Task: Full-Stack Task Manager (MERN)

Goal: Build a full-stack task management application with user authentication, role-based access, task CRUD operations, and filtering.

📌 Features & Requirements

1️⃣ User Authentication

✅ Implement **JWT-based authentication** with the following:

- **Sign Up & Login** (passwords encrypted using `bcryptjs`).
- Store JWT in **HTTP-only cookies**.
- **Logout Route**: Access to the protected routes must be removed upon Logout.
- Users must be **logged in** to access any task-related operations.

2️⃣ Role-Based Access Control (RBAC)

✅ Implement **two user roles**:

- **User:** Can create, view, update, and delete their own tasks.
- **Admin:** Can **view and delete** any task across all users.

✅ Admins cannot modify a user's task but can **delete** them.

✅Note:

- Users **must be assigned a role** at registration (`user` by default).

3️⃣ Task Management (CRUD + Filtering)

✅ **CRUD Operations for Tasks**

✅ **Task Filtering & Sorting**

- **Sort tasks** by **priority, due date, or status**.
- **Filter tasks** based on **assigned user, priority, or completion status**.

4️⃣ Frontend (React.js)

✅ **Use React Router** for page navigation.

✅ **Global State Management**

- Use **Redux Toolkit** (preferred) or **Context API**.
- Create a `useAuth()` hook to handle authentication state.

✅ **Reusable Components**

- **TaskCard** for displaying tasks.

✅ **Note**:

- Prevent users from modifying tasks **directly in the frontend** (must go through API).
- Hide admin-only features (delete others' tasks) for regular users.
- Use **protected routes** (`<PrivateRoute>`) to restrict access.

5️⃣ Backend (Node.js + Express.js)

✅ **Authentication & Authorization Middleware**

- Protect routes using JWT middleware..

6️⃣ Database (MongoDB)

✅ **Use MongoDB Atlas (cloud storage)**.

7️⃣ Deployment

✅ **Frontend** → Deploy on **Vercel**.
✅ **Backend** → Deploy on **Render**.
✅ **Database** → Use **MongoDB Atlas**.

NOTE

If the deployment is not complete, you can record a video showcasing the final output screen with all functionalities and upload it to your Google Drive (ensuring link-based access is enabled for anyone). Additionally, host the entire code on GitHub.

You have been provided with only rough requirements, so you are free to choose the implementation approach as long as the core task management functionality remains intact.