Assignment: Build a Task Management Dashboard with the Kanban Board

Objective:

To build a task management dashboard in Next.js that includes two screens: a task list and a Kanban board. The app should enable users to manage tasks through a list view and a Kanban board where tasks can be moved between statuses. The app should include user authentication, a backend API, and a MongoDB database. The UI should be visually appealing and implemented using **Shadcn**. The app should be deployed on Vercel for review.

Requirements:

1. **User Authentication:**
    - Implement user authentication (sign up, log in, and login) using JWT (JSON Web Tokens).
    - Protect routes to ensure that only authenticated users can access the task management features.
2. **Task Management:**
    - Users should be able to create, edit, delete, and view tasks.
    - Each task should have the following fields:
        - Title (string, required)
        - Description (string, optional)
        - Status (enum: "To Do", "In Progress", "Completed")
        - Priority (enum: "Low", "Medium", "High")
        - Due Date (date, optional)
    - Tasks should be filterable and sortable by status, priority, and due date.
3. **Two Screens:**
    - **Task List Screen:**
        - Display a list of tasks with filtering and sorting options.
        - Allow users to perform CRUD operations directly from the list view.
    - **Kanban Board Screen:**
        - Display tasks in a Kanban board format with columns for each status ("To Do", "In Progress", "Completed").
        - Enable drag-and-drop functionality for moving tasks between statuses.
        - Update the task status dynamically when moved between columns.
4. **Backend API:**
    - Build a backend API using Node.js and Express to handle the CRUD operations for tasks and user authentication.
    - Connect to a MongoDB database to store user and task data.
5. **Frontend Features:**
    - Use Next.js to create a responsive and dynamic user interface.
    - Implement a task list view and a Kanban board view.
    - Use state management (React context or a library like Redux) to manage the application state.
    - Provide a form for creating and editing tasks.
    - Display error messages and handle form validation.
6. **UI Design:**

    - Use **Shadcn** to create a modern and visually appealing user interface.
    - Ensure the UI is responsive, user-friendly, and provides a good user experience on both desktop and mobile devices.
    - To enhance interactivity, use components like modals, tooltips, drag-and-drop areas, dropdowns, and notifications.
7. **Deployment:**
    - Deploy the app on Vercel and ensure that it is fully functional.
    - Provide a link to the deployed app and any test credentials needed to access the features.
8. **Code Quality:**
    - Follow best practices for React and MERN stack development.
    - Write clean, maintainable, and modular code.
    - Include comments where necessary to explain complex logic.
    - Use TypeScript to define types for better maintainability (optional but recommended).
9. **Additional Considerations:**
    - Use a version control system (like Git) and provide a link to the repository.
    - Implement at least one custom hook in React to manage some part of the application logic.

Evaluation Criteria:

- **Functionality:** How well does the app meet the requirements?
- **UI/UX:** The quality and aesthetics of the UI created with Shadcn, including responsiveness and user experience.
- **Kanban Functionality:** Properly implementing drag-and-drop functionality and status updates on the Kanban board.
- **Code Quality:** Is the code clean, modular, and easy to understand?
- **React Skills:** Proper use of React hooks, state management, component architecture, and handling of side effects.
- **MERN Stack Understanding:** Correct implementation of the backend API, database interactions, and integration with the front end.
- **Logic and Problem-Solving:** Efficient and logical handling of features and edge cases.
- **Deployment:** Successful deployment on Vercel with a working link.