**Module-13**:React js (Task manager App) – Applying Redux

## Case Overview

* **Scenario:** You are tasked with building a scalable React + Redux application to manage customer complaints. The system must integrate with a REST API using Redux Toolkit, allow form submissions, manage records with search and filter features, and support create, update, and delete operations.

## Core Functionality

**Tech Stack**

React Form Handling & Controlled Components

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Responsive Web Form | * **Fields**: Task Title (text) ,Description (textarea) ,Due Date (date),Priority (select: High, Medium, Low) |
| Data Submission | * Use createSlice for crud. On valid submission, send POST request to API. Update Redux store and UI instantly. |
| 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](http://record.on). 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
* Redux Toolkit (createSlice, createAsyncThunk, useDispatch, useSelector)
* API Integration (GET, POST, PUT, DELETE)  
  Search & Filter Logic
* Real-time UI Updates via Redux Store
* Context API for UI state
* RTK Query for advanced API handling

## 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?