Task: Creating an API Endpoint to Fetch Data in Nest.js

Description: As an intern working with Nest.js, your task is to create a simple API endpoint that allows users to fetch data from a predefined dataset. This will help you understand the basics of creating routes, controllers, and handling API requests in Nest.js.

Steps:

Project Setup:

Set up a new Nest.js project using the Nest CLI or any other method you're comfortable with.

Create a basic module, let's call it DataModule, using the CLI or manually.

Data Service:

Create a service, DataService, within the DataModule.

Define a method in the service to return a predefined dataset (e.g., an array of objects).

Controller and Route:

Create a controller, DataController, within the DataModule.

Define a route using the @Get() decorator that maps to an endpoint like /api/data.

Inside the route handler method, call the method from the DataService to retrieve the data.

Testing:

Write unit tests for the DataService method to ensure it returns the correct data.

Write integration tests for the /api/data endpoint using testing frameworks like Jest.

Validation and Error Handling:

Implement input validation using Nest.js validation pipes (e.g., query parameters).

Handle potential errors, such as empty datasets or invalid queries, by sending appropriate HTTP responses.

Documentation:

Add basic documentation to the endpoint using comments or decorators to describe the purpose and usage of the /api/data route.

Code Quality:

Ensure your code follows coding standards, is well-documented, and uses meaningful variable and function names.

Submission:

Push your code to a version control repository (like GitHub). USE YOUR OFFICE EMAIL

Provide a brief summary of your implementation and any challenges you faced in a written document.

This task will help you practice creating a basic API endpoint, handling API requests, and ensuring code quality and documentation in Nest.js.