

BÀI THỰC HÀNH 5

XÂY DỰNG FRONTEND VỚI REACTJS

Bài 1: Kết nối tới Backend.

1.1 Cài đặt axios cho dự án hiện tại.

1.2 Tạo lớp dịch vụ có tên MovieDataService trong thư mục .src/services/movies.js

1.3 Tạo các lời gọi dịch vụ tới backend, sử dụng axios để gọi bao gồm:

getAll();

get(id);

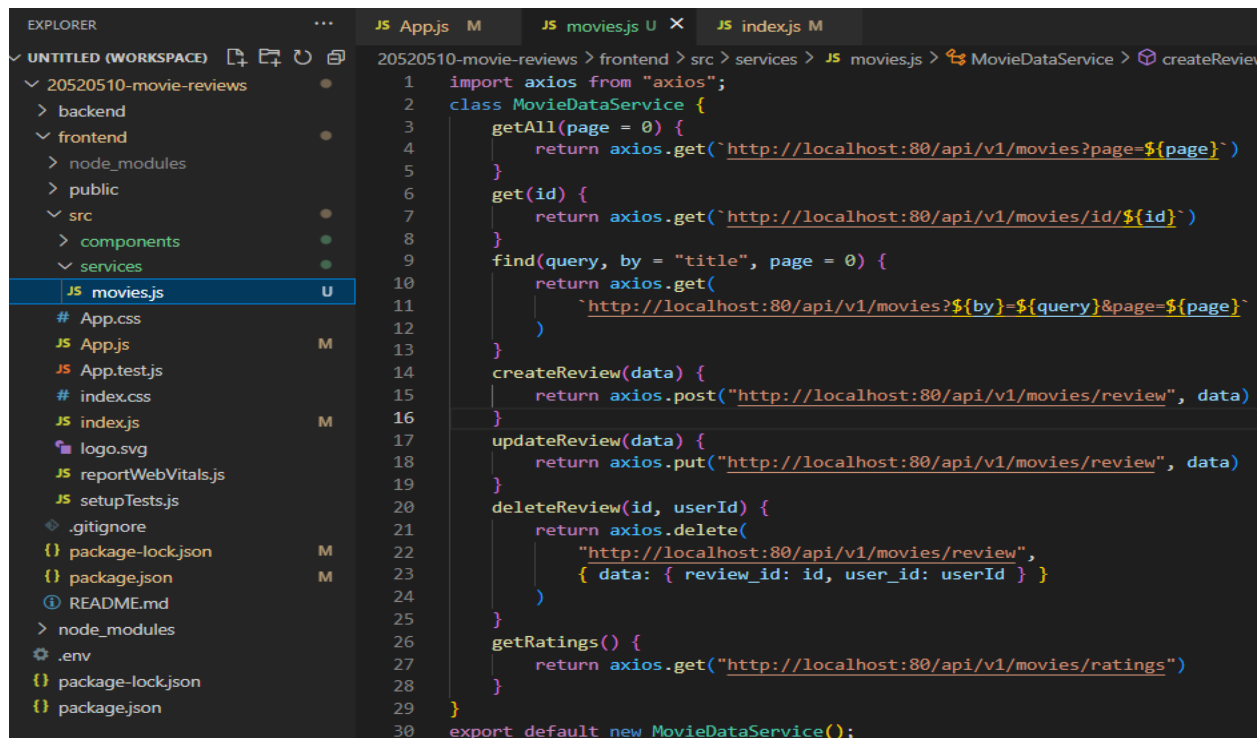
createReview(data);

updateReview(data);

deleteReview(data);

getRatings();

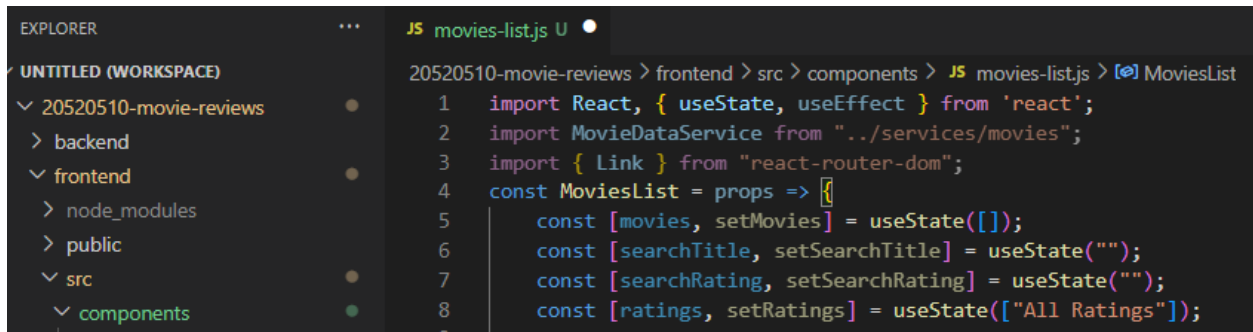
Tương ứng với các endpoint đã xây dựng ở phần backend ở các bài thực hành trước.



```
1 import axios from "axios";
2 class MovieDataService {
3   getAll(page = 0) {
4     return axios.get(`http://localhost:80/api/v1/movies?page=${page}`)
5   }
6   get(id) {
7     return axios.get(`http://localhost:80/api/v1/movies/id/${id}`)
8   }
9   find(query, by = "title", page = 0) {
10    return axios.get(
11      `http://localhost:80/api/v1/movies?${by}=${query}&page=${page}`
12    )
13  }
14  createReview(data) {
15    return axios.post("http://localhost:80/api/v1/movies/review", data)
16  }
17  updateReview(data) {
18    return axios.put("http://localhost:80/api/v1/movies/review", data)
19  }
20  deleteReview(id, userId) {
21    return axios.delete(
22      "http://localhost:80/api/v1/movies/review",
23      { data: { review_id: id, user_id: userId } }
24    )
25  }
26  getRatings() {
27    return axios.get("http://localhost:80/api/v1/movies/ratings")
28  }
29 }
30 export default new MovieDataService();
```

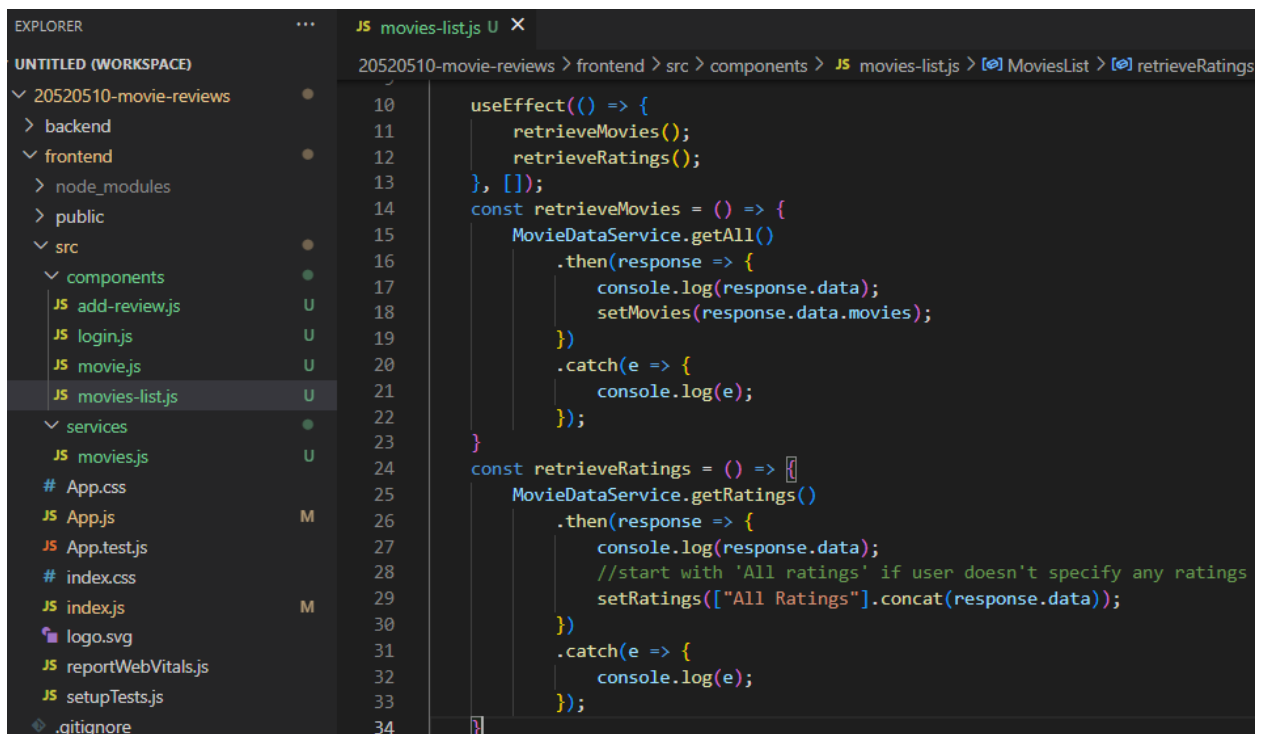
Bài 2: Xây dựng MoviesList Component.

2.1 Tạo các biến trạng thái: movies, searchText, searchRating, ratings sử dụng useState().



```
1 import React, { useState, useEffect } from 'react';
2 import MovieDataService from "../services/movies";
3 import { Link } from "react-router-dom";
4 const MoviesList = props => {
5   const [movies, setMovies] = useState([]);
6   const [searchTitle, setSearchTitle] = useState("");
7   const [searchRating, setSearchRating] = useState("");
8   const [ratings, setRatings] = useState(["All Ratings"]);
```

2.2 Tạo 2 phương thức retrieveMovies() và retrieveRatings() để lấy thông tin movie cùng danh sách các loại ratings. Và dùng useEffect() để gọi chung sau khi giao diện kết xuất xong.



```
10 useEffect(() => {
11   retrieveMovies();
12   retrieveRatings();
13 }, []);
14 const retrieveMovies = () => {
15   MovieDataService.getAll()
16     .then(response => {
17       console.log(response.data);
18       setMovies(response.data.movies);
19     })
20     .catch(e => {
21       console.log(e);
22     });
23 }
24 const retrieveRatings = () => {
25   MovieDataService.getRatings()
26     .then(response => {
27       console.log(response.data);
28       //start with 'All ratings' if user doesn't specify any ratings
29       setRatings(["All Ratings"].concat(response.data));
30     })
31     .catch(e => {
32       console.log(e);
33     });
34 }
```

2.3 Tạo 2 search form gồm tìm theo title, và tìm theo rating.

```
EXPLORER
20520510-movie-reviews
  backend
  frontend
    node_modules
    public
    src
      components
        JS movies-list.js U X

20520510-movie-reviews > frontend > src > components > JS movies-list.js
42 const onChangeSearchTitle = e => {
43   const searchTitle = e.target.value;
44   setSearchTitle(searchTitle);
45 }
46 const onChangeSearchRating = e => {
47   const searchRating = e.target.value;
48   setSearchRating(searchRating);
49 }
```

```
EXPLORER
20520510-movie-reviews
  backend
  frontend
    node_modules
    public
    src
      components
        JS add-review.js U
        JS login.js U
        JS movie.js U
        JS movies-list.js U
      services
        JS movies.js U
      # App.css
      JS App.js M
      JS App.test.js
      # index.css
      JS index.js M
      logo.svg
      JS reportWebVitals.js
      JS setupTests.js
      .gitignore
      {} package-lock.json M
      {} package.json M
      README.md
      > node_modules
      .env
      {} package-lock.json
      {} package.json

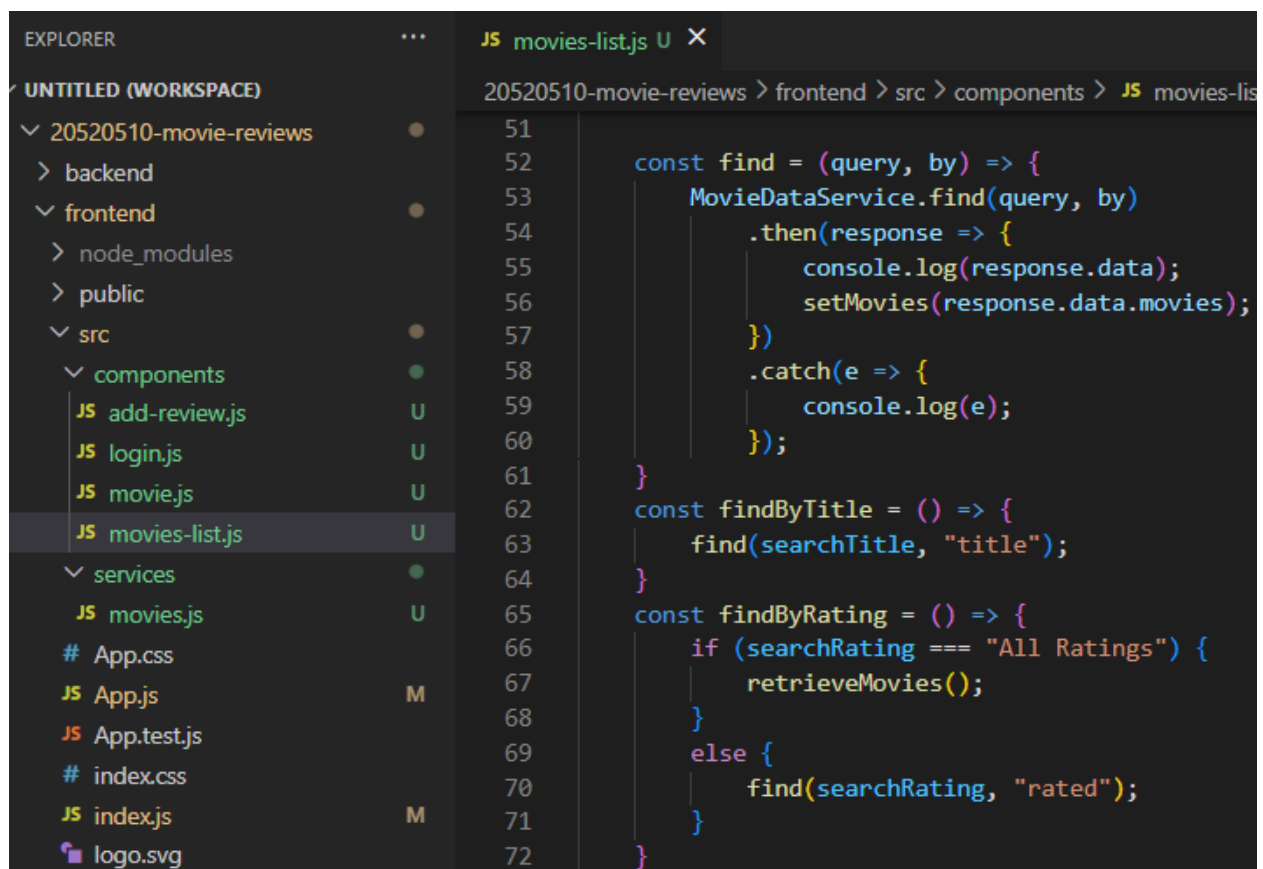
20520510-movie-reviews > frontend > src > components > JS movies-list.js > MoviesList
51 return (
52   <div className="App">
53     <Container>
54       <Form>
55         <Row>
56           <Col>
57             <Form.Group>
58               <Form.Control
59                 type="text"
60                 placeholder="Search by title"
61                 value={searchTitle}
62                 onChange={onChangeSearchTitle}
63               />
64             </Form.Group>
65             <Button
66               variant="primary"
67               type="button"
68               onClick={findByTitle}
69             >
70               Search
71             </Button>
72           </Col>
73           <Col>
74             <Form.Group>
75               <Form.Control
76                 as="select" onChange={onChangeSearchRating} >
77                 {ratings.map(rating => {
78                   return (
79                     <option value={rating}>{rating}</option>
80                   )
81                 })}
82               </Form.Control>
83             </Form.Group>
84             <Button
85               variant="primary"
86               type="button"
87               onClick={findByRating}
88             >
89               Search
90             </Button>
91           </Col>
92         </Row>
93       </Form>
94     </Container>
95   </div>
```

2.4 Hiển thị các movie bằng <Card> của React-bootstrap.



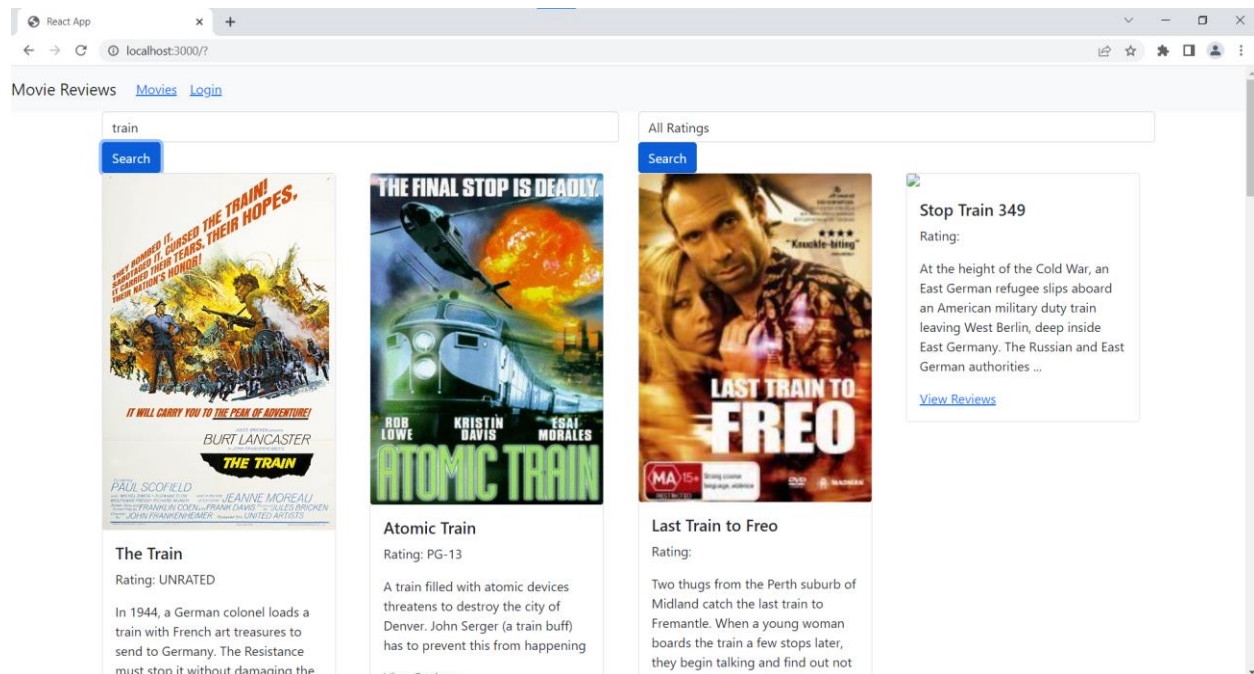
```
82      </Row>
83    </Form>
84    <Row>
85      {movies.map((movie) => {
86        return (
87          <Col>
88            <Card style={{ width: "18rem" }}>
89              <Card.Img src={movie.poster + "/100px180"} />
90              <Card.Body>
91                <Card.Title>{movie.title}</Card.Title>
92                <Card.Text>Rating: {movie.rated}</Card.Text>
93                <Card.Text>{movie.plot}</Card.Text>
94                <Link to={"/movies/" + movie._id}>View Reviews</Link>
95              </Card.Body>
96            </Card>
97          </Col>
98        );
99      })}
100    </Row>
```

2.5 Hiện thực 2 phương thức `findByTitle()` và `findByRating()` để tìm phim theo Title hoặc Rating.



```
51
52    const find = (query, by) => {
53      MovieDataService.find(query, by)
54        .then(response => {
55          console.log(response.data);
56          setMovies(response.data.movies);
57        })
58        .catch(e => {
59          console.log(e);
60        });
61    }
62    const findByTitle = () => {
63      find(searchTitle, "title");
64    }
65    const findByRating = () => {
66      if (searchRating === "All Ratings") {
67        retrieveMovies();
68      }
69      else {
70        find(searchRating, "rated");
71      }
72    }
```

Chạy ứng dụng:



Bài 3. Hiển thị thông tin trang movie khi nhấn vào ‘View Reviews’.

3.1 Thiết lập mã nguồn cho component Movie trong tệp tin ./components/movie.js gồm:

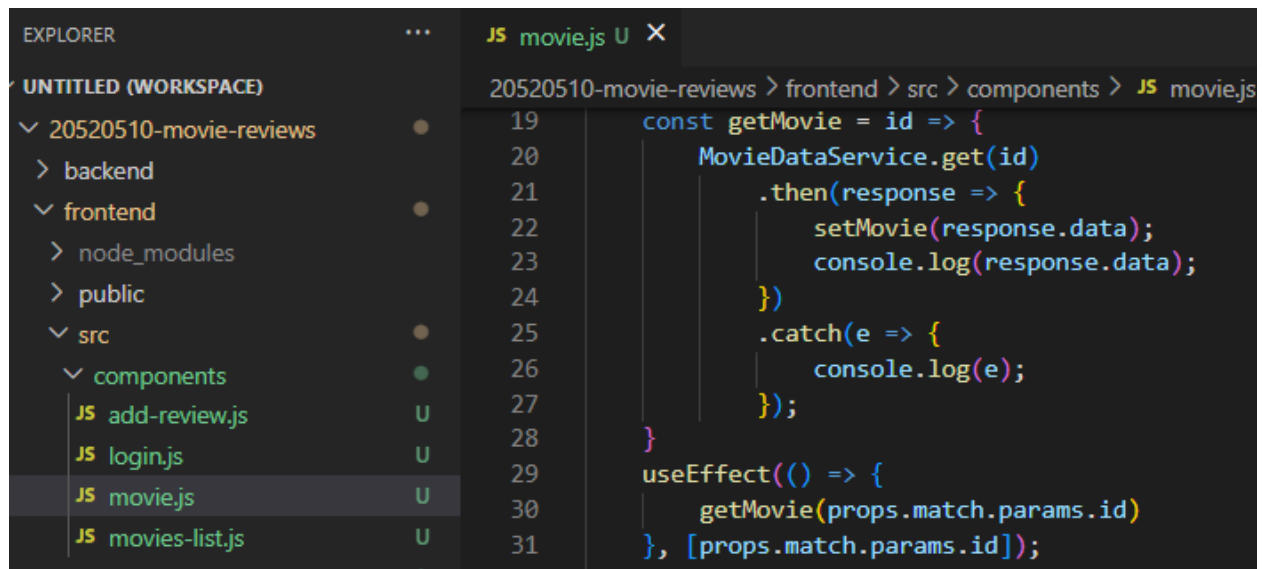
- Biến trạng thái movie để lưu trữ thông tin chi tiết của movie như id, title, rated, reviews.

```

EXPLORER
20520510-movie-reviews
  backend
  frontend
    node_modules
    public
    src
      components
JS movie.js U X
20520510-movie-reviews > frontend > src > components > JS
11
12 const Movie = props => {
13   const [movie, setMovie] = useState({
14     id: null,
15     title: "",
16     rated: "",
17     reviews: []
18   });

```

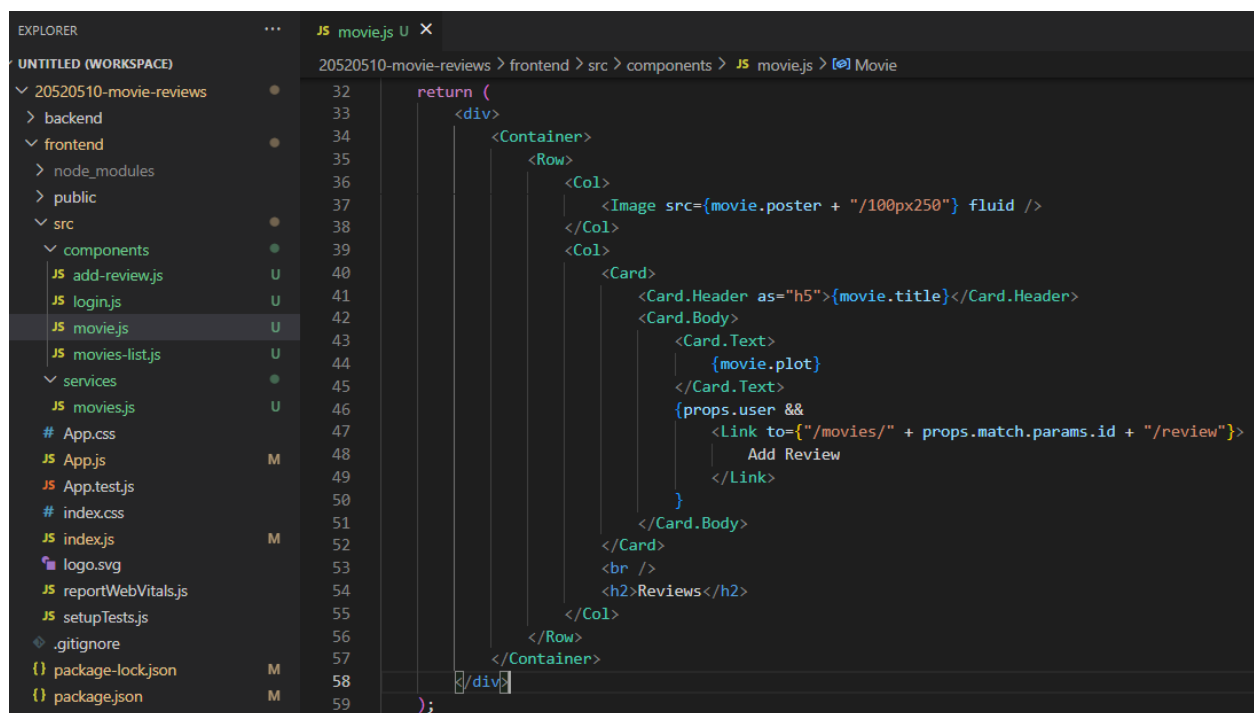
3.2 Xây dựng mã nguồn cho phương thức getMovie() trong component này để gọi phương thức get() trong MovieDataService.



```
EXPLORER 20520510-movie-reviews > frontend > src > components > JS movie.js U X
UNTITLED (WORKSPACE)
20520510-movie-reviews
  backend
  frontend
    node_modules
    public
    src
      components
        JS add-review.js U
        JS login.js U
        JS movie.js U
        JS movies-list.js U
      services
      JS movies.js U
  # App.css
  JS App.js M
  JS App.test.js
  # index.css
  JS index.js M
  logo.svg
  JS reportWebVitals.js
  JS setupTests.js
  .gitignore
  package-lock.json M
  package.json M

19 const getMovie = id => {
20   MovieDataService.get(id)
21     .then(response => {
22       setMovie(response.data);
23       console.log(response.data);
24     })
25     .catch(e => {
26       console.log(e);
27     });
28 }
29 useEffect(() => {
30   getMovie(props.match.params.id)
31 }, [props.match.params.id]);
```

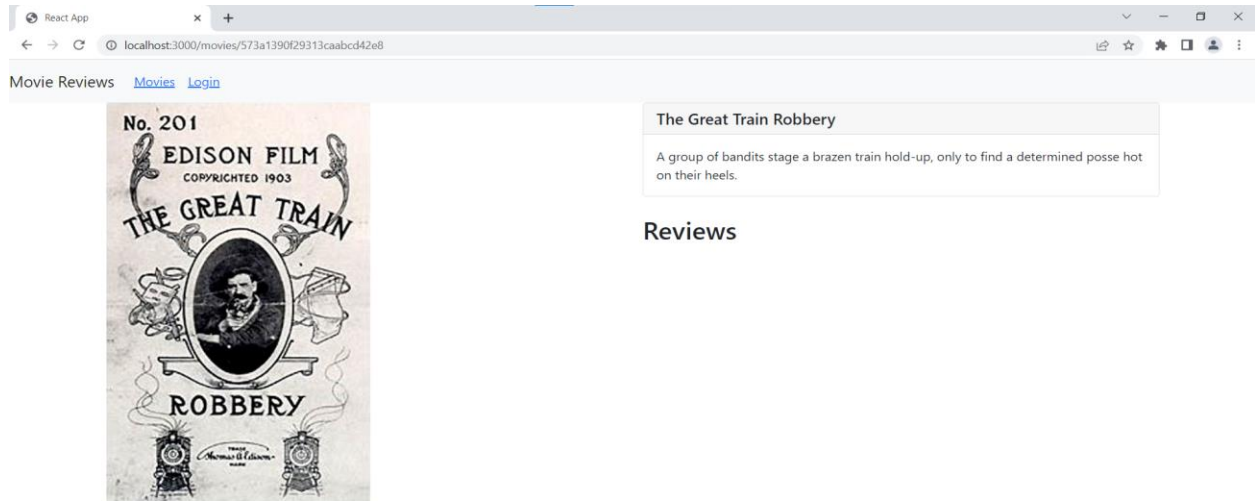
3.3 Trang trí cho phần JSX trả về để hiển thị như hình:



```
EXPLORER 20520510-movie-reviews > frontend > src > components > JS movie.js U X
UNTITLED (WORKSPACE)
20520510-movie-reviews
  backend
  frontend
    node_modules
    public
    src
      components
        JS add-review.js U
        JS login.js U
        JS movie.js U
        JS movies-list.js U
      services
      JS movies.js U
  # App.css
  JS App.js M
  JS App.test.js
  # index.css
  JS index.js M
  logo.svg
  JS reportWebVitals.js
  JS setupTests.js
  .gitignore
  package-lock.json M
  package.json M

32 return (
33   <div>
34     <Container>
35       <Row>
36         <Col>
37           <Image src={movie.poster + "/100px250"} fluid />
38         </Col>
39         <Col>
40           <Card>
41             <Card.Header as="h5">{movie.title}</Card.Header>
42             <Card.Body>
43               <Card.Text>
44                 {movie.plot}
45               </Card.Text>
46               {props.user &&
47                 <Link to={"/movies/" + props.match.params.id + "/review"}>
48                   Add Review
49                 </Link>
50               }
51             </Card.Body>
52           </Card>
53           <br />
54           <h2>Reviews</h2>
55         </Col>
56       </Row>
57     </Container>
58   </div>
59 );
```

Kết quả:



Bài 4. Hiện thị danh sách review tương ứng cho từng phim dưới phần Plot.

4.1 Viết đoạn mã nguồn JSX cho phép hiện thị danh sách review cho phim.

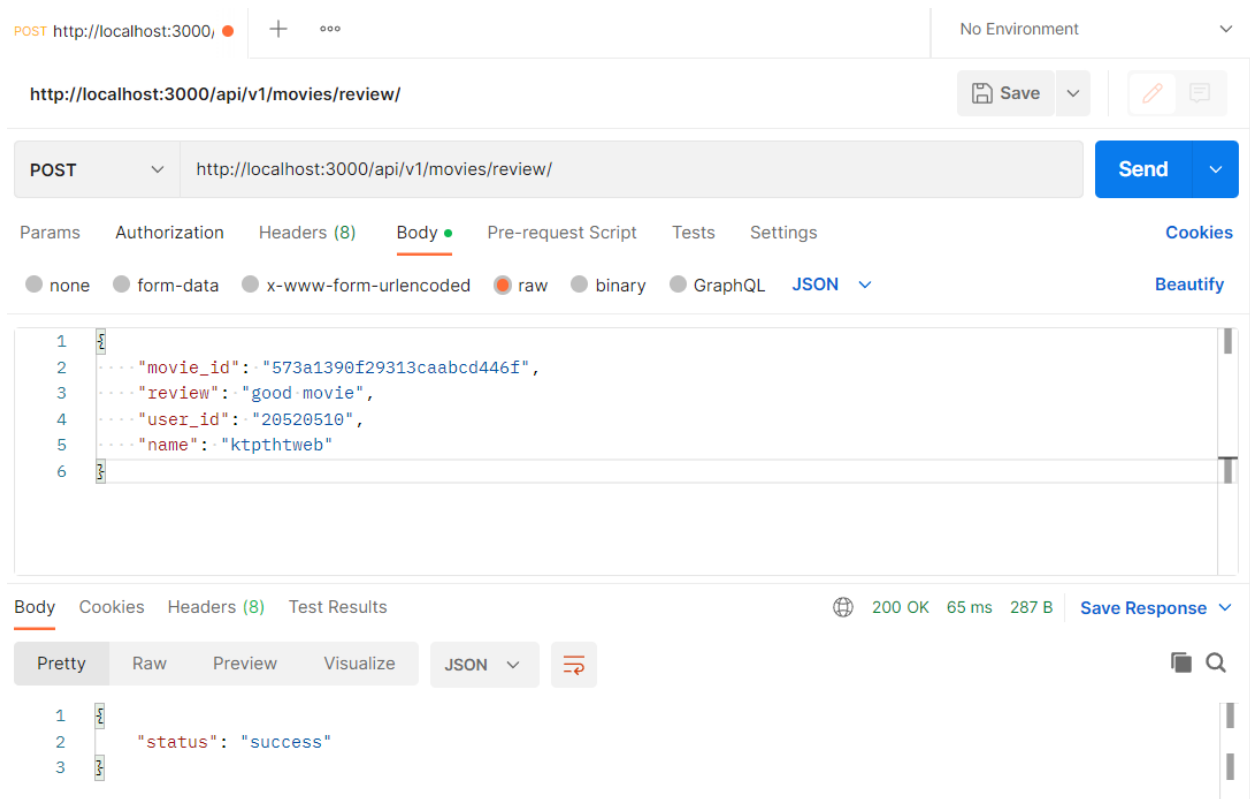
```

EXPLORER
  20520510-movie-reviews > frontend > src > components > JS movie.js > movie.js > movie.reviews.map() callback
  54
  55
  56
  57
  58
  59
  60
  61
  62
  63
  64
  65
  66
  67
  68
  69
  70
  71
  72
  73
  74
  75
  76
  77
  78

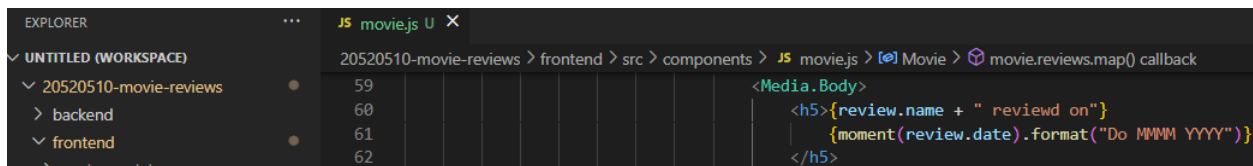
  <h2>Reviews</h2>
  <br />
  {movie.reviews.map((review, index) => {
    return (
      <Media key={index}>
        <Media.Body>
          <h5>{review.name + "reviewd on" + review.date}</h5>
          <p>{review.review}</p>
          {props.user && props.user.id === review.user_id &&
            <Row>
              <Col>
                <Link to={{
                  pathname: "/movies/" + props.match.params.id +
                    "/review",
                  state: { currentReview: review }
                }}>Edit</Link>
              </Col>
              <Col><Button variant="link">Delete</Button></Col>
            </Row>
          </Media.Body>
        </Media>
      );
    )
  })}
  </Col>

```

4.2 Xem lại slide 73 – chương 2 để hiểu cách thêm 1 review cho phim, và tiến hành thêm một số review thông qua các công cụ hỗ trợ như Postman, Insomnia.



4.2 Điều chỉnh lại cách hiển thị giờ với momentjs.



Kết quả:

