PROJECT :

새로운 가치를 창조하는 개발자들

TEAM. JJOL

팀 원

팀장 : 이형철

팀원 : 고원

팀원 : 김동욱

팀원 : 김희선

발표날짜 : 2024년 7월 19일

프로젝트 개요

저희는 온라인 학습의 장벽을 낮추고, 누구나 어디서나 양질의 교육을 받을 수 있도록 돕기 위해 온라인 인터넷 강의 플랫폼을 개발하게 되었습니다.
특히, 시간과 장소에 구애받지 않고 공부할 수 있는 환경을 제공하여, 바쁜 일상 속에서도 학습을 지속하고자 하는 사람들에게 도움이 되고자 했습니다.

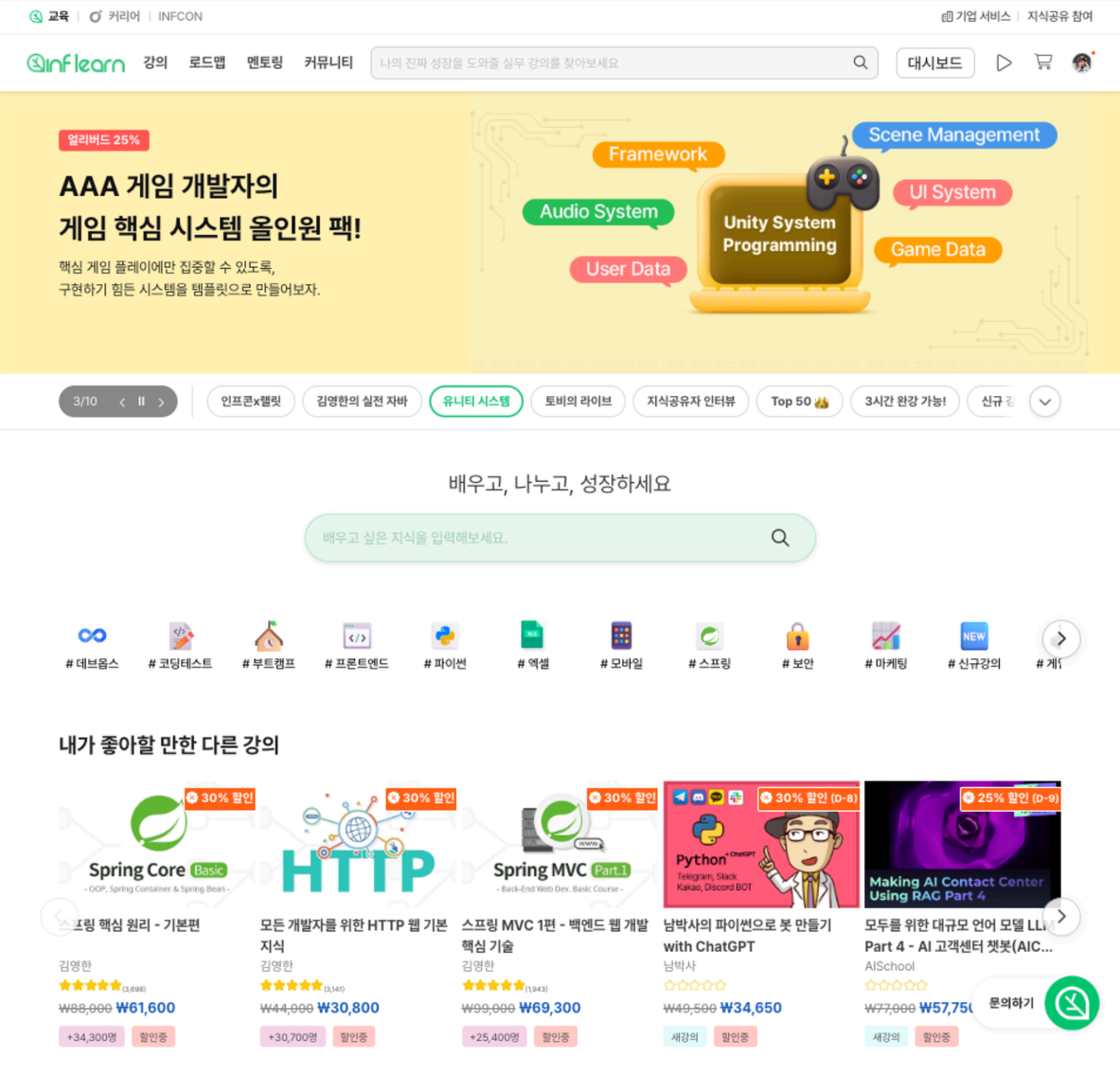
주요 기능

- 소셜네트워크 로그인, 회원가입
- 강의 결제
- 챗봇
- 스터디 그룹 커뮤니티 페이지
- 실시간 채팅
- 실시간 스트리밍 기능
- 동영상 마지막 시청 시간 저장 및 이어서 보기

기존 서비스

인프런

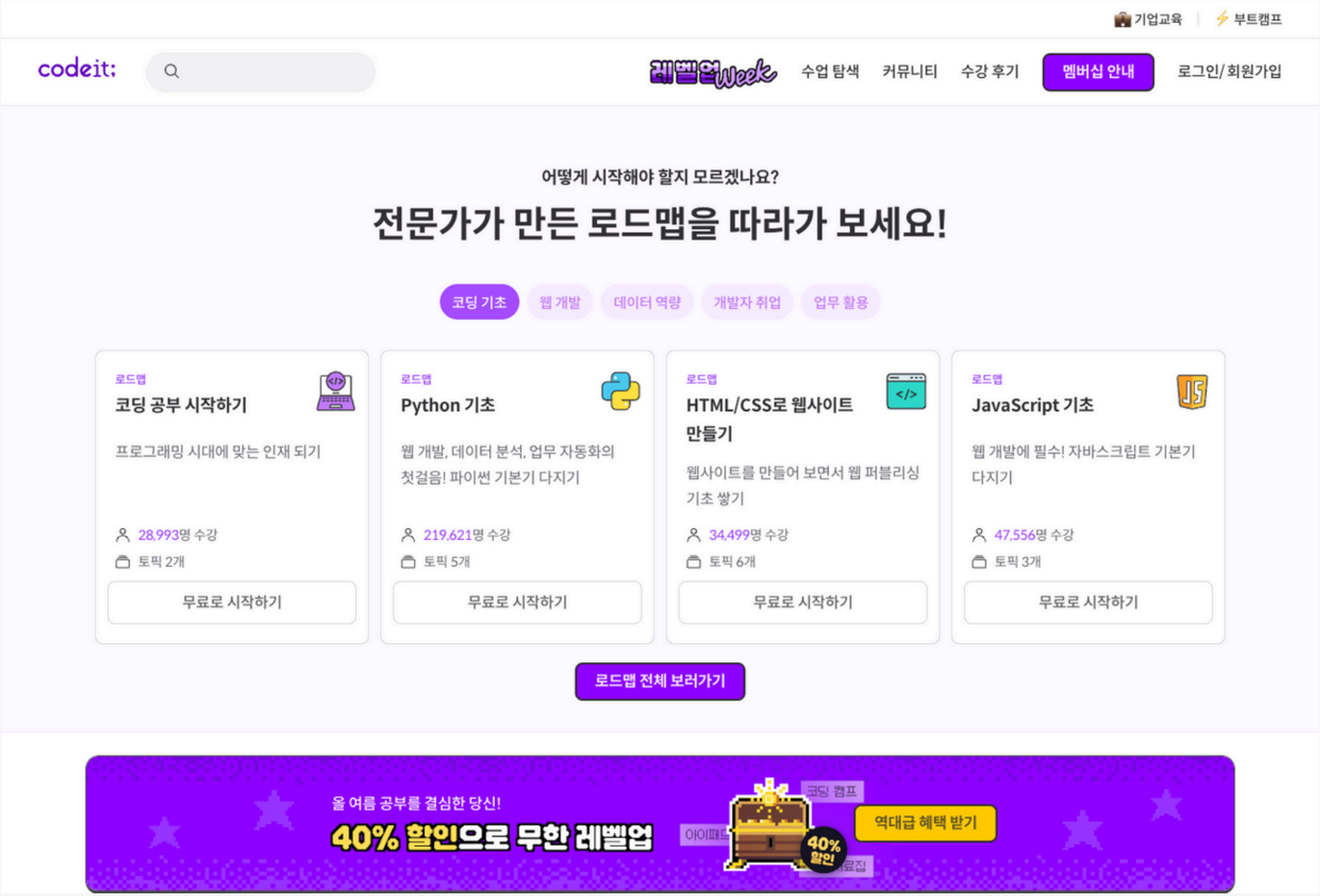
프로그래밍, 데이터 과학, 디자인,
비즈니스 등 다양한 분야의
강의를 제공하여
폭넓은 학습 기회를 제공함.



기존 서비스

코드잇

초보자부터 고급 사용자까지 단계별로
학습할 수 있는 체계적인
코딩 교육 커리큘럼을 제공하여,
학습자들이 효과적으로
실력을 향상시킬 수 있음.



제공 서비스

다양한 분야의 인터넷강의
스트리밍 서비스

유저간의 실시간 채팅

간편결제 및 포인트를
이용한 편리한 결제 시스템

그룹 스터디를 위한
커뮤니티 및 자료 공유 기능

사용자의 질문에 답변하는
인공지능 챗봇

시험 일정을 저장하고 디
데이를 알려주는 기능

기대 효과

학습 효율성 향상

누구나 쉽게 접근 가능한
학습 환경

사용자 만족도 증대

기본 구조

학생

소셜 로그인

강의 결제

시험 일정 알림

강의 스트리밍

실시간 채팅

강사

강의 관리

Q&A 관리

자료 업로드

관리자

사용자 관리

커뮤니티 관리

개발 환경

FRONT

HTML

CSS

jQuery

JS

B

Thymeleaf

AJAX

BACK

Java

spring boot

개발 환경

DB

MySQL

TOOL

MyBatis

git

GitHub

Fliki

IDE

Spring Tools 4

개발 환경

API

websockets

Dialogflow

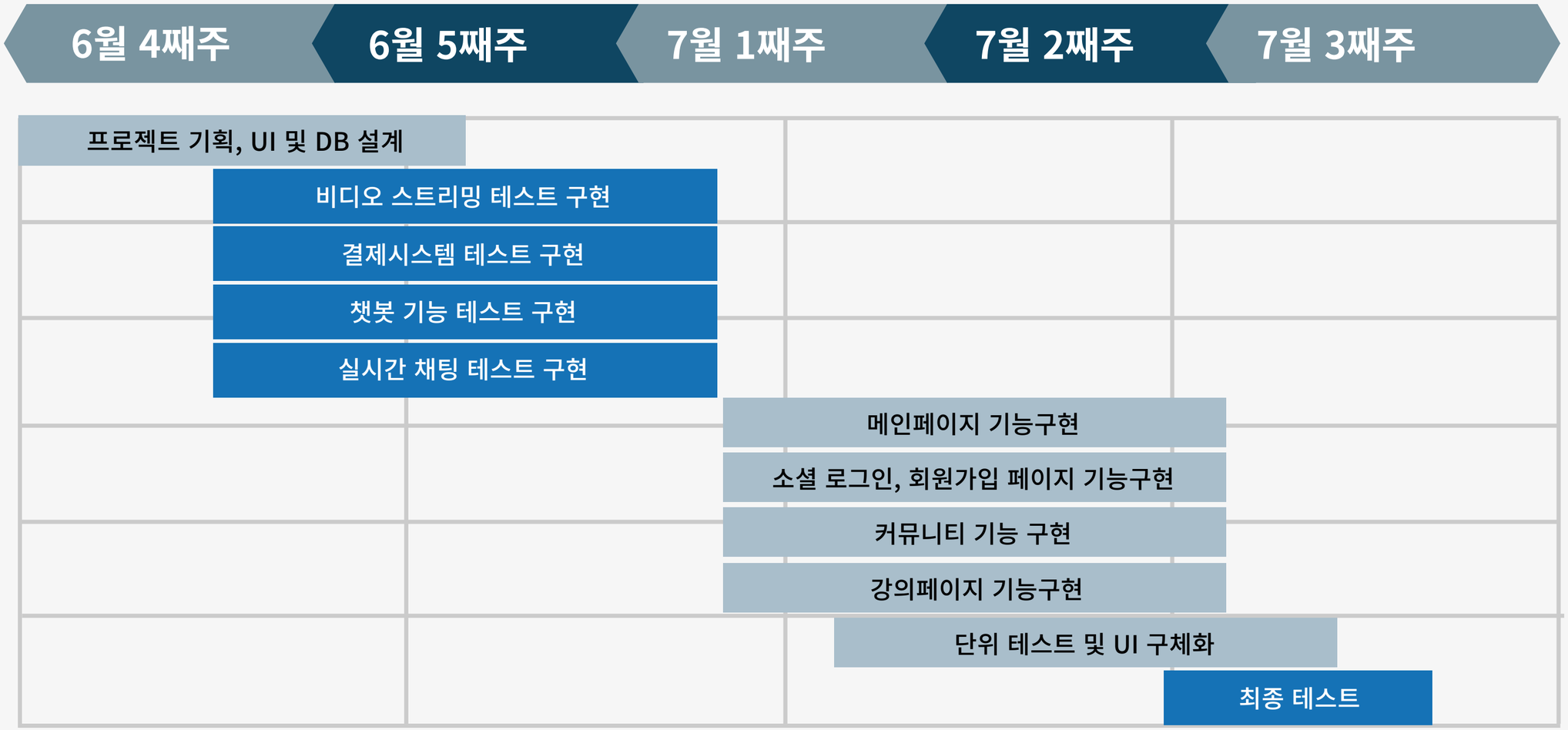
aws | S3

pay

NGINX

프로젝트 타임라인

오전수업이후 프로젝트 진행



팀원 소개

이형철

- 초기 세팅
- GIT 브랜치및 버전 관리
- PPT 제작
- 헤더 및 푸터 제작
- 소셜네트워크 로그인 및 회원가입
- 강의 검색
- 강의 평가 및 리뷰 작성
- 강의 노트 기능
- 실시간 스트리밍 기능
- 동영상 마지막 시청 시간 저장 및 이어서 보기
- 동영상 배속 재생
- 강의 콘텐츠 플레이어 개발 및 관리
- 수료증 발급
- 강의 Q&A
- 강의 정보, 설명
- 강의 신청
- 내 강의 페이지 기능

고원

- 카카오페이 연동 강의 결제 기능
- 포인트 결제 기능
- 사용자 관리 기능
- 사용자 인증 및 권한 관리
- 학생 마이페이지
- 강사 마이페이지
- 강의 관리 시스템
- 강의 수정 및 삭제
- 강의 영상 제작
- 발표

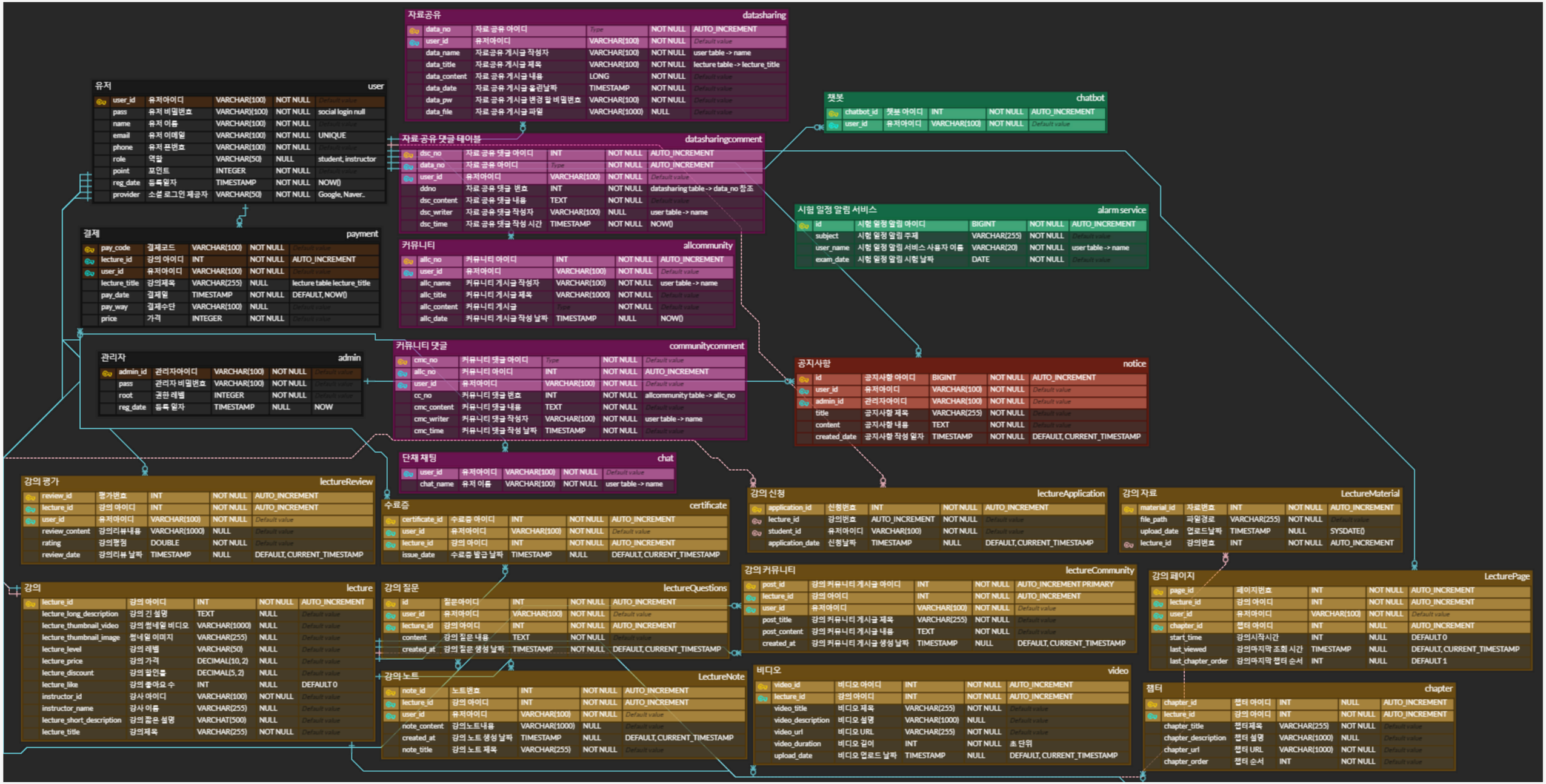
김동욱

- 메인 페이지 디자인
- 메인 페이지 콘텐츠 배치
- 챗봇 기본 설정
- 챗봇 대화 시나리오 작성
- 일정 알림 설정
- 일정 알림 관리 페이지
- 헤더 네비게이션 요소 추가
- 광고 영상 제작
- 발표

김희선

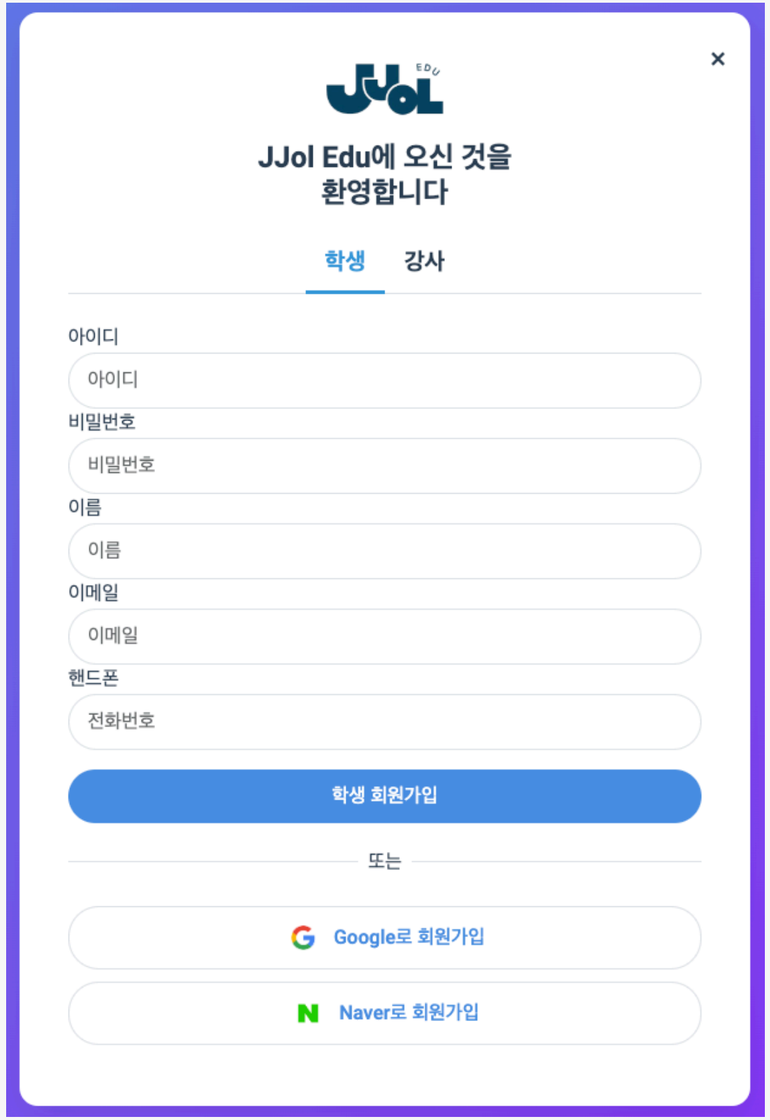
- 스터디 그룹 커뮤니티 게시판
- 실시간 단체 채팅
- 자료 업로드 및 다운로드
- 자료 분류 및 태그 기능
- 자료 게시판 검색 기능
- 자료 게시판 수정 기능
- 자료 게시판 삭제 기능
- PPT 제작
- 강의 영상 제작

ERD


자료공유 datasharing
data_no 자료 공유 아이디 Type NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
data_name 자료공유 게시글 작성자 VARCHAR(100) NOT NULL user table -> name
data_title 자료공유 게시글 제목 VARCHAR(100) NOT NULL lecture table -> lecture_title
data_content 자료 공유 게시글 내용 LONG NOT NULL Default value
data_date 자료 공유 게시글 올린날짜 TIMESTAMP NOT NULL Default value
data_pw 자료 공유 게시글 변경 할 비밀번호 VARCHAR(100) NOT NULL Default value
data_file 자료 공유 게시글 파일 VARCHAR(1000) NULL Default value
유저 user
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
pass 유저 비밀번호 VARCHAR(100) NOT NULL social login null
name 유저 이름 VARCHAR(100) NOT NULL Default value
email 유저 이메일 VARCHAR(100) NOT NULL UNIQUE
phone 유저 폰번호 VARCHAR(100) NOT NULL Default value
role 역할 VARCHAR(50) NULL student, instructor
point 포인트 INTEGER NOT NULL Default value
reg_date 등록일자 TIMESTAMP NOT NULL NOW()
provider 소셜 로그인 제공자 VARCHAR(50) NOT NULL Google, Naver..
챗봇 chatbot
chatbot_id 챗봇 아이디 INT NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
자료 공유 댓글 테이블 datasharingcomment
dsc_no 자료 공유 댓글 아이디 INT NOT NULL AUTO_INCREMENT
data_no 자료 공유 아이디 Type NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
ddno 자료 공유 댓글 번호 INT NOT NULL datasharing table -> data_no 참조
dsc_content 자료 공유 댓글 내용 TEXT NOT NULL Default value
dsc_writer 자료 공유 댓글 작성자 VARCHAR(100) NULL user table -> name
dsc_time 자료 공유 댓글 작성 시간 TIMESTAMP NOT NULL NOW()
시험 일정 알림 서비스 alarm service
id 시험 일정 알림 아이디 BIGINT NOT NULL AUTO_INCREMENT
subject 시험 일정 알림 주제 VARCHAR(255) NOT NULL Default value
user_name 시험 일정 알림 서비스 사용자 이름 VARCHAR(20) NOT NULL user table -> name
exam_date 시험 일정 알림 시험 날짜 DATE NOT NULL Default value
결제 payment
pay_code 결제코드 VARCHAR(100) NOT NULL Default value
lecture_id 강의 아이디 INT NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
lecture_title 강의제목 VARCHAR(255) NULL lecture table lecture_title
pay_date 결제일 TIMESTAMP NOT NULL DEFAULT, NOW()
pay_way 결제수단 VARCHAR(100) NULL Default value
price 가격 INTEGER NOT NULL Default value
커뮤니티 allcommunity
allc_no 커뮤니티 아이디 INT NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
allc_name 커뮤니티 게시글 작성자 VARCHAR(100) NOT NULL user table -> name
allc_title 커뮤니티 게시글 제목 VARCHAR(1000) NOT NULL Default value
allc_content 커뮤니티 게시글 Type NOT NULL Default value
allc_date 커뮤니티 게시글 작성 날짜 TIMESTAMP NULL NOW()
관리자 admin
admin_id 관리자아이디 VARCHAR(100) NOT NULL Default value
pass 관리자 비밀번호 VARCHAR(100) NOT NULL Default value
root 권한 레벨 INTEGER NOT NULL Default value
reg_date 등록 일자 TIMESTAMP NULL NOW
커뮤니티 댓글 communitycomment
cmc_no 커뮤니티 댓글 아이디 Type NOT NULL Default value
allc_no 커뮤니티 아이디 INT NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
cc_no 커뮤니티 댓글 번호 INT NOT NULL allcommunity table -> allc_no
cmc_content 커뮤니티 댓글 내용 TEXT NOT NULL Default value
cmc_writer 커뮤니티 댓글 작성자 VARCHAR(100) NOT NULL user table -> name
cmc_time 커뮤니티 댓글 작성 날짜 TIMESTAMP NOT NULL Default value
공지사항 notice
id 공지사항 아이디 BIGINT NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
admin_id 관리자아이디 VARCHAR(100) NOT NULL Default value
title 공지사항 제목 VARCHAR(255) NOT NULL Default value
content 공지사항 내용 TEXT NOT NULL Default value
created_date 공지사항 작성 일자 TIMESTAMP NOT NULL DEFAULT, CURRENT_TIMESTAMP
단체 채팅 chat
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
chat_name 유저 이름 VARCHAR(100) NOT NULL user table -> name
강의 평가 lectureReview
review_id 평가번호 INT NOT NULL AUTO_INCREMENT
lecture_id 강의 아이디 INT NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
review_content 강의리뷰내용 VARCHAR(1000) NULL Default value
rating 강의평점 DOUBLE NOT NULL Default value
review_date 강의리뷰 날짜 TIMESTAMP NULL DEFAULT, CURRENT_TIMESTAMP
수료증 certificate
certificate_id 수료증 아이디 INT NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
lecture_id 강의 아이디 INT NOT NULL AUTO_INCREMENT
issue_date 수료증 발급 날짜 TIMESTAMP NULL DEFAULT, CURRENT_TIMESTAMP
강의 신청 lectureApplication
application_id 신청번호 INT NOT NULL AUTO_INCREMENT
lecture_id 강의번호 AUTO_INCREMENT NOT NULL Default value
student_id 유저아이디 VARCHAR(100) NOT NULL Default value
application_date 신청날짜 TIMESTAMP NULL DEFAULT, CURRENT_TIMESTAMP
강의 자료 LectureMaterial
material_id 자료번호 INT NOT NULL AUTO_INCREMENT
file_path 파일경로 VARCHAR(255) NOT NULL Default value
upload_date 업로드날짜 TIMESTAMP NULL SYSDATE()
lecture_id 강의번호 INT NOT NULL AUTO_INCREMENT
강의 lecture
lecture_id 강의 아이디 INT NOT NULL AUTO_INCREMENT
lecture_long_description 강의 긴 설명 TEXT NULL Default value
lecture_thumbnail_video 강의 썸네일 비디오 VARCHAR(1000) NULL Default value
lecture_thumbnail_image 썸네일 이미지 VARCHAR(255) NULL Default value
lecture_level 강의 레벨 VARCHAR(50) NULL Default value
lecture_price 강의 가격 DECIMAL(10, 2) NULL Default value
lecture_discount 강의 할인률 DECIMAL(5, 2) NULL Default value
lecture_like 강의 좋아요 수 INT NULL DEFAULT 0
instructor_id 강사 아이디 VARCHAR(100) NOT NULL Default value
instructor_name 강사 이름 VARCHAR(255) NULL Default value
lecture_short_description 강의 짧은 설명 VARCHAT(500) NULL Default value
lecture_title 강의제목 VARCHAR(255) NOT NULL Default value
강의 질문 lectureQuestions
id 질문아이디 INT NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
lecture_id 강의 아이디 INT NOT NULL AUTO_INCREMENT
content 강의 질문 내용 TEXT NOT NULL Default value
created_at 강의 질문 생성 날짜 TIMESTAMP NOT NULL DEFAULT, CURRENT_TIMESTAMP
강의 커뮤니티 lectureCommunity
post_id 강의 커뮤니티 게시글 아이디 INT NOT NULL AUTO_INCREMENT PRIMARY
lecture_id 강의 아이디 INT NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
post_title 강의 커뮤니티 게시글 제목 VARCHAR(255) NOT NULL Default value
post_content 강의 커뮤니티 게시글 내용 TEXT NOT NULL Default value
created_at 강의 커뮤니티 게시글 생성 날짜 TIMESTAMP NULL DEFAULT, CURRENT_TIMESTAMP
강의 페이지 LecturePage
page_id 페이지번호 INT NOT NULL AUTO_INCREMENT
lecture_id 강의 아이디 INT NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
chapter_id 챕터 아이디 INT NULL AUTO_INCREMENT
start_time 강의시작시간 INT NULL DEFAULT 0
last_viewed 강의마지막 조회 시간 TIMESTAMP NULL DEFAULT, CURRENT_TIMESTAMP
last_chapter_order 강의마지막 챕터 순서 INT NULL DEFAULT 1
강의 노트 LectureNote
note_id 노트번호 INT NOT NULL AUTO_INCREMENT
lecture_id 강의 아이디 INT NOT NULL AUTO_INCREMENT
user_id 유저아이디 VARCHAR(100) NOT NULL Default value
note_content 강의노트내용 VARCHAR(1000) NULL Default value
created_at 강의 노트 생성 날짜 TIMESTAMP NULL DEFAULT, CURRENT_TIMESTAMP
note_title 강의 노트 제목 VARCHAR(255) NOT NULL Default value
비디오 video
video_id 비디오 아이디 INT NOT NULL AUTO_INCREMENT
lecture_id 강의 아이디 INT NOT NULL AUTO_INCREMENT
video_title 비디오 제목 VARCHAR(255) NOT NULL Default value
video_description 비디오 설명 VARCHAR(1000) NULL Default value
video_url 비디오 URL VARCHAR(255) NOT NULL Default value
video_duration 비디오 길이 INT NOT NULL 초 단위
upload_date 비디오 업로드 날짜 TIMESTAMP NULL DEFAULT, CURRENT_TIMESTAMP
챕터 chapter
chapter_id 챕터 아이디 INT NULL AUTO_INCREMENT
lecture_id 강의 아이디 INT NOT NULL AUTO_INCREMENT
chapter_title 챕터제목 VARCHAR(255) NOT NULL Default value
chapter_description 챕터 설명 VARCHAR(1000) NULL Default value
chapter_url 챕터 URL VARCHAR(1000) NOT NULL Default value
chapter_order 챕터 순서 INT NOT NULL Default value

페이지 설명

소셜네트워크 로그인, 회원가입



사용 기술

OAuth2

OAuth2는 자원 소유자가 타사 애플리케이션에 자신이 소유한 리소스에 접근할 수 있도록 권한을 부여하는 프로토콜

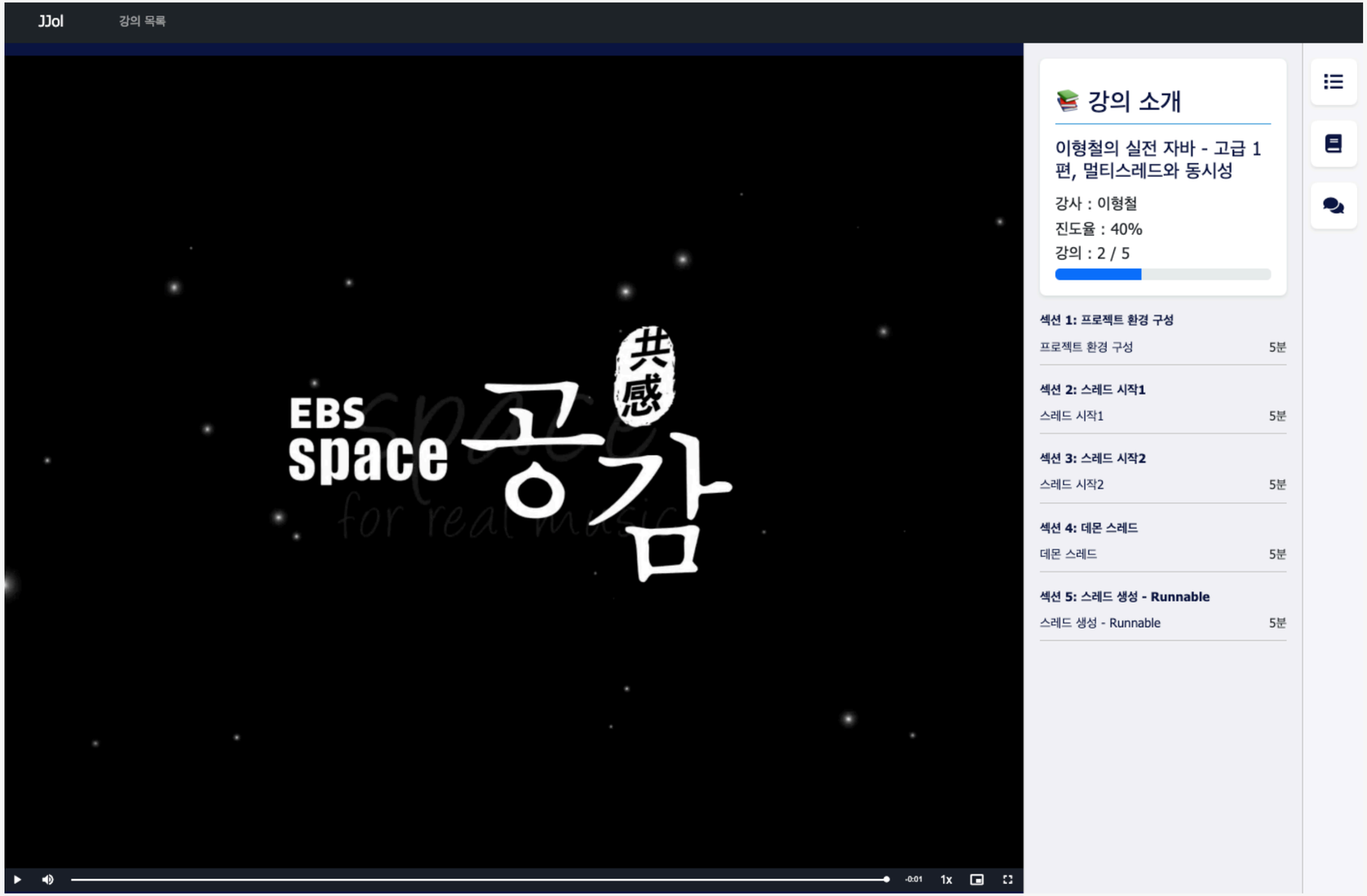
```java
package com.example.project_jjol.service;

import com.example.project_jjol.model.User;
import jakarta.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.authority.SimpleGrantedAuthority;
import
org.springframework.security.oauth2.client.userinfo.DefaultOAuth2UserService;
import org.springframework.security.oauth2.client.userinfo.OAuth2UserRequest;
import org.springframework.security.oauth2.client.userinfo.OAuth2UserService;
import org.springframework.security.oauth2.core.OAuth2AuthenticationException;
import org.springframework.security.oauth2.core.user.DefaultOAuth2User;
import org.springframework.security.oauth2.core.user.OAuth2User;
import org.springframework.stereotype.Service;

import java.util.Collections;
import java.util.Map;

@Service
public class CustomOAuth2UserService implements
OAuth2UserService<OAuth2UserRequest, OAuth2User> {

    private final UserService userService;
    private final HttpSession session;

    @Autowired
    public CustomOAuth2UserService(UserService userService, HttpSession session)
{
        this.userService = userService;
        this.session = session;
    }
```

사용 기술

1. 프로젝트 생성
2. OAuth2 설정
3. google-auth-httplib2 라이브러리 설치
4. 로그인 URL 생성
5. 토큰 교환 및 사용자 정보 가져오기

```
        } else if ("kakao".equals(provider)) {
            Map<String, Object> kakaoAccount = (Map<String, Object>)
oAuth2User.getAttributes().get("kakao_account");
            if (kakaoAccount != null) {
                email = (String) kakaoAccount.get("email");
                Map<String, Object> profile = (Map<String, Object>)
kakaoAccount.get("profile");
                if (profile != null) {
                    name = (String) profile.get("nickname");
                }
            }
        } else {
            email = oAuth2User.getAttribute("email");
            name = oAuth2User.getAttribute("name");
        }

        if (email == null) {
            throw new OAuth2AuthenticationException("이메일을 가져올 수 없습니다.");
        }
```

동영상 스트리밍


JJol
강의 목록
EBS space 공감 共感
for real music
-0:01
1x
강의 소개
이형철의 실전 자바 - 고급 1편, 멀티스레드와 동시성
강사 : 이형철
진도율 : 40%
강의 : 2 / 5
섹션 1: 프로젝트 환경 구성
프로젝트 환경 구성 5분
섹션 2: 스레드 시작1
스레드 시작1 5분
섹션 3: 스레드 시작2
스레드 시작2 5분
섹션 4: 데몬 스레드
데몬 스레드 5분
섹션 5: 스레드 생성 - Runnable
스레드 생성 - Runnable 5분

사용 기술

NGINX

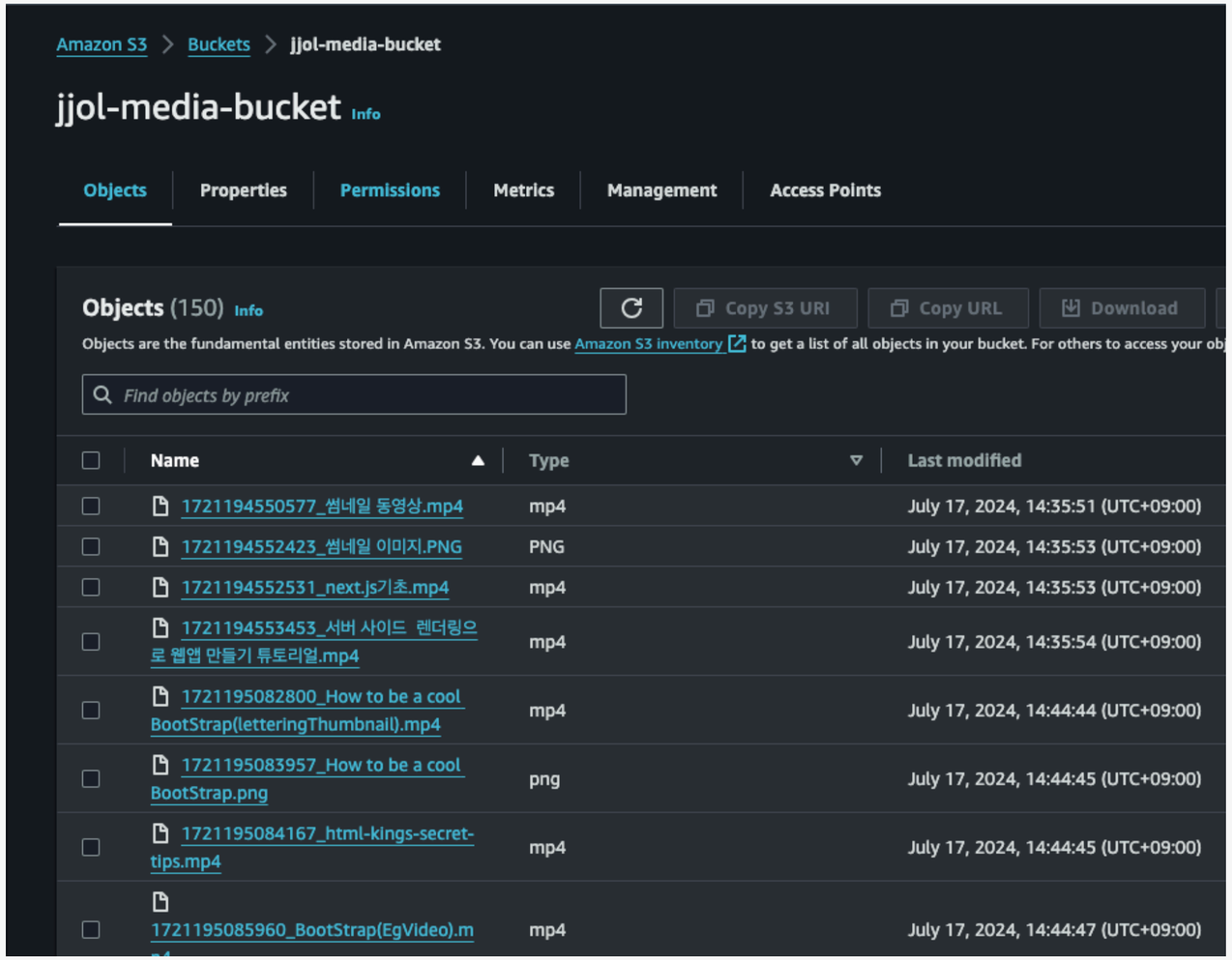
nginx는 RTMP 모듈을 통해 실시간 동영상 스트리밍 서버를 구축 및 라이브 방송, VOD 스트리밍 지원을 한다.

```java
@GetMapping("/videos/player")
    public String videoPlayer(@RequestParam("lectureId") int lectureId,
HttpSession session, Model model) {
        Object loggedInUser = session.getAttribute("loggedInUser");
        if (loggedInUser == null) {
            return "redirect:/login";
        }

        User user = (User) loggedInUser;
        List<Chapter> chapters =
lectureService.getChaptersByLectureId(lectureId);
        Lecture lecture = lectureService.getLectureById(lectureId);
        Chapter lastViewedChapter =
lectureService.getLastViewedChapter(lectureId, user.getUserId());

        if (!chapters.isEmpty()) {
            model.addAttribute("firstChapterUrl", lastViewedChapter != null ?
lastViewedChapter.getChapterUrl() : chapters.get(0).getChapterUrl());
        } else {
            model.addAttribute("firstChapterUrl",
lecture.getLectureThumbnailVideo());
        }

        model.addAttribute("chapters", chapters);
        model.addAttribute("lecture", lecture);
        return "views/videoPlayer"; // JSON 응답을 보내지 않고 HTML 템플릿을 렌더링
    }
```

사용 기술

Amazon S3

Amazon S3 에서 제공하는
객체 스토리지 서비스
간단하게 말해, 데이터를 클라우드에
안전하게 저장하고, 언제 어디서나
접근할 수 있게 해준다.



NGINX 와 AMAZON S3

강의 업로드

클라이언트 디바이스 → Nginx서버 (RTMP) → 스트리밍처리 → Amazon S3 (저장소)

강의 스트리밍

Amazon S3 (저장소) → Nginx서버 (RTMP) → 미디어 콘텐츠 접근

```
2024/06/20 17:22:35 [notice] 31988#0: signal process started
2024/06/20 17:45:16 [notice] 32347#0: signal process started
2024/06/20 17:52:15 [emerg] 32508#0: "types" directive is not al
/usr/local/nginx/conf/nginx.conf:127
2024/06/20 17:54:06 [warn] 32538#0: duplicate extension "mp4", c
"video/mp4", previous content type: "video/mp4" in
/usr/local/nginx/conf/nginx.conf:22
2024/06/20 17:54:06 [warn] 32538#0: duplicate extension "mov", c
"video/quicktime", previous content type: "video/quicktime" in
/usr/local/nginx/conf/nginx.conf:23
2024/06/20 17:54:06 [notice] 32538#0: signal process started
2024/06/21 13:10:25 [notice] 9330#0: signal process started
2024/06/21 13:10:25 [error] 9330#0: open() "/usr/local/nginx/log
(2: No such file or directory)
2024/06/21 13:10:52 [notice] 9345#0: signal process started
2024/06/21 13:10:52 [error] 9345#0: open() "/usr/local/nginx/log
(2: No such file or directory)
```

NGINX

1. Nginx 설치
2. RTMP 모듈 설정
3. Nginx 시작
4. 스프링부트 준비
5. OBS 설정
6. 스트리밍 확인

```
// 비디오 플레이어를 초기화하는 함수
    async function initializePlayer() {
        if (!player) {
            player = videojs('my-video', {
                controls: true,
                autoplay: false,
                preload: 'auto'
            });

            const { lastChapterOrder, startTime } = await
fetchLastViewedTimeAndChapter(lectureId);

            console.log(lastChapterOrder);

            if (startTime > 0) {
                const resumeFromLastTime = confirm("마지막 시청 지점부터 재생하시겠습니
까?");
                if (resumeFromLastTime) {
                    currentChapterOrder = lastChapterOrder;
                    loadChapter(currentChapterOrder, startTime);
                } else {
                    await resetLectureProgress(lectureId);
                    loadChapter(1, 0);
                }
            } else {
```

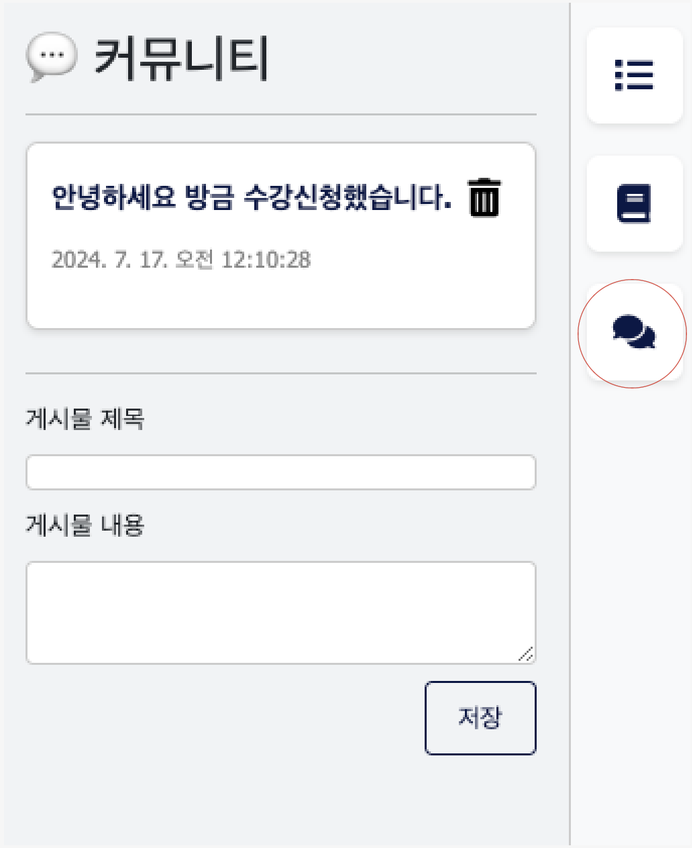
AMAZON S3

1. AWS 계정 생성 및 로그인
2. S3 버킷 생성
3. 버킷 정책 설정
4. 파일 업로드
5. 파일 URL 확인
6. 데이터 액세스 설정

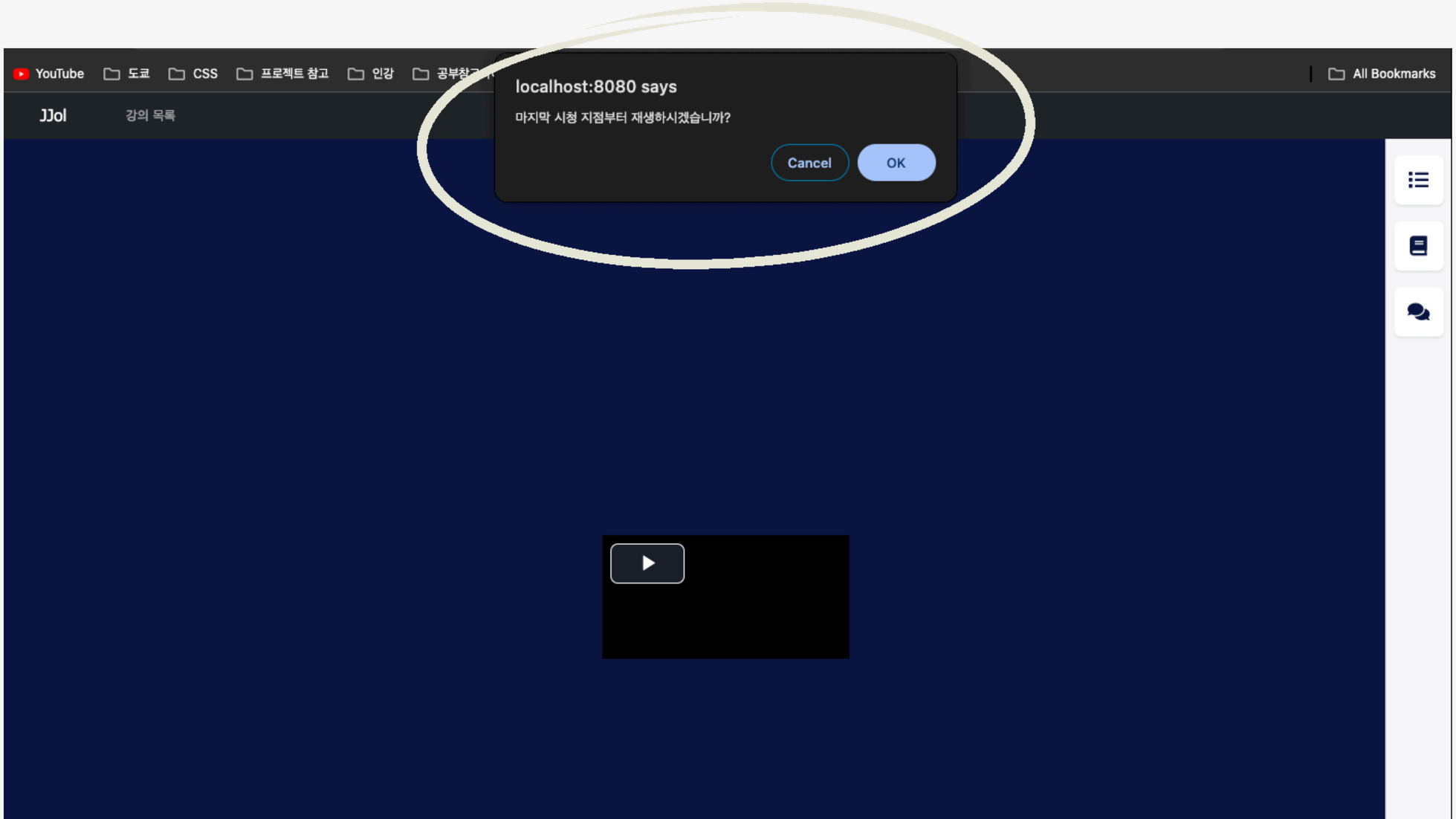
```java
package com.example.project_jjol.config;

@Configuration
public class S3Config {

    @Value("${amazon.aws.accesskey}")
    private String accessKey;

    @Value("${amazon.aws.secretkey}")
    private String secretKey;

    @Value("${amazon.s3.bucket}")
    private String bucketName;

    @Bean
    public S3Client s3Client() {
        AwsBasicCredentials awsCreds = AwsBasicCredentials.create(accessKey,
secretKey);
        return S3Client.builder()
                .region(Region.AP_NORTHEAST_2)  // 리전을 ap-northeast-2로 설정
                .credentialsProvider(StaticCredentialsProvider.create(awsCreds))
                .build();
    }

    public String getBucketName() {
        return bucketName;
```

강의노트, 강의커뮤니티

학습 노트
스프링부트 개요
스프링부트는 자바 기반의 간편한 웹 애플리케이션 개발 프레임워크입니다. 설정이 간소화되어 개발자가 빠르게 시작할 수 있고, 내장형 서버를 제공하여 배포가 간편합니다.
2024. 7. 17. 오전 12:08:23
노트 제목
노트 작성
저장


커뮤니티
안녕하세요 방금 수강신청했습니다.
2024. 7. 17. 오전 12:10:28
게시물 제목
게시물 내용
저장

동영상 마지막 시청 시간 저장 및 이어서 보기



사용 기술

1. 비디오 플레이어 요소 가져오기
2. 마지막 재생 지점을 저장할 변수 초기화
3. timeupdate 이벤트 리스너 등록
4. 현재 재생 중인 비디오의 시간 추적
5. 페이지 이탈 시 마지막 재생 지점을 서버에 전송

```
// 특정 챕터를 로드하고 시작 시간을 설정하는 함수
    async function loadChapter(chapterOrder, startTime) {
        const chapterElement = document.querySelector(`.lecture[data-chapter
order="${chapterOrder}"]`);
        if (!chapterElement) {
            console.error(`Chapter element with order ${chapterOrder} not
found`);
            return;
        }
        const videoUrl = chapterElement.getAttribute('data-url');
        player.src({ type: 'video/mp4', src: videoUrl });
        player.ready(function() {
            player.currentTime(startTime);
            player.play();
        });
    }
```

동영상 배속 재생, 강의 콘텐츠 플레이어 개발 및 관리

사용 기술

VIDEO.JS

**HTML5기반의 브라우저에서
다양한 미디어를 실행시킬 수 있는
비디오 플레이어 라이브러리**

VIDEO.JS

```
<!DOCTYPE html>
<html lang="ko" xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>JJOL EDU</title>

<link href="/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<script src="/bootstrap/js/bootstrap.bundle.min.js"></script>

<link
    href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css"
    rel="stylesheet">
<link href="/css/styles.css" rel="stylesheet">

<link href="//vjs.zencdn.net/7.10.2/video-js.min.css" rel="stylesheet">
<script src="//vjs.zencdn.net/7.10.2/video.min.js"></script>

<script src="/js/jquery-3.7.1.min.js"></script>

<script
    src="https://uicdn.toast.com/editor/latest/toastui-editor-all.min.js">
</script>
<link rel="stylesheet"
    href="https://uicdn.toast.com/editor/latest/toastui-editor.min.css" />
```

사용 기술

1. **HTML 파일에 Video.js 링크 추가****
2. **비디오 태그 추가**
3. **Video.js 초기화**
4. **비디오 옵션 설정**

```
<div class="video-section">
            <video id="my-video" class="vjs-matrix video-js" controls
                preload="auto"
                data-setup='{"playbackRates": [2, 1.5, 1.25, 1, 0.75, 0.5]}'
                autoplay>
                <source id="video-source" th:src="@{${firstChapterUrl}}"
                    type="video/mp4">
            </video>
        </div>
```

수료증

Certificate of Completion

This is to certify that

남궁혁

has successfully completed the course

서버 사이드 렌더링과 Next.js

on

2024-07-17 16:33:57

Instructor: 김희선

Director: JJol Education

JJol Online Learning Platform

jjol education

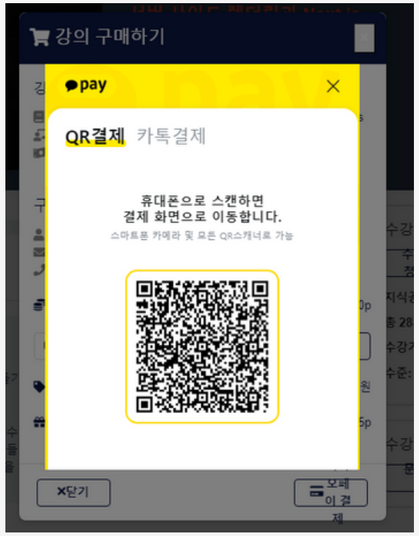
JJol

수료증 인쇄

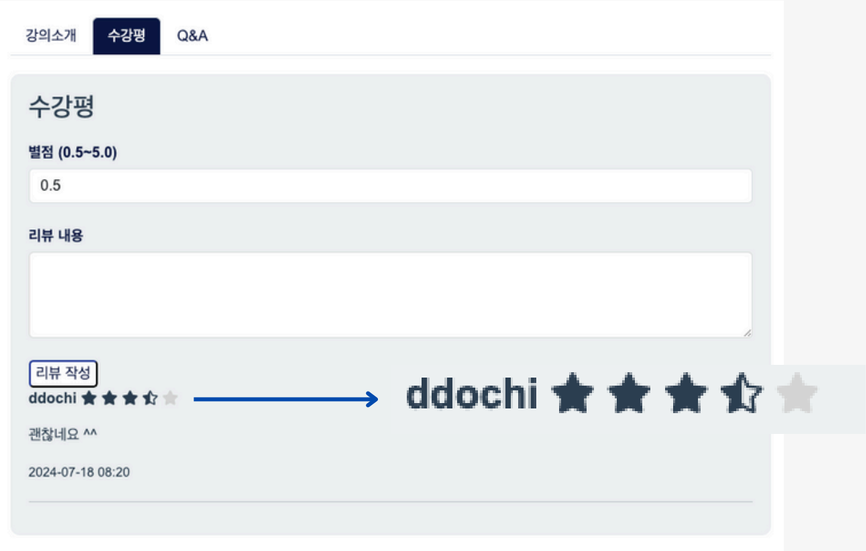
Jjol에 오신 걸 반갑게 환
까요?

수강신청

수강 신청
수강 신청하기
지식공유자: 김희선
수강기한: 무제한
수준: 하

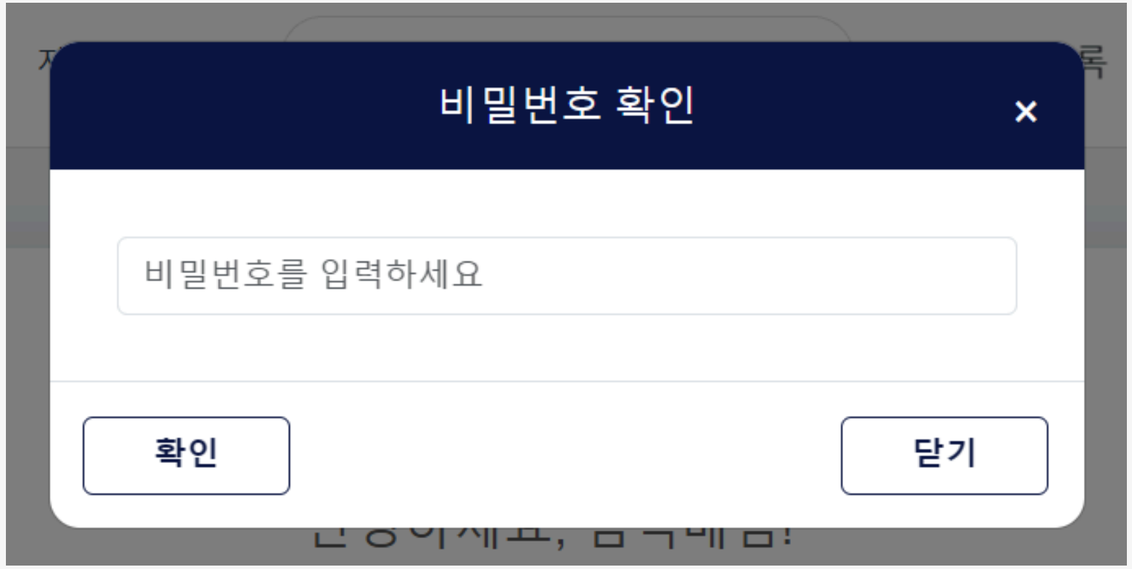

강의 구매하기
pay
QR결제 카톡결제
휴대폰으로 스캔하면
결제 화면으로 이동합니다.
스마트폰 카메라 및 모든 QR스캐너로 가능
×닫기
페이결제

학습중
학습하기
지식공유자: 김희선
수강기한: 무제한
수준: 하


강의소개
수강평
Q&A
수강평
별점 (0.5~5.0)
0.5
리뷰 내용
리뷰 작성
ddochi
괜찮네요 ^^
2024-07-18 08:20
ddochi

알아보겠습니다.
서버 사이드 렌더링과 Next.js
강사: 김희선
평점: 0 점 (0개의 수강평)
3명의 수강생
가격: 207100원 (할인 5.0% 적용)
공유하기
3명의 수강생

나의 강의 관리 - 수정

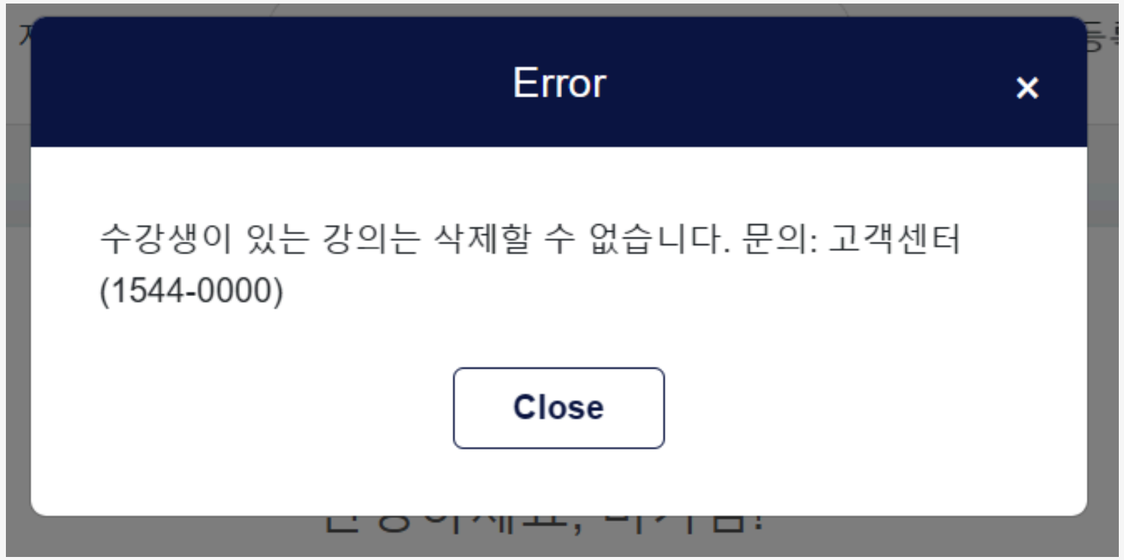



```
User loggedInUser = (User) session.getAttribute("loggedInUser");

if (password.equals(loggedInUser.getPass())) {
    Lecture lecture = myLecturesService.findLectureByLectureId(lectureId);
    List<Chapter> chapters =
            myLecturesService.findChaptersByLectureId(lectureId);

    model.addAttribute("lecture", lecture);
    model.addAttribute("loggedInUser", loggedInUser);
    model.addAttribute("chapters", chapters);
```

user 정보 get

일치
-> 정보 저장
-> return view

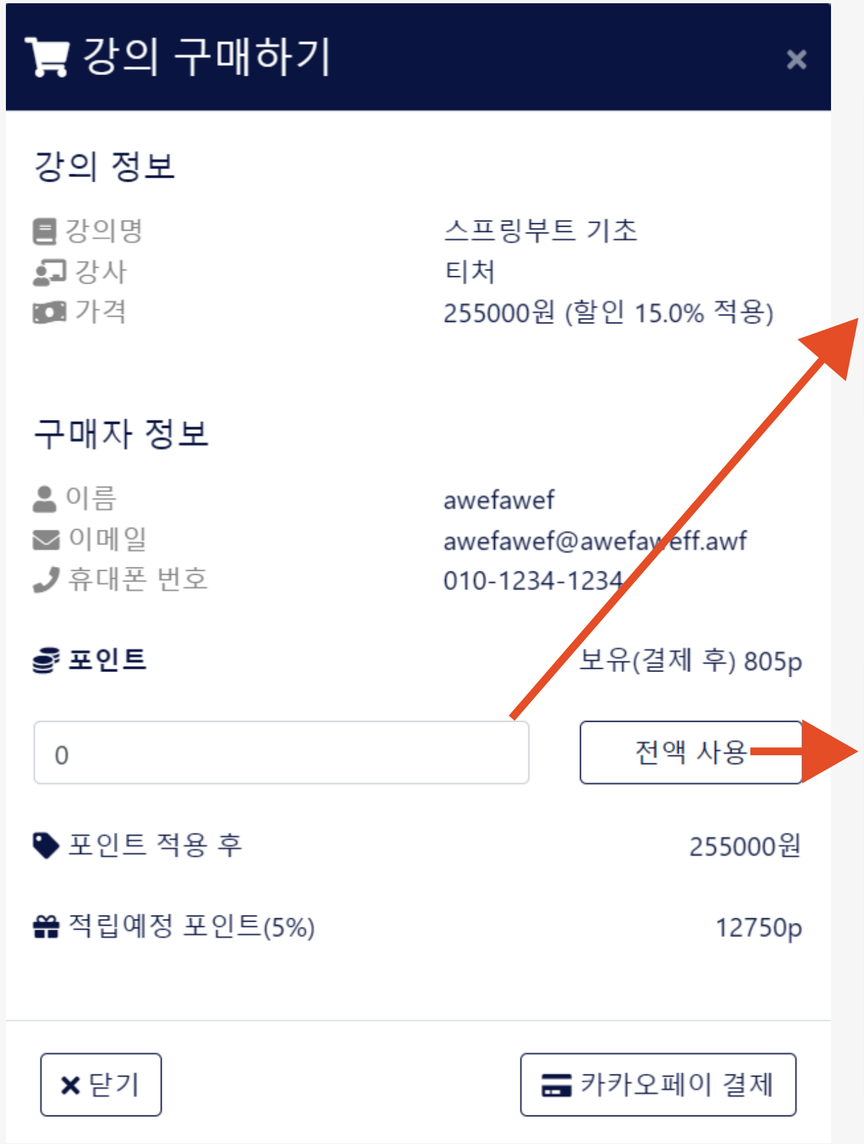
나의 강의 관리 - 삭제




```
            try {
                myLecturesService.deleteLecture(lectureId);
                redirectAttributes.addFlashAttribute(
                  "successMessage", "강의가 성공적으로 삭제되었습니다.");
            } catch(Exception e) {
                redirectAttributes.addFlashAttribute(
                  "errorMessage", "수강생이 있는 강의는 삭제할 수 없습니다. 문의: 고객센터
(1544-0000)");
                return "redirect:/myLectures";
            }
```

DataIntegrityViolationException

포인트 조정



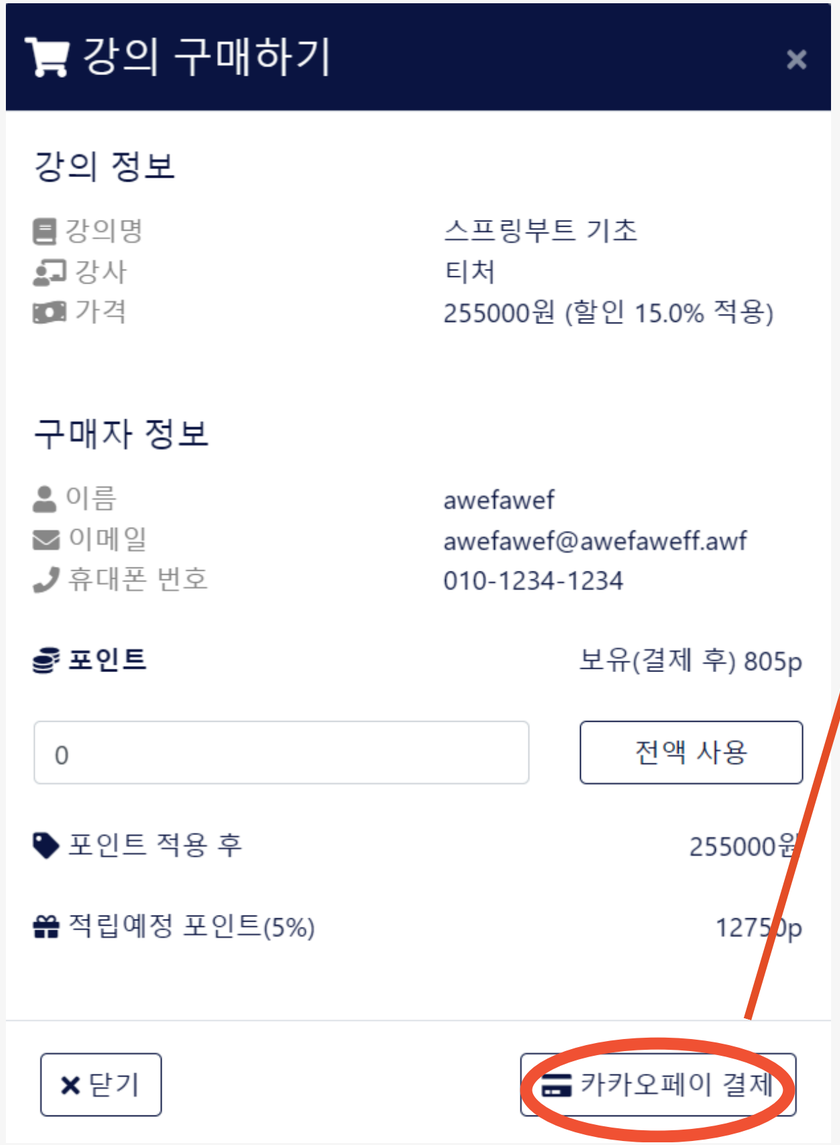
상한선 설정 : 가격과 보유 포인트 비교

```
$("#lectureDetailPointInput").on("keyup change", function() {
    if (lecturePrice > havePoint) {
        $("#lectureDetailPointInput").attr("max", havePoint);
    } else {
        $("#lectureDetailPointInput").attr("max", lecturePrice);
    }
```

전액 사용 버튼 : 보유 포인트와 사용 포인트 비교

```
    var usingPoint = parseInt($("#lectureDetailPointInput").val());

    if (usingPoint > havePoint) {
        $("#lectureDetailPointInput").val(havePoint);
    }
    if (usingPoint > lecturePrice) {
        $("#lectureDetailPointInput").val(lecturePrice);
    }
```

결제 시스템 - 검증




```
const dataForValidate = {
    userId: userId,
    lectureId: lectureId,
    lecturePrice: lecturePriceAfterPoint,
    usingPoint: usingPoint,
}
```

```
// 검증
await $.ajax({
    type: "POST",
    url: "/validatePayment",
    contentType: "application/json",
    data: JSON.stringify(dataForValidate),
    dataType: "json",
    success: function() {
        // 결제
        IMP.init("imp62227326");
        IMP.request_pay({
            pg: "kakaopay",
            pay_method: "kakaopay",
            merchant_uid: merchantUid,
            name: lectureTitle,
            amount: lecturePriceAfterPoint,
            buyer_name: userId,
            lecture_id: lectureId
        }, function (rsp) {
```

1. DATA -> 서버에 전송 -> 검증

2. 결제

결제 시스템 - 검증

```java
@RestController
public class PaymentController {
```

payment.js

```java
// ajax 데이터 추출
int lectureId = Integer.parseInt(requestData.get("lectureId").toString());
String userId = (String) requestData.get("userId");
int finalPriceFromView = Integer.parseInt(
        requestData.get("lecturePrice").toString());
int usingPoint = Integer.parseInt(
        requestData.get("usingPoint").toString());

// DB 데이터 추출
int discountedPriceFromDB = paymentService.getDiscountedPrice(lectureId);
int pointFromDB = paymentService.getUserPoint(userId);
```

```java
@Mapper
public interface PaymentMapper {

@Service
public class PaymentService {
```

```java
// 검증
if((finalPriceFromView != (discountedPriceFromDB - usingPoint))
        || ((pointFromDB - usingPoint) < 0)) {
```

응답 보냄

```java
// 응답 데이터
Map<String, Object> response = new HashMap<>();
response.put("message", "Data received successfully");
response.put("status", "success");
return ResponseEntity.ok(response);
```

결제 시스템 - DB 반영

2. 결제

서버 응답 데이터

3. DATA -> 서버에 전송
-> DB 반영

```
success: function() {

    // 결제
    IMP.init("imp62227326");
    IMP.request_pay({
        pg: "kakaopay",
        pay_method: "kakaopay",
        merchant_uid: merchantUid,
        name: lectureTitle,
        amount: lecturePriceAfterPoint,
        buyer_name: userId,
        lecture_id: lectureId
    }, function (rsp) {
        if(rsp.success) {
            // DB 반영
            $.ajax({
                type: "POST",
                url: "/addPayment",
                contentType: "application/json",
                data: JSON.stringify(dataForAddPayment),
                dataType: "json",
                success: function() {
                    console.log('결제 성공');
                    // 수강신청
                    $('#lectureDetailAppleyForm').submit();
                },
```

결제 시스템 - DB 반영

(payment.js → AJAX 추출)

```
@Mapper
public interface PaymentMapper {

@Service
public class PaymentService {
```

DB update →

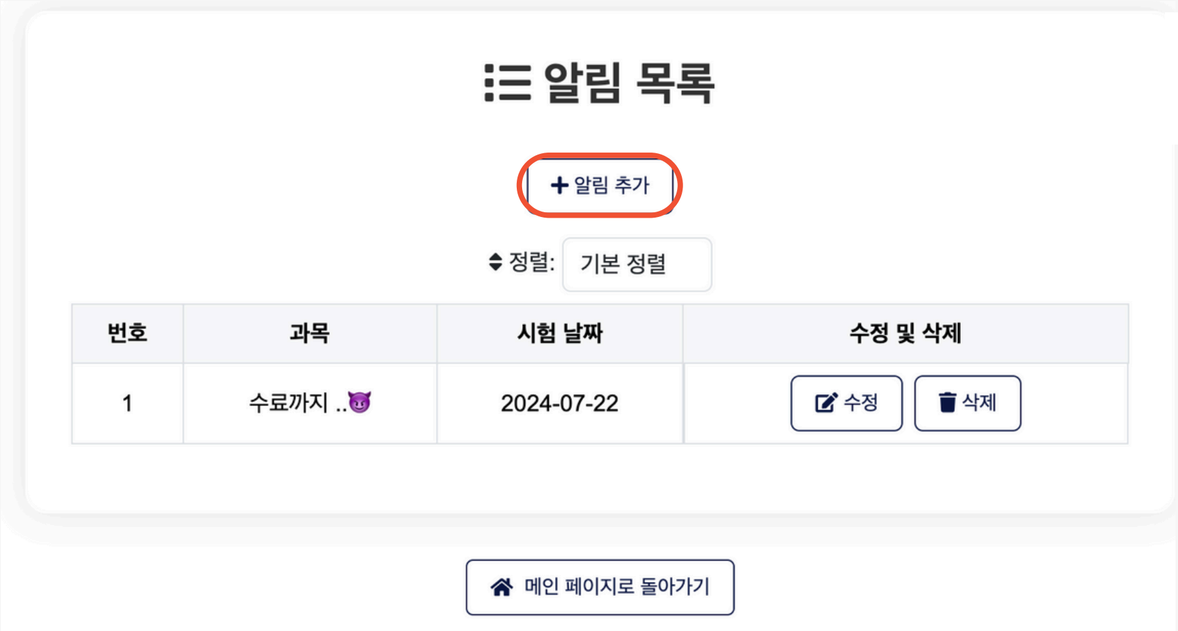
```
@RestController
public class PaymentController {
```

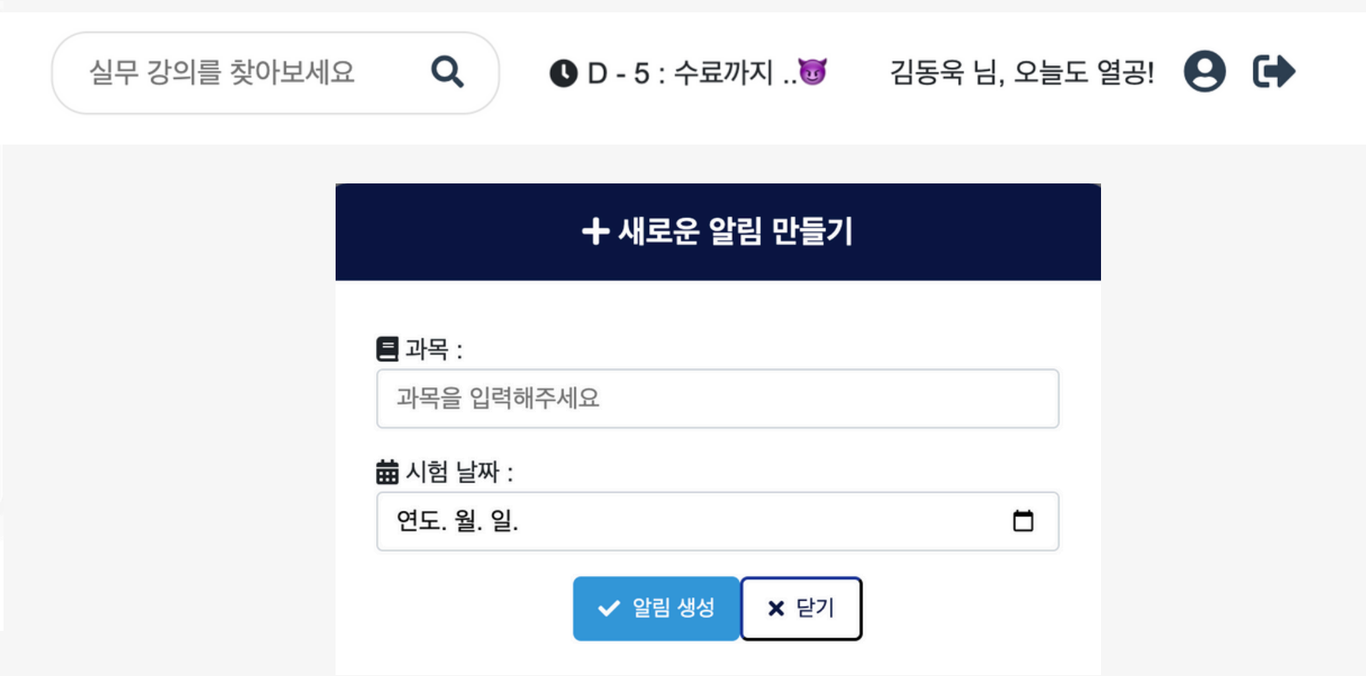
```
// DB 기록(Payment table)
Payment payment = new Payment();
payment.setLectureId(lectureId);
payment.setPrice(lecturePrice);
payment.setUserId(userId);
payment.setLectureTitle(lectureTitle);
payment.setMerchantUid(merchantUid);

paymentService.savePayment(payment);

// DB 기록(User table에서 포인트 업데이트)
int finalPoint = point + lecturePrice * 5 / 100;
User user = new User();
user.setUserId(userId);
user.setPoint(finalPoint);

paymentService.updatePoint(user);
```

응답 보냄

유저ID	강의제목	금액	결제일	결제ID
dlehrua	Intelligent Spring	96000원	2024-07-17 07:59:18	2
inst1	서버 사이드 렌더링과 Next.js	207100원	2024-07-17 08:01:51	3
studentking	vscode 내 몸 처럼 사용하기	85500원	2024-07-17 08:24:49	4
inst1	React와 Flask로 웹 애플리케이션 구현하기	203545원	2024-07-17 08:25:48	5
dlehrua	서버 사이드 렌더링과 Next.js	207100원	2024-07-17 08:46:58	6

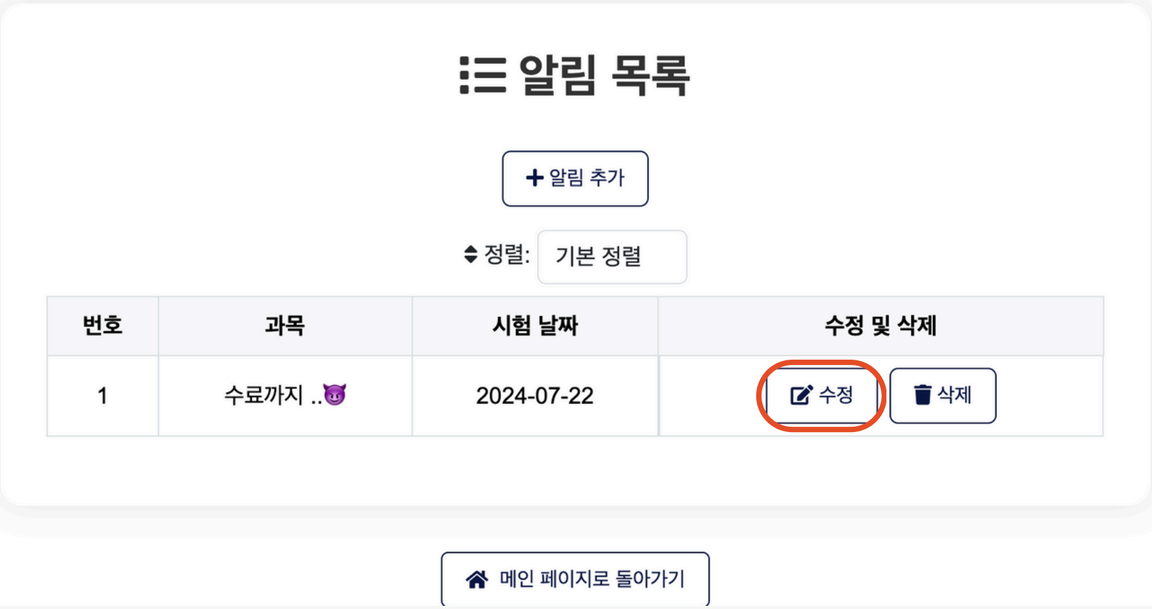
알림 시스템 - 추가

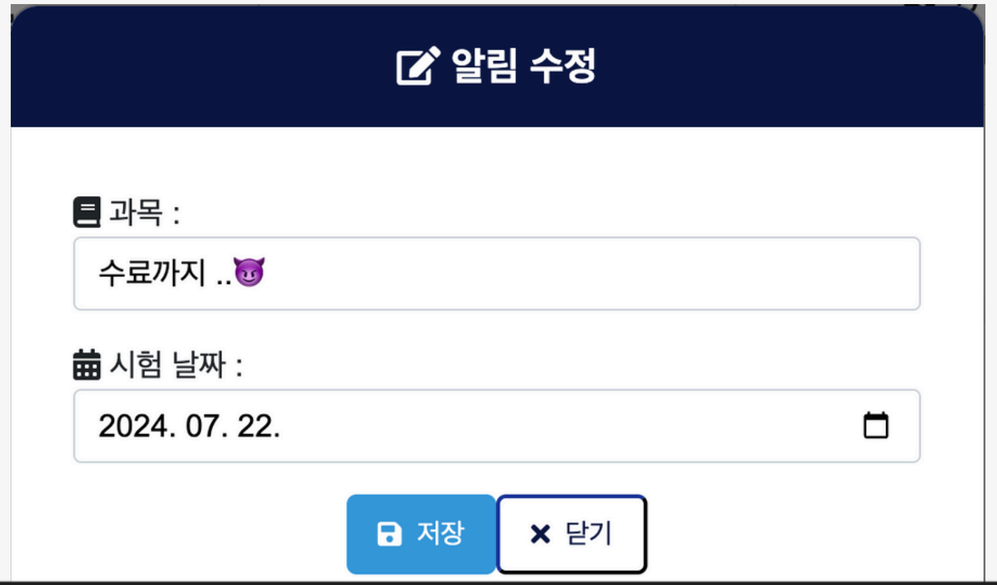





```
// 생성 모달창 열기
function notification_form_openCreateModal() {
    document.getElementById('notification_form_subject').value = '';
    document.getElementById('notification_form_examDate').value =
''; $('#notification_form_createModal').modal('show');
}
```

알림 시스템 - 수정



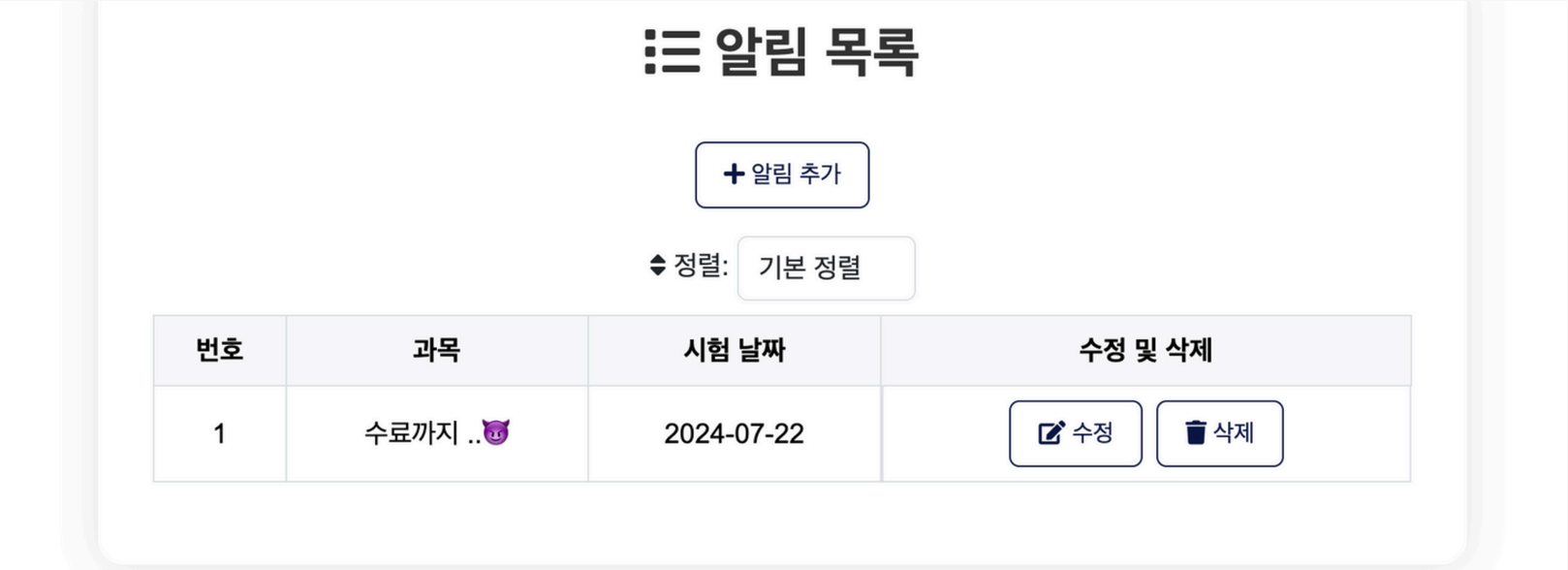



```java
// 여기서 수정 로직을 처리합니다.
    // id를 사용하여 DB에서 해당 알림을 가져와서 수정할 수도 있습니다.
    @PostMapping("/update")
    public String updateNotification(@RequestParam("id") Long id,
                                     @RequestParam("subject") String subject,
                                     @RequestParam("examDate") LocalDate examDate)
{       notificationService.updateNotification(id, subject, examDate);
        return "redirect:/notifications/form"; // 수정 후 폼 페이지로 리다이렉트
    }
```

```java
@Transactional
    public void updateNotification(Long id, String subject, LocalDate examDate) {
        // 1. id를 사용하여 기존의 알림을 찾습니다.
        Notification existingNotification = notificationMapper.findById(id);

        // 2. 기존 알림이 존재하는지 확인합니다.
        if (existingNotification == null) {
            throw new IllegalArgumentException("Notification with id " + id + " not
found");}

        // 3. 알림을 업데이트합니다.
        existingNotification.setSubject(subject);
        existingNotification.setExamDate(examDate);

        // 4. 매퍼를 사용하여 데이터베이스에 업데이트를 수행합니다.
        notificationMapper.updateNotification(existingNotification);
    }
```

원래 존재하고 있던 알림 내용이 있는지 확인하고 내용을 수정 한 후에 다시 MAPPER를 이용해 데이터베이스에 업데이트를 실행한다.

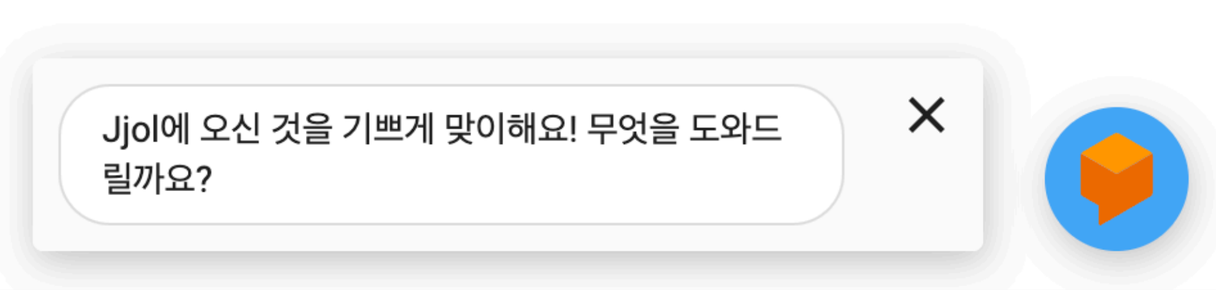
알림 시스템 - 삭제

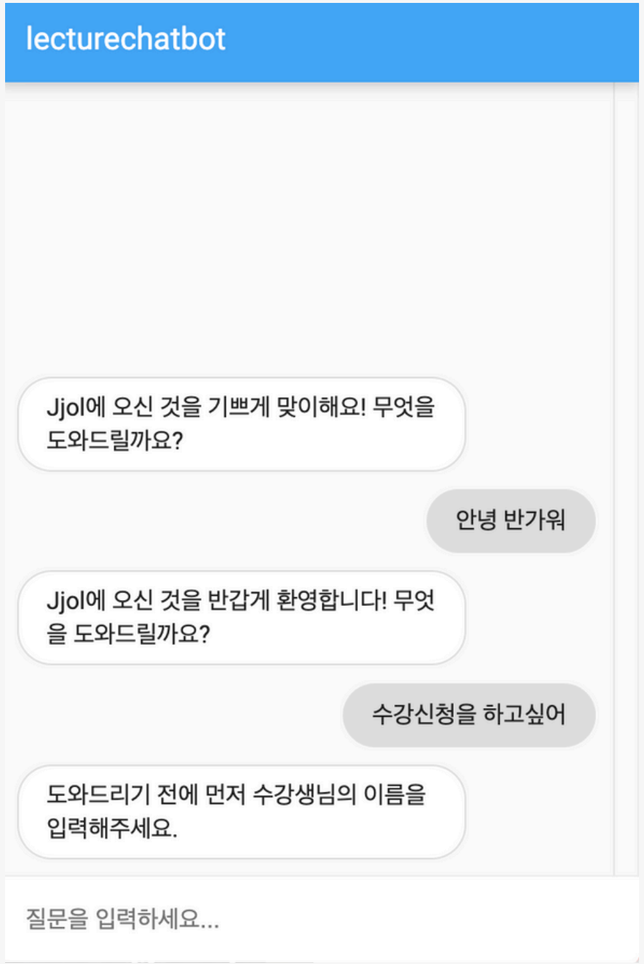



```
@PostMapping("/delete")
    public String deleteNotification(@RequestParam("id") Long id) {
        notificationService.deleteNotificationById(id);
        return "redirect:/notifications/form"; // 삭제 후 폼 페이지로
redirect
    }
```

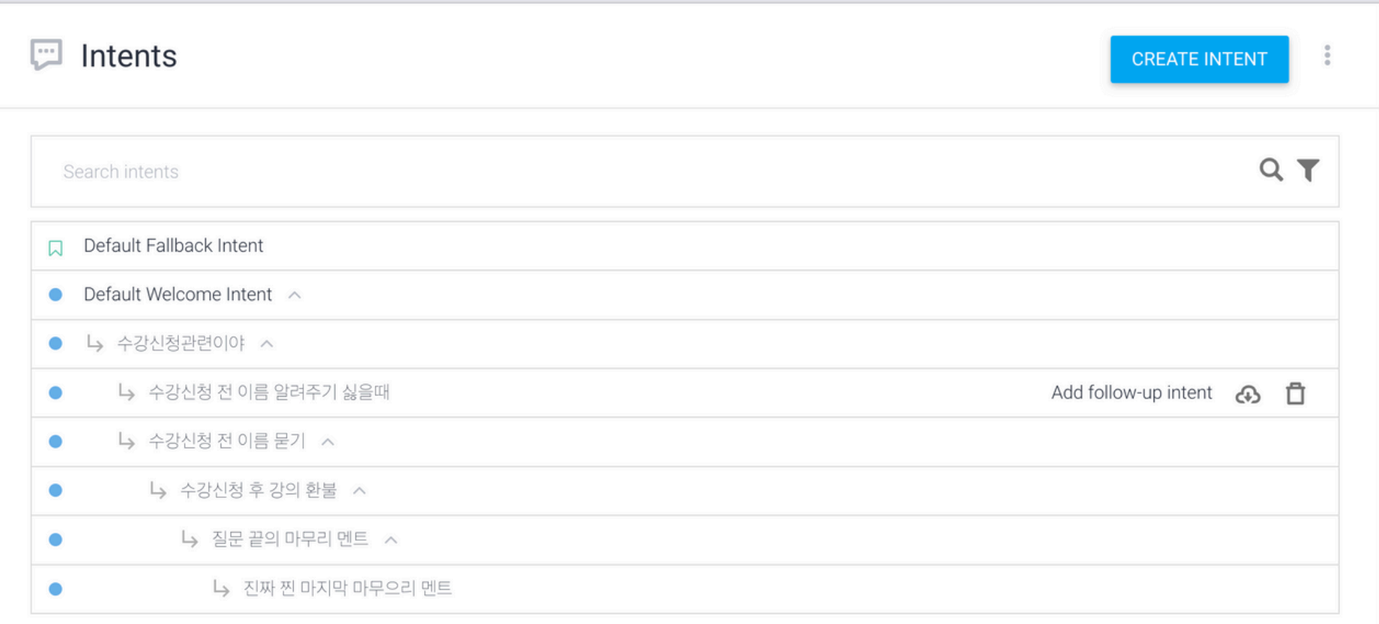
챗봇 (DIALOGFLOW)

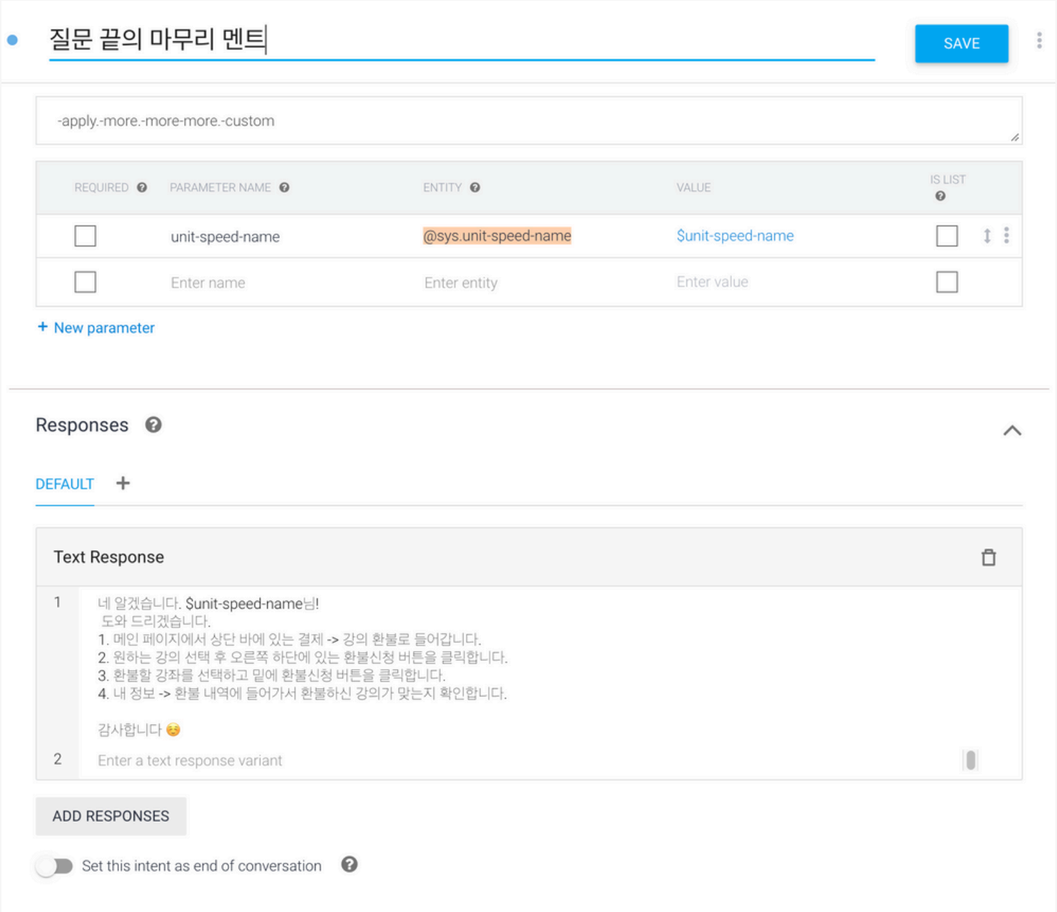
DIALOGFLOW는 GOOGLE CLOUD PLATFORM(GCP)에서 제공하는 자연어 이해(NLU) 플랫폼으로, 챗봇과 같은 대화형 인터페이스를 구축하는 데 사용된다.





챗봇 (DIALOGFLOW)

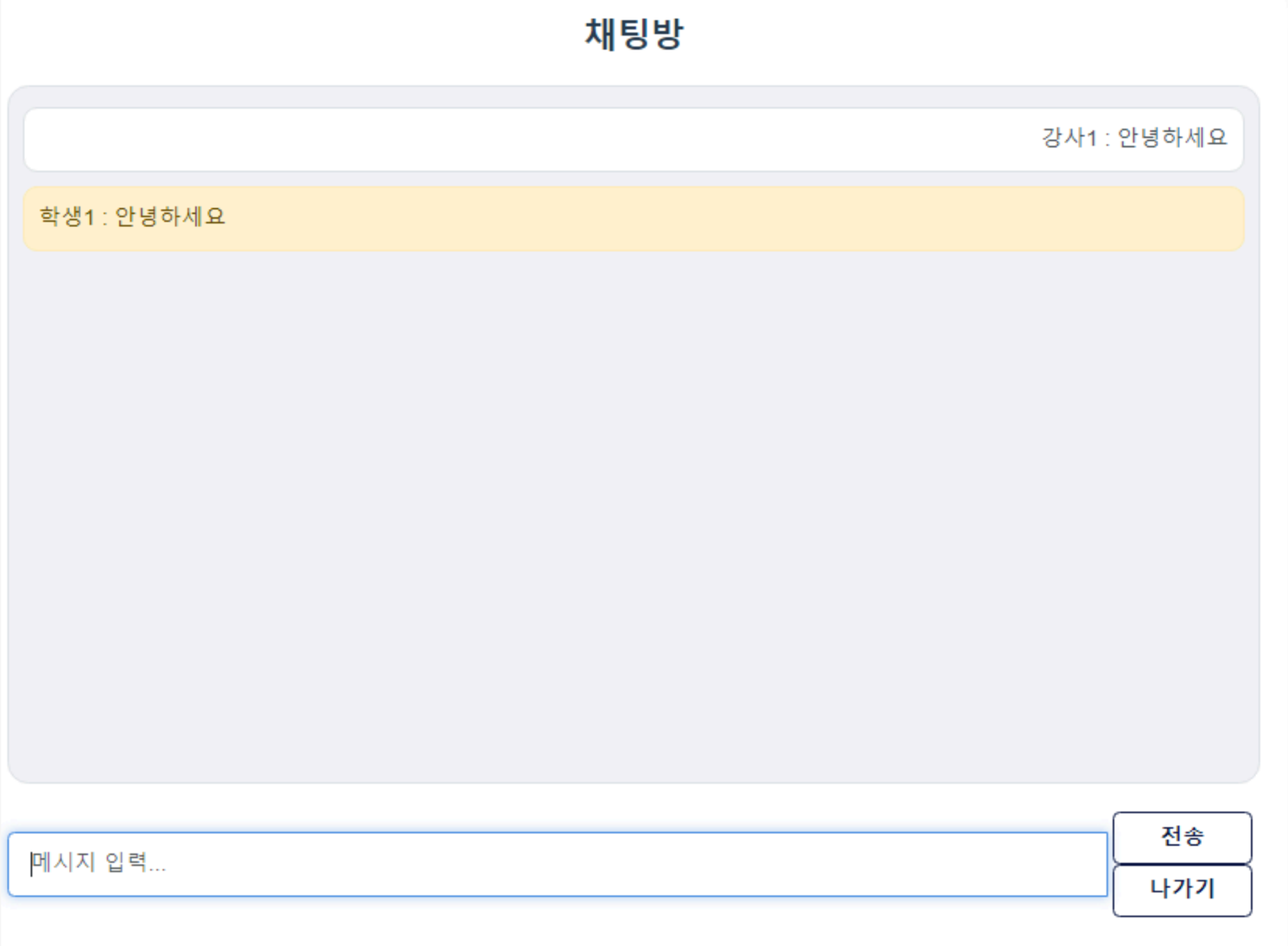





```
<script
src="https://www.gstatic.com/dialogflow-console/fast/messenger/bootstrap.js?v=1"></script>
<script
src="https://www.gstatic.com/dialogflow-console/fast/messenger/bootstrap.js?v=1"></script>

<df-messenger
  intent="WELCOME"
  chat-title="lecturechatbot"
  agent-id="f210b238-719f-4702-8593-9794ed407478"
  language-code="ko"
></df-messenger>
```

DIALOGFLOW에게 교육 시켜주고 챗봇 코드를 가져와 페이지에 적용시킨다.

로그인 유저 전체 채팅



TEXTWEBSOCKETHANDLER

연결 설정 → 메시지 처리 → 연결 해제

CONFIG작성으로 웹소켓 환경 설정 후
실제 기능 처리할 수 있는 핸들러로 이동하는 메시지 처리 구문이다.

```java
@Override
    public void afterConnectionEstablished(WebSocketSession session) throws Exception {
        // 클라이언트가 WebSocket 연결을 맺었을 때 호출됩니다.
        HttpSession httpSession = (HttpSession) session.getAttributes().get("HTTP_SESSION");

        if (httpSession != null) {
            // HTTP 세션에서 로그인한 사용자 정보를 가져옵니다.
            User loggedInUser = (User) httpSession.getAttribute("loggedInUser");

            if (loggedInUser != null) {
                // WebSocket 세션을 리스트에 추가하여 관리합니다.
                list.add(session);
                String entranceMessage = "entrance:" + loggedInUser.getName() + ":" + loggedInUser.getName() + "님 입장";
                for (WebSocketSession sess : list) {
                    sess.sendMessage(new TextMessage(entranceMessage));
                }
                log.info(session + " 클라이언트 접속");
            } else {
                // 로그인한 사용자 정보가 없으면 연결을 거부하고 세션을 닫습니다.
                session.close(CloseStatus.NOT_ACCEPTABLE.withReason("No logged in user"));
            }
        } else {
            // HTTP 세션이 없으면 연결을 거부하고 세션을 닫습니다.
            session.close(CloseStatus.NOT_ACCEPTABLE.withReason("No HTTP session"));
        }
    }
```

로그인 유저 전체 커뮤니티

커뮤니티

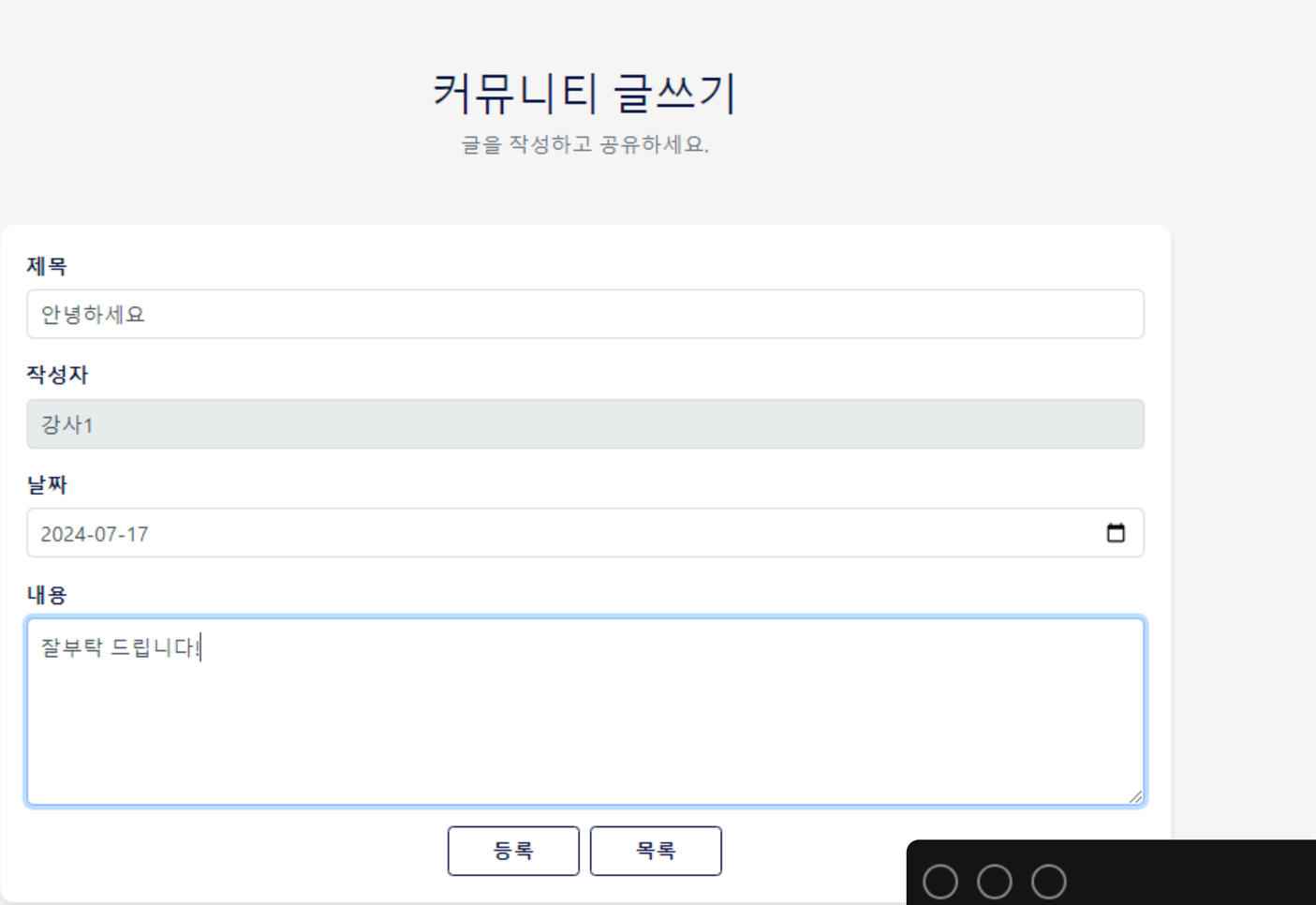
필요한 정보를 검색하고 커뮤니티에 참여하세요.

작성자 | 검색어를 입력해주세요. | 검색

글쓰기

제목	작성자	등록일
개발자에게 가장 중요한 건 마인드셋입니다. ^^	김덕배	2024-07-18
React와 Redux 사용 후기	비키	2024-07-18
Spring Security 적용 방법	홍지수	2024-07-18
Kubernetes 시작하기	황철진	2024-07-17
Docker 실습 강의	정철운	2024-07-15
Python 프로그래밍 도전기	최선중	2024-07-13
HTML/CSS 강의 후기	홍팍	2024-07-12
JavaScript 기초 강의 추천	김동욱	2024-07-10
MySQL 최적화 팁	김덕배	2024-07-09
React 강의 Q&A	이도겸	2024-07-07

로그인 유저 전체 커뮤니티




```
Timestamp myDate = new Timestamp(dateAllc.getTime());
        allcommunity.setAllcDate(myDate);
```

로그인 유저 전체 커뮤니티

안녕하세요

잘부탁^^

작성자: 강사1

작성일: 2024-07-17

목록 수정 🗑

댓글

와 강사님이다!
작성자: 학생1

학생님 반가워요! 🗑
작성자: 강사1

안녕하세요!
작성자: 학생1

댓글 내용

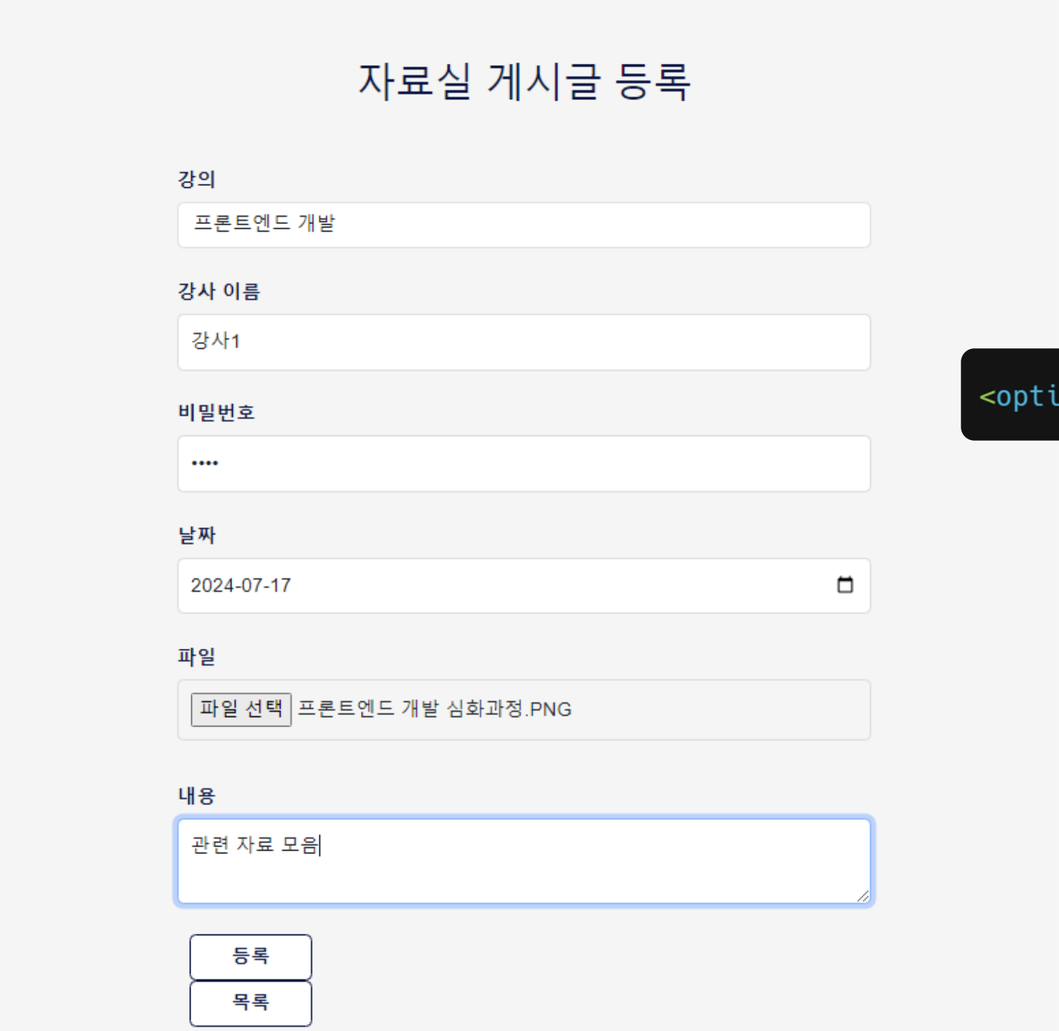
```
// 댓글 추가
    @PostMapping("/comments/cccommentadd")
    @ResponseBody
    public List<CommunityComment> cccommentadd(@RequestBody CommunityComment communitycomment, HttpSession session) {
```

자료실(강사 로그인)

자료실(학생 로그인)

```
<a href="/InstructorWrite" class="btn btn-outline-primary data-sharing-view-write-button"
   th:if="${session.loggedInUser != null and session.loggedInUser.role == 'instructor'}">
   글쓰기</a>
```

자료실
강사 로그인 유저 작성, 수정, 삭제 기능



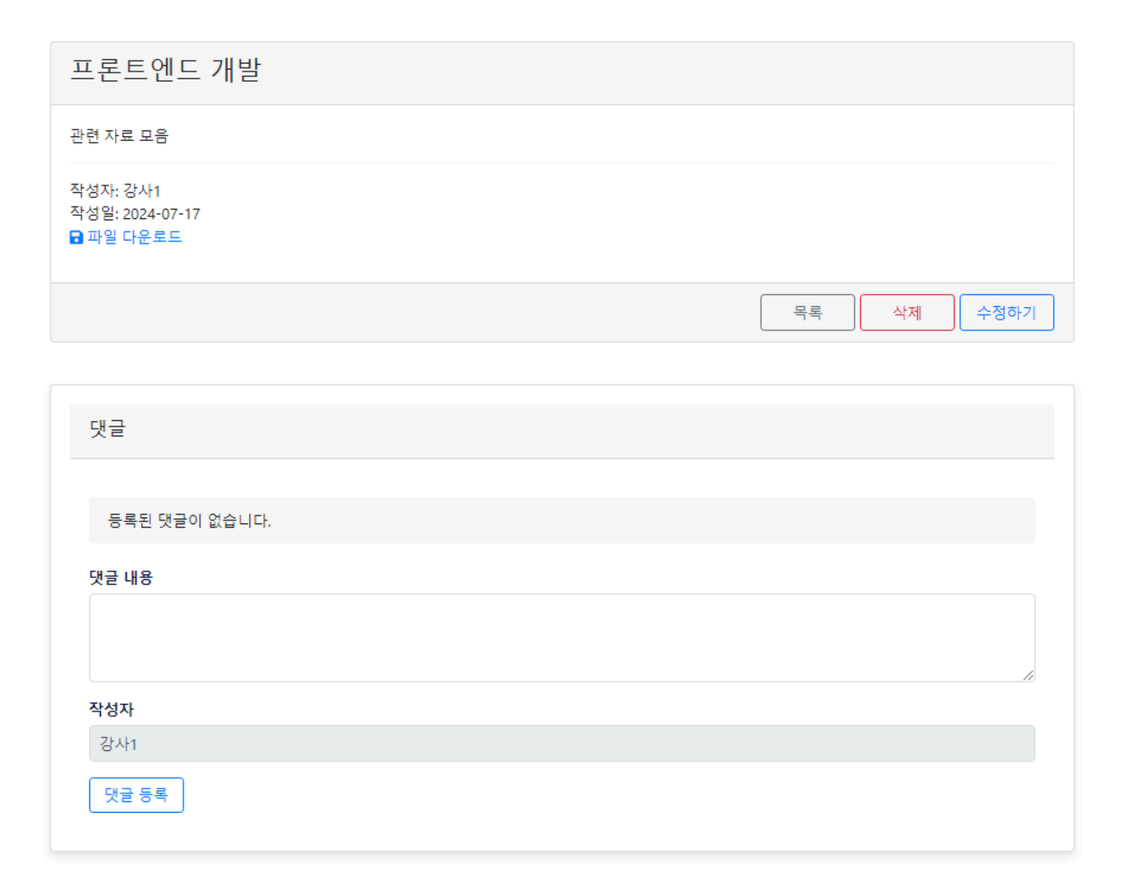
`th:each`를 사용하여 강사가 강의를 올렸을 때
강의 제목을 그대로 가져와 `<option>`으로 생성한다 .

```
<option th:each="lecture : ${lectures}" th:value="${lecture}" th:text="${lecture}"></option>
```

자료실 강사 이름 작성 부분은 세션으로 받아 온다..

```
th:value="${session.loggedInUser.name}"
```

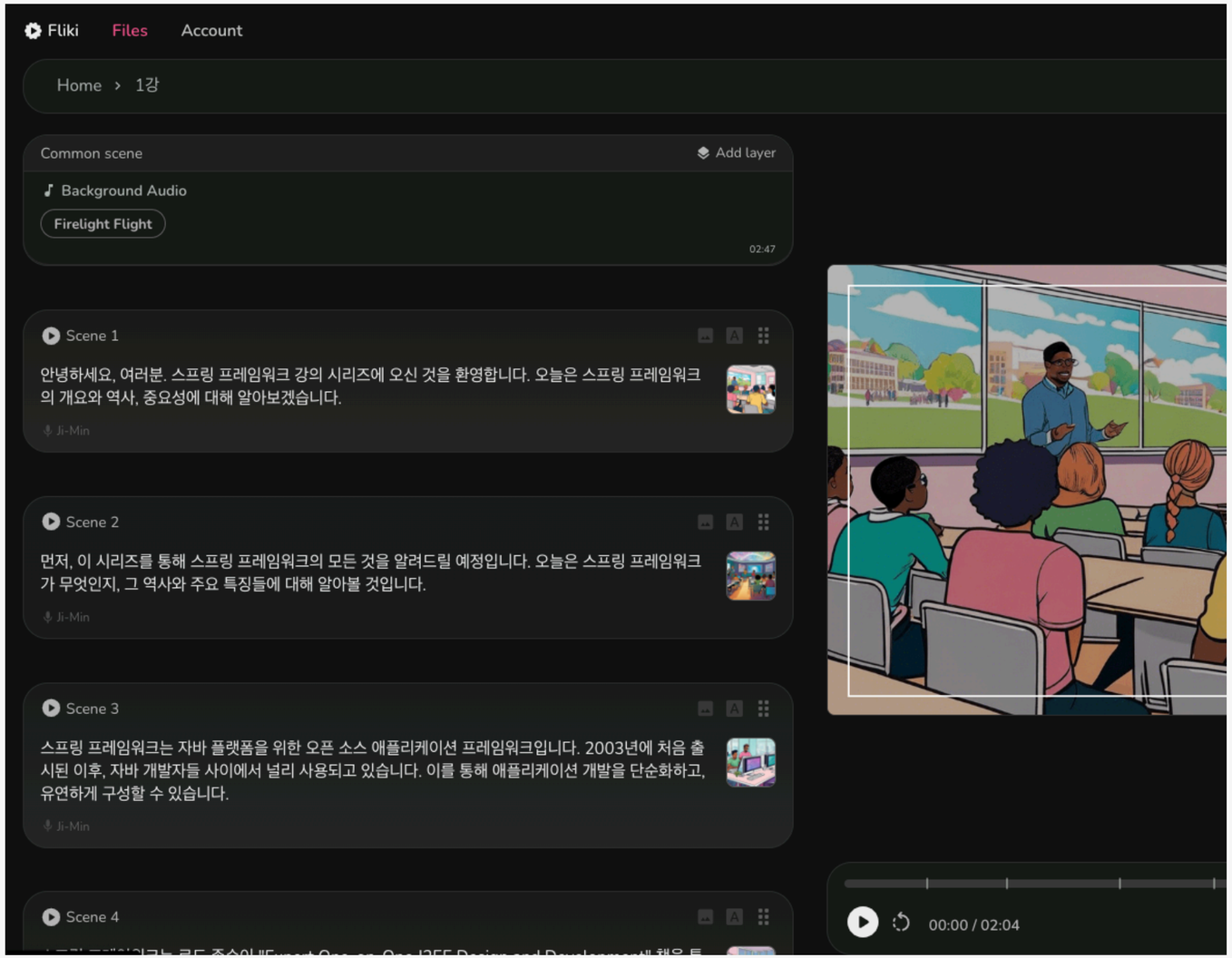
자료실
강사 로그인 유저 작성, 수정, 삭제 기능



사용 기술

Fliki

AI를 활용한 텍스트-음성 변환(TTS) 및 비디오 생성 서비스



프로젝트 시연

후기 및 느낀점

후기 및 느낀점

이형철

이번 파이널 프로젝트는 주제 선정부터 마무리까지 힘들지만 동시에 재미있던 소중한 경험이었습니다. 먼저, 항상 물심양면으로 도와주신 선생님께 감사의 말씀을 드립니다. 그리고 함께 열심히 노력해준 팀원들에게도 감사의 인사를 전하고 싶습니다.

프로젝트를 진행하면서 하고 싶은 기능과 디자인이 많았지만, 적절한 선에서 타협하여 가장 좋은 결과물을 만들어낼 수 있었습니다. 기본 CRUD 기능뿐만 아니라 새로운 API와 라이브러리를 활용하며 많은 것을 배울 수 있었던 시간이었습니다.

이번 프로젝트를 통해 얻은 다양한 경험과 지식은 앞으로의 개발에 큰 도움이 될 것 같습니다.
다시 한번, 모든 팀원들과 선생님께 감사드립니다.👏

고원

결제 API를 프로젝트의 CRUD와 자연스럽게 연결되게 하는 것이 어려웠으나 깊이 파고들다 보니 구현이 되고 깊이 이해하게 되어서 재미를 느꼈습니다. 팀장부터 시작해서 모두가 솔선수범해서 협력한 덕분에 즐겁게 완성할 수 있었습니다.

후기 및 느낀점

김동욱

해보지 않았던 기능들이나 API를 쓰려고 하다보니까 정말 난관이 많았는데 팀원분들께서 도움을 많이 받았고 또 선생님께서도 많이 도움을 주셔서 좋은 결과가 나온거같습니다.
저희 반 그리고 원진 선생님 정말 고생 많으셨고 다들 좋은 결과 있기 바랍니다. 수고하셨습니다.

김희선

- 어려운 부분 : 처음에 퀴즈를 만든다고 했을 때 시간이 없지만 일단 도전이라도 해보려고 해봤지만 테이블 끼리 연결하는 것과 어떻게 퀴즈를 내고 저장하고 빼 와야 하는지 고민이 많았습니다. 그러다 보니 시간이 없어 포기를 했지만 학원 일정이 끝난 후 어렵지만 혼자 다시 도전 해보려고 합니다.

- 해결한 부분 : 댓글 작성 후 삭제 등을 배우긴 했지만 저한테 꽤나 어렵다고 생각 한 부분이였지만 선생님께 여쭤보고 설명을 듣고 어떻게든 고쳐지는게 제 스스로서 신기한 경험이였습니다.

- 후기 : 모르는 부분을 팀원들께 공유하면서 배우고 공부할 수 있어서 좋았습니다.

THANK YOU.

도움주신분

원진 선생님 👨‍🏫

양은정님

우황희님

이현학님