DG Technical Test - NestJS Interview

Build a retail store api that exposes endpoints to retrieve products (Name, Upc, …) and operate all basic CRUD operations. Here are some requirements :

1. All endpoints should only be accessible for authenticated users.
Implement a simple user module that perform data validation and use cookie-based authentication.

2. Implement a caching mechanism into your application. You can for example cache the user information, or the respond when you retrieve product data for a specified period.

3. Create an event emitter each time a new product is created. Implement an event listener that logs the product information.

4.Write unit tests for the Product service using testing libraries like Jest.
Test various scenarios, including creating a new product, edge cases, entering wrong/ duplication information.

5. Test the performance of you application (Optional)
- Identify a performance bottleneck in your application (e.g., a slow endpoint or database query).
- Implement a performance improvement using any relevant technique.
- Measure and document the performance improvement achieved.

Good to have :

- Dependency injection for services. Ensure that services are properly injected into controllers and modules.
- Protect routes that require authentication with Guards
- Multithreading scenario. For example create an endpoint that performs a CPU-intensive task using multiple threads and returns the result.

Submission Instructions:

Please share your code repository (e.g., a GitHub repo) with the completed NestJS application, along with any additional instructions or documentation. You can also provide a description of any optional tasks you completed (multithreading) and the details of the performance improvement you achieved.