**Setup Instructions**

**Steps**

1.**Set Up Environment**

* Install NestJS CLI: npm i -g @nestjs/cli
* Initialize a new NestJS project: nest new project-management-api
* Install Prisma and PostgreSQL client
* Initialize Prisma: npx prisma init
* Configure DATABASE\_URL in .env for PostgreSQL.

2. **Define Prisma Models**

* Define the database schema in prisma/schema.prisma.
* **Run Migrations**: After defining the models, run Prisma migrations to create the tables in PostgreSQL.

3. **Set Up NestJS Modules**

* Create User, Project, and Task modules.

4. **Create DTOs (Data Transfer Objects)**

* Define DTOs for creating and updating User, Project, and Task data.

5. **Implement CRUD Services and Controllers**

* Implement services and controllers for users, projects, and tasks.

6. **Define API Endpoints**

* Set up routes in the controllers for each of the required endpoints:
* Users: /users, /users/:id
* Projects: /projects, /projects/:id
* Tasks: /projects/:projectId/tasks, /tasks/:id

7. **Implement JWT Authentication**

* Use @nestjs/jwt to protect routes, ensuring only authenticated users can manage data.
* Define guards in auth.module.ts to secure endpoints.

8. **Filtering and Search**

* Implement filtering in the TasksController by using query parameters for task status and assigned user.

9. **Test API Endpoints**

* Use Postman or another tool to verify each endpoint and ensure the correct handling of CRUD operations, relationships, and filtering.

10. **Deploy and Document**

* Add detailed setup instructions in a README.md file.
* Make a short screen recording to demonstrate testing the endpoints.