**BÀI THỰC HÀNH 6**

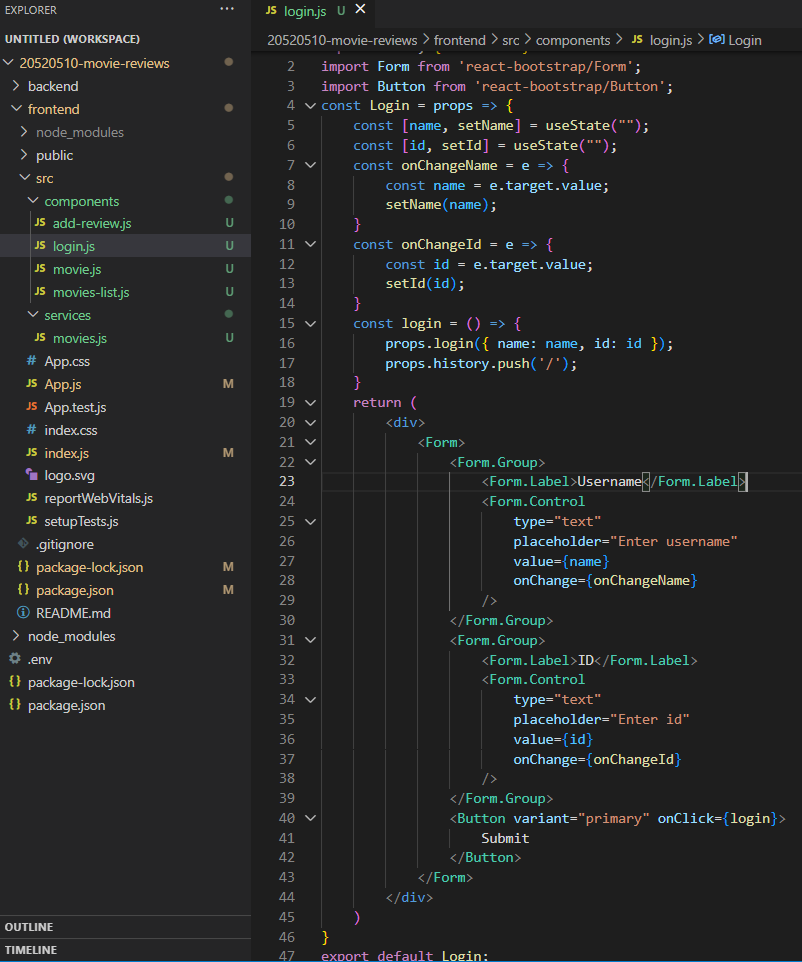
**XÂY DỰNG FRONTEND VỚI REACTJS (tt)**

Bài 1: Thêm và Sửa Review.

1.1 Tạo login component.

Yêu cầu khi thiết lập login component thì khi người dùng đăng nhập thành công, họ sẽ thấy được các chức năng như Edit và Delete review của chính họ.

Sau khi login thành công, người dùng sẽ được redirect về lại trang Home.



1.2 Thêm review

Tạo các biến như hướng dẫn sau:

- Biến editing sẽ có giá trị true khi component đang ở chế độ Editing.

- Ngược lại là chế độ thêm review.

- Biến trạng thái review được thiết lập thông qua biến initialReviewState.

- Trong chế độ editing, initialReviewState sẽ được thiết lập có nội dung text.

- Biến trạng thái submitted để theo dấu nếu như có một review được thêm mới.

Tạo các hàm phù hợp:

- Hàm onChangeReview() theo vết khi người dùng thêm giá trị review dưới form.

- Hàm saveReview() được gọi khi nút submit được nhấn.

- Trong hàm này, đầu tiên ta tạo một object tên data chứa các giá trị thuộc tính của review.

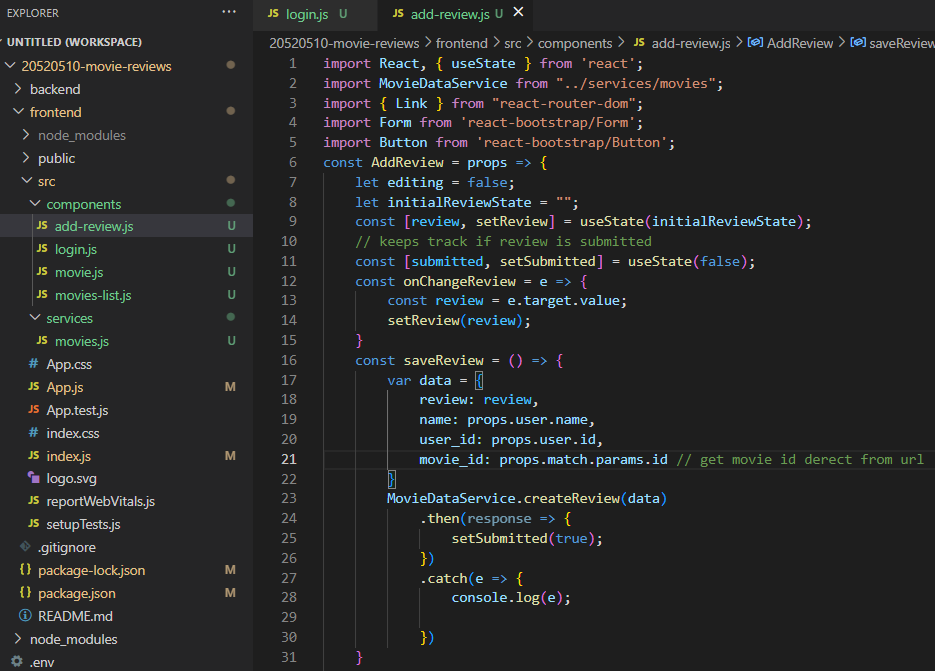
- 2 giá trị name và user\_id sẽ nhận từ props được gửi từ App.js.

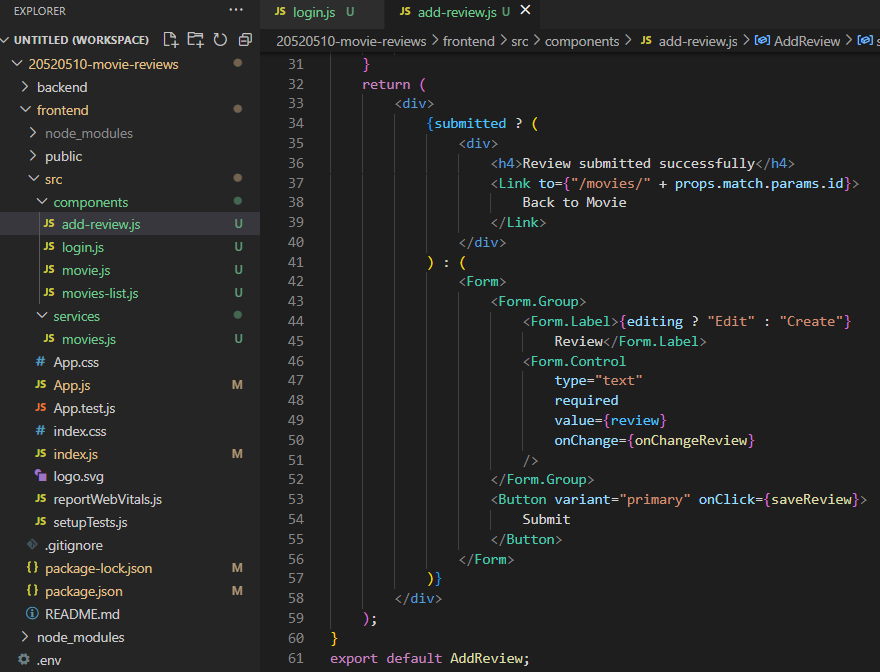
- Lấy movie\_id trực tiếp từ url (xem lại nội dung movie.js).

- Sau đó gọi hàm createReview(data) trong MovieDataServiece.

- Định tuyến này gọi tới ReviewsController trong backend và gọi apiPostReview(), trích xuất data từ request’s body params.

Phần return() chứa nội dung JSX giúp hiển thị và xử lý tính năng thêm review:





1.3 Sửa review

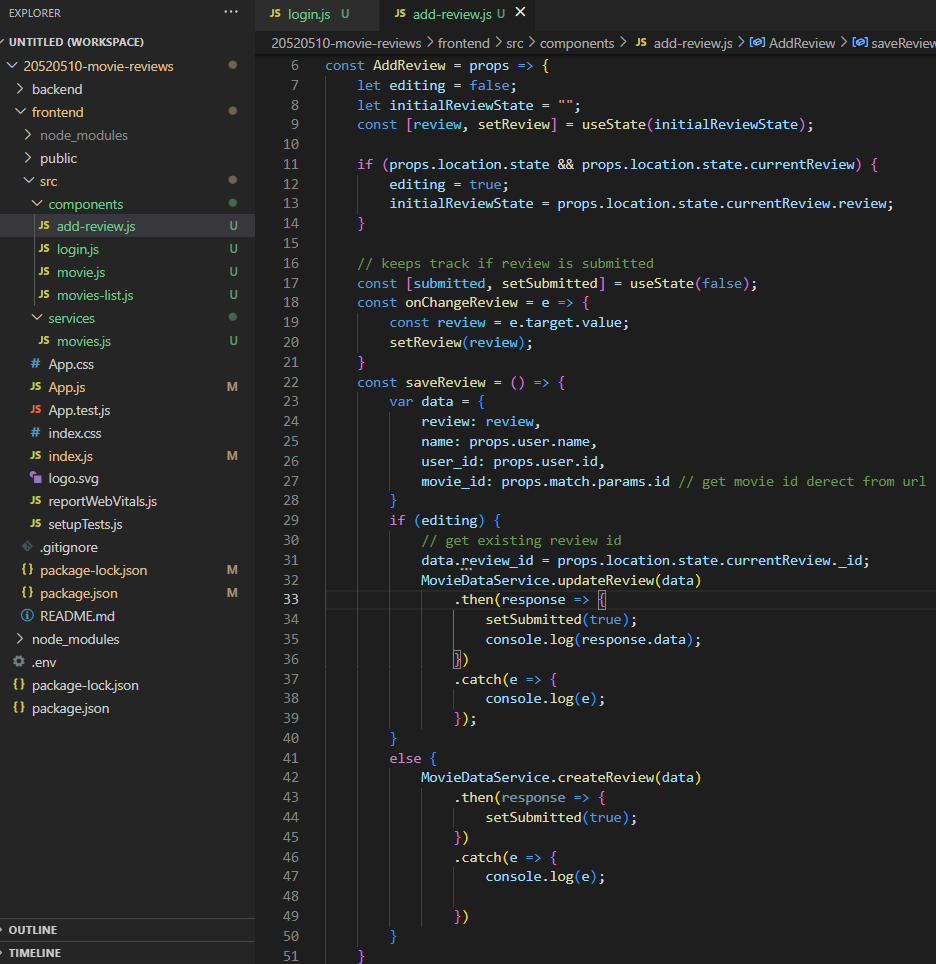
Viết mã nguồn thực hiện các việc sau:

- Đầu tiên, kiểm tra trạng thái truyền vào cho AddReview (xem lại tệp tin movie.js sẽ thấy prop state).

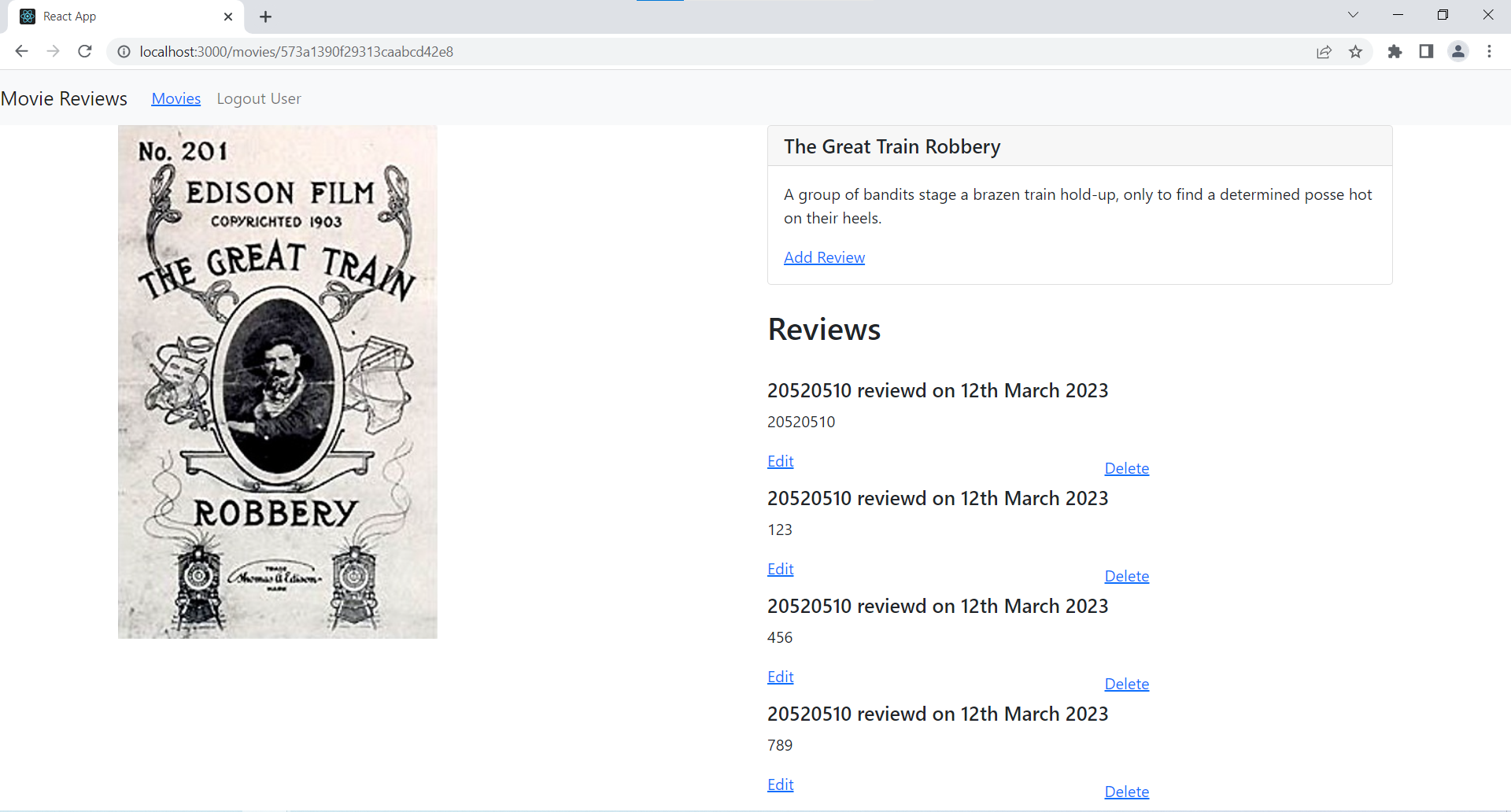
- Nếu state được truyền vào chứa thuộc tính currentReview thì chuyển editing thành true và initialReviewState thành currentReview.review.

- Nếu editing là true thì gọi updateReview() trong MovieDataService.

- Phương thức apiUpdateReview() trong ReviewsController ở backend sẽ được gọi, tương tự apiPostReview.



Kiểm tra chạy ứng dụng sẽ cho phép cập nhật lại review:

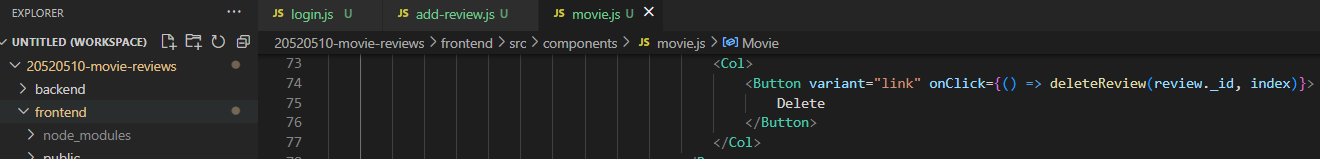


Bài 2: Xoá review

Thêm mã nguồn vào movie.js để xử lý phần xoá review:

- Trong nút delete, ta truyền review id và index chúng ta nhận được từ movie.reviews.map()

vào phương thức deleteReview().

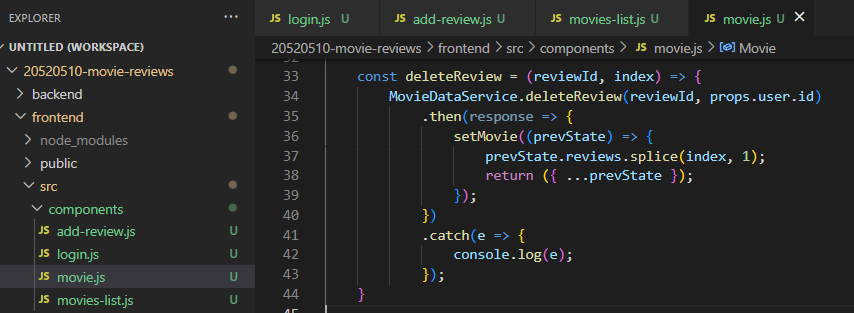


- Trong phương thức deleteReview(), ta gọi hàm deleteReview() trong MovieDataService.

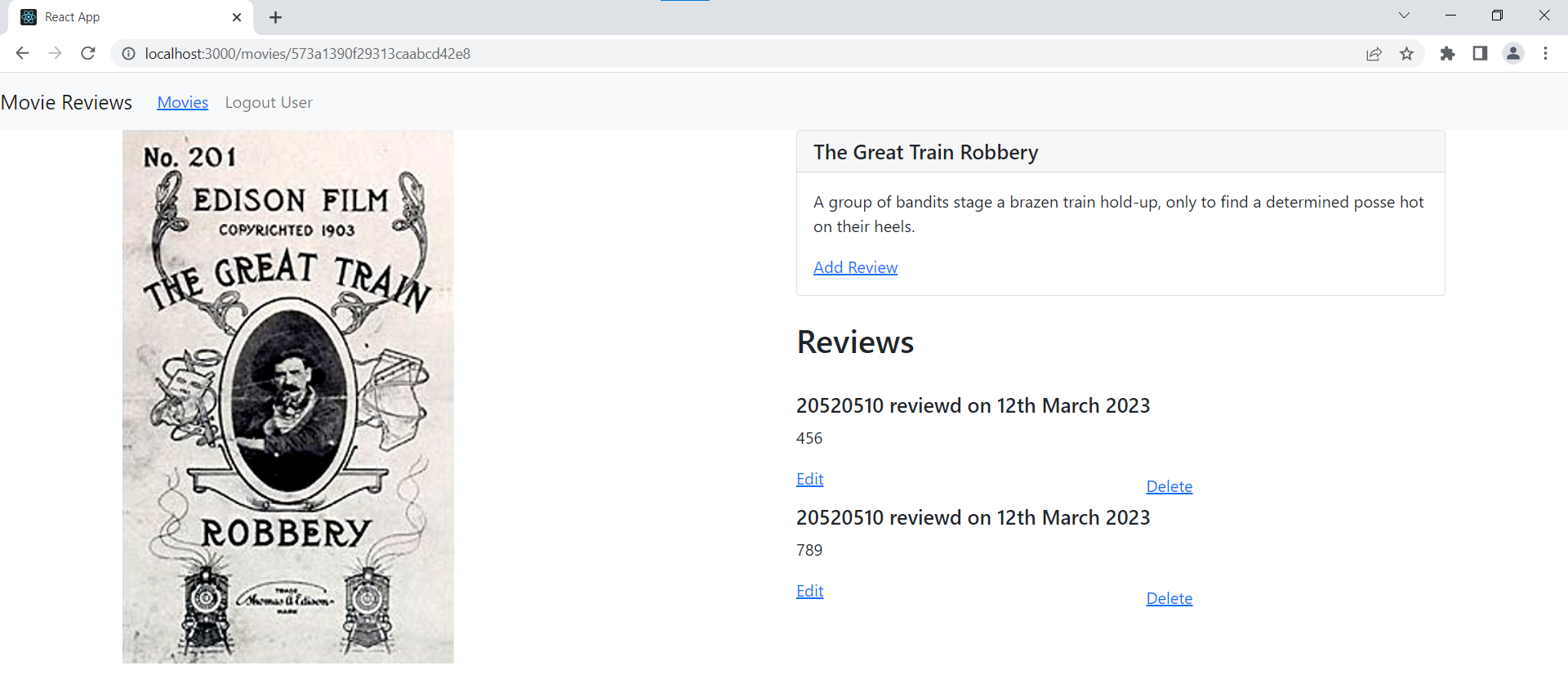
- Sau đó, tạo một callback .then() sẽ được gọi sau khi deleteReview() được gọi xong.

- Trong callback này, ta lấy mảng các reviews trong trạng thái hiện tại. Sau đó cung cấp index của review sẽ bị xoá cho phương thức splice() để xoá nó đi.

- Cuối cùng là cập nhật lại mảng reviews như là trạng thái mới.



Chạy lại ứng dụng -> log in -> chọn một movie cụ thể có review mà mình đã đăng.



Bài 3: Lấy dữ liệu cho trang tiếp theo

3.1 getAll()

Trong component movies-list.js thêm mã nguồn để thực hiện yêu cầu.

- Thêm 2 biến trạng thái currentPage và entriesPerPage.

- Thiết lập 2 biến trạng thái này trong phương thức retrieveMovies().

- Thêm một useEffect hook:

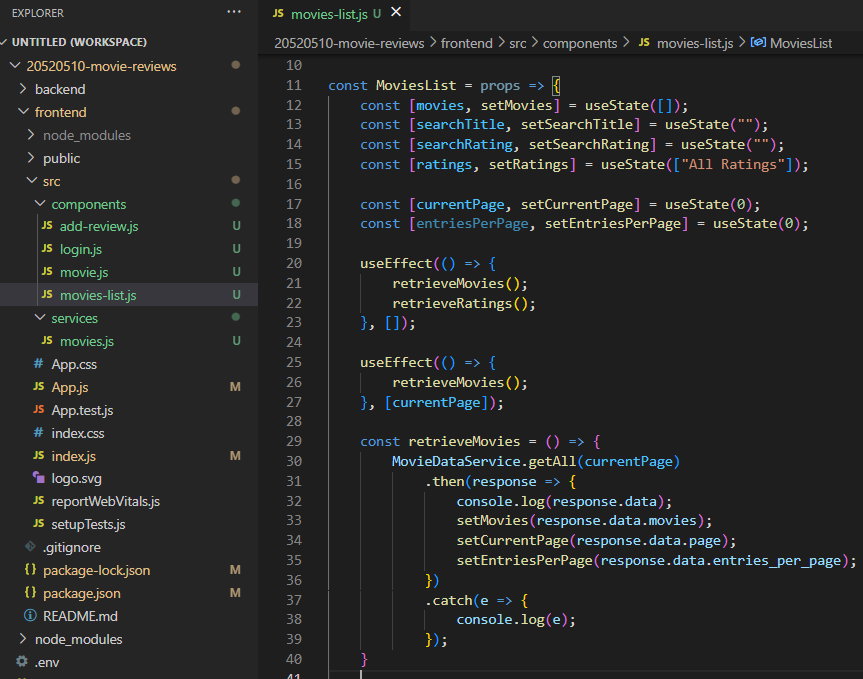
useEffect(() => {

retrieveMovies();

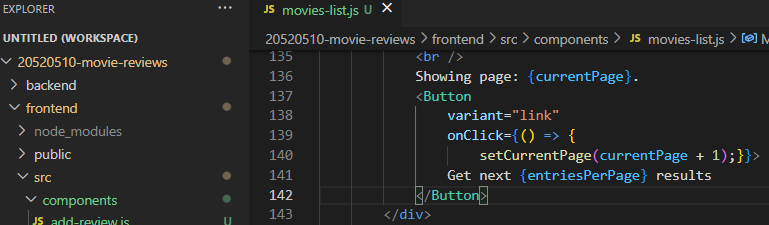
}, [currentPage]);

- Biến trạng thái currentPage được truyền trong tham số thứ 2 (trong mảng) nhằm mục đích khi giá trị của nó thay đổi thì hàm retrieveMovies() sẽ được gọi.

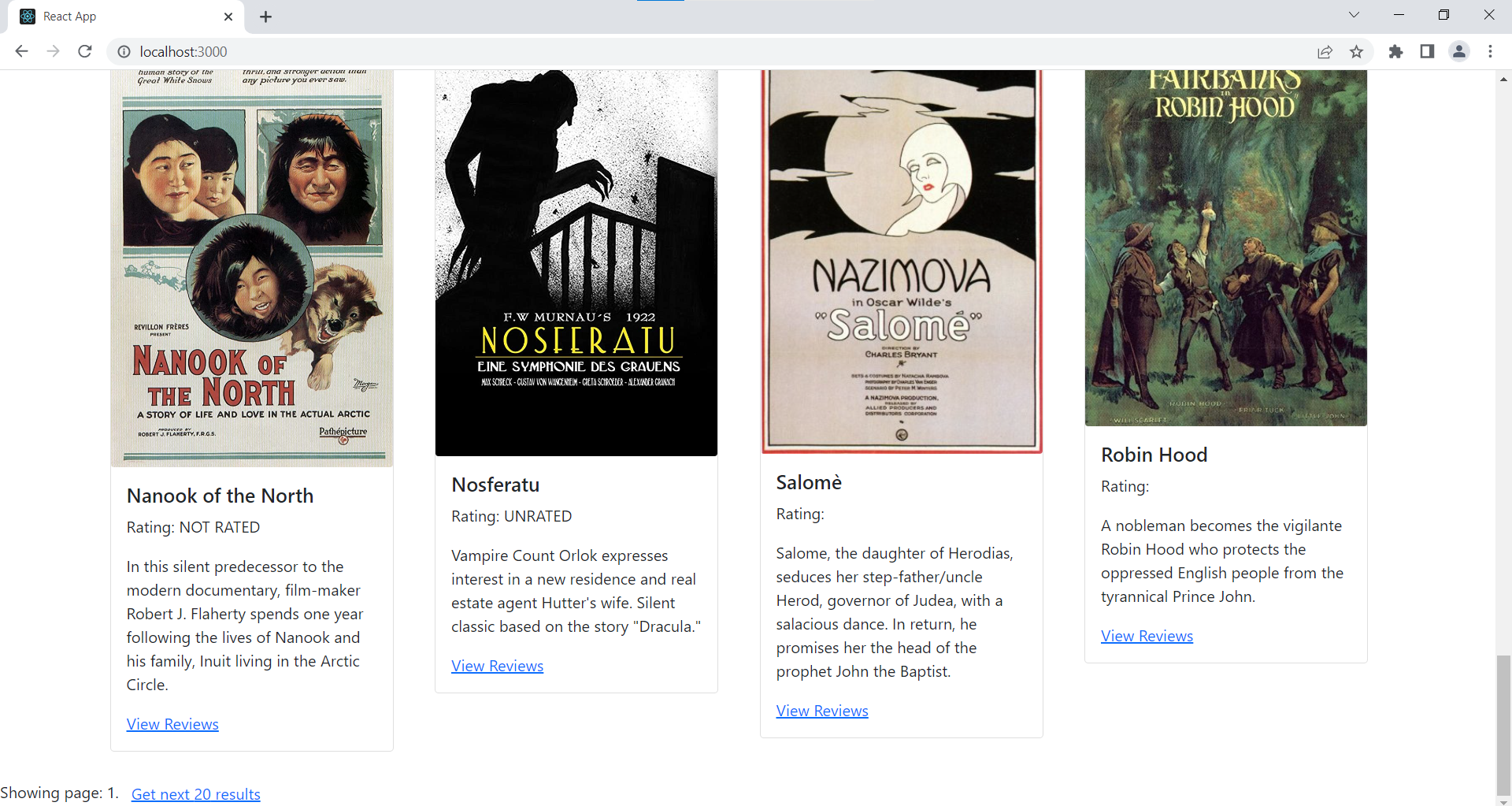
- Quan trọng: nhớ truyền currentPage vào lời gọi MovieDataService.get().



- Đồng thời, thêm đoạn mã xử lý vào hàm return().



Kết quả:



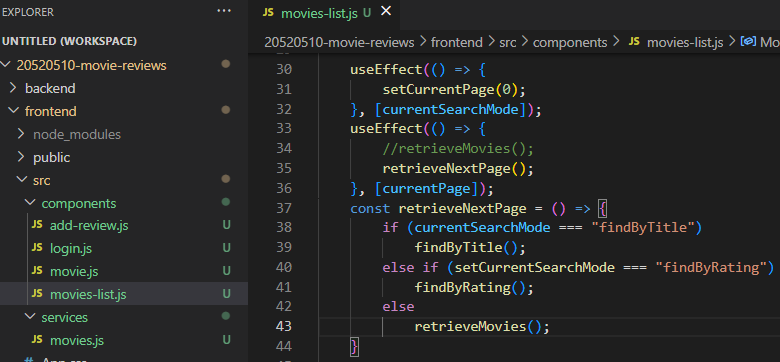
3.2 find()

Trong movies-list.js ta điều chỉnh mã nguồn theo các yêu cầu sau:

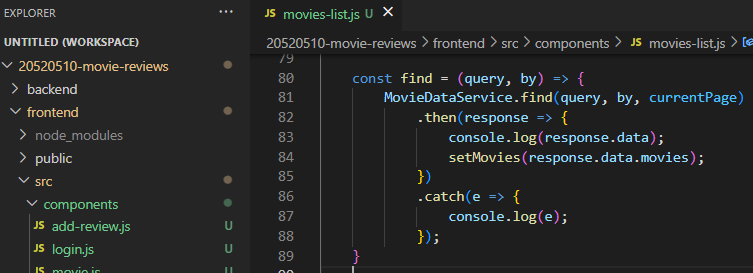
- Đầu tiên, tạo biến trạng thái currentSearchMode để nhận 2 giá trị “findByTitle” hoặc “findByRating”.

- Sử dụng một useEffect() để khi nào currentSearchMode thay đổi thì thiết lập lại currentPage về 0.

- Tạo phương thức mới retrieveNextPage() -> dựa vào currentSearchMode để gọi các hàm tương ứng.



- Thêm tham số currentPage vào trong lời gọi MovieDataService.find().

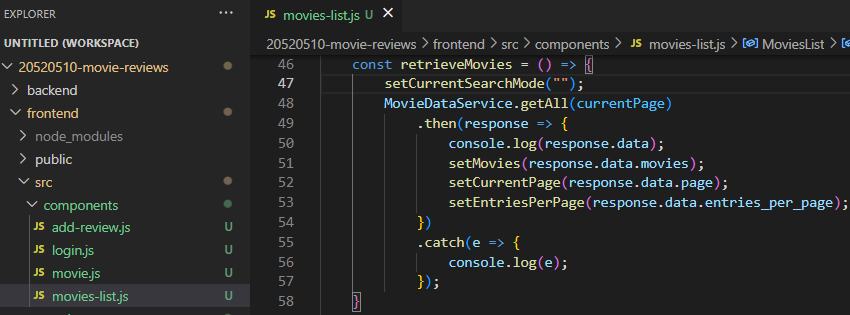


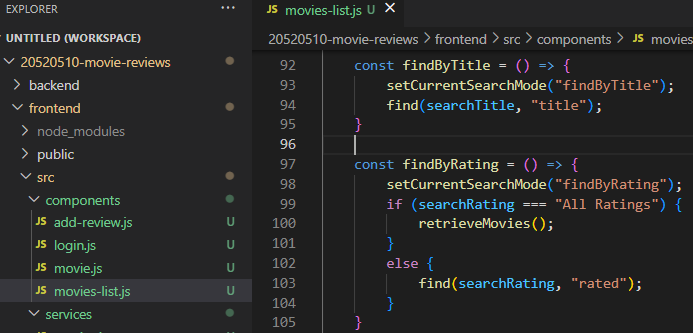
- Lần lượt thêm các dòng mã lệnh setCurrentSearchMode() tương ứng vào 3 phương thức điều khiển:

- retrieveMovies();

- findByTitle();

- findByRating();





Kết quả:

