

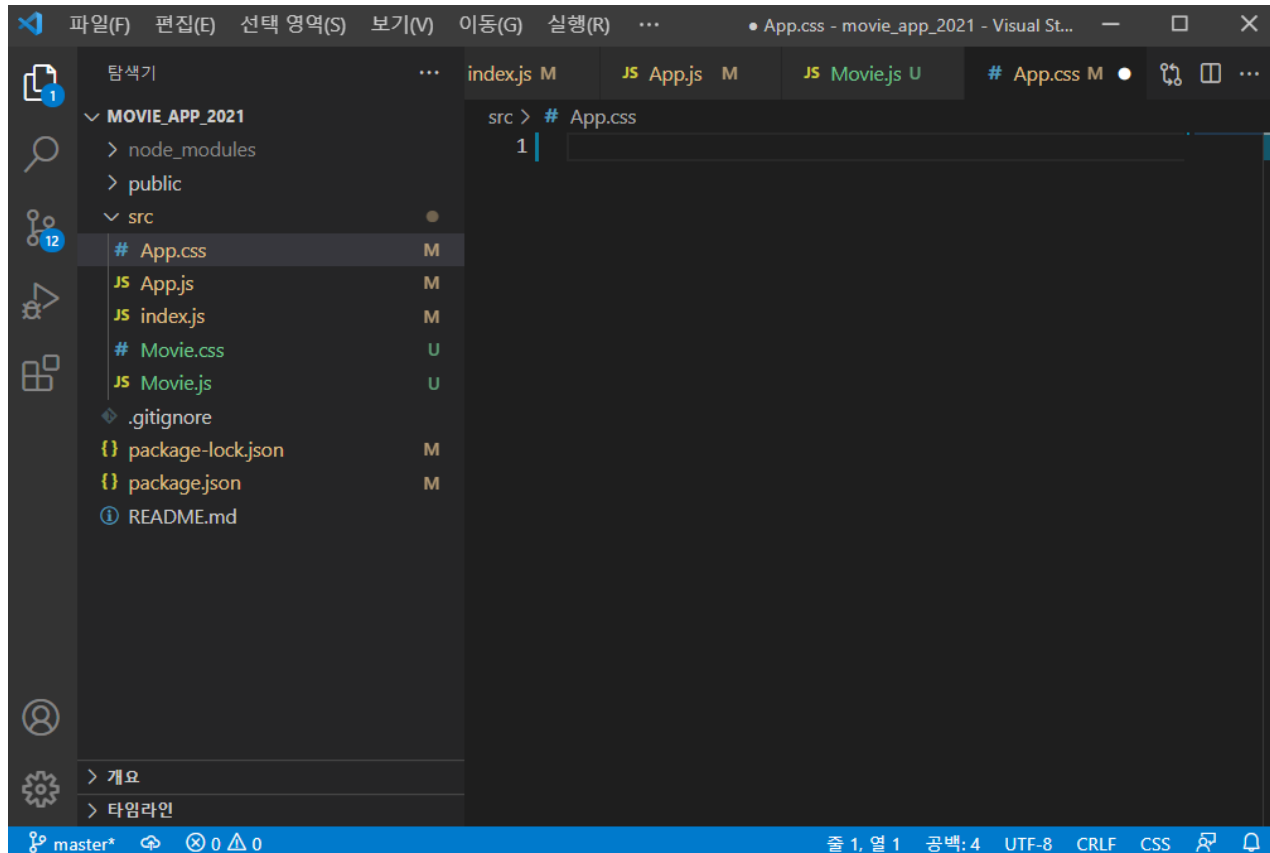
영화 앱 다듬기

1. 영화 앱 전체 모습 수정하기

1 영화 앱 전체 모습 수정하기

❖ App.css 내용 모두 지우기

- 본격적으로 영화 앱을 다듬는 작업으로 들어가기 전에 **App.css**의 모든 내용을 지우자.



1 영화 앱 전체 모습 수정하기

❖ 영화 API에서 장르 키 살펴보기

- 아직 추가하지 않은 영화 데이터는 '장르'이다.
- 다음 주소에 접속하여 영화 API에서 장르 키가 무엇인지 다시 한번 살펴보자.

https://yts-proxy.now.sh/list_movies.json?sort_by=rating

- **runtime** 키 아래에 **genres** 키가 있다.
- 이걸 영화 앱에 추가하자.

```
"movies": [  
  {  
    "id": 37384,  
    "url": "https://yts.mx/movies/jai-bhim-2021",  
    "imdb_code": "tt15097216",  
    "title": "Jai Bhim",  
    "title_english": "Jai Bhim",  
    "title_long": "Jai Bhim (2021)",  
    "slug": "jai-bhim-2021",  
    "year": 2021,  
    "rating": 9.4,  
    "runtime": 164,  
    "genres": [  
      "Crime",  
      "Drama"  
    ],  
  },  
]
```

1 영화 앱 전체 모습 수정하기

❖ Movie 컴포넌트에 genres props 넘겨주기

- App 컴포넌트에서 Movie 컴포넌트에 genres props를 넘겨준다 가정하고 Movie 컴포넌트를 수정해 보자.
- Movie 컴포넌트 인자에 genres를 추가하고, Movie.propTypes에는 genres props가 문자열 배열 arrayOf(PropTypes.string)이며 반드시 필요함(isRequired)을 추가하자.

| | |
|----|--|
| 1 | import React from "react"; |
| 2 | import propTypes from "prop-types"; |
| 3 | import './Movie.css'; |
| 4 | |
| 5 | function Movie({ title, year, summary, poster, genres }) { |
| 6 | return (|
| 7 | <div class="movie"> |
| 8 | |
| 9 | <div class="movie__data"> |
| 10 | <h3 class="movie__title">{title} </h3> |
| 11 | <h5 class="movie__year">{year}</h5> |
| 12 | <p class="movie__summary">{summary}</p> |
| 13 | </div> |

1 영화 앱 전체 모습 수정하기

❖ Movie 컴포넌트에 genres props 넘겨주기

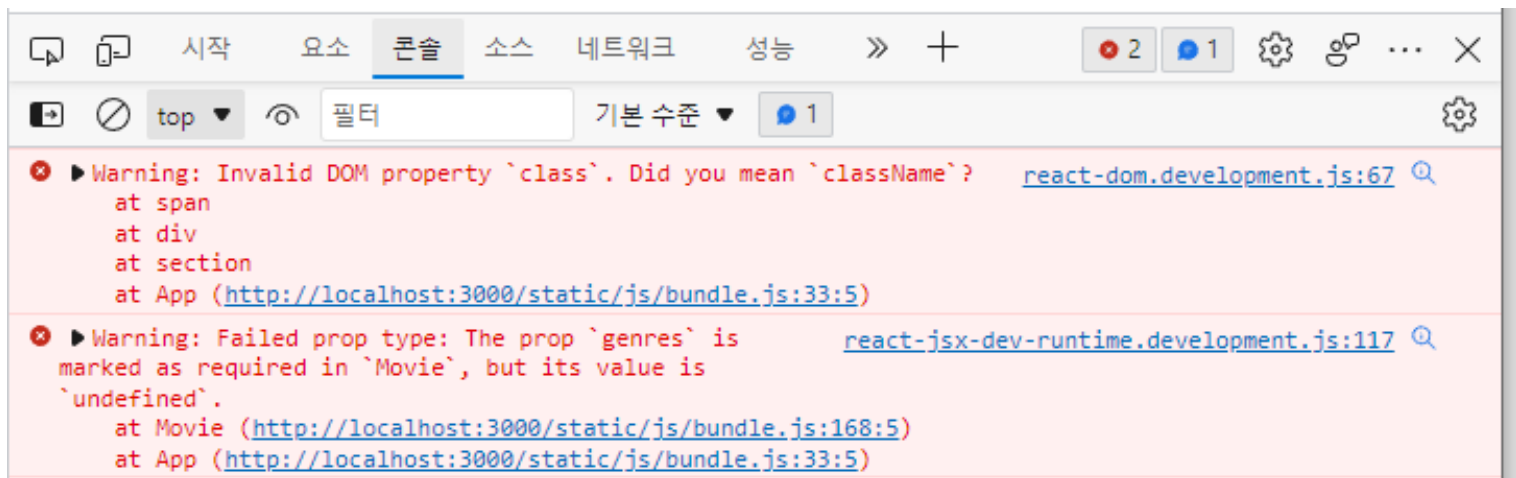
- App 컴포넌트에서 Movie 컴포넌트에 genres props를 넘겨준다 가정하고 Movie 컴포넌트를 수정해 보자.
- Movie 컴포넌트 인자에 genres를 추가하고, Movie.propTypes에는 genres props가 문자열 배열 arrayOf(propTypes.string)이며 반드시 필요함(isRequired)을 추가하자.

| | |
|----|--|
| 14 | <code></div></code> |
| 15 | <code>);</code> |
| 16 | <code>}</code> |
| 17 | |
| 18 | <code>Movie.propTypes = {</code> |
| 19 | <code> year: propTypes.number.isRequired,</code> |
| 20 | <code> title: propTypes.string.isRequired,</code> |
| 21 | <code> summary: propTypes.string.isRequired,</code> |
| 22 | <code> poster: propTypes.string.isRequired,</code> |
| 23 | <code> genres: propTypes.arrayOf(propTypes.string).isRequired,</code> |
| 24 | <code>};</code> |
| 25 | |
| 26 | <code>export default Movie;</code> |

1 영화 앱 전체 모습 수정하기

❖ Movie 컴포넌트에 genres props 넘겨주기

- 영화 앱을 수정한 다음 실행하여 [콘솔] 탭을 열면 경고 메시지 2개가 보인다.



- 첫 번째 경고 메시지는 JSX에 사용한 속성 중 `class` 속성이 `className`으로 사용되어야 한다는 뜻이고, 두 번째 경고 메시지는 `genres props`가 필수(required)인데 `Movie` 컴포넌트에 `undefined`로 넘어왔다는 뜻이다.
- `class` 속성은 `className`으로 바꿔 주고, `genres`가 잘 넘어오도록 `App.js`를 수정해 보자.

1 영화 앱 전체 모습 수정하기

❖ App 컴포넌트 수정하기

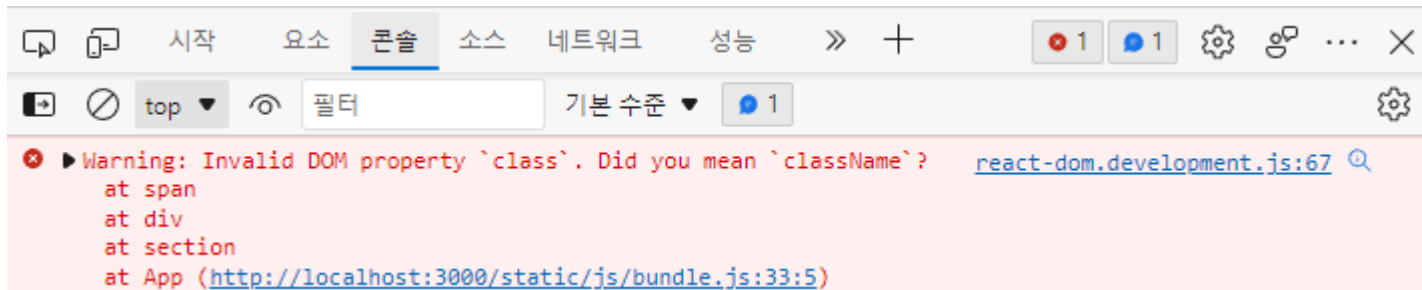
- App.js에서 우선 **genres**가 **undefined**인 상태를 고쳐 보자.
- App 컴포넌트에서 **Movie** 컴포넌트로 **genres props**를 전달하면 된다.

| | |
|----|---|
| | ... |
| 6 | class App extends React.Component{ |
| | ... |
| 25 | render(){ |
| 26 | const {isLoading, movies} = this.state; |
| 27 | return (|
| | ... |
| 34 | <div class="movies"> |
| 35 | {movies.map(movie => (|
| 36 | <Movie |
| 37 | key={movie.id} |
| 38 | ... |
| 39 | poster={movie.medium_cover_image} |
| 40 | genres={movie.genres} |
| 41 | /> |
| 42 |)}}} |

1 영화 앱 전체 모습 수정하기

❖ App 컴포넌트 수정하기

- App 컴포넌트에서 Movie 컴포넌트로 `genre props`를 넘겨주므로 이제 `genre props`가 `undefined`라는 경고 메시지는 사라졌다.
- 그러면 `class` 속성의 이름을 `className`으로 바꿔 보자.
- 그나저나 HTML에서는 `class`가 맞는데 왜 리액트에서는 `class`가 아닌 `className`으로 수정하라고 경고할까?
- 그건 JSX 때문이다.
- HTML의 `class`와 자바스크립트의 `class`라는 이름이 겹치면 리액트가 혼란스러울 수 있으므로 하나는 다른 이름을 쓰도록 하는 것이다.



1 영화 앱 전체 모습 수정하기

❖ class 속성 이름 className으로 바꿔 주기

- 앞에서 작성한 모든 코드에 있는 JSX의 class 속성 이름을 className으로 바꿔 주자.
- App.js, Movie.js 순서로 파일을 수정하자.

| | |
|----|---|
| | ... |
| 25 | render(){ |
| 26 | const {isLoading, movies} = this.state; |
| 27 | return (|
| 28 | <section className="container"> |
| 29 | {isLoading ? (|
| 30 | <div className="loader"> |
| 31 | Loading... |
| 32 | </div> |
| 33 |) : (|
| 34 | <div className="movies"> |
| 35 | {movies.map(movie => (|
| | ... |

1 영화 앱 전체 모습 수정하기

❖ class 속성 이름 className으로 바꿔 주기

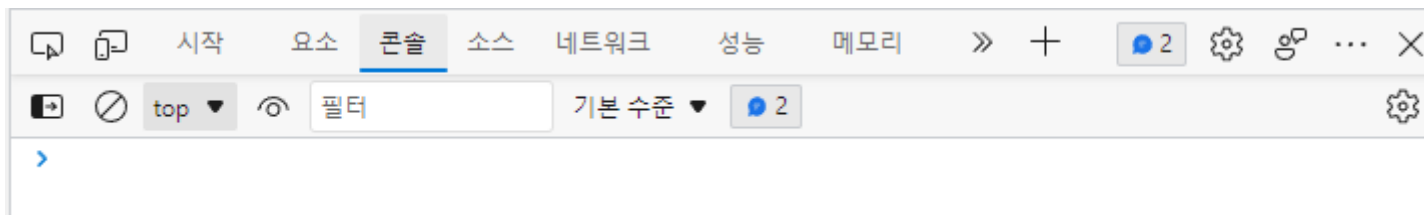
- 앞에서 작성한 모든 코드에 있는 JSX의 class 속성 이름을 className으로 바꿔 주자.
- App.js, Movie.js 순서로 파일을 수정하자.

| | |
|----|--|
| | ... |
| 5 | function Movie({ title, year, summary, poster, genres}){ |
| 6 | return (|
| 7 | <div className="movie"> |
| 8 | |
| 9 | <div className="movie__data"> |
| 10 | <h3 className="movie__title">{title} </h3> |
| 11 | <h5 className="movie__year">{year}</h5> |
| 12 | <p className="movie__summary">{summary}</p> |
| 13 | </div> |
| 14 | </div> |
| 15 |); |
| 16 | } |
| | ... |

1 영화 앱 전체 모습 수정하기

❖ class 속성 이름 className으로 바꿔 주기

- 이제 class 속성 이름에 대한 경고 메시지도 사라졌다.
- 이와 비슷한 경우를 하나 더 살펴보자.
- HTML의 label 엘리먼트에는 for라는 속성을 추가할 수 있는데, 이 속성 이름 역시 자바스크립트의 for 문 이름과 겹친다.
- 그래서 JSX에는 `<label for="name">`이 아니라 `<label htmlFor="name">`와 같이 작성해 줘야 한다.



1 영화 앱 전체 모습 수정하기

❖ 영화 장르 출력하기

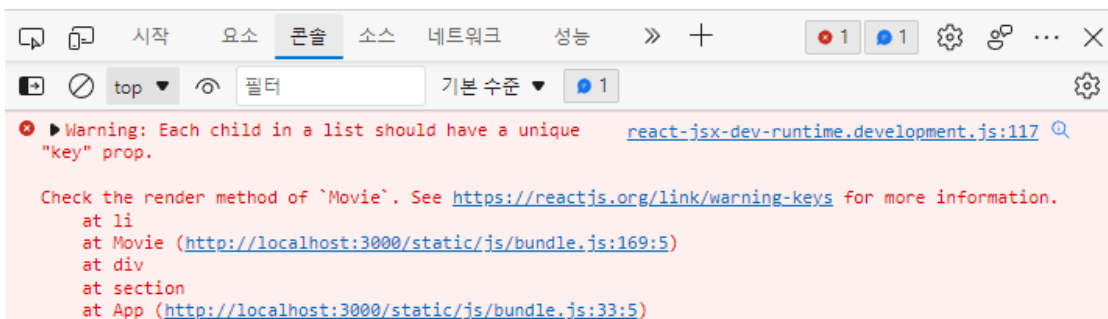
- 이제 **Movie** 컴포넌트에서 장르를 출력하도록 코드를 수정해 보자.
- **genres props**가 배열이므로 **map()** 함수를 사용한다.
- **Genres props**를 **ul, li** 엘리먼트로 감싸 출력해 보자.

| | |
|----|--|
| | ... |
| 5 | function Movie({ title, year, summary, poster, genres}){ |
| 6 | return (|
| 7 | <div className="movie"> |
| 8 | |
| 9 | <div className="movie__data"> |
| 10 | <h3 className="movie__title">{title} </h3> |
| 11 | <h5 className="movie__year">{year}</h5> |
| 12 | <ul className="movie__genres"> |
| 13 | {genres.map((genre) => { |
| 14 | return <li className="movie__genres">{genres}; |
| 15 |)}} |
| 16 | |
| 17 | <p className="movie__summary">{summary}</p> |
| | ... |

1 영화 앱 전체 모습 수정하기

❖ 영화 장르 출력하기

- 이제 장르까지 잘 출력한다.



```
Warning: Each child in a list should have a unique "key" prop.  
    Check the render method of `Movie`. See https://reactjs.org/link/warning-keys for more information.  
    at li  
    at Movie (http://localhost:3000/static/js/bundle.js:169:5)  
    at div  
    at section  
    at App (http://localhost:3000/static/js/bundle.js:33:5)
```



Jai Bhim

2021

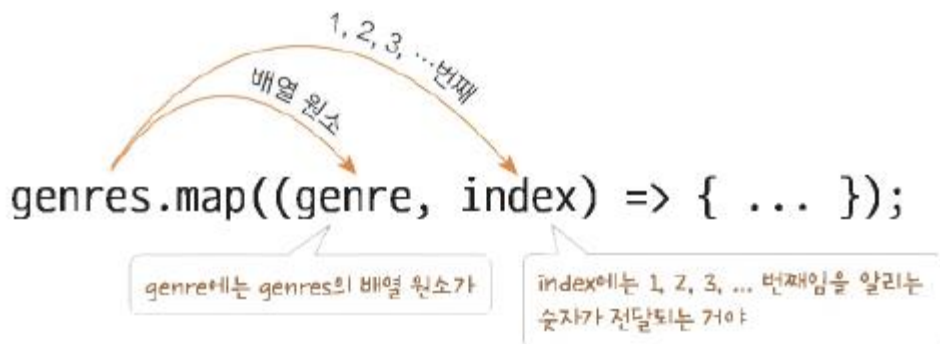
- CrimeDrama
- CrimeDrama

Rajakannu and his wife Sengeneni belong to a lower cast and works as laborer. They live a life of poverty but are happy with what they have. Rajakannu and Sengeneni give good news once Rajakannu is called to the house of a friend. Inside his house, the next day theft of jewelry is reported in the same hotel. The cops go to arrest Rajakannu but he leaves the town for work following the rest of family members asking them details about the Rajakannu. The cops find his brothers in jail asking him to confess the crime they did not commit later. They have eloped from the prison to escape torture. Mythra who teaches tribal people fights for tribal people and after hearing story of Sengeneni files a Habeas Corpus. alex.mjacks@gmail.com

1 영화 앱 전체 모습 수정하기

❖ 영화 장르 출력하기

- 그런데 [콘솔] 탭을 보면 경고 메시지가 또 있다!
- 이 경고 메시지는 장르를 출력할 때 사용한 li 엘리먼트에 **key props**를 추가하지 않아서 발생한 것이다.
- 그런데 장르는 API에서 id와 같은 값을 매겨 주지 않았다.
- 따라서 li 엘리먼트에 제공할 마땅한 **key**가 없다.
- 이런 경우에는 어떻게 해야 할까?
- **map()** 함수에 전달할 함수에 두 번째 인자를 전달하면 된다.
- **map()** 함수에 전달할 함수의 2번째 인자에는 **map()** 함수가 반복 실행하며 반환할 배열 원소의 인덱스가 자동으로 들어온다.
- 이 값을 이용하면 **key props**를 손쉽게 추가할 수 있다.



1 영화 앱 전체 모습 수정하기

❖ li 엘리먼트에 key props 추가하기

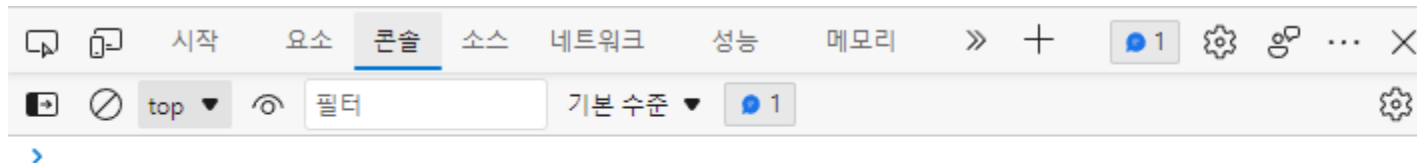
- 보통 `map()` 함수의 2번째 인자 이름은 `index`라고 지어 주므로 `genres.map((genre, index) => <li key={index} className="genres _genre"> {genre})`와 같이 코드를 작성하자.

| | |
|----|--|
| | ... |
| 5 | function Movie({ title, year, summary, poster, genres}){ |
| 6 | return (|
| 7 | <div className="movie"> |
| | ... |
| 12 | <ul className="movie_genres"> |
| 13 | {genres.map((genre, index) => { |
| 14 | return(|
| 15 | <li key={index} className="movie_genres"> |
| 16 | {genres} |
| 17 | |
| 18 |); |
| 19 |)}} |
| 20 | <p className="movie_summary">{summary}</p> |
| | ... |

1 영화 앱 전체 모습 수정하기

❖ li 엘리먼트에 key props 추가하기

- 영화 앱을 실행해 보면 key props 경고 메시지가 사라졌다.
- **Movie** 컴포넌트에 장르가 추가되었다.
- 이제 스타일링만 하면 된다!



2 영화 앱 스타일링하기

❖ App.css 파일 수정하기

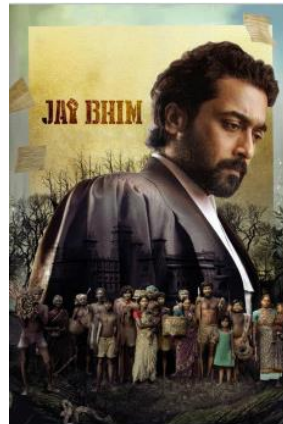
- 우선 **App.css** 파일부터 수정하자.
- **App.css** 파일에는 영화 앱 전체에 적용할 글꼴, 배경색 등을 적용했다.

| | |
|----|--|
| 1 | <code>*{</code> |
| 2 | <code> box-sizing: border-box;</code> |
| 3 | <code>}</code> |
| 4 | |
| 5 | <code>body{</code> |
| 6 | <code> margin: 0;</code> |
| 7 | <code> padding: 0;</code> |
| 8 | <code> font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;</code> |
| 9 | <code> background-color: #eff3f7;</code> |
| 10 | <code> height: 100%;</code> |
| 11 | <code>}</code> |

2 영화 앱 스타일링하기

❖ App.css 파일 수정하기

- App.css를 수정한 다음 영화 앱을 실행해 보면 배경색이 달라졌다.
- 하지만 영화 카드 모양으로 출력되지는 않는 상태이다.
- 영화 카드 모양은 Movie.css 를 수정해야 적용된다.



Jai Bhim

2021

- CrimeDrama
- CrimeDrama

Rajakannu and his wife Sengeni belong to a lower cast and works as labors in the field life of poverty but are happy with what they have. Rajakannu and Sengeni plan a second good news once Rajkannu is called to the house of an upper caste man as a snake has stolen jewelry is reported in the same house suspicious raising towards Rajkannu. He leaves the town for work following which the cops detain a pregnant Sengeni and rest about the Rajkannu. The cops trace Rajkannu and torture him and his brothers in jail as he does not commit later Sengeni finds that Rajkannu and his brother have eloped from the prison. A tribal village comes across a lawyer Chandru who fights for tribal people and a Habeas corpus case in court. —alex.mjacko@gmail.com

2 영화 앱 스타일링하기

❖ **Movie.css** 파일 수정하기

- 이어서 **Movie.css** 파일을 수정하자.

| | |
|----|---|
| 1 | .movies .movie{ |
| 2 | background-color: white; |
| 3 | margin-bottom: 70px; |
| 4 | font-weight: 300; |
| 5 | padding: 20px; |
| 6 | border-radius: 5px; |
| 7 | color: #adaeb9; |
| 8 | box-shadow: 0 13px 27px -5px rgba(50, 50, 93, 0.25), 0 8px 16px -8px rgba(0, 0, 0, 0.3), 0 -6px 16px rgba(0, 0, 0, 0.025); |
| 9 | } |
| 10 | |
| 11 | .movies .movie a{ |
| 12 | display: grid; |
| 13 | grid-template-columns: minmax(150px, 1fr) 2fr; |
| 14 | grid-gap: 20px; |
| 15 | text-decoration: none; |
| 16 | color: inherit; |
| 17 | } |

2 영화 앱 스타일링하기

❖ **Movie.css** 파일 수정하기

- 이어서 **Movie.css** 파일을 수정하자.

| | |
|----|--|
| 18 | |
| 19 | .movie img{ |
| 20 | position: relative; |
| 21 | top: -50px; |
| 22 | max-width: 150px; |
| 23 | width: 100%; |
| 24 | margin-right: 30px; |
| 25 | box-shadow: 0 30px 60px -12px rgba(50, 50, 93, 0.25), 0 18px 36px -18px rgba(0, 0, 0, 0.3), 0 -12px 36px -8px rgba(0, 0, 0, 0.025); |
| 26 | } |
| 27 | |
| 28 | .movie .movie_title, |
| 29 | .movie .movie__year{ |
| 30 | margin: 0; |
| 31 | font-weight: 300; |
| 32 | } |
| 33 | |

2 영화 앱 스타일링하기

❖ **Movie.css** 파일 수정하기

- 이어서 **Movie.css** 파일을 수정하자.

```
34 .movie .movie_title{
```

```
35     margin-bottom: 5px;
```

```
36     font-size: 24px;
```

```
37     color: #2c2c2c;
```

```
38 }
```

```
40 .movie .movie_genres{
```

```
41     list-style: none;
```

```
42     padding: 0;
```

```
43     margin: 0;
```

```
44     display: flex;
```

```
45     flex-wrap: wrap;
```

```
46     margin: 5px 0px;
```

```
47 }
```

```
48
```

2 영화 앱 스타일링하기

❖ **Movie.css** 파일 수정하기

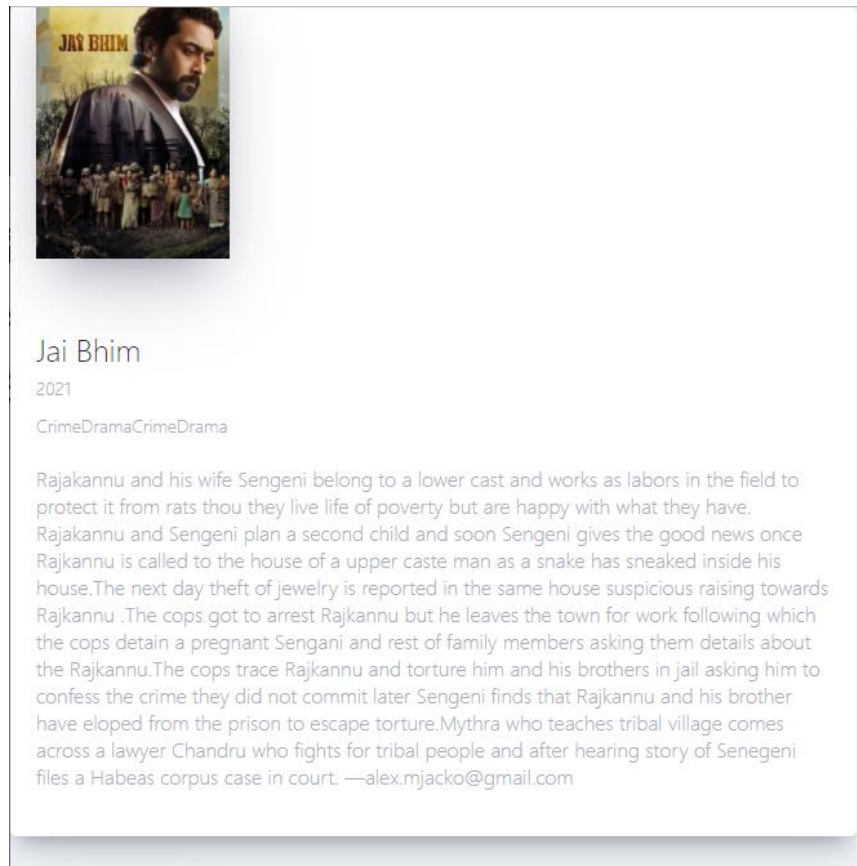
- 이어서 **Movie.css** 파일을 수정하자.

| | |
|----|----------------------------------|
| 49 | <code>.movie_genres li,</code> |
| 50 | <code>.movie .movie_year{</code> |
| 51 | <code>margin-right: 10px;</code> |
| 52 | <code>font-size: 14px;</code> |
| 53 | <code>}</code> |
| | |

2 영화 앱 스타일링하기

❖ Movie.css 파일 수정하기

- **Movie.css** 파일을 수정하면 영화 정보가 카드 모양으로 출력된다.
- 이때 영화 카드는 브라우저 폭에 따라 크기가 조절된다.



2 영화 앱 스타일링하기

❖ Movie.css 파일 수정하기

- 조금 더 수정해 보자.
- 여기서 시놉시스를 자르고 싶다.
- 다시 말해 영화 카드 안에 있는 시놉시스는 180자로 제한하고 싶다.
- 알다시피 시놉시스는 `summary props`에 저장되어 있는 문자열을 출력한 것이다.
- `Summary props`에 있는 문자열을 180자로 제한하려면 어떻게 해야 할까?
- 바로 자바스크립트의 `slice()` 함수를 사용하면 된다.
- `Slice()` 함수를 문자열에 적용하려면 `[문자열].slice(시작, 끝)`와 같이 사용하면 된다.

```
"hereisstring".slice(0, 10) // "hereisstri"
```

012 ... 11

시작은 0부터고

↓ ↓
시작 끝

012 ... 9

끝은 포함되지 않음에 주의해!

2 영화 앱 스타일링하기

❖ 시놉시스 180 자로 제한하기

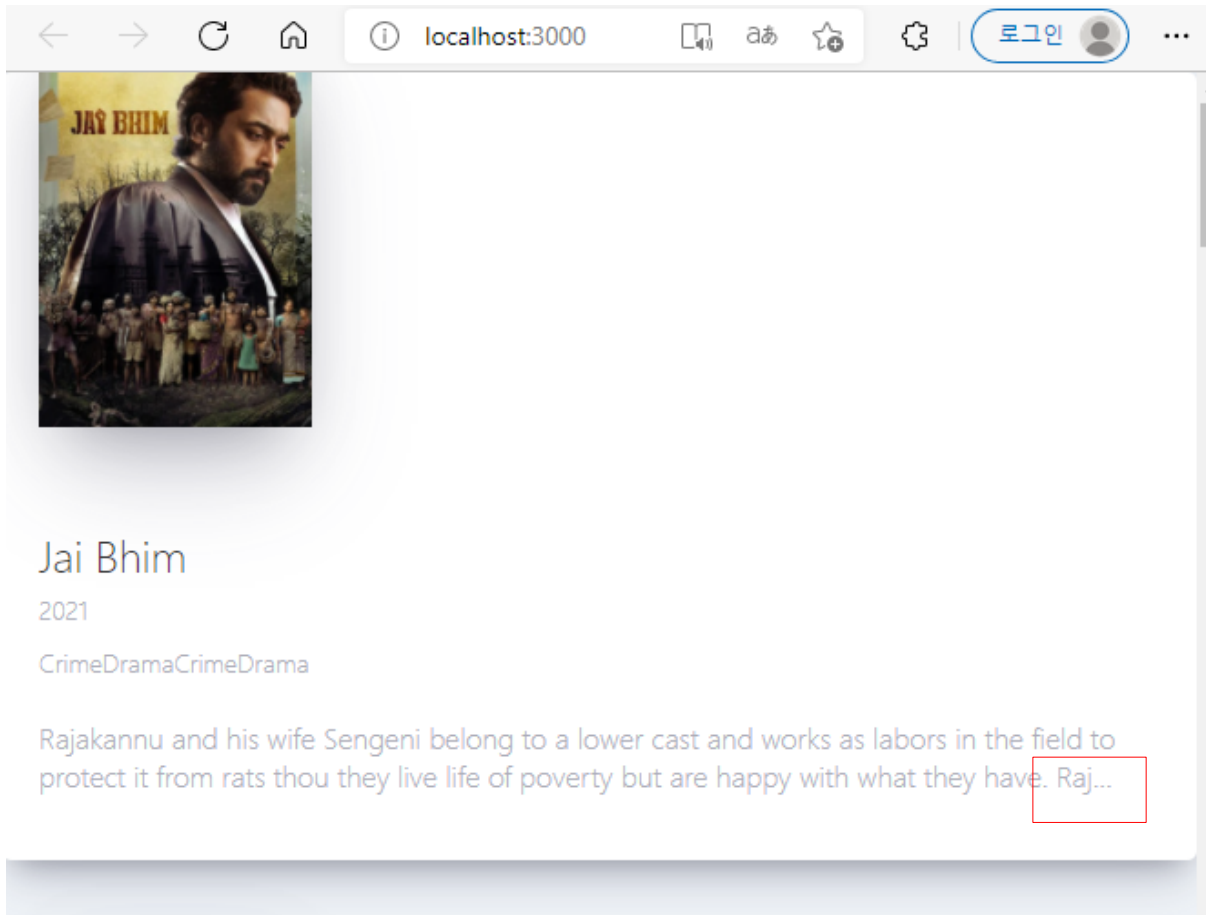
- {summary}를 {summary.slice(0, 180)}..}로 수정하자.

| | |
|----|--|
| | ... |
| 5 | function Movie({ title, year, summary, poster, genres}){ |
| 6 | return (|
| 7 | <div className="movie"> |
| | ... |
| 12 | <ul className="movie__genres"> |
| 13 | {genres.map((genre, index) => { |
| 14 | return(|
| 15 | <li key={index} className="movie__genres"> |
| 16 | {genres} |
| 17 | |
| 18 |); |
| 19 |)}} |
| 20 | <p className="movie__summary">{summary.slice(0, 180)}...</p> |
| | ... |

2 영화 앱 스타일링하기

❖ 시놉시스 180 자로 제한하기

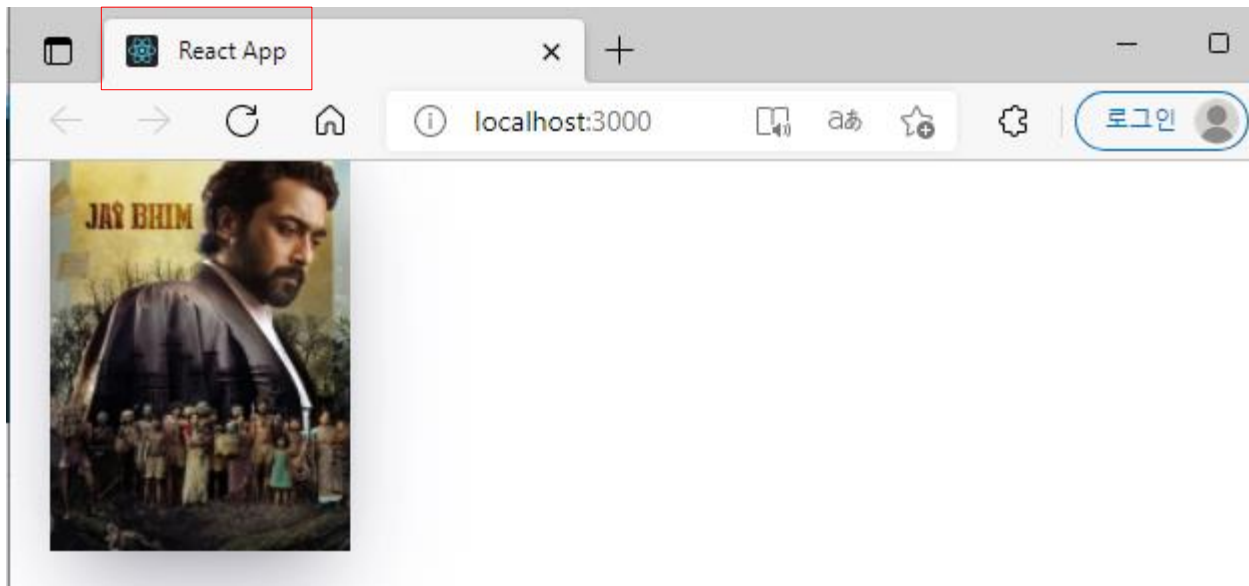
- 자, 시놉시스가 모두 180자로 제한되고 생략 표시로 ...이 추가됐다.
- 이런 식으로 긴 글은 줄여서 표현할 수 있다.



2 영화 앱 스타일링하기

❖ 영화 앱 제목 살펴보기

- 우리 영화 앱 제목은 뭐로 되어 있을까?
- 한번 살펴보자.
- 영화 앱의 제목은 브라우저 탭에서 볼 수 있다.
- 지금은 **create-react-app**에서 지정한 기본값인 **React App**이라고 되어 있을 것이다.

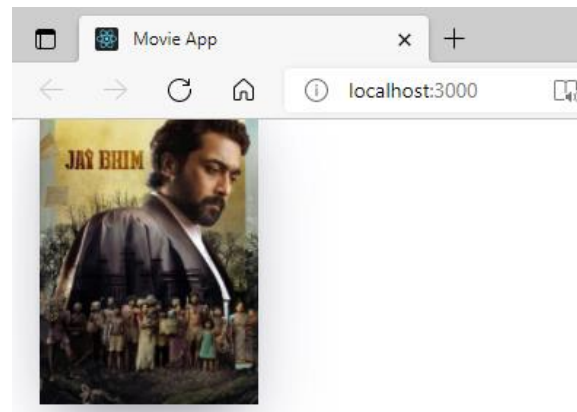


2 영화 앱 스타일링하기

❖ 영화 앱 제목 바꾸기

- 영화 앱 제목은 **public** 폴더의 **index.html** 파일을 열어 **title** 엘리먼트 사이에 넣으면 된다.
- **title** 엘리먼트는 **head** 엘리먼트 안에 있다.

| | |
|----|---|
| 1 | <code><!DOCTYPE html></code> |
| 2 | <code><html lang="en"></code> |
| 3 | <code><head></code> |
| | <code>...</code> |
| 27 | <code><title>Movie App</title></code> |
| 28 | <code></head></code> |
| 29 | <code><body></code> |
| | <code>...</code> |



2 영화 앱 스타일링하기

❖ 영화 앱 제목 바꾸기

- 자! 이제 영화 앱을 제대로 만든 것 같다.
- 영화 앱 제목도 새로 바꾸고, 영화 카드 모양도 만들고, 장르도 추가하고, 시놉시스는 180자로 제한했다.
- 그런데 아직 끝난 게 아니다!
- 다음에는 영화 앱에 내비게이션, 상세 페이지 이동 기능을 추가해보자.



Thank You !