**Module 09**: React.js – Components ,State and Props  
 **Module 10**: Lists , Hooks , Localstorage , Api Project

**Module-11**: React -Advance React- Styling , Routing

**Module-12**: React – JSON-server and Firebase Real Time Database

## Case Overview

* **Scenario:** You are tasked with building a **React.js app(Task Manager App)** that interacts with a **REST API** to handle Task requests. The app must provide form submission, API integration, and record management with search and filtering features.

## Core Functionality

## Tech Stack

React Form Handling & Controlled Components

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Responsive Task manager Web Form | Fields: Task Title (text) ,Description (textarea) ,Due Date (date),Priority (select: High, Medium, Low) |
| Data Submission (API) | * On valid submission,sent POST request. Show success message and reset form fields.Use json-server for localAPI. |
| View Submission | * Fetch data from api and display in table with columns:Name,Phone,Email,Address.Include search filter by Name.Show “No records found” if empty. |
| Delete Record | * Add Delete button for each [record.On](http://record.on) click,confirm action, send DELETE request to api, remove from UI immediately. |
| Edit Record | * Add edit button for each record.Put request use for update. |

## Key Competencies Tested

* React Form Handling & Controlled Components
* JavaScript Input Validation
* API Integration (GET, POST, PUT, DELETE)
* React Router Navigation
* Search & Filter Logic
* Real-time UI Updates

## Reflective Thinking Questions

* How would Redux or Context API improve state management in this app?
* Could RTK Query make API integration more efficient?
* What authentication method would you implement to secure CRUD operations?