

TEAM_NTP 권지현 노은실 정유진

CONTENTS

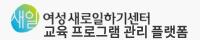
01/ 프로젝트개요

02/ 프로젝트설계

03/ 수행과정및개선사항

04/ 진행상황 및 자체 평가

05/ 시연영상

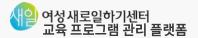


기존방식

(학교) 출석부, 훈련일지 양식 인쇄 (선생님) 출석부, 훈련일지 수기 작성 (학교) 작성한 출석부, 훈련일지 센터로 전달 (센터) 출석부, 훈련일지 재작성 후 출력 (센터) 학교 방문, 학생 날인 받기 (센터) 최종 결재

플랫폼도입후

(전산) 출석부, 훈련일지 작성, 수정 및 조회 ↓ 최종 결재



학생

(기존) 센터 혹은 선생님께 요청 후 본인 출결 상태 조회 (도입 후) 웹 상에서 본인의 출결 상태 조회 가능

선생님

(기존)출석부와 훈련일지 수기로 작성 (도입 후)출석부와 훈련일지 작성 전산화

센터

(기존)복잡하고 번거로운 서류 처리 과정 (도입 후)간소화된 서류 처리로 시간과 비용 절감 효과



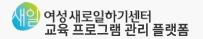
<mark>권지현(팀장)</mark> UI, FRONT-END, BACK-END, DB설계, 문서 로그인(세션,구글)/로그아웃/회원가입 공지사항/훈련과정/수강등록/출석부

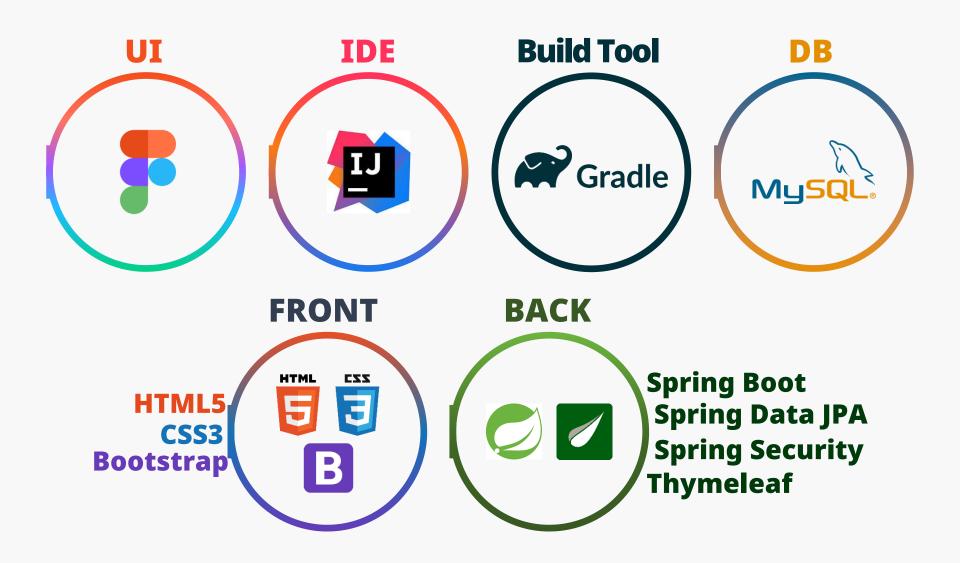


<mark>노은실(팀원)</mark> UI, BACK-END, 문서 훈련일지/출석부/파일 업로드/파일 다운로드

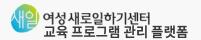


<mark>정유진(팀원)</mark> UI, BACK-END, 문서 훈련과정/사유서/출석부/파일 업로드/파일 다운로드





02. 프로젝트 설계 - UI / 로그인, 회원가입







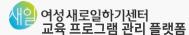
	외권 정보들 합력에 무세요.
	아이디
	비밀번호
	비밀번호확인
	이름
3	휴대전화번호
	e-mail
	주소 검색
	우편번호
	주소 -
	상세주소
	가입 취소
6	그 저나로 이려해 주비이
Ŷ	가 정보를 입력해 주세요.
이름	
휴대	전화번호
주소	검색
우편	번호
주소	
상세=	주소

취이 거나로 이러센 지네이

Description

- 1 일반-회원가입
- 2 일반-회원유형
- 3 일반-회원정보
- 4 구글-로그인
- 5 구글-유형
- 6 구글-회원정보

02. 프로젝트 설계 – UI / 레이아웃







사일 여성새로일하기센터

대구남부새일센터 1

email@email.com

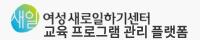


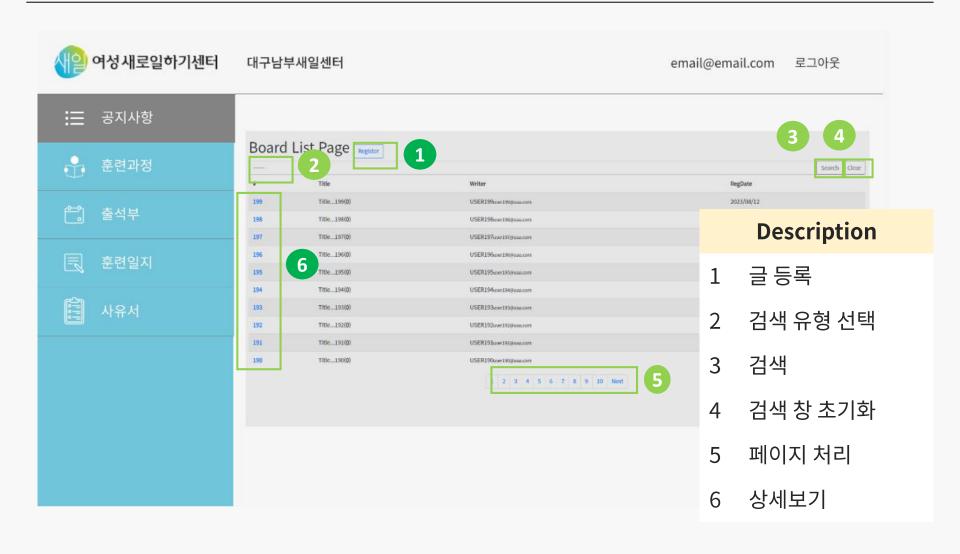


Description

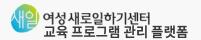
- 상단바
- 로그인 회원 정보
- 로그아웃
- 사이드바

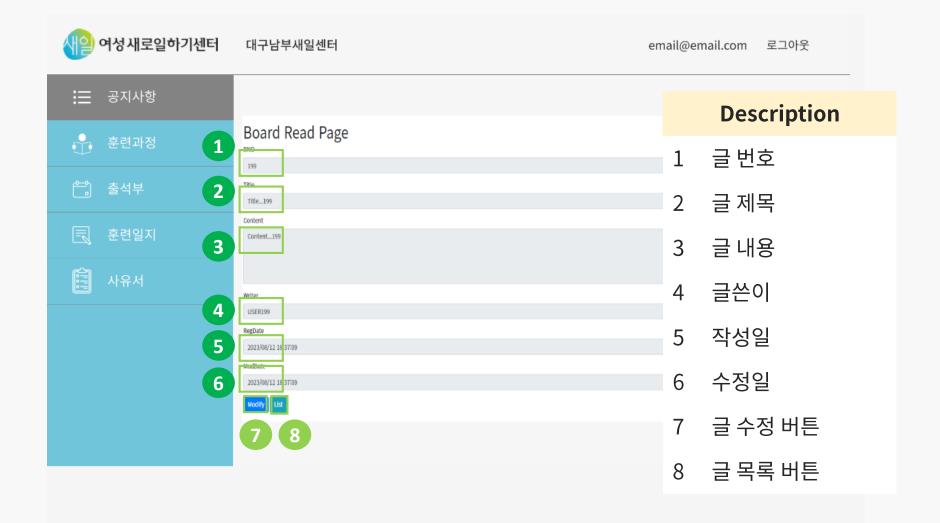
02. 프로젝트 설계 - UI / 공지사항



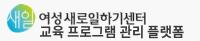


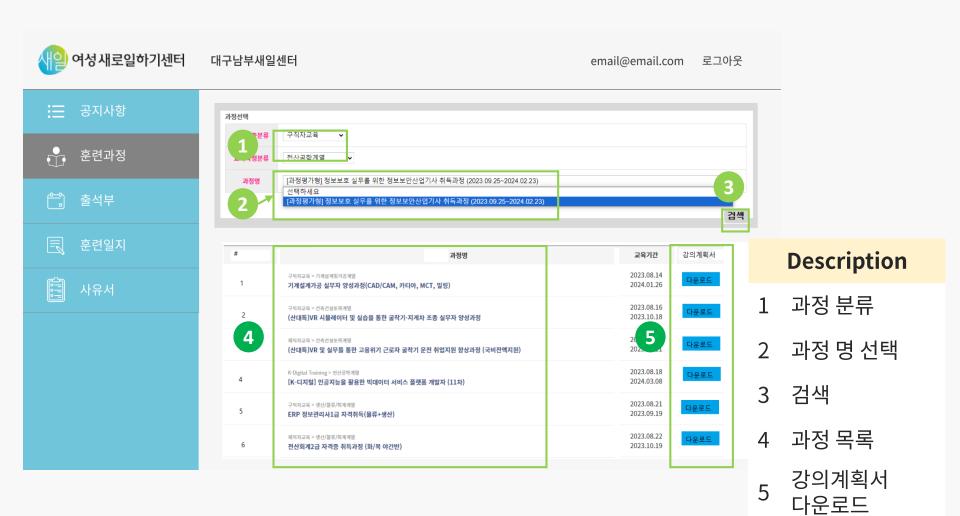
02. 프로젝트 설계 - UI / 공지사항



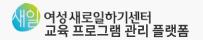


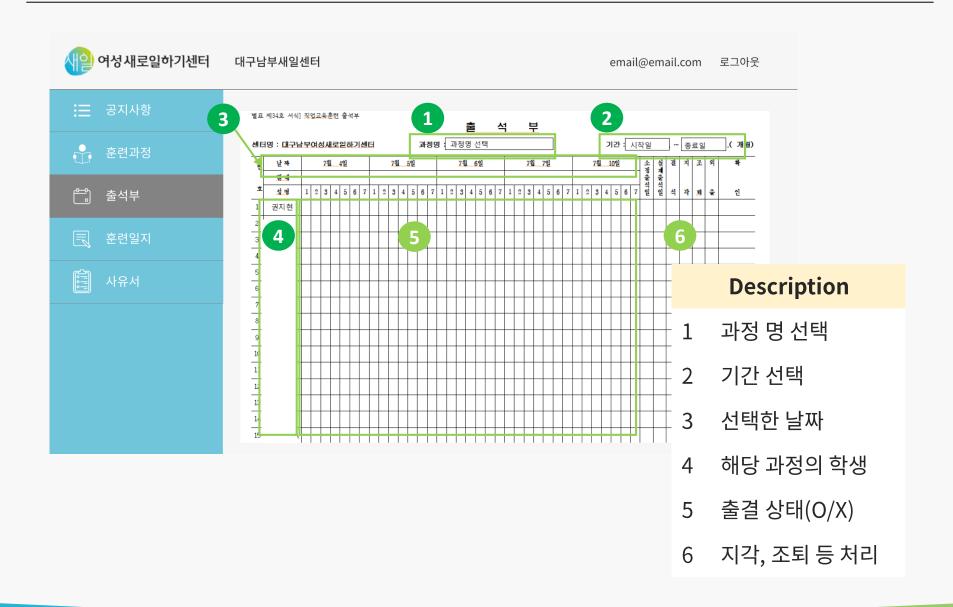
02. 프로젝트 설계 - UI / 훈련 과정



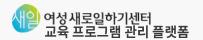


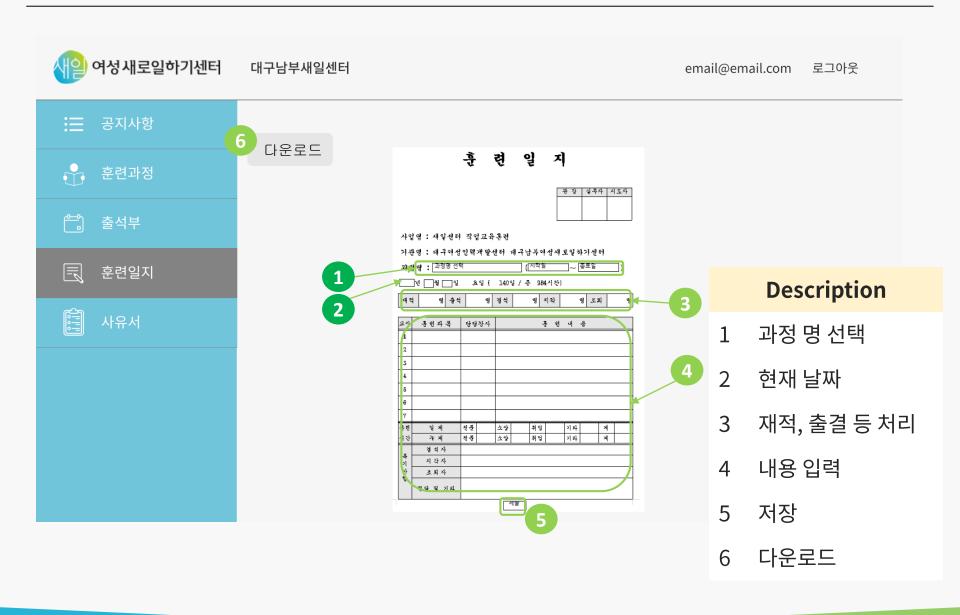
02. 프로젝트 설계 - UI / 출석부



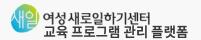


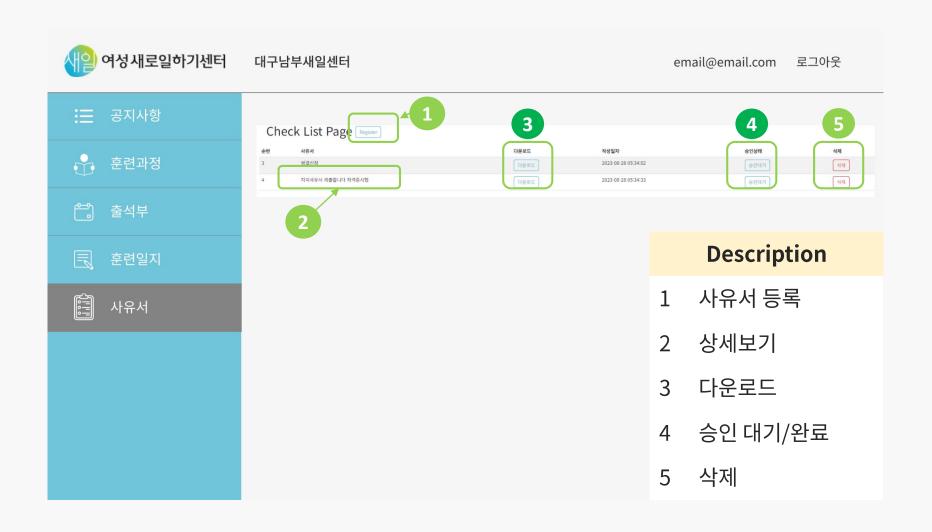
02. 프로젝트 설계 - UI / 훈련일지



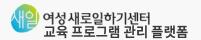


02. 프로젝트 설계 - UI / 사유서



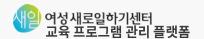


02. 프로젝트 설계 – 요구사항 정의서



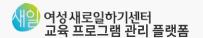
	회원				
회원가입	/sign(회원 유형)->/signUp(회원 정보 입력)				
로그인	/login				
로그아웃	/logout				
공지사항					
목록	/member/board/list				
등록	/member/board/register				
수정	/member/board/modify				
삭제	/member/board/remove				
출석부					
목록	/member/attendance/form, get				
등록	록 /member/attendance/form, post				

02. 프로젝트 설계 – 요구사항 정의서



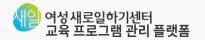
훈련과정 목록 /member/course/list 등록 /member/course/register 상세 /member/course/read 삭제 /member/course/delete 강의계획서 업로드 /uploadAjax 조회, 다운로드 /display					
등록 /member/course/register 상세 /member/course/read 삭제 /member/course/delete 강의계획서 업로드 /uploadAjax					
상세 /member/course/read 삭제 /member/course/delete 강의계획서 업로드 /uploadAjax					
삭제 /member/course/delete 강의계획서 업로드 /uploadAjax					
강의계획서 업로드 /uploadAjax					
업로드 /uploadAjax					
조회, 다운로드 /display					
 훈련일지					
목록 /member/training/form, get					
등록 /member/training/form, post					
 학생					
수강 등록 /member/student/register, get(form 이동), post(실제 등록 처리)					

02. 프로젝트 설계 – 요구사항 정의서



사유서				
목록	/member/check/list			
등록	/member/check/CWriteForm			
수정	/member/check/CUpdateForm			
삭제	/member/check/Delete			
사유서 파일				
업로드	/member/check/CWriteForm			
다운로드	/member/check/download			

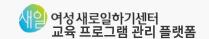
02. 프로젝트 설계 - DB / 공통 코드



공통코드	상위공통코드	공통코드명	공통코드영문명	사용여부(Y/N)	등록일시	수정일시
100		정부부처	Government Department			
10001	100	여성가족부	Gender Equality and Family			
101		새일센터	Saeil Center			
10101	101	대구남부여성새로일하기센터	Daegu Nam-bu Saeil Center			
200		훈련과정분류	Training Job Classification			
20001	200	직업교육훈련	Vocational education and training			
300		계정상태	Account Status			
30001	300	대기	standby			
30002	300	승인	approval			
30003	300	거부	rejection			
400		게시물상태	Post Status			
40001	400	임시저장	temporary			
40002	400	승인	approval			

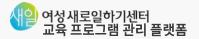
^{***} 해당 프로젝트 진행에 필요한 코드만 작성하였습니다.

02. 프로젝트 설계 - DB / 테이블 설계

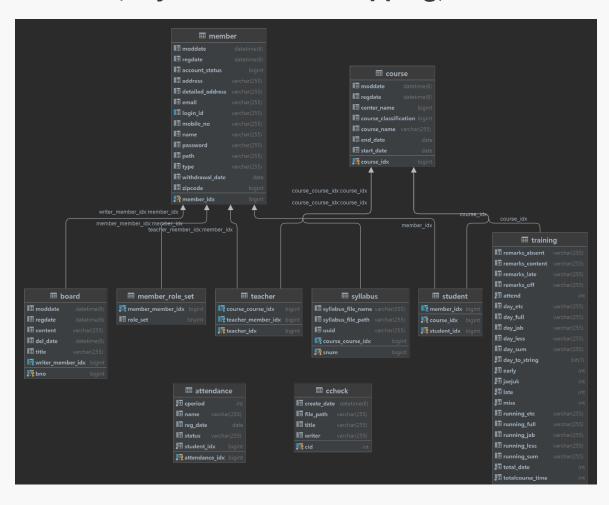


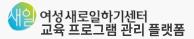
회원	훈련과정(course)	공지사항(board)	출석부(attendance)	사유서(check)	훈련일지(training)
회원idx(PK)	훈련과정idx(PK)	bno(PK)	출석부idx(PK)	사유서idx(PK)	훈련일지idx(PK)
loginId	훈련과정분류(직업교육훈련)	회원idx(FK)	훈련과정idx(FK)	출석부idx(FK)	출석부idx(FK)
Password	센터명(대구남부여성새로일하	제목	회원idx(FK)	과정idx(FK)	훈련과정idx(FK)
joindate	시작날짜	내용	날짜(연도,월,일)	회원idx(FK)	회원idx(FK)
가입경로(form,google)	종료날짜	등록일	교시(1~7교시)	사유서파일위치	교시(1~7교시)
withdrawaldate	강의계획서다운로드파일위치	수정일	교시별출석상태(출석/결석)	사유서파일이름	교시별훈련과목
계정 상태(대기,승인,거부)	강의계획서다운로드파일이름	게시여부	출석상태(출석,결석,지각,조퇴)	사유(내용)	교시별담당강사
권한(학생/선생님/행정/관리자)					교시별훈련내용
회원상세정보id(FK)			교시별출결상태		
이름			교시별출결상태idx(PK)		날인란
생년윌일			출석부idx(FK)		관장
성별			교시(1~7교시)		실무자
연락처			교시별출석상태(출석,결석)		지도자
이메일					
우편번호			교시별훈련내용		제적
주소			교시별훈련내용idx(PK)		출석
상세주소			훈련일지idx(FK)		결석
훈련과정ID(FK)			교시(1~7교시)		지각
			교시별훈련과목		조퇴
			교시별담당강사		
			교시별훈련내용		특기사항

^{*} 다음 프로젝트에 진행: 교시별 테이블 분리, 날인 전산화(회원 도장 이미지)



ORM(Object Relational Mapping) 현재 상태



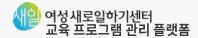


```
QConfiguration
public class WebConfig implements WebMvcConfigurer {

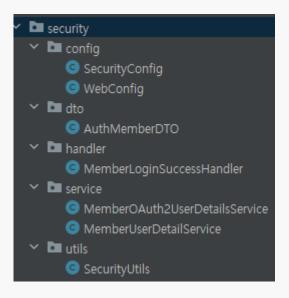
3 usages
QOverride
public void addViewControllers(ViewControllerRegistry registry) {
WebMvcConfigurer.super.addViewControllers(registry);
// root 경로 redirect 처리
registry.addViewController( urlPathOrPattern: "/").setViewName("redirect:/login");
}
```

현재 구현된 메인 페이지가 없어서 루트 경로를 /login 페이지로 설정 개선 사항: 메인 페이지 구현

03. 수행 과정 및 개선 사항 – 로그인, 로그아웃

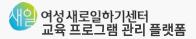


프로젝트 구조



Security Config

```
.formLogin((login) -> login
        .loginPage("/login")
        .defaultSuccessUrl("/")
        .loginProcessingUrl("/login")
        .usernameParameter("loginId")
        .successHandler(memberLoginSuccessHandler())
        .permitAll()
.logout((logout) -> logout
        .logoutRequestMatcher(new AntPathRequestMatcher(⊘"/logout"))
        .logoutSuccessUrl("/")
        .invalidateHttpSession(true)
        .permitAll())
.oauth2Login(oAuth2LoginConfigurer -> oAuth2LoginConfigurer
        .loginPage("/login")
        .defaultSuccessUrl("/")
        .authorizationEndpoint(authorizationEndpointConfig -> authorizationEndpointConfig
                .baseUri( authorizationRequestBaseUri: "/oauth2/authorize"))
        .successHandler(memberLoginSuccessHandler()))
// JS 와 함께 사용하는 경우 withHttpOnlyFalse() 활성화
// 보통 CSRF 옵션은 POST, DELETE, UPDATE 방식에 적용함
.csrf((csrf) -> csrf
        .csrfTokenRepository(CookieCsrfTokenRepository.withHttpOnlyFalse()))
```



SecurityUtils – getAuthList(Authentication authentication)

```
public class SecurityUtils {

1 usage

public static List<String> getAuthList(Authentication authentication) { // 회원 권한 List<String>로 가져오기

Collection<? extends GrantedAuthority> authorities = authentication.getAuthorities();

Iterator<? extends GrantedAuthority> auth = authorities.stream().iterator();

List<String> authList = new ArrayList<>();

while (auth.hasNext()) {

String strAuth = auth.next().toString();

authList.add(strAuth);

}

// 자체회원(form)인지 구글회원(google) 확인할 때 필요한 변수들

String path = authMemberDTO.getPath(); // form(자체), google(구권)

boolean passwordChecker = passwordEncoder.matches( TawPasswordC "@GGOOGLEGG", authMemberDTO.getPassword());

return authList;

}

// 전한 정보 List<String> 에 당아서 가져오기

List<String> authList = SecurityUtils.getAuthList(authentication);

// form 회원인지 google 회원인지 구분 후 권한에 따라 되어지 분기
```

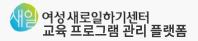
SuccessHandler 페이지 분기 처리

```
log.info("This Member is from " + path);

// 권한 정보 List<String> authList = SecurityUtils.getAuthList(authentication);

// form 회원인지 google 회원인지 구분 후 권한에 따라 페이지 분기
if (path.equals("form")) { // 자체 회원
    if (authList.contains("ROLE_ADMIN")) {
        redirectStrategy.sendRedirect(request, response, luft "/admin");
    } else if (authList.contains("ROLE_MEMBER")) {
        redirectStrategy.sendRedirect(request, response, luft "/member/board/list");
    }
} else if(path.equals("google")) {
    if (authList.contains("ROLE_ADMIN")) {
        redirectStrategy.sendRedirect(request, response, luft "/admin");
    } else if (authList.contains("ROLE_MEMBER")) {
        redirectStrategy.sendRedirect(request, response, luft "/member/board/list");
    }
} else {
    HttpSession session = request.getSession();
        session.setAttribute( name "authMemberDTO", authMemberDTO);
        redirectStrategy.sendRedirect(request, response, luft "/snsSign?path=google");
}
```

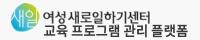
03. 수행 과정 및 개선 사항 – 로그인, 로그아웃



로그아웃 POST 방식 처리(CSRF)

```
<!-- 로그아웃 post 방식 처리 Start -->
<a class="nav-link" href="#" onclick="document.getElementById('logout-form').submit();">로그아웃</a>
<form id="logout-form" th:action="@{/logout}" method="post">
<input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}"/>
</form>
<!-- 로그아웃 post 방식 처리 End -->
```

개선 사항 센터, 관리자 페이지 구현 후 권한 별 페이지 분기 처리 로그아웃을 하지 않고 페이지 접근했을 경우 처리

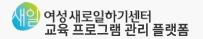


최초 로그인 시 임시 가입 처리

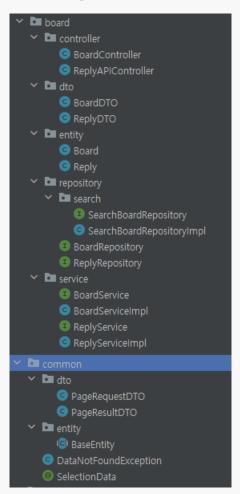
임시 가입 처리 후회원 가입 폼으로 리다이렉트

```
} else if(path.equals("google")) {
    if(!passwordChecker) {
        if (authList.contains("ROLE_ADMIN")) {
            redirectStrategy.sendRedirect(request, response, url: "/admin");
        } else if (authList.contains("ROLE_MEMBER")) {
            redirectStrategy.sendRedirect(request, response, url: "/member/board/list");
        }
    } else {
        HttpSession session = request.getSession();
        session.setAttribute( name: "authMemberDTO", authMemberDTO);
        redirectStrategy.sendRedirect(request, response, url: "/snsSign?path=google");
    }
}
```

03. 수행 과정 및 개선 사항 – 공지사항



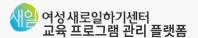
프로젝트 구조



PageResultDTO

```
public PageResultDTO(Page<EN> result, Function<EN,DTO> fn) {
    dtoList = result.stream().map(fn).collect(Collectors.toList());
    totalPage = result.getTotalPages();
   makePageList(result.getPageable());
public void makePageList(Pageable pageable) {
    this.page = pageable.getPageNumber() + 1; // 0부터 시작하므로 + 1
    this.size = pageable.getPageSize();
   // todo temp end page (아직 개선의 여지가 있음..)
    start = tempEnd - 9;
    end = totalPage > tempEnd ? tempEnd : totalPage;
    pageList = IntStream.rangeClosed(start,end).boxed().collect(Collectors.toList());
```

03. 수행 과정 및 개선 사항 – 공지사항

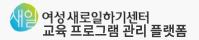


```
0verride
oublic Page<Object[]> <mark>searchPage</mark>(String type, String keyword, Pageable pageable) {
   QBoard board = QBoard.board;
   QMember member = QMember.member;
   QReply reply = QReply.reply;
   JPQLQuery<Board> jpqlQuery = from(board);
   jpqlQuery.leftJoin(reply).on(reply.board.eq(board));
   JPQLQuery<Tuple> tuple = jpqlQuery.select(board, member, reply.count());
   BooleanBuilder booleanBuilder = new BooleanBuilder();
   BooleanExpression expression = board.bno.qt( right: 0L);
   booleanBuilder.and(expression);
                                                                                                                       , pageable
       String[] typeArr = type.split( regex: "");
       BooleanBuilder conditionalBuilder = new BooleanBuilder();
       for (String t:typeArr) {
           switch (t){
                   conditionalBuilder.or(board.title.contains(keyword));
                   conditionalBuilder.or(board.content.contains(keyword));
```

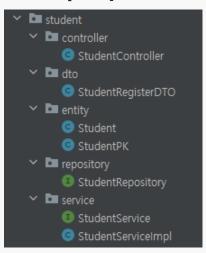
```
tuple.groupBy(board);
this.getQuerydsl().applyPagination(pageable, tuple);
long count = tuple.fetchCount();
```

검색, 페이지 처리

03. 수행 과정 및 개선 사항 – 수강 등록



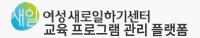
프로젝트 구조



학생 코드

```
public class StudentRegisterDTO {
   private static Long serialNum = 1000L; // 학생 일련번호(4자리)
   @Builder.Default
   private static Long year = (long) LocalDate.now().getYear(); // 연도(4자리)
   private Long studentIdx;
   private Long memberIdx;
   private Long courseIdx; // 과정코드(3자리)
   private LocalDate startDate;
   public Long createStudentIdx(long courseIdx) { // 학생 코드 변수 처리
       String year = String.valueOf(LocalDate.now().getYear());
       String strCourseIdx = String.valueOf(courseIdx);
       String serialNum = String.valueOf(this.serialNum);
       String strStudentIdx = year + strCourseIdx + serialNum;
       long studentIdx = Long.parseLong(strStudentIdx);
       this.serialNum++;
       return studentIdx;
```

03. 수행 과정 및 개선 사항 – 수강 등록



```
@Builder
@AllArgsConstructor
@NoArgsConstructor
@Getter
@EqualsAndHashCode // 필수
@ToString(exclude = {"member","course"})
@IdClass(StudentPK.class) // 식별자 클래스에 아이디가 모두 존재해야 함
public class Student {
   // todo 복합키 기존(studentIdx,courseIdx) -> 변경(memberIdx, courseIdx)
   @Column(name="student_idx")
   private Long studentIdx; // 학생코드-연도(4자리)과정(3자리)학생일련번호(4자리)
   @ManyToOne(fetch = FetchType.LAZY)
   @JoinColumn(name="course_idx")
   private Course course;
   @ManyToOne(fetch = FetchType.LAZY)
   @JoinColumn(name="member_idx")
   private Member member;
```

```
1 usage

@Getter @Setter

@EqualsAndHashCode // 필수

@NoArgsConstructor

@@AllArgsConstructor

public class StudentPK implements Serializable {

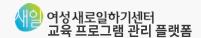
   private Long studentIdx;

   private Long course;

}
```

복합키(중복 수강 방지)

03. 수행 과정 및 개선 사항 - 출석부

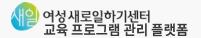


프로젝트 구조

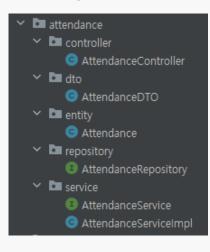
```
$('#combinedSubmit').on('click', function () {
   let formData2 = $('.cperiod').map((idx, item) => {
        idx = idx % 35;
        if (-1 < idx && idx < 7)
            regDate = day1;
        if (6 < idx && idx < 14)
            regDate = day2;
        if (13 < idx && idx < 21)
            regDate = day3;
        if (20 < idx && idx < 28)
        if (27 < idx && idx < 35)
            regDate = day5;
           member_idx: item.name.split("-")[1],
           name: item.name.split("-")[0],
           cperiod: item.name.split("-")[2],
            regDate,
            status: item.value
   }).get();
    console.log(formData2)
            "list": JSON.stringify(formData2)
          beforeSend: function (xhr) { // csrf 처리
```

FRONT JSON 전송 처리

03. 수행 과정 및 개선 사항 - 출석부



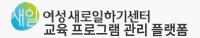
프로젝트 구조



```
@PostMapping(@>"/form")
public @ResponseBody HashMap<String, Object> save(HttpServletRequest request) {
   String jsonStr = request.getParameter( name: "list");
   System.out.println(jsonStr);
       JSONArray jsonArray = new JSONArray(jsonStr);
       for (int i = 0; i < jsonArray.length(); i++) {</pre>
           JSONObject jsonObj = jsonArray.getJSONObject(i);
           String cperiod = (String) jsonObj.get("cperiod");
           String name = (String) jsonObj.get("name");
           String member_idx = (String) jsonObj.get("member_idx");
           String regDate = (String) jsonObj.get("regDate");
           String status = (String) jsonObj.get("status");
           AttendanceDTO attendanceDTO = AttendanceDTO.builder()
                    .cperiod(Integer.parseInt(cperiod))
                    .courseIdx(11)
                    .studentIdx(Long.parseLong(member_idx))
                    .regDate(LocalDate.parse(regDate))
                    .name(name)
                    .build();
           System.out.println(attendanceDTO);
           attendanceService.save(attendanceDTO);
   } catch (JSONException e) {
       e.printStackTrace();
   HashMap<String, Object> retVal = new HashMap<>();
   retVal.put("message", "success");
```

BACK DB INSERT 처리

03. 수행 과정 및 개선 사항 – 훈련 일지



TrainingDTO

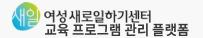
```
private String trainingCourse1;
private String trainingTeacher1;
private String trainingContent1;
private String trainingCourse2;
private String trainingTeacher2;
private String trainingContent2;
private String trainingCourse3;
private String trainingTeacher3;
private String trainingContent3;
private String trainingCourse4;
private String trainingContent4;
private String trainingCourse5;
private String trainingContent5;
private String trainingCourse6;
private String trainingTeacher6;
private String trainingContent6;
private String trainingCourse7;
private String trainingTeacher7;
private String trainingContent7;
```

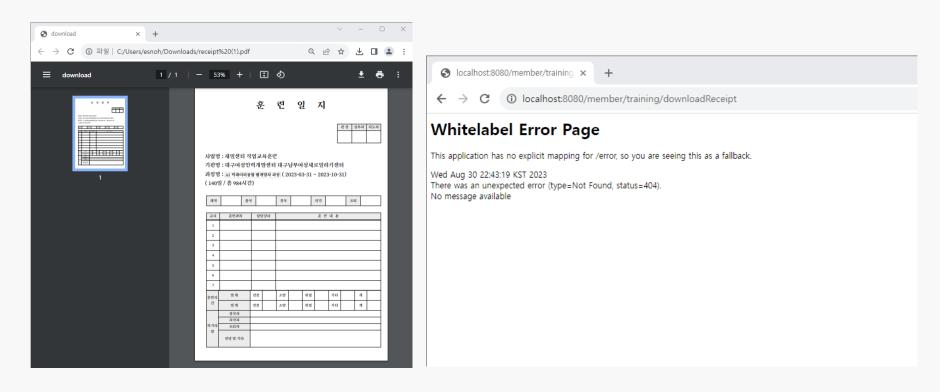
교시	훈련과목	담당강사	훈 런 내 용
1			
2			
3			
4			
5			
6			
7			

모든 입력 값에 컬럼 명을 설정해 데이터를 저장

개선 사항 훈련과목, 담당강사, 훈련내용으로 컬럼을 만들고 배열을 이용해서 입력 값을 저장하도록 변경

03. 수행 과정 및 개선 사항 – 훈련 일지

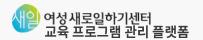


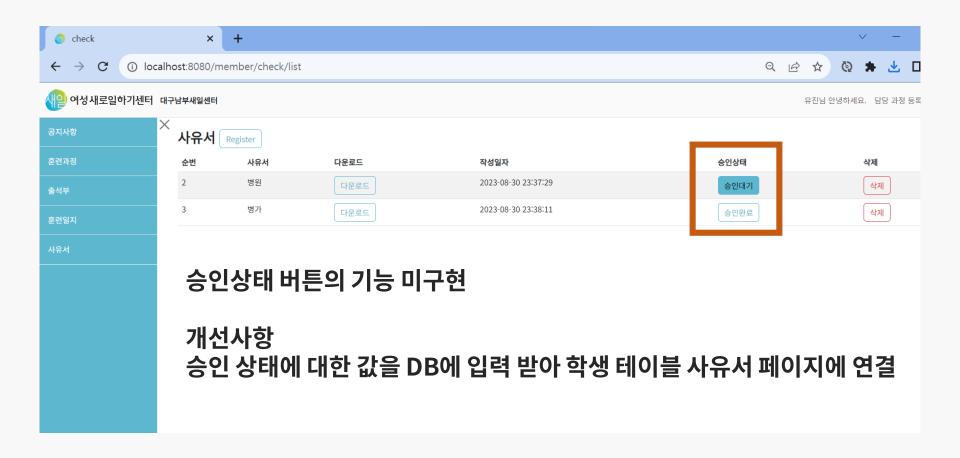


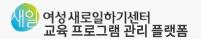
훈련일지 pdf 다운로드 시 DB에서 값을 받아서 다운로드 하는데 실패

개선 사항 다운로드 기능 구현

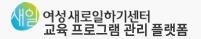
03. 수행 과정 및 개선 사항 – 사유서







1						
	센터	선생님	학생			
회원가입						
공지사항						
훈련과정						
출석부						
훈련일지						
사유서						
마이페이지						
관리자페이지						





권지현(팀장)

설계 부분에서 많은 어려움을 느꼈습니다. 설계 미스와 소요 기간을 제대로 판단하지 못해 팀원에게 부담을 많이 주었습니다. 협업을 하면서 동료의 소중함을 느꼈습니다. 혼자서는 이만큼 진행할 수 없었습니다. 다들 감사합니다.



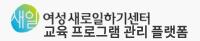
노은실(팀원)

프로젝트를 하면서 수업에서 배운 것과 직접 적용하는 것은 다르다는 것을 알게 되었습니다. 다양한 오류가 답답하긴 했지만 스스로 해결하면서 많이 배울 수 있었습니다.



정유진(팀원)

팀 프로젝트를 하면서 처음으로 모듈 별 작업을 해보았습니다. 각자 개발을 하더라도 얽히지 않고 작업이 가능해서 재미있었습니다. 협업하면서 많이 배울 수 있는 유익한 시간이었습니다. **05. 시연 영상** 용량문제 따로저장



감사합니다. 많은 조언 부탁드립니다.