**System Name:**

Inventory Management System (IMS)

**Prepared for:**  
Royal Organization Retailers  
**Prepared by:**  
GROUP:13

**Course Number:**

BIS698\_W1700

**Course Name:**

Capstone Project

**Contents**

1. Background
2. Business Problems
3. Project Description
   * 3.1 System Purpose
   * 3.2 Core Functional Requirements
   * 3.3 Out-of-Scope Features
4. Project Feasibility
   * 4.1 Economic Feasibility
   * 4.2 Technical Feasibility
   * 4.3 Schedule Feasibility
   * 4.4 Operational Feasibility
5. Use Cases
   * 5.1 Admin Login
   * 5.2 Product Management
   * 5.3 Inventory Tracking
   * 5.4 Customer Login and Order Placement
6. Process Model with DFD

6.1 Context Level

6.2 Level 0 DFD

6.3 Workflow Description

1. Task List
2. Critical Path Model
3. Entity Relationship Diagram
4. UI Designs
5. Conclusion

**1. Background**

The **Inventory Management System (IMS)** is being developed for **the Royal Organization Retailers**, a small-scale retail store that manages a variety of products. The store currently uses a manual inventory tracking system, which is time-consuming and prone to errors. As sales and inventory volumes increase, an automated system is needed to **streamline inventory tracking and product management**. The system will be built using **Python Tkinter** for the user interface and **MySQL** for secure inventory data storage.

**2. Business Problems**

**Royal Organization Retailers** faces the following inventory management challenges:

* **Manual Inventory Tracking:** The store relies on paper or spreadsheets for inventory tracking, causing errors, duplications, and inefficiencies.
* **Limited Data Accuracy:** Without real-time updates, maintaining an accurate stock record is challenging, affecting product availability and sales.
* **Slow Stock Replenishment:** The manual system delays stock level identification, leading to stockouts and late replenishments.
* **Lack of Reporting:** The current system lacks analytical reports, making it difficult to track inventory levels and sales trends.

These challenges **affect stock control, cause missed sales opportunities, and lead to operational inefficiencies**.

**3. Project Description**

The **Inventory Management System (IMS)** is a **desktop-based** solution that enables **inventory tracking, product management, and reporting** for **Royal Organization Retailers**. The system will be built using **Python Tkinter** for the GUI and **MySQL** for database management.

**3.1 System Purpose**

The primary objectives of IMS are to:

* **Track Inventory:** Maintain a structured record of all products, stock levels, prices, and supplier details.
* **Stock Management:** Allow users to **add, update, and remove products** efficiently.
* **Reporting:** Provide **inventory reports and analytics** to aid decision-making.

The system will **automate inventory tracking, improve stock accuracy, and offer reporting tools** to enhance management efficiency.

**3.2 Core Functional Requirements**

The system will include the following functionalities:

* **User Authentication:** Secure login access for **Admin and Customer** roles.
* **Admin Functionalities:**
  + **Add Products:** Admin can add new products to the inventory.
  + **Edit Product Details:** Admin can update product price, quantity, and supplier details.
  + **Delete Products:** Admin can remove products from the system.
  + **Generate Reports:** Admin can generate inventory reports and analytics.
* **Customer Functionalities:**
  + **Add Products to Cart:** Customers can browse products and add them to their cart.
  + **Place Orders:** Customers can place orders for selected products.
* **Inventory Tracking:** The system **automatically adjusts stock levels** after product modifications.
* **Database Integration:** The system will store inventory data securely in **MySQL**.

**3.3 Out-of-Scope Features**

The initial version of IMS will **not include**:

* **Online Sales Integration:** No integration with e-commerce platforms.
* **Advanced Analytics:** No predictive sales forecasting or complex data visualization.
* **Mobile Application:** The system will be **desktop-based only**.

**4. Project Feasibility**

**4.1 Economic Feasibility**

IMS will provide **multiple benefits** to **Royal Organization Retailers**, such as:

* **Improved Accuracy:** Automated tracking reduces errors and enhances stock records.
* **Time Savings:** Eliminates the need for manual inventory tracking and reporting.
* **Better Stock Control:** **Real-time updates** ensure timely replenishment.
* **Enhanced Decision-Making:** Reports provide insights into **inventory trends and stock movement**.

**Developer Cost Estimate:**

* **Number of Developers:** 5 Junior Developers
* **Hourly Rate:** $13 per hour
* **Work Duration:** 14 weeks
* **Weekly Work Hours:** 40 hours per developer

**Total Cost Calculation:**

**Total Estimated Development Cost: $36,400**

**5. Use Cases**

**5.1 Admin Login**

* Secure login for administrators to manage the system.

|  |
| --- |
| **Use Case Name:** Admin Login **ID:** UC-1 **Priority:** High |
| **Actor:** Admin |
| **Description:** The administrator logs into the system to manage inventory and system settings. |
| **Trigger:** Admin must have valid login credentials. |
| **Type:** External |
| **Preconditions:**   1. Admin must have valid login credentials. |
| **Normal Course:**   1. The admin navigates to the login page. 2. The admin enters their username and password. 3. The system validates credentials. 4. Upon successful authentication, the admin gains access to the dashboard. |
| **Postcondition:**   1. Admin is logged in and can access system functionalities. |

**5.2 Product Management**

* **5.2.1 Add Products:** Allows admins to add new products to the inventory.

|  |
| --- |
| **Use Case Name:** Add Products **ID:** UC-2 **Priority:** High |
| **Actor:** Admin |
| **Description:** Allows admins to add new products to the inventory. |
| **Trigger:** New product needs to be added. |
| **Type:** External |
| **Preconditions:**   * 1. Admin must be logged in.   2. System is online |
| **Normal Course:**   1. Admin navigates to the 'Add Product' page. 2. Admin enters product details (name, price, quantity, supplier, etc.). 3. The system validates and saves the product details. 4. Confirmation messages are displayed. |
| **Postcondition:**   1. Products are successfully added to the inventory. |

* **5.2.2 Edit Product Details:** Admins can modify product price, quantity, and supplier details.

|  |
| --- |
| **Use Case Name:** Edit Product Details **ID:** UC-3 **Priority:** High |
| **Actor:** Admin |
| **Description:** Admins can modify product price, quantity, and supplier details**.** |
| **Trigger:** When the product details needs to be changed. |
| **Type:** External |
| **Preconditions:**  Admin must be logged in. |
| **Normal Course:**   1. Admin navigates to the 'Product List' page. 2. Admin selects a product to edit. 3. Admin modifies the required fields. 4. The system updates the product details. 5. Confirmation messages are displayed. |
| **Postcondition:**  Product details are successfully updated. |

* **5.2.3 Delete Products:** Admins can remove outdated or incorrect products from the system.

|  |
| --- |
| **Use Case Name:** Delete Products **ID:** UC-4 **Priority:** High |
| **Actor:** Admin |
| **Description:** Admins can remove outdated or incorrect products from the system. |
| **Trigger:** When we want to delete the product from system. |
| **Type:** External |
| **Preconditions:**  Admin must be logged in. |
| **Normal Course:**   1. Admin navigates to the 'Product List' page. 2. Admin selects a product to delete. 3. The system prompts confirmation. 4. Admin confirms deletion. 5. The system removes the product from inventory. |
| **Postcondition:**  Product is successfully removed from the inventory. |

* **5.2.4 Generate Reports:** Admins can generate reports on inventory trends and stock levels.

|  |
| --- |
| **Use Case Name:** Generate Reports **ID:** UC-5 **Priority:** High |
| **Actor:** Admin |
| **Description:** Admins can generate reports on inventory trends and stock levels. |
| **Trigger:** When the admin wants to generate reports. |
| **Type:** External |
| **Preconditions:**  Admin must be logged in. |
| **Normal Course:**   1. Admin navigates to the 'Reports' section. 2. Admin selects a report type (e.g., inventory summary, stock levels). 3. The system generates and displays reports. 4. Admin can download or print the report. |
| **Postcondition:**  The report is successfully generated. |

**5.3 Inventory Tracking**

* Tracks product stock levels and updates inventory in real time.

|  |
| --- |
| **Use Case Name:** Inventory Tracking  **ID:** UC-6 **Priority:** High |
| **Actor:** Admin |
| **Description:** Tracks product stock levels and updates inventory in real-time. |
| **Trigger:** When we want to track stock levels. |
| **Type:** External |
| **Preconditions:**  Admin must be logged in.  Product data must be available in the system. |
| **Normal Course:**   1. Admin navigates to the inventory dashboard. 2. The system displays real-time stock levels. 3. Admin updates stock if necessary. 4. The system records all changes. |
| **Postcondition:**  Inventory data is updated and stored securely. |

**5.4 Customer Login and Order Placement**

* Customers can log in to browse, add products to their cart, and place orders.

|  |
| --- |
| **Use Case Name:** Customer Login and Order Placement **ID:** UC-7 **Priority:** High |
| **Actor:** Customer |
| **Description:** Customers can log in, browse products, add them to their cart, and place orders. |
| **Trigger:** When Customer wants to use IMS. |
| **Type:** External |
| **Preconditions:**  Customer must be registered in the system.  Customer has valid credentials. |
| **Normal Course:**   1. Customer navigates to the login page. 2. Customer enters login credentials. 3. System authenticates the customer. 4. Customer browses products and adds items to the cart. 5. Customer proceeds to check out. 6. System processes the order and confirms placement. |
| **Postcondition:**  Order is successfully placed, and inventory is updated. |

**6.Process Model with DFD**

**6.1 Context Level**

A diagram with text and numbers

AI-generated content may be incorrect.

**6.2 Level 0 DFD**

A diagram of a product management system

AI-generated content may be incorrect.

**6.3 Workflow Description:**

The Inventory Management System (IMS) follows a structured workflow to ensure efficient inventory tracking, product management, and order processing for Royal Organization Retailers. The system consists of two main user roles: Admin and Customer, each performing specific operations within the system.

**Admin Workflow**

The admin is responsible for managing inventory, tracking stock levels, processing orders, and generating reports. Upon logging in, the admin can add new products, update existing product details (such as price, stock quantity, and supplier information), and remove outdated or incorrect items from the inventory. The admin dashboard provides real-time stock updates, ensuring timely replenishment of low-stock items. Additionally, Admin can generate inventory **reports to analyze stock** movement trends, identify fast-moving products, and optimize procurement strategies.

**Customer Workflow**

The Customer interacts with the system by browsing the inventory, adding products to the cart, and placing orders. After logging in, the Customer can view product details, check availability, and proceed with order placement. Once an order is confirmed, the system automatically updates stock levels, ensuring inventory accuracy. The customer can also track their orders through the system, receiving real-time updates on order status.

**Inventory Tracking & Order Processing**

The IMS continuously monitors stock levels to maintain inventory accuracy. Whenever an order is placed, the system checks stock availability, updates inventory records, and notifies the admin of any low-stock items. The system also maintains a record of all transactions, allowing admins to review past orders, generate analytical reports, and make informed inventory decisions.

The system is designed to improve stock control, reduce errors, and automate key processes, ultimately enhancing the efficiency of inventory management for Royal Organization Retailers.

1. **Task List:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Task Name** | **Description** | **Assigned To** |
| **Week 1** | Requirement Analysis | Gather business needs and define project scope | **Ankith Reddy Chintalapelli** |
| **Week 2** | Stakeholder Meetings | Discuss project goals, constraints, and expectations | **Aravind Goud Bathini** |
| **Week 3** | System Design | Develop system architecture, DFDs, and ERD | **Ankith Reddy Chintalapelli** |
| **Week 4** | Database Setup | Implement MySQL database schema and relationships | **Aravind Goud Bathini** |
| **Week 5** | UI Development (Phase 1) | Design customer and admin dashboards | **Aravind Goud Bathini** |
| **Week 6** | UI Development (Phase 2) | Implement navigation, styling, and input forms | **Aravind Goud Bathini** |
| **Week 7** | Backend Development (Phase 1) | Develop inventory tracking and product management | **Tharun Prattipati** |
| **Week 8** | Backend Development (Phase 2) | Implement order processing and stock management | **Pavan Kalyan Mutakoduru** |
| **Week 9** | System Integration | Connect frontend, backend, and database | **Pavan Kalyan Mutakoduru** |
| **Week 10** | Testing & Debugging (Phase 1) | Conduct functional and usability testing | **Pavan Kalyan Mutakoduru** |
| **Week 11** | Testing & Debugging (Phase 2) | Identify and resolve system bugs | **Viswanadh Muttavarapu** |
| **Week 12** | Deployment & Documentation | Deploy final system and prepare user manuals | **Tharun Prattipati** |

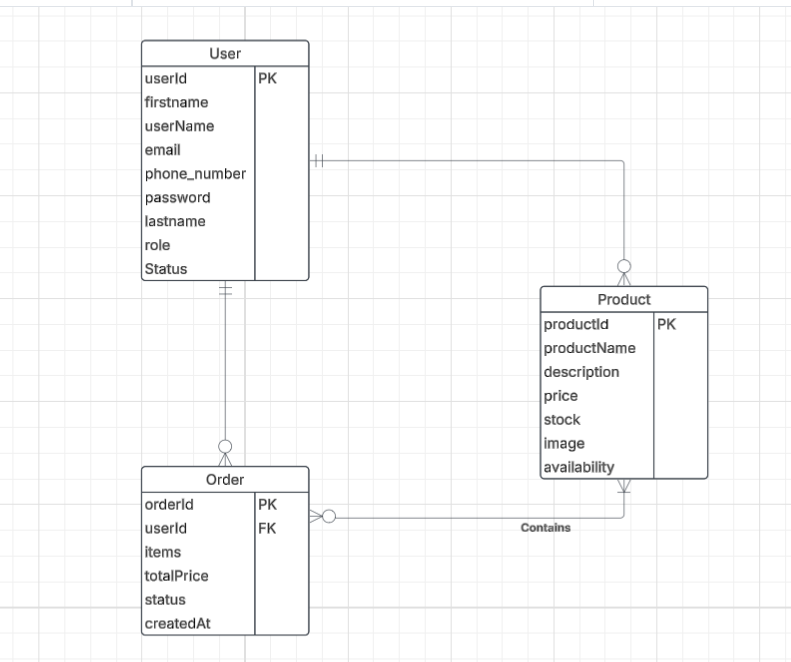
1. **Critical Path Model:**

**A screenshot of a project

AI-generated content may be incorrect.**

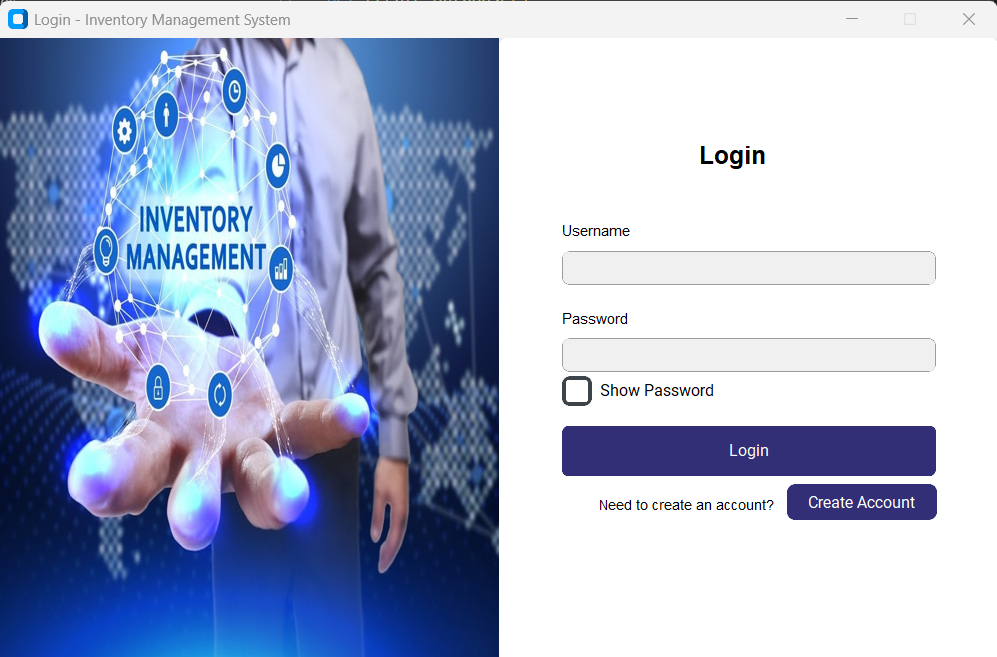
The critical path in project management represents the longest sequence of dependent tasks that determines the minimum project duration. Any delay in these tasks will directly impact on the project’s completion date. In the provided Gantt chart, red-colored tasks indicate critical activities such as System Design, Database Setup, Backend Development, System Integration, and Testing & Debugging. To ensure timely project delivery, these tasks must be closely monitored and completed on schedule.

1. **Entity Relationship Diagram:**

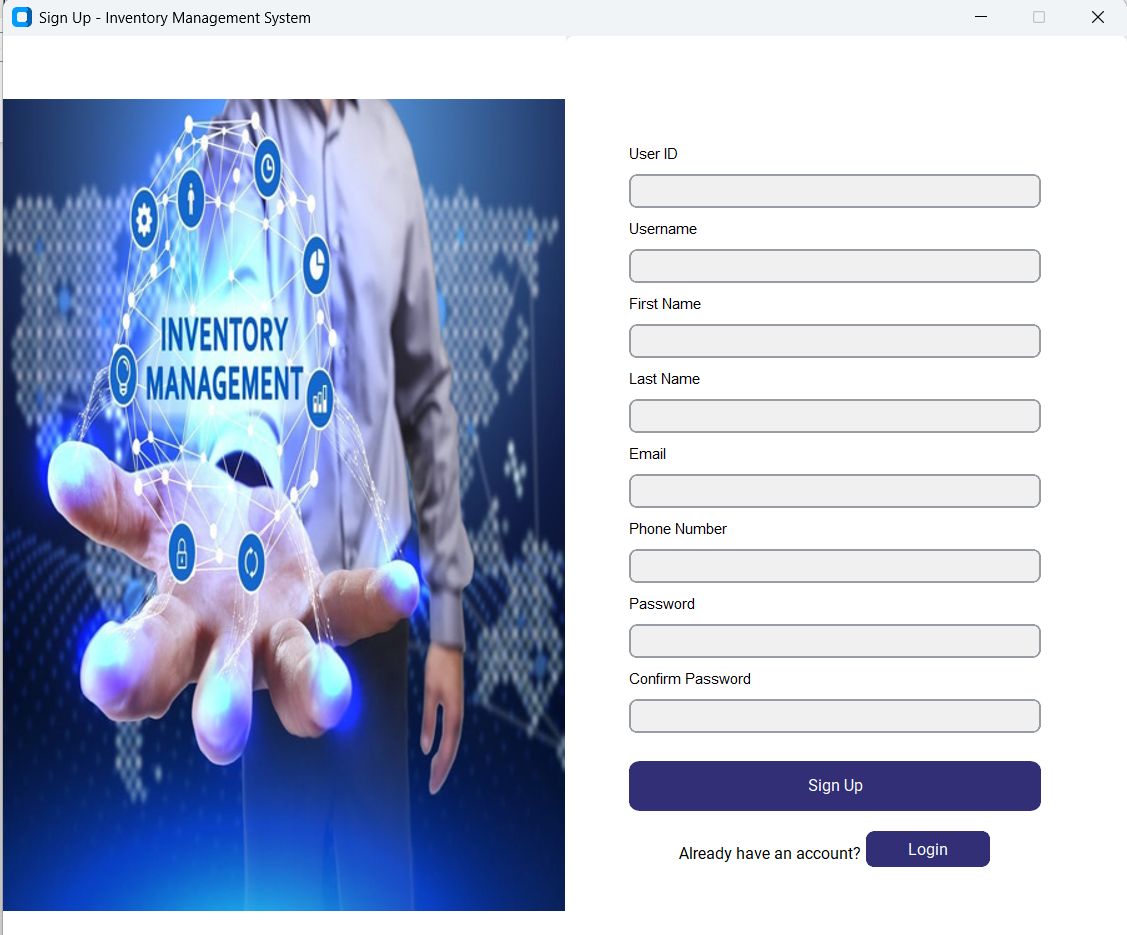
The ERD represents an e-commerce system where Users can be either Customers or Admins (disjoint specialization). Customers place Orders, which include multiple Products. Admins manage Products and update the Inventory. The Inventory table links Products to the Admin responsible for stock updates, ensuring accurate tracking. Orders store details such as items, total price, status, and timestamp, linking to Users who place them. The Product table contains product details, price, stock quantity, and images. The relationships enforce business rules, ensuring Customers can place orders, while Admins manage Inventory and Products.

1. **UI Designs:**

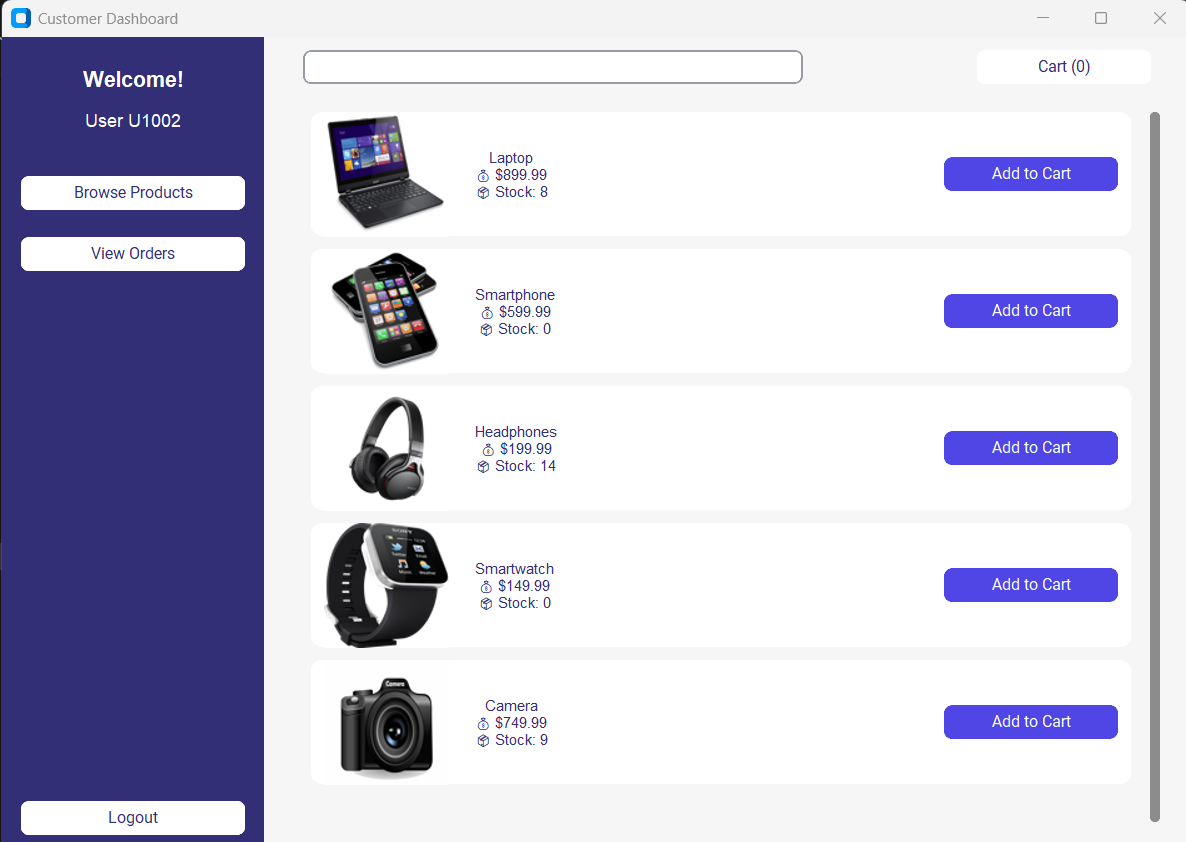
**Login:**

****

**Sign Up:**

****

**Customer Home:**

****

**Browse Products:**

**A screenshot of a computer

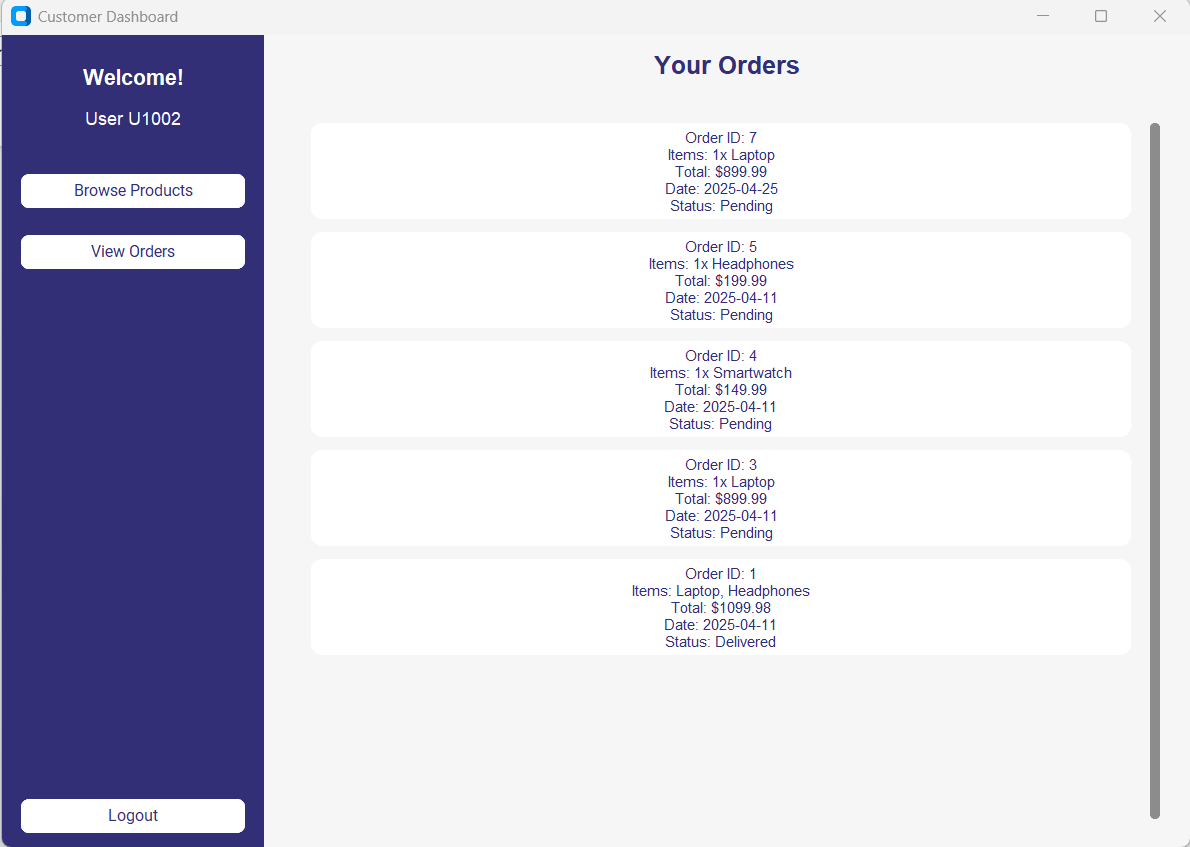
AI-generated content may be incorrect.**

**Cart:**

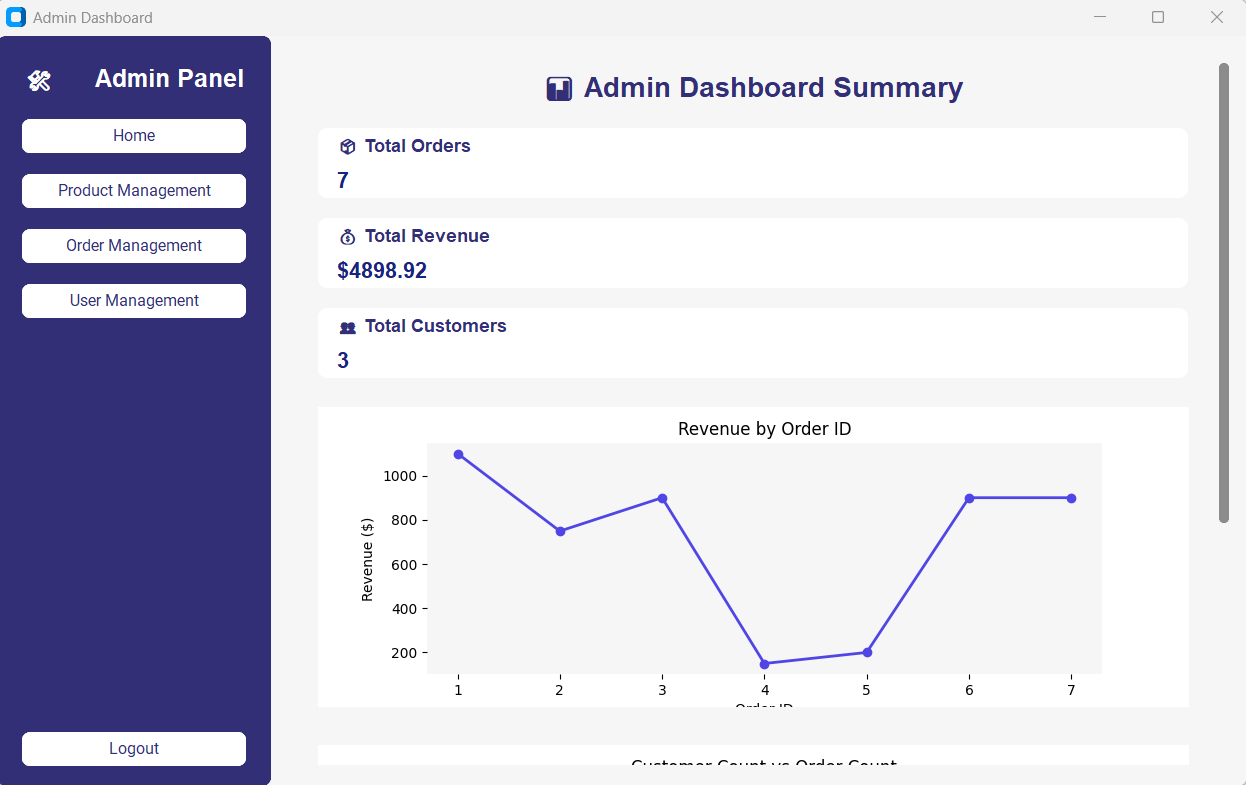
**A computer screen shot of a computer screen

AI-generated content may be incorrect.**

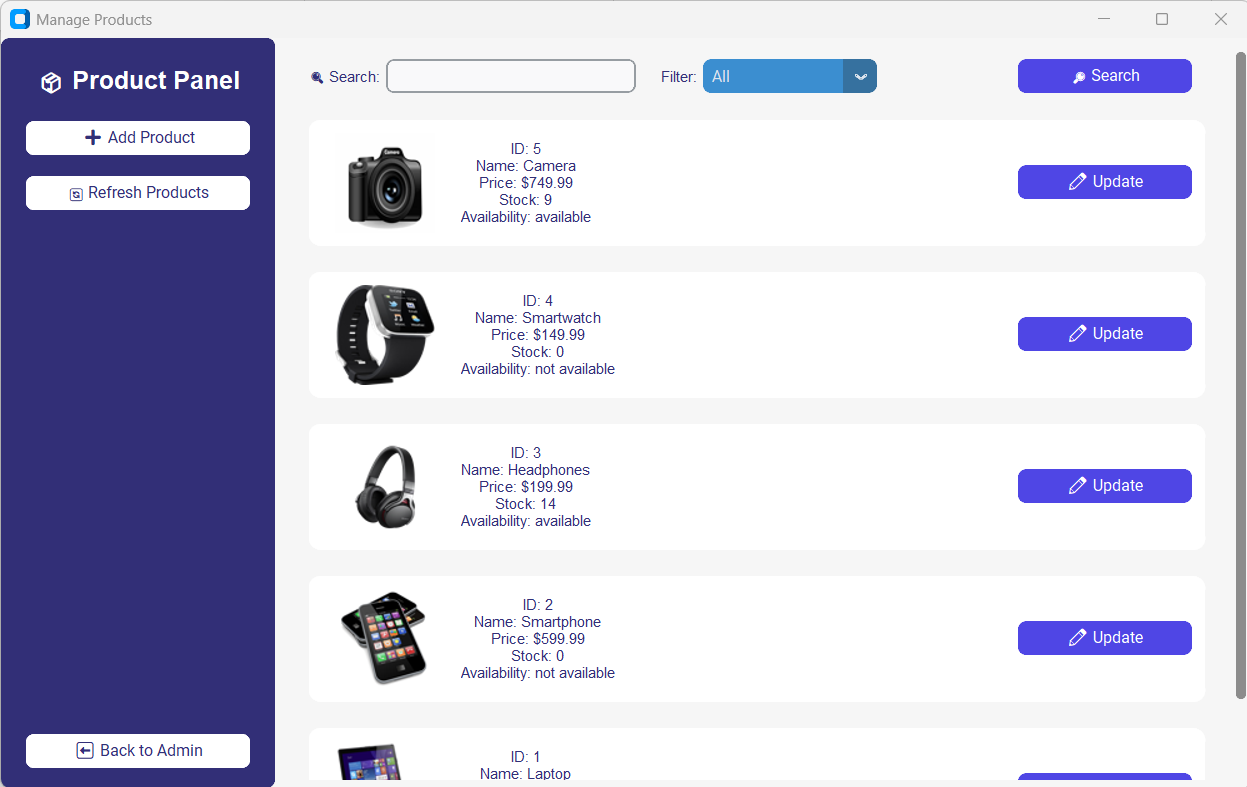
**Orders and Status:**

****

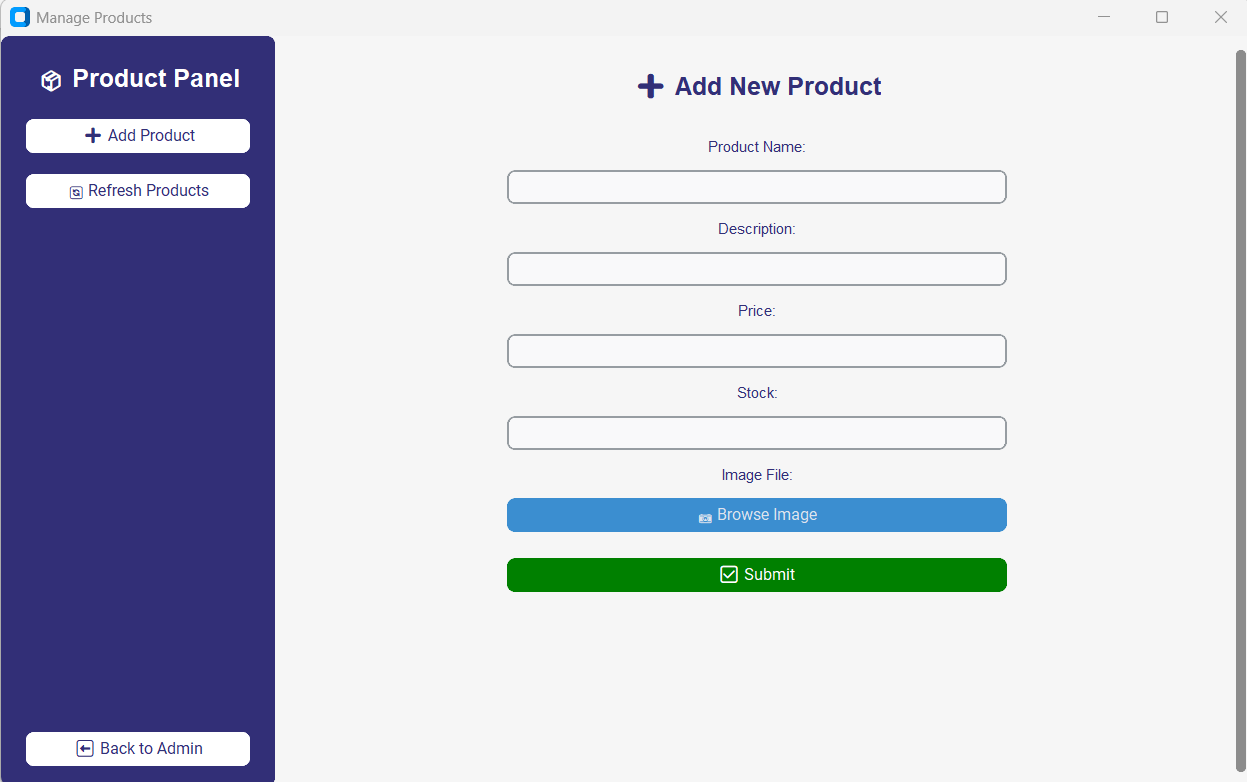
**Admin:**

****

**Product Management:**

****

**Add Product:**

****

**11. Conclusion**

This document outlines the **Inventory Management System (IMS)**, designed to enhance **inventory tracking, stock control, and product management** at **Royal Organization Retailers**. The **Python Tkinter & MySQL**-based system will offer **automation, accuracy, and better decision-making capabilities**. IMS will be completed **on time and within budget**, ensuring **a smooth transition to an optimized inventory management process**.