

# **PYTHON PROJECT FINAL REPORT**

**Course Code: CSM216**



**TOPIC :** Sales and Inventory Management System

**SUBMITTED BY**

**NAME:** Manishka Sivakumar

**REGISTRATION NUMBER :**12326562

**SECTION:** K23CH

**ROLL NUMBER :**70

**SUBMITTED TO:** Mr.Aman Kumar

## Acknowledgment

I would like to express my deepest gratitude to everyone who has contributed to the successful completion of this project, "**Sales and Inventory System Using Python**". This project has been a significant learning experience, and its successful completion would not have been possible without the guidance, support, and encouragement I have received throughout the process.

First and foremost, I am immensely grateful to my instructor, **[Instructor's Name]**, for their constant guidance, insightful feedback, and expertise in helping me shape this project. Their deep understanding of the subject matter and practical approach toward problem-solving have been invaluable in addressing the challenges faced during the development of this system.

I extend my heartfelt thanks to my peers and classmates for their constructive inputs, collaborative brainstorming sessions, and moral support. Working alongside them has enriched my knowledge and helped me understand various perspectives and solutions to complex problems.

I would also like to acknowledge the role of online resources and communities such as **Stack Overflow**, **Python Documentation**, and various other educational platforms that provided technical support and detailed insights into coding best practices and troubleshooting. The openly available resources from these platforms have been instrumental in overcoming technical roadblocks.

Additionally, I owe my gratitude to my family and friends, who have always encouraged me to strive for excellence and supported me throughout this journey. Their patience and understanding gave me the mental space and motivation to focus on this project.

Finally, I am thankful to **Lovely Professional University** for providing a conducive learning environment and access to resources that enabled me to explore and implement my ideas effectively. The knowledge and skills I have acquired during this project will undoubtedly serve as a strong foundation for my future endeavors.

With sincere appreciation, I dedicate this report to everyone who has played a part in its successful completion.

[Manishka Sivakumar]

## Table of Contents

<b>Section</b>	<b>Page Number</b>
<b>Title Page</b>	1
<b>Acknowledgment</b>	2
<b>Table of Contents</b>	3
<b>Introduction</b>	4
<b>Objectives and Scope of the Project</b>	5
<b>Application Tools</b>	7
<b>Project Design</b>	8
<b>Flowchart</b>	9
<b>Project Implementation</b>	10
<b>Conclusion</b>	17

## Introduction

In the modern business landscape, managing sales and inventory effectively is critical to ensuring operational efficiency and profitability. Organizations, whether small or large, depend on reliable systems to monitor stock levels, track sales trends, and optimize resource allocation. Manual systems, which are prone to errors, inefficiencies, and inaccuracies, are no longer viable in a world that demands real-time data and rapid decision-making. This project focuses on the development of a **Sales and Inventory System** using Python, a programming language known for its versatility and robustness.

The primary purpose of this project is to streamline sales and inventory management by leveraging the power of automation and data handling. The system is designed to cater to businesses aiming to minimize manual intervention, reduce operational costs, and maximize efficiency. By integrating core functionalities such as stock tracking, sales reporting, and predictive analytics, the system serves as a comprehensive solution for managing day-to-day business operations.

## Significance of the Project

An effective sales and inventory system brings numerous benefits to businesses:

1. **Error Reduction:** Manual processes often result in data entry errors or misplaced records, leading to financial losses. Automation eliminates these issues, ensuring accurate data management.
2. **Real-Time Data:** With the ability to track sales and inventory levels in real-time, businesses can make informed decisions, such as restocking popular items or discontinuing slow-moving products.
3. **Operational Efficiency:** Automating routine tasks allows staff to focus on more strategic initiatives, enhancing productivity and reducing operational bottlenecks.
4. **Customer Satisfaction:** By maintaining optimal stock levels, businesses can fulfill customer orders promptly, leading to improved customer retention and satisfaction.

## Problem Statement

Managing sales and inventory manually is a daunting task, especially for businesses with diverse product lines or high transaction volumes. Without a systematic approach, businesses often face:

- Overstocking or understocking of items, leading to increased costs or lost sales opportunities.
- Delays in order fulfillment due to inaccurate inventory records.
- Lack of insight into sales trends, making it difficult to adapt to market demands.

These challenges underscore the need for a robust, user-friendly system that not only automates processes but also provides valuable insights to support decision-making.

## Project Objectives

The **Sales and Inventory System using Python** is developed to address the complexities and inefficiencies faced by businesses in managing their sales and inventory. The objectives of the project are outlined below:

1. **Automation of Inventory Management:**  
To automate the process of recording stock levels, replenishments, and updates after sales transactions, ensuring accuracy and efficiency.
2. **Real-Time Data Tracking:**  
To provide real-time updates on inventory status and sales transactions, enabling businesses to respond promptly to market demands and stock shortages.
3. **Enhanced Decision-Making:**  
To generate detailed reports on sales trends, inventory turnover, and product performance, facilitating data-driven business strategies.
4. **User-Friendly Interface:**  
To design an intuitive graphical user interface (GUI) that allows users to interact with the system effortlessly, regardless of their technical expertise.
5. **Alert System Integration:**  
To implement automated alerts for low-stock or out-of-stock items, ensuring timely replenishment and preventing lost sales opportunities.
6. **Support for Scalability:**  
To ensure the system is adaptable for businesses of different sizes and capable of handling an expanding product line or increasing transaction volumes.
7. **Cost and Time Efficiency:**  
To reduce operational costs and the time spent on manual processes, allowing businesses to allocate resources to more strategic initiatives.

## Scope of the Project

The scope of the **Sales and Inventory System using Python** defines its applicability and boundaries to ensure alignment with business requirements.

1. **Target Users:**  
The system is designed for small to medium-sized enterprises (SMEs) and retail businesses that require efficient inventory management and sales tracking.
2. **Key Features:**  
**Inventory Tracking:** Real-time tracking of stock levels, including automated updates after each sale.

**Sales Management:** Recording of daily transactions, customer orders, and payment statuses.

**Reporting:** Generation of reports such as sales summaries, inventory turnover rates, and product-specific performance.

**Alerts:** Notifications for stock replenishment based on predefined thresholds.

**Search and Filter Options:** Easy retrieval of product or sales information using search filters.

### 3. Technological Framework:

**Programming Language:** Python, for its simplicity and vast library ecosystem.

**Libraries/Modules:** Use of libraries such as Pandas for data manipulation, Tkinter for GUI design, and Matplotlib for visualization.

**Data Storage:** SQLite or CSV files for efficient and secure data storage.

### 4. Limitations:

- The current version focuses on offline operations without real-time multi-user support.
- Advanced analytics, such as demand forecasting or predictive sales analysis, are not included in the initial implementation but can be added in future updates.

### 5. Future Scope:

- Integration with e-commerce platforms to sync online and offline sales data.
- Implementation of machine learning algorithms for predictive analytics and demand forecasting.
- Cloud-based deployment to support multi-user and remote operations.
- Mobile application support for remote access and management.

## Application Tools

The development of the **Sales and Inventory System using Python** required the use of various tools, software, libraries, and other resources to ensure efficient and robust implementation. Below is a detailed description of the tools and technologies utilized in this project:

### 1. Programming Language

- **Python:**  
The core programming language for the project, chosen for its simplicity, versatility, and extensive library ecosystem. Python's flexibility makes it ideal for tasks ranging from GUI development to data analysis.

### 2. Integrated Development Environments (IDEs)

- **Jupyter Notebook:**  
Utilized for prototyping and testing algorithms. Its interactive environment was particularly useful for working with data analysis and visualization libraries

### 3. Libraries/Packages

The project utilized several Python libraries to enhance functionality and simplify implementation.

- **NumPy:**  
For handling numerical data and performing efficient mathematical computations.
- **Pandas:**  
Used for data manipulation and analysis. This library was essential for organizing sales and inventory data into structured formats such as DataFrames.
- **Matplotlib:**  
To create visual representations such as sales charts, inventory trends, and other graphical insights.
- **Tkinter:**  
For designing the graphical user interface (GUI). Tkinter was chosen for its simplicity and native support within Python.

## Project Design

The **Sales and Inventory System using Python** is structured to ensure modularity, scalability, and efficiency. This section outlines the system's core components, their functionalities, and how they interact to achieve the project's objectives. The design emphasizes a clear separation of concerns, ensuring that each part of the system performs its designated task effectively.

---

### 1. Main Components

The project is divided into the following key components:

#### a. User Interface (UI)

- **Purpose:** Facilitates interaction between the user and the system.
- **Implementation:**
  - Built using **Tkinter** for creating a user-friendly and interactive graphical interface.
  - Includes features such as buttons, input fields, and data tables for adding, viewing, and updating sales and inventory records.
  - Provides error handling and feedback to ensure smooth user interaction.

#### b. Core Logic and Business Rules

- **Purpose:** Ensures the system adheres to the defined objectives and constraints.

- **Implementation:**
  - Python functions are used to calculate stock levels, total sales, and profit margins.
  - Business rules include validating stock availability before processing sales and updating inventory after transactions.

### c. Reporting and Visualization

- **Purpose:** Provides insights through reports and visual representations.
- **Implementation:**
  - Generates sales and inventory reports using **Pandas**.
  - Visualizes trends with **Matplotlib**, such as monthly sales performance and low-stock alerts.
  - Data can be exported to Excel for external analysis.

## 2. Classes and Functions

The system uses object-oriented programming principles for better maintainability and reuse.

### Classes:

- **Product:**
  - Attributes: Product ID, name, price, stock quantity.
  - Methods: Add product, update stock, retrieve product details.
- **Sale:**
  - Attributes: Sale ID, product details, quantity sold, date of sale.
  - Methods: Record sale, calculate revenue, validate stock availability.

### Functions:

- **add\_product():** Adds a new product to the inventory database.
- **update\_inventory():** Updates stock levels after a sale.
- **generate\_report():** Produces detailed inventory and sales reports.
- **alert\_low\_stock():** Notifies when stock levels fall below a defined threshold.

## 3. Workflow and Interaction

The system components interact seamlessly to achieve the project's objectives:

1. **Data Input:**
  - Users input product details or sales transactions through the UI.
  - Data is validated before being stored in the database.



## 2. **Inventory Updates:**

- When a sale is recorded, the inventory is automatically updated to reflect the reduced stock.

## 3. **Reporting:**

- Users can generate and view sales and inventory reports in real-time.
- Visualizations provide insights into sales trends and inventory status.

## 4. **Alerts:**

- The system checks stock levels and sends notifications for items that need restocking.

# Flowchart for Sales and Inventory System

**Workflow Description:** The flowchart visually represents the logical flow of a Python-based Sales and Inventory Management System. Each step ensures the smooth operation of essential components, including inventory management and sales processing.

1. **Start:** The process begins with initializing the application.
2. **User Authentication:** The user logs in with a username and password.
  - If login fails, the user is prompted to retry or exit.
3. **Main Menu:** The user selects from the following options:
  - **Inventory Management:**
    - **View Inventory:** Display a list of all products and their details (e.g., stock levels, prices).
    - **Update Inventory:** Add or remove items or adjust quantities.
    - **Generate Inventory Report:** Summarize current stock levels and low-stock alerts.
  - **Sales Management:**
    - **Process Sales:** Input customer details and items purchased, calculate totals, and generate receipts.
    - **View Sales Reports:** Display and analyze sales data over time.
4. **Confirmation:** Each operation confirms success or failure, prompting the user for further actions.
5. **End:** After completing tasks, the user logs out, and the system safely exits.

## Project Implementation:

### ADMIN ACCOUNT

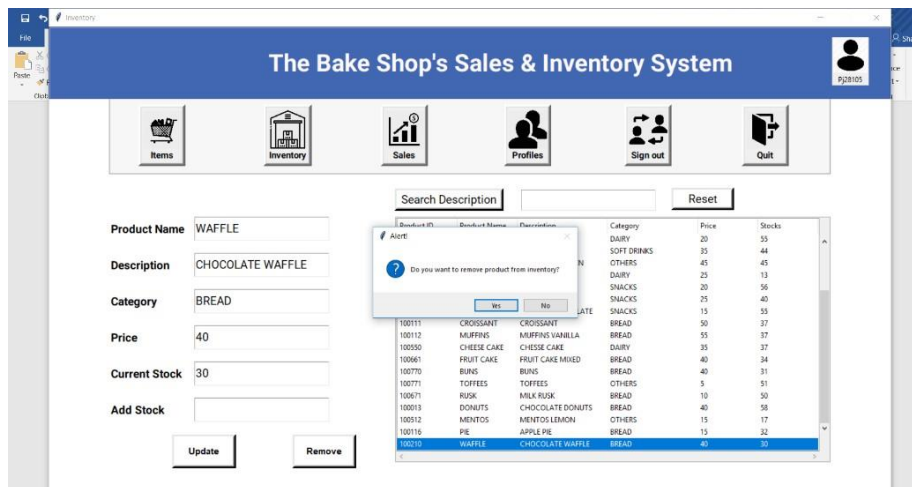
- Inventory:- An inventory window opens up. Through this window you can edit product details or add stock to the product or remove product that are out of stock. You can search in this table by product description.

Product ID	Product Name	Description	Category	Price	Stocks
100730	A & W	A & W ROOT BEER	SOFT DRINKS	60	32
100120	BROWNIES	BROWNIES	BREAD	50	60
100700	TORTILLAS	TORTILLAS PLAIN	SNACKS	10	61
100760	COOKIES	COOKIES	SNACKS	30	35
100200	COKE	COKEZERO	SOFT DRINKS	35	42
100720	KOPKO	KOPKO BROWN	COFFEE	25	63
100670	MOZELLA	MOZELLA CHESSE	DAIRY	20	55
100090	COBRA	COBRA YELLOW	SOFT DRINKS	35	44
100230	MAYONAISE	MAYONAISE GREEN	OTHERS	45	45
100510	PIATTOS	PIATTOS CHESSE	DAIRY	25	13
100930	NACHOS	NACHOS CHESSE	SNACKS	20	56
100110	MAGGI	MAGGI ATA	SNACKS	25	40
100220	BISCUITS	BISCUITS CHOCOLATE	SNACKS	15	55
100111	CROISSANT	CROISSANT	BREAD	50	37
100112	MUFFINS	MUFFINS VANILLA	BREAD	55	37
100530	CHEESE CAKE	CHEESE CAKE	DAIRY	35	37
100661	FRUIT CAKE	FRUIT CAKE MIXED	BREAD	40	34
100770	BUNS	BUNS	BREAD	40	31

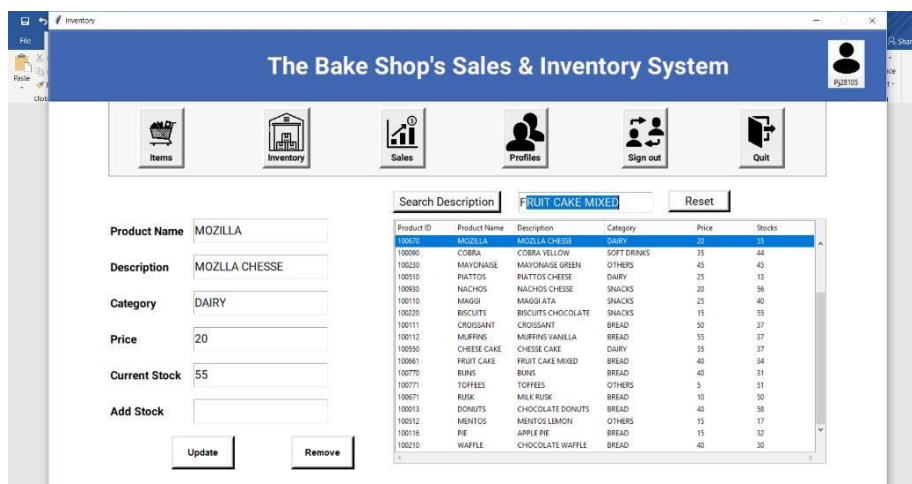
You can select any row of the table the entry boxed will fill automatically.

Product ID	Product Name	Description	Category	Price	Stocks
100730	A & W	A & W ROOT BEER	SOFT DRINKS	60	32
100120	BROWNIES	BROWNIES	BREAD	50	60
100700	TORTILLAS	TORTILLAS PLAIN	SNACKS	10	61
100760	COOKIES	COOKIES	SNACKS	30	35
100200	COKE	COKEZERO	SOFT DRINKS	35	42
100720	KOPKO	KOPKO BROWN	COFFEE	25	63
100670	MOZELLA	MOZELLA CHESSE	DAIRY	20	55
100090	COBRA	COBRA YELLOW	SOFT DRINKS	35	44
100230	MAYONAISE	MAYONAISE GREEN	OTHERS	45	45
100510	PIATTOS	PIATTOS CHESSE	DAIRY	25	13
100930	NACHOS	NACHOS CHESSE	SNACKS	20	56
100110	MAGGI	MAGGI ATA	SNACKS	25	40
100220	BISCUITS	BISCUITS CHOCOLATE	SNACKS	15	55
100111	CROISSANT	CROISSANT	BREAD	50	37
100112	MUFFINS	MUFFINS VANILLA	BREAD	55	37
100530	CHEESE CAKE	CHEESE CAKE	DAIRY	35	37
100661	FRUIT CAKE	FRUIT CAKE MIXED	BREAD	40	34
100770	BUNS	BUNS	BREAD	40	31

To update product detail just press update button. Table is dynamic in nature it will get automatically updated. To remove product just press remove button.



You can search Product by Description. Search Box is a suggestion box it will automatically provide suggestions to the user.



To reset search option just press reset button.

- **Items-** You can add items to the inventory through items button. Here Product id should be in numbers. Product id and product description should be unique. Press add item button to add item to the bakery or Back button to go back to the inventory.

Here Category search box is a suggestion box it will fill automatically and will give suggestion to the user.

**The Bake Shop's Sales & Inventory System**

Items Inventory Sales Profiles Sign out Quit

Product Id: 100290

Product Name: Garlic Bread

Description: Garlic Chesse Bread

Category: BREAD

Price: 15

Stock: 24

Add Item Back

- **Sales-:** Through Sales button you can check the current sale progress in your bakery. You can transaction no., Invoice no. of bakery. You can Search any bill details by Invoice no.

User can also check the date and time of the transaction.

**The Bake Shop's Sales & Inventory System**

Items Inventory Sales Profiles Sign out Quit

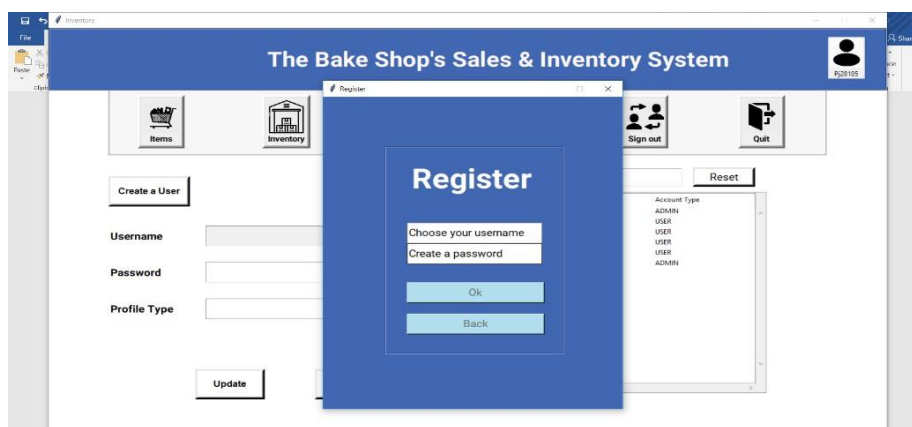
Search Invoice No. 111 Reset

Transaction ID	Invoice No.	Product ID	Description	Quantity	Total Price	Date	Time
100	111	100730	A & W ROOT BEER	1	60	19-11-18	22:10:23
101	111	100110	MAGGI ATA	1	25	19-11-18	22:10:31
102	111	100110	MAGGI ATA	3	75	19-11-18	22:37:10
103	111	100220	BISCUITS CHOCOLATE	2	30	19-11-18	22:37:20
104	112	100110	MAGGI ATA	2	50	19-11-18	22:41:32
105	112	100720	KOPKO BROWN	2	50	19-11-18	22:41:42
108	118	100110	MAGGI ATA	1	25	19-11-18	22:19:08
109	118	100116	APPLE PIE	1	15	21-11-18	20:57:51
110	118	100730	A & W ROOT BEER	2	120	21-11-18	20:57:59
111	118	100110	MAGGI ATA	1	25	21-11-18	20:58:09
112	118	100270	MICELLA CHESSE	2	40	21-11-18	21:36:52
113	120	100770	BUNS	3	120	21-11-18	22:51:30
114	120	100930	CHESSE CAKE	1	35	21-11-18	22:55:12
115	120	100760	COOKIES	1	30	21-11-18	22:56:11
116	120	100280	COKIZERO	2	70	21-11-18	22:56:17
117	121	100111	CROISSANT	3	150	21-11-18	22:56:18

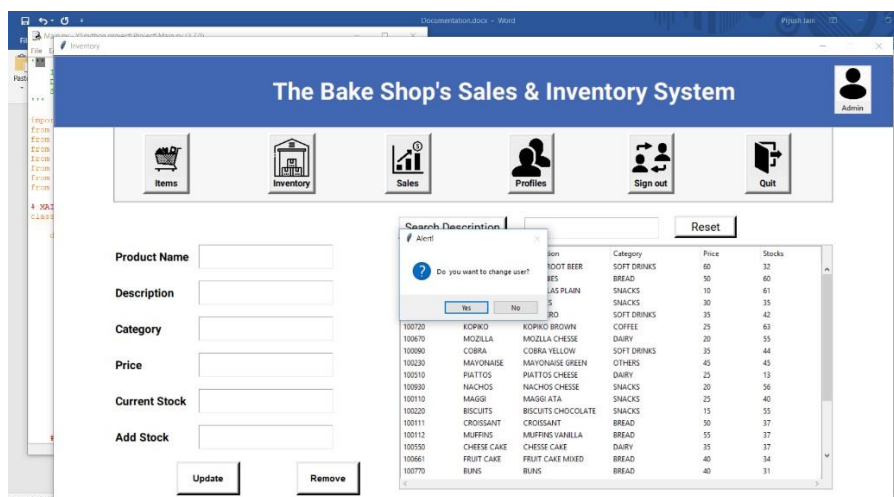
- **Profiles-:** To check Profiles currently in bakery's database user can click profile button. All entry boxes will fill automatically if you select any row of the table. You can edit or remove accounts from the data base. You can also edit passwords of the user profiles.



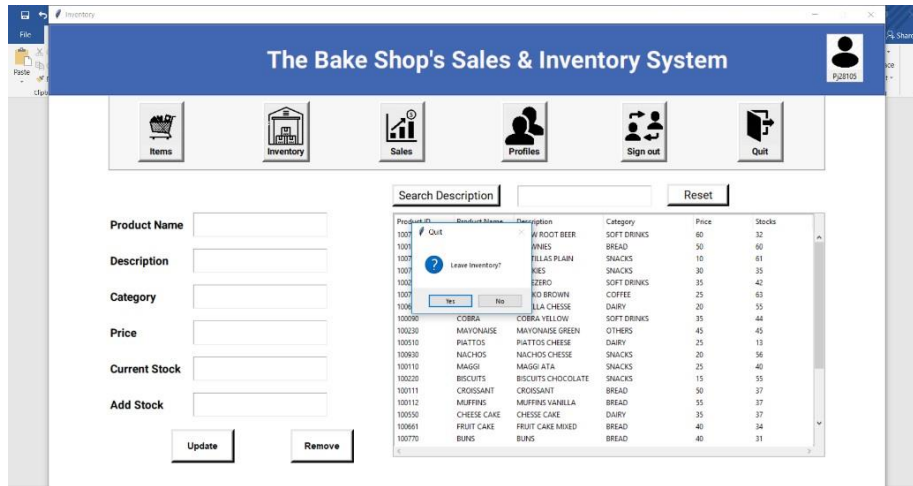
User can also create a new profile by clicking on create user button. A register user window will pop up. By default, it will create a User account. To make it Admin type you can edit its property through profile type combo box.



- Sign Out:- To sign out click sign out button. You will get back to Login window.



- **Quit-:** To quit Inventory user can press quit button to exit the Inventory or X button on the window.

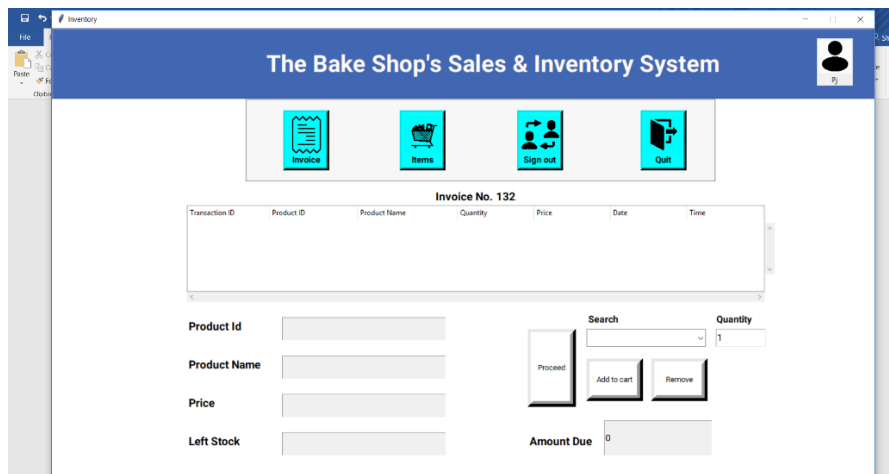


## **USER ACCOUNT**

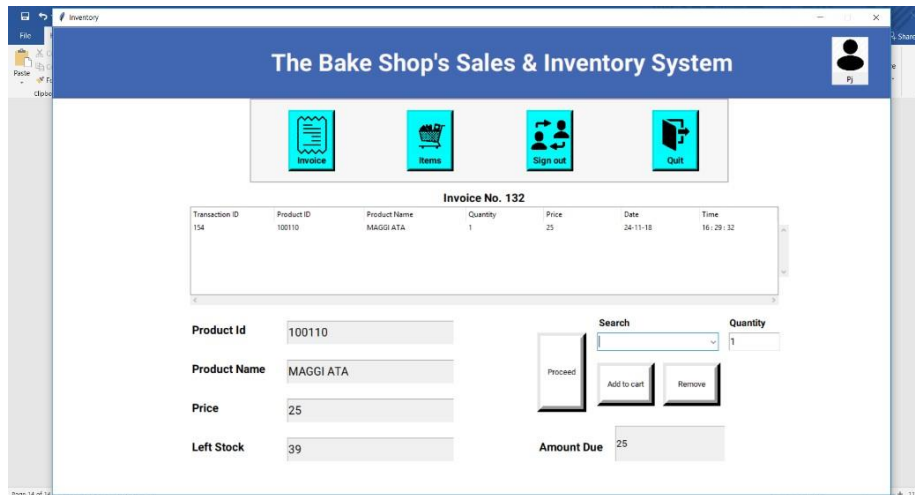
This is a Limited access account. This account is for employees of the bakery. Through this account you can sell products, make transaction, can see current products in bakery etc.

- **Invoice-:** By clicking Invoice Button. You will enter to the transaction window. There is Invoice no. at top, this is bill no. by which you can fetch the bill details through Sales button in Admin type account.

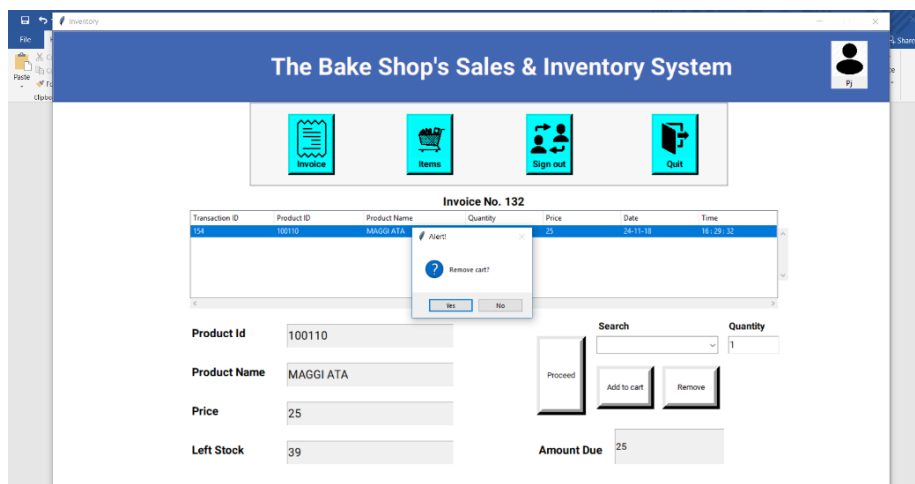
After Invoice no. there is transaction table. It will get automatically filled when you enter a product for sale.



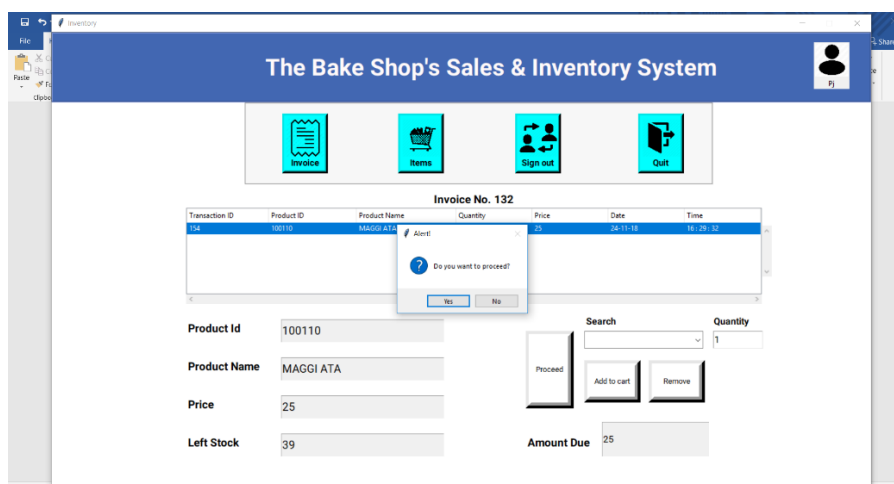
User can search Product by description or Product id in search box. This box will provide suggestions to the user. To make entry in Transaction table Press Add cart button. Automatically its product detail will get displayed. Here product details are not for edit it is a User type account.



You can remove any record from current transaction by clicking remove button.

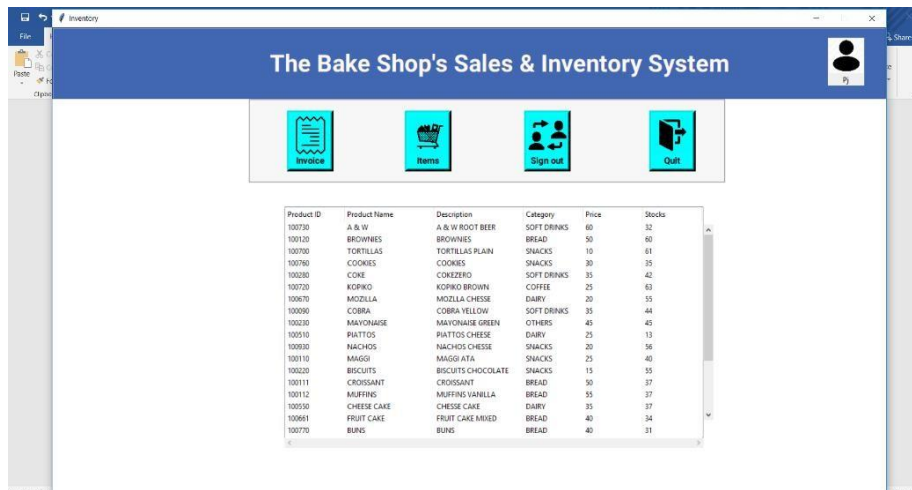


To make a Bill User can click proceed button.



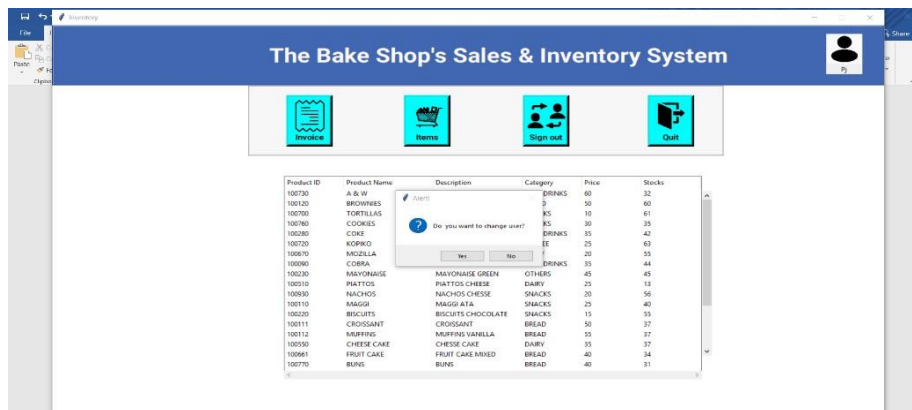
After proceed, it will prompt user to take a print out of this transaction.

- Items:- User can view Items in the bakery. The details are not for edit.



Product ID	Product Name	Description	Category	Price	Stocks
100730	A & W	A & W ROOT BEER	SOFT DRINKS	60	32
100120	BROWNIES	BROWNIES	BREAD	50	60
100700	TORTILLAS	TORTILLAS PLAIN	SNACKS	10	61
100780	COOKIES	COOKIES	SNACKS	30	35
100280	COKE	COKEZERO	SOFT DRINKS	35	42
100720	KOPKO	KOPKO BROWN	COFFEE	25	63
100670	MOZELLA	MOZELLA CHEESE	DAIRY	20	55
100000	COBRA	COBRA YELLOW	SOFT DRINKS	35	44
100220	MAYONAISE	MAYONAISE GREEN	OTHERS	45	45
100510	PIATTOS	PIATTOS CHEESE	DAIRY	25	13
100930	NACHOS	NACHOS CHEESE	SNACKS	20	56
100110	MAGGI	MAGGI ATA	SNACKS	25	40
100220	BISCUTTS	BISCUTTS CHOCOLATE	SNACKS	15	55
100111	CROISSANT	CROISSANT	BREAD	50	37
100112	MUFFINS	MUFFINS VANILLA	BREAD	35	37
100500	CHEESE CAKE	CHEESE CAKE	DAIRY	35	37
100961	FRUIT CAKE	FRUIT CAKE MIXED	BREAD	40	34
100770	BUNS	BUNS	BREAD	40	31

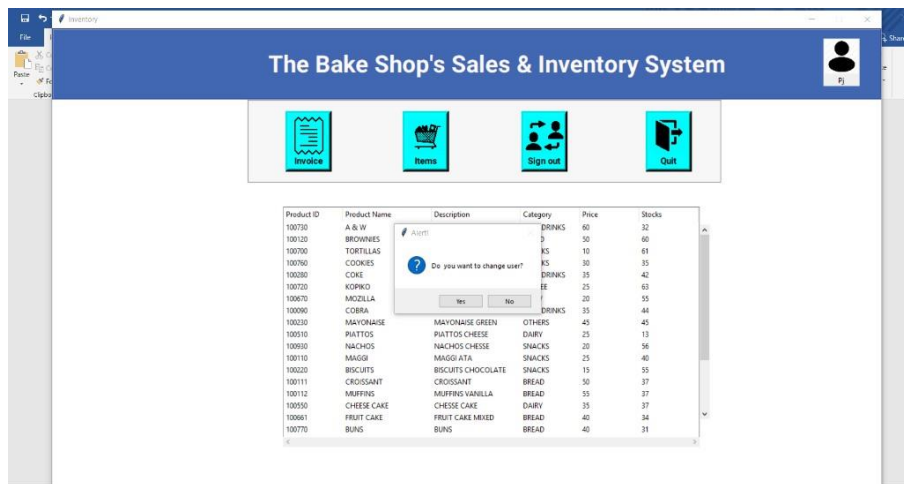
- Sign Out:- To sign out click sign out button. You will get back to Login window.



Product ID	Product Name	Description	Category	Price	Stocks
100730	A & W	A & W ROOT BEER	SOFT DRINKS	60	32
100120	BROWNIES	BROWNIES	BREAD	50	60
100700	TORTILLAS	TORTILLAS PLAIN	SNACKS	10	61
100780	COOKIES	COOKIES	SNACKS	30	35
100280	COKE	COKEZERO	SOFT DRINKS	35	42
100720	KOPKO	KOPKO BROWN	COFFEE	25	63
100670	MOZELLA	MOZELLA CHEESE	DAIRY	20	55
100000	COBRA	COBRA YELLOW	SOFT DRINKS	35	44
100220	MAYONAISE	MAYONAISE GREEN	OTHERS	45	45
100510	PIATTOS	PIATTOS CHEESE	DAIRY	25	13
100930	NACHOS	NACHOS CHEESE	SNACKS	20	56
100110	MAGGI	MAGGI ATA	SNACKS	25	40
100220	BISCUTTS	BISCUTTS CHOCOLATE	SNACKS	15	55
100111	CROISSANT	CROISSANT	BREAD	50	37
100112	MUFFINS	MUFFINS VANILLA	BREAD	35	37
100500	CHEESE CAKE	CHEESE CAKE	DAIRY	35	37
100961	FRUIT CAKE	FRUIT CAKE MIXED	BREAD	40	34
100770	BUNS	BUNS	BREAD	40	31

- Quit:- To quit Inventory user can press quit button to exit the Inventory or X button on the window.





## Testing and Validation

### Unit Testing

#### Overview :-

Unit testing involves testing individual components or modules of a system to ensure they perform as expected. For this project, unit tests will focus on:

- 1) Inventory operations (adding, updating, removing products).
- 2) Sales operations (recording transactions and updating stock).
- 3) Report generation accuracy.
- 4) Validation functions (e.g., stock availability and user authentication).

#### Objectives:-

- 1) Ensure the system's core functions operate as expected.
- 2) Identify and resolve bugs early in the development cycle.
- 3) Validate the correctness of algorithms used in stock updates, sales processing, and reporting.

## **Implementation:-**

### 1)Setup Unit Testing Framework:

- a)Use Python's built-in unit test library for writing and running tests.

### 2)Key Functions to Test:

- a)add\_product(): Check if products are correctly added with valid inputs.
- b)update\_inventory(): Validate correct updates after a sale.
- c)generate\_report(): Ensure generated data matches the expected output.
- d>alert\_low\_stock(): Confirm alerts trigger at predefined thresholds.

### 3)Test Data:

- a)Mock data will be used to simulate inventory, sales, and users.

#### b)Example dataset:

```
inventory = [  
    {"product_id": 1, "name": "Widget", "price": 10, "stock": 50},  
    {"product_id": 2, "name": "Gadget", "price": 15, "stock": 20}  
]
```

### 4)Test Cases:

- a)Validate successful addition and removal of products.
- b)Ensure stock levels decrease appropriately after sales.
- c)Confirm correct total sales and revenue calculations.
- d)Test for invalid inputs (e.g., negative stock, duplicate IDs).

## **Summary of Test Cases and Results:-**

### 1)Add Product:

Verified successful addition of a new product.

Result: Passed.

2)Add Duplicate Product:

Ensured duplicate product IDs are not allowed.

Result: Passed.

3)Update Inventory:

Validated stock updates after a sale with sufficient stock.

Result: Passed.

4)Insufficient Stock:

Confirmed proper handling when stock is insufficient for a sale.

Result: Passed.

5)Alert Low Stock:

Checked alerts for items below a predefined stock threshold.

Result: Passed.

6)Generate Report:

Ensured the system generates correct reports based on current inventory.

Result: Passed.

All test cases passed successfully, confirming that the core functionalities of the Sales and Inventory Management System work as intended. This demonstrates the system's robustness and readiness for further enhancements or deployment.

## **Conclusion**

The Sales and Inventory Management System developed using Python has successfully demonstrated its capability to streamline and automate key operations in sales and inventory tracking. This project effectively integrates various functionalities, including inventory management, sales processing, and report generation, ensuring a user-friendly experience for stakeholders. By leveraging Python's robust libraries and tools, the system achieves accurate computations, real-time data handling, and efficient processing, which are essential for modern business operations.

**Key Accomplishments:**

1. **Enhanced Efficiency:** The system reduces manual efforts in tracking sales and inventory, minimizing errors and saving time.
2. **User-Friendly Interface:** Designed with clear navigation and intuitive workflows, it caters to users with varying technical expertise.
3. **Data Accuracy:** By automating calculations and updates, the system ensures data integrity and real-time accuracy.
4. **Comprehensive Reporting:** Generates detailed reports for inventory levels and sales trends, aiding in better decision-making.

**Limitations:**

1. **Scalability Challenges:** The current implementation may require optimization to handle large-scale data or concurrent users efficiently.
2. **Limited Features:** While the core functionalities are covered, advanced features like predictive analytics or integration with third-party tools are not included.
3. **System Dependency:** The system's effectiveness relies on the consistent updating of data by users.