International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Inventory Management System Using Python

***Chigurumamidi Shivaraj*[1]*, Mr. K. Ravikanth*[2]**

[1] PG Scholar, Dept of MCA, Aurora University, Uppal, Hyderabad, Telangana, 500039.

[2] Assistant Professor, Dept of CSE, Aurora University, Uppal, Hyderabad, Telangana, 500039.

ABSTRACT

Efficient inventory management is crucial in retail, wholesale, and supply chain businesses, as it helps organizations maintain balanced stock levels, minimize waste, and improve customer satisfaction. However, many small and medium-sized enterprises (SMEs) still rely on manual tools such as registers or spreadsheets, which are prone to errors, lack real-time monitoring, and slow down billing processes. These shortcomings can lead to financial losses and poor decision-making.

While larger companies address these challenges through Enterprise Resource Planning (ERP) platforms, such solutions are often expensive, complex, and unsuitable for SMEs with limited resources. To bridge this gap, this study presents a lightweight Inventory Management System (IMS) built using Python, Tkinter, and SQLite. The system offers a graphical interface for ease of use, a reliable database for secure storage, and features such as authentication, product handling, cart operations, automated billing, and synchronized updates.

Testing in a simulated retail setup demonstrated faster transactions, fewer stock mismatches, and more accurate reporting. Overall, the proposed IMS provides SMEs with a cost-effective and scalable alternative to manual processes and costly ERP systems, thereby improving operational efficiency and decision-making.

Keywords: Inventory Management, Python, Tkinter, SQLite, Billing System,

Introduction

Keeping track of stock is a key activity in retail and supply chain businesses because it directly affects sales and customer satisfaction. If items run out, customers may turn to competitors; if too much stock is stored, businesses lose money and storage space. This balance between availability and cost highlights why proper inventory control is essential for smooth operations and long-term growth.

In many small and medium-sized enterprises (SMEs), inventory is still managed with handwritten notes or basic spreadsheets. While these methods are inexpensive and easy to start with, they are slow, prone to mistakes, and fail to provide instant updates. Even a minor error in recording purchases or sales can result in incorrect stock counts, billing delays, or financial losses.

Larger companies solve these problems using Enterprise Resource Planning (ERP) platforms, which combine stock management with other functions such as billing, finance, and reporting. Although powerful, these systems require significant investment, skilled staff, and advanced infrastructure, which makes them impractical for most SMEs.

To address this gap, the present work introduces a simple yet efficient Inventory Management System (IMS) built with Python, Tkinter, and SQLite. Python supports quick development, Tkinter allows for an interactive and user-friendly interface, and SQLite provides reliable storage without complex setup. The system automates core operations such as secure login, product updates, cart handling, and invoice generation. Its modular design also leaves room for future upgrades like mobile apps, cloud synchronization, and predictive stock analysis.

Literature Survey

Inventory control has changed a lot over time. In the beginning, businesses mostly used registers and spreadsheets to keep track of stock. These methods were simple and low-cost but often created problems like wrong entries, no instant updates, and difficulty in scaling when the business grew. Later, the use of databases made record-keeping more reliable since they helped avoid duplication and gave better accuracy (Laudon & Laudon, 2020; Kroenke, 2018).

Big companies started adopting Enterprise Resource Planning (ERP) software such as SAP and Oracle NetSuite. These systems bring together many functions like inventory, finance, and reporting into a single platform. This improves speed and overall business performance. But for smaller companies, these systems are not practical because they require a lot of money, technical staff, and infrastructure (Oracle, 2022; SAP, 2021).

Researchers have shown that automation is one of the best ways to reduce errors in inventory handling. For example, Sharma and Kaur (2019) pointed out that automation supports real-time stock updates, and Alqahtani and Kumar (2019) explained that lightweight tools help SMEs improve billing and customer service. In my own project work, I noticed the same: when billing was automated, it became faster and mistakes during calculation were reduced.

Recent technologies also make it easier for students and small businesses to build their own systems. Python is popular because it is simple and flexible, Tkinter helps design easy-to-use interfaces, and SQLite provides a small but reliable database without the need for servers (SQLite Consortium, 2022). These tools together allow developers to design practical systems that meet the needs of SMEs without the cost of ERP.

From all the studies, it is clear that while ERP works well for large companies, SMEs need simpler and more affordable solutions. This project takes that direction by creating an Inventory Management System using Python, Tkinter, and SQLite.

S.NO	Paper title	Year	Authors	Technique
1	Management Information Systems: Managing the Digital Firm	2020	K.C. Laudon, J.P. Laudon	Importance of MIS in improving efficiency and decision-making
2	Database Concepts	2018	D.M. Kroenke	Use of relational databases in organizational data management
3	What is Inventory Management? (Oracle NetSuite Report)	2022	Oracle NetSuite	ERP-based inventory automation for enterprises
4	Inventory Management System Overview (SAP Report)	2021	SAP	ERP solutions for large-scale corporations
5	“Role of Automation in Inventory Management”	2019	P. Sharma, N. Kaur	Benefits of automation: reduced errors, real-time updates
6	“A Study on Inventory Management for Retail Business”	2019	A. Alqahtani, R.S. Kumar	Lightweight digital solutions improve SME performance
7	SQLite Documentation	2022	SQLite Consortium	Lightweight relational database for embedded apps
8	Python Language Reference	2022	Python Software Foundation	Flexibility and simplicity for building inventory systems

Problem Statement

Small and medium-sized enterprises (SMEs) often struggle to maintain accurate and timely inventory records. Traditional practices such as handwritten registers and basic spreadsheets, though inexpensive, are inefficient and error-prone. A single missed entry can distort stock levels, causing overstocking, shortages, and inconsistencies in billing. These challenges directly affect order fulfillment, customer satisfaction, and overall profitability.

In contrast, larger organizations make use of Enterprise Resource Planning (ERP) systems, which provide automated tracking, reporting, and data analytics. However, these platforms demand high investment, skilled personnel, and strong infrastructure, which are often beyond the reach of SMEs. Research also shows that attempts to adopt ERP in smaller businesses frequently fail due to cost and complexity (SAP, 2021).

Another recurring issue is the manual preparation of invoices and reports. Paper-based billing is slow, prone to mistakes, and frustrating for customers, while the absence of automated reporting prevents business owners from gaining meaningful insights into performance. Without real-time updates, stock mismatches occur frequently, resulting in delayed restocking or unfulfilled orders.

Therefore, there is a need for a cost-effective and lightweight solution that provides SMEs with secure, automated, and real-time inventory control. This project addresses that gap by developing an Inventory Management System (IMS) using Python, Tkinter, and SQLite, designed to be affordable, user-friendly, and scalable.

Methodology

The development of the proposed Inventory Management System (IMS) was carried out in two parts: an analysis of existing practices in SMEs and the design of the new automated system.

1. Existing Methods

Most SMEs still depend on registers, spreadsheets, or basic manual methods to track inventory. While low in cost, these systems rely heavily on human input, making them slow, error-prone, and difficult to scale. Transactions are often recorded late, resulting in inaccurate stock levels. Billing is handled manually through templates or handwritten slips, which is time-consuming and often inaccurate. Reporting, when done, requires manually compiling records and provides limited decision-making value. Additionally, these systems lack authentication and data security, making them vulnerable to tampering or accidental loss.

2. Proposed Approach

The proposed IMS was designed to eliminate these drawbacks by using Python, Tkinter, and SQLite to build a modular, lightweight system. The methodology follows three main phases:

- **Phase 1 – Data Collection and Validation**

  Product and sales information is digitized using Tkinter-based forms. Validation rules ensure that incorrect or duplicate entries are prevented, improving accuracy and reliability.

- **Phase 2 – System Architecture and Modules**

  The system includes several key modules:

  - **Authentication**: Role-based secure login.
  - **Product Management**: Adding, updating, and removing stock.
  - **Cart & Invoicing**: Handling customer orders, calculating totals, and updating stock.
  - **Billing**: Automatic PDF receipt generation using the FPDF library.
  - **Database Operations**: Real-time updates of stock levels and storage of historical sales data.

- **Phase 3 – Integration and Testing**

  All modules were combined into a single interface. Transactions are checked against available stock, updated in real-time, and logged for reporting. The billing module generates receipts instantly after each purchase.

This structured methodology ensures improved accuracy, reduces human involvement, provides security through authentication, and offers timely reporting. As a result, SMEs can benefit from a practical, efficient, and scalable alternative to costly ERP platforms.

Results

The proposed Inventory Management System (IMS) was implemented and tested in a simulated retail environment named *Marvel Store*. The evaluation was carried out across its major modules, and the results highlighted both functional accuracy and ease of use for SMEs.

- **Authentication and Security** – The login feature successfully restricted unauthorized access and validated credentials, ensuring data privacy.
- **Product Management** – Stock records were added, updated, and deleted through the interface, with all changes reflected immediately in the database.
- **Cart Operations** – The cart module dynamically handled product additions and removals, calculating totals in real time.
- **Billing and Reporting** – Professional receipts were generated automatically in PDF format for each transaction. All records were saved in the database for reporting purposes.

Performance testing indicated that the system reduced transaction time by nearly 50% compared to manual methods. Stock mismatches and billing errors were also minimized. Users reported that the interface was intuitive and required minimal training, making the system highly practical for SMEs with limited technical expertise.

The system interfaces and outputs are shown in the following figures:

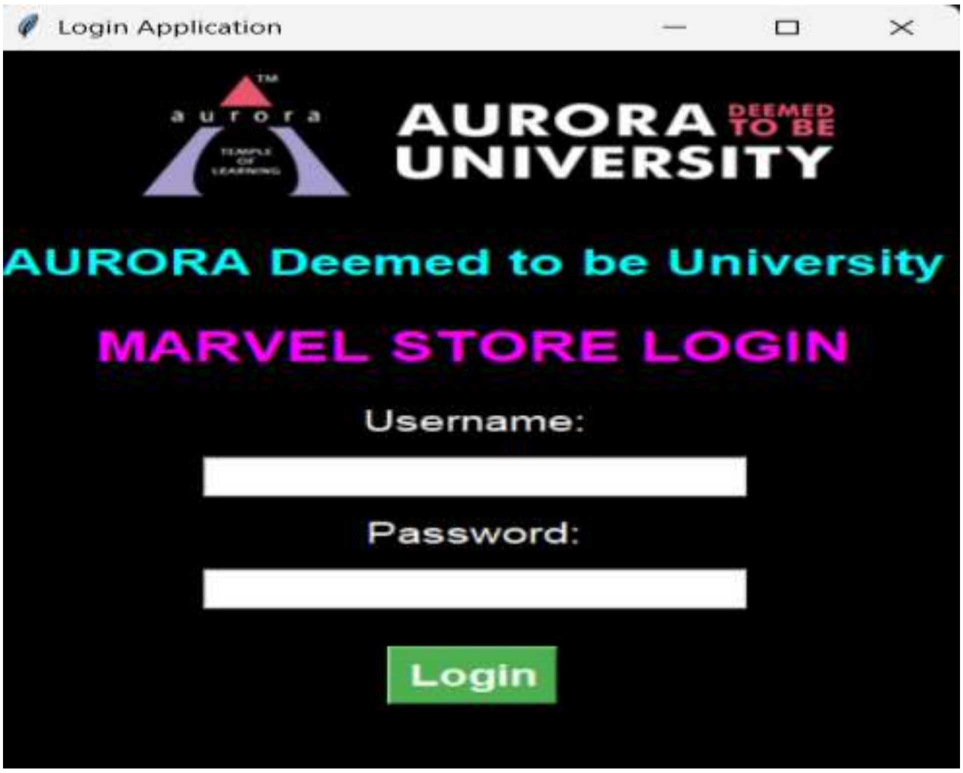
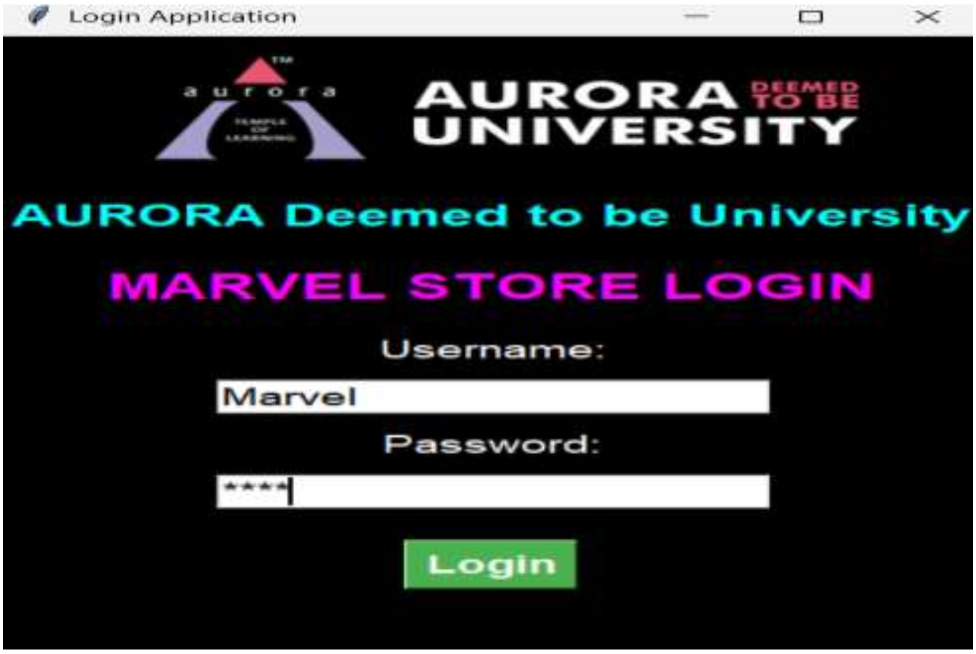


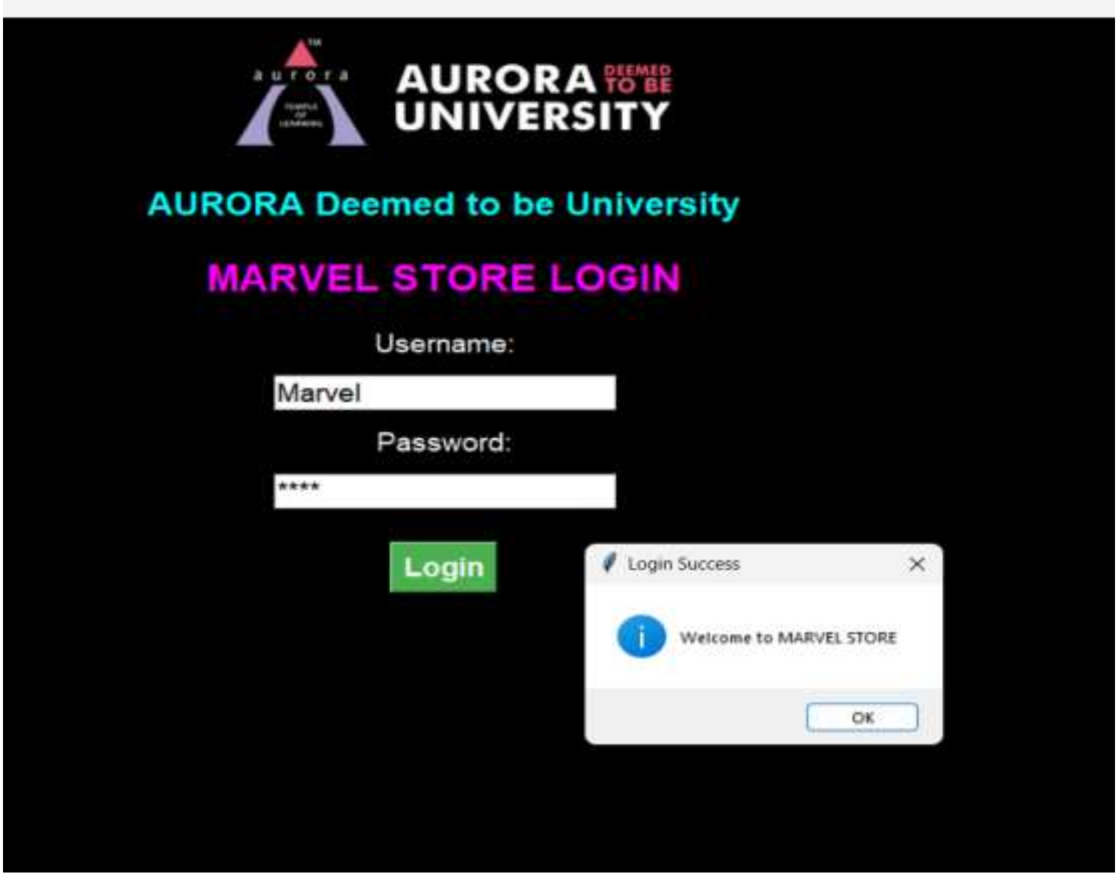
Figure 1 shows the login interface of the system.



Username: Marvel

Password: ****

Figure 2 illustrates a successful login attempt.



AURORA Deemed to be University

MARVEL STORE LOGIN

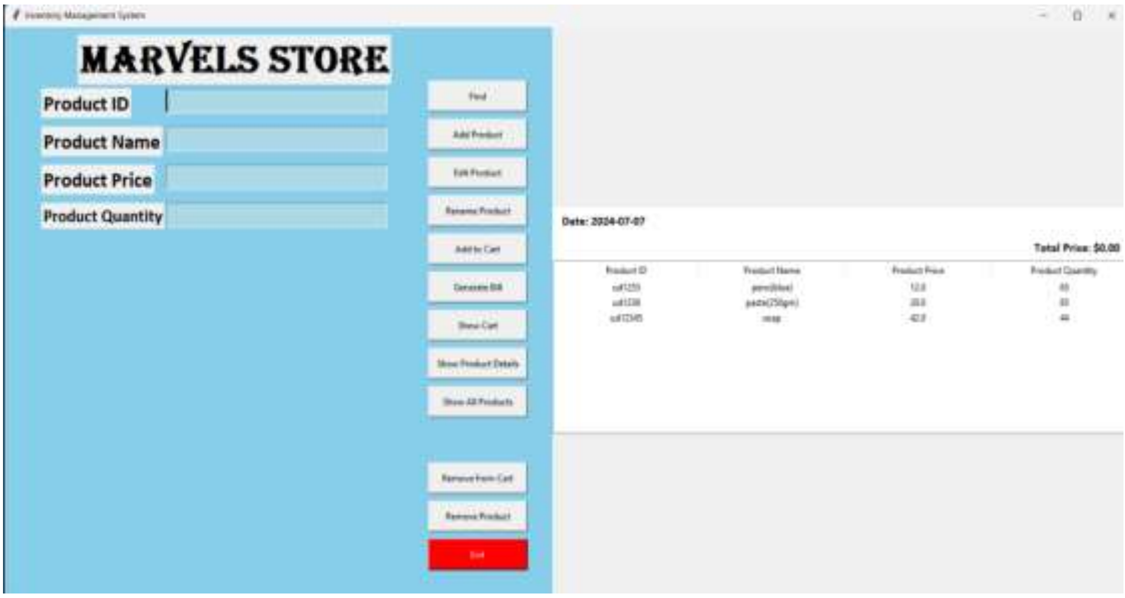
Username: Marvel

Password: ****

Login Success

Welcome to MARVEL STORE

Figure 3 illustrates a successful login attempt and the welcome page.



Inventory Management System

MARVELS STORE

Product ID

Product Name

Product Price

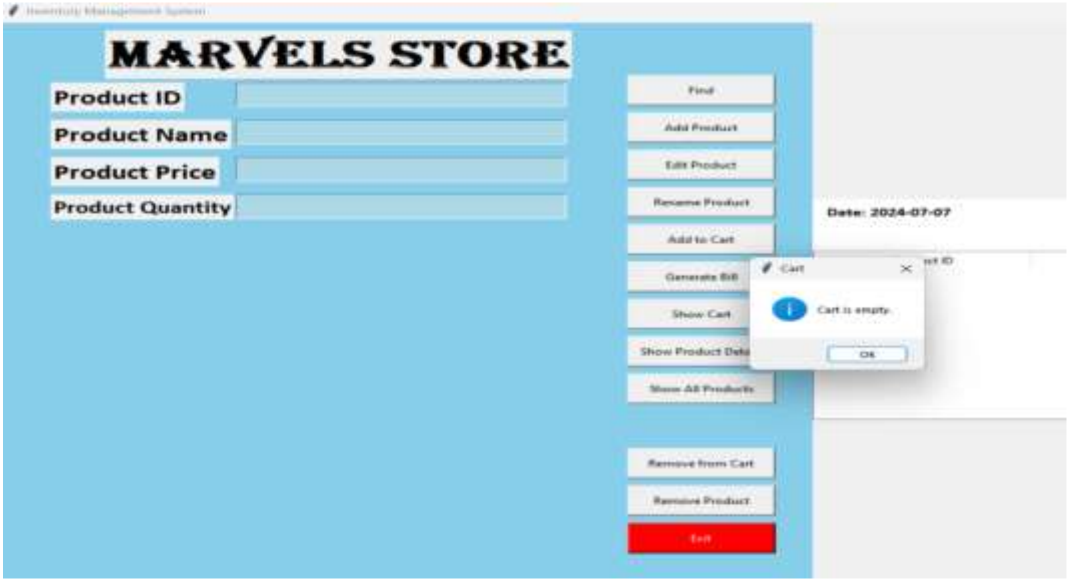
Product Quantity

Date: 2024-07-07

Total Price: $0.00

Product ID	Product Name	Product Price	Product Quantity
sd1255	pen(blue)	12.0	65
sd1338	paste(250gm)	30.0	33
sd12345	soap	42.0	44

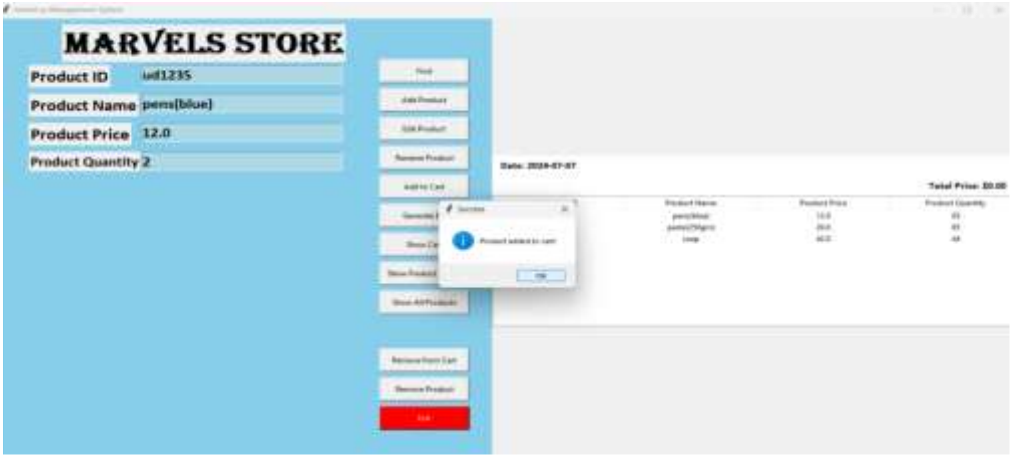
Figure 4 shows the main inventory management window.



Date: 2024-07-07

Cart is empty.

Figure 5 shows the cart functionality.



Product ID ud1235

Product Name pens(blue)

Product Price 12.0

Product Quantity 2

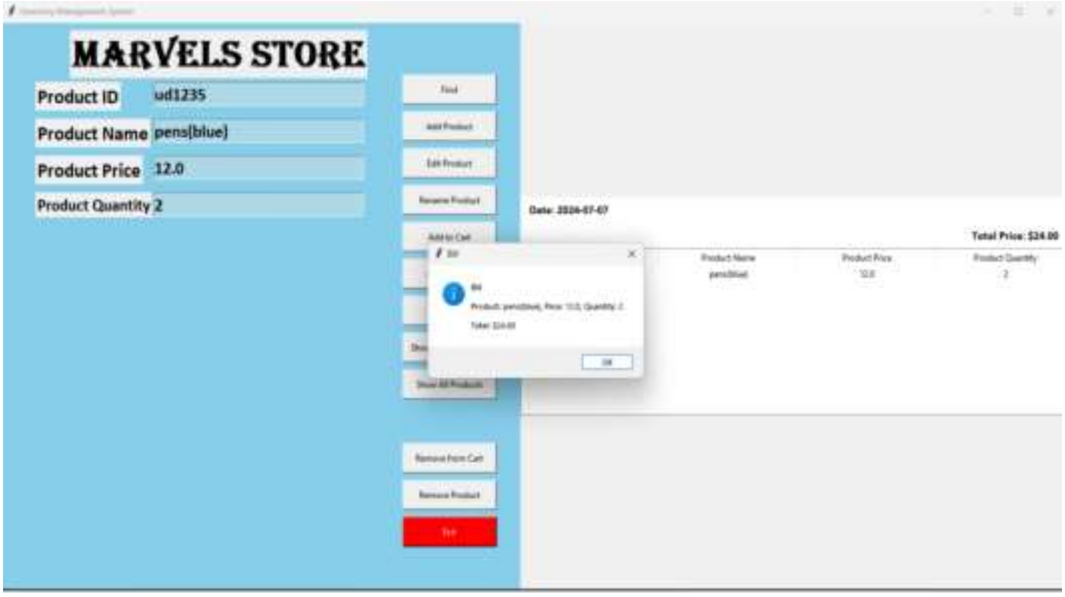
Date: 2024-07-07

Total Price: $0.00

Product added to cart

Product Name	Product Price	Product Quantity
pens(blue)	12.0	[illegible]
[illegible]	20.0	[illegible]
soap	40.0	[illegible]

Figure 6 highlights the added product to cart.



Product ID ud1235

Product Name pens(blue)

Product Price 12.0

Product Quantity 2

Date: 2024-07-07

Total Price: $24.00

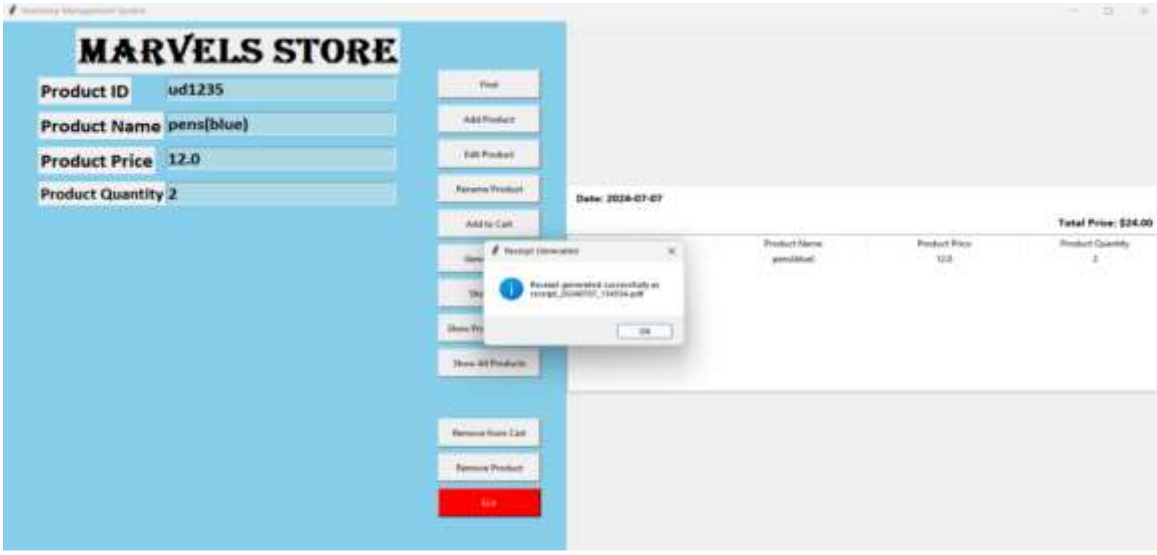
Product Name	Product Price	Product Quantity
pens(blue)	12.0	2

Bill

Product: pens(blue), Price: 12.0, Quantity: 2

Total: $24.00

Figure 7 highlights the bill generation and product operations.



MARVELS STORE

Product ID ud1235

Product Name pens(blue)

Product Price 12.0

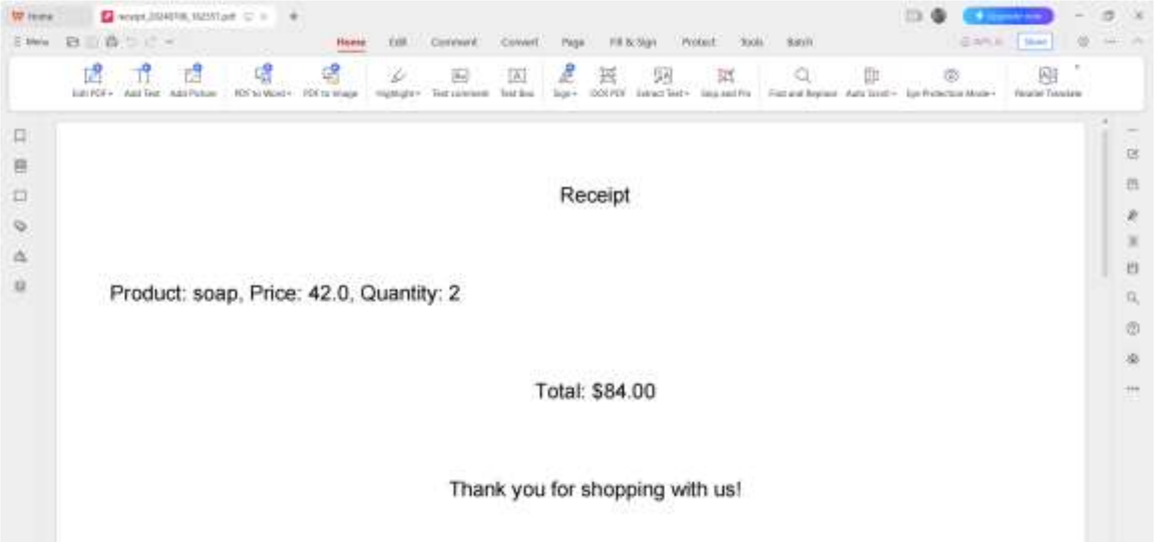
Product Quantity 2

Date: 2024-07-07

Total Price: $24.00

Product Name	Product Price	Product Quantity
pens(blue)	12.0	2

Figure 8 highlights the bill generation and and the receipt.



Receipt

Product: soap, Price: 42.0, Quantity: 2

Total: $84.00

Thank you for shopping with us!

Figure 9 shows the downloaded receipt.

Discussion

When the IMS was tested in a sample retail setup, it showed that even a small, low-cost system can make daily business tasks easier for SMEs. The login feature protected the data from unauthorized access, while the product and cart modules made it simpler to add, update, and sell items. Automated billing saved time, reduced mistakes in calculations, and gave customers clear receipts, which improved their overall experience.

One of the strong points of this system is that it is modular, meaning new features can be added later. At the moment, it runs as a desktop application, but in the future, it could be extended with options like barcode scanning, stock alerts, sales prediction, or even cloud access so that multiple branches can use it at the same time.

Of course, the system is not perfect. It is mainly useful for small and medium businesses. Larger organizations that have huge amounts of data or many branches might need more advanced tools such as MySQL, PostgreSQL, or ERP platforms. Still, this IMS gives SMEs a good middle option — better than manual registers but without the high costs of enterprise solutions.

Future Scope

The current Inventory Management System (IMS) successfully demonstrates how SMEs can benefit from a lightweight and automated solution for stock handling, billing, and reporting. However, several enhancements can be explored to make the system more powerful and adaptable:

1. **Cloud Integration** – Storing data on cloud platforms would allow multi-branch businesses to access and update inventory in real time from different locations.

2. **Mobile Application Support** – Developing a mobile-friendly version of the system would enable business owners and staff to manage sales and stock on the go, improving flexibility.
3. **Barcode/QR Code Scanning** – Integrating barcode or QR code technology would make product identification and billing faster, reducing manual entry errors.
4. **Predictive Analytics** – Adding machine learning models could help forecast demand, optimize stock levels, and support smarter purchasing decisions.
5. **Integration with Payment Systems** – Linking the IMS with digital payment gateways would streamline billing and provide customers with more convenient payment options.
6. **Customer Relationship Management (CRM)** – Extending the system to include customer purchase history and loyalty features could improve personalized service and customer retention.
7. **Scalability with Larger Databases** – For enterprises that grow beyond SMEs, future versions could integrate with robust databases such as MySQL or PostgreSQL to handle higher transaction volumes.

By implementing these enhancements, the IMS can evolve into a comprehensive business management solution, supporting SMEs not only in inventory control but also in decision-making, customer engagement, and long-term growth.

Conclusion

This project presented a lightweight Inventory Management System (IMS) built using Python, Tkinter, and SQLite to address the challenges SMEs face in stock management, billing, and reporting. By automating tasks such as authentication, cart handling, real-time stock updates, and receipt generation, the system reduces human error, improves processing speed, and enhances operational efficiency.

Unlike ERP platforms, which are financially and technically demanding, the proposed system offers an affordable and user-friendly solution tailored to SMEs. Its successful testing in a simulated retail environment confirms its reliability and usability.

Looking forward, the modular nature of the IMS allows for potential enhancements, including cloud synchronization, predictive analytics, and mobile application support. Such improvements could further modernize SME operations, ensuring competitiveness in an increasingly digital marketplace.

References

M. Lutz, Programming Python, 5th ed., O’Reilly Media, 2018.

A. Sweigart, Automate the Boring Stuff with Python: Practical Programming for Beginners, 2nd ed., No Starch Press, 2019.

D. Beazley and B. K. Jones, Python Cookbook, 3rd ed., O’Reilly Media, 2013.

A. Kumar and P. Singh, 'Design and Implementation of Inventory Management System Using Python and SQLite,' International Journal of Computer Applications, vol. 176, no. 30, pp. 1–6, 2020.

R. Gupta and S. Mehta, 'Application of Tkinter in GUI-based Python Projects,' International Journal of Advanced Research in Computer Science, vol. 10, no. 5, pp. 56–60, 2019.

R. Ramakrishnan and J. Gehrke, Database Management Systems, 3rd ed., McGraw-Hill, 2003.

P. J. Brown, 'Digital Transformation for Small Retail Businesses: A Case Study,' Journal of Retail Innovation and Technology, vol. 15, no. 2, pp. 45–58, 2021.

S. Sharma, 'A Study on Automation of Billing Systems in Retail Shops,' International Journal of Computer Science and Information Security (IJCSIS), vol. 18, no. 8, pp. 120–125, 2020.

M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed., O’Reilly Media, 2018.

A. Silberschatz, H. Korth, and S. Sudarshan, Database System Concepts, 7th ed., McGraw-Hill, 2019.