Functional Analysis: Product and Inventory Manager

Introduction

The **Product and Inventory Manager** is built using a modern technology stack to ensure scalability, security, and performance. Each technology was carefully chosen to meet the specific requirements of the project.

Technology Stack

1. NestJS

NestJS provides a modular architecture that promotes scalability and maintainability. It integrates well with TypeScript, ensuring type safety and developer productivity. It also supports microservices, which is ideal for handling complex backend systems.

2. TypeScript

TypeScript enhances JavaScript with static typing, preventing common errors and improving code quality. It also offers better tooling, autocompletion, and refactoring. This leads to more efficient development and easier debugging.

3. JWT (JSON Web Tokens)

JWT enables secure, stateless authentication for the system.

4. KafkaJS

KafkaJS allows us to handle real-time event-driven architecture efficiently

5. Redis

Redis is used for caching and real-time messaging.

6. TypeORM

TypeORM simplifies database interactions using an ORM pattern, allowing for easy CRUD operations.

7. MySQL

MySQL is a reliable and scalable relational database system. It supports ACID-compliant transactions, ensuring data consistency.

8. Axios

Axios simplifies making HTTP requests from the frontend to the backend.

9. Docker

Docker provides isolated, consistent environments for development, testing, and production. It enables easy deployment and scaling of the system using containers.