

Design and Implementation of a Clothing E-Commerce Web Application

Team Members: Adham Walid , Omar El Farouk , Youssef Shamy

NEW GIZA UNIVERSITY



Abstract

This project report details the design and implementation of a clothing e-commerce web application. The platform allows users to browse products, manage a shopping cart and wishlist, place orders, and track purchases. The backend architecture ensures secure transactions, while the frontend provides a seamless user experience. The project incorporates advanced features like dynamic product filtering, order management, and prompt-engineered GUI development.

Table of Contents

1. Introduction
2. Requirements
3. System Design
4. GUI Development Using Prompt Engineering
5. Implementation
6. Testing and Validation
7. Future Enhancements
8. Conclusion

1. Introduction

1.1 Purpose

The purpose of this project is to create a comprehensive e-commerce platform for clothing, providing users with convenient and secure shopping experience. It facilitates browsing, cart management, order placement, and Wishlist functionalities.

1.2 Scope

- Core functionalities include user authentication, product browsing, cart management, and secure checkout.
- Advanced features such as wishlists and dynamic filtering enhance the user experience.
- Scalability and cross-platform compatibility ensure the platform meets future growth demands.

2. Requirements

2.1 Functional Requirements

ID	Requirement	Category
REQ_001	The system must allow users to register and log in using email.	Must Have
REQ_002	The system must allow users to browse clothing items .	Must Have
REQ_003	The system must display product details, including images, sizes, colors, materials, Price.	Must Have
REQ_004	The system must allow users to add products to a shopping cart and review/edit the cart before checkout.	Must Have
REQ_005	The system must support secure payment options, including credit/debit cards, e-wallets, and Cash on Delivery (COD).	Must Have
REQ_006	The system must allow users to create and manage wishlists.	Must Have
REQ_007	The system must allow users to track the status of their orders (e.g., Processing, Shipped, Delivered).	Must Have
REQ_008	The system must allow admins to add, update, or remove products, prices, and discounts.	Must Have
REQ_009	The system won't allow purchases to proceed without user authentication.	Won't Have

2.2 Non-Functional Requirements

ID	Requirement	Category
REQ_010	The system must handle at least 1000 concurrent users during peak traffic.	Performance
REQ_011	The system must secure user data with encryption hashed for passwords.	Security
REQ_012	The system must load product pages within 2 seconds under normal conditions.	Performance
REQ_013	The system should have 99.9% uptime annually.	Reliability
REQ_014	The system must support both web and mobile platforms (responsive design and mobile app).	Usability
REQ_015	The system must include an intuitive user interface with easy navigation and clear visuals.	Usability
REQ_016	The system should log all key user actions (e.g., login, purchases, reviews) for auditing.	Security

2.3 Use case scenarios for functional and non-functional Requirements

Use Case ID: UC-001

Title: User Registration and Login

Preconditions: The user is on the platform's registration or login page.

Postconditions: The user is successfully logged in.

Main Flow:

1. The user accesses the platform and clicks "Sign Up" or "Log In."
2. The system prompts the user for email, phone number, or social media credentials.
3. The user enters the credentials.
4. The system validates the input and authenticates the user.
5. The user is logged in successfully.

Alternative Flow:

If the user opts for "Forgot Password," the system sends a password reset email.

Exception Flow:

If the entered credentials are invalid, the system displays an error message and prompts the user to try again.

Related Requirements: REQ_01

Use Case ID: UC-002**Title:** Browse Clothing Items**Preconditions:** The user is on the platform's homepage.**Postconditions:** The user can view a list of clothing items.**Main Flow:**

1. The user navigates to the "Clothing" section.
2. The system displays a list of available clothing items.
3. The user can filter or sort the items by category, size, color, or price.
4. The user selects an item to view more details.

Alternative Flow:

If no items are available, the system displays a message indicating no results.

Exception Flow:

If the system fails to load items, an error message is displayed.

Related Requirements: REQ_02**Use Case ID:** UC-003**Title:** View Product Details**Preconditions:** The user has selected a clothing item.**Postconditions:** The user can see detailed information about the product.**Main Flow:**

1. The user clicks on a product from the list.
2. The system displays product details, including images, sizes, colors, materials, and price.
3. The user can choose to add the product to the shopping cart or wishlist.

Alternative Flow:

If the product is out of stock, the system displays a notification.

Exception Flow:

If the product details fail to load, an error message is shown.

Related Requirements: REQ_03**Use Case ID:** UC-004**Title:** Add Product to Shopping Cart**Preconditions:** The user is viewing a product's details.**Postconditions:** The product is added to the user's shopping cart.**Main Flow:**

1. The user selects the desired size, color, and quantity.
2. The user clicks "Add to Cart."
3. The system adds the product to the shopping cart and updates the cart total.
4. The user can proceed to checkout or continue shopping.

Alternative Flow:

If the user is not logged in, the system prompts for login or registration.

Exception Flow:

If the product is no longer available, the system notifies the user.

Related Requirements: REQ_04

Use Case ID: UC-005**Title:** Secure Payment Processing**Preconditions:** The user has items in the shopping cart and proceeds to checkout.**Postconditions:** The payment is processed securely.**Main Flow:**

1. The user selects a payment method (credit/debit card, e-wallet, COD).
2. The system prompts for payment details.
3. The user enters the required information.
4. The system validates and processes the payment.
5. The user receives a confirmation of the successful transaction.

Alternative Flow:

If the payment fails, the system prompts the user to try another method.

Exception Flow:

If the system detects fraudulent activity, the transaction is blocked.

Related Requirements: REQ_05**Use Case ID:** UC-006**Title:** Manage Wishlist**Preconditions:** The user is logged in and viewing a product.**Postconditions:** The product is added to or removed from the user's wishlist.**Main Flow:**

1. The user clicks "Add to Wishlist" on a product page.
2. The system adds the product to the user's wishlist.
3. The user can view and manage their wishlist from their account page.

Alternative Flow:

If the product is already in the wishlist, the user can remove it.

Exception Flow:

If the system fails to update the wishlist, an error message is displayed.

Related Requirements: REQ_06**Use Case ID:** UC-007**Title:** Track Order Status**Preconditions:** The user has placed an order.**Postconditions:** The user can view the current status of their order.**Main Flow:**

1. The user navigates to the "Order History" section.
2. The system displays a list of the user's orders with their current status (e.g., Processing, Shipped, Delivered).
3. The user selects an order to view detailed tracking information.

Alternative Flow:

If the order status is delayed, the system provides an estimated delivery update.

Exception Flow:

If the order status cannot be retrieved, the system displays an error message.

Related Requirements: REQ_07

Use Case ID: UC-008

Title: Admin Manage Products

Preconditions: The admin is logged into the admin dashboard.

Postconditions: Products, prices, or discounts are updated or removed.

Main Flow:

1. The admin navigates to the "Product Management" section.
2. The admin selects to add, update, or remove a product.
3. The system processes the changes and updates the product catalog.

Alternative Flow:

If the admin updates a product's price, the system applies the change to all relevant orders.

Exception Flow:

If the system fails to save changes, an error message is displayed.

Related Requirements: REQ_08

Use Case ID: UC-009

Title: User Authentication Required for Purchase

Preconditions: The user attempts to proceed to checkout.

Postconditions: The user must log in or register to complete the purchase.

Main Flow:

1. The user clicks "Proceed to Checkout."
2. The system checks if the user is logged in.
3. If not logged in, the system redirects the user to the login or registration page.
4. After successful authentication, the user can complete the purchase.

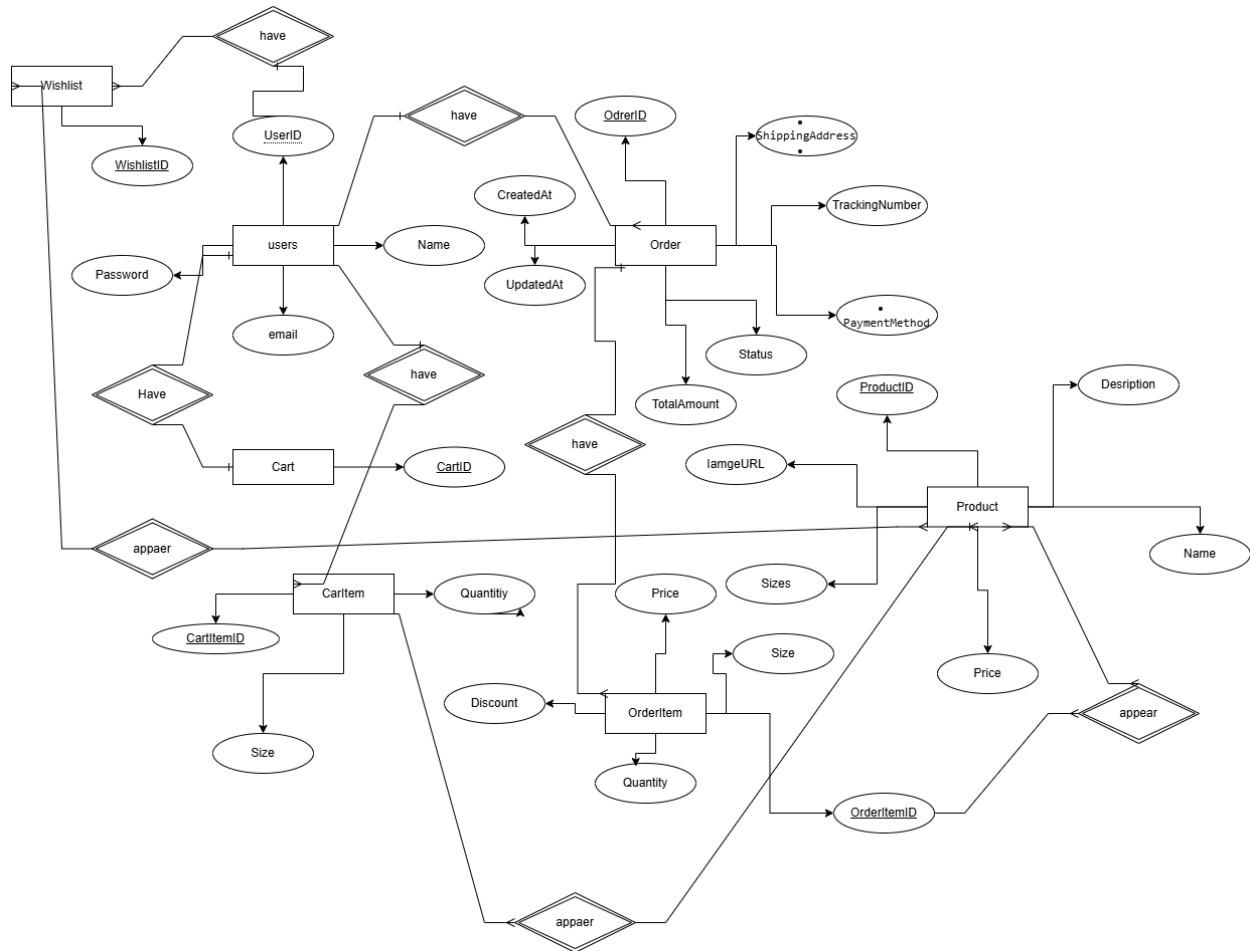
Alternative Flow:

If the user opts to register, they are guided through the registration process.

Exception Flow:

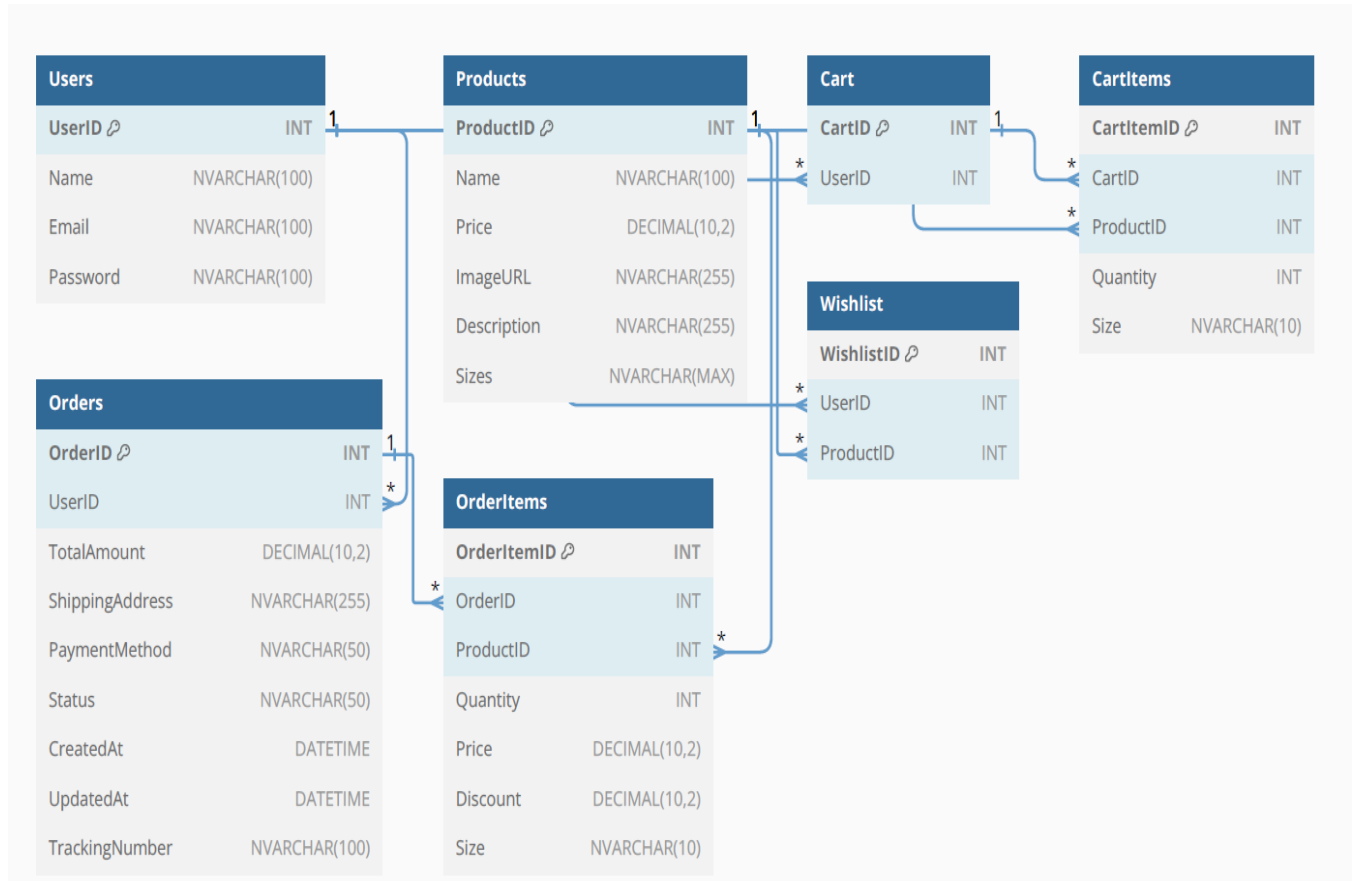
If authentication fails, the system displays an error message.

Related Requirements: REQ_09



3.2 Relational Database Schema

The relational schema defines tables such as Users, Products, Orders, CartItems, and Wishlist. Each table has well-defined attributes and relationships, aligning with the ERD



4. SQL Schema and Queries

Table: Users

Column Name	Data Type	Constraints
UserID	INT	PRIMARY KEY, IDENTITY
Name	NVARCHAR(100)	NOT NULL
Email	NVARCHAR(100)	NOT NULL, UNIQUE
Password	NVARCHAR(100)	NOT NULL

Table: Products

Column Name	Data Type	Constraints
ProductID	INT	PRIMARY KEY, IDENTITY
Name	NVARCHAR(100)	NOT NULL
Price	DECIMAL(10, 2)	NOT NULL
ImageURL	NVARCHAR(255)	NOT NULL
Description	NVARCHAR(255)	NOT NULL
Sizes	NVARCHAR(MAX)	NOT NULL

Table: Cart

Column Name	Data Type	Constraints
CartID	INT	PRIMARY KEY, IDENTITY
UserID	INT	FOREIGN KEY REFERENCES Users(UserID)

Table: CartItems

Column Name	Data Type	Constraints
CartItemID	INT	PRIMARY KEY, IDENTITY
CartID	INT	FOREIGN KEY REFERENCES Cart(CartID)
ProductID	INT	FOREIGN KEY REFERENCES Products(ProductID)
Quantity	INT	NOT NULL DEFAULT 1
Size	NVARCHAR(10)	NOT NULL

Table: Orders

Column Name	Data Type	Constraints
OrderID	INT	PRIMARY KEY, IDENTITY
UserID	INT	FOREIGN KEY REFERENCES Users(UserID)
TotalAmount	DECIMAL(10, 2)	NOT NULL
ShippingAddress	NVARCHAR(255)	NOT NULL
PaymentMethod	NVARCHAR(50)	CHECK (PaymentMethod IN ('Visa', 'COD'))
Status	NVARCHAR(50)	CHECK (Status IN ('Delivered', 'Shipped', 'Pending'))
CreatedAt	DATETIME	DEFAULT GETDATE()
UpdatedAt	DATETIME	DEFAULT GETDATE()
TrackingNumber	NVARCHAR(100)	NULL

Table: Wishlist

Column Name	Data Type	Constraints
WishlistID	INT	PRIMARY KEY, IDENTITY
UserID	INT	FOREIGN KEY REFERENCES Users(UserID)
ProductID	INT	FOREIGN KEY REFERENCES Products(ProductID)

5. GUI Development Using Prompt Engineering

5.1 Stage 1: Initial Prototype

Prompt: "Create a basic website for a clothing e-commerce platform with pages for home, products, login, signup, and cart. Ensure basic layout with simple navigation elements such as a top navigation bar, product grid, and footer. The product page should have placeholder images and product details. Include a user-friendly layout with clear call-to-action buttons like 'Add to Cart,' 'Login,' and 'Sign Up.'"

Techniques Used:

- 1. Clear and Actionable Instruction:** The prompt clearly instructs to "create a basic website," focusing on essential elements (pages, navigation, and product display).
- 2. Breakdown of Core Features:** The prompt outlines specific tasks: create pages for home, products, login, signup, and cart. This gives clear guidelines on the expected content and structure.
- 3. User-Centered Focus:** The prompt highlights the importance of "user-friendly layout" and includes call-to-action buttons like 'Add to Cart,' 'Login,' and 'Sign Up' to guide user interaction.
- 4. Minimalism and Clarity:** The prompt avoids overwhelming details and emphasizes simplicity by focusing on a basic layout with placeholders for prototyping.
- 5. Testable Outcome:** The deliverables (clear navigation, placeholder images, and buttons) can be easily assessed for completeness.

5.2 Stage 2: Adding Basic Interactivity

Prompt: "Enhance the website to include interactive elements like product filtering by category, price, and size. Implement dropdown menus or checkboxes for filtering, allowing users to easily refine their search results. Ensure the product grid dynamically updates to reflect user selections. Also, ensure the design is responsive, adjusting seamlessly to different screen sizes and devices."

Techniques Used:

1. **Incremental Complexity:** This stage adds complexity by introducing interactivity (product filtering) that builds upon the previous stage, creating a natural progression from a static to a dynamic website.
2. **Explicit Functional Requirements:** The prompt details the expected functionalities (dropdown menus, checkboxes for filtering) and specifies that the product grid should update dynamically without page reloads.
3. **Real-Time Action and Result:** The prompt emphasizes the dynamic nature of the site, specifically the real-time updates to the product grid based on user selections.
4. **Design Consistency:** The prompt focuses on maintaining a responsive design, ensuring the layout adapts to different devices. This encourages attention to consistency in user experience across platforms.
5. **User Interface (UI) Design:** Clear instructions for designing UI components (dropdowns, checkboxes, sliders) to make the filtering process intuitive.
6. **Testable Outcome:** The success of this stage can be verified by testing the interactive features (filtering) and responsiveness (layout adjustment).

5.3 Stage 3: Integrating User Accounts

Prompt: "Allow users to log in to the platform, view their cart, and manage their wishlist. Create a user authentication system with login and registration functionality. Once logged in, users should be able to view and modify their cart and wishlist. Ensure the implementation of session management to persist user login states across page reloads and sessions."

Techniques Used:

1. **Task Decomposition:** This prompt breaks down the overall task into smaller, manageable sections: user login, cart management, and wishlist functionality. This makes the process more digestible and easier to follow.
2. **Functional and User-Oriented Language:** The prompt focuses on the features that are most relevant to users, such as viewing and modifying the cart and wishlist, which are core to e-commerce sites.

3. **Contextual Instructions:** By specifying "session management" and persistence of login states across page reloads, the prompt directs the model to consider both the user experience and the technical requirements for user data retention.
4. **Systematic Flow:** The prompt encourages the creation of a logical flow for the user experience, ensuring that the user progresses from logging in to managing their cart and wishlist seamlessly.
5. **Testable Outcome:** The ability to check user login persistence and cart/wishlist functionality makes the output easily testable.

5.4 Stage 4: Finalizing Design

Prompt: "Refine the website's design with a modern, clean aesthetic, and seamless navigation. Implement a cohesive color scheme and typography for better branding. Add animations and transitions for a smoother user experience. Enhance UI elements such as buttons, inputs, and product cards with hover and focus effects. Ensure a consistent layout across all pages, and test for accessibility and responsiveness."

Techniques Used:

1. **Comprehensive Detail:** The prompt provides extensive instructions covering various aspects (design aesthetics, animations, transitions, and responsiveness), ensuring a polished final result.
2. **Focus on User Experience:** The language emphasizes a "smoother user experience" and "seamless navigation," guiding the model to focus on usability and interaction flow.
3. **Design Coherence:** The prompt stresses a cohesive visual identity, including color schemes and typography, which is vital for branding and user experience.
4. **Testing and Optimization:** The prompt introduces a focus on accessibility and responsiveness, ensuring that the design works across various devices and adheres to web accessibility standards.
5. **UI/UX Design:** The inclusion of hover and focus effects enhances the interactive nature of UI elements, encouraging attention to detail in the user interface.
6. **Cross-Browser Compatibility:** By emphasizing the need for consistent layout across pages, the prompt guides the model to ensure that the design works in various browsers and devices.
7. **Testable Outcome:** The success of the design refinement can be easily tested through cross-device checks, usability testing, and accessibility audits.

6. Implementation

6.1 Frontend Development

- Utilized HTML, CSS, and JavaScript for responsive design.
- Dynamic product loading using API calls.

6.2 Backend Development

- Built using Node.js and Express.js.
- Integrated MongoDB for database management.
- Implemented RESTful APIs for CRUD operations and secure payment handling.

7. Testing and Validation

7. Test Plan

7.1 Objectives

- Verify the functionality of user registration, login, product management, cart, wishlist, and order placement.
- Ensure integration between modules (e.g., cart, order, and payment gateway).
- Validate both functional and non-functional requirements (e.g., performance, security).

7.2 Scope

- **In Scope:**
 - User registration and login.
 - Product management (add, delete) by admin.
 - Cart and wishlist functionality.
 - Order placement and payment integration.
 - End-to-end workflows for user and admin actions.
 - Non-functional testing (performance, security).
- **Out of Scope:**
 - Third-party integrations (e.g., email notifications, external APIs).
 - Mobile application testing.

7.3 Test Strategy

- **Manual Testing:**
 - Used for exploratory testing, usability testing, and scenarios requiring human judgment.
 - Applied to test cases such as user registration, login, and end-to-end workflows.
- **Automation Testing:**
 - Used for repetitive tasks, regression testing, and integration testing.
 - Applied to test cases such as cart integration, order placement, and payment gateway testing.

7.4 Test Environment

- **OS:** Windows 10
- **Browser:** Chrome 116
- **Application Version:** 1.0

7.5 Test Deliverables

- Test cases document.
- Defect report.
- Test execution summary.
- Final test report.

7.6 Test Cases

7.6.1 Manual Test Cases

Manual testing will be performed to validate the following scenarios:

- **User Registration and Login:**
 - Verify that a new user can register successfully (TC_002).
 - Verify that a registered user can log in successfully (TC_001).
- **Product Management:**
 - Verify that a user can add and remove products from the cart (TC_003, TC_004).
 - Verify that a user can add and remove products from the wishlist (TC_008, TC_009).
 - Verify that an admin can add and delete products (TC_006, TC_007).
- **Order Placement:**
 - Verify that a user can place an order successfully (TC_005).
 - Verify that the cart module integrates with the order module (TC_012).
 - Verify that the order module integrates with the payment gateway (TC_013).
- **End-to-End Workflows:**
 - Verify the complete workflow from product selection to order placement (TC_014).
 - Verify the workflow for user registration and login (TC_015).
- **Functional and Non-Functional Requirements:**
 - Verify that all functional requirements are met (TC_017).
 - Verify that the website meets non-functional requirements such as performance and security (TC_018).

7.6.2 Automation Testing

Automation testing will be implemented for the following scenarios to ensure repeatability and efficiency:

- **User Registration and Login:**
 - Automate the registration and login process to ensure consistent behavior across multiple test runs.
- **Product Management:**
 - Automate the addition and removal of products from the cart and wishlist.
 - Automate the admin functionality for adding and deleting products.
- **Order Placement:**
 - Automate the order placement process, including integration with the payment gateway.
- **End-to-End Workflows:**
 - Automate the complete workflow from product selection to order placement.
 - Automate the workflow for user registration and login.
- **Performance and Security Testing:**
 - Automate load testing to ensure the website can handle multiple users.
 - Automate security testing to verify that sensitive data is encrypted.

4. Defect Management

4.1 Critical Defects

- **DEF_003 (Critical): Payment did not occur during checkout.**
 - **Description:** When a user attempts to complete the payment during checkout, the payment does not occur.
 - **Impact:** This defect prevents users from completing their orders, directly impacting the core functionality of the application.
 - **Action:** Immediate investigation and resolution are required.

4.2 High-Priority Defects

- **DEF_001 (High): Admin failed to add a new product.**
 - **Description:** The system returns a "Failed to fetch" error when an admin attempts to add a new product.
 - **Impact:** Admins cannot add new products, affecting the product catalog.
 - **Action:** High priority should be given to resolving this issue.
 - **DEF_002 (High): Admin failed to delete a product.**
 - **Description:** The system returns a "Failed to fetch" error when an admin attempts to delete a product.
 - **Impact:** Admins cannot delete products, leading to potential issues with the product catalog.
 - **Action:** High priority should be given to resolving this issue.
-

5. Test Execution

- **Manual Testing:** Manual test cases will be executed first to identify any major issues. The results will be documented, and defects will be reported.
- **Automation Testing:** Once manual testing is complete, automation scripts will be executed to validate the functionality across multiple test runs.
- **Defect Retesting:** After defects are resolved, retesting will be performed to ensure that the issues have been fixed and no new issues have been introduced.

8. Future Enhancements

- Incorporate AI for personalized recommendations.
- Add advanced analytics for admin dashboards.
- Enable multilingual support for global accessibility.

9. Conclusion

This project successfully delivers a feature-rich e-commerce platform for clothing. The robust backend architecture and user-friendly interface ensure a seamless shopping experience. Future enhancements aim to make the platform even more adaptive and user-centric.