

# Bài Tập Cuối Chương

## Đề bài: Xây dựng ứng dụng "Library Membership Manager"

### Mô tả ứng dụng

Học viên sẽ xây dựng một ứng dụng quản lý thư viện với hai thực thể chính: **sách (books)** và **thành viên (members)**. Ứng dụng có các tính năng:

- **Quản lý sách:** Hiển thị danh sách sách, thêm, xóa, đánh dấu trạng thái mượn, và lọc sách theo trạng thái ("All", "Borrowed", "Available").
- **Quản lý thành viên:** Hiển thị danh sách thành viên, thêm, xóa thành viên, và liên kết thành viên với sách đã mượn.
- Điều hướng giữa các trang: Home, Book Detail, Member Detail, NotFound (dùng React Router DOM v7).
- Quản lý dữ liệu toàn cục bằng **Redux Toolkit** với **hai slice**: `bookSlice` và `memberSlice`.

### Cấu trúc thư mục

```
src/
├── components/           # Các component dùng chung
│   ├── NavBar/          # Thanh điều hướng
│   │   ├── NavBar.jsx
│   │   └── NavBar.css
│   ├── BookItem/        # Hiển thị từng cuốn sách
│   │   ├── BookItem.jsx
│   │   └── BookItem.css
│   ├── BookList/        # Danh sách sách
│   │   ├── BookList.jsx
│   │   └── BookList.css
│   └── MemberItem/      # Hiển thị từng thành viên
```

```

| | | └─ MemberItem.jsx
| | | └─ MemberItem.css
| | └─ MemberList/      # Danh sách thành viên
| |   | └─ MemberList.jsx
| |   | └─ MemberList.css
| | └─ FilterButtons/   # Nút lọc danh sách sách
| |   | └─ FilterButtons.jsx
| |   | └─ FilterButtons.css
└─ screens/            # Các màn hình chính
  | └─ Home/           # Trang chủ
  |   | └─ Home.jsx
  | └─ BookDetail/     # Chi tiết sách
  |   | └─ BookDetail.jsx
  | └─ MemberDetail/   # Chi tiết thành viên
  |   | └─ MemberDetail.jsx
  └─ NotFound/         # Trang lỗi
    | └─ NotFound.jsx
└─ redux/             # Quản lý Redux Toolkit
  | └─ slices/         # Các slice
  |   | └─ bookSlice.js # Slice quản lý sách
  |   | └─ memberSlice.js # Slice quản lý thành viên
  └─ store.js          # Cấu hình store
└─ App.jsx            # File chính kết nối routes
└─ main.jsx           # Entry point

```

## Yêu cầu chi tiết

### 1. Thiết lập môi trường

- Sử dụng Vite để khởi tạo dự án:

```
npm create vite@latest library-membership-manager -- --template react
```

- Cài đặt các thư viện cần thiết:

```
npm install react-router-dom @reduxjs/toolkit react-redux
```

- Tạo cấu trúc thư mục như trên.

## 2. Chức năng cụ thể

### `src/redux/slices/bookSlice.js`

- **Mô tả:** Quản lý danh sách sách và trạng thái lọc.
- **Yêu cầu:**
  - **Initial State:**
    - `books`: Mảng sách, mỗi sách có các thuộc tính: `id`, `title`, `borrowed` (boolean), `borrowedBy` (ID thành viên mượn, nếu có), `addedAt`.
    - `filter`: Trạng thái lọc ("all", "borrowed", "available").
  - **Reducers:**
    - `addBook`: Thêm sách mới vào mảng `books`.
    - `removeBook`: Xóa sách theo `id`.
    - `toggleBorrowed`: Chuyển đổi trạng thái mượn ( `borrowed` ), nhận payload là `{ bookId, memberId }` để cập nhật `borrowedBy` (hoặc `null` nếu trả sách).
    - `setFilter`: Cập nhật trạng thái lọc.
  - **Export:** Action creators và reducer.

### `src/redux/slices/memberSlice.js`

- **Mô tả:** Quản lý danh sách thành viên.
- **Yêu cầu:**
  - **Initial State:**
    - `members`: Mảng thành viên, mỗi thành viên có các thuộc tính: `id`, `name`, `joinedAt`.
  - **Reducers:**
    - `addMember`: Thêm thành viên mới vào mảng `members`.

- `removeMember` : Xóa thành viên theo `id` , chỉ cho phép xóa nếu thành viên không mượn sách nào ( `borrowedBy` không trỏ đến thành viên này trong bất kỳ sách nào).

◦ **Export:** Action creators và reducer.

### src/redux/store.js

- **Mô tả:** Thiết lập Redux Store.
- **Yêu cầu:**
  - Sử dụng `configureStore` để kết hợp hai reducer từ `bookSlice` và `memberSlice` .

### src/App.jsx

- **Mô tả:** File chính của ứng dụng.
- **Yêu cầu:**
  - Hiển thị `<NavBar />` ở cấp cao nhất (ngoài `<Routes>` để luôn xuất hiện trên mọi trang).
  - Thiết lập các route:
    - `/` : Trang `Home` .
    - `/book/:id` : Trang `BookDetail` .
    - `/member/:id` : Trang `MemberDetail` .
    - : Trang `NotFound` .
  - Wrap toàn bộ ứng dụng trong `<Provider>` từ `react-redux` với Redux Store.

### src/components/NavBar/NavBar.jsx

- **Mô tả:** Thanh điều hướng.
- **Yêu cầu:**
  - Chứa liên kết đến trang chủ ( `/` ).
  - Áp dụng style cơ bản từ `NavBar.css` .

### src/screens/Home/Home.jsx

- **Mô tả:** Trang chính quản lý sách và thành viên.

- **Yêu cầu:**

- Input để thêm sách mới (dùng `useState`), nút "Add Book" để dispatch action `addBook`.
- Input để thêm thành viên mới (dùng `useState`), nút "Add Member" để dispatch action `addMember`.
- Gọi `<FilterButtons />` để lọc sách.
- Gọi `<BookList />` và `<MemberList />` để hiển thị danh sách.
- Dữ liệu lấy từ Redux Store qua `useSelector`.

### `src/components/BookList/BookList.jsx`

- **Mô tả:** Hiển thị danh sách sách đầy đủ.

- **Yêu cầu:**

- Lấy `books` và `filter` từ Redux Store.
- Lọc danh sách sách dựa trên `filter` ("all", "borrowed", "available").
- Map qua `books` để render `<BookItem />`.
- Áp dụng style từ `BookList.css`.

### `src/components/BookItem/BookItem.jsx`

- **Mô tả:** Hiển thị từng cuốn sách trong danh sách.

- **Yêu cầu:**

- Tiêu đề sách là liên kết đến `/book/:id` (dùng `Link`).
- Dropdown hoặc select để chọn thành viên mượn sách (danh sách từ `memberSlice`, bao gồm tùy chọn "Available" để trả sách), dispatch `toggleBorrowed` khi thay đổi.
- Nút "Delete" để dispatch `removeBook`.
- Áp dụng style từ `BookItem.css`.

### `src/components/MemberList/MemberList.jsx`

- **Mô tả:** Hiển thị danh sách thành viên đầy đủ.

- **Yêu cầu:**

- Lấy `members` từ Redux Store.
- Map qua `members` để render `<MemberItem />`.
- Áp dụng style từ `MemberList.css`.

### `src/components/MemberItem/MemberItem.jsx`

- **Mô tả:** Hiển thị từng thành viên trong danh sách.

- **Yêu cầu:**

- Tên thành viên là liên kết đến `/member/:id` (dùng `Link`).
- Nút "Delete" để dispatch `removeMember` (kiểm tra không được xóa nếu thành viên đang mượn sách).
- Áp dụng style từ `MemberItem.css`.

### `src/components/FilterButtons/FilterButtons.jsx`

- **Mô tả:** Nút lọc danh sách sách.

- **Yêu cầu:**

- 3 nút: "All", "Borrowed", "Available".
- Khi nhấn, dispatch `setFilter` để cập nhật trạng thái lọc trong `bookSlice`.
- Áp dụng style từ `FilterButtons.css`.

### `src/screens/BookDetail/BookDetail.jsx`

- **Mô tả:** Trang chi tiết của một cuốn sách.

- **Yêu cầu:**

- Lấy `id` từ URL bằng `useParams`.
- Hiển thị thông tin sách (title, borrowed, borrowedBy, addedAt) từ `bookSlice`.
- Nếu `borrowedBy` tồn tại, hiển thị tên thành viên mượn (lấy từ `memberSlice`) dưới dạng liên kết đến `/member/:id`.
- Nếu không tìm thấy sách, hiển thị thông báo "Book Not Found" và nút quay lại `/`.

- Nút "Back" dùng `Link` để quay lại `/`.

#### `src/screens/MemberDetail/MemberDetail.jsx`

- **Mô tả:** Trang chi tiết của một thành viên.
- **Yêu cầu:**
  - Lấy `id` từ URL bằng `useParams`.
  - Hiển thị thông tin thành viên (name, joinedAt) từ `memberSlice`.
  - Hiển thị danh sách sách đang mượn (lọc từ `bookSlice` dựa trên `borrowedBy`).
  - Nếu không tìm thấy thành viên, hiển thị thông báo "Member Not Found" và nút quay lại `/`.
  - Nút "Back" dùng `Link` để quay lại `/`.

#### `src/screens/NotFound/NotFound.jsx`

- **Mô tả:** Trang lỗi.
- **Yêu cầu:**
  - Hiển thị thông báo "404 - Page Not Found" cho các route không tồn tại.

## Gợi ý mở rộng

- Thêm tính năng "trả sách" trong `BookItem` (xóa `borrowedBy` khi sách được trả).
- Tối ưu hiệu năng bằng cách sử dụng `createSelector` để lọc danh sách sách hoặc thành viên.
- Kiểm tra logic xóa thành viên: Không cho phép xóa nếu thành viên đang mượn sách.