

○ Ezen_Jeonju

A team's Second Web Project

Ezen_Developer

Git: https://github.com/ProjectDaon/Ezen_Jeonju_project.git

○ INDEX

TEAM

팀 구성원 및 역할

PROJECT



프로젝트 소개 ----- 4P

프로젝트 목표 ----- 5P

프로젝트 일정 ----- 6P

사용기술 ----- 7P

요구사항 ----- 8P

DEVELOPMENT



DB 설계 (ERD) ----- 9P

다이어그램 ----- 11P

사이트맵 ----- 14P

UI/UX ----- 15P

주요기능 ----- 27P

확장 가능성 ----- 85P

시연

Q & A

소감

TEAM Introduction

팀 구성원 및 역할

Team members and roles

[FULLSTACK]



문다온

Project Master

- 일정관리
- 역할분배
- Contents 구현
- Mypage 구현



박건도

Project Assistant

- 화면편집
- 디자인
- MainPage 구현
- MainPage 디자인
- MainPage 설정 기능



김민정

Project Leader

- 기획
- 화면편집
- Notice 구현
- 페이지 디자인



윤대희

Project Assistant

- DB 설계
- 테스트
- Schedule 구현
- 일정 등록
- 일정 상세페이지

PROJECT Introduction

프로젝트 소개

Project Introduction

Project



클라우드 및 Spring Framework를 활용한
SNS 데이터활용 지역홍보시스템



Project Name

Ezen_Jeonju

PROJECT Introduction

프로젝트 목표

Project Objectives



시스템 목표

전주 관광지 소개

리뷰를 통한 관광객들간의 소통

여행 루트를 직접 계획



팀 목표

Spring Framework 기술 습득

각종 API 기술 습득

Frontend 기술 향상

PROJECT Introduction

프로젝트 일정
Project Schedule

1. 착수 및 분석

요구사항 정의서 작성

Use Case Diagram 작성

착수 계획서 작성

2023-12-04 ~ 2023-12-08

2. 설계

메뉴구조 사이트맵

스토리보드 작성

2023-12-04 ~ 2023-12-08

DB 설계

프로그램 설계

2023-12-08 ~ 2023-12-15

3. 개발

페이지별 디자인

프로그램 개발

2023-12-18 ~ 2024-01-12

업체참여 멘토링

2024-01-05

4. 테스트

체크리스트 작성

개발자 단위 테스트

애플리케이션 성능 테스트

2024-01-12 ~ 2024-01-16

5. 완료

최종발표 : 2024-01-17

PROJECT Introduction

사용기술

Technology used

개발도구

프로그래밍



Spring Tool Suite 3
version
3.9.13.RELEASE
Eclipse 2020-06(4.16.0)

DB 개발 및 설계



SQLGate
version
9.17.5.0

그래픽 편집



PowerPoint
version
2016

관리도구

서버



Tomcat
version
9.0

형상관리



GitHub

브라우저



Chrome
version
120.0.6099

프로그래밍 언어

프로그래밍



Java
version
1.8

DB



MySQL
version
8.0

웹 프로그램



jQuery
version
3.7.1



HTML
version
5



Javascript
version
1.5

디자인



CSS
version
3

PROJECT Introduction

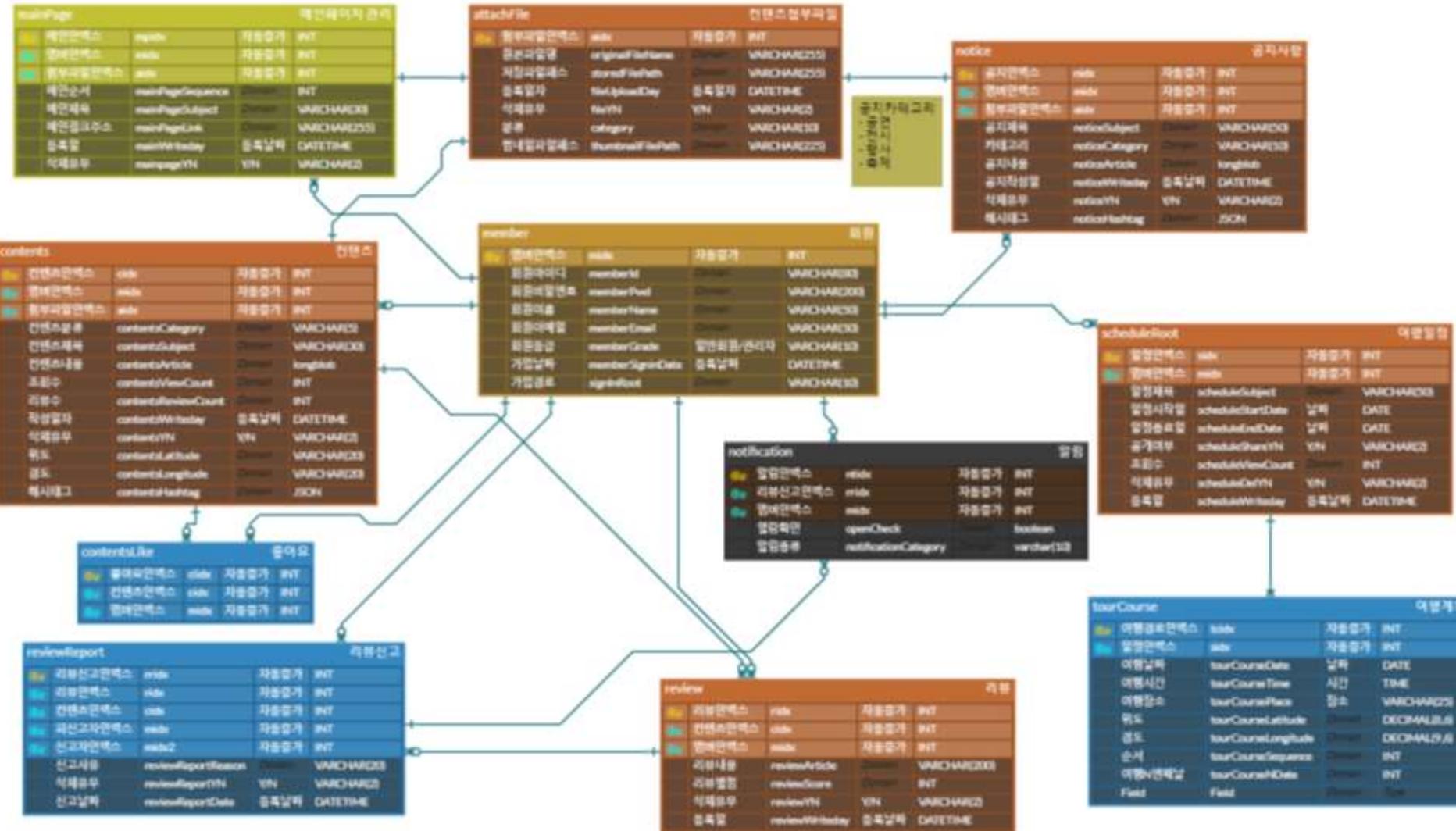
요구사항

Requirements

	회원/공통	관리자
 회원관리	로그인/로그아웃 SNS 로그인 회원가입 비밀번호 변경 아이디/비밀번호 찾기	로그인 로그아웃
 컨텐츠	컨텐츠 조회 지도 조회 리뷰 좋아요 블로그 리뷰 조회	글 작성 수정/삭제
 리뷰	리뷰 쓰기 리뷰 조회 본인 리뷰 삭제 신고하기 신고알림	신고 조회 리뷰 삭제
 여행일정	일정 만들기 일정 조회 일정 공유 일정 삭제	
 공지	공지사항 조회	공지 작성 수정/삭제
 마이페이지	나의 리뷰 나의 좋아요 나의 일정	신고 목록 신고 처리

DEVELOPMENT

DB 설계(ERD)
DataBase Design



DEVELOPMENT

DB 설계(ERD)

DataBase Design

contents		컨텐츠	
컨텐츠인덱스	cidx	자동증가	INT
멤버인덱스	midx	자동증가	INT
첨부파일인덱스	aidx	자동증가	INT
컨텐츠분류	contentsCategory	Domain	VARCHAR(5)
컨텐츠제목	contentsSubject	Domain	VARCHAR(30)
컨텐츠내용	contentsArticle	Domain	longblob
조회수	contentsViewCount	Domain	INT
리뷰수	contentsReviewCount	Domain	INT
작성일자	contentsWrittenDay	등록날짜	DATETIME
삭제유무	contentsYN	Y/N	VARCHAR(2)
위도	contentsLatitude	Domain	VARCHAR(20)
경도	contentsLongitude	Domain	VARCHAR(20)
해시태그	contentsHashtag	Domain	JSON

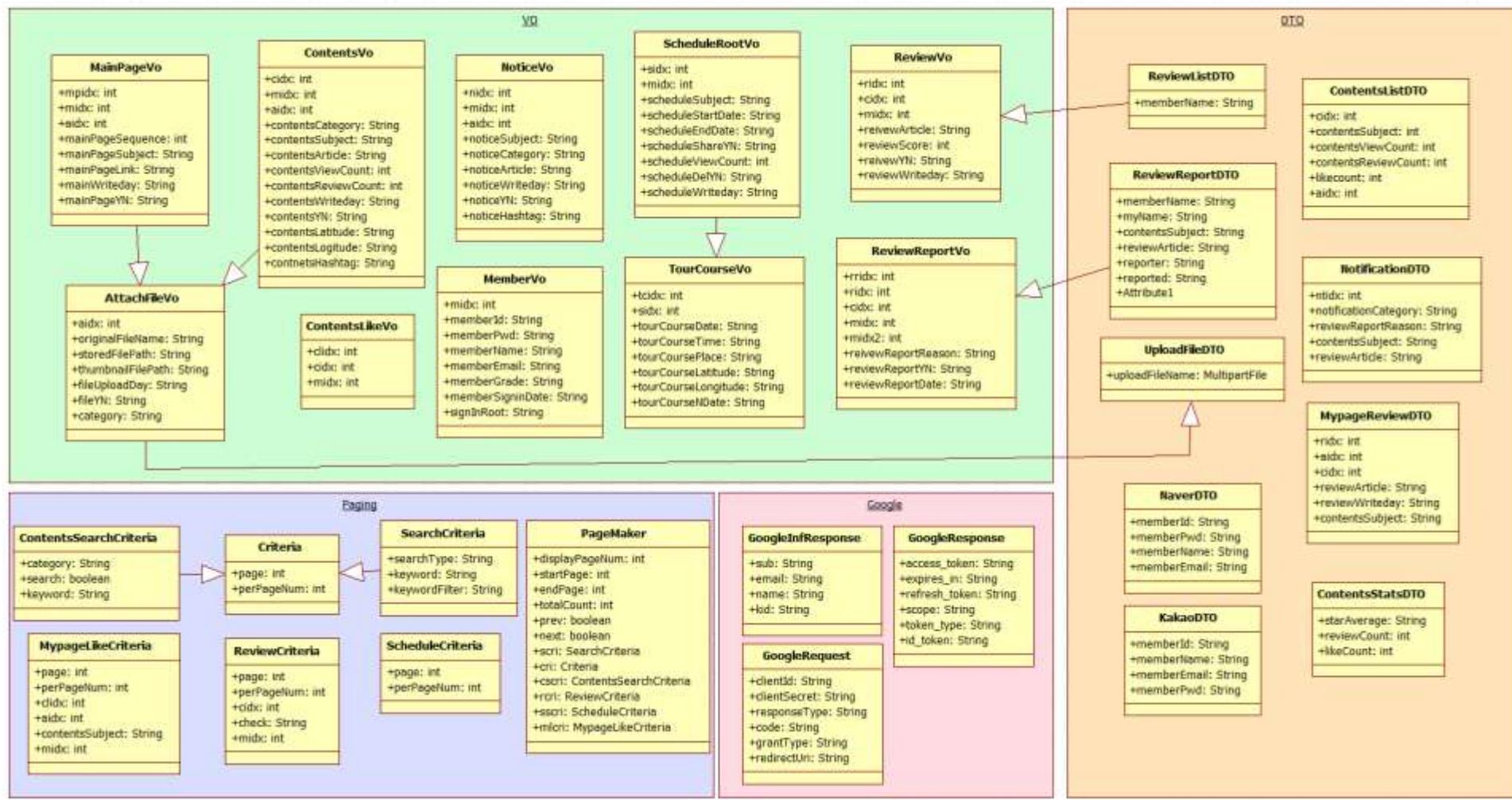
트리거 사용

```
CREATE TRIGGER `ezenjeonju`.`review_count`
AFTER INSERT ON `review`          리뷰테이블 INSERT 명령어 실행 후
FOR EACH ROW
begin
    declare in_cidx int;
    set in_cidx = NEW.cidx;
    update contents set contentsReviewCount = contentsReviewCount+1 where cidx = in_cidx;
end
```

```
CREATE TRIGGER `ezenjeonju`.`review_count_minus`
AFTER UPDATE ON `review`          리뷰테이블 UPDATE 명령어 실행 후
FOR EACH ROW
begin
    declare in_cidx int;
    set in_cidx = NEW.cidx;
    update contents set contentsReviewCount = contentsReviewCount-1 where cidx = in_cidx;
end
```

DEVELOPMENT

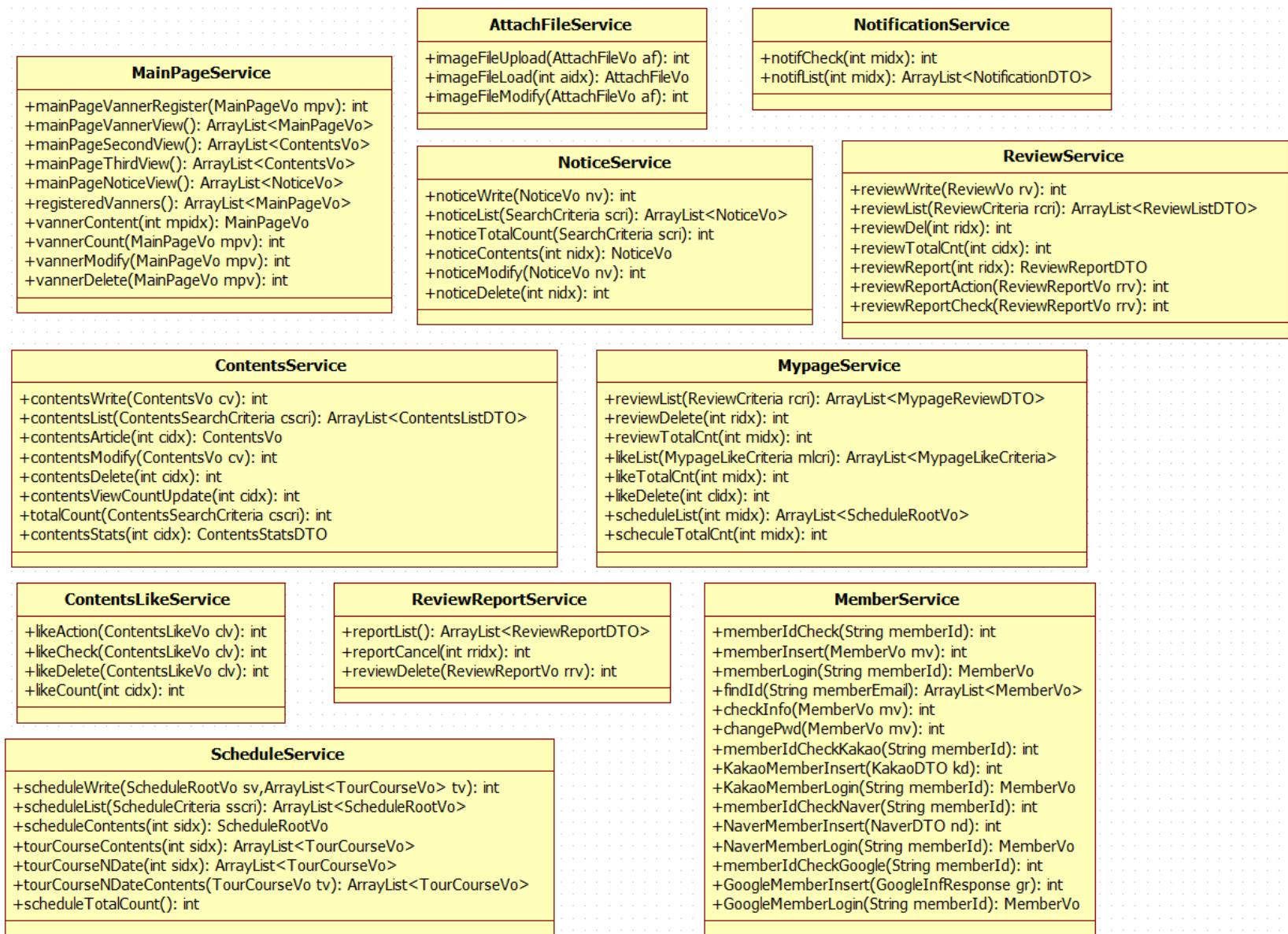
클래스다이어그램(Domain) Class Diagram



DEVELOPMENT

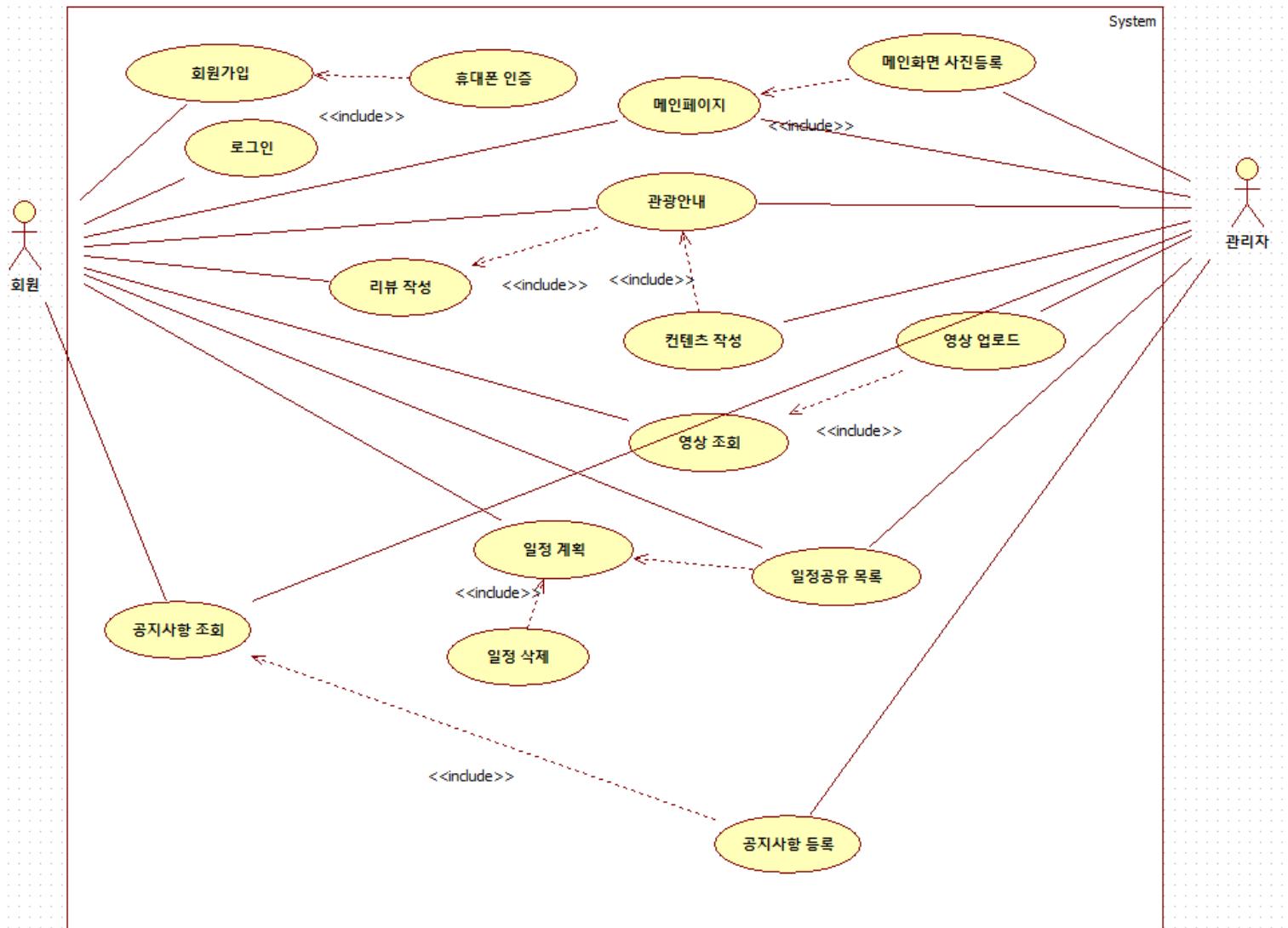
클래스 다이어그램(Service)

Class Diagram



DEVELOPMENT

Use Case Diagram
Use Case Diagram



DEVELOPMENT

사이트맵
SiteMap

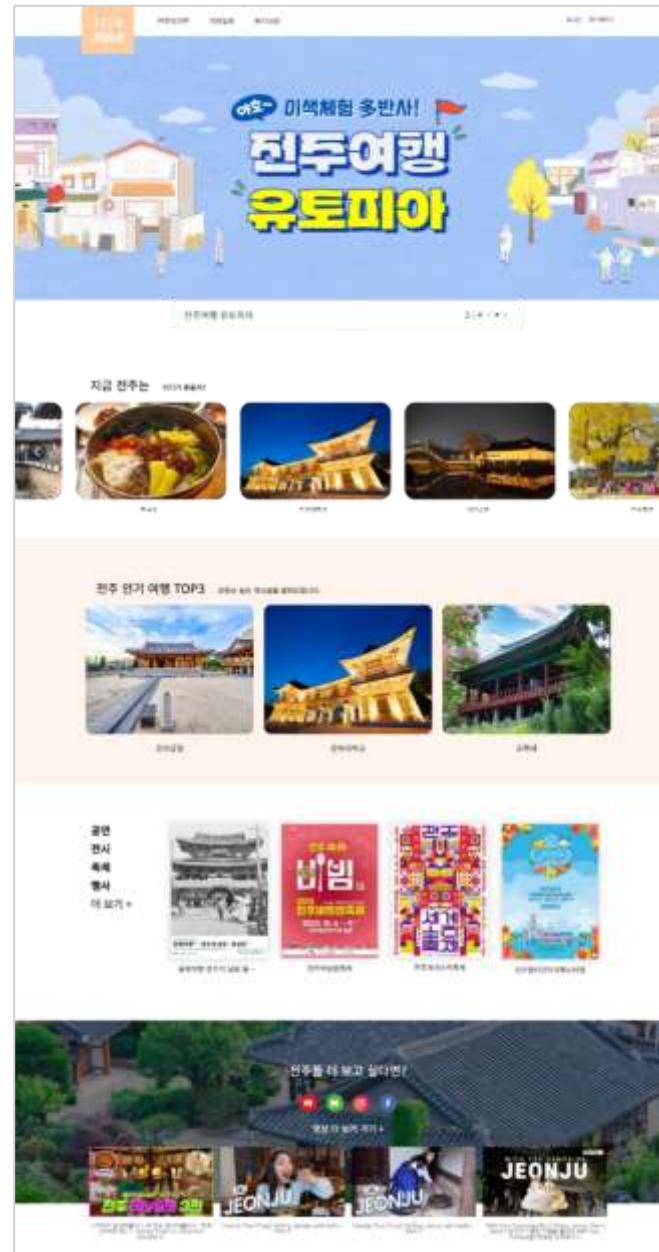
메인페이지				
로그인	마이페이지	전주에가면	여행일정	공지사항
로그인	나의 여행일정	명소	여행공유	공지사항
SNS 로그인	나의 리뷰	음식		
회원가입	나의 좋아요	영상		
ID/PW 찾기				

DEVELOPMENT

UI/UX

UserInterface/UserExperience

메인페이지



DEVELOPMENT

UI/UX

UserInterface/UserExperience

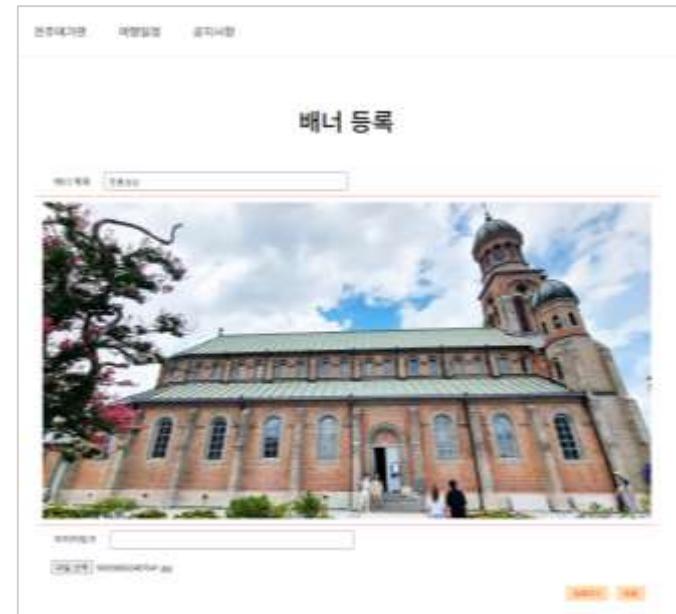
관리페이지

신고목록						
No	신고자	쪽번호	제시물	신고사유	등록내용	신고일자
1	김민아	010-1234-5678	화재신고	화재신고	화재발생 시 신고내용에 일부 허위로 신고된 경우 신고내용은 수정해도 오피셜서버로 등록됩니다.	2024-01-08 10:00:00
2	이수영	010-1234-5678	화재신고	화재신고	화재발생 시 신고내용에 일부 허위로 신고된 경우 신고내용은 수정해도 오피셜서버로 등록됩니다.	2024-01-08 10:00:00
3	김민아	010-1234-5678	화재신고	화재신고	화재발생 시 신고내용에 일부 허위로 신고된 경우 신고내용은 수정해도 오피셜서버로 등록됩니다.	2024-01-08 10:00:00
4	이수영	010-1234-5678	화재신고	화재신고	화재발생 시 신고내용에 일부 허위로 신고된 경우 신고내용은 수정해도 오피셜서버로 등록됩니다.	2024-01-08 10:00:00
5	김민아	010-1234-5678	화재신고	화재신고	화재발생 시 신고내용에 일부 허위로 신고된 경우 신고내용은 수정해도 오피셜서버로 등록됩니다.	2024-01-08 10:00:00
6	이수영	010-1234-5678	화재신고	화재신고	화재발생 시 신고내용에 일부 허위로 신고된 경우 신고내용은 수정해도 오피셜서버로 등록됩니다.	2024-01-08 10:00:00
7	김민아	010-1234-5678	화재신고	화재신고	화재발생 시 신고내용에 일부 허위로 신고된 경우 신고내용은 수정해도 오피셜서버로 등록됩니다.	2024-01-08 10:00:00
8	이수영	010-1234-5678	화재신고	화재신고	화재발생 시 신고내용에 일부 허위로 신고된 경우 신고내용은 수정해도 오피셜서버로 등록됩니다.	2024-01-08 10:00:00
9	김민아	010-1234-5678	화재신고	화재신고	화재발생 시 신고내용에 일부 허위로 신고된 경우 신고내용은 수정해도 오피셜서버로 등록됩니다.	2024-01-08 10:00:00
10	이수영	010-1234-5678	화재신고	화재신고	화재발생 시 신고내용에 일부 허위로 신고된 경우 신고내용은 수정해도 오피셜서버로 등록됩니다.	2024-01-08 10:00:00

리뷰 신고목록

등록된 배너 목록			
배너 순서	등록된 배너 제목	배너 등록일	삭제
1	여행일정 홍보	2024-01-09 11:23:13	<button>배너 삭제</button>
2	진주여행 유튜브	2024-01-10 09:22:28	<button>배너 삭제</button>
3	진주의 맛	2024-01-10 10:59:54	<button>배너 삭제</button>
4	진주청연루	2024-01-10 10:42:21	<button>배너 삭제</button>
5	<button>등록 완료</button>		

배너등록



DEVELOPMENT

UI/UX

UserInterface/UserExperience

로그인

회원가입

Login

Social Login

아이디
비밀번호
로그인



아이디 / 비밀번호 찾기 | 회원가입

회원가입

아이디
비밀번호
비밀번호 확인
이름
이메일
인증번호 찾기

가입하기

ID/PW 찾기

아이디 찾기

이메일을 입력해주세요.

이메일 주소

다음

비밀번호 찾기

아이디와 이메일을 입력해주세요.

아이디

이메일 주소

다음

DEVELOPMENT

UI/UX

UserInterface/UserExperience

마이페이지

마이페이지

이젠님의 전주여행

나의 여행일정 나의 리뷰 나의 좋아요

일정 목록 (4개)

제목	날짜	조회수
2024 새해 첫 친구들과의 여행	2024-01-01 ~ 2024-01-02	5
먹거리 여행	2024-01-25 ~ 2024-01-27	12
한자서 달일치기	2024-01-12 ~ 2024-01-12	8
엄마랑 1박2일	2024-01-11 ~ 2024-01-12	5

나의 일정

마이페이지

다른님의 전주여행

나의 여행일정 나의 리뷰 나의 좋아요

리뷰 목록 (7개)

	한국전통정원	2024-01-01	500

나의 리뷰

마이페이지

다른님의 전주여행

나의 여행일정 나의 리뷰 나의 좋아요

좋아요 목록 (8개)

	한국전통정원	500

나의 좋아요

DEVELOPMENT

UI/UX

UserInterface/UserExperience

전주에가면

The image displays a grid of 12 photographs arranged in three rows of four. Each photo is accompanied by a caption and a small icon indicating its popularity or rating.

- 전복대교** (Top of the 1st row, left): A traditional Korean bridge at night, illuminated from below. Popularity: 213, Rating: 2.
- 직진봉** (Top of the 1st row, middle): A traditional Korean building reflected in water at night. Popularity: 121, Rating: 4.
- 동국관** (Top of the 1st row, right): A large traditional Korean building during the day. Popularity: 404, Rating: 2.
- 술래자** (Second row, left): A traditional Korean gate (Nagwajeon) surrounded by greenery. Popularity: 107, Rating: 0.
- 모래밭** (Second row, middle): A traditional Korean pavilion with a tiled roof. Popularity: 121, Rating: 1.
- 화엄산 산자** (Second row, right): A large, weathered wooden cross standing in a forest. Popularity: 131, Rating: 2.
- 학부님** (Third row, left): A traditional Korean building with a tiled roof. Popularity: 107, Rating: 0.
- 한재단** (Third row, middle): A traditional Korean pavilion with a tiled roof. Popularity: 120, Rating: 1.
- 관수정** (Third row, right): A traditional Korean building with a tiled roof, surrounded by trees. Popularity: 200, Rating: 2.

DEVELOPMENT

UI/UX

UserInterface/UserExperience

전주에가면 (명소/음식)

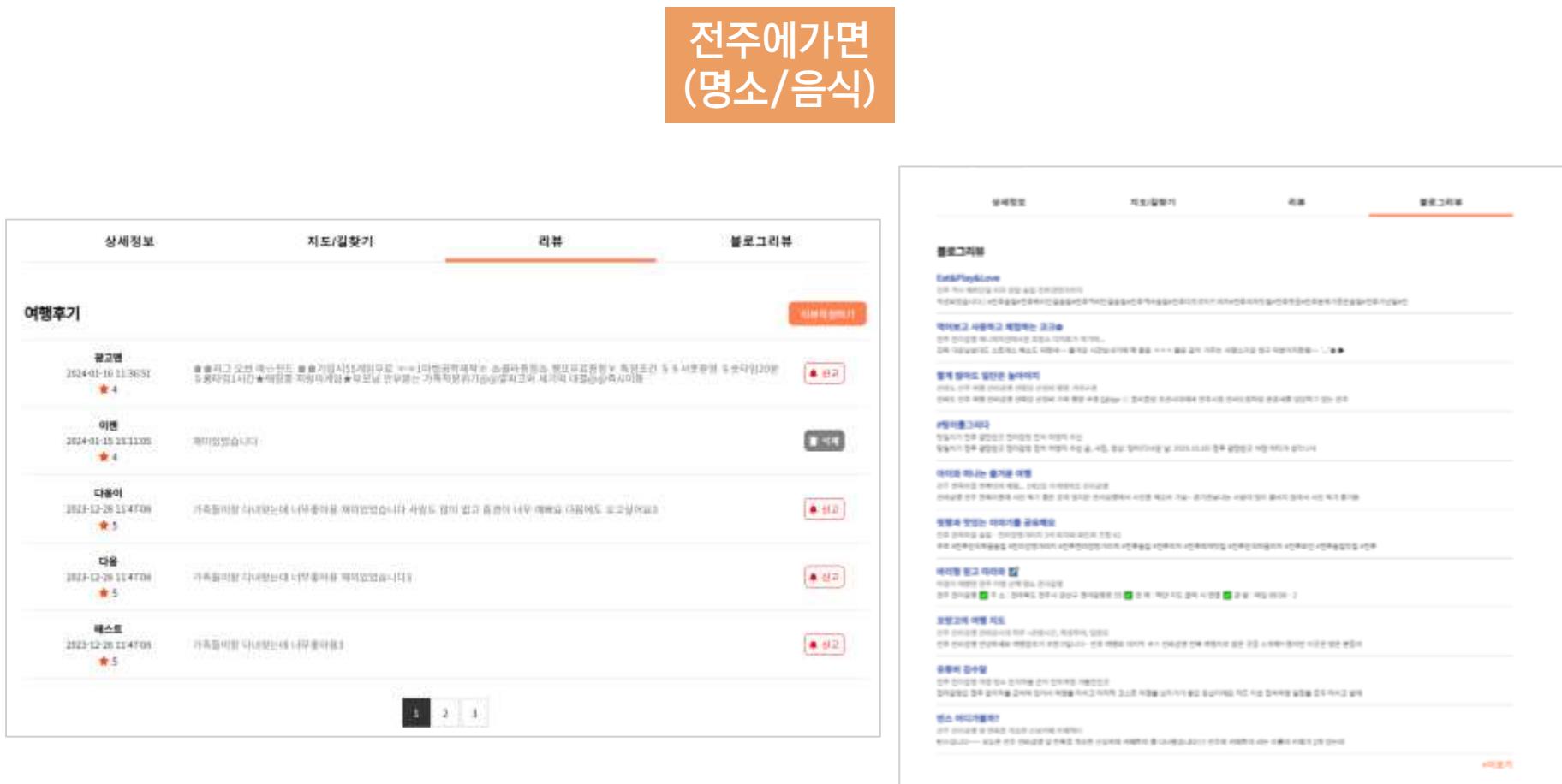
The screenshot shows a travel guide application interface. At the top, there are tabs for '기획여행', '여행지', '문화상', and '여행상'. Below this, the title '전라감영' is displayed above a large image of a traditional Korean building complex. A navigation bar at the bottom includes '날씨정보', '지도/길찾기', '리뷰', and '블로그리뷰'. Underneath the main image, there are two smaller images: one of the exterior and one of the interior of the building. At the bottom, there is a section titled '전라감영 소개' with a detailed description and a table titled '관람시간' (Viewing Hours) with specific times for different days.

The screenshot shows a map-based travel guide application. The top navigation bar includes '설정', '지도/길찾기' (selected), '리뷰', and '블로그리뷰'. The main area is a map of Gwangju with various landmarks marked by icons. Below the map, there are buttons for '길찾기' (Get Directions), '출발지' (Starting Point), '도착지' (Destination), and '전라감영' (Jeonju Gamyeong). A red '검색' (Search) button is located at the bottom right.

DEVELOPMENT

UI/UX

UserInterface/UserExperience



DEVELOPMENT

UI/UX

UserInterface/UserExperience

전주에가면



리뷰 등록



리뷰 신고

DEVELOPMENT

UI/UX

UserInterface/UserExperience

전주에가면 (영상)

EZEN JEDNU

전주에가면 여행일정 공지사항

로그마우 마이페이지

전주 영상

전체 21건

[Visiting Korea Jeonju] Trip to Korea Jeonju Episode 1.1 - Traditional Korean Dishes

전주여행이랑 함께 전주 봄 맛집여행! 보금, 찬금수, 궁극수, 탕수육, 화남제과... 오원점, 배전면 전미집, 다가당조정, 동영커피, 친구들 국수, 광복회관... 또 갈래

[Visiting Korea Jeonju] Trip to Korea Jeonju Episode 1.2 - Best 13 Travel Destinations

전주 찾을 모양다 1박 2일 푸백이 전주여행 코스 종장리 | 모집 예행자, 사진스팟 이대로만 가면 실패 없음!

[Visiting Korea Jeonju] Lovers Trip to Korea Jeonju Episode 5. Jeonju Zoo

전주 찾을 모양다 1박 2일 푸백이 전주여행 코스 종장리 | 모집 예행자, 사진스팟 이대로만 가면 실패 없음!

[Visiting Korea Jeonju] Friends Trip to Korea Jeonju Episode 2 - Late-Night Store

전주 찾을 모양다 1박 2일 푸백이 전주여행 코스 종장리 | 모집 예행자, 사진스팟 이대로만 가면 실패 없음!

[Visiting Korea Jeonju] Friends Trip to Korea Jeonju Episode 3 - Middle-aged Couple like MZ

전주 찾을 모양다 1박 2일 푸백이 전주여행 코스 종장리 | 모집 예행자, 사진스팟 이대로만 가면 실패 없음!

'60sec. of Jeonju Tour' Dog ver.

Jeonju Day 2 Marcella

전주 찾을 모양다 1박 2일 푸백이 전주여행 코스 종장리 | 모집 예행자, 사진스팟 이대로만 가면 실패 없음!

[Jeonju Tour V-log] Visiting Jeonju with PRUSA - Part 1

전주 찾을 모양다 1박 2일 푸백이 전주여행 코스 종장리 | 모집 예행자, 사진스팟 이대로만 가면 실패 없음!

[Visiting Korea Jeonju] Friends Trip to Korea Jeonju Episode 4. Nobel Chinese Restaurant

전주 찾을 모양다 1박 2일 푸백이 전주여행 코스 종장리 | 모집 예행자, 사진스팟 이대로만 가면 실패 없음!

[Jeonju People] Family Ep.4 Bibimbap House

전주 찾을 모양다 1박 2일 푸백이 전주여행 코스 종장리 | 모집 예행자, 사진스팟 이대로만 가면 실패 없음!

[Jeonju People] Family Ep.5 Teddy Island Park

전주 찾을 모양다 1박 2일 푸백이 전주여행 코스 종장리 | 모집 예행자, 사진스팟 이대로만 가면 실패 없음!

[ENG sub] [Jeonju Beep] Ep.02 Traveling Around Jeonju Ep.02 Jeonju Hanok Village

전주 찾을 모양다 1박 2일 푸백이 전주여행 코스 종장리 | 모집 예행자, 사진스팟 이대로만 가면 실패 없음!

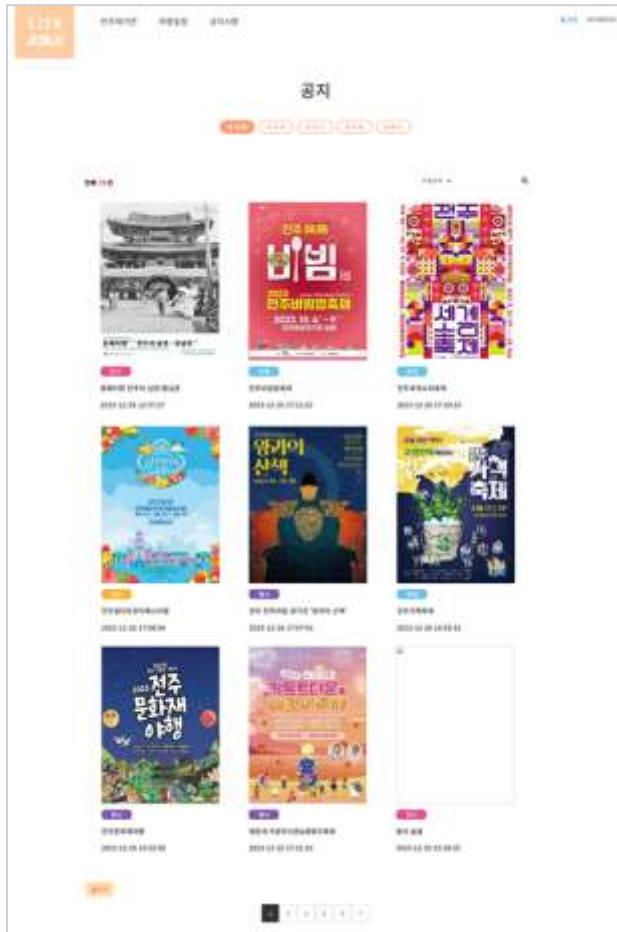
1 2

DEVELOPMENT

UI/UX

UserInterface/UserExperience

공지사항



공지목록

This screenshot shows the 'Notice Write' form on a website. At the top, there is a header with the text '공지사항 작성하기' (Write Notice) and a 'Submit' button. Below the header, there are input fields for '카테고리' (Category) set to '공지', '제목' (Title) set to '전시회관 전시회 개관 및 운영방침', '메시지' (Message) containing '[공지] 전시회 개관 및 운영방침', and '첨부파일' (Attached File) with a file named '남원 한남문.jpg'. The main area contains a rich text editor with various formatting tools. Below the editor, there is a large preview image of a traditional Korean building with people walking in front of it. At the bottom, there is a summary section with the text '공지사항' (Notice), '전시회관 전시회 개관 및 운영방침' (Exhibition Hall Exhibition Opening and Operation Policy), and a link to 'www.visitkorea.or.kr'. There are also icons for a camera and a QR code.

공지작성

DEVELOPMENT

UI/UX

UserInterface/UserExperience

공지사항

전주에가면 이행일정 공지사항

선택 풍패지향 전주의 남문:풍남문

#전시 #풍남문 #풍패지향
작성일: 2023-12-29 12:37:27



풍패지향 전주의 남문 : 풍남문
2023-12-29 ~ 2024-01-27
전시장: 전주한옥마을
작성자: 관리자

첨부파일: 풍제지향 전주의 남문_풍남문.jpg

수정하기 제출하기

공지상세

공지사항 수정하기

카테고리: 전시 | 글번호: 1234567890
제목: 풍제지향 전주의 남문 풍남문
작성자: 관리자 | 조회수: 100 | 좋아요: 0 | 댓글: 0
첨부파일: 풍제지향 전주의 남문 풍남문.jpg
작성일: 2023-12-29 12:37:27
작성자: 관리자
수정하기



공지수정

DEVELOPMENT

UI/UX

UserInterface/UserExperience

여행일정

일정 만들기

여행 일정을 만들 수 있는 화면입니다. 왼쪽에는 여행지 목록이 있고, 중앙에는 일정 테이블과 지도가 있습니다.

여행지 목록:

- 여행지 1: 2024-01-25 10:00 ~ 2024-01-26 10:00
- 여행지 2: 2024-01-26 10:00 ~ 2024-01-27 10:00
- 여행지 3: 2024-01-27 10:00 ~ 2024-01-28 10:00
- 여행지 4: 2024-01-28 10:00 ~ 2024-01-29 10:00
- 여행지 5: 2024-01-29 10:00 ~ 2024-01-30 10:00
- 여행지 6: 2024-01-30 10:00 ~ 2024-01-31 10:00
- 여행지 7: 2024-01-31 10:00 ~ 2024-02-01 10:00
- 여행지 8: 2024-02-01 10:00 ~ 2024-02-02 10:00
- 여행지 9: 2024-02-02 10:00 ~ 2024-02-03 10:00
- 여행지 10: 2024-02-03 10:00 ~ 2024-02-04 10:00
- 여행지 11: 2024-02-04 10:00 ~ 2024-02-05 10:00
- 여행지 12: 2024-02-05 10:00 ~ 2024-02-06 10:00
- 여행지 13: 2024-02-06 10:00 ~ 2024-02-07 10:00
- 여행지 14: 2024-02-07 10:00 ~ 2024-02-08 10:00
- 여행지 15: 2024-02-08 10:00 ~ 2024-02-09 10:00
- 여행지 16: 2024-02-09 10:00 ~ 2024-02-10 10:00
- 여행지 17: 2024-02-10 10:00 ~ 2024-02-11 10:00
- 여행지 18: 2024-02-11 10:00 ~ 2024-02-12 10:00
- 여행지 19: 2024-02-12 10:00 ~ 2024-02-13 10:00
- 여행지 20: 2024-02-13 10:00 ~ 2024-02-14 10:00
- 여행지 21: 2024-02-14 10:00 ~ 2024-02-15 10:00
- 여행지 22: 2024-02-15 10:00 ~ 2024-02-16 10:00
- 여행지 23: 2024-02-16 10:00 ~ 2024-02-17 10:00
- 여행지 24: 2024-02-17 10:00 ~ 2024-02-18 10:00
- 여행지 25: 2024-02-18 10:00 ~ 2024-02-19 10:00
- 여행지 26: 2024-02-19 10:00 ~ 2024-02-20 10:00
- 여행지 27: 2024-02-20 10:00 ~ 2024-02-21 10:00
- 여행지 28: 2024-02-21 10:00 ~ 2024-02-22 10:00
- 여행지 29: 2024-02-22 10:00 ~ 2024-02-23 10:00
- 여행지 30: 2024-02-23 10:00 ~ 2024-02-24 10:00
- 여행지 31: 2024-02-24 10:00 ~ 2024-02-25 10:00
- 여행지 32: 2024-02-25 10:00 ~ 2024-02-26 10:00
- 여행지 33: 2024-02-26 10:00 ~ 2024-02-27 10:00
- 여행지 34: 2024-02-27 10:00 ~ 2024-02-28 10:00
- 여행지 35: 2024-02-28 10:00 ~ 2024-02-29 10:00
- 여행지 36: 2024-02-29 10:00 ~ 2024-02-30 10:00
- 여행지 37: 2024-02-30 10:00 ~ 2024-02-31 10:00
- 여행지 38: 2024-02-31 10:00 ~ 2024-03-01 10:00
- 여행지 39: 2024-03-01 10:00 ~ 2024-03-02 10:00
- 여행지 40: 2024-03-02 10:00 ~ 2024-03-03 10:00
- 여행지 41: 2024-03-03 10:00 ~ 2024-03-04 10:00
- 여행지 42: 2024-03-04 10:00 ~ 2024-03-05 10:00
- 여행지 43: 2024-03-05 10:00 ~ 2024-03-06 10:00
- 여행지 44: 2024-03-06 10:00 ~ 2024-03-07 10:00
- 여행지 45: 2024-03-07 10:00 ~ 2024-03-08 10:00
- 여행지 46: 2024-03-08 10:00 ~ 2024-03-09 10:00
- 여행지 47: 2024-03-09 10:00 ~ 2024-03-10 10:00
- 여행지 48: 2024-03-10 10:00 ~ 2024-03-11 10:00
- 여행지 49: 2024-03-11 10:00 ~ 2024-03-12 10:00
- 여행지 50: 2024-03-12 10:00 ~ 2024-03-13 10:00
- 여행지 51: 2024-03-13 10:00 ~ 2024-03-14 10:00
- 여행지 52: 2024-03-14 10:00 ~ 2024-03-15 10:00
- 여행지 53: 2024-03-15 10:00 ~ 2024-03-16 10:00
- 여행지 54: 2024-03-16 10:00 ~ 2024-03-17 10:00
- 여행지 55: 2024-03-17 10:00 ~ 2024-03-18 10:00
- 여행지 56: 2024-03-18 10:00 ~ 2024-03-19 10:00
- 여행지 57: 2024-03-19 10:00 ~ 2024-03-20 10:00
- 여행지 58: 2024-03-20 10:00 ~ 2024-03-21 10:00
- 여행지 59: 2024-03-21 10:00 ~ 2024-03-22 10:00
- 여행지 60: 2024-03-22 10:00 ~ 2024-03-23 10:00
- 여행지 61: 2024-03-23 10:00 ~ 2024-03-24 10:00
- 여행지 62: 2024-03-24 10:00 ~ 2024-03-25 10:00
- 여행지 63: 2024-03-25 10:00 ~ 2024-03-26 10:00
- 여행지 64: 2024-03-26 10:00 ~ 2024-03-27 10:00
- 여행지 65: 2024-03-27 10:00 ~ 2024-03-28 10:00
- 여행지 66: 2024-03-28 10:00 ~ 2024-03-29 10:00
- 여행지 67: 2024-03-29 10:00 ~ 2024-03-30 10:00
- 여행지 68: 2024-03-30 10:00 ~ 2024-03-31 10:00
- 여행지 69: 2024-03-31 10:00 ~ 2024-04-01 10:00
- 여행지 70: 2024-04-01 10:00 ~ 2024-04-02 10:00
- 여행지 71: 2024-04-02 10:00 ~ 2024-04-03 10:00
- 여행지 72: 2024-04-03 10:00 ~ 2024-04-04 10:00
- 여행지 73: 2024-04-04 10:00 ~ 2024-04-05 10:00
- 여행지 74: 2024-04-05 10:00 ~ 2024-04-06 10:00
- 여행지 75: 2024-04-06 10:00 ~ 2024-04-07 10:00
- 여행지 76: 2024-04-07 10:00 ~ 2024-04-08 10:00
- 여행지 77: 2024-04-08 10:00 ~ 2024-04-09 10:00
- 여행지 78: 2024-04-09 10:00 ~ 2024-04-10 10:00
- 여행지 79: 2024-04-10 10:00 ~ 2024-04-11 10:00
- 여행지 80: 2024-04-11 10:00 ~ 2024-04-12 10:00
- 여행지 81: 2024-04-12 10:00 ~ 2024-04-13 10:00
- 여행지 82: 2024-04-13 10:00 ~ 2024-04-14 10:00
- 여행지 83: 2024-04-14 10:00 ~ 2024-04-15 10:00
- 여행지 84: 2024-04-15 10:00 ~ 2024-04-16 10:00
- 여행지 85: 2024-04-16 10:00 ~ 2024-04-17 10:00
- 여행지 86: 2024-04-17 10:00 ~ 2024-04-18 10:00
- 여행지 87: 2024-04-18 10:00 ~ 2024-04-19 10:00
- 여행지 88: 2024-04-19 10:00 ~ 2024-04-20 10:00
- 여행지 89: 2024-04-20 10:00 ~ 2024-04-21 10:00
- 여행지 90: 2024-04-21 10:00 ~ 2024-04-22 10:00
- 여행지 91: 2024-04-22 10:00 ~ 2024-04-23 10:00
- 여행지 92: 2024-04-23 10:00 ~ 2024-04-24 10:00
- 여행지 93: 2024-04-24 10:00 ~ 2024-04-25 10:00
- 여행지 94: 2024-04-25 10:00 ~ 2024-04-26 10:00
- 여행지 95: 2024-04-26 10:00 ~ 2024-04-27 10:00
- 여행지 96: 2024-04-27 10:00 ~ 2024-04-28 10:00
- 여행지 97: 2024-04-28 10:00 ~ 2024-04-29 10:00
- 여행지 98: 2024-04-29 10:00 ~ 2024-04-30 10:00
- 여행지 99: 2024-04-30 10:00 ~ 2024-05-01 10:00
- 여행지 100: 2024-05-01 10:00 ~ 2024-05-02 10:00

번호	제목	날짜	작성일
101	제주도 여행 일정	2024-01-25 ~ 2024-01-26	2024-01-25 10:28:34
102	부산 가족 주변	2024-01-26 ~ 2024-01-27	2024-01-26 17:25:31
103	내려 올정	2024-01-27 ~ 2024-01-28	2024-01-26 20:16:23:18
104	부산 달달하기 꽃축제	2024-01-28 ~ 2024-01-29	2024-01-26 21:33:00:31
105	부산 여행 일정	2024-01-29 ~ 2024-01-30	2024-01-26 22:04:28:28
106	한진주부	2024-01-30 ~ 2024-01-31	2024-01-26 22:16:29:31
107	2024 새해 첫 친구들과의 여행	2024-01-31 ~ 2024-02-01	2024-01-26 22:17:11:08
108	여기서 여행	2024-02-01 ~ 2024-02-02	2024-01-26 22:16:09:12
109	한국에서 달달하기	2024-02-02 ~ 2024-02-03	2024-01-26 21:49:34:03:01
110	한국 여행 일정	2024-02-03 ~ 2024-02-04	2024-01-26 21:48:49:48

여행 일정

여행 일정을 확인하는 화면입니다. 일정은 2024-01-25부터 2024-01-27까지입니다.

여행 일정:

날짜	제목	내용
2024-01-25	제주도 여행 일정	제주도 여행 일정
2024-01-26	부산 가족 주변	부산 가족 주변
2024-01-27	내려 올정	내려 올정

지도:



DEVELOPMENT

주요기능

Main Skill

Api 활용



로그인

KAKAO 로그인

NAVER 로그인

GOOGLE 로그인

KAKAO MAP

카카오 Map API 활용

YOUTUBE

Playlist 목록 크롤링

TourAPI

관광정보 OpenAPI

DEVELOPMENT

주요기능

Main Skill

기술 응용



보안	Spring security
쿠키	조회수 중복 방지
세션	이메일 인증
스크래핑	블로그 정보 스크래핑
필터	XSS방지, 비속어 필터

DEVELOPMENT

주요기능
Main Skill

CSS, JS 활용



슬라이더

메인페이지 슬라이드

알림창

SweetAlert

일정표

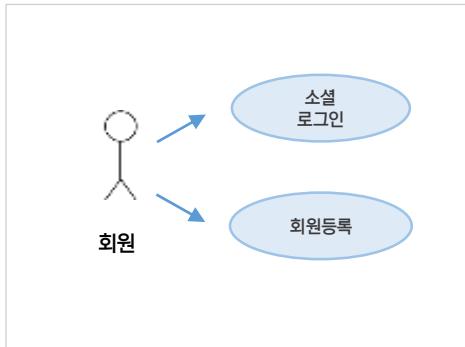
유동적 일정등록

써머노트

컨텐츠 등록

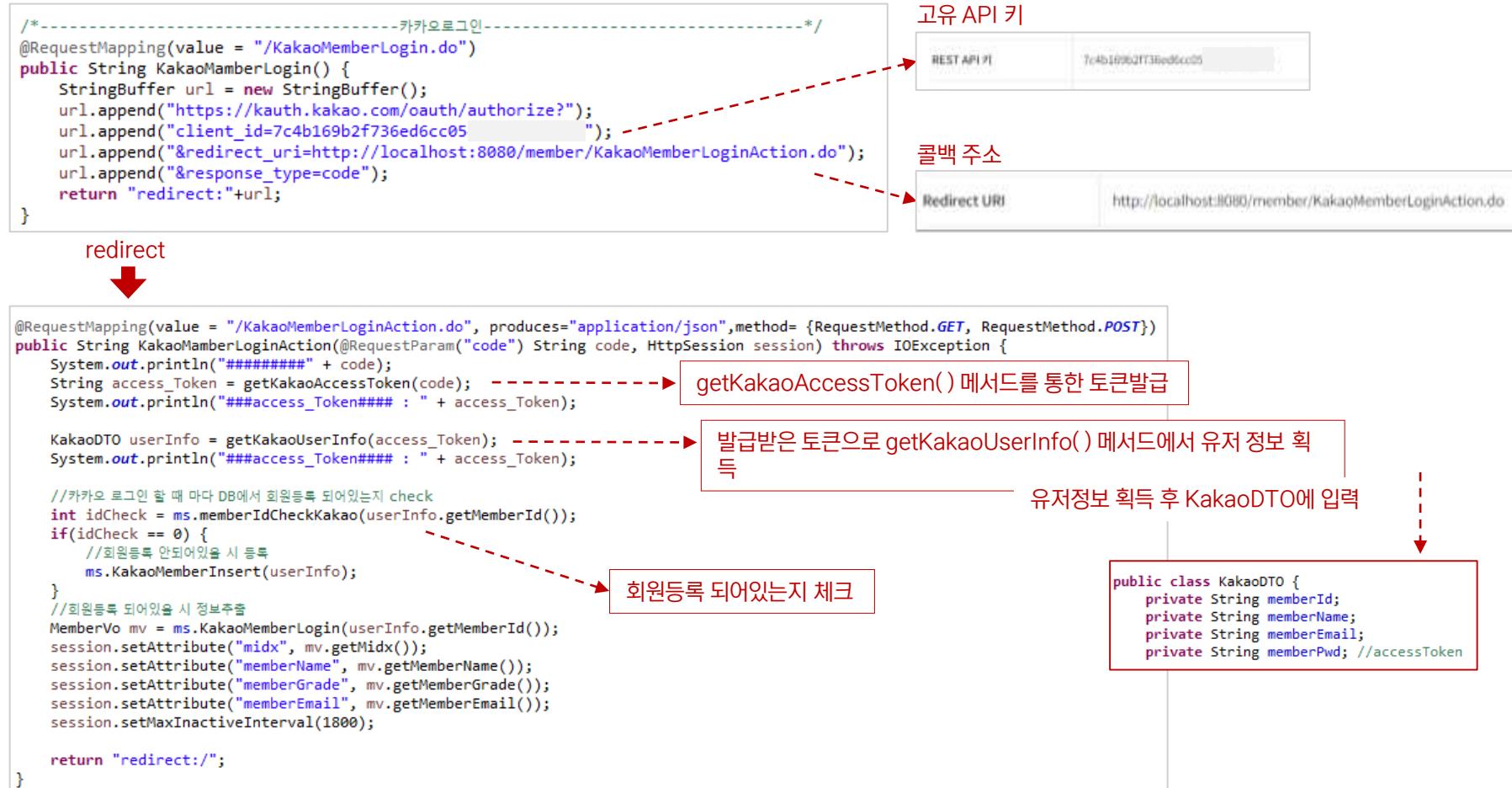
SNS 로그인

Social Login



- 각 SNS 로그인 클릭시 해당 SNS 정보를 이용해 로그인한다.
- 각 SNS에 해당하는 고유 ID가 DB에 저장되어있지 않으면 저장하여 회원등록한다.
- 저장되어 있으면 해당 member정보들을 세션에 저장하여 로그인한다.

SNS 로그인(Kakao, Naver)



DEVELOPMENT

주요기능
Main Skill

SNS 로그인(Kakao, Naver)

토큰발급을 위한 getKakaoAccessToken() 메서드

```
//토큰발급
public String getKakaoAccessToken (String authorize_code) {
    String access_Token = "";
    String refresh_Token = "";
    String reqURL = "https://kauth.kakao.com/oauth/token";

    try {
        URL url = new URL(reqURL);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();

        // URL연결은 일률적에 사용 될 수 있고, POST 혹은 PUT 요청을 하려면 setDoOutput을 true로 설정해야함.
        conn.setRequestMethod("POST");
        conn.setDoOutput(true);

        // POST 요청에 필요로 요구하는 파라미터 스트립을 통해 전송
        BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(conn.getOutputStream()));
        StringBuilder sb = new StringBuilder();
        sb.append("grant_type=authorization_code");
        sb.append("&client_id=7c4b169b2f736ed6cc05e445e9a99cf9");
        sb.append("&redirect_uri=http://localhost:8080/member/KakaoMemberLoginAction.do");
        sb.append("&code=" + authorize_code);
        bw.write(sb.toString());
        bw.flush();

        // 결과 코드가 200이라면 성공
        int responseCode = conn.getResponseCode();
        System.out.println("responseCode : " + responseCode);

        // 요청을 통해 얻은 JSON타입의 Response 메세지 읽어오기
        BufferedReader br = new BufferedReader(new InputStreamReader(conn.getInputStream()));
        String line = "";
        String result = "";

        while ((line = br.readLine()) != null) {
            result += line;
        }
        System.out.println("response body : " + result);

        // Gson 라이브러리에 포함된 클래스로 JSON파싱 객체 생성
        JsonParser parser = new JsonParser();
        JsonElement element = parser.parse(result);

        access_Token = element.getAsJsonObject().get("access_token").getAsString();
        refresh_Token = element.getAsJsonObject().get("refresh_token").getAsString();

        br.close();
        bw.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    return access_Token;
}
```

회원정보 획득을 위한 getKakaoUserInfo() 메서드

```
//유저정보
public KakaoDTO getKakaoUserInfo (String access_Token) {
    // 요청하는 클라이언트마다 가진 정보가 다를 수 있기에 HashMap타입으로 선언
    KakaoDTO userInfo = new KakaoDTO();
    String reqURL = "https://kapi.kakao.com/v2/user/me";
    try {
        URL url = new URL(reqURL);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");

        // 요청에 필요한 Header에 포함될 내용
        conn.setRequestProperty("Authorization", "Bearer " + access_Token);

        int responseCode = conn.getResponseCode();
        System.out.println("responseCode : " + responseCode);

        BufferedReader br = new BufferedReader(new InputStreamReader(conn.getInputStream()));

        String line = "";
        String result = "";

        while ((line = br.readLine()) != null) {
            result += line;
        }
        System.out.println("response body : " + result);

        JsonParser parser = new JsonParser();
        JsonElement element = parser.parse(result);

        JsonObject properties = element.getAsJsonObject().get("properties").getAsJsonObject();
        JsonObject kakao_account = element.getAsJsonObject().get("kakao_account").getAsJsonObject();
        String id = element.getAsJsonObject().get("id").getAsString();

        String nickname = properties.getAsJsonObject().get("nickname").getAsString();
        String email = kakao_account.getAsJsonObject().get("email").getAsString();

        userInfo.setMemberPwd(access_Token);
        userInfo.setMemberName(nickname);
        userInfo.setMemberEmail(email);
        userInfo.setMemberId(id);

    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    return userInfo;
}
```

DEVELOPMENT

주요기능

Main Skill

SNS 로그인(Google)

고유 API 키

클라이언트 ID

859504643059-8kf7...

```
/*
-----구글로그인-----
@RequestMapping(value="/googleMemberLogin.do")
public String googleMemberLogin() {
    StringBuffer url = new StringBuffer();
    url.append("https://accounts.google.com/o/oauth2/v2/auth?");
    url.append("client_id=859504643059-8kf7");
    url.append("&redirect_uri=http://localhost:8080/member/googleLoginAction.do&response_type=code&scope=email profile openid");
    return "redirect:"+url;
}
```

콜백 주소

redirect

```
@RequestMapping(value="googleLoginAction.do", method = RequestMethod.GET)
public String googleLoginAction(@RequestParam("code") String code, HttpSession session) {
    RestTemplate restTemplate = new RestTemplate();
    GoogleRequest googleAuthRequestParam = new GoogleRequest
        .Builder()
        .clientId("859504643059-8kf7")
        .clientSecret("GOCSPX-_RYY")
        .code(code)
        .redirectUri("http://localhost:8080/member/googleLoginAction.do")
        .grantType("authorization_code").build();

    ResponseEntity<GoogleResponse> resultEntity = restTemplate.postForEntity("https://oauth2.googleapis.com/token", googleAuthRequestParam, GoogleResponse.class);
    String jwtToken=resultEntity.getBody().getId_token();
    Map<String, String> map=new HashMap<>();
    map.put("id_token",jwtToken);
    ResponseEntity<GoogleInfResponse> resultEntity2 = restTemplate.postForEntity("https://oauth2.googleapis.com/tokeninfo", map, GoogleInfResponse.class);
    String sub = resultEntity2.getBody().getSub(); //사용자ID
    String name = resultEntity2.getBody().getName(); //사용자이름
    String email=resultEntity2.getBody().getEmail(); //사용자 이메일
    String kid = resultEntity2.getBody().getKid(); //키ID
    GoogleInfResponse googleDto = new GoogleInfResponse();
    googleDto.setSub(sub);
    googleDto.setName(name);
    googleDto.setEmail(email);
    googleDto.setKid(kid);

    int idCheck = ms.memberIdCheckGoogle(sub);
    if(idCheck==0) {
        ms.GoogleMemberInsert(googleDto);
    }
    MemberVo mv = ms.GoogleMemberLogin(sub);
    session.setAttribute("midx", mv.getMidx());
    session.setAttribute("memberName", mv.getMemberName());
    session.setAttribute("memberGrade", mv.getMemberGrade());
    session.setAttribute("memberEmail", mv.getMemberEmail());
    session.setMaxInactiveInterval(1800);

    return "redirect:/index.do";
}
```

클라이언트 보안 비밀번호
호

GOCSPX-_RYY

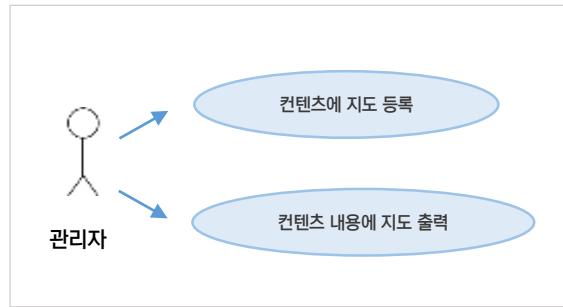
```
public class GoogleRequest {
    private String clientId;
    private String clientSecret;
    private String responseType;
    private String code;
    private String grantType;
    private String redirectUri;
}
```

```
public class GoogleResponse {
    private String access_token; // 애플리케이션이 Google API 요청을 승인하기 위해 보내는 토큰
    private String expires_in; // Access Token의 남은 수명
    private String refresh_token; // 새 액세스 토큰을 얻는 데 사용할 수 있는 토큰
    private String scope;
    private String token_type; // 반환된 토큰 유형(Bearer 고정)
    private String id_token;
}
```

```
public class GoogleInfResponse {
    private String sub; //db에서 id
    private String email;
    private String name;
    private String kid; //db에서 pw
}
```

OAuth 정책에 의해 승인되지 않은 페이지는 localhost 외의 서버 등록 불가로
다른 컴퓨터에서 실행 불가

Kakao Map API(관리자)



- 컨텐츠 작성시 해당 장소를 지도 등록하여 출력 한다.
- 지도의 기본 중심 좌표는 ‘전주시청’으로 설정한다.
- 키워드로 검색하여 장소 목록을 출력하며 마커를 생성하여 지도에 표시한다.
- 검색장소 결과 목록은 페이징 처리한다.
- 지도의 확대나 좌표가 변경됐을 때 지도 중심 좌표에 대한 주소 정보를 우측상단에 표시한다.

Kakao Map API(관리자)

발급받은 지도 API의 appkey

실제 지도를 그리는 Kakao Map API services 라이브러리 불러오기

```
<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=dbee45
<script src="../js/contentswrite-map.js"></script>
```

```
var markers = [];
var container = document.getElementById('map'); //지도를 담을 영역의 DOM 레퍼런스
var options = { //지도를 생성할 때 필요한 기본 옵션
    center: new kakao.maps.LatLng(35.82406050330023, 127.14816812319762), //지도의 중심좌표.
    level: 4 //지도의 레벨(확대, 축소 정도)
}
var map = new kakao.maps.Map(container, options); //지도 생성 및 객체 리턴
```

지도를 띄우는 코드 작성

```
//마커 이미지 설정
var markerImageUrl = 'https://cdn-icons-png.flaticon.com/512/5860/5860579.png',
markerImageSize = new kakao.maps.Size(40, 42), // 마커 이미지의 크기
markerImageOptions = {
    offset: new kakao.maps.Point(18, 40) // 마커 좌표에 일치시킬 이미지 안의 좌표
};

// 마커 이미지를 생성한다
var markerImage = new kakao.maps.MarkerImage(markerImageUrl, markerImageSize, markerImageOptions);
```

지도에 마커 생성하고 표시하는 코드작성

```
// 지도에 마커를 생성하고 표시한다
var setmarker = new kakao.maps.Marker({
    position: new kakao.maps.LatLng(35.82406050330023, 127.14816812319762), // 마커의 좌표
    image: markerImage, // 마커의 이미지
    map: map // 마커를 표시할 지도 객체
});
```

지도의 중심좌표인
'전주시청'에 마커 표시

```
// 마커 클릭 이벤트 핸들러
function onMarkerClick(marker, index) {
    setmarker.setMap(null);
    // 클릭한 마커의 정보를 출력하거나 원하는 작업을 수행합니다.
    var clickedPosition = marker.getPosition();
    var contentsLatitude = clickedPosition.getLat();
    var contentsLongitude = clickedPosition.getLng();
    setmarker.setMap(map);
    setmarker.setPosition(clickedPosition);
    $('#contentsLatitude').val(contentsLatitude);
    $('#contentsLongitude').val(contentsLongitude);
}
```

마커를 클릭하면 해당 장소의 위도와
경도 정보를 담음

Kakao Map API(관리자)

키워드로 장소를 검색하면 검색결과를 목록과 마커로 표시하는 코드 작성

```

var mapContainer = document.getElementById("map");
var coord = document.getElementById("clickLatlng");
...
// 주소-좌표 변환 객체를 생성합니다
var geocoder = new kakao.maps.services.Geocoder();
...
// 장소 검색 객체를 생성합니다
var ps = new kakao.maps.services.Places();
...
// 검색 결과 목록이나 마커를 클릭했을 때 장소명을 표출할 인포윈도우를 생성합니다
var infowindow = new kakao.maps.InfoWindow({zIndex:1});
searchPlaces();
...
// 키워드 검색을 요청하는 함수입니다
function searchPlaces() {
    ...
    var keyword = document.getElementById('keyword').value;
    ...
    if (!keyword.replace(/\s+|\s+/g, '')) {
        return false;
    }
    ...
    // 전주지역 범위 설정
    var bounds = new kakao.maps.LatLngBounds();
    ...
    new kakao.maps.LatLng(35.7936, 127.0856), // 서쪽 위 좌표
    ...
    new kakao.maps.LatLng(35.8549, 127.1921) // 동쪽 아래 좌표
    );
    ...
    // 장소검색 객체를 통해 키워드로 장소검색을 요청합니다
    ps.keywordSearch(keyword, function(data, status, pagination) {
        placesSearchCB(data, status, pagination);
    });
}

```

```

// 장소검색이 완료됐을 때 호출되는 콜백함수 입니다
function placesSearchCB(data, status, pagination) {
    if (status === kakao.maps.services.Status.OK) {
        ...
        // 정상적으로 검색이 완료됐으면
        // 검색 결과 목록과 마커를 표출합니다
        displayPlaces(data, bounds);
        ...
        // 페이지 번호를 표출합니다
        displayPagination(pagination);
    }
    ...
    // 첫 번째 장소에 포커스를 맞추고 지도 레벨을 4로 설정합니다
    if (data.length > 0) {
        var placePosition = new kakao.maps.LatLng(data[0].y, data[0].x);
        map.setCenter(placePosition);
        map.setLevel(4);
    }
}
...
} else if (status === kakao.maps.services.Status.ZERO_RESULT) {
    alert('검색 결과가 존재하지 않습니다.');
    return;
}
...
} else if (status === kakao.maps.services.Status.ERROR) {
    alert('검색 결과 중 오류가 발생했습니다.');
    return;
}
}
}

```

장소검색결과를 리스트로 출력할 때 페이징처리

검색된 장소로 센터 포커스 설정

Kakao Map API(관리자)

장소검색결과 목록 하단에 페이지번호를 표시하는 코드 작성

```
// 검색결과 목록 하단에 페이지번호를 표시하는 함수입니다.
function displayPagination(pagination) {
    var paginationEl = document.getElementById('pagination'),
        fragment = document.createDocumentFragment(),
        i;

    // 기존에 추가된 페이지번호를 삭제합니다.
    while (paginationEl.hasChildNodes()) {
        paginationEl.removeChild(paginationEl.lastChild);
    }

    for (i=1; i<=pagination.last; i++) {
        var el = document.createElement('a');
        el.href = "#";
        el.innerHTML = i;

        if (i==pagination.current) {
            el.className = 'on';
        } else {
            el.onclick = (function(i) {
                return function() {
                    pagination.gotoPage(i);
                }
            })(i);
        }

        fragment.appendChild(el);
    }
    paginationEl.appendChild(fragment);
}
```

지도 우측상단에 중심좌표에 대한 주소정보를 표시하는 코드 작성

```
// 중심 좌표나 확대 수준이 변경됐을 때 지도 중심좌표에 대한 주소 정보를 표시하도록 이벤트를 등록합니다.
kakao.maps.event.addListener(map, 'idle', function() {
    searchAddrFromCoords(map.getCenter(), displayCenterInfo);
});

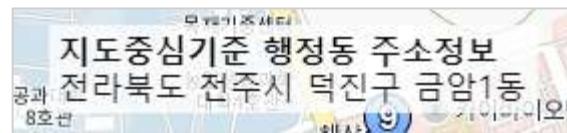
function searchAddrFromCoords(coords, callback) {
    // 좌표로 행정동 주소 정보를 요청합니다.
    geocoder.coord2RegionCode(coords.getLng(), coords.getLat(), callback);
}

function searchDetailAddrFromCoords(coords, callback) {
    // 좌표로 법정동 상세 주소 정보를 요청합니다.
    geocoder.coord2Address(coords.getLng(), coords.getLat(), callback);
}

// 지도 우측상단에 지도 중심좌표에 대한 주소정보를 표출하는 함수입니다.
function displayCenterInfo(result, status) {
    if (status === kakao.maps.services.Status.OK) {
        var infoDiv = document.getElementById('centerAddr');

        for(var i = 0; i < result.length; i++) {
            // 행정동의 region_type 값은 'H' 이므로
            if (result[i].region_type === 'H') {
                infoDiv.innerHTML = result[i].address_name;
                break;
            }
        }
    }
}
```

지도의 중심좌표나 확대수준이 변경될 때마다
Geocoder객체의 coord2addr 메소드를 이용해
지도중심 좌표에 대한 행정동 주소를 검색



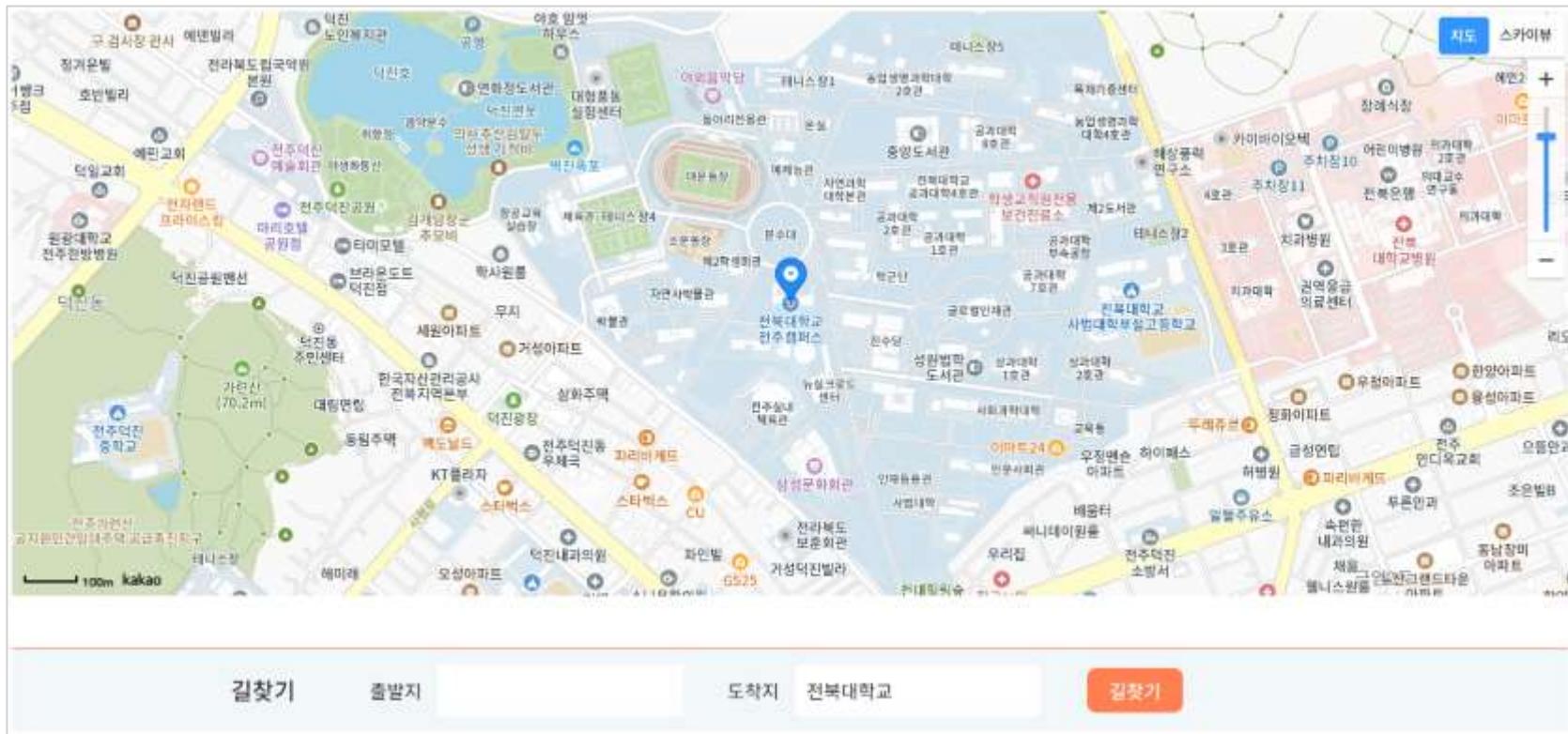
DEVELOPMENT

주요기능
Main Skill

Kakao Map API(회원)



- 해당 컨텐츠 장소 위치를 마커로 표시한다.
- 지도 탑입을 전환할 수 있는 지도타입 컨트롤을 생성한다.
- 해당 컨텐츠 장소가 도착지로 설정되어 길찾기 버튼을 클릭하면 카카오맵으로 이동한다.



Kakao Map API(회원)

```
<script>
  var container = document.getElementById('map'); //지도를 담을 영역의 DOM 레퍼런스
  var latitude = document.getElementById('latitude').value;
  var longitude = document.getElementById('longitude').value;
  var options = { //지도를 생성할 때 필요한 기본 옵션
    center: new kakao.maps.LatLng(latitude, longitude), //지도의 중심좌표
    level: 4 //지도의 레벨(확대, 축소 정도)
  };
  var map = new kakao.maps.Map(container, options);

```

지도 출력

```
// 마커가 표시될 위치입니다.
var markerPosition = new kakao.maps.LatLng(latitude, longitude);

// 마커를 생성합니다
var marker = new kakao.maps.Marker({
  position: markerPosition
});
marker.setMap(map);
```

해당 장소 지도 위에 마커 표시

```
// 일반 지도와 스카이뷰로 지도 타입을 전환할 수 있는 지도타입 컨트롤을 생성합니다
var mapTypeControl = new kakao.maps.MapTypeControl();
map.addControl(mapTypeControl, kakao.maps.ControlPosition.TOPRIGHT);

// 지도 확대 축소를 제어할 수 있는 줌 컨트롤을 생성합니다
var zoomControl = new kakao.maps.ZoomControl();
map.addControl(zoomControl, kakao.maps.ControlPosition.RIGHT);
```

우측 상단에 지도 타입과 확대 축소를 제어 하는 컨트롤 생성

```
function relayout() {
  // 지도를 표시하는 div 크기를 변경한 이후 지도가 정상적으로 표출되지 않을 수도 있습니다
  // 크기를 변경한 이후에는 반드시 map.relayout() 함수를 호출해야 합니다
  // window의 resize 이벤트에 의한 크기변경은 map.relayout() 함수가 자동으로 호출됩니다
  map.relayout();
  map.setCenter(new kakao.maps.LatLng(latitude, longitude));
}

</script>
```

지도/길찾기 템을 클릭하여 조회할 때
정상적으로 지도를 출력하기 위해 relayout 함수 사용

```
/* 지도크기 재설정 */
$('#tab2').click(function(){
  relayout();
});
```

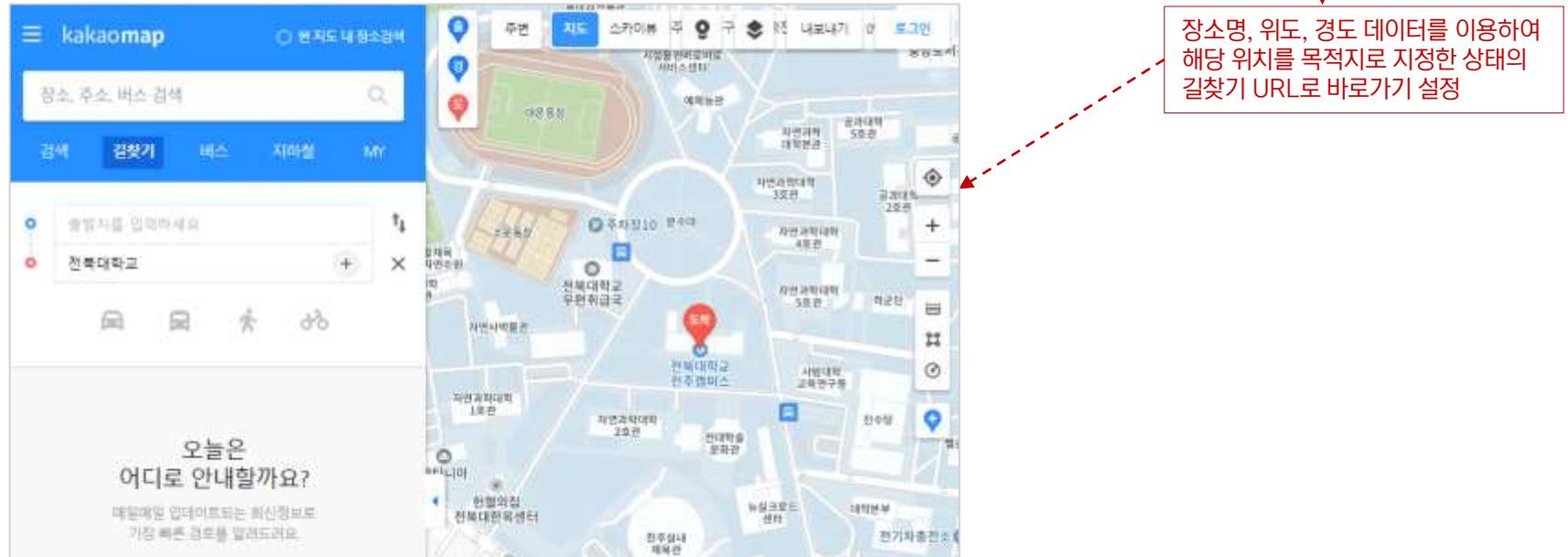
DEVELOPMENT

주요기능
Main Skill

Kakao Map API(회원)

```
<div>.class="loadSearch">
  <div>.class="inner">
    <strong>.class="title">길찾기</strong>
    <div>.class="loadPutBox">
      <div>.class="start">
        <span>출발지</span>
        <input type="text" id="findPath_start" class="loadInput">
      </div>
      <div>.class="end">
        <span>도착지</span>
        <input type="text" id="findPath_end" class="loadInput" value="${cv.contentsSubject}">
      </div>
    </div>
    <div>.class="LoadBtn">
      <a href="https://map.kakao.com/link/to/${cv.contentsSubject},${cv.contentsLatitude},${cv.contentsLongitude}" target="_blank">길찾기</a>
    </div>
  </div>
</div>
```

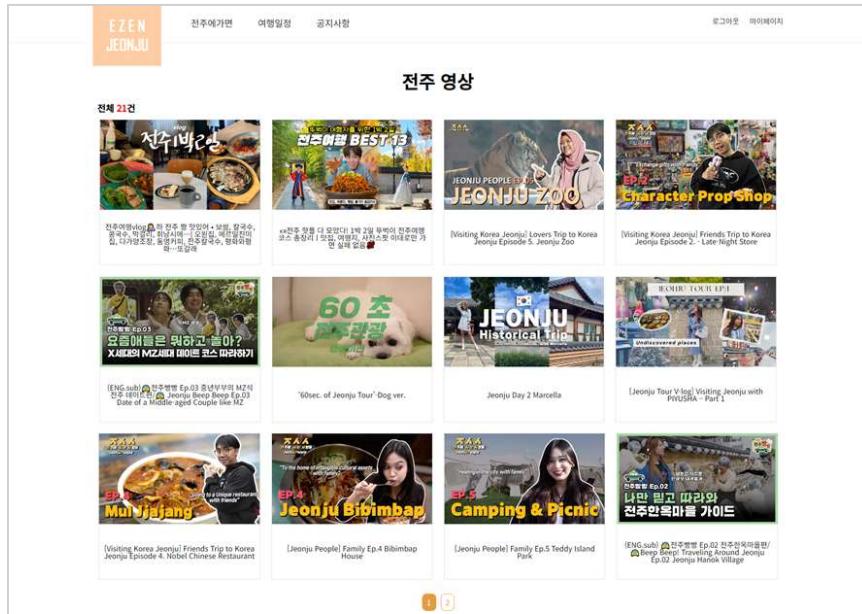
해당 컨텐츠의 장소명을 도착지로 출력



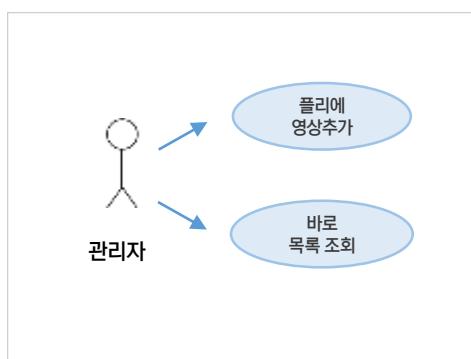
DEVELOPMENT

주요기능
Main Skill

YouTube Data API (Playlist 크롤링)



- 유튜브 특정 계정의 플레이리스트에 영상을 추가하고 그 영상들을 가져와 출력한다.
- 플레이리스트에 추가하는 영상은 바로 목록에 반영되도록 한다.
- 목록 출력 순서는 최근 추가된 영상 순으로 한다.
- 영상은 한 페이지에 12개씩 출력한다.



YouTube 플레이리스트



DEVELOPMENT

주요기능
Main Skill

YouTube Data API (Playlist 크롤링)

Google api 활용을 위한 javascript 라이브러리

```
<script src="https://apis.google.com/js/api.js"></script>
```

고유 API 키

API Key
AIzaSyBpmMGQU3g

```
const API_KEY = 'AIzaSyBpmMGQU3g';  
  
// 플레이언트 라이브러리 초기화  
gapi.load('client', init);  
  
function init() {  
    // YouTube API 플레이언트 초기화  
    gapi.client.init({  
        apiKey: API_KEY,  
        discoveryDocs: ["https://www.googleapis.com/discovery/v1/apis/youtube/v3/rest"],  
    }).then(() => {  
        // 최초 데이터 호출  
        getPlaylistItems();  
    });  
}
```

```
function getPlaylistItems() {  
    // 플레이리스트 아이디 설정  
    const PLAYLIST_ID = 'PL1GT5AS9KmUeXAxHkPeW3-fyFj9pKL000';  
    const maxResults = 50;  
    // API 호출하여 플레이리스트의 아이템 목록 가져오기  
    gapi.client.youtube.playlistItems.list({  
        part: 'snippet',  
        playlistId: PLAYLIST_ID,  
        maxResults: maxResults,  
        order: 'date',  
    }).then(response => {  
        const playlistItems = response.result.items;  
  
        if (playlistItems) {  
            playlistItems.forEach(item => {  
                console.log(item.snippet.publishedAt);  
            });  
        }  
    });  
}
```

가져오고자 하는 플레이리스트의 고유 아이디

가져올 최대 영상 갯수

최신순으로 출력하기 위해 order: 'date' 로 설정

가져온 영상이 있으면 if 구문 내의 코드 실행

YouTube Data API (Playlist 크롤링)

```

let tableRows = '';
let TableCell = '';
var size = playlistItems.length; // 가져온 총 영상 개수
var total_cnt = "<span>" +size+ "</span>건";
$('#total_cnt').append(total_cnt);
console.log("size: "+size);
var totalpage = Math.ceil(size/12); // 생성될 총 페이지 -----> 페이지를 위한 변수들
console.log("totalpage: "+totalpage);
var startnum = (page - 1)*12; // 각 페이지별 시작할 영상index
var endnum = startnum + 11;
for (let i = startnum; i <= endnum; i++) {
    if (i < size) {
        const playlistItem = playlistItems[i];
        // 각 아이템에서 제목, 썸네일 및 영상 링크 가져오기
        const videoTitle = playlistItem.snippet.title; //제목
        const thumbnailUrl = playlistItem.snippetthumbnails.maxres.url; //썸네일
        const videoId = playlistItem.snippet resourceId.videoId; //영상id
        const videoLink = 'https://www.youtube.com/watch?v=' + videoId; //영상링크

        // 테이블에 추가할 HTML 생성
        TableCell += '<td><div><a href="' + videoLink + '" target="_blank">' +
                     '</a><p>' + videoTitle +
                     '</p></div></td>';
        if ((i+1) % 4 === 0) {
            // 각 행의 시작에 새로운 <tr> 추가
            tableRows += '<tr>' + TableCell + '</tr>';
            TableCell = '';// 행 초기화
        }
        else{
            break;
        }
    }
}

// 마지막 행에 추가된 셀이 있을 경우 추가
if (TableCell !== '') {
    tableRows += '<tr>' + TableCell + '</tr>';
}

// 테이블에 동영상 추가
$('#results').html(tableRows); -----> generatePagination(totalpage);

```

페이지를 위한 변수들

```

<td>
  <div>
    <a href="https://www.youtube.com/watch?v=8JPk6bzSOsI" target="_blank">
      
    </a>
    <p>Jeonju Day 2 Marcella</p>
  </div>
</td>

```

tableCell로 생성되는 <td>

```

<div class="innerList">
  <div class="total_cnt" id="total_cnt">전체 </div>
  <table class="video-table" id="results"></table>
  <div class="innerwrap">
    <div class="pagination" id="pagination"></div>
  </div>
</div>

```

생성된 <tr><td>를 table에 삽입

YouTube Data API (Playlist 크롤링)

페이지 생성 함수

```
function generatePagination(totalpage) {
    const paginationElement = $('#pagination');
    paginationElement.empty();
    var pageLink = "";
    for (let i = 1; i <= totalpage; i++) {
        if(page==i){
            pageLink = '<a class="active" href="<%=request.getContextPath()%>/contents/youtube.do?page=' + i + '>' + i + '</a>';
        }else{
            pageLink = '<a href="<%=request.getContextPath()%>/contents/youtube.do?page=' + i + '>' + i + '</a>';
        }
        paginationElement.append(pageLink);
    }
}
```

```
<div class="innerList">
    <div class="total_cnt" id="total_cnt">전체 </div>
    <table class="video-table" id="results"></table>
    <div class="innerwrap">
        <div class="pagination" id="pagination"></div>
    </div>
</div>
```

생성된 <a>목록을 <div>에 삽입

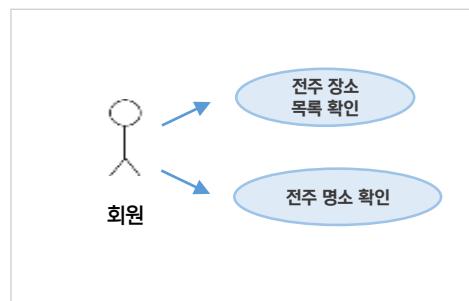
```
<a class="pagination" id="pagination">
    <a class="active" href="/contents/youtube.do?page=1">1</a>
    <a href="/contents/youtube.do?page=2">2</a>
</a>
```

생성된 페이지 목록

공공데이터 API

The screenshots show three different search results pages:

- 1/13**: A list of local guidebooks (지도) from various museums and historical sites.
- 1/2**: A list of recommended places (추천 장소) including a traditional Korean restaurant (한옥마을) and a general store (경기전).
- 1/1**: A list of recommended food places (맛집 추천) including traditional Korean restaurants (한옥마을), general stores (경기전), and specialty stores (역전공원). The results are sorted by popularity (별점).



- 일정 등록할 때 추천 장소, 명소, 음식점들이 출력된다.
- 음식점과 관광지는 장소의 이미지를 제공하고 이미지가 없을 시 다른 이미지로 대체한다.(API)
- 맛집추천은 별점이 4점이상인 전주 음식점들만 리스트에 출력한다. (네이버 인기도 API)
- 지도를 누르면 공공데이터 API에서 가져온 위도, 경도를 이용해 카카오맵 API와 같이 사용해서 지도 자세히보기 목록을 출력한다.
- 명소 추천은 공공데이터에 제공신청한 전주 명소 리스트를 공공데이터가 제시해준 순서대로 랭킹을 부여한다.

공공데이터 API(명소)

처리상태	신청 공공데이터 명칭	기관담당자	접수기관	신청일자	처리일자
제공	전라북도 전주시 관광 명소	신영술	전라북도 전...	2023-12-04	2023-12-15

공공데이터 포털에 제공 신청한 API

```

function getPlace() {
    event.preventDefault();
    let obj = document.getElementById("jj");
    let existingPs = obj.querySelectorAll('p');
    let top = 0;
    existingPs.forEach((p) => {
        p.remove();
    });
    let startIdx = (currentPagePlace - 1) * itemsPerPage;
    let endIdx = startIdx + itemsPerPage;
}

fetch("https://api.odcloud.kr/api/15125431/v1/uddi:3f24f89a-940d-4512-88d7-80b993cd28b8?page=" + currentPagePlace + "&perPage=10&serviceKey=%2BUFeyGq0"
    .then((response) => response.json())
    .then((data) => {
        data.data.slice(startIdx, endIdx).forEach((place) => {
            top++;
            let newP = document.createElement("p");
            let placeName = place['관광명소명'];
            let placelatitude = place['위도'];
            let placelongitude = place['경도'];
            newP.innerHTML = "<span id='top'>Top " + top + " : &ampnbsp</span>";
            newP.innerHTML += "<a href='javascript:addToTable(\"" + placeName + "," + placelatitude + "," + placelongitude + "\")'" + placeName + "</a>";
            newP.innerHTML += "&ampnbsp&ampnbsp&ampnbsp" + '<a href=' + 'https://map.kakao.com/link/map/' + placeName + ',' + placelatitude + ',' + placelongitude + '>' + placeName + '</a>';
            obj.appendChild(newP);
        });
    })
    .then(() => { // fetch가 완료된 후에 실행되도록 이동
        $('#pageIndexPlace').text(currentPagePlace + " / 1");
        $('#placeBtn').addClass('active');
        $('#foodBtn').removeClass('active');
        $('#tourBtn').removeClass('active');
        $('#restBtn').removeClass('active');
    })
}

```

페이지 번수

Javascript fetch로 api 가져옴
currentPagePlace => 가져온 데이터들을
페이지하기 위해 변수를 둠,
perPage=10 데이터가 10개씩 출력,
serviceKey = api 인증키

가져온 data들을 페이지 처리를 위해
slice, forEach문을 사용함.
JSON 형식으로 가져온 api 중
관광명소명, 위도, 경도를 key값으로
넣어 value를 변수로 등록해 가져옴

DEVELOPMENT

주요기능
Main Skill

공공데이터 API(음식추천)

활용신청 [승인] 전라북도 전주시_음식점기본정보

계정 개발 신청일 2023-12-04

```
function getFood() {
    event.preventDefault();
    let obj = document.getElementById("jj");
    let existingPs = obj.querySelectorAll('p');
    existingPs.forEach((p) => {
        p.remove();
    });

    let startIdx = (currentPageFood - 1) * itemsPerPage;
    let endIdx = startIdx + itemsPerPage;

    fetch("https://api.odcloud.kr/api/15076735/v1/uddi:98edad48-0892-4621-8741-cdff64f99c79?page=" + currentPageFood + "&perPage=" + perPage + "&serviceKey"
        .then((response) => response.json())
        .then((data) => {
            let List = "<ul>";
            let filteredData = data.data.filter((place) => place['네이버 인기도'] > 4.0);

            // 필터된 데이터의 갯수를 가져오기
            filteredDataCnt = filteredData.length;

            filteredData.slice(startIdx, endIdx).forEach((place) => {
                let newP = document.createElement("p");
                let placeName = place['식당명'];
                let placelatitude = place['식당위도'];
                let placelongitude = place['식당경도'];
                let placeScore = Math.floor(place['네이버 인기도']); // 정수로 내림

                // 별 등수문자로 갯수 표현
                let stars = "*".repeat(placeScore);
                let emptyStars = "*".repeat(5 - placeScore); // 비어있는 별

                newP.innerHTML = "<a href='javascript:addToTable(\"" + placeName + "," + placelatitude + "," + placelongitude + "\")'>" + placeName + "</a>";
                newP.innerHTML += "&nbsp;&nbsp;&nbsp;";
                newP.innerHTML += "별점 : <span id='stars'>" + stars + emptyStars + "</span>(" + place['네이버 인기도']+ ")";
                newP.innerHTML += "&nbsp;&nbsp;&nbsp;";
                newP.innerHTML += '<a href=\'' + https://map.kakao.com/link/map/' + placeName + ',' + placelatitude + ',' + placelongitude + '\'\>지도</a>';

                obj.appendChild(newP);
            });
        });
}
```

[네이버인기도]라는 키값을 이용해 4점이상의 데이터만 출력,
실수인 value값을 이용해 별이 출력되고 네이버인기도가 같이 출력되게 함

DEVELOPMENT

주요기능
Main Skill

공공데이터 API(관광지, 음식점)

활용신청 [승인] 한국관광공사_국문 관광정보 서비스_GW

계정 개발 신청일 2024-01-04 만료예정일 2026-01-04

```
function getTour() {
    event.preventDefault();
    let obj = document.getElementById("jj");
    let existingPs = obj.querySelectorAll('p');
    existingPs.forEach((p) => {
        p.remove();
    });

    let startIdx = (currentPageTour - 1) * itemsPerPage;
    let endIdx = startIdx + itemsPerPage;

    fetch("https://apis.data.go.kr/B551011/KorService1/areaBasedList1?numOfRows=82&pageNo=1&MobileOS=ETC&MobileApp=AppTest&ServiceKey=%2BUFeyGq0yCyRGAnfn2BZHmp")
        .then((response) => response.json())
        .then((data) => {
            // items -> item 배열을 가져와서 forEach
            data.response.body.items.item.slice(startIdx, endIdx).forEach((place) => {
                let newP = document.createElement("p");
                let newImage = document.createElement("img");
                newImage.classList.add("placeimg");
                let placeName = place['title'];
                let placelatitude = place['mapy'];
                let placelongitude = place['mapx'];
                let placeImage = place['firstimage'];
                if (placeImage) {
                    newImage.src = placeImage;
                } else {
                    // 사진없을 때
                    newImage.src = 'https://via.placeholder.com/100x100/CCCCCC/FFFFFF?text>No+Image';
                }

                newP.appendChild(newImage);
                newP.innerHTML += " &ampnbsp&ampnbsp" +
                    '<a href="javascript:addToTable(\\"' + placeName + "," + placelatitude + "," + placelongitude + '\")">' + placeName + "</a>" +
                    "&ampnbsp&ampnbsp&ampnbsp" +
                    '<a href="\https://map.kakao.com/link/map/' + placeName + ',' + placelatitude + ',' + placelongitude + '\">' + placeName + "</a>"';

                obj.appendChild(newP);
            });
        });
}
```

[firstimage]라는 키값을 이용해 대표이미지를 같이 출력,
이미지의 value가 존재하지 않으면 noImage가 출력되게 placeholder 이용

Spring security

pom.xml – security를 사용하기 위한 의존성 추가

```
<!-- ***** 비밀번호 암호화 ***** -->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-core</artifactId>
    <version>5.2.1.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-web</artifactId>
    <version>5.2.1.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-config</artifactId>
    <version>5.2.1.RELEASE</version>
</dependency>
```

web.xml – xml파일 로드

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
        classpath:config/root-context.xml
        classpath:config/security.xml
    </param-value>
</context-param>
```

config/security.xml – 빈 정의

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:security="http://www.springframework.org/schema/security"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/security https://www.springframework.org/schema/security/spring-security.xsd">

    <!-- bcryptPasswordEncoder -->
    <bean id="bcryptPasswordEncoder"
        class="org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder"/>
</beans>
```

비밀번호를 안전하게 저장하기 위해 사용되는 해시 함수

Spring security

MemberController.java – memberJoinAction()

```
if(authSession!=null && authSession.equals("pass")) {
    String memberPwd2 = bcryptPasswordEncoder.encode(mv.getMemberPwd());
    mv.setMemberPwd(memberPwd2);
```

```
public String encode(CharSequence rawPassword) {
    String salt;
    if (random != null) {
        salt = BCrypt.gensalt(version.getVersion(), strength, random);
    } else {
        salt = BCrypt.gensalt(version.getVersion(), strength);
    }
    return BCrypt.hashpw(rawPassword.toString(), salt);
```

솔트 생성 후 비밀번호 해시화(높은 강도)
후에 해시된 비밀번호 반환

MemberController.java – memberLoginAction()

```
if(mv!=null&&bcryptPasswordEncoder.matches(memberPwd, mv.getMemberPwd())) {
    session.setAttribute("midx",mv.getMidx());
    session.setAttribute("memberName", mv.getMemberName());
    session.setAttribute("memberGrade", mv.getMemberGrade());
    session.setMaxInactiveInterval(1800);
```

```
public boolean matches(CharSequence rawPassword, String encodedPassword) {
    if (encodedPassword == null || encodedPassword.length() == 0) {
        logger.warn("Empty encoded password");
        return false;
    }

    if (!BCRYPT_PATTERN.matcher(encodedPassword).matches()) {
        logger.warn("Encoded password does not look like BCrypt");
        return false;
    }

    return BCrypt.checkpw(rawPassword.toString(), encodedPassword);
}
```

입력된 비밀번호가 Bcrypt 해시 패턴에 맞는지 검사 후 실제로
평문 비밀번호와 해시된 비밀번호가 일치하는지 확인

쿠키를 사용한 조회수 중복 방지

컨텐츠에 이동

```
@RequestMapping(value = "/contentsArticle.do")
public String contentsArticle(@RequestParam("cidx") int cidx, Model model, HttpServletRequest req, HttpServletResponse res) throws Exception {
    viewCountUp(cidx, req, res);
    ContentsVo cv = cs.contentsArticle(cidx);
    String boardList = cv.getContentsBoardList();
}
↓ 쿠키 체크/입력하는 메서드
```

```
private void viewCountUp(int cidx, HttpServletRequest req, HttpServletResponse res) {
    Cookie oldCookie = null;

    Cookie[] cookies = req.getCookies();
    if (cookies != null) {
        for (Cookie cookie : cookies) {
            if (cookie.getName().equals("boardView")) { -----> 쿠키를 전부 가져와서 "boardView" 쿠키가 있는지 체크
                oldCookie = cookie;
            }
        }
    }

    if (oldCookie != null) {
        if (!oldCookie.getValue().contains("[" + Integer.toString(cidx) + "]")) { -----> 해당 cidx값을 가진 쿠키가 없으면
            cs.contentsViewCountUpdate(cidx);
            oldCookie.setValue(oldCookie.getValue() + "[" + cidx + "]");
            oldCookie.setPath("/");
            oldCookie.setMaxAge(60 * 60 * 24);
            res.addCookie(oldCookie);
        } else {
            System.out.println("쿠키있음");
        }
    } else {
        cs.contentsViewCountUpdate(cidx);
        Cookie newCookie = new Cookie("boardView", "[" + cidx + "]");
        newCookie.setPath("/");
        newCookie.setMaxAge(60 * 60 * 24);
        res.addCookie(newCookie);
    }
}
```

쿠키를 전부 가져와서 "boardView" 쿠키가 있는지 체크

해당 cidx값을 가진 쿠키가 없으면
contentsViewCountUpdate(cidx) 메서드 실행 후에
해당 쿠키 생성

이름	값	Domain	Path	Expires / Max-Age	크기
boardView	[82]	192.168.0.30	/	2024-01-12T06:35:54.359Z	13

쿠키값은 하루동안 저장됨

DEVELOPMENT

주요기능
Main Skill

세션을 사용한 이메일 인증

회원가입

이메일

ezen@naver.com

인증하기

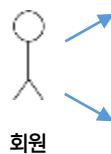
비밀번호 찾기

인증번호 확인

이메일로 전송된 인증번호를 입력해주세요.

[] 남은시간: 178초

확인



- 이메일 입력 후 해당 메일로 인증번호를 보낸다.
- 인증번호 입력란에 인증번호를 입력한 후 확인 버튼을 눌러 인증완료한다.
- 인증번호는 세션에 180초동안 저장한다.

DEVELOPMENT

주요기능
Main Skill

세션을 사용한 이메일 인증

이메일

ekdhs@naver.com

인증하기

확인 남은시간: 157초

메일인증을 바로 받기위해 Ajax 활용

```
function mailAuth(){
    var memberEmail = $('input[name="memberEmail"]').val();
    if(!CheckEmail(memberEmail)){
        alert("이메일 형식이 유효하지 않습니다.");
        return;
    }
    $.ajax({
        type: "post",
        url: "<%=request.getContextPath()%>/member/mailAuth.do",
        data: {"memberEmail" : memberEmail},
        dataType: "json",
        success: function(data){
            alert("인증번호가 전송되었습니다");
            $('#authNum').css("display", "block");
        },
        error: function(){
            alert("실패");
        }
    });
    updateSessionTimer(); // 초기화
    // 1초마다 업데이트
    var timerInterval = setInterval(function () {
        updateSessionTimer();
    }, 1000);
}
```

```
@ResponseBody
@RequestMapping(value="/mailAuth.do")
public JSONObject mailAuth(@RequestParam("memberEmail") String memberEmail, HttpSession session) {
    JSONObject js = new JSONObject();
    NaverMailSend nm = new NaverMailSend();
    try {
        nm.sendEmail(memberEmail, session);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return js;
```

NaverMailSend 클래스의 sendEmail 메서드를 이용해 메일 전송

세션에 저장되는 시간(180초) 타이머

```
function updateSessionTimer() {
    console.log('updateSessionTimer 호출'); // 디버깅 메시지 추가
    var sessionTimeOut = $('#sessionTimeOut').val();
    var sessionTimerElement = $('#sessionTimer');
    var txt = "남은시간: " + sessionTimeOut + "초";
    // 세션 타이머 값이 0 이상일 때만 업데이트
    if (sessionTimeOut > 0) {
        sessionTimerElement.html(txt);
        $('#sessionTimeOut').val(sessionTimeOut-1);
    } else {
        // 세션이 만료되었을 때의 처리를 추가할 수 있습니다.
        sessionTimerElement.html('세션이 만료되었습니다.');
        clearInterval(timerInterval); // 세션이 만료되면 타이머 중지
    }
}
```

DEVELOPMENT

주요기능
Main Skill

세션을 사용한 이메일 인증

```

public final class NaverMailSend {
    private final String host = "smtp.naver.com";           // SMTP 서버명
    private final String user = "ekdh...";                  // 발신자의 이메일 계정
    private final String password = "5KNG...";             // 발신자의 SMTP 패스워드

    public String sendEmail(String to, HttpSession forsession) throws Exception {
        String authenCode = null;

        /* Property 객체에 SMTP 서버 정보 설정 */
        Properties props = new Properties();
        props.put("mail.smtp.host", host);
        props.put("mail.smtp.port", "465");          // SMTP 포트
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.ssl.enable", "true");
        props.put("mail.smtp.ssl.trust", host);

        /* SMTP 서버 정보와 사용자 정보를 기반으로 Session 클래스의 인스턴스를 생성*/
        Session session = Session.getDefaultInstance(props, new Authenticator() {
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(user, password);
            }
        });

        /* Message 객체에 수신자와 내용, 제목의 메시지를 작성 */
        try {
            // 인증번호 생성
            authenCode = makeAuthenticationCode(); -> makeAuthenticationCode() 메소드로 인증번호 생성

            Message message = new MimeMessage(session);

            // 발신자 설정
            message.setFrom(new InternetAddress(user, "Ezen_Jeonju"));

            // 수신자 메일주소 설정
            message.addRecipient(Message.RecipientType.TO, new InternetAddress(to));

            // 메일제목 설정
            message.setSubject("Ezen_Jeonju :: 이메일 인증번호입니다.");

            // 메일 내용 설정
            message.setText("이메일 인증번호는 [ " + authenCode + " ] 입니다.");

            // Send the message
            Transport.send(message); -> Transport.send(message) 메소드로 메일 전송

            // 세션에 authenCode 저장
            forsession.setAttribute("authenCode", authenCode);
            forsession.setMaxInactiveInterval(180); //세션 시간 (초)

        } catch (MessagingException e) {
            e.printStackTrace();
        }

        return authenCode;
    }
}

```

```

private String makeAuthenticationCode() throws Exception {
    int pwdLength = 8;
    final char[] pwdTable = {
        'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
        'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X',
        'Y', 'Z', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j',
        'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
        'w', 'x', 'y', 'z', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0' };

    // System.currentTimeMillis(): 중복 방지 처리
    Random ran = new Random(System.currentTimeMillis());

    StringBuffer bf = new StringBuffer();
    for(int i=0; i<pwdLength; i++) {
        bf.append(pwdTable[ran.nextInt(pwdTable.length)]);
    }

    return bf.toString();
}

```

전송된 메일

★ Ezen_Jeonju :: 이메일 인증번호입니다. ↗

▼ 보낸사람 Ezen_Jeonju <ekdl...>

2024년 1월 11일 (목) 오후 3:54

이메일 인증번호는 [GMx4lqNa] 입니다.

인증코드를 세션에 저장 후 이 세션은 저장시간 180초로 지정

세션을 사용한 이메일 인증

Ajax로 새로고침 없이 확인

이메일

ekdhs2939@naver.com 인증하기

확인 남은시간: 157초

```
function mailAuthCheck(){
    var authNumber = $('input[name="authNumber"]').val();
    $.ajax({
        type: "post",
        url: "<%=request.getContextPath()%>/member/mailAuthCheck.do",
        data: {"authNumber" : authNumber},
        dataType: "json",
        success: function(data){
            if(data.txt=='pass'){
                $('#sessionTimer').css("display","none");
                $('#passCheckOK').css("display","inline-block");
                $('#passCheckOK').html("인증완료");
                $('input[name="authNumber"]').attr('readonly',true);
            }
        },
        error: function(){
            alert("실패");
        }
    });
}
```

```
@ResponseBody
@RequestMapping(value="/mailAuthCheck.do")
public JSONObject mailAuthCheck(@RequestParam("authNumber") String authNumber, HttpSession session) {
    JSONObject js = new JSONObject();
    String savedPass = (String) session.getAttribute("authenCode");
    String txt = "";
    if(authNumber.equals(savedPass)) {
        txt = "pass";
        session.setAttribute("mailPass", "pass");
    }else {
        txt = "인증실패";
    }

    js.put("txt", txt); -----> 세션 확인 후 결과값을 전송
    return js;
}
```

전송된 값이 세션에 저장된 코드와 일치하면 'pass' 메시지 세션에 저장

이메일

ekdhs2939@naver.com 인증하기

LML0VEU3 확인 인증완료

Pass 메시지가 전송되면 인증완료 메시지를 띄우고 인증번호는 수정불가 상태로 변경

```
@RequestMapping(value = "/memberJoinAction.do")
public String memberJoinAction(@Valid MemberVo mv, HttpSession session, Model model, BindingResult bindingResult)
    if (bindingResult.hasErrors()) {

        return "member/memberJoin"; // 오류 페이지로 리다이렉트 또는 프워드
    }
    String authSession = (String) session.getAttribute("mailPass");

    if(authSession!=null && authSession.equals("pass")) {
        String memberPwd2 = bCryptPasswordEncoder.encode(mv.getMemberPwd());
        mv.setMemberPwd(memberPwd2);
        int value = ms.memberInsert(mv);
        return "redirect:/"; // 프워드방식이 아닌 sendRedirect 방식
    }else {
        model.addAttribute("error", "메일인증이 처리되지 않았습니다.");
        return "member/memberJoin";
    }
}
```

Controller에서 메일인증 2차확인

네이버 블로그 스크래핑

블로그리뷰

Eat&Play&Love

전주 객사 퀘리단길 피자 원팀 솔직·전라감영가라지
작성되었습니다.] #전주술집#전주웨리단길술집#전주객사술집#전주디트로이드피자#전주피자맛집#전주핫플#전주분위기좋은술집#전주기념일#전

먹어보고 사용하고 체험하는 코코★

전주 전라감영 애니메이션에서본 프랑스 디자트가 여기에...
진짜 다른날보다도 스트레스 해소도 되면서→ 즐거운 시간보내기에 딱 좋음 ㅋㅋㅋ 물론 같이 가주는 사랑스러운 친구 더본이지만용→ 'ㄷ'★ ►

할게 많아도 일단은 놀아야지

전라도 전주 여행 전라감영 선화당 선정비 행랑 가마구경
전라도 전주 여행 전라감영 선화당 선정비 가마 행랑 구경 Editor ⓤ 호리호릿 조선시대에서 전주시청 전라도청처럼 관공서를 담당하고 있는 전주

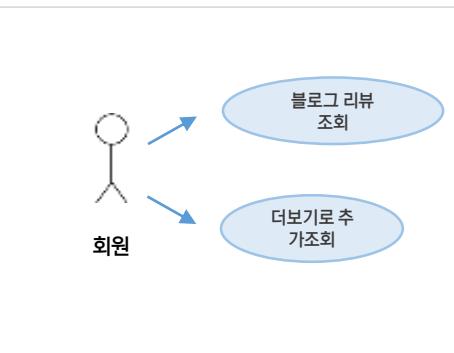
#팅이를그리다

당일치기 전주 갈만한곳 전라감영 전복 여행지 추천
당일치기 전주 갈만한곳 전라감영 전복 여행지 추천 글, 사진, 영상: 킹이[다녀온 날: 2023.10.13] 전주 갈만한곳 하면 어디가 생각나시

릿짱과 맛있는 이야기를 공유해요

전주 한옥마을 술집 - 전라감영가라지 3세 피자와 와인의 조합 #2
무료 #전주한옥마을술집 #전라감영가라지 #전주전라감영가라지 #전주술집 #전주피자 #전주피자맛집 #전주한옥마을피자 #전주와인 #전주술집맛집 #전주

+더보기



- 해당 컨텐츠와 관련된 키워드로 검색된 블로그 목록을 가져온다.
- 처음 5개 출력 후 더보기버튼을 이용해서 5개씩 목록이 추가되도록 한다.

DEVELOPMENT

주요기능
Main Skill

네이버 블로그 스크래핑

더보기 기능을 위해서 Ajax 활용

```
var start = 4; // 시작 인덱스
var batchSize = 5; // 한 번에 뽑아줄 항목의 개수
function blogReviewLead() {
    var subject = "전주 ${cv.contentsSubject}";
    $.ajax({
        type: "post",
        url: "${pageContext.request.contextPath}/review/blogReview.do",
        dataType: "json",
        data: {
            "subject": subject
        },
        cache: false,
        success: function (data) {
            displayBlogItems(data);
        },
        error: function () {
            alert("통신 오류 실패");
        }
    });
}
```

```
function displayBlogItems(data) {
    var txt = "";
    var end = Math.min(start + batchSize, data.url.length);

    for (var i = start; i < end; i++) {
        txt += "<a href='" + data.url[i] + "'>" +
            "<div class='blogCon'>" +
            "<div class='blogName'>" + data.blogname[i] + "</div>" +
            "<div class='blogSubject'>" + data.title[i] + "</div>" +
            "<div class='blogContents'>" + data.contents[i] + "</div>" +
            "</div>" +
            "</a>";
    }

    $(".blogList").append(txt);
    start = end;

    // 더 블라우저 데이터가 있는 경우에만 "더 보기" 버튼을 표시
    if (start < data.url.length) {
        $("#loadMoreBtn").show();
    } else {
        $("#loadMoreBtn").hide();
    }
}
```

검색할 키워드

```
<!-- https://mvnrepository.com/artifact/org.jsoup/jsoup -->
<dependency>
    <groupId>org.jsoup</groupId>
    <artifactId>jsoup</artifactId>
    <version>1.15.3</version>
</dependency>
```

Jsoup 라이브러리 추가

```
@RequestMapping(value="blogReview.dp")
public JSONObject blogReview(@RequestParam("subject") String subject) throws Exception {
    JSONObject js = new JSONObject();

    ArrayList<String> al1 = new ArrayList<>();
    ArrayList<String> al2 = new ArrayList<>();
    ArrayList<String> al3 = new ArrayList<>();
    ArrayList<String> al4 = new ArrayList<>();

    String address = "https://search.naver.com/search.naver?query=" + subject + "&nso=&where=blog&sm=tab_opt";
    Document rawData = Jsoup.connect(address).get();

    Elements blogOption = rawData.select("li.bx");

    String realURL = "";
    String realTitle = "";
    String blogName = "";
    String realContents = "";
    for(Element option : blogOption) {
        //System.out.println(option);
        //블로그 링크
        realURL = option.select(".title_area a").attr("href");
        //게시글 제목
        Elements titleDiv = option.select(".title_area");
        realTitle = titleDiv.select("a").text();
        //게시글 내용
        Elements contentsDiv = option.select(".dsc_area");
        realContents = contentsDiv.select("a").text();
        if(realContents.length()>=30) realContents = realContents.substring(0, 80);
        //블로그 이름
        Elements blognameDiv = option.select(".user_info");
        blogName = blognameDiv.select("a").text();

        al1.add(realURL);
        al2.add(realTitle);
        al3.add(blogName);
        al4.add(realContents);
    }

    js.put("url", al1);
    js.put("title", al2);
    js.put("blogname", al3);
    js.put("contents", al4);
}
```

검색 후 페이지에서 가져올 요소 선택

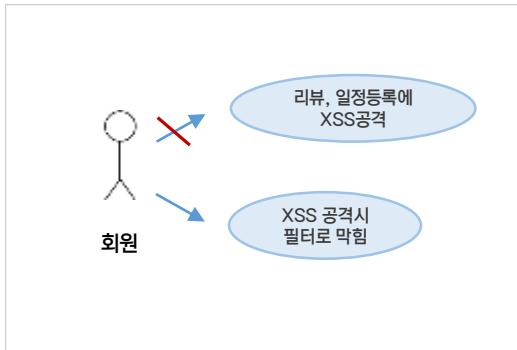
필요한 요소들을 JSON형식으로 전달

XSS 필터

리뷰장소	덕진공원
평가	★★★★★
리뷰내용	<script>alert('바보')</script>



여행후기		
광고맨		
2024-01-12 13:54:18	alert('바보')	
★ 3		



- XSS공격시도가 있을 것을 방지해 필터를 설정 한다.
- script 등 위험한 태그들을 공백으로 치환하도록 한다.

XSS 필터

```

public class CrossScriptingFilter implements Filter{
    public FilterConfig filterConfig;

    public void init(FilterConfig filterConfig) throws ServletException {
        this.filterConfig = filterConfig;
    }

    public void destroy() {
        this.filterConfig = null;
    }

    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
        throws IOException, ServletException {
        chain.doFilter(new RequestWrapper((HttpServletRequest) request), response);
    }
}

```

크로스 사이트 스크립트를 처리하기 위해
스프링의 FilterChain 사용

doFilter 메소드를 활용하여 실행될 클래스
RequestWrapper()

```

public class RequestWrapper extends HttpServletRequestWrapper {
    public RequestWrapper(HttpServletRequest servletRequest) {
        super(servletRequest);
    }

    public String[] getParameterValues(String parameter) {
        String[] values = super.getParameterValues(parameter);
        if (values==null) {
            return null;
        }
        int count = values.length;
        String[] encodedValues = new String[count];
        for (int i = 0; i < count; i++) {
            encodedValues[i] = cleanXSS(values[i]);
        }
        return encodedValues;
    }
}

```

```

public String getParameter(String parameter) {
    String value = super.getParameter(parameter);
    if (value == null) {
        return null;
    }
    return cleanXSS(value);
}

public String getHeader(String name) {
    String value = super.getHeader(name);
    if (value == null)
        return null;
    return cleanXSS(value);
}

```

입력되는 parameter, header들이
cleanXSS 메서드를 거치도록 함

XSS 필터

RequestWrapper.java

```

private String cleanXSS(String value) {
    StringBuffer sb = null;
    String[] checkStr_arr = {
        "<script>","</script>",
        "<javascript>","</javascript>",
        "<iframe>","</iframe>",
        "<vbscript>","</vbscript>",
        "<object>","</object>",
        "<img>","</img>",
        "<marquee>","</marquee>",
        "onerror","onclick"," onload","onmouseover","onstart"
    };
    for(String checkStr : checkStr_arr) {
        while(value.indexOf(checkStr)!=-1) {
            value=value.replaceAll(checkStr, "");
        }
        while(value.toLowerCase().indexOf(checkStr)!=-1) {
            sb=new StringBuffer(value);
            sb.replace(value.toLowerCase().indexOf(checkStr), value.toLowerCase().indexOf(checkStr)+checkStr.length(), "");
            value=sb.toString();
        }
    }
    value = value.replaceAll("eval\\\\((.*))","");
    value = value.replaceAll("[\\\\\"\\\\'\\\\'][\\\\s]*javascript:(.*)[\\\\\"\\\\']","");
    return value;
}

```

기본적인 script, javascript 태그를 포함한 문제가 될 태그들을 전부 치환하도록 함

web.xml

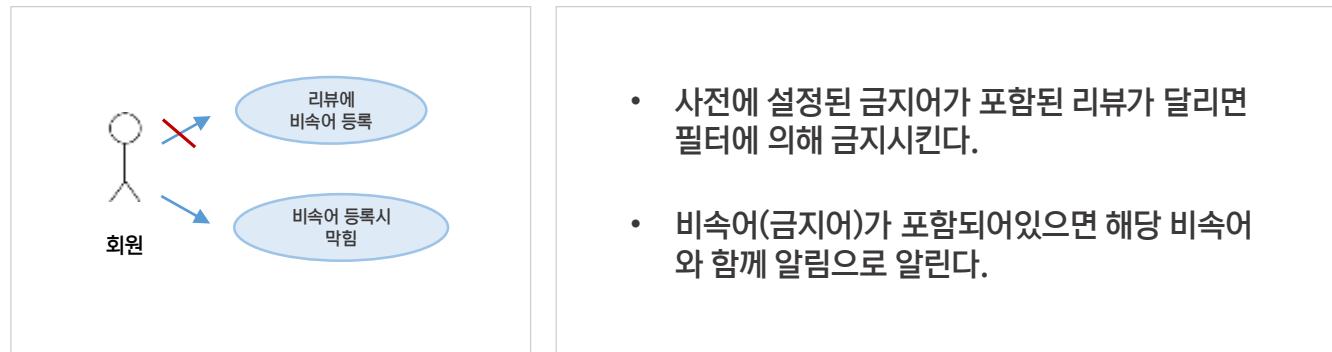
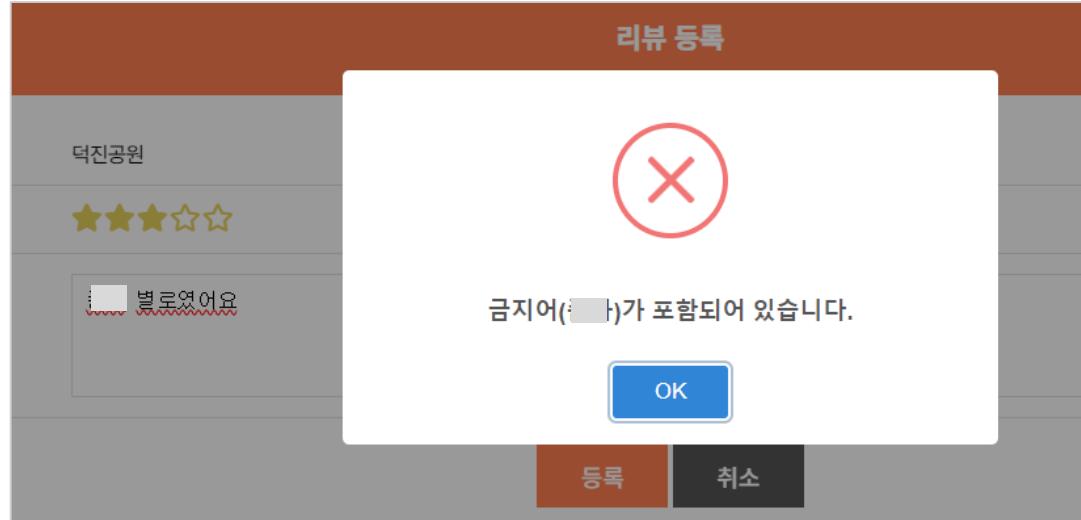
```

<filter>
    <filter-name>XSS</filter-name>
    <filter-class>com.ezen_jeonju.myapp.filter.CrossScriptingFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>XSS</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

```

filter가 모든 페이지에 적용되도록 /*로 기재

비속어 필터



비속어 필터

리뷰 등록시 실행되는 script

```

$.ajax({
    type : "post",
    url : "${pageContext.request.contextPath}/review/reviewWrite.do",
    dataType : "json",
    data : {
        "cidx" : cidx,
        "reviewScore" : score,
        "reviewArticle" : article
    },
    cache : false,
    success : function(data){
        if(data.bad != null){
            swal(
                '',
                '금지어(' + data.bad + ')가 포함되어 있습니다.',
                'error'
            )
            return;
        }
        if(data.txt === "pass"){
            swal(
                '',
                '리뷰가 등록되었습니다.',
                'success'
            )
            $('#writeReview').css('display','none');
            reviewList();
        }else{
            alert(data.txt);
        }
    },
    error : function(){
        swal(
            '',
            '등신오류 실패',
            'error'
        );
    }
});

```

```

@RequestMapping(value="/reviewWrite.do")
public JSONObject reviewWrite(ReviewWo rv, HttpSession session) {
    JSONObject jo = new JSONObject();
    String txt = "";
    String badWord = "";
    if(session.getAttribute("midx") == null) {
        txt = "로그인 이후 이용바랍니다.";
        jo.put("txt", txt);
        return jo;
    }else {
        int midx = Integer.parseInt(session.getAttribute("midx").toString());
        rv.setMidx(midx);
        String article = rv.getReviewArticle();
        if(bwf.checkBadWord(article)) {
            badWord = bwf.FindBadWord(article);
            System.out.println(badWord);
            jo.put("bad", badWord);
            return jo;
        }
        int value = rs.reviewWrite(rv);
        if(value != 0) {
            txt = "pass";
        }else {
            txt = "작성오류";
        }
        jo.put("txt", txt);
        return jo;
    }
}

```

“bad”키가 존재하면 해당 금지어와 함께
alert으로 알림

등록 시 비속어가 포함되면 “bad”라는 key
에 비속어를 넣어 JSON으로 넘김

비속어가 포함되어있지 않을 때만
reviewWrite 실행하여 “pass” 전달

비속어 필터

BadWordFiltering.java

```
public class BadWordFiltering implements BadWords{
    private final Set<String> badWordsSet = new HashSet<>(List.of(badWords));
    private final Map<String, Pattern> badWordPatterns = new HashMap<>();

    @PostConstruct
    public void compileBadWordPatterns() {
        for (String word : badWordsSet) {
            String wordPattern = buildWordPattern(word);
            badWordPatterns.put(word, Pattern.compile(wordPattern));
        }
    }

    private String buildWordPattern(String word) {
        String[] chars = word.split("");
        StringBuilder wordPatternBuilder = new StringBuilder();

        for (int i = 0; i < chars.length; i++) {
            wordPatternBuilder.append(Pattern.quote(chars[i]));

            if (i < chars.length - 1) {
                wordPatternBuilder.append("(");
                for (String delimiter : delimiters) {
                    wordPatternBuilder.append(Pattern.quote(delimiter)).append("|");
                }
                wordPatternBuilder.deleteCharAt(wordPatternBuilder.length() - 1);
                wordPatternBuilder.append(")");
            }
        }

        return wordPatternBuilder.toString();
    }
}
```

```
public interface BadWords {
    String[] badWords = {"나", "나1", "나2", "나3", "나4", "나5", "나6", "나7", "나8", "나9"}; //비속어
    String[] delimiters = {" ", ",", ";", ".", "!", "?", "@", "1", "2"}; //비속어 변형 방지
```

‘나1쁜말’ 같은 조합 방지를 위해 badWords에 저장된 비속어 배열들을 delimiters 배열과의 조합되는 모든 경우의 수를 만들어냄.

```
public boolean checkBadWord(String input) {
    for (Pattern pattern : badWordPatterns.values()) {
        Matcher matcher = pattern.matcher(input);
        if (matcher.find() && matcher.start() == 0) {
            return true;
        }
    }
    return false;
}
```

리뷰 등록시 모든 패턴과 매칭하여 비속어가 없는지 확인

```
public String FindBadWord(String input) {
    for (Map.Entry<String, Pattern> entry : badWordPatterns.entrySet()) {
        String badWord = entry.getKey();
        Pattern pattern = entry.getValue();

        Matcher matcher = pattern.matcher(input);
        if (matcher.find() && matcher.start() == 0) {
            return badWord;
        }
    }
    return null;
}
```

비속어가 존재하면 해당 비속어를 찾아내서 출력

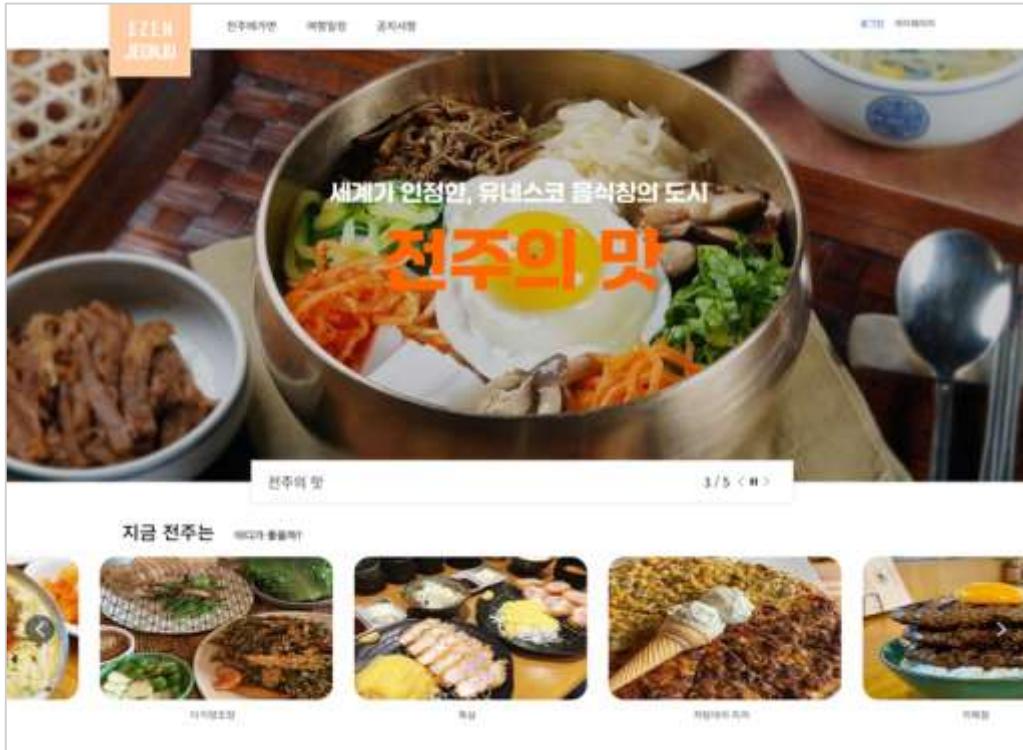
금지어(나)가 포함되어 있습니다.

OK

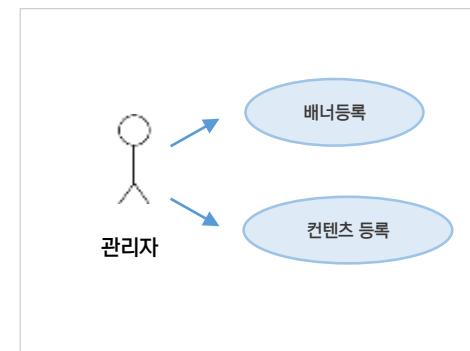
DEVELOPMENT

주요기능
Main Skill

메인페이지 슬라이드



- 등록된 배너가 메인 슬라이드 화면에 나온다.
- 컨텐츠에 등록된 글의 썸네일이 세컨트 슬라이드 화면에 나온다.
- 최근 날짜 7개의 게시글을 가져온다.



DEVELOPMENT

주요기능
Main Skill

메인페이지 슬라이드

Jquery 슬라이더 라이브러리 slick-slider

```
// 첫번째 슬라이더 플레이, 중지 버튼
$('#btn-play').hide();

$('#btn-play').click(function(){
    $('.first-text-list').slick('slickPlay');
    $(this).hide();
    $('#btn-pause').show();
});

$('#btn-pause').click(function(){
    $('.first-text-list').slick('slickPause');
    $(this).hide();
    $('#btn-play').show();
});

// 메인배너 리스트 슬라이드
$('.first-visual-list').slick({
    infinite: true,
    fade: true,
    arrows: false,
    asNavFor: '.first-text-list'
});

// 메인배너 텍스트 슬라이드
$('.first-text-list').slick({
    autoplay: true,
    autoplaySpeed: 3000,
    slidesToShow: 1,
    slidesToScroll: 1,
    vertical : true,
    verticalSwiping : true,
    asNavFor: '.first-visual-list',
    prevArrow : $('.btn-prev'),
    nextArrow : $('.btn-next'),
});
```

버튼과 재생, 중지 버튼을 생성 및 asNavFor를 이용 이미지 슬라이더와 텍스트 슬라이드 동기화

```
// 초기에 페이지 번호 및 트랄 넘버 설정
var initialCurrentSlide = $('.first-text-list').slick('slickCurrentSlide') + 1;
var totalSlides = $('.first-text-list').slick('getSlick').slideCount;

// 초기 페이지 번호 설정
$('.first-current-num').text(initialCurrentSlide);
// 초기 트랄 넘버 설정
$('.first-total-num').text(totalSlides);

// 슬라이더의 afterChange 이벤트 핸들러
$('.first-text-list').on('afterChange', function (event, slick, currentSlide) {
    // 현재 슬라이드 번호를 가져옵니다.
    var currentSlideNumber = currentSlide + 1;

    // 페이지 번호 업데이트
    $('.first-current-num').text(currentSlideNumber);
});
```

페이지와 총 페이지 수를 설정하여 해당 div에 값 등록

```
function calculateSlidesToShow() {
    var viewportWidth = $(window).width();
    //console.log("뷰포트" + viewportWidth);

    if(viewportWidth >= 1500){
        var imageWidth = 500;
    }else{
        var imageWidth = 330;
    }
    //console.log("이미지넓이" + imageWidth);

    var totalwidth = imageWidth;
    //console.log("토탈넓이" + totalwidth);

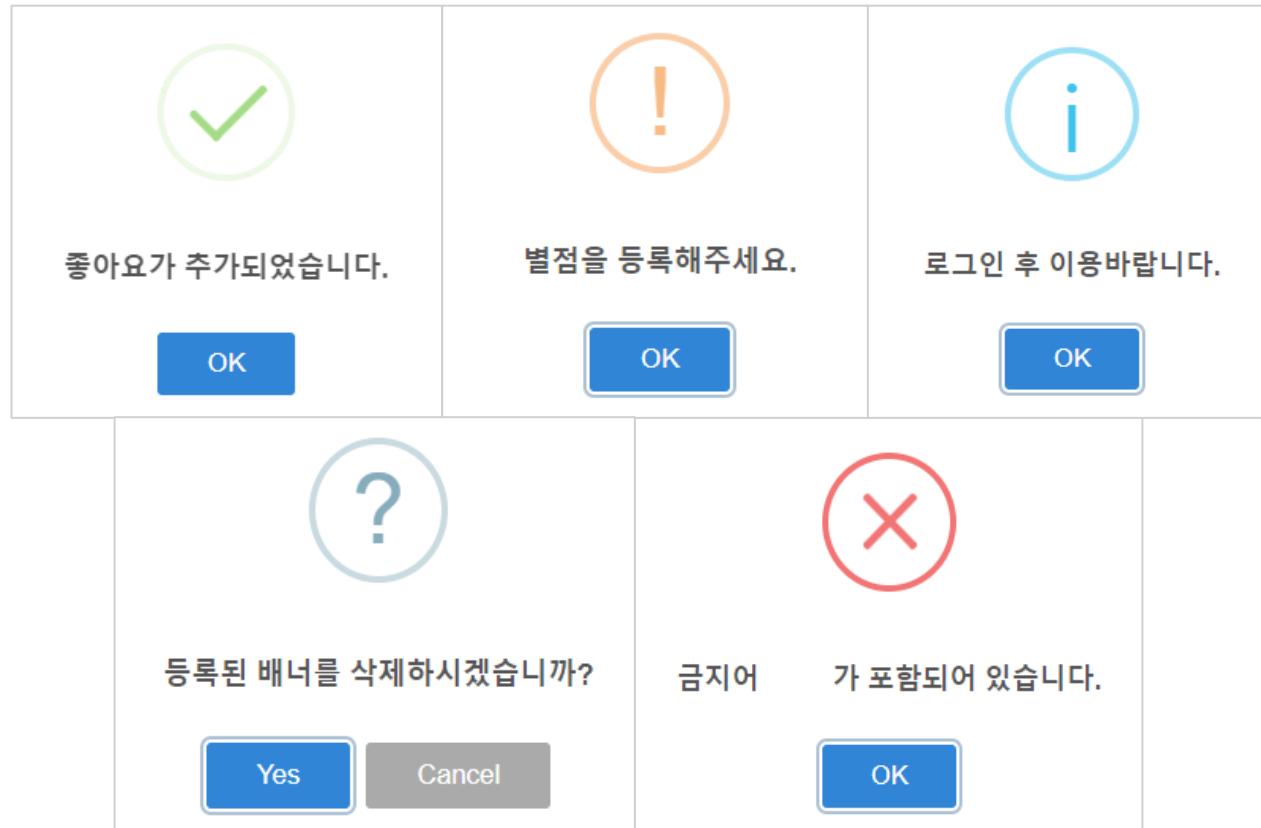
    // 반올림하여 소수점 한자리까지 표시
    var slidesToShow = Math.round(viewportWidth / totalwidth * 10) / 10;
    //console.log("개수" + slidesToShow);

    return slidesToShow;
}
```

```
// 두번째 슬라이드
$('.second-visual-list').slick({
    pauseOnHover: true,
    autoplay: true,
    autoplaySpeed: 3000,
    centerMode: true,
    centerPadding: '60px',
    slidesToShow: calculateSlidesToShow()
});
```

뷰포트를 사용하여 실시간으로 슬라이더 수 개수 조절.

SweetAlert



- 기존의 alert 대신 sweetAlert 을 사용하여 UI와 기능적인 요소를 향상시킨다.
- 회원에게 출력되는 모든 alert 을 sweetAlert으로 대체한다.
 - 좋아요 추가, 취소하기
 - 리뷰 작성, 삭제, 신고하기
 - 로그인 후 이용 안내

SweetAlert

sweetAlert을 사용하기 위해 CDN 추가

```
<!-- SweetAlert2 -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/7.2.0/sweetalert2.min.css">
<script src="https://cdnjs.cloudflare.com/ajax/libs/limonte-sweetalert2/7.2.0/sweetalert2.all.min.js"></script>
```

```
function likeThis(event){}
  $.ajax({
    type : "post",
    url : "${pageContext.request.contextPath}/contentsLike/contentsLike.do",
    dataType : "json",
    data : {
      "cidx" : cidx
    },
    cache : false,
    success : function(data){
      if(data.value === "좋아요"){
        swal(
          '',
          '',
          '<b style="font-weight:bold;">좋아요가 추가되었습니다.</b>',
          'success'
        )
      }else if(data.value === "좋아요 취소"){
        swal(
          '',
          '',
          '<b style="font-weight:bold;">좋아요가 취소되었습니다.</b>',
          'success'
        )
      }else if(data.value === "로그인 후 이용바랍니다."){
        swal(
          '',
          '',
          '<b style="font-weight:bold;">로그인 후 이용바랍니다.</b>',
          'info'
        )
      }
      likeCheck();
    },
    error : function(){
      swal(
        '',
        '',
        '<b style="font-weight:bold;">통신오류 실패</b>',
        'error'
      );
    }
  });
}
```

기존에 호출했던 alert 대신 swal로 변경하여 코드 작성

Icon type



title

좋아요가 추가되었습니다.

OK

SweetAlert

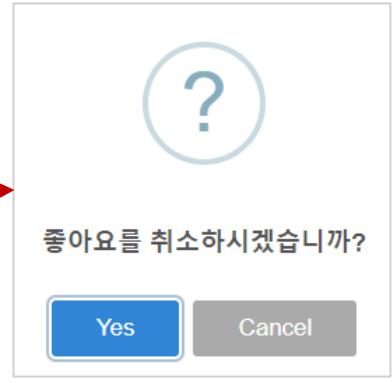
```

function likeDel(clidx){
    swal({
        title: '',
        text: "좋아요를 취소하시겠습니까?",
        type: "question",
        showCancelButton: true,
        confirmButtonText: "Yes",
        cancelButtonText: "Cancel"
    }).then((result) => {
        if(result.value){
            $.ajax({
                type : "post",
                url: "${pageContext.request.contextPath}/mypage/likeDelete.do",
                dataType : "json",
                data : {
                    "clidx": clidx
                },
                cache : false,
                success : function(data){
                    if(data.txt === "좋아요를 취소하였습니다."){
                        swal(
                            '',
                            '좋아요가 취소되었습니다.',
                            'success'
                        );
                        likelist();
                    }else{
                        alert(data.txt);
                    }
                },
                error : function(){
                    swal(
                        '',
                        '등신오류 실패',
                        'error'
                    );
                }
            });
        }
    });
}

```

Sweet Confirm으로 사용하여 confirm / cancel 버튼의 값에 따라 ajax request 실행

- Yes 버튼 ⇒ {"value":true}
- Cancel 버튼 ⇒ {"dismiss":"cancel"}



- Yes 버튼을 클릭하여 result로 value를 받으면 ajax가 실행됨
- Cancel 버튼을 클릭하면 sweetAlert 창이 사라짐

일정표 등록

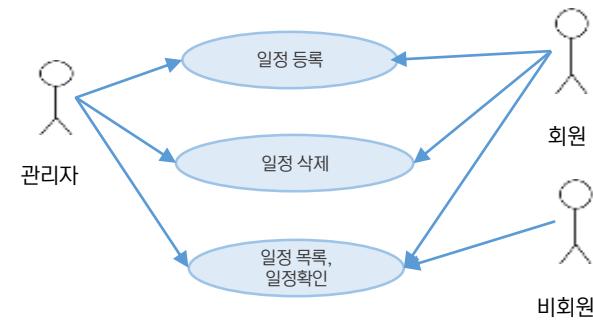
일정 만들기

제목 공개여부

기간 2024-01-08 ~ 2024-01-10

저장

1/13



- 일정을 등록할 때 공개여부를 선택할 수 있다.
- 기간등록을 통해 기간 등록 후 각 날짜에 해당하는 일정을 등록할 수 있다.
- 왼쪽목록의 일정 이름을 선택해 일정을 드래그로 자유롭게 설정할 수 있다.
- 테이블 셀을 누르거나 드래그시 해당 열이 강조되면서 해당하는 마커와 경로가 출력된다.
- 목록에 원하는 일정이 없을 때 지도로 직접 찾을 수 있다.
- 등록한 여행 일정을 다운로드 받을 수 있다.

DEVELOPMENT

주요기능
Main Skill

일정표 등록

```
// 테이블 요소 생성
var table = document.createElement('table');
table.id = 'dragDropTable';

var thead = document.createElement('thead');
var trHeader = document.createElement('tr');
for (var i = 1; i <= columnCount; i++) {
    var th = document.createElement('th');
    th.textContent = 'Day ' + i + ' : ' +(startDate.getMonth()+1) +'월' + startDate.getDate()+'일';

    // 헤더 셀에 스타일 추가
    th.style.height = '57px';
    th.style.width = '600px';
    th.style.maxHeight = '57px';
    th.style.minWidth = '200px';
    th.style.whiteSpace = 'nowrap';
    th.classList.add("fixed");

    startDate.setDate(startDate.getDate()+1);
    trHeader.appendChild(th);
}

thead.appendChild(trHeader);
table.appendChild(thead);

// 테이블 바디 생성 (임시 데이터로 5개의 행 추가)
var tbody = document.createElement('tbody');
for (var j = 1; j <= 16; j++) {
    var trBody = document.createElement('tr');
    for (var k = 1; k <= columnCount; k++) {
        if((j==1) && (k>1)){
            continue;
        }

        var td = document.createElement('td');
        td.textContent = "";

        // 바디 셀에 스타일 추가
        td.style.height = '57px';
        td.style.width = '300px';
        td.style.maxHeight = '57px';
        td.style.whiteSpace = 'nowrap';

        //행 열 아이디 부여
        td.id = j+'_'+k;

        //날짜 시간 장소
        var hour = j+6;

        $(td).attr("name", addDays($('#startDate').val(), k-1) + "_" + hour + "_" + k);
        trBody.appendChild(td);
    }
}

tbody.appendChild(trBody);
```

일정기간 등록에 따른 유동적 테이블 생성

일정 시작날짜와 마지막날짜를
입력하면 일정에 맞는
table th생성

시간	Day 1: 1월17일	Day 2: 1월18일
08 ~ 22:00	장소를 누르시고 원하는 시간대에 드래그하세요	
08:00		
09:00		
10:00		
11:00		
12:00		
13:00		
14:00		

// 첫 번째 행의 colspan 설정
var firstRow = tbody.querySelector('tr');
var firstCell = firstRow.querySelector('td');
firstCell.colSpan = columnCount;
firstCell.textContent = "장소를 누르시고 원하는 시간대에 드래그하세요";
firstCell.id = 'addSchedule';
firstRow.classList.add("fixed_1");
table.appendChild(tbody);

DEVELOPMENT

주요기능
Main Skill

일정표 등록

```
// 음식점 이름을 클릭하면 테이블 셀에 정보를 추가
function addToTable(placeName, placeLatitude, placeLongitude) {
    if(placeLatitude){
        placeName = placeName+","+placeLatitude+","+placeLongitude;
    }
    $("#scheduletbl").scrollTop(0);
    $("#scheduletbl").scrollLeft(0);
    let tableCell = document.getElementById("addSchedule");
    let placeArray = placeName.split(',');
    // 이름 추가
    if(tableCell == null){
        alert('기간을 먼저 등록해주세요');
        return
    }else{
        if (childWindow && !childWindow.closed) {
            childWindow.close();
        }
    }
    tableCell.innerHTML = placeArray[0];

    // X 버튼 및 추가 정보를 담은 input 태그 추가
    tableCell.innerHTML += "<input type='hidden' value='"+ placeArray[0] + "'>";
    tableCell.innerHTML += "<input type='hidden' value='"+ placeArray[1] + "'>";
    tableCell.innerHTML += "<input type='hidden' value='"+ placeArray[2] + "'>";
    tableCell.innerHTML += "&nbsp;&nbsp;&nbsp;&nbsp;<a href='#' style='text-decoration:none'; onclick='Xclose(this, \"' + placeArray[0] + "\")';>X</a>";
}
```

The screenshot shows a user interface for travel planning. On the left, there's a sidebar with tabs for '음식점' (Food), '관광지' (Attraction), '맛집 추천' (Recommended Restaurants), and '명소 추천' (Recommended Landmarks). Below these are dropdown menus for '음식점' and '관광지', and a search button labeled '검색'. A thumbnail image of a meal is shown with the text '가운데집' and '지도' (map). On the right, there's a large schedule table titled '시간' (Time) with two rows for 'Day 1: 1월17일' (January 17th) and 'Day 2 : 1월18일' (January 18th). The table has columns for time intervals like '08 ~ 22:00' and specific times like '08:00' and '09:00'. A red box highlights the text '가족회관' (Family Hall) in one of the table cells. A dashed red arrow points from this cell to a callout box at the bottom right. The callout box contains the text: '장소를 클릭하면 일정테이블로 이동, 맵에 마커표시 inputType=hidden으로 해당 정보 저장' (Clicking the location moves to the schedule table, marks the map with a marker, saves the information using inputType=hidden).

DEVELOPMENT

주요기능
Main Skill

일정표 등록

일정을 원하는 날짜/시간에 드래그하기 위한 함수

```
// 드래그 시작 이벤트 핸들러
function handleDragStart(e) {
    if (this.innerText === "장소를 누르시고 원하는 시간대에 드래그하세요" || this.innerText === "") {
        e.preventDefault();
        return;
    }

    // 드래그 시작한 요소의 참조 저장
    dragSrcElement = this;
    // 드래그 효과 설정
    e.dataTransfer.effectAllowed = 'move';
    // 드래그 데이터 설정 (텍스트/HTML 형식)
    e.dataTransfer.setData('text/html', this.innerHTML);
    // 드래그 중인 행에 클래스 추가하여 투명도 적용
    this.classList.add('dragged');
}

// 드래그 중인 상태에서 다른 요소 위로 이동할 때 발생하는 이벤트 핸들러
function handleDragOver(e) {
    // 기본 동작 방지
    if (e.preventDefault) {
        e.preventDefault();
    }
    // 드래그 가능한 효과 설정
    e.dataTransfer.dropEffect = 'move';
    return false;
}

// 드래그 중인 상태에서 요소에 진입했을 때 발생하는 이벤트 핸들러
function handleDragEnter() {
    // 드래그 중인 행에 클래스 추가하여 시작적 효과 적용
    this.classList.add('over');
}

// 드래그 중인 상태에서 요소에서 빠져나갈 때 발생하는 이벤트 핸들러
function handleDragLeave() {
    // 드래그 중인 행에서 클래스 제거하여 시작적 효과 제거
    this.classList.remove('over');
}
```

시간	Day 1 : 1월17일
08 ~ 22:00	이곳에 생성된 일정을 드래그로 이동
08:00	
09:00	
10:00	1: 가족회관 X
11:00	

```
// 테이블의 모든 행을 선택하여 드래그 이벤트 리스너 등록
const items = document.querySelectorAll('#dragDropTable tbody td');
items.forEach(function(item) {
    item.draggable = true;
    item.addEventListener('dragstart', handleDragStart, false);
    item.addEventListener('dragover', handleDragOver, false);
    item.addEventListener('dragenter', handleDragEnter, false);
    item.addEventListener('dragleave', handleDragLeave, false);
    item.addEventListener('drop', handleDrop, false);
    item.addEventListener('dragend', handleDragEnd, false);
    item.addEventListener('click', handleClick, false);
});
```

```
body{
    -webkit-touch-callout: none; /* iOS Safari */
    -webkit-user-select: none; /* Safari */
    -ms-user-select: none; /* 인터넷익스플로러 */
    user-select: none;
}
```

일정 이외의 요소들의 드래그 방지

DEVELOPMENT

주요기능
Main Skill

일정표 등록

드래그 후 드롭이 일어났을 때 실행되는 함수

```
function handleDrop(e) {
    // 이벤트 전파 방지
    if (e.stopPropagation) {
        e.stopPropagation();
    }

    // 드래그 소스와 드롭 대상이 다른 경우만 처리
    if (dragSrcElement !== this) {
        // 드래그 소스의 내용을 드롭 대상으로 이동
        dragSrcElement.innerHTML = this.innerHTML;
        // 드롭 대상의 내용을 드래그 소스로 이동
        this.innerHTML = e.dataTransfer.getData('text/html');

        // 기존 맵에 있는 marker 모두 지우기
        for (var i = 0; i < markers.length; i++) {
            markers[i].setMap(null);
        }
        // markers 초기화
        markers = [];
        positions = [];
        $(".highlight").removeClass("highlight");
        $(".highlight2").removeClass("highlight2");
        var columnIndex = $(this).index();
    }
}
```

```
function addRankToTable(tableId) {
    var table = document.getElementById(tableId);
    var rows = table.getElementsByTagName('tr');
```

```
for (var j = 0; j < rows[0].cells.length; j++) {
    var rank = 1;

    for (var i = 2; i < rows.length; i++) {
        var cell = rows[i].cells[j];

        // td에 값이 있을 경우에만 랭크 추가
        if (cell.innerHTML.trim() !== "") {
            var existingRank = cell.querySelector('#sequence');
            if (existingRank) {
                existingRank.parentNode.removeChild(existingRank);
            }

            cell.innerHTML = "<Strong id='sequence'>" + "<span id='rank'>" +
                rank + "</span>" + " : </Strong>" + cell.innerHTML;
            rank++;
        }
    }
}
```

```
// 모든 행에 대해 현재 열에 해당하는 td에 highlight 클래스 추가
$('#dragDropTable tr:not(:eq(1))').find('td: eq(' + columnIndex + ')').addClass('highlight');
$('#dragDropTable th: eq(' + columnIndex + ')').addClass('highlight2');
$('#nDate').val((columnIndex+1)+" 일자");
// 새로운 marker 생성
$("tr").each(function (index) {
    var highlightedCell = $(this).find('.highlight');

    var cellText = highlightedCell.text();
    var inputs = highlightedCell.find('input');
    var placeName = inputs.eq(0).val();
    var tourCourseLatitude = inputs.eq(1).val();
    var tourCourseLongitude = inputs.eq(2).val();

    if (cellText !== "") {
        // 마커 이미지 생성
        var markerImage = new kakao.maps.MarkerImage(
            'https://placehold.jp/23/3d4070/fffff/25x25.png?text=' + (markers.length + 1),
            new kakao.maps.Size(25, 25),
            { offset: new kakao.maps.Point(13, 25) }
        );

        var markerPosition = new kakao.maps.LatLng(tourCourseLatitude, tourCourseLongitude);
        var marker = new kakao.maps.Marker({
            position: markerPosition,
            title: placeName,
            image: markerImage // 마커에 이미지 설정
        });

        marker.setMap(map);
        positions.push(markerPosition);
        panTo(tourCourseLatitude, tourCourseLongitude);
        markers.push(marker);
        drawLine(positions);
    }
});
```

각 날짜별 일정 순서 생성



DEVELOPMENT

주요기능
Main Skill

일정표 등록

일정 테이블 클릭시 해당 날짜에 포커스(highlight)
되고 지도는 해당 날짜에 해당하는 루트로 출력

```

function handleClick() {
    for (var i = 0; i < markers.length; i++) {
        markers[i].setMap(null);
    }
    markers = [];
    positions = [];
    clearBounds(); // Clear bounds before adding new markers

    $(".highlight").removeClass("highlight");
    $(".highlight2").removeClass("highlight2");
    var columnIndex = $(this).index();
    $("#nDate").val((columnIndex + 1) + " 일자");

    $('#dragDropTable tr:not(:eq(1))').find('td:eq(' + columnIndex + ')').addClass('highlight');
    $('#dragDropTable th:eq(' + columnIndex + ')').addClass('highlight2');

    $("tr").each(function (index) {
        var $highlightedCell = $(this).find('.highlight');
        var cellText = $highlightedCell.text();

        var inputs = $highlightedCell.find('input');
        var placeName = inputs.eq(0).val();
        var tourCourseLatitude = inputs.eq(1).val();
        var tourCourseLongitude = inputs.eq(2).val();

        if (cellText !== "") {
            var markerImage = new kakao.maps.MarkerImage(
                'https://placehold.jp/23/3d4070/ffffff/25x25.png?text=' + (markers.length + 1),
                new kakao.maps.Size(25, 25),
                { offset: new kakao.maps.Point(13, 25) }
            );

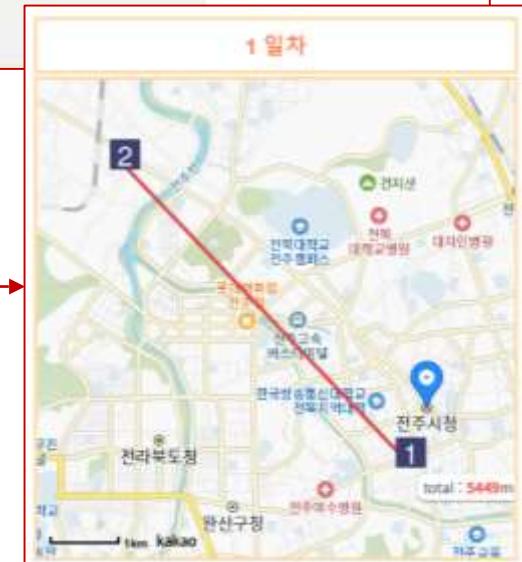
            var markerPosition = new kakao.maps.LatLng(tourCourseLatitude, tourCourseLongitude);
            var marker = new kakao.maps.Marker({
                position: markerPosition,
                title: placeName,
                image: markerImage
            });

            marker.setMap(map);
            markers.push(marker);
            positions.push(markerPosition);
            bounds.extend(markerPosition);
        } else {
            content = '<div class="dotOverlay distanceInfo">total : <span class="number">0</span>m</div>';
            positions.push(new kakao.maps.LatLng(35.8240808, 127.1481404));
            bounds.extend(new kakao.maps.LatLng(35.8240808, 127.1481404));
        }
    });
}

setBounds(); // Set bounds after adding new markers
drawLine(positions);
}

```

시간	Day 1: 1월17일	Day 2: 1월18일
08 ~ 22:00		
08:00		
09:00		
10:00	1: 가회회관 X	
11:00		
12:00	2: 갑기회관 본점 X	
13:00		



해당 날짜에 등록된 일정이 없으면 전주시청 좌표를 중심으로 출력

DEVELOPMENT

주요기능
Main Skill

일정표 등록

contentsWrite의 지도를 응용

장소 등록하기

카워드 검색 후 목록에서 원하는 장소를 클릭해주세요.

입력될 장소: 북대



원하는 이름으로 장소 등록 가능

Day 1: 1월 18일

2 일차



목록 선택시 입력될 장소에 자동 입력

```
function onMarkerClick(marker, index) {
    setmarker.setMap(null);
    // 클릭한 마커의 정보를 출력하거나 원하는 작업을 수행합니다.
    var clickedPosition = marker.getPosition();
    var contentName = marker.getTitle();

    var contentsLatitude = clickedPosition.getLat();
    var contentsLongitude = clickedPosition.getLng();

    setmarker.setMap(map);
    setmarker.setPosition(clickedPosition);
    $('.searchedPlaedName').val(contentName);
    $('#contentsLatitude').val(contentsLatitude);
    $('#contentsLongitude').val(contentsLongitude);
}
```

```
function passPlace(){
    var latitude = $('#contentsLatitude').val();
    var longitude = $('#contentsLongitude').val();
    var searchedPlaedName = $('input[name=searchedPlaedName]').val();
    if(searchedPlaedName == ""){
        alert("입력할 장소의 이름을 작성해주세요");
        return;
    }else if(longitude == ""){
        alert("장소를 선택해주세요");
        return;
    }
    window.opener.addToTable(searchedPlaedName, latitude, longitude);
}
```

입력된 장소와 위/경도를 받아서 부모창의 addToTable에 넘긴다

DEVELOPMENT

주요기능

Main Skill

일정표 등록

일정 등록시 실행되는 함수

```
function goWrite() {  
  
    var fm = document.frm;  
    var tbodyCells = document.querySelectorAll('#dragDropTable tbody td');  
    var tourCourseTime  
    var tourCourseDate  
    var tourCourseNDate  
    var jsonArray = new Array();  
    var scheduleSubject = fm.scheduleSubject.value;  
    var scheduleStartDate = fm.scheduleStartDate.value;  
    var scheduleEndDate = fm.scheduleEndDate.value;  
    var scheduleShareYN = fm.scheduleShareYN.value;  
    //각 td 엘리먼트를 순회하면서 맵스트가 있는지 확인  
    tbodyCells.forEach(function(td) {  
        // 맵스트가 있는 경우  
  
        if ((td.textContent.trim() !== "") &&(td.textContent !== "장소를 누르시고 원하는 시간대에 드래그하세요")) {  
            // 원하는 작업 수행  
  
            var jsonObj = new Object();  
  
            var inputs = td.getElementsByTagName('input');  
            var tourCoursePlace = inputs[0].value;  
            var tourCourseLatitude = inputs[1].value;  
            var tourCourseLongitude = inputs[2].value;  
  
            nameArray = td.getAttribute('name').split('_');  
            tourCourseDate = nameArray[0];  
            tourCourseTime = nameArray[1];  
            tourCourseNDate = nameArray[2];  
  
            jsonObj.tourCourseDate = tourCourseDate;  
            jsonObj.tourCourseTime = tourCourseTime;  
            jsonObj.tourCourseNDate = tourCourseNDate;  
            jsonObj.tourCoursePlace = tourCoursePlace;  
            jsonObj.tourCourseLatitude = tourCourseLatitude;  
            jsonObj.tourCourseLongitude = tourCourseLongitude;  
  
            jsonObj = JSON.stringify(jsonObj);  
            jsonArray.push(JSON.parse(jsonObj));  
  
        }  
    });  
    let arrays = JSON.stringify(jsonArray);  
    모든 객체를 json 형식으로 만들고 scheduleWriteAction.do로 전달
```

	tourCourseDate*	tourCourseTime	tourCourseNDate
1	2023/12/29		8
2	2023/12/30		9
3	2023/12/30		10
4	2023/12/30		11
5	2023/12/27		9
6	2023/12/28		9
7	2023/12/29		9

각 td에 부여된 name

name="2024-01-17_11_1"

name="2024-01-18_11_2"

일정표 조회

여행 일정

제목	1박2일 여행하기	기간	2024-01-23 ~ 2024-01-24	조회수	33
시간	2024-01-23	2024-01-24			
08:00					
09:00	체육관				
10:00		가족회관			
11:00					
12:00	개운데집	견지산			
13:00					
14:00					
15:00					
16:00					
17:00					
18:00					



The map displays the travel route for Day 1, starting from point 1 (한국대학) and ending at point 2 (체육관). The total distance is 18.8km. Key locations marked include the National Museum of Korea, Gyeongbokgung Palace, Changdeokgung Palace, Deoksugung Palace, and the National Folk Museum of Korea.

- 여행 일정을 들어가면 1 일차 열이 강조되고 지도에서도 1일차 해당 일정 좌표와 경로가 출력된다.
- 테이블 셀 하나를 클릭하면 해당 일차가 강조되고 해당 일정좌표 와 경로가 출력된다.
- selectBox를 누르면 일정이 등록된 일차만 출력되고 일차를 누르면 해당 일차가 강조되고 해당 경로가 출력된다.
- 일정삭제는 작성자와 관리자만 가능하다.
- 여행 일정 다운로드가 가능하다.

DEVELOPMENT

주요기능
Main Skill

일정표 조회

일자 선택시 실행되는 함수

```
$('#selectDate').on('change', function () {
    hideMarkers();
    tourCourseNDate = $(this).val();
    getTourCourseNDate(sidx, tourCourseNDate);
    $('#nDate').val(tourCourseNDate + " 일자");
    $(".highlight").removeClass("highlight");
    $(".highlight2").removeClass("highlight2");
    $('#timetbl tr').find('td: eq(' + tourCourseNDate + ')').addClass('highlight');
    $('#timetbl th: eq(' + tourCourseNDate + ')').addClass('highlight2');
});
```

```
$('#timetbl td').on('click', function () {
    var rowIndex = $(this).parent().index();
    var columnIndex = $(this).index();

    if (columnIndex === 0) {
        return;
    }
    $(".highlight").removeClass("highlight");
    $(".highlight2").removeClass("highlight2");
    // 첫 번째 행 또는 첫 번째 열인 경우 이벤트 발생하지 않도록 처리

    $('#timetbl tr').find('td: eq(' + columnIndex + ')').addClass('highlight');
    $('#timetbl th: eq(' + columnIndex + ')').addClass('highlight2'); // 수정된 부분
    hideMarkers();
    tourCourseNDate = columnIndex; // 수정된 부분
    // n일자 지도 출력
    getTourCourseNDate(sidx, tourCourseNDate);
    $('#nDate').val(tourCourseNDate + " 일자");
});
```

```
function getTourCourseNDate(sidx, tourCourseNDate) {
    $.ajax({
        type: "post",
        url: "${pageContext.request.contextPath}/schedule/getTourCourseNDate.do",
        data: {
            "sidx": sidx,
            "tourCourseNDate": tourCourseNDate
        },
        dataType: "json",
        success: function (data) {
            var positions = [];

            $(data).each(function () {
                positions.push({
                    title: this.tourCoursePlace,
                    latlng: new kakao.maps.LatLng(this.tourCourseLatitude, this.tourCourseLongitude)
                });
            });

            if(positions.length != 0){
                clearBounds();

                for (var i = 0; i < positions.length; i++) {
                    addMarker(positions[i].latlng, positions[i].title, (i + 1));
                    bounds.extend(positions[i].latlng);
                }

                setBounds();
                drawline(positions);
            }
            else{
                polyline.setMap(null);
                distanceOverlay.setMap(null);
                distanceOverlay = null;
            }
        }
    });
}
```

테이블의 셀이 클릭되거나 selectBox가 바뀔 때 해당 함수 실행,
ajax를 이용하여 바로 해당 일자의 위도경도가 배열에 담겨져 맵에 출력되게 함

DEVELOPMENT

주요기능
Main Skill

일정표 조회

```
function getTourCourse(sidx){
    $.ajax({
        type : "post",
        url : "${pageContext.request.contextPath}/schedule/getTourCourse.do",
        data : {
            "sidx" : sidx
        },
        dataType : "json",
        success : function(data){
            $(data).each(function(){
                var td_id = "#" + this.tourCourseDate + "_" + this.tourCourseTime;
                $(td_id).html(this.tourCoursePlace);
            });
        },
        error: function(request, status, error){
            alert("code:"+request.status+"\n"+ "message:" +request.responseText+"\n"+ "error:" +error);
        }
    });
}
```

scheduleController.java – boardContents()

```
String scheduleStartDate = "";
String scheduleEndDate = "";
scheduleStartDate = sv.getScheduleStartDate();
scheduleEndDate = sv.getScheduleEndDate();

LocalDate startDate = LocalDate.parse(scheduleStartDate, formatter);
LocalDate endDate = LocalDate.parse(scheduleEndDate, formatter);

long differenceInDays = endDate.toEpochDay() - startDate.toEpochDay() +1;
int totalCnt = (int) differenceInDays;

ArrayList<LocalDate> dateList = new ArrayList<>();

for (LocalDate currentDate = startDate; !currentDate.isAfter(endDate); currentDate = currentDate.plusDays(1)) {
    dateList.add(currentDate);
}
model.addAttribute("dateList", dateList);
```

시작	2024-01-16	2024-01-17	2024-01-18	2024-01-19
08:00				
09:00	가족여행			
10:00				
11:00				
12:00	힐링여행			
13:00	힐링여행			
14:00		힐링여행		
15:00				
16:00				
17:00				

일자와 시간에 맞는 일정이 들어가기 위해 td_id를 일자_시간으로 부여하고 jquery를 사용하여 해당 td에 해당 html부여

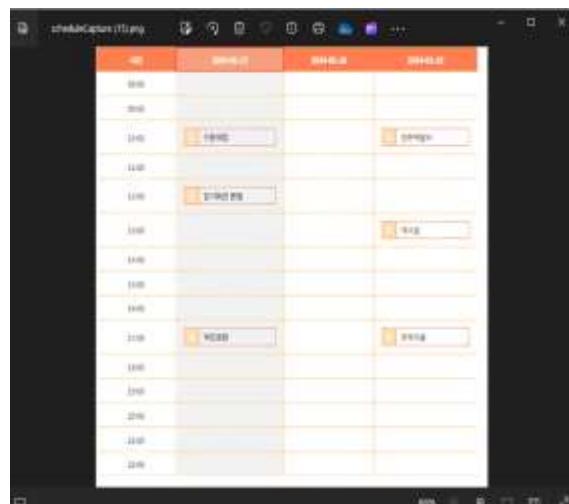
```
<tbody>
<c:forEach var="hour" begin="8" end="22">
    <tr>
        <c:choose>
            <c:when test="${hour lt 10}">
                <td>0${hour}:00</td>
            </c:when>
            <c:otherwise>
                <td>${hour}:00</td>
            </c:otherwise>
        </c:choose>
        <c:forEach var="dl" items="${dateList}">
            <td id="${dl}_${hour}"></td>
        </c:forEach>
    </tr>
</c:forEach>
</tbody>
```

시작날짜와 끝나는 날짜를 가지고와서 dateList 배열을 만들고 여행일정을 넣어줌으로써 일정 테이블의 th생성

DEVELOPMENT

주요기능
Main Skill

일정표 조회



```
<script src="https://html2canvas.hertzen.com/dist/html2canvas.js"></script>
```

```
<script>
function download(canvas, filename) {
    const data = canvas.toDataURL("image/png;base64");
    const downloadLink = document.querySelector("#download");
    downloadLink.download = filename;
    downloadLink.href = data;
}

const targetElement = document.getElementById("scheduletbl");
//로드 되는거 좀 기다려야함
setTimeout(() => {
    html2canvas(targetElement).then((canvas) => {
        download(canvas, "scheduleCapture");
    });
}, 500);
</script>
```

Javascript의 HTML2Canvas 라이브러리를 사용
하여 HTML 콘텐츠를 스크린샷을 찍음

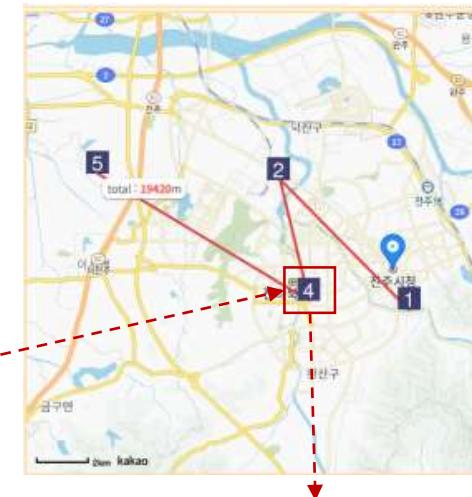
DEVELOPMENT

주요기능
Main Skill

일정표 조회

scheduleContents-map.js

```
var markers = [];  
  
function addMarker(position,placeName,index) {  
    // 마커 이미지 생성  
    var markerImage = new kakao.maps.MarkerImage(  
        'https://placehold.jp/23/3d4070/ffffff/25x25.png?text=' + (index),  
        new kakao.maps.Size(25, 25),  
        { offset: new kakao.maps.Point(13, 25) }  
    );  
  
    // 마커를 생성합니다  
    var marker = new kakao.maps.Marker({  
        position: position,  
        image : markerImage,  
        title : placeName  
    });  
  
    // 마커가 지도 위에 표시되도록 설정합니다  
    marker.setMap(map);  
  
    // 생성된 마커를 배열에 추가합니다  
    markers.push(marker);  
}  
  
//마커 생성 혹은 삭제  
function setMarkers(map) {  
    for (var i = 0; i < markers.length; i++) {  
        markers[i].setMap(map);  
    }  
}  
  
function setInfoWindows(map) {  
    for (var i = 0; i < infowindows.length; i++) {  
        infowindows[i].setMap(map);  
    }  
}
```



Parameter로 index값을 placeholder url 뒤쪽에 추가해서 해당 마커의 이미지에 인덱스를 포함하여 마커에도 순서를 보여줌

DEVELOPMENT

주요기능
Main Skill

일정표 조회

scheduleContents-map.js

```
function drawLine(positions) {
    var clickPosition = [];

    for (var i = 0; i < positions.length; i++) {
        clickPosition.push(positions[i].latlng);
    }

    // 선이 있으면 지울
    if (polyline) {
        polyline.setMap(null);
    }

    // clickPosition 배열이 비어있으면 content를 "0m"로 설정
    var content;
    if (clickPosition.length === 0) {
        content = '<div class="dotOverlay distanceInfo">total : <span class="number">0</span>m</div>';

        showDistance(content,new kakao.maps.LatLng(35.8240808, 127.1481404));
    } else {
        // 새로운 polyline 생성
        polyline = new kakao.maps.Polyline({
            map: map,
            path: clickPosition,
            strokeWeight: 3,
            strokeColor: '#db4040',
            strokeOpacity: 1,
            strokeStyle: 'solid'
        });

        polyline.setMap(map);

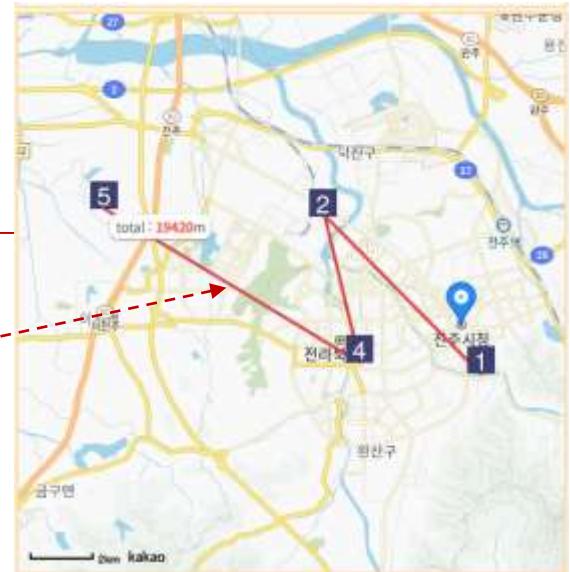
        // 선의 총 거리를 계산
        var distance = Math.round(polyline.getLength());

        // content 생성
        content = '<div class="dotOverlay distanceInfo">total : <span class="number">' + distance + '</span>m</div>';

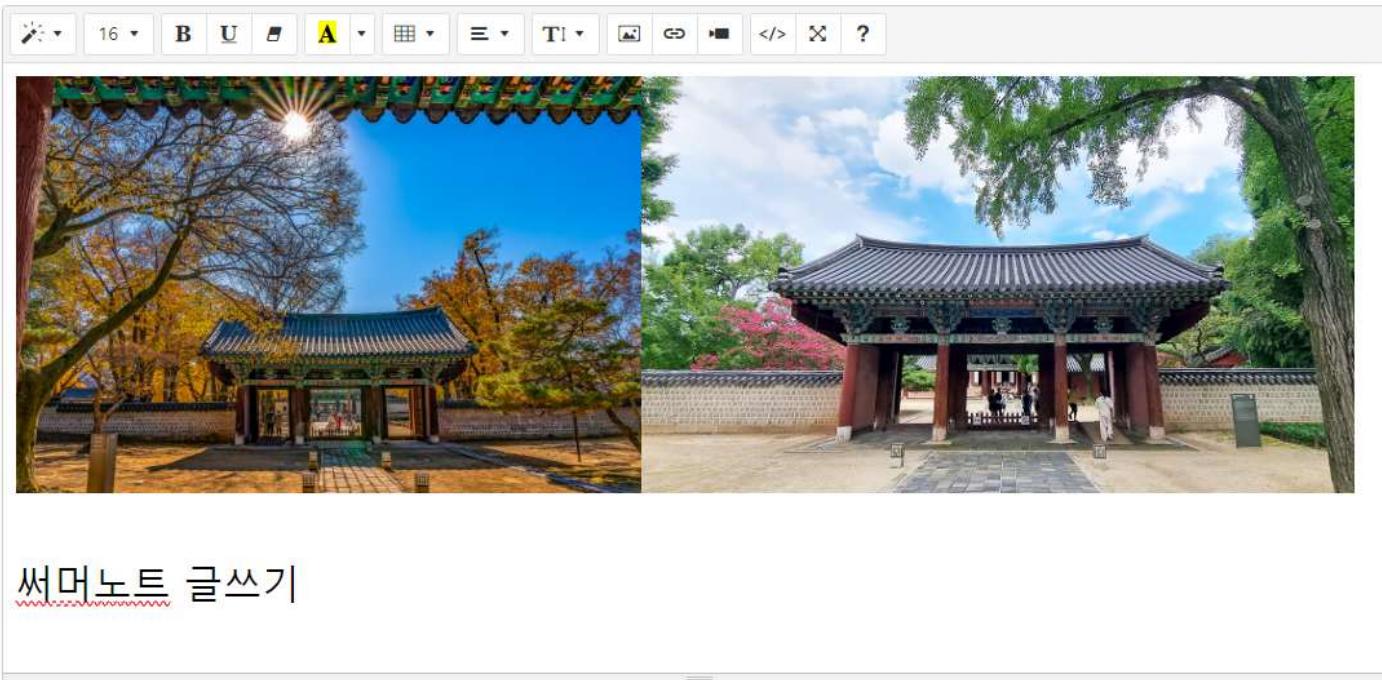
        // 각각의 점에 대해 거리 표시
        for (var i = 0; i < clickPosition.length; i++) {
            showDistance(content, clickPosition[i]);
        }
    }

    // 결과로 얻은 content를 어디가에 사용할 수 있음
}
```

경로를 출력하는 함수

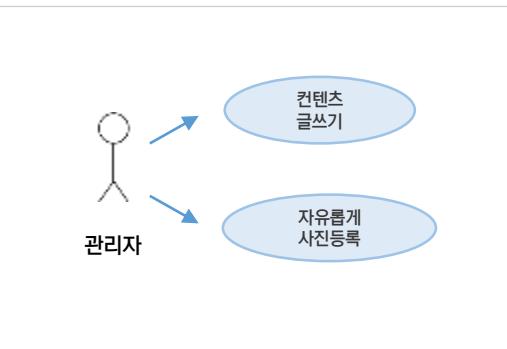


써머노트



The screenshot shows a rich-text editor interface with a toolbar at the top containing various styling options like bold, italic, underline, and alignment tools. Below the toolbar are two images of traditional Korean buildings (Hanok) with tiled roofs and wooden structures, set against a backdrop of trees and a clear blue sky.

써머노트 글쓰기



- 컨텐츠 테이블의 컬럼 contentsArticle에 저 장될 내용은 써머노트를 이용해서 사진과 내용을 자유롭게 저장한다.
- 외부 사진과 표 등을 붙여넣기로 편하게 옮길 수 있다.

DEVELOPMENT

주요기능
Main Skill

써머노트

contentsWrite.jsp

```
<!-- include summernote css/js -->
<link href="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote-lite.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote.min.js"></script>
<script src="../summernote/summernote-ko-KR.js"></script>
```

CDN 방식으로 연결

```
<script type="text/javascript">
$(document).ready( function() {
    // 가져올때 navbar.css도 같이 가져올 것
    $('#headers').load("../nav/nav.jsp");

    $('#summernote').summernote({
        lang: 'ko-KR', // default: 'en-US'
        toolbar: [
            // 스타일 관련 기능
            ['style', ['style']],
            // 글자 크기 설정
            ['fontsize', ['fontsize']],
            // 글꼴 스타일
            ['font', ['bold', 'underline', 'clear']],
            // 글자 색상
            ['color', ['color']],
            // 테이블 삽입
            ['table', ['table']],
            // 문단 스타일
            ['para', ['paragraph']],
            // 에디터 높이 설정
            ['height', ['height']],
            // 이미지, 링크, 동영상 삽입
            ['insert', ['picture', 'link', 'video']],
            // 코드 보기, 전체화면, 도움말
            ['view', ['codeview', 'fullscreen', 'help']],
        ],
        fontSizes: [
            // 글자 크기 선택 옵션
            '8', '9', '10', '11', '12', '14', '16', '18', '20', '22', '24', '28', '30', '36', '50', '72'
        ],
        styleTags: [
            // 스타일 태그 옵션
            'p',
            { title: 'Blockquote', tag: 'blockquote', className: 'blockquote', value: 'blockquote' },
            'pre',
            { title: 'code_light', tag: 'pre', className: 'code_light', value: 'pre' },
            { title: 'code_dark', tag: 'pre', className: 'code_dark', value: 'pre' },
            'h1', 'h2', 'h3', 'h4', 'h5', 'h6'
        ]
    });
});
```

<textarea id="summernote" name="contentsArticle"></textarea>

원하는 위치에 summernote 연결

툴바에 추가할 옵션들

PK	열	데이터 유형
	cidx	int
	contentsCategory	varchar(5)
	contentsSubject	varchar(30)
	contentsArticle	longblob
	contentsViewCount	int
	contentsReviewCount	int
	contentsWriteday	timestamp
	contentsYN	varchar(2)
	midx	int
	contentsLatitude	varchar(20)
	contentsLongitude	varchar(20)
	contentsHashtag	json
	aidx	int

↓

DB엔 longblob 타입으로 저장

DEVELOPMENT

확장가능성
Scalability



Spring security



API를 활용한 컨텐츠 글쓰기



해시태그 활용



피신고자 패널티

프로젝트 시연

<http://192.168.0.30:8080/>

Q & A

문다온

한달이라는 짧은 기한 안에 많은 기술들을 사용해 볼 수 있어서 재미있는 프로젝트였습니다. 팀원들도 각자 맡은 파트에서 열심히 구현해줘서 결과적으로 꽤 괜찮은 프로젝트가 완성된 것 같습니다. 시간이 좀 더 있었으면 하는 아쉬움은 있지만 그만큼 역량을 파악할 수 있어서 좋은 기회였습니다.

김민정

맡은 파트를 성공적으로 구현해낸 것이 가장 보람차고 다양한 기능들을 찾아보고 시도해보면서 많이 배우고 성장한 것 같습니다. 짧은 기간이었지만 팀원들의 노력으로 예상했던 것보다 더 완성도가 높은 결과물을 얻은 것 같아 뿌듯합니다.

박건도

스프링으로 활용한 프로젝트를 하면서 스프링에 대한 전반적인 부족한 지식을 스스로 깨닫는 되는 시간이었습니다. 더불어 프론트에 대한 이해와 고충을 알 수 있었습니다. 짧은 프로젝트 기간이었지만, 첫 프로젝트만큼 많은 것을 배울 수 있었기에 만족스러운 시간이었습니다.

윤대희

짧은 기한 내 구현하고자 하는 기능은 완성해내 다행이고 짧은 기한에 비해 숙련도가 많이 올라간거 같아 보람찹니다. 다만 아쉬운점은 스스로 생각하는 시간이 짧았습니다. 또한 익숙한 방법으로 해결하려 했던 점이 새로운 스킬을 익히게 하지 못한 거 같아 다소 아쉽습니다.

○ Ezen_Jeonju

감사합니다.

Ezen_Developer