

## PYTHON PROJECT REPORT

## INVENTORY MANAGEMENT SYSTEM

***Submitted by***

***Sudharsanprakalathan Vm***

***2303722810622163***

**BACHELOR OF ENGINEERING**

## *in*

**COMPUTERAND COMMUNICATION ENGINEERING**

**SRI ESHWAR COLLEGE OF ENGINEERING**

**(AN AUTONOMOUS INSTITUTION)**

**COIMBATORE – 641 202**

**JUNE - JULY 2024**

**BONAFIDE CERTIFICATE**

Certified that this project report **“ INVENTORY MANAGEMENT SYSTEM** ” is the bonafide work of

Sudharsanprakalathan Vm

2303722810622163

who carried out the project work under my supervision

|  |  |
| --- | --- |
|  | …………………………………  **SIGNATURE**  **Dr . V . Kiruthika**  Assistant Professor,  Dept. of Electronics & Communication Engineering,  Sri Eshwar College of Engineering,  Coimbatore-641202. |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
| **1** | **INTRODUCTION** |  |
| **2** | **PROBLEM DESCRPTION** |  |
| **3** | **OBJECTIVE** |  |
| **4** | **SOFTWARE SPECIFICATION** |  |
| **5** | **METHODOLOGY** |  |
| **6** | **IMPLEMENTATION** |  |
| **7** | **RESULT** |  |
| **8** | **CONCLUSION** |  |
| **9** | **FUTURE SCOPE** |  |

**TABLE OF CONTENTS**

**INTRODUCTION**

Inventory management is a critical aspect of any business that deals with physical goods. It involves the supervision of non-capitalized assets (inventory) and stock items. The primary objective is to ensure that the right amount of each product is available at the right time to meet customer demand while minimizing costs. Effective inventory management helps businesses improve their cash flow, minimize waste, and increase profitability..

**PROBLEM DESCRPTION**

Many businesses struggle with maintaining accurate inventory records. Manual processes are prone to errors, leading to overstocking or stockouts, which can result in lost sales and customer dissatisfaction. The need for an automated inventory management system is paramount to streamline operations, reduce human error, and enhance efficiency. The system should allow administrators to manage products, monitor inventory levels, and handle user transactions seamlessly..

**OBJECTIVE**

The objective of the Inventory Management System is to develop a robust software application that helps businesses manage their inventory efficiently. The system should enable administrators to add, update, and delete products, view inventory levels, and generate user bills. Additionally, it should support user registration and login functionalities, ensuring secure access and smooth operation.

**SOFTWARE SPECIFICATION**

The Inventory Management System is developed using Python, which offers simplicity and a wide range of libraries for various functionalities. Key specifications include:

* **Programming Language**: Python
* **User Interface**: Command-line interface
* **Data Storage**: In-memory data structures (dictionaries) for simplicity
* **Security**: Basic password protection for admin access
* **Features**: Product management, inventory tracking, user billing, user registration

**METHODOLOGY**

The development process follows these steps:

1. **Requirement Analysis**: Understanding the needs of the business and the problems with current inventory management practices.
2. **System Design**: Designing the system architecture, including classes and their interactions.
3. **Implementation**: Writing the code to implement the designed functionalities.
4. **Testing**: Verifying that the system works as expected and is free of bugs.
5. **Deployment**: Making the system available for use.
6. **Maintenance**: Regular updates and improvements based on user feedback.

**IMPLEMENTATION**

**class Product:**

**def \_\_init\_\_(self, name, price, quantity):**

**self.name = name**

**self.price = price**

**self.quantity = quantity**

**class InventoryManagementSystem:**

**def \_\_init\_\_(self):**

**self.products = {}**

**self.admin\_password = "admin123"**

**self.admin\_logged\_in = False**

**self.users = {}**

**def add\_product(self, name, price, quantity):**

**if name in self.products:**

**print(f"Product {name} already exists.")**

**else:**

**self.products[name] = Product(name, price, quantity)**

**print(f"Product {name} added successfully.")**

**def update\_product(self, name, price=None, quantity=None):**

**if name in self.products:**

**if price is not None:**

**self.products[name].price = price**

**if quantity is not None:**

**self.products[name].quantity = quantity**

**print(f"Product {name} updated successfully.")**

**else:**

**print(f"Product {name} does not exist.")**

**def delete\_product(self, name):**

**if name in self.products:**

**del self.products[name]**

**print(f"Product {name} deleted successfully.")**

**else:**

**print(f"Product {name} does not exist.")**

**def inventory(self):**

**if not self.products:**

**print("No products in inventory.")**

**else:**

**for name, product in self.products.items():**

**print(f"Name: {name}, Price: {product.price}, Quantity: {product.quantity}")**

**def admin\_login(self):**

**password = input("Enter admin password: ")**

**if password == self.admin\_password:**

**print("Login successful!")**

**self.admin\_logged\_in = True**

**else:**

**print("Incorrect password. Access denied.")**

**def admin\_menu(self):**

**while self.admin\_logged\_in:**

**print("\nAdmin Menu")**

**print("1. Add Product")**

**print("2. Update Product")**

**print("3. Delete Product")**

**print("4. Inventory")**

**print("5. Logout")**

**choice = input("Enter your choice: ")**

**if choice == '1':**

**name = input("Enter product name: ")**

**price = float(input("Enter product price: "))**

**quantity = int(input("Enter product quantity: "))**

**self.add\_product(name, price, quantity)**

**elif choice == '2':**

**name = input("Enter product name: ")**

**price = float(input("Enter new product price: "))**

**quantity = int(input("Enter new product quantity: "))**

**self.update\_product(name, price, quantity)**

**elif choice == '3':**

**name = input("Enter product name to delete: ")**

**self.delete\_product(name)**

**elif choice == '4':**

**self.inventory()**

**elif choice == '5':**

**self.admin\_logged\_in = False**

**print("Logged out.")**

**def user\_bill(self):**

**user\_id = input("Enter user ID or mobile number: ")**

**if user\_id in self.users:**

**total\_price = 0**

**while True:**

**product\_name = input("Enter product name to purchase (or 'done' to finish): ")**

**if product\_name.lower() == 'done':**

**break**

**if product\_name in self.products:**

**quantity = int(input("Enter quantity: "))**

**if quantity <= self.products[product\_name].quantity:**

**self.products[product\_name].quantity -= quantity**

**total\_price += self.products[product\_name].price \* quantity**

**else:**

**print("Not enough quantity available.")**

**else:**

**print("Product not found.")**

**print(f"Total bill for user {user\_id}: ${total\_price}")**

**else:**

**print("User not found.")**

**def register\_user(self):**

**user\_id = input("Enter user ID or mobile number: ")**

**if user\_id in self.users:**

**print("User already exists.")**

**else:**

**self.users[user\_id] = {}**

**print(f"User {user\_id} registered successfully.")**

**if \_\_name\_\_ == "\_\_main\_\_":**

**inventory\_system = InventoryManagementSystem()**

**while True:**

**print("\nWelcome to Inventory Management System")**

**print("1. Admin Login")**

**print("2. User Bill")**

**print("3. Register User")**

**print("4. Exit")**

**user\_choice = input("Enter your choice: ")**

**if user\_choice == "1":**

**inventory\_system.admin\_login()**

**if inventory\_system.admin\_logged\_in:**

**inventory\_system.admin\_menu()**

**elif user\_choice == "2":**

**inventory\_system.user\_bill()**

**elif user\_choice == "3":**

**inventory\_system.register\_user()**

**elif user\_choice == "4":**

**print("Exiting the system. Goodbye!")**

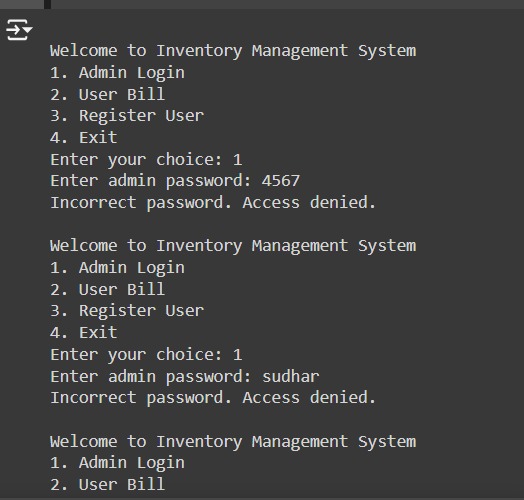
**break**

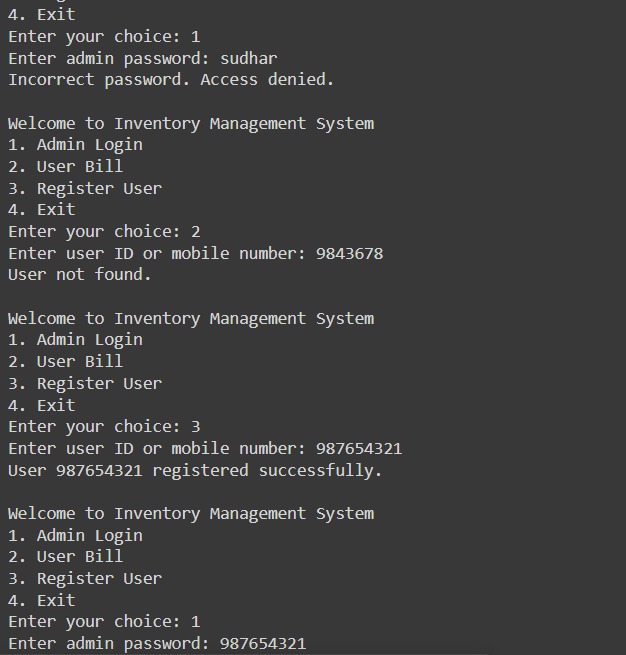
**else:**

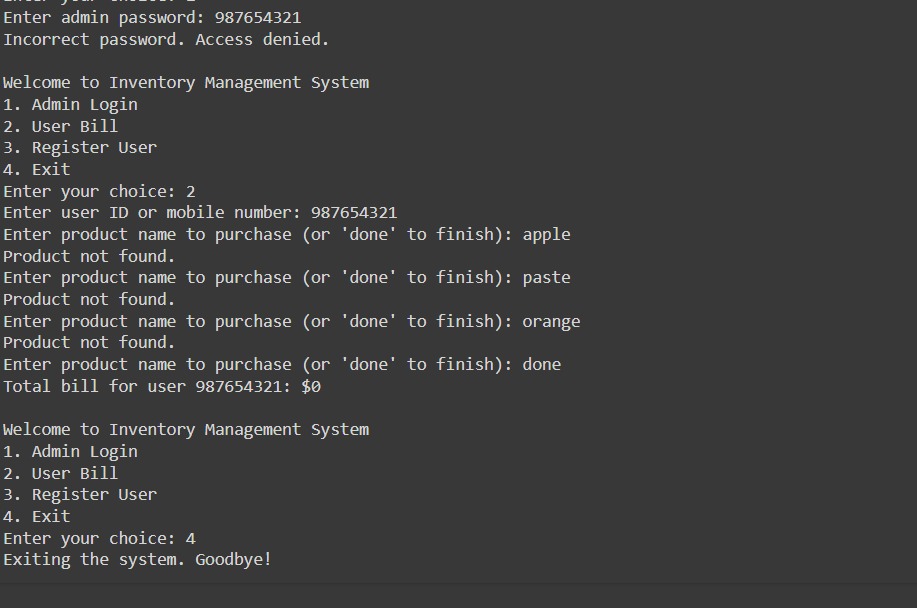
**print("Invalid choice. Please try again.")**

**RESULT**

The implemented system successfully manages inventory, allowing administrators to add, update, delete products, and view current inventory levels. Users can be registered, and their transactions are handled efficiently, ensuring accurate billing and inventory updates.







**CONCLUSION**

The Inventory Management System effectively addresses the challenges of manual inventory management. It provides a reliable and user-friendly platform for managing products, tracking inventory, and handling user transactions. The system improves operational efficiency, reduces errors, and ensures that inventory levels are maintained accurately.

**FUTURE SCOPE**

Future improvements and enhancements could include:

* **Database Integration**: Implementing a database for persistent storage.
* **Graphical User Interface (GUI)**: Developing a GUI for better user experience.
* **Advanced Security**: Implementing more robust security measures.
* **Reporting**: Adding functionality to generate detailed reports and analytics.
* **Integration with Other Systems**: Ensuring compatibility with accounting software and other business systems.
* .