Project Management System

### ReactJS Case Study

#### Queries?? Reach out to [Karthik Srinivasan – 2055776]

Here’s a sample case study for a React project that you can use for practice. This project is designed to cover various aspects of React development and will be graded out of 100 marks.

# Project Title: Product Management System

## Objective:

Build a web application for managing products. The application should allow users to view, add, and delete products.

## Requirements:

### 1) User Interface (30 marks)

Create a responsive UI using React components using Material-UI for styling.

Implement a navigation bar with links to different sections (Home, Products, Add Product).

### 2) Product List (20 marks)

Display a list of products with details such as name, price, and description.

Use React state to manage the list of products.

Implement pagination or infinite scrolling for the product list.

### 3) Add Product (20 marks)

Create a form to add new products.

Implement form validation to ensure all fields are filled correctly.

Use React state and props to manage form data.

### 4) Delete Product (10 marks)

Implement functionality to delete a product from the list.

Confirm deletion with a modal or alert.

### 5) API Integration (10 marks)

Use a mock API or a real backend service to fetch and update product data.

Handle API requests and responses using Axios or Fetch API.

### 6) State Management (10 marks)

Use React Context API to manage global state.

Ensure state updates are reflected across the application.

### Bonus (Optional):

### 7) Search and Filter (10 marks)

Implement search functionality to filter products by name or category.

Add sorting options (e.g., by price, by name).

## Evaluation Criteria:

### 1) Code Quality

Clean, readable, and well-documented code.

Proper use of React best practices and design patterns.

### 2) Functionality

All required features are implemented and work correctly.

The application is responsive and user-friendly.

### 3) Creativity and Design

Use MUI

Aesthetic design and user experience.

### 4) Error Handling

Loading Page or Icons.

Proper error handling for API requests and form validations. User-friendly error messages.

### 5) Performance

Efficient rendering and state management.

Optimized for performance.

## Getting Started:

### Setup:

Install Node.js and npm.

Create a new React application using Create React App: npx create-react-app product-management.

### Development:

Start by setting up the project structure.

Implement each feature step-by-step, testing as you go.

### Testing:

Manually Test the application thoroughly to ensure all features work as expected.

Fix any bugs or issues that arise.

### Submission:

Submit the project before 3rd October 2024. Ping me one to one so we can setup a call and discuss.

Good luck with your project! If you have any questions or need further assistance, feel free to ask me (Karthik Srinivasan - 2055776).