

Contents

[1.PROBLEM DESCRIPTION: 1](#_Toc171354864)

[2. ABSTRACT: 1](#_Toc171354865)

[3. DISTINGUISHING FEATURES OF OUR PROJECT: 2](#_Toc171354866)

[3.1 User-Friendly Interface: 2](#_Toc171354867)

[3.2 Efficient Item Management 2](#_Toc171354868)

[3.3 Discounts and Promotions 2](#_Toc171354869)

[3.4 Object-Oriented Design: 2](#_Toc171354870)

[3.5 Comprehensive Error Exceptions: 2](#_Toc171354871)

[3.6 Data Analytics: 2](#_Toc171354872)

[3.7 Dynamic Pricing and Inventory: 2](#_Toc171354873)

[3.8 Demonstration of Python's Versatility: 3](#_Toc171354874)

[3.9 Designed an Admin panel for Runtime enhancements and updates: 3](#_Toc171354875)

[3.10 Accessible for All Users: 3](#_Toc171354876)

[3.11 Feedback and Reviews: 3](#_Toc171354877)

[4. FLOW OF THE PROJECT: 3](#_Toc171354878)

[5. MOST CHALLENGING PART OF THE PROJECT: 4](#_Toc171354879)

[5.1 File Syntax Complexity: 4](#_Toc171354880)

[5.2 Creating The Admin Panel Class: 4](#_Toc171354881)

[5.3 Making The Shopping History for Users: 4](#_Toc171354882)

[5.4 Testing and Debugging: 4](#_Toc171354883)

[5.5 Scaling with Users 4](#_Toc171354884)

[6. NEW LEARNINGS FROM THE PROJECT: 4](#_Toc171354885)

[6.1 Types of classes and methods: 4](#_Toc171354886)

[6.2 File Handling in Python: 4](#_Toc171354887)

[6.3 Data Structures: 4](#_Toc171354888)

[6.4 Modular Programming: 5](#_Toc171354889)

[7. INDIVIDUAL CONTRIBUTIONS: 5](#_Toc171354890)

[a) Fatima Shahid’s Endeavor(CS-007): 5](#_Toc171354891)

[b) Arfa Tariq’s Effort(CS-013): 5](#_Toc171354892)

[c) Muhammad Abdullah’s Provision(CS-033): 5](#_Toc171354893)

[8. FUTURE EXPANSIONS: 5](#_Toc171354894)

[8.1 GUI Development: 5](#_Toc171354895)

[8.2 Database Integration: 5](#_Toc171354896)

[8.3 Product Search and Filtering: 5](#_Toc171354897)

[8.4 Inventory Management: 5](#_Toc171354898)

[8.5 Functions which can help us with Responsive Customer Support: 6](#_Toc171354899)

[8.6 Feature Enhancements: 6](#_Toc171354900)

[9.TEST RUNS: 6](#_Toc171354901)

[9.1 CREATING AN ACCOUNT: 6](#_Toc171354902)

[a) **CREATING AN ACCOUNT**: 6](#_Toc171354903)

[**b)** **LOGIN INTO EXISTING ACCOUNT:** 7](#_Toc171354904)

[9.2 WHAT OUR PROGRAM OFFERS (LIST OF ITEMS): 7](#_Toc171354905)

[9.3 FEATURES AVAILABLE IN THE PROGRAM: 8](#_Toc171354906)

[**a)** **ADDING TO CART:** 8](#_Toc171354907)

[**b)** **REMOVING FROM CART:** 8](#_Toc171354908)

[**c)** **VIEWING CART:** 9](#_Toc171354909)

[**d)** **USER HISTORY:** 10](#_Toc171354910)

[9.4 SUCCESSFUL TRANSACTION: 10](#_Toc171354911)

[9.5 PROMO CODE: 11](#_Toc171354912)

[9.6 ADDING A REVIEW: 11](#_Toc171354913)

[9.7 FEATURES OF THE ADMIN PANEL: 12](#_Toc171354914)

[**a) Logging in as an Admin** 12](#_Toc171354915)

[**b) Viewing all users** 12](#_Toc171354916)

[**c) Updating stock of existing phones:** 13](#_Toc171354917)

[**d) Adding a new phone to the inventory:** 14](#_Toc171354918)

[**e) Removing an existing phone from the inventory:** 14](#_Toc171354919)

[**f) Viewing the cart with updated products:** 14](#_Toc171354920)

[10) CLASS DIAGRAM: 15](#_Toc171354921)

# 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: We created 9 classes utilizing all concepts of Objected Oriented Programming, and making it work to the best of its abilities, leveraging Python's object-oriented programming, the system is designed for efficiency, making it scalable and adaptable for future enhancements.

3.5 Comprehensive Error Exceptions: Robust error-handling mechanisms have been implemented to enhance the program's reliability, ensuring a smooth experience for users even in unexpected scenarios. We have provided every possible exception to ensure a hassle-free experience of our system, not even allowing the user to type only the ‘Enter key’ in the username.

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 Designed an Admin panel for Runtime enhancements and updates:

With the inclusion of an Admin panel, it allows the admin to login during runtime and update the shopping list or view all the user’s that have ever logged in to the system, however the Admin doesn’t have access to the user’s passwords as they are meant to be strictly confidential.

3.10 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.

3.11 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 Admin Panel Class:

Creating the "Admin" class involved manipulating data structures, in an unorthodox way. The challenge raised from identifying how to access all User’s data and add or remove a product from our list and maintain sufficient stock.

## 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 Types of classes and methods:

We took a deeper dive into types of classes and methods in Python. Learning about static, abstract methods, how to provide implementation for abstract classes and use it to the best of its capabilities 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 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 classes, 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 classes and methods amongst 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:

## a) Fatima Shahid’s Endeavor(CS-007):

Fatima showcased her skills in implementing Abstract class, Method overriding, Different classes like Checkout, RemoveFromCart and Shopping\_history. She was the main hand behind to ensure a smooth experience for our customers. She also keeps track of all transactions.

## b) Arfa Tariq’s Effort(CS-013):

Arfa assisted the project with help of her programming skills, designing different classes like User and Admin. She was also the main hand behind our stock and inventory management, ensuring our products get restocked as soon as they run out for our customers. She also provided exception handling for any and every possible error possibility, ensuring a hassle-free experience and provided Client code.

## c) Muhammad Abdullah’s Provision(CS-033):

Abdullah asserted that our customers like the UI, organizing all the inventory and displaying it in a way that looks aesthetically pleasing. He also designed AddToCart and the Cart class. He also created the report and designed the Class diagram and flowchart of the project.

# 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:

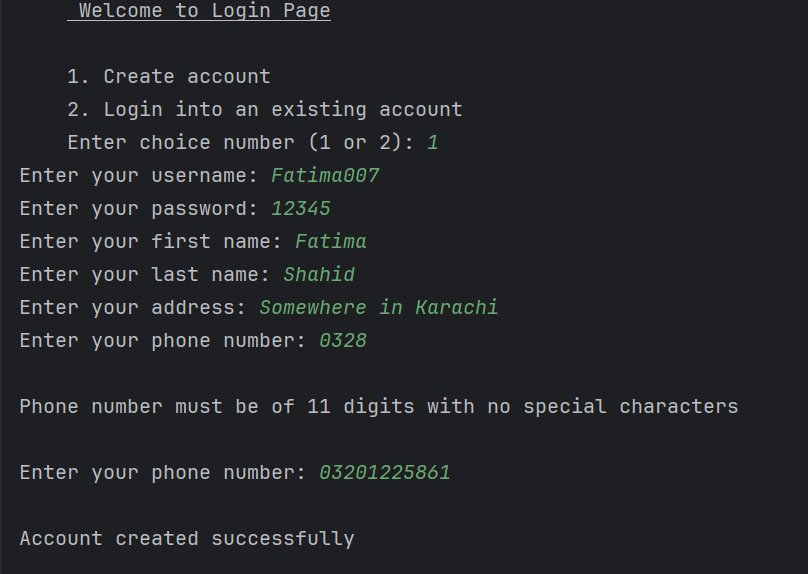
In the future, we can add new features or enhance existing ones based on user feedback and emerging trends in e-commerce. This could include product recommendations, wish lists, or social sharing features.

# 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.

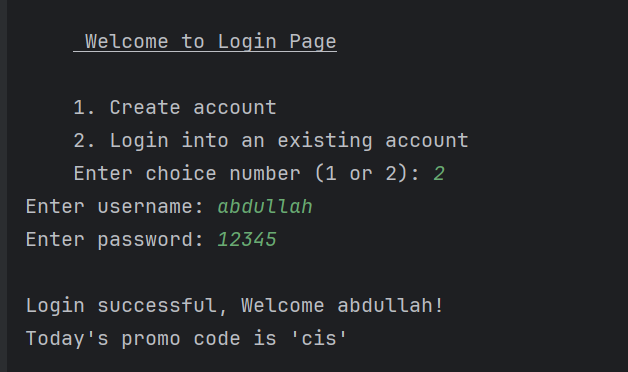
## 9.1 CREATING AN ACCOUNT:

### **CREATING AN ACCOUNT**:



We have set a logical rule that the phone number must contain 11 digits with no special characters or alphabets, taking real life error handling possibilities to a whole new level.

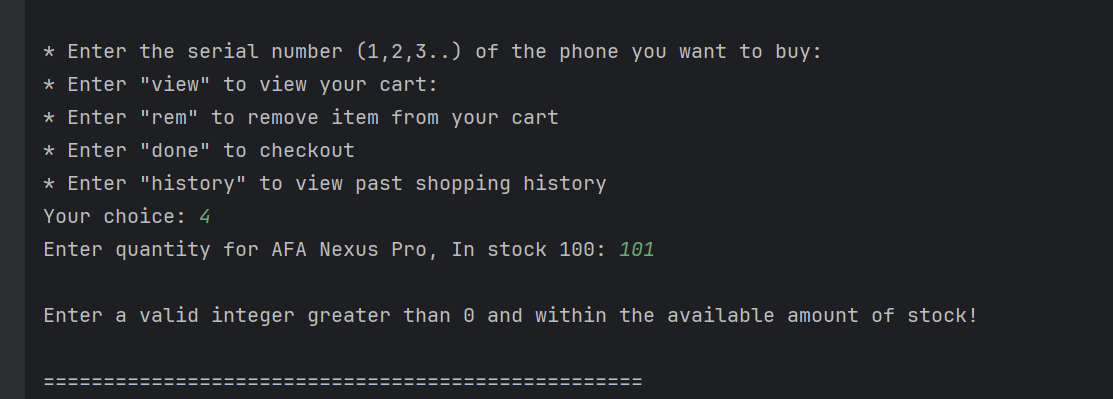
### **LOGIN INTO EXISTING ACCOUNT:**



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

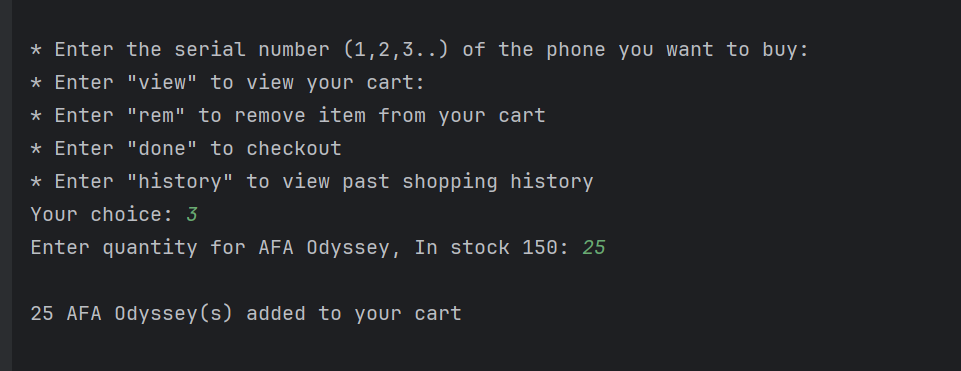


## 9.3 FEATURES AVAILABLE IN THE PROGRAM:

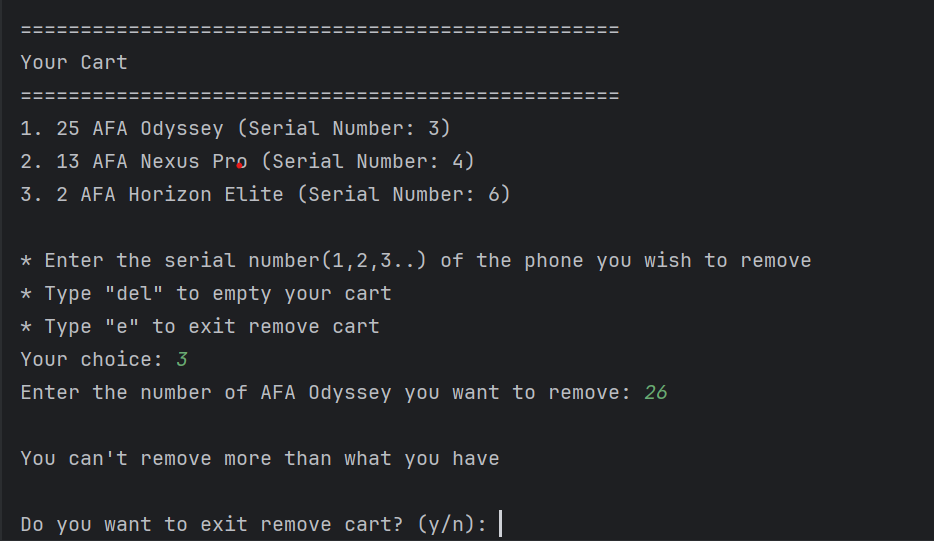


In the Above test case, you can see how a user can’t buy a phone which exceeds our stock limit, depicting real life scenarios.

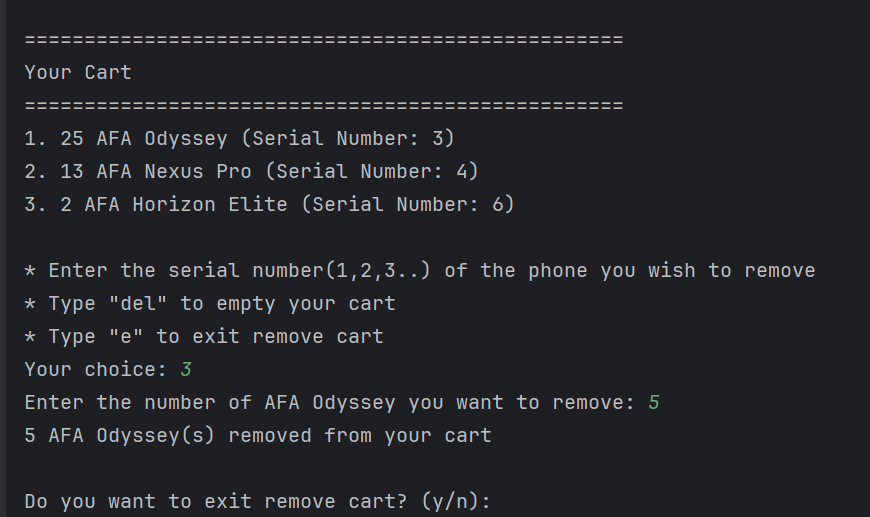
### **ADDING TO CART:**



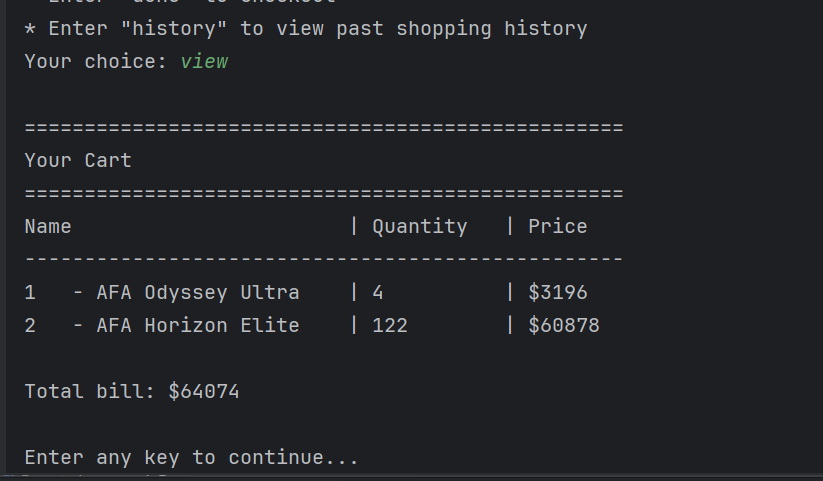
### **REMOVING FROM CART:**



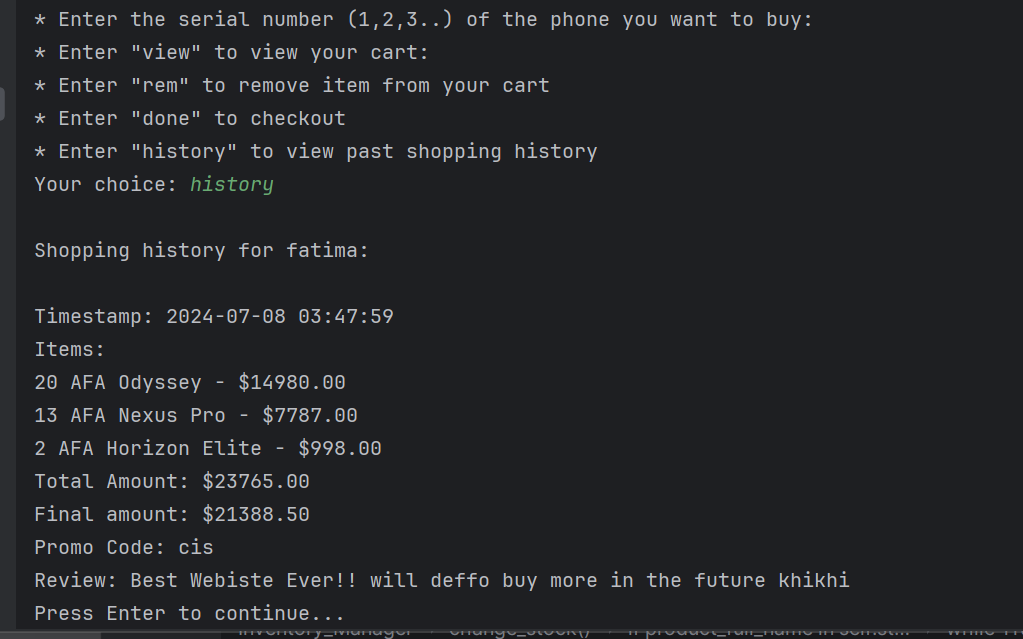
In the Above test case, you can see how our program handles unorthodox user inputs, if the user tries to remove more quantity of a phone than he has in his cart, our program will prompt him with a message stating ‘You cannot remove more than what you have”.



### **VIEWING CART:**

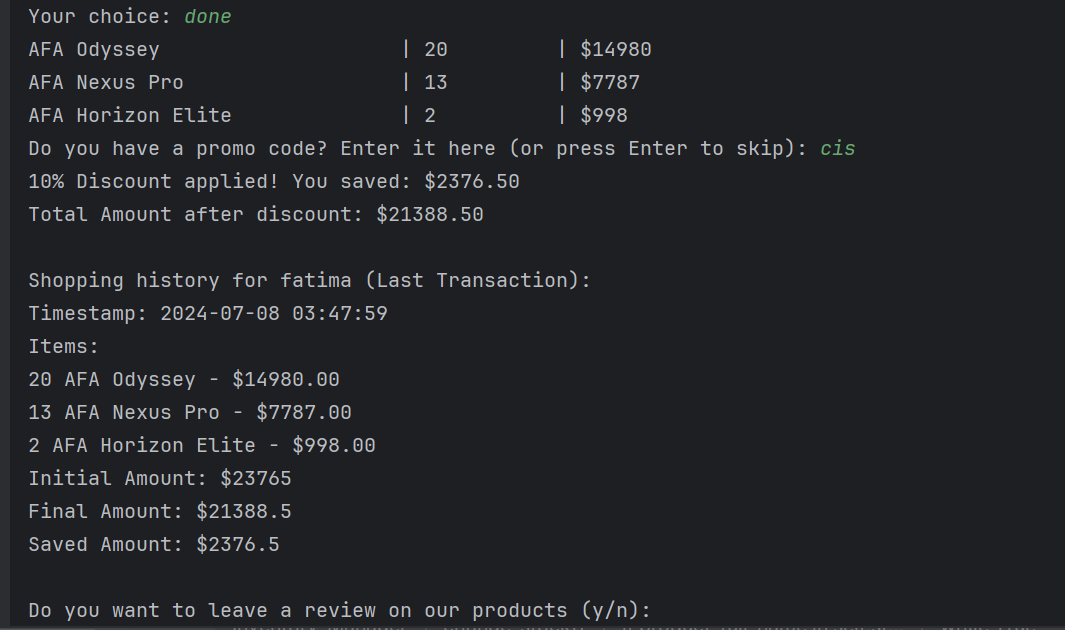


### **USER HISTORY:**

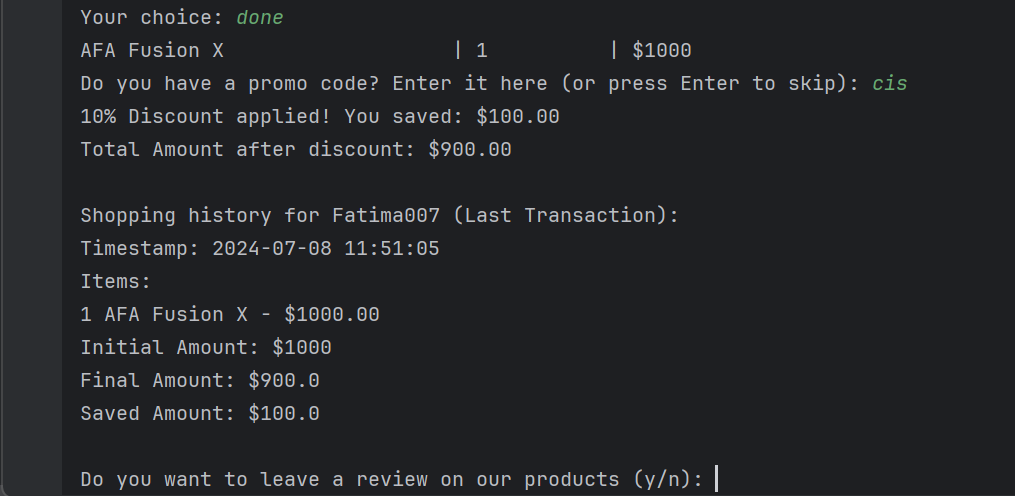


We saved and showed the user’s review in the history as well, to keep track of all the positive and negative feedback which will be useful in making our program even better in the future.

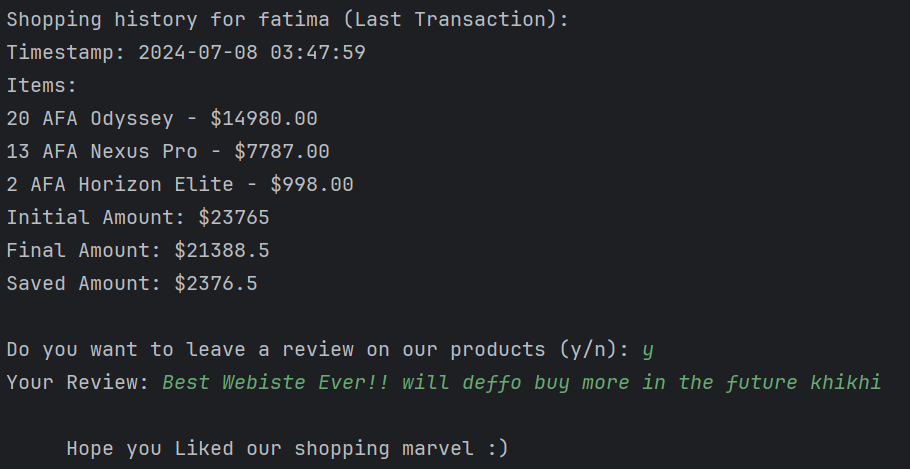
## 9.4 SUCCESSFUL TRANSACTION:



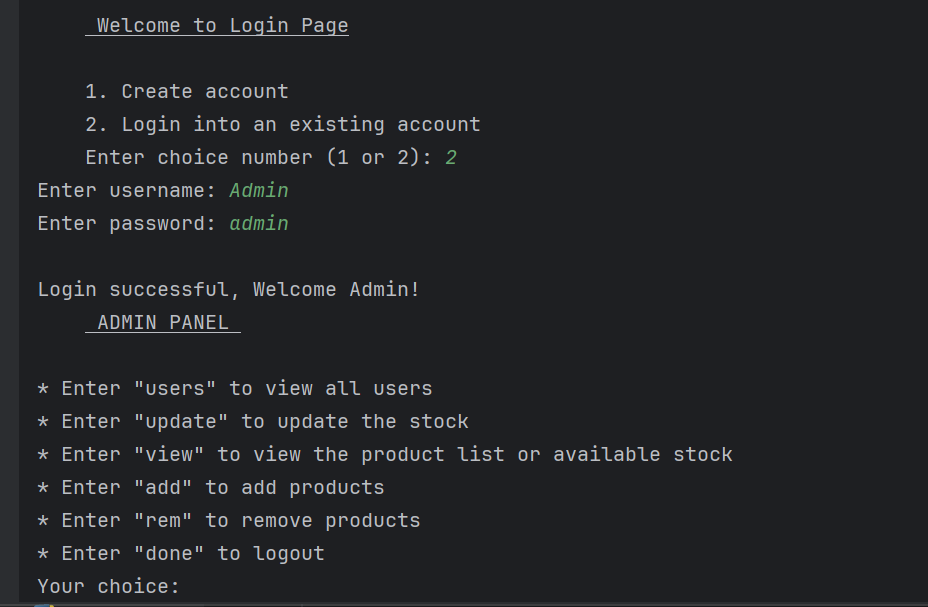
## 9.5 PROMO CODE:



## 9.6 ADDING A REVIEW:



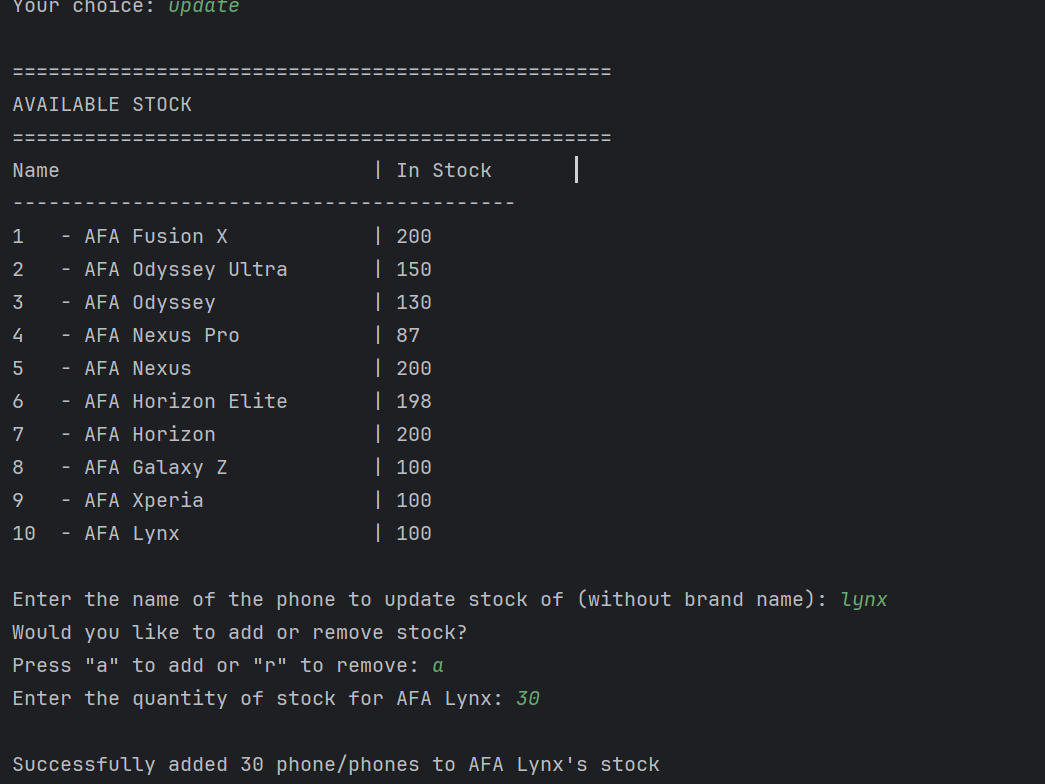
## 9.7 FEATURES OF THE ADMIN PANEL:

**a) Logging in as an Admin:**

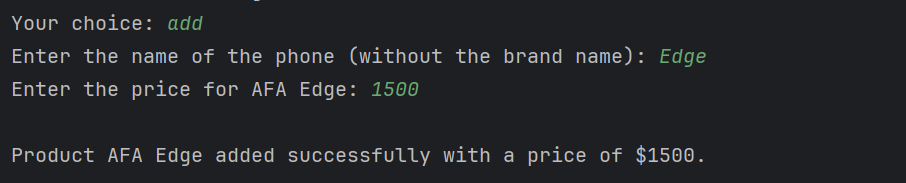
**b) Viewing all users:**

****

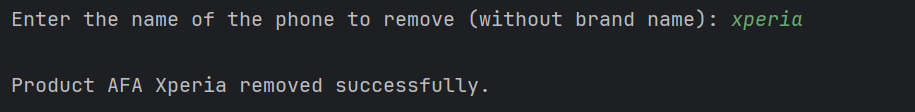
### **c) Updating stock of existing phones:**



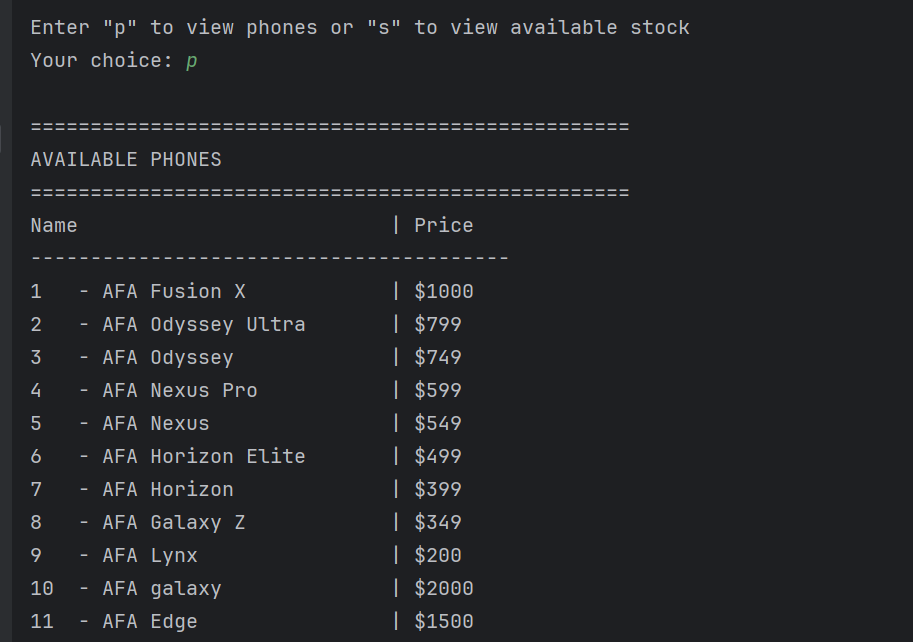
### **d) Adding a new phone to the inventory:**

****

### **e) Removing an existing phone from the inventory:**

****

### **f) Viewing the cart with updated products:**

****

# 10) **CLASS DIAGRAM**:

