

# ReactJS Part-4 - Lab Assignment

## Project - 1

### Small Project: Student Management System

**Concepts Covered:** JSX, Components, Props, State (`useState`), Event Handling, Lists, Forms, React Router (for navigation)

---

#### Project Structure

```
📁 student-management
├── 📁 src
│   ├── 📁 components
│   │   ├── 📄 StudentList.js
│   │   ├── 📄 StudentForm.js
│   │   └── 📄 StudentDetails.js
│   ├── 📄 App.js
│   └── 📄 index.js
├── 📄 package.json
└── 📄 README.md
```

---

### Modules to Implement

#### Module 1: Student List

- Display a list of students with **name, age, and grade**.
- Each student should have a **"View Details"** button.
- Implement **React Router** to navigate to a detailed student page.

#### Hints:

- Use `useState` to store the student list.
  - Use `.map()` to render the list dynamically.
  - Implement a `Link` from `react-router-dom` for navigation.
- 

#### Module 2: Add a New Student (Form Handling)

- Create a form with inputs for **name, age, and grade**.
- Add a **"Submit"** button to update the student list.

#### Hints:

- Use **controlled inputs** with `useState`.
  - On form submission, update the student list.
  - Prevent empty form submissions using `trim()`.
- 

### Module 3: Student Details Page

- Show the details of a selected student on a new page.
- Display additional information such as **email, phone number, and address**.

#### Hints:

- Pass the selected student's details using **React Router params**.
  - Retrieve and display the student's information dynamically.
- 

### Optional Enhancements

- ✓ **Delete a Student** – Add a delete button to remove students from the list.
  - ✓ **Edit Student Information** – Allow users to update details.
  - ✓ **Local Storage Integration** – Save data so it persists after page refresh.
  - ✓ **Styled Components** – Use inline styles or CSS modules for better UI.
-

# Project - 2

## Small Project: Online Bookstore

**Concepts Covered:** JSX, Components, Props, State (`useState`), Event Handling, Lists, Forms, React Router, Conditional Rendering

---

### Project Structure

```
online-bookstore
├── src
│   ├── components
│   │   ├── BookList.js
│   │   ├── BookDetails.js
│   │   └── AddBook.js
│   ├── App.js
│   └── index.js
├── package.json
└── README.md
```

---

## Modules to Implement

### Module 1: Book List (Homepage)

- Display a list of available books with **title, author, and price**.
- Each book should have a **"View Details"** button.
- Implement **React Router** to navigate to a detailed book page.

#### Hints:

- Use `useState` to manage the book list.
  - Use `.map()` to render the list dynamically.
  - Implement a `Link` from `react-router-dom` for navigation.
- 

### Module 2: Book Details Page

- Show **title, author, price, description, and availability** of a book.
- Display an **"Add to Cart"** button.

#### Hints:

- Pass the selected book's details using **React Router params**.
- Retrieve and display the book's information dynamically.

- Use `useState` to manage cart status (e.g., "Added to Cart").
- 

### Module 3: Add a New Book (Admin Panel)

- Create a form with inputs for **title, author, price, and description**.
- Add a "**Submit**" button to update the book list.

#### Hints:

- Use **controlled inputs** with `useState`.
  - On form submission, update the book list.
  - Prevent empty form submissions using `trim()`.
- 

### Optional Enhancements

- ✅ **Cart Management** – Allow users to add/remove books from a shopping cart.
- ✅ **Local Storage Integration** – Save book data so it persists after page refresh.
- ✅ **Styled Components** – Improve UI with CSS or Tailwind.
- ✅ **Search Functionality** – Implement a search bar to filter books dynamically.