

날씨 기반 의상 추천 서비스

클라우드 기반 데이터베이스 연동



2팀: 박정환, 강성대, 이동훈, 김유진, 김도윤

2025년 7월 25일

프로젝트 개요 및 목적

프로젝트 소개

날씨 기반 의상 추천 서비스는 Flask 웹 애플리케이션을 활용하여 현재 날씨 정보를 기반으로 적절한 의상을 추천해주는 서비스입니다.

개발 목적

클라우드 환경 실습

클라우드 환경에서 데이터베이스 연동 기술 학습

외부 API 연동






기상청 날씨 정보 API를 활용한 실시간 데이터 처리

사용자 중심 서비스

날씨 정보를 기반으로 실용적인 의상 추천 제공



팀원 및 역할

| | | | |
|---|-----|--------------|---|
|  | 박정환 | PM | 프로젝트 기획 및 전체 관리, 테스트 담당. 팀원 간 의사소통 조율. |
|  | 강성대 | 기술총괄 | 서버 연결 및 배포, 배포용 DB 구축 및 API 연동. |
|  | 이동훈 | 프론트엔드 | 사용자 인터페이스 개발, Jinja2 템플릿 구현 및 DB 연동 뷰 구성. |
|  | 김유진 | API-DB 연결 | 외부 API와 데이터베이스 연동, 데이터 처리 및 가공 로직 구현. |
|  | 김도윤 | API-뷰 연결, DB | API 데이터를 뷰에 표시하는 로직 구현, 데이터베이스 스키마 설계. |

주요 기능

현재 날씨(온도) 조회 기능

- 사용자가 선택한 지역의 현재 온도 정보 제공
- 기상청 API를 통한 실시간 날씨 데이터 조회
- 지역별 좌표(nx, ny) 기반 정확한 날씨 정보 검색

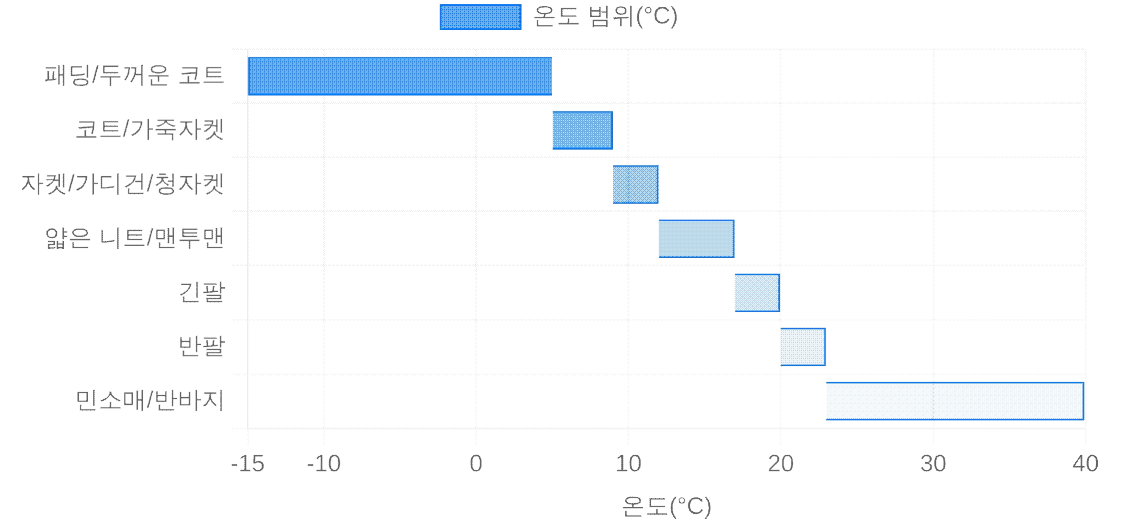
온도별 의상 추천기능

- 현재 온도에 적합한 의상 추천
- 데이터베이스에 저장된 온도별 의상 정보 활용
- 사용자 친화적인 추천 메시지 제공

지역별 날씨 정보 조회

- 전국 17개 시/도 지역 선택 가능
- 지역별 맞춤형 날씨 정보 제공
- 사용자 위치 기반 서비스 확장 가능성

온도별 의상 추천 범위



온도별 옷차림 추천



프로젝트 구조

✧ 구조 개요

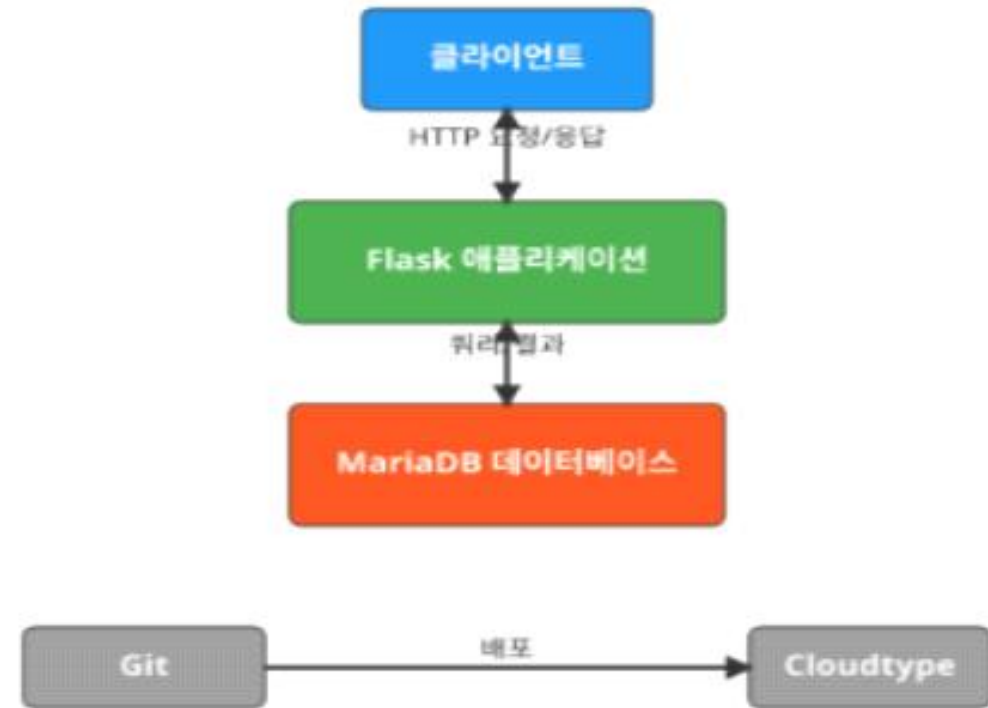
📁 프론트엔드: Jinja2, HTML/CSS, Javascript

📄 백엔드: Flask(라우팅, 로직)

🗄️ DB: MariaDB(데이터 저장)

☁️ 배포: Cloudtype

🔑 버전 관리: Git



데이터베이스 설계

region 테이블

| 필드명 | 데이터 타입 | 설명 |
|------|-------------|----------------|
| ④ id | INT | 고유 식별자 |
| name | VARCHAR(50) | 지역명 (예: 서울특별시) |
| nx | INT | 기상청 X 좌표 |
| ny | INT | 기상청 Y 좌표 |

지역별 기상청 API 좌표(nx, ny)를 저장하여 해당 지역의 날씨 정보를 조회할 수 있도록 합니다.

temclo 테이블

| 필드명 | 데이터 타입 | 설명 |
|----------|--------------|----------|
| ④ id | INT | 고유 식별자 |
| min_temp | FLOAT | 최저 온도 |
| max_temp | FLOAT | 최고 온도 |
| clothes | VARCHAR(200) | 추천 의상 설명 |

온도 범위별 추천 의상 정보를 저장하여 현재 온도에 맞는 의상을 추천 할 수 있도록 합니다.

핵심 코드 분석

요 주요 함수구현

```
def get_connection():  
    return pymysql.connect(  
        host='localhost',  
        user='root',  
        password='[REDACTED]',  
        db='test',  
        charset='utf8mb4',
```

```
@app.route("/search")  
def weather():  
    city=request.args.get('city')  
    conn=get_connection()  
    cursor=conn.cursor()  
  
    sql = "SELECT * FROM region WHERE name = %s"
```

코드 설명

데이터베이스 연결 설정

get_connection() 함수는 MySQL 데이터베이스에 연결하기 위한 설정을 제공합니다. DictCursor를 사용하여 결과를 딕셔너리 형태로 반환합니다.

의상 추천 로직

clothes(temp) 함수는 입력된 온도에 맞는 의상을 데이터베이스에서 조회합니다. 온도 범위 (min_temp, max_temp)를 기준으로 적절한 의상을 찾아 반환합니다.

날씨 API 연동

weather() 함수는 기상청 API를 호출하여 현재 날씨 정보를 가져옵니다. 지역명을 기준으로 데이터베이스에서 좌표(nx, ny)를 조회합니다.

데이터 처리 및 응답

API에서 받은 JSON 데이터를 파싱하여 온도 정보(T1H)를 추출하고, 추출된 온도를 기반으로 의상 추천 함수를 호출합니다.

시연

웹 애플리케이션 시연 과정

1 지역 선택

드롭다운 메뉴에서 원하는 지역을 선택합니다.

2 검색 버튼 클릭

'검색' 버튼을 클릭하여 날씨 정보를 요청합니다.

3 API 호출 및 데이터 처리

백엔드에서 기상청 API를 호출하고 데이터베이스에서 의상 정보를 조회합니다.

4 결과 확인

현재 온도와 추천 의상 정보가 화면에 표시됩니다.

날씨&추천

인천광역시

검색

현재는 25도 입니다. 추천 의상은 반팔, 반바지입니다!

시연 포인트

사용자 친화적인 인터페이스
실시간 날씨 정보 반영 온도에
맞는 의상 추천 정확도 빠른 응답
속도

결론 및 향후 과제

✧ 프로젝트를 통해 얻은 점

클라우드 데이터베이스 활용 경험

클라우드 환경에서 MySQL 데이터베이스를 구축하고 연동하는 실무 경험 습득

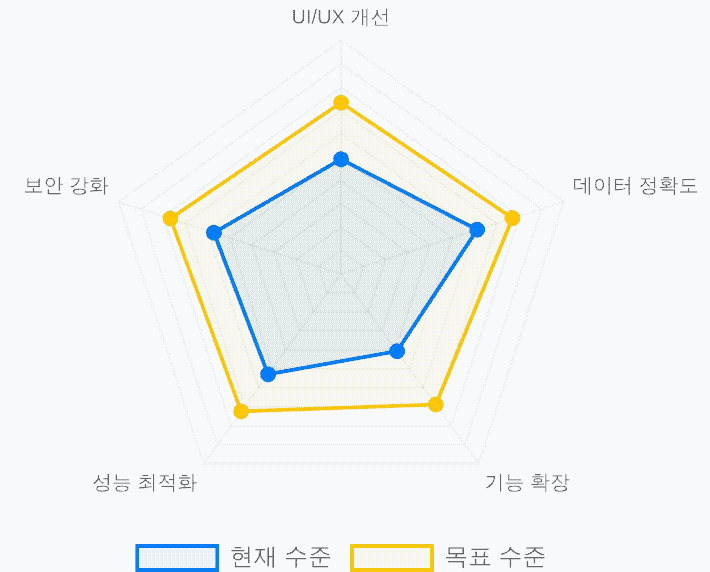
외부 API 연동 기술

기상청 공공데이터 API를 활용한 실시간 데이터 처리 및 가공 능력 향상

웹 애플리케이션 개발 역량

Flask 프레임워크를 활용한 웹 서비스 개발 및 프론트엔드-백엔드 연동 기술 습득

향후 개선 방안



사용자 맞춤형 추천 시스템

사용자 선호도와 과거 선택 데이터를 기반으로 한 개인화된 의상 추천 기능 개발

모바일 앱 개발

React Native 또는 Flutter를 활용한 크로스 플랫폼 모바일 애플리케이션 개발

팀원 소감

✧ 프로젝트를 통해 얻은 점

박정환 (PM)

"프로젝트 기획과 테스트를 담당하면서 팀원들의 다양한 아이디어를 조율하는 과정이 도전적이었습니다. Flask 프레임워크의 구조를 더 깊이 이해하게 되었습니다."

이동훈 (프론트엔드)

"Jinja2 템플릿 엔진을 활용한 프론트엔드 개발은 새로운 경험이었습니다. 백엔드와 프론트엔드의 연결 과정을 이해하며 웹 개발의 전체 흐름을 파악할 수 있었습니다."

김도윤 (백엔드)

"API에서 받아온 데이터를 뷰에 표시하는 과정과 DB 설계에서 많은 것을 배웠습니다. 특히 Flask에서 데이터를 처리하고 템플릿에 전달하는 방식에 대한 이해도가 크게 향상되었습니다."

강성대 (기술총괄)

"외부 서버 연결과 Cloudtype을 통한 배포 과정에서 많은 기술적 도전이 있었지만, 이를 통해 실제 서비스 배포 경험을 쌓을 수 있었습니다."

김유진 (백엔드)

"카카오맵 API를 활용하여 지역 데이터를 가져오고 이를 DB와 연결하는 과정이 복잡했지만, 실제 데이터 파이프라인 구축 경험을 통해 데이터 처리 능력을 향상시킬 수 있었습니다."

질문과 답변



 프로젝트 링크: https://github.com/jpminlak/AI_bracket2