DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
BACHELORS IN COMPUTER SYSTEMS ENGINEERING
Course Code: CS-116
Course Title: Object-Oriented Programming

Complex Engineering Problem

FE Batch 2023, Spring Semester 2024
Grading Rubric

TERM PROJECT

Group Members:

Student No.	Name	Roll No.
S1	Muhammad Anas	CS-094
S2	Syed Hamza	CS-093
S3	Iqra Abid	CS-119

CRITERIA AND SCALES				**Marks Obtained**		
				S1	**S2**	**S3**
Criterion 1: Does the class diagram meet the desired specifications and produce the desired outputs? (CPA-1, CPA-3) **[4 marks]**						
1	2	3	4			
The class diagram does not meet the desired specifications and is producing incorrect outputs.	The class diagram partially meets the desired specifications and is producing incorrect or partially correct outputs.	The class diagram meets the desired specifications but is producing incorrect or partially correct outputs.	The class diagram meets all the desired specifications and is producing correct outputs.			
Criterion 2: Does the application meet the desired specifications and produce the desired outputs? (CPA-1, CPA-3) **[6 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 3: 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 4: 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 5: 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 6: Does the report adhere to the given format and requirements? **[2 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

Term Project Title:

ONLINE SHOPPING CART

Problem Description:

An online shopping cart is a virtual shopping trolley, where shoppers can put all of their want-to-buy products in, review to make adjustments in quantity, product attributes, etc., and remove it before or during the checkout if they change their mind. The application also allows users to view the history of their past purchases.

Problem Statement:

Develop a software application in Python using the basic concepts and structures of object-oriented programming. Your application must fulfill the following design constraints:

- Organize the application into at least 6 interlinked user-defined classes.
- Your application must exhibit at least 5 of the following object-oriented features:
- Inheritance
- Association
- Method overriding
- Operator overloading
- Abstract classes
- Exception handling

DISTINGUISHING FEATURES OF THE PROJECT

There are various features of our project. We will discuss every feature in detail. The details are as follows:

Project Overview

Our Online Shopping Cart is a Graphical User Interface (GUI) application designed to streamline the shopping experience for users. Its primary purpose is to facilitate online purchases by providing an intuitive interface for selecting and managing items in a virtual shopping cart.

User Authentication:

To enhance security, the application may require user login or registration. This ensures that only authorized users can access their cart and make purchases.

Product Selection:

Users can browse through a catalog of products, view details, and add items to their cart. The application displays product images, prices, and availability.

Cart Management:

Once items are added to the cart, users can review their selections, adjust quantities, and remove items if needed. The cart keeps track of the total cost.

Checkout Process:

When ready to complete the purchase, users proceed to the checkout.

Order History:

The system maintains a record of past orders, allowing users to track their purchase history and reorder items easily.

GUI Design:

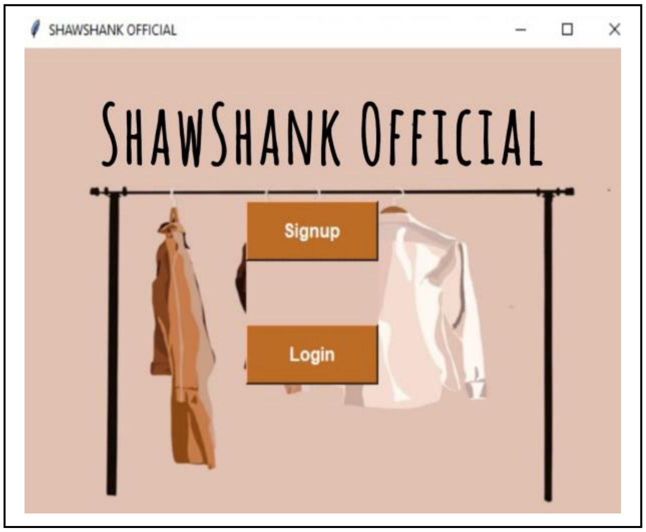
The GUI features an aesthetically pleasing layout with clear navigation elements. It includes buttons for adding/removing items, updating quantities, and proceeding to checkout. Visual cues (such as icons and color-coded buttons) enhance usability.

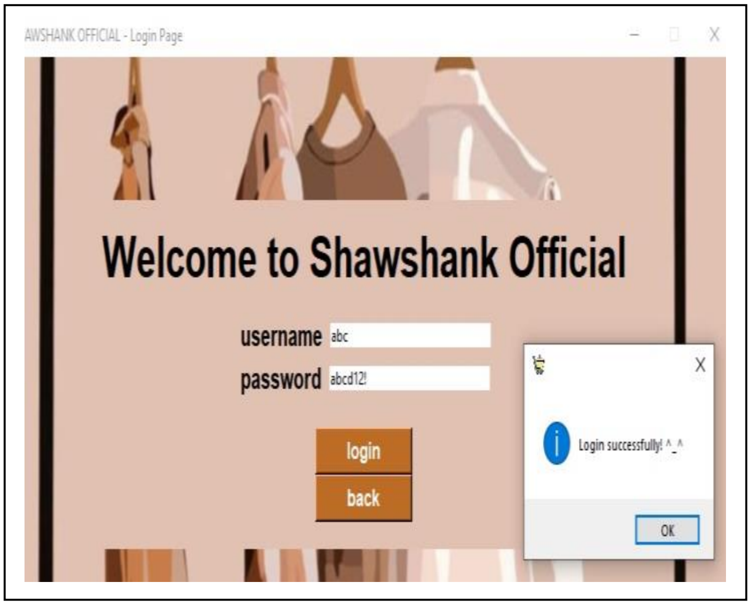
FUNCTIONALITY WITH CLASSES:

The program consists of six classes which increase the modularity of code.

1. Start class:

- The start class inherits from the Frame class.
- The initializer fetches the background image and deals with the geometry of the background image.
- The DisplayWelcomeMessage function asks for login or signup from the user.
- The resize_image function resizes the window.
- Next up the login_page function shows the login page where it asks the user for a username and password.
- It contains a sign_up function which deals with the registration of a new user.





username abc

password abcd12!

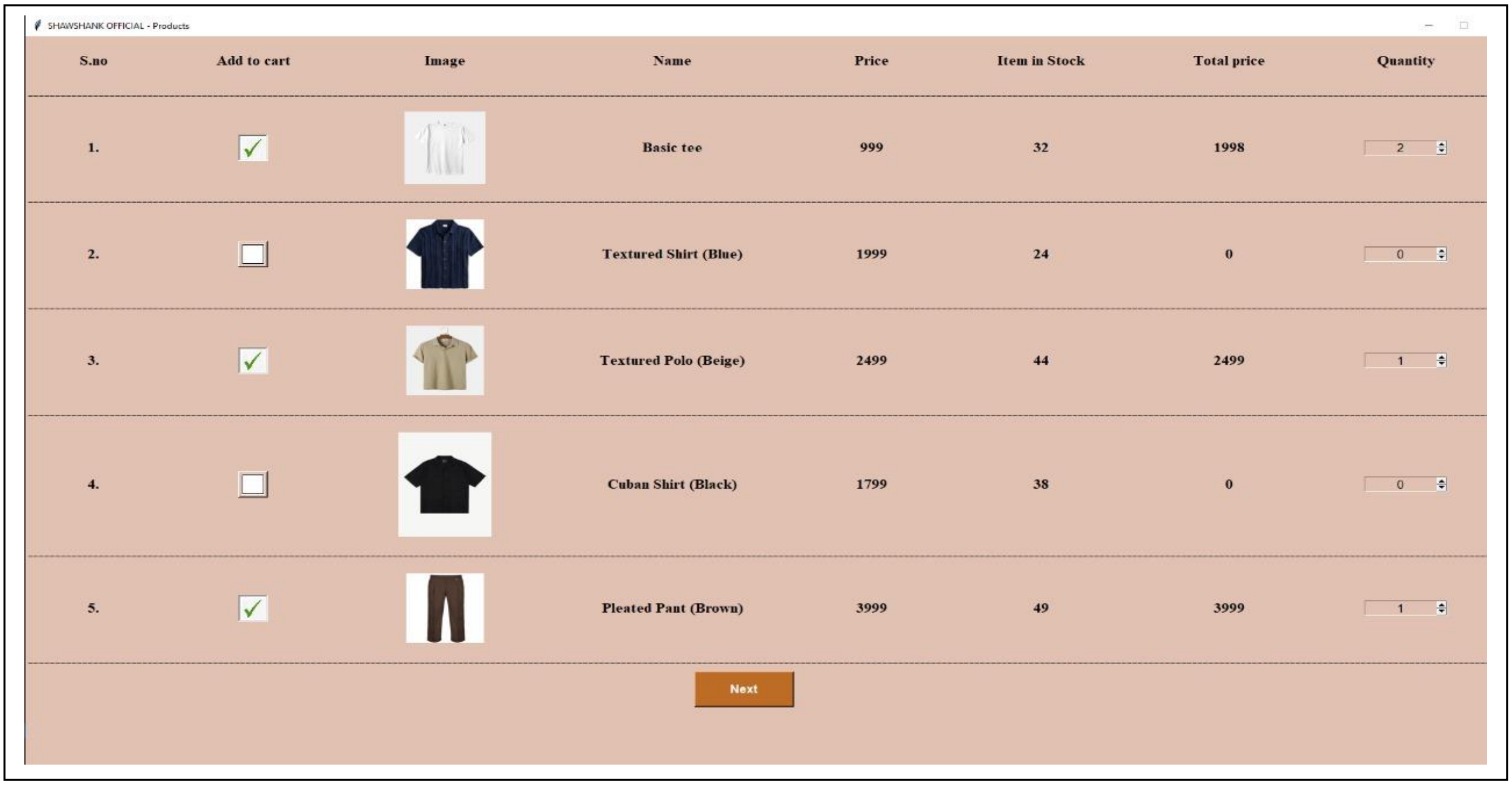
Login successfully! ^_^

2. User class:

- This class deals with the saving of the signup information of the user in a text file.
- If the user logs in, it will fetch his/her information from the text file.

3. Product Class:

- This class displays all the products to a user with the image, price and stock availability.
- Users can add the products to their cart in the quantity they want, using an increment or decrement button.

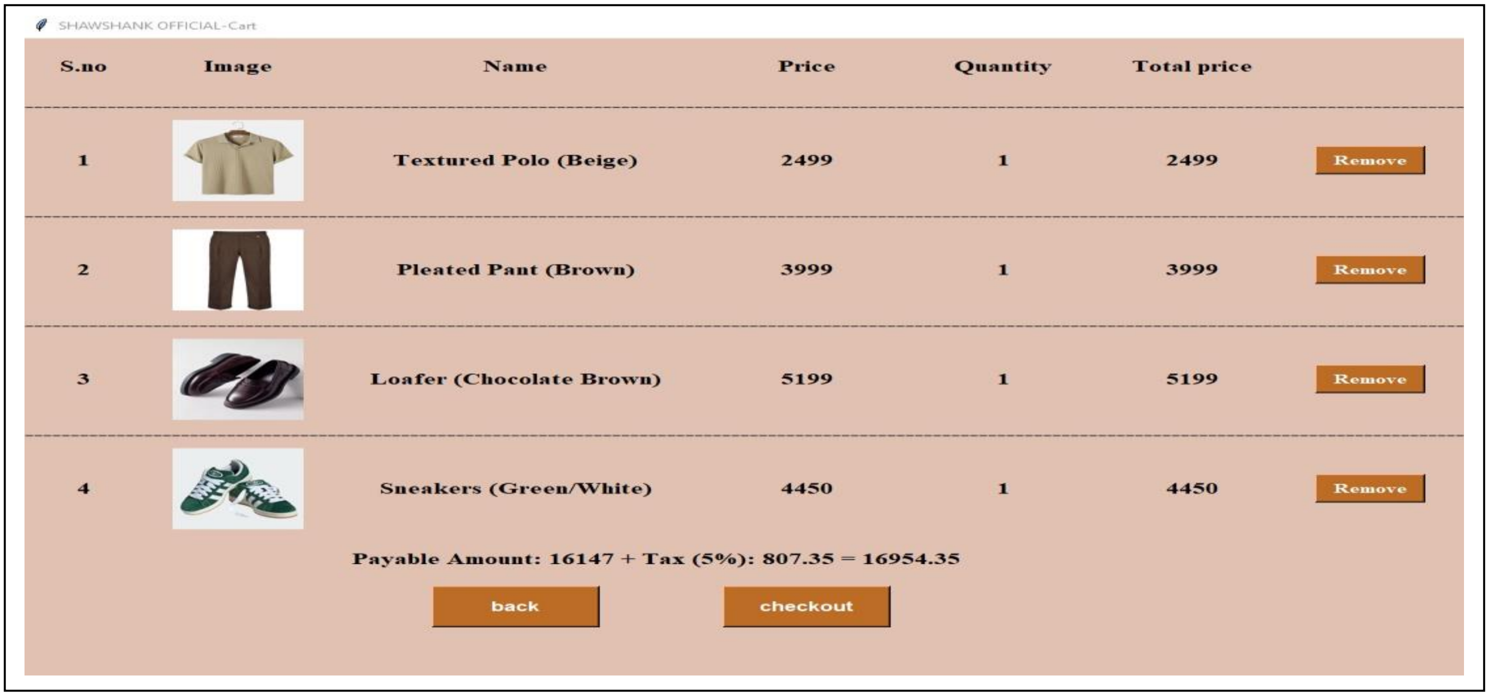


SHAWSHANK OFFICIAL - Products

S.no	Add to cart	Image	Name	Price	Item in Stock	Total price	Quantity
1.	☑		Basic tee	999	32	1998	2
2.	□		Textured Shirt (Blue)	1999	24	0	0
3.	☑		Textured Polo (Beige)	2499	44	2499	1
4.	□		Cuban Shirt (Black)	1799	38	0	0
5.	☑		Pleated Pant (Brown)	3999	49	3999	1

4. Cart class:

- This class allows user to remove a product from their cart.
- It calculates the total bill with tax and displays it to the user.
- The cart history is then saved to a text file.



SHAWSHANK OFFICIAL-Cart

S.no	Image	Name	Price	Quantity	Total price	
1		Textured Polo (Beige)	2499	1	2499	Remove
2		Pleated Pant (Brown)	3999	1	3999	Remove
3		Loafer (Chocolate Brown)	5199	1	5199	Remove
4		Sneakers (Green/White)	4450	1	4450	Remove

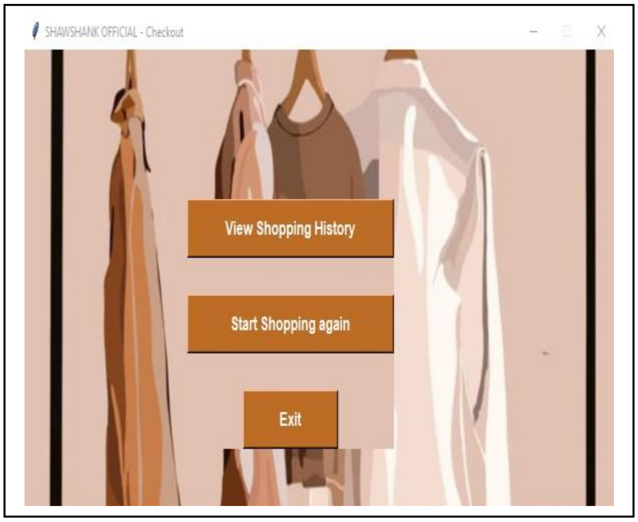
Payable Amount: 16147 + Tax (5%): 807.35 = 16954.35

5. **Checkout class:**

- The checkout class asks the user if he/she wants to view their shopping history.
- Or if they want to start shopping again.
- And finally, exit.
- The exit button terminates the program.

6. **Person Class(Abstract class):**

- It contains the abstract methods which we override in another classes.



OVERALL FLOW OF THE PROGRAM

1. Main Page

- The application starts by displaying the main page with options to 'Login' or 'Signup'.

2. User Authentication

- Login:

- The user selects the 'login' option.

- The application prompts the user to enter their username and password.

- After successful authentication, the user is directed to the product list page.

- Signup:

- The user selects the 'signup' option.

- The application prompts the user to enter necessary information and credentials.

- After successful registration, the user is redirected to the product list page.

3. Product List Page

- The application displays a list of available products with pictures and stock availability information.

- Each product has options to:

- Add to Cart:

Clicking a check button adds the product to the shopping cart.

- Increase Quantity:

An increment button allows the user to increase the quantity of the selected product.

4. Cart Management

- The user clicks the ‘Next’ button after selecting products.

- The application displays the shopping cart with the selected products and their quantities.

- The user has options to:

- Remove Products:

The user can remove any product from the cart if desired.

- View Bill:

The application displays the total bill for the items in the cart.

5. Checkout Options

- The user clicks the ‘Next’ button to proceed to the final step.

- The application presents three options:

- View Shopping History:

The user can view their shopping history.

- Start Shopping Again:

The user can return to the product list page to begin a new shopping session.

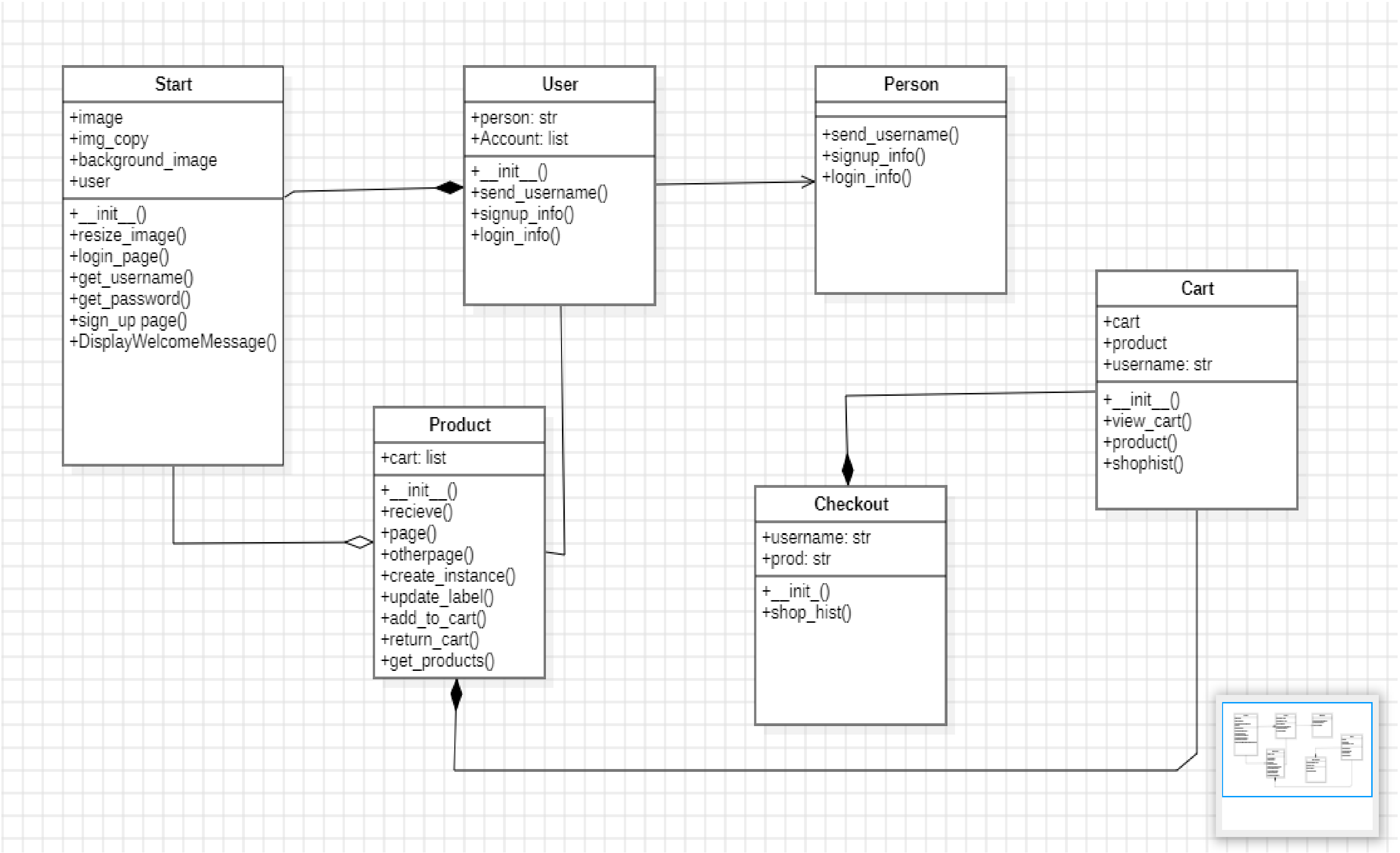
- Exit:

This option terminates the application.

6. Program Termination

- Selecting the Exit option ends the application, completing the shopping session.

CLASS DIAGRAM



MOST CHALLENGING PART

One of the most significant challenges we encountered during the development of our online shopping cart application was implementing and maintaining the graphical user interface (GUI).

- Labels for Multiple Products: Managing labels for multiple products was particularly challenging. We added 10 products to the application, each requiring individual labels to display product information. Coordinating these labels to ensure they were correctly positioned and updated as needed was a complex task.
- Increment Buttons: Adding increment buttons for each product to adjust quantities required detailed event handling and state management. Ensuring that each button correctly updated the corresponding product quantity without conflicts was challenging.
- Integration with Program Logic: The most significant challenge was integrating the GUI with the overall program logic. Ensuring that the GUI accurately reflected the underlying data and provided a responsive user experience.
- Maintaining Consistency: Maintaining consistency across different GUI elements while ensuring that the application remained user-friendly and intuitive was a continuous challenge. Each change in the GUI needed to be carefully coordinated with the program's functionality to avoid introducing bugs or inconsistencies.

NEW LEARNINGS

During the development of our online shopping cart application, we gained several new skills and insights related to Object-Oriented Programming and GUI development.

GUI Development

- New GUI Features: We explored various advanced GUI features, enhancing the user experience of the application.
- Labels and Grids: We learned how to effectively use labels for displaying text and grids for organizing the layout of the interface, which significantly improved the visual structure and usability of the application.
- Increment Button: Implementing an increment button was a valuable learning experience, as it required understanding how to capture and handle user input to dynamically adjust product quantities in the shopping cart.

Object-Oriented Programming

- Maintaining Code in Classes: Throughout the project, we deepened our understanding of maintaining and organizing code within classes. This included:
  - Encapsulation: Keeping data and methods that operate on the data within the same class, enhancing modularity and readability.
  - Inheritance: Utilizing inheritance to create a hierarchy of classes, promoting code reusability and reducing redundancy.

FUTURE EXPANSIONS

Future Expansion Opportunities are:

1. Product Categories

- Description: Organize products into categories for easier navigation and a more streamlined shopping experience.
- Implementation: Create category labels and group products accordingly. Update the GUI to include a dropdown menu or side panel for category selection.

2. Search and Filters

- Description: Implement a search bar and filter options (e.g., by price, brand, rating) to help users find products quickly.
- Implementation: Add a search input field to the GUI and implement filter options. Use basic algorithms to filter the product list based on user input.

3. Save for Later

- Description: Add a feature that allows users to save products for later, enabling them to return to their selections in future sessions.
- Implementation: Add a "Save for Later" button next to each product. Store the saved items in a separate list that users can access later.

4. Wish list

- Description: Implement a wishlist feature where users can save products they are interested in purchasing in the future.
- Implementation: Add an "Add to Wishlist" button for each product. Create a wishlist page where users can view and manage their saved items.

5. Discount Codes and Promotions

- Description: Allow users to apply discount codes or take advantage of promotions during checkout.
- Implementation: Add an input field for discount codes in the checkout process. Implement basic validation and apply discounts to the total bill if the code is valid.

INDIVIDUAL CONTRIBUTION

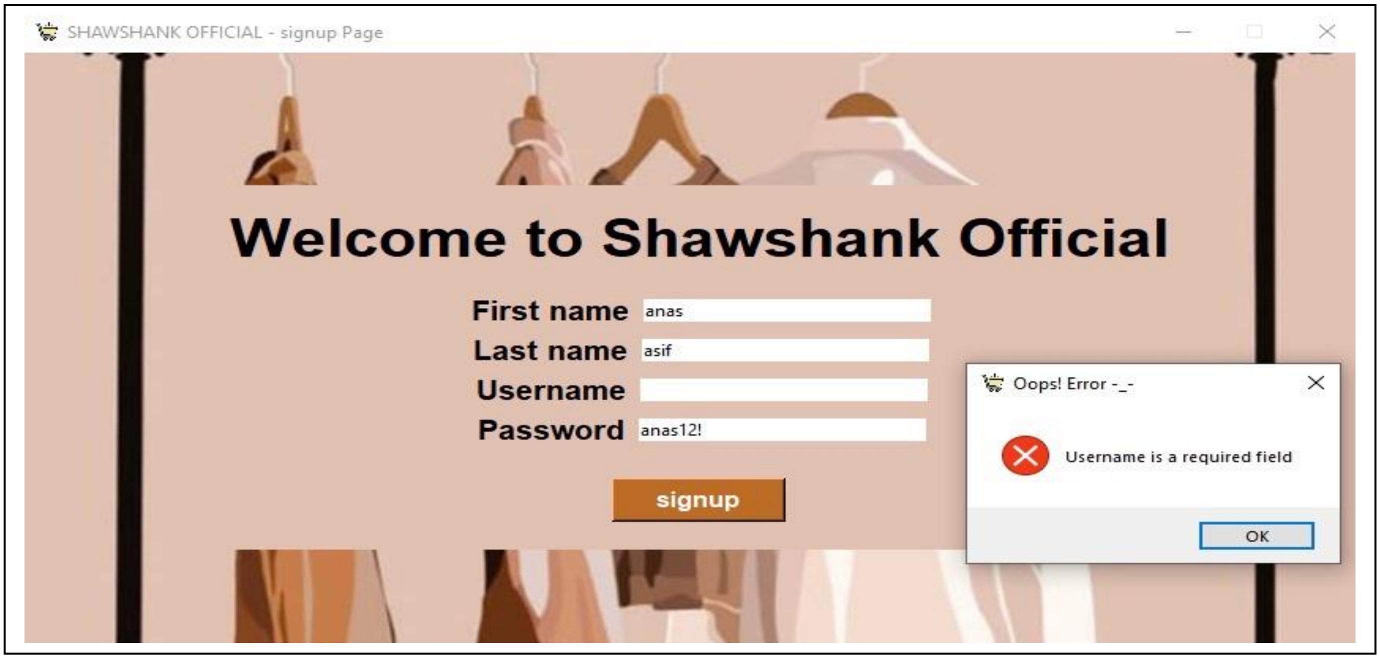
Muhammad Anas – Class start, class cart, and class checkout.
Syed Hamza – Class user and class person.
Iqra Abid – Class product.

LIST OF REFERENCES

- YouTube Tutorials.
- ChatGPT by OpenAI.
- Class Notes.
- Different websites.

TEST CASE RUNS

Test Case 1: User registration



SHAWSHANK OFFICIAL - signup Page

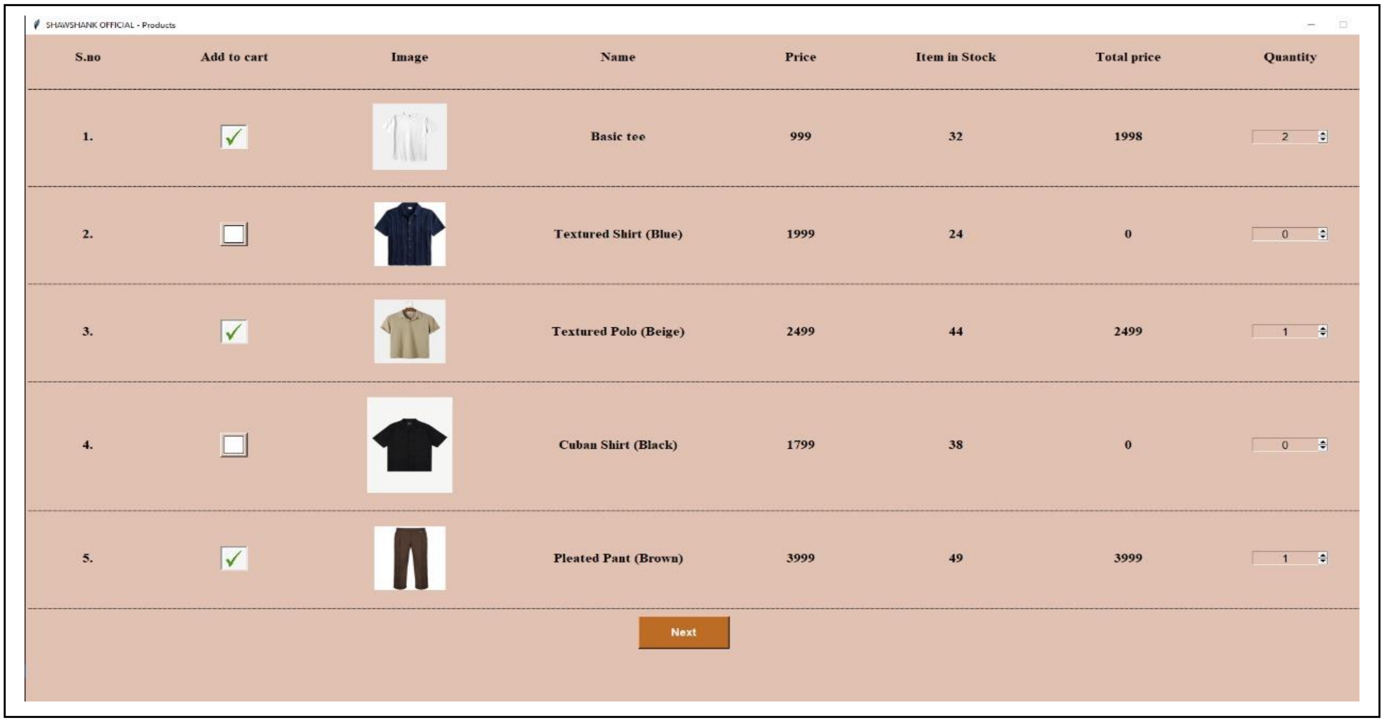
Welcome to Shawshank Official

First name: anas
Last name: asif
Username:
Password: anas12!

Oops! Error -_-

Username is a required field

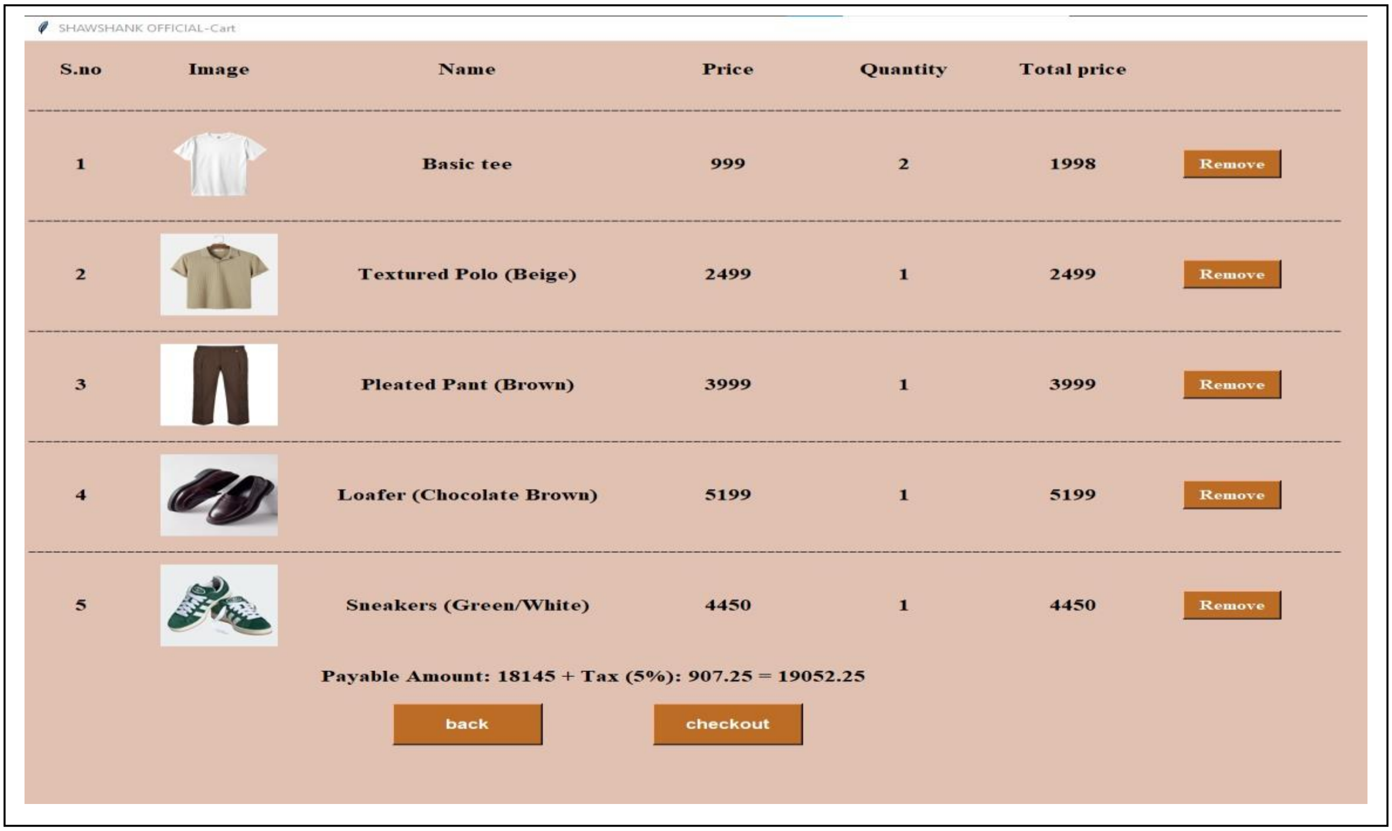
Test Case 2: Adding products to cart



SHAWSHANK OFFICIAL - Products

S.no	Add to cart	Image	Name	Price	Item in Stock	Total price	Quantity
1.	☑		Basic tee	999	32	1998	2
2.	☐		Textured Shirt (Blue)	1999	24	0	0
3.	☑		Textured Polo (Beige)	2499	44	2499	1
4.	☐		Cuban Shirt (Black)	1799	38	0	0
5.	☑		Pleated Pant (Brown)	3999	49	3999	1

Test Case 3: View Cart and checkout

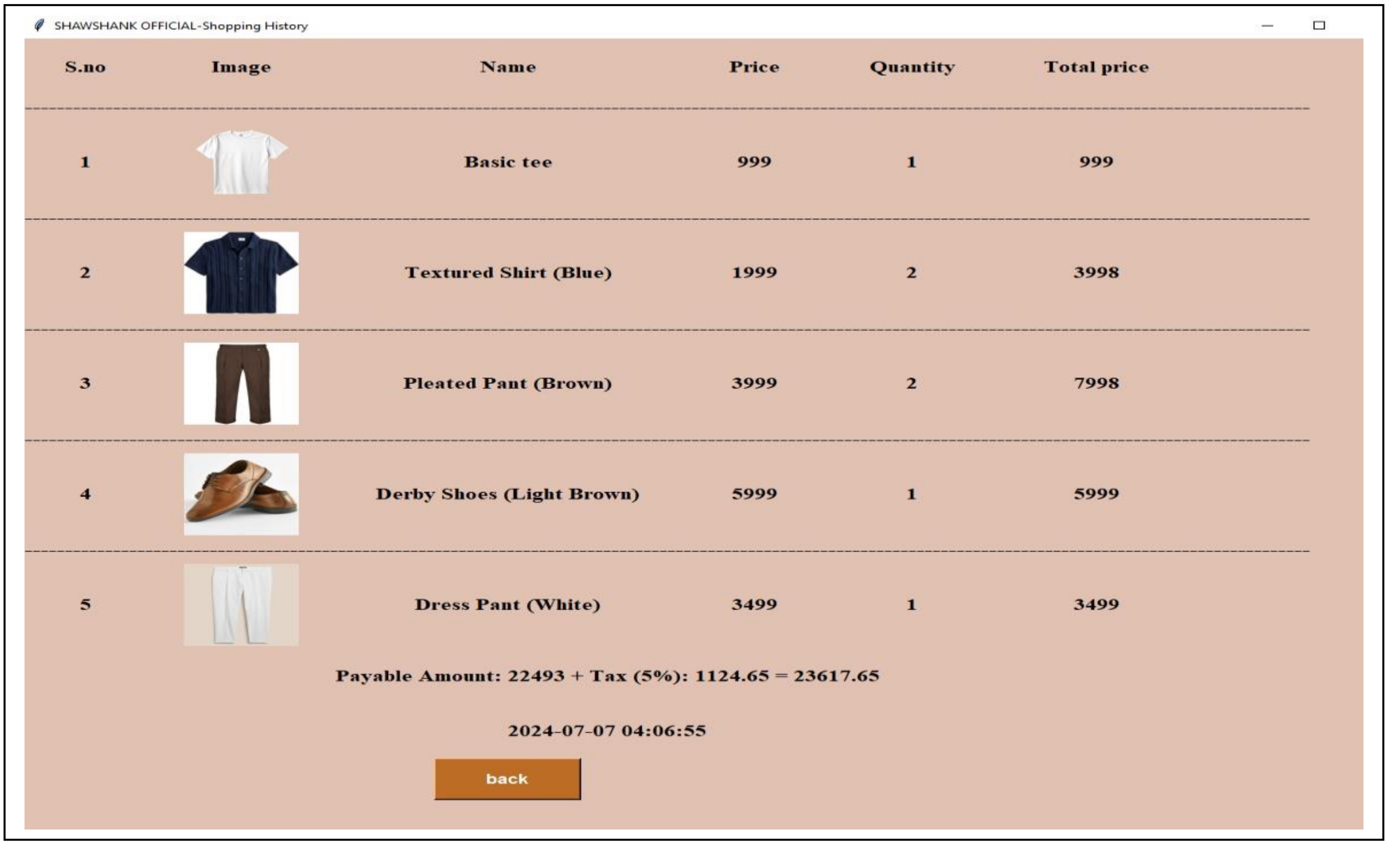


SHAWSHANK OFFICIAL-Cart

S.no	Image	Name	Price	Quantity	Total price	
1		Basic tee	999	2	1998	Remove
2		Textured Polo (Beige)	2499	1	2499	Remove
3		Pleated Pant (Brown)	3999	1	3999	Remove
4		Loafer (Chocolate Brown)	5199	1	5199	Remove
5		Sneakers (Green/White)	4450	1	4450	Remove

Payable Amount: 18145 + Tax (5%): 907.25 = 19052.25

Shopping History



SHAWSHANK OFFICIAL-Shopping History

S.no	Image	Name	Price	Quantity	Total price
1		Basic tee	999	1	999
2		Textured Shirt (Blue)	1999	2	3998
3		Pleated Pant (Brown)	3999	2	7998
4		Derby Shoes (Light Brown)	5999	1	5999
5		Dress Pant (White)	3499	1	3499

Payable Amount: 22493 + Tax (5%): 1124.65 = 23617.65

2024-07-07 04:06:55