

Runnnnnner



러닝 코스 추천 사이트

2024-1 AI융합캡스톤디자인과 창업 중간과제 | 20191138 이현규

프로젝트 동기 !!

러닝 코스에 대한 정보 부족 😭

- 학교 근처에는 다양한 러닝 코스가 존재하지만, 이에 대한 정보는 부족
- 전용 러닝 앱을 사용하거나, 러닝 모임에 들어가야 코스 정보를 알 수 있음
- 러닝을 하기 위해선, 길이, 난이도, 급수대/편의점/화장실 존재 등의 정보 필요
- 별도의 앱 설치 없이, 사용자의 현재 위치를 기반으로 다양한 길이의 러닝 코스를 추천하는 사이트 개발
- 사이트 기획, 개발, 배포 전 과정에 있어 GPT의 도움을 받음으로써 GPT 응용력 개발

데이터셋

문화빅데이터플랫폼 - 내 주변 산책로 데이터 (한국문화정보원)

- 국토교통부의 정보를 기반으로 한 산책로 데이터
- 코스명, 설명, 길이, 예상 소요시간, 편의시설, 위도, 경도 등 포함
- 사이트에서는 이름, 설명, 위치(경도/위도), 총 거리, 편의시설, 화장실 유무 등 사용
- https://www.bigdata-culture.kr/bigdata/user/data_market/detail.do?id=9d4e73e0-41e6-11eb-af9a-4b03f0a582d6

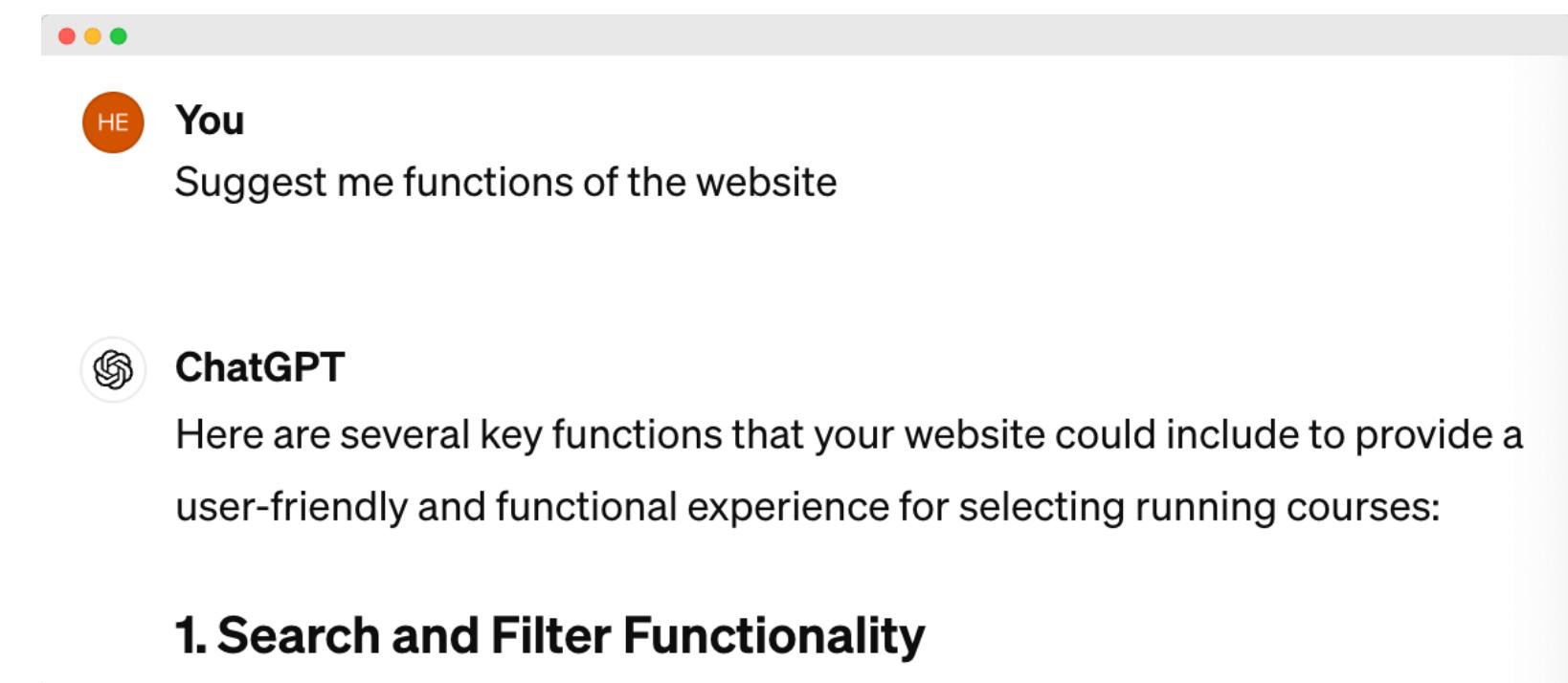
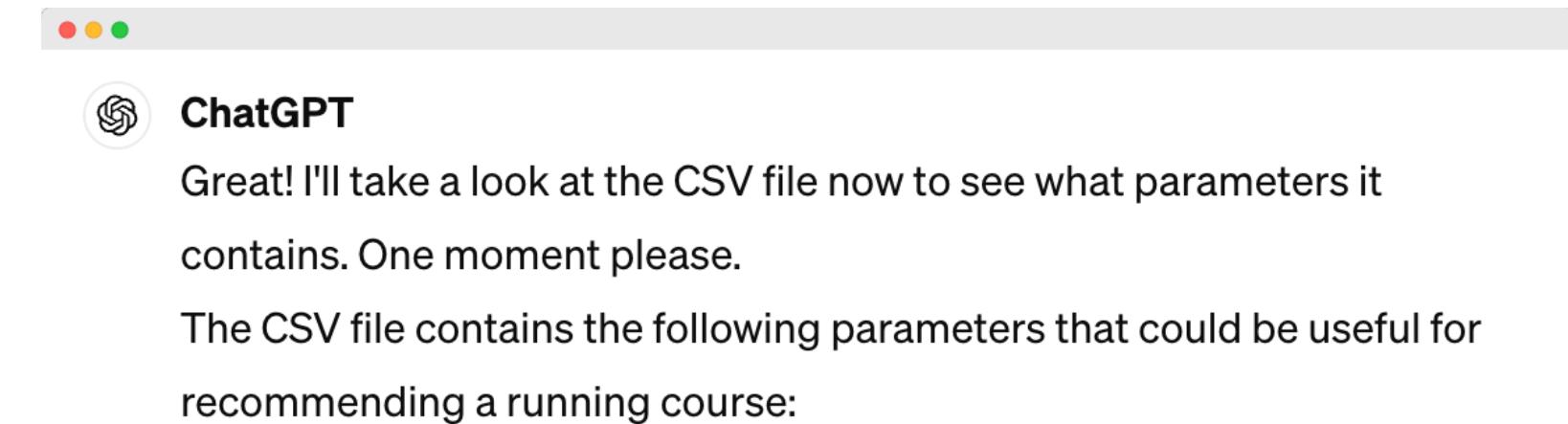
ESNTL_ID	WLK_COURS_FLAG_NM	WLK_COURS_NM	COURS_DC
KCCWSPO20N0000000001	남산 녹색 둘레길	남산 녹색 둘레길	지천생태길 ~ 녹색길 ~ 벚꽃길 ~ 고향길 ①지천생태길(4.2km) : 생태공원⇒백세공원⇒지천뚝방길⇒적누리 ↗
KCCWSPO20N0000000002	거북이 마을 솔바람길	01코스	전통체험관~보살바위~말바위~자라바위~전용석고택~호랑이가 잡아준묘~사랑바위~전통체험관(2.1km)
KCCWSPO20N0000000003	거북이 마을 솔바람길	02코스	자라바위~북방성황당~범바위~할매바위~삼형제바위~호랑이굴바위~산제바위~사랑바위(2.6km)
KCCWSPO20N0000000004	거북이 마을 솔바람길	03코스	삼형제바위 ~ 곰보바위 ~ 감투봉 ~ 전통체험관(2.8km)
KCCWSPO20N0000000005	내포문화숲길	내포 역사인물길	용봉산~(2.7km)고암이응노생가~(2.8km)백월산~(4.3km)남산~(2.1km)보개산~(1.8km)상지천~(6.1km)곧
KCCWSPO20N0000000006	내포문화숲길	17코스 내포 역사인물길	면천향교~(1.6km)면천읍성~(1.9km)영탑사~(2.3km)서산~(0.3km)의두암~(5.5km)승전곡~(3.6km)역천

웹 사이트 기획



ChatGPT를 활용한 기능 추출 및 기획

- ChatGPT를 활용한 데이터셋 분석
 - 헤더 분석을 통한 **메타데이터 추출** 💥
- 기존의 아이데이션 및 ChatGPT를 활용한 **기능 추가**
 - 내 주변 코스 추천 뿐만 아니라, 날씨 정보를 추가



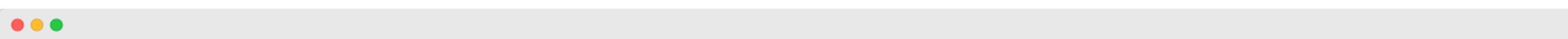
기능 리스트업 및 GPT 프롬프트 작성



- 기능 리스트
 - 위치 받아오기
 - 내 주변 코스 찾아오기
 - 5km 이하
 - 5km ~ 10km
 - 10km 이상
 - 각 코스 정보 보여주기
 - 각 코스와 나의 거리 찾기
 - 각 코스의 위치 지도로 보여주기
 - 내 주변 날씨 (강우) 보여주기
- 프롬프트
 앞서 보여준 데이터셋을 활용한 웹사이트를 제작할건데, 코드를 작성해 줘. 디자인을 위해 필요하다면 bootstrap을 사용해. 우선 **유저의 위치 정보를 제공받기** 위해서 **권한을 허가**받고, 그리고 경도값과 위도값을 사용해서 나와 **가장 가까운 코스**를 찾아줘. 가장 가까운 코드 중 **거리가 5km 이하, 5~10km, 10km 이상인 코스를 하나씩** 찾아주면 돼. 그리고 **그 코스의 이름, 나이도, 길이, 설명, 음수대 정보, 화장실 정보, 주소**를 알려줘. 나와 그 코스 시작점의 **거리를 계산**해서 결과값에 포함시켜 줘. 그 주소를 클릭하면 **Google Maps**로 연결되게 해줘. 주소가 어렵다면 위도와 경도로 길을 안내해줘도 괜찮아. 그리고 그 아래 **내 주변 지역의 날씨를 보여줘**.

ChatGPT의 응답

Python 기반 백엔드 프레임워크인 Flask(백엔드) 및 css, Bootstrap(프론트엔드) 코드 작성



ChatGPT

Here's a basic web application using Flask, a Python web framework, along with Bootstrap for styling. This application will request the user's location, find nearby running courses from the dataset based on specified distance ranges, and integrate with Google Maps for directions. It will also include a search functionality for courses by region and display local weather conditions.

Prerequisites:

1. Install Flask, pandas, and requests library in your Python environment. You can install these using pip:

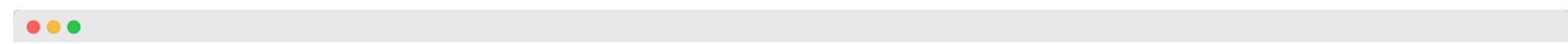
```
Copy code  
pip install Flask pandas requests
```

2. Ensure you have a basic understanding of Flask applications and have access to Google Maps API for embedding maps.

Flask Application Code

Here is the Python code for your Flask application. Save this as `app.py`:

```
python  
  
from flask import Flask, request, render_template  
import pandas as pd
```



HTML Templates

You'll need to create HTML templates for rendering the views. Use Bootstrap for styling. Here are basic templates:

1. **index.html** - Home page with request for location permission.
2. **results.html** - Display the closest courses and weather.
3. **search_results.html** - Display courses filtered by region.

For `index.html`:

```
html  
  
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Running Courses</title>  
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">  
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>  
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>  
</head>  
<body>  
    <div class="container mt-5">  
        <h1>Find Nearby Running Courses</h1>  
        <p>Allow location access to find courses near you.</p>  
        <button onclick="getLocation()">Get My Location</button>  
    </div>
```

로컬 개발 및 재생성



1. 위치 정보를 받아오는 과정에서의 오류

A screenshot of a web browser window. On the left, there's a user message from 'Anonymous' which reads: 'I got the page to work, and I pressed get my location, and it asked for permission, but nothing happens'. On the right, there's a response from 'ChatGPT' which reads: 'If pressing the "Get My Location" button prompts for permission but then nothing happens, there could be a few different issues to troubleshoot in your Flask application and JavaScript handling. Here are some steps and considerations to help resolve the issue:

- ChatGPT에게 코드와 오류를 입력하니 스스로의 문법 오류와 잘못된 코드를 수정
- 오류는 발생하지 않으나 추가 동작 하지 않는 오류 역시 콘솔, 터미널의 상태를 입력하니 해결책 제시

1. Check JavaScript Console for Errors

Open the browser's developer tools (usually by pressing F12) and go to the console tab. Check if there are any JavaScript errors reported when you click the button. Common errors could be related to:

- JavaScript syntax errors.
- Issues with making the POST request.
- Problems with the function not being defined properly.

2. Verify the POST Request Handling

Ensure that the JavaScript function making the POST request is correctly defined and executed. The function should look something like this:

로컬 개발 및 재생성

2. 날씨 및 지도 서비스 API 발급 ☀️

- ChatGPT가 준 코드의 Placeholder 대신 OpenWeather 및 Google Maps API 발급 ✓

The screenshot shows the OpenWeather API keys management interface. At the top, there's a navigation bar with the OpenWeather logo, a search bar, and links for Guide, API, Dashboard, Marketplace, and Pricing. Below the navigation, there are tabs for New Products, Services, API keys (which is highlighted in red), Billing plans, Payments, and Block logs. A message box states: "You can generate as many API keys as needed for your subscription. We accumulate the total". The main table lists one API key:

Key	Name	Status	Actions
c7b[REDACTED]	Default	Active	

The screenshot shows the Google Cloud Platform API keys management interface. At the top, there's a navigation bar with the Google Cloud logo, a dropdown menu for project STS2013-MID, and a search bar. Below the navigation, there are sections for Google Maps Platform and API 키. The API 키 section shows a table with one entry:

이름	생성일	제한사
⚠️ Maps API Key	2024. 4. 28.	없음

Below the API 키 section, there's an OAuth 2.0 클라이언트 ID section with a table:

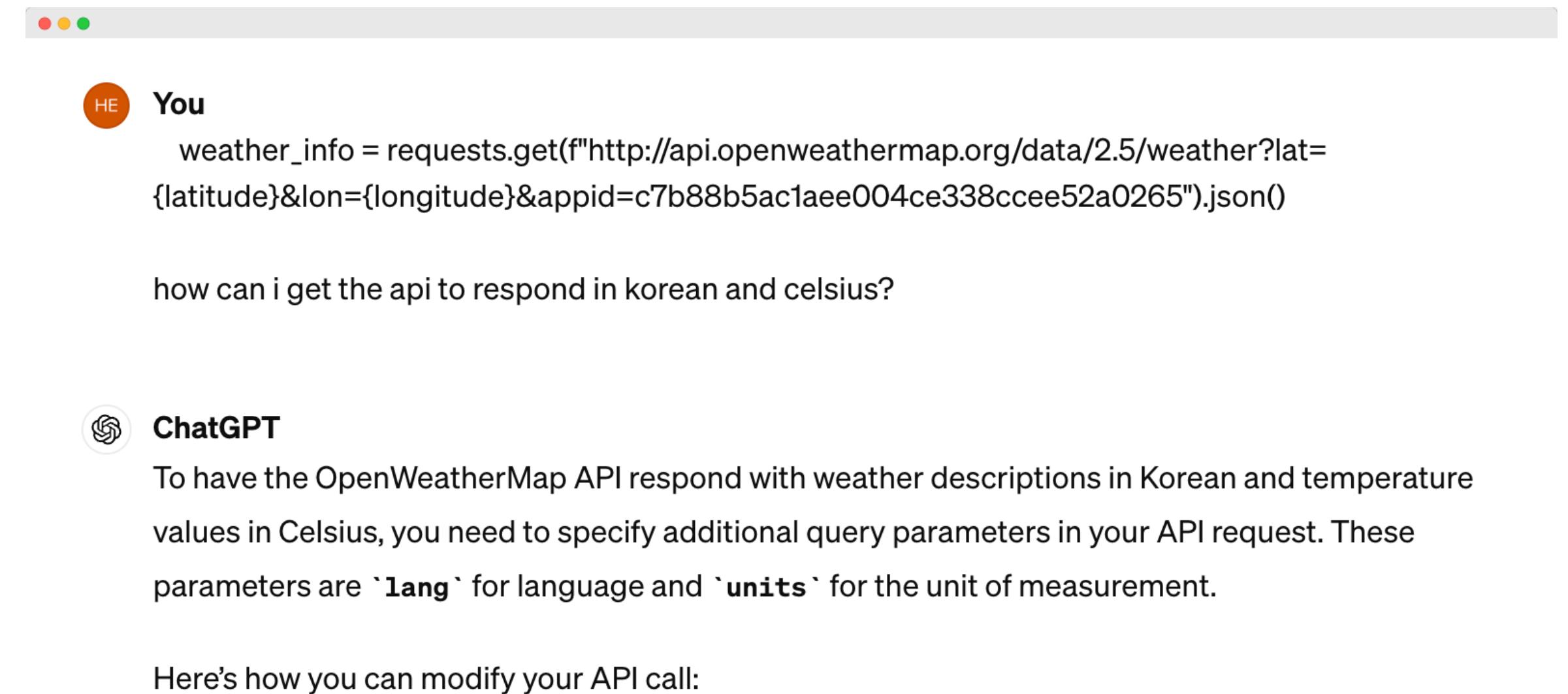
이름	생성일	유형
[REDACTED]	[REDACTED]	[REDACTED]

```
{"coord":{"lon":127.2131,"lat":0.2343}, "weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04d"}], "base":"stations", "main":{"temp":302.79,"feels_like":306.89,"temp_min":302.79,"temp_max":302.79,"pressure":1006,"humidity":68,"sea_level":1006,"grnd_level":1006}, "visibility":10000, "wind":{"speed":0.68,"deg":325,"gust":0.65}, "clouds":{"all":99}, "dt":1714291531, "sys":{"country":"ID", "sunrise":1714253091, "sunset":1714296730}, "timezone":32400, "id":1958070, "name":"North Maluku", "cod":200}
```

- API 정상 리턴 확인
temp: 302.79,
description: overcast clouds

로컬 개발 및 재생성

3. BootStrap을 활용한 디자인 개선 및 로케일 적용



```
{"coord": {"lon": 127.2131, "lat": 0.2343}, "weather": [{"id": 804, "main": "Clouds", "description": "온흐림", "icon": "04d"}], "base": "stations", "main": {"temp": 29.64, "feels_like": 33.74, "temp_min": 29.64, "temp_max": 29.64, "pressure": 1006, "humidity": 68, "sea_level": 1006, "visibility": 10000, "wind": {"speed": 0.68, "deg": 325, "gust": 0.65}, "clouds": {"all": 99}, "dt": 1714291531, "sys": {"country": "ID", "sunrise": 1714253091, "sunset": 1714296730}, "timezone": 32400, "id": 1958070, "name": "North Maluku", "cod": 200}
```

- 섭씨, 한국어 반영 완료 (temp: 29.64, description: 온흐림)

사이트 업로드



🔗 AWS EC2를 활용한 사이트 호스팅

- Github은 무료이고 index.html만 완성하면 곧바로 호스팅이 가능하지만, 정적 웹(컨텐츠가 변화하지 않는 앱)만 호스팅 가능
- 사용자의 위치를 받아온 후, 보여줄 컨텐츠를 계산해서 보여줘야 하는 사이트에는 사용할 수 없음
- AWS(Amazon Web Services)의 EC2 가상 서버를 활용하여 사이트 호스팅
- VSCode을 Github Repository에 연결해 작업 후, EC2 서버에 SSH로 연결, git pull한 후 곧바로 코드 동작

프로젝트 결과물



<https://3.34.193.55:5000/>



가장 가까운 거리별 러닝 코스

현재 날씨: 28.16°C의 체감 온도로 온흐림

5km 이하

A+B코스

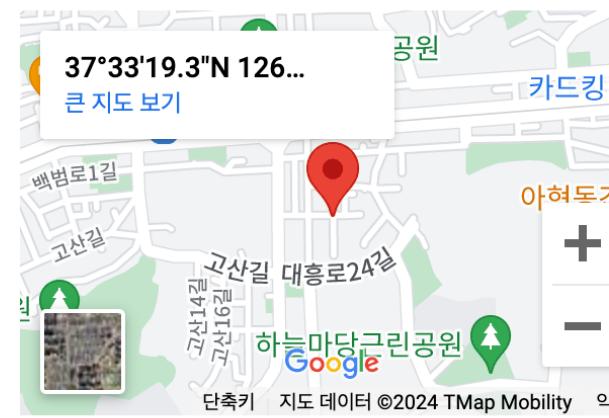
난이도: 매우쉬움
나로부터 거리: 1.28km

설명: A구간 : 가로등 따라 1-3-4-8-13-17-62-68 (1.1km 25분 소요) B구간 : 가로등 따라 59-55-46-38-32-29-19 (0.6km 15분 소요)

길이: 1.7 km

식수: 본인이 구비

화장실: 소금나루 무더위쉼터, 이대역



5~10km

신촌 대학탐방 나들길

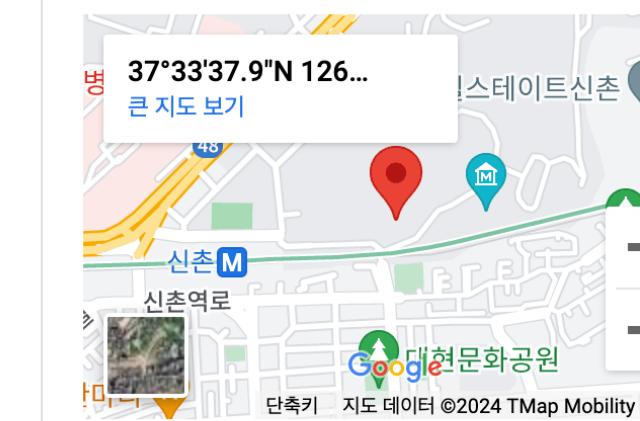
난이도: 보통
나로부터 거리: 1.86km

설명: 서강대정문~노고산~이대정문~연세대동문

길이: 5.12 km

식수: nan

화장실: nan



10km 이상

탕춘대성숲 나들길

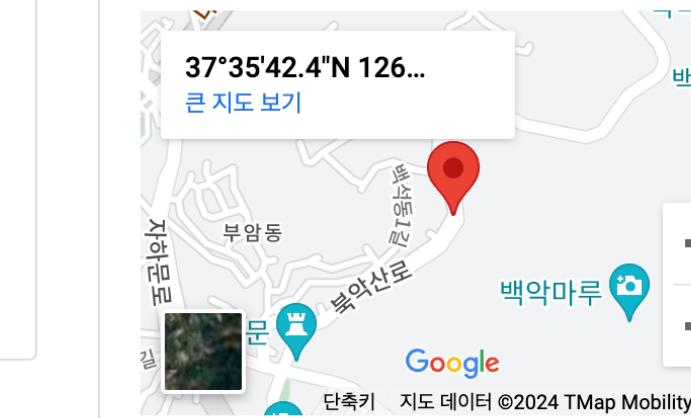
난이도: 어려움
나로부터 거리: 6.01km

설명: 흥제역 ~ 인왕중학교 ~ 인왕산 지키미 초소 ~ 약수입구 ~ 기차바위 ~인왕산~ 창의문~백사실 계곡~ 세검정 ~ 탕춘대성~흥제역

길이: 12.26 km

식수: 사전 준비 필요

화장실: 개미마을 언덕 꼭대기



전 세계 어디서나 접속해 주변 러닝 코스를 볼 수 있는 사이트

아이폰 기준, [세부사항 보기](#) → [이 웹사이트에 방문](#) → [웹사이트 방문](#) 클릭해야 접속 가능

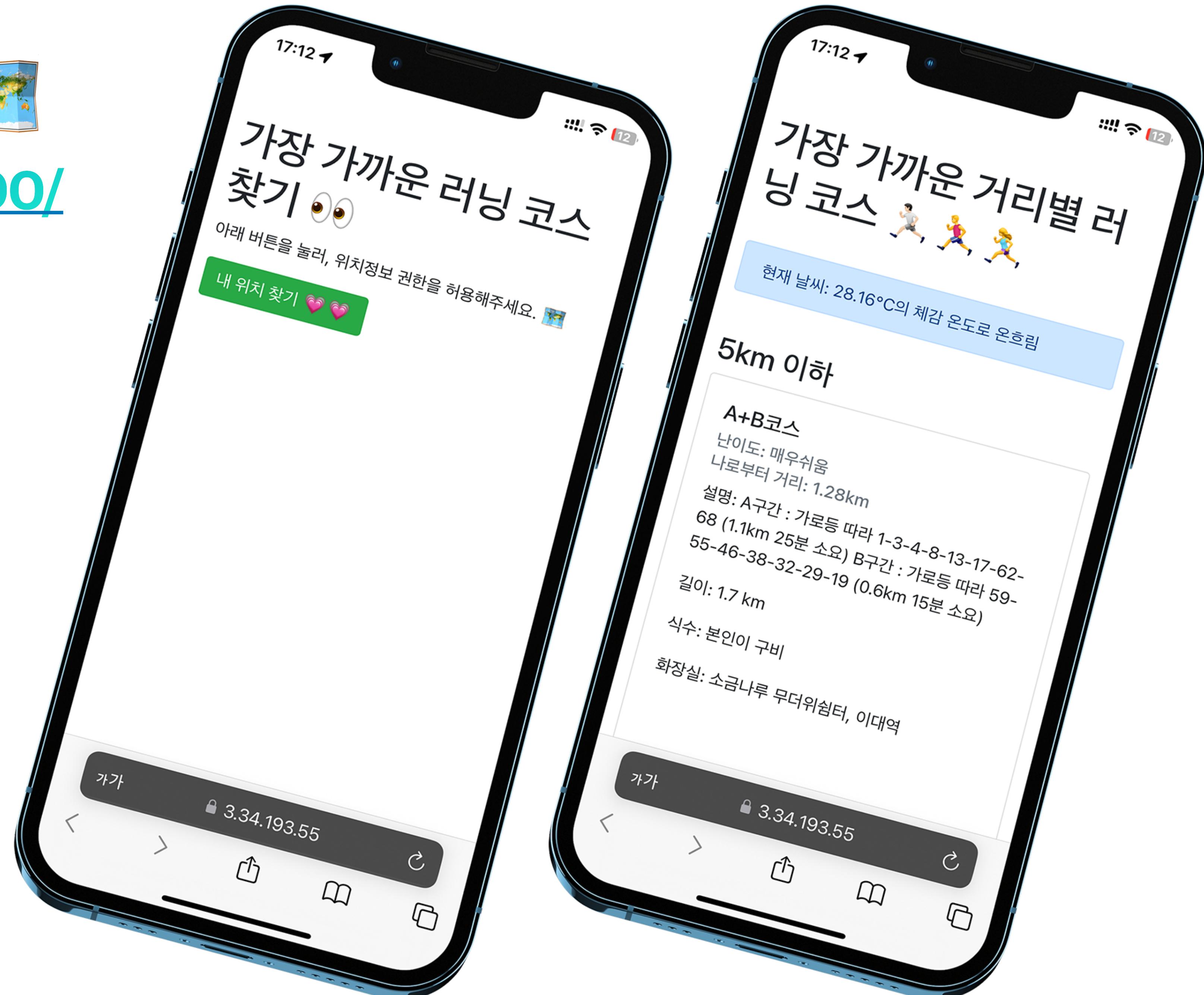
프로젝트 결과물



<https://3.34.193.55:5000/>

모바일 환경에 최적화

- 사이트 접속 후, 버튼 눌러 위치정보 권한 부여
- 위치정보로 날씨 및 가장 가까운 코스 정보 반환
- 가장 가까운 코스 정보, 지도 클릭 → 시작지점으로 Google Maps 실행



아이폰 기준,
[세부사항 보기](#) → 이 웹사이트에 방문 →
웹사이트 방문 클릭해야 접속 가능

프로젝트 의의 및 한계



- 서버 호스팅, API 발급 및 디자인을 제외한 모든 개발 과정에 ChatGPT를 활용
 - 특히 외부 라이브러리인 Bootstrap, Flask를 GPT가 활용할 수 있다는 점에서 놀라움
 - ChatGPT Plus는 이미지 분석도 지원하는 만큼, 원하는 디자인을 직접 그려 ChatGPT로 쉽고 빠르게 프론트엔드 코드로 변환할 수 있을 것으로 보임
- EC2 서버로 사이트를 배포함으로써 전 세계 누구나 사용할 수 있는 서비스 완성
 - 기말 프로젝트에서 SNS 및 모델 간 서버 구축시 ChatGPT 및 서버를 어떤 구조로 활용할 지 고려할 수 있었음
- 개선점
 - 위치를 제외한 요소를 사용한 추천이나 검색 기능 도입
 - 한국어 이외의 언어에도 대응(데이터셋 역시 번역하거나 번역기능을 도입해야 함)
 - 한국 이외의 산책 코스도 도입