A

PROJECT REPORT ON

**“BOUTIQUE MANAGEMENT SYSTEM”**

SUBMITTED BY

**Ms. Sidra Ajaz Mulla**

SUBMITTED TO

SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

IN PARTIAL FULFILLMENT OF DEGREE

**MASTER OF COMPUTER APPLICATION [ SEM I ]**

UNDER THE GUIDANCE OF

**Ms. Samiksha Yeola**

Through



**Sadhu Vasvani Institute of Management Studies for Girls,**

**Koregoan Park, Pune – 411001**

2024 – 2025

CERTIFICATE



This is to certify that the Mini Project report entitled, **“Boutique Management System**” being submitted herewith in partial fulfilment of the requirement of the award of the degree of **MASTER OF COMPUTER APPLICATION (SEM I )** to Savitribai Phule Pune University, Pune is result of the original project work completed by **Sidra Ajaz Mulla** under the supervision and guidance and to the best of my knowledge and belief , the work embodied in this project has not formed earlier the basis for the award of any degree of similar little o any other University or examining body.

Date :

Place :

Dr. B.H. Nanwani

Director

Dr. Neeta Raskar

HOD

Ms. Samiksha Yeola

Project Guide

Examiner 2 sign

Examiner 1 sign

DECLARATION BY STUDENT

To,

The Director,

SVIMS, Koregoan Park, Pune

I undersigned hereby declare that this project titled, “Boutique Management System ” written and submitted by me to SPPU, Pune in partial fulfilment of the requirement of the degree of MASTER OF COMPUTER APPLICATION (MCA I ) under the guidance of Ms. Samiksha Yeola, is my original work.

I further declare that to the best of my knowledge and belief , this project has not been submitted to this or any other University or Institution for the award of any Degree.

Place : Pune

Date :

Sidra Ajaz Mulla

ACKNOWLEDGEMENT

I extend my sincere gratitude to Dr. B.H. Nanwani, Dr. Neeta Raskar and Ms. Samiksha Yeola for allowing me to carry out the study and for their constant encouragement, valuable suggestions, and guidance during the research work.

I extend my special gratitude to my family members and friends who encouraged and motivated me to complete the project report.

Place : Pune

Date :

Sidra Ajaz Mulla

|  |  |  |
| --- | --- | --- |
| CHAPTER | DETAILS | PAGE NO |
| 1 | **CHAPTER 1: INTRODUCTION**   * 1. Client/Organization Profile   2. Need for System   3. Scope & Feasibility of Work   4. Operating Environment – H/w & S/w   5. Architecture of system   6. Detail Description of Technology Used |  |
| 2 | **CHAPTER 2: PROPOSED SYSTEM**   * 1. Proposed System   2. Objectives of System   3. User Requirements |  |
| 3 | **CHAPTER 3: ANALYSIS & DESIGN**   * 1. DFD   2. Table specifications (Database)   3. ERD   4. Object Diagram   5. Class Diagram   6. Use Case Diagrams   7. Web Site Map Diagram (if Website) |  |
| 4 | **CHAPTER 4: USER MANUAL**   * 1. User Interface Design (Screens etc.)   2. Limitations   3. Future enhancement   **BIBLIOGRAPHY**  **ANNEXURE:** Sample program code |  |

INDEX

CHAPTER 1 : INTRODUCTION

* 1. **CLIENT / ORGANIZATION PROFILE:**

Boutique is a fashion villa with collection of enormous designs. To an extent, in recent times the word ‘Boutique’ plays an active role in the day-to-day life. The main attraction is towards the ethnic designs, standard colours, desired patterns and availability of Ready-to-wear garments. Further they sell fashionable goods including apparel, jewellery, handbags, shoes and hair accessories. Many of the boutiques are specialized in producing handmade clothing and customer made apparels.

Successful management of the boutique is based on user friendly system. It is the system to manage the clothing, accessories and the accounts associated to the boutique. A boutique can be a successful one by managing it in a correct way. Boutique is a place where the collection of designers is displayed or it is the place where the fashionable garments and accessories are sold.

Boutiques can be generally classified into Higher End boutiques and Lower end boutiques. Further divided into Stand – Alone and Chain. The upcoming fashion trends and sales are analyzed and scrutinized by Fashion Experts, especially a team of fashion merchandisers.

Stand – Alone boutiques are owned by single ownership and a perfect location whereas Chain boutiques have the ownership by larger companies and widespread locations.

    ▪      High End Boutiques

▪      Low End Boutiques

Boutiques are available to all people of different strata, say, for example, the rich people can afford the price and the other one is the place where the people with moderate range can buy the desired collections.

* 1. **NEED OF THE SYSTEM:**

A Boutique Management System is designed to streamline and automate the operations of a boutique store. The need for such a system arises from several business requirements, including inventory management, customer service, sales tracking, and financial monitoring. Here's an outline of why this system is needed:

* Efficient Inventory Management:

A boutique needs to manage its stock effectively to avoid overstocking or understocking. The system helps track inventory levels, manage product categories, and receive alerts when stock levels are low, ensuring timely reordering.

* Sales Tracking and Reporting:

Manual sales tracking can lead to errors and inefficiencies. The system automates the sales process, records transactions, and generates sales reports. This helps analyze sales trends, identify best-selling products, and make data-driven decisions.

* Customer Relationship Management (CRM):

Maintaining customer data and preferences is key for a boutique’s growth. The system stores customer details, tracks purchase history, and allows for personalized marketing, improving customer satisfaction and loyalty.

* Streamlined Billing and Payment Process:

The system automates the billing process, calculating prices, discounts, and taxes accurately, reducing the chances of errors. It also integrates with payment gateways for easy processing of payments.

* Reduction of Human Error and Increased Efficiency:

Manual handling of inventory, billing, and reporting is time-consuming and prone to mistakes. The system reduces human errors, ensuring smooth operations, faster transactions, and greater efficiency.

* Data Security and Backup:

The system ensures that sensitive data (customer details, sales, and financial information) is securely stored and backed up, protecting it from data loss, theft, or corruption.

* 1. **SCOPE AND FEASIBLITY OF WORK**

SCOPE OF WORK ::

The following are not part of the current project scope, and any future changes or additions will require separate approval:

* **E-commerce Integration**: While the system will manage in-store operations, integration with an online store will not be included in the first phase.
* **Mobile App Development**: The development of a dedicated mobile app for customers or staff will not be included.
* **Advanced AI Features**: Advanced AI-based forecasting, personalized recommendations, or chatbots for customer support are not part of this version of the system.
* **Internationalization/Localization**: The system will not support multiple languages or currencies in the first version. It will initially operate in the local language and currency.
* **Assumptions**:
  + The client will provide timely access to necessary resources and personnel (e.g., product data, employee details).
* **Constraints**:
  + The system will initially only be deployed for in-store operations.
  + The project will adhere to the budget and time constraints set forth in the contract.

FEASIBILITY OF WORK ::

A feasibility study for a Boutique Management System is a comprehensive analysis that assesses the practicality, viability, and potential success of implementing such a system in a boutique setting. The study typically covers various aspects, including technical, operational, economic, legal, and scheduling considerations. Here is an outline of key components that might be considered in a feasibility study for a Boutique Management System:

**Technical Feasibility:**

Technical feasibility in the context of a Boutique Management System refers to the assessment of whether the proposed system can be developed, implemented, and maintained using the available technology infrastructure and resources. It involves evaluating the technical requirements, capabilities, and constraints associated with the implementation of the system.

**Operational Feasibility:**

Operational feasibility in the context of a Boutique Management System involves assessing whether the proposed system aligns with the current operations of the boutique and whether it can be effectively integrated into the daily workflow. It focuses on evaluating the impact of the system on the people, processes, and overall functioning of the boutique.

**Economical Feasibility:**

Economic or financial feasibility in the context of a Boutique Management System involves assessing whether the proposed system is financially viable and whether the benefits outweigh the costs associated with its implementation. This feasibility study helps determine the economic justification and financial sustainability of the system.

* 1. OPERATING ENVIRONMENT :: H/W AND S/W

In a boutique management system, both hardware (H/W) and software (S/W) work together to enable efficient operations like inventory management, sales processing, customer management, and financial reporting. Below is an overview of the hardware and software components needed for such a system.

HARDWARE AND SOFTWARE ::

|  |  |  |
| --- | --- | --- |
| **Component** | **Hardware (H/W)** | **Software (S/W)** |
| **Point-of-Sale (POS) System** | POS Terminal, Barcode Scanner, Receipt Printer, Cash Drawer, Touch Screen | POS Software (Sales Interface, Payment Gateway Integration) |
| **Server/Computers** | Desktop/Laptop, Server (for large systems) | Boutique Management Software (Inventory, CRM, Reporting) |
| **Networking Equipment** | Router, Wi-Fi / LAN, Switches | Cloud-based Solutions (for remote access, centralized data) |
| **Security Devices** | CCTV, Barcode Labels, Uninterruptible Power Supply (UPS) | Security Software (Antivirus, Firewall, Encryption) |
| **Mobile Devices (Optional)** | Tablets or Smartphones for staff and customer interaction | Mobile App for Staff (Inventory Check, Sales, Customer Management) |
|  | | |

* 1. ARCHITECTURE OF SYSTEM

The Boutique Management System is a three-tier architecture, ensuring scalability, maintainability, and ease of deployment. It includes a user-friendly interface for boutique employees and customers, a robust business logic layer for processing, and a secure database for storage.

Three-Tier Architecture

* **Presentation Layer:**
* **Description**: This is the user-facing part of the system. It provides the interface for users to interact with the system.
* **Components:**

Web Application or Mobile App: For boutique staff and customers.

* **Key Features:**

Inventory Dashboard

Sales Interface (Point of Sale)

Customer Management Module

Reporting Interface

* **Technologies:**

Frontend Frameworks: React.js, Angular, or Flutter.

Communication Protocols: HTTPS, REST APIs.

* **Business Logic Layer:**
* **Description**: Contains the core logic of the system and processes all business rules.
* **Components:**

Authentication & Authorization: Secure login and role-based access control.

Inventory Management: Tracks stock levels, updates products, and generates alerts.

Sales Processing: Handles orders, payment processing, and invoices.

Customer Management: Tracks customer details, purchase history, and loyalty programs.

Reporting Engine: Generates analytics and reports on sales, inventory, and customer behavior.

* **Technologies:**

Programming Languages: Python, Java, or Node.js.

Frameworks: Django, Spring Boot, or Express.js.

* **Data Layer:**
* **Description:** Manages the storage, retrieval, and security of data.
* **Components:**

Database: Stores data related to products, customers, sales, and analytics.

Data Backup: Ensures redundancy and recovery in case of failure.

* **Technologies:**

Database Systems: MySQL, PostgreSQL, or MongoDB.

Cloud Services: AWS RDS, Google Cloud SQL, or Azure Database

* 1. DETAIL DESCRIPTION OF TECHNOLOGY USED

Using HTML and CSS, the Boutique Management System will deliver a professional, user-friendly interface tailored for boutique staff and customers. Let me know if you’d like to add interactivity with JavaScript or advanced frontend tools.

* **HTML (HyperText Markup Language):::**

HTML (HyperText Markup Language) is the standard language used to create and design the structure of web pages. It acts as the backbone of every website, defining the content and layout that is displayed in web browsers.HTML is the standard language used to structure the content of web pages.

**Key Features of HTML**

* **Structure:**

Organizes content into meaningful sections like headers, footers, navigation bars, and main content areas.

* **Flexibility**:

Easily integrates with CSS for styling and JavaScript for interactivity.

* **Accessibility**:

Semantic elements improve usability for screen readers and other assistive technologies.

* **Cross-Browser Compatibility:**

Supported by all major web browsers, ensuring consistent user experiences.

HTML in Boutique Management System

* **HTML will be used to build:**

**User Interfaces**: Inventory management, sales tracking, customer management, and reports.

**Forms:** Data input for adding products, processing sales, and updating inventory.

Product Display: Showcases products with images, descriptions, and prices.

**Purpose in the System**

**Page Structure:**

Defines the layout and elements such as headers, footers, navigation bars, forms, and tables.

Ensures semantic structure for accessibility (e.g., <header>, <section>, <article>).

**User Interaction:**

Provides the foundation for interactive elements like buttons, forms, and links.

**Integration:**

Embeds multimedia (e.g., images and videos) for showcasing products in the boutique catalog.

* CSS (Cascading Style Sheets):::

CSS is used to style and enhance the appearance of HTML elements.

CSS (Cascading Style Sheets) is a language used to describe the presentation and layout of a web page. It works alongside HTML to style and format content, making websites visually appealing and easier to navigate.

**Key Characteristics of CSS**

* **Styling Language:**

Controls the look and feel of web pages, such as colors, fonts, and spacing.

* **Separation of Content and Design:**

Allows developers to keep HTML for structure and CSS for styling, promoting cleaner and more manageable code.

* **Cascading:**

Resolves styling conflicts by prioritizing rules based on specificity, importance, and source order.

* **How CSS Works**

CSS styles are applied to HTML elements using:

**Inline CSS:**

Directly within an HTML tag using the style attribute.

**Internal CSS:**

Defined within a <style> tag in the <head> section of an HTML document.

**External CSS:**

Written in a separate .css file and linked to the HTML document using the <link> tag.

**Purpose in the System**

**Styling:**

Adds colors, fonts, and background images to improve visual appeal.

**Layout Design:**

Manages the positioning of elements on the page using techniques like flexbox, grid, or CSS positioning.

**Responsive Design:**

Ensures compatibility with various devices (e.g., desktops, tablets, and mobile phones).

**Theming:**

Provides consistency across pages by defining reusable styles.

* **PYTHON**

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. It is widely used in web development, data science, machine learning, artificial intelligence, automation, and more.

**Key Characteristics of Python**

* **Simple and Readable:**

Python's syntax is clean and easy to understand, making it beginner-friendly.

* **Dynamically Typed:**

Variable types are determined at runtime, so you don’t need to declare types explicitly.

* **Interpreted Language:**

Python code is executed line by line, allowing for immediate feedback during development.

* **Extensive Libraries:**

Python has a vast standard library and third-party modules for various applications, from data analysis to web development.

* **Cross-Platform:**

Python runs on multiple operating systems, including Windows, macOS, and Linux.

**Applications of Python**

**Web Development:**

Frameworks like Django and Flask enable the creation of dynamic websites.

**Data Science:**

Libraries such as Pandas, NumPy, and Matplotlib make Python ideal for data manipulation and visualization.

**Machine Learning and AI:**

Frameworks like TensorFlow, PyTorch, and scikit-learn power AI applications.

**Automation and Scripting:**

Automates file management, web scraping, and testing tasks.

**Game Development:**

Libraries like Pygame enable the development of 2D games.

**Desktop GUI Applications:**

Libraries like Tkinter, PyQt, or Kivy support GUI development.

* **DJANGO**

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It is well-suited for building web applications, including a Boutique Management System, because it provides built-in features like an admin interface, user authentication, and database handling.

Why Use Django for a Boutique Management System?

* **Rapid Development:**

Django allows developers to build a web application quickly due to its extensive set of pre-built features (e.g., authentication, admin interface).

* **Database Integration:**

Django includes an ORM (Object-Relational Mapping) system that allows easy interaction with the database. It handles tasks like adding, retrieving, and deleting records with minimal SQL.

* **Security:**

Django comes with built-in protections against common security threats like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

* **Scalability:**

Django is built to scale, meaning it can handle a growing boutique's needs, from a small shop to a larger enterprise system.

* **Admin Panel:**

Django’s admin panel provides a powerful interface for managing data and content, making it especially useful for boutique managers who need to update product information, view sales data, or manage employees.

**Components of the Boutique Management System Using Django**

**Models:**

The models.py file defines the data structures, representing the products, customers, orders, and other entities in the system.

**Views:**

Views handle the business logic of the application and interact with models to retrieve or update data.

**Templates**:

Django uses HTML templates to render dynamic content. The template system allows you to easily separate the user interface (UI) from business logic.

**URLs**:

Django uses a URL dispatcher to map URLs to corresponding views. This is essential for creating specific pages like product listings, order history, or customer management.

**Forms**:

Django forms handle user input, such as adding new products, registering customers, or processing orders.

CHAPTER 2 : PROPOSED SYSTEM

2.1 PROPOSED SYSTEM

A proposed system in the context of a Boutique Management System would typically refer to the plan or design for a new and improved software or technology solution to manage the operations of a boutique. The proposed system aims to address the limitations or shortcomings of the existing system and introduce enhancements or new features to streamline processes, improve efficiency, and provide better services.

Here are some key aspects that might be considered in a proposed system for a Boutique Management System:

1].Inventory Management: Implementing a more robust inventory management system to track the stock of products, manage stock levels, and automate reorder processes.

2].Sales and Point of Sale (POS): Enhancing the POS system to facilitate smoother and faster transactions, integrated with inventory to update stock in real-time.

3].Customer Relationship Management (CRM): Introducing or improving CRM functionalities to better understand customer preferences, track purchase history, and enhance customer interactions.

4].Reporting and Analytics: Implementing a reporting system to generate insights into sales trends, popular products, and overall boutique performance. This could aid in strategic decision-making.

5].Online and E-commerce Integration: If the boutique wishes to expand its reach, an online platform or e-commerce integration may be proposed to enable online sales and reach a wider audience.

6].Employee Management: Enhancing features related to employee management, including payroll processing, scheduling, and performance tracking.

7].Security: Implementing robust security measures to protect sensitive customer data, financial transactions, and ensure the overall integrity of the system.

8].Mobile Accessibility: Developing a mobile-friendly application or interface to enable boutique management on-the-go for owners or staff.

9].Multi-location Support: If applicable, adding support for managing multiple boutique locations efficiently.

10].Integration with Other Systems: Ensuring compatibility and integration with other relevant systems or tools, such as accounting software, supplier databases, or CRM systems.

The specifics of the proposed system would depend on the unique needs and goals of the boutique in question. It's crucial to involve stakeholders, including boutique owners, managers, and staff, in the planning and design process to ensure the proposed system aligns with their requirements and expectations.

2.2 OBJECTIVE OF THE SYSTEM

The objective of a Boutique Management System (BMS) is to provide a comprehensive software solution that automates and simplifies various aspects of boutique operations. The system is designed to manage inventory, track sales, handle customer orders, and streamline administrative tasks, improving overall efficiency and customer satisfaction. The primary goal is to enhance the operational workflow of the boutique, ensuring smooth management of the store while improving customer service and supporting business growth.

Below are the key objectives of the Boutique Management System:

* Efficient Inventory Management
* Streamlined Order Processing
* Customer Management and Engagement
* Sales and Financial Reporting
* User Access Control and Security
* Multi-Channel Sales Management
* Easy-to-Use Interface
* Scalability and Flexibility
* Support for Promotions and Discounts

2.3 USER REQUIREMENTS

The user requirements for a Boutique Management System specify the functionalities and features that the system must support to ensure it meets the needs of different users, including customers, staff, managers, and system administrators. These requirements focus on usability, efficiency, and streamlining operations in a boutique, while also improving the shopping experience for customers.

**Here are the key user requirements:**

Customers interact with the system primarily to browse products, make purchases, and manage their accounts. The system should ensure a smooth and satisfying shopping experience.

**Functional Requirements:**

* **Product Catalog:**

View available products, including details like name, description, size, price, and images.

Filter products by categories (e.g., clothing, accessories), size, color, price, etc.

Search for products by keywords, type, or other attributes.

* **Account Management:**

Ability for customers to register, log in, and manage their accounts.

Store personal information such as name, contact details, and shipping address.

Modify account details such as email, phone number, or shipping address.

* **Order Placement:**

Ability to add products to the shopping cart and modify quantities before checkout.

Secure and simple checkout process with options for different payment methods (e.g., credit card, cash on delivery, etc.).

View order summary and total cost before confirming the purchase.

* **Order Tracking:**

Customers can track the status of their orders (e.g., processing, shipped, delivered).

Receive order confirmation, shipping updates, and delivery notifications.

* **Customer Support:**

Access to customer service via chat, email, or phone for inquiries or issues.

Ability to request order cancellations or returns based on store policies.

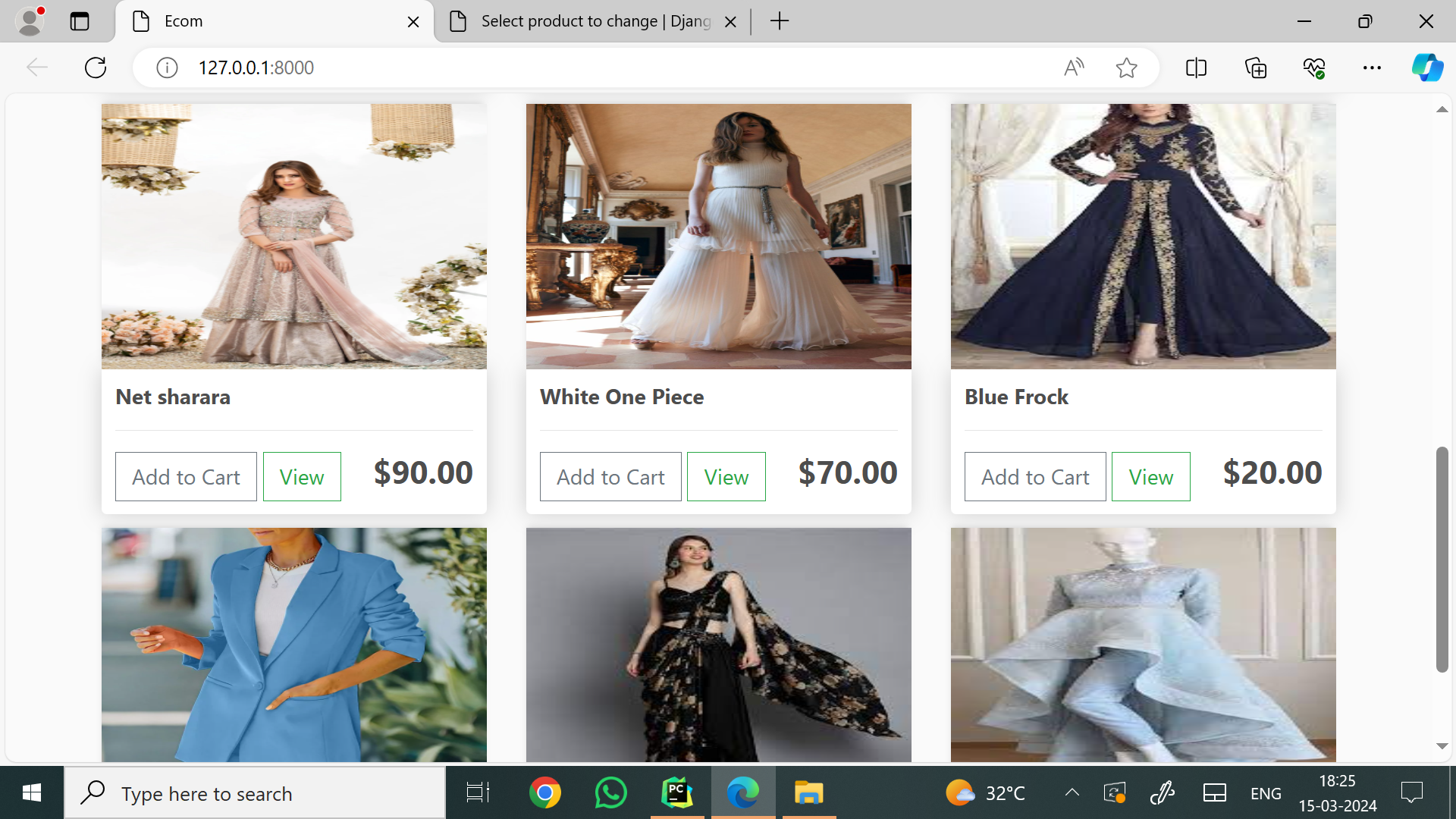
* **Discounts and Loyalty Programs:**

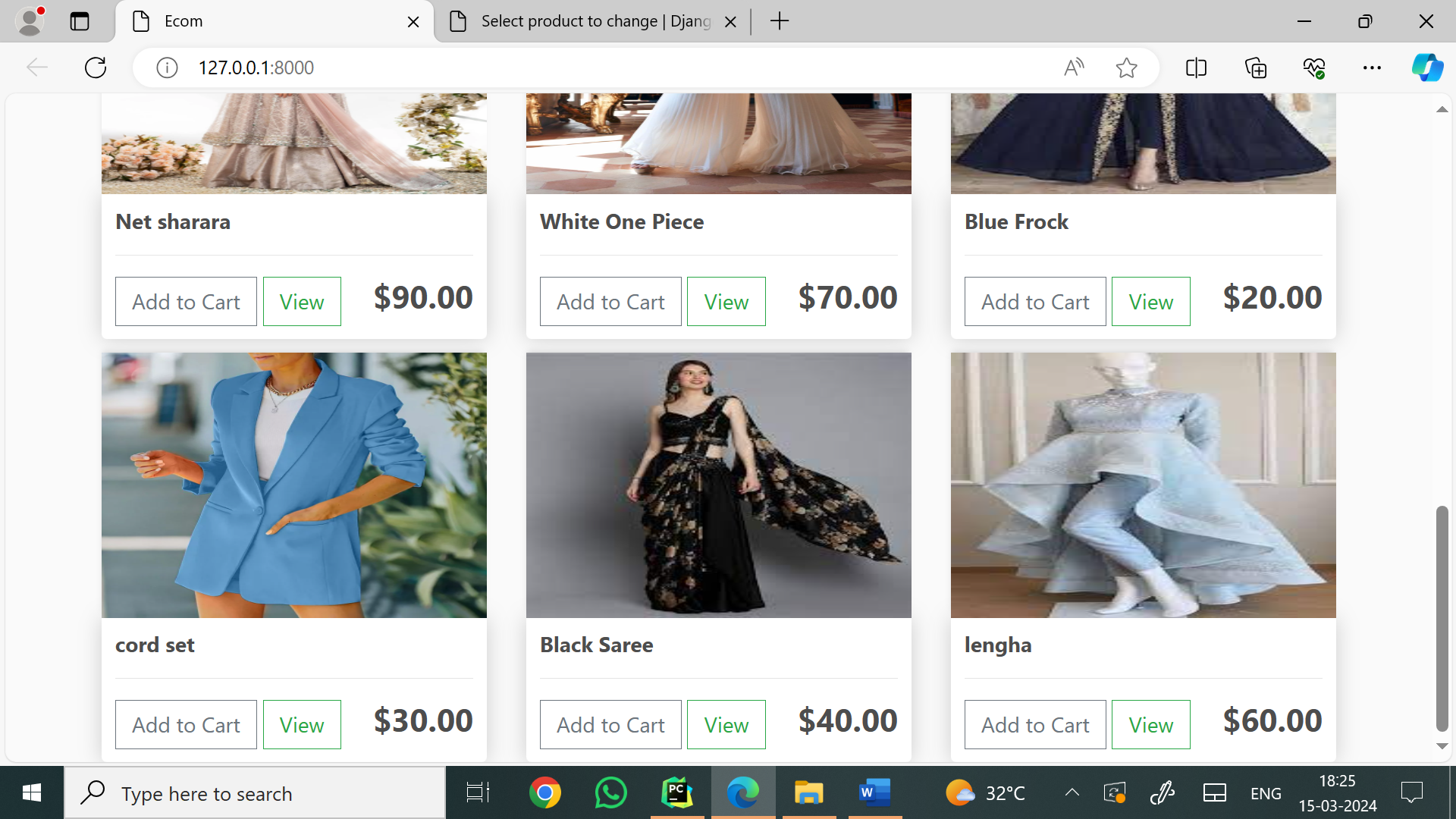
Ability to apply discount codes or promotional offers at checkout.

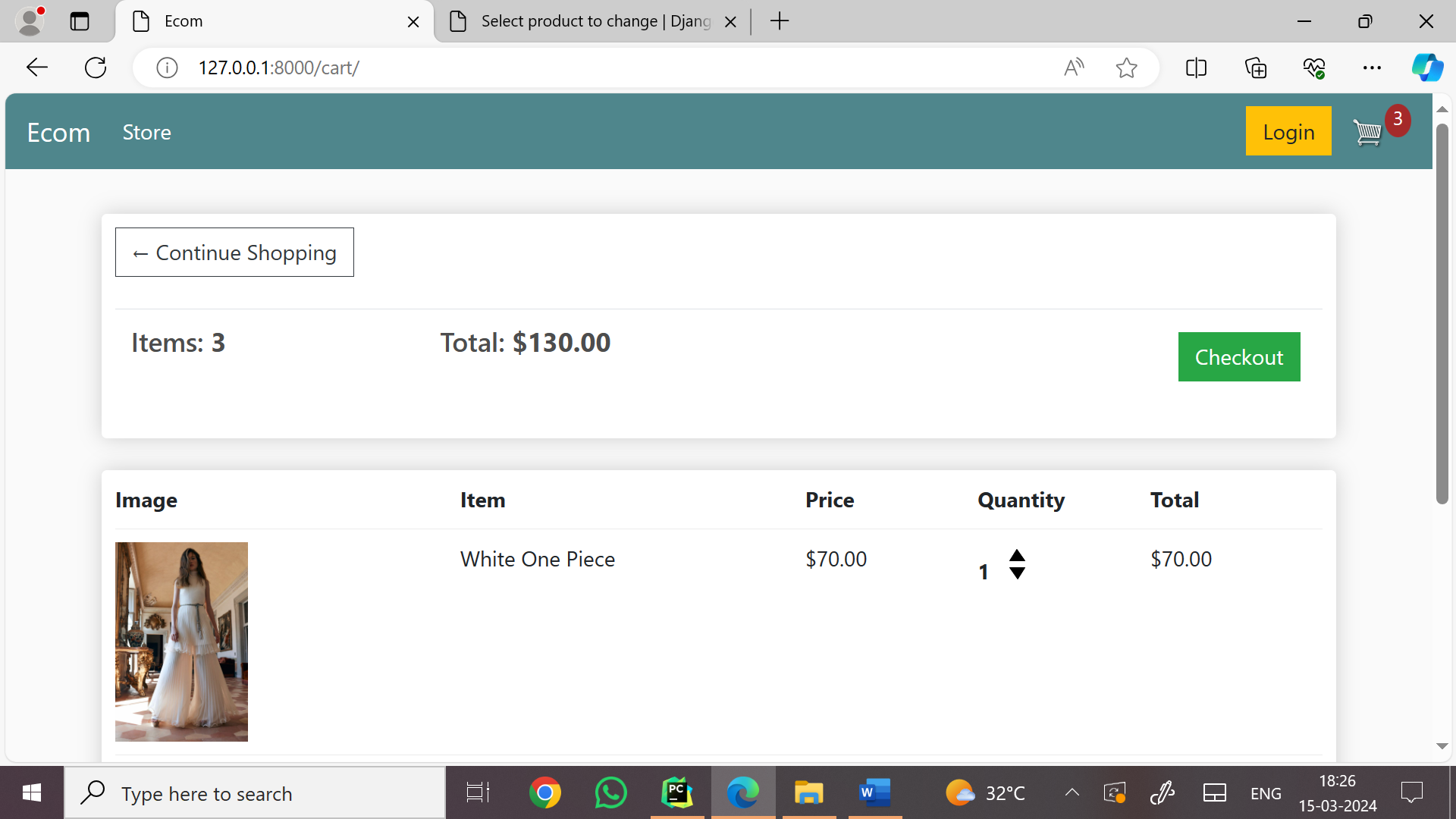
Track and participate in customer loyalty programs for future discounts or rewards.

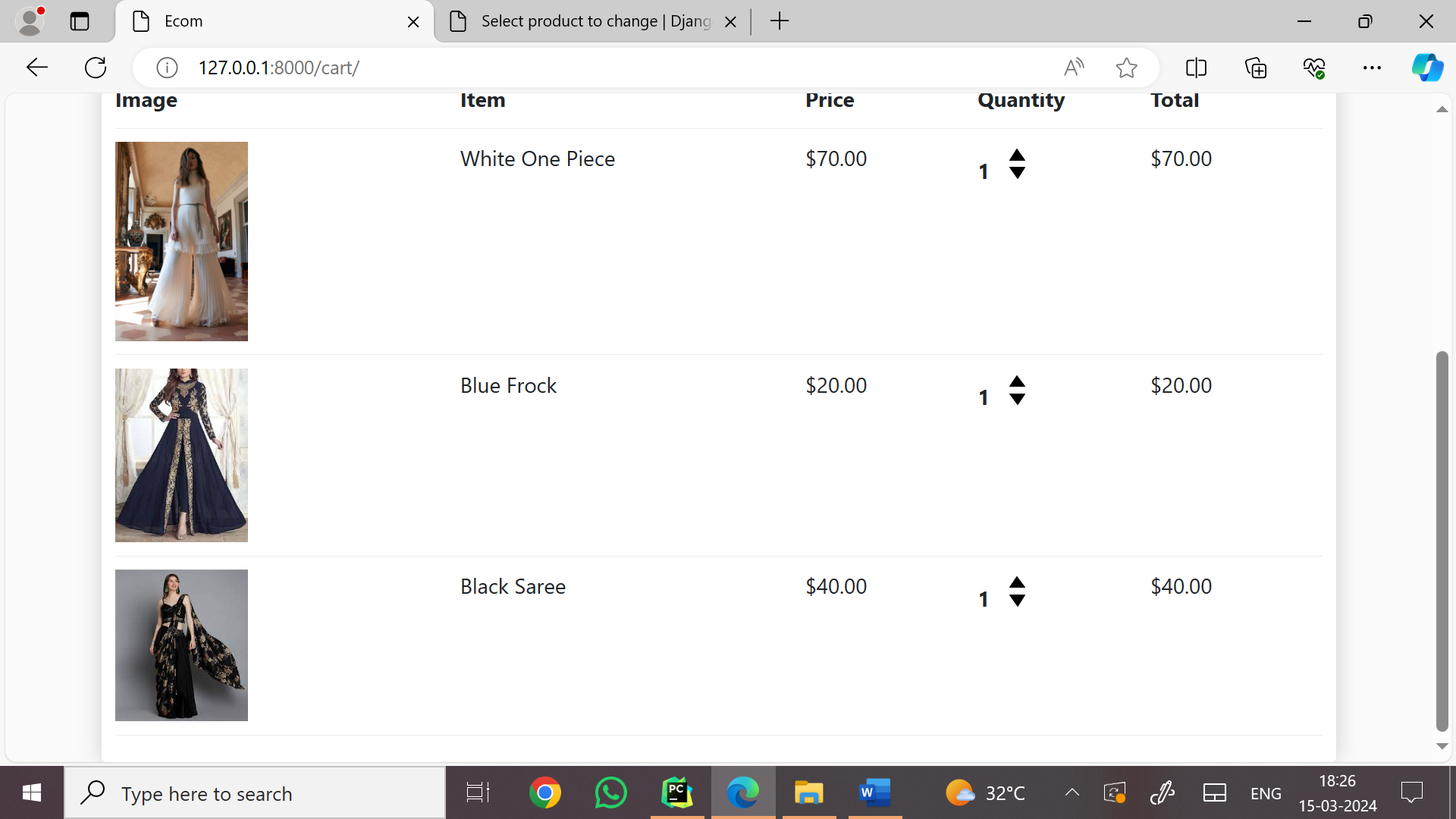
CHAPTER 4 : USER MANUAL

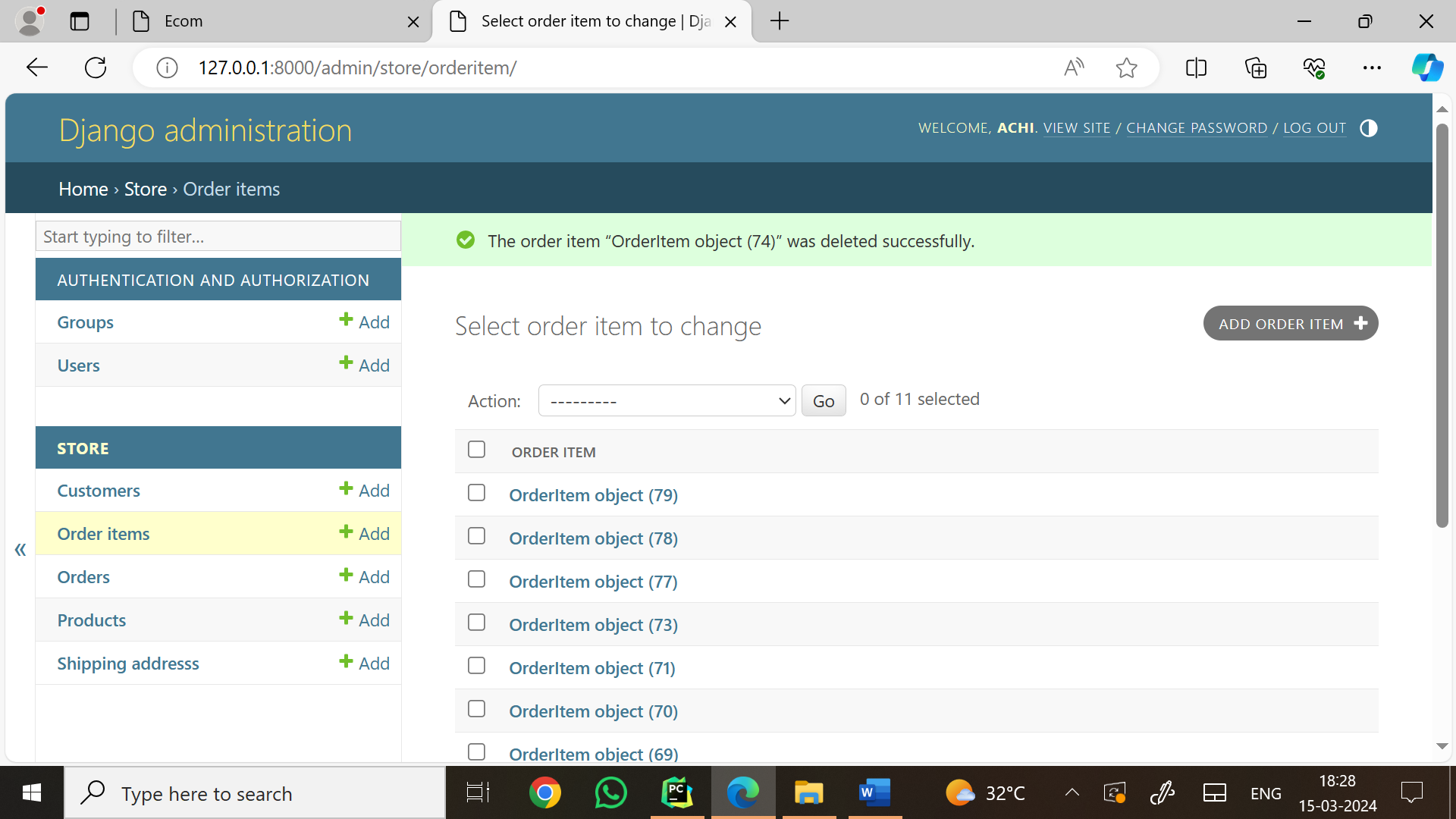
4.1 USER INTERFACE DESIGN (SCREENS ETC.)

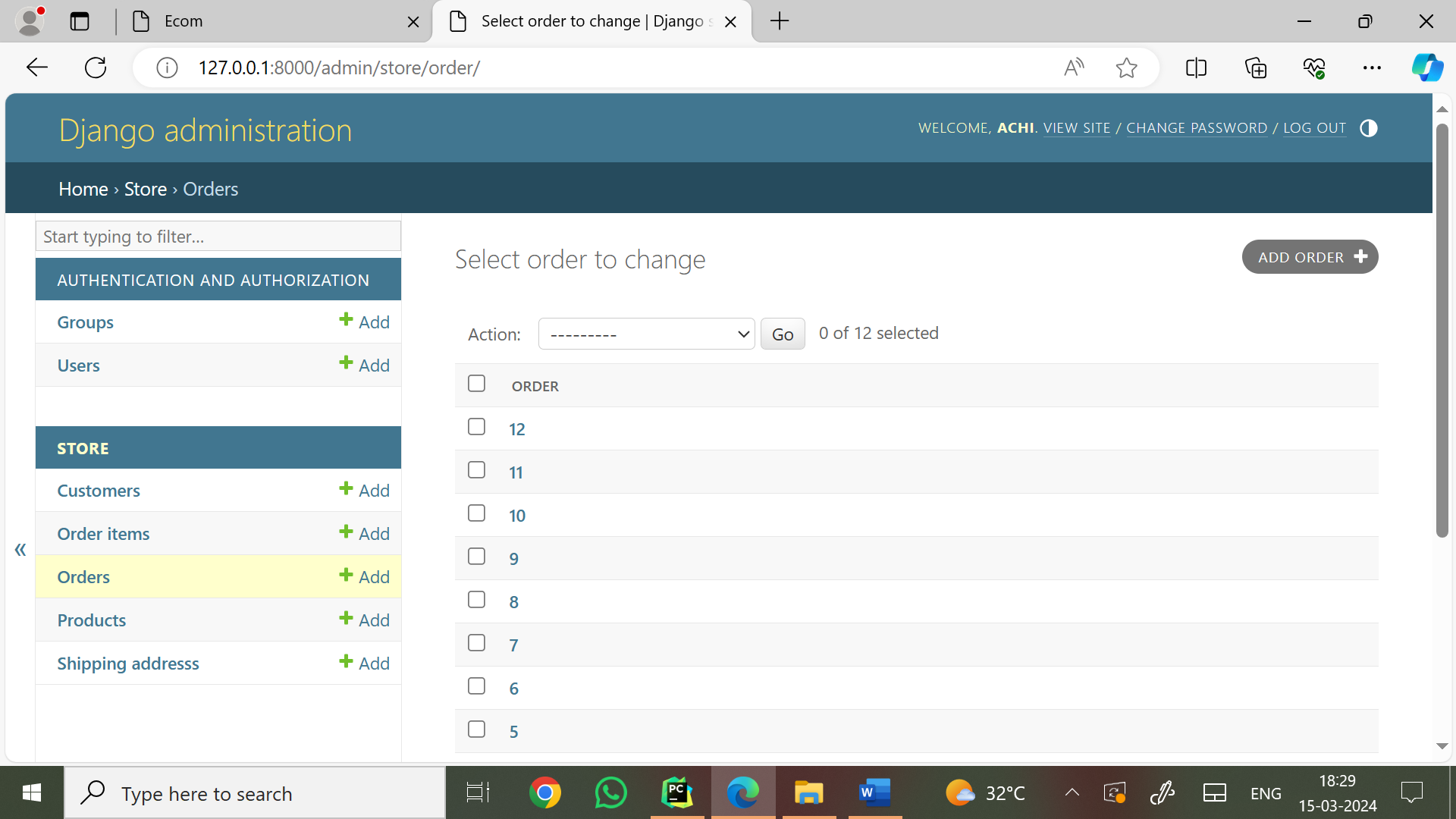


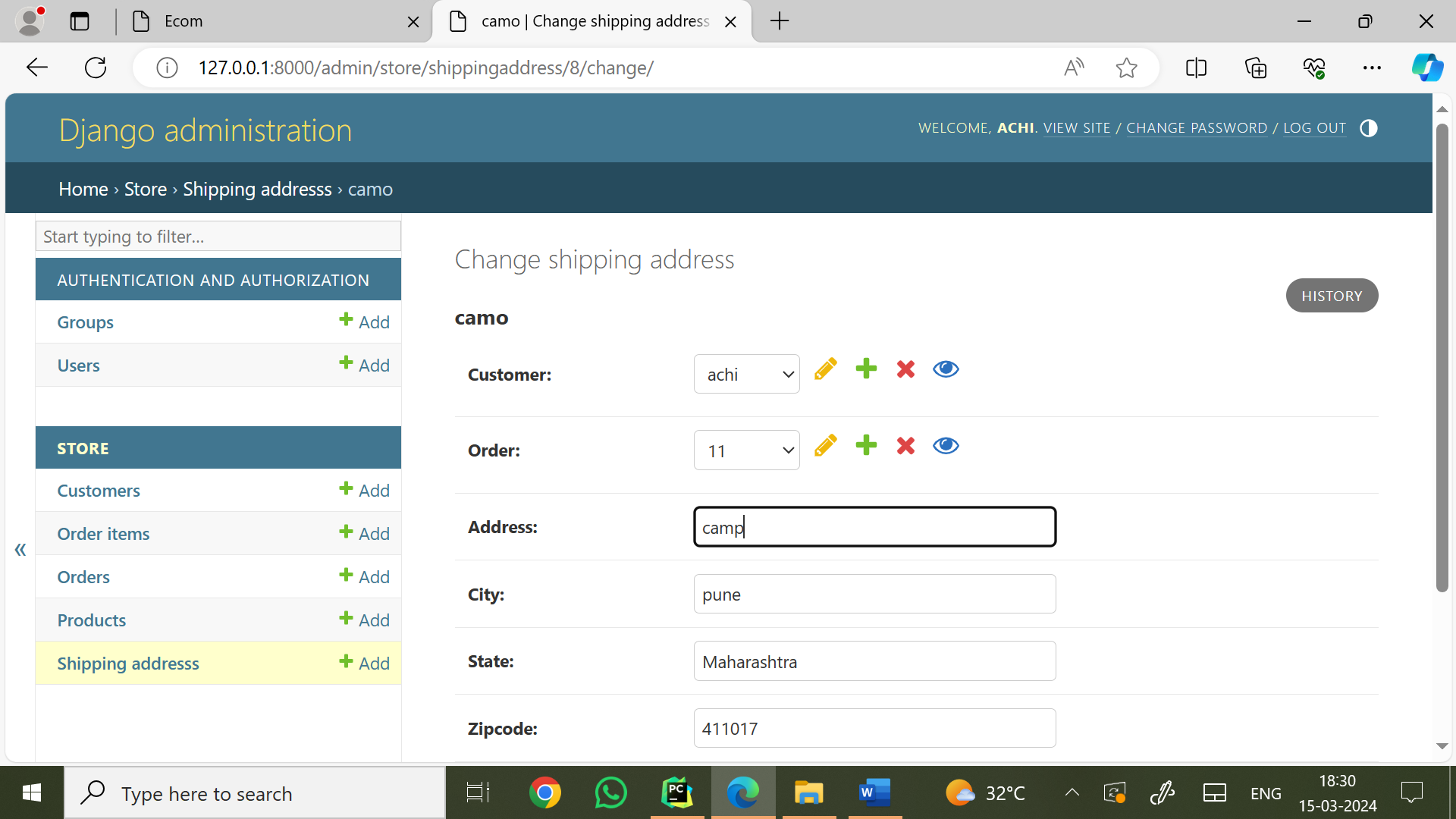












4.2 LIMITATIONS

A boutique management system can be a useful tool for managing inventory, sales, and customer relationships in small-scale retail businesses like boutiques. However, it comes with certain limitations:

1. **Cost Constraints**

High initial setup and subscription costs for advanced systems may be prohibitive for small boutique owners. Customization or upgrades might require additional expenditure.

1. **Complexcity**

Some systems may be complex to learn and operate, requiring training for employees. Overwhelming features can distract from essential functions.

1. **Limited Scalability**

Many boutique systems are designed for small businesses and may not scale well as the business grows or expands to multiple locations. Migrating to a more robust system later can be cumbersome.

1. **Integration Issues**

May not integrate seamlessly with other software like accounting systems, marketing tools, or e-commerce platforms. Manual data transfer between systems increases the risk of errors.

1. **Dependency on Internet**

Cloud-based systems require a stable internet connection. Connectivity issues can disrupt operations. Offline functionality might be limited or unavailable.

1. **Security Risks**

Cybersecurity vulnerabilities can put sensitive business and customer data at risk. Small businesses are less likely to invest in robust security measures.

1. **Limited Customization**

Many boutique management systems offer standardized features that might not cater to unique business requirements. Customization options may be limited or expensive.

1. **Data Accuracy**

Poor data entry or lack of regular updates can lead to inaccurate inventory or sales reports. System errors or bugs may also affect data reliability.

1. **Support and Maintenance**

Access to customer support may be limited, especially for free or low-cost systems.

Regular maintenance and updates might disrupt business operations.

1. **Staff Resistance**

Employees may resist adopting new technology, especially if they are used to manual processes. Lack of user-friendliness can lead to underutilization of the system.

1. **Overdependence on Technology**

A heavy reliance on the system may make the business vulnerable to disruptions caused by technical failures or outages. Addressing these limitations requires careful selection of a boutique management system tailored to the business's size, needs, and budget, as well as proper training and contingency planning.

* 1. FUTURE ENHANCEMENT

Enhancing a Boutique Management System involves adding features and functionalities to improve efficiency, customer satisfaction, and operational effectiveness. Here are some future enhancements you can consider:

1. Customer Experience Enhancements

Personalized Recommendations: Use AI to suggest products based on customer preferences, purchase history, and trending items.

Virtual Try-On: Integrate augmented reality (AR) for customers to virtually try clothing or accessories.

Loyalty Program: Implement a points-based system to reward repeat customers.

Feedback System: Enable customers to provide feedback on their purchases and shopping experience.

2. Inventory Management

Real-Time Tracking: Enable real-time stock updates across multiple locations.

Smart Restocking Alerts: Use predictive analytics to alert when stock is low or in high demand.

Barcode/QR Integration: Simplify stock management with barcode or QR code scanning.

Batch and Expiry Management: For items with limited shelf life, track batch numbers and expiry dates.

3. Sales and Marketing

Omni-Channel Integration: Sync online and in-store inventory for a seamless shopping experience.

Promotional Campaigns: Create automated marketing campaigns based on seasonality or customer segments.

Social Media Integration: Enable direct selling through platforms like Instagram or Facebook.

Analytics Dashboard: Provide detailed insights into sales trends, customer demographics, and performance metrics.

4. Employee and Vendor Management

Task Automation: Automate routine tasks like generating reports and sending reminders.

Performance Monitoring: Track employee performance and provide training suggestions.

Vendor Management: Manage vendor contracts, orders, and payments more efficiently.

5. Technological Integration

Cloud-Based Access: Allow secure, remote access to the system from anywhere.

Mobile App: Create an app for boutique owners and employees to manage the system on the go.

AI Chatbots: Use chatbots to assist customers with queries, product search, and order placements.

Blockchain Integration: Enhance transparency and security in payments and inventory tracking.

6. Payment and Billing

Multiple Payment Options: Include mobile wallets, cryptocurrencies, and BNPL (Buy Now, Pay Later) options.

Dynamic Pricing: Implement pricing adjustments based on demand and inventory levels.

Automated Invoicing: Simplify billing with automated invoice generation.

7. Customer Support

24/7 Chat Support: Provide round-the-clock support using chatbots and live agents.

Order Tracking: Let customers track their orders in real time.

Complaint Management: Create a streamlined system for handling complaints and returns.

8. Sustainability Features

Eco-Friendly Reporting: Track and report on the sustainability of materials used in products.

Second-Hand Selling: Enable the resale of boutique items through the platform.

Carbon Footprint Tracker: Show customers the environmental impact of their purchases.

9. Advanced Security

Two-Factor Authentication: Add extra security layers for account access.

Role-Based Access Control: Restrict access to sensitive areas of the system based on user roles.

Data Encryption: Protect customer and business data with end-to-end encryption.

10. Customizable Modules

Allow boutiques to tailor the system based on their unique needs, such as adding specific product categories or services.

**BIBLIOGRAPHY**

**ANNEXURE:** Sample program code

<!DOCTYPE html>  
{% load static %}  
<html>  
<head>  
 <title>Ecom</title>  
  
 <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, minimum-scale=1" />  
  
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">  
  
 <script type="text/javascript">  
 var *user* = '{{request.user}}'  
  
 function getToken(name) {  
 var cookieValue = null;  
 if (*document*.cookie && *document*.cookie !== '') {  
 var cookies = *document*.cookie.split(';');  
 for (var i = 0; i < cookies.length; i++) {  
 var cookie = cookies[i].trim();  
 // Does this cookie string begin with the name we want?  
 if (cookie.substring(0, name.length + 1) === (name + '=')) {  
 cookieValue = decodeURIComponent(cookie.substring(name.length + 1));  
 break;  
 }  
 }  
 }  
 return cookieValue;  
 }  
 var *csrftoken* = getToken('csrftoken')  
  
 </script>  
  
 <style>  
 body{  
 background-color: hsl(0, 0%, 98%);  
}  
  
h1,h2,h3,h4,h5,h6{  
 color:hsl(0, 0%, 30%);  
}  
  
.box-element{  
 box-shadow:hsl(0, 0%, 80%) 0 0 16px;  
 background-color: #fff;  
 border-radius: 4px;  
 padding: 10px;  
}  
  
.thumbnail{  
 width: 100%;  
 height: 200px;  
 -webkit-box-shadow: -1px -3px 5px -2px rgba(214,214,214,1);  
 -moz-box-shadow: -1px -3px 5px -2px rgba(214,214,214,1);  
 box-shadow: -1px -3px 5px -2px rgba(214,214,214,1);  
}  
  
.product{  
 border-radius: 0 0 4px 4px;  
}  
  
.bg-dark{  
 background-color: #4f868c**!important**;  
}  
  
#cart-icon{  
 width:25px;  
 display: inline-block;  
 margin-left: 15px;  
}  
  
#cart-total{  
 display: block;  
 text-align: center;  
 color:#fff;  
 background-color: brown;  
 width: 20px;  
 height: 25px;  
 border-radius: 50%;  
 font-size: 14px;  
}  
  
.col-lg-4, .col-lg-6, .col-lg-8, .col-lg-12{  
 margin-top: 10px;  
}  
  
.btn{  
 border-radius: 0;  
}  
  
.row-image{  
 width: 100px;  
}  
  
.form-field{  
 width:200px;  
 display: inline-block;  
 padding: 5px;  
}  
  
.cart-row{  
 display: flex;  
 align-items: stretch;  
 padding-bottom: 10px;  
 margin-bottom: 10px;  
 border-bottom: 1px solid #ececec;  
  
}  
  
.quantity{  
 display: inline-block;  
 font-weight: 700;  
 padding-right:10px;  
  
  
}  
  
.chg-quantity{  
 width: 12px;  
 cursor: pointer;  
 display: block;  
 margin-top: 5px;  
 transition:.1s;  
}  
  
.chg-quantity:hover{  
 opacity: .6;  
}  
  
  
.hidden{  
 display: none**!important**;  
}  
</style>  
</head>  
<body>  
  
 <nav class="navbar navbar-expand-lg navbar-dark bg-dark">  
 <a class="navbar-brand" href="{% url 'store' %}">Ecom</a>  
 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">  
 <span class="navbar-toggler-icon"></span>  
 </button>  
  
 <div class="collapse navbar-collapse" id="navbarSupportedContent">  
 <ul class="navbar-nav mr-auto">  
 <li class="nav-item active">  
 <a class="nav-link" href="{% url 'store' %}">Products <span class="sr-only">(current)</span></a>  
 </li>  
  
 </ul>  
 <div class="form-inline my-2 my-lg-0">  
 <a href="#"class="btn btn-warning">Login</a>  
  
 <a href="{% url 'cart' %}">  
 <img id="cart-icon" src="{% static 'images/cart.png' %}">  
 </a>  
 <p id="cart-total">{{cartItems}}</p>  
  
 </div>  
 </div>  
 </nav>  
  
 <div class="container">  
 <br>  
 {% block content %}  
  
  
 {% endblock content %}  
 </div>  
  
  
 <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>  
  
 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>  
  
 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>  
  
<script type="text/javascript" src="{% static 'js/cart.js' %}"></script>  
</body>  
</body>  
</html>