

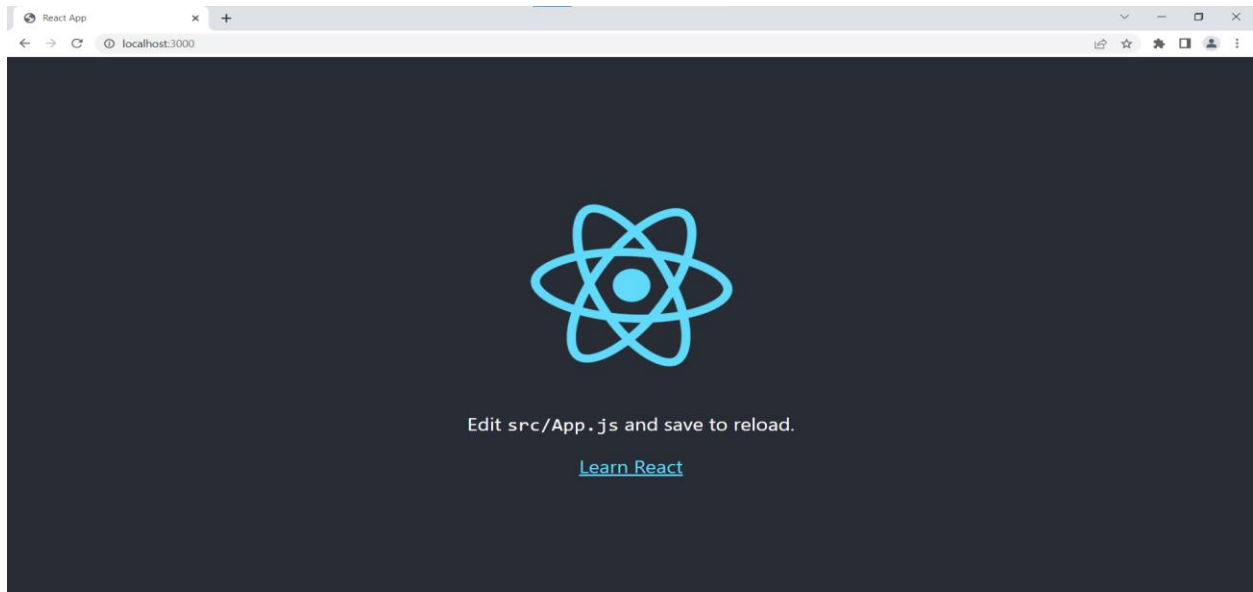
BÀI THỰC HÀNH 4

THIẾT LẬP FRONTEND VỚI REACTJS

Bài 1: Thiết lập nơi làm việc với frontend của dự án.

1.1 Tạo template frontend với React trong thư mục Movie Review (ứng dụng minh họa).

Chạy ứng dụng lên với câu lệnh `npm start` để thấy được kết quả như hình (nhớ cd vào thư mục frontend vừa tạo bên trên):



1.2 Cài đặt một số package hỗ trợ xây dựng dự án:

- Bootstrap: hỗ trợ xây dựng UI.
- React router dom: hỗ trợ định tuyến.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS F:\hk6\ie213-web\TH\lab2\20520510-movie-reviews\frontend> npm install react-bootstrap bootstrap
added 19 packages, and audited 1500 packages in 12s
233 packages are looking for funding
  run `npm fund` for details
6 high severity vulnerabilities
To address all issues (including breaking changes), run:
  npm audit fix --force
Run `npm audit` for details.
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Run `npm audit` for details.
PS F:\hk6\ie213-web\TH\lab2\20520510-movie-reviews\frontend> npm install react-router-dom

added 3 packages, and audited 1503 packages in 5s

233 packages are looking for funding
  run `npm fund` for details

6 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

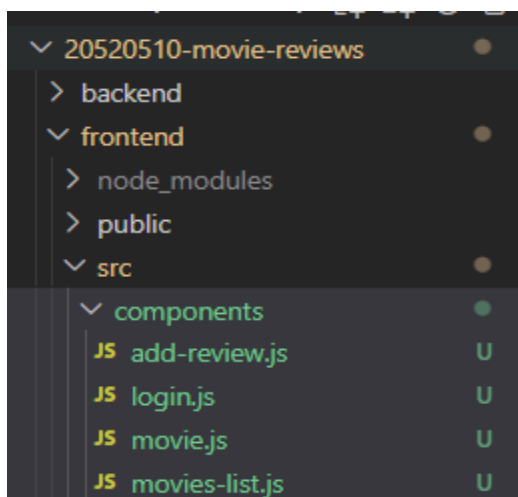
Run `npm audit` for details.
```

Bài 2: Xây dựng Navigation Header bar cho ứng dụng.

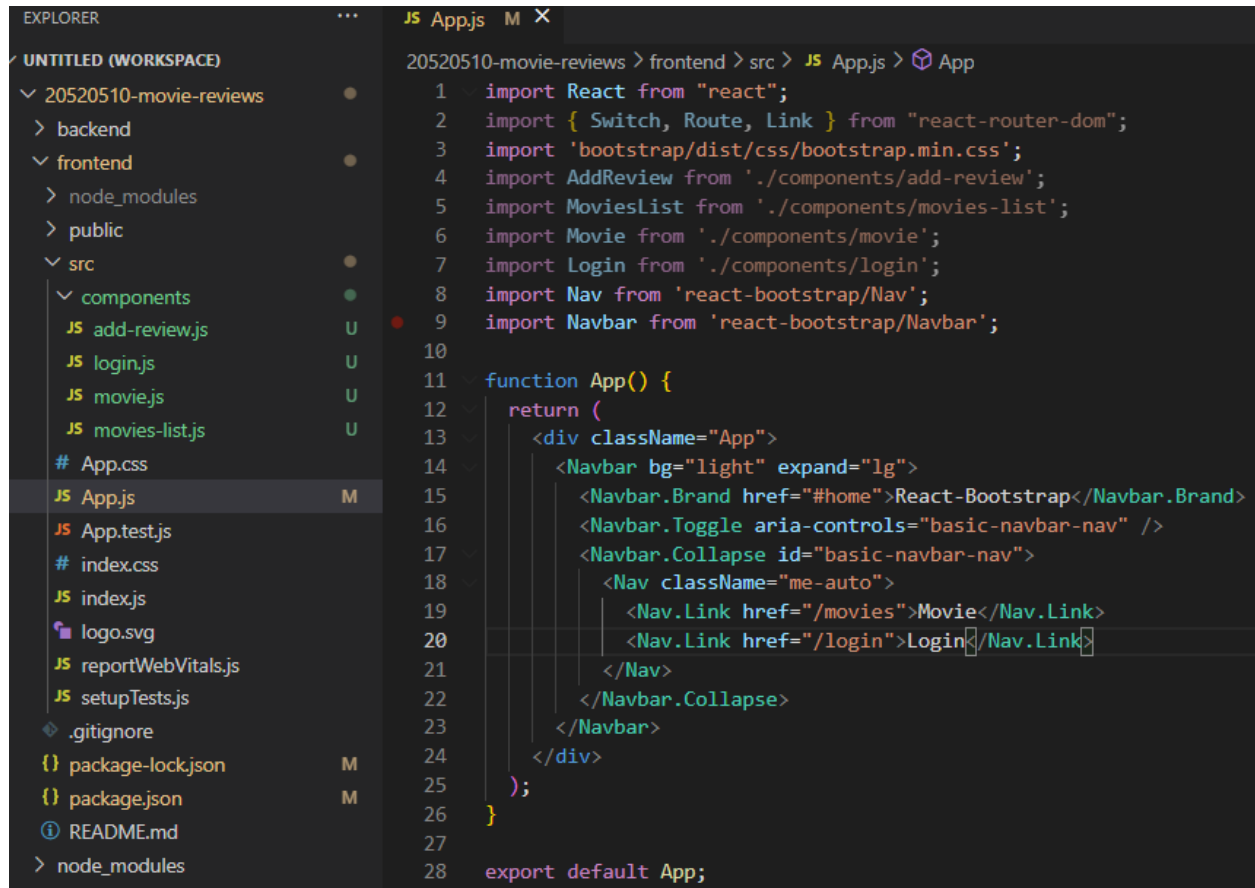
2.1 Navigation bar sẽ giúp người dùng định tuyến tới các nội dung của ứng dụng, do đó ta cần xây dựng các component như:

- movies-list: hiển thị thông tin danh sách phim.
- movie: hiển thị phim với các review.
- add-review: hỗ trợ thêm review cho khách.
- login: trang đăng nhập cho khách.

Lưu ý: các component này sẽ được tạo trong thư mục components (được tạo trong thư mục frontend), và lần lượt import vào tệp tin App.js để sử dụng về sau.



2.2 Lấy Navbar Component từ React-Bootstrap và đưa vào trong phần mã nguồn JSX của function App() trong tệp tin App.js.



```
1 import React from "react";
2 import { Switch, Route, Link } from "react-router-dom";
3 import 'bootstrap/dist/css/bootstrap.min.css';
4 import AddReview from './components/add-review';
5 import MoviesList from './components/movies-list';
6 import Movie from './components/movie';
7 import Login from './components/login';
8 import Nav from 'react-bootstrap/Nav';
9 import Navbar from 'react-bootstrap/Navbar';
10
11 function App() {
12   return (
13     <div className="App">
14       <Navbar bg="light" expand="lg">
15         <Navbar.Brand href="#home">React-Bootstrap</Navbar.Brand>
16         <Navbar.Toggle aria-controls="basic-navbar-nav" />
17         <Navbar.Collapse id="basic-navbar-nav">
18           <Nav className="me-auto">
19             <Nav.Link href="/movies">Movie</Nav.Link>
20             <Nav.Link href="/login">Login</Nav.Link>
21           </Nav>
22         </Navbar.Collapse>
23       </Navbar>
24     </div>
25   );
26 }
27
28 export default App;
```

2.3 Điều chỉnh một số thông tin:

- Tên logo: Movie Reviews.
- Liên kết thứ nhất thay Home thành Movies.
- Liên kết thứ hai thay Link thành trạng thái Login/Logout của người dùng.

Lưu ý: mã nguồn ở đây cần sử dụng React hook useState để lưu giữ và thay đổi trạng thái đăng nhập của người dùng, ví dụ: [user, setUser] = useState(null);

```
11 function App() {
12   const [user, setUser] = React.useState(null);
13   async function login(user = null) { // default user to null
14     setUser(user);
15   }
16   async function logout() {
17     setUser(null);
18   }
19   return (
20     <div className="App">
21       <Navbar bg="light" expand="lg">
22         <Navbar.Brand href="#home">Movie Reviews</Navbar.Brand>
23         <Navbar.Toggle aria-controls="basic-navbar-nav" />
24         <Navbar.Collapse id="basic-navbar-nav">
25           <Nav className="me-auto">
26             <Nav.Link>
27               <Link to={"/movies"}>Movies</Link>
28             </Nav.Link>
29             <Nav.Link>
30               {user ? (
31                 <a onClick={logout}>Logout User</a>
32               ) : (
33                 <Link to={"/login"}>Login</Link>
34               )}
35             </Nav.Link>
36           </Nav>
37         </Navbar.Collapse>
38       </Navbar>
39     </div>
40   );
41 }
```

Bài 3: Thiết lập các định tuyến cho các component vừa tạo ở trên.

3.1 Trong tệp tin App.js cần sử dụng thẻ <Switch> hoặc <Routes> (import từ react-router-dom) để định tuyến cho 4 component tạo ở bài 2.1.

3.2 Các định tuyến bao gồm:

- “/”: đến component MoviesList.
- “/movies/:id/review”: đến component AddReview.
- “/movies/:id”: đến component Movie.
- “/login”: đến component Login.

The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays the file structure of a project named '20520510-movie-reviews'. The structure includes a 'frontend' folder containing 'node_modules', 'public', and 'src'. Inside 'src', there is a 'components' folder with files like 'add-review.js', 'login.js', 'movie.js', and 'movies-list.js'. The main 'App.js' file is highlighted. On the right, the code editor shows the content of 'App.js'. It features a `<Switch>` component with four routes: a root route for `/` and `/movies` using `MoviesList`; a route for `/movies/:id/review` using `AddReview`; a route for `/movies/:id/` using `Movie`; and a `/login` route using `Login`. The code is wrapped in a `<div>` and ends with a closing semicolon.