**Student Have To Prepare Report In Format**

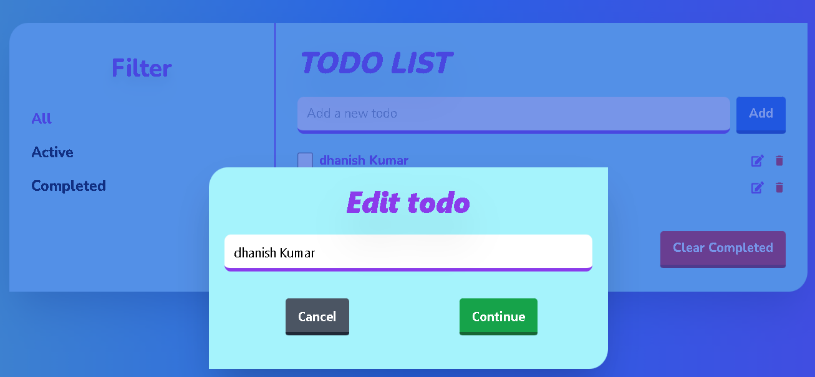
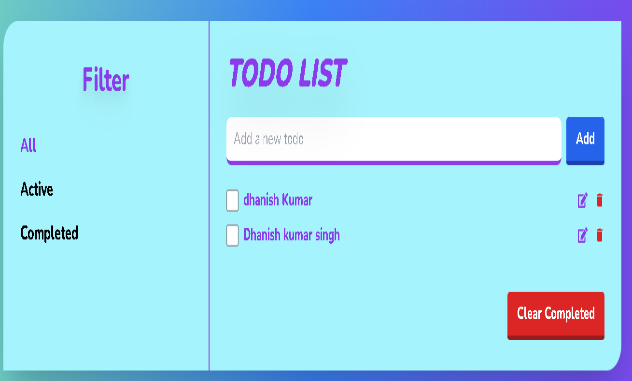
1. Add Task Description
2. Attach Screenshot Of Output.
3. Describe Widget/Algorithm Used In Task
4. Add Report In Your Task Zip File

**Sample Example :**

1. **Task Description**

This project involves creating a **Todo List App** that offers a seamless and user-friendly experience for managing tasks. The application will utilize **Redux** for state management, ensuring efficient and scalable handling of the app's data flow.

1. **Task Output Screenshot**



1. **Widget/Algorithm Used In Task**

* **1. Redux for State Management**
* **Widget/Algorithm:**  
  Redux is used as a predictable state container to manage the app's global state.
* **How It Works:**
* **Actions:** Encapsulate the changes (e.g., adding, removing, or marking a task).
* **Reducers:** Take the current state and an action, returning a new state.
* **Store:** Holds the global state and allows components to access it via the connect function or React-Redux hooks like useSelector and useDispatch.
* **2. Task Rendering Logic (React Components)**
* **Widget/Algorithm:**  
  Tasks are displayed using React functional components.
* **How It Works:**
* Each task is rendered as a list item (<li>) dynamically based on the state from Redux.
* Conditional rendering is applied to differentiate between completed and pending tasks using styles or icons.
* **3. Toggle Status Algorithm**
* **Widget/Algorithm:**  
  A simple algorithm toggles the isCompleted status of a task when the user interacts with it.
* **How It Works:**
* When the user clicks a "Mark as Complete" button, an action is dispatched to update the task’s status in the Redux store.
* The reducer modifies the isCompleted property for the corresponding task.
* **4. Filtering and Mapping**
* **Widget/Algorithm:**  
  Mapping is used to iterate over the task list, and optional filters can be implemented for task views (e.g., show only pending tasks).
* **How It Works:**
* Array.map() is used to render tasks dynamically in the UI.
* Filtering tasks by status can use Array.filter(), enabling features like "Show Completed" or "Show All."
* **5. Input Validation**
* **Widget/Algorithm:**  
  Basic input validation ensures users cannot add empty tasks.
* **How It Works:**
* A check is performed on the task input field before dispatching the "Add Task" action.
* If the input is invalid, an error message or warning can be displayed.