Technical Test: Inventory Management System

This technical test is designed to evaluate your full-stack development skills, your understanding of software architecture, and your ability to build a robust, production-ready application. Your task is to develop a comprehensive inventory management system.

The Challenge

You are tasked with creating a full-stack application that provides users with a secure platform to manage product inventory. The system must support user authentication and provide a complete set of features for managing assets in a store or warehouse.

System Architecture

Your solution should be composed of distinct and well-defined layers to demonstrate a clear separation of concerns. The system consists of a backend API and a frontend user interface.

1. Backend (API)

The backend is the core of your application, responsible for handling all business logic and data persistence.

- **Technology Stack:**
  - **Language:** Python
  - **Framework:** FastAPI
  - **Database:** A SQL database (e.g., SQLite, PostgreSQL).
- **Key Functionality:**
  - **User Authentication:** Implement a user management system with secure authentication. A token-based approach, such as JWT, is recommended.
  - **Product Management (CRUD):** Develop a set of RESTful API endpoints for the following operations on inventory products:
    - Create: Add a new product to the inventory.
    - Retrieve: Get a list of all products or fetch a single product by its unique identifier.
    - Update: Modify the details of an existing product.
    - Delete: Remove a product from the inventory.
  - **Data Models:** Define the data models for User and Product. Each Product must have a unique ID, along with attributes such as name, description, price, and quantity.
  - **Repository Pattern:** The data access layer must be abstracted using the repository pattern. Your service or business logic layer should interact with repositories to perform database operations, ensuring a clear separation from the underlying database technology.

2. Frontend (User Interface)

The frontend provides the user interface for interacting with the backend API.

- **Technology Stack:** You have the flexibility to choose a technology that best showcases your skills. Options include:
  - A modern JavaScript framework (e.g., React, Vue, Svelte).
  - Plain HTML, CSS, and JavaScript.
- **Key Functionality:**

    - **Login/Logout:** A user-friendly interface for authentication.
    - **Inventory Dashboard:** A responsive dashboard that displays all products, including their details.
    - **CRUD Interface:** Intuitive forms and controls for creating, viewing, updating, and deleting products.

Optional Enhancements

These additions are not mandatory but will be highly regarded as they demonstrate a deeper understanding of software development principles.

- **Design System:** Apply a professional and consistent design to the user interface. Using a CSS framework like Tailwind CSS or a custom design system is encouraged.
- **Asset Management:** Add support for including an image URL for each product.
- **Advanced Features:** Implement search, filtering, or sorting functionality for the product list.
- **State Management:** If using a frontend framework, demonstrate a clear and efficient state management approach.
- **Unit & Integration Testing:** Include automated tests for your FastAPI endpoints and data repositories to ensure code quality and reliability.

Evaluation Criteria

Your submission will be evaluated on the following criteria:

1. **Architectural Design (40%):** The clarity and effectiveness of the repository pattern implementation and the overall separation of concerns.
2. **Functionality (30%):** The completeness and correctness of the required features, including CRUD operations and authentication.
3. **Code Quality (20%):** The readability, maintainability, and adherence to best practices (e.g., PEP 8 for Python). This also includes the quality of your documentation and tests, if provided.
4. **User Experience (10%):** The creativity, usability, and visual appeal of the frontend interface.

We look forward to reviewing your solution.