("/admin/memberLevelList");
        authUser.add("/admin/loginHistory");

        //미인증 회원은 이 주소로 못 들어가도록 설정
        List<String> user = new ArrayList<String>();
        user.add("/admin/memberList");
        user.add("/mypage");

        // 로그인 안하면 로그인 페이지로 이동
        if(sessionEmail == null) {
            response.sendRedirect("/login");
            return false;
        }
        // 로그인 했을 때 로그인 한 회원 권한에 따라 주소에 못 들어가도록 설정
        }else {
            String sessionLevelCode = (String) session.getAttribute("SLEVEL");

            // 이메일 인증을 한 일반 회원의 경우 위에 선언한 주소로 못 들어간다.
            if(sessionLevelCode != null && "levelCode002".equals(sessionLevelCode)) {
                if(authUser.contains(requestURI)) {
                    response.sendRedirect("/");
                    return false;
                }
            }
        }
    }
}

```



```

// 이메일 인증을 한 일반 회원의 경우 /admin의 주소는 못 들어간다.
    if(requestURI.indexOf("/admin") > -1) {

        response.sendRedirect("/");
        return false;
    }
// 이메일 미인증 회원의 경우 위에 선언한 주소로 못 들어간다.
}else if(sessionLevelCode != null &&
"levelCode003".equals(sessionLevelCode)) {
    if(user.contains(requestURI)) {
        response.sendRedirect("/");
        return false;
    }

//이메일 미인증 회원의 경우 /admin의 주소는 못 들어간다.
    if(requestURI.indexOf("/admin") > -1) {

        response.sendRedirect("/");
        return false;
    }
//이메일 미인증 회원의 경우 /mypage의 주소는 못 들어간다.
    if(requestURI.indexOf("/mypage") > -1) {

        response.sendRedirect("/");
        return false;
    }

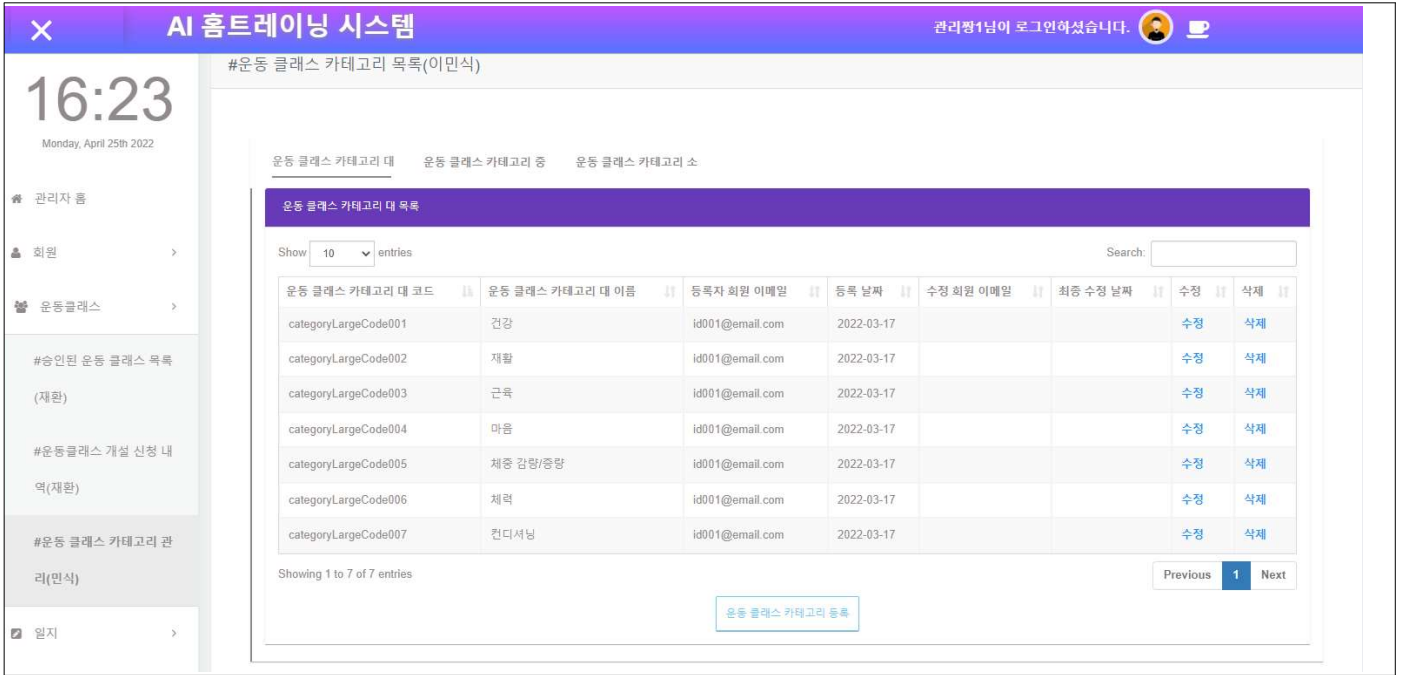
}

}

return true;
}
}

```

6.3.3 운동 클래스 카테고리 조회



운동 클래스 카테고리 조회

■ javascript

```

<!-- start: Javascript -->
<th:block layout:fragment="customBodyScript">

    <!-- plugins -->
    <script th:src="@{/asset/js/plugins/moment.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.dataTables.min.js}"></script>
    <script th:src="@{/asset/js/plugins/dataTables.bootstrap.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.nicescroll.js}"></script>

    <!-- custom -->
    <script th:src="@{/asset/js/main.js}"></script>
    <script type="text/javascript">
    $(document).ready(function(){
        $('#eClassCategoryLargeList').DataTable();
    });

    </script>
    <script type="text/javascript">
    $(document).ready(function(){
        $('#eClassCategoryMediumList').DataTable();
    });

    </script>
    <script type="text/javascript">
    $(document).ready(function(){
        $('#eClassCategorySmallList').DataTable();
    });
    
```

```
</script>
</th:block>
<!-- end: Javascript -->
```

■ Controller

```
/*
 * 운동 클래스 카테고리 목록 조회
 * */

@GetMapping("/eClassCategoryList")
public String eClassCategoryList(Model model
                                ,@RequestParam(value="searchKey",
required =false) String searchKey
                                ,@RequestParam(value="searchValue",
required =false) String searchValue) {

    log.info("운동 클래스 카테고리 목록 요청");
    log.info("searchValue:{}", searchValue);

    if(searchKey !=null) {
        if("eClassCategoryLargeCode".equals(searchKey)) {
            searchKey = "eClassCategoryLargeCode";
        }else if("eClassCategoryLargeName".equals(searchKey)) {
            searchKey = "eClassCategoryLargeName";
        }
    }
    if(searchKey !=null) {
        if("eClassCategoryMediumCode".equals(searchKey)) {
            searchKey = "eClassCategoryMediumCode";
        }else if("eClassCategoryMediumName".equals(searchKey)) {
            searchKey = "eClassCategoryMediumName";
        }
    }
    if(searchKey !=null) {
        if("eClassCategorySmallCode".equals(searchKey)) {
            searchKey = "eClassCategorySmallCode";
        }else if("eClassCategorySmallName".equals(searchKey)) {
            searchKey = "eClassCategorySmallName";
        }
    }

    List<EClassCategoryLarge> eClassCategoryLargeList =
eClassCategoryService.getEClassCategoryLargeList(searchKey, searchValue);
    List<EClassCategoryMedium> eClassCategoryMediumList =
eClassCategoryService.getEClassCategoryMediumList(searchKey, searchValue);
    List<EClassCategorySmall> eClassCategorySmallList =
eClassCategoryService.getEClassCategorySmallList(searchKey, searchValue);
    model.addAttribute("leftMenuList", "운동 클래스");
    model.addAttribute("title", "운동 클래스 카테고리 관리");
    model.addAttribute("eClassCategoryLargeList", eClassCategoryLargeList);
    model.addAttribute("eClassCategoryMediumList", eClassCategoryMediumList);
    model.addAttribute("eClassCategorySmallList", eClassCategorySmallList);

    return "eClass/eClassConfig/eClassCategoryList";
}
```

■ Service

```
//운동 클래스 카테고리 대 목록 조회
public List<EClassCategoryLarge> getEClassCategoryLargeList(String searchKey,
    String searchValue) {

    List<EClassCategoryLarge>          eClassCategoryLargeList          =
eClassCategoryMapper.getEClassCategoryLargeList(searchKey, searchValue);

    return eClassCategoryLargeList;
}
//운동 클래스 카테고리 중 목록 조회
public List<EClassCategoryMedium> getEClassCategoryMediumList(String searchKey,
    String searchValue) {

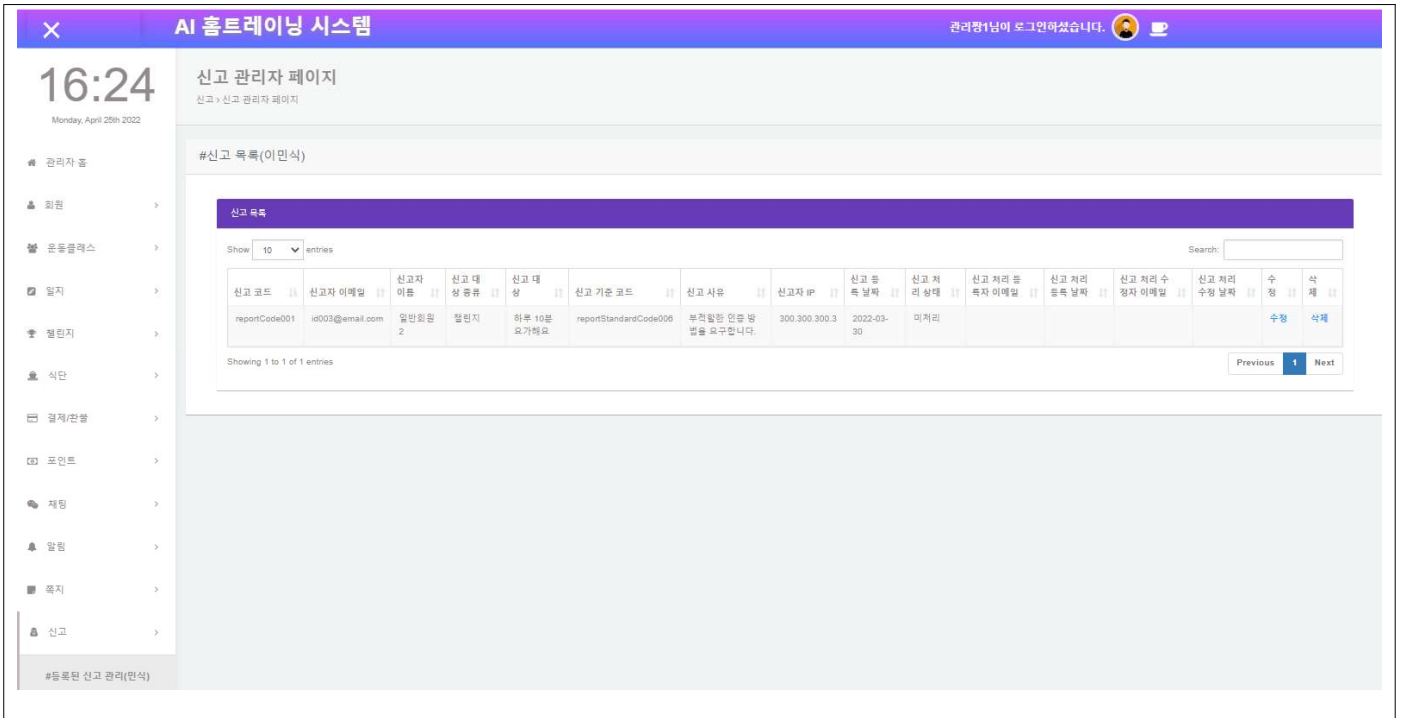
    List<EClassCategoryMedium>          eClassCategoryMediumList          =
eClassCategoryMapper.getEClassCategoryMediumList(searchKey, searchValue);

    return eClassCategoryMediumList;
}
//운동 클래스 카테고리 소 목록 조회
public List<EClassCategorySmall> getEClassCategorySmallList(String searchKey,
    String searchValue) {

    List<EClassCategorySmall>          eClassCategorySmallList          =
eClassCategoryMapper.getEClassCategorySmallList(searchKey, searchValue);

    return eClassCategorySmallList;
}
```

6.3.4 신고 목록 조회



신고 목록 조회

등록된 신고를 조회하는 화면

■ javascript

```

<!-- start: Javascript -->
<th:block layout:fragment="customBodyScript">

    <!-- plugins -->
    <script th:src="@{/asset/js/plugins/moment.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.dataTables.min.js}"></script>
    <script th:src="@{/asset/js/plugins/dataTables.bootstrap.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.nicescroll.js}"></script>

    <!-- custom -->
    <script th:src="@{/asset/js/main.js}"></script>
    <script type="text/javascript">
        $(document).ready(function(){
            $('#reportList').DataTable();
        });
    </script>
</th:block>
<!-- end: Javascript -->
    
```

■ Controller

```

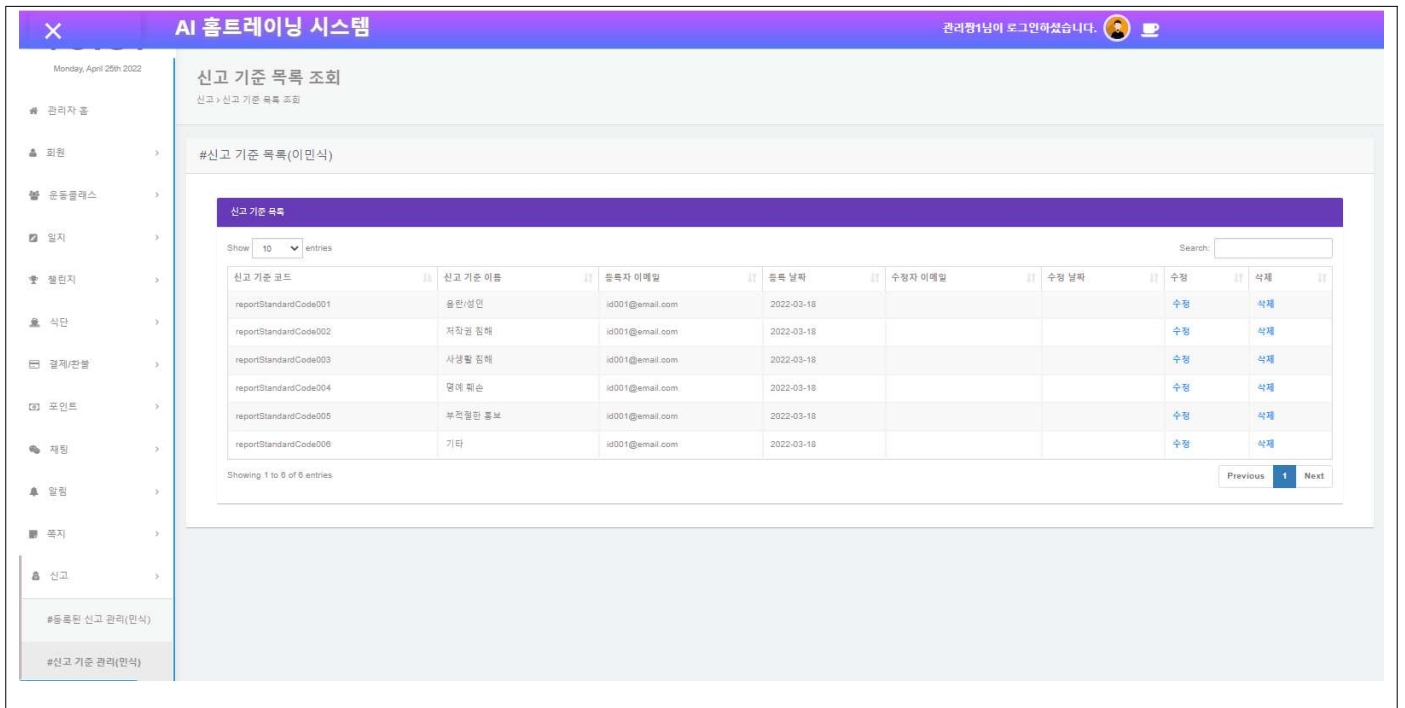
/*신고목록조회*/
@GetMapping("/reportList")
public String reportList(Model model) {
    
```

```
List<Report> reportList = reportService.getReportList()
model.addAttribute("leftMenuList", "신고")
model.addAttribute("title", "신고관리자페이지")
model.addAttribute("reportList", reportList)
return "report/reportList"
}
```

■ Service

```
public List<Report> getReportList(){
    List<Report> reportList = reportMapper.getReportList();
    return reportList;
}
```

6.3.5 신고 기준 목록 조회



신고 기준 목록 조회

등록된 신고 기준 목록을 조회하는 화면

■ JavaScript

```

<!-- start: Javascript -->
<th:block layout:fragment="customBodyScript">

    <!-- plugins -->
    <script th:src="@{/asset/js/plugins/moment.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.dataTables.min.js}"></script>
    <script th:src="@{/asset/js/plugins/dataTables.bootstrap.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.nicescroll.js}"></script>

    <!-- custom -->
    <script th:src="@{/asset/js/main.js}"></script>
    <script type="text/javascript">
        $(document).ready(function(){
            $('#reportStandardList').DataTable();
        });
    </script>
</th:block>
<!-- end: Javascript -->
    
```

■ Controller

```

/*신고 기준 목록 조회*/
@GetMapping("/reportStandardList")
    
```

```

public String reportStandardList(Model model) {

    List<ReportStandard> reportStandardList = reportService.getReportStandardList();

    model.addAttribute("leftMenuList", "신고");
    model.addAttribute("title", "신고 기준 목록 조회");
    model.addAttribute("reportStandardList", reportStandardList);

    return "report/reportStandardList";

}

```

■ Service

```

/**
 * 신고 기준 목록 조회
 */
public List<ReportStandard> getReportStandardList(){

    List<ReportStandard> reportStandardList = reportMapper.getReportStandardList();

    return reportStandardList;

}

```


6.3.6 클래스 신청 폼

소개하기

카테고리를 선택해주세요

운동 카테고리 선택을 선택해주세요

▶

▶

내용 작성은 필수로 작성해주세요

작가 이름을 등록해주세요

작성 완료

신청서 제출하기

섹션 정하기

내용 제목을 입력해주세요

작성하기

최소 100자 이상을 입력해주세요

최대 1000자까지 입력 가능

최소 100자 이상을 입력해주세요

최대 1000자까지 입력 가능

클래스에 대한 질문 및 답변

질문 또는 답변을 등록해주세요

작성하기

답변을 등록해주세요

작성하기

세부 설정

수강생 모집 시작 날짜

연도: 월: 일

수강생 모집 종료 날짜

연도: 월: 일

참가 인원

최소

1

명

최대

10

명

시간 및 금액 설정

1

시간

금액 설정

1

원

클래스 이름을 정해주세요

계속 신청하기

운동 클래스 개설 신청 폼

운동 클래스를 개설하기 위한 내용을 담은 폼입니다.
 카테고리 대 선택시 eClassCategoryLargeCode를 컨트롤러에서 받아와 AJAX로 중분류 소분류를 반환해줍니다

```

$(document).on('change', "#eClassCategoryLargeCode", function(e) {
    var eClassCategoryLargeCode = $('select[name="eClassCategoryLargeCode"]').val()
    console.log(eClassCategoryLargeCode)
    $.ajax({
        url : '/CategoryLarge',
        type : 'post',
        data : {'eClassCategoryLargeCode' : eClassCategoryLargeCode},
        dataType : 'json',
        success : function(data) {
            console.log(JSON.stringify(data))
            console.log(data.length)
            var txt1='<option value="none">운동카테고리중분류선택하세요</option>'
            for(var i =0i<data.lengthi++){
                txt1 +=<option value="" +data[i].eClassCategoryMediumCode+"" ' ' +>'+
                    data[i].eClassCategoryMediumName +</option>" }
                console.log(txt1)
                $('#eClassCategoryMediumCode').empty()
                $('#eClassCategoryMediumCode').append(txt1)
                $('#eClassCategoryMediumCode').niceSelect('update')
            }
        }
    })
});

$(document).on('change', "#eClassCategoryMediumCode", function(e) {
    var eClassCategoryMediumCode = $('select[name="eClassCategoryMediumCode"]').val()
    console.log(eClassCategoryMediumCode)
    $.ajax({
        url : '/CategoryMedium',
        type : 'post',
        data : {'eClassCategoryMediumCode' : eClassCategoryMediumCode},
        dataType : 'json',
        success : function(data) {
            console.log(JSON.stringify(data))
            console.log(data.length)
            var txt1='<option value="none">운동카테고리소분류선택하세요</option>'
            for(var i =0i<data.lengthi++){
                txt1 +=<option value="" +data[i].eClassCategorySmallCode+"" ' ' + data-cate="" +
                    data[i].eClassCategorySmallName +>'+ data[i].eClassCategorySmallName
                    +</option>" }
                console.log(txt1)
                $('#eClassCategorySmallCode').empty()
                $('#eClassCategorySmallCode').append(txt1)
                $('#eClassCategorySmallCode').niceSelect('update')
            }
        }
    })
});

```

■ Controller

```

@PostMapping("/CategoryLarge")
@ResponseBody
public List<EClassCategorySmall> EClassLarge(@RequestParam(value="eClassCategoryLargeCode", required
=false) String eClassCategoryLargeCode){
    log.info("eClassController EClassLarge 데이터: {}", eClassCategoryLargeCode)
    List<EClassCategorySmall> categoryLarge = eClassMapper.eClassCategoryLarge(eClassCategoryLargeCode)
    return categoryLarge
}

```

```

@PostMapping("/CategoryMedium")
@ResponseBody
public List<EClassCategorySmall> EClassMedium (@RequestParam(value = "eClassCategoryMediumCode",
required =false) String eClassCategoryMediumCode){
List<EClassCategorySmall> categoryMedium =
eClassMapper.eClassCategoryMedium(eClassCategoryMediumCode)
return categoryMedium
}
@GetMapping("/openAppley")
public String EClassOpenAppleyFormInsert( EClassCategorySmall eClassCategorySmall
, EClassIntroduce eClassIntroduce
, EClassSectionTitle eClassSectionTitle
, EClassSectionCurriculum eClassSectionCurriculum
, EClassQuestion eClassQuestion
, EClassAnswer eClassAnswer
, EClassApproved eClassApproved
, HttpSession session) {
String mamberEmail = (String) session.getAttribute("SEMAIL")
eClassService.EClassIntroduceInsert(eClassIntroduce, mamberEmail)
eClassService.EClassSectionTitleInsert(eClassSectionTitle, mamberEmail)
eClassService.EClassSectionCurriculumInsert(eClassSectionCurriculum, mamberEmail, eClassSectionTitle)
eClassService.EClassQuestionInsert(eClassQuestion, mamberEmail)
eClassService.EClassAnswerInsert(eClassAnswer, mamberEmail, eClassQuestion)
eClassService.EClassPriceInsert( eClassApproved
, eClassCategorySmall
, mamberEmail
, eClassIntroduce
, eClassSectionTitle
, eClassSectionCurriculum
, eClassQuestion
, eClassAnswer)
log.info("운동클래스신청폼에서입력받은데이터: {}",eClassCategorySmall)
log.info("운동클래스신청폼에서입력받은데이터: {}",eClassIntroduce)
log.info("운동클래스신청폼에서입력받은데이터: {}",eClassSectionTitle)
log.info("운동클래스신청폼에서입력받은데이터: {}",eClassSectionCurriculum)
log.info("운동클래스신청폼에서입력받은데이터: {}",eClassQuestion)
log.info("운동클래스신청폼에서입력받은데이터: {}",eClassAnswer)
return "redirect:/eClassOpenAppleyComplete"
}

```

■ Service

```

//클래스카테고리조회
public List<EClassCategorySmall> eClassCategoryLargeList() {
List<EClassCategorySmall> eClassCategoryLargeList = eClassMapper.eClassCategoryLargeList()
return eClassCategoryLargeList
}
public List<EClassCategorySmall> eClassCategoryMediumList() {
List<EClassCategorySmall> eClassCategoryMediumList = eClassMapper.eClassCategoryMediumList()
return eClassCategoryMediumList
}
public List<EClassApproved> MyApplyList(String eClassOpenAppleyMemberEmail){
List<EClassApproved> eClassOpenApply =
eClassMapper.eClassOpenAppleyList(eClassOpenAppleyMemberEmail)
return eClassOpenApply
}
//클래스신청폼introduce 등록처리
public int EClassIntroduceInsert(EClassIntroduce eClassIntroduce, String mamberEmail) {

```

```

log.info("EClassIntroduceInsert eClassIntroduce 데이터: {}", eClassIntroduce)
String introduceCode = commonMapper.getNewCode("eClassIntroduceCode", "eClassIntroduce")
log.info("eClassService EClassIntroduceInsert introduceCode: {}",introduceCode)
eClassIntroduce.seteClassIntroduceCode(introduceCode)
eClassIntroduce.setMemberEmail(memberEmail)
int result = eClassMapper.EClassIntroduceInsert(eClassIntroduce)
return result
}
//클래스신청폼sectiontitle 등록처리
public int EClassSectionTitleInsert(EClassSectionTitle eClassSectionTitle, String memberEmail) {
log.info("EClassSectionTitleInsert eClassSectionTitle 데이터: {}", eClassSectionTitle)
String sectiontitleCode = commonMapper.getNewCode("eClassSectionTitleCode", "eClassSectionTitle")
log.info("eClassService EClassSectionTitleInsert eClassSectionTitle", sectiontitleCode)
eClassSectionTitle.seteClassSectionTitleCode(sectiontitleCode)
eClassSectionTitle.setMemberEmail(memberEmail)
int result = eClassMapper.EClassSectionTitleInsert(eClassSectionTitle)
return result
}
//클래스신청폼sectionculum 등록처리
public int EClassSectionCurriculumInsert( EClassSectionCurriculum eClassSectionCurriculum
, String memberEmail
, EClassSectionTitle eClassSectionTitle) {
log.info("EClassSectionCurriculumInsert eClassSectionCurriculum 데이터: {}", eClassSectionCurriculum)
String curriculumCode = commonMapper.getNewCode("eClassSectionCurriculumCode",
"eClassSectionCurriculum")
String titleCode = eClassSectionTitle.geteClassSectionTitleCode()
log.info("eClassService EClassSectionCurriculumInsert eClassSectionCurriculum", curriculumCode)
eClassSectionCurriculum.seteClassSectionCurriculumCode(curriculumCode)
eClassSectionCurriculum.setMemberEmail(memberEmail)
eClassSectionCurriculum.seteClassSectionTitleCode(titleCode)
int result = eClassMapper.EClassSectionCurriculumInsert(eClassSectionCurriculum)
return result
}
//클래스신청폼question 등록처리
public int EClassQuestionInsert(EClassQuestion eClassQuestion, String memberEmail) {
log.info("EClassQuestionInsert eClassQuestion 데이터: {}", eClassQuestion)
String questionCode = commonMapper.getNewCode("eClassQuestionCode", "eClassQuestion")
log.info("eClassService EClassQuestionInsert eClassQuestion", questionCode)
eClassQuestion.seteClassQuestionCode(questionCode)
eClassQuestion.setMemberEmail(memberEmail)
int result = eClassMapper.EClassQuestionInsert(eClassQuestion)
return result
}
//클래스신청폼answer 등록처리
public int EClassAnswerInsert(EClassAnswer eClassAnswer
, String memberEmail
, EClassQuestion eClassQuestion) {
log.info("EClassAnswerInsert EClassAnswer 데이터: {}", eClassAnswer)
String answerCode = commonMapper.getNewCode("eClassAnswerCode", "eClassAnswer")
String questionCode = eClassQuestion.geteClassQuestionCode()
log.info("eClassService EClassQuestionInsert eClassQuestion", answerCode)
eClassAnswer.seteClassAnswerCode(answerCode)
eClassAnswer.setMemberEmail(memberEmail)
eClassAnswer.seteClassQuestionCode(questionCode)
int result = eClassMapper.EClassAnswerInsert(eClassAnswer)
return result
}
//클래스신청폼price 등록처리
public int EClassPriceInsert( EClassApproved eClassApproved
, EClassCategorySmall eClassCategorySmall
, String memberEmail
, EClassIntroduce eClassIntroduce

```

```

, EClassSectionTitle eClassSectionTitle
, EClassSectionCurriculum eClassSectionCurriculum
, EClassQuestion eClassQuestion
, EClassAnswer eClassAnswer) {
String priceCode = commonMapper.getNewCode("eClassApprovedCode", "eClassApproved")
log.info("eClassService EClassQuestionInsert eClassQuestion", priceCode)
eClassApproved.seteClassApprovedCode(priceCode)
eClassApproved.setMemberEmail(memberEmail)
String categoryCode = eClassCategorySmall.geteClassCategorySmallCode()
eClassApproved.seteClassCategorySmallCode(categoryCode)
String introduceCode = eClassIntroduce.geteClassIntroduceCode()
eClassApproved.seteClassIntroduceCode(introduceCode)
String sectionCode = eClassSectionTitle.geteClassSectionTitleCode()
eClassApproved.seteClassSectionTitleCode(sectionCode)
String curriculumCode = eClassSectionCurriculum.geteClassSectionCurriculumCode()
eClassApproved.seteClassSectionCurriculumCode(curriculumCode)
String questionCode = eClassQuestion.geteClassQuestionCode()
eClassApproved.seteClassQuestionCode(questionCode)
String answerCode = eClassAnswer.geteClassAnswerCode()
eClassApproved.seteClassAnswerCode(answerCode)
log.info("eClassService EClassPriceInsert eClassApproved 데이터: {}", eClassApproved)
log.info("eClassService EClassPriceInsert memberEmail 데이터: {}", memberEmail)
log.info("eClassService EClassPriceInsert priceCode 데이터: {}", priceCode)
log.info("eClassService EClassPriceInsert introduceCode 데이터: {}", introduceCode)
log.info("eClassService EClassPriceInsert sectionCode 데이터: {}", sectionCode)
log.info("eClassService EClassPriceInsert curriculumCode 데이터: {}", curriculumCode)
log.info("eClassService EClassPriceInsert questionCode 데이터: {}", questionCode)
log.info("eClassService EClassPriceInsert answerCode 데이터: {}", answerCode)
int result = eClassMapper.EClassPriceInsert(eClassApproved)
return result
}

```

6.3.7 등록 신청폼 수정

The screenshot shows a web form titled '수정하기' (Modify) for editing a class registration. The form is organized into several sections:

- 소개하기 (Introduction):** Contains a large text area for '클래스 소개를 소개해주세요' (Please introduce the class) and a smaller text area for '특기사항을 소개해주세요' (Please introduce special features). There are '등록 취소하기' (Cancel registration) and '수정하기' (Modify) buttons.
- 섹션 정하기 (Section Setting):** Includes a '수강 목적을 선택하세요' (Select your purpose of taking the course) dropdown, a '수강료 및 수업료' (Course fee and tuition) input field, and a '수강료 및 수업료' (Course fee and tuition) dropdown.
- 클래스에 대한 질문 및 답변 (Questions and Answers about the Class):** Features a '질문 또는 답변을 등록해주세요' (Please register a question or answer) text area, a '등록 취소하기' (Cancel registration) button, and a '등록하기' (Register) button.
- 세부 설정 (Detailed Settings):** Contains several dropdown menus for '수강생 모집 시작 날짜' (Start date of student recruitment), '수강생 모집 종료 날짜' (End date of student recruitment), '참가 인원' (Number of participants), '수강료 범위' (Course fee range), and '수업 범위' (Classroom range).
- 클래스 이름을 정해주세요 (Please specify the class name):** A large text area for entering the class name, with a '수정 취소하기' (Cancel registration) button below it.

등록 수정 폼

신청한 클래스를 수정하는 폼입니다

승인이 되면 해당 화면으로 이동할수 없고 개설 승인된 클래스 상세페이지 또는 마이페이지에서 승인된 클래스를 수정해야합니다.

■ Controller

```
@PostMapping("/openAppleyUpdate")
public String EClassAppleyFormUpdate( EClassCategorySmall eClassCategorySmall
, EClassIntroduce eClassIntroduce
, EClassSectionTitle eClassSectionTitle
, EClassSectionCurriculum eClassSectionCurriculum
, EClassQuestion eClassQuestion
, EClassAnswer eClassAnswer
, EClassApproved eClassApproved
, HttpSession session) {
String mamberEmail = (String) session.getAttribute("SEMAIL")
eClassService.EClassIntroduceUpdate(eClassIntroduce, mamberEmail)
eClassService.EClassSectionTitleUpdate(eClassSectionTitle, mamberEmail)
eClassService.EClassSectionCurriculumUpdate(eClassSectionCurriculum, mamberEmail)
```

```

eClassService.EClassQuestionUpdate(eClassQuestion, memberEmail)
eClassService.EClassAnswerInsert(eClassAnswer, memberEmail, eClassQuestion)
eClassService.EClassPriceInsert(eClassApproved
, eClassCategorySmall
, memberEmail
, eClassIntroduce
, eClassSectionTitle
, eClassSectionCurriculum
, eClassQuestion
, eClassAnswer)
log.info("운동클래스신청폼에서입력받은데이터: {}",eClassCategorySmall)
log.info("운동클래스신청폼에서입력받은데이터: {}",eClassIntroduce)
log.info("운동클래스신청폼에서입력받은데이터: {}",eClassSectionTitle)
log.info("운동클래스신청폼에서입력받은데이터: {}",eClassSectionCurriculum)
log.info("운동클래스신청폼에서입력받은데이터: {}",eClassQuestion)
log.info("운동클래스신청폼에서입력받은데이터: {}",eClassAnswer)
return "redirect:eClass/eClassOpenAppleyComplete"
}

```

■ Service

```

//클래스신청폼price 수정처리
public int EClassPriceUpdate( EClassApproved eClassApproved
, EClassCategorySmall eClassCategorySmall
, String memberEmail
, EClassIntroduce eClassIntroduce
, EClassSectionTitle eClassSectionTitle
, EClassSectionCurriculum eClassSectionCurriculum
, EClassQuestion eClassQuestion
, EClassAnswer eClassAnswer) {
log.info("eClassService EClassPriceUpdate setMemberUpdateEmail : {}",memberEmail)
int result = eClassMapper.EClassPriceInsert(eClassApproved)
return result
}
//클래스신청폼answer 수정처리
public int EClassAnswerUpdate(EClassAnswer eClassAnswer
, String memberEmail) {
log.info("EClassAnswerInsert EClassAnswer 데이터: {}", eClassAnswer)
eClassAnswer.setMemberUpdateEmail(memberEmail)
int result = eClassMapper.EClassAnswerInsert(eClassAnswer)
return result
}
//클래스신청폼question 수정처리
public int EClassQuestionUpdate(EClassQuestion eClassQuestion, String memberEmail) {
log.info("EClassQuestionInsert eClassQuestion 데이터: {}", eClassQuestion)
eClassQuestion.setMemberUpdateEmail(memberEmail)
int result = eClassMapper.EClassQuestionUpdate(eClassQuestion)
return result
}
//클래스신청폼sectionculum 수정처리
public int EClassSectionCurriculumUpdate( EClassSectionCurriculum eClassSectionCurriculum
, String memberEmail) {
log.info("EClassSectionCurriculumInsert eClassSectionCurriculum 데이터: {}", eClassSectionCurriculum)
eClassSectionCurriculum.setMemberUpdateEmail(memberEmail)
int result = eClassMapper.EClassSectionCurriculumInsert(eClassSectionCurriculum)
return result
}
//클래스신청폼sectiontitle 수정처리
public int EClassSectionTitleUpdate(EClassSectionTitle eClassSectionTitle, String memberEmail) {
log.info("EClassSectionTitleInsert eClassSectionTitle 데이터: {}", eClassSectionTitle)
eClassSectionTitle.setMemberUpdateEmail(memberEmail)
int result = eClassMapper.EClassSectionTitleInsert(eClassSectionTitle)
}

```

```
return result
}
//클래스신청폼introduce 수정처리
public int EClassIntroduceUpdate(EClassIntroduce eClassIntroduce, String memberEmail) {
log.info("EClassIntroduceInsert eClassIntroduce 데이터: {}", eClassIntroduce)
eClassIntroduce.setMemberUpdateEmail(memberEmail)
int result = eClassMapper.EClassIntroduceInsert(eClassIntroduce)
return result
}
//클래스신청수정화면
public EClassApproved eClassApprovedByCode(String eclassapprovedCode){
EClassApproved eClassApproved = eClassMapper.eClassApprovedByCode(eclassapprovedCode)
log.info("eClassService eClassApprovedList eClassApproved : {}", eclassapprovedCode)
return eClassApproved
}
```


6.3.8 수강신청

<p>수강 정보</p> <p>운동클래스 이름 다이어트댄스</p> <p>강사 이름 일반회원2</p> <p>커리큘럼 기간 3개월</p> <hr/> <p>이름 일반회원3</p> <p>이메일 id004@email.com</p> <p>휴대폰 번호 <input type="text"/></p>	<p>가격 정보 *기간내 미 결제시, 자동으로 수강 신청이 취소됩니다.</p> <p>클래스 금액 45000원 (15000원 x 3개월)</p> <p>월 정기 결제 금액 15000</p> <div style="text-align: center; border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>수강신청서 제출</p> </div>
--	---

수강신청

수강 신청 화면

■ JavaScript

```

<script type="text/javascript">
    $(function(){
        //휴대폰번호 유효성 검사
        $('#eClassTakeForm').validate({
            rules: {
                memberPhone: {
                    required: true,
                    digits : true,
                    minlength : 11,
                    maxlength : 11,
                }
            },
            messages: {
                memberPhone: {
                    required: "*휴대폰 번호를 입력해주세요",
                    digits : "숫자만 입력 해주세요",
                    minlength : "휴대폰 번호 11자리를 입력해주세요",
                    maxlength : "휴대폰 번호 11자리를 입력해주세요"
                }
            }
        });
        //수강신청 버튼 클릭 이벤트 - 수강신청 처리
        $('#eClassTakeBtn').click(function(){
            var result =confirm('수강신청 하시겠습니까?');
            if(result ==true){
                $('#eClassTakeForm').submit();
            }
        });
    });

```

```
</script>
```

■ Controller

```
/*DI 의존성 주입*/
private PaymentService paymentService;
private PaymentMapper paymentMapper;
private ChallengeEnterMapper challengeEnterMapper;
private MemberMapper memberMapper;
public PaymentController(PaymentService paymentService
    ,PaymentMapper paymentMapper
    ,ChallengeEnterMapper challengeEnterMapper
    ,MemberMapper memberMapper) {
    this.paymentService = paymentService;
    this.paymentMapper = paymentMapper;
    this.challengeEnterMapper = challengeEnterMapper;
    this.memberMapper = memberMapper;
}
//수강신청 화면
@GetMapping("/signUpForClass")
public String eClassTake(Model model , HttpServletRequest request
    ,@RequestParam(name="eClassCode") String eClassCode
    ,@RequestParam(name="memberEmail") String memberEmail) {

    Member member = paymentService.getEClassTakeMember(memberEmail);
    EClassApproved eClassApproved = paymentService.getEClassApproved(eClassCode);

    model.addAttribute("title", "수강신청");
    model.addAttribute("member", member);
    model.addAttribute("eClassApproved", eClassApproved);

    return "eClass/eClassTake";
}
//수강신청 처리
@PostMapping("/signUpForClass")
public String eClassTake(EClassTake eClassTake, RedirectAttributes reAttr) {

    paymentService.addEClassTake(eClassTake);
    reAttr.addAttribute("paymentGroupCode", eClassTake.getPaymentGroupCode());

    //위시리스트 삭제 처리

    return "redirect:/payment";
}
```

■ Service

```
private PaymentMapper paymentMapper;
@Autowired
public PaymentService(PaymentMapper paymentMapper) {
    this.paymentMapper = paymentMapper;
}
//수강신청 멤버 정보
public Member getEClassTakeMember(String memberEmail) {
    return paymentMapper.getEClassTakeMember(memberEmail);
}
//수강신청 운동클래스 정보
```

```

public EClassApproved getEClassApproved(String eClassCode) {
    return paymentMapper.getEClassApproved(eClassCode);
}
//수강신청
public void addEClassTake(EClassTake eClassTake) {
    String eClassApprovedCode = eClassTake.getEClassApprovedCode();

    int len = eClassTake.getMemberEmail().indexOf("@");
    String memberId = eClassTake.getMemberEmail().substring(0, len);

    String eClassTakeCode = eClassApprovedCode+"_"+memberId;

    SimpleDateFormat date =new SimpleDateFormat("yyyyMMddhhmm");

    String paymentGroupCode ="e"+date.format(new Date())+"_"+memberId;

    //dto에 위에서 만들어진 코드를 세팅해주기
    eClassTake.setEClassTakeCode(eClassTakeCode);
    eClassTake.setPaymentGroupCode(paymentGroupCode);

    paymentMapper.addEClassTake(eClassTake);
}

```

6.3.9 결제

결제 정보

결제상품명
다이어트댄스

금액
15000

정기결제 기간
3개월

이름
일반회원3

이메일
id004@email.com

휴대폰 번호
01012345678 인증번호 전송

인증하기

할인 적용

포인트 사용
보유 포인트 : 50p 포인트 전체사용

적용 취소

결제 수단

신용카드 가상계좌 카카오페이

총 결제금액 **15000** 결제하기

결제

결제 화면

■ JavaScript

```
<script type="text/javascript">
$(function(){
    //인증번호 유효성 검사
    $('#paymentForm').validate({
        rules: {
            memberPhoneCertify: {
                required: true,
                digits : true
            },
        },
    },
},
```

```

        messages: {
            memberPhoneCertify: {
                required: "*인증번호를 입력해주세요",
                digits : "*전송된 인증번호를 입력해주세요"
            },
        }
    });

    //결제방식 클릭할때 색 바뀌게 하기
    $('.paymentMethodBtn').click(function(){
        if($(this).children('input[type="radio"]').prop('checked') ==true){
            $(this).parents('div.row').find('.paymentMethodBtn').removeClass('active');
            $(this).addClass("active");
        }
    });

    //총 결제금액
    $('.totalprice').text($('.input[name="paymentTotalOrderPrice"]').val());
    $('#paymentTotalPayPrice').val($('.totalprice').text());

    //포인트 전체사용 클릭 이벤트
    $('.allPointUse').click(function(){
        var point = $('.finalPoint').text();
        if($(this).prop('checked') ==true){
            $('.input[name="paymentUsePoint"]').val(point);
        }else {
            $('.input[name="paymentUsePoint"]').val("");
        }
    });

    //포인트 적용 버튼 클릭 이벤트
    $('#pointUseBtn').click(function(){
        if($('.finalPoint').text() >0 && $('#pointUseBtn').hasClass('active') ==true){
            $('.finalPoint').text($('.finalPoint').text() -
            $('.input[name="paymentUsePoint"]').val());
            $('.totalprice').text($('.input[name="paymentTotalOrderPrice"]').val() -
            $('.input[name="paymentUsePoint"]').val());
            $('.allPointUse').attr('disabled', 'disabled');
            if($('.input[name="paymentUsePoint"]').val() >0){
                $('#pointUseBtn').removeClass('active');
                $('#pointUseCancelBtn').addClass('active');
                $('.input[name="paymentUsePoint"]').attr('readonly',
                'readonly');
                $('#paymentTotalPayPrice').val($('.totalprice').text());
            }else {
                alert('사용할 포인트를 입력해주세요');
                $('.input[name="paymentUsePoint"]').focus();
            }
        }
    });

    //포인트 적용 취소
    $('#pointUseCancelBtn').click(function(){
        if($(this).hasClass('active') ==true){
            $('.finalPoint').text(parseInt($('.finalPoint').text())
            +parseInt($('.input[name="paymentUsePoint"]').val()));
            $('.input[name="paymentUsePoint"]').val('').removeAttr('readonly');
            $('.allPointUse').removeAttr('disabled').prop('checked', false);
            $(this).removeClass('active');
            $('#pointUseBtn').addClass('active');
            $('.totalprice').text($('.input[name="paymentTotalOrderPrice"]').val());
            $('#paymentTotalPayPrice').val($('.totalprice').text());
        }
    });

```

```

    });

    /*아임포트 카카오페이 결제*/
    $('#paymentBtn').click(function () {
        var IMP =window.IMP;
        IMP.init('imp03955010');
        var paymentGoods = $('input[name="paymentOrderName"]').val();
        var money = $('.totalprice').text();
        var memberName = $('input[name="memberName"]').val();
        var memberEmail = $('input[name="memberEmail"]').val();
        var memberPhone = $('input[name="memberPhone"]').val();

        var len = memberEmail.indexOf("@");
        var memberId = memberEmail.substr(0, len);
        IMP.request_pay({
            pg: 'card',
            paymentCode: 'p'+new Date().getTime()+'_'+memberId,
            name: paymentGoods,
            amount: money,
            buyer_email: memberEmail,
            buyer_name: memberName,
            buyer_tel: memberPhone,
        }, function (rsp) {
            if (rsp.success) {
                var msg = '결제가 완료되었습니다.';
                if(rsp.apply_num !=null || rsp.apply_num !='')msg += '카드 승인번호 : '+ rsp.apply_num;

                $('#paymentForm').submit();
            } else {
                var msg = '결제에 실패하였습니다.';
                msg += '에러내용 : '+ rsp.error_msg;
                document.location.href="/mypage/mySignUpForClassList"; //alert
            }
            alert(msg);
        });
    });

    //휴대폰번호 인증번호 보내기 버튼 클릭 이벤트
    $('#memberPhoneCheck').click(function(){

        var to = $('input[name="memberPhone"]').val();
        $.ajax({
            url : "/memberPhoneCheck",
            type : "POST",
            data : "to="+ to,
            dataType : "json",
            success : function(data) {
                const checkNum = data;
                alert('checkNum:'+ checkNum);
                $('#certifyCheck').addClass('active');
                $('#certifyCheck').click(function(){
                    const userNum =
                    $('#input[name="memberPhoneCertify"]').val();

                    if(checkNum == userNum){
                        alert('인증 성공하였습니다.');
```

```
해주세요.');
```

```
        }  
    });  
},  
error : function() {  
    alert("에러")  
}  
});  
});  
});  
</script>
```

■ Controller

```
//결제처리  
@PostMapping("/payment")  
public String payment(Model model, Payment payment, RedirectAttributes reAttr) {  
  
    model.addAttribute("title", "결제");  
  
    paymentService.addPayment(payment);  
    System.out.println(payment.getPaymentUsePoint());  
  
    reAttr.addAttribute("paymentCode", payment.getPaymentCode());  
  
    return "redirect:/mypage/mypaymentList/paymentDetail";  
}  
//결제화면  
@GetMapping("/payment")  
public String payment(Model model, @RequestParam(name="paymentGroupCode") String  
paymentGroupCode) {  
  
    model.addAttribute("title", "결제");  
    String paymentGoodsName = null;  
    int paymentGoodsPrice = 0;  
    int paymentGoodsSetDate = 0;  
    String memberEmail = null;  
    String memberPhone = null;  
  
    if(paymentGroupCode.startsWith("e")) {  
        Map<String,String> eClassTakeInfo =  
paymentMapper.getEClassTake(paymentGroupCode);  
        String eClassCode = eClassTakeInfo.get("eClassApprovedCode");  
        memberEmail = eClassTakeInfo.get("memberEmail");  
        memberPhone = eClassTakeInfo.get("memberPhone");  
  
        EClassApproved eClass= paymentMapper.getEClassApproved(eClassCode);  
  
        if(Objects.nonNull(eClass)) {  
            paymentGoodsName = eClass.geteClassApprovedName();  
            paymentGoodsPrice = eClass.geteClassApprovedPrice();  
            paymentGoodsSetDate = eClass.geteClassApprovedSetDate();  
        }  
  
    }else {  
        Map<String,String> challengeEnterInfo =  
paymentMapper.getChallengeEnter(paymentGroupCode);  
        String challengeCode = challengeEnterInfo.get("challengeGatherCode");  
        memberEmail = challengeEnterInfo.get("memberEmail");  
  
        ChallengeGather challenge =  
challengeEnterMapper.getChallengeEnterByCode(challengeCode);
```

```

        if(Objects.nonNull(challenge)) {
            paymentGoodsName = challenge.getChallengeGatherName();
            paymentGoodsPrice = challenge.getChallengeEnterDeposit();
        }
    }

    Member member = memberMapper.getMemberInfoByEmail(memberEmail);

    model.addAttribute("paymentGoodsName", paymentGoodsName);
    model.addAttribute("paymentGoodsPrice", paymentGoodsPrice);
    model.addAttribute("paymentGoodsSetDate", paymentGoodsSetDate);
    model.addAttribute("member", member);
    model.addAttribute("memberPhone", memberPhone);
    model.addAttribute("paymentGroupCode", paymentGroupCode);

    return "payment/payment";
}
//문자 인증
@PostMapping("/memberPhoneCheck")
public @ResponseBody String memberPhoneCheck(@RequestParam(value="to") String to) throws
CoolsmsException {

    return paymentService.PhoneNumberCheck(to);
}

```

■ Service

```

//결제처리
public void addPayment(Payment payment) {

    int len = payment.getMemberEmail().indexOf("@");
    String memberId = payment.getMemberEmail().substring(0, len);
    SimpleDateFormat date =new SimpleDateFormat("yyyyMMddhhmmss");

    String paymentCode = "p"+date.format(new Date())+"_"+memberId;

    String paymentMethodCode
paymentMapper.getPaymentMethodCode(payment.getPaymentMethodName());

    payment.setPaymentCode(paymentCode);
    payment.setPaymentMethodCode(paymentMethodCode);

    paymentMapper.addPayment(payment);
}

//휴대폰번호 인증문자 보내기
public String PhoneNumberCheck(String to) throws CoolsmsException{
    String api_key = "--";
    String api_secret = "--";
    Message coolsms =new Message(api_key, api_secret);

    Random rand =new Random();
    String numStr = "";
    for(int i=0; i<4; i++) {
        String ran = Integer.toString(rand.nextInt(10));
        numStr += ran;
    }

    HashMap<String, String> params =new HashMap<String, String>();

```



```
params.put("to", to); // 수신전화번호 (ajax로 view 화면에서 받아온 값으로 넘김)
params.put("from", "01000000000"); // 발신전화번호. 테스트시에는 발신,수신 둘다 본인 번호로
하면 됨
params.put("type", "sms");
params.put("text", "AI 쌍방향 홈트레이닝 인증번호는 [+ numStr +] 입니다.");
coolsms.send(params); // 메시지 전송

    return numStr;
}
```

6.3.10 식단기능

개별 식단 페이지

2022-04-19
식단은행 가져오기

아침식사
점심식사
저녁식사

	식품명	1회제공량	에너지(kcal)	탄수화물(g)	단백질(g)	지방(g)	나트륨(mg)	
완료	도토리묵말이	1300	462	90	7	9	3206	취소 삭제
미완료	치즈크러스트 디핑소스	11	60	0	0	0	58	실행 삭제
미완료	그뤼에르 풍두소스	60	106	0	3	0	250	실행 삭제

식단 계획 추가하기

영양소 섭취

칼로리(kcal)	462kcal	/	628kcal
탄수화물(g)	90g	/	90g
단백질(g)	7g	/	10g
지방(g)	9g	/	9g
나트륨(mg)	3206mg	/	3514mg

개별 식단 페이지 - 날짜선택

2022-04-25

« 4월 2022 »

일	월	화	수	목	금	토
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Today

저녁식사

1회제공량

600

영양소 섭취

개별 식단 페이지 - 식단은행 불러오기

dietMyList

dietBankCode1(id001@email.com,2022-04-13)

식단 계획에 추가하기

요일	아침식사	점심식사	저녁식사
월요일	초콜릿케이크 100g 420kcal	간자장 650g 825kcal	가자미조림 300g 298kcal
	초콜릿케이크 100g 420kcal	기스면 1000g 607kcal	복어조림 100g 182kcal
	치즈케이크 100g 329kcal	굴짬뽕 900g 681kcal	동태조림 250g 276kcal
화요일	계국지 1100g 415kcal	고추장불고기 삼각김밥 100g 165kcal	고추장불고기 삼각김밥 100g 165kcal
	곰치국 1000g 419kcal	곤드레나물밥 350g 522kcal	농어초밥 250g 397kcal
	굴국밥 650g 683kcal	농어초밥 250g 397kcal	곤드레나물밥 350g 522kcal

개별 식단 페이지 - 식단 추가하기

e/dietMyList

22-04-25

생채및 무침류

나물.채소류무침

분류를 선택하고 검색할 식품명을 입력하세요

식품명	1회제공량	에너지(kcal)	탄수화물(g)	단백질(g)	지방(g)	나트륨(mg)	
노각무침	150	84	14	3	2	817	추가
단무지무침	50	20	3	1	1	429	추가
달래나물무침	150	136	24	4	3	867	추가
더덕무침	150	216	43	4	3	1247	추가
숙갓나물무침	150	96	7	5	6	710	추가
우거지나물무침	150	129	9	5	9	773	추가
							닫기

₩1님이 로그인하셨

식단은행 가져오기

삭제

1377kcal

0g / 244

0g / 28

0g / 33

mg / 5499m

사용자별 식단

입력한 아이디를 통해 자신의 식단 계획 화면을 보여줌
 식단 계획 화면에서 식단을 계획/ 실행/ 식단 은행에서 가져오기 등을 수행할 수 있음.
 계획 및 실행한 비율을 계산해 식단 실행 진행상황을 프로그레스 바로 보여줌

■ javascript

```
<script th:inline="javascript">
```

```
<!-- 페이지의 기본 javascript -->
```

```
$(document).ready(function(){
    dayChange();
    NutrChange();
})
```

```
var memberEmail = [[${session.SEMAIL}]];
var dietPlanTime = "아침식사";
var NutrChange =function(){
    $('.NutrientBar').each(function(){
        var DoneNu =
```

```

$(this).children().eq(1).text();

var Nu = $(this).children().eq(3).text();
var result = DoneNu/Nu*100+"%";

console.log($(this).next().find('.progress-bar').attr('style',"width:"+result+";"));
console.log(DoneNu);
console.log(Nu);

});
}

var NutrChange =function(){

console.log("실행은 하니?");

var dietPlanDay = $('#datePicker').val();

$.ajax({

url:
'/diet/mypage/updateProgress'

,type: 'POST'
,data:
{ 'memberEmail' : memberEmail,
'dietPlanDay' : dietPlanDay
}
,dataType: 'json'
,success: function(data){

var

nutrientsAPIKcalDone = data.nutrientsAPIKcalDone
,nutrientsAPIKcal
= data.nutrientsAPIKcal
,nutrientsAPICarboDone = data.nutrientsAPICarboDone
,nutrientsAPICarbo = data.nutrientsAPICarbo
,nutrientsAPIProDone = data.nutrientsAPIProDone
,nutrientsAPIPro
= data.nutrientsAPIPro
,nutrientsAPIFatDone = data.nutrientsAPIFatDone
,nutrientsAPIFat
= data.nutrientsAPIFat
,nutrientsAPINatrDone = data.nutrientsAPINatrDone
,nutrientsAPINatr
= data.nutrientsAPINatr;

[nutrientsAPIKcalDone, nutrientsAPICarboDone, nutrientsAPIProDone, nutrientsAPIFatDone,
nutrientsAPINatrDone];

var DoneList =
[nutrientsAPIKcal, nutrientsAPICarbo, nutrientsAPIPro, nutrientsAPIFat, nutrientsAPINatr];

var KcalList =

var count =0;

```

```

$('.NutrientBar').each(function(){
    var
    degree =null;
    if(count
    ==0){
    degree ="kcal";
    } else
    if(count ==4){
    degree ="mg"
    }else{
    degree ="g";
    }
    DoneNu = $(this).children().eq(1).text(DoneList[count]+degree);
    var Nu
    = $(this).children().eq(3).text(KcalList[count]+degree);
    var
    result = DoneList[count]/KcalList[count]*100+"%";
    console.log(result);
    $(this).next().find('.progress-bar').attr('style','width:'+result+"%");
    count
    ++;
});
}
});
}

```

<!-- 데이트피커 -->

```

var dayChange =function(){
    var DayCheck =null;
    $('.Diettime-btn').each(function(){
        var BtnCheck = $(this).hasClass('active');
        if(BtnCheck ==true){
            console.log($(this).text());
            DayCheck = $(this).text();
        }
    })
}

var count =0;
$('.planListTr').each(function(){

```

```

        var ListDayVal = $(this).attr('value');

        if(ListDayVal == DayCheck){
            $(this).show();
            count ++;
        }else{
            $(this).hide();
        }
    });

    console.log(count);

    if(count ==0){
        $('#MyPlanedDietTable').hide();
    } else{
        $('#MyPlanedDietTable').show();
    }
}

$('#datePicker').datepicker({
    format : "yyyy-mm-dd"
    ,language : "kr"
    ,todayHighlight : true
    ,todayBtn : true
    ,autoclose : true
}).datepicker("setDate", 'now');

$('#datePicker').on('changeDate',function(e){

    var dietPlanDay = $('#datePicker').val();

    $.ajax({
        url:
'/diet/mypage/selectChangedDietPlanList'
        ,type: 'POST'
        ,data:
        { 'memberEmail' : memberEmail,
          'dietPlanDay' : dietPlanDay
        }
        ,dataType: 'text'
        ,success: function(data){

            $('#myDietPlanedList').replaceWith(data);

            NutrChange();

        }
    });

    console.log(e);
});

$('#Diettime-btn').click(function(e){

    $('#Diettime-btn').removeClass( 'active' );
    $(e.target).addClass('active');
});

```

```
dietPlanTime = $(e.target).html()
```

```
dayChange();
```

```
});
```

```
<!-- 실행/취소 버튼 클릭 update -->
```

```
$('.planDo-Btn').click(function(e){
```

```
    var dietPlanDoValue = 1;
```

```
    var dietPlanCode = $(e.target).attr('value');
```

```
    var dietPlanDay = $('#datePicker').val();
```

```
    $.ajax({
```

```
        url: '/diet/mypage/updateUserDietPlan'
```

```
    ,type: 'POST'
```

```
    ,data:
```

```
    {'dietPlanDoValue' : dietPlanDoValue,
```

```
     'dietPlanCode' : dietPlanCode
```

```
    }
```

```
    ,dataType: 'text'
```

```
    ,success: function(data){
```

```
        if(data =1){
```

```
            console.log("돼");
```

```
            $.ajax({
```

```
                ,type: 'POST'
```

```
            ,data:
```

```
            {'memberEmail' : memberEmail,
```

```
             'dietPlanDay' : dietPlanDay,
```

```
            }
```

```
            ,dataType: 'text'
```

```
            ,success: function(data){
```

```
                '/diet/mypage/selectChangedDietPlanList'
```

```
url:
```

```
                $('#myDietPlannedList').replaceWith(data);
```

```
                NutrChange();
```

```
            }
```

```
        });
```

```
    }
```

```
    }
```

```
});
```

```
});
```

```
$('.planCancel-Btn').click(function(e){
```

```
    var dietPlanDoValue = 0;
```

```
    var dietPlanCode = $(e.target).attr('value');
```

```
    var dietPlanDay = $('#datePicker').val();
```

```
    $.ajax({
```

```
        url: '/diet/mypage/updateUserDietPlan'
```

```
    ,type: 'POST'
```

```
    ,data:
```

```
    {'dietPlanDoValue' : dietPlanDoValue,
```

```
     'dietPlanCode' : dietPlanCode
```

```
    }
```

```
    ,dataType: 'text'
```

```
    ,success: function(data){
```

```
        if(data =1){
```

```
            console.log("돼");
```

```
            $.ajax({
```



```

url:
'/diet/mypage/selectChangedDietPlanList'
,type: 'POST'
,data:
{'memberEmail' : memberEmail,
'dietPlanDay' : dietPlanDay,
}
,dataType: 'text'
,success: function(data){

$('#myDietPlannedList').replaceWith(data);
NutrChange();
}
});
}
});
});

<!-- 삭제 버튼 클릭 해당 줄 delete -->
$('.planDel-Btn').click(function(e){
var dietPlanCode = $(e.target).attr('value')
var dietPlanDay = $('#datePicker').val();
console.log(memberEmail);
console.log(dietPlanDay);

$.ajax({ url: '/diet/mypage/deleteUserDietPlan'
,type: 'POST'
,data:
{'memberEmail' : memberEmail,
'dietPlanDay' : dietPlanDay,
'dietPlanCode' : dietPlanCode
}
,dataType: 'text'
,success: function(data){
if(data =1){
console.log("돼");
$.ajax({
url:
'/diet/mypage/selectChangedDietPlanList'
,type: 'POST'
,data:
{'memberEmail' : memberEmail,
'dietPlanDay' : dietPlanDay,
}
,dataType: 'text'
,success: function(data){

$('#myDietPlannedList').replaceWith(data);
NutrChange();
}
});
}
});
});

<!-- 식단은행 불러오기 -->
$('#selectDietBank-btn').click(function(){
console.log("식단은행 실행된다");

```

```

$.ajax({
    url: '/diet/mypage/selectDietBankList'
    ,type: 'POST'
    ,dataType: 'text'
    ,success: function(data){
        $('#selectDietBankModalIn').replaceWith (data);
    }
});

```

```
});
```

```

<!-- Ajax select:mealSort -->
$('#updateDetailBank').click(function(){

```

```

$.ajax({
    url: '/admin/diet/selectDietMealLevelList'
    ,type: 'POST'
    ,dataType: 'text'
    ,success: function(data){

        $("#mainMealSort").replaceWith(data);
    }
});

```

```
});
```

```

<!-- 대 분류 선택 -->

```

```

function MainSortchageSelect(){
    var mainMealSort = $('#mainMealSort option:selected').val()

```

```

$.ajax({
    url: '/admin/diet/selectDietMealListDetail'
    ,type: 'POST'
    ,data: {'mainMealSort' : mainMealSort}
    ,dataType: 'text'
    ,success: function(data){
        $("#testDmealSort").replaceWith(data);
    }
});

```

```

$.ajax({
    url: '/diet/mypage/selectUserDietMealList'
    ,type: 'POST'
    ,data: {'mainMealSort' : mainMealSort}
    ,dataType: 'text'
    ,success: function(data){

```

```
$("#DietNutrientListTable").replaceWith(data);
```

```
});
```

```
}
```

```

<!-- 소 분류 선택 -->

```

```

function DetailSortchageSelect(){

    var
        DetailMealSort = $('#DetailMealSort option:selected').val()
        ,NutrientListLine = $('.NutrientListLine');

    NutrientListLine.each(function(){

        var searchVal = $(this).attr('value');

        if(DetailMealSort == searchVal){
            $(this).show();
        }else{
            $(this).hide();
        }
    });
}

<!-- 검색어 입력 -->

$('#searchNutrient').keyup(function(){
    var searchKeyword = $(this).val();
    var NutrientListLine = $('.NutrientListLine');

    NutrientListLine.each(function(){
        var searchLineName = $(this).children().first().text();
        var searchResult = searchLineName.indexOf(searchKeyword);

        if(searchResult >-1){
            $(this).show();
        }else{
            $(this).hide();
        }
    });
});

<!-- 추가모달 on --->
$(document).ready(function(){
    $('#myModal').on('show.bs.modal', function (e) {
        $.ajax({
            url: '/admin/diet/selectDietMealLevelList'
            ,type: 'POST'
            ,dataType: 'text'
            ,success: function(data){

                $('#testMmealSort').replaceWith(data);
                $('#DetailMealSort').removeAttr('style');
            }
        });
    });
});

<!-- 추가모달 off -->
$(document).ready(function(){
    $('#myModal').on('hidden.bs.modal', function () {

```

```
selected disabled style="display:none">상세분류</option>');
                                                    $('#DetailMealSort').empty();
                                                    $('#DetailMealSort').append('<option
                                                    $('#DietNutrientListTable').empty();
                                                    });
                                                    });

</script>
```

■ Controller

```
@Controller
@RequestMapping("/diet/mypage/")
public class DietUserController {

    private static final Logger log = LoggerFactory.getLogger(DietUserController.class);

    //DI 의존성 주입
    private DietService dietService;
    private DietMapper dietMapper;

    public DietUserController(DietService dietService, DietMapper dietMapper) {
        this.dietService = dietService;
        this.dietMapper = dietMapper;
    }

    @GetMapping("/dietBankList")
    public String selectDietBankList(Model model) {

        model.addAttribute("title", "식단 은행 테스트");
        model.addAttribute("role", "식단은행 테스트페이지");

        return "diet/dietBankList";
    }

    //개인 식단 페이지 이동, 음식 대분류 조회, 개인 식단 계획 조회
    @GetMapping("/dietMyList")
    public String selectDietMyList(Model model, HttpSession session) {
        model.addAttribute("title", "개인 식단페이지");
        model.addAttribute("role", "개인 식단 페이지 테스트");

        //음식 대분류 조회
        List<DietMealLevelList> mainMealSort = dietMapper.selectDietMealLevelList();
        model.addAttribute("mainMealSort", mainMealSort);

        //개인 식단 계획 조회(오늘자)
        DietPlan dietplan = new DietPlan();
        String memberEmail = (String) session.getAttribute("SEMAIL");
        String today = LocalDate.now().toString();

        dietplan.setMemberEmail(memberEmail);
        dietplan.setDietPlanDay(today);

        List<HashMap<String, Object>> UserdietPlanList = dietService.selectUserDietPlan(dietplan);
        model.addAttribute("UserdietPlanList", UserdietPlanList);

        HashMap<String, Object> UserdietPlanListNutrient =
        dietService.UserdietPlanList(UserdietPlanList);
        model.addAttribute("UserdietPlanListNutrient", UserdietPlanListNutrient);

        return "diet/dietMyList";
    }
}
```

```

//Ajax 대분류 선택 후 음식 조회(user)
@PostMapping("/selectUserDietMealList")
public String selectDietMealList(Model model
,@RequestParam(name="mainMealSort",required =false) String mainMealSort) {

    List<DietNutrientList> DietNutrientList = dietMapper.selectDietNutrientList(mainMealSort);

    model.addAttribute("DietNutrientList", DietNutrientList);

    return "diet/AjaxTable/DietUserMealListAjax";
}

//Ajax 개인 식단 계획에 음식 삽입
@PostMapping("/insertUserDietPlan")
@ResponseBody
public int insertUserDietPlan(Model model,
                                DietPlan dietplan) {

    //삽입시 필요한 새 코드명
    String newCode = dietService.selectDietBankListNewCode("dietplan", "dietPlanCode");
    dietplan.setDietPlanCode(newCode);

    int insertResult = dietService.insertUserDietPlan(dietplan);

    return insertResult;
}

@PostMapping("/selectChangedDietPlanList")
public String selectChangedDietPlanList(Model model
                                , DietPlan
dietplan) {

    List<HashMap<String, Object>> UserdietPlanList = dietService.selectUserDietPlan(dietplan);
    model.addAttribute("UserdietPlanList",UserdietPlanList);

    HashMap<String, Object> UserdietPlanListNutrient =
dietService.UserdietPlanList(UserdietPlanList);
    model.addAttribute("UserdietPlanListNutrient", UserdietPlanListNutrient);

    log.info("UserdietPlanListNutrient : {}", UserdietPlanListNutrient);

    return "diet/AjaxTable/DietUserMealplanedListAjax";
}

@PostMapping("/deleteUserDietPlan")
@ResponseBody
public int deleteUserDietPlan(DietPlan dietplan) {

    int deleteResult = dietService.deleteUserDietPlan(dietplan);

    return deleteResult;
}

```

```

//Ajax 식단 계획 페이지에서 실행 눌렀을 때, 혹은 취소 눌렀을 때 update
@PostMapping("/updateUserDietPlan")
@ResponseBody
public int updateUserDietPlan(DietPlan dietPlan) {

    int updateResult = dietService.updateUserDietPlan(dietPlan);

    return updateResult;
}

//Ajax User식단표 progress-bar
@PostMapping("/updateProgress")
@ResponseBody
public HashMap<String, Object> updateProgress(Model model, DietPlan dietplan){

    List<HashMap<String, Object>> UserdietPlanList = dietService.selectUserDietPlan(dietplan);
    model.addAttribute("UserdietPlanList",UserdietPlanList);

    HashMap<String, Object> UserdietPlanListNutrient =
dietService.UserdietPlanList(UserdietPlanList);
    model.addAttribute("UserdietPlanListNutrient", UserdietPlanListNutrient);

    return UserdietPlanListNutrient;
}

//Ajax user diet에서 식단은행 불러오기
@PostMapping("/selectDietBankList")
public String selectDietBankListByUser(Model model) {
    List<DietBank> dietBankList = dietMapper.getDietBankListAdmin();
    model.addAttribute("dietBankList", dietBankList);

    return "diet/AjaxTable/dietMyListSelectBankAjax";
}

//Ajax user diet에서 식단은행 디테일 불러오기
@PostMapping("/selectDietBankListDetail")
public String selectDietBankListDetail(Model model,DietBank dietBank) {

    //select 시간, 요일 넘겨주기
    List<List<HashMap<String, Object>>> selectBankDay =
dietService.selectBankDay();

    model.addAttribute("Banktime", selectBankDay.get(0));
    model.addAttribute("Bankday", selectBankDay.get(1));

    //select 음식 조회, 화면에 뿌려주기
    String dietBankCode = dietBank.getDietBankCode();
    DietOnemealConnection dietOnemealConnection =new DietOnemealConnection();
    dietOnemealConnection.setDietBankCode(dietBankCode);

    List<HashMap<String, Object>> selectOneMealConn =
dietMapper.selectDietOneMealConnectionByBankCode(dietOnemealConnection);
    model.addAttribute("selectOneMealConn", selectOneMealConn);
}

```

```

        return "/diet/AjaxTable/DietMyListSelectBankDetailAjax";
    }

    //Ajax user 식단에 dietBank 내용 삽입
    @PostMapping("/insertDietBankListDetail")
    @ResponseBody
    public void insertDietBankListDetail(DietBank dietBank, HttpSession session) {

        String memberEmail = (String) session.getAttribute("SEMAIL");
        dietBank.setMemberEmail(memberEmail);

        dietService.insertDietOneMealConnectionAll(dietBank);
    }

```

■ Service

```

//User DietPlan 추가버튼 눌렀을 때 insert
public int insertUserDietPlan(DietPlan dietplan) {

    int result = dietMapper.insertUserDietPlan(dietplan);

    return result;
}

//UserDietPlan 페이지 select
public List<HashMap<String, Object>> selectUserDietPlan(DietPlan dietplan){

    List<HashMap<String, Object>> DietPlanList = dietMapper.selectUserDietPlan(dietplan);

    return DietPlanList;
}

//UserDietPlan 에서 삭제
public int deleteUserDietPlan(DietPlan dietplan) {
    int deleteReuslt = dietMapper.deleteUserDietPlan(dietplan);
    return deleteReuslt;
}

//UserDietPlanUpdate
public int updateUserDietPlan(DietPlan dietplan) {
    int updateResult = dietMapper.updateUserDietPlan(dietplan);
    return updateResult;
}

//total 영양소 계산기
public HashMap<String, Object> UserdietPlanList(List<HashMap<String, Object>> UserdietPlanList){

    //총 계획 영양소들
    Integer nutrientsAPIKcal =0;
    Integer nutrientsAPICarbo =0;
    Integer nutrientsAPIPro =0;
    Integer nutrientsAPIFat =0;
    Integer nutrientsAPINatr =0;

    for(int i=0; i<UserdietPlanList.size(); i++) {

        HashMap<String, Object> mealList = UserdietPlanList.get(i);
    }
}

```



```

        nutrientsAPIKcal += (Integer)mealList.get("nutrientsAPIKcal");
        nutrientsAPICarbo += (Integer)mealList.get("nutrientsAPICarbo");
        nutrientsAPIPro += (Integer)mealList.get("nutrientsAPIPro");
        nutrientsAPIFat += (Integer)mealList.get("nutrientsAPIFat");
        nutrientsAPINatr += (Integer)mealList.get("nutrientsAPINatr");
    }

```

```

    HashMap<String, Object> planedNutrient =new HashMap<>();

```

```

    planedNutrient.put("nutrientsAPIKcal", nutrientsAPIKcal);
    planedNutrient.put("nutrientsAPICarbo", nutrientsAPICarbo);
    planedNutrient.put("nutrientsAPIPro", nutrientsAPIPro);
    planedNutrient.put("nutrientsAPIFat", nutrientsAPIFat);
    planedNutrient.put("nutrientsAPINatr", nutrientsAPINatr);

```

```

    //총 실행 영양소들

```

```

    nutrientsAPIKcal =0;
    nutrientsAPICarbo =0;
    nutrientsAPIPro =0;
    nutrientsAPIFat =0;
    nutrientsAPINatr =0;

```

```

    for(int i=0; i<UserdietPlanList.size(); i++) {

```

```

        HashMap<String, Object> mealList = UserdietPlanList.get(i);

```

```

        if (mealList.get("dietPlanDoValue").equals(1)) {

```

```

            nutrientsAPIKcal += (Integer)mealList.get("nutrientsAPIKcal");
            nutrientsAPICarbo += (Integer)mealList.get("nutrientsAPICarbo");
            nutrientsAPIPro += (Integer)mealList.get("nutrientsAPIPro");
            nutrientsAPIFat += (Integer)mealList.get("nutrientsAPIFat");
            nutrientsAPINatr += (Integer)mealList.get("nutrientsAPINatr");

```

```

        }

```

```

    }

```

```

    planedNutrient.put("nutrientsAPIKcalDone", nutrientsAPIKcal);
    planedNutrient.put("nutrientsAPICarboDone", nutrientsAPICarbo);
    planedNutrient.put("nutrientsAPIProDone", nutrientsAPIPro);
    planedNutrient.put("nutrientsAPIFatDone", nutrientsAPIFat);
    planedNutrient.put("nutrientsAPINatrDone", nutrientsAPINatr);

```

```

    return planedNutrient;

```

```

}

```

```

//받아온 식단은행 내 식단에 넣기

```

```

public int insertDietOneMealConnectionAll(DietBank dietBank) {

```

```

    //select 음식 조회

```

```

    String dietBankCode = dietBank.getDietBankCode();
    String pickerToday = dietBank.getDietBankUpdateDate();
    DietOnemealConnection dietOnemealConnection =new DietOnemealConnection();
    dietOnemealConnection.setDietBankCode(dietBankCode);

```

```

    List<HashMap<String, Object>> selectOneMealConn =

```

```

    dietMapper.selectDietOneMealConnectionByBankCode(dietOnemealConnection);

```

```

        for(int i=0; i<selectOneMealConn.size(); i++) {

                String                CheckGroup                =                (java.lang.String)
selectOneMealConn.get(i).get("dietOneMealConnectionGroupNum");

                String TimeValue = CheckGroup.substring(CheckGroup.lastIndexOf(" ") + 1);
                String DayValue = CheckGroup.substring(0, CheckGroup.lastIndexOf(" "));

                DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");

                LocalDate now = LocalDate.parse(pickerToday, formatter);

                String dietPlanCode = selectDietBankListNewCode("dietplan", "dietPlanCode");
                String                nutrientsAPICode                =                (java.lang.String)
selectOneMealConn.get(i).get("nutrientsAPICode");
                String memberEmail = dietBank.getMemberEmail();
                String dietPlanDay = null;
                String dietPlanTime = TimeValue;

                DietPlan dietplan = new DietPlan();

                dietplan.setDietPlanCode(dietPlanCode);
                dietplan.setNutrientsAPICode(nutrientsAPICode);
                dietplan.setMemberEmail(memberEmail);
                dietplan.setDietPlanTime(dietPlanTime);

                if(DayValue.equals("월요일")) {
                }else if(DayValue.equals("화요일")) {
                        now = now.plusDays(1);
                }else if(DayValue.equals("수요일")) {
                        now = now.plusDays(2);
                }else if(DayValue.equals("목요일")) {
                        now = now.plusDays(3);
                }else if(DayValue.equals("금요일")) {
                        now = now.plusDays(4);
                }else if(DayValue.equals("토요일")) {
                        now = now.plusDays(5);
                }else if(DayValue.equals("일요일")) {
                        now = now.plusDays(6);
                }

                dietPlanDay = now.format(formatter);

                dietplan.setDietPlanDay(dietPlanDay);

                if(nutrientsAPICode != null) {
                        log.info("최종 들어간값들!!:{", dietplan);
                        log.info("날짜!!:{", DayValue);

                        dietMapper.insertUserDietPlan(dietplan);
                }
        }

        return 0;
}

```


6.3.11 식단은행

식단 은행 - 전체 목록 조회 페이지

검색▼
식단 추가하기 +

식단은행 번호	운동클래스	작성자기메일	작성일	수정자기메일	최종수정일	확인/수정	삭제
dietBankCode1		id001@email.com	2022-04-13	id001@email.com	2022-04-13	확인/수정	삭제
dietBankCode2		id001@email.com	2022-04-13	id001@email.com	2022-04-13	확인/수정	삭제
dietBankCode3		id001@email.com	2022-04-13	id001@email.com	2022-04-13	확인/수정	삭제
dietBankCode4		id001@email.com	2022-04-19			확인/수정	삭제
dietBankCode5		id001@email.com	2022-04-19	id001@email.com	2022-04-19	확인/수정	삭제
dietBankCode6		id001@email.com	2022-04-21			확인/수정	삭제

showing 0-10 of 30 items

« 1 »

식단 은행 - 생성 정보 확인 페이지

시스템
관리짱1님이 로그인

새 식단 생성

아래와 같이 식단을 생성합니다

생성자Email	<input style="width: 100%;" type="text" value="id001@email.com"/>
연동 클래스	<input style="width: 100%;" type="text" value="내가 생성한 eclass 1번"/> ▼
생성 날짜	<input style="width: 100%;" type="text" value="2022/04/25"/>

확인 및 생성
취소

식단 은행 상세 페이지 - 상세 정보 전체 조회

식단은행 코드	dietBankCode1
작성자 이메일	id001@email.com
현재 연동된 클래스	클래스 연동(클래스 데이터 생성후 구현)
작성연월일	2022-04-13
수정자 이메일	id001@email.com
최종수정일	2022-04-13

식단 상세

시간	월요일	화요일	수요일	목요일	금요일	토요일	일요일
아침식사	초콜릿케이크 100g 420kcal 초콜릿케이크 100g 420kcal 치즈케이크 100g 329kcal +	계국지 1100g 415kcal 곰치국 1000g 419kcal 굴국밥 650g 683kcal +	계국지 1100g 415kcal 곰치국 1000g 419kcal 굴국밥 650g 683kcal +	고추장불고기 삼각김밥 100g 165kcal 농어초밥 250g 397kcal 돼지국밥 1200g 911kcal +	고들빼기 50g 56kcal 얼무얼말이김치 50g 18kcal 고들빼기 50g 56kcal +	평불고기 500g 369kcal 달걀비 400g 596kcal 달걀비 300g 559kcal +	고들빼기 50g 56kcal 얼무얼말이김치 50g 18kcal 고들빼기 50g 56kcal +
점심식사	간자장 650g 825kcal 가스면 1000g 607kcal 굴팍팍 900g 681kcal	고추장불고기 삼각김밥 100g 165kcal 곤드레나물밥 350g 522kcal 농어초밥 250g	고들빼기 50g 56kcal 고들빼기 50g 56kcal 고들빼기 50g 56kcal	수정과 150g 133kcal 식혜 150g 130kcal 수정과 150g 133kcal	미나리나물 50g 30kcal 취나물 50g 74kcal 콩나물 50g 25kcal	곰보항 70g 277kcal 만주 70g 223kcal 참치샌드위치 250g 555kcal	계적 250g 488kcal 소고기산적 200g 453kcal 툰베기적 100g 160kcal

식단 은행 상세 페이지 - 일별, 요일별 개별 식단 조회 페이지

레이닉 시스템
관리자1님이 로그인하셨습니다.

월요일 아침식사

현재 식단

식품명	1회제공량	에너지(kcal)	탄수화물(g)	단백질(g)	지방(g)	나트륨(mg)	
초콜릿케이크	100	420	44	5	25	170	제거
초콜릿케이크	100	420	44	5	25	170	제거
치즈케이크	100	329	29	7	21	201	제거

메뉴검색

기타 기타 분류를 선택하고 검색할 식품명을 입력하세요

식품명	1회제공량	에너지(kcal)	탄수화물(g)	단백질(g)	지방(g)	나트륨(mg)	
도토리묵말이	1300	462	90	7	9	3206	추가
도토리묵	100	46	10	0	0	124	추가
프렌치 라파두이	100	135	0	2	0	315	추가
그뤼에르 콩유소스	60	106	0	3	0	250	추가
치즈크리스티 디핑소스	11	60	0	0	0	58	추가

식단 은행

식단 은행 목록을 미리 작성해 놓을 수 있음.
이 목록을 통해 개인 식단에 미리 작성한 계획을 삽입 할 수 있음.
계획은 요일/시간별로 가능함

■ JavaScript

전체 조회페이지(식단은행)

```
<script type="text/javascript">
    $(document).ready(function(){
        //$('#datatables-example').DataTable();
    });

    <!-- jquery insert -->
    $('#insertDietBank').click(function(){
        var memberEmail ="id001@email.com";
        console.log(memberEmail+"임시 이메일 나중엔 로그인 기록으
로 변환");

        $.ajax({
            url: '/admin/diet/insertDietBankList'
            ,type: 'POST'
            ,data: {'memberEmail' : memberEmail}
            ,dataType: 'json'
            ,success: function(data){

                var today =new Date();
                var                dd
                =String(today.getDate()).padStart(2, '0');
                var                mm
                =String(today.getMonth() +1).padStart(2, '0');
                var                yyyy
                =
                today.getFullYear();
                var                yyyy
                =
                yyyy+"/"+mm+"/"+dd;

                $('#insertTable').find('input').eq(0).val(memberEmail);
                $('#insertTable').find('input').eq(1).val(today);

                $.each(data,function(index,item){

                $('#insertTable').find('select').append("<option value="+index+">"+item+"</option>");
                });
                }
            });

            $('#cancel-insert').click(function){
                $('#insertTable').find('select').children().remove();
            });
    });
</script>
```

```

</script>
상세 조회페이지(식단은행)
<script type="text/javascript">
    $(document).ready(function(){
        //$('#datatables-example').DataTable();
    });

    $('#oneDayDelBtn').click(function(e){

        var dietOneMealConnectionCode = $(e.target).attr('value');
        var dietBankCode = $('#DietUpdietBankCode').text()
        var titleName = $('#modal-title').text();
        $.ajax({
            url:
'/admin/diet/deleteDietOneMealConnection'
            ,type: 'POST'
            ,data:
            {
                'dietOneMealConnectionCode' :
                dietOneMealConnectionCode,
                'dietBankCode' : dietBankCode }
            ,dataType: 'text'
            ,success: function(data){

                $('#MealListContent').replaceWith(data);

                $('#todayTrLine').each(function(index,item){
                    var groupName =
                    $(this).attr('value');
                    if(groupName ==
                    titleName){
                        $(this).show();
                    } else{
                        $(this).hide();
                    }
                });
            }
        });

        <!-- Ajax select:mealSort -->
        $('#updateDetailBank').click(function(){
            $.ajax({
                url: '/admin/diet/selectDietMealLevelList'
                ,type: 'POST'
                ,dataType: 'text'
                ,success: function(data){

                    $('#mainMealSort').replaceWith(data);

```

```

    });
}

});

<!-- 대 분류 선택 -->
function MainSortchageSelect(){
    var mainMealSort = $('#mainMealSort option:selected').val()

    $.ajax({
        url: '/admin/diet/selectDietMealListDetail'
        ,type: 'POST'
        ,data: {'mainMealSort' : mainMealSort}
        ,dataType: 'text'
        ,success: function(data){

$("#DetailMealSort").replaceWith(data);

        }
    });

    $.ajax({
        url: '/admin/diet/selectDietMealList'
        ,type: 'POST'
        ,data: {'mainMealSort' : mainMealSort}
        ,dataType: 'text'
        ,success: function(data){

$("#DietNutrientListTable").replaceWith(data);

        }
    });

}

<!-- 소 분류 선택 -->
function DetailSortchageSelect(){

    var
        DetailMealSort = $('#DetailMealSort option:selected').val()
        ,NutrientListLine = $('.NutrientListLine');

    NutrientListLine.each(function(){

        var searchVal = $(this).attr('value');

        console.log(searchVal);

        if(DetailMealSort == searchVal){
            $(this).show();

        }else{
            $(this).hide();

        }

    });

}

<!-- 검색어 입력 -->

```



```

$('#searchNutrient').keyup(function(){
    var searchKeyword = $(this).val();
    var NutrientListLine = $('.NutrientListLine');

    NutrientListLine.each(function(){
        var searchLineName = $(this).children().first().text();
        var searchResult = searchLineName.indexOf(searchKeyword);

        if(searchResult >-1){
            $(this).show();
        }else{
            $(this).hide();
        }
    });
});

<!-- 수정모달 on ---->
$(document).ready(function(){
    $('#myModal').on('show.bs.modal', function (e) {
        var button = $(e.relatedTarget);
        var recipient = button.data('whatever');
        var modal = $(this);

        modal.find('.modal-title').text(recipient);
        var titleName = modal.find('.modal-title').text();

        $('.todayTrLine').each(function(index,item){

            var groupName = $(this).attr('value');

            if(groupName == titleName){
                $(this).show();
            } else{
                $(this).hide();
            }
        });
    });
});

<!-- 수정모달 off -->
$(document).ready(function(){
    $('#myModal').on('hidden.bs.modal', function () {
        location.reload();
    });
});

</script>

```

6.3.12 챌린지 관리 설정

X
관리자님이 로그인하였습니다.

챌린지 관리 설정

챌린지 > 챌린지 관리 설정

챌린지 카테고리 목록

카테고리 검색 | 카테고리명 | 검색키워드를 입력해주세요.

등록일자 | 시작일 | 부터 | 종료일 | 까지 | 오늘 | 1주일 | 1개월 | 3개월 | 1년

검색 Q | 초기화

전체 19건

챌린지 카테고리코드	챌린지 카테고리명	운동 클래스 카테고리명	등록자 이메일	등록 날짜	수정	삭제
challengeCategoryCode_1	라틴댄스	춤	h402@email.com	2021-03-29	수정	삭제
challengeCategoryCode_2	아쉬랑카요가	춤	h402@email.com	2022-03-29	수정	삭제
challengeCategoryCode_3	기초는기초	기초 요가	h403@email.com	2022-03-29	수정	삭제
challengeCategoryCode_4	초급 걸런지 요가UP	기초 요가	h403@email.com	2022-03-29	수정	삭제
challengeCategoryCode_5	이지올딩과 하는 방충댄스	댄스	h403@email.com	2022-03-29	수정	삭제
challengeCategoryCode_6	방송방송댄스	댄스	h402@email.com	2022-03-29	수정	삭제
challengeCategoryCode_7	춤4	춤	h402@email.com	2022-03-29	수정	삭제
challengeCategoryCode_8	중급기초요가	기초 요가	h403@email.com	2022-03-29	수정	삭제
challengeCategoryCode_10	방송댄스2	댄스	h403@email.com	2021-03-30	수정	삭제
challengeCategoryCode_11	춤춤5	춤	h402@email.com	2021-12-01	수정	삭제
challengeCategoryCode_12	댄스댄스댄스	댄스	h403@email.com	2022-03-21	수정	삭제
challengeCategoryCode_17	춤7개선UP	춤	h402@email.com	2022-03-25	수정	삭제
challengeCategoryCode_20	현충운동	댄스	h403@email.com	2022-04-08	수정	삭제
challengeCategoryCode_22	잠자기 요가	춤	h402@email.com	2022-04-11	수정	삭제
challengeCategoryCode_24	성심탄카데	춤	h402@email.com	2022-04-13	수정	삭제

이전 1 2 다음

카테고리 목록

챌린지 세팅 목록

카테고리 검색 | 세팅코드 | 검색키워드를 입력해주세요.

검색 Q | 초기화

전체 3건

세팅코드	인증 기간 여부 등록 가능 여부	수정	삭제
challengeSettingCode_1	Yes	수정	삭제
challengeSettingCode_2	No	수정	삭제
challengeSettingCode_3	세팅등록UP	수정	삭제

이전 1 다음

세팅 목록

챌린지 관리 설정

챌린지 카테고리 목록 / 챌린지 세팅 목록

등록된 챌린지 카테고리 목록을 조회하고 입력값으로 해당 내용을 검색해서 조회된 목록을 다시 출력시킨다.

챌린지 카테고리를 삭제가 가능하다. 등록페이지, 수정페이지로 이동이 가능하다.

챌린지 세팅 목록을 조회하고 입력값으로 해당 내용을 검색해서 조회된 목록을 다시 출력시킨다.

세팅 등록페이지, 수정페이지로 이동이 가능하다.

■ javascript

```
/* datepicker Start */
```

```

$('.input-daterange').datepicker({
    format: "yyyy-mm-dd", //데이터 포맷 형식(yyyy : 년 mm : 월 dd : 일 )
    /* startDate: '-3d', //달력에서 선택 할 수 있는 가장 빠른 날짜. 이전으로는
    선택 불가능 ( d : 일 m : 달 y : 년 w : 주) */
    autoclose : true, //사용자가 날짜를 클릭하면 자동 캘린더가 닫히는 옵션
    calendarWeeks : false, //캘린더 옆에 몇 주차인지 보여주는 옵션 기본값 false
    보여주려면 true
    clearBtn : false, //날짜 선택한 값 초기화 해주는 버튼 보여주는 옵션 기본값
    false 보여주려면 true
    datesDisabled : ['2019-06-24','2019-06-26'],//선택 불가능한 일 설정 하는
    배열 위에 있는 format 과 형식이 같아야함.
    disableTouchKeyboard : false, //모바일에서 플러그인 작동 여부 기본값
    false 가 작동 true가 작동 안함.
    immediateUpdates: false, //사용자가 보는 화면으로 바로바로 날짜를 변경
    할지 여부 기본값 :false
    multigate : false, //여러 날짜 선택할 수 있게 하는 옵션 기본값 :false
    multigateSeparator : ",", //여러 날짜를 선택했을 때 사이에 나타나는 글자
    2019-05-01,2019-06-01
    templates : {
        leftArrow: '&laquo;',
        rightArrow: '&raquo;'
    }, //다음달 이전달로 넘어가는 화살표 모양 커스텀 마이징
    showWeekDays : true // 위에 요일 보여주는 옵션 기본값 : true
    title: "등록일", //캘린더 상단에 보여주는 타이틀
    todayHighlight : true , //오늘 날짜에 하이라이팅 기능 기본값 :false
    toggleActive : true, //이미 선택된 날짜 선택하면 기본값 : false인경우 그대로
    유지 true인 경우 날짜 삭제
    weekStart : 0 ,//달력 시작 요일 선택하는 것 기본값은 0인 일요일
    language : "ko" //달력의 언어 선택, 그에 맞는 js로 교체해줘야한다.
});
/* datepicker End */

$('.searchBtn').click(function(){
    $('select[name="searchKey"]').attr("required" , true);
    $('input[name="searchValue"]').attr("required" , true);
    $('select[name="searchKey2"]').attr("required" , true);
    $('input[name="searchValue2"]').attr("required" , true);
    $('input[name="searchStart"]').attr("required" , true);
    $('input[name="searchEnd"]').attr("required" , true);
});

$('.searchResetBtn').click(function(){
    $('select[name="searchKey"]').attr("required" , false);
    $('input[name="searchValue"]').attr("required" , false);
    $('select[name="searchKey2"]').attr("required" , false);
    $('input[name="searchValue2"]').attr("required" , false);
    $('input[name="searchStart"]').attr("required" , false);
    $('input[name="searchEnd"]').attr("required" , false);
});

```

```

/*오늘 버튼 클릭시 */
$('.todayBtn').click(function(){
    /* Swal.fire($(this).val()); */
    var btnValue =$(this).val();
    $('select[name="searchKey"]').attr("required" , false);
    $('input[name="searchValue"]').attr("required" , false);
    $('select[name="searchKey2"]').attr("required" , false);
    $('input[name="searchValue2"]').attr("required" , false);
    $('input[name="searchStart"]').attr("required" , false);
    $('input[name="searchEnd"]').attr("required" , false);

});

/*일주일 버튼 클릭시 */
$('.weekBtn').click(function(){
    /* Swal.fire($(this).val()); */
    var btnValue =$(this).val();
    $('select[name="searchKey"]').attr("required" , false);
    $('input[name="searchValue"]').attr("required" , false);
    $('select[name="searchKey2"]').attr("required" , false);
    $('input[name="searchValue2"]').attr("required" , false);
    $('input[name="searchStart"]').attr("required" , false);
    $('input[name="searchEnd"]').attr("required" , false);

});

/*한달 버튼 클릭시 */
$('.monthBtn').click(function(){
    /* Swal.fire($(this).val()); */
    var btnValue =$(this).val();
    $('select[name="searchKey"]').attr("required" , false);
    $('input[name="searchValue"]').attr("required" , false);
    $('select[name="searchKey2"]').attr("required" , false);
    $('input[name="searchValue2"]').attr("required" , false);
    $('input[name="searchStart"]').attr("required" , false);
    $('input[name="searchEnd"]').attr("required" , false);

});

/*3개월 버튼 클릭시 */
$('.3monthBtn').click(function(){
    /* Swal.fire($(this).val()); */
    var btnValue =$(this).val();
    $('select[name="searchKey"]').attr("required" , false);
    $('input[name="searchValue"]').attr("required" , false);
    $('select[name="searchKey2"]').attr("required" , false);
    $('input[name="searchValue2"]').attr("required" , false);
    $('input[name="searchStart"]').attr("required" , false);
    $('input[name="searchEnd"]').attr("required" , false);

```

```

});
/*1년 버튼 클릭시 */
$('.yearBtn').click(function(){
    /* Swal.fire($(this).val()); */
    var btnValue = $(this).val();
    $('select[name="searchKey"]').attr("required" , false);
    $('input[name="searchValue"]').attr("required" , false);
    $('select[name="searchKey2"]').attr("required" , false);
    $('input[name="searchValue2"]').attr("required" , false);
    $('input[name="searchStart"]').attr("required" , false);
    $('input[name="searchEnd"]').attr("required" , false);

});

/*sweetalret start*/
$('.cateDelete').click(function(){
    console.log($(this).parents('tr').find('.cateCode').text());
    var cateCode = $(this).parents('tr').find('.cateCode').text();

    Swal.fire({
        title: '정말 삭제 하시겠습니까?',
        text: "삭제한 내용은 되돌릴 수 없습니다.",
        icon: 'warning',
        showCancelButton: true,
        confirmButtonColor: '#3085d6',
        cancelButtonColor: '#d33',
        confirmButtonText: '삭제',
        cancelButtonText: '취소'
    }).then((result) => {
        if (result.isConfirmed) {
            location.href =
"/admin/challengeCategoryDelete?challengeCategoryId="+cateCode;

        }
    })
});
/*sweetalret End*/

/* 챌린지 카테고리 데이터 테이블 생성*/
var table1 = $('#challengeCategory').DataTable({
    order : false,
    lengthChange : false,
    pageLength : 15,
    filter: false,
    dom: '<"top">rt<"bottom">p<"clear">',
    "language": {
        "info": "전체 _TOTAL_건",

```

```

        "paginate": {
            "first":    "처음",
            "last":     "마지막",
            "next":     "다음",
            "previous": "이전"
        }
    }
});

/* 레코드 수 포맷 */
var info1 = table1.page.info();
$('#challengeCategory_info').html('전체 ' + info1.recordsTotal + '건')
).css('font-weight','bold');

$('#challengeCategory').click(function(){
    $('#challengeCategory_info').html('전체 ' + info1.recordsTotal + '건');
});

/*챌린지 세팅 데이터 테이블 생성*/
var table2 = $('#challengeSetting').DataTable({
    order : false,
    lengthChange : false,
    pageLength : 15,
    filter: false,
    dom: '<"top">rt<"bottom">p><"clear">',

    "language": {
        "info": "전체 _TOTAL_건",
        "paginate": {
            "first":    "처음",
            "last":     "마지막",
            "next":     "다음",
            "previous": "이전"
        }
    }
});

/* 레코드 수 포맷 */
var info2 = table2.page.info();
$('#challengeSetting_info').html('전체 ' + info2.recordsTotal + '건')
).css('font-weight','bold');

$('#challengeSetting').click(function(){
    $('#challengeSetting_info').html('전체 ' + info2.recordsTotal + '건');
});

```

```
});
```

■ Controller

```
//챌린지 관리설정
@GetMapping("/configList")
public String challengeConfigList(Model model

,@RequestParam(value="searchKey", required = false) String searchKey

,@RequestParam(value="searchValue", required = false) String searchValue

,@RequestParam(value="searchDate", required = false) String searchDate

,@RequestParam(value="searchStart", required = false) String searchStart

,@RequestParam(value="searchEnd", required = false) String searchEnd

,@RequestParam(value="searchKey2", required = false) String searchKey2

,@RequestParam(value="searchValue2", required = false) String searchValue2

,@RequestParam(name="btnKey", required = false) String btnKey) {

    log.info("챌린지 카테고리 조회 btnKey : {}", btnKey);

    Map<String, Object> paramMap = new HashMap<String, Object>();
    /* paramMap.put("memberEmail", "id001@email.com"); */
    //searchKey = challengeCategoryName challengeCategoryName(property) ->
challengeCategoryName(column) => searchKey = challengeCategoryName
    //searchKey = eClassCategorySmallName eClassCategorySmallName(property) ->
eClassCategorySmallName(column) => searchKey = eClassCategorySmallName

    if(searchKey != null) {
        if("challengeCategoryName".equals(searchKey)) {
            searchKey = "challengeCategoryName";
            searchDate = "challengeCategoryRegDate";
        }else if("eClassCategorySmallName".equals(searchKey)) {
            searchKey = "eClassCategorySmallName";
            searchDate = "challengeCategoryRegDate";
        }
    }

    if(searchKey2 != null) {
        if("challengeSettingCode".equals(searchKey2)) {
            searchKey2 = "challengeSettingCode";
        }else if("challengeSettingUpdateDo".equals(searchKey2)) {
            searchKey2 = "challengeSettingUpdateDo";
        }
    }

    if(btnKey != null) {
```

```

        if("오늘".equals(btnKey)) {
            btnKey = "challengeCategoryRegDate";
        }else if("1주일".equals(btnKey)) {
            btnKey = "-1 WEEK";
        }else if("1개월".equals(btnKey)) {
            btnKey = "-1 MONTH";
        }else if("3개월".equals(btnKey)) {
            btnKey = "-3 MONTH";
        }else if("1년".equals(btnKey)) {
            btnKey = "-1 YEAR";
        }
    }

    paramMap.put("searchKey", searchKey);
    paramMap.put("searchValue", searchValue);
    paramMap.put("searchKey2", searchKey2);
    paramMap.put("searchValue2", searchValue2);
    paramMap.put("searchDate", searchDate);
    paramMap.put("searchStart", searchStart);
    paramMap.put("searchEnd", searchEnd);
    paramMap.put("btnKey", btnKey);

    //List<ChallengeCategory> challengeCategoryList =
challengeConfigService.getChallengeCategoryList(paramMap);
    List<Map<String, Object>> challengeCategoryList =
challengeConfigService.getChallengeCategoryList(paramMap);
    List<Map<String, Object>> challengeSettingList =
challengeConfigService.getChallengeSettingList(paramMap);

    paramMap = null;

    log.info("챌린지 카테고리 조회 challengeCategoryList : {}", challengeCategoryList);

    model.addAttribute("title", "챌린지 관리 설정");
    model.addAttribute("leftMenuList", "챌린지");
    model.addAttribute("challengeCategoryList", challengeCategoryList);
    model.addAttribute("challengeSettingList", challengeSettingList);
    return "challenge/challengeConfig/challengeConfigList";

}
//챌린지 카테고리 삭제처리
@GetMapping("/challengeCategoryDelete")
public String challengeCategoryDelete(Model model

@RequestParam(name="challengeCategoryCode", required = false) String challengeCategoryCode) {

    log.info("챌린지 카테고리 삭제버튼 눌렀을때 스트링 값 challengeCategoryCode : {}",
challengeCategoryCode);

```



```

//챌린지 카테고리 코드에 따른 신고내역 삭제(챌린지 카테고리 삭제처리 )
challengeConfigService.categoryDeleteBycateCode(challengeCategoryCode);

return "redirect:/admin/configList";

```

```

}

```

■ Service

```

//챌린지 카테고리 조회
public List<Map<String, Object>> getChallengeCategoryList(Map<String, Object> paramMap){
    //List<ChallengeCategory> challengeCategoryList =
challengeConfigMapper.getChallengeCategoryList(paramMap);
    List<Map<String, Object>> challengeCategoryList =
challengeConfigMapper.getChallengeCategoryList(paramMap);

    return challengeCategoryList;
}

//챌린지 세팅 목록 조회
public List<ChallengeSetting> getChallengeSettingList(){
    List<ChallengeSetting> challengeSettingList =
challengeConfigMapper.getChallengeSettingList();

    return challengeSettingList;
}

//카테고리 삭제처리 프로세스
public void categoryDeleteBycateCode(String challengeCategoryCode) {
    //챌린지 카테고리 코드에 따른 신고내역 삭제(챌린지 카테고리 삭제처리 )
    challengeConfigMapper.reportDeleteBycateCode(challengeCategoryCode);

    //챌린지 카테고리 코드에 따른 포인트내역 삭제(챌린지 카테고리 삭제처리 )
    challengeConfigMapper.pointDeleteBycateCode(challengeCategoryCode);

    //챌린지 카테고리 코드에 따른 실행내역 삭제(챌린지 카테고리 삭제처리 )
    challengeConfigMapper.planDoDeleteBycateCode(challengeCategoryCode);

    //챌린지 카테고리 코드에 따른 참가내역 삭제(챌린지 카테고리 삭제처리 )
    challengeConfigMapper.enterDeleteBycateCode(challengeCategoryCode);

    //챌린지 카테고리 코드에 따른 모집계획 내역 삭제(챌린지 카테고리 삭제처리 )
    challengeConfigMapper.planDeleteBycateCode(challengeCategoryCode);

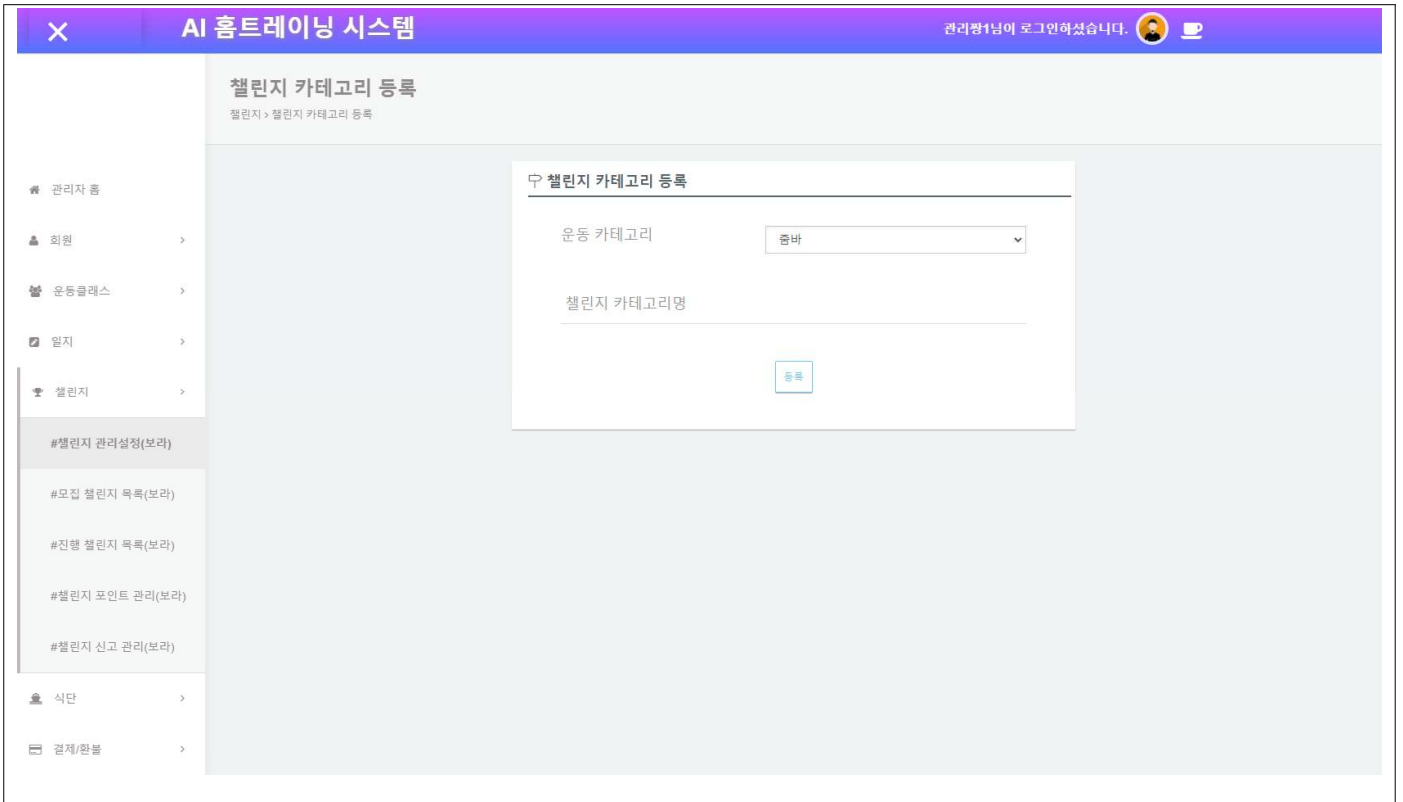
    //챌린지 카테고리 코드에 따른 모집내역 삭제(챌린지 카테고리 삭제처리 )
    challengeConfigMapper.gatherDeleteBycateCode(challengeCategoryCode);
}

```

```
//챌린지 카테고리 코드에 따른 카테고리 삭제(챌린지 카테고리 삭제처리 )  
challengeConfigMapper.challengeCategoryDelete(challengeCategoryId);
```

```
}
```

6.3.13 챌린지 카테고리 등록



챌린지 카테고리 등록

챌린지 카테고리 등록이 가능한 화면

■ javascript

```
$("#challengeCategoryForm").validate({
    errorElement: "em",
    errorPlacement: function(error, element) {
        $(element.parent("div")).addClass("form-animate-error");
        error.appendTo(element.parent("div"));
    },
    success: function(label) {
        $(label.parent("div")).removeClass("form-animate-error");
    },
    rules: {
        validate_categoryName: "required",
        validate_eClassCategory: "required"
    },
    messages: {
        validate_categoryName: "카테고리명을 입력해주세요.",
        validate_eClassCategory: "카테고리를 선택해주세요."
    }
});
```

```
});
```

■ Controller

```
//챌린지 카테고리 등록폼
@GetMapping("/challengeCategoryInsert")
public String challengeCategoryInsert(Model model) {

    Map<String, Object> paramMap = new HashMap<String, Object>();
    /* paramMap.put("memberEmail", "id001@email.com"); */

    //운동클래스카테고리small 조회
    List<EClassCategorySmall> classCategoryList =
challengeConfigService.getClassCategoryList();

    paramMap = null;

    log.info("운동클래스카테고리small 조회 classCategoryList : {}", classCategoryList);

    model.addAttribute("title", "챌린지 카테고리 등록");
    model.addAttribute("leftMenuList", "챌린지");
    model.addAttribute("classCategoryList", classCategoryList);

    return "challenge/challengeConfig/challengeCategoryInsert";

}

//챌린지 카테고리 등록 처리
@PostMapping("/challengeCategoryInsert")
public String challengeCategoryInsert(ChallengeCategory challengeCategory, HttpSession
session) {

    log.info("챌린지 카테고리 등록 폼에서 입력받은 데이터: {}", challengeCategory); //받은
내용이 여기{}에 담긴다.

    String sEmail = (String) session.getAttribute("SEMAIL");

    challengeConfigService.challengeCategoryInsert(challengeCategory, sEmail);

    return "redirect:/admin/configList";

}
```

■ Service

```
//카테고리 등록 처리
public int challengeCategoryInsert(ChallengeCategory challengeCategory, String sEmail) {
```

```
//pk컬럼에 들어갈 코드를 자동으로 만들어주는 Mapper //pk로 쓸 db의 컬럼명
//코드가 들어갈 db의 테이블명
String newCode = commonMapper.getNewCode("challengeCategoryCode",
"challengecategory");

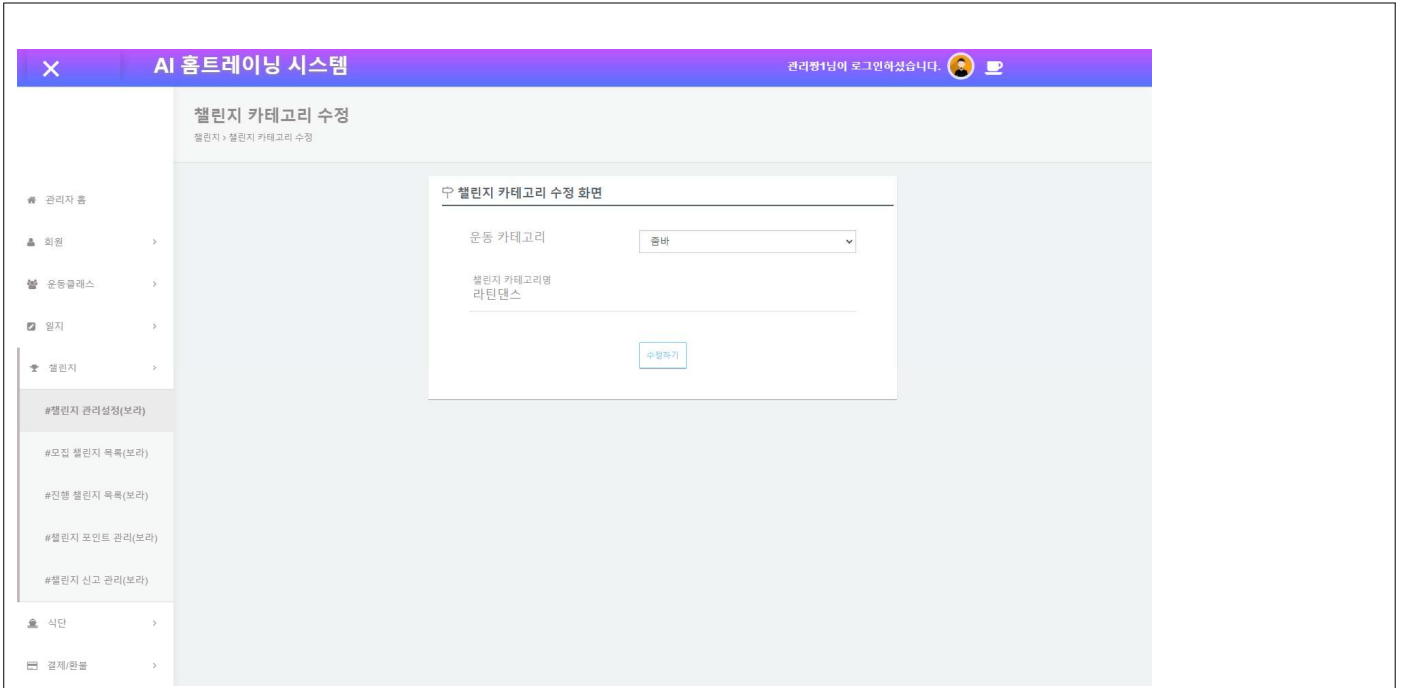
//dto에 위에서 만들어진 코드를 세팅해주기
challengeCategory.setChallengeCategoryCode(newCode);
challengeCategory.setMemberEmail(sEmail);

int result = challengeConfigMapper.challengeCategoryInsert(challengeCategory);

return result;

}
```

6.3.14 챌린지 카테고리 수정



챌린지 카테고리 수정

챌린지 카테고리 수정이 가능한 화면

■ JavaScript

```

$("#challengeCategoryUpdate").validate({
    errorElement: "em",
    errorPlacement: function(error, element) {
        $(element.parent("div")).addClass("form-animate-error");
        error.appendTo(element.parent("div"));
    },
    success: function(label) {
        $(label.parent("div")).removeClass("form-animate-error");
    },
    rules: {
        validate_categoryName: "required",
        validate_eClassCategory: "required"
    },
    messages: {
        validate_categoryName: "카테고리명을 입력해주세요.",
        validate_eClassCategory: "카테고리를 선택해주세요."
    }
});

```

■ Controller

```
//챌린지 카테고리 수정 화면
```

```

    @GetMapping("/challengeCategoryUpdate")
    public String challengeCategoryUpdate(Model model

    @RequestParam(name="challengeCategoryId", required = false) String challengeCategoryId) {

        log.info("챌린지 카테고리 수정 화면 폼 쿼리 스트링 challengeCategoryId : {}",
challengeCategoryId);

        //카테고리 코드별 챌린지 카테고리 정보 조회
        ChallengeCategory challengeCategory
challengeConfigService.getChallengeCategoryByCode(challengeCategoryId);

        //운동클래스카테고리small 조회
        List<EClassCategorySmall> classCategoryList
challengeConfigService.getClassCategoryList();

        log.info("카테고리 코드별 챌린지 카테고리 정보 조회 challengeCategory : {}",
challengeCategory);
        log.info("운동클래스카테고리small 조회 classCategoryList : {}", classCategoryList);

        model.addAttribute("title", "챌린지 카테고리 수정");
        model.addAttribute("leftMenuList", "챌린지");
        model.addAttribute("challengeCategory", challengeCategory);
        model.addAttribute("classCategoryList", classCategoryList);

        return "challenge/challengeConfig/challengeCategoryUpdate";
    }

    //챌린지 카테고리 수정 처리
    @PostMapping("/challengeCategoryUpdate")
    public String challengeCategoryUpdate(ChallengeCategory challengeCategory, HttpSession
session) {

        log.info("챌린지 카테고리 수정 폼에서 입력받은 데이터: {}", challengeCategory); //받은
내용이 여기{}에 담긴다.

        String sEmail = (String) session.getAttribute("SEMAIL");

        challengeConfigService.challengeCategoryUpdate(challengeCategory, sEmail);

        return "redirect:/admin/configList";
    }
}

```

■ Service

```
//운동 클래스 카테고리 small 조회
```

```

public List<EClassCategorySmall> getClassCategoryList() {

    List<EClassCategorySmall>                classCategoryList                =
challengeConfigMapper.getClassCategoryList();

    return classCategoryList;

}

//카테고리 코드별 챌린지 카테고리 정보 조회
public ChallengeCategory getChallengeCategoryByCode(String challengeCategoryCode) {

    ChallengeCategory                challengeCategory                =
challengeConfigMapper.getChallengeCategoryByCode(challengeCategoryCode);

    return challengeCategory;

}
//카테고리 수정처리
public int challengeCategoryUpdate(ChallengeCategory challengeCategory, String sEmail) {

    //session에서 받아온 이메일값
    challengeCategory.setMemberUpdateEmail(sEmail);

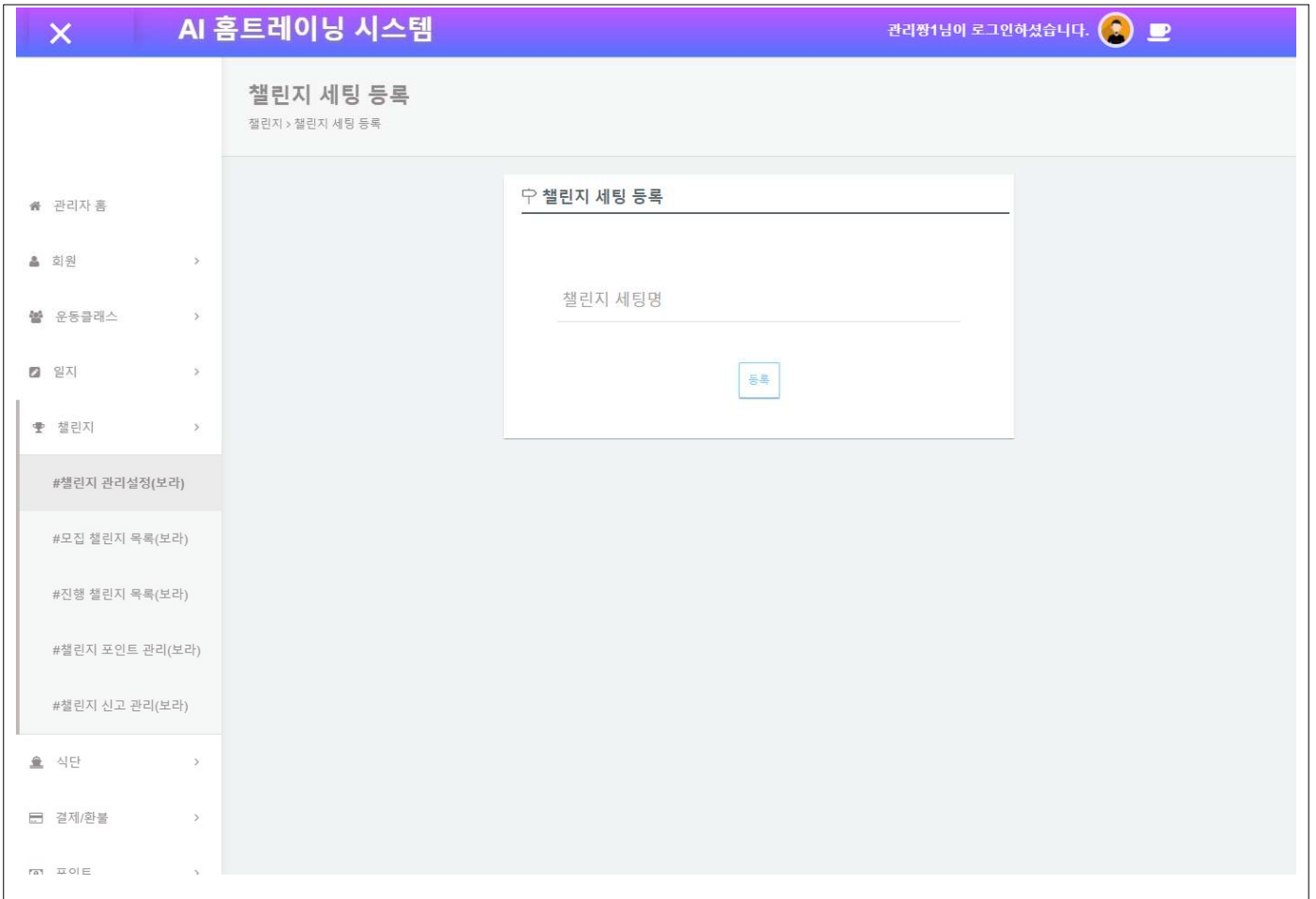
    int result = challengeConfigMapper.challengeCategoryUpdate(challengeCategory);

    return result;

}

```


6.3.15 챌린지 세팅 등록



챌린지 세팅 등록

챌린지 세팅 등록이 가능한 화면

■ javascript

```
$("#challengeSettingForm").validate({
    errorElement: "em",
    errorPlacement: function(error, element) {
        $(element.parent("div").addClass("form-animate-error"));
        error.appendTo(element.parent("div"));
    },
    success: function(label) {
        $(label.parent("div").removeClass("form-animate-error"));
    },
    rules: {
        validate_settingName: "required",
    },
    messages: {
        validate_settingName: "카테고리명을 입력해주세요.",
    }
});
```

```
});
```

■ Controller

```
//챌린지 세팅 등록
@GetMapping("/challengeSettingInsert")
public String challengeSettingInsert(Model model) {

    model.addAttribute("title", "챌린지 세팅 등록");
    model.addAttribute("leftMenuList", "챌린지");

    return "challenge/challengeConfig/challengeSettingInsert";

}

//챌린지 세팅 등록 처리
@PostMapping("/challengeSettingInsert")
public String challengeSettingInsert(ChallengeSetting challengeSetting) {

    log.info("챌린지 세팅 폼에서 입력받은 데이터: {}", challengeSetting);

    challengeConfigService.challengeSettingInsert(challengeSetting);

    return "redirect:/admin/configList";

}
```

■ Service

```
//카테고리 등록 처리
public int challengeCategoryInsert(ChallengeCategory challengeCategory, String sEmail) {
    //pk컬럼에 들어갈 코드를 자동으로 만들어주는 Mapper //pk로 쓸 db의 컬럼명
    //코드가 들어갈 db의 테이블명
    String newCode = commonMapper.getNewCode("challengeCategoryCode",
"challengecategory");

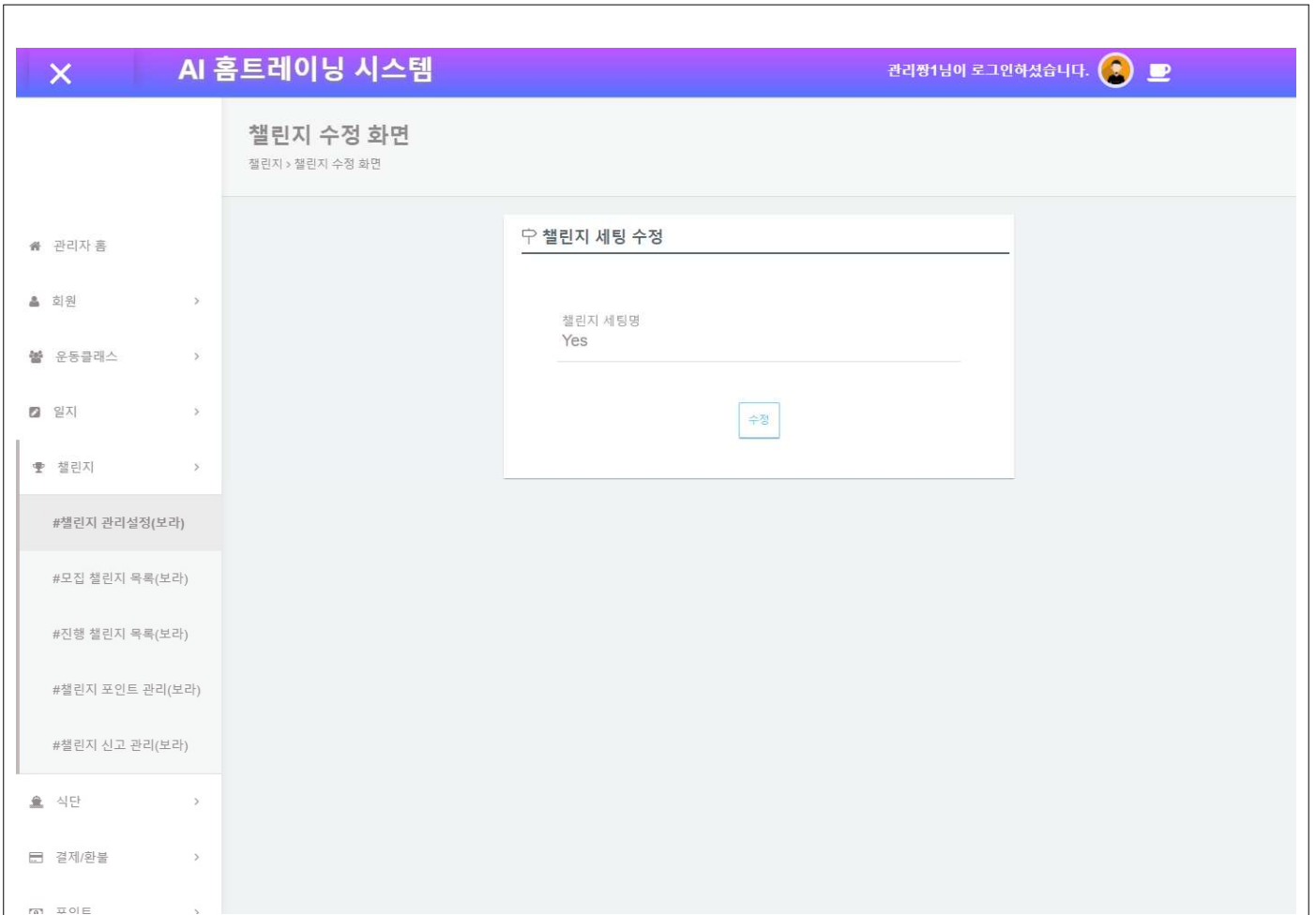
    //dto에 위에서 만들어진 코드를 세팅해주기
    challengeCategory.setChallengeCategoryCode(newCode);
    challengeCategory.setMemberEmail(sEmail); //로그인 처리가 아직 안이루어져서 임의로
해놓음!

    int result = challengeConfigMapper.challengeCategoryInsert(challengeCategory);

    return result;

}
```

6.3.16 챌린지 세팅 수정



챌린지 카테고리 수정

챌린지 세팅 수정이 가능한 화면

■ javascript

```
$("#challengeSettingForm").validate({
    errorElement: "em",
    errorPlacement: function(error, element) {
        $(element.parent("div").addClass("form-animate-error"));
        error.appendTo(element.parent("div"));
    },
    success: function(label) {
        $(label.parent("div").removeClass("form-animate-error"));
    },
    rules: {
        validate_settingName: "required",
    },
    messages: {
        validate_settingName: "카테고리명을 입력해주세요.",
    }
});
```

```
});
```

■ Controller

```
//챌린지 세팅 수정 화면
@GetMapping("/challengeSettingUpdate")
public String challengeSettingUpdate(Model model,
@RequestParam(name="challengeSettingCode", required = false) String challengeSettingCode) {

    log.info("챌린지 세팅 수정화면 품 쿼리 스트링 challengeSettingCode : {}",
challengeSettingCode);

    //챌린지 세팅 코드 별 세팅정보 조회
    ChallengeSetting challengeSetting =
challengeConfigService.getChallengeSettingByCode(challengeSettingCode);

    model.addAttribute("title", "챌린지 수정 화면");
    model.addAttribute("leftMenuList", "챌린지");
    model.addAttribute("challengeSetting", challengeSetting);

    return "challenge/challengeConfig/challengeSettingUpdate";

}

//챌린지 세팅 수정 처리
@PostMapping("/challengeSettingUpdate")
public String challengeSettingUpdate(ChallengeSetting challengeSetting) {

    log.info("챌린지 세팅 수정 품에서 입력받은 데이터 challengeSetting : {}",
challengeSetting);

    challengeConfigService.challengeSettingUpdate(challengeSetting);

    return "redirect:/admin/configList";

}
```

■ Service

```
//챌린지 세팅 수정 처리
public void challengeSettingUpdate(ChallengeSetting challengeSetting) {

    challengeConfigMapper.challengeSettingUpdate(challengeSetting);

}
```

6.3.17 모집 챌린지 목록

모집 챌린지 목록

모집 > 모집 챌린지 목록

모집 챌린지 목록

카테고리 검색 | 모집코드 | 검색키워드를 입력해주세요.

등록일 모집일

기간 검색 | 시작일 | 부터 | 종료일 | 까지 | 오늘 | 1주일 | 1개월 | 3개월 | 1년

검색 Q | 초기화

전체 5 건

챌린지 모집코드	챌린지 등록자 이메일	챌린지 카테고리명	챌린지 모집명	챌린지 등록날짜	챌린지 모집 시작일	챌린지 모집 종료일
challengeGatherCode001	id002@email.com	라틴댄스	하루 10분 요가해요	2022-03-18	2022-03-18	2022-03-29
challengeGatherCode002	id005@email.com	아쉬탕카요가	아쉬탕카 중급	2022-03-17	2022-03-17	2022-03-22
challengeGatherCode003	id006@email.com	기초는기초	한달 넥스트레벨 완성하기	2022-03-15	2022-03-15	2022-03-24
challengeGatherCode_1	id002@email.com	이시슬님과 하는 방송댄스	헤이마마	2022-04-06	2022-04-16	2022-04-30
challengeGatherCode_2	id002@email.com	댄스댄스댄스	하루 100개 스쿼트	2022-04-11	2022-03-27	2022-04-09

이전 1 다음

하루 10분 요가해요

챌린지 소개	하루 10분 요가하는 챌린지입니다.
챌린지 최대참여인원수	5
챌린지 인증방법	사진
챌린지 인증내용	사진을 찍어서 2장 이상 올려주세요
챌린지 인증빈도	30
챌린지 참가자 이메일	id003@email.com
챌린지 시작일	2022-03-30
챌린지 종료일	2022-04-30

닫기

모집 챌린지 목록 조회

모집 챌린지 목록 조회가 가능한 화면
 행 클릭시 해당 챌린지의 세부 내용 확인 가능

■ JavaScript

```

/* datepicker Start */
$('.input-daterange').datepicker({
  format: "yyyy-mm-dd", //데이터 포맷 형식(yyyy : 년 mm : 월 dd : 일 )
  /* startDate: '-3d', //달력에서 선택 할 수 있는 가장 빠른 날짜. 이전으로
  */
});
  
```

```

는 선택 불가능 ( d : 일 m : 달 y : 년 w : 주) */
autoclose : true, //사용자가 날짜를 클릭하면 자동 캘린더가 닫히는 옵션
calendarWeeks : false, //캘린더 옆에 몇 주차인지 보여주는 옵션 기본값
false 보여주려면 true
clearBtn : false, //날짜 선택한 값 초기화 해주는 버튼 보여주는 옵션 기본
값 false 보여주려면 true
datesDisabled : ['2019-06-24','2019-06-26'],//선택 불가능한 일 설정 하
는 배열 위에 있는 format 과 형식이 같아야함.
disableTouchKeyboard : false, //모바일에서 플러그인 작동 여부 기
본값 false 가 작동 true가 작동 안함.
immediateUpdates: false, //사용자가 보는 화면으로 바로바로 날짜를 변
경할지 여부 기본값 :false
multidate : false, //여러 날짜 선택할 수 있게 하는 옵션 기본값 :false
multidateSeparator : ",", //여러 날짜를 선택했을 때 사이에 나타나는 글자
2019-05-01,2019-06-01
templates : {
    leftArrow: '&laquo;',
    rightArrow: '&raquo;'
}, //다음달 이전달로 넘어가는 화살표 모양 커스텀 마이징
showWeekDays : true ,// 위에 요일 보여주는 옵션 기본값 : true
title: "등록일", //캘린더 상단에 보여주는 타이틀
todayHighlight : true , //오늘 날짜에 하이라이팅 기능 기본값 :false
toggleActive : true, //이미 선택된 날짜 선택하면 기본값 : false인경우 그
대로 유지 true인 경우 날짜 삭제
weekStart : 0 ,//달력 시작 요일 선택하는 것 기본값은 0인 일요일
language : "ko" //달력의 언어 선택, 그에 맞는 js로 교체해줘야한다.
});
/* datepicker End */

$('.searchBtn').click(function(){
    $('select[name="searchKey"]').attr("required" , true);
    $('input[name="searchValue"]').attr("required" , true);
    $('input[name="searchStart"]').attr("required" , true);
    $('input[name="searchEnd"]').attr("required" , true);
    $('input[name="searchDate"]').attr("required" , true);
});
$('.searchResetBtn').click(function(){
    $('select[name="searchKey"]').attr("required" , false);
    $('input[name="searchValue"]').attr("required" , false);
    $('input[name="searchStart"]').attr("required" , false);
    $('input[name="searchEnd"]').attr("required" , false);
    $('input[name="searchDate"]').attr("required" , false);
});

/* 데이터 테이블 생성*/
var table = $('#challengeList').DataTable({
    order : [ 0, 'asc' ],

```

```

lengthChange : false,
pageLength : 15,
filter: false,
dom: '<"top">rt<"bottom">p><"clear">',

    "language": {
        "info": "전체 _TOTAL_건",
        "paginate": {
            "first": "처음",
            "last": "마지막",
            "next": "다음",
            "previous": "이전"
        }
    }
});

/* 레코드 수 포맷 */
var info = table.page.info();
$('#challengeList_info').html('전체 ' + info.recordsTotal + ' 건')
).css('font-weight','bold');

$('#challengeList').click(function(){
    $('#challengeList_info').html('전체 ' + info.recordsTotal + ' 건');
});

var $datatr = $('#challengeList tbody tr');

$datatr.click(function (event) {
    event.preventDefault();
    console.log($(this).find('#challengeGatherCode').text(), 'find
challengeGatherCode');
    var challengeGatherCode =
$(this).find('#challengeGatherCode').text();

    $.ajax({
        url: '/admin/listDetail'
        ,type : 'post'
        ,data : {
            'challengeGatherCode' : challengeGatherCode
        }
        ,dataType : 'json'
        ,success: function(data){
            console.log(data);
            console.log(data.length);
            for(var i=0; i<data.length; i++){

                var modalhader = '<h5 class="modal-title"
id="GatherListModalLabel" style="font-weight: bold; font-size: large;">' +data[i].challengeGatherName

```

```

+ '</h5>';

var modalbody = '';

modalbody = '<table class="table-striped
table-bordered" cellpadding="5" cellspacing="0" border="0" style="width: 100% ">'+
            '<tr>'+
            '<td>챌린지 소
개</td>'+
            '<td>'+data[i].challengeGatherIntro+'</td>'+
            '</tr>'+
            '<tr>'+
            '<td>챌린지 최
대참여인원수</td>'+
            '<td>'+data[i].challengeGatherMaxPeople+'</td>'+
            '</tr>'+
            '<tr>'+
            '<td>챌린지 인
증방법</td>'+
            '<td>'+data[i].challengeGatherProveMethod+'</td>'+
            '</tr>'+
            '<tr>'+
            '<td>챌린지 인
증내용</td>'+
            '<td>'+data[i].challengeGatherProveContent+'</td>'+
            '</tr>'+
            '<tr>'+
            '<td>챌린지 인
증빈도</td>'+
            '<td>'+data[i].challengeGatherProveFrequency+'</td>'+
            '</tr>'+
            '<tr>'+
            '<td>챌린지 참
가자 이메일</td>'+
            '<td>'+data[i].memberEnterEmail+'</td>'+
            '</tr>'+
            '<tr>'+
            '<td>챌린지 시
작일</td>'+
            '<td>'+data[i].challengeStartDate+'</td>'+
            '</tr>'+
            '<tr>'+
            '<td>챌린지 종

```



```

료일</td>'+
'<td>'+data[i].challengeEndDate+'</td>'+
'</tr>'+
'</table>';

$('.modal-header').empty();
$('.modal-body').empty();

$('.modal-header').prepend(modalheader);
$('.modal-body').append(modalbody);

$('#GatherListModal').modal();
    }
    }
});
});
});

```

■ Controller

```

//챌린지 모집 중인 목록
@GetMapping("/challengeList")
public String challengeList(Model model
    ,@RequestParam(value="searchKey",
required = false) String searchKey
    ,@RequestParam(value="searchValue",
required = false) String searchValue
    ,@RequestParam(value="searchDate",
required = false) String searchDate
    ,@RequestParam(value="searchStart",
required = false) String searchStart
    ,@RequestParam(value="searchEnd",
required = false) String searchEnd
    ,@RequestParam(value="gatherStartDate",
required = false) String gatherStartDate
    ,@RequestParam(value="gatherEndDate",
required = false) String gatherEndDate) {

    Map<String, Object> paramMap = new HashMap<String, Object>();

    if(searchKey != null) {
        if("challengeGatherCode".equals(searchKey)) {
            searchKey = "challengeGatherCode";
            if(searchDate != null) {
                if("challengeGatherRegDate".equals(searchDate)) {
                    searchDate = "challengeGatherRegDate";

```

```

                }else{
                    gatherStartDate = "challengeGatherStartDate";
                    gatherEndDate = "challengeGatherEndDate";
                }
            }
        }else if("memberEmail".equals(searchKey)) {
            searchKey = "cg.memberEmail";
            if(searchDate != null) {
                if("challengeGatherRegDate".equals(searchDate)) {
                    searchDate = "challengeGatherRegDate";
                }else{
                    gatherStartDate = "challengeGatherStartDate";
                    gatherEndDate = "challengeGatherEndDate";
                }
            }
        }
        }else if("challengeCategoryName".equals(searchKey)) {
            searchKey = "challengeCategoryName";
            if(searchDate != null) {
                if("challengeGatherRegDate".equals(searchDate)) {
                    searchDate = "challengeGatherRegDate";
                }else{
                    gatherStartDate = "challengeGatherStartDate";
                    gatherEndDate = "challengeGatherEndDate";
                }
            }
        }
        }else if("challengeGatherName".equals(searchKey)) {
            searchKey = "challengeGatherName";
            if(searchDate != null) {
                if("challengeGatherRegDate".equals(searchDate)) {
                    searchDate = "challengeGatherRegDate";
                }else{
                    gatherStartDate = "challengeGatherStartDate";
                    gatherEndDate = "challengeGatherEndDate";
                }
            }
        }
    }
}

```

```

paramMap.put("searchKey", searchKey);
paramMap.put("searchValue", searchValue);
paramMap.put("searchDate", searchDate);
paramMap.put("gatherStartDate", gatherStartDate);
paramMap.put("searchStart", searchStart);
paramMap.put("searchEnd", searchEnd);
paramMap.put("gatherEndDate", gatherEndDate);

```

```

        List<Map<String, Object>> gatherList =
challengeConfigService.getGetherList(paramMap);

        paramMap = null;

        log.info("모집챌린지 gatherList : {}", gatherList);

        model.addAttribute("title", "모집 챌린지 목록");
        model.addAttribute("leftMenuList", "모집");
        model.addAttribute("gatherList", gatherList);

        return "challenge/challengeConfig/challengeList";
    }

    //ajax 챌린지 모집 목록 자세한 내용 조회
    @PostMapping("/listDetail")
    @ResponseBody
    public List<ChallengeGather> getListDetailByCode(String challengeGatherCode){

        log.info("모집챌린지 String challengeGatherCode : {}", challengeGatherCode);

        List<ChallengeGather> listDetail =
challengeConfigService.getListDetailByCode(challengeGatherCode);
        log.info("모집 챌린지 세부 내용 조회 ListDetail : {}", listDetail);

        return listDetail;
    }
}

```

■ Service

```

//모집 챌린지 목록 조회
    public List<Map<String, Object>> getGetherList(Map<String, Object> paramMap){

        List<Map<String, Object>> gatherList =
challengeConfigMapper.getGetherList(paramMap);

        return gatherList;
    }

    //모집 챌린지 코드별 모집 챌린지 세부 정보 조회
    public List<ChallengeGather> getListDetailByCode(String challengeGatherCode){

        List<ChallengeGather> listDetail =
challengeConfigMapper.getListDetailByCode(challengeGatherCode);
    }
}

```

```
return listDetail;
```

```
}
```

6.3.18 진행챌린지 목록 조회

The screenshot shows a web interface for 'AI 홈트레이닝 시스템'. The main content area displays a list of '진행 챌린지 목록' (Ongoing Challenge List) with columns for challenge ID, email, name, category, start/end dates, and status. A modal window is open over the '하루 10분 요가해요' challenge, showing details such as introduction, maximum participants (5), submission method (photo), submission content (upload 2 photos), submission limit (30), participant email (id003@email.com), start date (2022-03-30), and end date (2022-04-30).

진행 챌린지 목록 조회

진행 중인 챌린지 목록 조회가 가능한 화면
 행 클릭시 해당 챌린지의 세부 정보 조회 가능

■ JavaScript

```

/* datepicker Start */
$('.input-daterange').datepicker({
    format: "yyyy-mm-dd", //데이터 포맷 형식(yyyy : 년 mm : 월 dd :
    
```

```

일 )
        /* startDate: '-3d', //달력에서 선택 할 수 있는 가장 빠른 날짜. 이전으로
는 선택 불가능 ( d : 일 m : 달 y : 년 w : 주) */
        autoclose : true, //사용자가 날짜를 클릭하면 자동 캘린더가 닫히는 옵션
        calendarWeeks : false, //캘린더 옆에 몇 주차인지 보여주는 옵션 기본값
false 보여주려면 true
        clearBtn : false, //날짜 선택한 값 초기화 해주는 버튼 보여주는 옵션 기본
값 false 보여주려면 true
        datesDisabled : ['2019-06-24','2019-06-26'],//선택 불가능한 일 설정 하
는 배열 위에 있는 format 과 형식이 같아야함.
        disableTouchKeyboard : false, //모바일에서 플러그인 작동 여부 기
본값 false 가 작동 true가 작동 안함.
        immediateUpdates: false, //사용자가 보는 화면으로 바로바로 날짜를 변
경할지 여부 기본값 :false
        multigate : false, //여러 날짜 선택할 수 있게 하는 옵션 기본값 :false
        multigateSeparator : ",", //여러 날짜를 선택했을 때 사이에 나타나는 글자
2019-05-01,2019-06-01
        templates : {
            leftArrow: '&laquo;',
            rightArrow: '&raquo;'
        }, //다음달 이전달로 넘어가는 화살표 모양 커스텀 마이징
        showWeekDays : true ,// 위에 요일 보여주는 옵션 기본값 : true
        title: "등록일", //캘린더 상단에 보여주는 타이틀
        todayHighlight : true , //오늘 날짜에 하이라이팅 기능 기본값 :false
        toggleActive : true, //이미 선택된 날짜 선택하면 기본값 : false인경우 그
대로 유지 true인 경우 날짜 삭제
        weekStart : 0 ,//달력 시작 요일 선택하는 것 기본값은 0인 일요일
        language : "ko" //달력의 언어 선택, 그에 맞는 js로 교체해줘야한다.
    });
    /* datepicker End */
$('.searchBtn').click(function(){
        $('select[name="searchKey"]').attr("required" , true);
        $('input[name="searchValue"]').attr("required" , true);
        $('input[name="searchStart"]').attr("required" , true);
        $('input[name="searchEnd"]').attr("required" , true);
        $('input[name="searchDate"]').attr("required" , true);
    });
$('.searchResetBtn').click(function(){
        $('select[name="searchKey"]').attr("required" , false);
        $('input[name="searchValue"]').attr("required" , false);
        $('input[name="searchStart"]').attr("required" , false);
        $('input[name="searchEnd"]').attr("required" , false);
        $('input[name="searchDate"]').attr("required" , false);
    });

    /* 데이터 테이블 생성*/
    var table = $('#IngList').DataTable({
        order : [ 0, 'asc' ],

```

```

lengthChange : false,
pageLength : 15,
filter: false,
dom: '<"top">rt<"bottom">p><"clear">',

    "language": {
        "info": "전체 _TOTAL_건",
        "paginate": {
            "first": "처음",
            "last": "마지막",
            "next": "다음",
            "previous": "이전"
        }
    }
});

/* 레코드 수 포맷 */
var info = table.page.info();
$('#IngList_info').html('전체 ' + info.recordsTotal + ' 건')
).css('font-weight','bold');

$('#IngList').click(function(){
    $('#IngList_info').html('전체 ' + info.recordsTotal + ' 건');
});

var $datatr = $('#IngList tbody tr');

$datatr.click(function (event) {
    event.preventDefault();
    console.log($(this).find('#challengeGatherCode').text(), 'find
challengeGatherCode');

    var challengeGatherCode =
$(this).find('#challengeGatherCode').text();

    $.ajax({
        url: '/admin/listDetail'
        ,type : 'post'
        ,data : {
            'challengeGatherCode' : challengeGatherCode
        }
        ,dataType : 'json'
        ,success: function(data){
            console.log(data);
            console.log(data.length);
            for(var i=0; i<data.length; i++){

                var modalhader = '<h5 class="modal-title"

```

```

id="ChallengeIngListModalLabel" style="font-weight: bold; font-size: large;">
+data[i].challengeGatherName + '</h5>';

var modalbody = '';

modalbody = '<table class="table-striped
table-bordered" cellpadding="5" cellspacing="0" border="0" style="width: 100% ">'+

'<tr>'+
'<td>챌린지 소
개</td>'+
'<td>'+data[i].challengeGatherIntro+'</td>'+
'</tr>'+
'<tr>'+
'<td>챌린지 최
대참여인원수</td>'+
'<td>'+data[i].challengeGatherMaxPeople+'</td>'+
'</tr>'+
'<tr>'+
'<td>챌린지 인
증방법</td>'+
'<td>'+data[i].challengeGatherProveMethod+'</td>'+
'</tr>'+
'<tr>'+
'<td>챌린지 인
증내용</td>'+
'<td>'+data[i].challengeGatherProveContent+'</td>'+
'</tr>'+
'<tr>'+
'<td>챌린지 인
증빈도</td>'+
'<td>'+data[i].challengeGatherProveFrequency+'</td>'+
'</tr>'+
'<tr>'+
'<td>챌린지 참
가자 이메일</td>'+
'<td>'+data[i].memberEnterEmail+'</td>'+
'</tr>'+
'<tr>'+
'<td>챌린지 시
작일</td>'+
'<td>'+data[i].challengeStartDate+'</td>'+
'</tr>'+
'<tr>'+
'<td>챌린지 종

```



```

료일</td>'+
'<td>'+data[i].challengeEndDate+'</td>'+
'</tr>'+
'</table>';

$('.modal-header').empty();
$('.modal-body').empty();

$('.modal-header').prepend(modalheader);
$('.modal-body').append(modalbody);

$('#ChallengeIngListModal').modal();
    }
}
});
});

```

■ Controller

```

//진행중인 챌린지 목록
@GetMapping("/challengeListIng")
public String challengeListIng(Model model

,@RequestParam(value="searchKey", required = false) String searchKey

,@RequestParam(value="searchValue", required = false) String searchValue

,@RequestParam(value="searchDate", required = false) String searchDate

,@RequestParam(value="ingStartDate", required = false) String ingStartDate

,@RequestParam(value="searchStart", required = false) String searchStart

,@RequestParam(value="searchEnd", required = false) String searchEnd

,@RequestParam(value="ingEndDate", required = false) String ingEndDate) {

    Map<String, Object> paramMap = new HashMap<String, Object>();

    if(searchKey != null) {
        if("challengeGatherCode".equals(searchKey)) {
            searchKey = "challengeGatherCode";
            if(searchDate != null) {
                if("challengeGatherRegDate".equals(searchDate)) {
                    searchDate = "challengeGatherRegDate";

```

```
                }else{
                    ingStartDate = "challengeGatherStartDate";
                    ingEndDate = "challengeGatherEndDate";
                }
            }
        }else if("memberEmail".equals(searchKey)) {
            searchKey = "cg.memberEmail";
            if(searchDate != null) {
                if("challengeGatherRegDate".equals(searchDate)) {
                    searchDate = "challengeGatherRegDate";
                }else{
                    ingStartDate = "challengeGatherStartDate";
                    ingEndDate = "challengeGatherEndDate";
                }
            }
        }else if("challengeCategoryName".equals(searchKey)) {
            searchKey = "challengeCategoryName";
            if(searchDate != null) {
                if("challengeGatherRegDate".equals(searchDate)) {
                    searchDate = "challengeGatherRegDate";
                }else{
                    ingStartDate = "challengeGatherStartDate";
                    ingEndDate = "challengeGatherEndDate";
                }
            }
        }else if("challengeGatherName".equals(searchKey)) {
            searchKey = "challengeGatherName";
            if(searchDate != null) {
                if("challengeGatherRegDate".equals(searchDate)) {
                    searchDate = "challengeGatherRegDate";
                }else{
                    ingStartDate = "challengeGatherStartDate";
                    ingEndDate = "challengeGatherEndDate";
                }
            }
        }
    }

    paramMap.put("searchKey", searchKey);
    paramMap.put("searchValue", searchValue);
    paramMap.put("searchDate", searchDate);
    paramMap.put("ingStartDate", ingStartDate);
    paramMap.put("searchStart", searchStart);
    paramMap.put("searchEnd", searchEnd);
    paramMap.put("ingEndDate", ingEndDate);

    List<Map<String, Object>> challengeIngList =
```

```

challengeConfigService.getChallengeIngList(paramMap);

    paramMap = null;

    log.info("진행 챌린지 목록 조회 challengeIngList: {}", challengeIngList);

    model.addAttribute("title", "진행 챌린지 목록");
    model.addAttribute("leftMenuList", "챌린지");
    model.addAttribute("challengeIngList", challengeIngList);

    return "challenge/challengeConfig/challengeListIng";

}
//ajax 챌린지 모집 목록 자세한 내용 조회
@PostMapping("/listDetail")
@ResponseBody
public List<ChallengeGather> getListDetailByCode(String challengeGatherCode){

    log.info("모집챌린지 String challengeGatherCode : {}", challengeGatherCode);

    List<ChallengeGather> listDetail =
challengeConfigService.getListDetailByCode(challengeGatherCode);
    log.info("모집 챌린지 세부 내용 조회 ListDetail : {}", listDetail);

    return listDetail;

}

```

■ Service

```

//진행 챌린지 목록 조회
public List<Map<String, Object>> getChallengeIngList(Map<String, Object> paramMap) {

    List<Map<String, Object>> challengeIngList =
challengeConfigMapper.getChallengeIngList(paramMap);

    return challengeIngList;

}

```

6.3.19 챌린지 포인트 관리

챌린지 포인트 관리

포인트 지급 예정인 챌린지 목록 조회가 가능
 포인트지급 버튼 클릭시 해당 챌린지의 포인트 관련 상세정보가 모달창에서 조회 가능
 모달창에서 포인트 지급 버튼 클릭시 포인트 지급 페이지로 이동

■ JavaScript

```

/* datepicker Start */
$('.input-daterange').datepicker({
    format: "yyyy-mm-dd", //데이터 포맷 형식(yyyy : 년 mm : 월 dd :
일 )
    /* startDate: '-3d', //달력에서 선택 할 수 있는 가장 빠른 날짜. 이전으로
는 선택 불가능 ( d : 일 m : 달 y : 년 w : 주) */
    autoclose : true, //사용자가 날짜를 클릭하면 자동 캘린더가 닫히는 옵션

```

```

선
false 보여주려면 true
값 false 보여주려면 true
는 배열 위에 있는 format 과
본값 false 가 작동 true가 작동 안함.
경할지 여부 기본값 :false
2019-05-01,2019-06-01
templates : {
    leftArrow: '&laquo;',
    rightArrow: '&raquo;'
}, //다음달 이전달로 넘어가는 화살표 모양 커스텀 마이징
showWeekDays : true ,// 위에 요일 보여주는 옵션 기본값 : true
title: "등록일", //캘린더 상단에 보여주는 타이틀
todayHighlight : true , //오늘 날짜에 하이라이팅 기능 기본값 :false
toggleActive : true, //이미 선택된 날짜 선택하면 기본값 : false인경우 그
대로 유지 true인 경우 날짜 삭제
weekStart : 0 ,//달력 시작 요일 선택하는 것 기본값은 0인 일요일
language : "ko" //달력의 언어 선택, 그에 맞는 js로 교체해줘야한다.
});
/* datepicker End */
$('.searchBtn').click(function(){
    $('select[name="searchKey"]').attr("required" , true);
    $('input[name="searchValue"]').attr("required" , true);
    $('input[name="searchStart"]').attr("required" , true);
    $('input[name="searchEnd"]').attr("required" , true);
});
$('.searchResetBtn').click(function(){
    $('select[name="searchKey"]').attr("required" , false);
    $('input[name="searchValue"]').attr("required" , false);
    $('input[name="searchStart"]').attr("required" , false);
    $('input[name="searchEnd"]').attr("required" , false);
});

/* 데이터 테이블 생성*/
var table1 = $('#challengePointList').DataTable({
    order : false,
    lengthChange : false,
    pageLength : 15,
    filter: false,
    dom: '< "top"i>rt< "bottom"p>< "clear">',
    "language": {

```

```

        "info": "전체 _TOTAL_건",
        "paginate": {
            "first": "처음",
            "last": "마지막",
            "next": "다음",
            "previous": "이전"
        }
    });

    /* 레코드 수 포맷 */
    var info1 = table1.page.info();
    $('#challengePointList_info').html('전체 ' + info1.recordsTotal + '건')
    .css('font-weight','bold');

    $('#challengePointList').click(function(){
        $('#challengePointList_info').html('전체 ' + info1.recordsTotal + '건');
    });

    var $selectTr = $('#challengePointList tbody .pointGive');

    $selectTr.click(function (event){
        event.preventDefault();
        var pointCode = $(this).attr('data-code');
        console.log(pointCode);

        $.ajax({
            url: '/admin/pointDetail'
            ,type : 'post'
            ,data : {
                'challengePointGiveCode' : pointCode
            }
            ,dataType : 'json'
            ,success: function(data){
                console.log(data);
                console.log(data.length);
                for(var i=0; i<data.length; i++){

                    var modalhader = '<h5 class="modal-title"
id="pointDetailLabel" style="font-weight: bold; font-size: large;">
+data[i].challengeGather.challengeGatherName + '</h5>';

                    var modalbody = '';

                    modalbody = '<table
class="table-striped table-bordered" cellpadding="5" cellspacing="0" border="0" style="width: 100%
">+

```

```

'<tr>'+
                                                                    '<td>챌
린지 참여자 이메일</td>'+
                                                                    '</td>챌
'<td>'+data[i].challengeEnter.memberEmail+'</td>'+
                                                                    '</tr>'+
                                                                    '<tr>'+
                                                                    '<td>챌
린지 달성률</td>'+
                                                                    '</tr>'+
                                                                    '<tr>'+
                                                                    '<td>포
인트 지급기준</td>'+
                                                                    '</tr>'+
                                                                    '<tr>'+
                                                                    '<td>지
급 예정 포인트</td>'+
                                                                    '</tr>'+
                                                                    '</table>'+
                                                                    '<input
type="hidden" class="mt-2 p-2 form-control" name="challengePointGiveCode" readonly value="'+
pointCode +'>';

if(data[i].challengeGather.challengeGatherAchieveRate=='30'){
                                                                    //달성률 100으로 변경해야
                                                                    함.
                                                                    console.log(pointCode,'포인
                                                                    트코드', '버튼보인다 30일때');
                                                                    $('#pointGive').show();
                                                                    }else{
                                                                    console.log('버튼 안보인다
                                                                    30아닐때');
                                                                    $('#pointGive').hide();
                                                                    }

                                                                    $('modal-header').empty();
                                                                    $('modal-body').empty();

                                                                    $('modal-header').prepend(modalheader);
                                                                    $('modal-body').append(modalbody);

                                                                    $('#pointDetail').modal();

```

```

    }
    }
});
});

```

■ Controller

```

//챌린지 포인트 관리
@GetMapping("/challengePoint")
public String challengePoint(Model model
                                ,@RequestParam(value="searchKey",
required = false) String searchKey
                                ,@RequestParam(value="searchValue",
required = false) String searchValue
                                ,@RequestParam(value="searchDate",
required = false) String searchDate
                                ,@RequestParam(value="searchStart",
required = false) String searchStart
                                ,@RequestParam(value="searchEnd",
required = false) String searchEnd) {

    Map<String, Object> paramMap = new HashMap<String, Object>();

    if(searchKey != null) {
        if("challengeCategoryName".equals(searchKey)) {
            searchKey = "challengeCategoryName";
            searchDate = "challengeCategoryRegDate";
        }else if("eClassCategorySmallName".equals(searchKey)) {
            searchKey = "eClassCategorySmallName";
            searchDate = "challengeCategoryRegDate";
        }
    }

    paramMap.put("searchKey", searchKey);
    paramMap.put("searchValue", searchValue);
    paramMap.put("searchDate", searchDate);
    paramMap.put("searchStart", searchStart);
    paramMap.put("searchEnd", searchEnd);

    //챌린지 포인트 관리 목록 조회
    List<Map<String, Object>> pointList =
challengeConfigService.getChallengePointList(paramMap);

    paramMap = null;

    log.info("챌린지 포인트 목록 조회 pointList: {}", pointList);
}

```



```

        model.addAttribute("title", "챌린지 포인트 관리");
        model.addAttribute("leftMenuList", "챌린지");
        model.addAttribute("pointList", pointList);

        return "challenge/challengeConfig/challengePoint";
    }

    //ajax
    //챌린지 포인트 코드별 포인트 관리 세부 조회
    @PostMapping("/pointDetail")
    @ResponseBody
    public List<ChallengePointGive> getPointDetailByCode(String challengePointGiveCode){

        log.info("챌린지 포인트 관리 String challengePointGiveCode : {}",
challengePointGiveCode);

        List<ChallengePointGive> pointDetailList =
challengeConfigService.getPointDetailByCode(challengePointGiveCode);

        return pointDetailList;
    }
    //챌린지 포인트 지급 페이지로 연결
    @PostMapping("/challengePoint")
    public String challengePointGiveUpdate(ChallengePointGive challengePointGive) {
        log.info("모달에서 보낸 값 조회 challengePointGive : {}", challengePointGive);

        return "point/givePoint";
    }
}

```

■ Service

```

//챌린지 포인트 관리 목록 조회
    public List<Map<String, Object>> getChallengePointList(Map<String, Object> paramMap) {

        List<Map<String, Object>> pointList =
challengeConfigMapper.getChallengePointList(paramMap);

        return pointList;
    }

    //챌린지 포인트 코드별 포인트 관리 세부 조회
    public List<ChallengePointGive> getPointDetailByCode(String challengePointGiveCode) {

        List<ChallengePointGive> pointDetailList =
challengeConfigMapper.getPointDetailByCode(challengePointGiveCode);
    }
}

```

```
}    return pointDetailList;
```

6.3.20 챌린지 모집 목록

챌린지 모집 목록

챌린지 | 챌린지 모집 목록

모집 중인 챌린지

하루 10분 요가해요

모집인원 : 5

하루 10분 요가하는 챌린지입니다.
하루 10분 요가하는 챌린지입니다.

2022-03-18 2022-03-29

아쉬탕카 중급

모집인원 : 10

일주일 3번 한달 챌린지 입니다.
일주일 3번 한달 챌린지 입니다.

2022-03-17 2022-03-22

한달 넥스트레벨 완성하기

모집인원 : 15

한달안에 완성하는겁니다
한달안에 완성하는겁니다

2022-03-15 2022-03-24

헤이마마

모집인원 : 10

만희님과 댄스를 배워요
만희님과 댄스를 배워요

2022-04-16 2022-04-30

하루 100개 스쿼트

모집인원 : 5

4월한달간 개같이 코딩해봐요
4월한달간 개같이 코딩해봐요

2022-03-27 2022-04-09

모집 목록

20220418/22-27 중 #오도 클래스 #챌린지 #시디

아쉬탕카 중급

챌린지 카테고리

아쉬탕카요가

챌린지 소개글

일주일 3번 한달 챌린지 입니다.

챌린지 최대 참여 인원수

10

인증 방법

영상

인증 빈도

12

인증 빈도

2022-03-23

인증 빈도

2022-04-23

참가하기

닫기

챌린지 모집 목록

챌린지 모집 목록 조회가 가능한 화면
모집 챌린지 클릭시 해당 챌린지 세부 정보 모달창에서 조회가능
참가하기 버튼 클릭시 결제 화면으로 이동
모집 등록 버튼 클릭시 등록화면으로 이동

■ javascript

```
$('.myModal a').on('click', function(event){
    /* var challengeGatherCode = $(this).data('code'); */
    //console.log(gatherCode);
    event.preventDefault();
    console.log($(this).attr("href"));
    var href = $(this).attr("href");

    $.ajax({
        url: href
        ,success: function(data){
            console.log(data);
            console.log(data.length);
            for(var i=0; i<data.length; i++){

                var modalhader = '<h5 class="modal-title"
id="GatherDetailModalLabel" style="font-weight: bold; font-size: large;">
+data[i].challengeGather.challengeGatherName + '</h5>';

                var modalbody = '';

                modalbody += '<div
class="form-group">'+

'<label class="inputTitle">챌린지 카테고리</label>'+

'<input class="mt-2 p-2 form-control" name="challengeCategoryCode" readonly value="'+
data[i].challengeCategory.challengeCategoryName +'"style="background-color: #f9f9ff; border:
none;">'+

'</div>'+

                modalbody += '<div
class="form-group">'+

'<label class="inputTitle">챌린지 소개글</label>'+

'<input class="mt-2 p-2 form-control" name="challengeGatherIntro" readonly value="'+
data[i].challengeGather.challengeGatherIntro +'"style="background-color: #f9f9ff; border: none;">'+

'</div>'+

                modalbody += '<div
```

```

class="form-group">'+
'<label class="inputTitle">챌린지 최대 참여 인원수</label>'+
'<input class="mt-2 p-2 form-control" name="challengeGatherMaxPeople" readonly value="" +
data[i].challengeGather.challengeGatherMaxPeople + "style="background-color: #f9f9ff; border:
none;">'+
'</div>'
modalbody += 'div
class="form-group">'+
'<label class="inputTitle">인증 방법</label>'+
'<input class="mt-2 p-2 form-control" name="challengeGatherProveMethod" readonly value="" +
data[i].challengeGather.challengeGatherProveMethod + "style="background-color: #f9f9ff; border:
none;">'+
'</div>'
modalbody += 'div
class="form-group">'+
'<label class="inputTitle">인증 빈도</label>'+
'<input class="mt-2 p-2 form-control" name="challengeGatherProveFrequency" readonly value="" +
data[i].challengeGather.challengeGatherProveFrequency + "style="background-color: #f9f9ff; border:
none;">'+
'</div>'
modalbody += 'div
class="form-group">'+
'<label class="inputTitle">인증 빈도</label>'+
'<input class="mt-2 p-2 form-control" name="challengeStartDate" readonly value="" +
data[i].challengeGather.challengeStartDate + "style="background-color: #f9f9ff; border: none;">'+
'</div>'
modalbody += 'div
class="form-group">'+
'<label class="inputTitle">인증 빈도</label>'+
'<input class="mt-2 p-2 form-control" name="challengeEndDate" readonly value="" +
data[i].challengeGather.challengeEndDate + "style="background-color: #f9f9ff; border: none;">'+
'</div>'
$('.modal-header').empty();
$('.modal-body').empty();
$('.modal-header').prepend(modalheader);
$('.modal-body').append(modalbody);

```

```

        $('#GatherDetailModal').modal();
    }
}
});
});

```

■ Controller

```

//ajax 모집 챌린지 상세정보 Modal로
@GetMapping("/gatherDetail")
@ResponseBody
public List<ChallengeGatherPlan> getGatherDatail(String challengeGatherCode){
    log.info("챌린지모집 세부내용 조회 challengeGatherCode : {}",
challengeGatherCode);
    List<ChallengeGatherPlan> gatherDetail =
challengeGatherMapper.getGatherDatail(challengeGatherCode);
    log.info("챌린지모집 세부내용 조회 gatherDetail : {}", gatherDetail);
    return gatherDetail;
}

//모집 챌린지 목록 화면
@GetMapping("/challengeGatherList")
public String challengeGather(Model model) {

    List<ChallengeGather> challengeGatherList =
challengeGatherService.getChallengeGatherList();

    log.info("모집챌린지 조회 challengeGatherList : {}", challengeGatherList);

    model.addAttribute("title", "챌린지 모집 목록");
    model.addAttribute("headerList", "챌린지");
    model.addAttribute("challengeGatherList", challengeGatherList);
    return "challenge/challengeGather/challengeGatherList";
}

```


■ Service




```

//챌린지 모집 목록 조회
public List<ChallengeGather> getChallengeGatherList() {
    List<ChallengeGather> challengeGatherLlist =
challengeGatherMapper.getChallengeGatherList();
    return challengeGatherLlist;
}

```

6.3.21 챌린지 모집 등록

홈 #운동 클래스 #챌린지 #식단 #일지 관리자1님이 로그인하셨습니다. #관리자메뉴



챌린지 모집하기

챌린지 | 챌린지 모집하기

챌린지 모집하기

챌린지 모집 기본 등록사항

챌린지 모집명

챌린지 카테고리
라틴댄스

챌린지 최대 참여 인원수

챌린지 소개

챌린지 모집 시작일

챌린지 모집 종료일

챌린지 모집 계획 등록 사항

주 인증번호(주단위) 회

인증 기간 후 인증 가능 여부
Yes

챌린지 시작일

챌린지 종료일

챌린지 인증방법

챌린지 인증내용

■ JavaScript

```

$('#gatherInsert').validate({
  rules: {
    challengeGatherName: {
      required: true
    },
    challengeGatherMaxPeople: {
      required: true
    },
    challengeGatherProveMethod: {
      required: true
    },
    challengeGatherProveContent: {
      required: true
    },
    challengeGatherPlanProveNumber: {
      required: true
    },
    challengeCategory: {
      required: true
    },
    challengeGatherStartDate: {
      required: true
    },
    challengeGatherEndDate: {
      required: true
    },
    challengeStartDate: {
      required: true
    },
    challengeEndDate: {
      required: true
    },
    challengeGatherProveFrequency: {
      required: true,
      digits : true,
      range:[1,7]
    },
    challengeGatherIntro: {
      required: true
    }
  },

```



```

messages: {
  challengeGatherName: {
    required: "챌린지명을 입력해주세요"
  },
  challengeGatherMaxPeople: {
    required: "최대인원을 입력해주세요"
  },
  challengeGatherProveMethod: {
    required: "인증방법을 입력해주세요"
  },
  challengeGatherProveContent: {
    required: "인증내용을 입력해주세요"
  },
  challengeGatherPlanProveNumber: {
    required: "인증회수를 입력해주세요"
  },
  challengeCategory: {
    required: "카테고리를 선택해주세요."
  },
  challengeGatherStartDate: {
    required: "모집 시작일을 선택해주세요."
  },
  challengeGatherEndDate: {
    required: "모집 종료일일 선택해주세요."
  },
  challengeStartDate: {
    required: "챌린지 시작일을 선택해주세요."
  },
  challengeEndDate: {
    required: "챌린지 종료일일 선택해주세요."
  },
  challengeGatherProveFrequency: {
    required: "인증빈도를 입력해주세요.",
    digits : "숫자만 입력 가능합니다.",
    range:"1~7 사이의 숫자만 입력 가능합니다."
  },
  challengeGatherIntro: {
    required: "챌린지 소개를 입력해 주세요."
  }
}
});

$('.input-daterange').datepicker({
  format: 'yyyy-mm-dd',
  autoclose: true,

```

```

calendarWeeks : true,
clearBtn: true,
disableTouchKeyboard: true

});

```

■ Controller

```

//모집 챌린지 등록 화면
@GetMapping("/challengeGatherInsert")
public String challengeGatherInsert(Model model) {

    //챌린지 카테고리조회
    List<ChallengeCategory> challengeCateList =
challengeGatherService.getChallengeCate();

    //챌린지 세팅조회
    List<ChallengeSetting> challengeSettingList =
challengeGatherService.getChallengeSetting();

    model.addAttribute("title", "챌린지 모집하기");
    model.addAttribute("headerList", "챌린지");
    model.addAttribute("challengeCateList", challengeCateList);
    model.addAttribute("challengeSettingList", challengeSettingList);

    return "challenge/challengeGather/challengeGatherInsert";

}

//모집챌린지 등록처리
@PostMapping("/challengeGatherInsert")
public String challengeGatherInsert(ChallengeGather challengeGather, ChallengeGatherPlan
challengeGatherPlan, HttpSession session) {

    log.info("모집챌린지 등록폼에서 입력받은 데이터: {}", challengeGather);

    String sEmail = (String) session.getAttribute("SEMAIL");

    challengeGatherService.challengeGatherInsert(challengeGather, challengeGatherPlan,
sEmail);

    return "redirect:/challenge/challengeGather/challengeGatherList";

}

```

■ Service

```

//챌린지 모집 등록 처리

```

```

public void challengeGatherInsert(ChallengeGather challengeGather, ChallengeGatherPlan
challengeGatherPlan, String sEmail) { //,dto하나더 추가하고(controller)
    //pk컬럼에 들어갈 코드를 자동으로 만들어주는 Mapper //pk로 쓸 db의 컬럼명
    //코드가 들어갈 db의 테이블명
    String newCode1 = commonMapper.getNewCode("challengeGatherCode",
"challengegather");
    challengeGather.setChallengeGatherCode(newCode1);

    String newCode2 = commonMapper.getNewCode("challengeGatherPlanCode",
"challengegatherplan");
    challengeGatherPlan.setChallengeGatherPlanCode(newCode2);

    //날짜계산 도전!
    String stringStart = challengeGather.getChallengeStartDate(); //챌린지 시작기간 가져오
기
    String stringEnd = challengeGather.getChallengeEndDate(); //챌린지 종료 기간 가져오
기
    String stringProveFrequency = challengeGather.getChallengeGatherProveFrequency();
//인증빈도수 가져오기

    //데이터픽커로 선택한 기간을 기준으로 차이 계산하기(주)
    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd",
Locale.US); //문자열을 날짜로 바꾸는법
    LocalDate startGatherDate = LocalDate.parse(stringStart, formatter); //문자열로 받은
챌린지 시작기간을 날짜로 바꿔줌
    LocalDate endGatherDate = LocalDate.parse(stringEnd, formatter); //문자열로 받은 챌
린지 종료기간을 날짜로 바꿔줌

    //여쭙보자!
    LocalDate startDate = startGatherDate;
    LocalDate endDate = endGatherDate;

    //인증빈도수 long형변환
    long proveFrequency = Long.parseLong(stringProveFrequency);

    //데이터픽커로 선택한 기간을 기준으로 차이 계산하기(기간차이를 '주'로 계산해줌)
    long weekProve = ChronoUnit.WEEKS.between(startDate, endDate);

    String proveNumber = Long.toString(proveFrequency * weekProve); //빈도수 * 챌린지
기간(주로계산됨)

    log.info("빈도수 * 챌린지기간(주로계산됨) proveNumber : {}", proveNumber);

    challengeGather.setMemberEmail(sEmail);

    //dto(커맨드객체2).set해당프로퍼티(?) (challengeGather.getChallengeCategoryCode());
    //challengeGatherPlan에 challengeGather의 입력값을 다시 넣는 과정
    challengeGatherPlan.setMemberEmail(challengeGather.getMemberEmail());

```

```
challengeGatherPlan.setChallengeGatherCode(challengeGather.getChallengeGatherCode());

challengeGatherPlan.setChallengeCategoryCode(challengeGather.getChallengeCategoryCode());

challengeGatherPlan.setChallengeGatherChallengeStartDate(challengeGather.getChallengeStartDate());

challengeGatherPlan.setChallengeGatherChallengeLastDate(challengeGather.getChallengeEndDate());
    challengeGatherPlan.setChallengeGatherPlanProveNumber(proveNumber);

    //챌린지 모집 등록 처리
    challengeGatherMapper.challengeGatherInsert(challengeGather);

    //챌린지 모집 계획 등록처리
    challengeGatherMapper.challengeGatherPlanInsert(challengeGatherPlan);

}
```

6.3.22 참가 챌린지 목록

2022041822:27 홈 #운동 클래스 #챌린지 #시단 #일지 알바회원2님이 로그인하였습니다. #회원 정보

참가 챌린지 목록

챌린지 | 참가 챌린지 목록

참가 중인 챌린지

헤이마마
한회님과 덤스를 바꿔요
인증하기

참가 중인 챌린지

참가하고 있는 챌린지가 없습니다.
챌린지 참가하러 가기

참가 챌린지 목록

참가 중인 챌린지 목록 조회가 가능한 화면
챌린지 이름 클릭시 참가 챌린지 세부 내용 확인 화면으로 이동 가능
인증하기 버튼 클릭시 인증하기 화면으로 이동
참가 중인 챌린지가 챌린지 참가하러가기 버튼 클릭시 모집 챌린지 목록 화면으로 이동 가능

■ javascript

■ Controller

```
@GetMapping("/challengeEnterList")
```

```

public String challengeEnterList(Model model, HttpSession session) {

    log.info("참가챌린지 목록조회시 session: {}", session);

    String sEmail = (String) session.getAttribute("SEMAIL");

    List<ChallengeEnter> challengeEnterList
challengeEnterService.getChallengeEnterListByEmail(sEmail);

    log.info("참가챌린지 목록조회 challengeEnterList: {}", challengeEnterList);

    model.addAttribute("title", "참가 챌린지 목록");
    model.addAttribute("headerList", "챌린지");
    model.addAttribute("challengeEnterList", challengeEnterList);

    return "challenge/challengeEnter/challengeEnterList";

}

```

■ Service

```

//회원 이메일별 참가 챌린지 목록 조회
public List<ChallengeEnter> getChallengeEnterListByEmail(String sEmail){

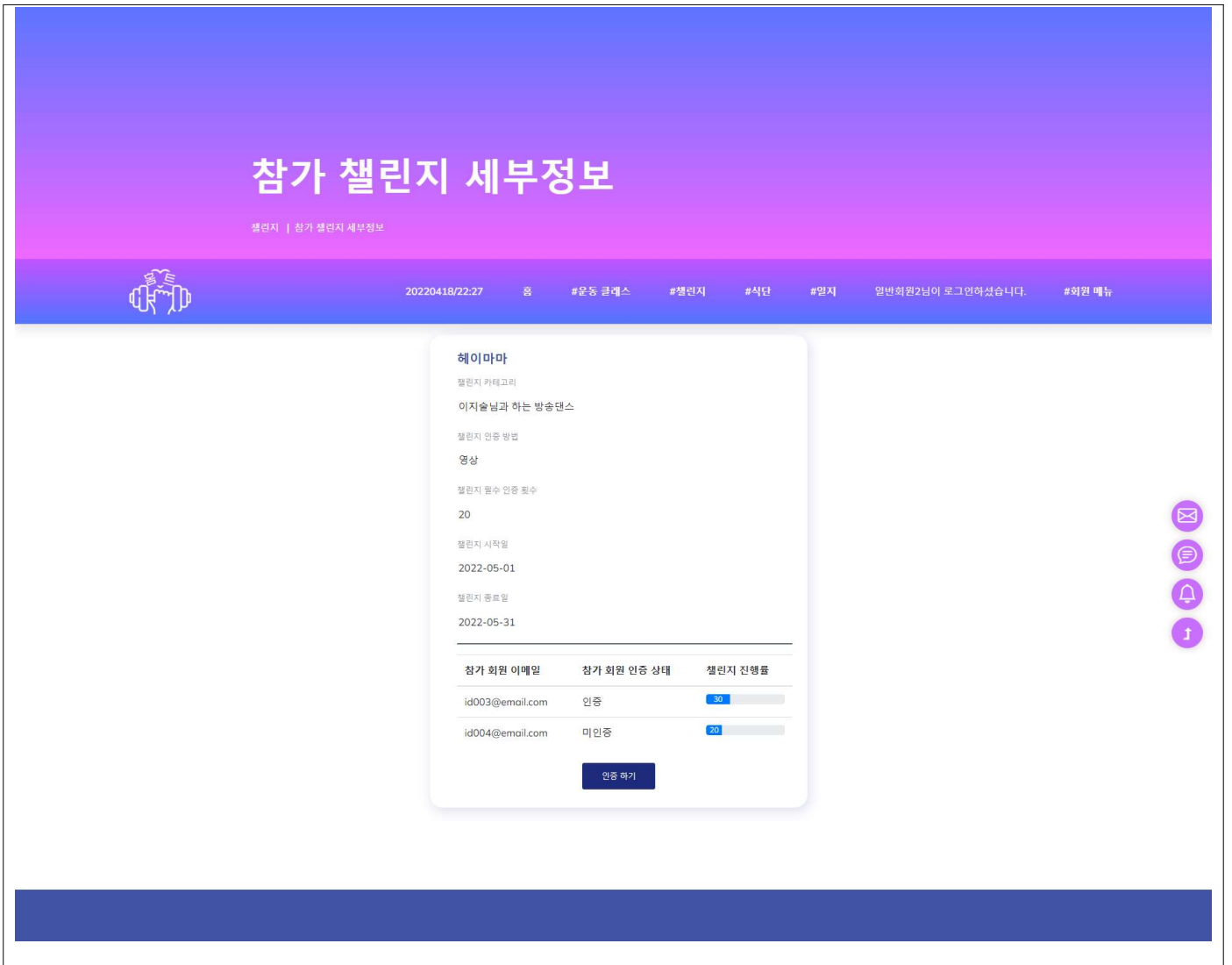
    List<ChallengeEnter> challengeEnterList
challengeEnterMapper.getChallengeEnterListByEmail(sEmail);

    return challengeEnterList;

}

```

6.3.23 참가챌린지 세부정보



참가 챌린지 세부정보

참가 챌린지의 세부 정보 조회 가능
 다른 참가자 조회 가능

■ JavaScript

■ Controller

```

@GetMapping("/challengeEnterDetail")
public String challengeEnterDetail(Model model
, @RequestParam(value="challengeGatherCode", required = false) String challengeGatherCode) {
    log.info("참가챌린지 세부조회 화면 스트링 값 challengeEnterCode: {}",
challengeGatherCode);
    }
    
```

```

        ChallengeGather                enterDetail
challengeEnterService.getChallengeEnterByCode(challengeGatherCode);

        log.info("참가챌린지 세부 조회 enterDetail: {}", enterDetail);

        model.addAttribute("title", "참가 챌린지 세부정보");
        model.addAttribute("headerList", "챌린지");
        model.addAttribute("enterDetail", enterDetail);

        return "challenge/challengeEnter/challengeEnterDetail";
    }

```

■ Service

```

//참가 챌린지의 세부 정보 조회
    public ChallengeGather getChallengeEnterByCode(String challengeGatherCode) {

        //챌린지 EnterDetail의 공통부분 목록 조회
        ChallengeGather                enterDetail
challengeEnterMapper.getChallengeEnterByCode(challengeGatherCode);

        //List<ChallengeGatherPlanDo>                ChallengeGatherPlanDoList
challengeEnterMapper.getChallengeEnterDetail(challengeGatherCode);
        //enterDetail.setChallengeGatherPlanDoList(ChallengeGatherPlanDoList);
        log.info("참가챌린지 세부 정보 조회(service) enterDetail: {}", enterDetail);

        return enterDetail;
    }

//참가 챌린지 코드별 챌린지 정보 조회
    public List<ChallengeEnter> getChallengeEnterListByCode(String challengeEnterCode) {

        List<ChallengeEnter>                challengeEnterByCode
challengeEnterMapper.getChallengeEnterListByCode(challengeEnterCode);

        return challengeEnterByCode;
    }

```


6.3.24 챌린지 인증

챌린지 | 참가 챌린지 인증하기

챌린지 인증하기

챌린지 인증 사항

헤이마마

이지솔님과 하는 방송댄스

인증 내용

파일 선택 선택된 파일 없음

인증 등록

챌린지 인증

챌린지 인증 등록이 가능한 화면

■ javascript

```
$('#gatherInsert').validate({
  rules: {
    challengeGatherPlanDoContent: {
      required: true
    }
  },
  messages: {
    challengeGatherPlanDoContent: {
      required: "인증 내용을 입력해주세요."
    }
  }
});
```

```

    }
});

$('.input-daterange').datepicker({
    format: 'yyyy-mm-dd',
    autoclose: true,
    calendarWeeks : true,
    clearBtn: true,
    disableTouchKeyboard: true
});

```

■ Controller

```

//챌린지 인증 화면
@GetMapping("/challengePlanDoInsert")
public String challengePlanDoInsert(Model model,

@RequestParam(value="challengeEnterCode", required = false) String challengeEnterCode) {

    log.info("참가 챌린지 challengeEnterCode: {}", challengeEnterCode);

    List<ChallengeEnter> challengeEnterByCode =
challengeEnterService.getChallengeEnterListByCode(challengeEnterCode);

    log.info("참가 챌린지 정보조회 challengeEnterByCode: {}", challengeEnterByCode);

    model.addAttribute("title", "참가 챌린지 인증하기");
    model.addAttribute("headerList", "챌린지");
    model.addAttribute("challengeEnterByCode", challengeEnterByCode);

    return "challenge/challengeEnter/challengePlanDoInsert";

}
//챌린지 인증 처리
@PostMapping("/challengePlanDoInsert")
public String challengePlanDoInsert(ChallengeGatherPlanDo challengeGatherPlanDo
, HttpSession
session
, @RequestParam
MultipartFile[] fileImage
,
HttpServletRequest request) {

    log.info("챌린지 인증 등록 폼에서 입력받은 데이터: {}", challengeGatherPlanDo); //받
은 내용이 여기{}에 담긴다.

```

```

String sEmail = (String) session.getAttribute("SEMAIL");

//파일 업로드
String serverName = request.getServerName();
String fileRealPath = "";
if("localhost".equals(serverName)) {
    fileRealPath = System.getProperty("user.dir") + "/src/main/resources/static/";
    //fileRealPath =
request.getSession().getServletContext().getRealPath("/WEB-INF/classes/static/");
} else {
    fileRealPath =
request.getSession().getServletContext().getRealPath("/WEB-INF/classes/static/");
}

challengeEnterService.challengePlanDoInsert(challengeGatherPlanDo, sEmail, fileImage,
fileRealPath);

return "redirect:/challenge/challengeEnter/challengePlanDoList";

}

```

■ Service

```

//챌린지 인증 등록 처리
public int challengePlanDoInsert(ChallengeGatherPlanDo challengeGatherPlanDo, String sEmail,
MultipartFile[] fileImage, String fileRealPath) {
    log.info("참가 인증 품 입력데이터(service) challengeGatherPlanDo: {}",
challengeGatherPlanDo);

    String newCode = commonMapper.getNewCode("challengeGatherPlanDoCode",
"challengegatherplando");
    log.info("새로 생성된 코드: {}", newCode);

    challengeGatherPlanDo.setChallengeGatherPlanDoCode(newCode);
    challengeGatherPlanDo.setMemberEnterEmail(sEmail);

    int result = challengeEnterMapper.challengePlanDoInsert(challengeGatherPlanDo);

    List<FileDto> fileList = fileUtil.parseFileInfo(fileImage, fileRealPath);

    fileMapper.addFiles(fileList);

    List<Map<String, String>> paramList = new ArrayList<Map<String, String>>();

    Map<String, String> paramMap = null;

    for(FileDto fileDto : fileList) {

```

```
        paramMap = new HashMap<String, String>();
        paramMap.put("referenceCode", newCode);
        paramMap.put("fileIdx", fileDto.getFileIdx());
        paramList.add(paramMap);
    }

    fileMapper.addFileControl(paramList);

    return result;
}
```

6.3.25 일지 관리

일지 관리

관리자일 경우 일지 관리하는 화면
새롭게 등록된 일지 목록과 등록된 전체 일지 목록을 조회 할 수 있습니다.
새롭게 등록된 일지 확인 후 좋아요 기능 가능합니다.
전체 일지 목록은 회원 이메일과 일지 제목으로 검색 및 기간별 검색도 가능합니다.

■ javascript

```
<!-- start: Javascript -->
  <th:block layout:fragment="customBodyScript">
    <!-- start: Javascript -->
    <script th:src="@{/asset/js/jquery.min.js}"></script>
    <script th:src="@{/asset/js/jquery.ui.min.js}"></script>
    <script th:src="@{/asset/js/bootstrap.min.js}"></script>

    <!-- plugins -->
    <script th:src="@{/asset/js/plugins/moment.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.dataTables.min.js}"></script>
    <script th:src="@{/asset/js/plugins/dataTables.bootstrap.min.js}"></script>
    <script th:src="@{/asset/js/plugins/bootstrap-datepicker.js}"></script>
    <script th:src="@{/asset/js/plugins/bootstrap-datepicker.ko.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.nicescroll.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.mask.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.validate.min.js}"></script>

    <!-- custom -->
    <script th:src="@{/asset/js/main.js}"></script>
    <script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>

    <script type="text/javascript">
      $(document).ready(function() {

        /* datepicker Start */
        $('input-daterange').datepicker({
          format : "yyyy-mm-dd", //데이터 포맷 형식
          (yyyy : 년 mm : 월 dd : 일 )
          /* startDate: '-3d', //달력에서 선택 할
          수 있는 가장 빠른 날짜. 이전으로는 선택 불가능 ( d : 일 m : 달 y : 년 w : 주) */
          autoclose : true, //사용자가 날짜를 클릭하면
          자동 캘린더가 닫히는 옵션
          calendarWeeks : false, //캘린더 옆에 몇 주
```

차인지 보여주는 옵션 기본값 false 보여주려면 true

주는 버튼 보여주는 옵션 기본값 false 보여주려면 true

'2019-06-26'],//선택 불가능한 일 설정 하는 배열 위에 있는 format 과 형식이 같아야함.

플러그인 작동 여부 기본값 false 가 작동 true가 작동 안함.

화면으로 바로바로 날짜를 변경할지 여부 기본값 :false

하는 옵션 기본값 :false

했을 때 사이에 나타나는 글자 2019-05-01,2019-06-01

스텀 마이징

옵션 기본값 : true

이틀

팅 기능 기본값 :false

하면 기본값 : false인경우 그대로 유지 true인 경우 날짜 삭제

기본값은 0인 일요일

는 js로 교체해줘야한다.

clearBtn : false, //날짜 선택한 값 초기화 해

datesDisabled : ['2019-06-24', '2019-06-26'],

disableTouchKeyboard : false, //모바일에서

immediateUpdates : false, //사용자가 보는

multidate : false, //여러 날짜 선택할 수 있게

multidateSeparator : ",", //여러 날짜를 선택

templates : {

leftArrow : '«',

rightArrow : '»'

}, //다음달 이전달로 넘어가는 화살표 모양 커

showWeekDays : true, // 위에 요일 보여주는

title : "등록일", //캘린더 상단에 보여주는 타

todayHighlight : true, //오늘 날짜에 하이라이

toggleActive : true, //이미 선택된 날짜 선택

weekStart : 0, //달력 시작 요일 선택하는 것

language : "ko" //달력의 언어 선택, 그에 맞

});

/* datepicker End */

\$('.searchBtn').click(function() {

\$('.select[name="searchKey"]').attr("required",

true);

\$('.input[name="searchValue"]').attr("required", true);

true);

\$('.input[name="searchStart"]').attr("required",

true);

\$('.input[name="searchEnd"]').attr("required",

});

\$('.searchResetBtn').click(function() {

\$('.select[name="searchKey"]').attr("required",

false);

\$('.input[name="searchValue"]').attr("required", false);

false);

\$('.input[name="searchStart"]').attr("required",

false);

\$('.input[name="searchEnd"]').attr("required",

```

});

/*오늘 버튼 클릭시 */
$('.todayBtn').click(function(){
    /* Swal.fire($(this).val()); */
    var btnValue =$(this).val();
    $('select[name="searchKey"]').attr("required"
, false);
    $('input[name="searchValue"]').attr("required"
, false);
    $('input[name="searchStart"]').attr("required"
, false);
    $('input[name="searchEnd"]').attr("required"
, false);

});

/*일주일 버튼 클릭시 */
$('.weekBtn').click(function(){
    /* Swal.fire($(this).val()); */
    var btnValue =$(this).val();
    $('select[name="searchKey"]').attr("required"
, false);
    $('input[name="searchValue"]').attr("required"
, false);
    $('input[name="searchStart"]').attr("required"
, false);
    $('input[name="searchEnd"]').attr("required"
, false);

});

/*한달 버튼 클릭시 */
$('.monthBtn').click(function(){
    /* Swal.fire($(this).val()); */
    var btnValue =$(this).val();
    $('select[name="searchKey"]').attr("required"
, false);
    $('input[name="searchValue"]').attr("required"
, false);
    $('input[name="searchStart"]').attr("required"
, false);
    $('input[name="searchEnd"]').attr("required"
, false);

});

/*3개월 버튼 클릭시 */
$('.3monthBtn').click(function(){
    /* Swal.fire($(this).val()); */

```

```

, false);
, false);
, false);
, false);

});

/*1년 버튼 클릭시 */
$('.yearBtn').click(function(){
    /* Swal.fire($(this).val()); */
    var btnValue =$(this).val();
    $('select[name="searchKey"]').attr("required"
    $('input[name="searchValue"]').attr("required"
    $('input[name="searchStart"]').attr("required"
    $('input[name="searchEnd"]').attr("required"

, false);
, false);
, false);
, false);

});

/* 데이터 테이블 생성*/
var table = $('#workoutLogList').DataTable({
    order : [ 0, 'desc' ],
    lengthChange : false,
    pageLength : 15,
    filter : false,
    dom : '<"top"i>rt<"bottom"p><"clear">',

    "language": {
        "info": "전체 _TOTAL_건",
        "paginate": {
            "first": "처음",
            "last": "마지막",
            "next": "다음",
            "previous": "이전"
        }
    }
});

/* 레코드 수 포맷 */
var info = table.page.info();

```



```

        $('#workoutLogList_info').html(
            '전체 ' + info.recordsTotal + ' 건'
        ).css('font-weight', 'bold');

        $('#workoutLogList').click(
            function() {

$('##workoutLogList_info').html('전체 ' + info.recordsTotal + ' 건');
            });

            //썸네일 슬라이더
            $('#myCarousel').carousel({
                interval : 10000
            })
            $('#myCarousel2').carousel({
                interval : 10000
            })

        });

        // 일지 좋아요 Ajax
        $(function() {
            $(".workoutLogLikeCount").on("click", function() {
                console.log('좋아요 되나유');
            });
            var workoutLogCode = $(this).attr('data-workoutLogCode');
            var text = $(this);
            console.log(workoutLogCode);
            $.ajax({
                url : '/workoutLog/workoutLogUser/workoutLogLikeCountUpdate',
                type : 'post',
                data : {
                    'workoutLogCode' : workoutLogCode
                },
                dataType : 'text',
                success : function(data) {
                    console.log(data);
                    console.log($(this));
                    text.empty();
                    text.append('<i class="fa fa-heart"></i>좋아요(' + data + ')');
                }
            });
        });
    });
</script>

</th:block>
<!-- end: Javascript -->

```

■ Controller

```
//운동 일지 관리자 메인 화면
    @GetMapping("/workoutLogConfigMain")
    public String workoutLogConfigMain(Model model

,@RequestParam(value="searchKey", required = false) String searchKey

,@RequestParam(value="searchValue", required = false) String searchValue

,@RequestParam(value="searchDate", required = false) String searchDate

,@RequestParam(value="searchStart", required = false) String searchStart

,@RequestParam(value="searchEnd", required = false) String searchEnd

,@RequestParam(name="btnKey", required = false) String btnKey) {

    log.info("챌린지 카테고리 조회 btnKey : {}", btnKey);

    Map<String, Object> paramMap = new HashMap<String,Object>();

    if(searchKey != null) {
        if("memberEmail".equals(searchKey)) {
            searchKey = "w.memberEmail"; //쿼리문과 동일하게(모호하다며 찾을
수 없다며)

            searchDate = "workoutLogUpdateFinalDate";
        }else if("workoutLogTitle".equals(searchKey)) {
            searchKey = "w.workoutLogTitle";
            searchDate = "workoutLogUpdateFinalDate";
        }
    }
    if(btnKey != null) {
        if("오늘".equals(btnKey)) {
            btnKey = "workoutLogUpdateFinalDate";
        }else if("1주일".equals(btnKey)) {
            btnKey = "-1 WEEK";
        }else if("1개월".equals(btnKey)) {
            btnKey = "-1 MONTH";
        }else if("3개월".equals(btnKey)) {
            btnKey = "-3 MONTH";
        }else if("1년".equals(btnKey)) {
            btnKey = "-1 YEAR";
        }
    }

    paramMap.put("searchKey", searchKey);
    paramMap.put("searchValue", searchValue);
    paramMap.put("searchDate", searchDate);
}
```

```

        paramMap.put("searchStart", searchStart);
        paramMap.put("searchEnd", searchEnd);
        paramMap.put("btnKey", btnKey);

        List<Map<String, Object>> workoutLogList =
workoutLogConfigService.getWorkoutLogList(paramMap);

        List<WorkoutLog> workoutLogNewList =
workoutLogConfigService.getNewWorkoutLog();
        log.info("새로운 일지 목록 workoutLogNewList : {}", workoutLogNewList);

        paramMap = null;

        log.info("일지 검색 workoutLogList : {}", workoutLogList);

        model.addAttribute("title", "일지 관리자 화면");
        model.addAttribute("leftMenuList", "일지");
        model.addAttribute("workoutLogList", workoutLogList);
        model.addAttribute("workoutLogNewList", workoutLogNewList);

        return "workoutLog/workoutLogConfig/workoutLogConfigMain";
    }

```

■ Service

```

// 운동 일지 목록 조회
    public List<Map<String, Object>> getWorkoutLogList(Map<String, Object> paramMap){

        List<Map<String, Object>> workoutLogList =
workoutLogConfigMapper.getWorkoutLogList(paramMap);

        return workoutLogList;

    }

// 새롭게 등록된 일지 조회
    public List<WorkoutLog> getNewWorkoutLog(){

        List<WorkoutLog> workoutLogNewList =
workoutLogConfigMapper.getNewWorkoutLog();

        return workoutLogNewList;

    }

```

6.3.26 운동 계획 카테고리 관리

운동 계획 카테고리 목록

일지 > 운동 계획 카테고리 목록

운동 계획 카테고리 목록

일지 검색 회원이메일 검색키워드를 입력해주세요.

등록일자 시작일 부터 종료일 까지 오늘 1주일 1개월 3개월 1년

검색 초기화

전체 6 건

운동 계획 카테고리명	수정	삭제
챌린지	수정	삭제
민식붓	수정	삭제
개인 운동 목표	수정	삭제
개인 운동 목표	수정	삭제
개인 과제	수정	삭제

Previous 1 Next

카테고리 등록

운동 계획 카테고리명 수정

운동 계획 카테고리명
챌린지

취소 저장

운동 계획 카테고리 관리

관리자일 경우 운동 계획 카테고리 등록, 수정, 삭제, 조회 가능합니다.

등록 버튼을 클릭하면 운동 카테고리명을 등록할 수 있습니다.

수정 버튼을 클릭하면 해당하는 카테고리명을 모달에 띄워주고 수정 후 저장 버튼을 클릭 시 Ajax로 수정처리가 가능합니다.

삭제 버튼을 클릭하면 해당하는 카테고리명이 삭제됩니다.

운동 계획 카테고리 등록 화면

일지 > 운동 계획 카테고리 등록 화면

운동 계획 카테고리 등록

운동 계획 카테고리명

입력

■ javascript

```
<!-- start: Javascript -->
  <th:block layout:fragment="customBodyScript">
    <!-- start: Javascript -->
    <script th:src="@{/asset/js/jquery.min.js}"></script>
    <script th:src="@{/asset/js/jquery.ui.min.js}"></script>
    <script th:src="@{/asset/js/bootstrap.min.js}"></script>

    <!-- plugins -->
    <script th:src="@{/asset/js/plugins/moment.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.dataTables.min.js}"></script>
    <script th:src="@{/asset/js/plugins/dataTables.bootstrap.min.js}"></script>
    <script th:src="@{/asset/js/plugins/bootstrap-datepicker.js}"></script>
    <script th:src="@{/asset/js/plugins/bootstrap-datepicker.ko.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.nicescroll.js}"></script>
    <script th:src="@{/asset/js/plugins/nouislider.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.mask.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.validate.min.js}"></script>

    <!-- custom -->
    <script th:src="@{/asset/js/main.js}"></script>
    <script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>
    <script type="text/javascript">

      $(document).ready(function(){

        /* datepicker Start */
        $('input-daterange').datepicker({
          format : "yyyy-mm-dd", //데이터 포맷 형식(yyyy : 년 mm :
월 dd : 일 )
```

```

/* startDate: '-3d', //달력에서 선택 할 수 있는 가장 빠
큰 날짜. 이전으로는 선택 불가능 ( d : 일
m : 달 y : 년 w : 주) */
autoclose : true, //사용자가 날짜를 클릭하면 자동 캘린더가
닫히는 옵션
calendarWeeks : false, //캘린더 옆에 몇 주차인지 보여주는
옵션 기본값 false 보여주려면 true
clearBtn : false, //날짜 선택한 값 초기화 해주는 버튼 보여주
는 옵션 기본값 false 보여주려면 true
datesDisabled : [ '2019-06-24', '2019-06-26' ],//선택 불가
능한 일 설정 하는 배열 위에 있는 format 과 형식이 같아야함.
disableTouchKeyboard : false, //모바일에서 플러그인 작동 여
부 기본값 false 가 작동 true가 작동 안함.
immediateUpdates : false, //사용자가 보는 화면으로 바로바로
날짜를 변경할지 여부 기본값 :false
multidate : false, //여러 날짜 선택할 수 있게 하는 옵션 기본
값 :false
multidateSeparator : ",", //여러 날짜를 선택했을 때 사이에 나
타나는 글짜 2019-05-01,2019-06-01
templates : {
    leftArrow : '&laquo;',
    rightArrow : '&raquo;'
}, //다음달 이전달로 넘어가는 화살표 모양 커스텀 마이징
showWeekDays : true,// 위에 요일 보여주는 옵션 기본값 :
true
title : "등록일", //캘린더 상단에 보여주는 타이틀
todayHighlight : true, //오늘 날짜에 하이라이팅 기능 기본값
:false
toggleActive : true, //이미 선택된 날짜 선택하면 기본값 :
false인경우 그대로 유지 true인 경우 날짜 삭제
weekStart : 0,//달력 시작 요일 선택하는 것 기본값은 0인 일요
일
language : "ko" //달력의 언어 선택, 그에 맞는 js로 교체해줘
야한다.
});
/* datepicker End */
$('.searchBtn').click(function() {
    $('select[name="searchKey"]').attr("required", true);
    $('input[name="searchValue"]').attr("required", true);
});
$('.searchResetBtn').click(function() {
    $('select[name="searchKey"]').attr("required", false);
    $('input[name="searchValue"]').attr("required", false);
});
/* 데이터 테이블 생성*/
var table = $('#workoutCategory').DataTable({
    order : [ 0, 'desc' ],
    lengthChange : false,
    pageLength : 15,
    filter: false,
    dom: '<"top">rt<"bottom"><"clear">',

```

```

});

/* 레코드 수 포맷 */
var info = table.page.info();
$('#workoutCategory_info').html('전체 ' + info.recordsTotal + ' 건
).css('font-weight','bold');

$('#workoutCategory').click(function(){
    $('#workoutCategory_info').html('전체 ' + info.recordsTotal +
' 건');
});

/* 모달 */
// Set iframe attributes when the show instance method is called
$('#categoryUpdate').on("show.bs.modal", function(event) {
    let button = $(event.relatedTarget); // Button that triggered the
modal
});
// Remove iframe attributes when the modal has finished being hidden
from the user
$('#categoryUpdate').on("hidden.bs.modal", function() {
    $('#categoryUpdate').removeAttr('.close');
});

/* 카테고리폼 */
$('#workoutCategoryForm').validate({
    errorElement: "em",
    errorPlacement: function(error, element) {
        $(element.parent("div").addClass("form-animate-error"));
        error.appendTo(element.parent("div"));
    },
    success: function(label) {
        $(label.parent("div").removeClass("form-animate-error"));
    },
    rules: {
        workoutGoalPlanCategoryContent: "required",
    },
    messages: {
        workoutGoalPlanCategoryContent: "카테고리 이름을 입력해주세요.",
    }
});

//모달에 데이터 가져오기
$(document).on('click','.workoutCategoryUpdateBtn',function(){
    console.log("나와라 오바");
    var workoutLogCategoryCode =

```

```

$(this).attr('data-cateCode');
                                                                    var      workoutGoalPlanCategoryContent      =
$(this).attr('data-cateName');
                                                                    console.log(workoutLogCategoryCode,
workoutGoalPlanCategoryContent);

$('#workoutGoalPlanCategoryContent').val(workoutGoalPlanCategoryContent);

$('#workoutGoalPlanCategoryCode').val(workoutLogCategoryCode);

                                                                    });

                                                                    //모달에서 카테고리명 수정 처리
                                                                    $(document).on('click', '.workoutCategoryUpdate', function(){
                                                                    console.log("나와라 수정");
                                                                    var      workoutGoalPlanCategoryCode      =
$('#workoutGoalPlanCategoryCode').val();
                                                                    var      workoutGoalPlanCategoryContent      =
$('#workoutGoalPlanCategoryContent').val();
                                                                    console.log(workoutGoalPlanCategoryCode,
workoutGoalPlanCategoryContent);

                                                                    $.ajax({
                                                                    url:
'/admin/workoutLogCategoryUpdateModal' //수정 컨트롤러
                                                                    ,type : 'post'
                                                                    ,data : {

                                                                    'workoutGoalPlanCategoryCode' : workoutGoalPlanCategoryCode,

                                                                    'workoutGoalPlanCategoryContent' : workoutGoalPlanCategoryContent
                                                                    }
                                                                    ,dataType : 'text'
                                                                    ,success: function(data){
                                                                    console.log(data);
                                                                    if(Number(data) > 0){

                                                                    alert('수정 성공');
                                                                    location.reload();

                                                                    }else{
                                                                    alert('수정 실패');

                                                                    }

                                                                    }
                                                                    , error : function(){
                                                                    alert('수정 실패');

                                                                    }

                                                                    }

                                                                    });

```



```

        });

    });

</script>

</th:block>
<!-- end: Javascript -->

```

■ Controller

```

//운동 계획 카테고리 목록
@GetMapping("/workoutCategoryList")
public String workoutCategoryList( Model model

,@RequestParam(value="searchValue", required = false) String searchValue) {

    Map<String, Object> paramMap = new HashMap<String,Object>();

    if(searchValue != null) {
        if("workoutGoalPlanCategoryContent".equals(searchValue)) {
            searchValue = "workoutGoalPlanCategoryContent"; //쿼리문과
동일하게(모호하다며 찾을 수 없다며)
        }
    }

    paramMap.put("searchValue", searchValue);

    List<Map<String, Object>> workoutLogCategoryList =
workoutLogConfigService.getWorkoutLogCategoryList(paramMap);

    paramMap = null;

    model.addAttribute("title", "운동 계획 카테고리 목록");
    model.addAttribute("leftMenuList", "일지");
    model.addAttribute("workoutLogCategoryList", workoutLogCategoryList);

    return "workoutLog/workoutLogConfig/workoutCategoryList";
}

//운동 계획 카테고리 등록 처리
@PostMapping("/workoutCategoryInsert")
public String workoutCategoryInsert(WorkoutLogCategory workoutLogCategory) {

    log.info("운동 카테고리 등록 폼에서 입력받은 데이터: {}", workoutLogCategory);

    workoutLogConfigService.workoutGoalCategoryInsert(workoutLogCategory);

    return "redirect:/admin/workoutCategoryList";
}

```

```

}

//운동 계획 카테고리 등록 화면
@GetMapping("/workoutCategoryInsert")
public String workoutCategoryInsert(Model model) {

    model.addAttribute("title", "운동 계획 카테고리 등록 화면");
    model.addAttribute("leftMenuList", "일지");

    return "workoutLog/workoutLogConfig/workoutCategoryInsert";
}

//A-jax : 운동 계획 카테고리명 수정처리
@PostMapping("/workoutLogCategoryUpdateModal")
@ResponseBody
public int workoutLogCategoryUpdateModal(@RequestParam Map<String, Object> paramMap){

    log.info("workoutLogCategoryUpdateModal : {}", paramMap.toString());

    return workoutLogConfigMapper.workoutLogCategoryUpdateModal(paramMap);
}

//운동 계획 카테고리 수정
@GetMapping("/workoutCategoryUpdate")
public String workoutCategoryUpdate(Model model) {

    model.addAttribute("title", "운동 계획 카테고리 수정");
    model.addAttribute("leftMenuList", "일지");

    return "workoutLog/workoutLogConfig/workoutCategoryUpdate";
}

//운동 계획 카테고리 삭제
@GetMapping("/workoutCategoryDelete")
public String workoutCategoryDelete(@RequestParam(name="workoutGoalPlanCategoryCode",
required = false) String workoutGoalPlanCategoryCode, WorkoutLogCategory workoutLogCategory) {

    log.info("운동 계획 카테고리 등록 폼에서 삭제 할 데이터: {}",
workoutGoalPlanCategoryCode);

    workoutLogConfigService.workoutGoalCategoryDelete(workoutGoalPlanCategoryCode);

    return "redirect:/admin/workoutCategoryList";
}

```

```
}
```

■ Service

```
// 운동 계획 카테고리 목록 조회
public List<Map<String, Object>> getWorkoutLogCategoryList(Map<String, Object> paramMap){

    List<Map<String, Object>> workoutLogCategoryList =
workoutLogConfigMapper.getWorkoutLogCategoryList(paramMap);

    return workoutLogCategoryList;

}

// 운동 계획 카테고리 등록 처리
public int workoutGoalCategoryInsert(WorkoutLogCategory workoutLogCategory) {

    String code = commonMapper.getNewCode("workoutGoalPlanCategoryCode",
"workoutgoalplancategory");
    workoutLogCategory.setWorkoutGoalPlanCategoryCode(code);

    int result = workoutLogConfigMapper.workoutGoalCategoryInsert(workoutLogCategory);

    return result;

}

// 운동 계획 카테고리 삭제 처리
public void workoutGoalCategoryDelete(String workoutGoalPlanCategoryCode) {

workoutLogConfigMapper.GoalPlanInCategoryCodeDelete(workoutGoalPlanCategoryCode);
    workoutLogConfigMapper.workoutLogCategoryDelete(workoutGoalPlanCategoryCode);

}

}
```

6.3.27 하루 일지

20220415/14:27 홈 #운동 클래스 #캘린저 #식단 #일지 일반회원6년여 로그인하셨습니다.

#회원 메뉴

하루 일지

] 하루 일지

15 Jan

오늘은 사이드런지

사이드 런지 사이드 런지 ...

♥ 좋아하는날 | 📌 피드백

Search Keyword

오늘의 일지 등록

전체 일지

오늘의 할 일

- 물 2리터 이상 마시기
- 스쿼트 400개 하기
- 책 읽기
- 필라소파

새로운 할 일 작성 후 Enter 키를

나의 목표

운동 목표
하루에 스쿼트 100개

목표 달성률

운동 목표
10kg 감량

목표 달성률

하루 일지

로그인한 일반 회원의 등록된 일지 목록 조회 가능합니다.
일지의 좋아요 수 조회 가능하며 일지 제목을 클릭하면 해당하는 일지의 상세 화면으로 전환됩니다.
오늘의 일지 등록을 클릭하면 일지 등록이 가능합니다.
오늘의 할 일을 추가하고 삭제 가능합니다.

■ JavaScript

```

<!-- 사용자 정의 Script -->
    <th:block layout:fragment="customScript">
        <script>

            // to do List
            $(document).ready(function($) {
                "use strict"
                //todo list
                $(".tdl-new").on('keypress', function(e) {
                    var code = (e.keyCode ? e.keyCode : e.which);
                    if (code == 13) {
                        var v = $(this).val();
                        var s = v.replace(/ +?/g, '');
                        if (s == "") {
                            return false;
                        } else {
                            $(".tdl-content ul").append("<li><label><input
type='checkbox'><i></i><span>" + v + "</span><a href='#' class='ti-trash'></a></label></li>");
                            $(this).val("");
                            return false; //리턴값 추가하여 엔터 먹힘
                        }
                    }
                });
                $(".tdl-content a").on("click", function() {
                    var _li = $(this).parent().parent("li");
                    _li.addClass("remove").stop().delay(100).slideUp("fast", function() {
                        _li.remove();
                    });
                    return false;
                });
                // for dynamically created a tags
                $(".tdl-content").on('click', "a", function() {
                    var _li = $(this).parent().parent("li");
                    _li.addClass("remove").stop().delay(100).slideUp("fast", function() {
                        _li.remove();
                    });
                    return false;
                });
            });

            // 일지 좋아요 Ajax
            $(function() {
                $(".workoutLogLikeCount").on("click", function() {

```

```

        var workoutLogCode = $(this).attr('data-workoutLogCode');
        var text = $(this);
        console.log(workoutLogCode);
        $.ajax({
            url : '/workoutLog/workoutLogUser/workoutLogLikeCountUpdate',
            type : 'post',
            data : {
                'workoutLogCode' : workoutLogCode
            },
            dataType : 'text',
            success : function(data) {
                console.log(data);
                console.log($(this));
                text.empty();
                text.append('<i class="fa fa-heart"></i>좋아요(' + data + ')');
            }
        });
    });
</script>
</th:block>

```

■ Controller

```

// 일지 메인 화면
@GetMapping("/workoutLogMain")
public String workoutLogMain(Model model) {

    //운동 목표 목록 조회
    List<WorkoutGoal> workoutGoalList = workoutLogUserService.getworkoutGoalList();

    //log.info("운동 목표 목록 조회 workoutGoalList : {}", workoutGoalList);

    //일지 목록 조회
    List<WorkoutLog> workoutLogList = workoutLogUserService.getworkoutLogList();

    log.info("일지 목록 조회 workoutLogList : {}", workoutLogList);

    model.addAttribute("title", "하루 일지");
    model.addAttribute("workoutGoalList", workoutGoalList);
    model.addAttribute("workoutLogList", workoutLogList);

    return "workoutLog/workoutLogUser/workoutLogMain";

}

// 일지 좋아요 Ajax

```

```

    @PostMapping("/workoutLogLikeCountUpdate")
    @ResponseBody
    public String workoutLogLikeCountUpdate(@RequestParam(value = "workoutLogCode") String
workoutLogCode

workoutLogLike

workoutLogLike){

        workoutLogUserService.workoutLogLikeCountUpdate(workoutLogCode);
        workoutLogLike.setWorkoutLogLikeEmail("id004@email.com");
        workoutLogLike.setWorkoutLogCode(workoutLogCode);
        workoutLogUserService.workoutLogLikeInsert(workoutLogLike);

        return workoutLogUserMapper.getworkoutLogLikeCount(workoutLogCode);
    }

```

■ Service

```

// 일지 좋아요 등록
public int workoutLogLikeInsert(WorkoutLogLike workoutLogLike) {

    String num = commonMapper.getNewCode("workoutLogLikeNum", "workoutloglike");
    workoutLogLike.setWorkoutLogLikeNum(num);

    int result = workoutLogUserMapper.workoutLogLikeInsert(workoutLogLike);

    return result;
}

//일지 좋아요 카운트
public void workoutLogLikeCountUpdate(String workoutLogCode) {

    workoutLogUserMapper.workoutLogLikeCountUpdate(workoutLogCode);
}

//일지 목록 조회
public List<WorkoutLog> getworkoutLogList(){

    List<WorkoutLog> workoutLogList = workoutLogUserMapper.getworkoutLogList();

    log.info("workoutLogList : {}", workoutLogList);

    return workoutLogList;
}

```

6.3.28 일지 수정

일지 수정 화면

| 로그인/로그아웃

일지 수정

일지 제목

토르가 운동해유

전체 보기

근육

헬스

중급 런닝 운동

일지 내용

토르가 살을 많이 찘어요
나도 빼야하는데
어려워요

일지 수정

일지 수정

일지 상세 화면에서 수정버튼 클릭 시 등록된 일지 정보를 불러와 보여줍니다.
운동 클래스 카테고리 및 공개범위 정보를 Ajax처리하여 대분류를 클릭 시 대분류에 해당하는 중분류를 보여주고 중분류 클릭 시 중분류에 해당하는 소분류와 등록되어 있던 공개범위 내용을 보여줍니다.
수정할 부분 변경하고 일지 수정 버튼을 클릭하면 수정 처리가 됩니다.

■ JavaScript

```

<!-- 사용자 정의 Script -->
    <th:block layout:fragment="customBodyScript">
        <!-- <script src="https://code.jquery.com/jquery-3.5.1.min.js"
crossorigin="anonymous"></script>

        <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvblyZFJoft+2mJbHaEWidlvi9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>

        <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-wfSDF2E50Y2D1uUdJ0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script> -->
        <script>

            $(function(){
                console.log("이거 되나? 이거 안되면");

                //운동 계획 카테고리 대-중 분류
                $(document).on('change','#eClassCategoryLargeCode', function(e) {
                    console.log("이거 되나?medium");

                    var eClassCategoryLargeCode =
$('select[name="eClassCategoryLargeCode"]').val();
                    console.log(eClassCategoryLargeCode);
                    $.ajax({
                        url : '/workoutLog/workoutLogUser/geteClassCategoryMedium',
                        type : 'post',
                        data : {
                            'eClassCategoryLargeCode' : eClassCategoryLargeCode
                        },
                        dataType : 'json',
                        success : function(data) {
                            console.log(JSON.stringify(data));
                            console.log(data.length);
                            var txt1='<option value="none">운동 카테고리 중분류 선택하세요
</option>;

                            for(var i =0;i<data.length;i++)
                            {
                                txt1 += '<option
value="'+data[i].eClassCategoryMediumCode+' ' + data[i].eClassCategoryMediumName
+'>';
                            }
                            console.log(txt1);
                            $('#eClassCategoryMediumCode').empty();
                            $('#eClassCategoryMediumCode').append(txt1);
                            $('#eClassCategoryMediumCode').niceSelect('update');

                        }
                    }
                )
            });
        }
    </th:block>

```

```

    });
    });

    //운동 계획 카테고리 중-소 분류
    $(document).on('change', '#eClassCategoryMediumCode', function(e) {
        console.log("이거 되나?small");

        var eClassCategoryMediumCode =
        $('select[name="eClassCategoryMediumCode"]').val();
        console.log(eClassCategoryMediumCode);
        $.ajax({
            url : '/workoutLog/workoutLogUser/geteClassCategorySmall',
            type : 'post',
            data : {
                'eClassCategoryMediumCode' : eClassCategoryMediumCode
            },
            dataType : 'json',
            success : function(data) {
                console.log(JSON.stringify(data));
                console.log(data.length);
                var txt1='<option value="none">운동 카테고리 소분류 선택하세요
</option>';

                for(var i =0;i<data.length;i++)
                {
                    txt1 += '<option
value="'+data[i].eClassCategorySmallCode+' ' + data-cate="'+ data[i].eClassCategorySmallName
+'>' + data[i].eClassCategorySmallName + "</option>";
                }
                console.log(txt1);
                $('#eClassCategorySmallCode').empty();
                $('#eClassCategorySmallCode').append(txt1);
                $('#eClassCategorySmallCode').niceSelect('update');

            }
        });
    });

    //등록 처리시 form을 submit 하기 전에 특정값을 추가
    $(document).on('click', '#workoutLogInsertbtn', function(){
        var datacate = $('#eClassCategorySmallCode
option:selected').attr('data-cate');
        //alert(datacate);
        var eClassCategorySmall = $("<input type='hidden' value="" +
datacate + " name='eClassCategorySmallName'/>");
        $("#workoutLogInsert").append(eClassCategorySmall);
        $("#workoutLogInsert").submit();
    });
});
</script>

```

</th:block>

■ Controller

```
// Ajax : 운동 계획 카테고리 대-중 분류
@PostMapping("/geteClassCategoryMedium")
@ResponseBody
public List<EClassCategoryMedium> geteClassCategoryMedium(@RequestParam(value = "eClassCategoryLargeCode", required = false) String eClassCategoryLargeCode){

    log.info("운동클래스 카테고리 medium 목록 조회 : {}", eClassCategoryLargeCode);

    return workoutLogUserMapper.geteClassCategoryMedium(eClassCategoryLargeCode);

}

// Ajax : 운동 계획 카테고리 중-소 분류
@PostMapping("/geteClassCategorySmall")
@ResponseBody
public List<EClassCategorySmall> geteClassCategorySmall(@RequestParam(value = "eClassCategoryMediumCode", required = false) String eClassCategoryMediumCode){

    log.info("운동클래스 카테고리 small 목록 조회 : {}", eClassCategoryMediumCode);

    return workoutLogUserMapper.geteClassCategorySmall(eClassCategoryMediumCode);

}

// Ajax : 일지 수정 시 공개범위 선택
@PostMapping("/getworkoutLogPrivacyboundsList")
@ResponseBody
public List<WorkoutLogPrivacybounds> getworkoutLogPrivacyboundsList(@RequestParam(value = "workoutLogPrivacyBoundsCode", required = false) String workoutLogPrivacyBoundsCode){

    log.info("수정 화면 - 공개 범위 선택 : {}", workoutLogPrivacyBoundsCode);

    return workoutLogUserMapper.getworkoutLogPrivacyboundsList();

}

// 일지 수정 처리
@PostMapping("/workoutLogUpdate")
public String workoutLogUpdate(WorkoutLog workoutLog, RedirectAttributes reAttr) {

    workoutLogUserService.workoutLogUpdate(workoutLog);
    log.info("일지 수정 처리: {}", workoutLog);
}
```

```

        reAttr.addAttribute("workoutLogCode", workoutLog.getWorkoutLogCode());

        return "redirect:/workoutLog/workoutLogUser/workoutLogList";
    }

    // 일지 수정 화면
    @GetMapping("/workoutLogUpdate")
    public String workoutLogUpdate( Model model
        , @RequestParam(name="workoutLogCode", required = false)
String workoutLogCode) {

        log.info("일지 수정화면 폼 쿼리 스트링 workoutLogCode : {}", workoutLogCode);

        //일지 공개범위 목록 조회
        List<WorkoutLogPrivacybounds> workoutLogPrivacyboundsList =
workoutLogUserService.getworkoutLogPrivacyboundsList();
        log.info("일지 공개범위 목록 조회 workoutlogPrivacyboundsList : {}",
workoutLogPrivacyboundsList);

        //운동 클래스 카테고리 large 목록 조회
        List<EClassCategoryLarge> eClassCategoryLargeList =
workoutLogUserService.geteClassCategoryLargeList();
        log.info("운동 클래스 카테고리 large 목록 조회 eClassCategoryLargeList : {}",
eClassCategoryLargeList);

        //수정화면 - 일지 코드로 일지 조회
        WorkoutLog workoutLog =
workoutLogUserService.getworkoutLogByLogCode(workoutLogCode);
        log.info("수정화면 - 일지 코드로 일지 조회 workoutLog : {}", workoutLog);

        model.addAttribute("title", "일지 수정 화면");
        model.addAttribute("workoutLogPrivacyboundsList", workoutLogPrivacyboundsList);
        model.addAttribute("eClassCategoryLargeList", eClassCategoryLargeList);
        model.addAttribute("workoutLog", workoutLog);

        return "workoutLog/workoutLogUser/workoutLogUpdate";
    }
}

```

■ Service

```

//일지 코드로 일지 목록 조회 public WorkoutLog getworkoutLogByCode(String workoutLogCode) {

        return workoutLogUserMapper.getworkoutLogByCode(workoutLogCode);

    }

    // 일지 수정 화면 - 일지 코드로 일지 조회

```

```
public WorkoutLog getworkoutLogByLogCode(String workoutLogCode){  
  
    return workoutLogUserMapper.getworkoutLogByLogCode(workoutLogCode);  
  
}  
  
// 일지 수정 처리  
public int workoutLogUpdate(WorkoutLog workoutLog) {  
  
    int result = workoutLogUserMapper.workoutLogUpdate(workoutLog);  
  
    return result;  
  
}
```

6.3.29 일지 등록

일지 등록

| 일지 등록

오늘 일지 등록

일지 제목

■ 공개 범위

■ 운동 카테고리 대

■ 운동 카테고리 중

■ 운동 카테고리 소

파일 선택 선택된 파일 없음

일지 내용

내용을 입력하세요

일지 등록

일지 등록

로그인한 일반 회원이 일지 제목, 공개범위, 운동 클래스 카테고리 대, 중, 소 및 이미지 파일 첨부, 일지 내용을 일반 회원만 등록할 수 있습니다.

■ JavaScript

```
<!-- 사용자 정의 Script -->
    <th:block layout:fragment="customBodyScript">
        <!--<script src="https://code.jquery.com/jquery-3.5.1.min.js"
crossorigin="anonymous"></script>
        <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvblyZFJoft+2mJbHaEWIidlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
        <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script>-->
        <script>
```

```

$(function(){
    console.log("이거 되나? 이거 안되면");

    //운동 계획 카테고리 대-중 분류
    $(document).on('change','#eClassCategoryLargeCode', function(e) {
        console.log("이거 되나?medium");
        var eClassCategoryLargeCode =
        $('#select[name="eClassCategoryLargeCode"]').val();
        console.log(eClassCategoryLargeCode);
        $.ajax({
            url : '/workoutLog/workoutLogUser/geteClassCategoryMedium',
            type : 'post',
            data : {
                'eClassCategoryLargeCode' : eClassCategoryLargeCode
            },
            dataType : 'json',
            success : function(data) {
                console.log(JSON.stringify(data));
                console.log(data.length);
                var txt1='<option value="none">운동 카테고리 중분류 선택하세요
</option>';
                for(var i =0;i<data.length;i++)
                {
                    txt1 +=<option
value="" +data[i].eClassCategoryMediumCode+" ' ' +>'+ data[i].eClassCategoryMediumName
+ "</option>";
                }
                console.log(txt1);
                $('#eClassCategoryMediumCode').empty();
                $('#eClassCategoryMediumCode').append(txt1);
                $('#eClassCategoryMediumCode').niceSelect('update');
            }
        });
    });

    //운동 계획 카테고리 중-소 분류
    $(document).on('change','#eClassCategoryMediumCode', function(e) {
        console.log("이거 되나?small");
        var eClassCategoryMediumCode =
        $('#select[name="eClassCategoryMediumCode"]').val();
        console.log(eClassCategoryMediumCode);
        $.ajax({
            url : '/workoutLog/workoutLogUser/geteClassCategorySmall',
            type : 'post',
            data : {
                'eClassCategoryMediumCode' : eClassCategoryMediumCode
            },
            dataType : 'json',
            success : function(data) {
                console.log(JSON.stringify(data));
                console.log(data.length);
                var txt1='<option value="none">운동 카테고리 소분류 선택하세요
</option>';
                for(var i =0;i<data.length;i++)
                {
                    txt1 +=<option
value="" +data[i].eClassCategorySmallCode+" ' ' + data-cate="" + data[i].eClassCategorySmallName
+ ">'+ data[i].eClassCategorySmallName + "</option>";
                }
                console.log(txt1);
            }
        });
    });
});

```

```

        $('#eClassCategorySmallCode').empty();
        $('#eClassCategorySmallCode').append(txt1);
        $('#eClassCategorySmallCode').niceSelect('update');
    }
});
});

//등록 처리시 form을 submit 하기 전에 특정값을 추가
$(document).on('click','#workoutLogInsertbtn',function(){
    var datacate = $('#eClassCategorySmallCode
option:selected').attr('data-cate');
    //alert(datacate);
    var eClassCategorySmall = $("<input type='hidden' value='"+
datacate +"' name='eClassCategorySmallName'/>");
    $("#workoutLogInsert").append(eClassCategorySmall);
    $("#workoutLogInsert").submit();
});

//회원만 일지 등록하는 처리

var sessionLevel = '[[${session.SLEVEL}]]';

$('#workoutLogInsertbtn').click(function(){

    if(sessionLevel == 'levelCode001'){
        alert('회원만 일지 등록할 수 있습니다.')
        return false;
    }
});

});
</script>

</th:block>

```

■ Controller

```

// 일지 등록 화면
@GetMapping("/workoutLogInsert")
public String workoutLogInsert(Model model) {

    //일지 공개범위 목록 조회
    List<WorkoutLogPrivacybounds> workoutLogPrivacyboundsList =
workoutLogUserService.getworkoutLogPrivacyboundsList();
    log.info("일지 공개범위 목록 조회 workoutlogPrivacyboundsList : {}",
workoutLogPrivacyboundsList);

    //운동 클래스 카테고리 large 목록 조회
    List<EClassCategoryLarge> eClassCategoryLargeList =
workoutLogUserService.geteClassCategoryLargeList();
    log.info("운동 클래스 카테고리 large 목록 조회 eClassCategoryLargeList : {}",
eClassCategoryLargeList);

    model.addAttribute("title", "일지 등록");
    model.addAttribute("workoutLogPrivacyboundsList", workoutLogPrivacyboundsList);
    model.addAttribute("eClassCategoryLargeList", eClassCategoryLargeList);
}

```



```

        return "workoutLog/workoutLogUser/workoutLogInsert";
    }

    // Ajax : 운동 계획 카테고리 대-중 분류
    @PostMapping("/geteClassCategoryMedium")
    @ResponseBody
    public List<EClassCategoryMedium> geteClassCategoryMedium(@RequestParam(value
    ="eClassCategoryLargeCode", required =false) String eClassCategoryLargeCode){

        log.info("운동클래스 카테고리 medium 목록 조회 : {}", eClassCategoryLargeCode);

        return workoutLogUserMapper.geteClassCategoryMedium(eClassCategoryLargeCode);

    }

    // Ajax : 운동 계획 카테고리 중-소 분류
    @PostMapping("/geteClassCategorySmall")
    @ResponseBody
    public List<EClassCategorySmall> geteClassCategorySmall(@RequestParam(value
    ="eClassCategoryMediumCode", required =false) String eClassCategoryMediumCode){

        log.info("운동클래스 카테고리 small 목록 조회 : {}", eClassCategoryMediumCode);

        return workoutLogUserMapper.geteClassCategorySmall(eClassCategoryMediumCode);

    }

    // Ajax : 일지 수정 시 공개범위 선택
    @PostMapping("/getworkoutLogPrivacyboundsList")
    @ResponseBody
    public List<WorkoutLogPrivacybounds> getworkoutLogPrivacyboundsList(@RequestParam(value
    ="workoutLogPrivacyBoundsCode", required =false) String workoutLogPrivacyBoundsCode){

        log.info("수정 화면 - 공개 범위 선택 : {}", workoutLogPrivacyBoundsCode);

        return workoutLogUserMapper.getworkoutLogPrivacyboundsList();

    }

    //일지 등록 처리
    @PostMapping("/workoutLogInsert")
    public String workoutLogInsert(WorkoutLog workoutLog
    , HttpSession session
    , @RequestParam MultipartFile[]
    fileImage
    , HttpServletRequest request) {

        String sessionEmail = (String) session.getAttribute("SEMAIL"); //형변환을 해줘라

        //파일 업로드
        String serverName = request.getServerName();
        String fileRealPath = "";
        if("localhost".equals(serverName)) {
            fileRealPath =System.getProperty("user.dir") +"/src/main/resources/static/";
            //fileRealPath
        }else {
            fileRealPath
            request.getSession().getServletContext().getRealPath("/WEB-INF/classes/static/");
        }
        fileRealPath
        request.getSession().getServletContext().getRealPath("/WEB-INF/classes/static/");
    }

```

```

    }

    log.info("일지 등록 폼에서 입력받은 데이터: {}", workoutLog);

    workoutLogUserService.workoutLogInsert(workoutLog, sessionEmail, fileImage,
fileRealPath );

    return "redirect:/workoutLog/workoutLogUser/workoutLogMain";

}

```

■ Service

```

// 일지 등록 처리
public int workoutLogInsert(WorkoutLog workoutLog, String sessionEmail, MultipartFile[]
fileImage, String fileRealPath) {

    String code = commonMapper.getNewCode("workoutLogCode", "workoutlog");
    workoutLog.setWorkoutLogCode(code);
    workoutLog.setMemberEmail(sessionEmail);

    int result = workoutLogUserMapper.workoutLogInsert(workoutLog);

    List<FileDto> fileList = fileUtil.parseFileInfo(fileImage, fileRealPath);

    fileMapper.addFiles(fileList);

    List<Map<String, String>> paramList =new ArrayList<Map<String, String>>();

    Map<String, String> paramMap =null;

    for(FileDto fileDto : fileList) {
        paramMap = new HashMap<String, String>();
        paramMap.put("referenceCode", code);
        paramMap.put("fileIdx", fileDto.getFileIdx());
        paramList.add(paramMap);
    }

    fileMapper.addFileControl(paramList);

    return result;
}

// 운동 목표 목록 조회
public List<WorkoutGoal> getworkoutGoalList(){

    List<WorkoutGoal> workoutGoalList = workoutLogUserMapper.getworkoutGoalList();

    return workoutGoalList;
}

// 일지 공개범위 목록 조회
public List<WorkoutLogPrivacybounds> getworkoutLogPrivacyboundsList(){

    List<WorkoutLogPrivacybounds> workoutLogprivacyboundsList =
workoutLogUserMapper.getworkoutLogPrivacyboundsList();

    return workoutLogprivacyboundsList;
}

```

```
    }  
  
    // 운동 클래스 카테고리 large 목록 조회  
    public List<EClassCategoryLarge> geteClassCategoryLargeList(){  
  
        List<EClassCategoryLarge> eClassCategoryLargeList =  
workoutLogUserMapper.geteClassCategoryLargeList();  
  
        return eClassCategoryLargeList;  
  
    }  
}
```

6.3.30 AI 운동 영상 목록

X
관리짱1님이 로그인하셨습니다.

09:20

Friday, April 22nd 2022

- 관리자 홈
- 회원 >
- 운동클래스 >
- 일지 >
- #일지 관리(민희)
- #운동계획 카테고리 관리 (민희)
- #운동계획 관리.트레이너 (민희)

AI 홈트레이닝 시스템

AI 운동 영상 목록

일지 > AI 운동 영상 목록

AI 운동 영상 목록

전체 4 건


운동 영상 제목	등록일	수정	삭제
토오오오르르르르	2022-04-21	수정	삭제
사이드 런지	2022-04-20	수정	삭제
사이드 런지	2022-04-21	수정	삭제
배틀로프	2022-04-21	수정	삭제

Previous 1 Next

AI 운동 영상 등록

목록

사이드 런지



취소

AI 운동 영상 목록

AI 운동 영상 목록(영상 제목, 이미지, 등록일)을 조회할 수 있습니다.
 영상 제목을 클릭하면 등록된 이미지를 Ajax를 사용하여 Modal에 띄워줍니다.

- 204 -

한국스마트정보교육원 ksmart.or.kr

```

<!-- start: Javascript -->
  <th:block layout:fragment="customBodyScript">
    <script th:src="@{/asset/js/jquery.min.js}"></script>
    <script th:src="@{/asset/js/jquery.ui.min.js}"></script>
    <script th:src="@{/asset/js/bootstrap.min.js}"></script>

    <!-- plugins -->
    <script th:src="@{/asset/js/plugins/moment.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.dataTables.min.js}"></script>
    <script th:src="@{/asset/js/plugins/dataTables.bootstrap.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.nicescroll.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.mask.min.js}"></script>
    <script th:src="@{/asset/js/plugins/jquery.validate.min.js}"></script>

    <!-- custom -->
    <script th:src="@{/asset/js/main.js}"></script>
    <script type="text/javascript">

      $(document).ready(function(){

        /* 데이터 테이블 생성 */
        var table = $('#AIVideoList').DataTable({
          order : [ 0, 'desc' ],
          lengthChange : false,
          pageLength : 15,
          filter : false,
          dom : '<"top"i>rt<"bottom"p><"clear">',
        });
        /* 레코드 수 포맷 */
        var info = table.page.info();
        $('#AIVideoList_info').html(
          '전체 ' + info.recordsTotal + ' 건').css(
            'font-weight', 'bold');
        $('#AIVideoList').click(
          function() {
            $('#AIVideoList_info').html(
              '전체 ' + info.recordsTotal +
              '건');
          });

        /* 모달에 시운동영상 이미지 불러오기 */
        $('#AIVideoImg').click(function(e){

          console.log($(e.target).data('code'));

          var AIVideoImgCodes =$(e.target).data('code');

          $.ajax({
            url: '/admin/workoutAIVideoImg'
            ,type : 'post'
            ,data : {
              'AIVideoImgCodes' : AIVideoImgCodes
            }
            ,dataType : 'text'
            ,success: function(data){

              $('#imgModalBody').replaceWith(data);

            }
          });
        });
      });
    </script>
  </th:block>
</!-- start: Javascript -->

```

```

        });

    });

});
</script>
</th:block>

```

■ Controller

```

//AI 운동 영상 목록
@GetMapping("/workoutAIVideoList")
public String workoutAIVideoList(Model model) {

    List<AIVideo> workoutAIVideoList = workoutLogConfigService.getWorkoutAIVideoList();

    log.info("AI 운동 영상 목록 조회 workoutAIVideoList : {}", workoutAIVideoList);

    model.addAttribute("title", "AI 운동 영상 목록");
    model.addAttribute("leftMenuList", "일지");
    model.addAttribute("workoutAIVideoList", workoutAIVideoList);

    return "workoutLog/workoutLogConfig/workoutAIVideoList";
}

//ajax 등록이미지 조회
@PostMapping("/workoutAIVideoImg")
public String workoutAIVideoImg (String AIVideoImgCodes, Model model){

    log.info("코드 뭐냐어냐:",AIVideoImgCodes);

    List<AIVideo> workoutAIVideoImgCodes = workoutLogConfigService.workoutAIVideoImg(AIVideoImgCodes);

    log.info("AI 등록이미지조회 workoutAIVideoImgCodes : {}", workoutAIVideoImgCodes);
    String imgPath = workoutAIVideoImgCodes.get(0).getFileList().get(0).getFilePath();

    model.addAttribute("imgPath", imgPath);

    return "workoutLog/workoutLogConfig/workoutAIVideoList-imgModal";
}

```

■ Service

```

//AIVideo 영상 목록 조회
public List<AIVideo> getWorkoutAIVideoList(){

    List<AIVideo> AIVideoList = workoutLogConfigMapper.getWorkoutAIVideoList();
}

```

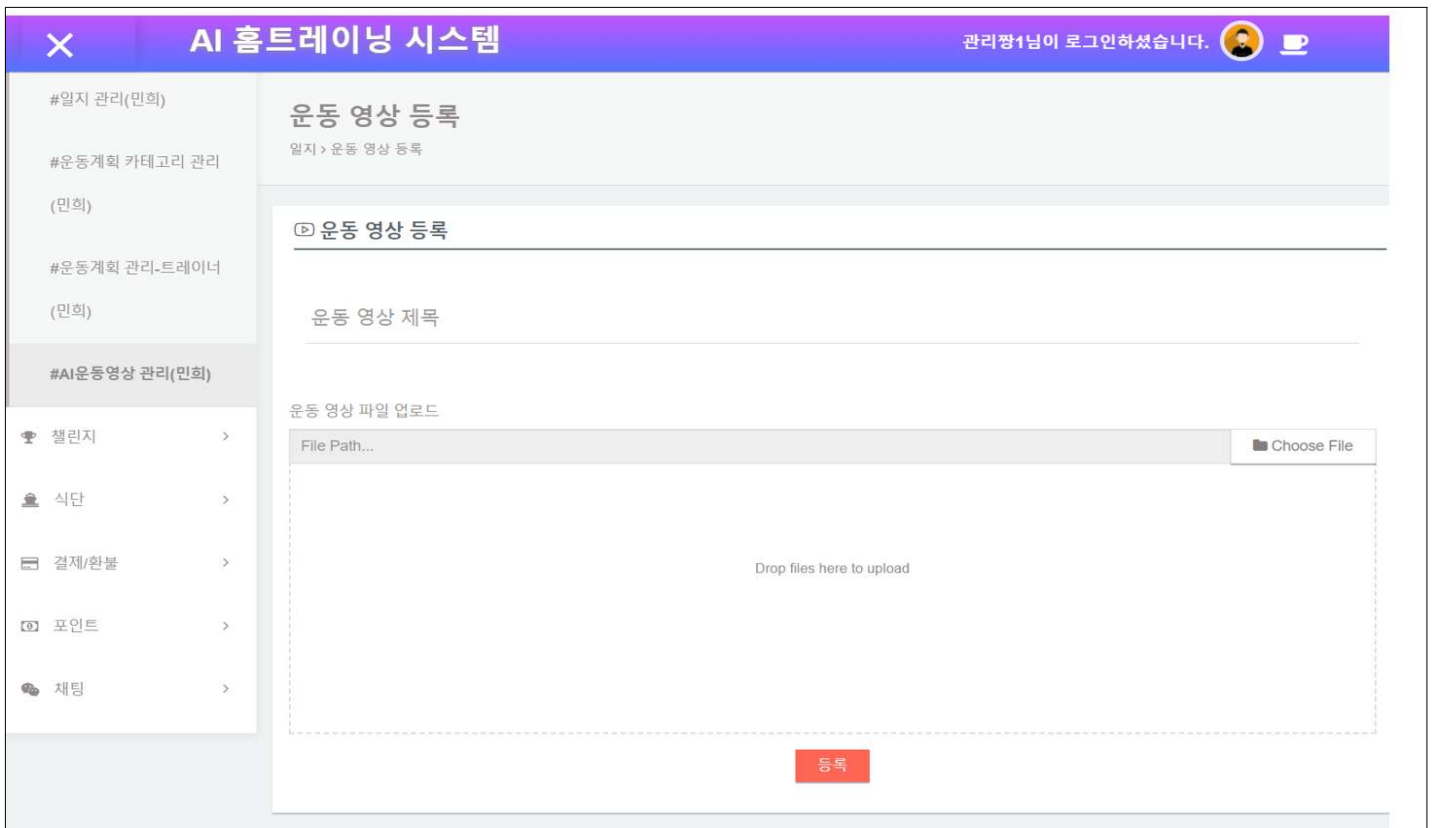
```
        return AIVideoList;
    }

    //ajax 등록이미지 조회
    public List<AIVideo> workoutAIVideoImg(String AIVideoImgCodes){

        List<AIVideo> workoutAIVideoImgCodes =
workoutLogConfigMapper.workoutAIVideoImg(AIVideoImgCodes);

        return workoutAIVideoImgCodes;
    }
```

6.3.31 AI 운동 영상 등록



AI 운동 영상 등록

운동 영상 제목과 파일을 첨부하여 등록할 수 있습니다.

■ JavaScript

```

<!-- 사용자 정의 Script -->
<th:block layout:fragment="customBodyScript">
  <script th:src="@{/asset/js/jquery.min.js}"></script>
  <script th:src="@{/asset/js/jquery.ui.min.js}"></script>
  <script th:src="@{/asset/js/bootstrap.min.js}"></script>
  <!-- plugins -->
  <script th:src="@{/asset/js/plugins/moment.min.js}"></script>
  <script th:src="@{/asset/js/plugins/dropzone.js}"></script>
  <script th:src="@{/asset/js/plugins/jquery.nicescroll.js}"></script>
  <script th:src="@{/asset/js/plugins/jquery.validate.min.js}"></script>

  <!-- custom -->
  <script th:src="@{/asset/js/main.js}"></script>
  <script type="text/javascript">
    $(document).ready(function(){

      $("#workoutAIVideoInsert").validate({
        errorElement: "em",
        errorPlacement: function(error, element) {
          $(element.parent("div")).addClass("form-animate-error");
          error.appendTo(element.parent("div"));
        }
      });
    });
  </script>
</th:block>

```



```

    },
    success: function(label) {
        $(label.parent("div").removeClass("form-animate-error"));
    },
    rules: {
        validate_workoutAIVideoInsert: "required",
    },
    messages: {
        validate_workoutAIVideoInsert: "AI 운동 영상 제목을 입력해주세요.",
    }
    });
});

</script>
<!-- end: Javascript -->
</th:block>

```

■ Controller

```

//AIVideo 영상(이미지 대체) 등록 처리
@PostMapping("/workoutAIVideoInsert")
public String workoutAIVideoInsert( AIVideo aiVideo
                                     , HttpSession session
                                     , @RequestParam
MultipartFile[] fileImage
                                     , HttpServletRequest
request) {

    String sessionEmail = (String) session.getAttribute("SEMAIL");

    //파일 업로드
    String serverName = request.getServerName();
    String fileRealPath = "";
    if("localhost".equals(serverName)) {
        fileRealPath =System.getProperty("user.dir") +"/src/main/resources/static/";
        //fileRealPath
request.getSession().getServletContext().getRealPath("/WEB-INF/classes/static/");
    }else {
        fileRealPath
request.getSession().getServletContext().getRealPath("/WEB-INF/classes/static/");
    }

    log.info("AI운동 영상 등록 폼에서 입력받은 데이터: {}", aiVideo);

    workoutLogConfigService.workoutAIVideoInsert(aiVideo, sessionEmail, fileImage,
fileRealPath);

    return "redirect:/admin/workoutAIVideoList";

}

```

■ Service

```

//AIVideo 영상(이미지 대체) 등록 처리
public int workoutAIVideoInsert(AIVideo aiVideo, String sessionEmail, MultipartFile[] fileImage,
String fileRealPath) {

```

```

String filecode = commonMapper.getNewCode("fileIdx", "aivideo");
aiVideo.setFileIdx(filecode);
aiVideo.setMemberEmail(sessionEmail);

int result = workoutLogConfigMapper.workoutAIVideoInsert(aiVideo);

List<FileDto> fileList = fileUtil.parseFileInfo(fileImage, fileRealPath);

fileMapper.addFiles(fileList);

List<Map<String, String>> paramList =new ArrayList<Map<String, String>>();

Map<String, String> paramMap =null;

for(FileDto fileDto : fileList) {
    paramMap = new HashMap<String, String>();
    paramMap.put("referenceCode", filecode);
    paramMap.put("fileIdx", fileDto.getFileIdx());
    paramList.add(paramMap);
}

fileMapper.addFileControl(paramList);

return result;
}

```

6.3.32 운동 목표 조회



운동 목표 조회

로그인 한 일반 회원의 운동 목표 조회(운동 목표, 수강 클래스, 시작일, 종료일)가 가능합니다.

■ JavaScript

■ Controller

```
// 운동 목표 목록 조회
@GetMapping("/workoutGoalList")
public String workoutGoalList(Model model, HttpSession session) {

    String sessionEmail = (String) session.getAttribute("SEMAIL");

    List<WorkoutGoal> workoutGoalList =
workoutLogUserService.getworkoutGoalList(sessionEmail);
```

```
        model.addAttribute("title", "운동 목표");
        model.addAttribute("workoutGoalList", workoutGoalList);

        return "workoutLog/workoutLogUser/workoutGoalList";
    }
}
```

■ Service

```
// 운동 목표 목록 조회
public List<WorkoutGoal> getworkoutGoalList(String sessionEmail){

    List<WorkoutGoal> workoutGoalList =
workoutLogUserMapper.getworkoutGoalList(sessionEmail);

    return workoutGoalList;
}
```

6.3.33 운동 목표 등록

운동 목표 등록

| 운동 목표 등록

운동 목표 등록하기

운동 목표 등록사항

운동 목표

수강 중인 클래스

다이어트댄스

시작일

종료일

목표 등록

운동 목표 등록

로그인 한 일반 회원의 운동 목표 등록(운동 목표, 수강 클래스, 시작일, 종료일)가 가능합니다.

■ JavaScript

```
<!-- 사용자 정의 Script --> <th:block layout:fragment="customBodyScript">
<!--<script src="https://code.jquery.com/jquery-3.5.1.min.js"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvblyZFJoft+2mJbHaEWIidlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExI3Og8ifwB6"
crossorigin="anonymous"></script>-->
<script>

$(function){
    console.log("이거 되나? 이거 안되면");
```

```

//운동 계획 카테고리 대-중 분류
$(document).on('change','#eClassCategoryLargeCode', function(e) {
    console.log("이거 되나?medium");
    var eClassCategoryLargeCode
$('select[name="eClassCategoryLargeCode"]').val();
    console.log(eClassCategoryLargeCode);
    $.ajax({
        url : '/workoutLog/workoutLogUser/geteClassCategoryMedium',
        type : 'post',
        data : {
            'eClassCategoryLargeCode' : eClassCategoryLargeCode
        },
        dataType : 'json',
        success : function(data) {
            console.log(JSON.stringify(data));
            console.log(data.length);
            var txt1='<option value="none">운동 카테고리 중분류 선택하세요
</option>';
            for(var i =0;i<data.length;i++)
            {
                txt1 += '<option
value="'+data[i].eClassCategoryMediumCode+'"' + '>'+ data[i].eClassCategoryMediumName
+'</option>';
            }
            console.log(txt1);
            $('#eClassCategoryMediumCode').empty();
            $('#eClassCategoryMediumCode').append(txt1);
            $('#eClassCategoryMediumCode').niceSelect('update');
        }
    });
});

//운동 계획 카테고리 중-소 분류
$(document).on('change','#eClassCategoryMediumCode', function(e) {
    console.log("이거 되나?small");
    var eClassCategoryMediumCode
$('select[name="eClassCategoryMediumCode"]').val();
    console.log(eClassCategoryMediumCode);
    $.ajax({
        url : '/workoutLog/workoutLogUser/geteClassCategorySmall',
        type : 'post',
        data : {
            'eClassCategoryMediumCode' : eClassCategoryMediumCode
        },
        dataType : 'json',
        success : function(data) {
            console.log(JSON.stringify(data));
            console.log(data.length);
            var txt1='<option value="none">운동 카테고리 소분류 선택하세요
</option>';
            for(var i =0;i<data.length;i++)
            {
                txt1 += '<option
value="'+data[i].eClassCategorySmallCode+'"' + ' data-cate="'+ data[i].eClassCategorySmallName
+'>'+ data[i].eClassCategorySmallName + "</option>";
            }
            console.log(txt1);
            $('#eClassCategorySmallCode').empty();
            $('#eClassCategorySmallCode').append(txt1);

```

```

        $('#eClassCategorySmallCode').niceSelect('update');
    }
    });
});

//등록 처리시 form을 submit 하기 전에 특정값을 추가
$(document).on('click','#workoutLogInsertbtn',function(){
    var datacate = $('#eClassCategorySmallCode
option:selected').attr('data-cate');
    //alert(datacate);
    var eClassCategorySmall = $("<input type='hidden' value='"+
datacate +" name='eClassCategorySmallName'/>");
    $("#workoutLogInsert").append(eClassCategorySmall);
    $("#workoutLogInsert").submit();
});

//회원만 일지 등록하는 처리

var sessionLevel = '[[${session.SLEVEL}]];

$('#workoutLogInsertbtn').click(function(){

    if(sessionLevel == 'levelCode001'){
        alert('회원만 일지 등록할 수 있습니다.')
        return false;
    }

});

});
</script>
</th:block>

```

■ Controller

```

// 운동 목표 등록 @GetMapping("/workoutGoalInsert")
public String workoutGoalInsert(Model model, @RequestParam(value = "memberEmail",
required = false) String memberEmail) {

    // 수강 중인 운동 클래스 목록 조회
    List<EClassTake> eClassTakeList =
workoutLogUserService.getEClassTakeList(memberEmail);
    log.info("수강 중인 운동 클래스 목록 조회 eClassTakeList : {}", eClassTakeList);

    model.addAttribute("title", "운동 목표 등록");
    model.addAttribute("eClassTakeList", eClassTakeList);
    return "workoutLog/workoutLogUser/workoutGoalInsert";

}

// 운동 목표 등록 처리
@PostMapping("/workoutGoalInsert")

```

```

public String workoutGoalInsert(WorkoutGoal workoutGoal, HttpSession session) {
    String sessionEmail = (String) session.getAttribute("SEMAIL");
    log.info("운동 목표 등록 폼에서 입력받은 데이터: {}", workoutGoal);
    workoutLogUserService.workoutGoalInsert(workoutGoal, sessionEmail);

    return "redirect:/workoutLog/workoutLogUser/workoutGoalList";
}

```

■ Service

```

// 운동 목표 등록 처리
public int workoutGoalInsert(WorkoutGoal workoutGoal, String sessionEmail) {

    String code = commonMapper.getNewCode("workoutGoalCode", "workoutgoal");
    workoutGoal.setWorkoutGoalCode(code);
    workoutGoal.setMemberEmail(sessionEmail);

    int result = workoutLogUserMapper.workoutGoalInsert(workoutGoal);

    return result;
}

// 수강 중인 운동 클래스 목록 조회
public List<EClassTake> getClassTakeList(String memberEmail){

    List<EClassTake> eClassTakeList =
workoutLogUserMapper.getClassTakeList(memberEmail);

    return eClassTakeList;
}

```