

이화여대 재학생들의 진짜 지도



숨겨져 있던
지름길까지 다 있다

안내 목차

잇다(It-da)
소개

개발 목적 소개

기능 소개

핵심 기능 및 부가 기능

기술 설명

사용된 AI, 백엔드,
프론트엔드 기술 설명

차별점

잇다맵만의 차별점,
향후 발전 가능성

당신의 길찾기를 도와 줄

잇다(It-da)를 소개합니다.

새내기 때 다들 길 잃어 본 적 있으시죠?

더 빠른 길, 더 편한 길은 없을까

고민해 본 적 없으신가요?

이제 잇다(It-da)가 도와드릴게요.



실용적인

경로를 예상 소요 시간과 함께

재학생들만 아는 지름길부터 건물 간 연결 통로까지,
모두 반영한 가장 빠른 길을 알려드려요.

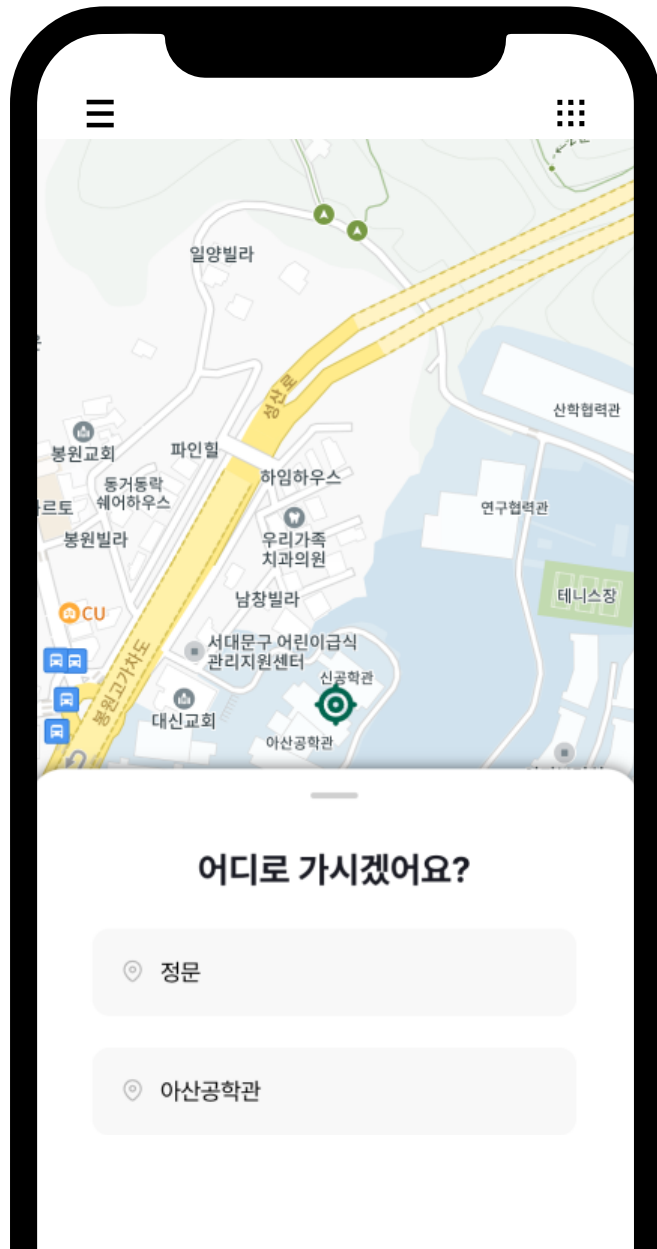


셔틀 정보 비교해서 선택하세요

급할 때 셔틀 탈지, 걸어갈지 고민되셨죠?
잇다가 알려드릴게요.

부가 기능

더 많은 사람들에게, 더 쉽게 보여드려요



1

번역 제공

2

계단, 경사로 표시

3

지름길 설명 정보 제공

부가 설명 작성란

아직 미정



문제 상황

- 기존 이화캠퍼스맵
(<https://www.ewha.ac.kr/ewha/intro/campus-map.do>)
- 카카오 맵 api 지도를 기반으로 하여
길찾기만 사용
- 큰 도로 위주의 길로, 학교의 모든 건물
검색이 힘들





개선점

- 카카오 맵 api의 지도만을 기반으로, polyline을 직접 그어 길을 표시하였음
- 학교내 모든 건물을 좌표값으로 표시하여 구분, 모든 건물 검색 가능



AI

사용 기술 스택

- [깃헙 링크]

(https://github.com/samsungchungso/image_analyst/blob/main/README.md)

- 딥러닝 모델 패키지 - Road Extraction - Global

- 입력: 8-bit, 3-band 고해상도 (1m급) 항공/위성영상

- 출력: 도로네트워크 피쳐

- 모델아키텍처: Songtao He 연구팀이 제작한 Sat2Graph 모델 기반

- 성능지표: 도시급 데이터셋에서 F1 score 값: 76.26, 정밀도: 0.807, 재현율: 0.723)

Sat2Graph 모델

- 고해상도(1미터) 항공/위성 이미지에서 도로를 추출

- 도로 레이어는 캠퍼스 개발, 변화 감지, 인프라 계획 등 다방면에서 사용 가능

- 일반적으로 이미지에서 도로를 디지털화하는 작업은 손으로 직접 디지털화해야하나, Sat2Graph 모델의 경우 넓은 캠퍼스 범위의 도로를 빠르게 디지털라이징 가능함

- 딥러닝 모델을 사용하여 캠퍼스를 안내하는 '잇다'의 길찾기 프로세스를 자동화하고 비용을 절감함

이미지에서 오브젝트 추출 - 이화여대 캠퍼스 도로



- 입력 래스터(위성 이미지)에 대해 훈련된 딥러닝 모델을 실행하여 탐색하고자 하는 객체를 피쳐 클래스로 생성(피쳐는 탐색 객체인 도로의 polyline)
- 딥러닝 모델 패키지로 산출된 TensorFlow로 훈련된 모델을 포함한 Detect Objects Using Deep Learning 도구를 이용

발전 방향 제시 - automatic road extraction using deep learning

- 타 대학 캠퍼스 도로 추출 및 서비스 확대
- 도로 추출 워크플로 자동화

Backend

사용 기술 스택

- Spring Boot
- Project Lombok (lombok)
- MariaDB JDBC Driver: Vertex, Edge, Building Database
- JGraphT (jgrapht-core) : Dijkstra Algorithm
- Spring Boot Test (spring-boot-starter-test) : Test

사용 언어

- Java : Server, API 구현
- Python3 : Vertex, Edge, Building JSON → SQL
- MySQL : 데이터 조작에 사용

ERD 설계

![[Untitled]](<https://prod-files-secure.s3.us-west-2.amazonaws.com/41b6-b3bc-a7c4b63f041e/ba2a430e-71b9-4b56-a656-41b6b3bc-a7c4b63f041e/Untitled1.png>)

핵심 기능 - 최단 경로 찾기

- 입력 : 출발지 건물명, 도착지 건물명
- 출력 : 최단 경로, 건물 정보
- 사용 알고리즘 : Dijkstra Algorithm
- API 작동 방식
- 사용자가 입력한 Building Name으로 Building Table 조회하여 Building 객체 저장 (start, end)
- Vertex Table에서 Building_id(FK)와 [Building.id] (<http://Building.id>)가 일치하는 Vertex 조회하여 Vertex 객체 저장
- Vertex, Edge 정보 모두 불러와 가중치(time)가 있는 Multigraph 생성
- Start vertex, end vertex, Multigraph를 가지고 Dijkstra Algorithm 적용
- Start vertex, end vertex, minutes(total times), 방문한 Vertex List가 담긴 PathResult 객체로 응답

<aside>

 셔틀 구현하면 변경될 수 있음

</aside>

부가 기능 - 건물명 검색

- 입력 : 건물명
- 출력 : 검색어가 포함된 건물명을 가진 건물 List
- API 작동 방식
- 사용자가 입력한 Building Name으로 Building Table 조회하여 Building 객체 저장
- Vertex Table에서 Building_id(FK)와 [Building.id] (<http://Building.id>)가 일치하는 Vertex 조회하여 Vertex 객체 저장
- Building 객체가 포함된 Vertex 객체 list로 응답

Frontend

사용 기술 스택

- React: create-react-app
- PWA(Progressive Web App) 구성
- Styled-Components
- Kakao Maps API
- 주요 라이브러리
- react-kakao-maps-sdk: 지도 구현
- framer-motion: bottom sheet modal 구현
- axios: 서버 연결

사용 언어: Typescript, Javascript

핵심 기능

- 장소 검색
- 길안내

