Grocery Store Inventory CRUD System

Submitted by: Pushkara Bhardwaj

Registration No: 25BCY10105

Table of Contents:-

1) Introduction

The **Grocery Store Inventory CRUD System** is a simple inventory management application developed in **Python**.

It allows users to perform basic **CRUD operations**:

- **Create:** Add new products to inventory
- **Read:** View all products
- **Update:** Modify existing product details
- **Delete:** Remove products

Product data is stored persistently using **CSV files**, making the system lightweight and easy to manage.

This project is Designed by using the concepts of Python functions, Basic CSV File handling, Using the CSV module, and also using the basic functionalities of Git and Github

Overall, this project demonstrates how even a simple program can make day-to-day business operations faster, more accurate, and more organized. It also provides a strong foundation for building advanced inventory management systems in the future using databases and graphical user interfaces.

It also helps develop logical thinking and problem-solving skills by breaking a real-life store management problem into smaller programmable tasks.

2) Tools used in creation of this project

- **Programming Language:** Python 3.14.1
- **Data Storage:** CSV (Comma Separated Values)
- **IDE:** Spyder (Python)
- **Version Control:** Git & GitHub
- **OS:** Windows

3) System Design

The system is divided into two main components:

3.1 main.py :-

- **Provides a menu-driven interface for the user.**
- **Takes input for all product-related operations.**
- **Calls functions from `storage.py`**

3.2 storage.py:-

Contains all CRUD functions:

- `add_product()`
- `view_products()`
- `update_product()`
- `delete_product()`

Uses relative path to access the inventory CSV file:

```
INVENTORY_FILE = os.path.join("..", "data", "inventory.csv")
```

Ensures separation between code and data, and creates the data folder if it does not exist.

3.3 Data Flow Diagram:-

User → main.py → storage.py → inventory.csv

4) Features

The project supports the following operations:-

- **Add Products: Input Product ID, Name, Quantity, and Price.**
- **View Products: Display the list of all products in inventory.**
- **Update Products: Modify name, quantity, or price of existing products.**
- **Delete Products: Remove a product using its ID.**
- **Exit: Close the program safely.**

5) How to Run the Project

- **Make sure Python is installed:**
- `python --version`
- **Clone or download the repository:**
- `git clone https://github.com/Pushkar10105/Grocery-Store-Inventory-CRUD-System.git`
- **Navigate to the project folder:**
- `cd Grocery-Store-Inventory-CRUD-System`
- `cd source_code`
- **Run the main program:**

- `python main.py`

- **Important:**
- `main.py` **and** `storage.py` **must remain in source_code**
- `inventory.csv` **must stay in data folder**
- **Run the program from inside the source_code folder, otherwise Python will not find the CSV file.**

6) Screenshots

6.1 Main Menu:-

```
--- Grocery Store Inventory ---
1. View Products
2. Add Product
3. Update Product
4. Delete Product
5. Exit
Enter your choice (1-5):
```

Above is the main displayed running main.py here we can select the desire option according to our needs

6.2 Adding a Product:-

```
--- Grocery Store Inventory ---
1. View Products
2. Add Product
3. Update Product
4. Delete Product
5. Exit
Enter your choice (1-5): 2
Enter Product ID: 25
Enter Product Name: Chips
Enter Quantity: 1
Enter Price: 20
Product Chips added successfully.
```

Here we choose option 2 so we can add the product in our inventory system which gets saved on to the inventory.csv files which we can access through the excel sheet or the note pad

6.3 Viewing Products:-

```
--- Grocery Store Inventory ---
1. View Products
2. Add Product
3. Update Product
4. Delete Product
5. Exit
Enter your choice (1-5): 1

Current Inventory:
['ProductID', 'ProductName', 'Quantity', 'Price']
['25', 'Chips', '1', '20.0']
```

In this operation we have chosen option 1 to check our recently added product we can access the product by using its product id

Then we are displayed with all the data about the product in a list format

6.4 Update Product

```
--- Grocery Store Inventory ---
1. View Products
2. Add Product
3. Update Product
4. Delete Product
5. Exit
Enter your choice (1-5): 3
Enter Product ID to update: 25
Enter new Name (or press Enter to skip):
Enter new Quantity (or press Enter to skip): 5
Enter new Price (or press Enter to skip): 100
Product 25 updated successfully.
```

Here we are performing the update operation , we can access the product by the product id then the program asks us to update the value of our desire or we can just skip the part pressing the enter key on our keyboard

7) Conclusion

This project demonstrates a practical implementation of Python programming and file handling for managing a simple grocery store inventory.

Learning Outcomes:-

Understanding modular programming with functions and separate files.

File handling with CSV for persistent storage.

Using Git and GitHub for version control.

Designing menu-driven CLI applications.