🧪 Test Report - Sweet Shop Management System

This document outlines the results of unit testing for the Sweet Shop Management System project.

The system was tested using Python's built-in unittest module with full test coverage for the following operations:

- Tested Features:
    - Add Sweet
    - Delete Sweet
    - View All Sweets
    - Search Sweets (by name, category, price range)
    - Sort Sweets (by name, category, price)
    - Purchase Sweet (with stock check and errors)
    - Restock Sweet (with input validation)

✔ Summary

Total Test Cases: 45

All tests passed successfully with edge case validation included.
Testing was done using a TDD approach. Each feature was developed only after writing corresponding failing test cases first.

📌 Edge Cases Tested

- Adding sweets with duplicate IDs

- Deleting non-existent sweets

- Purchasing more than available quantity

- Purchasing with zero/negative quantity

- Restocking with invalid input

- Searching with no matching results

- Sorting empty inventory

- Chaining operations (add → delete → view/sort/search)

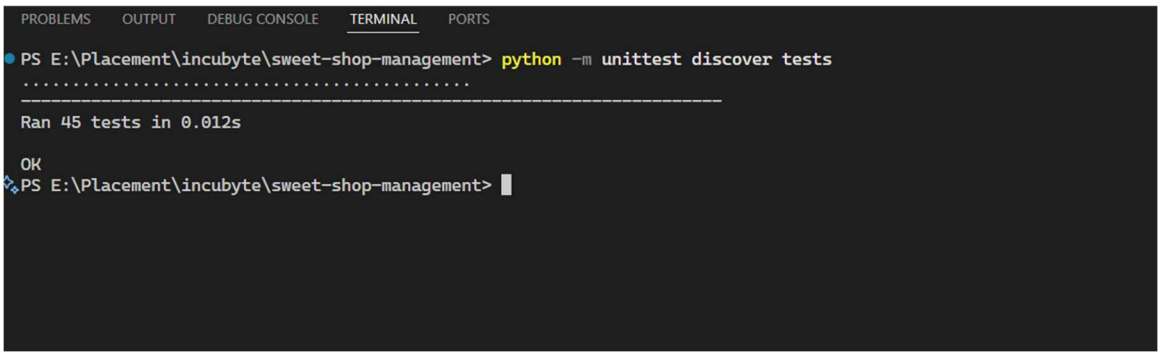
🛠️ Tools & Frameworks

- Python 3.11+
- unittest (standard library)
- Manual CLI testing via main.py

📸 Screenshots

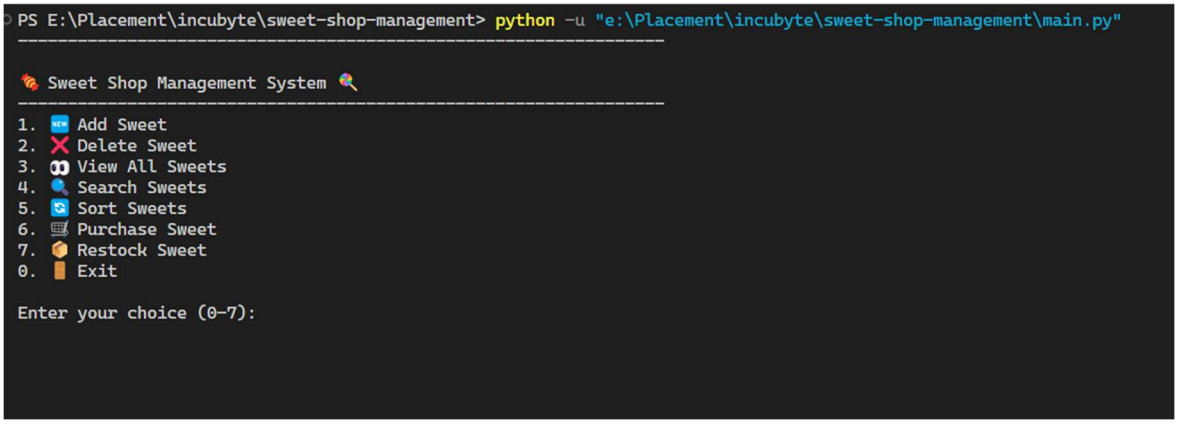
✅ All Unit Tests Passing

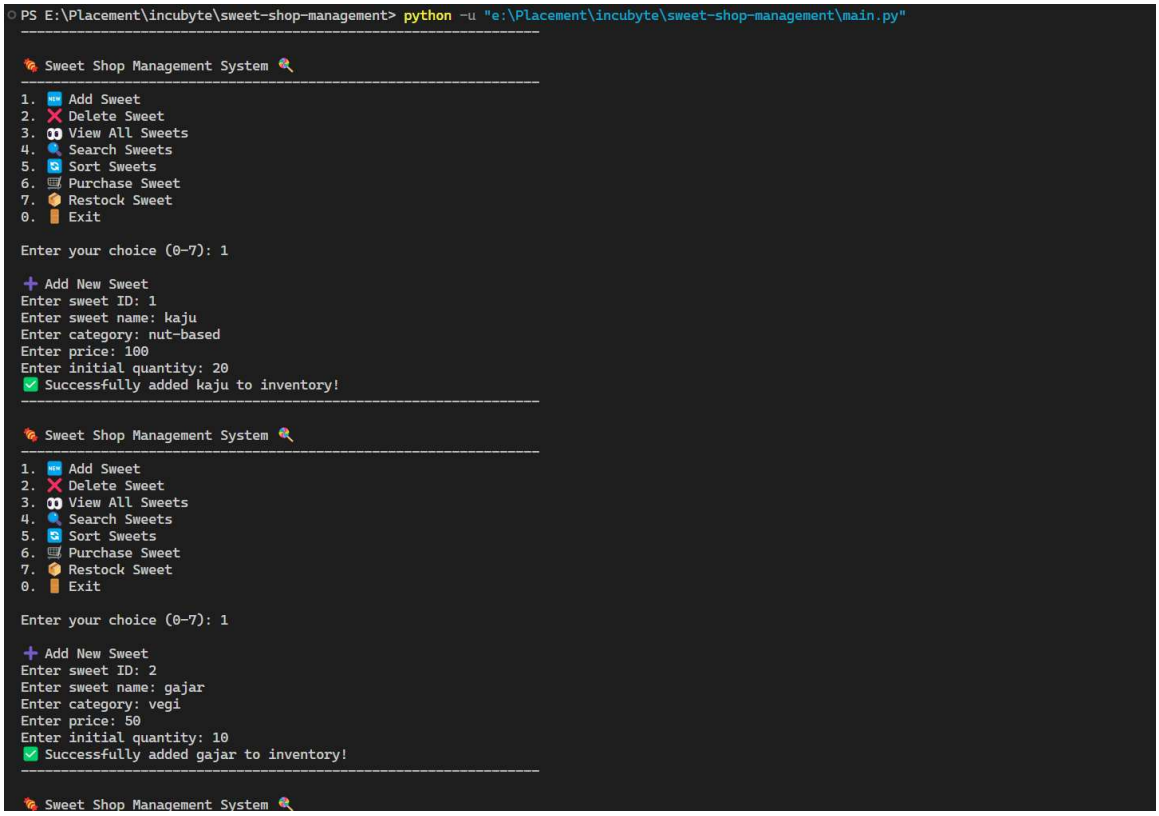
This screenshot demonstrates the successful execution of all unit tests using Python's `unittest` framework. It reflects full test coverage for core functionalities such as adding, deleting, purchasing, searching, sorting, and restocking sweets. Tests were developed and executed using a Test-Driven Development (TDD) approach.

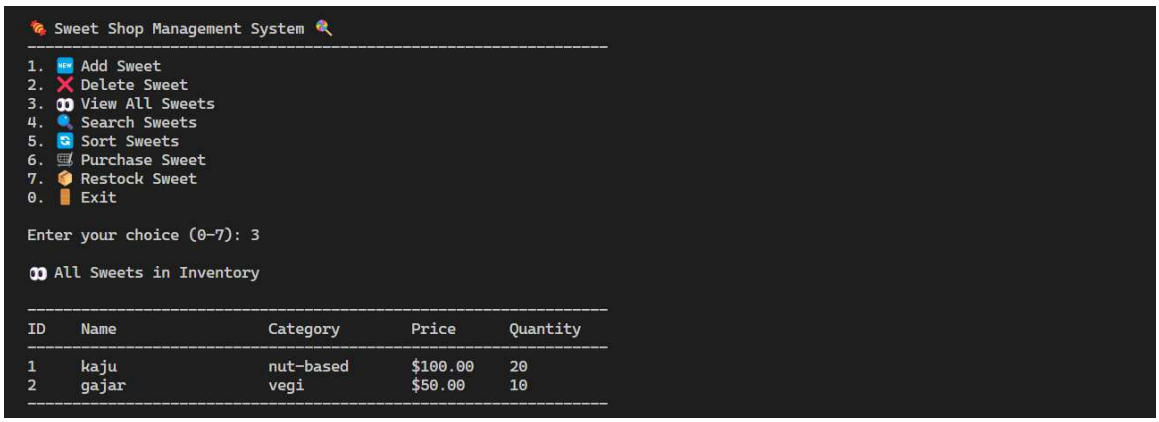


💉 Interactive CLI in Action

This screenshot captures the custom-built interactive command-line interface (main.py) designed for manual testing and validation. It allows real-time execution of operations like adding sweets, purchasing, and performing search/sort actions—ensuring live state updates and a seamless user experience for validation or demo purposes.




PS E:\Placement\incubyte\sweet-shop-management> python -u "e:\Placement\incubyte\sweet-shop-management\main.py"
Sweet Shop Management System
1. Add Sweet
2. Delete Sweet
3. View All Sweets
4. Search Sweets
5. Sort Sweets
6. Purchase Sweet
7. Restock Sweet
0. Exit
Enter your choice (0-7): 1
Add New Sweet
Enter sweet ID: 1
Enter sweet name: kaju
Enter category: nut-based
Enter price: 100
Enter initial quantity: 20
Successfully added kaju to inventory!
Sweet Shop Management System
1. Add Sweet
2. Delete Sweet
3. View All Sweets
4. Search Sweets
5. Sort Sweets
6. Purchase Sweet
7. Restock Sweet
0. Exit
Enter your choice (0-7): 1
Add New Sweet
Enter sweet ID: 2
Enter sweet name: gajar
Enter category: vegi
Enter price: 50
Enter initial quantity: 10
Successfully added gajar to inventory!
Sweet Shop Management System


Sweet Shop Management System
1. Add Sweet
2. Delete Sweet
3. View All Sweets
4. Search Sweets
5. Sort Sweets
6. Purchase Sweet
7. Restock Sweet
0. Exit
Enter your choice (0-7): 3
All Sweets in Inventory
ID Name Category Price Quantity
1 kaju nut-based $100.00 20
2 gajar vegi $50.00 10