Mini Inventory Management System

# ✅ Overview

A small system for managing items in a store or warehouse, where interns will build full CRUD APIs to:  
- Add new products  
- View product list  
- Update product details (price, quantity, etc.)  
- Delete products  
- (Optional) Filter/search by category, price, or stock status

# 🧠 Why it's a great fit

- Simple domain (products) — no need to explain complex business logic  
- Both software and AI/ML interns can contribute:  
 - Software interns: build the backend CRUD APIs using Node.js (Express/NestJS)  
 - AI/ML interns: analyze stock patterns or simulate demand predictions using Python (Flask)

# 📦 Data Model Example

{  
 "id": 1,  
 "name": "Wireless Mouse",  
 "category": "Electronics",  
 "price": 25.99,  
 "quantity": 50,  
 "inStock": true  
}

# 🛠 Technologies

- Software Dev: Node.js + Express or NestJS, MySQL  
- AI/ML: Python + Flask, optional analysis  
- Tools: Postman (for API testing), Swagger (optional for documentation)

# 🚀 Extended API Ideas

To deepen their understanding of backend queries, interns can implement the following additional APIs:

- Get the \*\*total inventory value\*\*: Sum of (price × quantity) of all products.

- Get the \*\*average product price\*\* across all items.

- Get the \*\*maximum and minimum price\*\* in the inventory.

- Get the \*\*total number of products\*\* per category.

- Get a list of \*\*out-of-stock items\*\* (quantity = 0).

- Get \*\*top 5 most expensive items\*\*.

- Get \*\*items within a price range\*\* (e.g., $10 to $50).

- Get \*\*products added in the last N days\*\* (add a `createdAt` timestamp).