**Bynry Backend Case Study**

**Gayatri Munukutla**

# Project: StockFlow (Inventory Management System)

Implemented a backend solution for a B2B inventory management platform that handles products, warehouses, suppliers, and low-stock alerts. This project fulfills all requirements outlined in the assignment PDF.

Google Drive Backup:  
<https://drive.google.com/file/d/1jErpWI3HcJWvhhRAgr60jO4QWNIDgMXC/view?usp=sharing>

Part 1: Code Review & Debugging

File: part1\_code\_review/fixed\_code.py

* Reviewed broken product creation endpoint.
* Fixed 7+ issues including:  
   - Missing validations  
   - SKU uniqueness  
   - Error handling  
   - Transaction safety

Technologies: Python, Flask, SQLAlchemy

# Part 2: Database Design

Files:

* part2\_database\_design/schema.sql
* part2\_database\_design/design\_notes.md
* Designed relational schema with: companies, warehouses, products, inventory, suppliers, bundles, sales
* Used SQL DDL format.
* Covered normalization, indexing, and business rules.

Questions Listed:

* How bundles are inventoried?
* Supplier–company relationships?
* Sales tracking and price histories?

# Part 3: API Implementation

Files:

* part3\_api\_implementation/app.py
* part3\_api\_implementation/models.py
* Implemented: GET /api/companies/<company\_id>/alerts/low-stock
* Returns all products below their low\_stock\_threshold and recently sold.
* Handles edge cases, supplier info, stockout estimates.

## How to Run Locally

1. In terminal:

python -m venv venv  
venv\Scripts\activate  
pip install -r requirements.txt

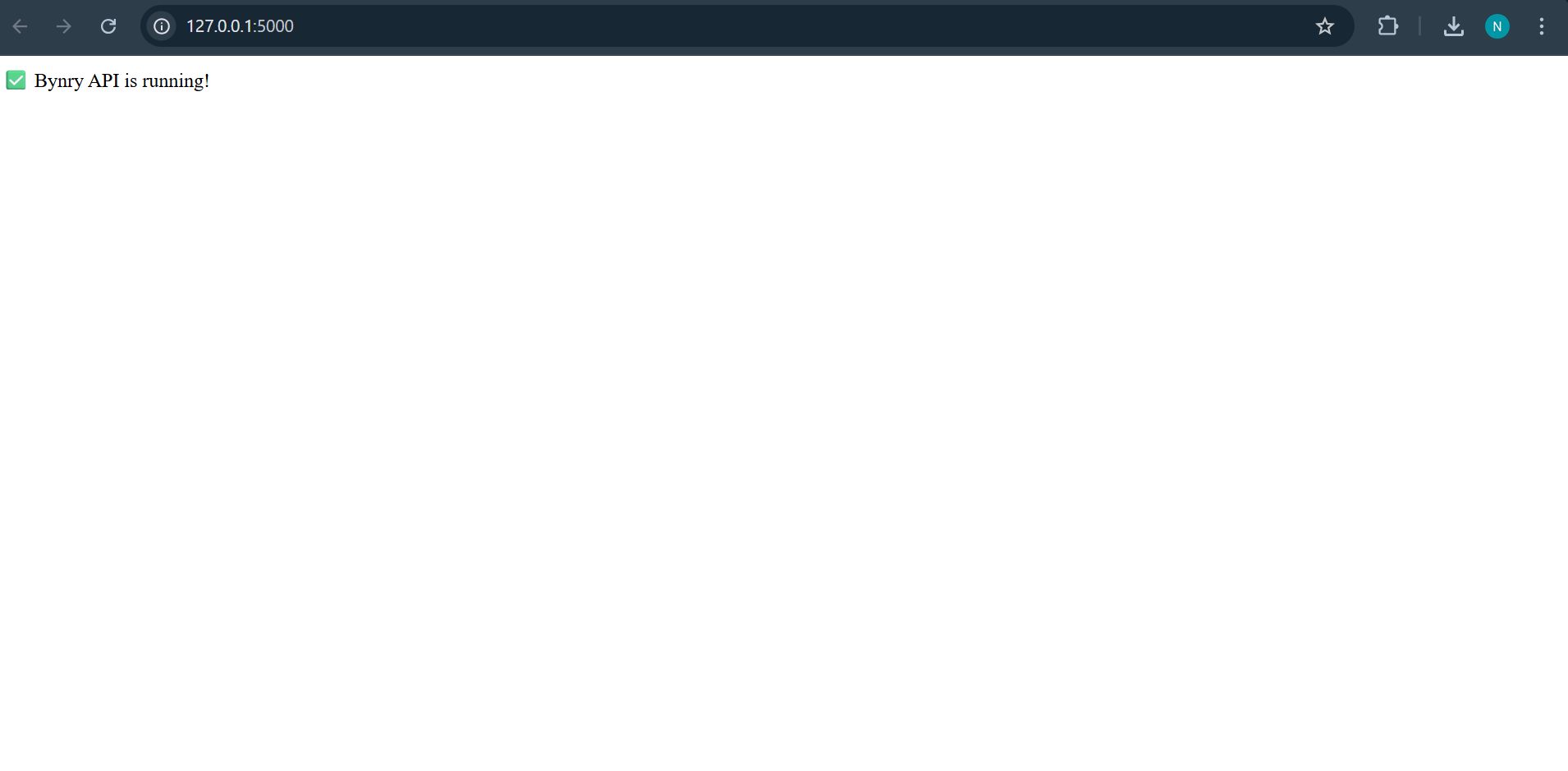
2. Then:

$env:FLASK\_APP = "app.py"  
flask run

3. Visit:

* Home: <http://127.0.0.1:5000/>
* Test: http://127.0.0.1:5000/api/companies/1/alerts/low-stock

# 📸 Screenshot – API Running Successfully



# Final Notes

File: assumptions.md

* Threshold and SKU assumptions
* Recent sales definition
* Inventory tracking logic
* Error-handling logic
* Bundle design strategy

**THANK YOU**