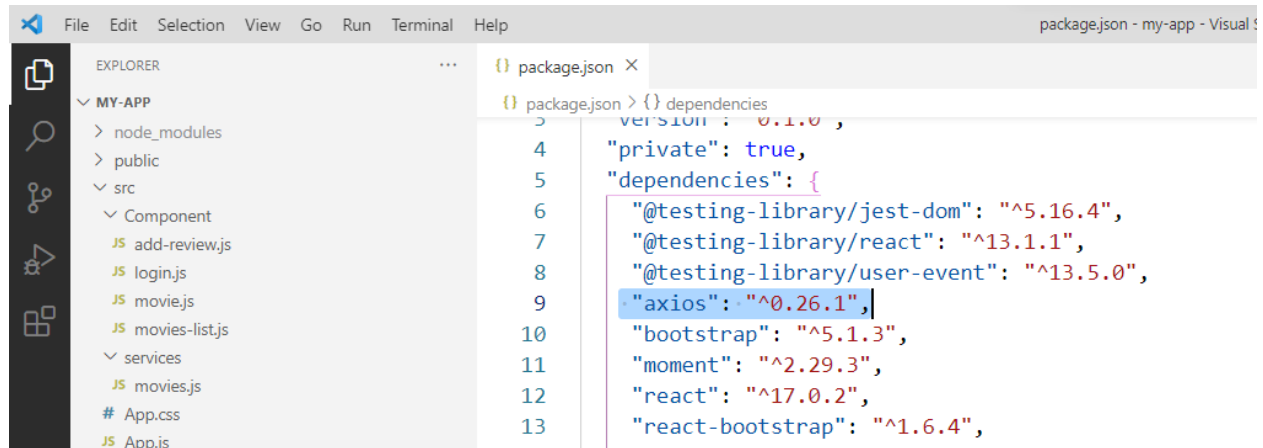


BÁO CÁO THỰC HÀNH 5

XÂY DỰNG FRONTEND VỚI REACTJS

Bài 1: Kết nối tới Backend.

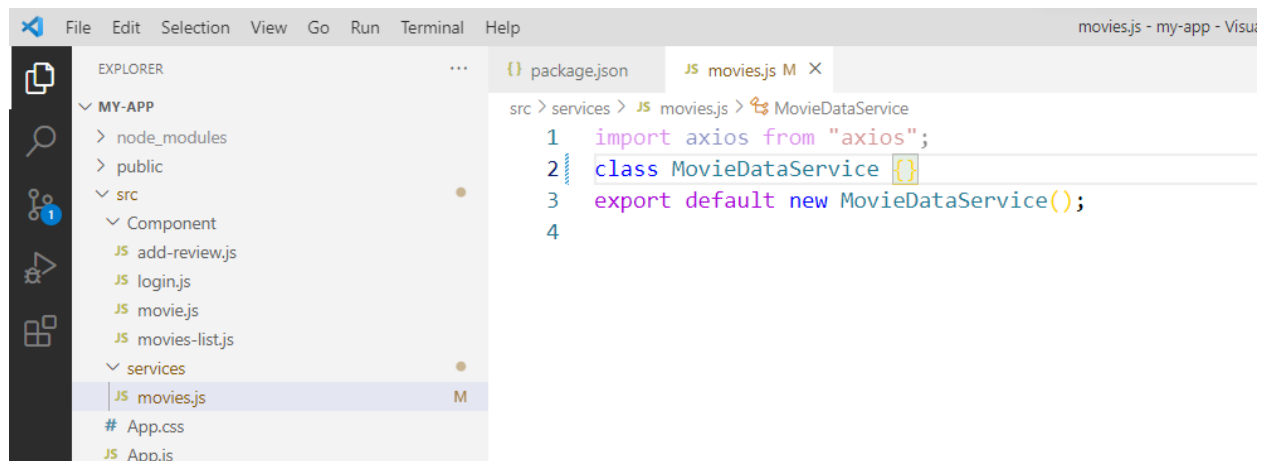
1.1 Cài đặt axios cho dự án hiện tại.



The screenshot shows the Visual Studio Code interface with the Explorer view on the left displaying the project structure. The main editor shows the `package.json` file. The `dependencies` section is expanded, and the `axios` package is being added to the list.

```
{} package.json > {} dependencies
1  "version": "0.1.0",
2  "private": true,
3  "dependencies": {
4    "@testing-library/jest-dom": "^5.16.4",
5    "@testing-library/react": "^13.1.1",
6    "@testing-library/user-event": "^13.5.0",
7    "axios": "^0.26.1",
8    "bootstrap": "^5.1.3",
9    "moment": "^2.29.3",
10   "react": "^17.0.2",
11   "react-bootstrap": "^1.6.4",
12   ...
13 }
```

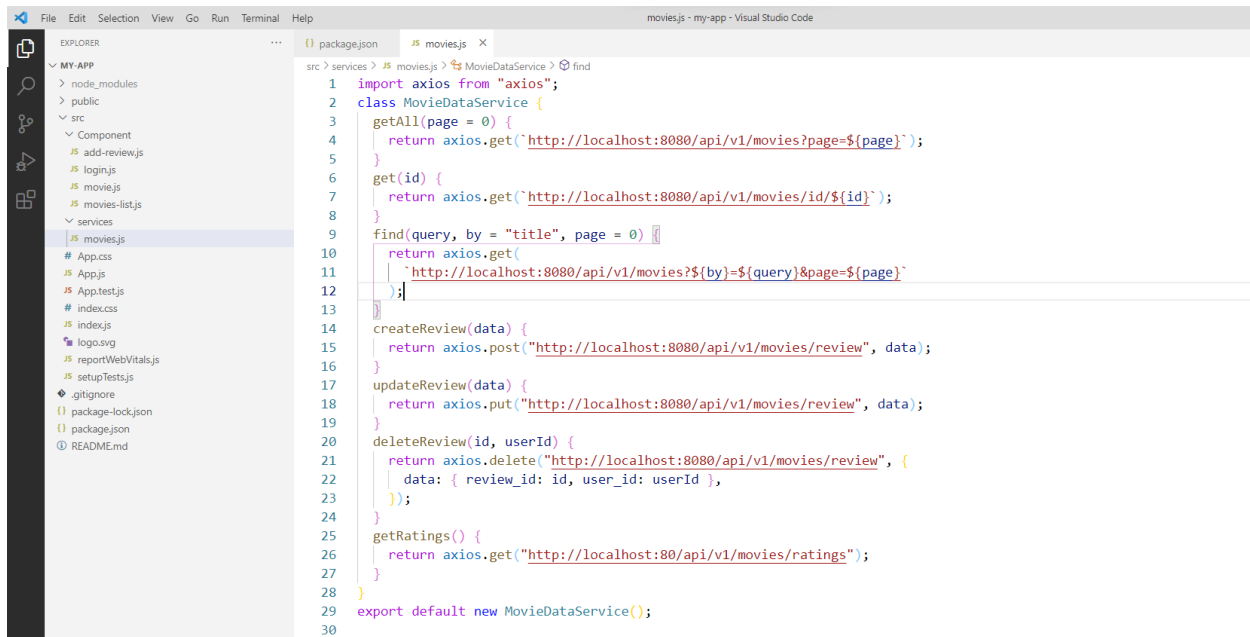
1.2 Tạo lớp dịch vụ có tên MovieDataService trong thư mục .src/services/movies.js



The screenshot shows the Visual Studio Code interface with the Explorer view on the left. The main editor shows the `movies.js` file in the `src > services` directory. The file contains the following code:

```
src > services > JS movies.js > MovieDataService
1  import axios from "axios";
2  class MovieDataService {}
3  export default new MovieDataService();
4
```

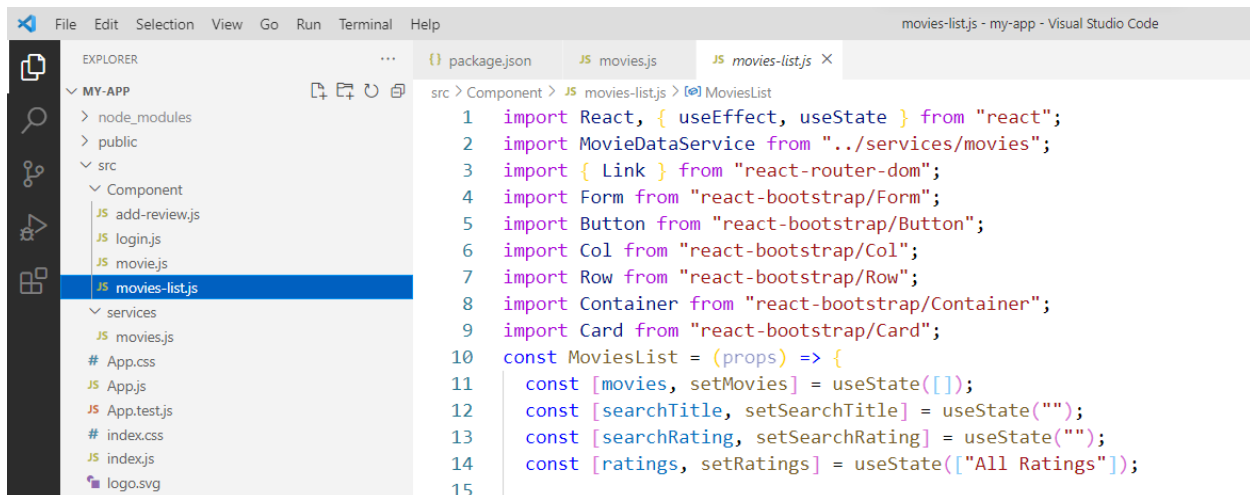
1.3 Tạo các lời gọi dịch vụ tới backend



```
1 import axios from "axios";
2 class MovieDataService {
3   getAll(page = 0) {
4     return axios.get(`http://localhost:8080/api/v1/movies?page=${page}`);
5   }
6   get(id) {
7     return axios.get(`http://localhost:8080/api/v1/movies/id/${id}`);
8   }
9   find(query, by = "title", page = 0) {
10    return axios.get(
11      `http://localhost:8080/api/v1/movies?${by}=${query}&page=${page}`
12    );
13  }
14  createReview(data) {
15    return axios.post("http://localhost:8080/api/v1/movies/review", data);
16  }
17  updateReview(data) {
18    return axios.put("http://localhost:8080/api/v1/movies/review", data);
19  }
20  deleteReview(id, userId) {
21    return axios.delete("http://localhost:8080/api/v1/movies/review", {
22      data: { review_id: id, user_id: userId },
23    });
24  }
25  getRatings() {
26    return axios.get("http://localhost:8080/api/v1/movies/ratings");
27  }
28 }
29 export default new MovieDataService();
30
```

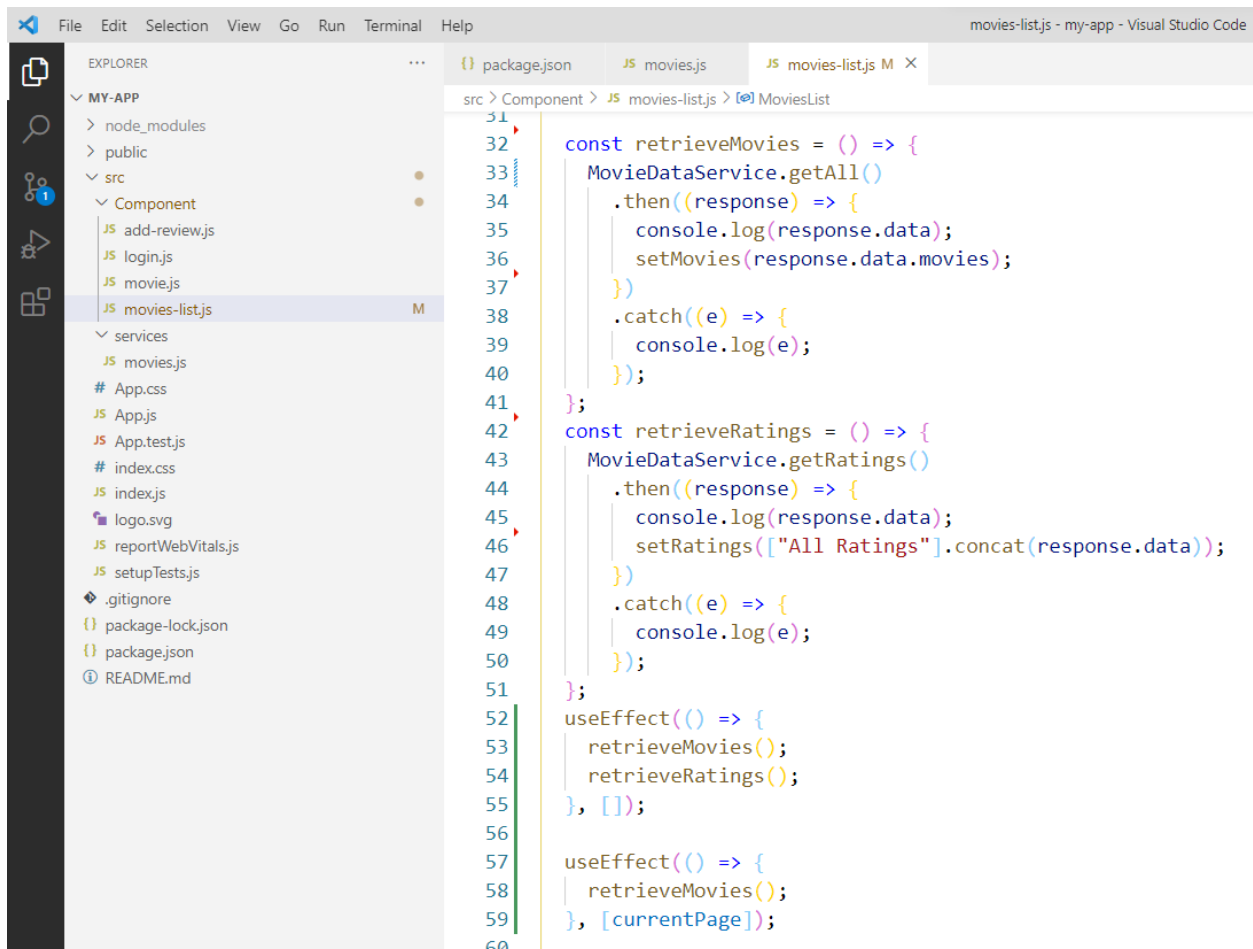
Bài 2: Xây dựng MoviesList Component.

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



```
1 import React, { useEffect, useState } from "react";
2 import MovieDataService from "../services/movies";
3 import { Link } from "react-router-dom";
4 import Form from "react-bootstrap/Form";
5 import Button from "react-bootstrap/Button";
6 import Col from "react-bootstrap/Col";
7 import Row from "react-bootstrap/Row";
8 import Container from "react-bootstrap/Container";
9 import Card from "react-bootstrap/Card";
10 const MoviesList = (props) => {
11   const [movies, setMovies] = useState([]);
12   const [searchTitle, setSearchTitle] = useState("");
13   const [searchRating, setSearchRating] = useState("");
14   const [ratings, setRatings] = useState(["All Ratings"]);
15
```

2.2 Tạo 2 phương thức retrieveMovies() và retrieveRatings()



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left and the Code editor on the right. The Explorer sidebar shows the project structure for 'MY-APP', with the 'src' folder expanded to show the 'Component' folder. The 'movies-list.js' file is selected. The Code editor shows the implementation of the 'MoviesList' component, with the following code:

```
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
const retrieveMovies = () => {  
  MovieDataService.getAll()  
    .then((response) => {  
      console.log(response.data);  
      setMovies(response.data.movies);  
    })  
    .catch((e) => {  
      console.log(e);  
    });  
};  
  
const retrieveRatings = () => {  
  MovieDataService.getRatings()  
    .then((response) => {  
      console.log(response.data);  
      setRatings(["All Ratings"].concat(response.data));  
    })  
    .catch((e) => {  
      console.log(e);  
    });  
};  
  
useEffect(() => {  
  retrieveMovies();  
  retrieveRatings();  
}, []);  
  
useEffect(() => {  
  retrieveMovies();  
}, [currentPage]);
```

2.3 + 2.5 Tạo 2 search form gồm tìm theo title, và tìm theo rating. Và hiện thực 2 phương thức findByTitle() và findByRating(). (Bước hiện thức ở dòng 104 và 117)

```
93 |  
94 |  
95 |  
96 |  
97 |  
98 |  
99 |  
100 |  
101 |  
102 |  
103 |  
104 |  
105 |  
106 |  
107 |  
108 |  
109 |  
110 |  
111 |  
112 |  
113 |  
114 |  
115 |  
116 |  
117 |  
118 |  
119 |  
120 |  
121 |  
122 |  
123 |
```

```
/* tạo form tìm kiếm */  
<Row>  
  <Col>  
    <Form.Group>  
      <Form.Control  
        type="text"  
        placeholder="Search by title"  
        value={searchTitle}  
        onChange={onChangeSearchTitle}  
      />  
    </Form.Group>  
    <Button variant="primary" type="button" onClick={findByTitle}>  
      Search  
    </Button>  
  </Col>  
  
  <Col>  
    <Form.Group>  
      <Form.Control as="select" onChange={onChangeSearchRating}>  
        {ratings.map((rating) => {  
          return <option value={rating}>{rating}</option>;  
        })}  
      </Form.Control>  
    </Form.Group>  
    <Button variant="primary" type="button" onClick={findByRating}>  
      Search  
    </Button>  
  </Col>  
</Row>  
/* tạo form tìm kiếm */
```

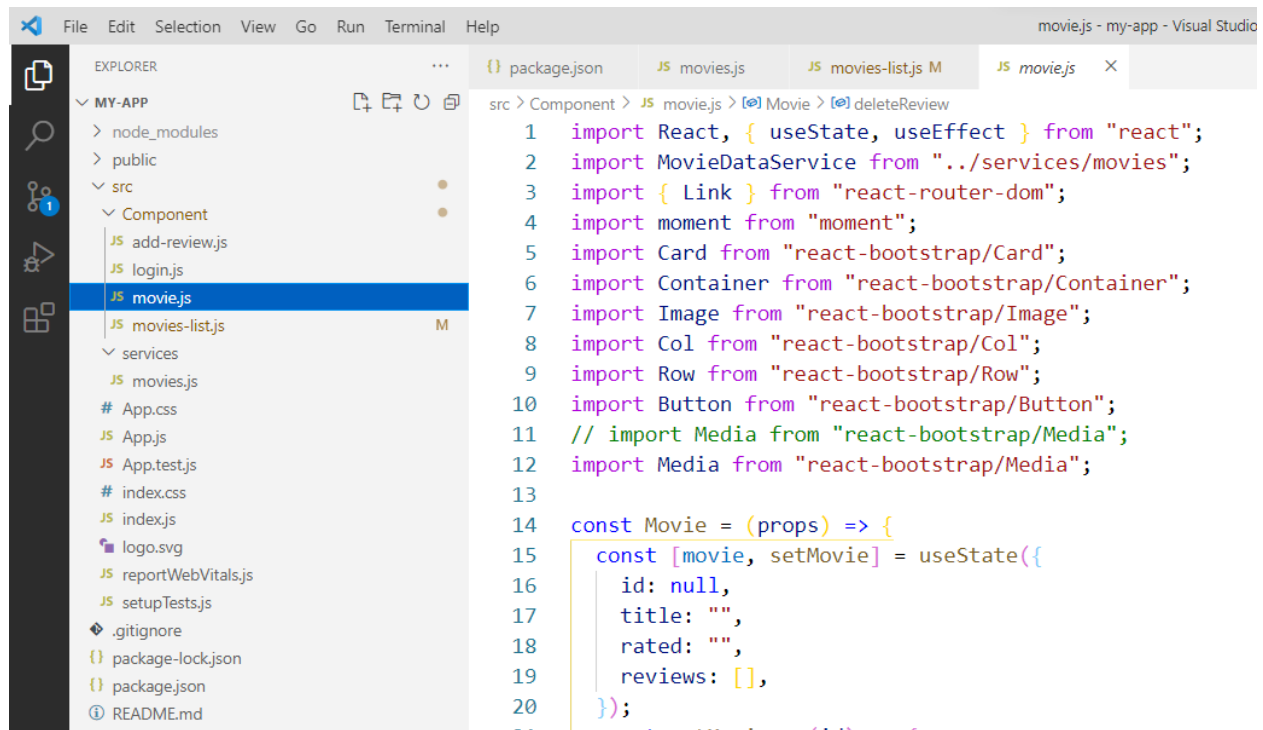
2.4 Hiện thị các movie bằng <Card> của React-bootstrap.

```
124 |  
125 |  
126 |  
127 |  
128 |  
129 |  
130 |  
131 |  
132 |  
133 |  
134 |  
135 |  
136 |  
137 |  
138 |  
139 |  
140 |  
141 |  
142 |
```

```
/* hiển thị phim */  
<Row>  
  {movies.map((movie) => {  
    return (  
      <Col>  
        <Card style={{ width: "18rem" }}>  
          <Card.Img src={movie.poster + "/100px180"} />  
          <Card.Body>  
            <Card.Title>{movie.title}</Card.Title>  
            <Card.Text>Rating: {movie.rating}</Card.Text>  
            <Card.Text>{movie.plot}</Card.Text>  
            <Link to={"/movies/" + movie._id}>View Reviews</Link>  
          </Card.Body>  
        </Card>  
      </Col>  
    );  
  })}  
</Row>  
/* hiển thị phim */
```

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

3.1 Thiết lập mã nguồn cho component Movie



```
File Edit Selection View Go Run Terminal Help moviejs - my-app - Visual Studio
```

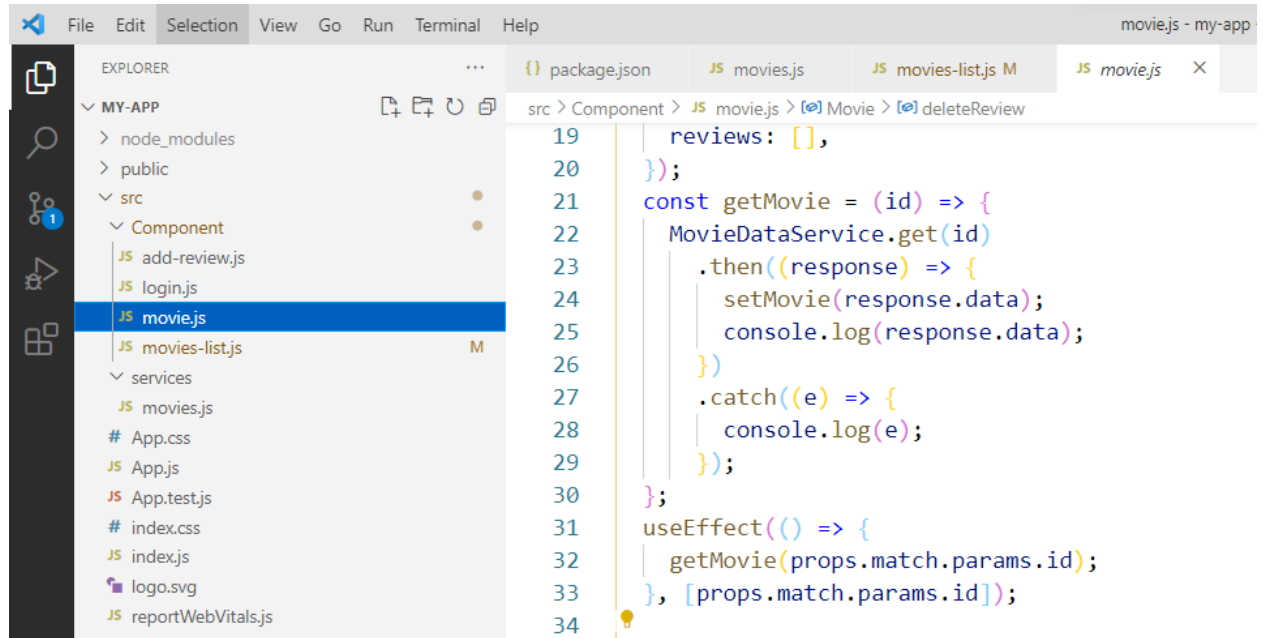
EXPLORER

- MY-APP
 - node_modules
 - public
 - src
 - Component
 - add-review.js
 - login.js
 - movie.js
 - movies-list.js
 - services
 - movies.js
 - App.css
 - App.js
 - App.test.js
 - index.css
 - index.js
 - logo.svg
 - reportWebVitals.js
 - setupTests.js
 - .gitignore
 - package-lock.json
 - package.json
 - README.md

src > Component > JS movie.js > deleteReview

```
1 import React, { useState, useEffect } from "react";
2 import MovieDataService from "../services/movies";
3 import { Link } from "react-router-dom";
4 import moment from "moment";
5 import Card from "react-bootstrap/Card";
6 import Container from "react-bootstrap/Container";
7 import Image from "react-bootstrap/Image";
8 import Col from "react-bootstrap/Col";
9 import Row from "react-bootstrap/Row";
10 import Button from "react-bootstrap/Button";
11 // import Media from "react-bootstrap/Media";
12 import Media from "react-bootstrap/Media";
13
14 const Movie = (props) => {
15   const [movie, setMovie] = useState({
16     id: null,
17     title: "",
18     rated: "",
19     reviews: [],
20   });
```

3.2 Xây dựng mã nguồn cho phương thức getMovie()



```
File Edit Selection View Go Run Terminal Help moviejs - my-app
```

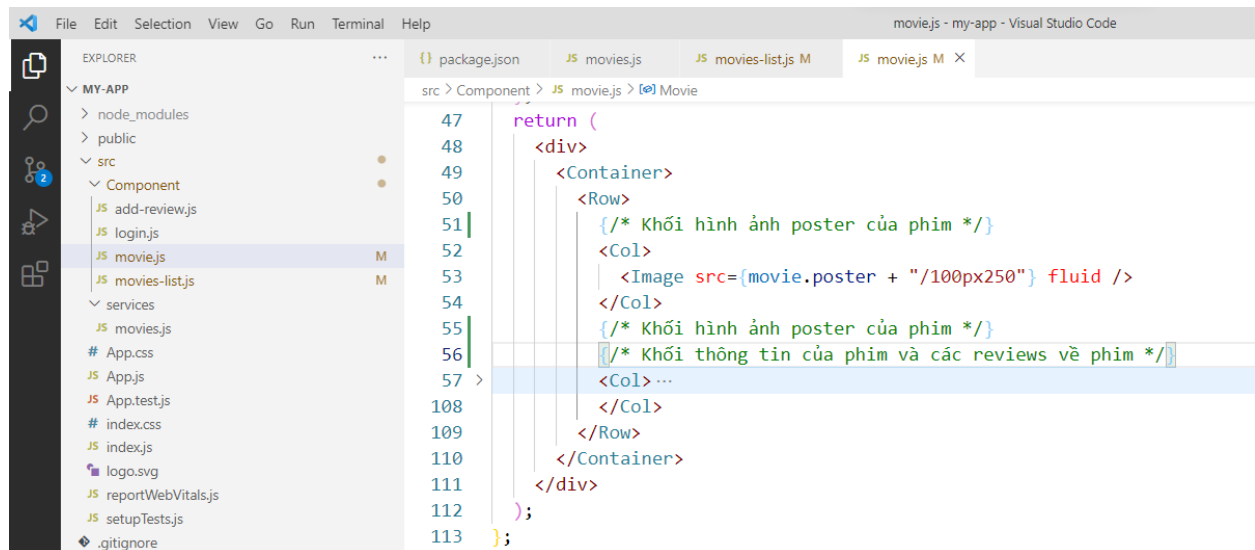
EXPLORER

- MY-APP
 - node_modules
 - public
 - src
 - Component
 - add-review.js
 - login.js
 - movie.js
 - movies-list.js
 - services
 - movies.js
 - App.css
 - App.js
 - App.test.js
 - index.css
 - index.js
 - logo.svg
 - reportWebVitals.js
 - setupTests.js

src > Component > JS movie.js > deleteReview

```
19   reviews: [],
20   });
21   const getMovie = (id) => {
22     MovieDataService.get(id)
23       .then((response) => {
24         setMovie(response.data);
25         console.log(response.data);
26       })
27       .catch((e) => {
28         console.log(e);
29       });
30   };
31   useEffect(() => {
32     getMovie(props.match.params.id);
33   }, [props.match.params.id]);
34
```

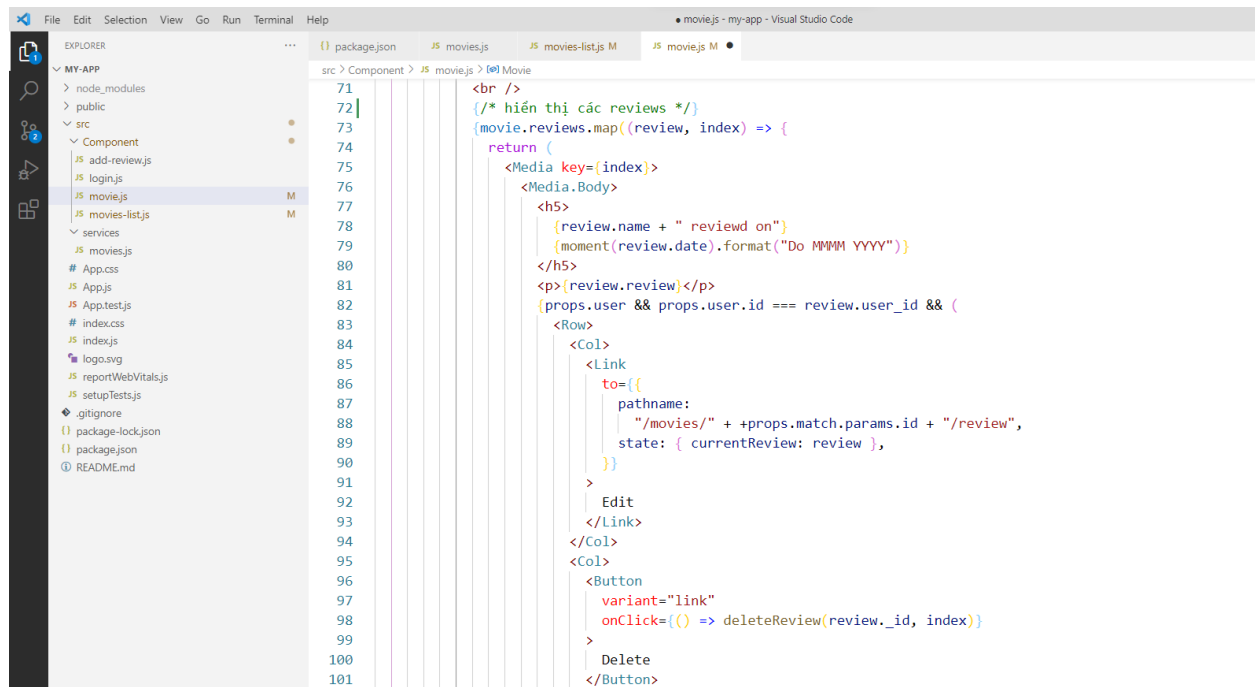
3.3 Xây dựng giao diện trang review phim:



```
47 return (  
48   <div>  
49     <Container>  
50       <Row>  
51         { /* Khối hình ảnh poster của phim */ }  
52         <Col>  
53           <Image src={movie.poster + "/100px250"} fluid />  
54         </Col>  
55         { /* Khối hình ảnh poster của phim */ }  
56         { /* Khối thông tin của phim và các reviews về phim */ }  
57       <Col> ...  
108     </Col>  
109   </Row>  
110 </Container>  
111 </div>  
112 );  
113 };
```

Bài 4: Hiển thị danh sách review tương ứng cho từng phim

4.1 + 4.2 Viết đoạn mã nguồn JSX cho phép hiển thị danh sách review cho phim. Và điều chỉnh lại cách hiển thị giờ với momentjs.(dòng 79)



```
71 <br />  
72 { /* hiển thị các reviews */ }  
73 {movie.reviews.map((review, index) => {  
74   return (  
75     <Media key={index}>  
76       <Media.Body>  
77         <h5>  
78           {review.name + " reviewd on"}  
79           {moment(review.date).format("Do MMMM YYYY")}  
80         </h5>  
81         <p>{review.review}</p>  
82         {props.user && props.user.id === review.user_id && (  
83           <Row>  
84             <Col>  
85               <Link  
86                 to={{  
87                   pathname:  
88                     "/movies/" + props.match.params.id + "/review",  
89                   state: { currentReview: review },  
90                 }}  
91             >  
92               Edit  
93             </Link>  
94           </Col>  
95           <Col>  
96             <Button  
97               variant="link"  
98               onClick={() => deleteReview(review._id, index)}  
99             >  
100               Delete  
101             </Button>
```

Kết quả :

Trang movies:

Võ Hồng Hòa 19521522 [Movies](#) [Login](#)

Search by title

Search

Blacksmith Scene

Rating: UNRATED

Three men hammer on an anvil and pass a bottle of beer around.

[View Reviews](#)

No. 201
EDISON FILM
COPYRIGHTED 1903
THE GREAT TRAIN ROBBERY

The Great Train Robbery

Rating: TV-G

A group of bandits stage a brazen train hold-up, only to find a

All Ratings

Search

A Corner in Wheat

Rating: G

A greedy tycoon decides, on a whim, to corner the world market in wheat. This doubles the price of bread, forcing the grain's producers into charity lines and further into poverty. The film...

[View Reviews](#)

The Land Beyond the Sunset

Rating: UNRATED

A young boy, oppressed by his mother, goes on an outing in the country with a social welfare group where he does to dream of a land

Trang reviews phim:

Võ Hồng Hòa 19521522 [Movies](#) [Login](#)

No. 201
EDISON FILM
COPYRIGHTED 1903
THE GREAT TRAIN ROBBERY

The Great Train Robbery

A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.

Reviews

Võ Hồng Hòa 19521522 reviewd on7th May 2022
bad!

ie213 reviewd on7th May 2022
phim khá hay ha

ie213 reviewd on9th May 2022
hay hayahaya ahyauahaha