

=====

PROJECT REPORT

=====

PROJECT TITLE: Grocery Store Management System

SUBMITTED BY: Rajdeep Kumar

COURSE: B.Tech

SUBJECT: Introduction to problem solving in python

DATE: November 23, 2025

=====

TABLE OF CONTENTS

=====

1. Introduction
 2. Problem Statement
 3. Objectives
 4. System Requirements
 5. System Design & Logic
 6. Implementation Details
 7. Limitations
 8. Future Scope
 9. Conclusion
- =====

1. INTRODUCTION

In the retail sector, efficient management of inventory is the backbone of

operational success. Traditional methods of maintaining stock records via paper ledgers are time-consuming, prone to human error, and difficult to update in real-time.

The "Grocery Store Management System" is a software solution developed using Python. It is a Console-Based Application designed to help store owners manage their stock efficiently. The system allows for the digital entry of products, real-time stock viewing, and easy modification of item details, ensuring that the inventory data remains accurate and accessible.

2. PROBLEM STATEMENT

Manual inventory management faces several challenges:

- Difficulty in tracking available stock levels.
- Time consumption in searching for product prices.
- Errors in calculation and record updates.
- Physical registers can be lost or damaged.

This project aims to solve these issues by digitizing the records into a centralized computer program.

3. OBJECTIVES

The primary objectives of this project are:

- To replace manual paperwork with a digital interface.
- To implement CRUD operations (Create, Read, Update, Delete) effectively.
- To provide a user-friendly menu-driven interface.

- To demonstrate the use of Python Data Structures (Dictionaries/Lists) for data management.

4. SYSTEM REQUIREMENTS

Hardware Requirements:

- Processor: Intel Core i3 or later.
- RAM: 4GB (Minimum 2GB).
- Hard Disk: 100MB free space.

Software Requirements:

- Operating System: Windows 10/11, Linux, or macOS.
- Programming Language: Python 3.x.
- IDE/Editor: VS Code, PyCharm, or Python IDLE.

5. SYSTEM DESIGN & LOGIC

The system operates on a standard Input-Process-Output cycle.

Data Structure Used:

The project utilizes a Python Dictionary for storing data because it allows for fast lookups ($O(1)$ complexity) based on unique Item IDs.

- Structure: `inventory = { 'ID': ['Name', Price, Quantity] }`

Flow of Control:

1. Start Program.
2. Display Main Menu.

3. User selects an option (1-5).
4. Based on selection, the corresponding function is executed.
5. Loop continues until the User selects "Quit".

6. IMPLEMENTATION DETAILS

The project is modularized into specific functions to ensure clean code management:

A. add_item():

- Prompts user for ID, Name, Price, and Quantity.
- Validates if ID already exists to prevent duplicates.
- Stores data in the global inventory dictionary.

B. show_items():

- Checks if inventory is empty.
- Uses a formatted loop to display data in a tabular format.

C. edit_item():

- Searches for an item by ID.
- If found, allows the user to input new details (Name, Price, Stock).
- Updates the dictionary value.

D. remove_item():

- Accepts an ID from the user.
- Uses the 'del' keyword to remove the item from the system.

E. main():

- The driver function containing the infinite 'while' loop.
- Handles menu navigation and function calls.

7. LIMITATIONS

While functional, the current version has specific limitations:

1. Volatile Memory: Data is stored in RAM (variables). Closing the program results in data loss as there is no file storage integration.
2. Single User: The system is designed for a standalone terminal, not for multiple concurrent users.
3. Input Validation: Basic validation is missing; entering text into numeric fields (like Price) causes the program to crash.

8. FUTURE SCOPE

Future enhancements for this project include:

- Database Connectivity: Connecting to SQL or using CSV/TXT files to save data permanently.
- Billing Module: Adding a feature to generate bills for customers based on items selected.
- GUI (Graphical User Interface): Migrating from a command-line interface to a window-based application using Tkinter.
- Auth System: Adding a login screen for security.

9. CONCLUSION

The Grocery Store Management System successfully demonstrates the application of Python programming concepts to solve a real-world business problem. It provides a structured way to manage inventory, reducing the hassle of manual record-keeping. Working on this project provided valuable insights into algorithm design, data structures, and modular programming.

END OF REPORT
