

# 중간 발표

팀 명 : 양까라

과 목 : 프로젝트(캡스톤 디자인)

담당교수님 : 성락운 교수님

날 짜 : 2024-05-10

# 목차

---

- 01 프로젝트 소개
- 02 계획 변경
- 03 중간결과물
- 04 문제 및 해결
- 05 이후 계획
- 06 피드백 및 느낀점

# 프로젝트 소개

---

사이트 이름 : 앙까라

주제 : 유튜브 랭킹 및 정보 플랫폼

설명 : 유튜브 동영상 및 뉴스 관련 다양한 정보와 커뮤니티를 제공하는 사이트

Part 1,

프로젝트 소개

시그널

시그널 실시간 검색어

현재 기준 사용자가 가장 많이 검색 하는 키워드입니다.

2024년 5월 10일 금요일 오전 12:48

가이드

1

특검엔 정치 공세

-

2

신변 위협글 수사

-

3

소프트뱅크 라인야후

-

4

돌싱글즈 5

▲

5

김미영 팀장 탈옥

-

6

김재중 바다

-

7

사고 도로공사 직원

▼

8

레알마드리드

-

9

최원권 감독 김상식

-


10

아파트서 아기 추락사

▼


언론사별 가장 많이 본 뉴스

각 언론사의 가장 많이 본 기사 1건을 제공합니다.




카페에 아이 두고 나간 여성...알바생에 "왜 애 혼자 화장실 보내" 버럭

뉴스1




유튜브로 생중계된 피습 순간...유튜버 간 갈등이 부른 '참극' ...

JTBC



법원 앞에서 흥기 피습...50대 유튜버 사망

KBS



[단독] '여친 살해' 의대생, 부모 통화 끝에 "악 놓고 와"...피해자 발견 90분 지체











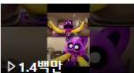




YTN

<

1 / 10

>

플레이 보드

순위	채널	조회수	영상
1 NEW	 김프로KIMPRO #김프로 #kimpro	+ 114,185,298	 >13.8만  >65.2만
2 NEW	 HYBE LABELS #HYBE #HYBE LABELS	+ 35,444,992	 >2.8만  >16만
3 NEW	 SBS 뉴스 #SBSNEWS #sbs	+ 24,012,149	 >204  >16만
4 ↓ 3	 게항쓰 GH'S #게항 #게항쓰	+ 22,738,889	 >1.4백만  >55.7만
5 ↓ 1	 승비니 Seungbini	+ 22,681,593	 >10.2만  >37.3만

©Saebyeol Yu. Saebyeol's PowerPoint

# 계획 변경

6주차: 04/26~05/02 - API 개발( 백엔드 개발 )

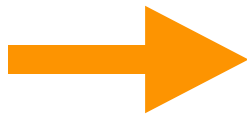
7주차: 05/03~05/09 - API 개발( 백엔드 개발 )

8주차: 05/10~05/16 - 프론트엔드 개발

9주차: 05/17~05/23 - 프론트엔드 개발

10주차: 05/24~05/30 - 통합 테스트 및 디버깅

11주차: 05/31~06/06 - 오류 수정



6주차: 04/26~05/02 - 프론트, 백엔드개발

7주차: 05/03~05/09 - 프론트, 백엔드개발

8주차: 05/10~05/16 - 프론트, 백엔드개발

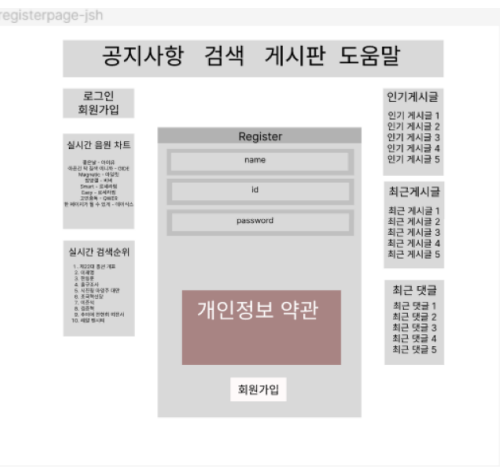
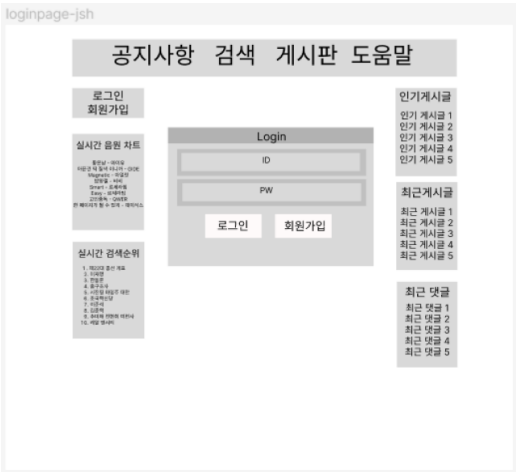
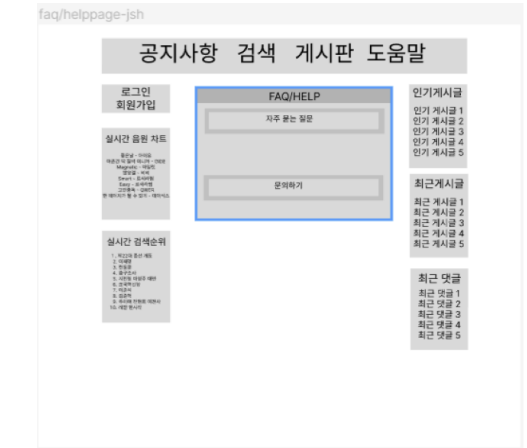
9주차: 05/17~05/23 - 프론트, 백엔드개발

10주차: 05/24~05/30 - 통합 테스트 및 디버깅

11주차: 05/31~06/06 - 배포

기초적인 웹페이지, 백엔드, 데이터수집코드

Part 3, **중간 결과물**

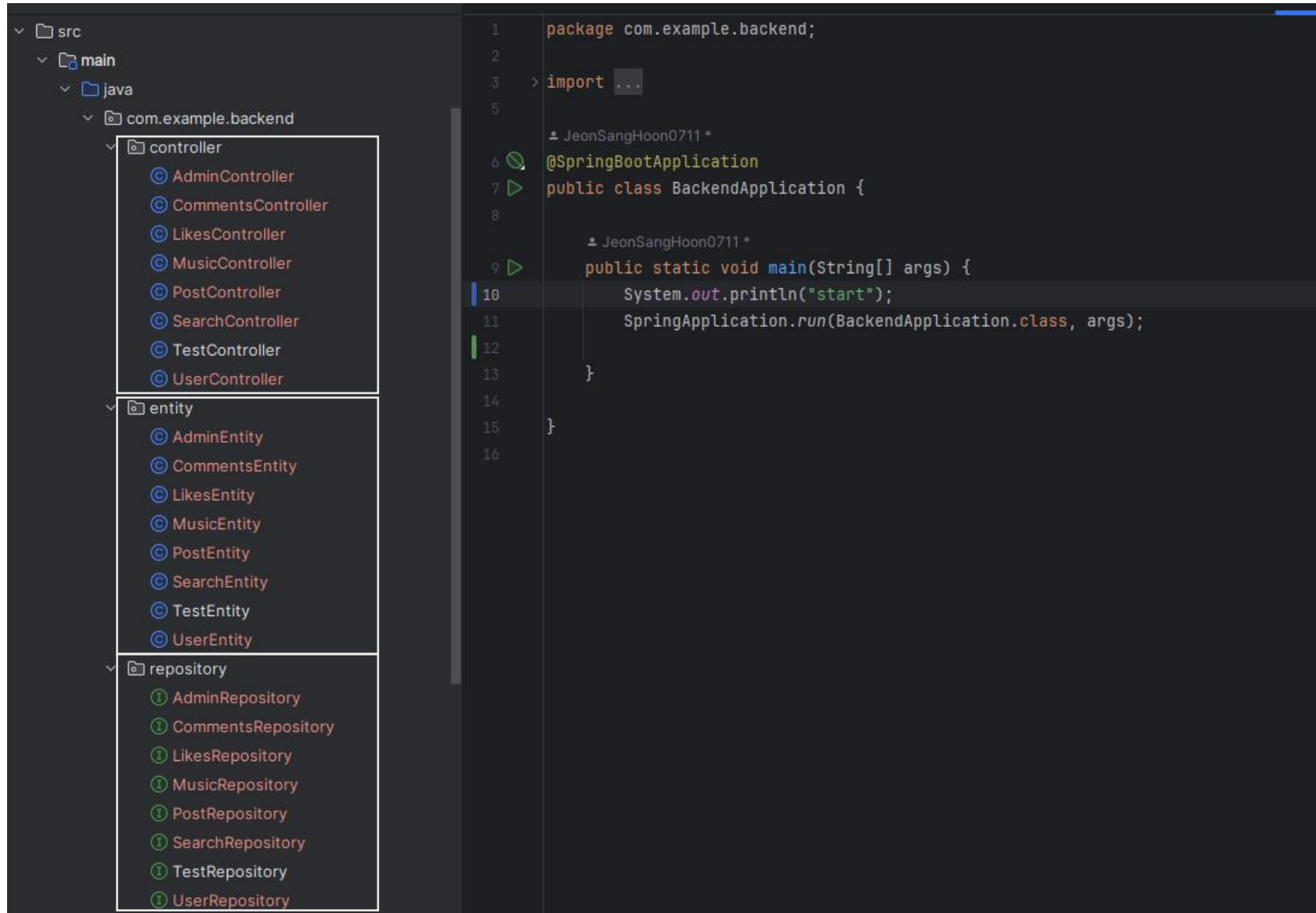


### Part 3,

## 중간 결과물







## 더미데이터

```
-- `topdb`.`user` 테이블에 들어갈 데이터
INSERT INTO `topdb`.`user` (`name`, `id`, `email`, `pw`) VALUES
('김철수', 'chulsoo', 'chulsoo@example.com', 'password123'),
('이영희', 'younghee', 'younghee@example.com', 'password456'),
('박지민', 'jimin', 'jimin@example.com', 'password789');

-- `topdb`.`post` 테이블에 들어갈 데이터
INSERT INTO `topdb`.`post` (`uid`, `title`, `content`, `date`, `views`, `likecount`, `category`) VALUES
(1, '오늘의 일기', '오늘은 맑았다.', '2024-05-09', 50, 5, '일상'),
(2, '여행 계획', '이번 여름에는 제주도로 가고 싶다.', '2024-05-10', 70, 8, '여행'),
(3, '책 추천', '이번 달에 읽은 책 중에서 추천하고 싶은 책은...', '2024-05-11', 30, 2, '취미');

-- `topdb`.`comments` 테이블에 들어갈 데이터
INSERT INTO `topdb`.`comments` (`pid`, `uid`, `date`, `content`) VALUES
(1, 2, '2024-05-12', '정말 재미있어 보여요!'),
(2, 3, '2024-05-13', '제주도 좋죠! 저도 가보고 싶어요.'),
(3, 1, '2024-05-14', '오, 저 책 읽어봐야겠어요. 추천 감사합니다!');

-- `topdb`.`likes` 테이블에 들어갈 데이터
INSERT INTO `topdb`.`likes` (`likedate`, `pid`, `uid`) VALUES
('2024-05-15', 1, 3),
('2024-05-16', 2, 1),
('2024-05-17', 3, 2);

-- `topdb`.`music` 테이블에 들어갈 데이터
INSERT INTO `topdb`.`music` (`title`, `artist`, `date`) VALUES
('하루의 시작', '김노래', '2024-04-03'),
('밤하늘의 별들', '이가수', '2024-04-04'),
('여름바람', '박음악', '2024-04-05');

-- `topdb`.`search` 테이블에 들어갈 데이터
INSERT INTO `topdb`.`search` (`title`, `date`) VALUES
('최신 영화', '2024-05-18'),
('맛집 탐방', '2024-05-19'),
('여름 휴가지 추천', '2024-05-20');

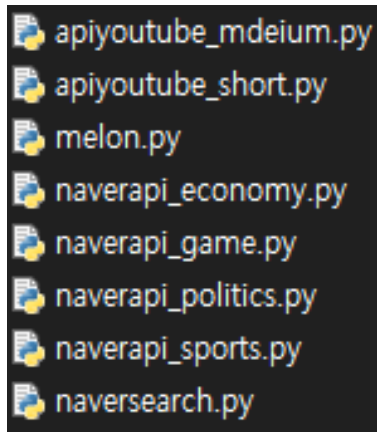
-- `topdb`.`admin` 테이블에 들어갈 데이터
INSERT INTO `topdb`.`admin` (`user_uid`) VALUES
(1),
(2),
(3);
```

## 호출 결과

```
localhost:8080/api/posts
pretty print 적용

[
  {
    "pid": 1,
    "user": {
      "uid": 1,
      "id": "chulsoo",
      "name": "김철수",
      "email": "chulsoo@example.com",
      "pw": "password123"
    },
    "title": "오늘의 일기",
    "content": "오늘은 맑았다.",
    "date": "2024-05-09",
    "views": 50,
    "likecount": 5,
    "category": "일상"
  },
  {
    "pid": 2,
    "user": {
      "uid": 2,
      "id": "younghee",
      "name": "이영희",
      "email": "younghee@example.com",
      "pw": "password456"
    },
    "title": "여행 계획",
    "content": "이번 여름에는 제주도로 가고 싶다.",
    "date": "2024-05-10",
    "views": 70,
    "likecount": 8,
    "category": "여행"
  },
  {
    "pid": 3,
    "user": {
      "uid": 3,
      "id": "jimin",
      "name": "박지민",
      "email": "jimin@example.com",
      "pw": "password789"
    },
    "title": "책 추천",
    "content": "이번 달에 읽은 책 중에서 추천하고 싶은 책은...",
    "date": "2024-05-11",
    "views": 30,
    "likecount": 2,
    "category": "취미"
  }
]
```

## 데이터 수집 코드



```
from googleapiclient.discovery import build

# API 정보 설정
YOUTUBE_API_SERVICE_NAME = "youtube"
YOUTUBE_API_VERSION = "v3"
DEVELOPER_KEY = "발급받은키" # 여기에 발급받은 API 키를 입력하세요.

# YouTube API 클라이언트 생성
youtube = build(YOUTUBE_API_SERVICE_NAME, YOUTUBE_API_VERSION, developerKey=DEVELOPER_KEY)

# 단계 1: 특정 날짜에 업로드된 영상 검색 (한국에서 업로드된 2분 미만 영상)
search_response = youtube.search().list(
    part="id",
    type="video",
    regionCode="KR", # 한국에서 업로드된 영상만 검색
    videoDuration="short", # 4분 미만 영상 검색 (API에서는 정확히 2분 미만을 지정할 수 없으며, 'short'는 4분 미만을 의미함)
    publishedAfter="2024-04-05T00:00:00Z",
    publishedBefore="2024-04-06T00:00:00Z",
    maxResults=10, # 최대한 많은 결과를 얻기 위해 (API 한계는 고려해야 함)
    order="viewCount" # 조회수가 높은 순으로 정렬
).execute()

# 단계 2: 얻은 영상 ID로부터 조회수를 확인
video_ids = [item['id']['videoId'] for item in search_response['items']]
videos_response = youtube.videos().list(
    part="snippet,statistics",
    id=",".join(video_ids),
).execute()

# 조회수에 따라 정렬하고 상위 10개 선택
videos_sorted = sorted(
    videos_response['items'],
    key=lambda x: int(x['statistics']['viewCount']),
    reverse=True)[:10]

# 결과 출력
print("Top 10 videos by view count:")
for video in videos_sorted:
    title = video["snippet"]["title"]
    video_id = video["id"]
    view_count = video["statistics"]["viewCount"]
    like_count = video["statistics"].get("likeCount", "N/A")
    comment_count = video["statistics"].get("commentCount", "N/A")
    tags = video["snippet"].get("tags", ["No Tags"]) # 태그 정보 가져오기, 태그 없을 시 ["No Tags"]
    tags_str = ", ".join(tags) # 태그 리스트를 문자열로 변환
    print(f"{title} (views: {view_count}, likes: {like_count}, comments: {comment_count}, tags: {tags_str}) - https://www.youtube.com/watch?v={video\_id}\n")
```

## 수집한 데이터

### Youtube

title : 남자친구를 놀라게 하고 싶었지만 놀라는 것은 그녀였다 Part1

views : 44604555

likes : 1434537

comments : 8583

tags : 김프로, kimpro, 김동준, 김프로유튜브, 유튜브김프로

adress : <https://www.youtube.com/watch?v=tU-5z90etel>

### News

"title": "SK증권 영업부PIB센터, 15일 고객초청 2분기 <b>경제</b>전망 세미나 개최",

"originallink": "https://www.etoday.co.kr/news/view/2349788",

"link": "https://www.etoday.co.kr/news/view/2349788",

"description": "2분기 <b>경제</b>전망 세미나를 개최한다고 11일 밝혔다.

이번 세미나는 SK증권 리서치센터의 강재현 연구위원... 2분기 <b>경제</b>전망을 주제로 강연을 진행한다.

세미나는 희망하는 누구나 참석할 수 있으며, 참석 문의는... ",

pubDate": "Thu, 11 Apr 2024 17:08:00 +0900"

## Cors 문제

개발 환경과 실제 데이터베이스 환경 간의 대소문자 구분 차이

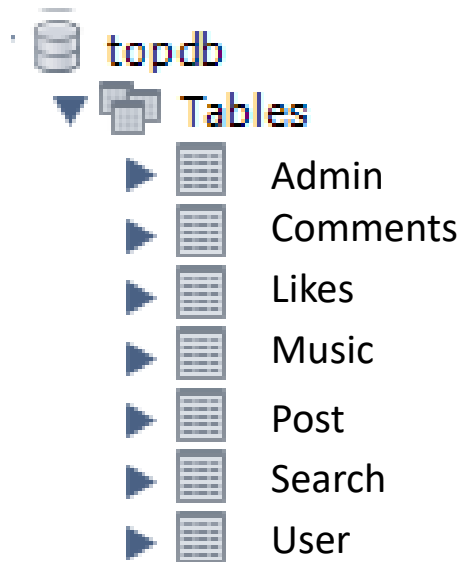
## Cors 문제

```
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.CorsRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

♣ JeonSangHoon0711
@Configuration
public class WebMvcConfig implements WebMvcConfigurer {
    0개의 사용위치 ♣ JeonSangHoon0711
    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping(pathPattern: "/api/**")
            .allowedOrigins("http://localhost:3000")
            .allowedMethods("*")
            .allowedHeaders("*")
            .allowCredentials(true);
    }
}
```

# 문제 및 해결

개발 환경과 실제 데이터베이스 환경 간의 대소문자 구분 차이

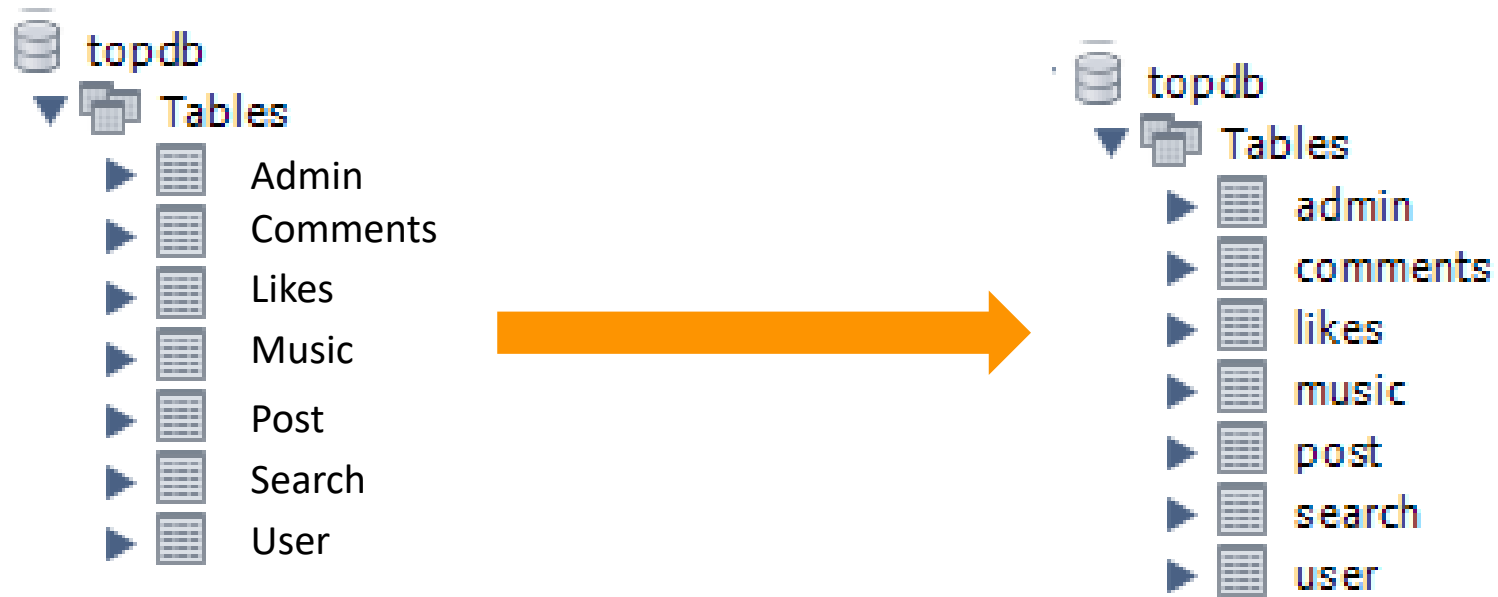


```
@Entity
@Table(name = "User") //
public class UserEntity {
```



```
java.sql.SQLException Create breakpoint : Table 'topdb.user' doesn't exist
```

# 문제 및 해결





# 이후 계획

---

## 현재까지의 계획

1주차: 03/22~03/28 - 개발환경 구성

2주차: 03/29~04/04 - 웹사이트 디자인

3주차: 04/05~04/11 - DB 모델링 구성

4주차: 04/12~04/18 - 중간고사 준비

5주차: 04/19~04/25 - 중간고사

6주차: 04/26~05/02 - 프론트,백엔드 개발

7주차: 05/03~05/09 - 프론트,백엔드 개발

## 이후 계획

8주차: 05/10~05/16 - 프론트,백엔드 개발

9주차: 05/17~05/23 - 프론트,백엔드 개발

10주차: 05/24~05/30 - 통합 테스트 및 디버깅

11주차: 05/31~06/06 - 배포

## 이후 계획

8주차 : 사용자 페이지 구현

8주차 : 도움말 페이지 구현

8주차 : 공지사항 페이지 구현

8주차 : 검색 페이지 구현

8,9주차 : 데이터 수집코드를 스케줄러 사용하여 자동화

8,9주차 : 데이터 수집 코드와 데이터베이스 연동

8,9주차 : 데이터베이스에 실제 데이터 삽입

8,9주차 : 로그인이나 기타 **CRUD**기능 백엔드 로직 구현

8,9주차 : 백엔드와 프론트의 **API** 연동

10주차 : 통합 테스트 및 디버깅

11주차 : 배포

## 피드백 및 느낀점

---

작은 오류들이 너무 많다

- 혼자 작업하는 것이 아니기 때문에 타인의 코드와 자신이 작성한 코드로 인한 오류 발생이 많다
- 팀원과 더 많은 소통이 필요한 것 같다

디자인 구현에 어려움이 있다

- 디자인 구현 과정에서 개개인의 시각 차이로 인한 의견 충돌이 발생한다.
- 자신의 의견에 대한 명확한 생각과 이유를 제안 해야 할것 같다

<https://gemini.google.com/>

<https://wrtn.ai/>

<https://www.signal.bz/>

<https://playboard.co/>

[https://youtu.be/x6i3\\_LfeTjY](https://youtu.be/x6i3_LfeTjY)

감사합니다