**DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING**

**BACHELORS IN COMPUTER SYSTEMS ENGINEERING**

**Course Code: CS-115**

**Course Title: Computer Programming**

**Complex Engineering Problem**

**FE Batch 2023, Fall Semester 2023**

#### Grading Rubric

**TERM PROJECT**

**Group Members:**

|  |  |  |
| --- | --- | --- |
| **Student No.** | **Name** | **Roll No.** |
| S1 | **Muhammad Ahmed Raza** | **CS-23097** |
| S2 | **Muhammad Yousuf Mateen** | **CS-23135** |

|  |  |  |  |
| --- | --- | --- | --- |
| **CRITERIA AND SCALES** | | | |
| **S1** | **S2** |
| Criterion 1: Does the application meet the desired specifications and produce the desired outputs? (CPA-1, CPA-3) **[8 marks]** | | | |  |  |
| 1 | 2 | 3 | 4 |
| The application does not meet the desired specifications and is producing incorrect outputs. | The application partially meets the desired specifications and is producing incorrect or partially correct outputs. | The application meets the desired specifications but is producing incorrect or partially correct outputs. | The application meets all the desired specifications and is producing correct outputs. |
| Criterion 2: How well is the code organization? [2 marks] | | | |  |  |
| 1 | 2 | 3 | 4 |
| The code is poorly organized and very difficult to read. | The code is readable only to someone who knows what it is supposed to be doing. | Some part of the code is well organized, while some part is difficult to follow. | The code is well organized and very easy to follow. |
| Criterion 3: How friendly is the application interface? (CPA-1, CPA-3) [2 marks] | | | |  |  |
| 1 | 2 | 3 | 4 |
| The application interface is difficult to understand and use. | The application interface is easy to understand and but not that comfortable to use. | The application interface is very easy to understand and use. | The application interface is very interesting/ innovative and easy to understand and use. |
| Criterion 4: How does the student performed individually and as a team member? (CPA-2, CPA-3) [4 marks] | | | |  |  |
| 1 | 2 | 3 | 4 |
| The student did not work on the assigned task. | The student worked on the assigned task, and accomplished goals partially. | The student worked on the assigned task, and accomplished goals satisfactorily. | The student worked on the assigned task, and accomplished goals beyond expectations. |

# \*Title: Report on Python Shopping Cart Application\*

## \*1. Introduction:\*

The Python Shopping Cart Application presented here is a versatile and user-friendly system designed to simulate an online shopping experience. This report provides an overview of the main features, functionalities, distinguishing characteristics, and proposed improvements of the

Table of Contents

[\*Title: Report on Python Shopping Cart Application\* 2](#_Toc170742099)

[\*1. Introduction:\* 2](#_Toc170742100)

[Abstract: 5](#_Toc170742101)

[Distinguishing Features: 6](#_Toc170742102)

[Flow of Our Project 7](#_Toc170742103)

[Most Challenging Part: 8](#_Toc170742104)

[1) Menu Navigation: 8](#_Toc170742105)

[2) File Handling Edge Cases: 8](#_Toc170742106)

[3) User History Management: 8](#_Toc170742107)

[4) Documentation Clarity: 8](#_Toc170742108)

[5) Code Breaking: 8](#_Toc170742109)

[What we Learn in this Project: 8](#_Toc170742110)

[1) Database (SQL) With Python: 8](#_Toc170742111)

[2) Command-Line Interface (CLI) Interaction: 8](#_Toc170742112)

[3) User Authentication and Authorization: 9](#_Toc170742113)

[4) Class Decomposition: 9](#_Toc170742114)

[5) Exception Handling: 9](#_Toc170742115)

[6) Understanding Customer and Seller Roles: 9](#_Toc170742116)

[7) Product Management for Seller: 9](#_Toc170742117)

[8) Project Documentation: 9](#_Toc170742118)

[Task Distribution: 9](#_Toc170742119)

[1) Muhammad Ahmed Raza CS-23097: 9](#_Toc170742120)

[2) Muhammad Yousuf Mateen CS-23135: 9](#_Toc170742121)

[Future Enhancements: 10](#_Toc170742122)

[1) User Authentication and Authorization: 10](#_Toc170742123)

[2) Graphical User Interface (GUI): 10](#_Toc170742124)

[3) Product Categories and Filtering: 10](#_Toc170742125)

[4) Discounts and Promotions: 10](#_Toc170742126)

[5) Inventory Management: 10](#_Toc170742127)

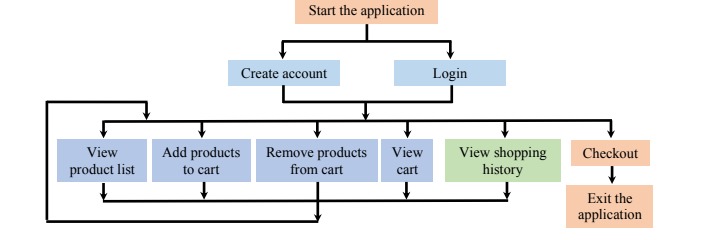
[6) Chatbot Support: 10](#_Toc170742128)

[7) Data Backup and Recovery: 10](#_Toc170742129)

[Reference: 10](#_Toc170742130)

# Abstract:

The goal of this project was to create a shopping mart management system with distinct functionalities for users and administrators. The outcome is a modular and user-friendly application that allows users to interact with a virtual shopping mart, add items to their carts, and check out. Admins can manage products, update stock, and perform account-related tasks. The project delves into user account management, file I/O, and efficient menu-driven interfaces.



# Distinguishing Features:

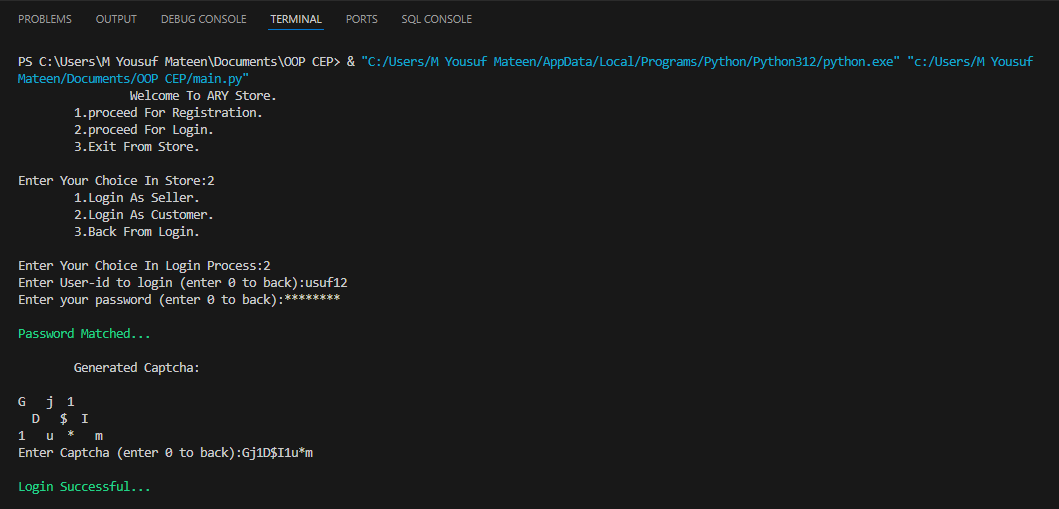


Figure 1: Captcha.

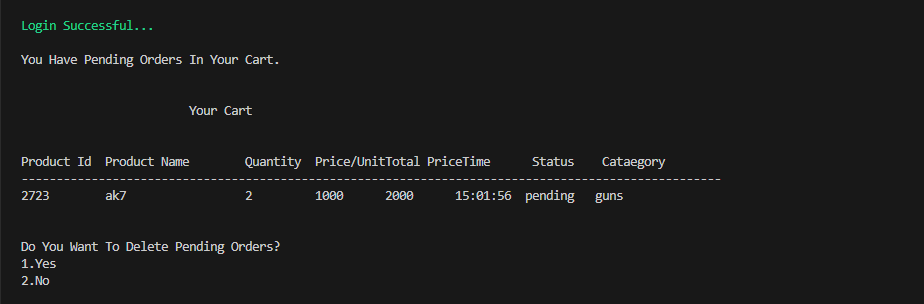


Figure 2: Pending orders will remain in cart.

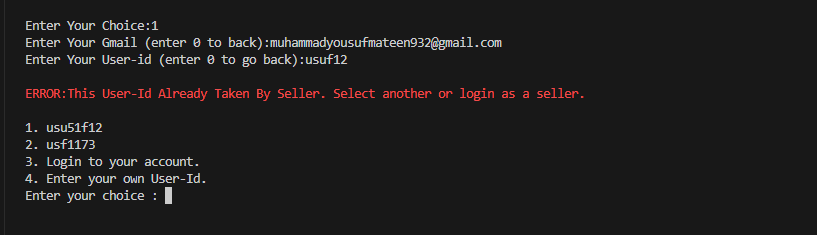


Figure 3: Suggestion if new user-id matched with old user-id.



Figure 4: Verification code will be sent when register via gmail.

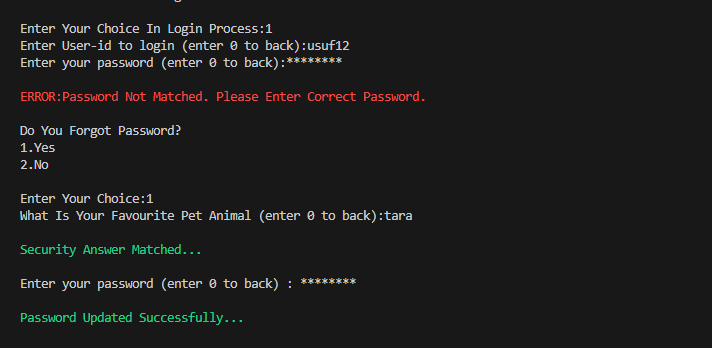
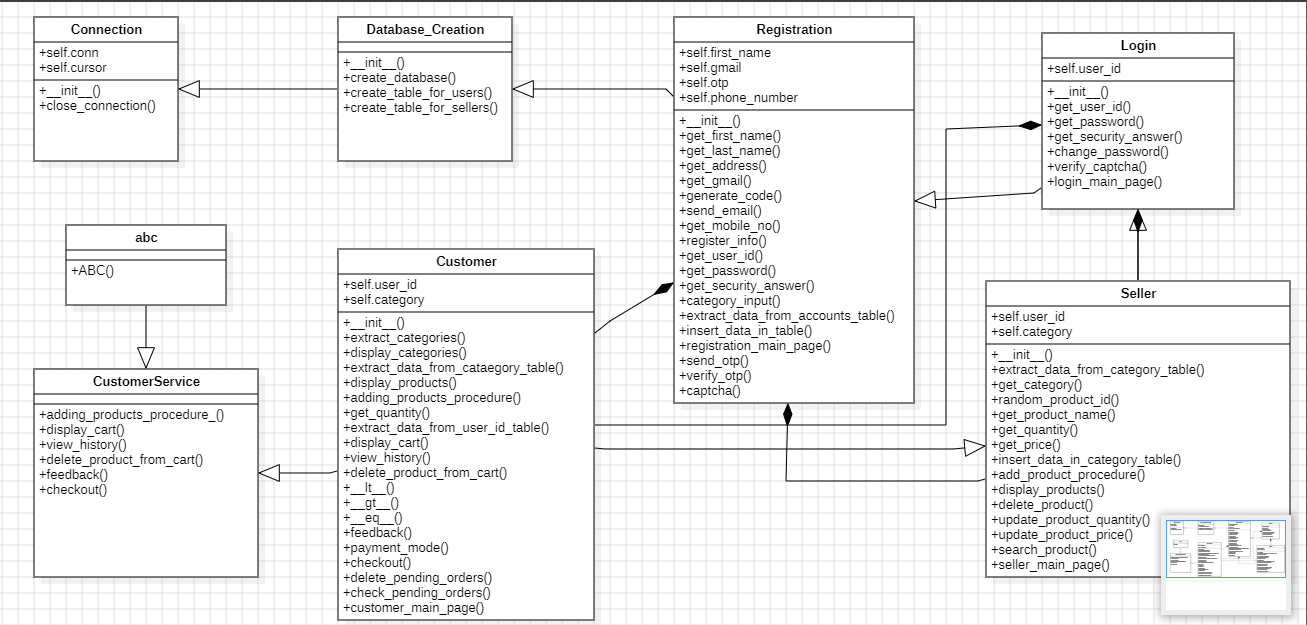


Figure 5: Forgot password option given, when yes required security answer set earlier during registration.

# Flow of Our Project



# Most Challenging Part:

## Menu Navigation:

* Designing a user-friendly menu system for both users and admins.

## Maintaining Different Sellers:

* To maintain different Sellers ,like every sellers can make changes in their respective

Dashboard.

## User History Management:

* Efficiently managing and displaying customer shopping history, especially if the application grows and accumulates a significant amount of historical data.
* Also in managing the history of items present in cart of customer will remain in the cart until next login.

## Documentation Clarity:

* Ensuring that the documentation is clear, comprehensive, and accessible for anyone who may need to understand or modify the code.

## Code Breaking:

* The major issue we faced in this project was the breaking of code into different modules.

# What we Learn in this Project:

## Database (SQL) With Python:

* Working with SQL operations, including creation of file and updating file wherever necessary.
* Handling exceptions related to SQL operations, such as my sql connector error..

## Command-Line Interface (CLI) Interaction:

* Implementing a basic command-line interface for user interaction.
* Processing user input either seller or customer and providing appropriate responses in different text color, distinguishing in between the terminal original color.

## User Authentication and Authorization:

* Implementing a basic user authentication system for both seller and customer using a username-password approach.
* Verification code sent when user signup via gmail and captcha is asked when register via mobile.

## Class Decomposition:

* Breaking down the application into modular classes to handle specific tasks.
* Understanding the importance of modular design for code organization and readability.

## Exception Handling:

* Implementing exception handling to manage errors and unexpected situations in the code.
* Using try-except blocks for graceful error handling.

## Understanding Customer and Seller Roles:

* Implementing different classes and access levels for customers and sellers.

## Product Management for Seller:

* Developing features for adding, updating, searching and deleting products in inventory system of seller.
* Managing product details, including product id, prices and quantities.

## Project Documentation:

* Creating and maintaining project documentation, including doc string, comments, to enhance code readability.

# Task Distribution:

## Muhammad Ahmed Raza CS-23097:

* Ahmed and Yousuf mutually maintain the database and configuration of SQL.
* Registration and Customer classes have been made by Ahmed. Project testing, code flow, additional features (captcha, verification code and many more) and class diagram is mutually maintained.

## Muhammad Yousuf Mateen CS-23135:

* Yousuf and Ahmed mutually maintain the database and configuration of SQL.
* Login and Seller classes have been made by Yousuf. Project testing, code flow , additional features (captcha, verification code and many more) and class diagram is mutually maintained.

# Future Enhancements:

## User Authentication and Authorization:

* Implement a more robust user authentication system, possibly incorporating encryption for storing passwords.

## Graphical User Interface (GUI):

* Develop a graphical user interface using a GUI using Python Frameworks like Django Or Flask. This can provide a more user-friendly experience compared to a command-line interface.

## Product Categories and Filtering:

* Add support for product categories, allowing customes to filter and search for products based on categories.

## Discounts and Promotions:

* Introduce a system for discounts, promotions, or coupon codes that users can apply during checkout.
* Implement logic for handling special events or sales.

## Inventory Management:

* Provide alerts for low-stock items to sellers.

## Data Backup and Recovery:

* Develop a system for regular data backups to prevent data loss. Include mechanisms for data recovery in case of unexpected issues.

## Add Admin Feature:

* Develop a class for admin that admin has complete access for the store , He can cancel sellers’s Registration at any moment.

# Test Cases:

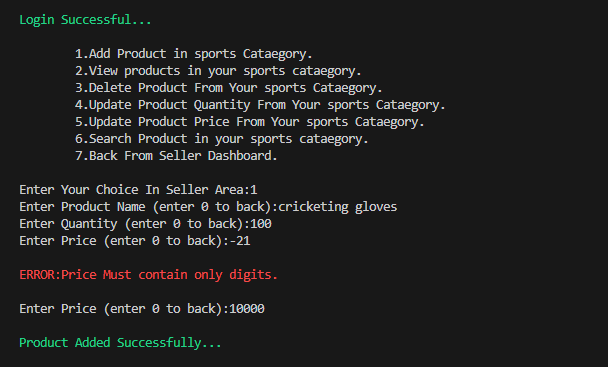


Figure : Seller adding product and its info.

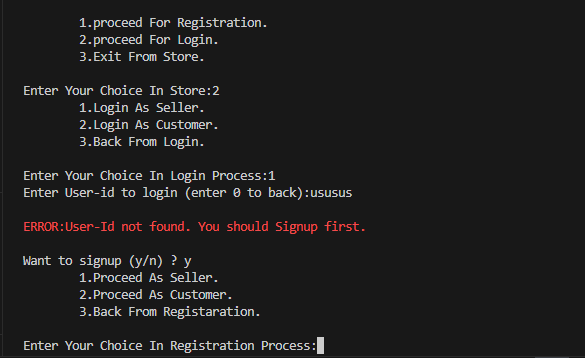


Figure : When user enter user-id during login while not in our database, can go to signup directly if user want.

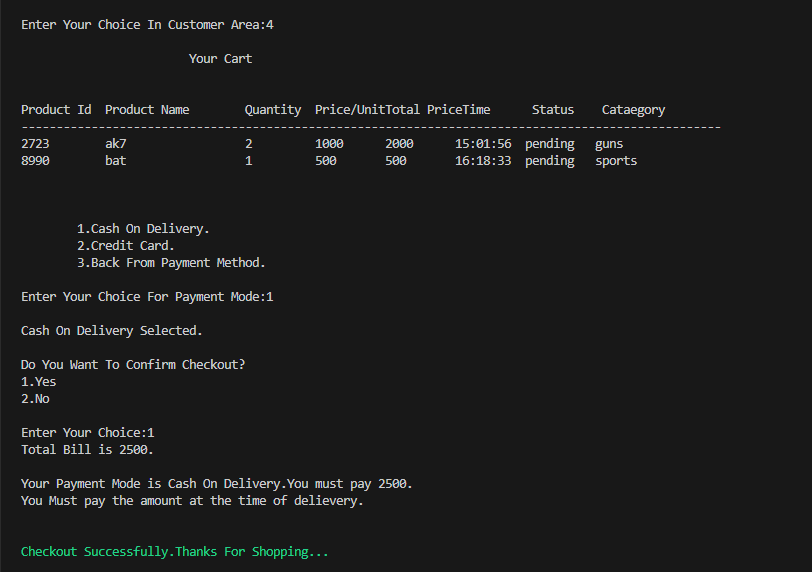
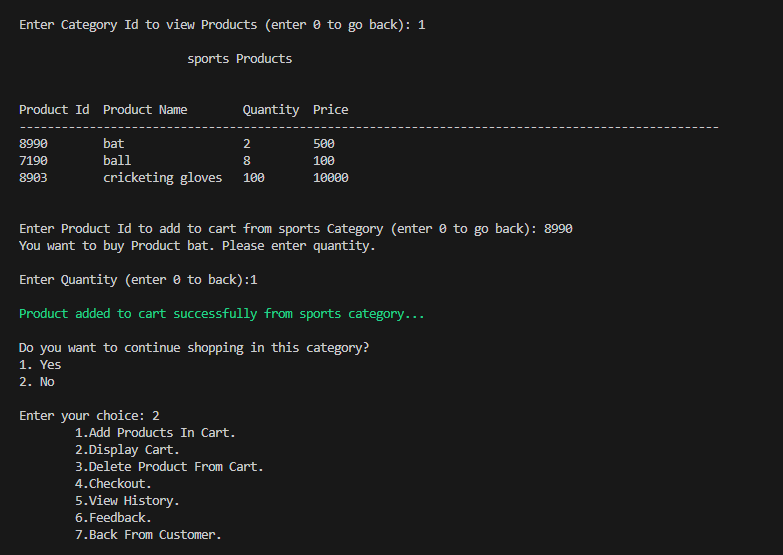
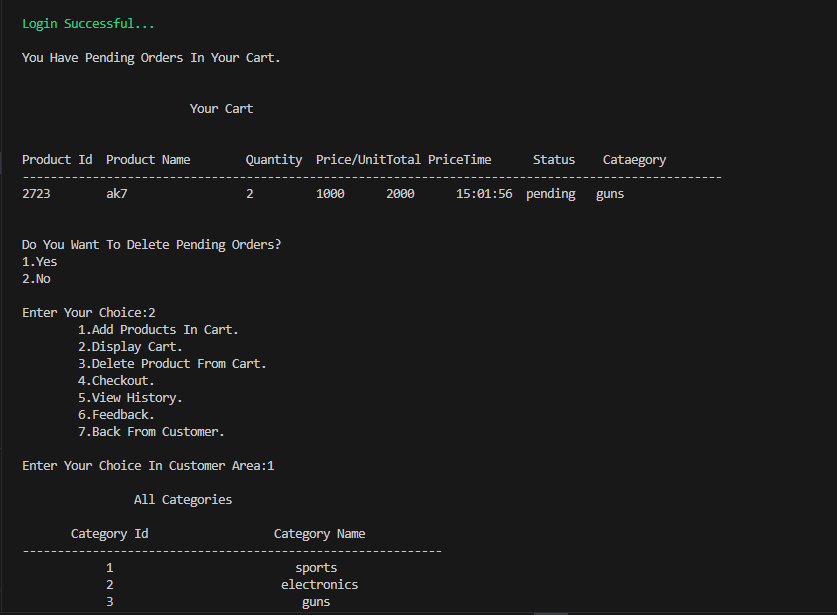


Figure : Above three pics showing that the customer adding product and doing checkout.

# Reference:

Functions and concepts for this project were derived from various sources and tutorials, including online platforms like Stack Overflow, official Python documentation, and educational resources.