**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 ABDULLAH** | **CS-033** |
| S2 | **FATIMA SHAHID** | **CS-007** |
| S3 | **ZOHA SAJID** | **CS-021** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CRITERIA AND SCALES** | | | | **Marks Obtained** | | |
| **S1** | **S2** | **S3** |
| 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. |
| **Criterion 5: Does the report adhere to the given format and requirements?** [4 marks] | | | |  |  |  |
| **1** | **2** | **3** | **4** |
| **The report does not contain the required information and is formatted poorly.** | **The report contains the required information only partially but is formatted well.** | **The report contains all the required information but is formatted poorly.** | **The report contains all the required information and completely adheres to the given format.** |
| **Total Marks:** | | | |  |  |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Teacher’s Signature

Contents

[1.PROBLEM DESCRIPTION: 4](#_Toc156164922)

[2. ABSTRACT: 4](#_Toc156164923)

[3. DISTINGUISHING FEATURES OF OUR PROJECT: 4](#_Toc156164924)

[3.1 User-Friendly Interface: 4](#_Toc156164925)

[3.2 Efficient Item Management 4](#_Toc156164926)

[3.3 Discounts and Promotions 4](#_Toc156164927)

[3.4 Object-Oriented Design: 4](#_Toc156164928)

[3.5 Comprehensive Error Handling: 5](#_Toc156164929)

[3.6 Data Analytics: 5](#_Toc156164930)

[3.7 Dynamic Pricing and Inventory: 5](#_Toc156164931)

[3.8 Demonstration of Python's Versatility: 5](#_Toc156164932)

[3.9 Accessible for All Users: 5](#_Toc156164933)

[3.10 Feedback and Reviews: 5](#_Toc156164934)

[4. FLOW OF THE PROJECT: 6](#_Toc156164935)

[5. MOST CHALLENGING PART OF THE PROJECT: 6](#_Toc156164936)

[5.1 File Syntax Complexity: 6](#_Toc156164937)

[5.2 Creating The Remove Cart Function: 7](#_Toc156164938)

[5.3 Making The Shopping History For Users: 7](#_Toc156164939)

[5.4 Testing And Debugging: 7](#_Toc156164940)

[5.5 Scaling With Users 7](#_Toc156164941)

[6. NEW LEARNINGS FROM THE PROJECT: 7](#_Toc156164942)

[6.1 Functions and Methods: 7](#_Toc156164943)

[6.2 File Handling in Python: 7](#_Toc156164944)

[6.3 Data Structures: 7](#_Toc156164945)

[6.4 Modular Programming: 7](#_Toc156164946)

[7. INDIVIDUAL CONTRIBUTIONS: 7](#_Toc156164947)

[1. Login and Create Account Function & save data function (Zoha ): 8](#_Toc156164948)

[2. View Cart, Add to Cart, and Display Product List Functions (Abdullah): 8](#_Toc156164949)

[3. History, Remove Items from Cart and Load Data Function (Fatima): 8](#_Toc156164950)

[8. FUTURE EXPANSIONS: 8](#_Toc156164951)

[8.1 GUI Development: 8](#_Toc156164952)

[8.2 Database Integration: 8](#_Toc156164953)

[8.3 Product Search and Filtering: 8](#_Toc156164954)

[8.4 Inventory Management: 8](#_Toc156164955)

[8.5 Functions which can help us with Responsive Customer Support: 8](#_Toc156164956)

[8.6 Feature Enhancements: 9](#_Toc156164957)

[9.TEST RUNS: 9](#_Toc156164958)

[9.1 CREATING AN ACCOUNT: 9](#_Toc156164959)

[a) CREATING AN ACCOUNT: 9](#_Toc156164960)

[b) LOGIN INTO EXISTING ACCOUNT: 9](#_Toc156164961)

[9.2 WHAT OUR PROGRAM OFFERS (LIST OF ITEMS): 10](#_Toc156164962)

[9.3 FEATURES AVAILABLE IN THE PROGRAM : 11](#_Toc156164963)

[a) ADDING TO CART: 11](#_Toc156164964)

[b) REMOVING FROM CART: 12](#_Toc156164965)

[c) VIEWING CART: 12](#_Toc156164966)

[d) USER HISTORY: 13](#_Toc156164967)

[9.4 SUCCESSFUL TRANSICTION: 13](#_Toc156164968)

[9.5 ADDING A REVIEW: 14](#_Toc156164969)

# 1.PROBLEM DESCRIPTION:

Ever found yourself wanting a simple way to keep track of the things you want to buy online? Many existing shopping cart programs are either too complex or lack user-friendliness. Our project addresses this issue by creating a straightforward shopping cart program using Python. The goal is to provide an uncomplicated tool for users to easily browse, select, and manage items in their virtual cart, making the online shopping experience more accessible for everyone. The problem we aim to solve is to offer a simple and user-friendly solution for online shoppers who prefer a hassle-free way of organizing their purchases

# 2. ABSTRACT:

The main objective of this project is to develop a comprehensive shopping cart program using Python. The program provides users with a command-in-line interface in which they can browse, select, search, and manage items. A virtual shopping cart where you can pick items, adjust quantities, and smoothly complete your purchases. We used a smart way of organizing things in our program using different lists and dictionaries, making it simple for the user to access their old shopping history and past purchases.

Our project demonstrates how Python can be used by beginners like us to create simple, practical, and user-friendly shopping tools for online stores and much more.

# 3. DISTINGUISHING FEATURES OF OUR PROJECT:

3.1 User-Friendly Interface: Our shopping cart program prioritizes simplicity, ensuring an easy and intuitive experience for users, even those unfamiliar with complex software.

3.2 Efficient Item Management: The program allows users to effortlessly add, remove, and adjust quantities of items in their virtual cart, streamlining the shopping process.

3.3 Discounts and Promotions: This program incorporates features for applying discounts, and promotional codes during the checkout process to incentivize purchases

3.4 Object-Oriented Design: Leveraging Python's object-oriented programming, the system is designed for efficiency, making it scalable and adaptable for future enhancements.

3.5 Comprehensive Error Handling: Robust error-handling mechanisms have been implemented to enhance the program's reliability, ensuring a smooth experience for users even in unexpected scenarios.

3.6 Data Analytics: Our program includes features for collecting and analyzing user data, helping us keep track of different users, their past purchases, account history, and other features

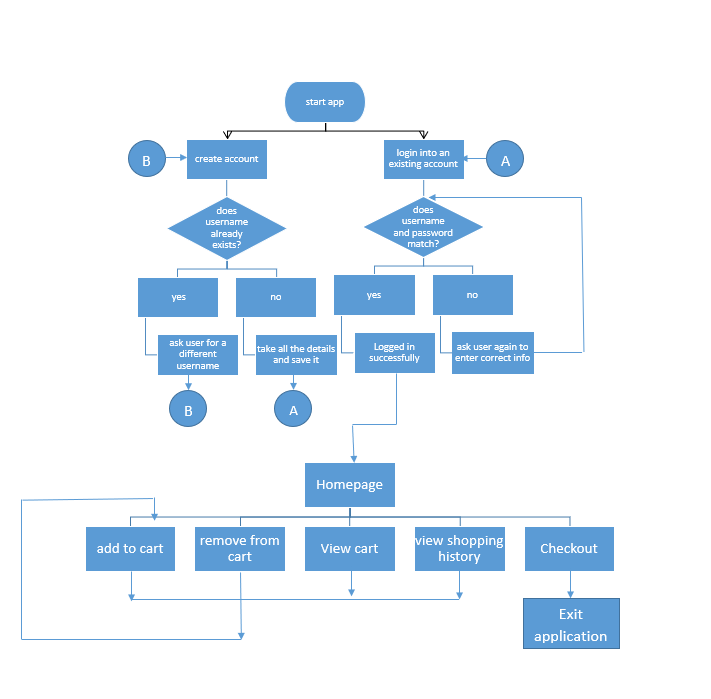
3.7 Dynamic Pricing and Inventory: The system manages product details, pricing, and inventory dynamically, providing accurate and up-to-date information for users as they shop and also updating the cart in runtime.

3.8 Demonstration of Python's Versatility: The project showcases the versatility of Python in creating practical applications, specifically tailored for the e-commerce domain.

3.9 Accessible for All Users: With a focus on simplicity, the shopping cart program is designed to cater to a wide range of users, making online shopping more accessible to everyone.

3.10 Feedback and Reviews: We integrated a system for customers to leave feedback and reviews on products, enhancing the credibility of our platform and helping others make informed decisions.

# 4. FLOW OF THE PROJECT:



# 5. MOST CHALLENGING PART OF THE PROJECT:

## 5.1 File Syntax Complexity:

Understanding the syntax for opening, reading, writing, and closing files in Python was a bit challenging, especially for us in the beginning. It was hard since it involves multiple steps and ensuring proper usage of functions like open(), read(), write(), and close().

## 5.2 Creating The Remove Cart Function:

Creating the "remove from cart" function involved manipulating data structures, for us a dictionary or a list representing the shopping cart. The challenge raised from ensuring that the removal process is accurate and doesn't disrupt the overall structure of the cart.

## 5.3 Making The Shopping History For Users:

Managing the structure of the shopping history data for each user was complex. Saving and retrieving this data effectively, involved dealing with file handling which was a difficult task. Especially associating each shopping history with the correct user required a reliable way to identify users ...., Thus making it essential for us to link all the users and passwords.

## 5.4 Testing And Debugging:

With increased complexity, testing becomes more critical. Ensuring that the shopping history feature, removing and adding items, and promo codes work as intended for various scenarios and handling potential bugs was time-consuming.

5.5 Scaling With Users:

As the number of users increases, managing individual lists or dictionaries for each user becomes more complex. This could impact performance and efficiency if not handled properly, so we tried to handle it as best as we could according to our capabilities

# 6. NEW LEARNINGS FROM THE PROJECT:

## 6.1 Functions and Methods:

We took a deeper dive into functions and methods in Python. Learning about function arguments, return values, and how to create reusable pieces of code.

## 6.2 File Handling in Python:

Got to learn more about reading from and writing to files in Python. Understanding different file formats, such as CSV or JSON, TXT, and how to manipulate data using file operations.

## 6.3 Data Structures:

We had the chance to brush up on basic data structures like lists, dictionaries, and sets in Python which allowed us to know when and how to use each data structure efficiently.

## 6.4 Modular Programming:

We took a deeper look at how to break down our program into smaller, manageable modules, the concept of functions, and how to organize your code for better readability and maintainability.

# 7. INDIVIDUAL CONTRIBUTIONS:

We have done most of our work together by mutual cooperation , but we divided the work of creating basic functions among ourselves and then complied them all together while adding other features and filling all the different features in one big program

Here's a breakdown of each member's individual contributions to the group project:

## 1. Login and Create Account Function & save data function (Zoha ):

- Developed the login and create account functionality, ensuring secure user authentication.

-Created the save data function ,which helped the information to be saved in a proper file.

## 2. View Cart, Add to Cart, and Display Product List Functions (Abdullah):

- Implemented the functionality to view the shopping cart, including a user-friendly display of added items.

- Developed the "Add to Cart" function to enable users to easily include items in their shopping cart.

- Created the product list display function, providing users with a clear overview of available products.

## 3. History, Remove Items from Cart, and Load Data Function (Fatima):

- Developed a history function to track and display user transaction history.

- Implemented the functionality to remove items from the shopping cart, ensuring a seamless user experience.

- Developed the load data function to enable proper functionality of dynamic cart and to make the user’s shopping history viewable.

Overall, each member's contributions were crucial to the success of the project, resulting in a fully functional and user-friendly shopping cart program in Python.

# 8. FUTURE EXPANSIONS:

## 8.1 GUI Development:

We can make our application a graphical user interface (GUI), and learn how to create it using libraries like Tkinter or PyQt. GUIs can enhance the user experience.

## 8.2 Database Integration:

We could Learn how to integrate a database (e.g., SQLite, MySQL, or PostgreSQL) into our program for more efficient data storage and retrieval.

## 8.3 Product Search and Filtering:

We can work on Implementing a robust search and filtering system to help users quickly find the products they're looking for.

## 8.4 Inventory Management:

We could work on a function to help with inventory tracking to prevent overselling items and notify users about low stock or unavailable items.

## 8.5 Functions which can help us with Responsive Customer Support:

We can establish responsive customer support channels by further study of Python and its libraries. This could include live chat, email support, or a helpdesk system to address user queries and concerns promptly.

## 8.6 Feature Enhancements:

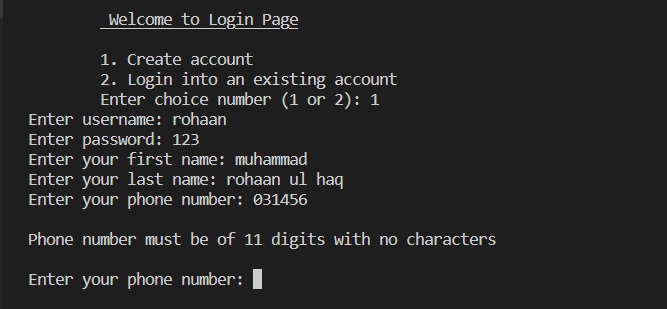
In the future, we can add new features or enhance existing ones based on user feedback and emerging trends in e-commerce. This could include product recommendations, wish lists, or social sharing features.

# 9.TEST RUNS:

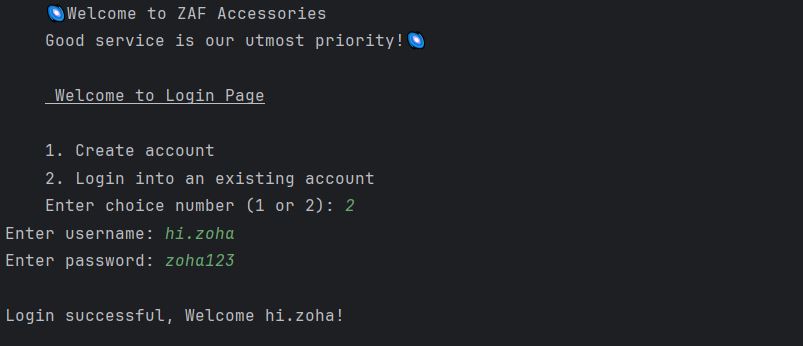
We did a lot of test runs and debugging to make sure all the functions are running properly and that all the history and information is being saved in their respective locations.

## 9.1 CREATING AN ACCOUNT:

### CREATING AN ACCOUNT:



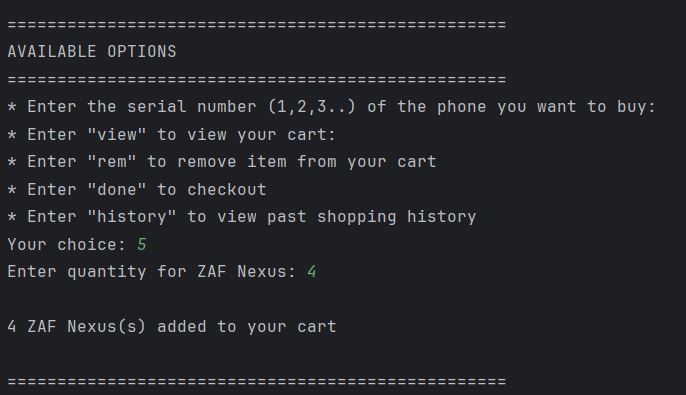
### LOGIN INTO EXISTING ACCOUNT:



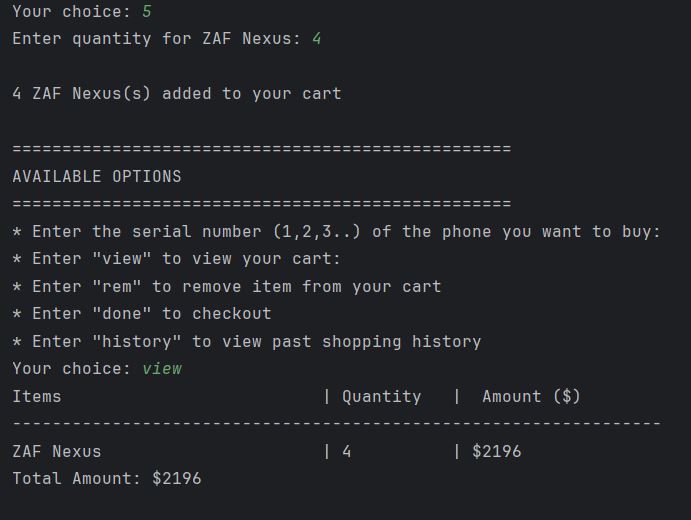
## 9.2 WHAT OUR PROGRAM OFFERS (LIST OF ITEMS):



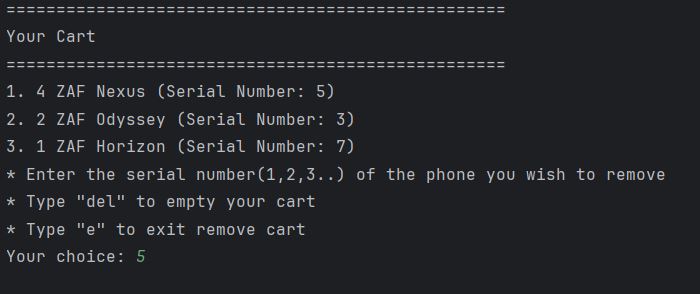
## 9.3 FEATURES AVAILABLE IN THE PROGRAM :

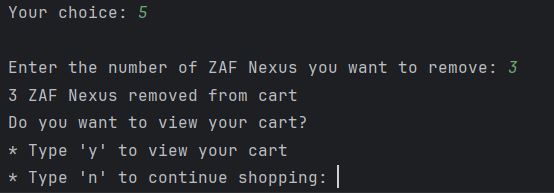


### ADDING TO CART:

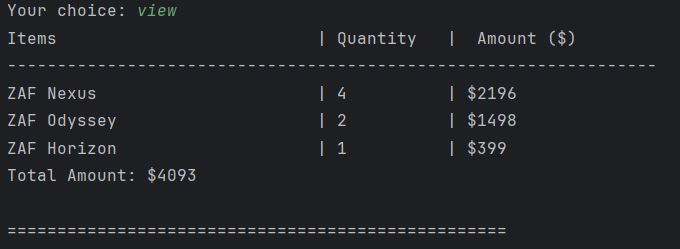


### REMOVING FROM CART:

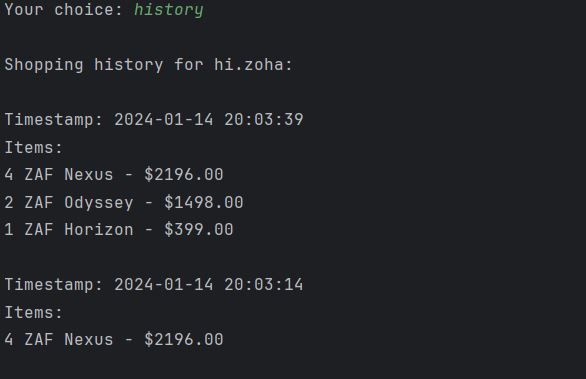




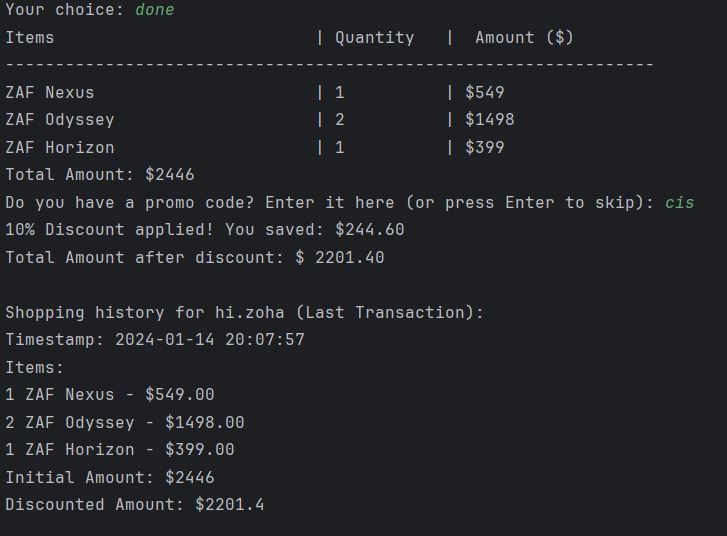
### VIEWING CART:



### USER HISTORY:



## 9.4 SUCCESSFUL TRANSICTION:



## 9.5 ADDING A REVIEW:

