



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: <https://doi.org/10.22214/ijraset.2026.78214>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Supermarket Management System Using Python Flask and MySQL

Shinde Snehal Ganesh¹, Bhagat Sakshi Sanjay², Prof.Lamkhade J.G³

Computer Engineering Department, Samarth Polytechnic Belhe

Abstract: *The Supermarket Management System is a web-based solution designed to automate the daily operations of small and medium supermarkets. The system integrates product management, customer management, supplier tracking, billing, and reporting in a single platform. Built using Python Flask, MySQL, and Docker, it provides a secure and user-friendly interface for different user roles. The system ensures accurate billing, real-time inventory management, and efficient data handling, reducing manual errors and improving operational efficiency.*

Keywords: *Supermarket Management System, Retail Automation, Python Flask, Inventory Management, Point of Sale, MySQL*

I. INTRODUCTION

The Supermarket Management System is a web-based software application developed to automate and simplify the daily operations of a supermarket. Traditional supermarket operations often rely on manual processes for billing, stock management, and record keeping. These manual methods are time-consuming and can lead to errors in billing, inventory tracking, and data management.

To overcome these problems, a computerized system is developed using modern technologies. The Supermarket Management System helps store owners and employees manage products, customers, suppliers, sale transactions, and inventory efficiently.

This system is built using Python Flask for the backend, MySQL for the database, and Docker for containerized deployment. The application provides a secure and user-friendly web interface that allows different types of users such as Admin, Manager, and Staff to access the system according to their roles.

The system improves accuracy, reduces manual workload, and allows real-time monitoring of supermarket operations.

II. PROBLEM STATEMENT

In the current retail environment, many supermarkets still rely on manual or semi-automated systems for billing, inventory management, and sales tracking. These traditional methods present several significant issues:

Inefficiency: Manual calculation and billing take time, leading to long queues and customer dissatisfaction.

Human Error: Manual data entry is prone to mistakes in calculation, stock updates, and pricing.

Inventory Tracking Issues: Without a centralized database, it is difficult to know real-time stock levels, leading to "out of stock" or "overstock" situations.

Lack of Analytics: Shop owners lack easy access to data regarding daily sales, most popular products, and profit margins.

The "Supermarket Management System" aims to solve these problems by automating the billing process, managing inventory in real-time, and generating accurate reports using a Python-based application backed by a robust MySQL database.

III. LITERATURE REVIEW

Several studies have focused on the development of automated retail management systems to improve operational efficiency in supermarkets and retail stores. Smith et al. (2020) proposed a web-based retail management platform that integrates billing and inventory tracking features to reduce manual workload and improve transaction accuracy. Their study demonstrated that digital retail systems significantly reduce billing errors and improve inventory visibility.

Kumar and Patel (2021) developed a Point-of-Sale (POS) system integrated with a centralized database to automate product sales and stock management. Their system allowed store owners to monitor product availability and generate sales reports in real time.

Similarly, Sharma et al. (2019) introduced a retail automation system that utilizes a web interface and database backend for managing product inventories and customer transactions. The system improved operational efficiency but required expensive proprietary software solutions.

Although these systems provide useful automation features, many of them are costly and difficult for small-scale supermarkets to adopt.

Therefore, there is a need for a lightweight and cost-effective solution built using open-source technologies. The proposed Supermarket Management System addresses this problem by implementing a scalable web-based platform using Python Flask and MySQL.

IV. SYSTEM ARCHITECTURE

The proposed Supermarket Management System follows a three-layer architecture consisting of the presentation layer, application layer, and database layer. This architecture separates the user interface, application logic, and data storage components, which improves system scalability, maintainability, and performance.

The Presentation Layer represents the user interface of the system. Users interact with the application through webpages such as the login page, product management interface, POS terminal, and reporting dashboard. This layer is responsible for displaying information to the user and collecting user input.

The Application Layer contains the backend logic of the system implemented using the Python Flask framework. This layer handles core business processes including user authentication, billing calculations, inventory updates, and report generation. It acts as an intermediary between the user interface and the database.

The Database Layer is responsible for storing and managing all system data. The MySQL database stores information related to users, products, customers, suppliers, and sales transactions. The database ensures data integrity and supports efficient retrieval of information required by the application.

V. SYSTEM MODULES

The Supermarket Management System consists of several functional modules designed to automate different aspects of supermarket operations.

The User Management Module is responsible for handling system authentication and user account management. It provides secure login functionality and role-based access control.

Different user roles such as administrator, manager, and staff members are defined within the system, and each role is granted specific permissions to access system features.

The Product Management Module manages product information stored in the system database. This module allows administrators to add new products, update existing product details, delete products, track product quantities, and manage pricing information.

The Inventory Management Module monitors stock levels of products available in the supermarket. It automatically updates product quantities after each sale transaction and provides alerts when product stock reaches a low level. This module also supports bulk product import using CSV files to simplify large-scale inventory updates. The Point-of-Sale (POS) Module is used by staff members during customer checkout. This module allows users to select products, scan barcodes, calculate the total bill amount, process sales transactions, and generate invoices. It ensures a fast and efficient billing process. The Customer Management Module stores customer information and purchase history. It allows administrators to add customer details, view customer purchase records, manage loyalty points, and maintain customer ledgers. This module helps improve customer relationship management.

The Supplier Management Module maintains records of suppliers who provide products to the supermarket. It allows users to add supplier information, update supplier details, track products supplied by vendors, and maintain supplier contact records.

The Reporting and Analytics Module generates reports that help administrators analyze supermarket performance. The system can produce daily sales reports, monthly sales reports, product sales statistics, and inventory status reports, enabling store owners to make informed business decisions.

VI. DATA BASE DESIGN

The system database is designed using a relational model implemented in MySQL. Several tables are used to manage different types of data associated with supermarket operations.

The User stable stores information about system users including their unique user ID, name, email address, password, and assigned role.

The Product stable maintains details of products available in the supermarkets such as product ID, product name, price, quantity, and product category.

Customer information is stored in the Customer stable, which includes fields such as customer ID, customer name, and phone number.

Supplier records are stored in the Supplier stable, which contains supplier ID, supplier name, and contact information.

Sale transactions are recorded in the Sale table, which includes fields such as sale ID, transaction date, customer reference, and total transaction amount.

Each sale transaction may contain multiple products, which are stored in the Sale_Items table. This table records item ID, sale ID, product ID, quantity purchased, and item price.

This relational database structure ensures efficient organization of supermarket data and supports accurate tracking of product sales and inventory levels.

VII. METHODOLOGY

The development of the Supermarket Management System follows a modular design approach to ensure flexibility, scalability, and ease of maintenance. The system workflow consists of several functional steps that automate daily supermarket operations.

First, the system performs user authentication through a secure login interface. Authorized users such as administrators and staff members can access different modules based on their assigned roles.

After successful authentication, the product management module allows administrators to add, update, or remove product information stored in the database. The inventory management component continuously monitors stock levels and updates product quantities after each sales transaction.

The Point-of-Sale (POS) module enables staff members to select products during customer checkout. The system automatically calculates the total bill based on selected products and their quantities.

Finally, all transaction data is recorded in the database, allowing the reporting module to generate daily or monthly sales reports. This workflow ensures efficient supermarket management while minimizing human errors in billing and inventory tracking.

VIII. IMPLEMENTATION

The Supermarket Management System is implemented using a combination of modern web technologies. The backend of the application is developed using the Python Flask framework, which provides a lightweight and efficient environment for handling web requests and application logic.

The frontend interface is designed using HTML, CSS, and JavaScript to provide an interactive and user-friendly web interface. These technologies enable responsive design and smooth interaction with the backend services.

MySQL is used as the relational database management system to store all application data including product information, customer records, and sales transactions.

To ensure portability and simplified deployment, the application environment is containerized using Docker. This allows the system to run consistently across different server environments while maintaining all dependencies required by the application.

IX. RESULTS AND DISCUSSION

The implemented Supermarket Management System was evaluated in a simulated retail environment to analyze its performance and operational efficiency. The system successfully automated key supermarket functions including billing, inventory tracking, and report generation.

The Point-of-Sale (POS) module enabled quick product selection and automatic bill calculation, significantly reducing the time required for customer checkout compared to manual billing systems. Additionally, the system automatically updated inventory levels after each completed transaction, ensuring accurate stock management.

The reporting module generated useful insights such as daily sales summaries and product demand trends, which can assist store administrators in making informed business decisions.

Overall, the experimental results indicate that the proposed system improves operational efficiency, reduces manual workload, and enhances the accuracy of retail management processes.

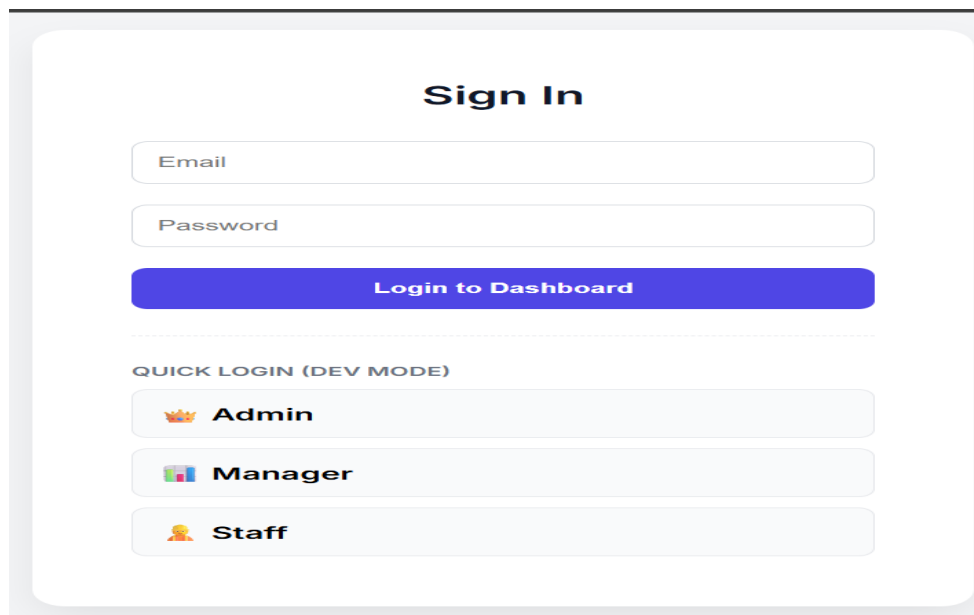


Fig.1 Login Interface of the Supermarket Management System

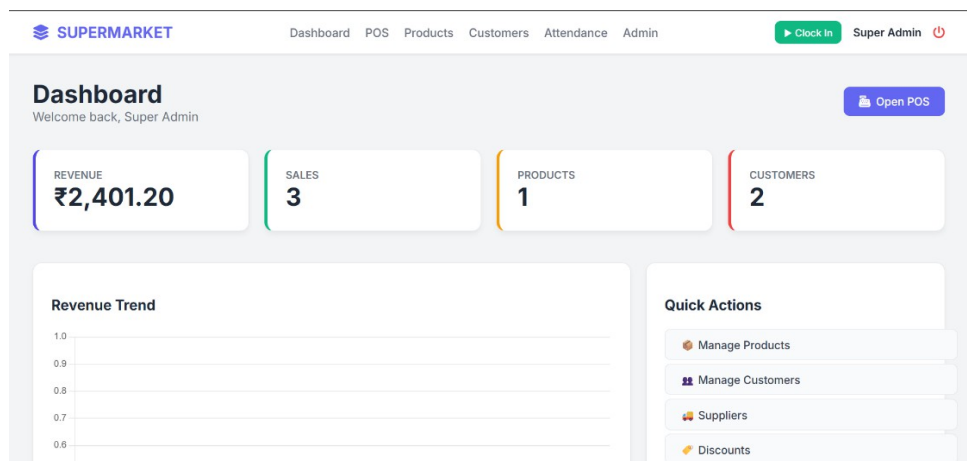


Fig.2 Dashboard of the Supermarket Management System

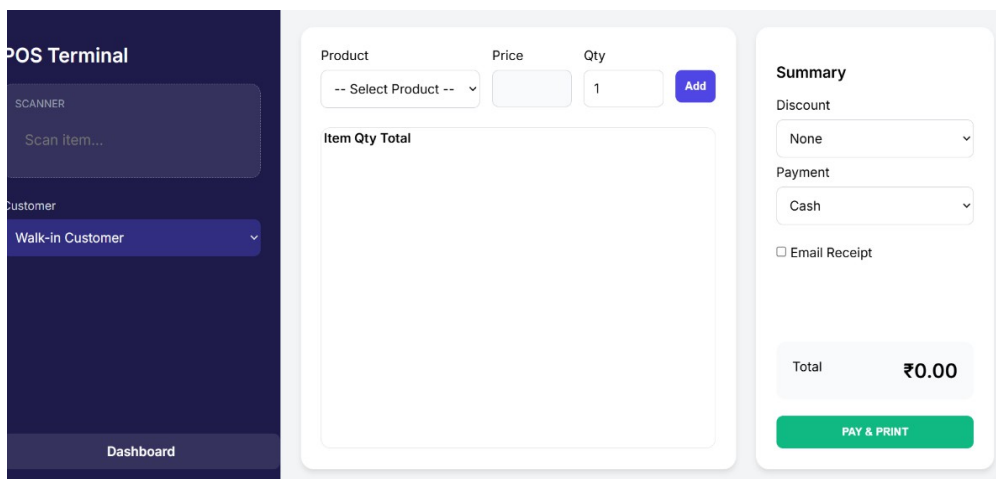
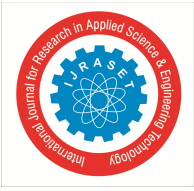


Fig.3 Point of Sale Billing Interface



X. CONCLUSION

The Supermarket Management System is designed to automate and simplify the daily operations of a supermarket. The system helps in managing product details, billing processes, and inventory records in an efficient way. By implementing this system, manual work is reduced and the chances of human errors in billing and stock management are minimized.

The developed system allows the user to easily add, update, and delete product information. It also generates bills automatically and updates the inventory after each transaction. This improves the accuracy and speed of supermarket operations.

Overall, the implementation of the Supermarket Management System provides a reliable and efficient solution for managing supermarket activities. It helps store owners maintain proper records, improve productivity, and save time.

REFERENCES

- [1] PythonSoftwareFoundation,PythonDocumentation.
- [2] FlaskFrameworkDocumentation,PalletsProjects.
- [3] OracleCorporation,MySQLReferenceManual.
- [4] MerkelD.,'Docker:LightweightLinuxContainersforConsistentDevelopmentand Deployment,' IEEE Software, 2014.
- [5] Pressman, R. S., Software Engineering: A Practitioner's Approach, McGraw-Hill Education.
- [6] Sommerville, I., Software Engineering, Pearson Education.
- [7] Python Software Foundation, Python Documentation, Available at: <https://www.python.org>◆
- [8] Oracle Corporation, MySQL Database Documentation, Available at: <https://www.mysql.com>◆
- [9] Various online tutorials and research articles related to supermarket management systems.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)