

09 PJT

Vue를 활용한 SPA 구성

INDEX

- 개요
- 준비사항
- 프로젝트 안내
- 요구사항
- 도전 과제
- 제출

개요

프로젝트 개요

- Vue를 활용하는 SPA 구성
- TMDB API에서 제공하는 영화 데이터를 활용한 영화 정보 제공 애플리케이션 완성
- Youtube API에서 제공하는 검색 기능을 활용 가능한 애플리케이션 완성

프로젝트 목표

- 영화 정보를 제공하는 SPA 제작
- AJAX와 JSON에 대한 이해
- 외부 API와의 요청과 응답을 통한 데이터 수집
- Vite Frontend Tooling에 대한 이해

준비사항

| 개발도구

- Visual Studio Code
- Google Chrome
- Node.js LTS
- Vue 3

| 사전 준비 - API 키 발급

- 공용 문서 (<https://abit.ly/pb-document>)
 - TMDB API key 발급 가이드
 - Youtube API key 발급 가이드

| 사용 API

- TMDB API v3
 - <https://developer.themoviedb.org/reference/movie-details>
(개별 영화 세부정보)
 - <https://developer.themoviedb.org/reference/movie-top-rated-list>
(평점별로 정렬된 영화 목록)
- Youtube API
 - <https://www.googleapis.com/youtube/v3/search>

프로젝트 안내

프로젝트

- 프로젝트 이름
 - my-vue-pj t

컴포넌트 구조



router 및 관련 컴포넌트

Path	Component	Description
/	HomeView	메인 페이지
/movies	MovieListView	전체 영화 목록 페이지
/:movieId	MovieDetailView	영화 세부정보 페이지
/review-search	ReviewSearchView	유튜브 검색 및 출력 페이지
/recommended	RecommendedView	추천 영화 출력 페이지 (도전 과제)

요구사항

| 공통 요구사항

- 명시된 요구사항 이외에는 자유롭게 작성해도 무관
- Bootstrap을 이용하여 자유롭게 스타일링 가능
- **.gitignore** 파일을 추가하여 불필요한 파일 및 폴더는 제출하지 않음
- **.env** 파일을 사용해 API Key를 비롯한 중요한 정보는 제출하지 않음
 - <https://ko.vitejs.dev/guide/env-and-mode.html>

.env 사용 예시

```
.env
```

```
VITE_TMDB_API_KEY=eyJhQ...iOIwZT
```

```
// Vue Component
```

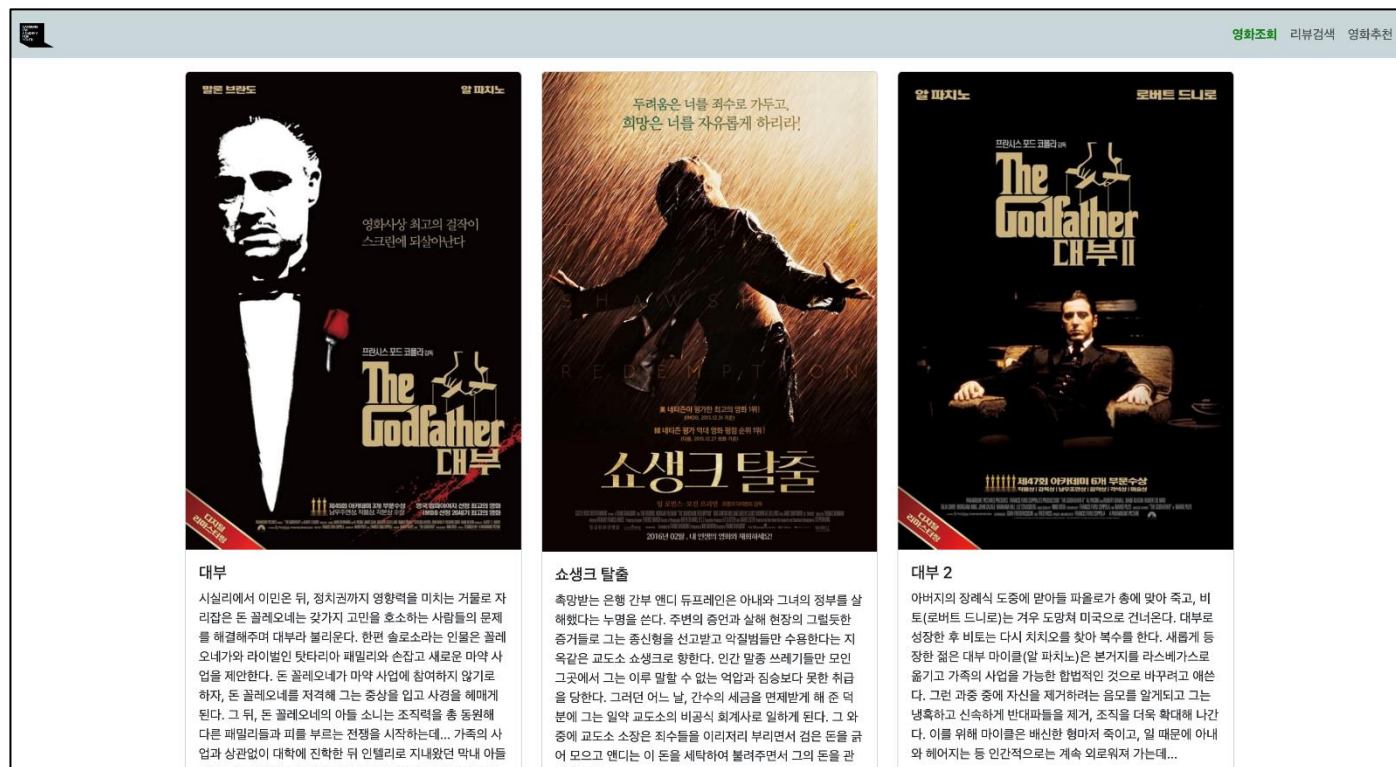
```
const API_KEY = import.meta.env.VITE_TMDB_API_KEY
```

| 필수 요구사항

- A. 최고 평점 영화 조회
- B. 최고 평점 영화의 세부정보 조회
 - 영화 제목, 개봉일, 러닝타임, 평점, 장르, 줄거리
 - 공식 예고편 영상 조회
- C. 영화 리뷰 영상 조회
- D. [도전 과제] 현재 날씨에 따른 영화 추천하기

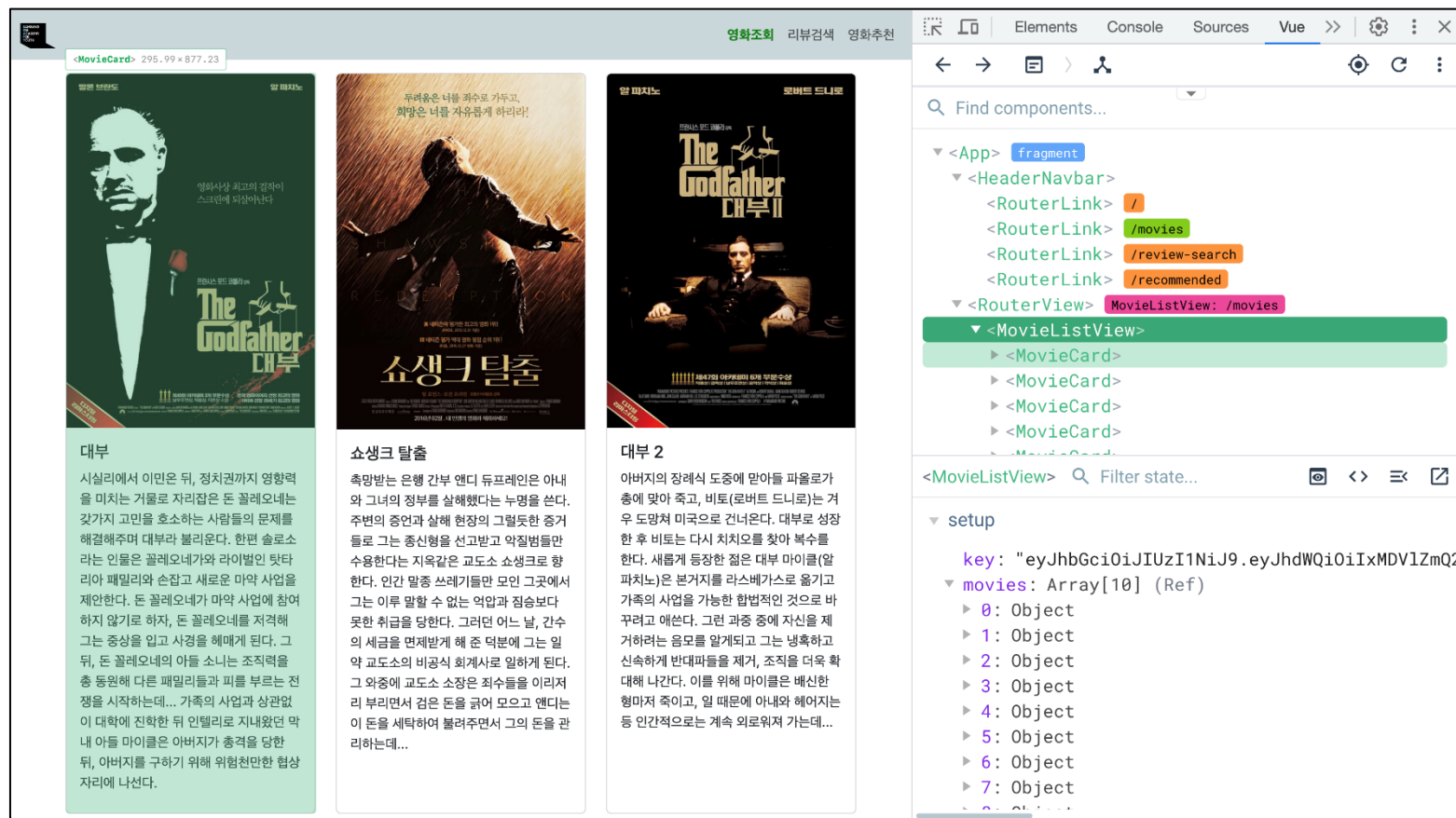
A. 최고 평점 영화 조회 (1/2)

- 내비게이션 바에서 영화 조회 링크(/movies)를 클릭하면 평점이 높은 순으로 정렬된 영화 목록을 출력 (TMDB API - MOVIES LISTS - Top Rated 활용)



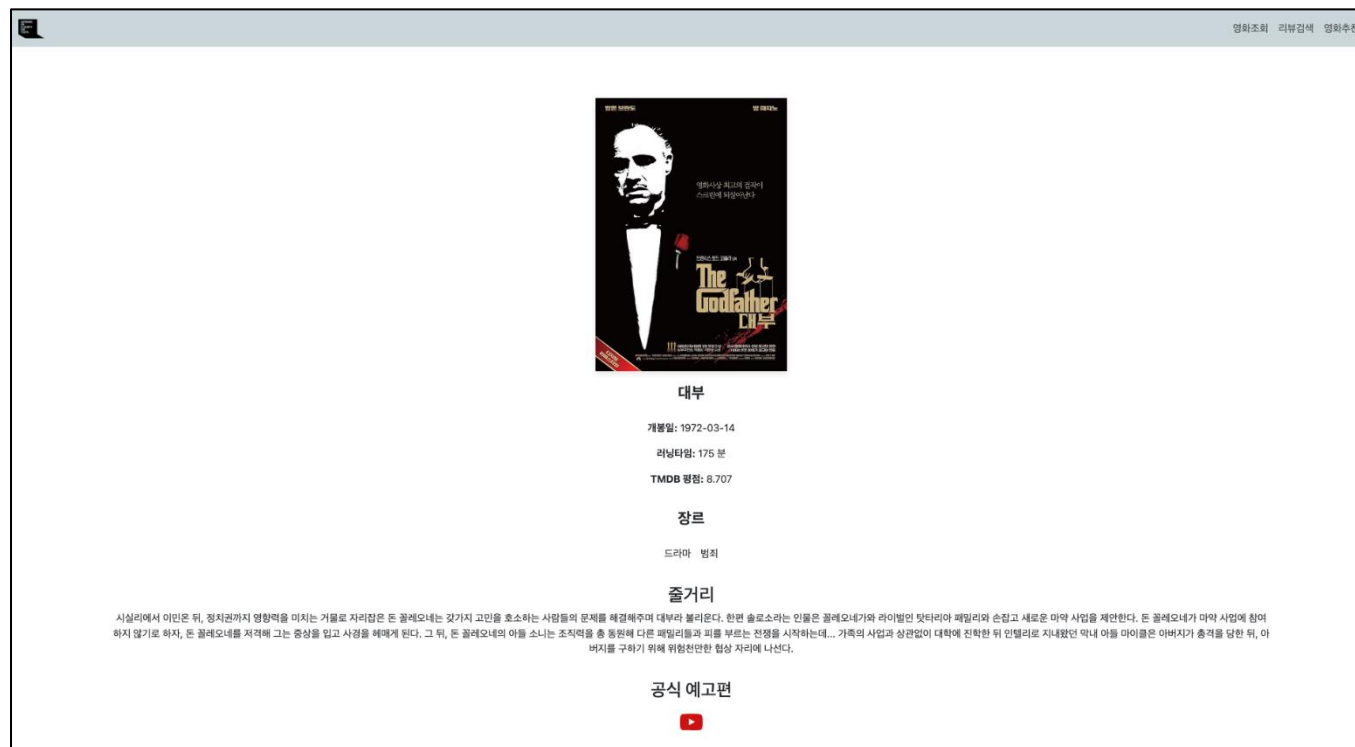
A. 최고 평점 영화 조회 (2/2)

- 컴포넌트 구조



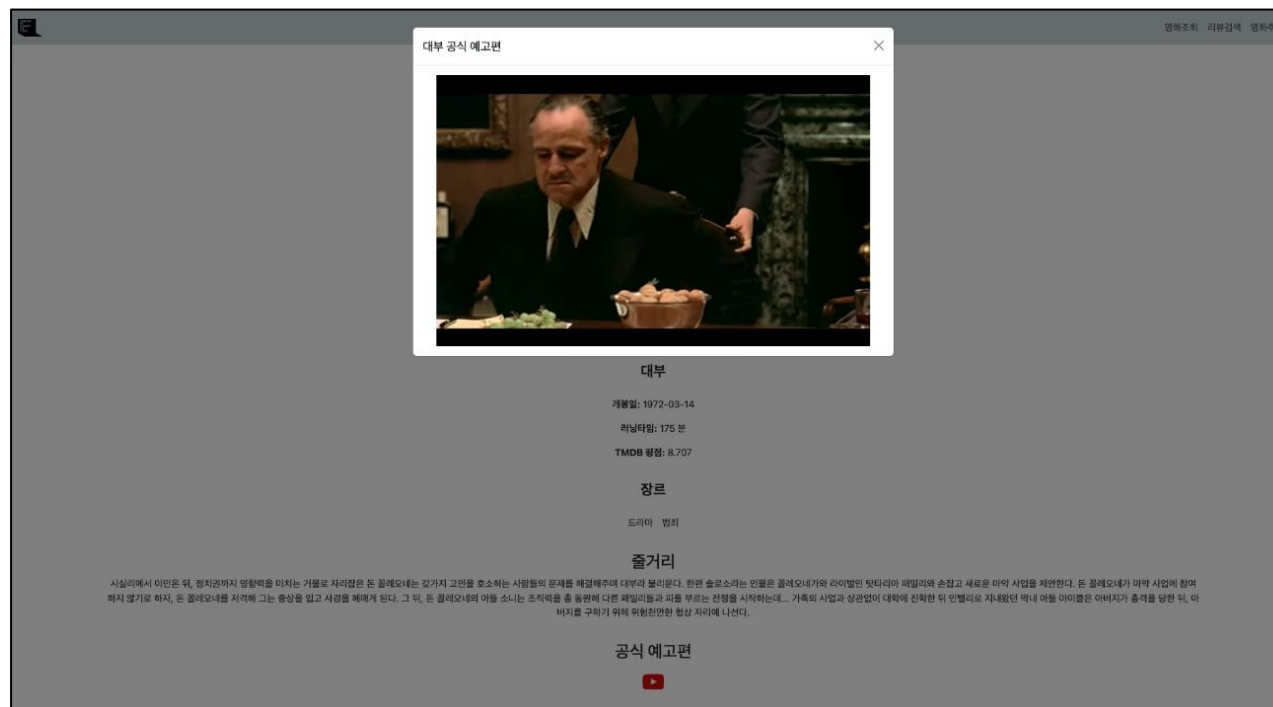
B. 최고 평점 영화의 세부정보 조회 (1/2)

- MovieCard를 클릭하면 RouterLink를 통해 MovieDetailView(/:movieId)로 이동 후 MovieDetailInfo를 출력 (TMDB API - MOVIES - Details 활용)



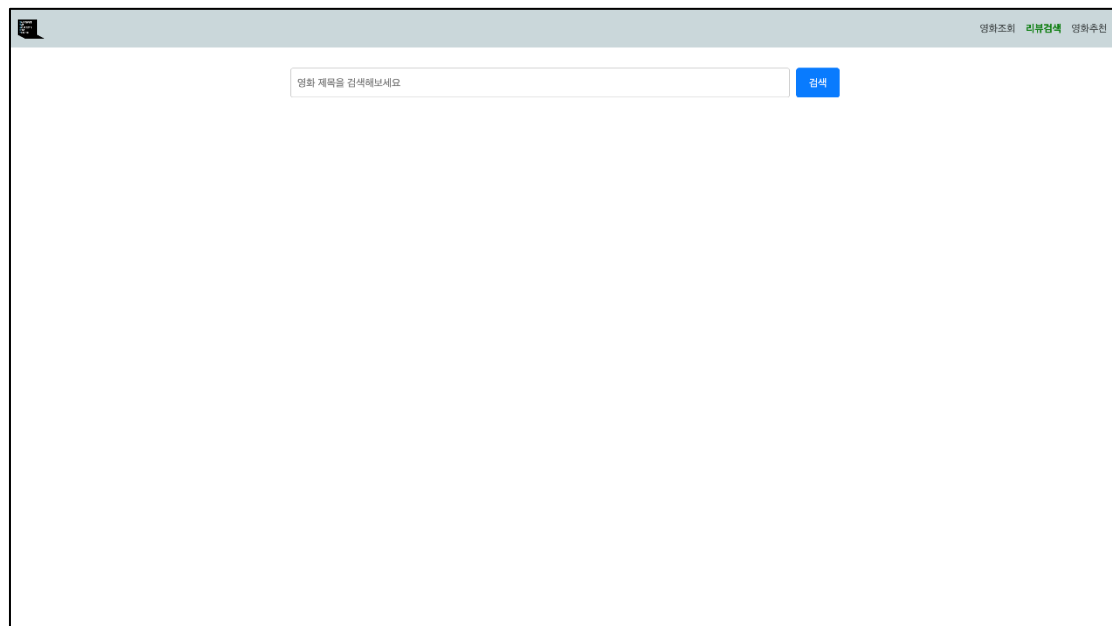
B. 최고 평점 영화의 세부정보 조회 (2/2)

- 세부정보에서 유튜브 버튼을 클릭하면 영화 제목 데이터를 활용해 해당 영화의 공식 예고편을 YoutubeTrailerModal에서 출력 (Youtube API 활용)



C. 영화 리뷰 영상 조회 (1/2)

- 내비게이션 바에서 리뷰검색(/review-search)을 클릭 후 검색어를 입력하면
검색한 영화에 대한 리뷰 영상 목록을 각 YoutubeCard에 출력 (Youtube API 활용)



검색 전



검색 후

C. 영화 리뷰 영상 조회 (2/2)

- YoutubeCard를 클릭하면 YoutubeReviewModal에서 해당 영상을 출력



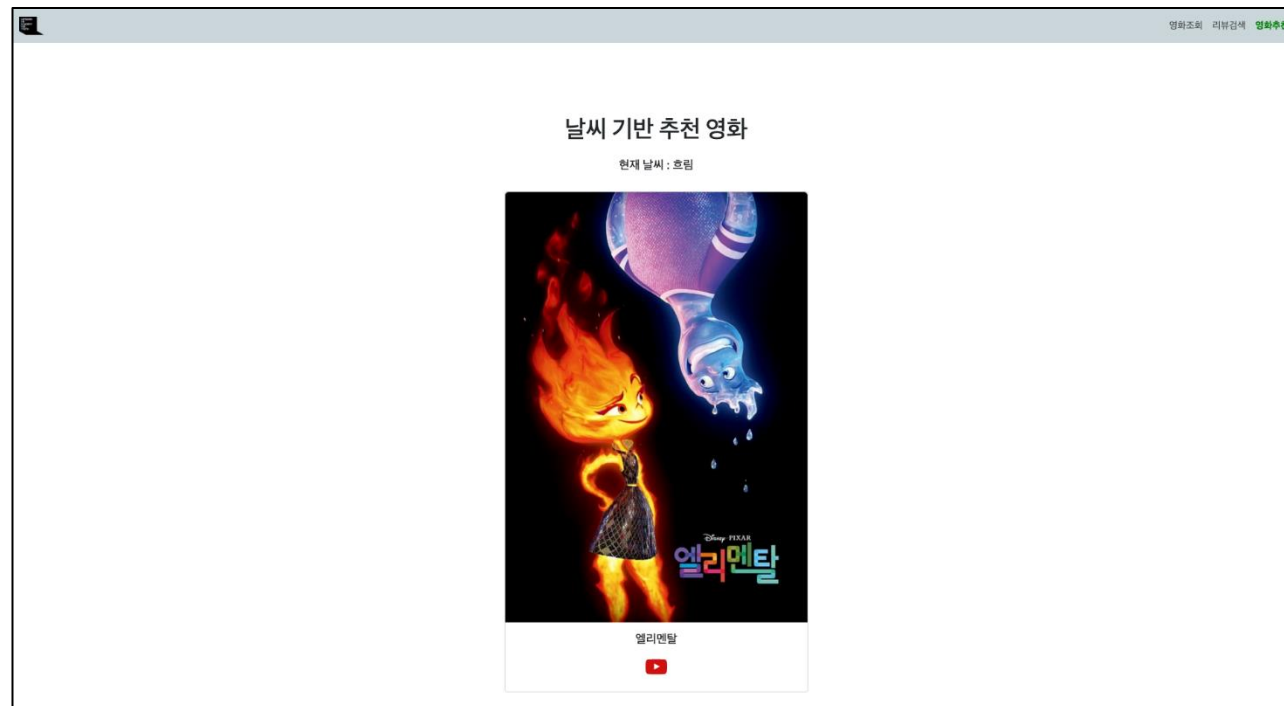
도전 과제

도전 과제 목록

- A. 현재 날씨에 따른 영화 추천하기
- B. Pinia를 도입하여 데이터 흐름 구성 변경하기

A. 현재 날씨에 따른 영화 추천하기 (1/3)

- RecommendedView에서 OpenWeather API로 부터 받아온 대한민국의 현재 날씨 정보와 영화 데이터의 특정 정보(ex. 장르)를 사용해 영화를 랜덤으로 출력



A. 현재 날씨에 따른 영화 추천하기 (2/3)

- OpenWeather API 요청 예시

Overview

GET https://api.openweathre

+

...

https://api.openweathermap.org/data/2.5/weather?q=Seoul,kor&APPID=MY-API-KEY

GET

▼

https://api.openweathermap.org/data/2.5/weather?q=Seoul,kor&APPID=MY-API-KEY

Params ●

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	q	Seoul,kor
<input checked="" type="checkbox"/>	APPID	MY-API-KEY
	Key	Value

A. 현재 날씨에 따른 영화 추천하기 (3/3)

- OpenWeather API 응답 예시

```

1  {
2    "coord": {
3      "lon": 126.9778,
4      "lat": 37.5683
5    },
6    "weather": [
7      {
8        "id": 701,
9        "main": "Mist",
10       "description": "mist",
11       "icon": "50n"
12     }
13   ],
14   "base": "stations",
15   "main": {
16     "temp": 281.18,
17     "feels_like": 281.18,
18     "temp_min": 279.81,
19     "temp_max": 282.84,
20     "pressure": 1025,
21     "humidity": 88
22   },
23   "visibility": 4000,
24   "wind": {
25     "speed": 1.03
  
```

B. Pinia를 도입하여 데이터 흐름 구성 변경하기

- 먼저 현재 프로젝트에 Pinia 도입이 필요한지 파악하기
- 도입했을 경우 데이터를 어떻게 재구성해야 할지 설계 후 적용하기

제출

제출 시 주의사항

- 제출기한은 금일 18시까지 입니다. 제출기한을 지켜 주시기 바랍니다.
- 반드시 **README.md** 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 등을 상세히 기록하여 제출합니다.
 - 단순히 완성된 코드만을 나열하지 않습니다.
- 위에 명시된 요구사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- <https://lab.ssafy.com/>에 프로젝트를 생성하고 제출합니다.
 - 프로젝트 이름은 ‘프로젝트 번호 + pjt’로 지정합니다. (ex. **01-pjt**)
- 반드시 각 반 담당 강사님을 Maintainer로 설정해야 합니다.