MERN Stack Coding Assignment with Authentication (2 Hour)

Objective:

Build a **Mini "Task Manager" App** with **user authentication** (JWT) and task CRUD operations. The aim is to test end-to-end MERN skills including auth, API design, and React integration.

Requirements

Backend (Node.js + Express + MongoDB)

User Authentication

1. Create a `User` model with fields:
   - `name`
   - `email` (unique)
   - `password` (hashed)
2. Implement:
   - `POST /api/auth/register` → Register a new user
   - `POST /api/auth/login` → Login user and return a **JWT token**
   - Use middleware to **protect routes** (check JWT in `Authorization` header).

You can use any password hashing method like bcrypt and generate JWT with `jsonwebtoken`.

Task Management

1. Create a `Task` model with fields:
   - `title` (string)
   - `status` (pending/completed, default pending)
   - `user` (ObjectId → references User)
2. Implement the following **protected** routes:
   - `POST /api/tasks` → Create a new task for the logged-in user
   - `GET /api/tasks` → Get all tasks of the logged-in user
   - `PUT /api/tasks/:id` → Update a task's status (e.g., pending → completed)
   - `DELETE /api/tasks/:id` → Delete a task

Frontend (React)

1. Create a simple React UI with:
   - **Login & Register** forms (email + password; name only for register)

    - On successful login, store the **JWT token in localStorage** and use it for authenticated API calls.
    - A **Task Dashboard**:
        - Input box to add new task
        - List all tasks for the logged-in user
        - Button to **mark task as completed**
        - Button to **delete task**
2. Redirect unauthenticated users to the login/register page.

Keep the UI clean but functional — design is secondary to working authentication + CRUD.

Bonus

- Add **filter** to view “All / Pending / Completed” tasks.
- Show logged-in user’s name on the dashboard.
- Implement logout.

Expected Deliverables

- **GitHub repo** containing both backend and frontend code (client + server folders).
- **README** with clear setup instructions.

□ Evaluation Criteria

Area	What We Look For
Authentication	Proper JWT flow, password hashing, route protection
Functionality	Register → Login → Create/View/Update/Delete tasks
MERN Integration	Smooth API integration with React
Code Quality	Structure, naming, clean routes, reusable components
Bonus	Filters, logout, UX touches
Time Management	How much is achieved within 2 hour
