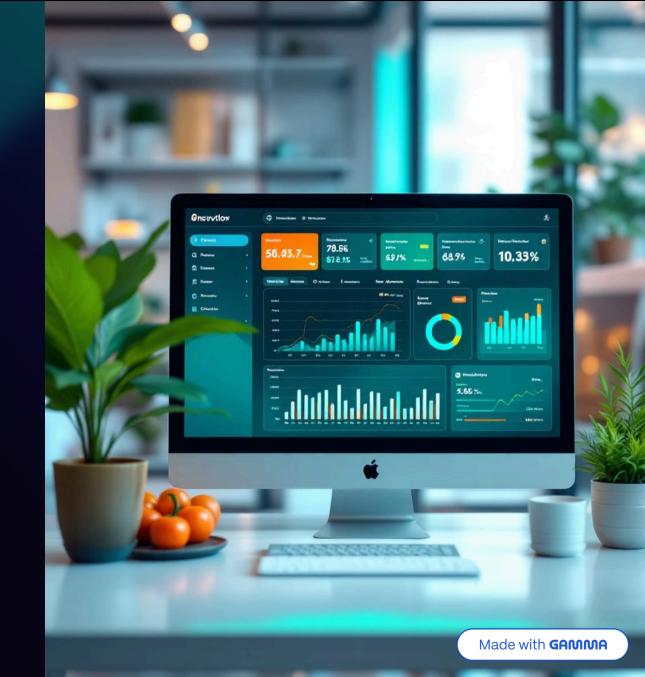
StockWise -Inventory Management **System Full-stack CRUD Application using FastAPI** and Vanilla JavaScript

Author: Jubair

Date: 24-10-2025



Project Overview

Purpose:

- Manage inventory products efficiently
- Perform CRUD operations (Create, Read, Update, Delete)
- Provide a simple, user-friendly frontend interface

Key Features:

- Add, update, delete, and view products
- Search products by ID, name, or description
- Frontend-backend integration with REST APIs
- Modular design for scalability

Project Structure

Folder Layout:

- backend: database connection, FastAPI app, models, schemas
- **frontend:** HTML, JavaScript, and CSS for UI

Backend Highlights:

- PostgreSQL database with SQLAlchemy ORM
- Pydantic schemas for validation
- CRUD APIs with CORS support

Frontend Highlights:

- Dynamic product table
- Add, edit, delete products
- Real-time search and success/error messages

Backend & API

Database & Models:

- Database connection and session management
- Product table with id, name, description, price, quantity
- Pydantic schemas for request and response validation

FastAPI App:

- Initializes backend with CORS
- Preloads sample products if database is empty
- Provides CRUD endpoints for products

Frontend & Workflow

Frontend Functionality:

- Form for adding and editing products
- Table displays all products dynamically
- Search input filters products in real time
- Messages show success or error feedback

Workflow:

- Fetch products from backend
- Display products in a table with Edit/Delete options
- Add new product → POST request
- Edit → PUT request; Delete → DELETE request
- Search filters displayed products dynamically

Integration:

- Frontend interacts with FastAPI backend
- Backend communicates with PostgreSQL database

Error Handling & Future Enhancements

Error Handling:

- Product not found returns error response
- Request fields validated to ensure correct data
- Frontend displays messages for success or failure

Future Enhancements:

- User authentication (JWT/0Auth2)
- Rate limiting to prevent abuse
- Pagination for large product lists
- Upgrade frontend to modern frameworks (React/Vue)
- Analytics dashboard for inventory insights

Conclusion

Fully functional inventory management system

Clean integration between FastAPI backend and JS frontend

Modular, scalable, and ready for additional features