WEB BASED CLOTHS ORDERING SYSTEM FOR LITE FASHION STORE

REPORT

ASSIGNMENT 03

Version 1.0

Prepared by

Dark Devils

KGDME Jayarathna - E2240438

GSS Lakshan - E2240402

PVMC Wickramasinghe - E2240158

Created: 05.11.2024

Abstract

The Online Clothing Ordering System (OCOS) is designed to digitize the traditional manual processes involved in clothing retail by implementing a comprehensive online platform. This system aims to streamline the shopping experience, allowing users to browse a wide selection of clothing items, view detailed product information, and place orders easily. By automating order processing, inventory management, and customer records, OCOS reduces the need for manual handling, enabling businesses to minimize labor costs while increasing efficiency. The system provides an intuitive, user-friendly interface that allows customers to explore collections, select sizes, and complete purchases securely, with options for online or cash-on-delivery payment methods.

OCOS prioritizes data security by assigning each customer a unique account with a secure login, ensuring personal information is kept confidential. Additionally, the platform includes easy-to-use tools for staff to manage products, track orders, and update inventory, resulting in a responsive, adaptable, and scalable solution for clothing retailers. This automated approach eliminates data redundancy, reduces errors, and improves the overall user experience, making it easier for businesses to expand their reach and compete in today's fast-paced digital market. In a time where customers value convenience, OCOS offers a seamless shopping experience, with visibility into available products, pricing, and efficient order management, all aimed at better performance and customer satisfaction.

Keywords – customer, employees, clothing, inventory, orders

Table of Contents

List of Tables

List of Figures

Chapter 01: Introduction

1.1 Overview

OCSS is an all-in-one solution to give both online customers and store owners the need in abundance of today's modern fashion retail. It is a complete platform with a good bundle of what it has to offer like user sign up, secure authentication, product browsing, cart management, order processing and real time tracking of the order. Listed as e-commerce and in-store Point of Sale (POS) system, it's a system that caters for both online and in store customers combined to handle physical transactions and inventory.

It believes that the OCSS provides an opportunity for customers to browse through collections of clothing in categories, to make secure purchases and to follow the progress of their orders from placement until delivery. For a store owner, it has tools for inventory management, sales monitoring and report generation, which helps them learn about their performance in sales on a daily, monthly or yearly basis. The platform also has features that help manage employee data and get customer feedback to constantly improve the platform based on real user input.

The detailed documentation of system functional and non-functional requirements is described in this Report along with the way the platform operates, what the platform will develop and what the platform should be developed to. This document is meant to be a reference for everyone within the stakeholder: developers, project managers, clients, and quality assurance teams to verify that the last system complies with stated requirements.

1.2 Background & Motivation

The way people shop has changed drastically with the rapid evolution of e-commerce hence, convenience, accessibility, and efficiency have become keys to the success of any successful online business. In particular, the fashion industry has seen a large increase in online shopping as people choose the convenience of browsing through the collections, buying and following their orders from their houses. As this demand was growing, the Online Clothing Shopping Store (OCSS) was conceptualized as a portal between sellers and customers that effectively blurs the boundaries.

The goal of this project is to deliver an easy-to-use platform for clothing ordering where customers can simply order clothes while also providing store owners with tools for storage control, sales monitoring, and performance analysis. We developed OCSS because the need for a robust, scalable solution to the complexity of online retail business in terms of product discovery, transaction security, order management and tracking, as well as in-store business with a Point of Sale (POS) system.

As an integrated OCSS and by integrating the online and offline capabilities the solution provides the balance between the benefits for customers and benefits for store managers, which leads to operational efficiency, customer satisfaction and result in the business successful.

1.3 Aim and Objectives

1.3.1 Aim

The aim of the Online Clothing Shopping Store (OCSS) project is to build a complete e-commerce platform which offers a comfortable, simple to use shopping expertise for prospects and powerful administration instruments for store owners. The objective of this project is to allow for world class online clothing transactions that includes secure user registration, user friendly product browsing, streamlined order processing, real time order tracking and effective inventory management. The OCSS objectives will be accomplished through achieving customer satisfaction, increased operational efficiency and a competitive position in the online retail market.

1.3.2 Objectives

- User Experience Enhancement: Create a user-friendly interface so customers can easily look through collections, put items in their cart and make safe purchasing.

- Seamless Order Management: Helping order to run a simple and effective order process that allows real time order tracking and notifications so customers know when their order shipped from process purchase to delivery.

- Secure User Registration & Authentication: Also, implement strong user registration and authentication, secure user data and transaction safety.

- Performance Analytics & Reporting: Provide store owners with the ability to generate sales reports, how they're performing as store employees, and create custom reports aligned with a certain set of criteria.

- Customer Engagement & Feedback: Build a feedback system that will empower customers, to share their experience and thus drive continual improvements to the platform.

- By combining these objectives, we will arrive at a full package of online and offline clothing retail system, which will not only make shopping ease but also bring the business intelligence and operational efficiency to store owners.

1.4 Summary

The Online Clothing Shopping Store (OCSS) is a comprehensive platform designed to streamline both online and in-store shopping experiences. It offers features like user registration, secure product browsing, cart management, order tracking, and a review system. Store owners benefit from sales tracking, and detailed reporting tools, along with a Point of Sale (POS) system for in-store transactions. OCSS enhances customer engagement through a user-friendly interface and ensures efficient order fulfillment, providing a scalable and secure solution for the fashion retail industry.

Chapter 02: Related Research

2.1 Overview

This chapter explores the foundational research and analysis that inform the development of the Online Clothing Sales System (OCSS). It begins with a literature review covering advancements in e-commerce technologies, industry best practices, and critical security measures for online transactions. By analyzing the strengths and weaknesses of existing systems, the chapter lays the groundwork for identifying unique features that will set OCSS apart in the competitive e-commerce landscape.

2.2 Introduction

The reason for conducting the literature review is to develop a working knowledge of prior studies on e-commerce technology, applications, and security measures. The purpose of this review is to explicate various systems' advantages and shortcomings, and prepare the reader for ideas presented in our Online Clothing Sales System (OCSS). All the practical ideas and technical solutions proposed in the course of the project are based on the results of previous researches and on the existing technologies that work for similar purposes and thus, avoid typical mistakes and fill in the gaps.

2.2 Literature Review

1. Existing Technologies

- Today's e-commerce technology environment is flexible because of the changes in web development frameworks, AI, and cloud computing. Cutting-edge technologies have revolutionized user experiences and streamlined backend processes:

- Responsive Web Design: Thanks to mobile shopping, separate versions of a platform are no longer needed, as platforms use responsive design to provide the optimal experience across devices.
- AI-Powered Search and Personalization: Almost all the platforms today use artificial intelligence to make the platform smart by applying intelligence search and relevant recommendations. Collecting the information about the actions of users, machine learning helps to choose what products to display and how to improve search results.

- Progressive Web Apps (PWAs): Hybird applications which bring the features of both, web site and application, and those can be used offline as well.
- Microservices Architecture: Microservice designs become popular in e-commerce systems to enhance system scalability and maintain improvement.

2. Best Practices in E-commerce Platforms

- Successful e-commerce systems follow a set of well-established best practices aimed at maximizing user satisfaction and driving conversions:

- User-Centric Design: These include the Showily and Amazon where the number of clicks and the simple search option matches with the actual product or page with a clear unpolluted font and background.
- Seamless Checkout Processes: To help with cart abandonment some of the leading platforms use one click check out options, auto fill forms, and multiple payment options. Returning users are also signposted to help reduce the number of steps for a new user during check out.
- Omni channel Strategies: Retailers such as Wal-Mart and Target blend in-store and online and mobile commerce for a Synchro modal strategy. This way the customers will have a good impression of the channel irrespective of the shopping channel.
- Customer Support: These include; catboats and the artificial intelligence customer supports due to their contribution towards increasing the efficiency of customer service by providing instant support and sending solutions to potential everyday questions and inquiries

3. Security Measures in Online Transactions

    - As more and more products and services are sold through portals on the World Wide Web, security is a key factor to protect the data of users and provide security against fraud. The following measures are standard in leading e-commerce platforms:

    - Two-Factor Authentication (2FA): This adds an extra layer of security on top of the traditional password in that to successfully login to an account, a user is expected to provide an additional piece of information such as a code sent to ones phone or email address.
    - PCI-DSS Compliance: Compliance with the Payment Card Industry Data Security Standard (PCI-DSS) is compulsory to any system that deals with cards. This includes the following; cardholder data encryption and transmission, as well as building secure network configurations.

- Anti-Fraud Mechanisms: AI is also used on platforms to identify and avoid fraud cases. Some of these models identify behaviour patterns such as; Unordinary spending or login patterns are also detected and blocked.
- SSL Encryption: An SSL (Secure Sockets Layer) certificates encrypt information exchanged in between users and the server and such as credit card details among other sensitive data.

4. Comparative Analysis

- To understand the competitive landscape, it is crucial to compare features available in top-tier e-commerce platforms:

- Amazon: Being a retail giant Amazon provides highly customized user experiences with especially noteworthy recommendation engine based on the machine learning. It also has a prowess in providing a good logistics and delivery system that will help in improving the satisfaction of the customer.
- eBay: eBay has also well developed its auctions and bidding processes, having a target audience in mind. It also has among the most elaborate feedback and review systems that serves to enhance buyer and seller trust.
- Shopify: Shopify is one of the leaders in the market of e-commerce solutions targeting the principal features of this platform are the focus on customization, the ability to scale the store and the clearly designed interface, allowing using it even without coding knowledge.
- OCSS Differentiators: The primary differentiation of our Online Clothing Sales System (OCSS) will be based on individual tailored clothing advice, integration of an artificial intelligence powered virtual fitting room and improved real time order tracking. All these features are intended for the fashion retail segment to enhance interaction with clients and the number of sales.

.

Chapter 03: System Analysis and Design

2.1 Introduction

Any software development process has system analysis and design as its backbone. In order to analyze the system requirements and design architecture of the system to meet the requirements of customer and store owners a complete approach was employed for Online Clothing Shopping Store (OCSS). In this chapter, I explore in detail how requirements were derived, how they were analyzed, and how they influenced the systems design. The collected data is the subject of the outline techniques used to collect it, it compares other systems, and the design approach that guarantees scalability, usability and security.

2.2 Requirement Analysis

This phase is critical stage in software development, as it identifies, collects and documents all the functional and non-functional requirements of the application. This process for the OCSS ensured alignment of the platform with user needs and business objectives. This process involved gathering comprehensive insights from these involved key stakeholders such as customers, store owners and staff. The system's features, designs and architecture were influenced directly by the outcome of the requirement analysis phase.

2.2.1 Requirement Gathering Techniques

Different requirement gathering techniques were used so as to train the system about all that the requirements of system are. This allowed them to find out what functionalities, user behaviors and system constraints are needed.

- **Similar Systems**

The features and functionalities implemented in local online clothing platforms were analyzed to understand the fashion e-commerce industry in Sri Lanka. Example of these is: Fashion Bug, House of Fashions, Kandy Selection and Thilakawardhana**.** In terms of making purchase and vendor operations more efficient, these platforms provide features such as categorized product browsing, user registration, secure methods of online payment and order tracking. Each of these systems takes advantage of a combination of traditional and contemporary elements to improve customer satisfaction.

Review of these systems revealed the need for the Online Clothing Shopping Store (OCSS) to have a user-friendly interface, secure payment gateways and robust order tracking mechanisms. These insights from these platforms informed OCSS to compete locally on the market and also help provide a unique and optimized user experience.

- **Manual System Documentation**

Insights from traditional order processing, inventory tracking and customer service methods used on manual systems were gathered through a review of manual systems previously used by small

garment stores. Efficiencies and bottlenecks in manual workflows like delayed feed of inventory updates and order handling issues were exposed by this documentation. In the aim to do away with these inefficiencies and automate pertinent processes such as real time stock update, order tracking and so on, OCSS digitized it.

- **Questioners**

A number of questionnaires were distributed among store owners, staff and customers to gather direct feedback from potential users. The questionnaires focused on user preferences for online shopping, preferred payment methods, feedback mechanisms and order tracking needs. The data collected helped us shape the user interface design, so it's intuitive and fits the ways that people use most often. The responses also included insights on the relevance of some elements such as secure payment gateways, personalized recommendations, the possibility to leave reviews.

- **Interviews**

We conducted interviews with key stakeholders: store managers, IT administrators and end users to get a better handle on what they expect and what exactly they are in pain. Through these one-on-one interviews we had the opportunity to delve into these items: system requirements, features and constraints in detail. Real time reporting, inventory management were mentioned as store owners stress, while customers demand a seamless shopping experience, quick order fulfillment and secure payment processing. These interviews helped in identifying the non-functional requirements i.e., system performance, security and scalability.

2.2.2 Functional Requirements

1. User account creation and account control

- Sign-Up Process: The new system will let users create an account with the application using their e-mail or through their social accounts like, face book or Google account. Then, an email will be sent for the validation of the created account.
- Profile Management: Registered users are able to change other particulars including name, other contact address information, and shipping and payment information. User’s profile details can be changed any time, through control panel/dashboard.
- Account Security: To make its protection even more reliable, users may apply two-factor authentication or 2FA whenever it is possible to guarantee that only those users who are allowed to enter the account can do it.

2. Product Browsing and Filtering

- Product Search: Some of the fields will allow users to enter keyword-searchable text for finding clothing (e.g., “red jacket” and “summer dress”). The search results will be shown in the form of product images, price and a short description of the product.

- Filtering Options: There will be many filter options like size, colour, price, gender, type and brand. The filters will help them narrow down the search results to their preferred option of the product in the market.
- Sorting: Customers can filter products according to their needs and preferences such as; price, rating, or even the newly listed products.

3. Cart and Checkout Process

    - Cart Addition: Each product can be added to a user's cart with a single click. The cart will update itself when an item is added and will display count and subtotal of order as well as the probable shipping charges.
    - Edit Cart: Consumers will be free to amend the contents of the cart either by altering the number in each item or even deleting an item.
    - Checkout Process: The checkout in the process will therefore not imply a mere point-of-sale:

    i) Review Cart: Still, shoppers get informed about the contents of the cart, or delete or continue.
    ii) Shipping Information: Input delivery information and select the delivery option.
    iii) Payment Options: Customers will be able to use their credit/debit cards, PayPal, or other integrated payment to make payments. For security, the system is going to encrypt payment information by using SSL/TLS protocols.

    - Order Confirmation: Once payment is effected, the users will be posted an order number with dispatch number and an expected time of arrival.

4. Tracking of orders and the notification of the clients.

    - Order Tracking: SEO users can track the stated order through their profiles as soon as they place the order. It will help to track the order's location and time of delivery in the shortest time possible.
    - Notifications: For general use the notification will be through emails and or messaging systems such as for order confirmation, shipment details for delivery details or if there was any complication during the processing. Patrons will also be allowed to set their notification priorities.

5. Admin Panel

- Product Management: The system will provide an opportunity to add, modify or delete products in the catalog for administrators only. They can modify product details, images, prices, and the item's availability in stock.

- Order Management: From the dashboard that is intended for administrators, one will be able to see all orders made by the user; the order status, payment details, shipping details, and any comment from the user. Admins will also be able to change the status of the order that contains processed, shipped, delivered and the likes.
- User Management: From this area, admins can trace the user's interaction with the site, interacting with complaints, or even providing refunds if needed.
- Analytics and Reports: The admin panel will offer various forms of figures and graphs, like sales, most bought products, customer distribution, traffic, etc. enabling the admins come up with sound business-related decisions.

2.2.3 Non-Functional Requirements

1. Security

Data Encryption: Any personal attribute of the user that is submitted to the system, such as payment information, will be encrypted using standard industry encryption techniques, including AES-256. All transactions will be protected with SSL certificates.

Fraud Detection: The system using machine learning techniques, to consider a particular purchasing pattern as secure or insecure, to consider an account as fraudulent. Only high-risk orders can be reviewed manually.

User Privacy: It is legal for people's information like under GDPR to be protected to avoid unauthorized use. We will anonymize or delete personal data where such request is received.

2. Performance

Optimized Load Time: The system itself will have low loading time and it will be achieved through means like compressing images, loading media only when needed, and caching of frequently accessed information.

Database Optimization: Optimized queries into the database will be employed so that other details such as products, users as well as order's status will be retrieved in the shortest time. Such measurements like indexing standards and data normalization discussed here will enhance database efficiency and decrease the load.

Content Delivery Network (CDN): To improve downloads of static content such as images, CSS, and JavaScript a Content Delivery Network shall be used to cache content from servers close to the user.

3. Scalability

Vertical and Horizontal Scaling: Thus, the architecture of the system will allow for vertical (extension of server capabilities) and horizontal (acquisition of more servers) scaling to accommodate traffic during the peak-selling seasons, such as the Black Friday.

Cloud Infrastructure: The system will be deployed in Amazon Web Services, Microsoft Azure or similar cloud services in order to accommodate the growing user client aggregate and product portfolio.

Database Shading: In the future, the database will be partitioned (sharded) as the platform expands in order to meet the demands of big data and maintain high availability.

4. Usability

User Interface Design: The design will consist of relatively simple aesthetics with fairly epitomized layout organization and a distinctly minimalistic interface that a user has to involve themselves into mostly merely to surf the website, shop, check out, accomplish point-of-purchase, and track orders.

Mobile Responsiveness: The platform will also be completely mobile and capable of being viewed just as well on a desktop, tablet or small smartphone screen.

Accessibility: Many aspects including accessibility for handicapped users, allowing screen reader, open with keyboard or high contrast mode, WCAG (Web Content Accessibility Guidelines) compatibility will be provided by the system.

5. Reliability

High Availability: Load balancing and fail over mechanisms will be incorporated in the system for a very high availability such that in the event of failure of a given server, the level of time required to come back online will be less.

Error Recovery: Proper error provide proper error handling mechanism will be implemented with potential solutions in the case of transactions, redirecting or broken links, etc.

Backups: This will be achieved through the following features: The system will backup as a way ensuring that no data gets lost. Data backups will be kept off-site and tested this year and annually thereafter for their ability to be restored.

2.3 Design Diagrams

A graphical representation of a user's potential interactions with a system is called a use case diagram. A use case diagram will frequently be accompanied by other types of diagrams and will display the various use cases and user types the system has. Either circles or ellipses are used to represent the use cases.

2.3.1 Use Case Diagram for Full System

Figure 1 - Full system use case diagram

2.3.2 Use case Diagram for Customer Registration

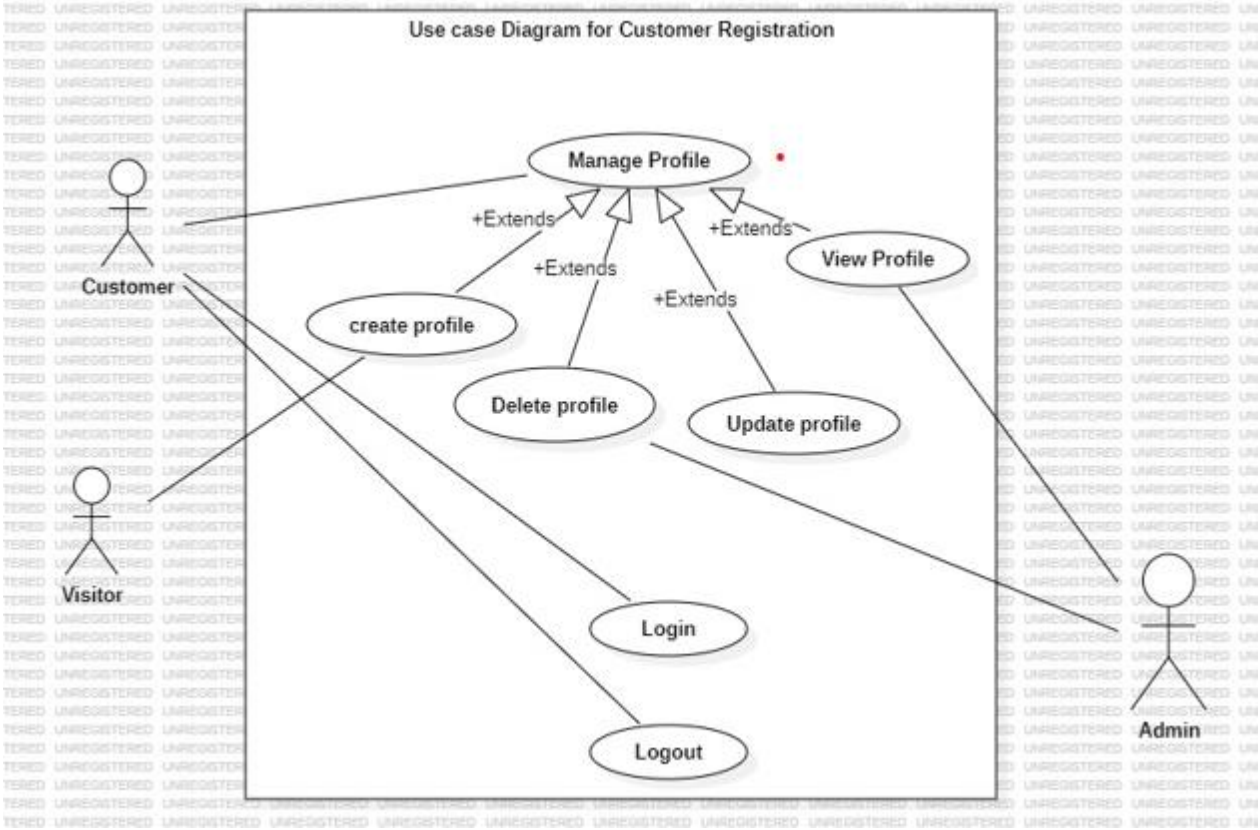

Figure 2 - Use case Diagram for Customer Registration

Use-Case Name	Customer Management
Use-Case Type	Business
Use-Case ID	FO-01
Requirements Priority	High Priority
Source	Online Form
Primary Business Actor	Admin
Other Participating Actors	Admin
Other Interested Stakeholders	Customer
Description	1. Customer Registration 2. Customer Login 3. Customer Edit Profile 4. Customer Delete Profile 5. Customer Logout

Table 1 - Use case customer management

2.3.3 Use case Diagram for Order Management

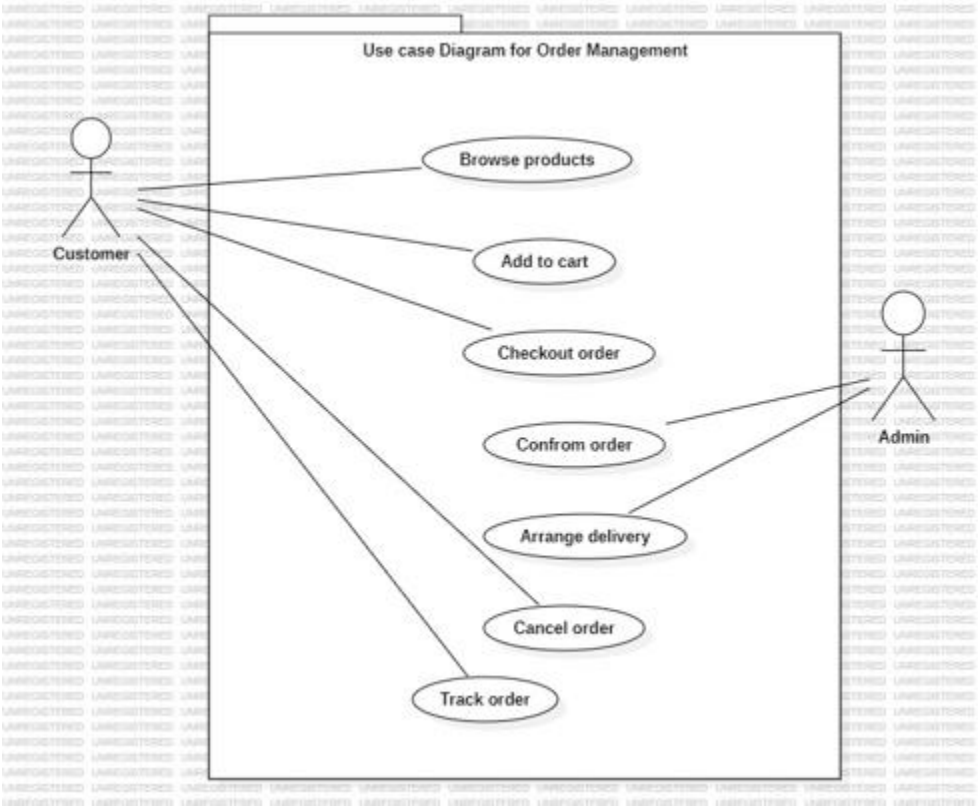

Figure 3 - Use case Diagram for Order Management

Use-Case Name	Order Management
Use-Case Type	Business
Use-Case ID	FO-02
Requirements Priority	High Priority
Source	Online Form
Primary Business Actor	Customer
Other Participating Actors	
Other Interested Stakeholders	
Description	1. View product 2. Add to cart 3. Checkout order 4. Confirm order

Table 2 - Use case Order Management

2.3.4 Activity Diagram for Admin Login

A class diagram in software engineering is a kind of static structure diagram that reveals the classes, attributes, operations, and relationships between objects in a system to describe the system's structure.

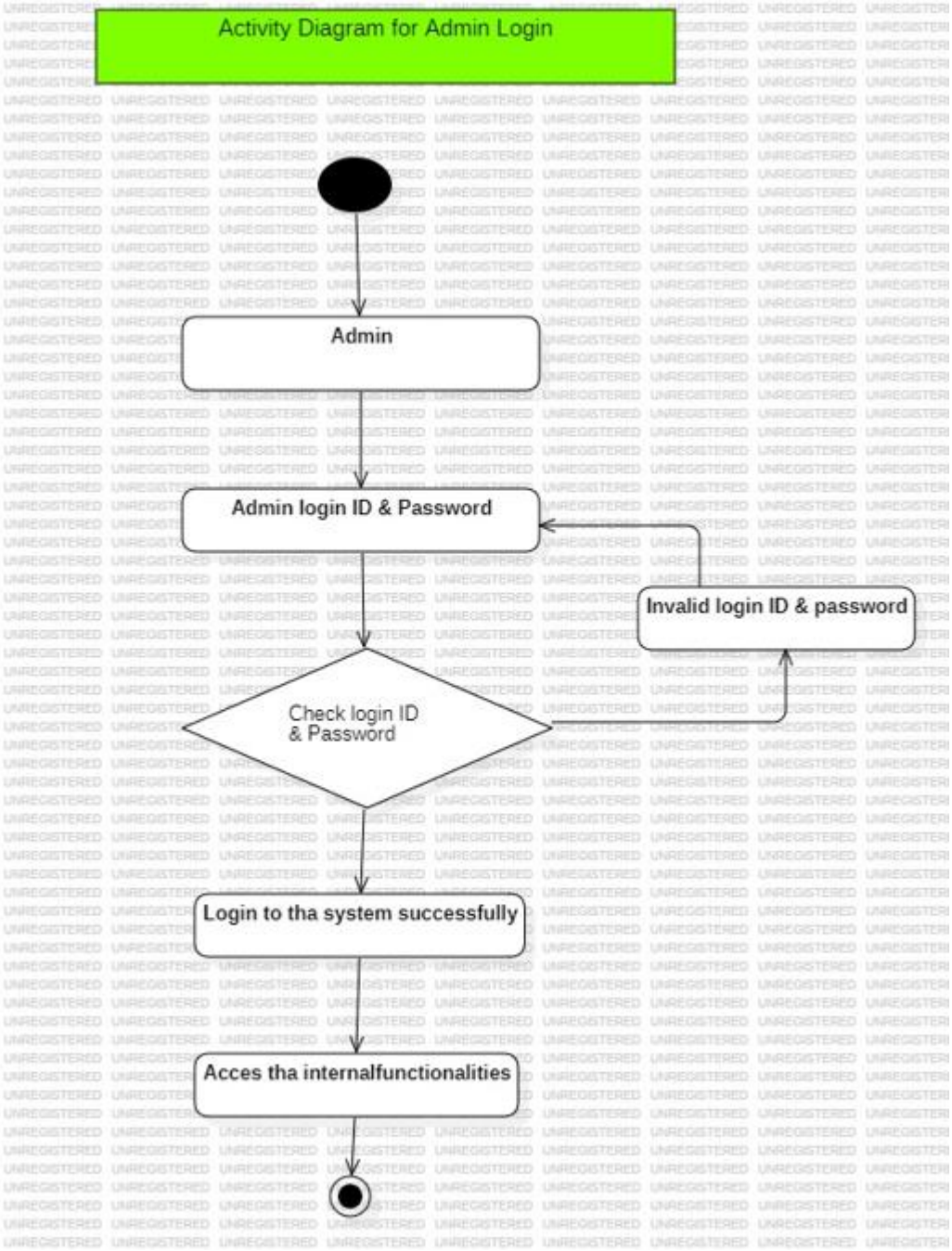

Figure 4 - Activity Diagram for Admin Login

2.3.5 Activity Diagram for Order

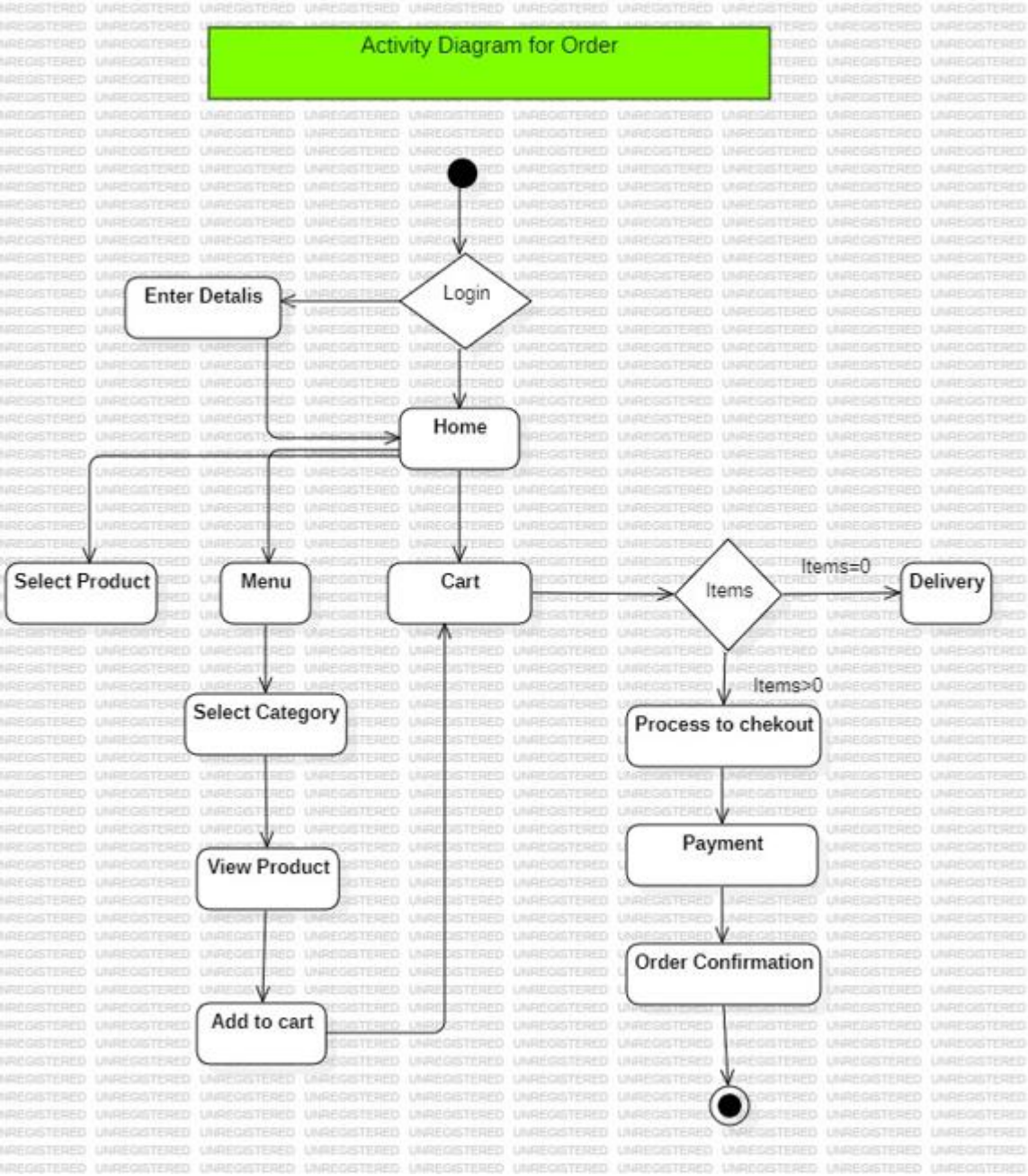

Figure 5 - Activity Diagram for Order

2.3.6 Activity Diagram for Registration

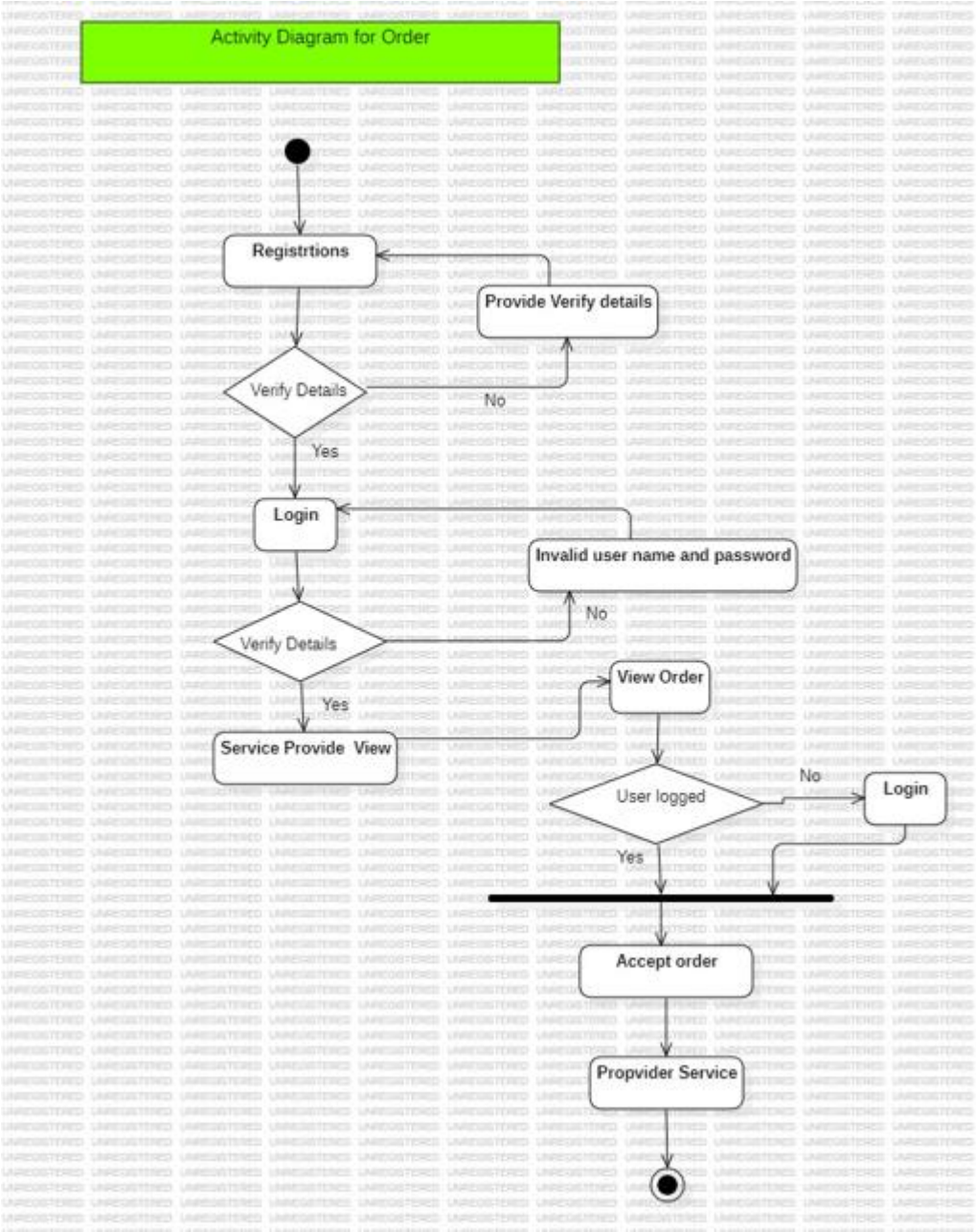

Figure 6 - Activity Diagram for Registration

2.3.7 Sequence Diagram for Admin Login

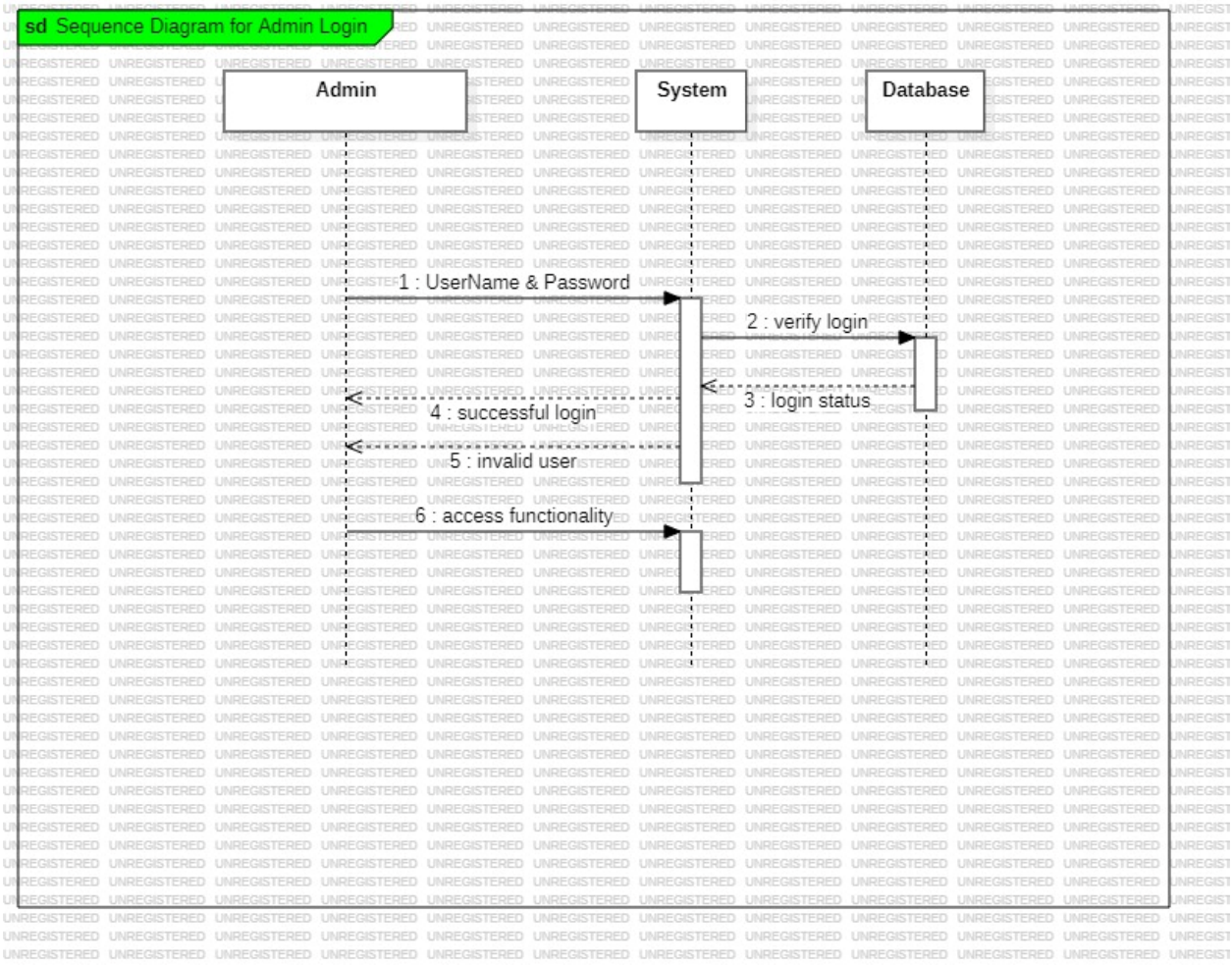

Figure 7 - Sequence Diagram for Admin Login

2.3.8 Sequence Diagram for Order

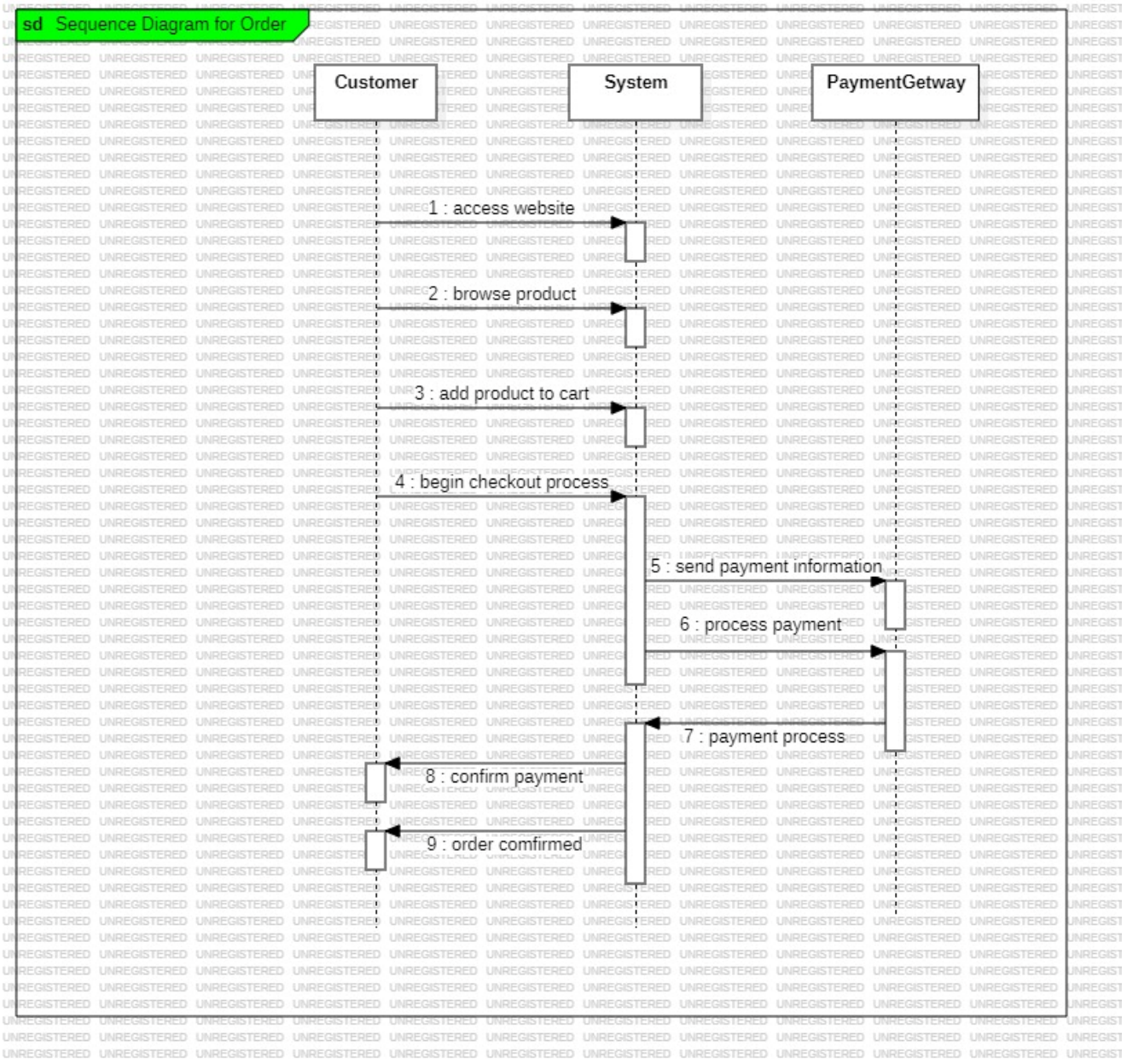

Figure 8 - Sequence Diagram for Order

2.3.9 Sequence Diagram for Registration

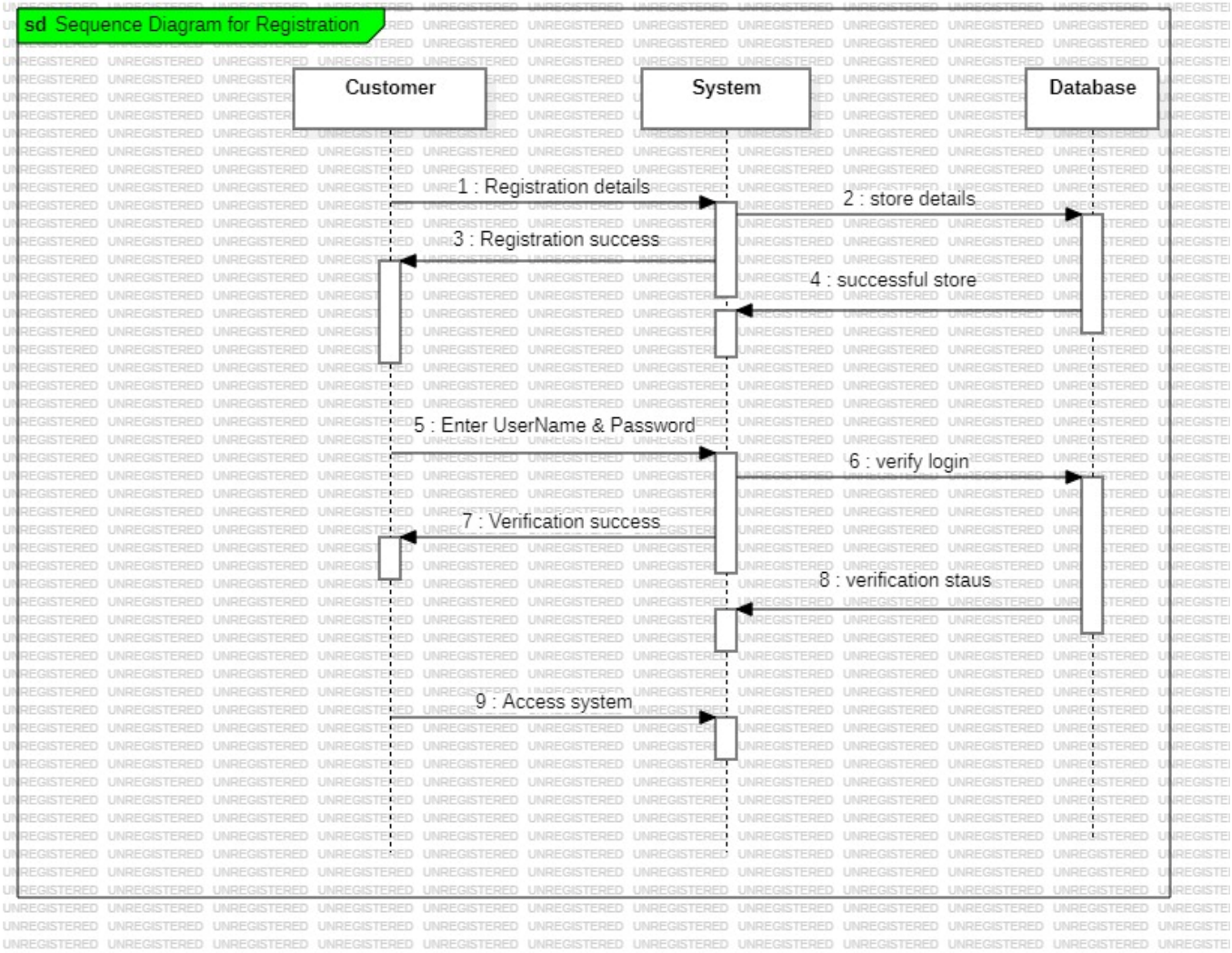

Figure 9 - Sequence Diagram for Registration

2.3.10 Class Diagram

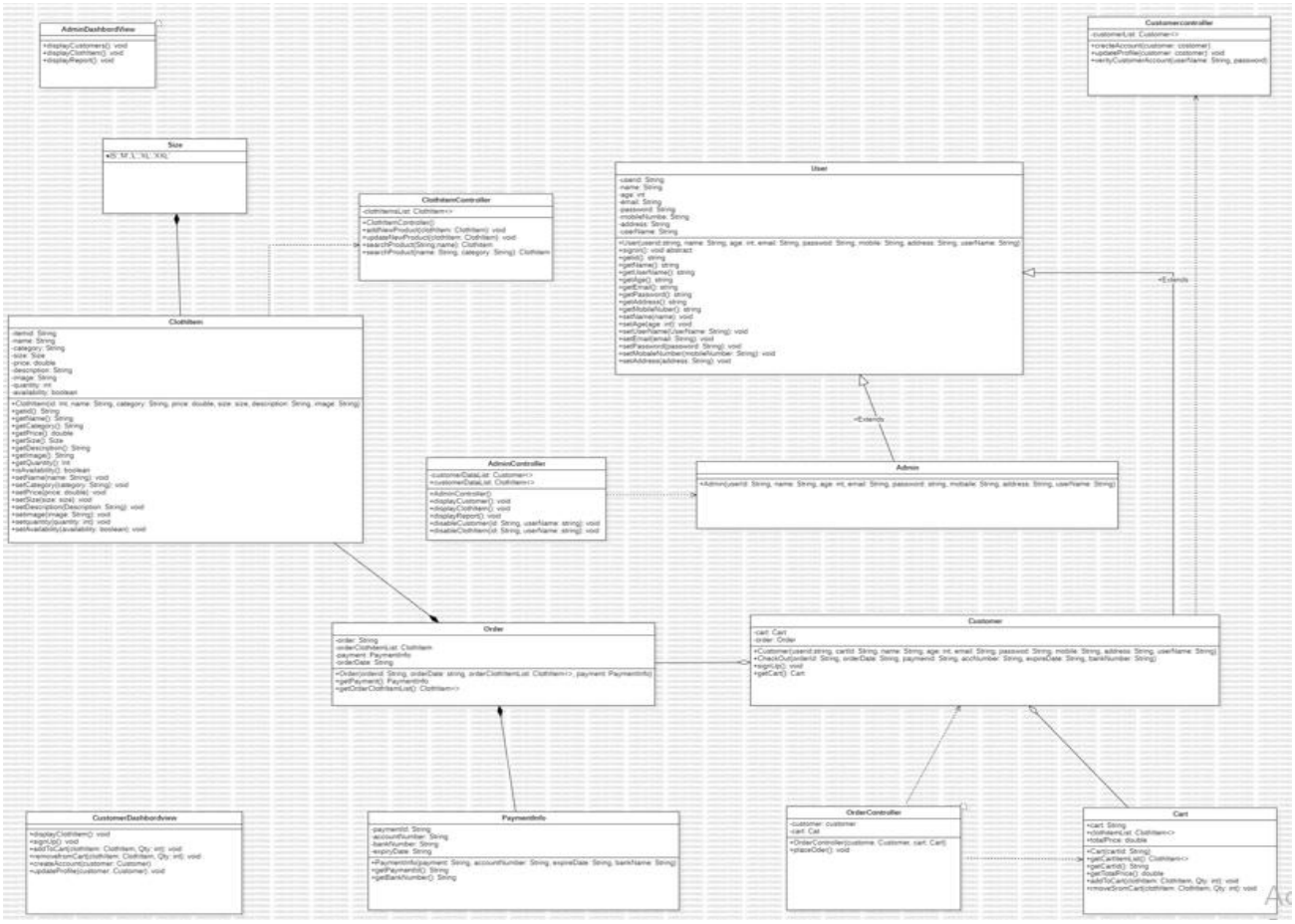

Figure 10 - Class Diagram

2.3.11 EER Diagram

Through high-level models and tools, EER Diagram, also known as "Enhanced Entity-Relationship Diagram" assists us in building and maintaining comprehensive databases. Additionally, they are an extended form of the fundamental ER diagrams and were created from them

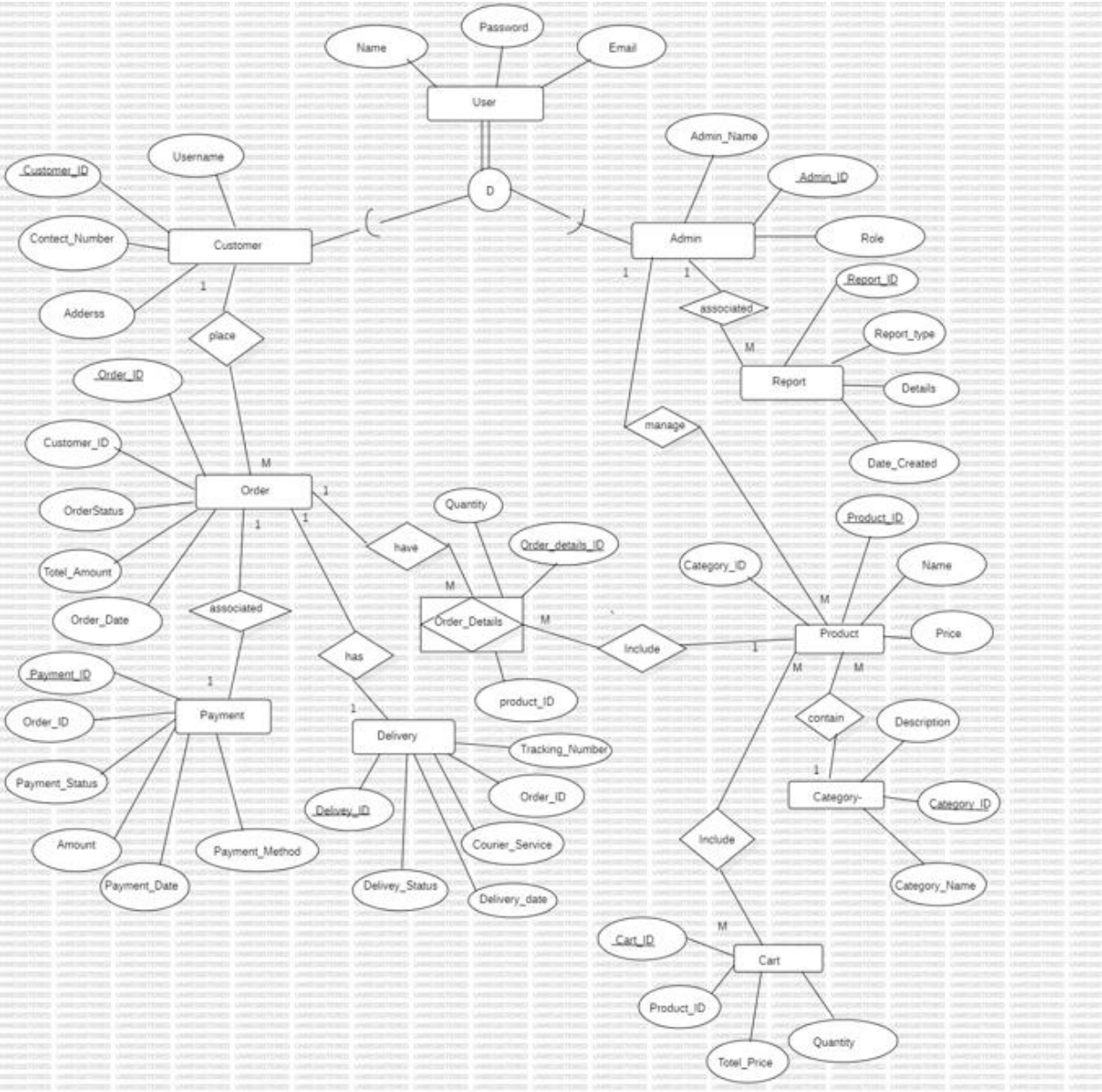

Figure 11 - EER Diagram

EER Diagram Assumptions

- Each clothing item belongs to only one category and one subcategory (e.g., Men, Women, Kids).
- The Customer ID uniquely identifies each customer, order, cart, payment, and shipping information.
- Customers can browse clothing items, add them to the cart, and make purchases.
- Customers can browse specific clothing items, including size and availability options.
- After placing an order, customers can track their order status and shipping information.

- Customers can pay for their orders using various payment methods (e.g., credit card, COD) at checkout.
- Only registered customers can place orders and manage their profiles, but unregistered users can browse items.
- Each customer can save multiple shipping addresses but can only select one per order.
- Orders are processed once payment is confirmed, and customers receive a confirmation email.
- Administrators can manage, update product information, and handle customer inquiries.

Chapter 04: Technologies Adopted

4.1 Introduction

The above chapter covered the topic of creating diagrams. There are several schematics for designing. Use case diagrams, Activity diagrams, Sequence diagrams, and Class diagrams are all located beneath the UML diagrams. EER diagram and Relational Schema are used in database design. In order to better comprehend the system, use case diagrams and use case narratives are generated for each module.

This chapter covered the topic of technologies adapted. There are requirement in this chapter, Software requirement, Hardware requirement, Technologies, and Other requirement.

4.2 Software Requirements

4.2.1 Star UML

Star UML is an open source software modeling tool that supports the UML (Unified Modeling Language) framework for system and software modeling. It is based on UML version 5.0.2, provides eleven different types of diagram and it accepts UML 2.0 notation. It actively supports the MDA (Model Driven Architecture) approach by supporting the UML profile concept and allowing to generate code for multiple languages. Version 5.0.2 is used for draw diagrams

4.2.2 phpMyAdmin

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc.) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

4.2.3 Operating System

The operating system (OS) controls all of the computer's software and hardware. Windows 10 has an official end of support date of October, 2025, with Windows 11 as its successor. This computer's operating system Version is Windows 10 pro.

4.2.4 Visual Code

An easier-to-use code editor called Visual Studio Code supports debugging and version control as well as other development-related tasks. It aims to provide only the tools necessary for a quick cycle of code-build-debugging, leaving more complex workflows to fully functional IDEs like Visual Studio IDE. Version 1.74.2 is used for implementation.

4.2.5 XAMPP

MariaDB, PHP, and Perl are all included in the totally free and easy-to-install Apache distribution known as XAMPP. Installing and using the XAMPP open-source software is incredibly straightforward. We used XAMPP Control panel V3.3.0.

4.3 Hardware Requirements

4.3.1 RAM

The RAM (random access memory) in the CPU is where data, programs, and program output are kept. During machine operation, data is stored in this read/write memory. When a machine is turned off, all data is erased. 4GB RAM is used for this.

4.3.2 Processor

A processor is the piece of logic hardware that receives and processes the fundamental commands that run a computer (CPU). Since it interprets the majority of the computer's commands, the CPU is regarded as the main and most significant integrated circuit (IC) chip in a computer. In there, the processor is Intel(R) Core(TM) i3-2348M CPU @ 2.30GHz 2.30GHz.

4.4 Technologies

4.4.1 Frontend

The frontend of the Online Clothing Sales System (OCSS) is developed using dynamic and contemporary web platforms to enhance the graphical user interface. The chosen technologies make it possible to enable a consistent experience across different form factors.

4.4.1.1 CSS

A simple and straightforward method for managing the style of web content is CSS. The term "Cascading Style Sheet" is CSS. Cascading Style Sheets, also known as CSS, is a straightforward design language created to streamline the process of making web pages aesthetically pleasing.

4.4.1.2 HTML

Hyper Text Markup Language, also known as HTML, is the most popular language for creating web pages on the Internet. Serves as the core of the application since it interprets the meaning of the web pages. Full multimedia support (audio and video) can be incorporated with no need for additional plugins, and it supports the concepts of accessibility for various user types in a single plugin.

4.4.1.3 JavaScript

JavaScript is a simple, interpreted programming language. The creation of network-centric applications is its intended use. It enhances and combines with Java. JavaScript is comparatively easy to implement because it is integrated with HTML. It is open and cross-platform

4.4.2.Backend

It specifically represents backend of the Online Clothing Sales System (OCSS) where every fundamental processing is performed and data is managed to maintain a smooth interaction of front end and database. These are strategic and architectural components that enable an e-commerce solution to be performance and scalability optimized and secure to meet fast changing market conditions and demand.

4.4.2.1 PHP

With the help of the programming language PHP, web designers can create dynamic content that communicates with databases. Most web-based software applications are made with PHP.

4.4.2.2 MySQL

The relational database management system MySQL is open source and free. In tables made up of rows and columns, MySQL stores data in a similar manner to other relational databases. Users can define, manipulate, control, and query data using SQL, or Structured Query Language. The most well-liked open-source database system, MySQL, is a strong and versatile tool.

4.5 Database

The OCSS database is focused on the most important information which includes user details, products and orders. MySQL is chosen for managing data because it commonly recognized as reliable and scalable for the needs of the platform.

4.5.1 Database Technology

MySQL is selected because it is reliable for big web-based systems and other ‘e-business’ solutions. Platforms. Key reasons for selection include:

- Reliability: MySQL is a pretty stable relational database management system, which allows preventing different issues with data integrity and protection.

- ACID Compliance: Offers security and validity which is a very important factor when it comes to business especially the online business firms.

- Community Support: A wealth of vehicles and help available when things go wrong or for bringing out the best or most from the system.

- Scalability: The fact that providers can add more depth to the solutions employed as well as extent of the architectures used, to meet both increasing numbers of active users and the data traffic they generate.

4.5.2 Database Design Principles

- Normalization:

Was normalized up to third normal form (3NF) to solve the problem of data redundancy and optimize storage.

Example: Transferring of users' data as well as records of transactions into different tables that are related by foreign keys.

- Indexes:

Developed on the most used columns that include the user_id, product_id, and order_id in an attempt to optimize the queries that may be used on this table.

- Constraints:

Primary and foreign or mandatory constraints have been employed to ensure data integrity.

Some constraints introduce the ability not to have similar records (for example, distinct email addresses for users).

- Backup and Recovery:

The use of backup and recovery measures is used in order to ensure the data is safe in case of failure.

4.5.3 Database Operations

- Data Retrieval: That is why efficient querying is important to organize further quick response to users' queries and filter products if needed.
- Transactions: Every operation pertaining to the order is managed using MySQL's transaction support leading to either atomicity, consistency, isolation, and durability (ACID).

- Scalability: Some of the features of the schema are developed to fit horizontal scaling so as to improve on the performance of the system when it is expanding.

4.5.4 Relational Schema

The schema is structured to ensure data consistency, efficiency, and scalability while supporting all key functionalities of OCSS:

1. **orders**:

Order ID	User ID (foreign key)	Product ID (foreign key)	Quantity	Total Price	Payment Status	Timestamp

2. **Order items**

Order Item ID	Order ID (foreign key)	Product ID (foreign key)	Quantity	Price.

3. **User Table**

Name	Email	Password

4. **Customer Table**

CustomerID	Username	Address	ContactNumber

5. **Admin Table**

AdminID	AdminName	Role

6. **Order Table**

OrderID	CustomerID	OrderStatus	TotalAmount	OrderDate

7. **Reports Table**

ReportID	ReportType	DateCreated	Details

8. **Products Table**

ProductID	ProductName	Description	Price	CategoryID

9. **Payment Table**

PaymentID	OrderID	PaymentMethod	Amount	PaymentStatus	PaymentDate

10. **Delivery Table**

DeliveryID	OrderID	CourierService	DeliveryStatus	DeliveryDate	TrackingNumber

11. **Category Table**

CategoryID	CategoryName

12. **Order Details Table**

OrderDetailsID	ProductID	Quantity

13. **Cart Table**

CartID	ProductID	Quantity	TotalAmount

4.6 Additional Tools

To enhance functionality and ensure security, the following additional tools and integrations were utilized:

APIs:

- Email sending using resend API service

Security Measures:

- SSL/TLS Encryption: Oversees the exchange of data between the client and the server making it secure by maintaining the privacy of the data.
- Bcrypt: Employed while creating password forms in that it takes an offender many tries to guess a common user's username and password.

Version Control: Git and GitHub were used as repository and version for the collaboration code purposes.

Deployment Tools: Containerization was performed with the help of Docker, making environments used in development, testing, and production stages equal.

Summary

Thus, effective frontend and backend frameworks, functional and optimized database management, and additional programs for improving capabilities and security are used to create the Online Clothing Sales System design to provide customer-oriented and safe shopping. If you need more or different information let me know!

4.7 Summary

This chapter discusses the adapted technologies described in the previous chapter. There are software requirements, hardware requirements, and technologies under "technologies adapted." In this chapter, we describe the software that we use to create the suggested system under the software requirements section. In this chapter, desired hardware and technologies like HTML, CSS, PHP, and MySQL are listed under hardware requirements. The proposed system's implementation section is shown in the following chapter. It shows a variety of frontend and backend interfaces for the suggested web application. It also shows the pertinent code segments.

Chapter 05: Implementation

5.1 Introduction

The fourth chapter describes the technologies needed to create the proposed system, as well as the hardware and software requirements. This will give users a better idea of how to meet their needs before a web-based application is implemented. The user interfaces of the suggested system are described in this chapter. The frontend and backend interfaces are shown in this chapter along with the corresponding code segments.

5.2 User Interfaces

5.2.1 Font End User interfaces

The layer above the back end is the front end and it includes all software or hardware that is part of a user interface. Human or digital users interact directly with various aspects of the front end of a program, including user-entered data, buttons, programs, websites and other features. Most of these features are designed by user experience professionals to be accessible, pleasant and easy to use.

5.2.1.1 Login Interface

The Login Interface which is the main point of entry into the accounts, is intended to offer an easily navigable yet secure way. This page includes:

Features:

- An input form with only two fields, either of which may contain the user's email or username, and the other must contain the password.
- Input validation on the fly for creating proper format and completeness of entries provided by a user.
- RETRIEVAL – Account recovery tab which includes "Forgot Password" link.

Security Measures:

- Password encryption aside from the use of a simple hashing algorithm such as bcrypt.
- Additional security measures that make use of two factor authentication methods.

User Experience:

- Uncluttered design with relatively few gimmicks to prevent users from getting distracted while carrying out authentication.

As shown in Figure 12, the login interface is simplistic and effective with greater emphasis placed on the overall aesthetics of the interface and accessibility in terms of login but also security considerations.

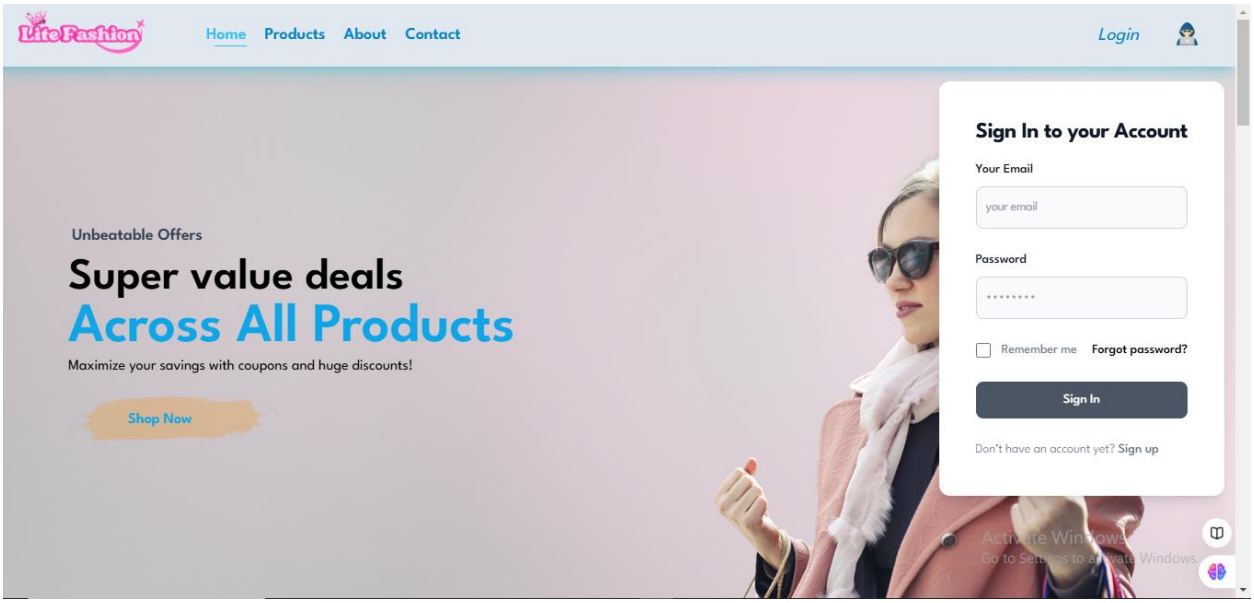

Figure 12 - Login Interface

5.2.1.2 Products Interface

The Products Interface is the window for presenting all the items that are being sold in the online store to help the users navigate through.

Features:

- Responsive grid system that adjusts the products' placement to all devices.
- Small images of the products including the name, price, and ratings.
- Search options (such as by products in categories, price, or number of hits).
- Flashing icon for users who just want the basic details of the product without having to go to another page.

Interactivity:

- Hover effects to enhance the activistic feature of browsing.
- Implementation of Ajax for effective loading of more products.

Figure 13 illustrates the products interface aspect, and the fact that it is adaptable.

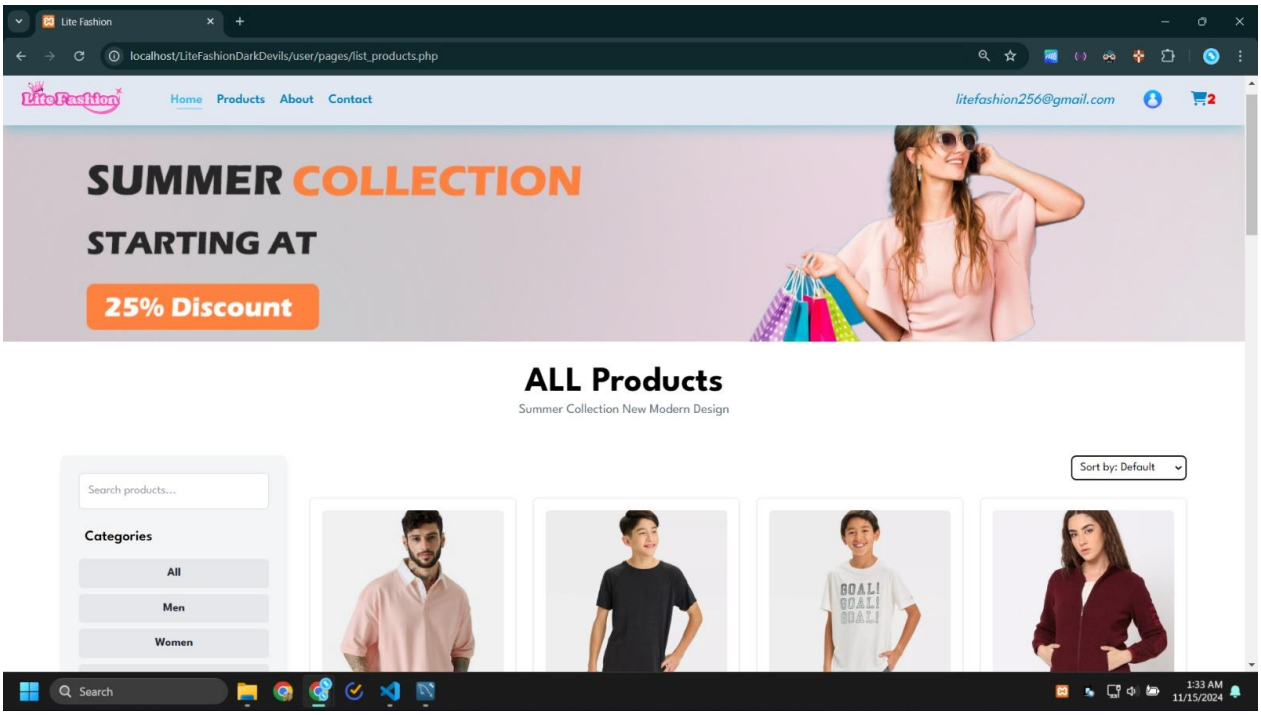

Figure 13 - Products Interface

5.2.1.3 Product Details Interface

The Product Details Interface contains detailed information on one or several products to help make the right choice.

Features:

- Good product images that zoom in on them.
- Technical product information, such as characteristics, constituent materials, and size.
- Efficiency of options for size, color or variant choice.
- Use of customers' feedback and rating to enable people to associate with the brand.
- For quick actions, the button is left with "Add to Cart" on it.

The pictorial view about the product details information in Figure 14 enables clients to view all details of the product at once.

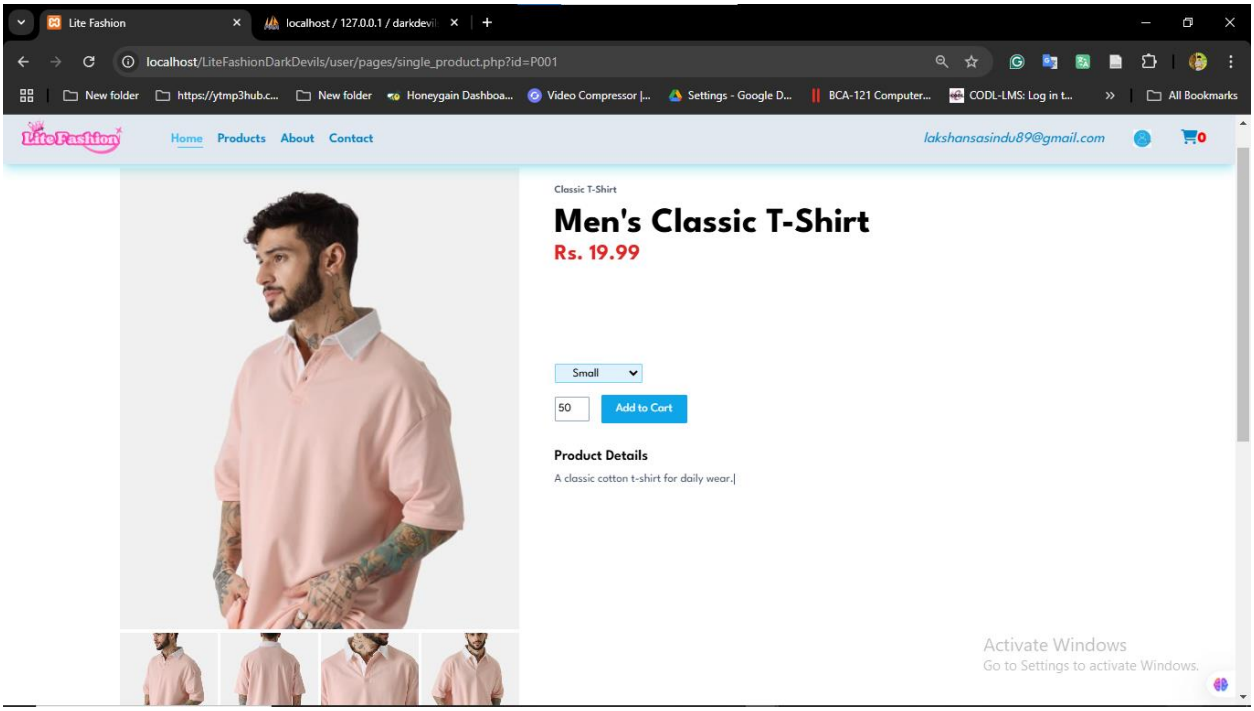

Figure 14 - Product Details Interface

5.2.1.4 Search Option

The enhanced functionality of search option enables users to find specific products easily, making the usage easier.

Features:

- An auto-suggest for the search bar that will display the results immediately a user starts typing.
- Lists of sites obtained via keywords identified based upon relevance, rating, or last updated.
- Additional sub-screens for fine tuning the search results (e.g., by price or category or manufacturer).

Performance:

- So enhanced by backend algorithms for faster query responses.
- Integration of the present work with AI to include search with the use of predictions.
- This is evident in figure 15 showing the search option as been more of an intuitive and highly powerful option when it comes to the movement of users from one section of the site to the other.

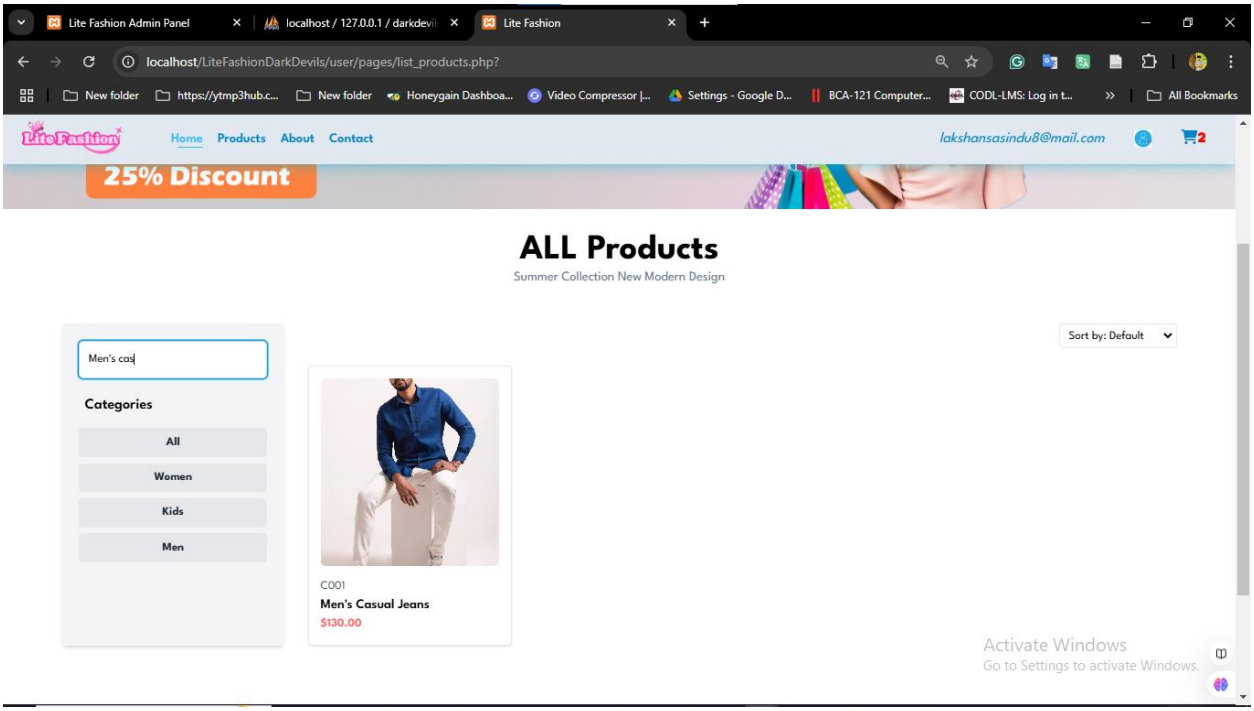

Figure 15 - Search Option

5.2.1.5 Cart Interface

The Cart Interface is the one that offers users an option of sorting through the items selected before going for the next process of paying for them.

Features:

- All the items added will be shown including name, image, price and quantity.
- Automated computation and display of total cost when new items are added, deleted or when individual cost rates are modified.
- Still as a shopping cart, it has features of enabling one to adjust the quantities of the items or even remove such items from the cart.
- 'Checkout' button that leads to check out that's where clients participate in the payment process.

User Convenience:

- It employs AJAX for display of cart details without the need for having to reload the entire page.
- Easy to identify aspects of the websites that show that discounts or some special promotions are available.
- As seen in figure 16 below, the cart interface is created with a user-friendly interface to easily managed purchases and improve the general shopping.

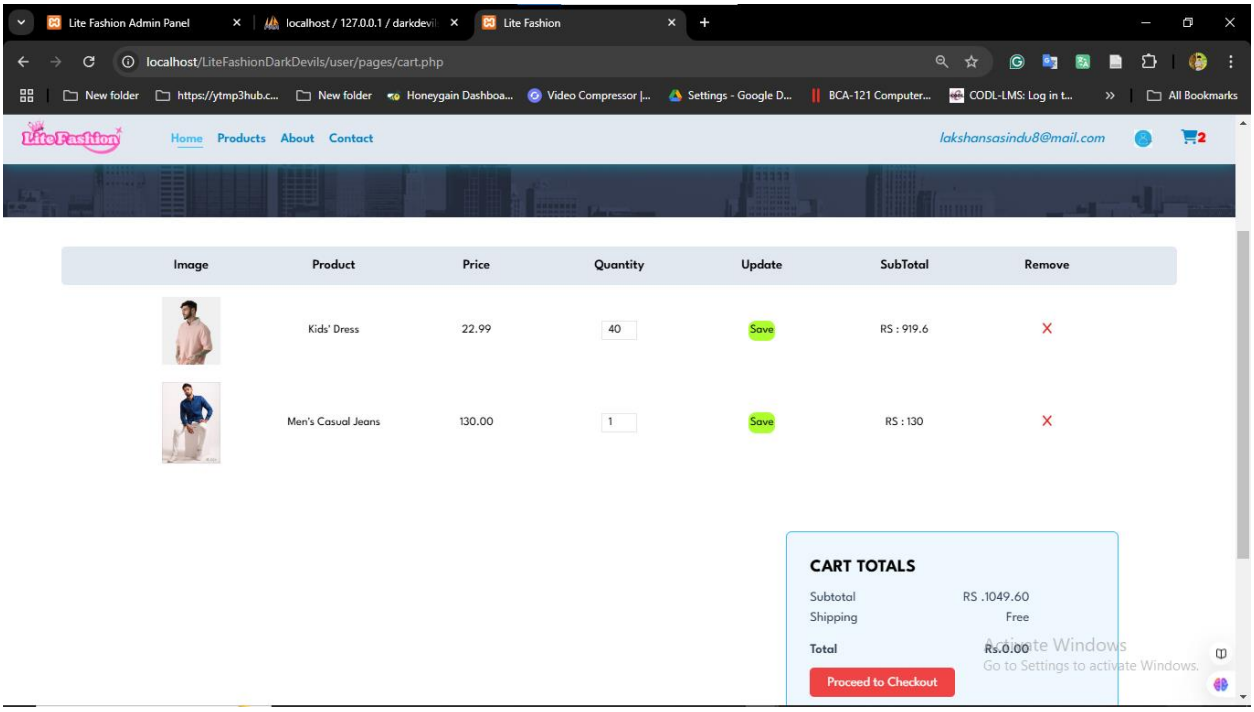

Figure 16 - Cart Interface

5.2. Back-end user interfaces

The back end refers to parts of a computer application or a program's code that allow it to operate and that cannot be accessed by a user. Most data and operating syntax are stored and accessed in 48 the back end of a computer system. Typically, the code is comprised of one or more programming languages.

The back end is also called the data access layer of software or hardware and includes any functionality that needs to be accessed and navigated to by digital means. The following figure interfaces which belong to administrator and it is use to add new administrator, delete an existing administrator and edit the current administrator information. After every deletion, or updating feedback is given to user whether the modification updates the database or not.

5.2.1 Admin Dashboard

The **Admin Dashboard** serves as the central hub for administrators to monitor and manage system activities.

Key Features:

- Overview of system metrics (e.g., total sales, new users, active orders).

- Charts and graphs for quick data visualization (e.g., sales trends, product performance).
- Notifications (e.g., low stock alerts, system updates).
- Quick navigation to other modules (e.g., Add Product, Product Dashboard).
- Recent activities or logs (e.g., products added, orders updated).

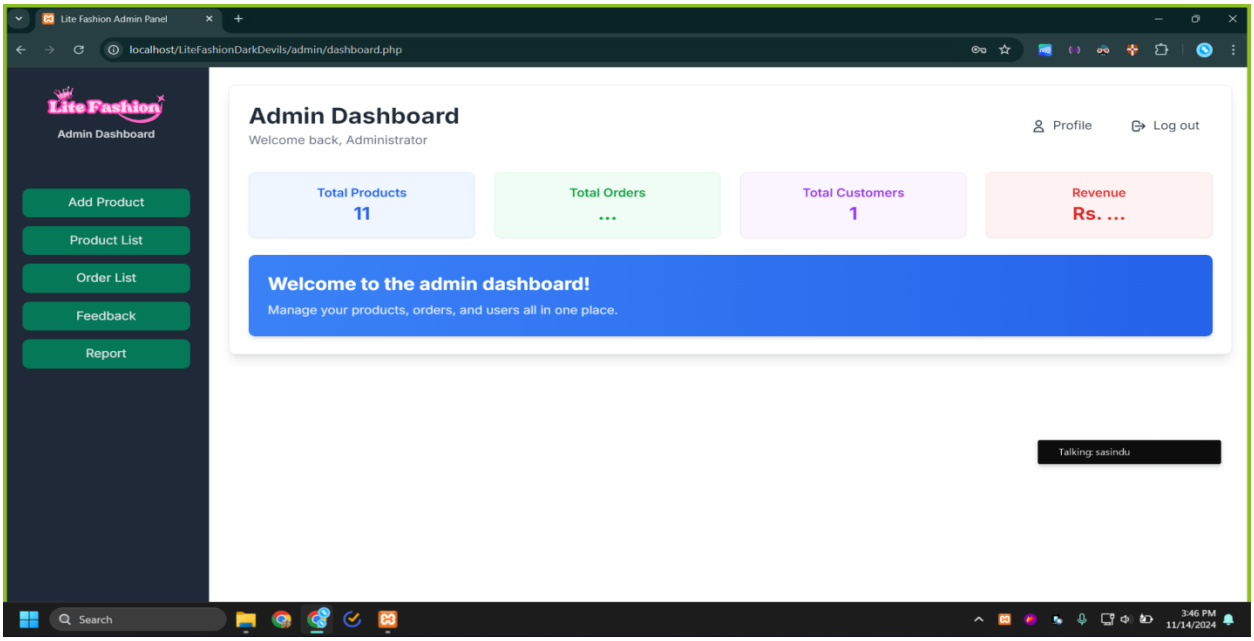

Figure 17 5.2.1 Admin Dashboard

5.2.2 Add New Product

Add New Product

This section allows admins to add new products to the system, ensuring that they are available for customers.

Key Features:

Input Fields:

- Product Name
- Description
- Price
- Quantity/Stock
- Images (upload feature)

- Product Category (dropdown or tag selection)
- SKU (Stock Keeping Unit) or product code

Validation:

- Mandatory fields check (e.g., Name, Price, Category).
- Image format and size validation.
- Save/Reset Options:
- Save the product or reset the input fields for corrections.

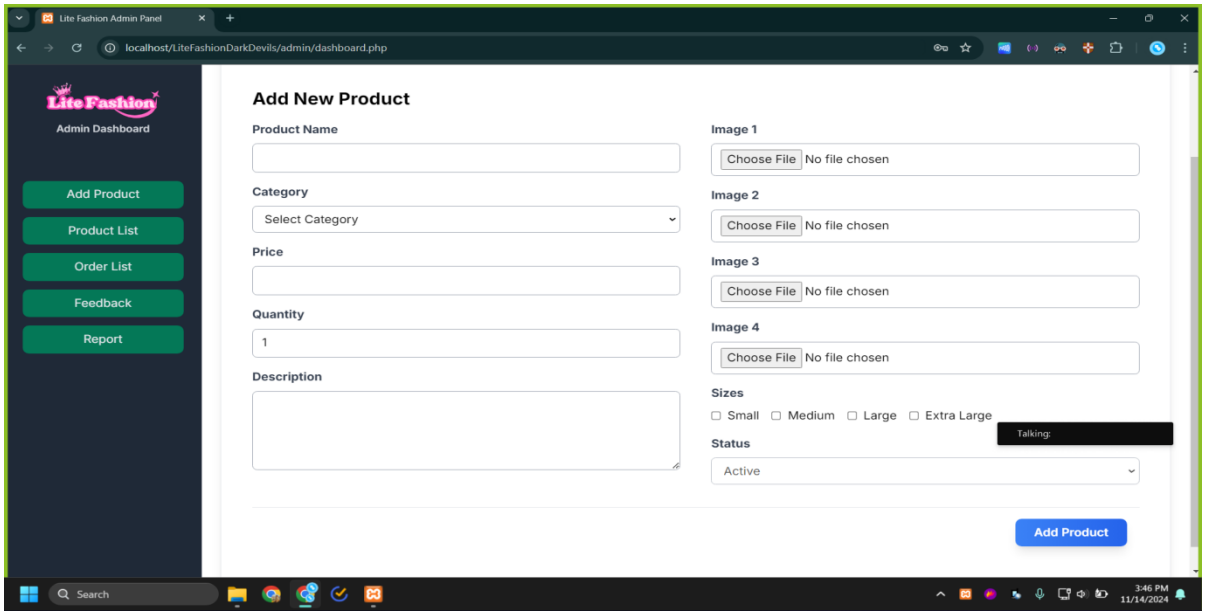

Figure 18 Add New Product

5.2.3 Product Dashboard

The Product Dashboard is used to view, search, edit, and delete products in the system.

Key Features:

Search and Filter:

- By name, category, price, or stock status.

Table View:

- Columns for product details (e.g., Name, Category, Price, Stock Quantity).

- Actions (e.g., Edit, Delete, View Details).

Batch Actions:

- Bulk edits or deletions.

Product Status Indicators:

- Highlight low-stock or out-of-stock products.

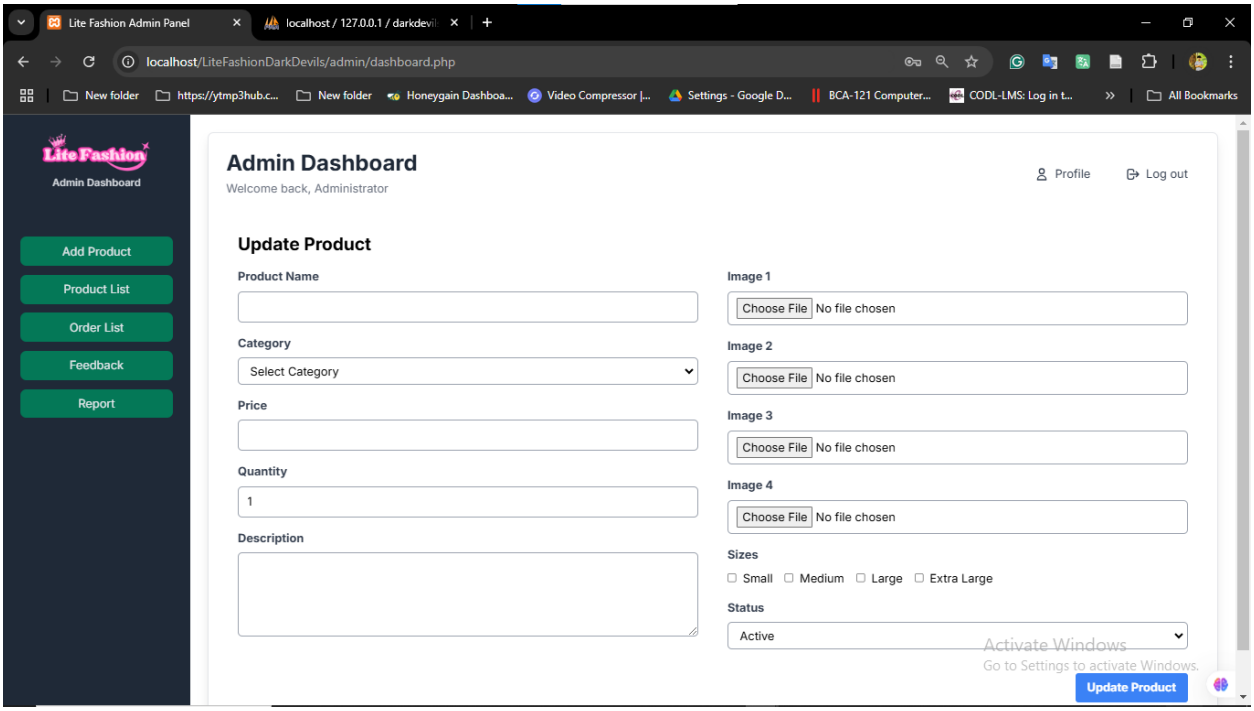

Figure 19 Product Dashbord

5.2.4 Order List

The Order List section allows administrators to view, manage, and track customer orders within the system. It serves as a centralized location to handle all order-related activities, ensuring efficient order processing and fulfillment.

Key Features:

1. Order Overview

- Display all orders in a tabular format, with columns such as:

- Order ID: Unique identifier for each order.
- Customer Name: Name of the customer who placed the order.
- Order Date: Date and time when the order was placed.
- Order Status: Current status (e.g., Pending, Processing, Shipped, Delivered, Canceled).
- Total Amount: Total cost of the order.
- Payment Status: Paid or Unpaid.

2. Search and Filter Options

- Search: Look up orders by:
- Order ID
- Customer Name
- Date Range
- Filters: Refine the list by:
- Order Status
- Payment Status
- Delivery Method (e.g., Pickup, Courier)

3. Order Details View

- Clicking on an order should open a detailed view with:
- Customer Information:
- Name, Address, Contact Information.
- Product List:
- Items in the order, including quantity, unit price, and subtotal.
- Payment Details:
- Payment Method (e.g., Credit Card, Cash on Delivery).
- Transaction ID (if applicable).
- Delivery Information:
- Shipping address, courier details, and estimated delivery date.

4. Bulk Actions

- Ability to perform actions on multiple orders at once, such as:
- Update Status (e.g., mark as shipped).
- Generate invoices.

5. Notifications and Alerts

- Highlight overdue or urgent orders.
- Notify admins of canceled or returned orders.

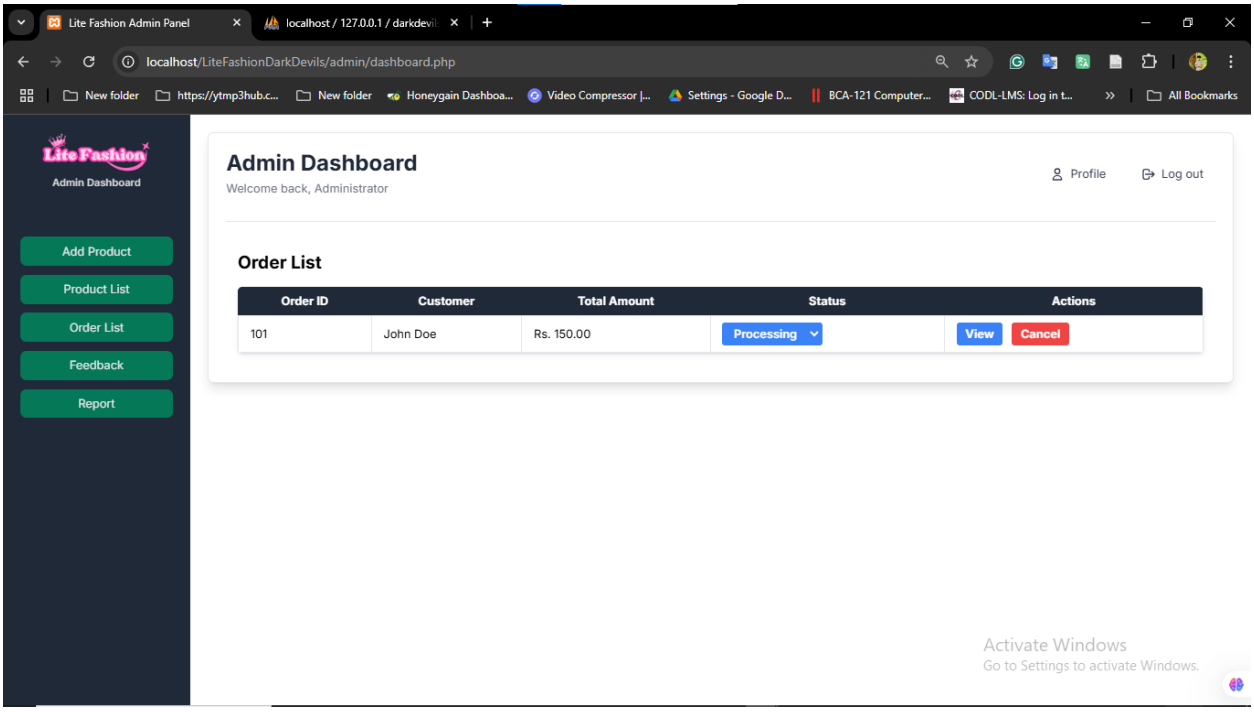

Figure 20 Order List

5.2.5 Admin Profile

The **Admin Profile** section allows administrators to manage their account details, view system privileges, and update personal information. This section is crucial for ensuring account security and customization of preferences.

Key Features:

1. Admin Information Display

- Display key details about the admin account, such as:
- **Name**: Full name of the admin.
- **Username/Email**: Unique identifier for login.
- **Role**: Admin role or access level (e.g., Super Admin, Manager).
- **Contact Information**: Phone number, secondary email (if applicable).
- **Profile Picture**: Option to upload or update a photo.

2. Edit Profile

- Allow administrators to update personal information:

- Name.
- Email or phone number.
- Profile picture.
- Provide validation for fields to ensure data integrity (e.g., valid email formats).

3. Activity Log

- View a log of recent actions performed by the admin, including:
- Last login time and IP address.
- Changes made to the system (e.g., product additions, order updates).
- Security alerts (e.g., failed login attempts, password changes).

4. Notifications and Preferences

- Enable admins to customize notifications:
- Email alerts for specific actions (e.g., low stock, new orders).
- Dashboard notifications (e.g., system updates, critical issues).
- Theme preferences (light/dark mode).

5. Logout Option

- Provide a secure way to log out of the system, with a confirmation prompt to prevent accidental logout.

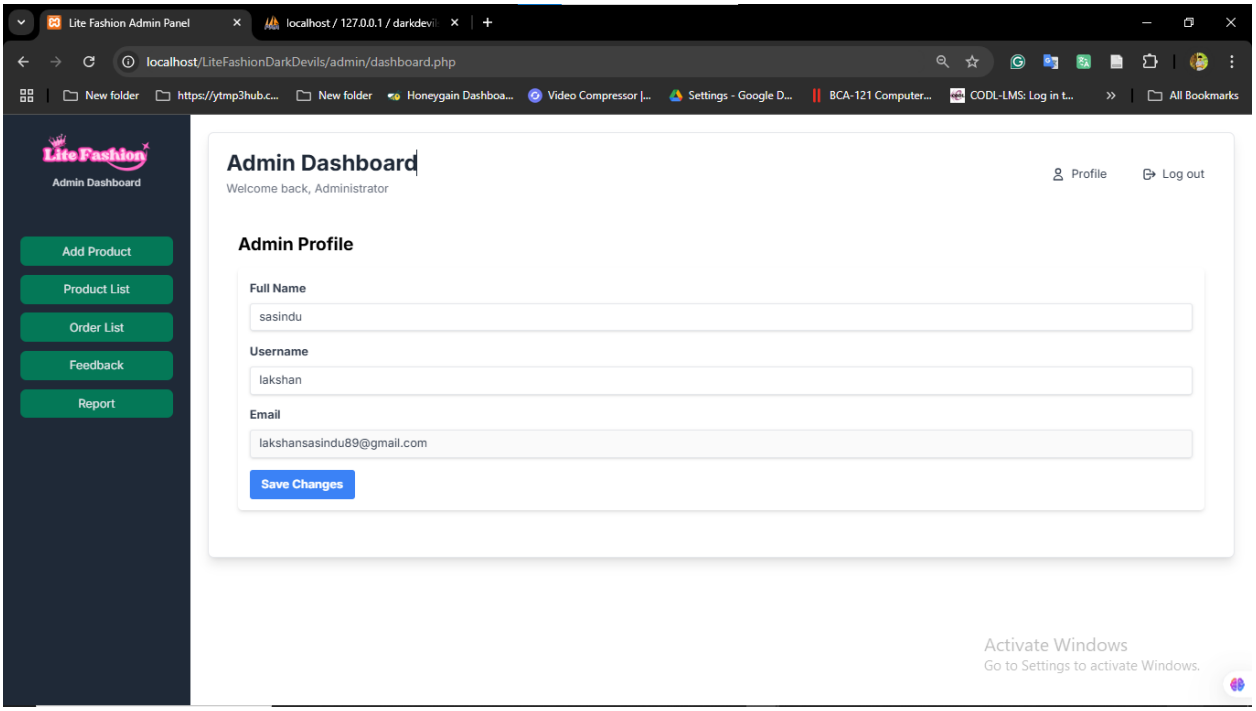

Figure 21 Admin Profile

Chapter 6 – Testing and Evaluation

6.1 Introduction

Details about the proposed system's implementation process are revealed in the fifth chapter. Several user interfaces for the suggested system are shown. A few of the front-end and back-end interfaces of the suggested system are shown in the aforementioned chapter. Additionally, it shows the system's front end and back end code segments.

This chapter describes the proposed system's testing and evaluation phases. We have selected unit testing, black box testing, security testing, integration testing, and system testing as the testing types. The test cases for the proposed system are revealed in this chapter.

6.2 Types of Software Testing

6.2.1 Unit Testing

A unit test is a way of testing a unit - the smallest piece of code that can be logically isolated in a system. That is a method, a function, a subroutine, or a property in the majority of programming languages. The definition's isolated portion is crucial. Michael Feathers, the author of "Working Effectively with Legacy Code," claims that such tests are not unit tests if they depend on external systems. "If it requires system configuration, can't be run concurrently with any other test, talks to the database, talks across the network, touches the file system.

6.2.2 Back Box Testing

Black box testing is a technique for testing software applications' functionalities without having access to their internal code structure, implementation specifics, or internal paths. Black Box Testing is entirely based on software requirements and specifications and primarily concentrates on the input and output of software applications. Additionally called behavioral testing.

6.2.3 Security Testing

Security testing is a type of software testing that uncovers vulnerabilities of the system and determines whether the data and resources of the system are protected from possible intruders. It ensures that the software system and application are free from any threats or risks that can cause a loss. Security testing of any system is focused on finding all possible loopholes and weaknesses of the system which might result in the loss of information or repute of the organization.

6.2.4 Integration Testing

Integration testing, also known as integration and testing (I&T), is a type of software testing in which the different units, modules, or components of a software application are tested as a combined entity. However, these modules may be coded by different programmers.

6.2.5 System Testing

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System testing is defined as a series of different tests whose sole purpose is to exercise the full computer-based system.

6.3 Test Cases

6.4 Summary

This chapter explains how testing and evaluation are carried out using a variety of testing techniques, including unit testing, black box testing, security testing, and system testing. Each testing technique illustrates how the testing is carried out using various project sections and how the final results are presented. The project's executive summary and plans for implementing the suggested system to further improve the hotel will be covered in the following chapter.

Chapter 7: Conclusion and Future Work

7.1 Introduction

In the previous chapter, we detailed the testing and evaluation phases of the proposed system, including the application of unit testing, black-box testing, security testing, integration testing, and system testing. Test cases for the proposed web-based clothing ordering system were also outlined.

This chapter focuses on summarizing the key outcomes of the project and exploring future enhancements for the system. The Lite Fashion Store's website has been successfully developed as a public platform, accessible to a wide audience. Ensuring an appealing design that satisfies the diverse preferences of customers was a significant challenge that was successfully addressed.

7.2 Conclusion

The web-based clothing ordering system for Lite Fashion Store was developed to provide a seamless and efficient shopping experience for customers while supporting the operational needs of the store. The system allows users to browse available clothing items, select products, and place orders online. By enabling customers to shop from the convenience of their homes, the system eliminates the need for physical visits, thus saving time and effort for both the customers and the store staff.

The platform also incorporates features such as real-time inventory management, secure payment gateways, and order tracking. These functionalities help the store maintain accurate stock records and streamline order fulfillment processes. Furthermore, the system's user-friendly interface enhances customer satisfaction by offering a visually appealing and intuitive shopping experience.

The implementation of this system has the potential to increase sales, improve customer retention, and strengthen the Lite Fashion Store's market presence. By embracing digital transformation, the store can adapt to evolving consumer behaviors and gain a competitive edge in the retail clothing industry.

7.3 Future Work

While the current iteration of OCSS fulfills its primary objectives, there are several opportunities for further enhancement and expansion:

Integration of Advanced AI Features:

- Extend recommendation methods using machine learning approaches for users' activity analysis.
- Extend the current application of virtual fitting room with augmented reality (AR) technology to make the fitting as real as possible.

Mobile Application Development:

- Increase the audience by developing specific mobile applications for Android and iOS operating systems.

Global Market Expansion:

- Extend the system to cover support for multiple languages and multiple currencies in view of global markets.

Advanced Analytics:

- To enhance the outcome of the complex sales strategies, it is highly advisable to introduce business intelligence tools to profoundly analyze the major peculiarities of sale's performance and future tendencies.

Blockchain Integration:

- Adopt blockchain to enable secure payment methods as well as to overhaul transparency in tracking products.

Sustainability Features:

- Supplement by integrating environment friendly tools like carbon footprint of products and promote sustainable choice.

7.4 Challenges Faced

During the development of the web-based clothing ordering system for Lite Fashion Store, several challenges were encountered. These challenges included:

1. Lack of Knowledge in Development Tools and Languages

The development team initially faced a significant hurdle due to a limited understanding of the tools and programming languages required for building the system. To overcome this, extensive self-learning was undertaken through online tutorials, developer forums, technical documentation, and books. This proactive approach helped the team acquire the necessary skills to complete the project.

2. Designing a User-Friendly Interface

Creating a visually appealing and intuitive user interface that caters to the diverse preferences of customers posed a challenge. Multiple iterations of design prototypes were tested and refined based on user feedback to ensure customer satisfaction.

3. Integrating Real-Time Inventory Management

Implementing a robust inventory management system that accurately reflects the stock in real-time was technically challenging. The integration of this feature required meticulous planning and testing to ensure synchronization between the database and the website.

4. Ensuring Security and Data Privacy

Building a secure platform to protect customer information, including payment details, was another critical challenge. Implementing secure payment gateways and ensuring compliance with data protection standards required additional research and testing.

5. Time and Resource Constraints

Balancing the scope of the project with available time and resources was a recurring challenge. Prioritizing tasks, setting realistic milestones, and efficient project management helped address this issue.

Despite these challenges, the team successfully delivered a functional and efficient web-based ordering system. These experiences have not only contributed to the successful completion of the project but have also enhanced the team's technical and problem-solving skills.

7.5 Summary

The development of the Web-Based Clothes Ordering System for Lite Fashion Store provided an invaluable learning experience, offering exposure to the entire software development lifecycle—from feasibility studies to the completion of the project. This journey allowed the application of critical theories and technologies learned during the BIT degree program in a practical context.

Initially, there was limited clarity on how to approach a project of this scope. However, by adhering to the university’s structured guidelines and processes, a clear path to success was established. The project emphasized the importance of systematic planning and time management, enabling daily activities to align efficiently with the project timeline.

Through this endeavor, technical skills were sharpened, intellectual abilities were enhanced, and a deeper understanding of professional software development practices was acquired. This project not only provided theoretical and practical insights but also prepared the team to tackle real-world challenges in future endeavors confidently.

6.3 Contributions

The OCSS project contributes to both the academic and professional realms in several significant ways:

Technical Contributions:

- Shows how modern Web technologies, backend frameworks, and database systems can be utilized in the creation of a large-scale e-business system.
- Includes new features such as an AI fitting room and list of inspiring recommendations for further use challenging the user experience.

Business Value:

- Serves as an effective, customer-oriented solution focusing on outline issues in online clothing selling, including order processing and user engagement.
- Helps the further development of the segment of fashion retail trade by offering small companies to use progressive e-commerce platforms.

Educational Impact:

- Being a sample application, it is the perfect learning resource for learning about the development of a contemporary e-business solution from design to implementation.

In conclusion I will say that OCSS has set the ground for the kind of change that the online retail industry. Because it confronts current difficulties and successfully assimilates future possibilities, it is ripe for prolonged stability and creativity.

Appendix A

For more user interfaces refer Appendix A

Code Segments

1. Front End

Customer Interface

```
<?php
require_once('./includes/config.php');
?>
<?php
```

```php
$categories1 = $connection->query("select * from clothProduct where P_status='1'
and P_categoryId='C001' LIMIT 4");
$categories1->execute();
$menCloth = $categories1->fetchAll(PDO::FETCH_OBJ);

$categories2 = $connection->query("select * from clothProduct where P_status='1'
and P_categoryId='C002' LIMIT 4");
$categories2->execute();
$womenCloth = $categories2->fetchAll(PDO::FETCH_OBJ);

$categories3 = $connection->query("select * from clothProduct where P_status='1'
and P_categoryId='C003' LIMIT 4");
$categories3->execute();
$kidsCloth = $categories3->fetchAll(PDO::FETCH_OBJ);
?>


<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Lite Fashion</title>
  <link
    rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.6.0/css/all.min.css" />
  <link rel="stylesheet" href="layout/css/styles.css" />
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script src="https://cdn.tailwindcss.com"></script>
</head>

<body>
  <!-- navbar -->
  <?php include_once 'includes/navbar.php' ?>

  <!-- hero section -->
  <section
    id="hero"
    class="flex flex-col py-0 px-20 flex-start justify-center">
    <h4 class="text-lg font-semibold mb-2 ml-1 text-slate-700">
      Unbeatable Offers
    </h4>
    <h2 class="text-5xl font-semibold">Super value deals</h2>
```

```
    <h1 class="text-6xl font-semibold text-sky-500">Across All Products</h1>
    <p>Maximize your savings with coupons and huge discounts!</p>
    <a
      href="pages/list_products.php"
      class="flex justify-start shop-button text-sky-500 hover:text-sky-400 font-
semibold cursor-pointer mt-5">
      Shop Now
    </a>
  </section>

  <!-- feature section -->
  <section
    id="feature"
    class="flex py-20 px-20 gap-10 justify-between text-sky-500 bg-slate-200/40
rounded-sm font-semibold">
    <div class="feature-box">
      <img src="layout/images/features/f1.png" alt="" class="p-5" />
      <p class="text-sky-500 bg-sky-200/40 rounded-sm font-semibold">
        Free Shipping
      </p>
    </div>
    <div class="feature-box">
      <img src="layout/images/features/f2.png" alt="" class="p-5" />
      <p class="text-sky-500 bg-sky-200/40 rounded-sm font-semibold">
        Online Order
      </p>
    </div>
    <div class="feature-box">
      <img src="layout/images/features/f3.png" alt="" class="p-5" />
      <p class="text-sky-500 bg-sky-200/40 rounded-sm font-semibold">
        Save Money
      </p>
    </div>
    <div class="feature-box">
      <img src="layout/images/features/f4.png" alt="" class="p-5" />
      <p class="text-sky-500 bg-sky-200/40 rounded-sm font-semibold">
        Promotions
      </p>
    </div>
    <div class="feature-box">
      <img src="layout/images/features/f5.png" alt="" class="p-5" />
      <p class="text-sky-500 bg-sky-200/40 rounded-sm font-semibold">
        Happy Sell
      </p>
    </div>
```

```
    <div class="feature-box">
      <img src="layout/images/features/f6.png" alt="" class="p-5" />
      <p class="text-sky-500 bg-sky-200/40 rounded-sm font-semibold">
        24/7 Service
      </p>
    </div>
  </section>



  <!-- men products section -->
  <section id="feature-products" class="py-8 px-20 mt-5">
    <h2 class="text-5xl font-bold  text-center">Men Category</h2>
    <p class="text-lg text-slate-500  text-center">Summer Collection New Modern
Design</p>
    <div class="flex justify-between py-5 flex-wrap">
      <?php foreach ($menCloth as $men) : ?>
        <div
          class="w-[23%] min-w-64 py-2 px-3 border-2 rounded-md shadow-md mt-4
mx-0 cursor-pointer hover:shadow-xl"
          onclick="window.location.href='./pages/single_product.php?id=<?php echo
($men->ProductId); ?>'">
          <img
src="http://localhost/LiteFashionDarkDevils/user/layout/images/products/<?php
echo $men->P_image1; ?>" alt="" />
          <div class="flex justify-between py-3 relative">
            <div class="px-2">
              <span class="text-lg text-slate-500">adidas</span>
              <p class="text-xl font-bold"><?php echo $men->P_name; ?></p>
              <div class="text-yellow-400 text-sm">
                <i class="fa-solid fa-star"></i>
                <i class="fa-solid fa-star"></i>
                <i class="fa-solid fa-star"></i>
                <i class="fa-solid fa-star"></i>
              </div>
              <p class="text-red-400 font-bold"><?php echo $men->P_price; ?></p>
            </div>
            <div>
              <a href="#">
                <div
                  class="h-11 w-11 bg-slate-200 rounded-full flex items-center
justify-center absolute right-2 bottom-4">
                  <i
                    class="fa-solid fa-cart-arrow-down text-xl text-sky-500"></i>
                </div>
```

```
            </a>
          </div>
        </div>
      </div>
    <?php endforeach; ?>
  </div>
</section>

<!-- women products section -->
<section id="feature-products" class="py-8 px-20 mt-5">  text-center
  <h2 class="text-5xl font-bold  text-center">Women Category</h2>
  <p class="text-lg text-slate-500  text-center">Summer Collection New Modern
Design</p>
  <div class="flex justify-between py-5 flex-wrap">
    <?php foreach ($womenCloth as $women) : ?>
      <div
        class="w-[23%] min-w-64 py-2 px-3 border-2 rounded-md shadow-md mt-4
mx-0 cursor-pointer hover:shadow-xl"
        onclick="window.location.href='./pages/single_product.php?id=<?php echo
($women->ProductId); ?>'">
        <img
src="http://localhost/LiteFashionDarkDevils/user/layout/images/products/<?php
echo $women->P_image1; ?>" alt="" />
        <div class="flex justify-between py-3 relative">
          <div class="px-2">
            <span class="text-lg text-slate-500">adidas</span>
            <p class="text-xl font-bold"><?php echo $women->P_name; ?></p>
            <div class="text-yellow-400 text-sm">
              <i class="fa-solid fa-star"></i>
              <i class="fa-solid fa-star"></i>
              <i class="fa-solid fa-star"></i>
              <i class="fa-solid fa-star"></i>
            </div>
            <p class="text-red-400 font-bold"><?php echo $women->P_price;
?></p>
          </div>
          <div>
            <a href="#">
              <div
                class="h-11 w-11 bg-slate-200 rounded-full flex items-center
justify-center absolute right-2 bottom-4">
                <i
                  class="fa-solid fa-cart-arrow-down text-xl text-sky-500"></i>
              </div>
            </a>
```

```
            </div>
          </div>
        </div>
      <?php endforeach; ?>
    </div>
  </section>

  <!-- kids products section -->
  <section id="feature-products" class="py-8 px-20 mt-5">
    <h2 class="text-5xl font-bold  text-center">Kids Category</h2>
    <p class="text-lg text-slate-500  text-center">Summer Collection New Modern
Design</p>
    <div class="flex justify-between py-5 flex-wrap">
      <?php foreach ($kidsCloth as $kids) : ?>
        <div
          class="w-[23%] min-w-64 py-2 px-3 border-2 rounded-md shadow-md mt-4
mx-0 cursor-pointer hover:shadow-xl"
          onclick="window.location.href='./pages/single_product.php?id=<?php echo
($kids->ProductId); ?>'">
          <img
src="http://localhost/LiteFashionDarkDevils/user/layout/images/products/<?php
echo $kids->P_image1; ?>" alt="" />
          <div class="flex justify-between py-3 relative">
            <div class="px-2">
              <span class="text-lg text-slate-500">adidas</span>
              <p class="text-xl font-bold"><?php echo $kids->P_name; ?></p>
              <div class="text-yellow-400 text-sm">
                <i class="fa-solid fa-star"></i>
                <i class="fa-solid fa-star"></i>
                <i class="fa-solid fa-star"></i>
                <i class="fa-solid fa-star"></i>
              </div>
              <p class="text-red-400 font-bold"><?php echo $kids->P_price; ?></p>
            </div>
            <div>
              <a href="#">
                <div
                  class="h-11 w-11 bg-slate-200 rounded-full flex items-center
justify-center absolute right-2 bottom-4">
                  <i
                    class="fa-solid fa-cart-arrow-down text-xl text-sky-500"></i>
                </div>
              </a>
            </div>
          </div>
```

```
        </div>
      <?php endforeach; ?>
    </div>
  </section>


  <!-- banner -->
  <section id="banner" class="py-8 px-20">
    <div
      class="flex flex-col text-center items-center text-white bg-sky-400/90 py-7
px-8 rounded-md">
      <p class="text-xl mb-1">Fast Delivery</p>
      <h2 class="text-5xl font-bold mb-2">
        Up to <span>60% Off</span> - All Fashions
      </h2>
      <a
        href="http://localhost/LiteFashionDarkDevils/user/pages/list_products.php
"
        class="py-2 px-4 border-2 font-semibold rounded-md hover:text-sky-400
hover:bg-white/90">
        Explore More
      </a>
    </div>
  </section>

  <!-- footer -->
  <?php include_once 'includes/footer.php' ?>

  <script src="layout/js/script.js"></script>
</body>

</html>
```

Customer Registration

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sign Up</title>
    <script src="https://cdn.tailwindcss.com"></script>
</head>
```

```
<body class="flex justify-center bg-gray-100">
    <div class="popup-contentLog bg-white rounded-xl shadow-lg w-1/3 my-10">
        <div class="p-10 space-y-4">
            <h1 class="text-2xl font-bold leading-tight tracking-tight text-gray-
900 text-center">
                Create a New Account
            </h1>
            <form class="space-y-4 md:space-y-6" method="POST"
action="signup_data_send.php">

                <!-- Full Name -->
                <div>
                    <label for="fname" class="block mb-2 text-sm font-medium
text-gray-900">Full Name</label>
                    <input type="text" name="fullName" id="fullName" class="bg-
gray-50 border border-gray-300 text-gray-900 rounded-lg focus:ring-primary-600
focus:border-primary-600 block w-full p-2.5" placeholder="Enter your full name"
required>
                </div>

                <!-- Username -->
                <div>
                    <label for="lname" class="block mb-2 text-sm font-medium
text-gray-900">Username</label>
                    <input type="text" name="userName" id="userName" class="bg-
gray-50 border border-gray-300 text-gray-900 rounded-lg focus:ring-primary-600
focus:border-primary-600 block w-full p-2.5" placeholder="Enter your username"
required>
                </div>

                <!-- Email -->
                <div>
                    <label for="email" class="block mb-2 text-sm font-medium
text-gray-900">Email Address</label>
                    <input type="email" name="email" id="email" class="bg-gray-50
border border-gray-300 text-gray-900 rounded-lg focus:ring-primary-600
focus:border-primary-600 block w-full p-2.5" placeholder="name@company.com"
required>
                </div>

                <!-- Password -->
                <div>
                    <label for="password" class="block mb-2 text-sm font-medium
text-gray-900">Password</label>
```

```
                    <input type="password" name="password" id="password"
class="bg-gray-50 border border-gray-300 text-gray-900 rounded-lg focus:ring-
primary-600 focus:border-primary-600 block w-full p-2.5" placeholder="••••••••"
required>
                </div>

                <!-- Submit Button -->
                <button type="submit" name="submitACC" class="w-full text-white
bg-gray-600 hover:bg-primary-700 focus:ring-4 focus:outline-none focus:ring-
primary-300 font-medium rounded-lg text-sm px-5 py-2.5 text-center">Create
Account</button>

                <!-- Redirect to Sign In -->
                <p class="text-sm font-light text-gray-500 text-center mt-5">
                    Already have an account?
                    <a href="http://localhost/LiteFashionDarkDevils/user/"
class="font-medium text-primary-600 hover:underline">Sign In</a>
                </p>
            </form>
        </div>
    </div>
</body>

</html>
```

Products Interface

```
<?php
require_once('../includes/config.php');

?>
<!DOCTYPE html>
<html lang="en">

<?php
include_once('../includes/header.php');
?>

<body>
  <!-- Navbar -->
  <?php include_once('../includes/navbar.php'); ?>

  <!-- Page header -->
```

```
  <section id="page-header-products" class="flex flex-col py-0 px-20 text-center
justify-center">
  </section>

  <!-- Title -->
  <div class="text-center pt-8 pb-4">
    <h2 class="text-5xl font-bold">ALL Products</h2>
    <p class="text-lg text-slate-500">Summer Collection New Modern Design</p>
  </div>

  <!-- Main Content Section -->
  <div class="flex py-8 px-20 space-x-8">

    <!-- Filter Sidebar -->
    <aside class="w-1/5 p-6 bg-gray-100 rounded-lg shadow-md">
      <!-- Search Form -->
      <form id="search-form" class="mb-6">
        <input
          type="text"
          id="search-input"
          class="w-full p-3 rounded-md border border-gray-300 focus:outline-none
focus:ring-2 focus:ring-sky-500"
          placeholder="Search products..." />
      </form>

      <!-- Category Filters -->
      <div>
        <h5 class="text-xl font-semibold mb-4 pl-2">Categories</h5>
        <div id="category-buttons" class="space-y-2">
          <button class="filter-btn w-full text-center font-bold text-left p-2
rounded-md text-gray-800 bg-gray-200 hover:bg-gray-300" data-
category="all">All</button>

          <?php

          $categories = $connection->query("
        SELECT DISTINCT c.CategoryId, c.C_name
        FROM clothProduct p
        JOIN category c ON p.P_categoryId = c.CategoryId
        WHERE p.P_status = 1
    ");

          while ($row = $categories->fetch(PDO::FETCH_ASSOC)) {
```

```
            echo "<button class='filter-btn text-center font-bold w-full text-
left p-2 rounded-md text-gray-800 bg-gray-200 hover:bg-gray-300' data-
category='{$row['CategoryId']}'>
                {$row['C_name']}
              </button>";
          }
          ?>
        </div>

      </div>
    </aside>

    <!-- Products Section -->
    <section id="products" class="flex-1">

      <!-- Sort Dropdown -->
      <div class="flex justify-end mb-6">
        <select id="sort-options" class="p-2 border rounded-md">
          <option value="default">Sort by: Default</option>
          <option value="low-to-high">Price: Low to High</option>
          <option value="high-to-low">Price: High to Low</option>
        </select>
      </div>

      <!-- Products Container -->
      <div id="products-container" class="grid grid-cols-1 sm:grid-cols-2
md:grid-cols-3 lg:grid-cols-4 gap-6">
        <?php
        $query = "SELECT * FROM clothProduct WHERE P_status = 1";
        $products = $connection->query($query);
        foreach ($products as $product):
        ?>
          <div
            class="p-4 border rounded-md shadow hover:shadow-lg transition
duration-200 cursor-pointer"
            onclick="window.location.href='single_product.php?id=<?php echo
$product['ProductId']; ?>'">
            <img src="http://localhost/LiteFashionDarkDevils/admin/uploads/<?php
echo $product['P_image1']; ?>" alt="<?php echo $product['P_name']; ?>" class="w-
full h-64 object-cover rounded-md" />
            <div class="mt-4">
              <span class="block text-gray-600"><?php echo
$product['P_categoryId']; ?></span>
              <h3 class="text-lg font-semibold"><?php echo $product['P_name'];
?></h3>
```

```
            <p class="text-red-400 font-bold">Rs. <?php echo
$product['P_price']; ?></p>
          </div>
        </div>
      <?php endforeach; ?>
    </div>
  </section>
</div>

<!-- Footer -->
<?php include_once '../includes/footer.php'; ?>

<script src="../layout/js/script.js"></script>
<script>
  document.addEventListener('DOMContentLoaded', () => {
    const searchInput = document.getElementById('search-input');
    const categoryButtons = document.querySelectorAll('.filter-btn');
    const sortOptions = document.getElementById('sort-options');

    function fetchProducts(queryParams = '') {
      const xhr = new XMLHttpRequest();
      xhr.open('GET', `fetch_products.php?${queryParams}`, true);
      xhr.onload = function() {
        if (this.status === 200) {
          document.getElementById('products-container').innerHTML =
            this.responseText;
        }
      };
      xhr.send();
    }

    searchInput.addEventListener('input', () => {
      const searchTerm = searchInput.value;
      fetchProducts(`search=${searchTerm}`);
    });

    categoryButtons.forEach((button) => {
      button.addEventListener('click', () => {
        const category = button.getAttribute('data-category');
        fetchProducts(`category=${category}`);
      });
    });

    sortOptions.addEventListener('change', () => {
      const sortOrder = sortOptions.value;
```

```
        fetchProducts(`sortOrder=${sortOrder}`);
      });
    });
  </script>
</body>

</html>
```

Product Details Interface

```
<?php
require_once('../includes/config.php');
?>
<?php

if (isset($_POST['submit'])) {
  $userid = trim($_POST['U_id']);
  $clothid = trim($_POST['P_id']);
  $name = trim($_POST['P_name']);
  $image = trim($_POST['P_image']);
  $qty = trim($_POST['P_qty']);
  $price = trim($_POST['P_price']);
  $size = trim($_POST['P_size']);

  if (empty($userid)) {
    echo "first login";
  } else {
    $insertdata = $connection->prepare("insert into
cart(ProductId,CustermerId,P_name,P_price,P_image1,S_qty,S_size)
          values(:ProductId, :CustermerId, :P_name, :P_price, :P_image1 , :S_qty
, :S_size)");

    $insertdata->execute([
      ":ProductId" => $clothid,
      ":CustermerId" => $userid,
      ":P_name" => $name,
      ":P_price" => $price,
      ":P_image1" => $image,
      ":S_qty" => $qty,
      ":S_size" => $size
    ]);
  }
}

if (isset($_GET['id'])) {
```

```
  $productID = trim($_GET['id']);
  $singleProduct = $connection->query("select * from clothProduct where
P_status=1 and ProductId='$productID'");
  $singleProduct->execute();
  $oneClothData = $singleProduct->fetch(PDO::FETCH_OBJ);


  $reletedCloth = $connection->query("select * from clothProduct where P_status=1
and P_categoryId='$oneClothData->P_categoryId' and ProductId != '$productID'
LIMIT 4");
  $reletedCloth->execute();
  $allReletedCloth = $reletedCloth->fetchAll(PDO::FETCH_OBJ);
} else {
}

?>

<!DOCTYPE html>
<html lang="en">

<!-- header -->
<?php include_once '../includes/header.php' ?>

<body>
  <!-- navbar -->
  <?php include_once '../includes/navbar.php' ?>

  <?php
  if (isset($_GET['id'])) {
    if (isset($_SESSION['customerId'])) {
      $isCartAdded = $connection->query("select * from cart where
ProductId='{$_GET['id']}' and CustermerId='{$_SESSION['customerId']}'");
      $isCartAdded->execute();
    }
  }
  ?>

  <!-- product details section -->
  <section id="product-details" class="py-8 px-40 mt-5 flex">
    <div class="w-[40%] mr-12">
      <img
        src="http://localhost/LiteFashionDarkDevils/admin/uploads/<?php echo
$oneClothData->P_image1; ?>"
        width="100%"
        class="mb-1"
```

```
        id="main-image" />
      <div class="flex justify-between gap-1">
        <img
          src="http://localhost/LiteFashionDarkDevils/admin/uploads/<?php echo
$oneClothData->P_image1; ?>"
          width="100%"
          class="cursor-pointer small-image" />

        <img
          src="http://localhost/LiteFashionDarkDevils/admin/uploads/<?php echo
$oneClothData->P_image2; ?>"
          width="100%"
          class="cursor-pointer small-image" />

        <img
          src="http://localhost/LiteFashionDarkDevils/admin/uploads/<?php echo
$oneClothData->P_image3; ?>"
          width="100%"
          class="cursor-pointer small-image" />

        <img
          src="http://localhost/LiteFashionDarkDevils/admin/uploads/<?php echo
$oneClothData->P_image4; ?>"
          width="100%"
          class="cursor-pointer small-image" />
      </div>
    </div>
    <!-- details -->
    <div class="w-[50%]">
      <p class="text-sm text-slate-600 font-semibold pt-5 pb-3">
        Classic T-Shirt
      </p>
      <p class="text-5xl font-bold"><?php echo $oneClothData->P_name; ?></p>
      <h2 class="text-3xl font-bold pb-3 text-red-600">Rs. <?php echo
$oneClothData->P_price; ?></h2>
      <form method="POST" id="form-data">

        <input class="form-control" type="hidden" name="P_id" value="<?php echo
$oneClothData->ProductId; ?>" required><br>
        <input class="form-control" type="hidden" name="U_id" value="<?php echo
((isset($_SESSION['username'])) ? $_SESSION['customerId'] : ''); ?>"
required><br>
        <input class="form-control" type="hidden" name="P_name" value="<?php echo
$oneClothData->P_name; ?>" required><br>
```

```
        <input class="form-control" type="hidden" name="P_price" value="<?php
echo $oneClothData->P_price; ?>" required><br>
        <input class="form-control" type="hidden" name="P_image" value="<?php
echo $oneClothData->P_image1; ?>" required><br>
        <select name="P_size"
          class="form-control block py-1 px-5 mb-2 bg-sky-100 border-2 border-
sky-300 rounded-sm mb-4">
          <?php if ($oneClothData->P_small == 1) : ?>
            <option value="Small">Small</option>
          <?php endif; ?>
          <?php if ($oneClothData->P_medium) : ?>
            <option value="Medium">Medium</option>
          <?php endif; ?>
          <?php if ($oneClothData->P_large) : ?>
            <option value="Large">Large</option>
          <?php endif; ?>
          <?php if ($oneClothData->P_extraLarge) : ?>
            <option value="Extra Large">Extra Large</option>
          <?php endif; ?>

        </select>
        <input type="number" name="P_qty" value="<?php echo $oneClothData-
>P_quantity; ?>"
          class="form-control w-12 focus:outline-none border-slate-400 rounded-sm
border-2 p-1 mr-3" required />
        <?php if (isset($_SESSION['username'])) : ?>
          <?php if ($isCartAdded->rowCount() > 0) : ?>

            <button class="btn-insert py-2 px-5 bg-sky-500 text-white rounded-sm"
type="submit" name="submit" disabled>Added to Cart</button>
          <?php else : ?>
            <button class="btn-insert py-2 px-5 bg-sky-500 text-white rounded-sm"
type="submit" name="submit">Add to Cart</button>
          <?php endif; ?>
        <?php else: ?>
          <h5 style="color: red;">Frist login website</h5>
        <?php endif; ?>

      </form>
      <h2 class="text-xl font-semibold pb-1 mt-8">Product Details</h2>
      <span class="text-slate-600"><?php echo $oneClothData->P_description;
?></span>
    </div>
  </section>
```

```
  <!-- related products section -->
  <section id="feature-products" class="py-8 px-20 mt-5">
    <div class="text-center">
      <h2 class="text-5xl font-bold">Feature Products</h2>
      <p class="text-lg text-slate-500">
        Summer Collection New Modern Design
      </p>
    </div>
    <div class="flex justify-between py-5 flex-wrap">
      <?php foreach ($allReletedCloth as $reletedCloth) : ?>
        <div
          class="w-[23%] min-w-64 py-2 px-3 border-2 rounded-md shadow-md mt-4
mx-0 cursor-pointer hover:shadow-xl"
          onclick="window.location.href='single_product.php?id=<?php echo
($reletedCloth->ProductId); ?>'">
          <img src="http://localhost/LiteFashionDarkDevils/admin/uploads/<?php
echo $reletedCloth->P_image1; ?>" alt="" />
          <div class="flex justify-between py-3 relative">
            <div class="px-2">
              <span class="text-lg text-slate-500">adidas</span>
              <p class="text-xl font-bold"><?php echo $reletedCloth->P_name;
?></p>
              <div class="text-yellow-400 text-sm">
                <i class="fa-solid fa-star"></i>
                <i class="fa-solid fa-star"></i>
                <i class="fa-solid fa-star"></i>
                <i class="fa-solid fa-star"></i>
              </div>
              <p class="text-red-400 font-bold">Rs. <?php echo $reletedCloth-
>P_price; ?></p>
            </div>
            <div>
              <a href="#">
                <div
                  class="h-11 w-11 bg-slate-200 rounded-full flex items-center
justify-center absolute right-2 bottom-4">
                  <i
                    class="fa-solid fa-cart-arrow-down text-xl text-sky-500"></i>
                </div>
              </a>
            </div>
          </div>
        </div>
      <?Php endforeach;  ?>
    </div>
```

```
  </section>

  <!-- footer -->
  <?php include_once '../includes/footer.php' ?>

  <script src="../layout/js/script.js"></script>
  <script>
    document.addEventListener("DOMContentLoaded", function() {
      // Check if elements exist for debugging
      const form = document.getElementById("form-data");
      const btnInsert = document.querySelector(".btn-insert");

      if (!form || !btnInsert) {
        console.error("Form or button element not found.");
        return;
      }

      // Prevent the user from entering 0 or less than 0 value
      document.querySelectorAll(".form-control").forEach(function(input) {
        input.addEventListener("input", function() {
          let value = input.value.replace(/^(0*)/, "");
          input.value = value || 1;
        });
      });

      // Prevent page reload on form submit
      btnInsert.addEventListener("click", function(e) {
        e.preventDefault();

        let formData = new FormData(form);
        formData.append('submit', 'submit');

        fetch("Single_product.php?id=<?php echo $_GET['id']; ?>", {
            method: "POST",
            body: formData
          })
          .then(response => response.text())
          .then(data => {
            console.log("Server response:", data);
            alert("Product added to cart");

<<<<<<< HEAD
                // Disable the add to cart button and change its text
                btnInsert.innerHTML = "<i class='addCss'></i>Added to Cart";
                btnInsert.disabled = true;
```

```
                //LoadRef();
            })
            .catch(error => console.error("Fetch error:", error));
        });
    });
    function LoadRef() {
  fetch("single_product.php")
    .then(response => response.text())
    .then(html => {
      document.body.innerHTML = html;
    })
    .catch(error => {
      console.error('Error loading content:', error);
    });
}
</script>
=======
            // Disable the add to cart button and change its text
            btnInsert.innerHTML = "<i class='addCss'></i>Added to Cart";
            btnInsert.disabled = true;
          })
          .catch(error => console.error("Fetch error:", error));
      });
    });
  </script>
>>>>>>> 7a5223b5cd2efda2a529275a0057ddc73ecfdccc

</body>

</html>
```

Cart Interface

```
<?php
require_once('../includes/config.php');
?>

<!DOCTYPE html>
<html lang="en">

<!-- header -->
<?php include_once '../includes/header.php' ?>

<body>
  <!-- navbar -->
```

```
  <?php include_once '../includes/navbar.php' ?>

  <?php

  if (!isset($_SESSION['username'])) {
    echo "<script>
window.location.href='http://localhost/LiteFashionDarkDevils/user/'; </script>";
  }

  $cartItems = $connection->query("select * from cart where
CustermerId='{$_SESSION['custormerId']}'");
  $cartItems->execute();

  $allCartItems = $cartItems->fetchAll(PDO::FETCH_OBJ);
  ?>

  <!-- page header -->
  <section
    id="page-header-about"
    class="flex flex-col py-0 px-20 text-center justify-center">
    <h1 class="text-6xl font-semibold text-white/90 p-3">Cart</h1>
    <p class="text-white/80">Add your coupon code & SAVE upto 60%</p>
  </section>

  <!-- cart items -->
  <section class="py-10 px-20">
    <div class="items-center text-center">
      <div
        class="grid grid-cols-7 px-20 py-3 mx-auto text-lg font-semibold bg-
slate-200 mb-1 text-center rounded-md">
        <p>Image</p>
        <p>Product</p>
        <p>Price</p>
        <p>Quantity</p>
        <p>Update</p>
        <p>SubTotal</p>
        <p>Remove</p>
      </div>
      <?php if (count($allCartItems) > 0) : ?>
        <?php foreach ($allCartItems as $cart) : ?>
          <div
            class="grid grid-cols-7 px-20 py-3 items-center text-center justify-
items-center">
            <img src="http://localhost/LiteFashionDarkDevils/admin/uploads/<?php
echo $cart->P_image1; ?>" alt="" class="w-20" />
```

```
            <p><?php echo $cart->P_name; ?></p>
            <p class="pro_price"><?php echo $cart->P_price; ?></p>
            <input type="number" value="<?php echo $cart->S_qty; ?>" class="
pro_qty w-12 pl-2 border" />
            <button value="<?php echo $cart->S_cartId; ?>" class="btn-update"
style="background-color: greenyellow; border-radius:10px;
padding:2px;">Save</button>
            <p class="total_price"><?php echo "RS : " . (($cart->S_qty) * ($cart-
>P_price)); ?></p>
            <button value="<?php echo $cart->S_cartId; ?>" class="btn-delete"><i
class="fa-solid fa-x text-red-600 cursor-pointer"></i></button>
          </div>
        <?php endforeach; ?>
      <?php else : ?>
        <h6 style="color: red;">no product have</h6>
      <?php endif; ?>

    </div>
  </section>

  <!-- coupon and total amount -->
  <section class="py-10 px-20 mx-20">
    <div class="flex justify-end">
      <div class="w-1/3 border-2 border-sky-400 bg-sky-100/50 p-8 rounded-lg">
        <h1 class="text-2xl font-bold pb-3">CART TOTALS</h1>
        <div class="w-[300px] text-lg text-slate-700 gap-2">
          <div class="flex justify-between">
            <p>Subtotal</p>
            <p class="full_price">Rs.0.00</p>
          </div>
          <div class="flex justify-between">
            <p>Shipping</p>
            <p>Free</p>
          </div>
          <div class="flex justify-between my-4">
            <p class="font-semibold">Total</p>
            <p class="font-semibold">Rs.0.00</p>
          </div>
          <!-- <button class="bg-red-500 text-white py-2 px-6 rounded-md mt-8">
              Proceed to Checkout
            </button> -->
          <a
            href="place_order.php"
            class="bg-red-500 text-white py-3 px-6 rounded-md mt-8">
            Proceed to Checkout
```

```
          </a>
        </div>
      </div>
    </div>
  </section>

  <!-- footer -->
  <?php include_once '../includes/footer.php' ?>

  <script src="../layout/js/script.js"></script>
  <script>
    document.querySelectorAll('.pro_qty').forEach(input => {
      input.addEventListener('mouseup', () => {
        const container = input.closest('div');

        const proQty = input.value;
        const proPrice =
parseFloat(container.querySelector('.pro_price').textContent);

        const total = proPrice * proQty;
        const totalPriceEl = container.querySelector('.total_price');
        totalPriceEl.textContent = "RS : " + total;
      });

      document.querySelectorAll('.btn-update').forEach(button => {
        button.replaceWith(button.cloneNode(true));
      });

      document.querySelectorAll('.btn-update').forEach(button => {
        button.addEventListener('click', function(e) {
          const id = this.value;
          const container = this.closest('div');
          const proQty = container.querySelector('.pro_qty').value;

          fetch("cart_update.php", {
              method: "POST",
              headers: {
                "Content-Type": "application/x-www-form-urlencoded"
              },
              body: `update=update&id=${id}&pro_qty=${proQty}`
            })
            .then(response => response.text())
            .then(data => {
              alert("Updated");
            })
```

```
            .catch(error => {
              alert('Error');
            });
        });

        document.querySelectorAll('.btn-delete').forEach(button => {
          button.replaceWith(button.cloneNode(true));
        });
        document.querySelectorAll('.btn-delete').forEach(button => {
          button.addEventListener('click', function(e) {
            const id = this.value;

            fetch("cart_delete.php", {
                method: "POST",
                headers: {
                  "Content-Type": "application/x-www-form-urlencoded"
                },
                body: `delete=delete&id=${id}`
              })
              .then(response => response.text())
              .then(data => {
                alert("Deleted");
                reload();
              })
              .catch(error => {
                alert('Error');
              });
          });

        });


      });

      calTotalPrice();

      function calTotalPrice() {
        setInterval(function() {
          let sum = 0.0;

          // Select all elements with the class 'total_price' and calculate the
sum
          document.querySelectorAll('.total_price').forEach(element => {
            sum += parseFloat(element.textContent.replace('RS : ', '')) || 0;
          });
```

```
          // Update the content of the element with the class 'full_price'
          document.querySelector('.full_price').textContent = 'RS .' +
sum.toFixed(2);
        }, 4000);
      }

      function reload() {
        fetch("cart.php")
          .then(response => response.text())
          .then(html => {
            document.body.innerHTML = html;
          })
          .catch(error => {
            console.error('Error loading content:', error);
          });
      }


    });
  </script>
</body>

</html>
```

Appendix B

For more user interfaces refer Appendix A

Code Segments

2. Backend End

Admin interface

```
<?php

require_once 'includes/config.php';

try {
    // SQL query total products
```

```
    $sqlProducts = "SELECT COUNT(*) FROM clothproduct";
    $stmtProducts = $connection->query($sqlProducts);
    $totalProducts = $stmtProducts->fetchColumn();

    // SQL query total customers
    $sqlCustomers = "SELECT COUNT(*) FROM customers";
    $stmtCustomers = $connection->query($sqlCustomers);
    $totalCustomers = $stmtCustomers->fetchColumn();
} catch (PDOException $e) {
    echo "Error: " . $e->getMessage();
}
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lite Fashion Admin Panel</title>
    <link
        rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.6.0/css/all.min.css" />
    <link rel="stylesheet" href="layout/css/styles.css" />
    <script src="https://cdn.tailwindcss.com"></script>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>

<body>

    <div class="flex">
        <!-- Sidebar -->
        <?php include_once 'includes/sidebar.php'; ?>

        <!-- Main Content -->
        <div class="ml-64 flex-1 p-6">
            <div class="bg-white p-6 rounded-lg shadow-lg border border-gray-200
backdrop-blur-sm backdrop-filter">
                <header class="flex justify-between items-center mb-8">
                    <div>
                        <h1 class="text-3xl font-bold text-gray-800">Admin
Dashboard</h1>
                        <p class="text-gray-500 mt-1">Welcome back,
Administrator</p>
```

```
                    </div>
                    <div class="flex items-center space-x-4">

                        <a href="#profile" id="profile" class="flex items-center
px-4 py-2 text-gray-700 hover:text-blue-600 transition-colors duration-200">
                            <svg class="w-5 h-5 mr-2" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                                <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M16 7a4 4 0 11-8 0 4 4 0 018 0zM12 14a7 7 0
00-7 7h14a7 7 0 00-7-7z" />
                            </svg>
                            Profile
                        </a>
                        <a
href="http://localhost/LiteFashionDarkDevils/admin/pages/logout.php" class="flex
items-center px-4 py-2 text-gray-700 hover:text-red-600 transition-colors
duration-200">
                            <svg class="w-5 h-5 mr-2" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                                <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2" d="M17 16l4-4m0 0l-4-4m4 4H7m6 4v1a3 3 0 01-3
3H6a3 3 0 01-3-3V7a3 3 0 013-3h4a3 3 0 013 3v1" />
                            </svg>
                            Log out
                        </a>
                    </div>
                </header>

                <main id="content-container" class="space-y-6">
                    <!-- Quick Stats -->
                    <div class="grid grid-cols-1 md:grid-cols-4 gap-6 text-
center">
                        <div class="bg-blue-50 p-4 rounded-lg border border-blue-
100 shadow-sm">
                            <h3 class="text-blue-600 font-semibold">Total
Products</h3>
                            <p class="text-2xl font-bold text-blue-600"><?php
echo $totalProducts ?></p>
                        </div>
                        <div class="bg-green-50 p-4 rounded-lg border border-
green-100 shadow-sm">
                            <h3 class="text-green-600 font-semibold">Total
Orders</h3>
                            <p class="text-2xl font-bold text-green-600">...</p>
                        </div>
```

```
                        <div class="bg-purple-50 p-4 rounded-lg border border-
purple-100 shadow-sm">
                            <h3 class="text-purple-600 font-semibold">Total
Customers</h3>
                            <p class="text-2xl font-bold text-purple-600"><?php
echo $totalCustomers ?></p>
                        </div>
                        <div class="bg-red-50 p-4 rounded-lg border border-red-
100 shadow-sm">
                            <h3 class="text-red-600 font-semibold">Revenue</h3>
                            <p class="text-2xl font-bold text-red-600">Rs.
...</p>
                        </div>
                    </div>

                    <!-- Welcome Message -->
                    <div class="bg-gradient-to-r from-blue-500 to-blue-600 text-
white p-6 rounded-lg shadow-md">
                        <h2 class="text-2xl font-bold mb-2">Welcome to the admin
dashboard!</h2>
                        <p class="text-blue-100">Manage your products, orders,
and users all in one place.</p>
                    </div>
                </main>
            </div>
        </div>
    </div>



    <script>
        $('#profile').click(function(event) {
            event.preventDefault();
            $('#content-container').load(
                'http://localhost/LiteFashionDarkDevils/admin/pages/profile.php'
            );
        });
    </script>
    <script
src="http://localhost/LiteFashionDarkDevils/admin/layout/js/script.js"></script>
</body>

</html>
```

Admin Login And Register

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lite Fashion Admin Panel</title>
    <link
        rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.6.0/css/all.min.css" />
    <link rel="stylesheet" href="layout/css/styles.css" />
    <script src="https://cdn.tailwindcss.com"></script>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>

<body class="flex justify-center bg-gray-100">
    <div class="popup-contentLog bg-white rounded-xl shadow-lg w-1/3 my-10">
        <div class="p-10 space-y-4">
 <!-- Tab Navigation -->
<div class="flex justify-center space-x-6 rounded-xl">
    <button id="loginTab" class="tab-button text-sm font-semibold text-gray-800
border-b-2 border-transparent hover:border-blue-500 hover:text-blue-500 py-3 px-6
transition-all duration-300">
        Sign In
    </button>
    <button id="signupTab" class="tab-button text-sm font-semibold text-gray-800
border-b-2 border-transparent hover:border-gray-500 hover:text-gray-500 py-3 px-6
transition-all duration-300">
        Sign Up
    </button>
</div>


            <!-- Login Form -->
            <div id="loginForm" class="form-container">
                <h1 class="text-2xl font-bold leading-tight tracking-tight text-
gray-900 text-center my-5">
                    Admin Login
                </h1>
                <form class="space-y-4 md:space-y-6" method="POST"
action="http://localhost/LiteFashionDarkDevils/admin/pages/backend/signUp.php">
                    <!-- Email -->
```

```
                    <div>
                        <label for="email" class="block mb-2 text-sm font-medium
text-gray-900">Email Address</label>
                        <input type="email" name="email" id="email" class="bg-
gray-50 border border-gray-300 text-gray-900 rounded-lg focus:ring-primary-600
focus:border-primary-600 block w-full p-2.5" placeholder="name@company.com"
required>
                    </div>

                    <!-- Password -->
                    <div>
                        <label for="password" class="block mb-2 text-sm font-
medium text-gray-900">Password</label>
                        <input type="password" name="password" id="password"
class="bg-gray-50 border border-gray-300 text-gray-900 rounded-lg focus:ring-
primary-600 focus:border-primary-600 block w-full p-2.5" placeholder="••••••••"
required>
                    </div>

                    <!-- Remember Me -->
                    <div class="flex items-center justify-between">
                        <div class="flex items-start">
                            <input type="checkbox" name="remember" id="remember"
class="w-4 h-4 text-primary-600 bg-gray-100 border-gray-300 rounded focus:ring-
primary-500">
                            <label for="remember" class="ml-2 text-sm text-gray-
900">Remember me</label>
                        </div>
                        <a href="forgot_password_form.php" class="text-sm font-
medium text-primary-600 hover:underline">Forgot password?</a>
                    </div>

                    <!-- Submit -->
                    <button type="submit" name="submitLogin" class="w-full text-
white bg-gray-600 hover:bg-primary-700 focus:ring-4 focus:outline-none
focus:ring-primary-300 font-medium rounded-lg text-sm px-5 py-2.5 text-
center">Sign In</button>
                </form>
            </div>

            <!-- Signup Form -->
            <div id="signupForm" class="form-container hidden">
                <h1 class="text-2xl font-bold leading-tight tracking-tight text-
gray-900 text-center my-5">
                    Create Admin Account
```

```
                </h1>
                <form class="space-y-4 md:space-y-6" method="POST"
action="http://localhost/LiteFashionDarkDevils/admin/pages/backend/signUp.php">
                    <!-- Full Name -->
                    <div>
                        <label for="fname" class="block mb-2 text-sm font-medium
text-gray-900">Full Name</label>
                        <input type="text" name="fname" id="fname" class="bg-
gray-50 border border-gray-300 text-gray-900 rounded-lg focus:ring-primary-600
focus:border-primary-600 block w-full p-2.5" placeholder="Enter your full name"
required>
                    </div>

                    <!-- Username -->
                    <div>
                        <label for="lname" class="block mb-2 text-sm font-medium
text-gray-900">Username</label>
                        <input type="text" name="lname" id="lname" class="bg-
gray-50 border border-gray-300 text-gray-900 rounded-lg focus:ring-primary-600
focus:border-primary-600 block w-full p-2.5" placeholder="Enter username"
required>
                    </div>

                    <!-- Email -->
                    <div>
                        <label for="email" class="block mb-2 text-sm font-medium
text-gray-900">Email Address</label>
                        <input type="email" name="email" id="email" class="bg-
gray-50 border border-gray-300 text-gray-900 rounded-lg focus:ring-primary-600
focus:border-primary-600 block w-full p-2.5" placeholder="name@company.com"
required>
                    </div>

                    <!-- Password -->
                    <div>
                        <label for="password" class="block mb-2 text-sm font-
medium text-gray-900">Password</label>
                        <input type="password" name="password" id="password"
class="bg-gray-50 border border-gray-300 text-gray-900 rounded-lg focus:ring-
primary-600 focus:border-primary-600 block w-full p-2.5" placeholder="••••••••"
required>
                    </div>

                    <!-- Submit -->
```

```
                    <button type="submit" name="submitACC" class="w-full text-
white bg-gray-600 hover:bg-primary-700 focus:ring-4 focus:outline-none
focus:ring-primary-300 font-medium rounded-lg text-sm px-5 py-2.5 text-
center">Create Account</button>
                </form>
            </div>
        </div>
    </div>

    <script>
        // Toggle forms

        document.getElementById('loginTab').addEventListener('click', () => {
            document.getElementById('loginForm').classList.remove('hidden');
            document.getElementById('signupForm').classList.add('hidden');
        });

        document.getElementById('signupTab').addEventListener('click', () => {
            document.getElementById('signupForm').classList.remove('hidden');
            document.getElementById('loginForm').classList.add('hidden');
        });
    </script>
</body>

</html>
```

Add product

```
<!DOCTYPE html>
<html lang="en">

<!-- header -->
<?php include_once '../includes/header.php'; ?>

<body>
  <hr>
  <div class="container mx-auto p-4">
    <h1 class="text-2xl font-bold mb-4">Add New Product</h1>

    <form
      id="postForm"
      enctype="multipart/form-data"
      class="flex flex-col h-full">
      <div class="grid grid-cols-2 gap-10 flex-grow">
        <!-- Left Side -->
```

```
        <div class="flex flex-col justify-between">
          <div>
            <div class="mb-4">
              <label for="P_name" class="block text-gray-700 font-bold mb-
2">Product Name</label>
              <input
                type="text"
                id="P_name"
                name="P_name"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200"
                required />
            </div>

            <div class="mb-4">
              <label
                for="P_categoryId"
                class="block text-gray-700 font-bold mb-2">Category</label>
              <select
                id="P_categoryId"
                name="P_categoryId"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200"
                required>
                <option value="">Select Category</option>
                <option value="C001">Men</option>
                <option value="C002">Women</option>
                <option value="C003">Kids</option>
              </select>
            </div>

            <div class="mb-4">
              <label for="P_price" class="block text-gray-700 font-bold mb-
2">Price</label>
              <input
                type="number"
                step="0.01"
                id="P_price"
                name="P_price"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200"
                required />
```

```
            </div>

            <div class="mb-4">
              <label
                for="P_quantity"
                class="block text-gray-700 font-bold mb-2">Quantity</label>
              <input
                type="number"
                id="P_quantity"
                name="P_quantity"
                value="1"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200"
                required
                disabled />
            </div>

            <div class="mb-4">
              <label
                for="P_description"
                class="block text-gray-700 font-bold mb-2">Description</label>
              <textarea
                id="P_description"
                name="P_description"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200"
                rows="4"></textarea>
            </div>
          </div>
        </div>

        <!-- Right Side -->
        <div class="flex flex-col justify-between">
          <div>
            <!-- images -->
            <div>
              <div class="mb-4">
                <label for="P_image1" class="block text-gray-700 font-bold mb-
2">Image 1</label>
                <input
                  type="file"
                  id="P_image1"
                  name="P_image1"
```

```
                  class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200" />
              </div>

              <div class="mb-4">
                <label for="P_image2" class="block text-gray-700 font-bold mb-
2">Image 2</label>
                <input
                  type="file"
                  id="P_image2"
                  name="P_image2"
                  class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200" />
              </div>

              <div class="mb-4">
                <label for="P_image3" class="block text-gray-700 font-bold mb-
2">Image 3</label>
                <input
                  type="file"
                  id="P_image3"
                  name="P_image3"
                  class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200" />
              </div>

              <div class="mb-4">
                <label for="P_image4" class="block text-gray-700 font-bold mb-
2">Image 4</label>
                <input
                  type="file"
                  id="P_image4"
                  name="P_image4"
                  class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200" />
              </div>
            </div>

            <div class="mb-4">
              <label class="block text-gray-700 font-bold mb-2">Sizes</label>
              <div class="flex items-center space-x-4">
```

```
              <label class="inline-flex items-center">
                <input
                  type="checkbox"
                  id="P_small"
                  name="P_small"
                  value="S"
                  class="form-checkbox" />
                <span class="ml-2">Small</span>
              </label>
              <label class="inline-flex items-center">
                <input
                  type="checkbox"
                  id="P_medium"
                  name="P_medium"
                  value="M"
                  class="form-checkbox" />
                <span class="ml-2">Medium</span>
              </label>
              <label class="inline-flex items-center">
                <input
                  type="checkbox"
                  id="P_large"
                  name="P_large"
                  value="L"
                  class="form-checkbox" />
                <span class="ml-2">Large</span>
              </label>
              <label class="inline-flex items-center">
                <input
                  type="checkbox"
                  id="P_extraLarge"
                  name="P_extraLarge"
                  value="XL"
                  class="form-checkbox" />
                <span class="ml-2">Extra Large</span>
              </label>
            </div>
          </div>

          <div class="mb-4">
            <label for="P_status" class="block text-gray-700 font-bold mb-
2">Status</label>
            <select
              id="P_status"
              name="P_status"
```

```
                disabled
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200">
                <option value="1">Active</option>
                <option value="0">Inactive</option>
              </select>
            </div>
          </div>
        </div>
      </div>

      <div class="flex justify-end items-center mt-4 gap-10 p-4 border-t border-
gray-200">
        <!-- Message -->
        <h1 class="text-sky-500 font-semibold text-lg" id="show"></h1>
        <!-- Button -->
        <button
          type="submit"
          name="submit"
          class="bg-gradient-to-r from-blue-500 to-blue-600 hover:from-blue-600
hover:to-blue-700 text-white font-semibold py-2 px-6 rounded-lg shadow-lg
focus:outline-none focus:ring-4 focus:ring-blue-300 transition-transform
duration-300 transform hover:scale-105">
          Add Product
        </button>
      </div>

    </form>
  </div>

  <script>
    document.getElementById('postForm').addEventListener('submit', postName);

    function postName(e) {
      e.preventDefault();

      let formData = new FormData(document.getElementById('postForm'));

      let xhr = new XMLHttpRequest();
      xhr.open('POST',
'http://localhost/LiteFashionDarkDevils/admin/pages/backend/add-product.php',
true);

      xhr.onload = function() {
```

```
        if (xhr.status === 200) {
          console.log(this.responseText);
          document.getElementById('show').innerHTML = this.responseText;
        } else {
          console.error('Error in form submission');
        }
      };

      xhr.send(formData);
    }
  </script>
</body>

</html>
```

Product List

```
<?php
require_once '../includes/config.php';
?>

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Lite Fashion Admin Panel</title>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.6.0/css/all.min.css" />
  <link rel="stylesheet"
href="http://localhost/LiteFashionDarkDevils/admin/layout/css/styles.css" />
  <script src="https://cdn.tailwindcss.com"></script>
  <style>
    .content {
      display: none;
```

```
    }

    .active {
      display: block;
    }
  </style>
</head>

<body>
  <div id="displayContent" class="content active">
    <div class="overflow-x-auto">
      <h1 class="text-2xl font-bold mb-4">Product List</h1>
      <table class="min-w-full bg-white rounded-lg overflow-hidden">
        <thead>
          <tr>
            <th class="py-2 px-4 bg-gray-800 text-white">ID</th>
            <th class="py-2 px-4 bg-gray-800 text-white">Image</th>
            <th class="py-2 px-4 bg-gray-800 text-white">Product Name</th>
            <th class="py-2 px-4 bg-gray-800 text-white">Category</th>
            <th class="py-2 px-4 bg-gray-800 text-white">Price</th>
            <th class="py-2 px-4 bg-gray-800 text-white">Quantity</th>
            <th class="py-2 px-4 bg-gray-800 text-white">Sizes</th>
            <th class="py-2 px-4 bg-gray-800 text-white">Status</th>
            <th class="py-2 px-4 bg-gray-800 text-white">Actions</th>
          </tr>
        </thead>
        <tbody class="text-gray-700">
          <?php
          try {
            $query = $connection->query("SELECT * FROM clothproduct");

            while ($product = $query->fetch(PDO::FETCH_ASSOC)) {
              $sizes = [];
              if ($product['P_small']) $sizes[] = 'S';
              if ($product['P_medium']) $sizes[] = 'M';
              if ($product['P_large']) $sizes[] = 'L';
              if ($product['P_extraLarge']) $sizes[] = 'XL';

              echo "<tr>
                <td class='border px-4 py-2'>{$product['ProductId']}</td>
                <td class='border px-4 py-2'>
                  <img
src='http://localhost/LiteFashionDarkDevils/admin/uploads/{$product['P_image1']}'
alt='Product Image' style='width: 50px; height: auto;' />
                </td>
```

```
                <td class='border px-4 py-2'>{$product['P_name']}</td>
                <td class='border px-4 py-2'>{$product['P_categoryId']}</td>
                <td class='border px-4 py-2'>Rs. {$product['P_price']}</td>
                <td class='border px-4 py-2'>{$product['P_quantity']}</td>
                <td class='border px-4 py-2'>" . implode(', ', $sizes) . "</td>
                <td class='border px-4 py-2'>" . ($product['P_status'] == '1' ?
'Active' : 'Inactive') . "</td>
                <td class='border px-4 py-2'>
                  <div class='flex'>
                    <button class='bg-blue-500 hover:bg-blue-700 text-white font-
semibold py-1 px-3 rounded' onclick=\"showContent('editContent')\">Edit</button>
                    <button class='bg-red-500 hover:bg-red-700 text-white font-
semibold py-1 px-3 rounded ml-2'
onclick=\"deleteProduct({$product['ProductId']})\">Delete</button>
                  </div>
                </td>
              </tr>";
            }
          } catch (PDOException $e) {
            echo "<tr><td colspan='10' class='border px-4 py-2 text-red-
500'>Error loading products</td></tr>";
          }
          ?>
        </tbody>
      </table>
    </div>
  </div>

  <!-- Edit Section -->
  <div id="editContent" class="content container mx-auto p-4">
    <h1 class="text-2xl font-bold mb-4">Update Product</h1>
    <form
      action="/update-product"
      method="POST"
      enctype="multipart/form-data"
      class="flex flex-col h-full">
      <div class="grid grid-cols-2 gap-10 flex-grow">
        <!-- Left Side -->
        <div class="flex flex-col justify-between">
          <div>
            <div class="mb-4">
              <label for="P_name" class="block text-gray-700 font-bold mb-
2">Product Name</label>
              <input
                type="text"
```

```
                id="P_name"
                name="P_name"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200"
                required />
            </div>

            <div class="mb-4">
              <label
                for="P_categoryId"
                class="block text-gray-700 font-bold mb-2">Category</label>
              <select
                id="P_categoryId"
                name="P_categoryId"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200"
                required>
                <option value="">Select Category</option>
                <option value="men">Men</option>
                <option value="women">Women</option>
                <option value="kids">Kids</option>
              </select>
            </div>

            <div class="mb-4">
              <label for="P_price" class="block text-gray-700 font-bold mb-
2">Price</label>
              <input
                type="number"
                step="0.01"
                id="P_price"
                name="P_price"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200"
                required />
            </div>

            <div class="mb-4">
              <label
                for="P_quantity"
                class="block text-gray-700 font-bold mb-2">Quantity</label>
              <input
```

```
                type="number"
                id="P_quantity"
                name="P_quantity"
                value="1"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200"
                required />
            </div>

            <div class="mb-4">
              <label
                for="P_description"
                class="block text-gray-700 font-bold mb-2">Description</label>
              <textarea
                id="P_description"
                name="P_description"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200"
                rows="4"></textarea>
            </div>
          </div>
        </div>

        <!-- Right Side -->
        <div class="flex flex-col justify-between">
          <div>
            <div class="mb-4">
              <label for="P_image1" class="block text-gray-700 font-bold mb-
2">Image 1</label>
              <input
                type="file"
                id="P_image1"
                name="P_image1"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200" />
            </div>

            <div class="mb-4">
              <label for="P_image2" class="block text-gray-700 font-bold mb-
2">Image 2</label>
              <input
                type="file"
```

```
                id="P_image2"
                name="P_image2"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200" />
            </div>

            <div class="mb-4">
              <label for="P_image3" class="block text-gray-700 font-bold mb-
2">Image 3</label>
              <input
                type="file"
                id="P_image3"
                name="P_image3"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200" />
            </div>

            <div class="mb-4">
              <label for="P_image4" class="block text-gray-700 font-bold mb-
2">Image 4</label>
              <input
                type="file"
                id="P_image4"
                name="P_image4"
                class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200" />
            </div>

            <div class="mb-4">
              <label class="block text-gray-700 font-bold mb-2">Sizes</label>
              <div class="flex items-center space-x-4">
                <label class="inline-flex items-center">
                  <input
                    type="checkbox"
                    id="P_small"
                    name="P_small"
                    value="S"
                    class="form-checkbox" />
                  <span class="ml-2">Small</span>
                </label>
                <label class="inline-flex items-center">
                  <input
```

```
                  type="checkbox"
                  id="P_medium"
                  name="P_medium"
                  value="M"
                  class="form-checkbox" />
                <span class="ml-2">Medium</span>
              </label>
              <label class="inline-flex items-center">
                <input
                  type="checkbox"
                  id="P_large"
                  name="P_large"
                  value="L"
                  class="form-checkbox" />
                <span class="ml-2">Large</span>
              </label>
              <label class="inline-flex items-center">
                <input
                  type="checkbox"
                  id="P_extraLarge"
                  name="P_extraLarge"
                  value="XL"
                  class="form-checkbox" />
                <span class="ml-2">Extra Large</span>
              </label>
            </div>
          </div>

          <div class="mb-4">
            <label for="P_status" class="block text-gray-700 font-bold mb-
2">Status</label>
            <select
              id="P_status"
              name="P_status"
              class="w-full px-3 py-2 bg-white rounded-md border border-gray-
400 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 transition-all
duration-200">
              <option value="1">Active</option>
              <option value="0">Inactive</option>
            </select>
          </div>
        </div>
      </div>
```

```
      <div class="flex justify-end mt-4">
        <button
          type="submit"
          class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4
rounded focus:outline-none focus:shadow-outline transition-colors duration-200">
          Update Product
        </button>
      </div>
    </form>
  </div>

  <script
src="http://localhost/LiteFashionDarkDevils/admin/layout/js/script.js"></script>
  <script>
    function showContent(contentId) {
      document.querySelectorAll('.content').forEach(content =>
content.classList.remove('active'));
      document.getElementById(contentId).classList.add('active');
    }

    function deleteProduct(productId) {
      if (confirm('Are you sure you want to delete this product?')) {
        window.location.href = `delete-product.php?ProductId=${productId}`;
      }
    }
  </script>
</body>

</html>
```

Appendix C

This appendix mainly related to database design

Admin Table

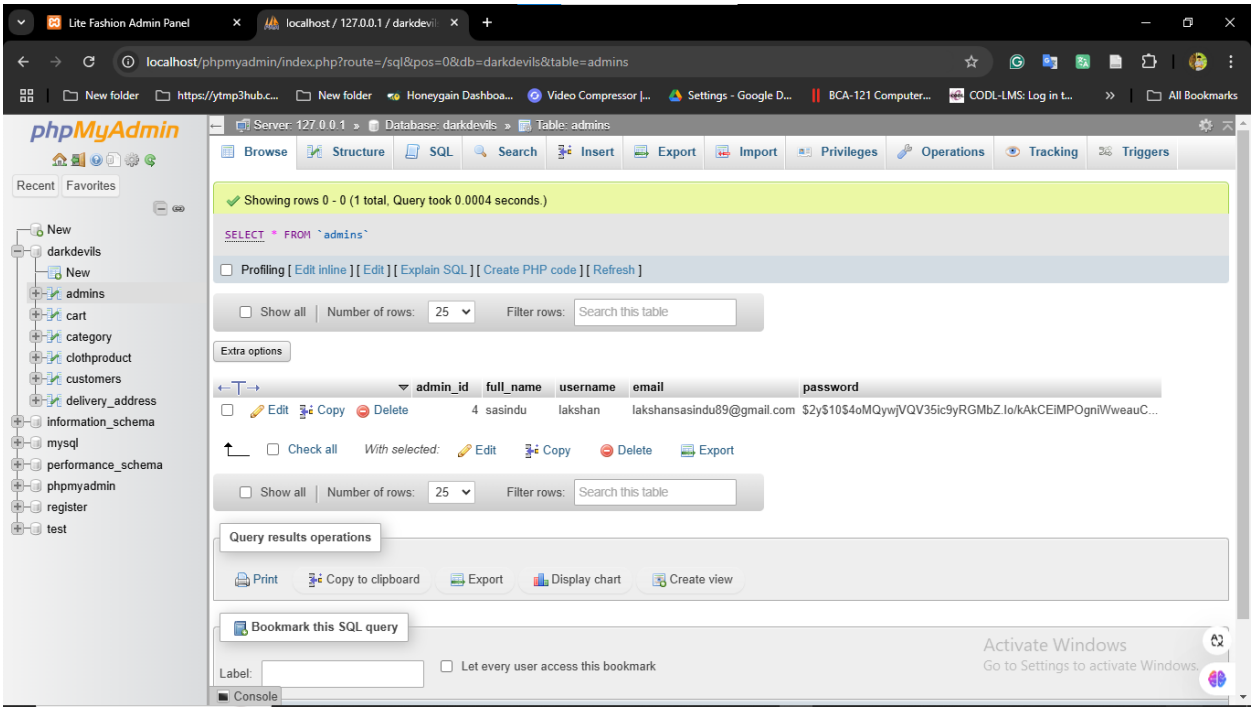

Figure 22 Admin Table

Cart Table

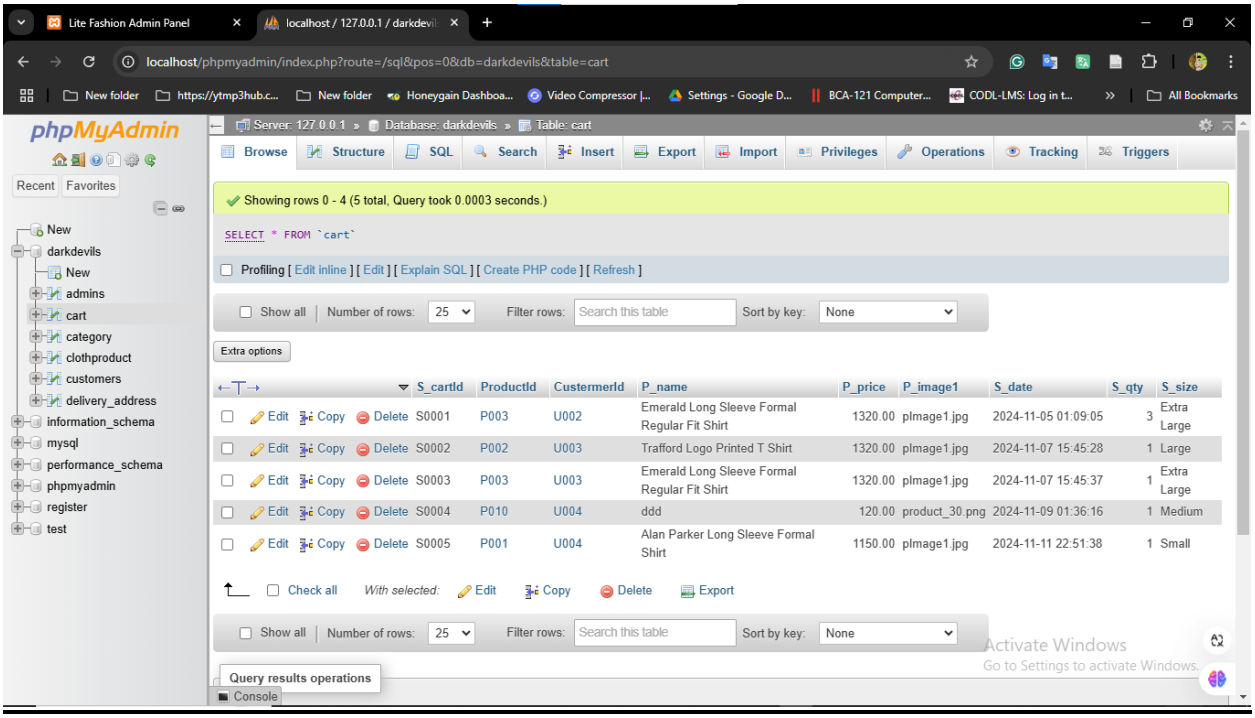

Figure 23 Cart Table

Category Table

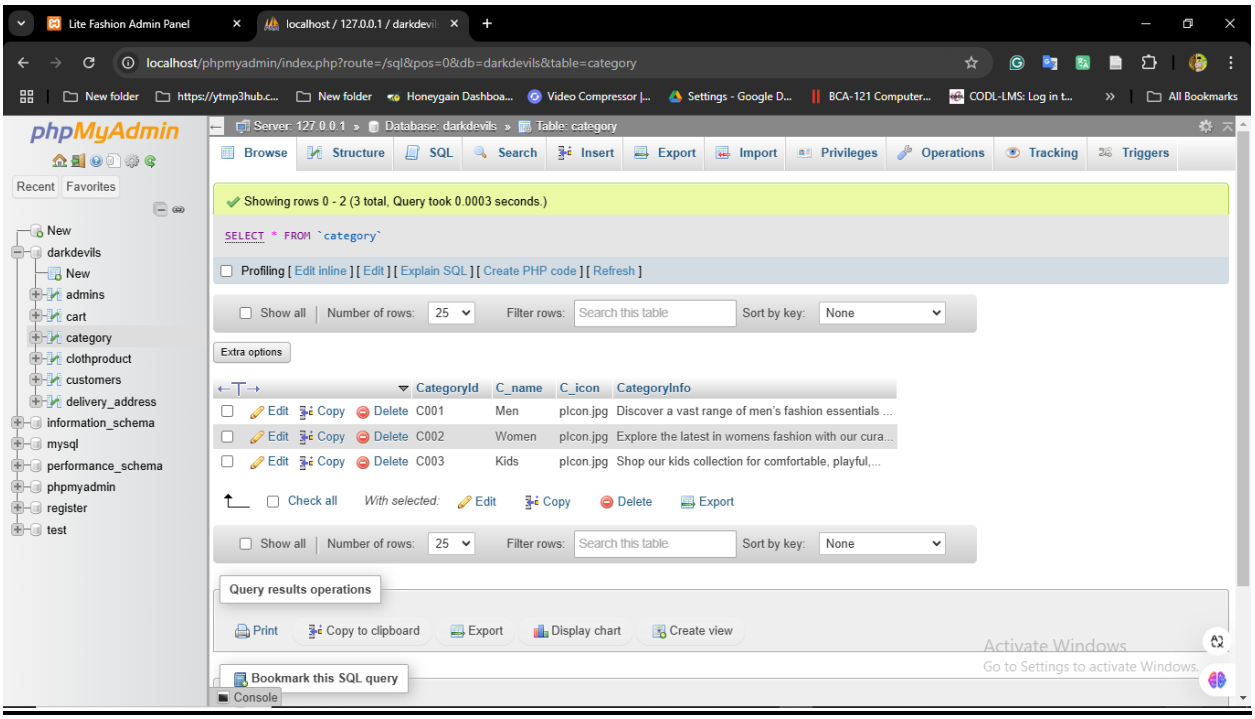

Figure 24 Category Table

Cloth product Table

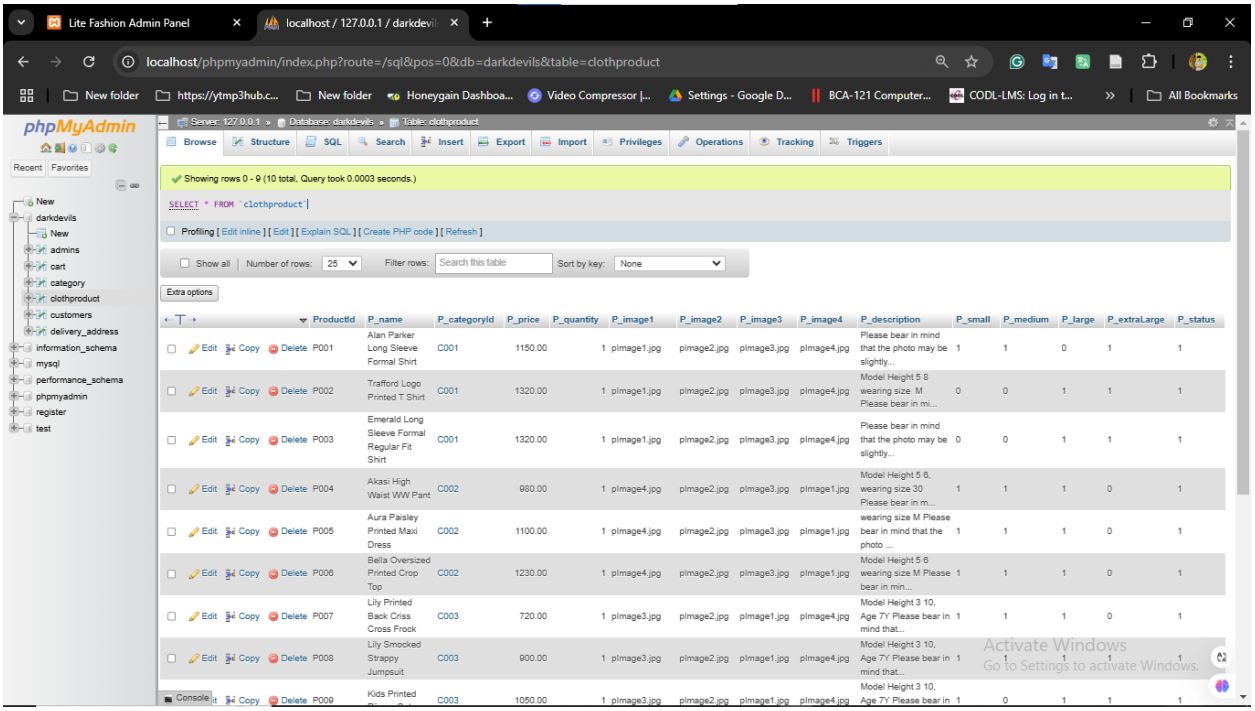

Figure 25 Cloth product Table

Customers Table

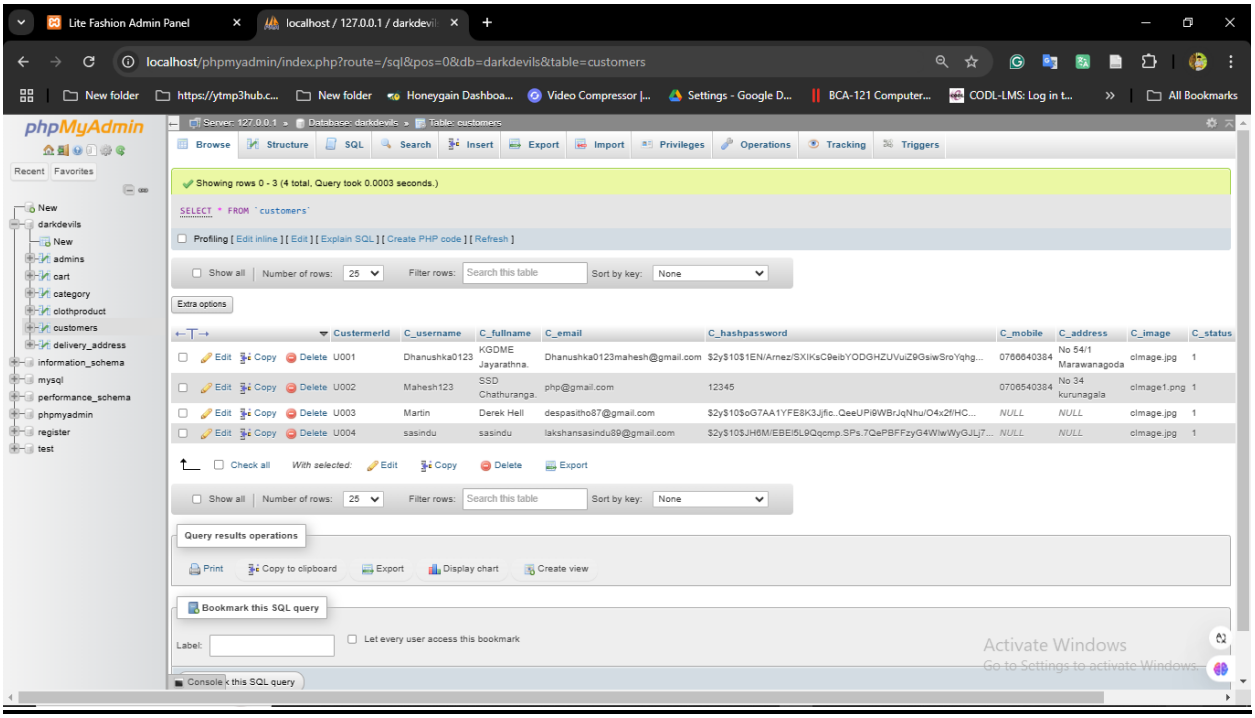

Figure 26 Customers Table

Delivery_Addres Table

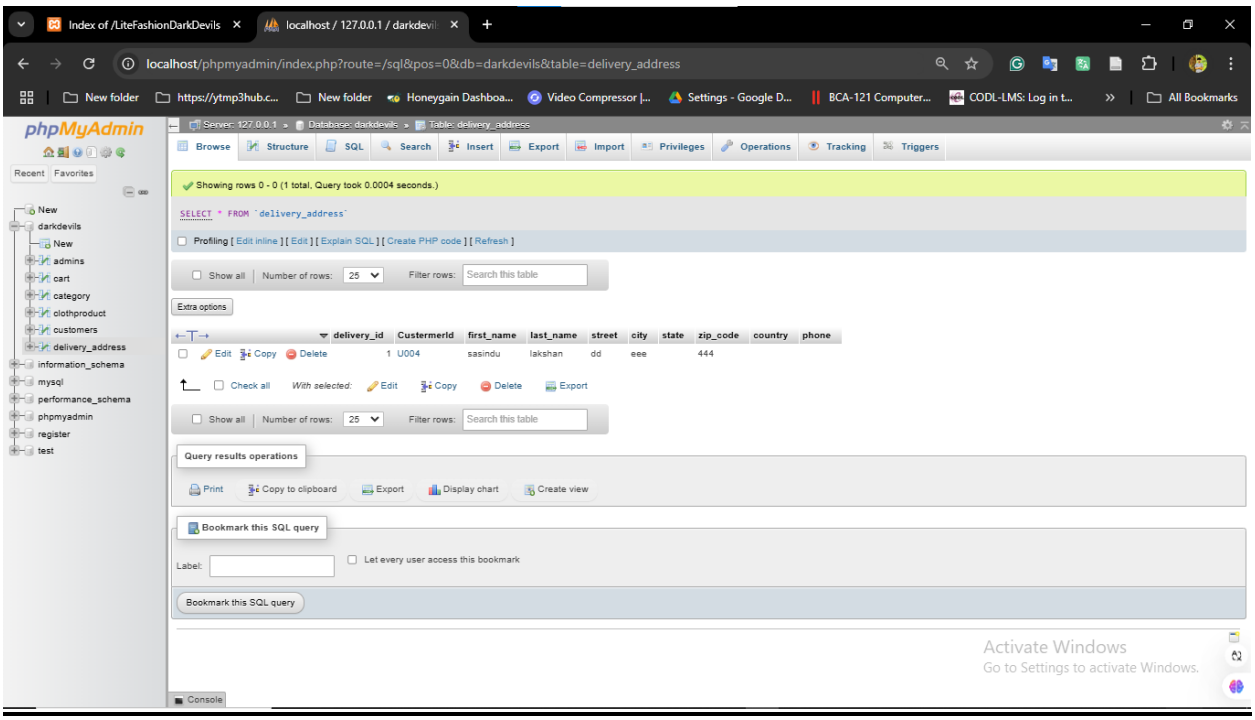

Figure 27 Delivery_Addres Table

Appendix D

This appendix mainly related chapter 6– Testing and Evaluation

Test case - Lite Fashion Website

Project Name	Lite Fashion Website Testing
Module Name	Check functionality
Reference Document	SRS
Created By	Charith , Sasindu, Mahesh
Date of creation	11/11/2024

Test Case ID	Test Description/Test case name	Prerequisite	Test Steps	Input Data	Expected Result	Actual Result	Status
TC_LF01	Verify successful user registration with valid data	User is not logged in	1. Navigate to the Lite Fashion registration page. 2. Enter valid details (name, email, password). 3. Submit the registration form.	Name: 'Amal Kumara', Email: 'amal@example.com', Password: 'Password123!'	User should be registered successfully and redirected to the welcome page.	as expected	pass
TC_LF02	Verify error message on registration with an already registered email	User email already registered	1. Navigate to the Lite Fashion registration page. 2. Enter details (name, already registered email, password). 3. Submit the registration form.	Name: 'Amal Kumara', Email: 'amal@example.com', Password: 'Password123!'	System should display an error message indicating the email is already in use.	as expected	pass
TC_LF03	Verify product search functionality with a valid keyword	At least one product available on the site	1. Navigate to the Lite Fashion homepage. 2. Enter a product keyword (e.g., 'pendrive') in the search bar. 3. Click on the search button.	Keyword: 'men shirt'	Relevant search results should be displayed on the page.	as expected	pass
TC_LF04	Verify the system handles boundary values for product search input	System has a defined character limit for search input	100 characters (assuming max limit)	System should display search results or an appropriate message if input exceeds the limit.		as expected	pass
TC_LF05	Verify adding a product to the cart	User is logged in and product is available	1. Search for a product. 2. Select a product from the search results. 3. Click on 'Add to Cart.	Product selection	Product should be successfully added to the cart, and a confirmation message should be displayed.	as expected	pass
TC_LF06	Verify checkout process with valid payment information	User is logged in and has items in the cart	1. Navigate to the cart. 2. Click on 'Checkout.' 3. Enter valid payment and shipping details. 4. Confirm the order.	Payment information, shipping address	Order should be placed successfully, and an order confirmation page should be displayed.	as expected	pass

Figure 28 Testing and Evaluation

References

[1] “Depop - Buy, Sell, Discover Unique Fashion.” Accessed August 27, 2024. https://www.depop.com/

[2] “Fashion Bug | Online Dress Shopping in Sri Lanka.” Accessed August 27, 2024. https://www.fashionbug.lk/

[3] “GFLOCK Clothing Sri Lanka.” Accessed August 27, 2024. https://gflock.lk/

[4] “House of Fashions Sri Lanka.” Accessed August 27, 2024. https://houseoffashions.lk/ [5] “Kandy Selection.” Accessed August 27, 2024. https://kandyselection.lk/

[6] NOLIMIT. “Home.” Accessed August 27, 2024. https://www.nolimit.lk/

[7] “Poshmark: Buy and Sell Fashion, Home Decor, Beauty & More.” Accessed August 27, 2024. https://poshmark.com/

[8] “Shop Women’s Clothing, Footwear & Accessories Online | Nils Store.” Accessed August 27, 2024. https://nilsonline.lk/

[9] “Thilakawardhana | Online Store.” Accessed August 27, 2024. https://thilakawardhana.com/

[10] “Vinted | Sell and Buy Clothes, Shoes and Accessories.” Accessed August 27, 2024. https://www.vinted.com/