

제주특별자치도

관광 안내

홈페이지 제작

발표일

2021년 11월 03일

개발담당

스프링클라우드 2팀

오수빈(팀장), 최웅빈, 최두식, 최진호, 양승모, 진희



목차

스프링클라우드 2팀

제주특별자치도 관광 안내 홈페이지



01

프로젝트 소개

팀원 소개	03
프로젝트 설명	04

02

문서

ERD	05
유스케이스 다이어그램	06
간트차트	07
깃허브	08

03

페이지 소개

메인 페이지	11
로그인 페이지	17
관리자 페이지	18
게시판	24
여행 경로 페이지	30

04

기능 소개

관리자 페이지	36
마이 페이지	41
로그인 페이지	43
게시판	49
여행 경로 페이지	54
폴리곤	62

01

프로젝트 소개

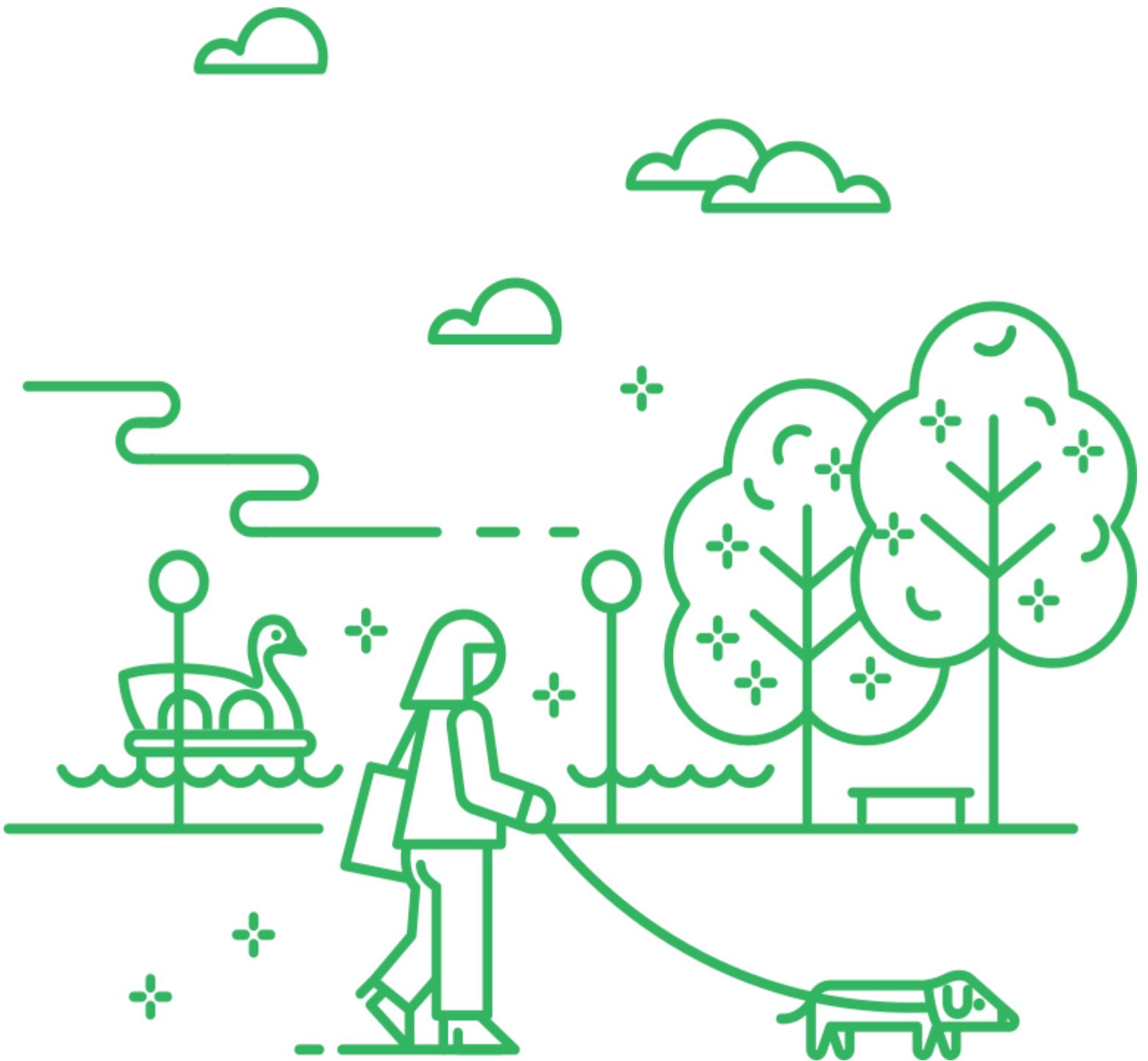
Project introduction

1. 팀원 소개

03

2. 프로젝트 설명

04



1. 팀원 소개

업무 분장

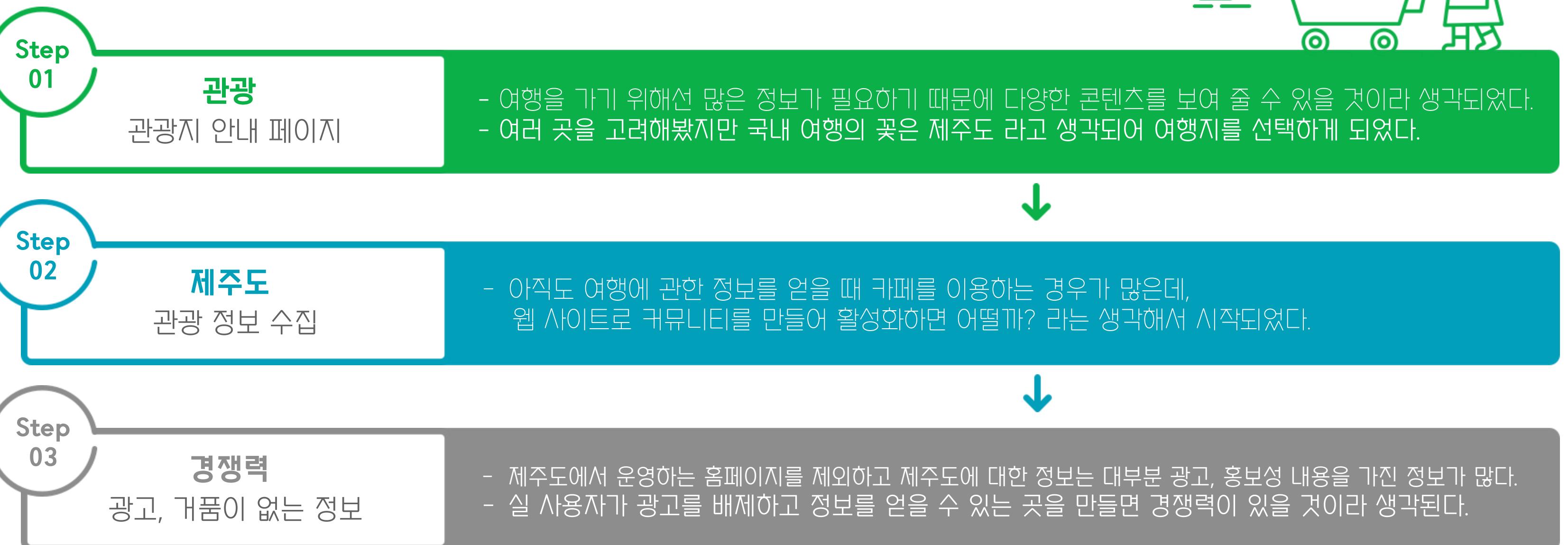
오수빈(팀장), 최웅빈, 최두식, 최진호, 양승모, 진희



이름	직책	업무내용
오수빈	팀장 / 1팀	로그인 / 마이 페이지 / 관리자 페이지 (오수빈 - 최종 검수)
최진호	팀원 / 1팀	
양승모	팀원 / 2팀	게시판 / 게시판 글 작성
진희	팀원 / 2팀	
최웅빈	팀원 / 3팀	맛집 / 관광지 여행 경로 페이지 게시판 페이지 수정
최두식	팀원 / 3팀	

2. 프로젝트 설명

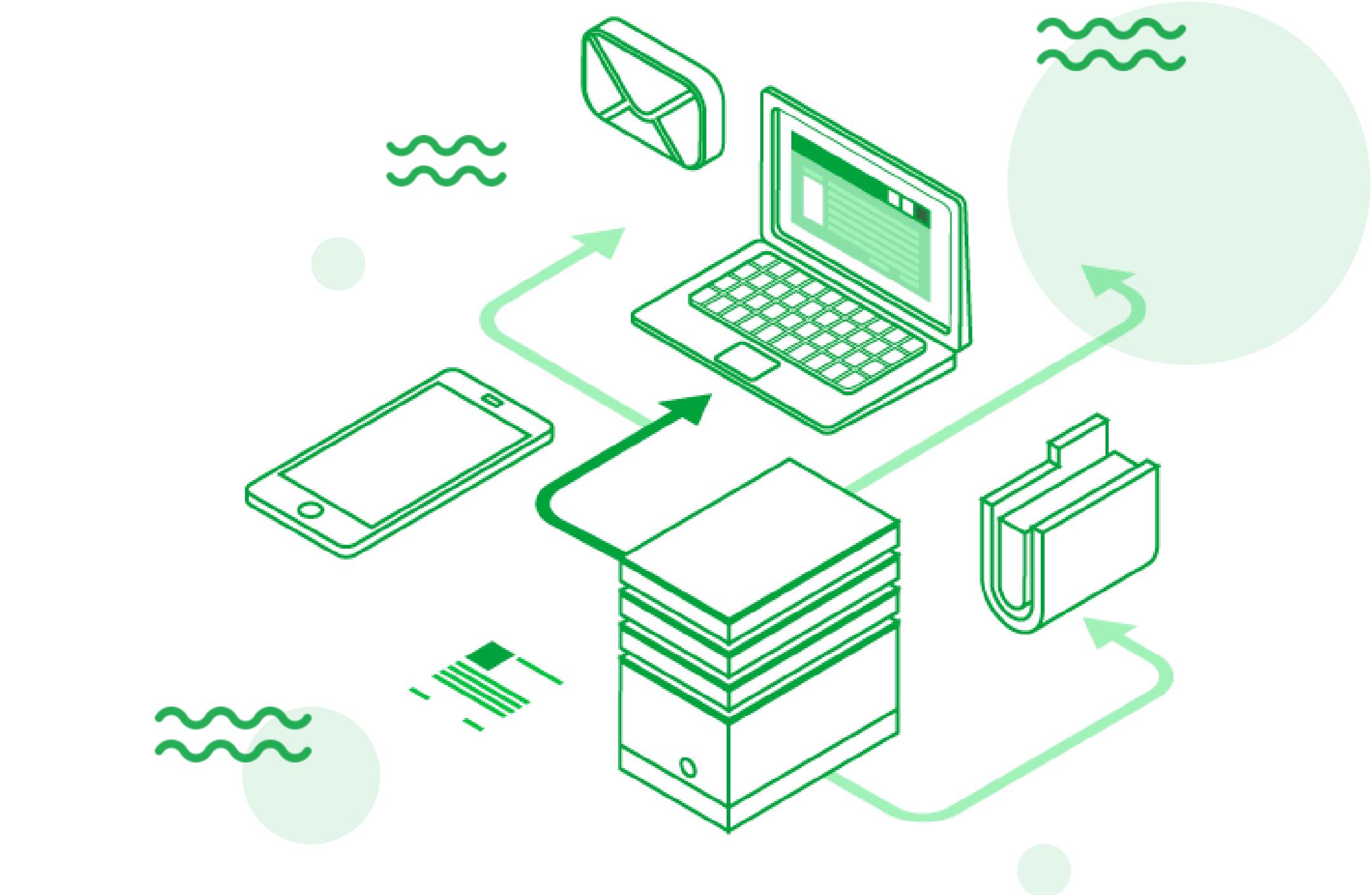
키워드 선정 과정



02 문서

Database, GitHub

1. ERD
2. 유스케이스
3. 간트차트
4. GitHub



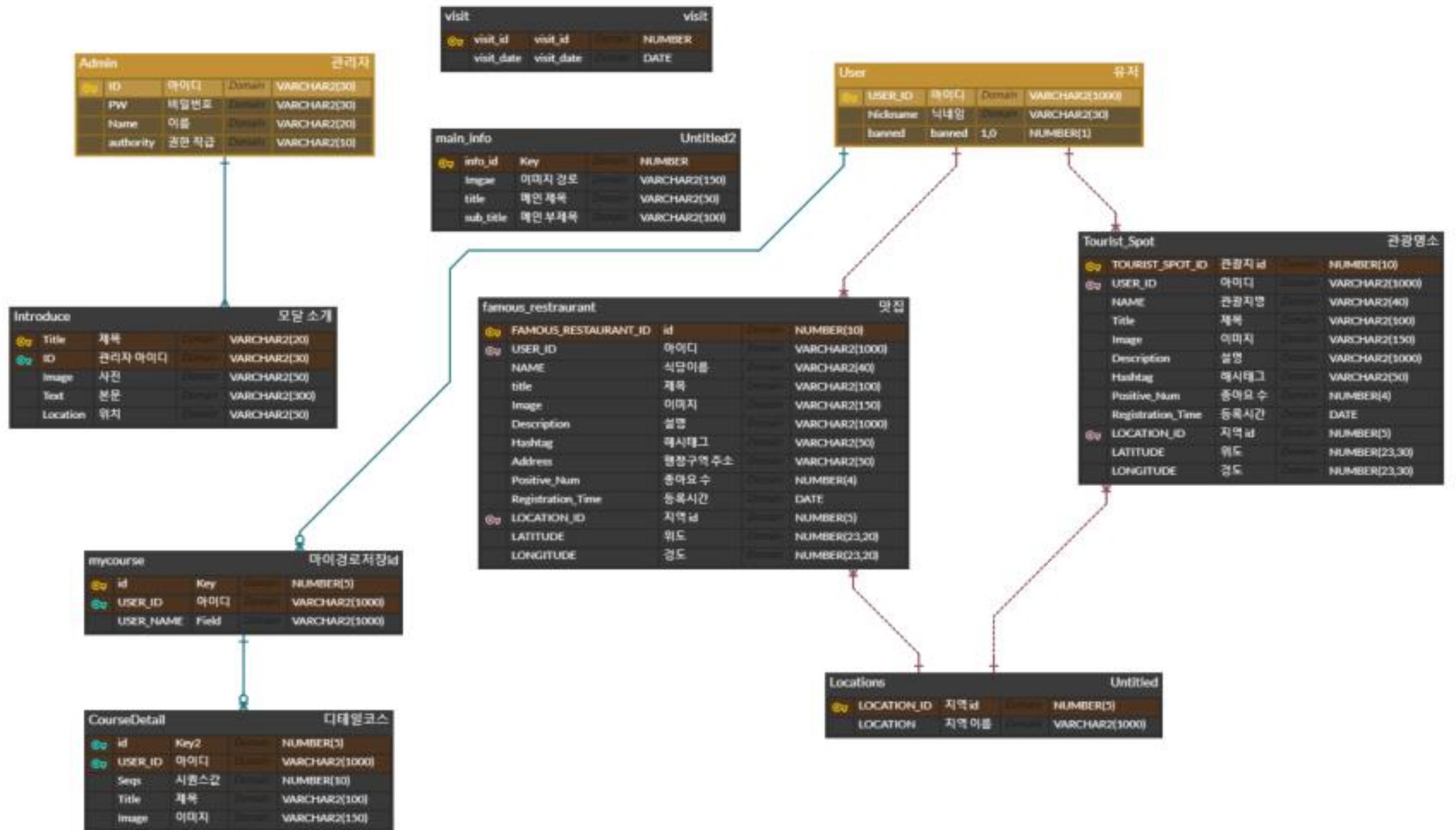
06

07

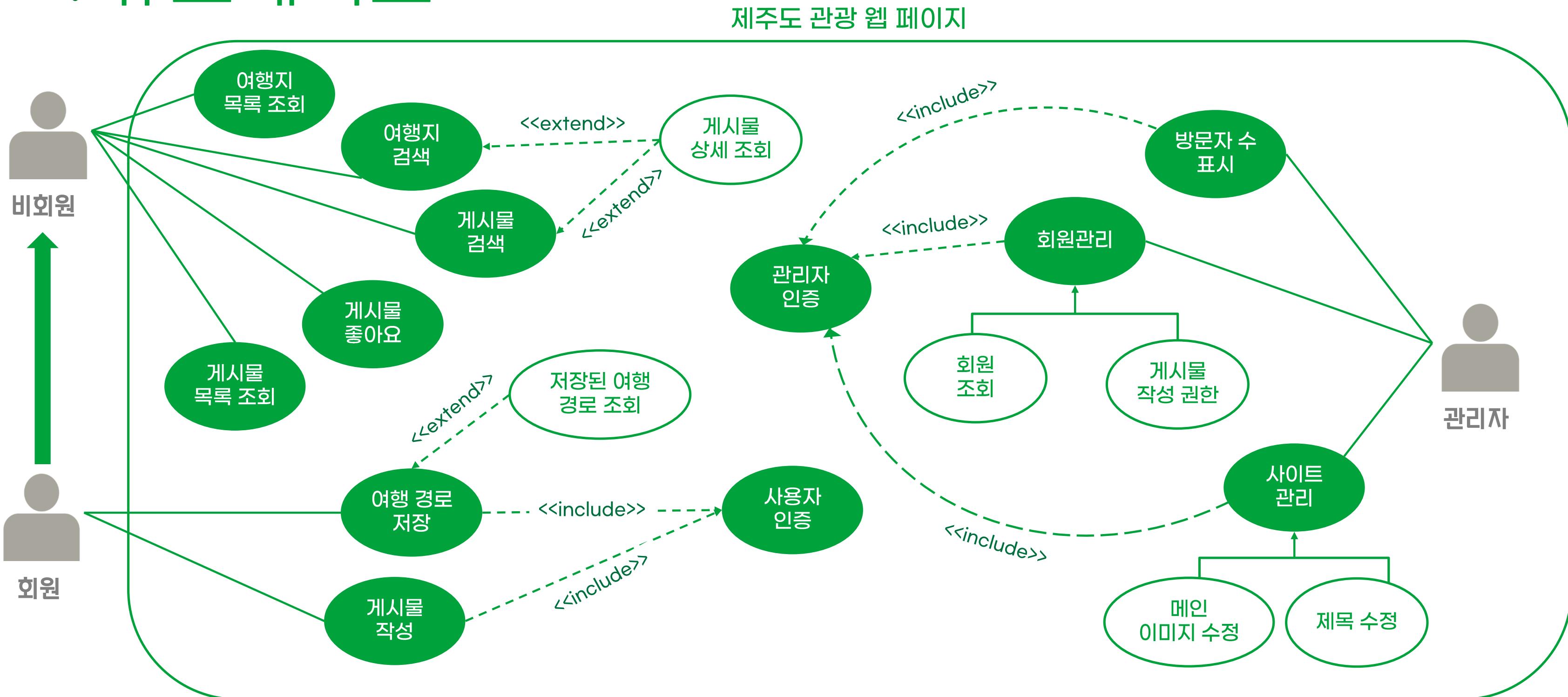
08

09

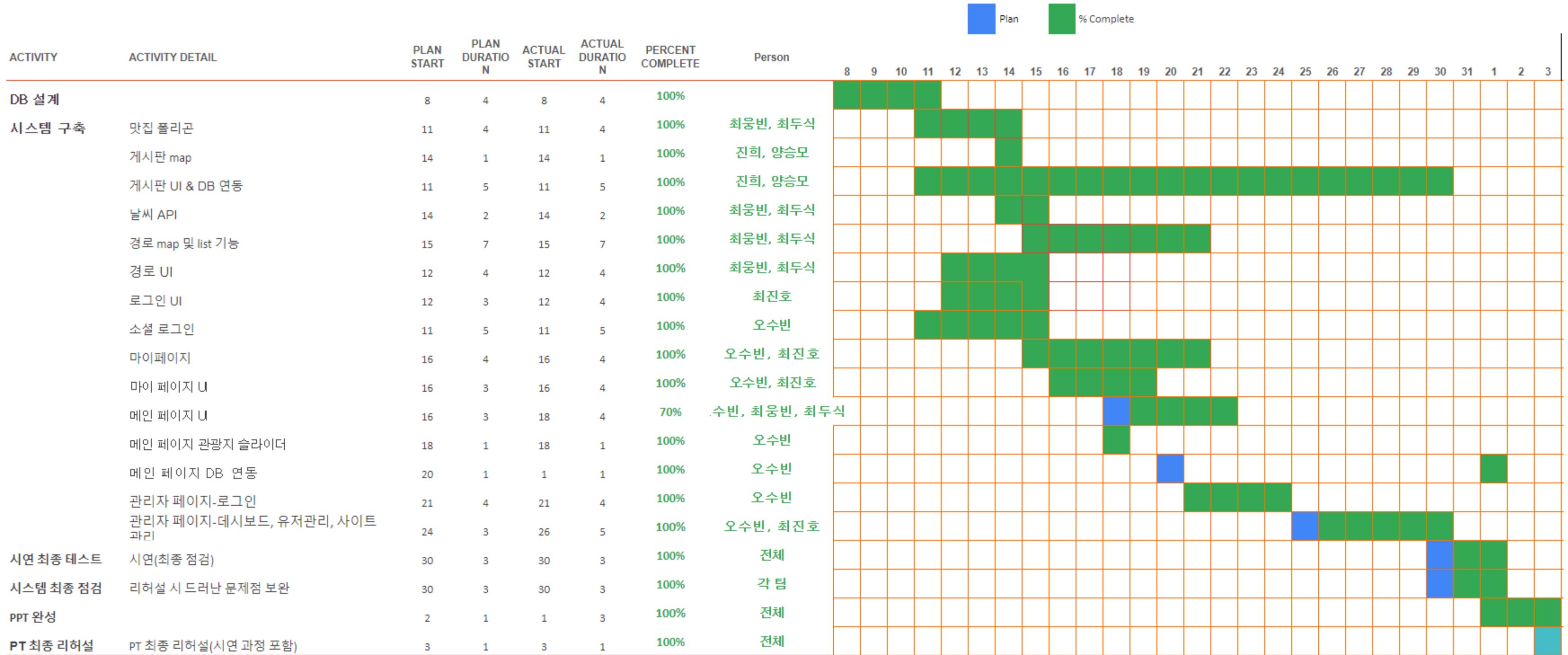
1. ERD



2. 유스케이스



3. 간트차트



4. GitHub

▼ GitHub Issues

<input type="checkbox"/> Author ▾	<input type="checkbox"/> Label ▾	<input type="checkbox"/> Projects ▾	<input type="checkbox"/> Milestones ▾	<input type="checkbox"/> Assignee ▾	<input type="checkbox"/> Sort ▾
<input type="checkbox"/>	<input checked="" type="checkbox"/> admin 상단 링크 미적용 수정 enhancement				
	#126 by cjh0919 was closed 15 hours ago				
<input type="checkbox"/>	<input checked="" type="checkbox"/> 게시판 저장기능 구성 enhancement				
	#118 by Doosic was closed 18 hours ago				
<input type="checkbox"/>	<input checked="" type="checkbox"/> Feature/23 main UI bug duplicate enhancement				
	#114 by ohsoou was merged 2 days ago				
<input type="checkbox"/>	<input checked="" type="checkbox"/> Feature/101 admin site bug duplicate enhancement				
	#102 by ohsoou was merged 4 days ago				
<input type="checkbox"/>	<input checked="" type="checkbox"/> admin manage site enhancement				
	#101 by ohsoou was closed 4 days ago				
<input type="checkbox"/>	<input checked="" type="checkbox"/> logout button add & admin side bar UI 수정 #99 enhancement				
	#100 by cjh0919 was merged 4 days ago				
<input type="checkbox"/>	<input checked="" type="checkbox"/> logout button add & admin side bar UI 수정 enhancement				
	#99 by cjh0919 was closed 2 days ago				
<input type="checkbox"/>	<input checked="" type="checkbox"/> add totalCount duplicate enhancement				
	#97 by ohsoou was merged 5 days ago				
<input type="checkbox"/>	<input checked="" type="checkbox"/> Feature/91 admin manage user bug enhancement				
	#96 by ohsoou was merged 5 days ago				
<input type="checkbox"/>	<input checked="" type="checkbox"/> 관리자 페이지 user ban 기능 enhancement				
	#91 by ohsoou was closed 5 days ago				
<input type="checkbox"/>	<input checked="" type="checkbox"/> Feature/19 mypage db bug duplicate enhancement				
	#90 by ohsoou was merged 5 days ago				

Id	Message
af52a13	[origin/update/1...] Merge branch 'dev' into update/131_이용방법_변경_및_링크
924e440	모달 크기 변경 및 이미지 변경 완료
ed2fc8d	모달 변경 및 수정사항 적용
5553358	delete text
79cba20	unstaged form.js
10a2d24	Merge pull request #130 from JejuislandToyProject/update/123_이미지_경로_및_location_수정
01d854b	Merge branch 'dev' into update/123_이미지_경로_및_location_수정
a867764	delete remember me button
2b1a343	Merge branch 'dev' of https://github.com/JejuislandToyProject/Jejulsland.git into dev
8fa7dd4	게시판 파일 업로드 완료
d356d0f	Merge branch 'dev' into admin_수정/#126
f67264c	Merge pull request #128 from JejuislandToyProject/feature/fix-error
ed866eb	반응형 UI 수정
a86c441	o
4d14cbc	반응형 UI 수정
9af195e	Merge branch 'ㅁㅇㅇㄴㅇ' into ㅁㅇㅇ
b8f3d32	좌측상단 관리자 페이지 링크 페이지 미적용 수정
11d5c7f	커밋
4da8ea9	list.jsp삭제/배너이미지 변경/검색바 수정/상세페이지
4b8bb74	1차 커밋
88e06f6	o o
67a8100	location값 수정 완료
ef7f42f	fix rest error
cd0e6cd	이미지 경로 수정 완료
5b8f586	Merge branch 'dev' into update/113_게시판_저장_DB연동
8ec30bb	게시판 작성기능 완료.
e2dacf7	Merge pull request #121 from JejuislandToyProject/113_board/boardViewUI
b5fe94d	(origin/113_boar...) 게시판/상세페이지 UI수정
191e356	한글 깨짐 수정

▲ branch

03

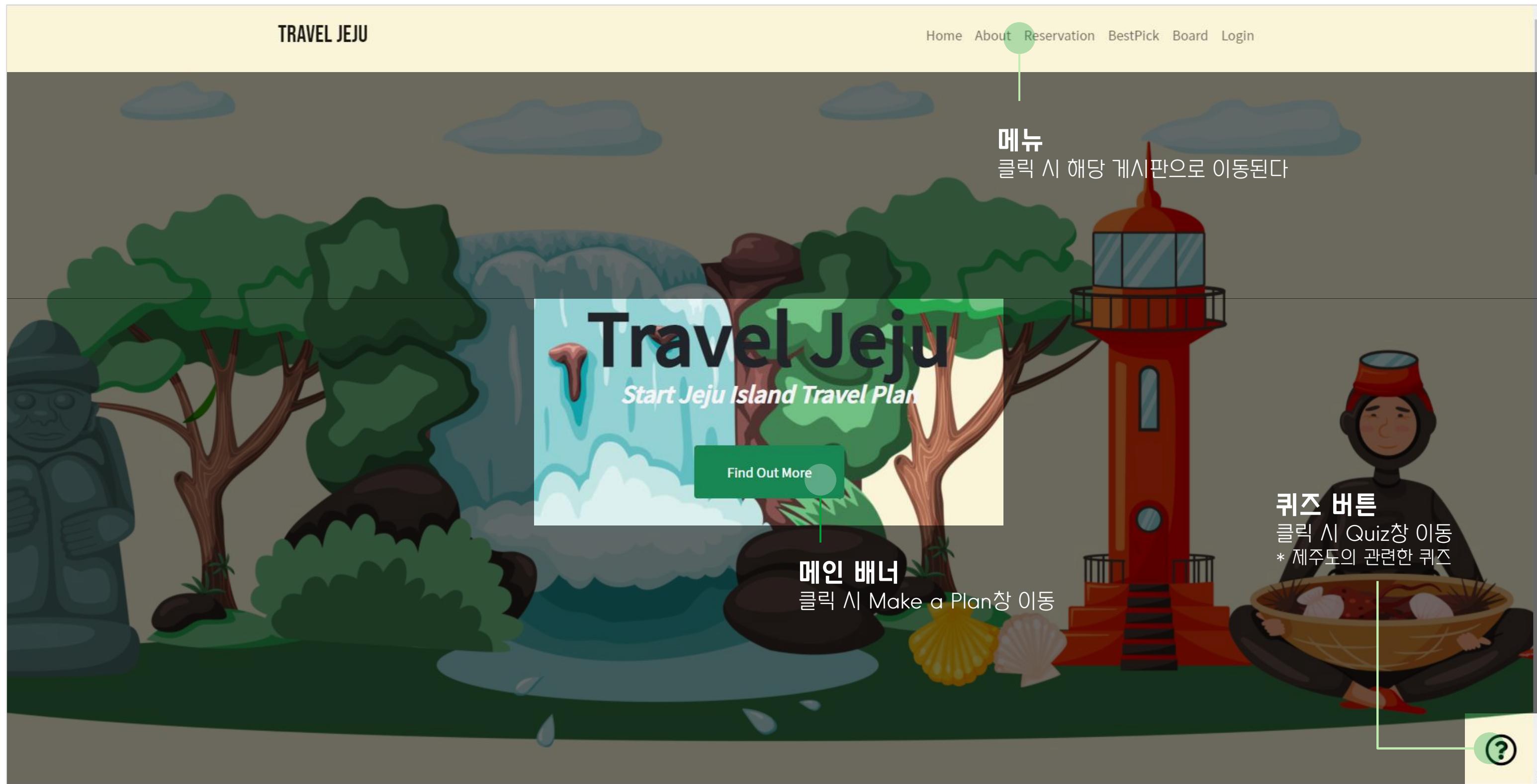
페이지 소개

Web Homepage

- | | | |
|--------------|-------|----|
| 1. 메인 페이지 | | 11 |
| 2. 로그인 페이지 | | 17 |
| 3. 관리자 페이지 | | 18 |
| 4. 게시판 | | 24 |
| 5. 여행 경로 페이지 | | 30 |



1. 메인 페이지



1. 메인 페이지

나는 제주도를 얼마나 알고 있을까?
나의 제주도 지식을 테스트 해보세요!

JEJU Quiz

퀴즈
제주도를 사랑하는 사람이라면
맞출 수 있는 문제

SEARCH
Travel Course



Kakao Map으로 보는 내 여행 경로
제주도 여행으로의 한걸음 설레는 마음으로 계획을 세워보세요!
다른 사람들이 올린 맛집과 관광지를 바탕으로
나만의 여행 경로를 만들고 저장해보세요

Make a Plan

여행코스
클릭 시 Make a Plan창 이동

1. 메인 페이지



행정 구역 별 맛집 찾기
제주도 행정 구역별 대표 맛집을
모달 창으로 띠워준다.

행정구역별 맛집 찾기

세분화된 맛집 로드뷰를 바탕으로 맛집투어를 기획해보세요!
신빙성 있는 자료를 위해 정회원 이상만 등록 가능합니다

× 정회원 등록하러 가기!

Seongsan
서귀포시 성산읍
제주도의 주요 관광지로써 성산일출봉과 성산항 등이 있고, 성산항의 배를 타고 우도면으로 이동할 수 있다. 대표적인 맛집으로는 부춘식당, 그리운 바다 성산포, 해贻 등이 있다

👍
⚡
🌐

상세보기

수 많은 관광지 속 숨은 진주까지 프로계획러들을 위한 서비스!






?

제주도 관광지 안내 홈페이지 제작

1. 메인 페이지

The image shows the homepage of a travel website for Jeju Island. The title "A trip to Jeju Island" is at the top, with a "MAKE PLAN" button above it. Below the title are four main service sections:

- AIRPLANE:** "최저가 항공권 예매하러 가기!" (Go to book the lowest airfare!). It features an illustration of an airplane flying over clouds.
- ACCOMMODATION:** "숙박시설 한번에 비교하여 최고의 예약을!" (Compare accommodation facilities and make the best reservation!). It shows a hotel building with a magnifying glass over a smartphone displaying a map.
- RENTAL CARS:** "제주도 렌트카 실시간 가격비교" (Real-time price comparison of Jeju Island rental cars). It features an illustration of a woman standing next to a car with a smartphone showing a map.
- A TOURIST ATTRACTION:** "제주도 관광명소&맛집 한번에 알아보기" (Learn about Jeju Island's tourist attractions and restaurants). It shows a map of Jeju Island with various icons representing attractions.

유용한 사이트
제주도의 항공권, 숙박시설,
렌트카, 관광 명소 및 맛집을
한번에 알아보고,
바로 확인할 수 있는 창

1. 메인 페이지



행정구역별 맛집 찾기
세분화된 맛집 로드뷰를 바탕으로 맛집투어를 기획해보세요!
신빙성 있는 자료를 위해 정회원 이상만 등록 가능합니다

정회원 등록
클릭 시 로그인 페이지로 이동 [정회원 등록하러 가기!](#)

관광 명소 찾기
제주도의 숨은 관광지를
이미지 슬라이드로 보여준다.

제주도 관광 명소 찾기
수 많은 관광지 속 숨은 진주까지 프로계획러들을 위한 서비스!



[?](#)

1. 메인 페이지

 이번 달 Best Pick



이번 달 맛집 : 우진해장국
loaction : 제주특별자치도 제주시 특별자치도 삼도2동 831



이번 달 관광지 : 빛의 병카
loaction : 제주특별자치도 서귀포시 특별자치도 성산읍 고성리 2039-22

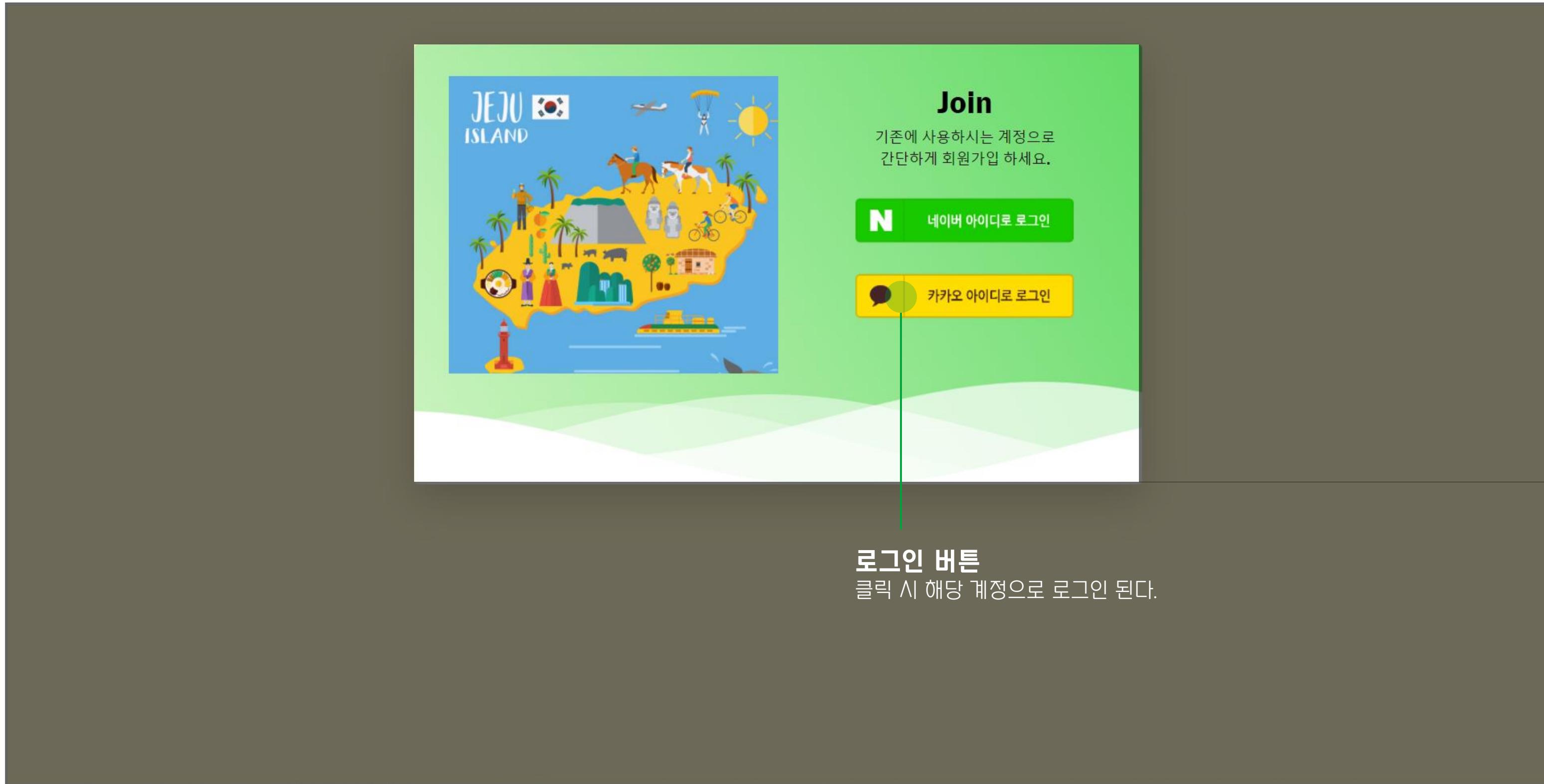
이번 달 Best Pick

한달 간 좋아요 수가 가장 많은 게시물
클릭 시 해당 게시물로 이동

Jeju Island

제주도 관광지 안내 홈페이지 제작

2. 로그인 페이지



3. 관리자 페이지

관리자 페이지

- Dashboard**
- INTERFACE**
 - 사이트 관리**
 - 회원 관리**
- LOGOUT**

대시보드
대시보드로 이동
(방문자 수 카운트)

오늘 방문자 수
47

총 방문자 수
267

사이트 관리
메인 페이지 관리를 담당
메인 배너/타이틀 변경이 가능하다

일간 방문자 수 그래프

날짜	방문자 수
2021/10/26	0명
2021/10/27	50명
2021/10/28	100명
2021/10/29	90명
2021/10/30	40명
2021/11/01	110명
2021/11/02	60명

회원관리
회원 정보 / 회원 권한을
관리하는 페이지이다.

네비게이션 메뉴
버튼 클릭 시 메뉴 크기 조정

Copyright © Jeju Website 2021

localhost:8081/jeju/admin/index.html

3. 관리자 페이지

관리자 페이지

- Dashboard
- INTERFACE
 - 사이트 관리
 - 회원 관리

LOGOUT

localhost:8081/juju/admin/index.html

오늘 방문자 수
47

총 방문자 수
267

방문자 수
일일 방문자 수 / 총 방문자수 표시

일간 방문자 수 그래프

날짜	방문자 수
2021/10/26	0명
2021/10/27	50명
2021/10/28	48명
2021/10/29	45명
2021/10/30	35명
2021/11/01	100명
2021/11/02	50명

일간 방문자 수
일주인 간 방문자 수를 나타낸다.
Session을 통해 로그인한 사용자는
하루에 여러 번 방문해도 카운트 되지 않는다.

Copyright © Jeju Website 2021

3. 관리자 페이지

사이트 관리

메인페이지 관리

Visual Image
./resources/img/bg-masthead 업로드

Visual Title
Travel Jeju

Visual Sub Title
이미지 업로드
기준의 배너를 수정할 수 있다

메인 배너 미리보기
삽입할 이미지를 미리 볼 수 있도록
띠워준다.

저장

This screenshot shows the 'Main Page Management' section of the administrator's page. It displays a large, colorful cartoon illustration of Jeju Island's natural beauty, including a waterfall, a red lighthouse, a traditional stone statue, and a person sitting cross-legged with a basket of seafood. Below the image, there are fields for 'Visual Image' (with a file path ./resources/img/bg-masthead), 'Visual Title' (Travel Jeju), and 'Visual Sub Title' (which includes a note about uploading banner images). A descriptive text box on the right explains the preview feature for the main banner. At the bottom right is a 'Save' button.

3. 관리자 페이지



3. 관리자 페이지

관리자 페이지

회원 관리

회원 리스트

Show 10 entries

NickName	ID	Lock & UnLock
AtlierQ	HP8SXtJEZrSLFdCT45c4QTgTX2-ghsCcy9SXdcy/Jlg	<input checked="" type="checkbox"/>
gae	SNxSQLneoXpsB4OHxYDpkPYg9U7Hhw9LnUQG8aVrUA	<input type="checkbox"/>
두식	1949701518	<input type="checkbox"/>
빈병	HOrjQ1w3m_bs_3cX3hsaUaOShY-RCVW3cYvI_lAUYNc	<input type="checkbox"/>
승모	1957246886	<input type="checkbox"/>
오수빈	1948254533	<input type="checkbox"/>
최진호	Lpn8s-3cEN3JwizOVCqsi1YiZlc9HW_uTfTWU4xNmHI	<input type="checkbox"/>
NickName	ID	Lock & UnLock

검색 원하는 사용자만 검색하여 보여줍니다.

Search:

목록
리스트의 개수를 정하는 버튼입니다.

권한
사용자를 차단 / 해제하여 사용자의 권한을 부여합니다.

Showing 1 to 7 of 7 entries

Previous 1 Next

3. 관리자 페이지 - 마이페이지

The screenshot shows the 'My Page' section of the TRAVEL JEJU website. At the top, it displays the user's name, '최진호님의 마이페이지'. Below this, there are two main sections: '내가 쓴 글' (Articles I wrote) and '내가 만든 경로' (Travel routes I created). The '내가 쓴 글' section includes a list of articles with titles like '제목', '글쓴이', and '날짜'. The '내가 만든 경로' section shows a list of travel routes. At the bottom right, there is a 'Logout' button. A callout box points to the 'Logout' button with the text '클릭 시 로그아웃 된다.' (Click here to log out).

TRAVEL JEJU

최진호님의 마이페이지

내가 쓴 글

제목

내가 만든 경로

내가 만든 경로

여행 경로 페이지에서 저장한 나만의 여행 경로를 확인할 수 있습니다.

내가 쓴 글

제목

글쓴이

날짜

홈

메인 페이지로 이동됩니다

Home Logout

로그아웃

클릭 시 로그아웃 된다.

4. 게시판

관광지 명소
제주도의 유명한 명소 소개합니다

관광지 명소

맛집 정보

여행 코스

관광지 템
클릭 시 해당 게시판으로 이동된다

성산일출봉	사려니숲길	카멜리아힐	협재해수욕장
성산일출봉 천연보호구역은 제주특별자치도 서귀포시 성산읍...	사려니 숲길은 비자림로를 시작으로 물찻오름과 사려 니오름...	카멜리아힐은 동양에서 가장 큰 동백 수목원으로 토종 등...	제주도 서쪽에 가볼만한 해수욕장을 꼽으라면 단연 1...
천연보호구역, 관광명소	숲길, 사진명소	수목원, 연중무휴	서쪽해수욕장, 관광명소
좋아요 7	좋아요 9	좋아요 9	좋아요 4

4. 게시판

관광지 명소
제주도의 유명한 명소 소개합니다

카드(관광지/맛집)
DB에 저장된 내용을 불러와 카드로 구성한다.
(이미지 / 제목 / 설명 / 해시태그 / 좋아요)

관광지 명소 맛집 정보 여행 코스

성산일출봉

성산일출봉 천연보호구역은 제주특별자치도 서귀포시 성산읍...

천연보호구역, 관광명소

좋아요 7

관광지/맛집 이름

이름 클릭 시 상세 페이지로 이동
해당 카드의 자세한 내용을 확인 할 수 있다

사려니숲길

사려니 숲길은 비자림로를 시작으로 물찻오름과 사려
니오름...

숲길, 사려니숲길

좋아요

카멜리아힐

카멜리아힐은 동양에서 가장 큰 동백 수목원으로 토종
등...

수목원, 연중무휴

좋아요 9

협재해수욕장

제주도 서쪽에 가볼만한 해수욕장을 꼽으라면 단연
1...

서쪽해수욕장, 관광명소

좋아요 4

4. 게시판

관광지 명소
제주도의 유명한 명소 소개합니다

관광지 명소

관광지 명소
에메랄드 빛 바다,관광명소
좋아요 8

관광지 명소
오름,천연기념물
좋아요 16 *로그인 시 이용 가능

맛집 정보
카드 추가
직접 추천하고 싶은
내용을 추가 할 수 있다 *
로그인 시 이용 가능

여행 코스
검색 창
원하는 내용의 카드를 검색할 수 있다
곳,동쪽코스
좋아요 9

페이지
원하는 페이지로 이동 할 수 있다
카드는 한 페이지당 8개씩 노출된다

Type here... 글쓰기

<< 1 2 >>

Copyright © Jeju Website 2021

4. 게시판



관광지 명소



관광지 명소

제목 : 새별오름

관광지명 : 새별오름

글쓴이 : 오수빈

좋아요 : 5

지역명: 애월

설명

새별오름은 바리메오름·누운오름·당오름·금오름 등 많은 오름이 있는 밀집해 있는 서부 중산간 오름지대 중에서 으뜸가는 서부의 대표 오름이다. 저녁 하늘에 샛별과 같이 외롭게 서있다 하여 붙여진 이름이라 한다.

해시태그 : 측화산, 애월 관광명소

back



성산일출봉

성산일출봉 천연보호구역은 제주특별자치도 서귀포시 성산읍...

천연보호구역, 관광명소

좋아요 7



관광지 명소

제목 : 새별오름

관광지명 : 새별오름

글쓴이 : 오수빈

좋아요 : 5



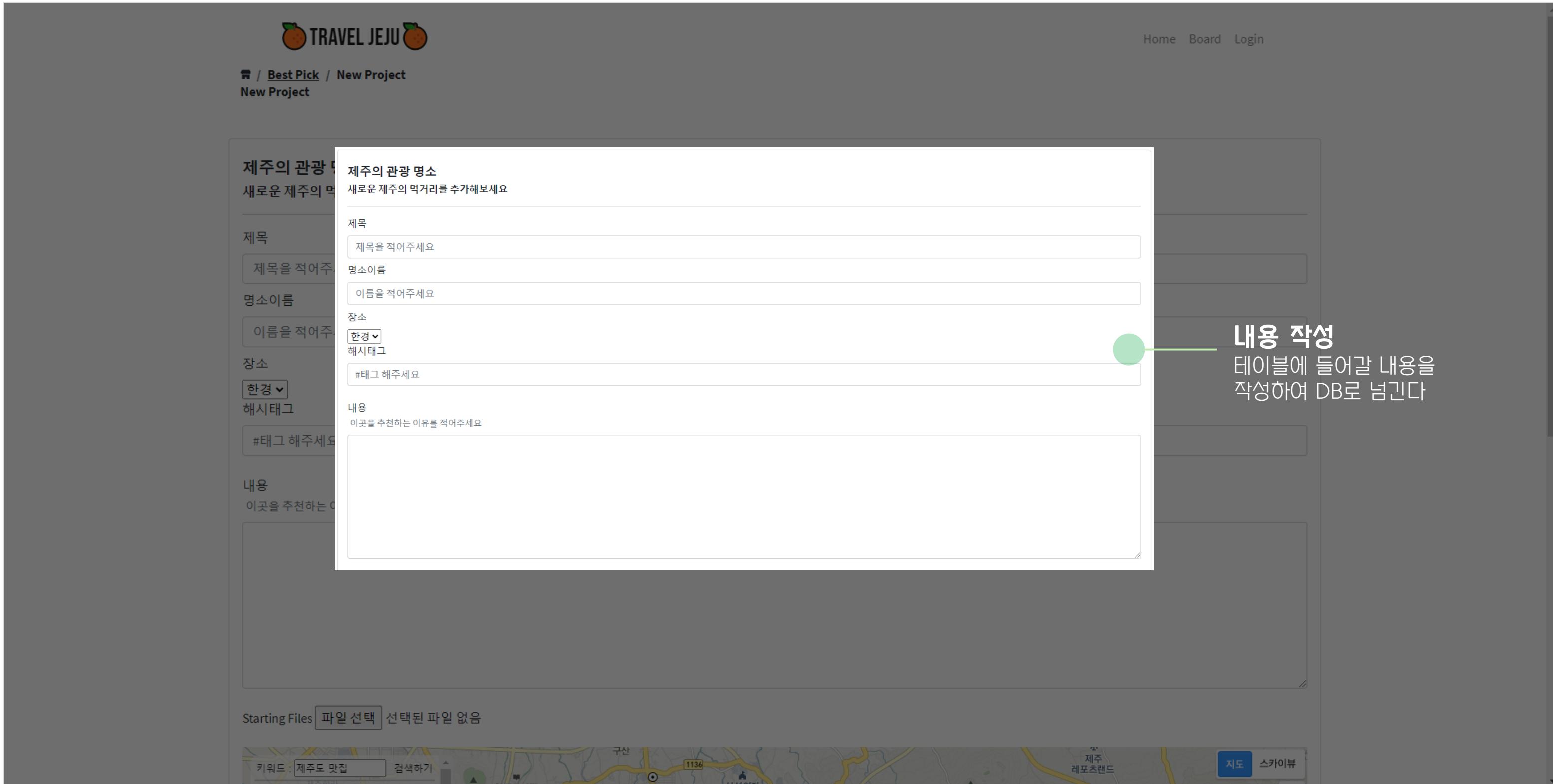
협재해수욕장

제주도 서쪽에 가볼만한 해수욕장을 꼽으라면 단연 1...

서쪽해수욕장, 관광명소

좋아요 4

4. 게시판



The screenshot shows a web application interface for creating a new travel project. At the top, there is a logo for "TRAVEL JEJU" with two orange icons. To the right of the logo are links for "Home", "Board", and "Login". Below the header, the URL "Best Pick / New Project" and the page title "New Project" are displayed.

The main content area contains a form for entering project details:

- 제목**: A text input field labeled "제목을 적어주세요".
- 명소이름**: A text input field labeled "이름을 적어주세요".
- 장소**: A dropdown menu set to "한경" and a text input field for hashtags labeled "#태그 해주세요".
- 내용**: A text area for writing reasons, with the placeholder "이곳을 추천하는 이유를 적어주세요".

At the bottom of the form, there is a file selection button labeled "파일 선택" and a message "선택된 파일 없음".

A green callout bubble on the right side points to the "내용" section with the text "내용 작성" and the sub-instruction "테이블에 들어갈 내용을 작성하여 DB로 넘긴다".

At the very bottom of the page, there is a map search bar with the keyword "제주도 맛집" and a "검색하기" button. To the right of the map are buttons for "지도" (Map) and "스카이뷰" (Aerial View).

4. 개시판

TRAVEL JEJU

Home Board Login

Best Pick / New Project
New Project

지도 스카이뷰

지도

2km kakao

위도 :
경도 :

Cancel Upload

Starting Files 파일 선택 선택된 파일 없음

지도 스카이뷰

주소
맛집/관광지의 주소를
지도에서 찾아 넣어준다

5. 여행 경로 페이지

TRAVEL JEJU MAKE YOUR TRAVEL COURSE

제주도 날씨
Jeju Weather

현재온도: 14°C
강수량: 20%

제주도 기상청 가기

선택 목록
Select List

선택목록 전체삭제

-  중문보말칼국수 좋아요: 72 X
-  낭뜰에 심팡 좋아요: 70 X
-  곰막식당 좋아요: 63 X
-  협재해수욕장 좋아요: 25 X
-  용눈이오름 좋아요: 25 X



제주도 날씨
Jeju Weather

여행경로 보기
여행경로 저장
관광명소 등록
맛집등록 하기
지도 원위치
이용방법

선택 목록
선택목록 전체삭제

제주도 기상청 가기

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

이용방법

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

여행경로 저장

관광명소 등록

맛집등록 하기

지도 원위치

선택 목록

제주도 날씨

여행경로 보기

<div style="position: absolute; top: 3025px; left: 3025px; background-color: black; color: white; padding: 5px

5. 여행 경로 페이지

TRAVEL JEJU MAKE YOUR TRAVEL COURSE

HOME About Best-Pick Login

검색 창
원하는 내용만 검색하여 표시
(맛집 테이블 / 관광지 테이블)

지도 **스카이뷰** **검색어를 입력해주세요** **🔍**

제주도 날씨
Jeju Weather

현재온도: 14°C
강수량: 20%

제주도 기상청 가기

선택 목록
Select List

선택목록 전체삭제

중문보말칼국수 좋아요 : 72	X
낭뜰에 쉼팡 좋아요 : 70	X
곰막식당 좋아요 : 63	X
협재해수욕장 좋아요 : 25	X
용눈이오름 좋아요 : 25	X

카테고리(맛집/관광지)
홈페이지 내에 저장된 정보를
각 카테고리 별로 좋아요 수가
많은 순으로 나열

카드 선택
카드의 우측 + 버튼을 누르면
좌측의 선택 목록으로 넘어가며
지도에 선택한 카드의 위치가
표시된다.

지도 **스카이뷰** **+** **-**

맛집 **관광지**

중문보말칼국수 좋아요 : 72
낭뜰에 쉼팡 좋아요 : 70
곰막식당 좋아요 : 63
춘미향식당 좋아요 : 50
옹포 별장가든 좋아요 : 39
흥성방 좋아요 : 37
고불락식당 좋아요 : 35
우진해장국 좋아요 : 27
그리운바다 성산포 좋아요 : 17
남원포구식당 좋아요 : 10

5. 여행 경로 페이지

TRAVEL JEJU MAKE YOUR TRAVEL COURSE

HOME About Best-Pick Login

제주도 날씨

변덕스러운 제주도 날씨에 여행 일정을 망치지 않도록 날씨를 확인하며 계획할 수 있다.

* 날씨는 3시간 간격으로 업데이트된다.

제주도 날씨
Jeju Weather

현재온도: 14°C
강수량: 20%

제주도 기상청 가기

선택 목록
Select List

선택목록 전체삭제

중문보말칼국수 좋아요: 72

낭뜰에 쉼팡 좋아요: 70

곰막식당 좋아요: 63

협재해수욕장 좋아요: 25

용눈이오름 좋아요: 25

제주도 날씨 이용방법 지도 원위치 맛집등록 하기 관광명소 등록 여행경로 보기 여행경로 저장

지도 스카이뷰 검색어를 입력해주세요

맛집 관광지

제주도 날씨 선택 목록 선택된 카드

선택 목록 우측에서 선택한 카드를 보여준다. 전체 삭제 버튼으로 편리하게 모든 일정을 비울 수 있다.

선택된 카드

- 카드의 우측 X 버튼을 누르면, 선택 취소가 되며 카드가 우측으로 다시 넘어가게 된다.
- 카드들의 순서는 Drag & Drop으로 변경할 수 있다.

5. 여행 경로 페이지

TRAVEL JEJU MAKE YOUR TRAVEL COURSE

이용방법

페이지 이용방법을
보여주는 창을 띄움

Jeju Weather

현재온도: 14°C
강수량: 20%

여행경로보기

선택된 카드들의 경로를
선택한 순서대로 지도에
표시해준다.

선택 목록

Select List

여행경로저장

선택된 모든 카드들이
경로들 모달 창에 띄워
보여준다

등록하기

우측에 원하는 카드가 없을 경우에
등록 버튼을 눌러 원하는 카테고리 별로
카드 등록을 할 수 있다.
* 비회원은 등록할 수 없다.

지도 **스카이뷰**

검색어를 입력해주세요

맛집 **관광지**

- 중문보말칼국수 좋아요: 72
- 낭뜰에 쉼팡 좋아요: 70
- 곰막식당 좋아요: 63
- 춘미향식당 좋아요: 50
- 옹포 별장가든 좋아요: 39
- 홍성방 좋아요: 37
- 고불락식당 좋아요: 35
- 우진해장국 좋아요: 27
- 그리운바다 성산포 좋아요: 17
- 남원포구식당 좋아요: 10

5. 여행 경로 페이지

TRAVEL JEJU MAKE YOUR TRAVEL COURSE

HOME About Best-Pick Login

지도 스카이뷰

검색어를 입력해주세요

맛집 관광지

제주도 날씨
Jeju Weather

현재온도: 14°C
강수량: 20%

제주도 기상청 가기

선택 목록
Select List

선택목록 전체삭제

중문보말칼국수 좋아요: 72

낭뜰에 쉼팡 좋아요: 70

곰막식당 좋아요: 63

춘미향식당 좋아요: 50

홍성방 좋아요: 39

옹포 별장가든 좋아요: 37

홍성방 좋아요: 37

고불락식당 좋아요: 35

우진해장국 좋아요: 27

그리운바다 성산포 좋아요: 17

남원포구식당 좋아요: 10

내가 선택한 여행 경로

저장하기

저장하기 버튼 클릭 후 로그인이 된 경우 저장된 여행 경로는 마이 페이지에서 확인할 수 있다.
비로그인의 경우 로그인 화면으로 이동한다.

03

기능 소개

Function

- | | | |
|--------------|-------|----|
| 1. 관리자 페이지 | | 36 |
| 2. 마이 페이지 | | 41 |
| 3. 로그인 페이지 | | 43 |
| 4. 게시판 페이지 | | 49 |
| 5. 여행 경로 페이지 | | 54 |
| 6. 폴리곤 | | 62 |



1. 관리자 페이지

1 - 1. 관리자 로그인



Spring security를 이용해 강력한 보안 구현
DB에 저장된 관리자 정보와 일치할 경우에만 해당 url
주소에 들어갈 수 있다.

2가지 권한(ROLE_MASTER/ROLE_ASSISTANCE)을
부여하여 관리자 페이지에 접근권한을 제한하였다.
ROLE_MASTER 권한은 관리자 페이지의 모든 부분에 접근
가능하다.

ROLE_ASSISTANCE 권한은 관리자 페이지의 사용자 관리
탭은 이용하지 못한다.

또한 비밀번호는 암호화되어 저장된 DB 값과 비교한다.

URL에 접근 권한 부여

```

9<security:http pattern="/admin/**" auto-config="true" use-expressions="true">
10<security:csrf disabled="true"/>
11<security:intercept-url pattern="/admin/main" access="hasAnyRole('ROLE_MASTER', 'ROLE_ASSISTANTE')"/>
12<security:intercept-url pattern="/admin/site" access="hasAnyRole('ROLE_MASTER', 'ROLE_ASSISTANTE')"/>
13<security:intercept-url pattern="/admin/user" access="hasAnyRole('ROLE_MASTER')"/>
14
15<security:form-login
16    username-parameter="id"
17    password-parameter="password"
18    login-page="/admin/Login"
19    default-target-url="/admin/main"
20    login-processing-url="/admin/perform_Login"
21    authentication-failure-url="/admin/Login?error=true"
22    always-use-default-target="true"
23/>
24<security:logout logout-url="/adminLogout" logout-success-url="/adminLogin"/>
25</security:http>
26
27
28<security:authentication-manager>
29    <security:authentication-provider user-service-ref="adminLogin">
30        <security:password-encoder ref="passwordEncoder"/>
31    </security:authentication-provider>
32</security:authentication-manager>
33
34<bean id="adminLogin" class="com.kgitbank.jeju.admin.service.AdminLoginService"/>
35<bean id="passwordEncoder" class="org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder"/>
36</beans>
37

```

Security 로그인 coustom

비밀번호 암호화

▲ Spring Security 설정

1. 관리자 페이지

1 - 2. 데시보드

Chart.js를 이용해 ajax로 방문자수 값을 가져와 방문자수
그래프 부분을 구현했다.
DB 쿼리문에서 최대 10일 치의 데이터만 가져오도록
제한해뒀다.

```
<select id="getDailyVisit" resultType="com.kgitbank.jeju.admin.dto.DailyVisit">
    SELECT * FROM (
        SELECT TO_CHAR(visit_date, 'YYYY/MM/DD') AS visit_date, COUNT(*) AS count
        FROM visit
        GROUP BY TO_CHAR(visit_date, 'YYYY/MM/DD')
    )
    WHERE <![CDATA[ ROWNUM <= 10]]>
</select>
```

▲ 일별 방문자수 가져오는 쿼리문

1. 관리자 페이지

1-2-1. 방문자수 체크

Session 값을 이용해 session이 새로 생성되면 방문자수를 증가시킨다. 단, 관리자 페이지 접근일 경우에는 방문자 수를 증가시키지 않는다.(하단의 코드)

SessionService 빈을 일반클래스에서는 Autowired로 가져올 수 없기 때문에, getSessionService 메서드를 보면 DispatcherServlet에 의해서 만들어지는 WebApplicationContext에 직접 접근해 꺼내왔다. (우측 코드)

```

3+ import javax.servlet.http.HttpSession;
0
1 @Service("sessionService")
2 public class SessionServiceImpl implements SessionService{
3@   @Autowired
4     VisitMapper visitMapper;
5
6@   @Override
7     public void addVisit(HttpSession session) {
8       if(session.getAttribute("admin") != null) {
9         return;
0       }
1       visitMapper.addVisit();
2     }
3 }
```

▲ DB visit 테이블 접근

```

3+ import javax.servlet.http.HttpSession;
13
14 @Log4j
15 public class VisitorCounter implements HttpSessionListener{
16
17@   @Override
18     public void sessionCreated(HttpSessionEvent se) {
19       if(se.getSession().isNew()) {
20         log.info("create new session");
21         getSessionService(se).addVisit(se.getSession());
22     } else {
23       log.info("already created session");
24     }
25
26   }
27
28@   @Override
29     public void sessionDestroyed(HttpSessionEvent se) {}
30
31@   private SessionService getSessionService(HttpSessionEvent se) {
32     WebApplicationContext context =
33       WebApplicationContextUtils.getWebApplicationContext(
34         se.getSession().getServletContext(),
35         FrameworkServlet.SERVLET_CONTEXT_PREFIX + "appServlet" );
36
37     return (SessionService) context.getBean("sessionService");
38 }
```

▲ Session 값을 이용한 방문자수 체크

1. 관리자 페이지

1 - 3. 회원 관리

DataTables 라이브러리를 이용해 ajax로 값을 가져와 회원 정보 테이블을 만들었다.

그리고 banned 버튼을 누르면 ajax로 데이터를 보내 DB 값이 update 되는 방식이다.

```

$(document).ready(function() {
    $('#dataTable').DataTable().destroy();

    $('#dataTable').DataTable( {
        "ajax": {url:"/jeju/admin/getUserData", dataSrc:""},
        "columns": [
            { "data": "nickname" },
            { "data": "user_id" },
            { "data": "banned" }
        ],
        "columnDefs": [
            {
                "targets": 2,
                "render": function ( data, row ) {
                    if ( data === 1 ) {
                        return `<label class="switch" style="margin-top: 5px;"><input type="checkbox" checked=""></input></label>`;
                    } else {
                        return `<label class="switch" style="margin-top: 5px;"><input type="checkbox"></input></label>`;
                    }
                }
            }
        ],
        "fnDrawCallback": function() {
            $(".checkbox").click(function (e) {
                let checkbox = e.target;
                let id = e.target.parentNode.parentNode.previousSibling.innerText;
                banned = checkbox.checked? 1: 0;
                console.log(checkbox.checked);
                $.ajax({
                    url: '/jeju/admin/user_ban',
                    dataType: 'application/json',
                    contentType: 'application/json',
                    type: 'post',
                    data: JSON.stringify ({
                        "user_id": id,
                        "banned": banned
                    }),
                    success: function() {
                        console.log('Success');
                    }
                })
            });
        }
    });
});
```

데이터 테이블 생성, ajax 코드 ►

1. 관리자 페이지

1 - 4. 사이트 관리

현재 DB에 저장된 Main 페이지의 Visual 부분 이미지와, title, sub-title을 불러와 input 태그에 값을 넣는다. 파일을 선택해 이미지를 불러오면, 미리보기 화면에 이미지가 보이게 js로 EventListener를 추가해 두었다.

- * 저장 버튼을 누르면 해당 값이 controller로 넘어가고, 이미지는 server에 저장
- * 이미지 경로와, title, sub-title 값은 DB에 저장한다.

만약 선택된 이미지가 없을 경우 DB에 저장되어있던 이미지 경로 값을 다시 저장한다.

```

public void uploadMainInfo(MultipartFile multi, Main main, String prevImage, HttpServletRequest request) {
    String fileName = multi.getOriginalFilename();

    try {
        if (!multi.getOriginalFilename().isEmpty()) {
            String root_path = request.getSession().getServletContext().getRealPath("/resources/img");
            root_path += File.separator + fileName;
            File file = new File(root_path);
            multi.transferTo(file);
            Log.info(file.getAbsolutePath());
        } else {
            String[] strArray = prevImage.split("/");
            int endIndex = strArray.length-1;
            fileName=strArray[endIndex];
        }
        Main updateMain = new Main(main.getInfo_id(),
                "./resources/img/" + fileName,
                main.getTitle(),
                main.getSub_title());
        mainMapper.updateMainInfo(updateMain);
    } catch (Exception e) {
        Log.info("error : " + e);
    }
}

```

▲ Form 데이터 처리

Form data DB 저장

2. 마이페이지

2-1. 내가 쓴 글

Ajax로 값을 받아와서 테이블에 띄워주는 탭이다.

Ajax로 관광지 게시판 값과, 맛집 관광지 값을 가져온 후 테이블 데이터를 생성해야 해서 promise.all을 이용해 비동기적 처리를 했다.

그리고 twbsPagination 플러그인으로 pagination을 구현했다.

```

myWrite.addEventListener('click', () => {
    promises = [makeRequest('GET', 'getMyFamousRestaurantWrite'), makeRequest('GET', 'getMyTouristSpotWrite')];

    /* process after sucess request-response */
    Promise.all(promises).then(responses => {
        let JSONString = "";
        responses.forEach((response)=> {
            JSONString += response.trim();
        });
        JSONString = JSONString.replace('\]\[', ', ');

        myWriteVariable.records = JSON.parse(JSONString);
        myWriteVariable.totalRecords = myWriteVariable.records.length;
        myWriteVariable.totalPages = Math.ceil(myWriteVariable.totalRecords/myWriteVariable.recordPerPage);
        applyMyWritePagination();

    }).catch((error) => {
        console.error(error);
    });
});

```

▲ Ajax 비동기 처리

2. 마이페이지

2-2. 내가 만든 경로

내가 쓴 글 템 처리와 비슷하다.

Ajax로 값을 가져온 후 table에 해당 값을 보여준다.

단 경로는 내가 만든 경로 배열과, 경로 안에 목적지들의 배열이 하나 더 있어서 받아온 json 데이터를 처리해줘야 했다.

key값은 경로 id 값으로, Value 값은 배열로 Object를 하나 만든다.

JSON 데이터가 id 값으로 정렬되어 들어오기 때문에, 이전 id값과 id가 같은

JSON 데이터는 Object[id] 배열에 추가하고, 이전 id 값과 id가 다르면 해당 id를 key로 만들고 value 배열에 데이터를 추가한다.

```

myCourse.addEventListener('click', () => {
  makeRequest('GET', 'getMyRoute').then(responses => {
    myCourseVariable.records = JSON.parse(responses);

    var coursesObj = {};
    let prevId = myCourseVariable.records[0].id;
    coursesObj[prevId] = [myCourseVariable.records[0]];

    for(let i = 1; i < myCourseVariable.records.length ; i++) {
      if(prevId !== myCourseVariable.records[i].id) {
        prevId = myCourseVariable.records[i].id
        coursesObj[prevId] = [myCourseVariable.records[i]]
      } else {
        coursesObj[prevId].push(myCourseVariable.records[i]);
      }
    }

    courses = Object.values(coursesObj);
    myCourseVariable.totalRecords = Object.keys(coursesObj).length;
    myCourseVariable.totalPages = Math.ceil(myCourseVariable.totalRecords/myCourseVariable.recordPerPage);

    applyMyCoursePagination();
  }).catch((error) => {
    console.error(error);
  });
});

```

▲ JSON 데이터 처리

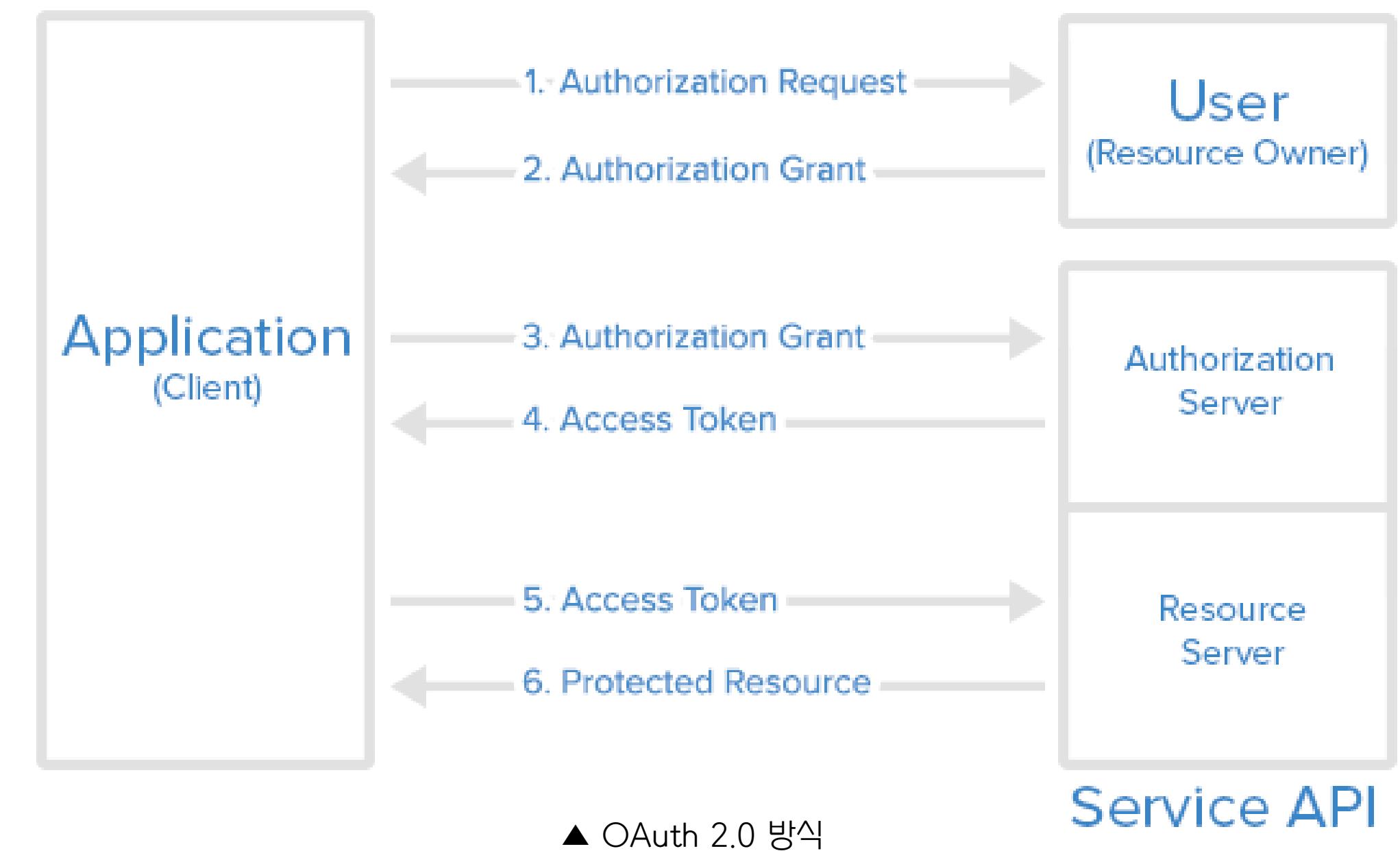
3. 로그인 페이지

3-1. 소셜 로그인

Kakao와 Naver 모두 OAuth2.0 방식으로 로그인 API를 제공한다.

그래서 OAuth 자바 라이브러리인 **ScribeJava**를 이용해 소셜 로그인을 구현했다.

Abstract Protocol Flow



3. 로그인 페이지

3-1. 소셜 로그인

소셜 로그인에 사용될 client id와 secret 값은 노출되면 안된다. 그래서 properties 파일로 따로 관리하고, gitignore을 지정해 github에는 올라가지 않도록 했다.

* DB 설정 값도 마찬가지다.

root-context.xml에 <util:properties.../>로 모든 properties 파일을 프로퍼티로 등록하고 사용한다.

```
<util:properties id="property" location="classpath:/config/*.properties" />
<bean id="config" class="com.zaxxer.hikari.HikariConfig" >
    <property name="driverClassName" value="#{property['driverClassName']}"/>
    <property name="jdbcUrl" value="#{property['url']}"/>
    <property name="username" value="#{property['username']}"/>
    <property name="password" value="#{property['password']}"/>
    <property name="minimumIdle" value="25"/>
    <property name="maximumPoolSize" value="50"/>
</bean>
```

▲ root-context.xml에 프로퍼티 등록과 DB config 빈 생성 시 프로퍼티를 사용한 코드

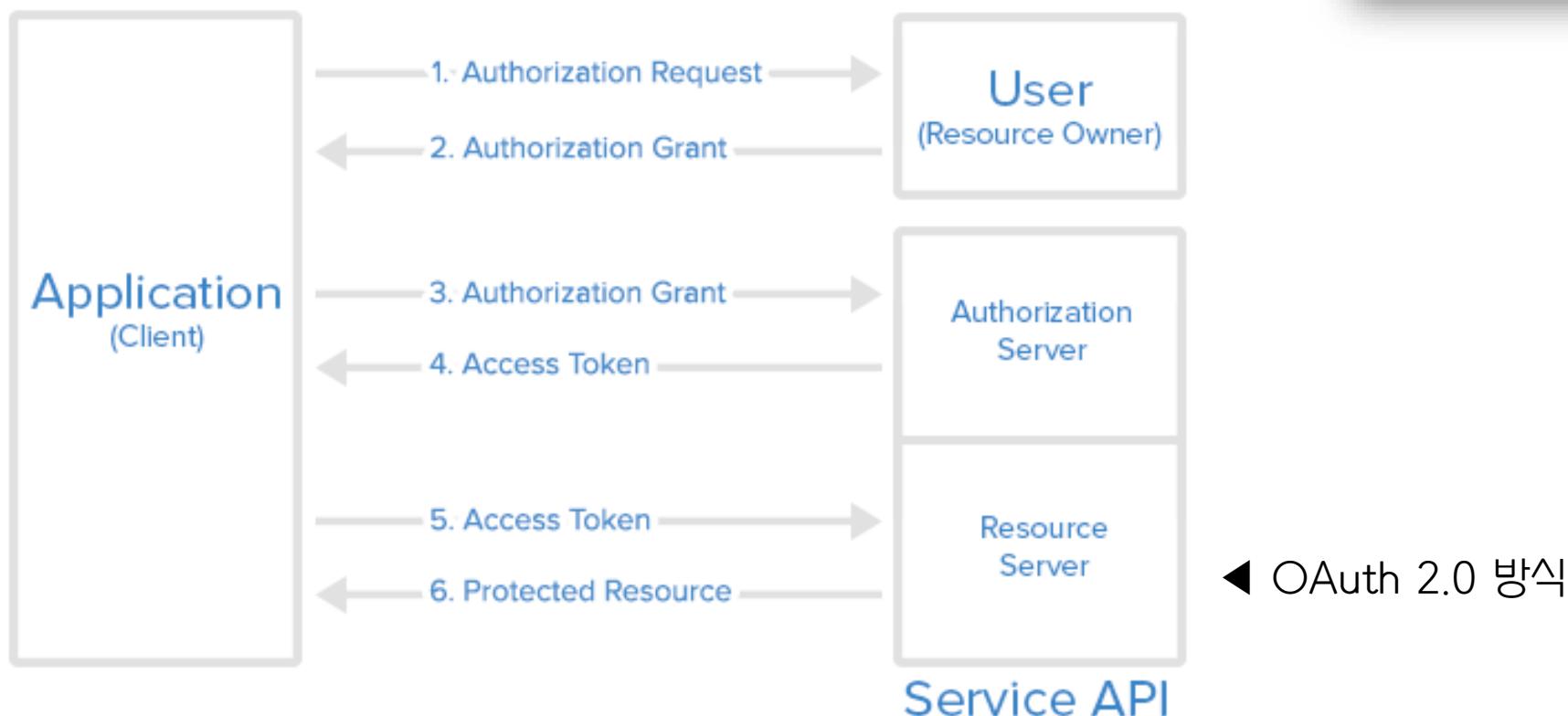
3. 로그인 폼이지

3-1. 소셜 로그인

먼저 ScribeJava에서 제공하는 DefaultApi20을 상속받아 OAuth에 필요한 1번의 Authorization code를 받아올 url를 getAuthorizationURL()메서드에 세팅해둔다.

그리고 3번을 수행해 Access Token을 얻어오기 위한 URL을 getAccessTokenEndpoint() 메서드에 세팅해둔다.

Abstract Protocol Flow



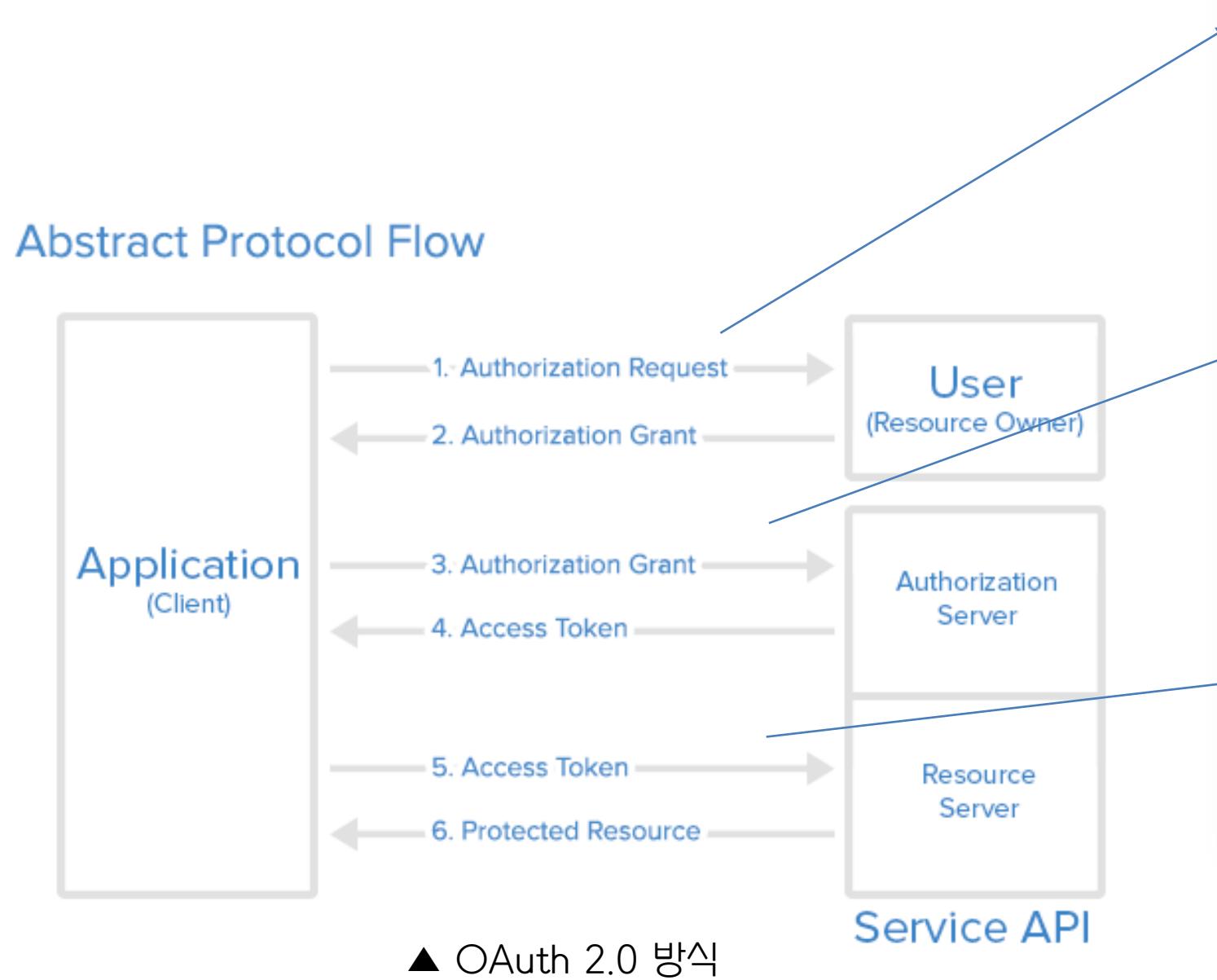
```
5 public class NaverLoginApi extends DefaultApi20{
6     private static class InstanceHolder{
7         private static final NaverLoginApi INSTANCE = new NaverLoginApi();
8     }
9
10    public static NaverLoginApi instance(){
11        return InstanceHolder.INSTANCE;
12    }
13    @Override
14    public String getAccessTokenEndpoint() {
15        return "https://nid.naver.com/oauth2.0/token?grant_type=authorization_code";
16    }
17
18    @Override
19    protected String getAuthorizationBaseUrl() {
20        return "https://nid.naver.com/oauth2.0/authorize";
21    }
22
23 }
```

▲ 네이버 API 코드

◀ OAuth 2.0 방식

3. 로그인 페이지

3-1-1. 네이버 로그인



```

public String getAuthorizationUrl(HttpServletRequest session, String serverUrl) {
    String state = generateRandomString();

    setSession(session, state);
    OAuth20Service oauthService = new ServiceBuilder(NAVER_CLIENT_ID)
        .apiSecret(NAVER_CLIENT_SECRET)
        .callback(serverUrl + NAVER_REDIRECT_URI)
        .build(NaverLoginApi.instance());
    return oauthService.getAuthorizationUrl(state);
}

/* 네이버아이디로 Callback 처리 및 AccessToken 획득 Method */
public OAuth2AccessToken getAccessToken(HttpServletRequest session, String code, String state, String serverUrl) throws Exception {

    String sessionState = getSession(session);

    if (StringUtils.pathEquals(sessionState, state)) {
        OAuth20Service oauthService = new ServiceBuilder(NAVER_CLIENT_ID)
            .apiSecret(NAVER_CLIENT_SECRET)
            .callback(serverUrl + NAVER_REDIRECT_URI)
            .build(NaverLoginApi.instance());
        OAuth2AccessToken accessToken = oauthService.getAccessToken(code);
        return accessToken;
    }
    return null;
}

public String getUserProfile(OAuth2AccessToken oauthToken, String serverUrl) throws Exception {
    OAuth20Service oauthService = new ServiceBuilder(NAVER_CLIENT_ID)
        .apiSecret(NAVER_CLIENT_SECRET)
        .callback(serverUrl + NAVER_REDIRECT_URI)
        .build(NaverLoginApi.instance());
    OAuthRequest request = new OAuthRequest(Verb.GET, PROFILE_API_URL);
    oauthService.signRequest(oauthToken, request);
    Response response = oauthService.execute(request);
    return response.getBody();
}

```

▲ 실제 NaverLogin 인증처리 하는 코드

3. 로그인 페이지

3-1-1. 네이버 로그인

```

@Override
public void naverLogin(HttpSession session, OAuth2AccessToken oauthToken, String serverUrl) throws Exception {
    // 로그인 사용자 정보를 읽어온다
    String apiResult = naverLoginBO.getUserProfile(oauthToken, serverUrl);
    Log.info("apiResult : " + apiResult);

    JsonElement element = JsonParser.parseString(apiResult);
    JsonObject object = element.getAsJsonObject();
    JsonObject response_obj = (JsonObject) object.get("response");

    String id = response_obj.get("id").getAsString();
    String nickname = response_obj.get("nickname").getAsString();

    Log.info("naver ID : " + id);
    Log.info("naver nickname : " + nickname);

    User user = new User(id, nickname, 0);

    if(usermapper.listUser(id) == null) {
        usermapper.addUser(user);
    }

    // 세션에 사용자 정보 등록
    session.setAttribute("id", id);
}

```

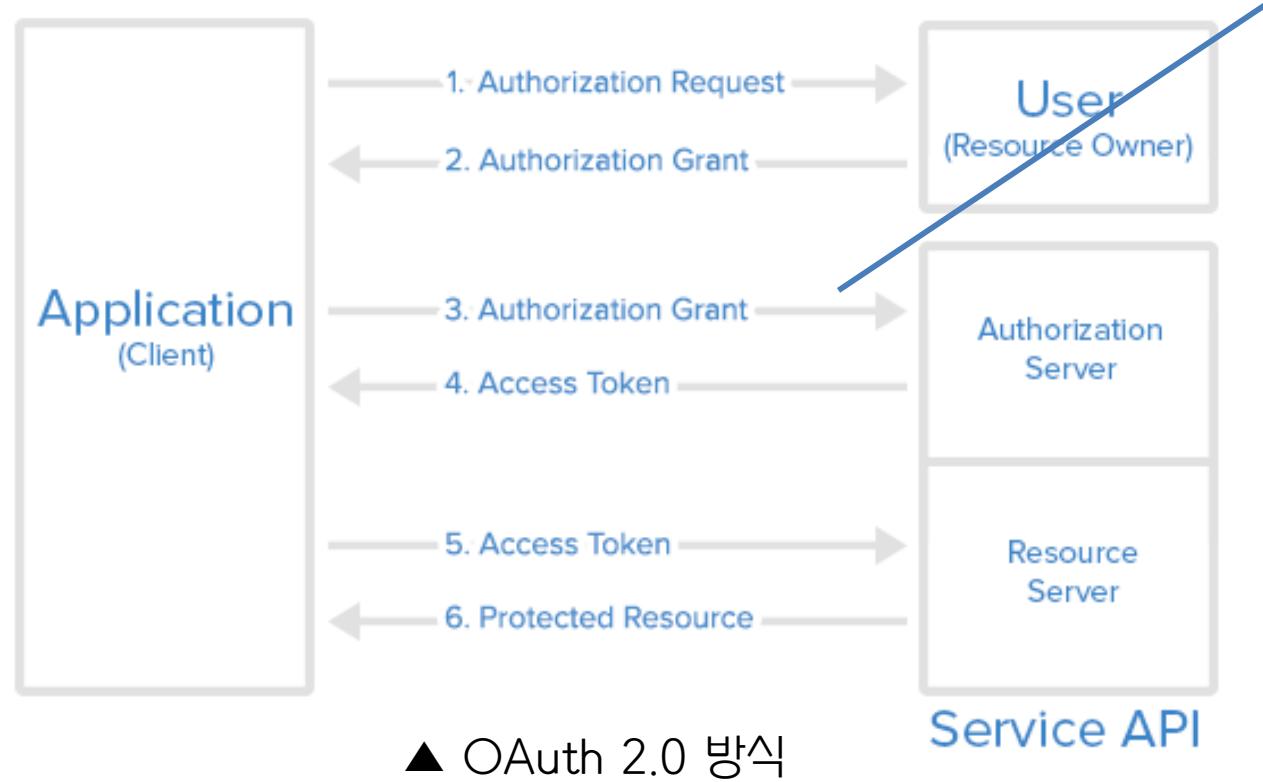
Response 값에서 필요한 값을 추출한다.
 id 값은 session Attribute에 저장한다.
 id, nickname으로 해당 값이 DB에 없으면 저장해둔다.

3. 로그인 페이지

3-1-2. 카카오 로그인

카카오 로그인 API는 ScribeJava를 이용하면 인증 과정에 문제가 없어도 AccessToken error가 일어나 라이브러리를 사용하지 않고 직접 AccessToken 요청을 보내 값을 받아온다. 그 이외 부분은 네이버 로그인과 같다.

Abstract Protocol Flow



```

public String getAccessToken(HttpServletRequest request, String authorizeCode) throws UnsupportedEncodingException {
    String clientId = KAKAO_CLIENT_ID;
    String clientSecret = KAKAO_CLIENT_SECRET;
    String code = authorizeCode;
    String state = request.getParameter("state");
    String redirectURI = URLEncoder.encode("http://localhost:8081/jeju/kakaoAuth", "UTF-8");
    String apiURL;
    apiURL = "https://kauth.kakao.com/oauth/token?grant_type=authorization_code&";
    apiURL += "client_id=" + clientId;
    apiURL += "&client_secret=" + clientSecret;
    apiURL += "&redirect_uri=" + redirectURI;
    apiURL += "&code=" + code;
    apiURL += "&state=" + state;

    String access_token = "";
    String refresh_token = "";

    try {
        URL url = new URL(apiURL);
        HttpURLConnection con = (HttpURLConnection)url.openConnection();
        con.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        con.setRequestMethod("POST");

        int responseCode = con.getResponseCode();

        BufferedReader br;
        if(responseCode==200) { // 정상 호출
            br = new BufferedReader(new InputStreamReader(con.getInputStream()));
        } else { // 에러 발생
            ...
        }

        String inputLine;
        StringBuffer res = new StringBuffer();

        while ((inputLine = br.readLine()) != null) {
            res.append(inputLine);
        }

        if(responseCode==200) {
            JsonElement element = JsonParser.parseString(res.toString());
            access_token = element.getAsJsonObject().get("access_token").getAsString();
            refresh_token = element.getAsJsonObject().get("refresh_token").getAsString();
        } else {
            access_token = null;
        }
        br.close();
    } catch (Exception e) {
        Log.error(e);
    }

    return access_token;
}

```

Kakao 로그인 API accessToken 가져오기

4. 게시판 페이지

1 - 1. 관광/맛집 게시판

`touristBoard.js / restaurantBoard.js` 를 이용해 ajax로 데이터를 가져와 카드를 구현했다.
`twbsPagination` 플러그인을 사용하여 페이지네이션을 하였다.

게시판 보드에서 name을 클릭하면 RequestParam 으로 `tourist_spot_id` 값을 넘겨받아 데이터의 자세한 내용을 DB에서 받아 띄워준다.

```
// detail and click count
@RequestMapping(value="board/listById", method= RequestMethod.GET)
public ModelAndView listById (@RequestParam("tourist_spot_id") int tourist_spot_id,
                             HttpSession session) throws Exception{

    ModelAndView mav = new ModelAndView();
    mav.setViewName("board/touristView");
    mav.addObject("tourist_spot_id", tourist_spot_id);
    mav.addObject("tourist_spot", touristSpotMapper.listView(tourist_spot_id));

    log.info(mav);
    return mav;
}
```

```
//page number
const applyPagination = ()=> {
    $pagination.twbsPagination({
        totalPages: totalPages,
        visiblePages: 5,
        startPage: 1,
        prev: "<<",
        next: ">>",
        first: '',
        last: '',
        onPageClick: function(e, page) {
            let displayCardsIndex = Math.max(page -1, 0) * cardPerPage;
            let endCard = (displayCardsIndex) + cardPerPage;

            displayCards = cards.slice(displayCardsIndex, endCard);
            generateCard();
        }
    });
}
```

▲ 카드 페이지 관련 코드

◀ 자세히 보기 관련 코드

4. 게시판 페이지

1 - 2. 게시물 좋아요

Ajax를 이용하여 좋아요를 클릭 한 카드정보의 PK값을 RestController에 전송한다
컨트롤러에서는 해당 키 값을 이용하여 DB상의 해당 카드의 좋아요 수를 업데이트 한 후 업데이트 된 좋아요 값을 조회하여 다시 JSP로 리턴한다

데이터를 전달받은 JSP에서는 조건문을 통해 변경할 카드를 찾은 후 해당 카드의 좋아요 수를 업데이트 하여 웹페이지에 띄워준다

```
//like addPositive
function like_func(spotId){
    const likeNum = document.querySelectorAll('#like_result');

    const xhttp = new XMLHttpRequest();
    xhttp.addEventListener('readystatechange', (e) => {

        const restaurant_id = document.querySelectorAll('.restaurant_id');

        const readyState = e.target.readyState;
        const httpStatus = e.target.status;
        if(readyState == 4 && httpStatus == 200) {

            for(var i = 0; i < restaurant_id.length; ++i){
                if(spotId == $(restaurant_id[i]).val()){
                    likeNum[i].innerHTML = '<i class="far fa-thumbs-up me-3">/i>' + Object.values(JSON.parse(xhttp.responseText))[9];
                }
            }
        });
        xhttp.open('GET', '/jeju/board/restlike/' + spotId, true);
        xhttp.setRequestHeader('content-type', 'application/json; charset=UTF-8');

        xhttp.send();
    });
}
```

▲ 좋아요 관련 Ajax 코드

```
<update id="addDTORestPositiveNum" >
UPDATE famous_restaurant SET positive_num  = positive_num +1
WHERE famous_restaurant_id = #{id}
</update>

<select id="getListByRestPositiveNum" resultType="com.kgitbank.jeju.dto.FamousRestaurant">
    SELECT * FROM famous_restaurant WHERE famous_restaurant_id = #{id}
</select>
```

◀ 좋아요 관련 SQL문

4. 게시판 페이지

1 - 3. 게시물 검색

검색은 두 가지의 유형이 있는데,

첫번째는 메인 페이지에 위치한 Polygon을 통해 행정구역별 맛집
검색으로 진입

두번째는 게시판에서의 직접적인 검색기능이 있다

JSP에서 두 가지 유형 중 어떻게 진입하였는지 판별하고,
각각의 유형에 맞는 데이터를 컨트롤러로 전송한다

컨트롤러에서는 받은 데이터를 이용하여 사용자가 원하는 정보를
DB에서 가져와 웹페이지상에 띄워준다

```
<select id="getRestSearch" resultType="com.kgitbank.jeju.dto.FamousRestaurant">
    SELECT * FROM famous_restaurant WHERE title Like '%'||#{textValue}||'%'
</select>

<select id="polygonSearch" resultType="com.kgitbank.jeju.dto.FamousRestaurant">
    SELECT * FROM famous_restaurant WHERE address = #{textValue}
</select>
```

▲ 검색 관련 SQL문

```
<c:choose>
    <c:when test="#{polygonSearch ne null}">
        <input id="searchValue" type="text"
            class="form-control border-0 ps-3" placeholder="Type here..." value="#{polygonSearch }">
    </c:when>
    <c:otherwise>
        <input id="searchValue" type="text"
            class="form-control border-0 ps-3" placeholder="Type here...">
    </c:otherwise>
</c:choose>
```

▲ 검색 관련 html 코드

```
searchBtn.addEventListener('click', () =>{
    $(document).ready(function() {
        $("#body").empty();
        var minValue = minValue.value;
        searchRequest(minValue);

    });
});

function searchRequest(minValue) {
    xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = setVariable;

    xhttp.open('GET', '/jeju/restSearch/' + minValue, true);
    xhttp.send();
}
```

▲ 검색 관련 ajax 코드

4. 게시판 페이지

2-1. 여행 경로 지도 API

카카오 맵 API를 이용하여 게시판 글 작성 시
사용자가 원하는 곳을 키워드로 검색하여 맵 상의 해당 위치에
마커를 찍어주고 마커를 클릭 시 해당 위치의
위도,경도 값을 글 작성의 위도,경도 input값에 넣어준다

사용자들이 작성한 맛집&관광지 글들이 추후 경로 찾기 페이지에서
좌표 값으로 마커를 찍어줘야 하기 때문에 기능을 추가하였다

게시판 맵에서는 키워드 검색, 마커 기능, 마커 클릭 시 맵 이동
& 좌표 값 추출, 스카이 뷰 기능들이 있다

```
var mapContainer = document.getElementById('map'), // 지도를 표시할 div
    mapOption = {
        center: new kakao.maps.LatLng(33.36209328052907,126.53147109994914), // 지도의 중심좌표
        level: 9 // 지도의 확대 레벨
    };

    // 지도를 생성합니다
var map = new kakao.maps.Map(mapContainer, mapOption);

    // 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});

    // 키워드로 장소를 검색합니다
searchPlaces();

    // 일반 지도와 스카이뷰로 지도 타입을 전환할 수 있는 지도타입 컨트롤을 생성합니다
var mapTypeControl = new kakao.maps.MapTypeControl();

    // 지도에 컨트롤을 추가해야 지도위에 표시됩니다
    // kakao.maps.ControlPosition은 컨트롤이 표시될 위치를 정의하는데 TOPRIGHT는 오른쪽 위를 의미합니다
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);
```

```
kakao.maps.event.addListener(marker, 'click', function() {

    var moveLatLon = new kakao.maps.LatLng(Object.values(position)[1],Object.values(position)[0]);
    map.setLevel(5);
    // 지도 중심을 부드럽게 이동시킵니다
    // 만약 이동할 거리가 지도 화면보다 크면 부드러운 효과 없이 이동합니다
    map.panTo(moveLatLon);

    // 지도를 1레벨 올립니다 (지도가 축소됩니다)

    latitude.innerHTML = Object.values(position)[1];
    longitude.innerHTML = Object.values(position)[0];
});
```

▲ 카카오 API를 이용한 지도 생성

◀ 지도 클릭 시 좌표 값 추출

4. 게시판 페이지

3. 게시물 작성

게시물 작성 후 저장버튼 클릭 시 필수 입력데이터에 빈 공간이 있는지 체크하여 주며

데이터를 전송 할 때에 위도,경도 값이 form바깥에 위치하여 값을 가져오기 위해 form안에 hidden 태입의 input태그를 만들어 두고 JS단에서 form태그 바깥에 위치한 위도,경도 값을 value로 넣어주어 값을 전송하여 준다.

컨트롤러에서는 넘겨받은 데이터들을 sql문을 이용하여 DB에 저장한다.

```
function boardPath(location,latitude, longitude){
  const sortByForm = document.getElementById('sortBy');
  const sortByLocation = document.querySelector('#sortBy > #location_path');
  const sortByLatitude = document.querySelector('#sortBy > #getlatitude');
  const sortByLongitude = document.querySelector('#sortBy > #getlongitude');

  sortByForm.action= './addTourist/success';
  sortByForm.method = 'post';
  sortByForm.enctype = 'multipart/form-data';
  sortByLocation.value = location;
  sortByLatitude.value = latitude;
  sortByLongitude.value = longitude;
  sortByForm.submit();
}
```

```
<form id="sortBy">
  <label for="projectName" class="form-label">제목</label> <input
    type="text" class="form-control" name="title" id="title"
    placeholder="제목을 적어주세요"> <label for="projectName"
    class="form-label mt-2">명소이름</label> <input type="text"
    class="form-control" name="name" id="name"
    placeholder="이름을 적어주세요">
  <label for="projectName" class="form-label mt-2">장소</label>
  <div class="col-6">
    <Select id="location_id" name="location_id ">
      <c:forEach items="${locations }" var="location" varStatus="i">
        <option id="location" name="location"
          value="${location.location_id }">${location.location }
        </option>
        <br>
      </c:forEach>
    </Select>
  </div>
  <label for="projectName" class="form-label">해시태그</label>
  <input type="text" class="form-control" name="hashtag" id="hashtag"
    placeholder="#태그 해주세요">

  <label class="mt-4">내용</label>
  <p class="form-text text-muted text-xs ms-1 mb-2">이곳을 추천하는 이유를
    적어주세요</p>
  <div id="editor" class="ql-container ql-snow">
    <textarea name="description" id="description" class="form-control"
      data-gramm="false" contenteditable="true" maxlength='250' rows="8"><c:out value="${content}" /></textarea>
  </div>
  <label
    class="mt-4 mb-4 form-label">Starting Files</label>
  <!-- file add -->

  <input type="file" name="imageFile" id="image" >
  <input type="hidden" id="user_id" name="user_id" value="${id }">
  <input type="hidden" id="location_path" name="location_path">
  <input type="hidden" id="getlatitude" name="getlatitude">
  <input type="hidden" id="getlongitude" name="getlongitude">
</form>
```

◀ 데이터를 전송해주는 Function

▲ 게시물 작성 Form

5. 경로페이지

1 - 1. 날씨 API

```

JSONObject obj = new JSONObject(resultMap);

JSONObject parse_response = (JSONObject) obj.get("response");
// response 로부터 body 찾기
JSONObject parse_body = (JSONObject) parse_response.get("body");
// body 로부터 items 찾기
JSONObject parse_items = (JSONObject) parse_body.get("items");
/* log.info(parse_items.toString()); */
// items로 부터 itemlist 를 받기
JSONArray parse_item = parse_items.getJSONArray("item");
String category;
JSONObject weather; // parse_item은 배열형태이기 때문에 하나씩 데이터를 하나씩 가져올때 사용

for(int i = 0 ; i < parse_item.length(); i++) {
    weather = (JSONObject) parse_item.get(i);
    Object fcstValue = weather.get("fcstValue");
    Object fcstDate = weather.get("fcstDate");
    Object fcstTime = weather.get("fcstTime");
    //double형으로 받고싶으면 아래내용 주석 해제
    //double fcstValue = Double.parseDouble(weather.get("fcstValue").toString());
    category = (String)weather.get("category");

    if(category.equals("SKY")) { // 하늘상태
        sky = (String) fcstValue;
    }else if(category.equals("PTY")) { // 눈 비
        pty = (String) fcstValue;
    }else if(category.equals("TMP")) { // 현재 기온
        temp = (String) fcstValue;
    }else if(category.equals("POP")) { // 강수 확률
        pop = (String) fcstValue;
    }
}

```

```

public Weather getWeather() throws Exception {
    Weather weather_dto = new Weather();
    // 1. weather_dto 필요한 날씨 정보를 받아 리턴해줄 객체
    String sky = null, temp = null, pty = null, pop = null;
    // 2. sky, pty, 강수 확률
    String skyState = null;
    // 3. skyState에 이모티콘 값을 실어서 반영한다.
    String url = "http://apis.data.go.kr/1360000/VillageFcstInfoService_2.0/getVillageFcst"
        + "?serviceKey=GGNGzKyHIGayjNsUzlbzneX46MH8FeY93vryk5cjBe6%2FcMzK75D51Vc%2BjdsVcXRTTrRkXeuBpkryYG4ndqX%2FXLA%3D%3D"
        + "&dataType=JSON" // 요청자료형식 : JSON, XML
        + "&numOfRows=10" // 한 페이지 결과 수 : Default: 10
        + "&pageNo=1" // 페이지 번호 : Default: 1
        + "&base_date=" + dataArray().get("day") // 발표일자 : 2021년10월15일 발표
        + "&base_time=0200" // 발표시간 : 08시(정시단위)
        + "&nx=52" // 예보지점 x좌표값 격자 좌표값을 이용한다.
        + "&ny=38"; // 예보지점 y좌표값
    /*
     * 4. GET방식으로 api서버에 원하는 정보를 설정하여 받는다.
     * 좌표는 격자좌표를 사용하여 격자좌표는 기상청 api를 받을때의 엑셀문서에 포함되어있다.
     * 요청자료 형식은 json과 xml만 가능하다.
     */
    boolean isPost = false;

    if ("post".equals("get")){
        isPost = true;
    }else {
        url = "".equals("") ? url : url + "?request=" + "";
    }

    String resultMap = getStringFromURL(url, "UTF-8", isPost, "", "application/json").toString();
}

```

기상청 서버에 JSON형식의 날씨 데이터를 요청하여 전달받은 데이터들을 JSON 라이브러리를 이용하여 JSONObject 와 JSONArray로 분리한 후 원하는 값만 추출하여 웹페이지상에 띄워준다

하늘상태, 강수량, 기온을 통해 해당 날씨에 맞게 이모지를 선정하고 그 값을 현재 제주도의 날씨상황에 맞게 상시 변경해주는 기능

이 밖의, 기상청 서버에서 데이터를 가져오지 못 할 경우 Socket exception이 발생하여 경로페이지의 다른 기능에도 영향을 끼쳐 Window.onload로 페이지를 로딩 한 후 비동기 처리 방식을 이용하여 해결하였다

5. 경로 페이지

2-1. 맛집 & 관광지 카드리스트

맛집 & 관광지 버튼 클릭 시 해당 버튼의 이름을 컨트롤러로 전송한다.
컨트롤러에서는 전달받은 이름을 이용하여 맛집과 관광지를 구분하고
SQL문을 실행시켜 사용자가 원하는 정보들을 DB에서 조회하여 가져온다

가져온 정보들은 JS코드를 사용하여
카드형식으로 만들고 웹페이지에 카드리스트를 띄워준다

```
restaurantBtn.addEventListener('click', event =>{
  const xhttp = new XMLHttpRequest();

  $(document).ready(function() {
    $("#right-box3").empty();
  });

  xhttp.addEventListener('readystatechange', (e) => {
    const readyState = e.target.readyState;
    const httpStatus = e.target.status;

    if(readyState == 4 && httpStatus == 200) {
      for(var i = 0; i < Object.keys(JSON.parse(e.target.responseText)).length; ++i){
        addToList(JSON.parse(e.target.responseText)[i]);
      }
    }
  });
  xhttp.open('GET', './course/matzip', true);
  xhttp.setRequestHeader('content-type', 'application/json;charset=UTF-8');

  xhttp.send();
});
```

▲ 맛집&관광지 버튼 클릭 이벤트

```
@GetMapping(value="/course/{title}", produces = "application/json; charset=UTF-8")
public List<?> getList(@PathVariable("title") String title) {
```

```
  if (title.equals("matzip")) {
    return matzipMapper.famousList();
  } else {
    return spotMapper.courseList();
  }
}
```

```
const addToList = (jsonData) => {
  const jsonDataDiv = document.createElement('div');
  jsonDataDiv.setAttribute('id', 'jsonData');
  const div1 = document.createElement('div');
  div1.setAttribute('class', 'card mb-1');
  const div2 = document.createElement('div');
  div2.setAttribute('class', 'card-main');
  const div3 = document.createElement('div');
  div3.setAttribute('class', 'direction');
  const div_image = document.createElement('div');
  div_image.setAttribute('id', 'jsonImg');
  div_image.innerHTML = "";
  const div4 = document.createElement('div');
  div4.setAttribute('class', 'col-5 card-text');
  const div_title = document.createElement('div');
  div_title.setAttribute('class', 'text-sm mb-0 text-capitalize font-weight-bold');
  div_title.setAttribute('id', 'jsonTitle');
  div_title.innerText = jsonData.title;
  const div_location = document.createElement('div');
  div_location.setAttribute('id', 'jsonLocation');
  div_location.innerText = '좋아요 : ' + jsonData.positive_num;
  const div5 = document.createElement('div');
  div5.setAttribute('class', 'col-1 card-btn');
  const btn_plus = document.createElement('button');
  btn_plus.setAttribute('class', "rightCardSmallBtn");
  btn_plus.setAttribute('id', jsonData.title);
  btn_plus.setAttribute('value', jsonData.latitude +'/' + jsonData.longitude);
  btn_plus.setAttribute('onClick', `deleteRow(this,'this.id','jsonData'),addCard(this.id, this.value)`);
  btn_plus.innerText = '+';
  right_box3.appendChild(jsonDataDiv);
  jsonDataDiv.appendChild(div1);
  div1.appendChild(div2);
  div2.appendChild(div3);
  div3.appendChild(div_image);
  div3.appendChild(div4);
  div3.appendChild(div5);
  div4.appendChild(div_title);
  div4.appendChild(div_location);
  div5.appendChild(btn_plus);
};
```

▲ Ajax 방식 컨트롤러

5. 경로 페이지

2-2. 웹페이지 상의 카드 데이터 교환

맛집 & 관광지 카드 리스트에서 원하는 카드를 +버튼을 이용하여 선택목록에 추가하였을 때, 해당 카드를 리스트에서 삭제해주고 맵 상에 마커와 커스텀 오버레이를 생성해준다

동시에 중복검사를 통해 이미 추가된 카드는 추가할 수 없게 하였다

```
// 맵에 표시된 마커와 커스텀오버레이 삭제해주는 기능
> function deleteMarkers(map, value) {
    var coord = value.split('/');
    var latitude = parseFloat(coord[0]).toPrecision(14);
    var longitude = parseFloat(coord[1]).toPrecision(14);

    for (var i = 0; i < lines.length; i++){
        lines[i].setMap(null);
    }
    lines = [];
    for (var i = 0; i < markers.length; i++) {
        if ((markers[i].getPosition().getLat()).toPrecision(14) == latitude &&
            (markers[i].getPosition().getLng()).toPrecision(14) == longitude){
            markers[i].setMap(null);
            overlays[i].setMap(null);
            markers.splice(i, 1);
            overlays.splice(i, 1);
            positions.splice(i, 1);
        }
    }
}
```

▲ 마커와 커스텀 오버레이 삭제 함수

```
// 카드가 반대쪽으로 넘어갈 때 카드를 삭제해주고 배열에서 해당카드 인덱스를 삭제
> function deleteRow(th, thId, idName){
    var ths = $(th);
    for(var i = 0; i < names.length; ++i){
        if(names[i] == thId){
            names.splice(i, 1);
        }
    }
    ths.closest('#'+idName).remove();
};
```

▲ 해당카드 삭제 함수

```
function addCard(cardTitle, value){
    var check = 'false';
    for(var i = 0; i < names.length; ++i){
        if(names[i] == cardTitle){
            check = 'true';
        }
    }
    if(check == 'true'){
        alert('이미 추가된 항목입니다!');
    }else{
        names.push(cardTitle);

        var checker = value.split('/');
        var markerPosition = new kakao.maps.LatLng(checker[0],checker[1]); // 마커가 표시될 위치입니다
```

▲ 선택목록 카드 생성 함수 / 코드 생략

5. 경로 페이지

2-3. 카드리스트 검색

사용자가 검색 시에 검색한 값을 컨트롤러로 전송하는데,
전달받은 데이터를 이용하여 DB 테이블을 조회할 때
두 개의 테이블을 동시에 조회하기 위해서
UNION 문법을 사용하였고 포함된 단어를 모두 검색하기 위해
Ibatis like을 사용하였다

또한 검색한 값만 표출하기 위해서 jQuery 문법으로
카드들이 담기는 리스트를 비워준다

이 후, 전달받은 데이터들을 카드형식으로 구성하여
웹페이지 카드리스트상에 띄워준다

```
<select id="getListBySearchTitle" resultType="com.kgitbank.jeju.dto.SpotAndRestaurant">
    SELECT title, image, positive_num, latitude, longitude FROM
    (SELECT title, image, positive_num, latitude, longitude FROM tourist_spot
     UNION SELECT title, image, positive_num, latitude, longitude FROM famous_restaurant)WHERE title LIKE '%'||#{textValue}||'%'
</select>
```

```
@GetMapping(value="/search/{textValue}", produces = "application/json; charset=UTF-8")
public List<SpotAndRestaurant> getSearch(@PathVariable("textValue") String textView) {
    return spotAndMatzipMapper.getListBySearchTitle(textValue);
}
```

▲ 검색기능 RestController

```
magBtn.addEventListener('click', event =>{
    $(document).ready(function() {
        $("#right-box3").empty();
    });
    var textView = searchBox.value;
    search(textView);
});

// 검색기능 구현
function search(textValue){
    const xhttp = new XMLHttpRequest();
    xhttp.addEventListener('readystatechange', (e) => {
        const readyState = e.target.readyState;
        const httpStatus = e.target.status;

        if(readyState == 4 && httpStatus == 200) {
            for(var i = 0; i < Object.keys(JSON.parse(e.target.responseText)).length; ++i){
                addToList(JSON.parse(e.target.responseText)[i]);
            }
        }
    });
    xhttp.open('GET', './search/' + textView, true);
    xhttp.setRequestHeader('content-type', 'application/json; charset=UTF-8');

    xhttp.send();
};
```

▲ 검색기능 Ajax 처리 방식

▲ 검색기능 SQL문

5. 경로 페이지

3. Drag & Drop

선택된 카드목록들을 유저가 원하는 순서대로 배치 할 수 있게 드래그 앤 드롭 기능을 추가하였고,

드롭시에 데이터교환이 이루어지는데
경로 보기 및 저장시 변경된 순서를 표기해주기 위해 데이터들을 담아두었던 배열들을 변경된 순서대로 초기화하여준다.

```
<!-- 드래그 앤 드롭 -->
<link href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="https://code.jquery.com/jquery-1.12.4.min.js" ></script>
<script type="text/javascript" src="https://code.jquery.com/ui/1.12.1/jquery-ui.js" ></script>
```

▲ Import

```
$($function() {
    $("#leftSidebar-3").sortable({
        stop:function(event,ui){
            positions = [];// 바뀐 순서대로 다시 넣어주어야 해서 비워둠
            names = [];
            var vcheck = document.querySelectorAll('.cardSmallBtn');
            for(var i = 0; i < markers.length; ++i){
                markers[i].setMap(null);
                overlays[i].setMap(null);
                lines[i % lines.length].setMap(null);
            }
            markers = [];
            overlays = [];
            lines = [];
            for(var j = 0; j < vcheck.length; ++j){
                var checker = vcheck[j].getAttribute('value').split('/');
                var passName = vcheck[j].getAttribute('id');
                markerPosition = new kakao.maps.LatLng(checker[0],checker[1]);
                positions.push(markerPosition); // 마커 텍스트 초기화위해 필요
                names.push(passName);
                var marker = new kakao.maps.Marker({
                    position: markerPosition,
                    image: markerImage
                });
                marker.setMap(map);
                markers.push(marker);
            }

            for(var j = 0; j < positions.length; ++j){
                var content = '<div class="customoverlay">' +
                    ' <a>' +
                    '   <span id="numbers"></span> <span class="title">' +names[j]+ '</span>' +
                    ' </a>' +
                    '</div>';
                addText(positions[j], content);
            }
        }
    });
    $("#leftSidebar-3").disableSelection();
});
```

▲ 카드 순서 변경 코드

5. 경로 페이지

4. Map

카카오 맵 API를 이용하여 웹페이지상에 맵을 띄운 후에,
카드를 추가할 때마다 맵에 마커를 표시하고
맵 위의 여행경로 보기 버튼을 이용하여
사용자가 지정한 순서대로 여행경로를 표시해준다

```
// 맵에 마커 위에 제목을 써주는 기능
function addText(pos, con){ // 줘야할 매개변수는 포지션(좌표), 콘텐츠(들어갈 값)
    var customOverlay = new kakao.maps.CustomOverlay({
        map: map,
        position: pos,
        content: con,
        yAnchor: 1
    });
    overlays.push(customOverlay);
}
```

▲ 커스텀 오버레이 생성

```
const lookCourseBtn = document.getElementById('lookCourseBtn');
lookCourseBtn.addEventListener('click', event=>{
    for (var i = 0; i < overlays.length; i++){
        overlays[i].setMap(null);
    }
    overlays = [];
    addLine(markers);

    for(var j = 0; j < positions.length; ++j){
        var content = '<div class="customoverlay">' +
            ' <a>' +
            ' <span id="numbers">' + (j+1) + '</span> <span class="title">' + names[j] + '</span>' +
            ' </a>' +
            '</div>';

        addText(positions[j], content);
    }
});

function addLine(markers){
    var linePath=[];
    for (i=0; i < markers.length; ++i){
        linePath.push(markers[i].getPosition());
    }
    // 지도에 표시할 선을 생성합니다
    var polyline = new kakao.maps.Polyline({
        path: linePath, // 선을 구성하는 좌표배열 입니다
        strokeWeight: 2, // 선의 두께 입니다
        strokeColor: 'red', // 선의 색깔입니다
        strokeOpacity: 0.7, // 선의 불투명도 입니다 1에서 0 사이의 값이며 0에 가까울수록 투명합니다
        strokeStyle: 'solid' // 선의 스타일입니다
    });
    lines.push(polyline);
    // 지도에 선을 표시합니다
    polyline.setMap(map);
}
```

▲ 여행경로 보기 클릭 이벤트

5. 경로 페이지

5. Modal

경로페이지에서 사용하는 모달은
이용방법과 여행경로 저장 모달이 있는데,
이용방법은 슬라이드식으로 모달을 띄우고
여행경로 저장은 사용자가 선택한 경로들을 스크롤 형식으로
보기 편하게 구성하였다
모달은 버튼 클릭 시 jQuery를 이용하여
Active 를 활성화시켜준다

```
@charset "UTF-8";
.introduce-modal {
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%); /* 위에서 아래로 나올수 있게 해줌 */
  /* transform:scale(0.7); 이거쓰면 작은상태에서 커지는 효과*/
  opacity: 0;
  transition: 0.5s;
  pointer-events: none;
}

.introduce-modal.active {
  opacity: 1;
  top: 50%;
  left: 50%;
  width: 80%;
  height: 71.5%;
  transform: translate(-50%, -50%);
  pointer-events: all;
  z-index: 1050;
}
```

▲ 모달 CSS

```
<div class="introduce-modal" style="width: inherit; max-width:100%; height: auto;">
  <div class="modal-content">
    <button class="introduce-close">&times;</button>
    <div id="carouselExampleControls" class="carousel slide"
      data-bs-ride="carousel">
      <div class="carousel-inner">
        <div class="carousel-item active">
          
        </div>
        <div class="carousel-item">
          
        </div>
        <div class="carousel-item">
          
        </div>
        <div class="carousel-item">
          
        </div>
      </div>
      <button class="carousel-control-prev" type="button"
        data-bs-target="#carouselExampleControls" data-bs-slide="prev">
        <span class="carousel-control-prev-icon" aria-hidden="true"></span>
        <span class="visually-hidden">Previous</span>
      </button>
      <button class="carousel-control-next" type="button"
        data-bs-target="#carouselExampleControls" data-bs-slide="next">
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
        <span class="visually-hidden">Next</span>
      </button>
    </div>
  </div>
</div>
<div class="overlay"></div>
```

▲ 이용방법 슬라이드 모달

5. 경로 페이지

6. 여행경로 저장

사용자가 여행경로 저장 모달을 통해
자신이 지정한 경로들을 확인한 후에 저장버튼을 클릭 시
POST 방식으로 컨트롤러로 여행경로 데이터들을 전송하는데
이 때, 비로그인 상태라면 로그인 페이지로 Redirect 시켜준다

로그인시에는 SQL문을 이용하여 여행경로 아이디(PK,시퀀스)를
조회하고 이 후 데이터들을 DB에 저장한다

저장시에는 업데이트 된 시퀀스의 마지막 값을 가져오기 위해
selectKey와 BEFORE 문법을 통해 구현하였다

```
@PostMapping("/course/save")
public String getTravelCourse(HttpServletRequest req, HttpSession session) {
    if(!LoginVerifier.isLogin(session)) {
        return "redirect:/login";
    }
    String user_id = session.getAttribute("id").toString();
    MyCourse course = new MyCourse();
    course.setUser_id(user_id);
    myCourseMapper.addDTOCourse(course);

    String[] images = req.getParameter("imagePath").split(",");
    String[] names = req.getParameter("namePath").split(",");

    for(int i = 0; i < images.length; ++i) {
        CourseDetail detail = new CourseDetail();
        detail.setSeqs(i+1);
        detail.setImage(images[i]);
        detail.setTitle(names[i]);
        courseDetailMapper.insertDTOCourse(detail);
    }
    return "redirect:/mypage";
}
```

▲ 여행경로 저장 Controller

```
<insert id="insertDTOCourse">
    <selectKey keyProperty="id" resultType="integer" order="BEFORE">
        SELECT (MAX(id)) FROM mycourse
    </selectKey>

    INSERT INTO courseDetail(id, seqs, title, image) VALUES(#{id}, #{seqs}, #{title}, #{image})
</insert>
    <select id="getListByCourseId" resultType="com.kgitbank.jeju.dto.CourseDetail">
        SELECT * FROM coursedetail WHERE id=#{id} ORDER BY id, seqs
    </select>
```

◀ 여행경로 저장 SQL문

6. Polygon

폴리곤을 만들기 위해서 대한민국 행정표준코드를 이용하여
제주도 부분만 QGIS 프로그램에 띄운다

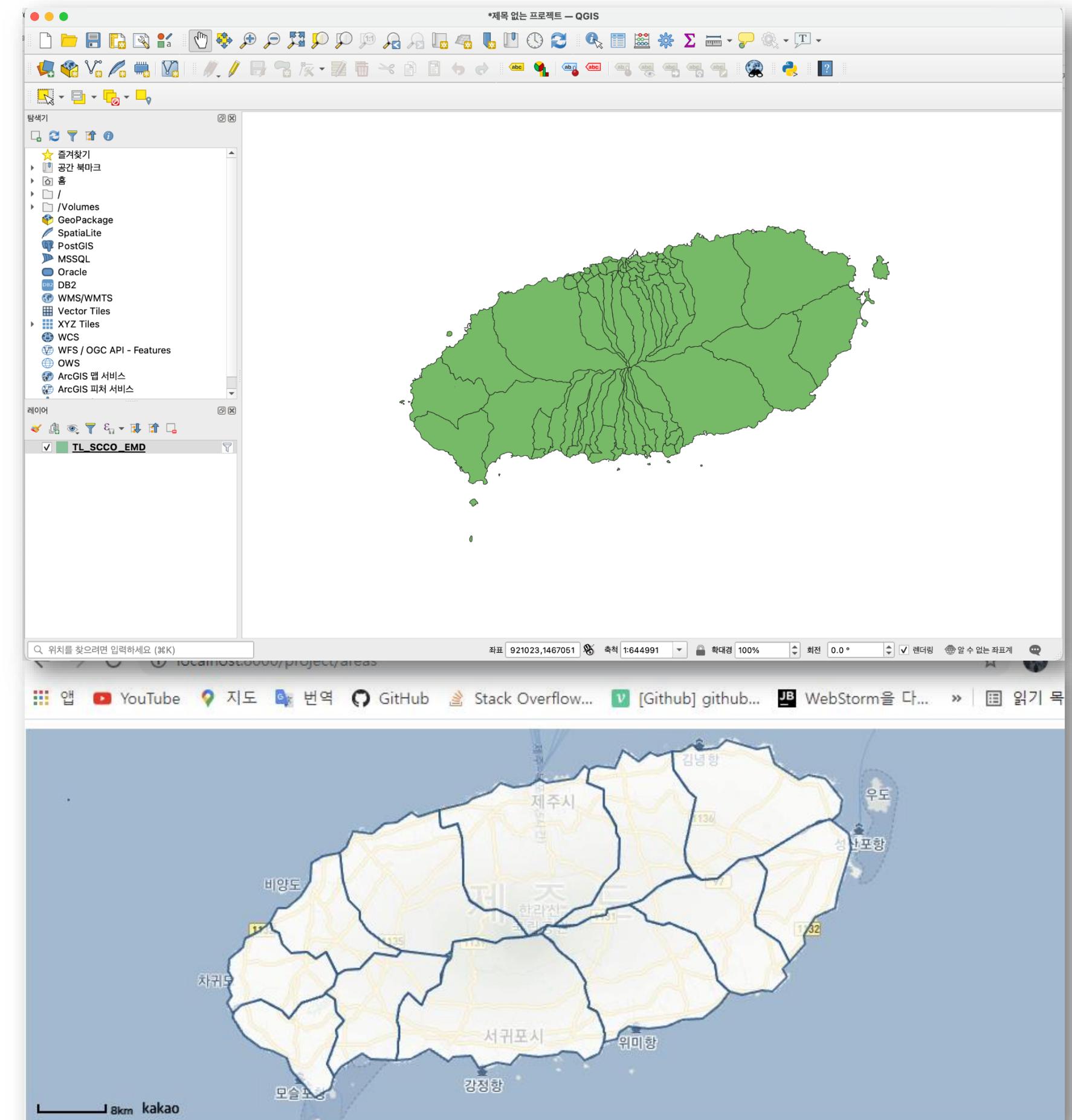
그 후에, 병합하고 싶은 부분들의 백터를 병합하고
저장시에 좌표계를 카카오 맵 API와 통일시켜준다
또한, 서버 과부하를 방지하여 맵을 단순화시켰다

다음으로는 .shp파일을 MyGeoData Converter를 이용하여 geojson
파일로 변환한다

마지막으로 json형식의 데이터를 서버에서 읽어 폴리곤을 만든다

폴리곤에는 마우스 이벤트들을 등록하였다(click,mouseover)

The screenshot shows the MyGeodata Converter interface. In the 'Input Data' section, a file named 'map.shp' is selected. In the 'Output Data' section, the 'Output Format' is set to 'GeoJSON'. In the 'Conversion' section, there is a preview map of Jeju Island with a red rectangle highlighting a specific area. Below the map are 'Show in a Map' and 'Convert now!' buttons.





제주특별자치도 관광 안내 홈페이지



연락처 010-0000-0000
이메일 jeju_spring@gmail.com