

“Inventory Management System using Python”

A PROJECT

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

In

INTERNET OF THINGS

By

**Devansh Rawat(0901EO211018)
Harshit Dwivedi(0901EO211026)**

Under the guidance of

Prof. Saurabh Kumar Rajput

Assistant Professor

Centre for IoT



**CENTRE FOR INTERNET OF THINGS
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE,
GWALIOR (M.P.) – 474005**

July - Dec 2023



Madhav Institute of Technology & Science, Gwalior (M.P.)

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CANDIDATE'S DECLARATION

We hereby declare that the work presented in this project entitled **Inventory Management System using Python which** is being submitted in the partial fulfilment of the requirement for the award of degree of Bachelor of Centre for Internet of Things is an authentic record of our own work carried out under the guidance of **Prof. Saurabh Kumar Rajput, Assistant Professor**, Centre for Internet of Things.

The matter presented in this project has not been submitted elsewhere by us for the award of any other degree/diploma.

Harshit Dwivedi

Devansh Rawat

Date:

Place: Gwalior

This is to certify that the above statement made by the candidates is correct to the best of my knowledge and belief.

Guided by

Prof. Saurabh Kumar Rajput
Assistant Professor
Centre for Internet of Things
MITS, Gwalior

Minor Project Coordinator

Forwarded by

Dr Yashwant Sawle
Assistant Professor,
Centre for Internet of Things
MITS, Gwalior

Dr. Praveen Bansal
Assistant Professor &
Coordinator of Centre for Internet
of Things, MITS, Gwalior

ACKNOWLEDGEMENTS

Madhav Institute of Technology & Science, Gwalior, who has been supportive of our project and career goals and who worked actively to provide us with valuable academic time to pursue these goals. We would also like to extend our regards to **Prof. Saurabh Kumar Rajput** who has supported us throughout this project with his knowledge and guidance. We are grateful to all of those with whom we have had the pleasure to work with during this project. Each of the members of the Centre for IoT provided us with extensive personal and professional guidance and taught us a huge deal about both scientific research and life in general. This work would not have happened without the academic support of the Madhav Institute of Technology & Science, Gwalior.

Executive Summary

The Python Inventory Management System Project represents a state-of-the-art solution aimed at transforming inventory control for businesses. Equipped with a user-friendly interface and real-time tracking features, the system ensures precise stock level management, minimizing errors and automating essential tasks. Through the use of advanced Python programming, the project provides functionalities such as automated data entry, order management, comprehensive reporting, and analytics. Its integration capabilities enable seamless connectivity with other business systems, ultimately enhancing overall efficiency. By optimizing inventory management, the system not only reduces operational costs but also equips businesses with timely and accurate data for informed decision-making.

.

ABBREVIATIONS

IoT	Internet of Things
IMS	Inventory Management System
IS	Inventory System
DB	Data Base
Py	Python
SQL	Structure Query Language

Table of Contents

ACKNOWLEDGEMENTS.....	3
Executive Summary	4
ABBREVIATIONS	5
LIST OF FIGURES	7
CHAPTER - 1	8
INTRODUCTION.....	8
1.1 Overview	8
1.2 What is Inventory Management System Project In Python?.....	8
1.3 Project Details.....	9
CHAPTER - 2 LITERATURE REVIEW.....	10
CHAPTER – 3	Error! Bookmark not defined.
METHODOLOGY.....	Error! Bookmark not defined.
3.1 Functions	Error! Bookmark not defined.
3.2 Use of VS CODE.....	Error! Bookmark not defined.
3.3 Hardware Used	Error! Bookmark not defined.
3.4 Installing mysql	Error! Bookmark not defined.
3.5 python programming	Error! Bookmark not defined.
CHAPTER – 4	20
RESULTS & DISCUSSIONS.....	20
CHAPTER – 5	21
CONCLUSION.....	21
REFERENCES	22

LIST OF FIGURES

S NO.	FIGURE NAME	PAGE NO.
1.	MySQL home page	12
2.	Inventory System database	13
3.	Inventory-table with columns	13
4.	for inventory table	14
5.	for transactions table	14
6.	Importing the libraries	15
7.	Output1 of Inventory Management System Project Python	15
8.	Output2 of Inventory Management System Project Python	17
9.	Output3 of Inventory Management System Project Python	17
10.	Output4 of Inventory Management System Project Python	18
11.	Output5 of Inventory Management System Project Python	19
12.	Output6 of Inventory Management System Project Python	19

CHAPTER - 1

INTRODUCTION

1.1 Overview

The Inventory Management System (IMS) Project in Python represents a strategic and user-centered solution aimed at transforming inventory control for businesses. Emphasizing simplicity and effectiveness, the project offers an intuitive interface, real-time tracking, and automated data entry and order management processes. By harnessing Python programming, the system delivers valuable insights to businesses through robust reporting and analytics. With rigorous security measures and integration capabilities, it ensures data integrity and seamless connectivity with other business systems. Key benefits of the project include improved operational efficiency, cost reduction, data-driven decision-making, and heightened customer satisfaction.

The Python Inventory Management System Project is a crucial tool in today's business landscape, providing a comprehensive approach to inventory management. Its real-time tracking feature allows businesses to anticipate stock fluctuations, thereby preventing stockouts and overstock situations. The project's focus on automated data entry streamlines operations, reducing the risk of errors and saving valuable employee time. Moreover, the order management system ensures seamless and timely order processing, significantly enhancing overall customer satisfaction. The robust reporting and analytics capabilities empower decision-makers with actionable insights, enabling the implementation of data-driven strategies to enhance performance and profitability.

Furthermore, the project's adaptability stands as a key strength. Its integration capabilities enable businesses to seamlessly link the IMS with existing software infrastructure, fostering a cohesive ecosystem. This interoperability not only boosts efficiency but also guarantees that the IMS aligns with the unique needs of various industries. Whether a small startup or a large enterprise, businesses can utilize the Inventory Management System Project to optimize their supply chain, decrease operational costs, and promote a more agile and responsive approach to inventory control. In today's digital era where precision and agility are essential, the IMS Project in Python emerges as a valuable asset for businesses maneuvering the complexities of modern inventory management.

1.2 What is Inventory Management System Project In Python?

In this project, we will explore the fundamentals of inventory management, encompassing the identification of various types of goods and determining optimal reorder points. This system diligently monitors the entire lifecycle of inventory, from procurement to distribution. By employing this inventory management approach, businesses can ensure the availability of sufficient stock to meet customer needs, thereby achieving cost savings and bolstering customer satisfaction. Therefore, an effectively managed supply chain is vital for the success of any enterprise.

The project titled **Python Inventory Management System with Source Code** utilizes the **Tkinter** module and MySQL server for programming, creating a GUI-based application.

1.3 Project Details

- **Project Name:** Inventory Management System Project In Python
- **Abstract:** This project is a Graphical User Interface (GUI) application developed using the Tkinter module. Its primary aim is to streamline the various components involved in inventory management.
- **Language/s Used:** Python
- **IDE:** VS CODE(Recommended)
- **Python Version (Recommended):** 3.8 or 3.9
- **Database:** MySQL
- **Type:** Desktop Application

CHAPTER - 2

LITERATURE REVIEW

The research and development of Inventory Management Systems (IMS) have been the focus of significant attention in the field of business operations. Numerous studies have stressed the critical role of efficient inventory management in boosting overall organizational performance and competitiveness. The adoption of Python programming in inventory management systems has become increasingly prominent due to its adaptability and seamless integration. Academics have emphasized the significance of real-time tracking capabilities, a key feature embedded in the IMS Project in Python, as an essential component for addressing stock-related challenges.

Automation plays a pivotal role in enhancing inventory management systems, with a particular focus on data entry processes. Research indicates that by minimizing manual data entry, data accuracy is improved while also freeing up time for employees to concentrate on more strategic responsibilities. The IMS Project underscores this research finding by integrating automated data entry functionalities, effectively addressing a prevalent challenge in inventory management.

Order management, an essential component of inventory control, has been a focal point in the literature due to its influence on customer satisfaction. Effective order processing is acknowledged as a driver for enhancing customer experiences. The IMS Project in Python reflects these conclusions by optimizing order management procedures, guaranteeing prompt order fulfillment, and consequently enhancing customer satisfaction.

Furthermore, scholarly literature emphasizes the importance of data-driven decision-making in modern business environments, particularly in the context of business intelligence and analytics. The robust reporting and analytics capabilities of the IMS Project provide decision-makers with valuable insights, aligning with broader trends identified in academic works. This literature review reaffirms the relevance and timeliness of the IMS Project in Python, establishing it as a leading contemporary inventory management system that incorporates insights and best practices from existing research.

CHAPTER – 3

Methodology

3.1 Functions of Inventory management system in python

1. Overseeing all products
2. Supervising sales and inventory management system
3. Creating invoices

3.2 Use of VS CODE IDE for the project

1. Begin by installing the VS CODE.
2. Next, create a new project by navigating to **File** and selecting **New Project**. Input the project name and then click on **Create**.
3. After creating the project, right-click on the project name and proceed to create three new Python files named "inventory.py," "Inserttodatabase.py," and "Updatedatabase.py".
4. Once the files are created, proceed to write the necessary code for the inventory management system and execute the Python program by clicking on the **Run** tab.

3.3 Installing MySQL Server Database and MySQL-connector-python

1. Please download [mysql-installer community 8.0.29.0](#) and proceed with the installation of the MySQL Database.
2. Prior to importing the mysql.connector, it is necessary to install the connector by executing "pip3 install mysql-connector-python" in the terminal or command prompt to enable the database connectivity.

3.4 Creating the database and tables

To establish a database in MySQL Workbench, please follow Step 1:

1. Access MySQL Workbench.
2. Select the Local Instance MySQL80.
3. Input the password when prompted.

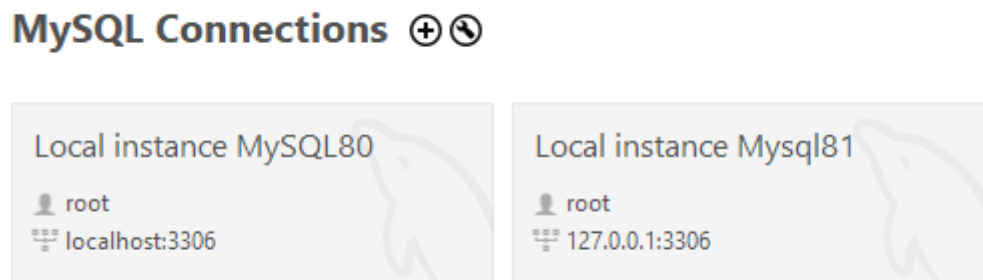


Fig. 1 MySQL home page

Establish a database, specifically a Schema, by performing a right-click action within the schemas section and proceed to select "create schema" to set up the "inventory_system."

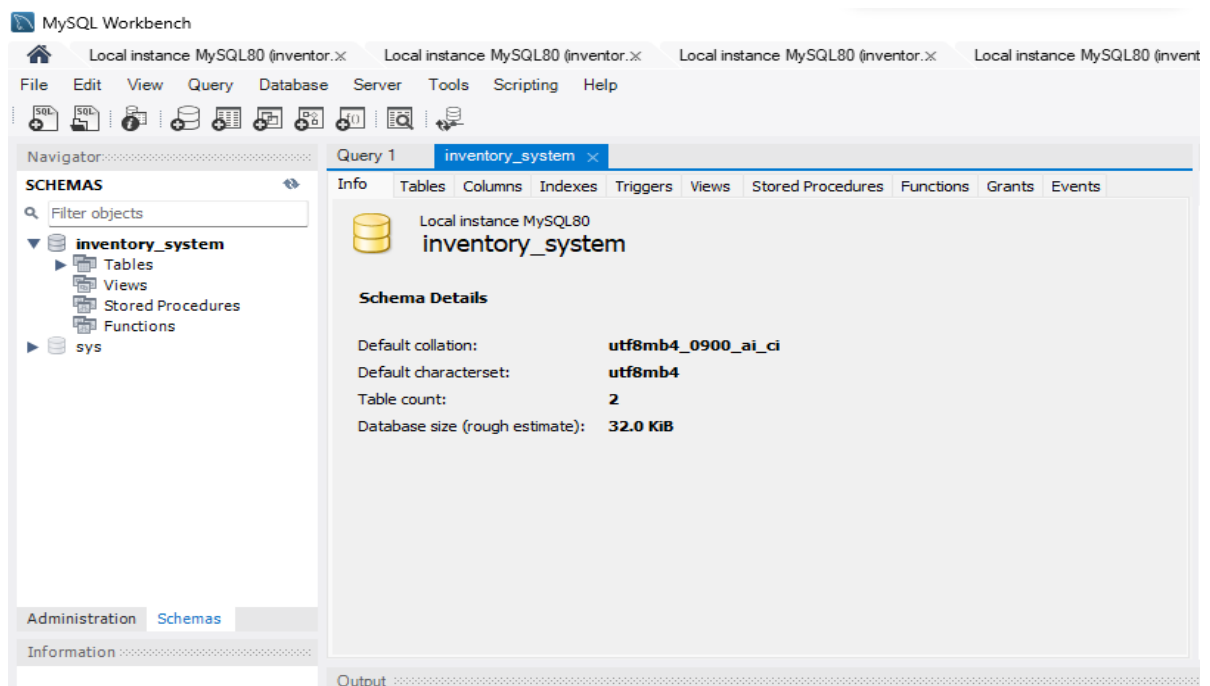


Fig.2 Inventory System database

Generate the inventory and transaction tables by right-clicking on the Tables within the inventory_system schema. Proceed to define the columns for both the inventory and transaction as indicated below:

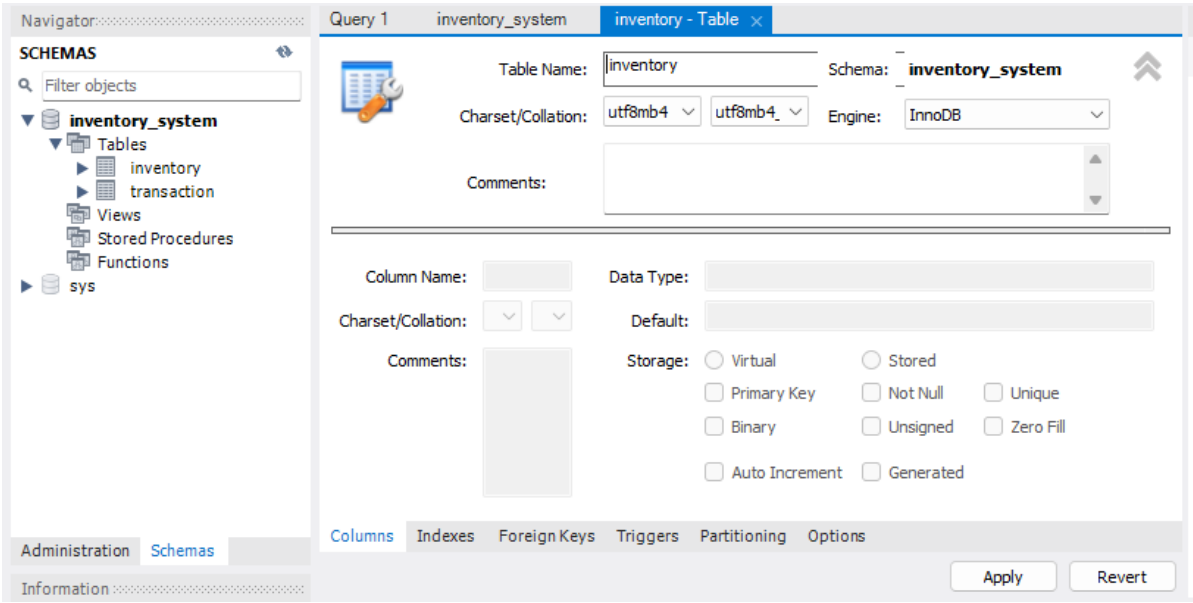


Fig.3 inventory-table with columns

Define the columns for the inventory and transaction tables as indicated below

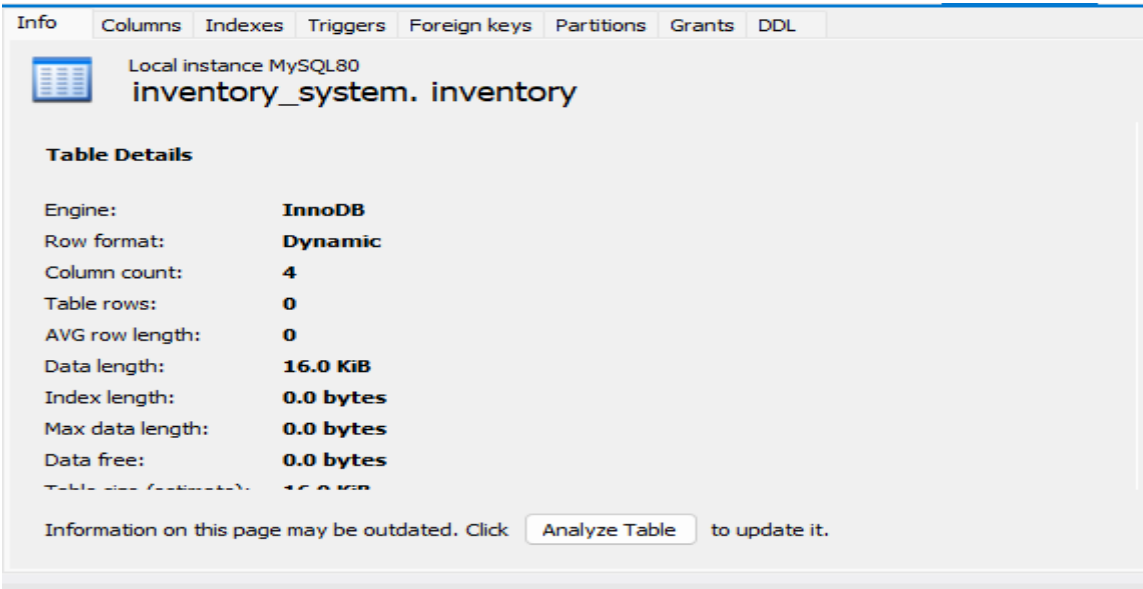


Fig.4 for inventory table

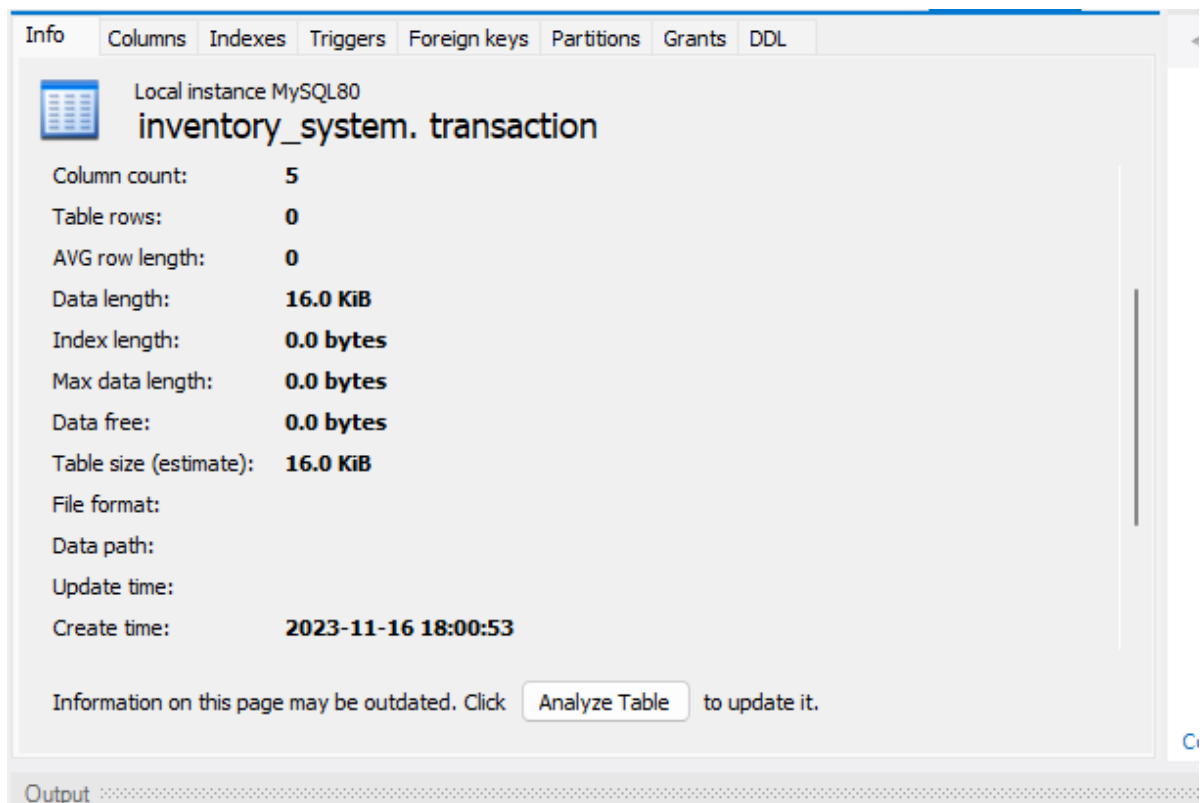


Fig.5 For Transaction Table

3.5 Python Program explanation of Inventory Management system with and Output

Create inventory.py file

```
1 from tkinter import *
2 from tkinter import messagebox
3 import mysql.connector
4 import tkinter.messagebox
5 import datetime
6
```

Fig.6 Importing the libraries

EXPLANATION:

- Utilize [Tkinter](#) for creating the GUI.
- Implement **mysql.connector** to establish database connections with the MySQL server.

- Use [Datetime.date](#) to convert the date from the tree to a [Datetime.date](#) instance for setting purposes.
- Incorporate **Tkinter.messagebox** to display a box with information or errors.
- Employ the **math** module for performing calculations.

Create the main form of the inventory management system

Within the Application class, we are tasked with generating the primary form that accommodates two frames - a left frame and a right frame. This involves placing labels, buttons, and text fields into frames of specified dimensions.



Fig.7 Output1 of Inventory Management System Project Python

The function of database connection and displaying the products information

In the **def ajax** code, the database connection to "inventory_system" is established using the connect function. Upon providing the ID in the Product's ID field, the SQL query is executed to retrieve information from the inventory table. Once the ID is found, the product's name and price are displayed on the form. Subsequently, quantity and discount offered labels and text fields are created, along with the display bills and generate bill buttons.

Function for add to cart and calculate change for inventory management system

```
def add_to_cart(self, quantity_value):
    """
```

The method `add_to_cart` takes the value for the quantity and checks whether the quantity provided matches the stock available.

It uses the if-else function to verify this condition and displays an error message if the quantity is higher than the available stock.

If the quantity is within the available stock, it takes the quantity and applies any potential discounts.

After clicking on 'add to cart', the product's name and quantity are displayed in the cart and the total amount is calculated and shown below the amounts field.

```
""""  
def calculate_change(self):
```

```
""""  
  
The calculate_change method is used to determine the change in the total amount.
```

Upon entering the amount and clicking on 'generate a bill', the changed amount is deducted from the total amount and displayed on the screen.

```
""""  
  
The generate_bill() function validates the ID and retrieves the corresponding information. It verifies the stock availability, deducts the used stock, generates new stock, updates the inventory table, and sets the new stock quantity for the specified ID. The system then records this data in the transaction table.
```



Fig.8 Output2 of Inventory Management System Project Python

Create inerrtdatabase.py

Creating the form for adding the data in the database of the inventory table

Within this class, we establish the structure of the database table "inventory," specifying the labels, entry fields, buttons, text size and colour, as well as the positioning of each field within the form.

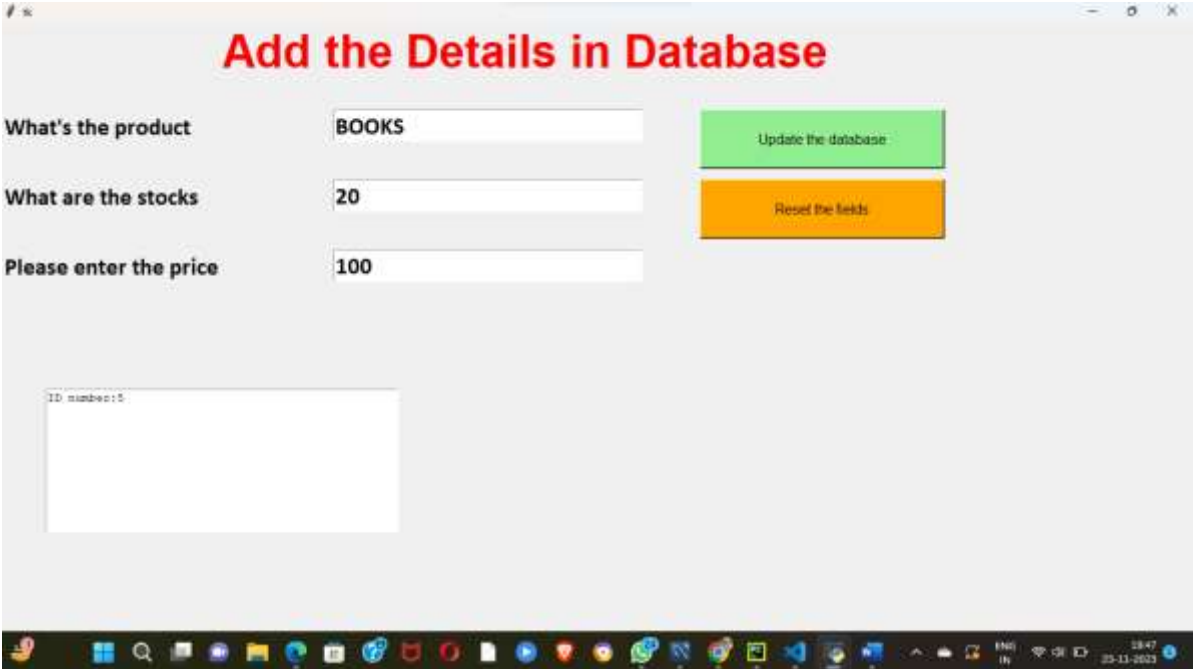


Fig.9 Output3 of Inventory Management System Project Python

Function for accepting the data and clearing the text fields

In the function `**def get_items(),**` the variables for name, stock, and price are initialized using the `**get() function**` to store the entered values. The if block checks if any of these fields are empty; if so, it displays the error message "Error: Please fill all the entries."

Otherwise, it executes the SQL query to insert the values into the inventory table and prints the message "Success: Successfully added to the database."

The `**def clear_all()**` function auto-increments the ID and clears all the text fields to facilitate adding another set of data to the table.



Fig.10 Output4 of Inventory Management System Project Python

Create a Updatedatabase.py file

Create a form of updating the inventory management system.

In this instance, the max(id) function exhibits the ID number on the display.

Database Class

In this section, we establish the structure of the form that encompasses all labels, entry fields, buttons, text size and color, and the positioning of each field within the form to facilitate updates on the table.



Fig.11 Output5 of Inventory Management System Project Python

Function to search and update the table

`def search_id():`

Upon entering the ID and clicking the search button, all the associated information will be displayed in the text fields.

`def update_changes():`

This function will apply the modifications made in the fields to update the inventory table.



Fig. 12 Output6 of Inventory Management System Project Python

CHAPTER – 4

RESULTS & DISCUSSIONS

The Python-based implementation of the Inventory Management System (IMS) Project has delivered substantial outcomes in enhancing inventory control for businesses. Its intuitive interface has simplified adoption across diverse organizational environments, leading to improved operational efficiency. Real-time tracking capabilities have played a crucial role in averting stockouts and overstock situations, empowering businesses with a proactive approach to inventory management. Furthermore, the automated data entry functionality has minimized manual errors, ensuring data precision and freeing up valuable employee time for more strategic responsibilities.

The streamlined processes of the order management system have resulted in efficient order processing, positively enhancing customer satisfaction by ensuring timely order fulfillment. Furthermore, the robust reporting and analytics capabilities have provided decision-makers with valuable insights into key performance indicators, enabling the implementation of data-driven strategies to improve operational performance.

The integration capabilities of the IMS Project have significantly improved overall business efficiency by seamlessly connecting with other systems, creating a more cohesive and interconnected operational environment. This has led to tangible benefits for businesses of all sizes and across industries, including increased operational efficiency, cost reduction, and enhanced customer satisfaction. The implementation of the IMS Project in Python serves as a testament to its effectiveness in addressing current challenges in inventory management and contributing to the success and competitiveness of businesses.

CHAPTER – 5

CONCLUSION

In summary, the Python Inventory Management System Project has proven effective in modernizing conventional inventory control methods, leading to streamlined, efficient, and data-driven processes. Its user-friendly interface, real-time tracking, and focus on automation have enhanced inventory management for businesses, ultimately improving operational efficiency and reducing costs. Additionally, the integration capabilities and comprehensive reporting features have equipped decision-makers with valuable insights, promoting a more adaptable and responsive inventory control approach.

The project's significance lies in its tangible outcomes, which include improved customer satisfaction and more informed strategic decision-making. As businesses navigate the complexities of modern inventory management, the IMS Project in Python emerges as a valuable asset, enhancing competitiveness and fostering sustainable growth. Its adaptability and alignment with contemporary research insights position it as a pivotal tool for businesses aiming to excel in an era where precision, efficiency, and data-driven strategies are paramount.

REFERENCES

- [1] [Inventory Management System Project In Python - CopyAssignment](#)
- [2]https://www.irjmets.com/uploadedfiles/paper//issue_6_june_2023/42137/final/fin_irjmets1686838032.pdf
- [3] <https://www.scribd.com/document/444986497/Online-Stock-Inventory-Management-System-Project-in-Python>
- [4]https://www.researchgate.net/publication/311216054_Inventory_Management_System_U sing_IOT
- [5]https://www.researchgate.net/publication/372653862_Inventory_Management_Systems_I MS