NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY
Nitte Karkala Taluk, Udupi-576124, Karnataka, India

A PROJECT REPORT ON

Python based Inventory Management System

NMAM INSTITUTE OF TECHNOLOGY, NITTE

(Off-Campus Centre, Nitte Deemed to be University, Nitte - 574 110, Karnataka, India)

In partial fulfilment of the requirements for the award of the

Degree of Bachelor of Technology
in
Information Science and Engineering

By

SOHAN A. NAIK
Reg. No: NNM24IS246

Under the Guidance of

Ms. Rashmi Hegde
Department of Humanities, Nitte

NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

Nitte Karkala Taluk, Udupi-576124, Karnataka, India

CERTIFICATE

This is to certify that the "Internship I report" submitted by SOHAN A. NAIK (NNM24IS246), of I year B.Tech., a Bonafide student of NMAM Institute of Technology, Nitte, has undergone two weeks of internship on "Python based Inventory Management System" during December 2024 fulfilling the partial requirements for the award of degree of Bachelor of Technology in Information Science Engineering at NMAM Institute of Technology, Nitte.

Name and Signature of Mentor

Signature of Dean(Academics)

ACKNOWLEDGEMENT

I would like to thank our beloved Principal, Vice-Principal, Dean of Academics, and Controller of Examinations for their motivation and support in carrying out this project work. I am happy to express our sincere gratitude to our project guide Ms. Rashmi Hegde, Department of Humanities, for the constant support and guidance in carrying out our project work. I also thank all our teachers and friends who supported us in our task of completing the project.

TABLE OF CONTENTS

INTRODUCTION

Effective inventory management is a crucial aspect of any business that deals with physical products. Poor stock tracking can lead to overstocking, stockouts, and inefficiencies in operations.

To address these challenges, my internship work focused on developing a Python-based Inventory Management System designed to automate and streamline the process of managing inventory in a small to medium-sized business environment.

The project aimed to create a simple yet functional system using Python, chosen for its readability, ease of development, and extensive library support.

A system has been designed to streamline the inventory process wherein any user can store, update, and retrieve key data effortlessly. It supports the management of product details, including product names, quantities, and prices helping business to maintain accurate stock records and avoid issues like overstocking or shortages. In addition to product tracking, the system also manages employee details, such as employee IDs, names, and age.

Technology like Tkinter was used for the graphical user interface, and SQLite served as the lightweight backend database.

This project provided valuable hands-on experience in application development, including requirement analysis, coding, testing, and deployment. It also offered insight into how software tools can be tailored to meet specific business needs and improve operational efficiency.

OBJECTIVES

1. To design and implement an inventory management system using Python

2. To support basic CRUD operations

3. To develop a user-friendly graphical user interface using Tkinter

4. To ensure efficient storage and retrieval of product information

5. To enable systematic management of employee details

6. To understand and apply real-world requirements of database integration

LITERATURE REVIEW

The paper by Chaudhary et al. (2020) addresses these challenges by developing an inventory management system using Python's Tkinter library for the interface and SQLite for the backend. This approach reflects a growing trend in the use of lightweight, open-source technologies to reduce costs and enhance scalability, as seen in previous works (Liao et al., 2018). The system's features, such as user authentication, transaction logging, and stock management, align with industry needs for real-time tracking, security, and data accuracy (Singh & Sharma, 2016). Moreover, the integration of SQLite as a serverless database and the use of Raspberry Pi as the deployment platform offer cost-effective solutions for small businesses, consistent with trends in embedded systems for inventory management.

The article by Python Mania presents a simple inventory management system using Python's Tkinter library for the GUI and an in-memory list for data storage. This project reflects the growing trend of utilizing Python in developing user-friendly applications for small-scale inventory management (Wang et al., 2017). The system allows users to add, edit, delete, and search for items, making it an excellent introductory project for developers. Its modular design, based on object-oriented principles, makes it adaptable for future expansion (Sommerville, 2011). The simplicity of the system is ideal for small businesses or educational purposes, offering an accessible tool for inventory management (Kumar & Sharma, 2019).

Nishant Gupta's GitHub project, *Inventory Management System*, exemplifies this trend by leveraging Python with Tkinter for the GUI and SQLite for backend data handling. This aligns with existing research that highlights the suitability of Python for rapid application development in business automation (Wang et al., 2017). Gupta's system is modular and feature-rich, including modules for employee, supplier, product, category, sales, and billing management. The use of object-oriented design and separation of functionality into distinct scripts reflects best practices in software engineering for maintainability and scalability (Sommerville, 2011). Moreover, the system's billing and invoice generation features address practical business needs, making it a robust tool for small and medium enterprises.

Ashley Gingeleski's tutorial, *"How to Build an Inventory App with Tkinter"*
(2021), offers a hands-on guide to developing such an application using Python's Tkinter for the GUI and SQLite for data storage. This reflects ongoing trends in leveraging open-source technologies for accessible and cost-effective business solutions (Wang et al., 2017). Gingeleski's implementation focuses on core functionalities—such as adding and displaying records—while maintaining a clean and user-friendly interface. Her modular, beginner-friendly approach supports the educational value of the project and aligns with software development practices that emphasize usability and maintainability (Sommerville, 2011).

Inventory management systems are essential for tracking stock levels, ensuring product availability, and supporting business operations. In their article, *"Inventory Management System using Python,"* the team at **Itxperts** outlines the development of a basic yet functional inventory system using Python's Tkinter for the graphical user interface and SQLite for the database backend. This aligns with current trends in using open-source technologies for building accessible and cost-effective business applications (Wang et al., 2017). The system features key operations such as adding, editing, deleting, and viewing inventory items, implemented with a modular design that promotes code readability and maintainability (Sommerville, 2011). Itxperts contribution stands out as a practical guide for beginners and small businesses, offering straightforward solution while also laying a foundation for more advanced development.

INVENTORY MANAGEMENT SYSTEM

A Python-based Inventory Management System is a software solution designed to help businesses manage their stock and inventory operations efficiently. Built this system using the Python programming language, which allows users to add, update, delete, and track inventory items in real-time.
By integrating it with a database (like SQLite or MySQL), the system securely stores the data and let users access or modify it whenever needed.
This system is especially useful for small to medium-sized business to reduce manual errors and improve inventory control.

IMPORTANCE

1. **Optimizing Stock Levels**

The system helps maintain accurate records of inventory in real time, preventing both overstocking and stockouts. By analyzing stock movement and demand patterns, it ensures that the right quantity of products is available at the right time, reducing storage costs and improving inventory turnover.

2. **Reducing Cost and Time**

Automation of inventory tasks like stock updates, stockout and expiry alerts, and graph generation reduces the time spent on manual work. This efficiency lowers labour costs, minimizes errors, and speeds up operations, leading to cost savings and better resource utilization.

3. **Enhancing Customer Satisfaction**

By ensuring product availability and quick order processing, the system helps businesses meet customer demands promptly. This leads to improved service reliability, faster delivery times, and a better overall customer experience.

4. **Supporting Decision Making**

The system provides detailed information and data analytics on stock levels and sales trends. These insights help business owners and managers make informed decisions about restocking, discontinuing slow-moving items, and planning future inventory needs.

TECHNOLOGY USED

The development of the Python-based Inventory Management System utilized lightweight, open-source technologies that are well-suited for building efficient desktop applications. The key technologies used in this project include:

1. Python

   Python was the primary programming language used for developing the system. It is a high-level, interpreted language known for its simplicity, readability, and extensive standard library. Python's syntax makes it easy to write and maintain code, which is particularly beneficial for rapid application development. Its wide range of libraries and community support made it ideal for handling the logic, file operations, and data manipulation required in the inventory system.

2. SQLite

   SQLite was used as the database management system for storing and retrieving inventory data. It is a lightweight, serverless, and self-contained relational database engine that is integrated into Python. SQLite is highly suitable for desktop applications and small-scale systems because it requires no separate server process and offers fast read/write performance.

3. Tkinter

   Tkinter was used to build the graphical user interface (GUI) of the application. It is Python's standard GUI library and provides tools to create windows, buttons, forms, and other interactive components. Tkinter allowed for the development of a clean and user-friendly desktop interface, making the system accessible to users with minimal technical experience. The use of Tkinter ensured that the application remained lightweight and cross-platform.

4. Pycharm

   PyCharm was used as the main IDE for developing the Python-based Inventory Management System. It provided features like code completion, error detection, and easy debugging, which helped streamline development. Its support for virtual environments and database integration made project management more efficient.

SYSTEM FEATURES

The Python-based Inventory Management System was designed to provide essential functionalities for managing inventory efficiently. Below are the key features of the system:

1. **Product Management (CRUD Operation)**

The system allows users to perform basic CRUD operations (Create, Read, Update, Delete) on inventory products. Users can add new products, update existing product details (such as name, price, or quantity), delete obsolete products, and view the full product list. This flexibility ensures that the inventory remains accurate and up-to-date.

2. **Employee Record Management (CRUD Operations)**

The system allows users to perform basic CRUD operations (Create, Read, Update, Delete) on employee details. Users can add new employee, update existing employee details (such as name, ID, or age), delete details, and view the full employee list. This flexibility ensures that the inventory remains accurate and up-to-date.

3. **Real-Time Stock Monitoring**

This system provides real-time visibility of stock levels, allowing users to monitor inventory quantities at any given time. This helps to ensure that businesses can manage stock more effectively and avoid situations such as overstocking or stockouts.

4. **Low Stock and Expiry Alerts**

To prevent shortages, the system includes automatic low-stock alerts. When a product's stock level falls below a predefined threshold it notifies the user. This inventory system provides real-time alerts in case of stockouts and product supply chain management.

5. **Graph Generation**

The inventory system includes built-in graphing features that visually display product trends over time. These graphs help track stock levels, monitor usage patterns, and identify fast- or slow-moving items, enabling better inventory planning and decision-making.

6. **Simple and User-Friendly Interface**

The system features a user-friendly interface built with Tkinter. The GUI is designed to be simple and intuitive, allowing even non-technical easily navigate through the various sections such as product and employee management, and graph generation.

SKILLS GAINED

The development of the Python-based Inventory Management System during the internship was a valuable and rewarding experience. It resulted in a functional software application tailored to the needs of a small business and also offered significant technical and professional growth.
The key outcomes and learnings from the project are outlined below:

1. Practical Application of Python Skills

The project provided an opportunity to apply core Python programming skills in a real-world context. Concepts such as data structures, file handling, modular programming, and exception handling were used extensively throughout the development process.

2. Experience with GUI Development (Tkinter)

Working with Tkinter to design the graphical user interface helped in understanding the basics of desktop application development. Building windows, handling user input, and organizing layout elements improved my ability to design user-friendly interfaces.

3. Database Management with SQLite

Integrating SQLite into the application deepened my understanding of database management, SQL queries, and how to interact with databases using Python. It also provided hands-on experience in designing database schemas, managing data relationships, and ensuring data integrity.

4. Improved Debugging and Testing Skills

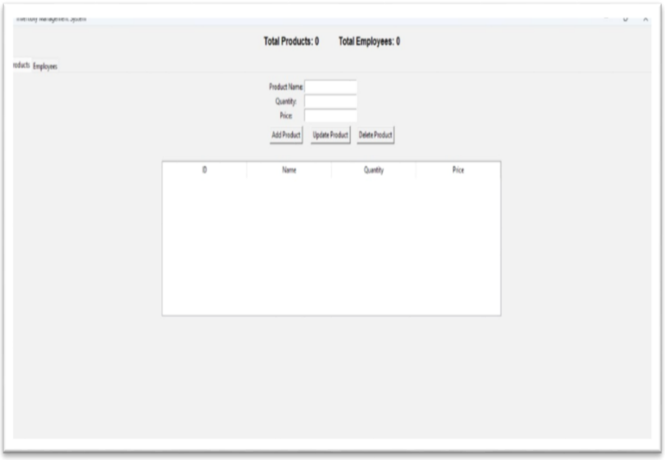
Developing the system required thorough testing and debugging to ensure functionality and reliability. I learned how to identify bugs, interpret error messages, and test individual components systematically.

5. Documentation and Reporting

Writing technical documentation, such as this report, helped me improve my ability to explain software features and technical decisions clearly. It also reinforced the importance of maintaining proper records for future maintenance and user guidance.

RESULTS AND SCREENSHOTS

1. DASHBOARD

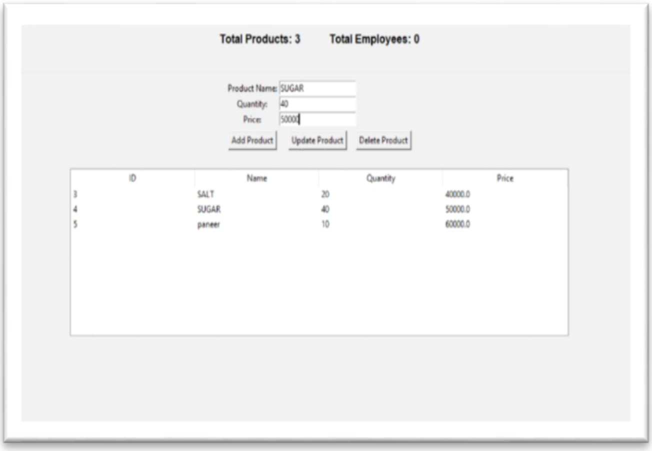


Total Products: 0 Total Employees: 0

Product Name:
Quantity:
Price:

ID	Name	Quantity	Price

2. ADD PRODUCTS

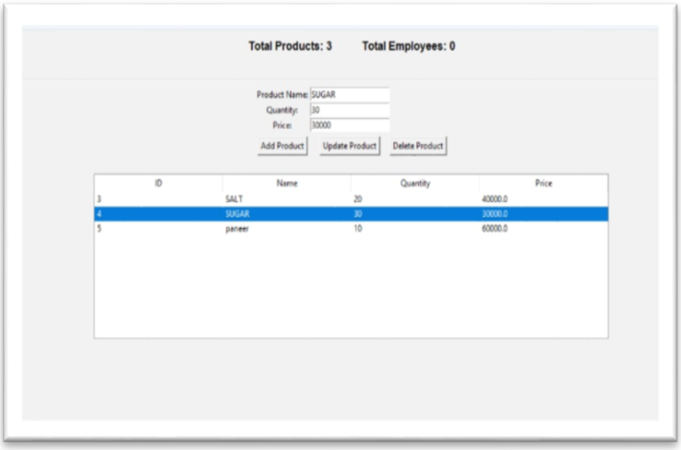


Total Products: 3 Total Employees: 0

Product Name: SUGAR
Quantity: 40
Price: 50000

ID	Name	Quantity	Price
3	SALT	20	40000.0
4	SUGAR	40	50000.0
5	paneer	10	60000.0

3. UPDATE PRODUCT DETAILS

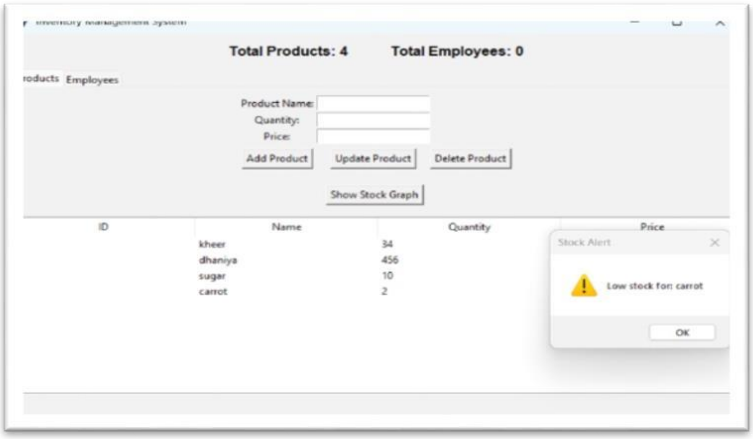


Total Products: 3 Total Employees: 0

Product Name: SUGAR
Quantity: 30
Price: 30000

ID	Name	Quantity	Price
3	SALT	20	40000.0
4	SUGAR	30	30000.0
5	paneer	10	60000.0

4. LOW STOCK ALERT

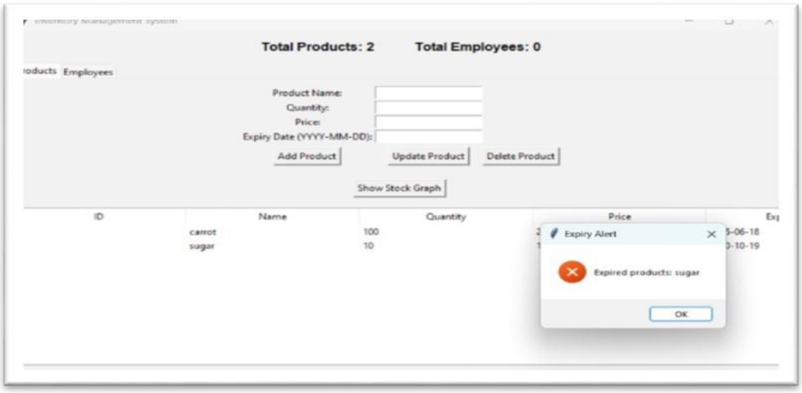


Total Products: 4 **Total Employees: 0**

ID	Name	Quantity	Price
	kheer	34	
	dhaniya	456	
	sugar	10	
	carrot	2	

Low stock for: carrot

5. EXPIRY ALERT

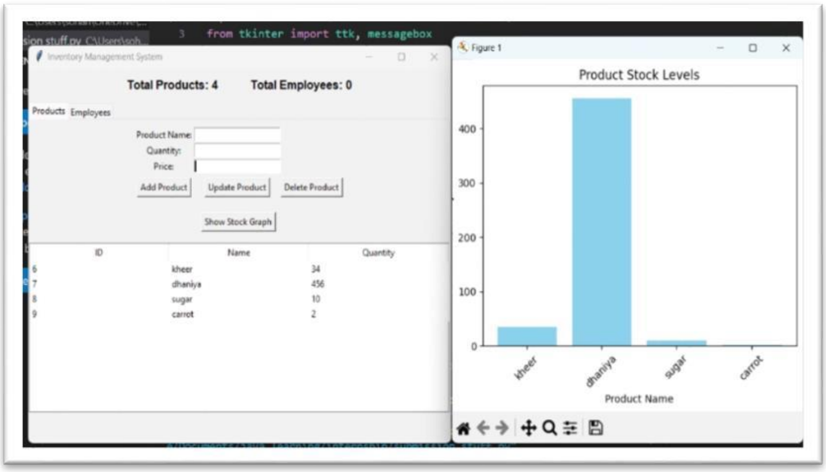


Total Products: 2 **Total Employees: 0**

ID	Name	Quantity	Price	Ex
	carrot	100	2	5-06-18
	sugar	10	1	3-10-19

Expired products: sugar

6. GRAPH GENERATION

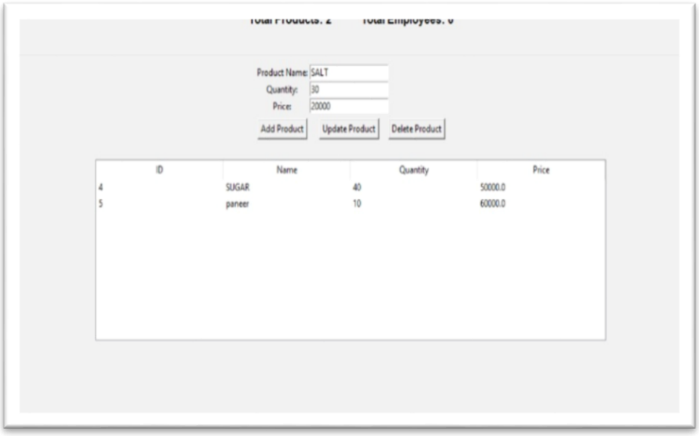


```
from tkinter import ttk, messagebox
```

Total Products: 4 **Total Employees: 0**

ID	Name	Quantity
6	kheer	34
7	dhaniya	456
8	sugar	10
9	carrot	2

7. DELETE PRODUCT

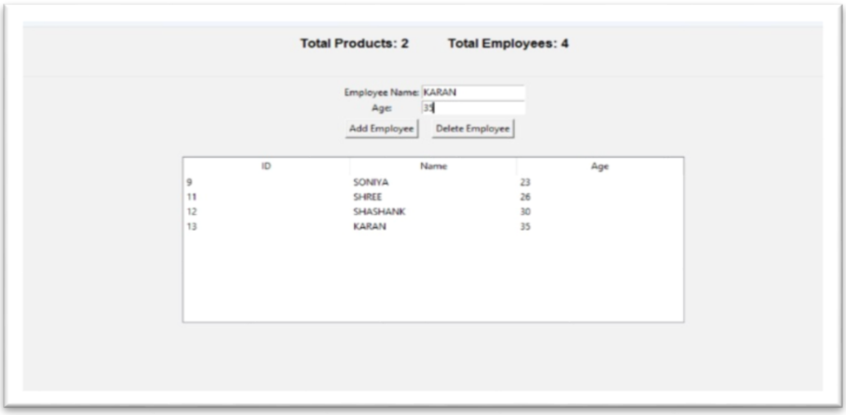


Total Products: 2 Total Employees: 0

Product Name: SALT
Quantity: 30
Price: 20000

ID	Name	Quantity	Price
4	SUGAR	40	50000.0
5	paneer	10	60000.0

8. ADD EMPLOYEE

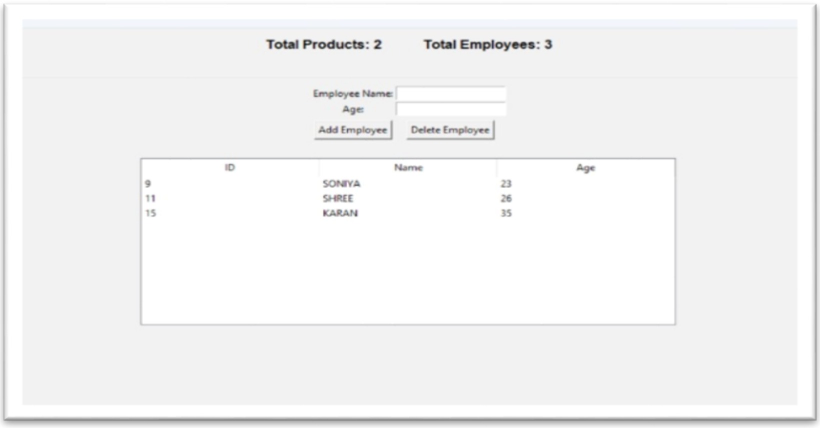


Total Products: 2 Total Employees: 4

Employee Name: KARAN
Age: 35

ID	Name	Age
9	SONIYA	23
11	SHREE	26
12	SHASHANK	30
13	KARAN	35

9. DELETE EMPLOYEE



Total Products: 2 Total Employees: 3

Employee Name:
Age:

ID	Name	Age
9	SONIYA	23
11	SHREE	26
15	KARAN	35

FUTURE IMPROVEMENTS

The inventory management system developed for small-scale industries has significant potential for future enhancements and scalability. Below are key areas for future development:

1. Incorporating predictive analytics to forecast inventory needs based on historical data and market trends, reducing stock outs and overstocking using AI and Machine Learning

2. Using IoT devices and RFID technology for real-time tracking of inventory items, enabling instant identification and location tracking.

3. Transitioning to cloud-based systems to enable remote access, scalability, and real time collaboration across multiple locations, reducing dependency on local hardware.

4. Leveraging blockchain technology to enhance supply chain transparency, traceability, and security, ensuring tamper-proof records of inventory transactions.

5. Integrating robotics and automated systems for efficient warehouse operations, including automated picking, packing, and replenishment tasks.

6. Developing interactive dashboards with advanced analytics capabilities to provide actionable insights into inventory trends and operational efficiency.

7. Creating a mobile version of the system for on-the-go inventory management, enabling users to perform tasks like stock updates and transaction logging via smartphones or tablets.

8. Implementing features like batch tracking for expiry dates and waste reduction strategies to minimize inventory wastage in industries dealing with perishable goods.

CONCLUSION

The development of the Python-based Inventory Management System was a successful and insightful project that provided both a functional solution for managing inventory and a strong learning experience. The system effectively addressed the common challenges associated with manual inventory tracking by offering features such as product management, real-time stock updates, low-stock alerts, and graph generation.

Furthermore, by utilizing Python, along with SQLite for database management and Tkinter for the graphical user interface, the application proved to be a lightweight, user-friendly, and easily maintainable tool suitable for small to medium-sized businesses. It streamlined daily operations and minimized the possibility of human error, ultimately supporting more accurate and efficient inventory control.

During the internship, I gained valuable technical skills in software development, database integration, and GUI design, as well as practical experience in troubleshooting, debugging, and documenting a real-world project. This hands-on exposure not only strengthened my programming abilities but also improved my understanding of how software can be tailored to solve specific business problems.

Overall, the project demonstrated the potential of open-source tools in developing cost-effective solutions and highlighted the importance of aligning technical design with user needs and organizational goals.

REFERENCES

1. Ashley Gingeleski, *How to build an Inventory App with Tkinter,* Google , https://ashleygingeleski.com/2021/01/30/how-to-build-aninventory-app-with-tkinter/

2. Gemini.google.com, https://gemini.google.com/?hl=en-IN

3. Openai.com ,https://openai.com/index/chatgpt/

4. Wikipedia.com ,https://www.wikipedia.org/

5. Rangesh-*Webcode, How to create Inventory Management System with database in Python,*Youtube https://www.youtube.com/watch?v=uxLuAz7b1tU&list=PL4P8s Y6zvjk6ef4lpm6XiwJVRahLCp6DI

6. Yuvraj Karuppan, Murugan and Pravin M.C, *Design and development of an applicaton for Database Inventory Management system using Tkinter and SQLite Platform,* researchgate.net, https://www.researchgate.net/publication/347805620_Design _And_Development_Of_An_Application_For_Database_Mainte nance_In_Inventory_Management_System_Using_Tkinter_And _Sqlite_Platform

7. Pythonmania.org, *Simple inventory management project using python and Tkinter,* Source –chatgpt.com, https://pythonmania.org/simple-inventory-managementproject-using-python-and-tkinter/?utm_source=chatgpt.com

8. Nishanth,Github.com ,*Inventory Managemnet system,*Source- chatgpt, https://github.com/nishant0820/Inventory-Management-System?utm_source=chatgpt.com

9. Itxperts.co,*Inventory Management System* ,Source- chatgpt, https://itxperts.co.in/blog/inventory-management-systemusing-python/?utm_source=chatgpt.com