

## 1. Objective:

Create a small web application that displays a list of products and allows users to view details of individual products. The application should use React, Redux, React Router, and Axios for API calls. API data can be mocked using JSON files or a mock server.

## 2. Requirements:

### a. Project Setup

- Initialize a new React project using Create React App.
- Set up Redux for state management.
- Set up React Router for navigation.

### b. Data Mocking

- Create a mock server or use JSON files to serve the product data.
- Use Axios for making API calls to fetch the product data.

### c. Product List Page

- Display a list of products fetched from the mock API.
- Each product should display at least the following information: product name, price, and a brief description.
- Include a "View Details" button/link for each product that navigates to the product detail page.

### d. Create product

- Create new product (Can hold new product in memory if using Json file for read list)

### e. Product Detail Page

- When a user clicks on "View Details", navigate to a product detail page.
- Display detailed information about the selected product: product name, price, description, and any other relevant details.

### f. Edit and Save product

- Data should be update using redux store and updated in main grid.

### g. Redux Store

- Use Redux to manage the state of the application.
- Store the list of products and the details of the selected product in the Redux store.
- Use Redux actions and reducers to handle state changes.

### h. Routing

- Set up routing to handle navigation between the product list page and the product detail page.
- Use React Router for navigation.

### i. API Calls with Axios

- Use Axios to fetch data from the mock API.
- Handle loading states and errors appropriately.

### j. Styling

- Apply basic styling to make the application visually appealing.
- Use CSS or a CSS-in-JS solution of your choice.

## 3. Deliverables:

- A GitHub repository with the complete source code.
- A README file with instructions on how to set up and run the project.

## 4. Bonus Points:

- Implement a search functionality to filter products on the product list page.
- Add pagination to the product list if there are many products.

- Use TypeScript for type safety.
- Write basic unit tests for Redux reducers and React components using a testing library like Jest.