

React와 NestJS를 활용한 맛집 지도 웹 애플리케이션 개발



부산대학교 정보컴퓨터공학부
202255673 최민기
202155510 권소현

지도교수: 손준영

목차

1. 연구 배경 및 목표
 - a. 연구 배경
 - b. 과제 목표
2. 요구조건 분석
3. 연구 방향
 - a. 위치 기반 지도 기능
 - b. 사용자 인증 및 보안
 - c. 이미지 기반 검색 기능
 - d. 전체 구성도 및 사용 기술
4. 현실적 제약 사항
5. 개발 일정 및 역할 분담
 - a. 개발 일정
 - b. 역할 분담
6. 참고문헌

1. 연구 배경 및 목표

a. 연구 배경

연구 배경 최근 사용자가 음식 사진이나 키워드를 통해 자신이 원하는 음식점 정보를 찾는 수요가 증가하고 있다. 그러나 기존 맛집 플랫폼은 위치 기반 추천은 가능하지만, 감성 분석을 통한 리뷰 품질 평가나 이미지 기반 음식 인식 기능은 부족한 실정이다. 이에 따라 본 프로젝트는 **Python**을 활용하여 크롤링, 감성분석, 이미지 인식, 지도 기반 시각화를 모두 포함하는 맛집 랭킹 웹 애플리케이션을 개발하고자 한다.

b. 과제 목표

React를 기반으로 한 웹 애플리케이션 형태로 맛집 랭킹 서비스를 개발하고자 한다. 사용자가 음식 키워드 또는 이미지를 입력하면, 관련된 맛집 데이터를 수집·분석하여 랭킹을 보여주는 기능을 제공한다. 이를 통해 결과를 지도와 워드클라우드 등 시각화 형태로 제공함으로써, 사용자의 검색 의도에 부합하는 맞춤형 정보를 제공한다. 웹 크롤링, 이미지 분석, 감성 분석 등을 복합적으로 활용하고 사용자 인증을 적용하여 개인화된 데이터 보호 기능 제공한다.

2. 요구사항 분석

- 사용자 위치 탐색 및 지도 중심 위치 설정 기능 (**navigator.geolocation** 활용)
- 맛집 리스트 및 블로그 리뷰 크롤링 (네이버 지도, 블로그 등)
- 텍스트 및 이미지 기반 음식 검색 처리 (**Vision API** 또는 **CLIP** 활용)
- 감성분석 모델을 통한 리뷰 품질 분석 및 랭킹 산출
- 지도 시각화 및 워드클라우드 출력 기능
- 웹에서 이미지 업로드 및 검색 기능 지원
- 사용자 인증(**JWT**)을 통한 데이터 접근 보호

3. 연구 방향

a. 위치 기반 지도 기능

Google Maps API 또는 Kakao Map API를 연동하여 사용자의 현재 위치를 기준으로 지도 화면을 구성한다. 사용자가 입력하거나 탐색한 위치 정보에 따라 주변 맛집 정보를 마커로 시각화하고, 사용자가 등록한 장소를 기반으로 마커 목록이 필터링된다. 마커 클릭 시 상세 정보를 확인할 수 있도록 구성하고, 반경 필터링을 통해 사용자가 설정한 거리 이내의 맛집만 조회할 수 있도록 구현한다.

b. 사용자 인증 및 보안

NestJS에서 JWT(Json Web Token)를 기반으로 사용자의 로그인 정보를 보호하며, 인증된 사용자만 본인의 데이터에 접근 및 수정이 가능하도록 백엔드를 구성한다. 토큰은 클라이언트에 저장되며 요청 시 함께 전송되어 서버에서 사용자 식별 및 권한 확인에 활용된다.

c. 이미지 기반 검색 기능

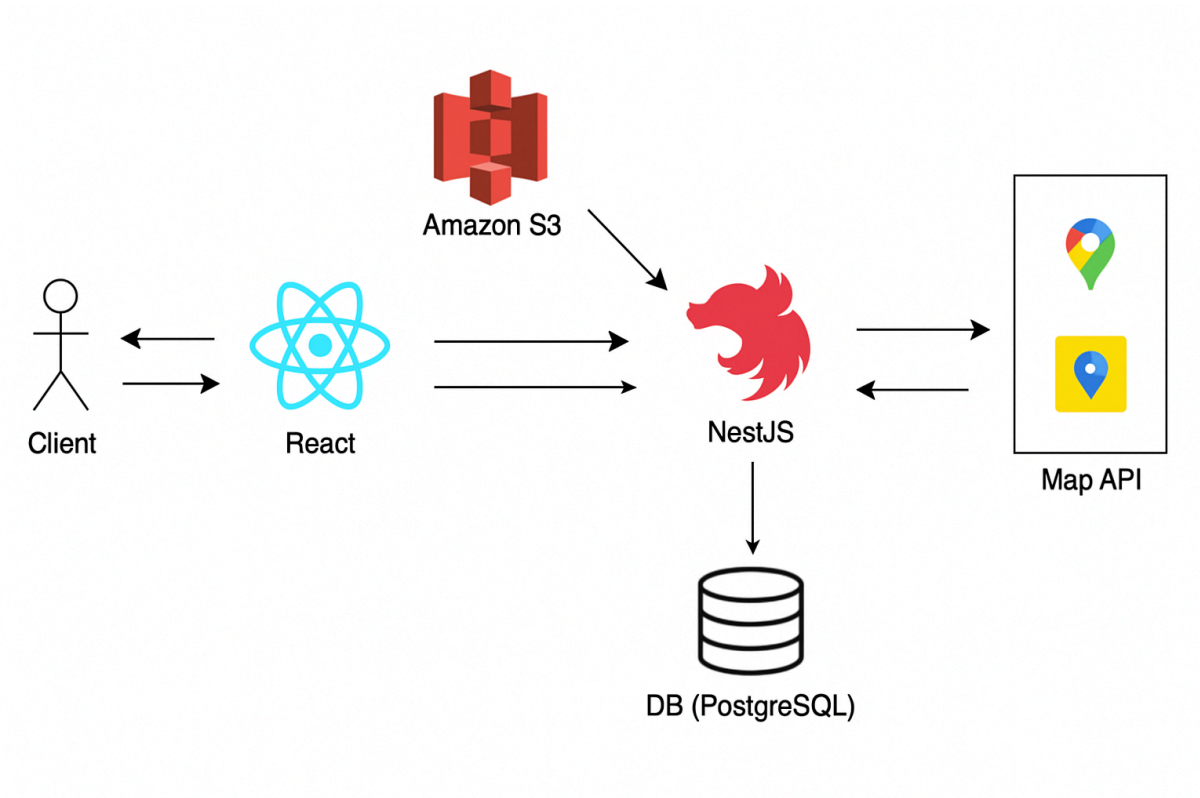
텍스트 및 이미지 기반 검색 사용자는 음식 키워드를 텍스트로 입력하거나 이미지 업로드를 통해 검색을 수행할 수 있다. 이미지의 경우 Google Vision API 또는 CLIP 모델을 활용하여 음식 이름을 추출하고, 이를 기반으로 관련 맛집 데이터를 필터링하여 제공한다. 텍스트와 이미지 검색은 공통된 랭킹 알고리즘에 의해 결과가 정렬된다.

d. 날씨 정보 연동 기능

사용자 위치 또는 선택된 맛집의 좌표를 기반으로 OpenWeatherMap API를 호출하여 해당 지역의 현재 날씨 정보를 확인할 수 있도록 구현한다. 지도 마커 클릭 시 또는 검색 결과와 함께 "현재 날씨: 맑음, 24°C"와 같은 정보를 간단히 제공하며, 날씨 상태에 따라 간단한 경고나 안내 문구도 출력할 수 있다. 이 기능은 사용자 경험 향상을 위한 보조 정보로 활용된다.

e. 전체 구성도 및 사용 기술

- 프론트엔드: React , HTML, CSS
- 백엔드: NestJS, Flask, JWT
- 인프라:PostgreSQL (DB), Amazon S3
- API 연동: Google Maps API 또는 Kakao Maps API, OpenWeatherMap API(추후 Vision API)
- 개발 환경: GitHub, Docker, VS Code
- 디자인: Figma 기반 UI 시안 제공 및 구현



4. 현실적 제약 사항

- 웹에서 제공하는 위치 정보 정확도가 기기 성능에 따라 차이 있음
- 이미지 분석 **API** 사용 시 요금 발생 가능성 존재
- 네이버 등 외부 웹사이트의 크롤링 차단 및 로봇배제정책 고려 필요
- 감성 분석 모델의 학습 및 튜닝에 시간 소요됨
- 실시간 **API** 응답 속도 최적화 필요

5. 개발 일정 및 역할 분담

a. 개발 일정 (월별)

월	작업 내용
5월	주제 선정 및 사전 조사, 요구사항 정리
6월	디자인, 기본 UI/UX 설계, 지도 연동 기능 구현
7월	프론트 구조 구축(이미지 업로드), 크롤링 및 감성분석 로직 개발(중간보고)
8월	백엔드 구축 (각종 API + 감성분석/크롤링 연동), 보안 시스템 및 DB 설계
9월	이미지 검색 기능 구현, 마커/피드 기능 개발
10월	테스트 및 배포, 발표 자료 정리

5월				6월				7월				8월				9월				10월			
W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
주제선정, 사전조사																							
	요구사항 정리																						
		디자인, UI/UX 설계																					
			지도 연동 기능 구현																				
				프론트 구축, 크롤링, 감성분석 기술 개발																			
							중간 보고																
								백엔드 구축, 보안 시스템 및 DB 설계															
												이미지 검색 기능 구현											
													마커/피드 기능 개발										
																			테스트 및 배포				
																						발표 준비	

b. 역할 분담

이름	담당 역할
권소현	프론트엔드 개발 (지도 연동, 마커 처리, 이미지 업로드/검색)
최민기	백엔드 개발 (NestJS API, DB 설계, 사용자 인증 보안 및 배포 환경)

6. 참고 문헌

- React 공식 문서
- NestJS 공식 문서
- Google Maps Platform
- Kakao Developers API 문서
- OpenWeatherMap API 문서
- OpenAI Vision API 문서