



# Bob Morgan

밥 먹언

JEJU MUST-GO SEARCH/RESERVATION WEBSITE

# TEAM MEMBERS

Frontend 박수연  
이범석  
이후경  
정예지

Backend 이태권



# CONTENTS

01 Sprint  
Routine

03 Code  
Review

02 Project  
Architecture

04 Project  
Review

# SPRINT ROUTINE

## NOTION SPRINT MEETING

### ! 필수 기능 구현

#### Frontend

1. 소셜 로그인 페이지 (수연님)
  - a. 카카오 로그인/회원가입 [/kakao/signin](#)
2. 메인 페이지 (예지님)
  - a. 슬라이드
  - b. 검색 기능(지역, 음식 종류)
3. (상세 검색 + 전체 맛집) 리스트 (범석님)
  - a. 지역 검색 [/regions](#)
  - b. 음식 종류[한, 중, 일, 양, 카] [/places](#)
  - c. 필터링
    - i. 가격대 필터링
    - ii. 지역 필터링
    - iii. 필터 초기화 버튼
  - d. 마우스 좌표 기능 애니메이션 배너
  - e. 랜덤 맛집 리스트 캐러셀?
  - f. 맛집 무한 스크롤
4. 맛집 상세 페이지 (후경님)
  - a. 맛집 이미지 캐러셀
  - b. 네이버 플레이스 연결 버튼
  - c. 예약 기능

#### Backend

- (태권님)
1. 소셜 로그인 API (GET)
  - a. 카카오
  2. 지역, 카테고리 API (GET)
  3. 리스트 페이지 API (GET)
  4. 상세 페이지 API (GET)
  5. 예약 API (POST)

1. 소셜 로그인  
2. 맵  
3. 예약 기능  
4. 필터링

### + 추가 기능 구현

1. 마이페이지 (+ 예약 조회 및 수정, 삭제)
2. 최근 검색 맛집

# SPRINT ROUTINE

## NOTION DAILY STANDUP MEETING

밥 먹언(Bob Morgan) / [밥 먹언] Daily Meeting Table

+ :: 7월 7일 (목)

팀원	어제 한 것	오늘 할 것	Blocker
박수연	-소셜 로그인(인증 코드 받고, 인가 토큰 받기) 구현 -태권님과 통신 시도	-소셜 로그인 재도전, 태권님과 통신 시도	-토큰을 어떻게 서버에게 전달해야 할지?
이범석	-검색 페이지 목데이터, 태권님과 상의하여 만들 -음식 가격 범위(range) 레이아웃 작성 -식당 이름 검색	-체크박스에 필터링(단일 조건)	-체크박스에 필터링(여러 조건) → 추가 기능 구현 해 보기
이후경	-상세 페이지 캐러셀(슬라이드) 부분 완성 → 대략적인 상세 페이지 완성	-모달창 구현 -한 줄 평(추가 기능)을 넣을 것이라면, 페이지 하단에 구현해야 함	-지도 API -달력 API
정예지	-메인 레이아웃, 메인 슬라이드 구현	-모달창 구현	-모달창
이태권	-장고 모델링 수정 사항 모두 반영 -csv 파일을 모두 데이터베이스에 담음 -카카오 로그인 코드 분석 및 개발자 문서 숙지	-수연님과 실제로 카카오 로그인 구현 -모델링에 관하여 프론트에게 설명 및 피드백 + 검색 페이지 API 범석님과 의논	-카카오 로그인 토큰 문제 + 전체적인 소셜 로그인 flow와 백엔드에서 처리할 것
공통			

# SPRINT ROUTINE

TRELLO

Sprint Routine Trello Board

This screenshot shows a Trello board titled "wecode 34th project" with five lists: Backlog, To Do, In Progress, In Review, and Completed. The board is set against a background image of a mountain landscape.

- Backlog (앞으로 해야 할 티켓)**
  - FE & BE 공통
  - Frontend
    - 상세페이지 리뷰 (댓글) 기능
    - 예약하기 모달창 (기능)
    - 마이페이지 기능 구현
  - Backend
    - '먹고 싶어요' 기능 구현
    - '마이 페이지' API
- To Do (스print에 해야 할 티켓)**
  - FE & BE 공통
  - Frontend
    - 모델링에 필요한 API 정리
  - Backend
- In Progress (진행 중인 티켓)**
  - FE & BE 공통
  - Frontend
    - 예약 component 레이아웃
    - 상세페이지 (기능) 받아온 정보 화면 렌더링
    - 상세페이지 캐러셀
    - 식당 검색기능
    - 상세페이지 (기능) 주소 지도
    - 리스트페이지 캐러셀
    - 페이지네이션 기능 구현
    - 상세페이지 리뷰 (댓글) 레이아웃
    - 상세페이지 레이아웃
    - 예약 component 기능 구현
    - Backend
  - Backend
    - 식당 필터링
    - 페이지네이션 기능 구현
    - 지역 체크박스 필터링 기능
    - 식당 구분에 따른 체크박스 필터링
- In Review (PR 이후 리뷰 진행 중인 티켓)**
  - FE & BE 공통
  - Frontend
    - 식당 검색기능
    - 리스트페이지 캐러셀
    - 페이지네이션 기능 구현
    - 상세페이지 리뷰 (댓글) 레이아웃
    - 상세페이지 레이아웃
    - 예약 component 기능 구현
    - Backend
  - Backend
    - 식당 필터링
    - 페이지네이션 기능 구현
    - 지역 체크박스 필터링 기능
    - 식당 구분에 따른 체크박스 필터링
- Completed (완료된 티켓)**
  - Frontend
    - 소셜로그인 레이아웃
    - 소셜 로그인 카카오톡 API 연결
    - 로그인 여부에 따른 버튼 상태 변화
    - 마이페이지 레이아웃
    - 로그인 페이지 리팩토링
    - Navigation 레이아웃
    - 초기 세팅
    - 메인페이지 캐러셀
    - 모달안에 달력기능추가
    - 메인 모달안 카카오맵 설정
    - 메인페이지 검색창레이아웃 만들기
    - 메인페이지 검색창 모달 기능구현
    - main 레이아웃
  - Backend
    - POST /reservations
    - '예약 하기' 기능 구현
    - TESTS / VIEWS / GET / PLACES
    - 맛집 상세 페이지 API

+ Add another list

4

# PROJECT ARCHITECTURE

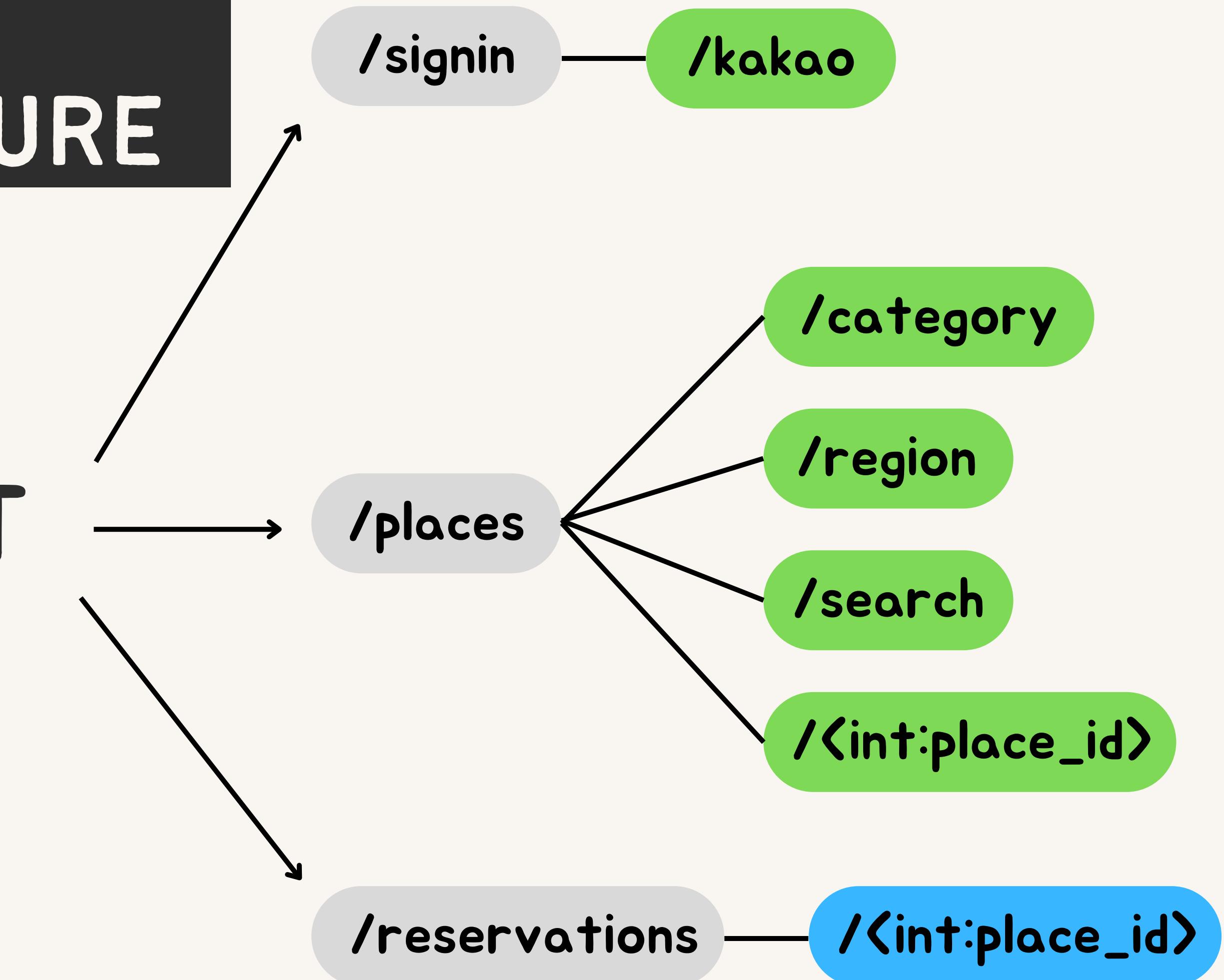
## MODELING



# PROJECT ARCHITECTURE

## ENDPOINT

GET  
POST



# 이태권

# CODE REVIEW

## PlaceSearchView

```
places = Place.objects.select_related('category', 'region').prefetch_related(
    'placemenu_set__price', 'placemenu_set__menu', 'menus', 'image_set').annotate(
        min_price = Min('placemenu__price__price'),
        max_price = Max('placemenu__price__price'),
        avg_price = Avg('placemenu__price__price')
    ).filter(q_region & q_category).order_by(order_key)[offset : offset + limit]

results = [
    'place_id' ..... : place.id,
    'place_name' ..... : place.name,
    'place_opening_hours' ..... : place.opening_hours,
    'place_maximum_number_of_subscriber' ..... : place.maximum_number_of_subscriber,
    'place_able_to_reserve' ..... : place.able_to_reserve,
    'place_closed_temporarily' ..... : place.closed_temporarily,
    'place_category' ..... : place.category.name,
    'place_region' ..... : place.region.name,
    'place_image' ..... : place.image_set.all().first().image_url
        if place.image_set.all() else None,
    'menus' : [
        'id' ..... : placemenu.menu.id,
        'name' ..... : placemenu.menu.name,
        'price' ..... : placemenu.price.price,
        'is_signature' ..... : placemenu.is_signature,
    } for placemenu in place.placemenu_set.all()],
    'menu_avg_price' ..... : place.avg_price,
    'menu_min_price' ..... : place.min_price,
    'menu_max_price' ..... : place.max_price,
} for place in places]

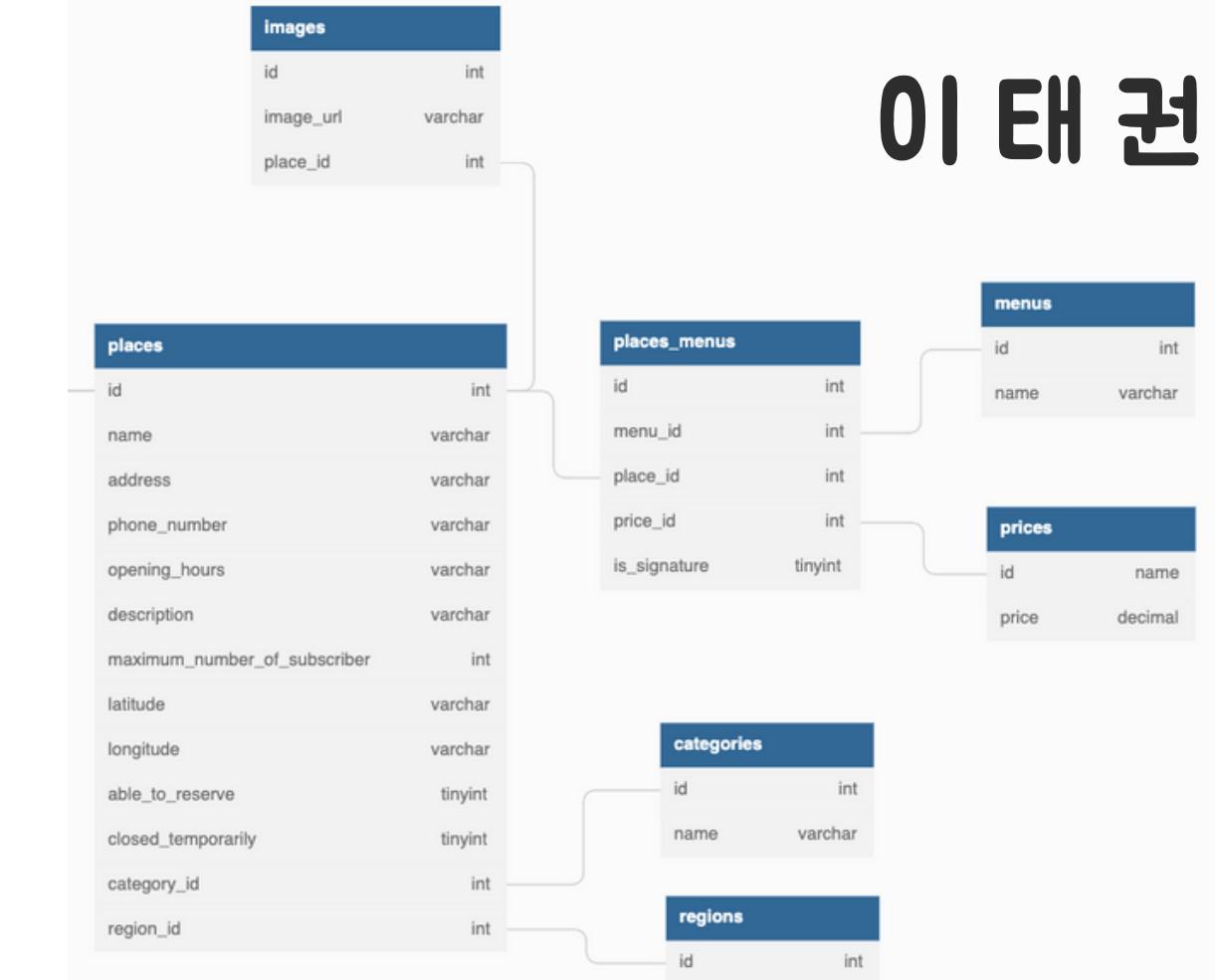
return JsonResponse({'results' : results, 'date' : date}, status = 200)
```

## ORM 최적화

- `select_related`
- `prefetch_related`



약 1/8 감소!



localhost:8000/places/search?region=1&date=2022-07-12

### before

```
-----
Function : get
Number of Queries : 170
Finished in : 0.15s
-----
```

### after

```
-----
Function : get
Number of Queries : 22
Finished in : 0.06s
-----
```

# WEBSITE DEMONSTRATION

The screenshot shows a website interface. At the top left is the logo "Bob Morgan" with an orange fruit icon. The top navigation bar includes links for "홈", "제주맛집", and "마이페이지". On the right side of the header are buttons for "친절상담 \* 8716-6728", "로그인", and a user icon. The main content area features a large, abstract image of a textured surface in shades of green and brown. Below this is a search form with fields for "지역" (Region) containing "제주 검색", "날짜" (Date) containing "날짜 선택", and a large orange "검색" (Search) button with a magnifying glass icon and the text "맛집 검색".

<https://scrawny-opera-4c6.notion.site/ebad7fd9511642fa9de656e1226aaf07>

# CODE REVIEW

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <ThemeProvider theme={theme}>
    <GlobalStyle />
    <RecoilRoot>
      <Router />
    </RecoilRoot>
  </ThemeProvider>
);
```

- RecoilRoot로 Router 감싸주기

```
import { atom } from 'recoil';

export const tokenState = atom({
  key: 'tokenState',
  default: '',
});

const [token, setToken] = useRecoilState(tokenState);
```

- 컴포넌트가 atom을 읽고 쓰게 하기 위해 useRecoilState()와 함께 사용하기.
  - atom의 값을 읽는 컴포넌트들은 암묵적으로 atom을 구독.
  - atom에 변화가 생기면 atom을 구독하는 모든 컴포넌트들이 재 렌더링.

- 원했던 것 : 로그인 여부에 따라 자동으로 바뀌는 버튼.
- 실제로 나온 것 : 수동 새로 고침으로만 바뀌는 로그인/로그아웃 버튼
- 원인 : 그저 Local Storage에 값이 있는지 없는지에 따라 상태를 변화하게 하려고 하니, Navigation에 업데이트 된 내용이 없어 렌더링이 일어나지 않음.
  - 개선 방법 : 전역 변수에 토큰을 담아 관리 해주기.  
(Recoil을 활용하여 로그인/로그아웃 버튼 상태 변화 구현)

# CODE REVIEW

```
/*global kakao*/
import React, { useState, useEffect, useRef } from 'react'

useEffect(() => {
  let options = {
    center: new kakao.maps.LatLng(33.50431022611561, 126.49432520073913),
    level: 4,
  };

  let map = new kakao.maps.Map(mapId.current, options);
  let geocoder = new kakao.maps.services.Geocoder();

  let imageSrc = '/images/bobMorgan-map-pick2.png';
  let imageSize = new kakao.maps.Size(36, 50);

</Tab>
<Tab eventKey="mapZone" title="지도">
  <S.MapBox>
    <S.MapContainerBox ref={mapId} />
  </S.MapBox>
  <S.MapContainerBoxTxt id="centerAddr">
</Tab>
```

- **Javascript로 만들어진 KAKAOMAP 리액트에 적용시키기**
- **/\*global kakao\*/ 이렇게 선언하는걸로 카카오맵 실행 가능!**
- **직접 DOM에 접근하기 보다는 useRef로해서 접근하기!**
- **useRef만 써서 되는게아닌 .current까지 써주면서 해야지 카카오 맵이 실행 가능해진다~**
- **그리고 카카오가 쓴 NAMING들은 꼭 공식문서에서 찾아서 무슨 기능으로 썼는지 꼭 확인해주기!**

# CODE REVIEW

```
const updateSearchInput = e => {
  setSearchInput(e.target.value);
};

const sortedPlaces = placeList.filter(place => {
  return place.place_name.includes(searchInput);
});
const range = sortedPlaces.filter(range => {
  return (
    values[0] <= range.menu_avg_price / 10000 &&
    range.menu_avg_price / 10000 <= values[1]
  );
});
```

- 필터 함수를 이용한 식당 이름, 가격 필터링 기능 구현
- 몬스터에서 배운 내용을 직접 적용해보고 싶어 직접 필터링 기능을 구현 했다.

```
{range.length === 0 && <Empty />}
{range.map(place => {
  return <PlaceList place={place} key={place.place_id}>
})}
```

# CODE REVIEW

\_KAKAOMAP\_API\_KEY=70

key=%REACT\_APP\_KAKAOMAP\_API\_KEY%&librar

202:8000/reservations/

선호하는 네트워크



WeWork



- 카카오 맵 API키는 개인 정보기 때문에, 소중하게 숨겨서 사용하기 .  
env파일로 관리하되 git ignore에 기재해서, git에도 올리지 않게 하기
- <script></script> 안에서 %여기% 기재해서 사용함.
- 제발 endpoint를 확인하기 제발...
- 제발 wifi 확인하기 제발...  
위워크는 왜 나에게 개인 와이파이를 줬을까....

# PROJECT REVIEW

BE : FE = 1 : 4... 그 어느 때보다 바빴지만 그만큼 얻을 수 있었던 값진 경험들

약한 소리 금지... 이것도 금지... 저것도 금지... 서로가 서로에게 금지... 금지도 금지...

뿌리가 탄탄하지 못했던 기획... 그만큼 팀원들을 고생시키지 않았는가... 😞

기능 구현에만 치우치지 않고 틈틈이 리팩토링을 놓치지 않았다

# PROJECT REVIEW

정 예 지 : 더 많은 기능 구현하고 싶었는데 😡 하루만 더 준다면.. 범석님 디자인 고쳐주고 싶다...

박 수연 : 다시는 소셜 로그인을 무시하지 않겠습니다... 잊지 못할 git rebase -i main...!

이 범석 : 자식 잃은 부모의 기분을 느껴보셨습니까...? 😢 하루만 더 준다면... 예지님..비상..비상..!

이 후 경 : 모든 단계가 거대한 산이었어요 그 산을 넘은 내가 최고야.. 도와준 당신들도 최고야..

THANK YOU

고맙습니다

밥 먹언 (BOB MORGAN)