**Steps to Implement the Todo App**

1. **Set Up the Project** – I created separate frontend and backend directories. I initialized **Node.js** for the backend using Express and set up **React.js** for the frontend.
2. **Built the Backend** – I set up an **Express server**, connected it to **MongoDB** using **Mongoose**, and configured middleware like **CORS** and express.json() to handle API requests.
3. **Created the Todo Model** – I defined a schema with fields for text, completed, and order to store todos and manage their position for reordering.
4. **Developed API Routes** – I created endpoints for adding, retrieving, updating, deleting, and reordering todos so that data is stored and manipulated in MongoDB.
5. **Ran the Backend** – I created a .env file with the **MongoDB connection string**, started the backend using npm run dev, and verified the API routes using **Postman**.
6. **Set Up the Frontend** – I initialized a **React app**, installed necessary dependencies like **Axios** for API communication, and used @hello-pangea/dnd for drag-and-drop functionality.
7. **Managed State in React** – I used **useState** and **useEffect** in App.jsx to fetch todos, add new todos, update their status, delete them, and handle filtering options.
8. **Created Components** – I built:
   * TodoInput.jsx for adding new todos with a text field.
   * TodoList.jsx to display todos and support drag-and-drop.
   * TodoItem.jsx to manage each todo with complete and delete actions.
   * ThemeToggle.jsx to switch between **light and dark mode**.
9. **Connected Frontend to Backend** – I used **Axios** to send requests to the backend for fetching, adding, updating, deleting, and reordering todos.
10. **Implemented Drag and Drop** – I used @hello-pangea/dnd to allow users to **reorder todos** and updated the backend accordingly to **persist changes** in MongoDB.
11. **Styled the Application** – I used **CSS** to ensure a **modern, responsive UI**, making sure the **filter buttons** highlight the active state, and dark/light mode changes were applied properly.
12. **Ran the Application** – I started both the **frontend (npm start)** and **backend (npm run dev)** servers, tested the **todo functionality**, and ensured **drag-and-drop and filtering** worked as expected.