1. **INTRODUCTION**

**1. INTRODUCTION**

***1.1 PROBLEM STATEMENT***

In today’s competitive online marketplace, e-commerce platforms are often overloaded with an extensive range of product categories, which can make finding specific items challenging for consumers. Shoppers seeking specialized products, such as kurta sets, may face difficulties navigating generic, multi-purpose e-commerce websites that do not prioritize or customize their experience for niche products. This lack of focus can lead to a frustrating shopping journey, where users struggle to find what they are looking for efficiently, reducing customer satisfaction and loyalty.

***1.2 PROPOSED SYSTEM***

To address these challenges, the project “Ethnica” proposes the development of a bespoke e-commerce platform dedicated exclusively to the sale of kurta sets. Unlike generalized e-commerce sites, Ethnica aims to create a user-centric platform tailored specifically for users interested in this type of apparel. By focusing on a single product category, Ethnica ensures a more streamlined, intuitive, and enjoyable shopping experience.

***1.3 FEATURES OF THE PROPOSED SYSTEM***

The proposed system will offer the following key features:

1. **User Registration and Authentication**: Users can create accounts and log in securely.
2. **Product Search**: Users can search for specific kurta based on keywords .
3. **Add to Cart**: Users can add items to their shopping cart and modify the quantity of products.
4. **Checkout Process**: A simple and secure checkout process for users to finalize their purchases.
5. **View Order History**: Users can view their previous orders.
6. **Download Invoice**: Users can download an invoice for completed orders.
7. **Admin Dashboard**: Admin users can manage products, process orders, and update inventory.

**2. FUNCTIONAL REQUIREMENTS**

**2. FUNCTIONAL REQUIREMENTS**

### *****2.1 Administrative Functions*****

The admin will have full access to the system's backend to manage products and orders:

* **Manage Products**: Admins can add, update, or delete kurta sets.
* **Order Management**: Admins can view and manage orders placed by users.

### *****2.2 User Registration*****

* **Sign Up**: Users will provide their firstname, lastname, email address, username, and password to create an account.
* **Login**: Registered users can log in using their credentials.

### *****2.3 Product Search and Viewing*****

* **Search Functionality**: The system will provide a search bar where users can enter keywords (e.g., "kurta") to find relevant products.
* **Product Details**: The system will display product details such as name, description, images, price, and size.

### *****2.4 Shopping Cart and Checkout*****

* **Add to Cart**: Users can add products to their cart, modify the quantity, or remove items.
* **Checkout Process**: Users will enter their shipping details (name, address, contact number) and proceed to order.
* **Order Confirmation**: After completing the checkout, users will receive an order confirmation, and the order will be stored in their order history.

### *****2.5 Order History and Invoice*****

* **Order History**: Users can view their previous orders.
* **Download Invoice**: Users can download an invoice for each completed order.

**3. NON-FUNCTIONAL REQUIREMENTS**

#### 3. NON-FUNCTIONAL REQUIREMENTS

**PERFORMANCE**

The system can optimized to handle multiple user requests simultaneously to provide a smooth shopping experience for users seeking products. Pages must load within 2 to 3 seconds, even during peak shopping periods. Database queries should be efficient to minimize response times, particularly for operations like searching and filtering organic products.

**SCALABILITY**

The platform is designed to accommodate growth, including an increasing number of users, products, and orders as the business expands. The infrastructure should support the integration of future features, such as personalized product recommendations and advanced analytics.

**USABILITY**

The interface is intuitive, with a user-friendly design for both desktop. Error messages should be clear, actionable, and not expose sensitive system information.

**DEPENDABILITY AND ACCESSIBILITY**

Ethnica website is available 24/7 to ensure consistent access for both users and administrators. Regular automated backups should be scheduled to prevent data loss. The system is designed to recover smoothly and resume normal operations following any unexpected crashes or failures.

**MAINTAINABILITY**

The system follows modular and well-documented coding practices to simplify updates and debugging. Logs should be generated for significant events to aid troubleshooting and auditing.

**COMPATIBILITY**

The website is compatible with modern browsers (e.g., Chrome, Firefox, Safari, Edge) and support both desktop.

**SUPPORTABLITY**

The code and supporting modules of the system are well documented and easy to understand. Online documentation and help system requirements.

**4. UML DIAGRAMS**

**4.1 USE CASES**

Ethnica Website Use Cases :

***4.1.1 Login***  
  
**Use Case ID:** E\_01  
**Use Case Name:** Login  
**Date Created:** 2024-11-16  
**Description:** This use case allows users and admin to log in to the ethnica website to access their account.  
**Primary Actor:** User  
**Secondary Actor:** None  
**Precondition:** User has a valid account.  
**Postcondition:** The system redirects the user to the home page.  
**Main Flow:**  
User enters their username and password.  
User clicks the "Login" button.  
The system validates the user's username and password.  
If the credentials are valid, the system redirects the user to the home page.  
If the credentials are invalid, user still in the login page.  
**Alternative Flows:**  
Invalid username or password: User still in the login page.

***4.1.2 Register***   
**Use Case ID:** E\_02  
**Use Case Name**: Register  
**Date Created:** 2024-11-16  
**Description:** This use case allows users to create a new account on the Ethnica website.  
**Primary Actor:** User  
**Secondary Actor:** None  
**Precondition:** User does not have an account on the website.  
**Postcondition:** User has a new account on the website.  
**Main Flow:**  
User clicks the icon link on the homepage.

User can click “Register” link to register   
User enters their personal information, including firstname, lastname, email address, username  and password.  
User clicks the "Register" button.  
The system verifies the user's information and creates a new account.  
User redirected to Login page .  
**Alternative Flows:**  
Invalid information: The system displays an message informing the user that their information is invalid.  
Password: if the password is similar to username system display a message ”The password is too similar to the username”. Also password must contain at least 8 characters.

***4.1.3 Search Products***

**Use case id:** E\_03

**Use case name:** Search Products.

**Date Created:** 2024-11-16

**Descriptions:** this use case allow user to search products

**Primary Actor:** User

**Secondary Actor :** None

**Preconditions:** user is on the website. Products are available in the website.

**Postconditions:** system display relevant product search result based on the user query’s.

**Mainflow :** User enters a product name in the search bar.

User clicks the search button.

The system processes the query and searches the inventory for matching products.

The system displays the search results with product name image and price.

User can further filter the result based on the criteria like price etc.

**Alternative flow :** if no products match the query the system display

item is found”.

***4.1.4 Add to Cart***

**Use case id:** E\_04

**Use case name :** Add to Cart

**Date Created:** 2024-11-16

**Descriptions:** this use case allow user to cart the products

**Primary actor :** User

**Secondary actor :** None

**Precondition:** User is logged in the website. The selected churidar set is available in stock.

**Precondition:** the item is added to the user cart.

**Mainflow:** user navigates to the “shop” where the product are displayed . .

User can select the product , can view the description also can add the product to My Cart.

User clicks the “Add to Cart” button.

The system verifies product availability .

After the verification the product is added to the cart.

In the Cart page user can add the quantity of the product  
User can choose to continue shopping or “checkout”.

Alternative flow: if the product is out of stock the user cannot add product to cart.

***4.1.5 Checkout***

**Use case id:** E\_05

**Use case name:** Checkout

**Description:** This use case allow user to place an order.

**Primary Actor :** User

**Secondary Actor:** None

**Precondition:** User had added products to their shopping cart. User is logged in. Product in the cart are in stock.

**Postcondition:** The order is successfully placed. The order details are saved in the user order history.

**Mainflow:** User can view the items in their shopping cart.

User can add the quantity of the item.

User clicks the "Checkout" button.

If the user can add name, shipping information including address, contact number and pincode, .

User clicks the "Place Order" button to finalize the purchase.

The system generates the order and stores it in the order management system.

The system displays an order confirmation message and sends a confirmation email to the user.

***4.1.6 Manage Product***

**Use case id:** E\_06

**Use case name :** Manage Products

**Date Created :** 2024-11-16

**Description:** This use case allow admin to manage the products in the website

**Primary Actor :** Admin

**Secondary Actor:** None  
**Precondition:** admin has login access to the e-commerce admin panel. Manage products that has available in the website.

**Postconditions:** Products are added or stock levels are modified.

**Mainflow:** Admin logs into the e-commerce admin panel.

Admin navigates to the product management section.

Admin selects “Product” to upload a new churidar set, or “Edit Product” to modify

an existing item.

Admin enters product details such as name, price, sizes, images, and stock quantity.

Admin saves the changes or additions.

System updates the inventory and displays the churidar set on the e-commerce website.

**Alternative Flow:** If the admin enters invalid data (e.g., missing product name, price), the

system prompts for corrections.

The system alerts the admin when stock levels for certain churidar sets are running low

**Other possible use cases include:**

User can view order history.

User can edit profile.  
Manage user .

Manage order.

***4.2 USECASE DIAGRAM***

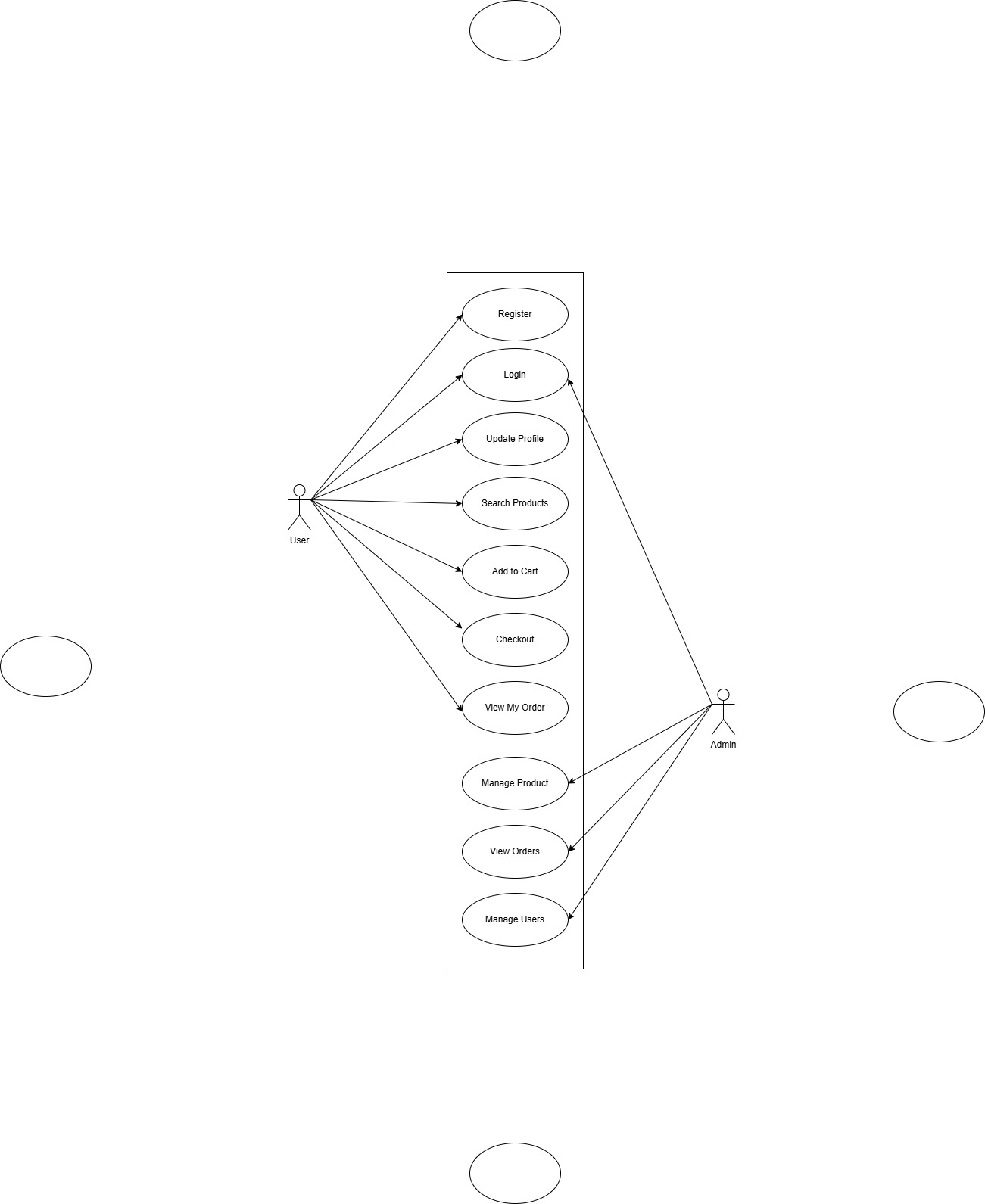


Fig . 4.1 Use Case Diagram

***4.3 ACTIVITY DIAGRAM***

**4.3.1 USER**

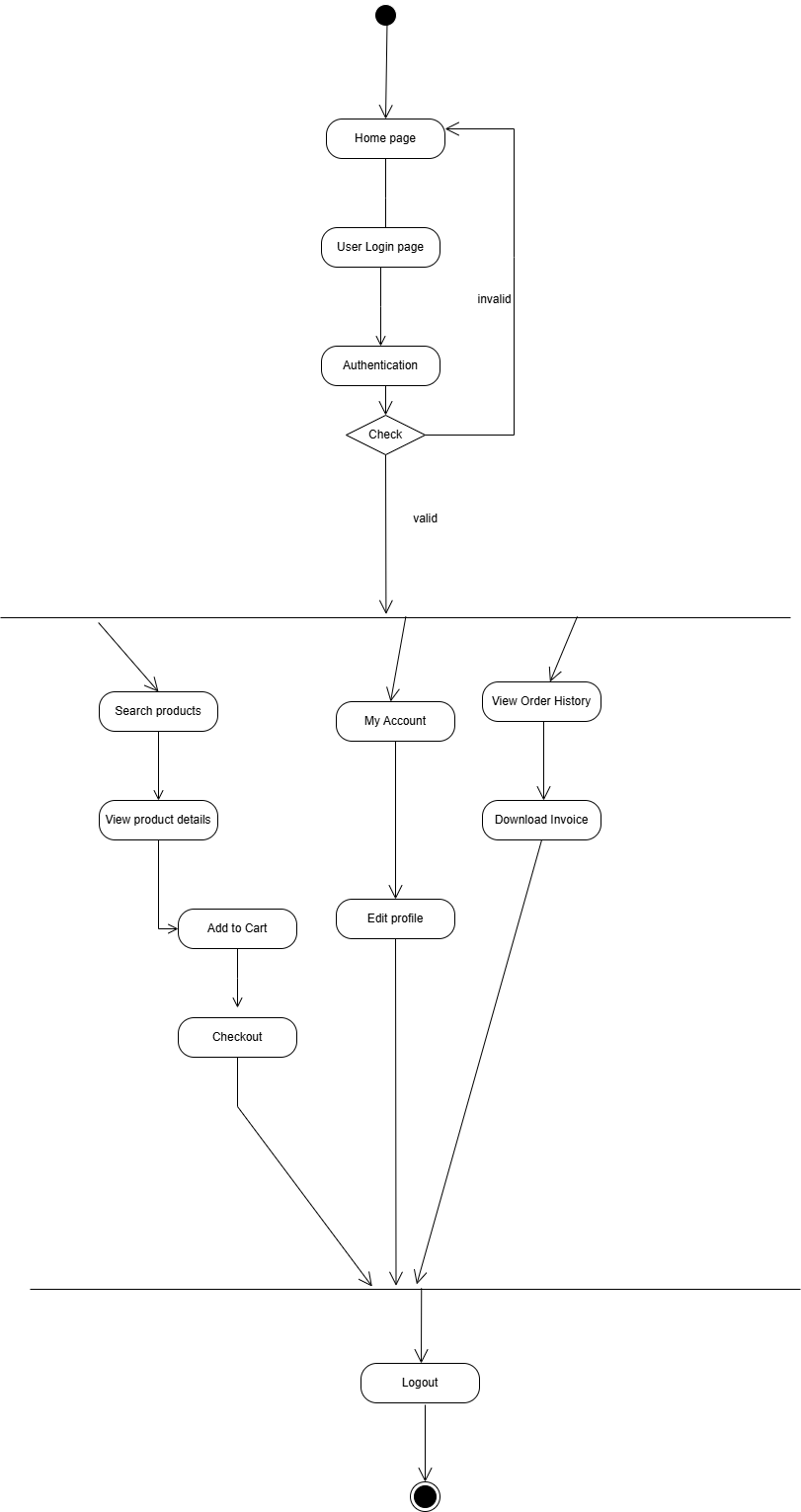
******

Fig . 4.2 Activity Diagram - User

**4.3.2 ADMIN**

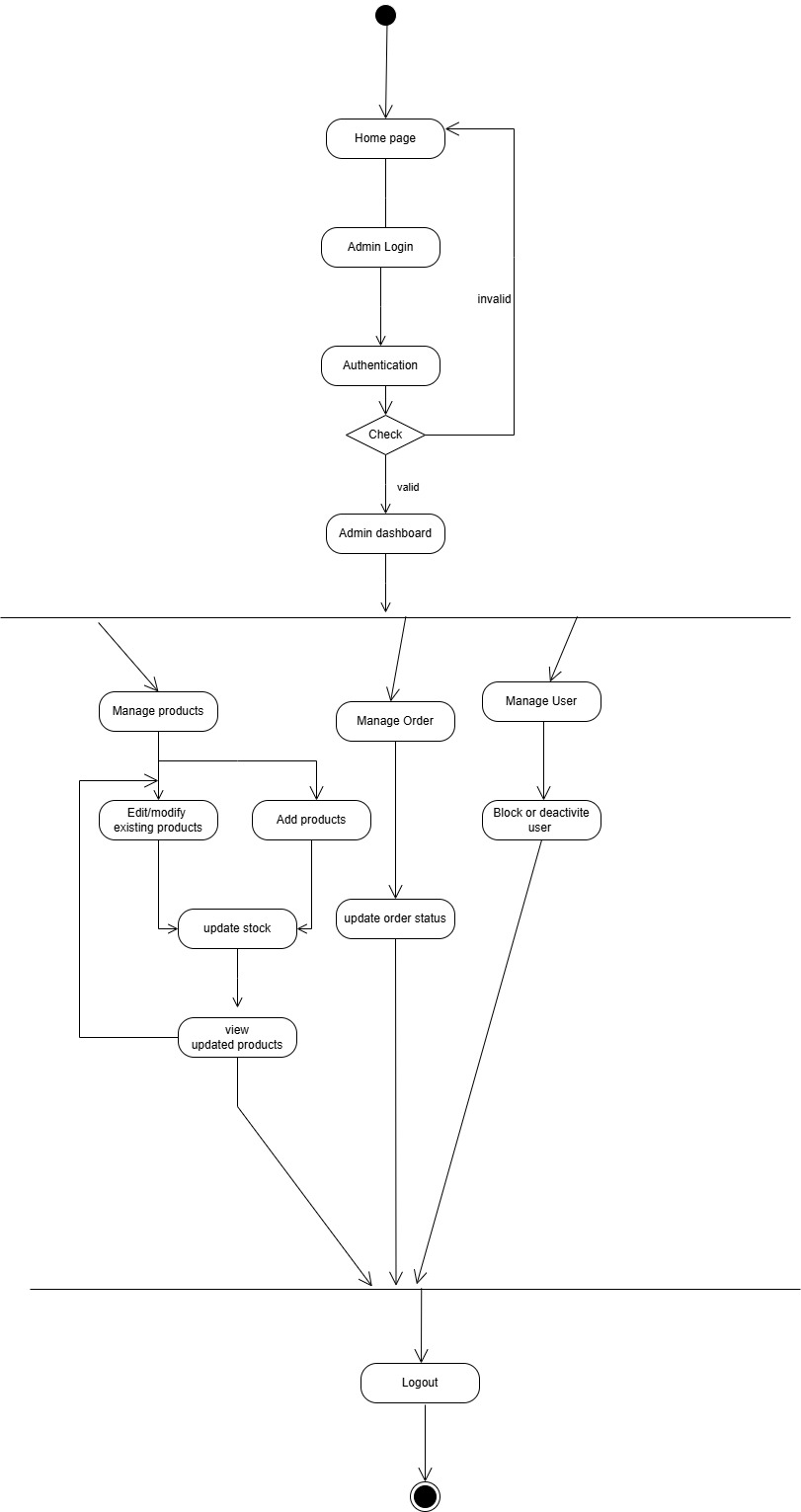
****

Fig . 4.3 Activity Diagram – Admin

* 1. ***CLASS DIAGRAM***

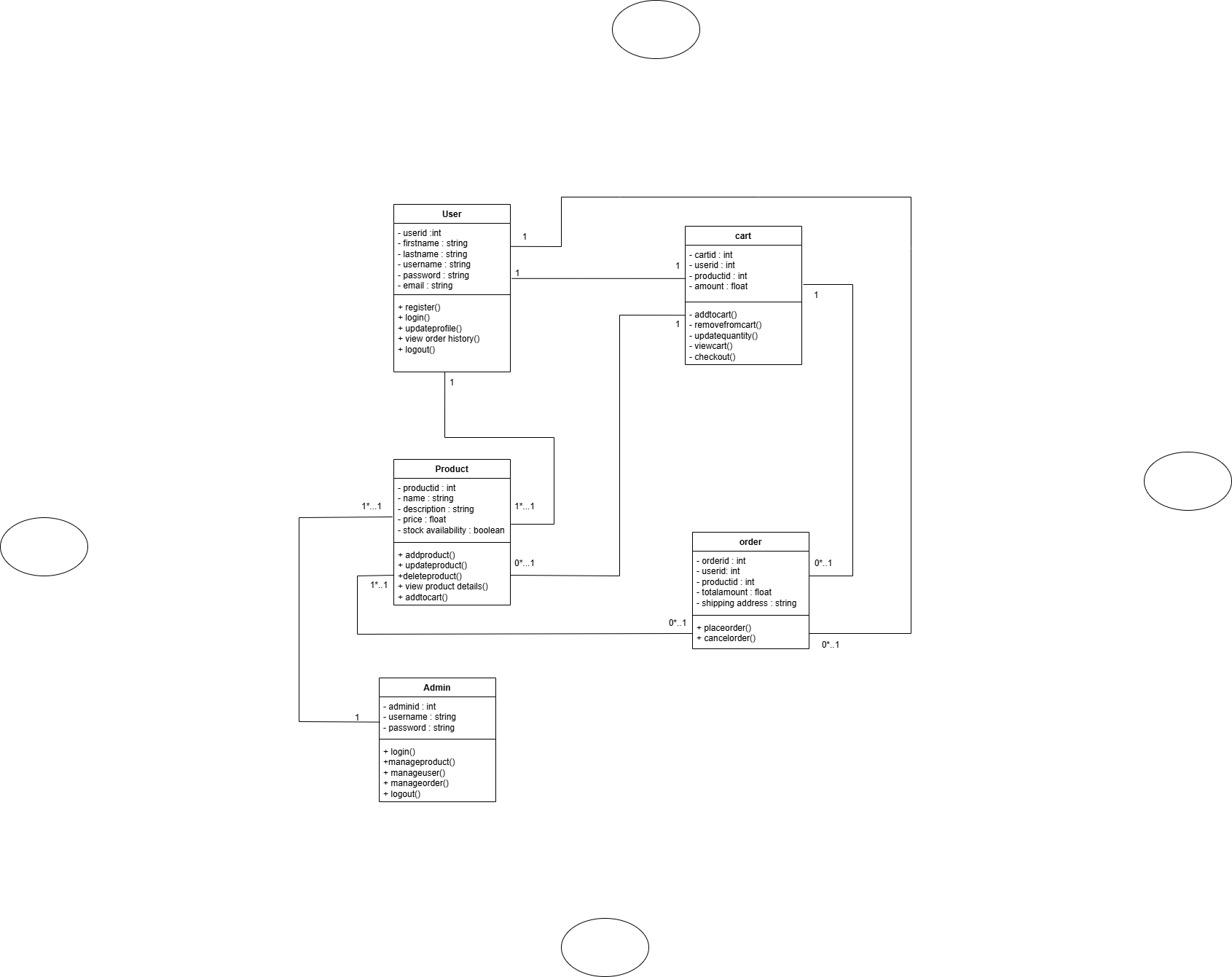


Fig . 4.4 Class Diagram

**5. TEST CASES**

**5. TEST CASES**

***5.1 LOGIN***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SR\_NO** | **TEST CASE** | **FEATURE** | **DESCRIPTION** | **STEPS TO EXECUTE** | **EXPECTED RESULTS** |
| **1** | TC-01 | User Interface | Check the appearance of the home page for the Ethnica website. | 1.Load the home page of the Ethnica website. | UI elements is properly aligned. Navigation buttons and links should be functional. |
| **2** | TC-02 | Login form | Verify user login with valid credentials | 1.Load the home page of the Ethnica website 2.Enter username and password. | User is redirected to the home page |
| **3** | TC-03 | Login form | Verify login with invalid credentials | 1.Load the home page of the Ethnica website 2..Enter username and password incorrect. | User will still in login page. |
| **4** | TC-04 | Password Recovery | Verify the forgot password functionality works | 1.Load the home page of the Ethnica website 2.Click on the forgot password link and enter username | A password reset kink is sent to the email. |

Table 5.1 Test Case For Login

***5.2 REGISTRATION***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SR\_NO** | **TEST CASE** | **FEATURE** | **DESCRIPTION** | **STEPS TO EXECUTE** | **EXPECTED RESULTS** |
| 1 | TC-01 | User Interface | Check the appearance of the registration page. | 1.Load the Ethnica website registration page. | UI elements is properly aligned and formatted.  Form fields should be clearly labeled and easy to understand.  Navigation buttons and links should be functional. |
| **2** | TC-02 | Registration Form | Check the behavior of the registration form with empty fields. | 1.Open the registration page.  2.Leave all form fields blank.  3.Click on the "Register" button. | Error messages is displayed for all empty fields. The error messages is informative and clear. |
| **3** | TC-03 | Registration Form | Check if invalid data is prevented during registration. | 1.Enter invalid data in the form fields, such as: Incorrect email format.Weak password with less than 8 characters.  2.Mismatched password confirmation. 3.Click on the "Register" button. | Error messages is displayed for invalid data.  The error messages is specific to the type of invalid data entered.  Registration should not be allowed. |
| **4** | TC-04 | Registration Form | Check successful registration with valid data. | 1.Enter valid data in all form fields, including: First name: ,last name :,Username :Email address. :, Password:, Password confirmation. 2.Click on the "Register" button. | Registration is successful.  Directed to login. |
| **5** | TC-05 | Login Functionality | Verify login with valid credentials. | 1.Enter the username in the username field.  2.Enter the registered password in the password field.  3.Click on the "Login" button. | Login is successful.  The user is redirected to the home page. |

Table 5.2 Test Case For Registration

**5.3 PRODUCT ADD PAGE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SR\_NO** | **TEST CASE** | **FEATURE** | **DESCRIPTION** | **STEPS TO EXECUTE** | **EXPECTED RESULTS** |
| **1** | TC-01 | Admin dashboard | Verify that admin can add a new product | 1.Log in as admin.  2.Navigate to Products page.  3.Enter the product name, description,  upload image, set price and stock quantity  4.Click Save button | Product is successfully added and visible in the product list. |
| **2** | TC-02 | Admin Dashboard | Verify all fields are required | 1.Log in as admin.  2.Navigate to 2.Products page.  4.Leave one or more fields as blank  5.Click on Save button | Error message is displayed for required fields |
| **3** | TC-03 | Admin Dashboard | Verify admin can update product details | 1.Log in as admin.  2.Navigate to Products page.  3.Select a product and click to edit.  4.Modify details and click on Save button | Product details is updated successfully. |
| **4** | TC-04 | Admin Dashboard | Verify that admin can remove a product | 1.Log in as admin.  2.Navigate to Products page.  3.Select a product and click to delete the product and confirm the deletion | Product is removed from the product list. |
| **5** | TC-05 | Admin Dashboard | Verify image upload functionality | 1.Log in as admin.  2.Navigate to Products page.  3.Upload an image for product and click to save. | Image should be uploaded and displayed on the product details page. |
| **6** | TC-06 | Admin Dashboard | Verify stock quantity update | 1.Log in as admin.  2.Navigate to Products page.  3.Select a product and update stock quantity  4Click to save the changes  . | Stock quantity is updated and reflected on the product page. |

Table 5.3 Test Case For Product Add Page

**6. INPUT DESIGN AND**

**OUTPUT DESIGN**

**6. INPUT DESIGN AND OUTPUT DESIGN**

***6.1 INPUT DESIGN***

**6.1.1 Introduction to Input Design**

The input design in Ethnica aims to facilitate user interaction by collecting accurate data efficiently and securely. This section covers the various forms and input fields on the homepage and across different user roles (Guest, Registered User, Admin), ensuring seamless data collection, validation, and handling.

**6.1.2 Forms and Input Fields**

Home Page Forms

* Search Form

Fields: Search Bar (Text Field): A search bar that allows users to look for products by name or keyword.

* Registration Form (For New Users)
* Fields:
  + - Username (Text Field): A unique username for identification.
    - Email (Email Field): For communication.
      * Validation: Valid email format required.
    - Password (Password Field): Password for account security.
      * Validation: Minimum 8 characters, with at least one uppercase letter, one digit, and one special character.
    - Confirm Password (Password Field): Confirms the password entered above.
      * Validation: Must match the "Password" field.
* Login Form
* Fields:
  + - Username (username Field): The user's registered username.
    - Password (Password Field): The password associated with the account.
      * Validation: Required and must match the registered password.
* Checkout Form (For Registered Users)
* Fields:
  + - Name (Text Field): The user's name for shipping.
      * Validation: Required, at least two words (first and last name).
    - Address (Text Field): Full delivery address.
      * Validation: Required, formatted correctly.
* Admin Forms
* Add Product Form
  + - Fields:
      * Product Name (Text Field): The name of the product.
        + Validation: Required, at least 3 characters.
      * Product Price (Number Field): The price of the product.
        + Validation: Must be a positive number.

**6.2.3 Validation and Error Handling**

* Client-Side Validation:
* HTML5 attributes such as required, maxlength, and pattern are used for real-time validation. For instance:
  + - Email field: type="email" for correct email format.
    - Password field: pattern="[A-Za-z0-9]{8,}" to ensure password strength.
    - Confirm Password: A JavaScript check to match the passwords.
* Server-Side Validation:
* Django form and model validation ensure correct data is saved:
  + - Email Validation: Django’s EmailField for proper email format.
    - Password Validation: Custom validation ensures the password meets security requirements.
    - Required Fields: Django’s required=True ensures no field is left empty.
* Error Messages:
* When validation fails, error messages are displayed next to the affected fields, such as "Password must be at least 8 characters," or "Email is required."
* Error messages are clear and help users fix the issue promptly.

***6.2 OUTPUT DESIGN***

**6.2.1. Introduction to Output Design**

The output design ensures users and admins receive information in a clear and structured format. The goal is to deliver product details, and order summaries in an easily understandable way. This section focuses on how the information is displayed for different user roles, such as customers and admins.

**6.2.2. User Interface Elements**

* Home Page
* The home page displays key sections such as Products, About Us, Contact, Login, Register, and a Search Bar for product search.
  + - Product Listings: Displays products with images, prices, names, and an "Add to Cart" button.
    - Quick Links: Easy access to "Register" and "Login" for new or returning users.
* Registered User Dashboard
* Once logged in, the Dashboard provides an overview of the user’s account with options for:
  + - Products: A list of products the user has purchased or added to their cart.
    - Orders: A history of past orders with order status (e.g., shipped, delivered).
    - Add to Cart: Items added to the shopping cart for checkout.
    - Profile/Account: Access to user information and settings.
* Admin Dashboard
* The Admin Dashboard provides access to manage:
  + - Products: Admins can view, add, edit, or delete products.
    - Users: Admins can view all registered users and their activity.
    - Orders: Admins can view, process, and update order statuses.
* Order Details and Invoice
* After a purchase, users are shown an Order Summary with itemized costs, taxes, and total. This includes a downloadable invoice.
  + - Key Elements: Product names, quantities, unit prices, subtotal, shipping fees, taxes, total cost.

**6.2.3. Error and Success Messages**

* Success Messages:
* After a successful registration or login, users see messages like "Welcome, [Username]!" .
* After placing an order, users receive a message such as, "Your order has been successfully placed! You will receive an email confirmation shortly."
* Error Messages:
* In cases of invalid form submission, error messages like "Please enter a valid email address" or "Password does not meet requirements" are displayed.

**7. SYSTEM IMPLEMENTATION**

# 7. System Implementation

***7.1 Introduction***

The Ethnica project is an innovative e-commerce platform designed for selling kurta items. The application was developed using Django as the backend framework, with HTML, CSS, and Bootstrap for the frontend. JavaScript was used for interactive elements to enhance user experience. The development environment includes Python 3.9, Django 4.x, and additional dependencies such as SQLite for the database, Pillow for image handling, and Django Crispy Forms for enhanced form styling.

***7.2 Project Structure***

The Ethnica project follows Django’s standard project structure, with a clean separation between the frontend, backend, and static resources. The project structure is as follows:

Ethnica/

│

├── kurta/ # Main project folder

│ ├── \_\_init\_\_.py

│ ├── settings.py # Django settings configuration

│ ├── urls.py # Project-level URL routing

│ ├── wsgi.py

│ └── asgi.py

│

├── products/ # Product-related app

│ ├── \_\_init\_\_.py

│ ├── models.py # Django models for products, cart,checkout etc.

│ ├── views.py # Views handling product-related requests

│ ├── urls.py # App-level URL routing

│ ├── forms.py # Forms for product management

│ ├── templates/ # Templates for product-related pages

│ └── static/ # Static files (images)

│

├── myapp/ # User authentication app

│ ├── \_\_init\_\_.py

│ ├── models.py # Custom user models (if any)

│ ├── views.py # Views for registration, login, etc.

│ ├── urls.py # App-level URL routing

│ └── templates/ # Templates for authentication

│

├── static/ # Static files for the entire project

├── templates/ # Project-wide HTML templates

│ └── base.html # Base template for inheritance

└── manage.py # Django management script

Each app in the project follows Django's convention of having its own models.py, views.py, urls.py, forms.py, and templates for better modularity and scalability.

***7.3. Database Design and Models***

The database is designed using Django’s with SQLite as the backend. Key models include:

* **Product**: Fields for name, price, image, description and stock.
* **CartProduct**: Fields for product rate ,quantity, size , subtotal. .
* **Checkout**: Fields for address, mobile, total, customer\_id, cart\_id are foregin key .

Relationships:

* One-to-Many: A user can place many orders.
* Many-to-Many: A user can have many products in the cart, with a many-to-many relationship between the Cart and Product.

**7.3.1 Table name : User Registration**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| Customer\_id | INTEGER | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for user |
| username | VARCHAR(50) | NOT NULL | User username |
| Email address | VARCHAR(50) | NOT NULL | User email |
| First name | VARCHAR(50) | NOT NULL | First name of user |
| Last name | VARCHAR(50) | NOT NULL | Last name of user |
| password | VARCHAR(50) | UNIQUE,NOT NULL | Password for user |

Table 7.1 User registration table database design

**7.3.2 Table name : Product**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| Product\_id | INTEGER | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for product |
| material | VARCHAR | NOT NULL | Material of the item |
| color | VARCHAR | NULLABLE | Color of the item |
| name | VARCHAR(200) | NULLABLE | Name of the product |
| image | VARCHAR(255) | NOT NULL | product image |
| stock | INTEGER | DEFAULT 0 | Quantity of product in stock |
| description | TEXT | NULLABLE | Product description |
| price | DECIMAL(10,2) | NULLABLE | Price of the product |

Table 7.2 Product table database design

***7.3.3 Table name : Cart***

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| Cart\_id | INTEGER | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for the cart |
| customer\_id | INTEGER | FOREIGN KEY, NULLABLE | Reference to `User` table |
| total | INTEGER | DEFAULT 0 | Total price of items in the cart |

Table 7.3 Cart table database design

**7.3.4 Table name : Cartproduct**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| id | INTEGER | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for the cart product |
| cart\_id | INTEGER | FOREIGN KEY, NOT NULL | Reference to `Cart` table |
| product\_id | INTEGER | FOREIGN KEY, NOT NULL | Reference to `Product` table |
| rate | INTEGER | NOT NULL | Price per unit of the product |
| quantity | INTEGER | NOT NULL | Quantity of the product |
| subtotal | INTEGER | NOT NULL | Total price for the cart product |
| size | VARCHAR(20) | NOT NULL | Size of the product |

Table 7.4 Cartproduct table database design

**7.3.5 Table name : Order**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Constraints | Description |
| order\_id | INTEGER | PRIMARY KEY, AUTO\_INCREMENT | Unique identifier for the order |
| cart\_id | INTEGER | FOREIGN KEY, NOT NULL | Reference to `Cart` table |
| customer\_id | INTEGER | FOREIGN KEY, NULLABLE | Reference to `User` table |
| total | INTEGER | NOT NULL | Total price of the order |
| order\_status | VARCHAR(50) | CHECK IN ('order recived', 'order processing', 'order on the way', 'order completed', 'order cancelled'), NOT NULL | Status of the order |
| address | VARCHAR(100) | NOT NULL | Shipping address |
| mobile | VARCHAR(50) | NOT NULL | Customer's mobile number |

Table 7.5 Order table database design

***7.4 URL Routing and Views***

* URL routing is organized into app-specific urls.py files, imported into the root urls.py.
* **Example URLs**:

products/ for product listings.

/cart/ for cart management.

/checkout/ for placing orders.

Admin URLs are handled through Django's built-in admin system (/admin/).

Views:

* Product Views: Handle displaying products.
* User Views: Handle login, registration, and user profile.
* Order Views: Manage user orders and checkout.
* Admin Views: Allow admin to add/edit/delete products, manage users, and view orders.

HTTP Methods:

* GET: Used for retrieving information, such as loading a product page or user profile.
* POST: Used for submitting data, such as creating an account or placing an order.

***7.5 Templates and Frontend Integration***

* Templates use Django’s template engine with features like loops and conditionals.
* **Template Inheritance**: Base layout (base.html) includes headers, footers, and a responsive navbar.
* **Static Files**: Used for load images for the system.

***7.6 User Authentication and Authorization***

* Django’s built-in authentication handles login, logout, and registration.
* **Custom Roles**: The admin manages products and orders, while regular users can browse and shop.
* Role-based access restricts admin-only actions like inventory updates.

***7.7 Forms and Validation***

* Django forms are used for user inputs, such as login and product search.
* **Validation**: Ensures required fields, proper email formats, and secure passwords. Custom error messages guide users in correcting inputs.

***7.8 Business Logic and Core Functionality***

* **Product Display**: Products are dynamically fetched from the database and displayed by category.
* **Search** : Users can search products by name.
* **Cart Management**: Users can add products to their cart, adjust quantities, and remove items.
* **Checkout**: Collects user shipping details and processes orders, storing them in the database for generating invoice.
* Custom utility functions handle price calculations and stock updates.

***7.9 Testing and Debugging***

* The testing process includes unit tests for models and views, ensuring accurate data handling.
* Django’s testing framework was used to validate form submissions, URL resolutions, and database operations.
* Debugging tools include Django’s runserver debug mode and browser developer tools for frontend troubleshooting.

**8. FUTURE ENHANCEMENTS**

**8. FUTURE ENHANCEMENT**

***1. Improved User Interface (UI/UX):***

* Optimize the website for mobile devices to enhance the shopping experience.
* Add personalized product recommendations based on user behavior.

***2. Payment Integration***:

* Integrate additional payment gateways like PayPal, UPI, and digital wallets.

***3. Product Management:***

* Implement product reviews and ratings for better user engagement.
* Add inventory management to track stock levels and notify admins about low inventory.

***4. User Engagement Features:***

* Introduce a wishlist feature for users to save products for future purchases.
* Implement a referral and loyalty program to boost customer acquisition and retention.

***5. Admin Panel Enhancements:***

* Expand reporting and analytics to track sales and user behavior.
* Allow bulk product uploads to simplify inventory management.

***6. Security Improvements:***

* Implement two-factor authentication (2FA) for user accounts and enhance data encryption.

***7. Performance Optimization:***

* Optimize the website for faster load times using caching and CDN integration.

**9. CONCLUSION**

**9. CONCLUSION**

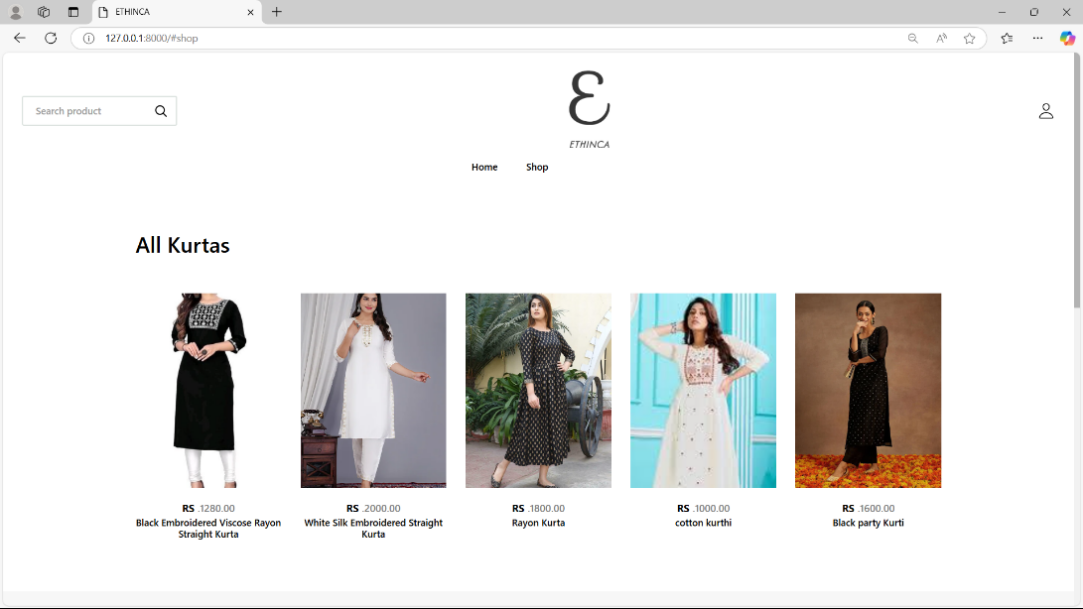
The Ethnica e-commerce platform is designed to offer a specialized, streamlined shopping experience for customers looking for kurta sets. By focusing on user-centric design and robust functionality, the platform simplifies online shopping for this niche. The website is easy to use and navigate, and it offers a number of features that make it a valuable resource for users.

In addition to the features that are currently available, the website is also planning to add a number of new features in the future, such as online payment. Future enhancements aim to further enrich the user experience through advanced technologies such as AI and mobile integration. The input and output design ensures an intuitive and efficient interaction for both users and administrators, while comprehensive security measures safeguard user data and transactions. These new features will make the website even more useful and convenient for both users and administrators. Overall, the online ethnica website is a valuable tool that can help users save time and improve their experience.

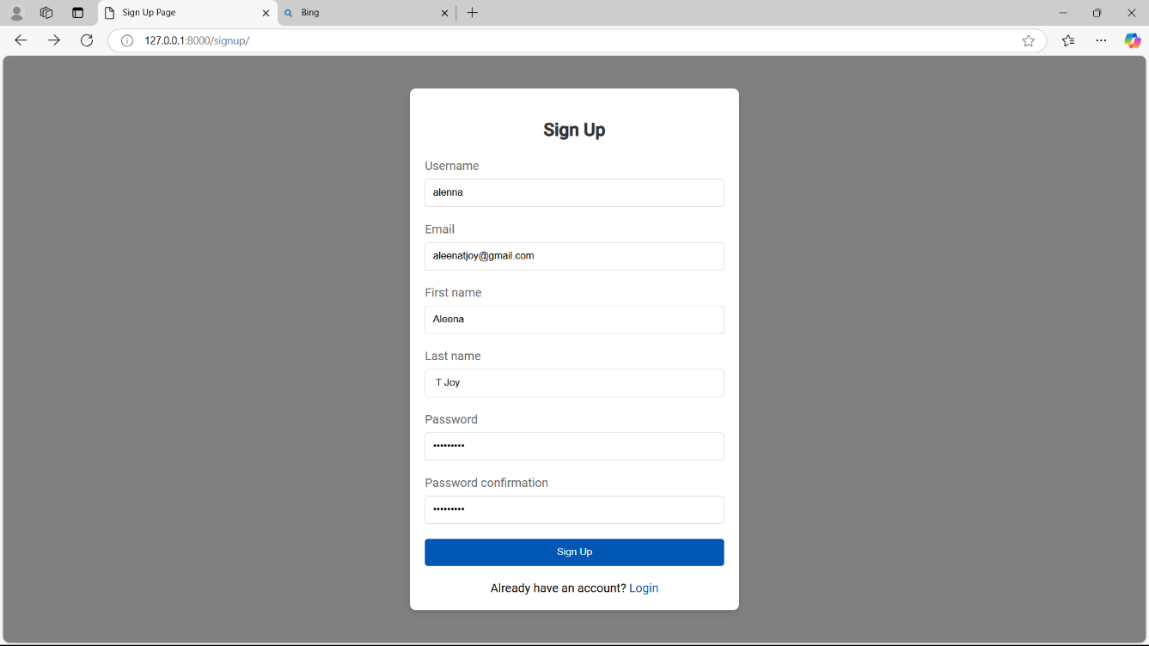
1. **ANNEXURE**

**10. SCREENSHOTS**

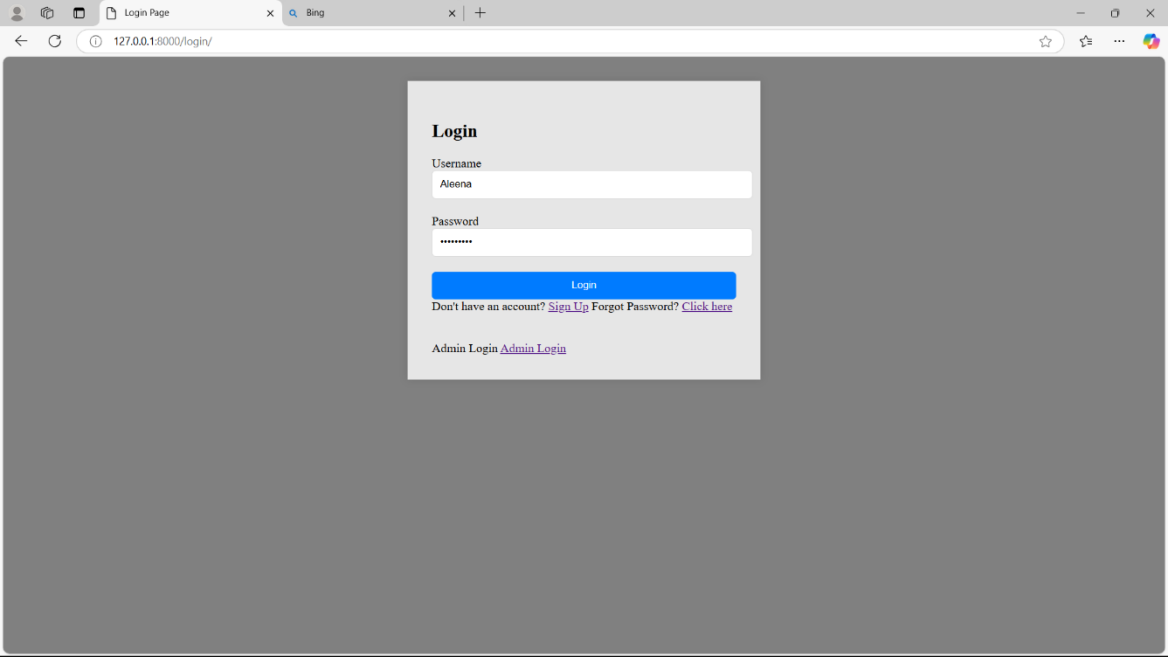
***10.1 Home page***

******

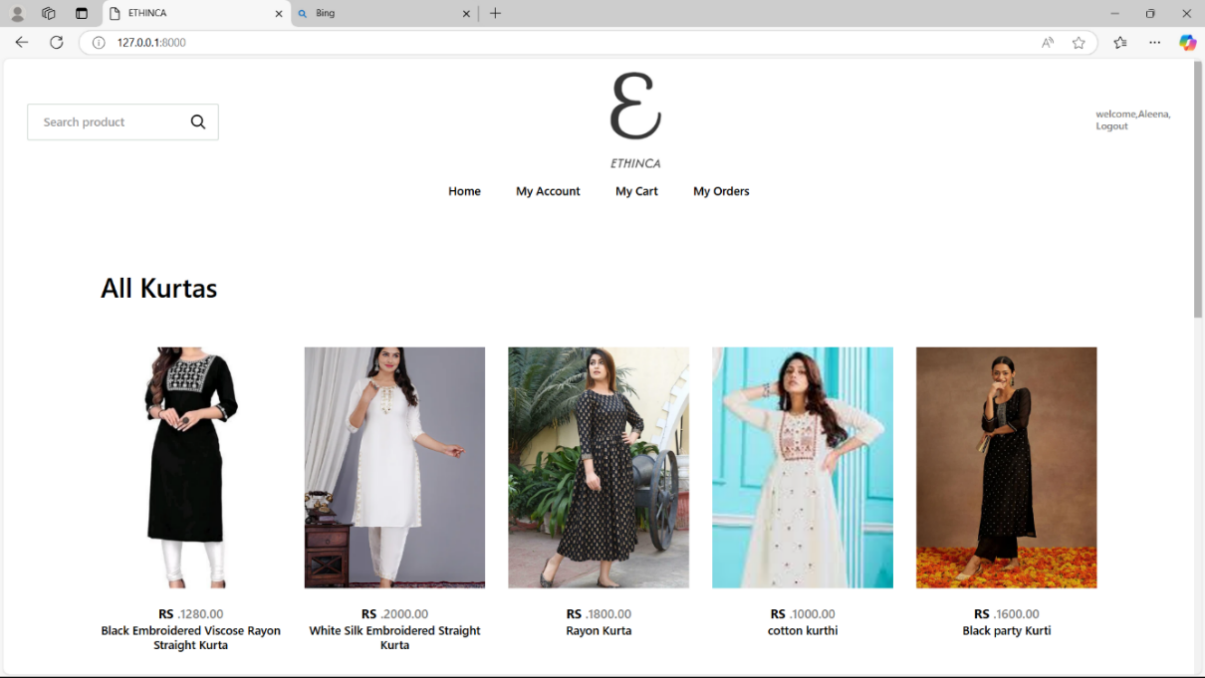
***10.2 Register page***

****

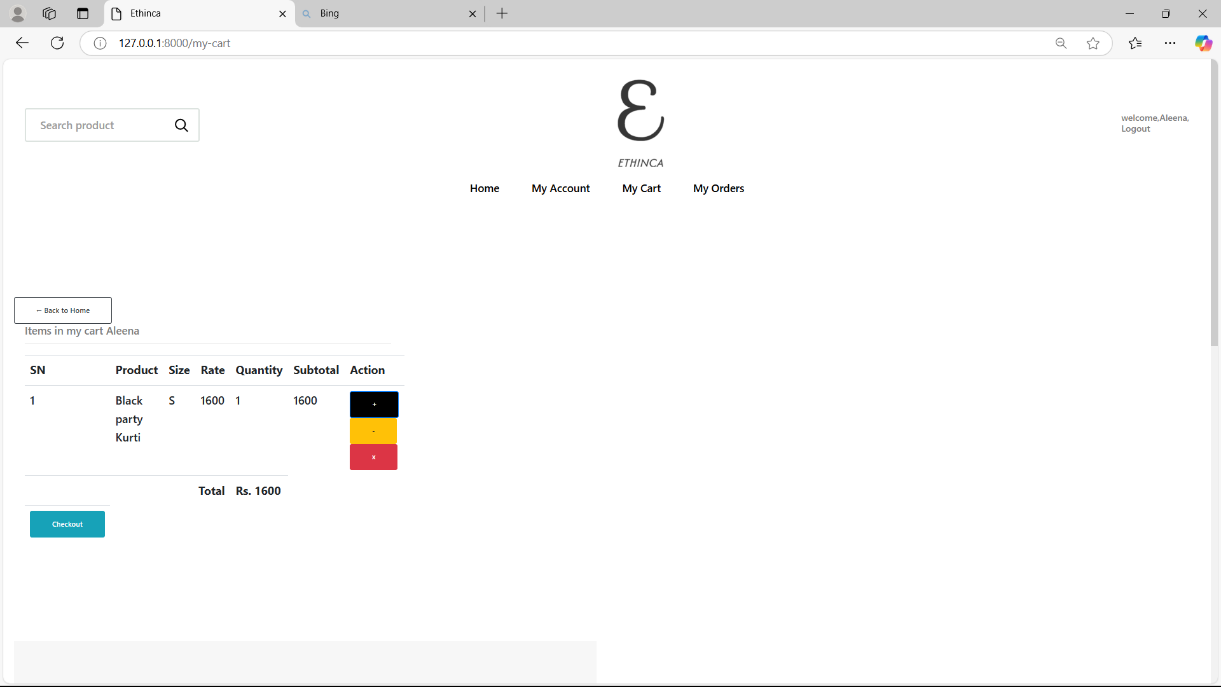
***10.3 Login page***

****

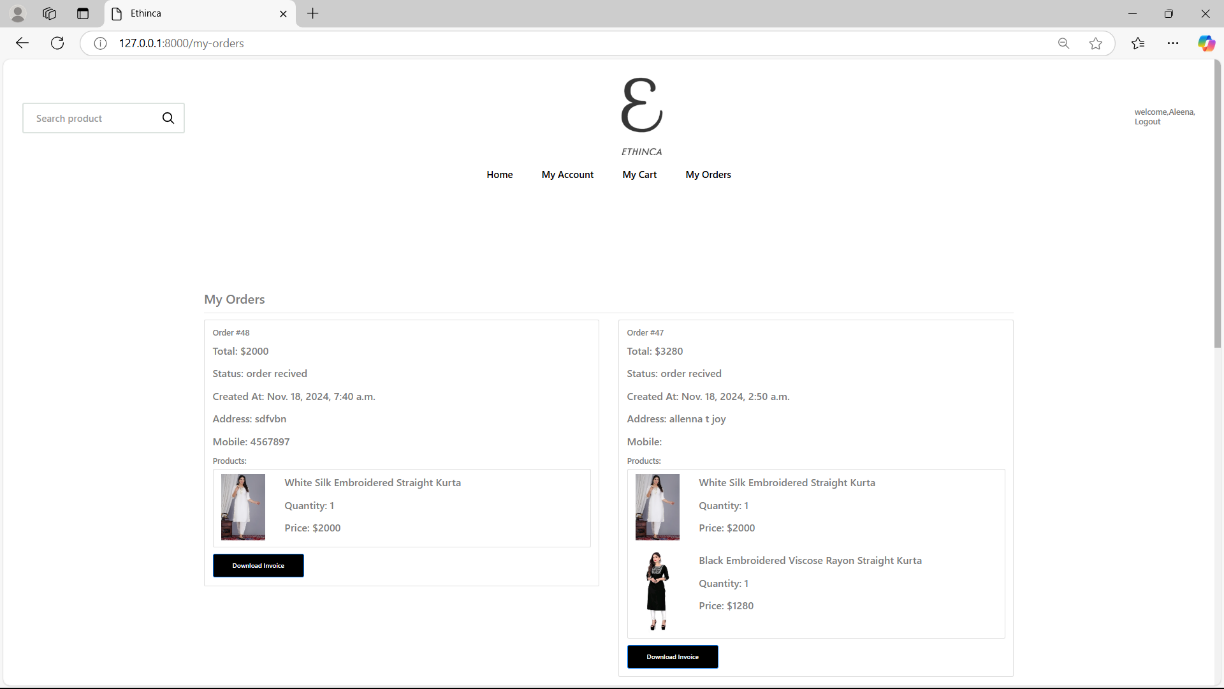
***10.4 User\_page***

****

***10.5 Cart \_page***

******

***10.6 Order History page***

******