

[프로젝트 중간 과제평가]

1. 과제 설명

[React기반 날씨 데이터 연동 어플리케이션 구현]

이 프로젝트 평가는 React에서의 이벤트 처리, useEffect 사용, 그리고 서버 데이터 핸들링과 관련된 기본 개념들을 다루고 있습니다. 수행 시 주어진 요구 사항과 평가 지표를 잘 따라 문제를 해결해 보세요.

React 애플리케이션에서 대한 다음 요구 사항을 구현해야 합니다.

이 애플리케이션은 사용자로부터 이름을 입력 받고, 서버로부터 얻은 날씨 데이터를 화면에 출력합니다.

2. 사전정보

- (1) React 컴포넌트에 대한 이해가 필요합니다.
- (2) React의 state와 이벤트 처리 방법에 대한 이해가 필요합니다.
- (3) useEffect hook의 사용법에 대한 이해가 필요합니다.
- (4) 서버로부터 데이터를 받아오기 위한 HTTP 요청에 대한 기본적인 이해가 필요합니다.
- (5) fetch() 함수를 이용하여 GET 방식으로 API 서비스 사용법에 대한 이해가 필요합니다.

3. 요구사항

- (1) 리액트 App 컴포넌트에서 버튼을 하나 만듭니다.
(컴포넌트 버튼 구성된 웹브라우저 화면 캡처 - 평가지표1)
- (2) 버튼을 클릭하면, 서버("https://openweathermap.org/current")로부터 날씨 데이터를 비동기적으로 받아와서 콘솔에 출력합니다. (하단 이미지 참조)
(날씨 데이터 출력된 웹브라우저 화면 캡처 - 평가지표2)

[API 연동 참고자료]

API call

```
https://api.openweathermap.org/data/2.5/weather?q={city  
name}&appid={API key}
```



- (3) useEffect를 사용하여 컴포넌트 마운트 시 날씨 데이터를 콘솔로 출력합니다.
(useEffect 소스코드 입력 또는 소스코드 화면 캡처 - 평가지표3)
- (4) useState를 사용하여 상태 관리를 구현합니다.
(useState 소스코드 입력 또는 소스코드 화면 캡처 - 평가지표4)

4. 평가지표_(70점이상 Pass)

[1단계] 리액트 기본문법을 이해하고 데이터 연동 구현

- (1) 리액트 App 컴포넌트에서 버튼을 만들었는가? (30점)
- (2) 서버로부터 데이터를 정상적으로 받아오고 화면에 출력하는가? (40점)

[2단계] 데이터 상태관리 구현

- (3) useEffect를 사용하여 마운트 시점의 로직을 정상적으로 구현하는가? (15점)
- (4) 상태 관리를 위해 useState를 올바르게 사용하는가? (15점)

5. 참고 강의 안내

- 1) 강의차시: 1~8차시(특히 6~8차시)
- 2) React에서의 이벤트 처리, useEffect 사용, 서버 데이터 핸들링
 - 기능 미구현시 이벤트 처리, Hooks 사용법, 비동기 처리 파트 추가 학습 피드백

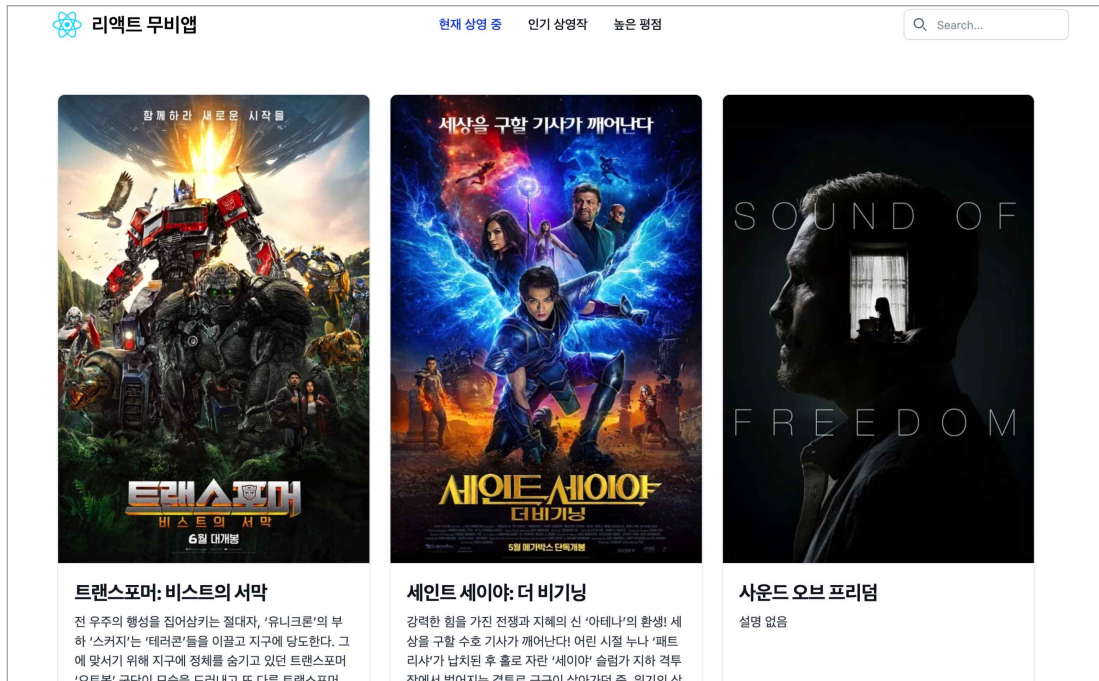
2. 프로젝트 최종 과제평가

1. 과제설명

이 과제는 React 또는 Vue를 활용하여 영화목록을 조회 및 검색하는 앱을 제작하는 것을 목표로 합니다.

주어진 요구 사항과 평가 지표에 따라 영화 정보를 효율적으로 검색하고 조회하는 기능 구현의 결과물을 제출해 주시기 바랍니다.

[구현 결과화면]



2. 사전정보

- 17~21차시 진행되는 프로젝트 강의를 참조하여 작성 하세요.
- 사용 API: TMDB (<https://www.themoviedb.org/?language=ko>)
- 3가지 API 항목중 현재상영중('Now Playing') 항목만 구현합니다.

3. 요구사항

- 프로젝트는 Vite를 활용하여 스캐폴딩 해야 합니다.
- 앱 내에 최소 5개 이상의 컴포넌트를 작성해야 합니다.
- 사용자는 앱을 통해 영화 목록을 조회하거나 검색할 수 있어야 합니다.
- 앱의 재사용성을 높이기 위해 Custom Hooks (React에서) 및 Composables (Vue에서)을 활용해야 합니다.

4. 제출 내용 (일부분만 구현하여 결과 화면이 없을시 작성된 소스코드 복사하여 제출)

- vite를 사용하여 React 또는 Vue 프로젝트가 정상적으로 생성되었는지 확인
제출자료(평가지표1): 프로젝트 생성 후 VSCode에서 디렉토리 구조 부분 화면 캡처
- 컴포넌트 작성이 잘 되었는지 확인
제출자료(평가지표2): VSCode 의 컴포넌트 디렉토리 부분 화면 캡처
- 현재 상영중 조회화면 동작 여부 확인
제출자료(평가지표3): 현재 상영중 조회 화면 캡처
(화면의 제목을 리액트 무비앱이 아닌 본인 이름을 포함 시킬 것)
- 영화제목으로 검색 시 검색결과 동작 여부 확인
제출자료(평가지표4): "korea" 로 조회하여 결과화면 캡처

5. 평가지표 (70점 이상 Pass)

[1단계] 리액트 구조/기능/문법을 이해를 기초로 컴포넌트를 생성하고 브라우저에 출력

- (1) vite를 사용하여 프로젝트를 정상적으로 생성 하였는가? (30점)
- (2) 컴포넌트 작성이 잘 되었는가? (40점)

[2단계] 실제 서비스에 활용 할 수 있는 API를 연동하고 응용 기능 구현

- (3) TMDB API를 사용하여 현재 상영중인 영화조회 구현을 잘 하였는가? (15점)
- (4) 조회 기능이 잘 구현되었는가? (15점)