Software Document

Table of Contents

1. Introduction:

A simple task management application made using React.js. The application allows the users to create, update, delete, and view tasks. It also allows the user to mark the task complete or incomplete by just one click.

2. Features:

- **Task List:** Display a list of tasks with their title, description, and status (complete or incomplete).
- **Create Task:** Users should be able to add new tasks with a title, description, and status (defaulted to incomplete).
- **Edit Task:** Users should be able to edit the title, description, and status of existing tasks.
- **Delete Task:** Users should be able to delete a task.
- **Mark Task Complete/Incomplete**: Users should be able to toggle the status of a task between complete and incomplete.

3. Requirements:

- Use React.js for building the user interface.
- Manage the state of tasks within the application using React state or a state management library like Redux.
- Implement a clear and user-friendly interface.
- Use CSS or a CSS framework (e.g., Bootstrap) for styling.
- Ensure the application is responsive and works well on different screen sizes.
- Validate user inputs and provide feedback on any errors.
- Use appropriate data structures (e.g., arrays, objects) to manage tasks efficiently.
- Store tasks temporarily within the application (no need for a backend or database).

- Write all sorts of tests for the entire application.

4. Codebase:

Following is the overview of codebase of project

4.1 Dependencies:

Project created with react redux toolkit template. Used npm to install following packages:

- "@fortawesome/fontawesome-svg-core": "^6.4.2",
- "@fortawesome/free-brands-svg-icons": "^6.4.2",
- "@fortawesome/free-regular-svg-icons": "^6.4.2",
- "@fortawesome/free-solid-svg-icons": "^6.4.2",
- "@reduxjs/toolkit": "^1.9.5",
- "@testing-library/jest-dom": "^5.17.0",
- "@testing-library/react": "^13.4.0",
- "@testing-library/user-event": "^14.4.3",
- "bootstrap": "^4.6.0",
- "bootstrap-icons": "^1.10.5",
- "react": "^18.2.0",
- "react-dom": "^18.2.0",
- "react-redux": "^8.1.2",
- "react-scripts": "5.0.1",
- "web-vitals": "^2.1.4"

4.2 State Management and Redux:

Redux toolkit used for state management:

- **store.jsx:** contains the configured store using toolkit.
- **TaskSlice.jsx**: contains a Redux slice that is a concept introduced by Redux Toolkit that represents a self-contained piece of the Redux store that includes a reducer function, initial state, and action creators. It respectively contains reducers for crud operations on store.
- **Objects properties**: { id: number, title: string , description: string, completed: Boolean}

4.3 Styling:

Bootstrap 4.6 used for styling with vanilla css in file App.css. Also,

- Bootstrap Icons: used for edit and delete button icons

4.4 Working:

Working of the App is explain in steps:

1. Index.js file wraps App.js component in Redux Provider
2. App.js renders only Task component that contains the UI for todo(Task management) App.

3. Task.js: Event handlers like onClick() and onChange() are used to get data from inputs in useStates and states are used to create an object that is passed as a payload using dispatcher to call reducer using slice so that we do not need to implement Actions separately.
4. Finally useSelector hook lets us access store data. Which is stored in a list and mapped on a screen to display.

4.5 Instructions on Running:

First clone the git repo than:

1. Run command **npm install**
2. Than **npm start**