**Project Phase 3**

Deliverable 5

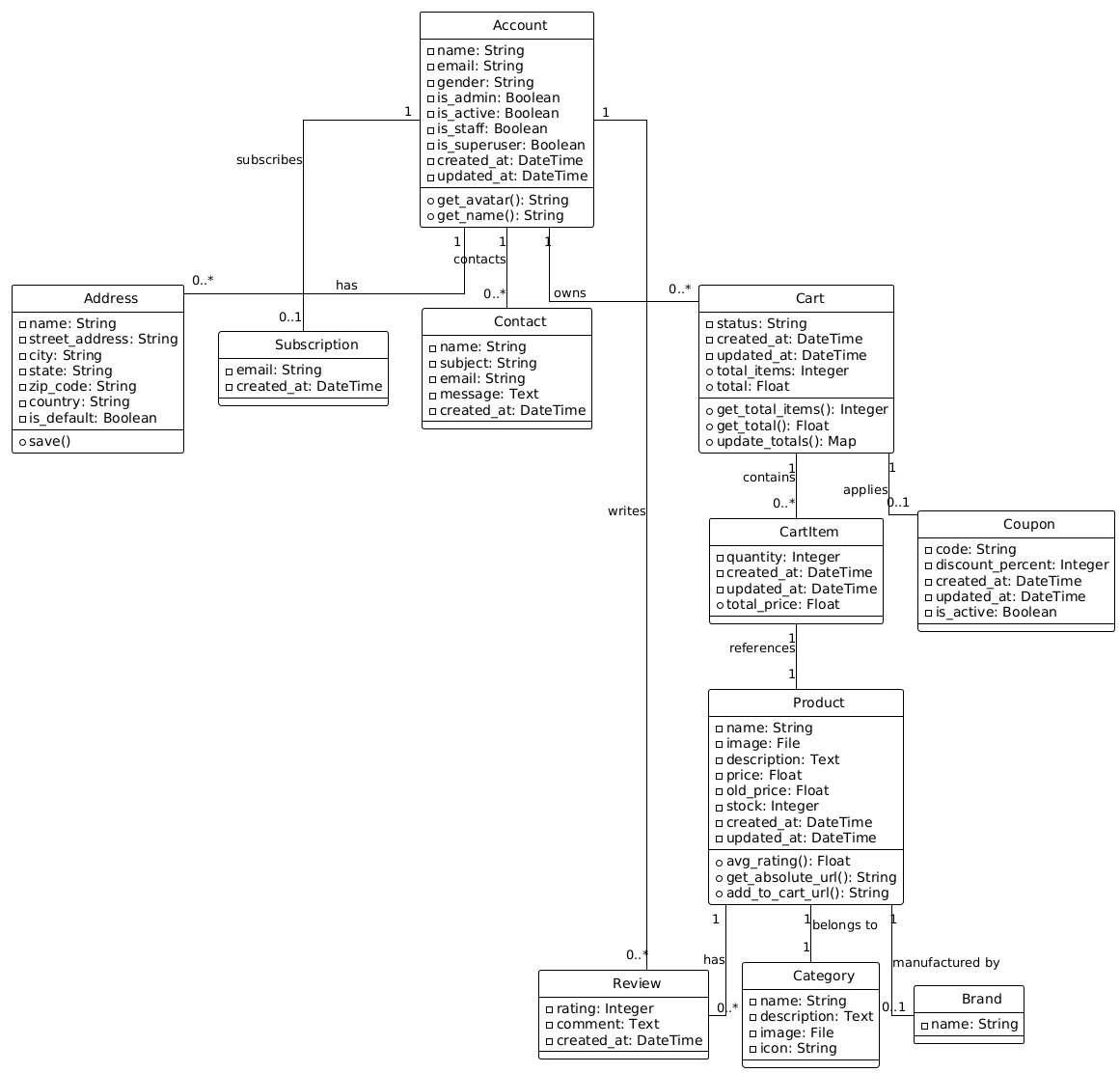
**Phase 3 Requirements**

The requirements for the development in **Phase 3: Secondary Feature Development** are derived from the initial project scope outlined in Phase 1 (as specified in Deliverable 2), with enhanced and detailed descriptions to ensure clarity and alignment with the objectives of this phase. The scope remains unchanged, and the requirements for this phase focus on the implementation of secondary features, addressing bugs in the core functionalities, and finalizing key components for the system's operational readiness.

**Requirements for Phase 3: Secondary Feature Development**

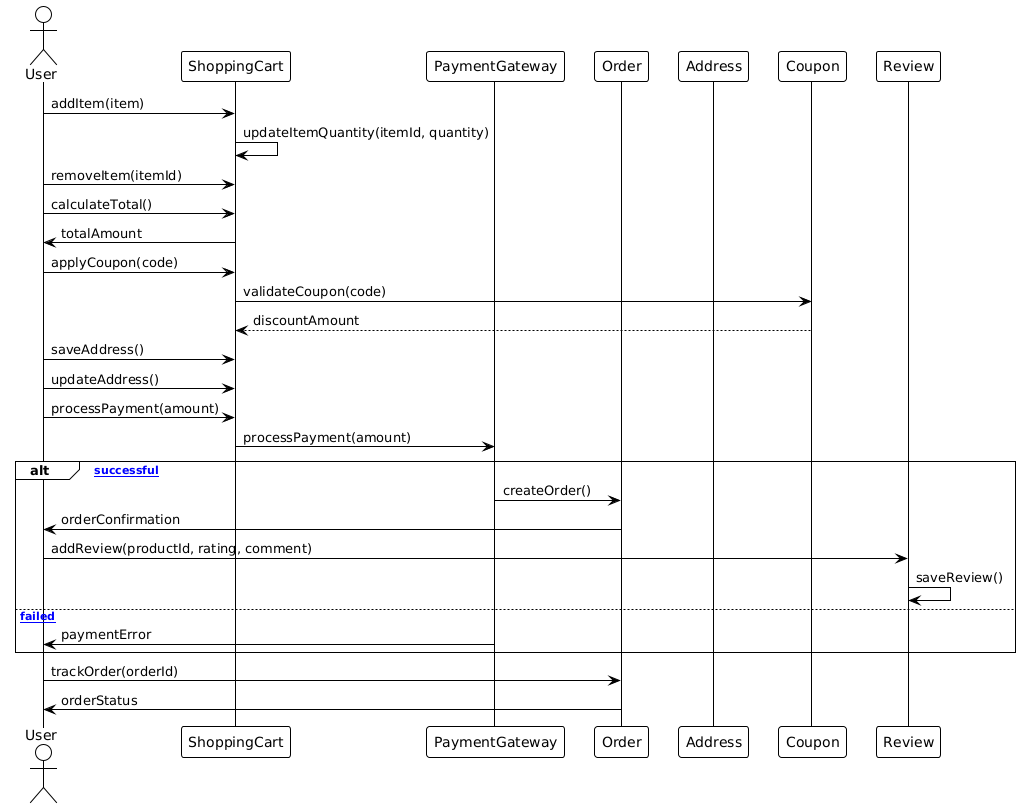
1. **Notification and Email Functionalities**
   * **Objective:** To enhance user engagement and communication by implementing a notification system and integrating email functionalities.
   * **Details:**
     + Develop a real-time notification system to alert users about updates such as order status and system messages.
     + Integrate email services for transactional emails (e.g., order confirmations) and promotional campaigns.
     + Ensure email templates are dynamic, responsive, and customizable.
     + Enable users to subscribe for notifications
2. **Discount and Coupons Functionality**
   * **Objective:** To provide users with the ability to apply discounts and coupons during the checkout process, enhancing customer satisfaction and increasing sales.
   * **Details:**
     + Design a system to generate, validate, and manage discount codes and promotional offers.
     + Enable users to apply valid coupons during checkout.
     + Implement backend validation to prevent misuse or duplication of coupons.
3. **Order History Functionality**
   * **Objective:** To allow users to track and review their past purchases for better user experience and transparency.
   * **Details:**
     + Design a viewable format for the order history of when the order was placed, what the order was, and how much it cost.
     + Description requirements include details for each order that you have delivered and for each order which is being processed delivery status and payment type.
     + Implement sorting and filtering options to enhance usability.
     + Ensure secure access to user-specific order history data.
4. **Data Validation and Security Measures**
   * **Objective:** To maintain the integrity of the system and protect sensitive user data.
   * **Details:**
     + Use strict input validation measures to eliminate the input of either erroneous or incorrect entries.
     + Introduce encryption for sensitive data, such as user credentials.
     + Enforce strong security measures such as password strengths.
5. **Finalization of the Recommendation Engine**
   * **Objective:** To complete the development and optimization of the recommendation engine for personalized user experiences.
   * **Details:**
     + Refine recommendation algorithms to provide accurate and relevant recommendations based on user behavior and preferences.
     + Test and optimize the recommendation engine to ensure scalability and performance.
     + Integrate the recommendation engine seamlessly into the user interface.

**Class Diagram**

****

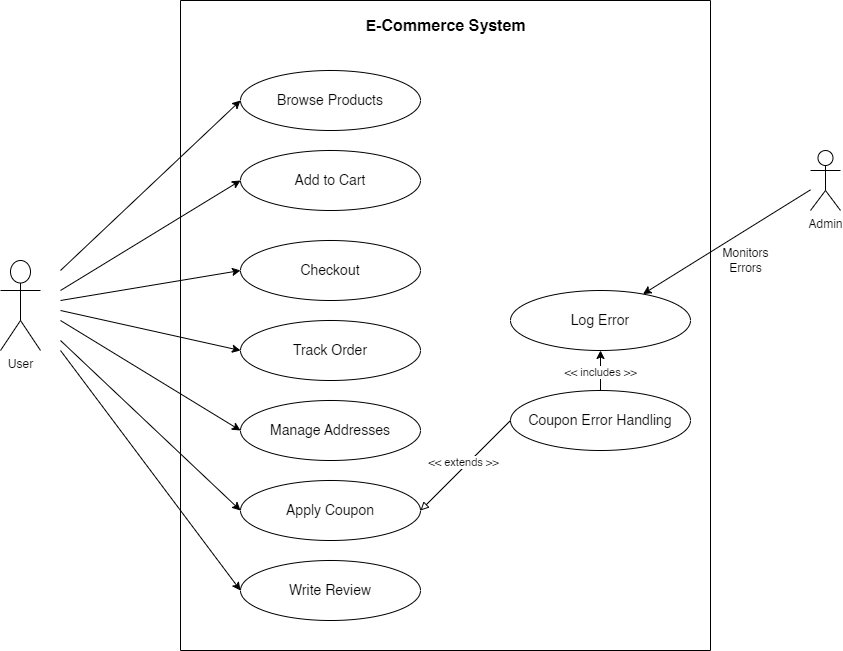
The revised class diagram shows how all the models in the system interrelate and their structure of the models and their interaction and data flow correspond to Phase 3 of development. As mentioned in this phase, it incorporates Account, Address, Subscription subsystems and supports Product, category, brand, review, cart, cart item and coupon subsystems at the same times. The Account class is the key reference point between the user, addresses, subscriptions, reviews or shopping carts. The connections figured are mostly right angled and orthogonal, clearly indicating the fact that the relations are quite well maintained and can be scaled up easily. This diagram gives a plan for Phase 3’s backend implementation in which user customization and e-commerce interfaces merge: default addresses/subscription, cart/interface, coupons.

**Sequence Diagram**



The updated sequence diagram for Phase 3 reflects the integration of new features, enhancing the e-commerce system's functionality. The user interacts with the ShoppingCart to add, remove, or update items, calculate totals, apply coupons (validated via the Coupon system), and process payments through the PaymentGateway. If the payment is successful, an order is created (Order), and the user receives a confirmation, along with the option to add reviews for purchased products via the Review system. In case of payment failure, an error message is returned. Additionally, users can now track their orders by querying the Order system for the current status. These updates streamline the shopping experience by incorporating coupon validation, product reviews, and order tracking into the workflow.

**Use Case Diagram**

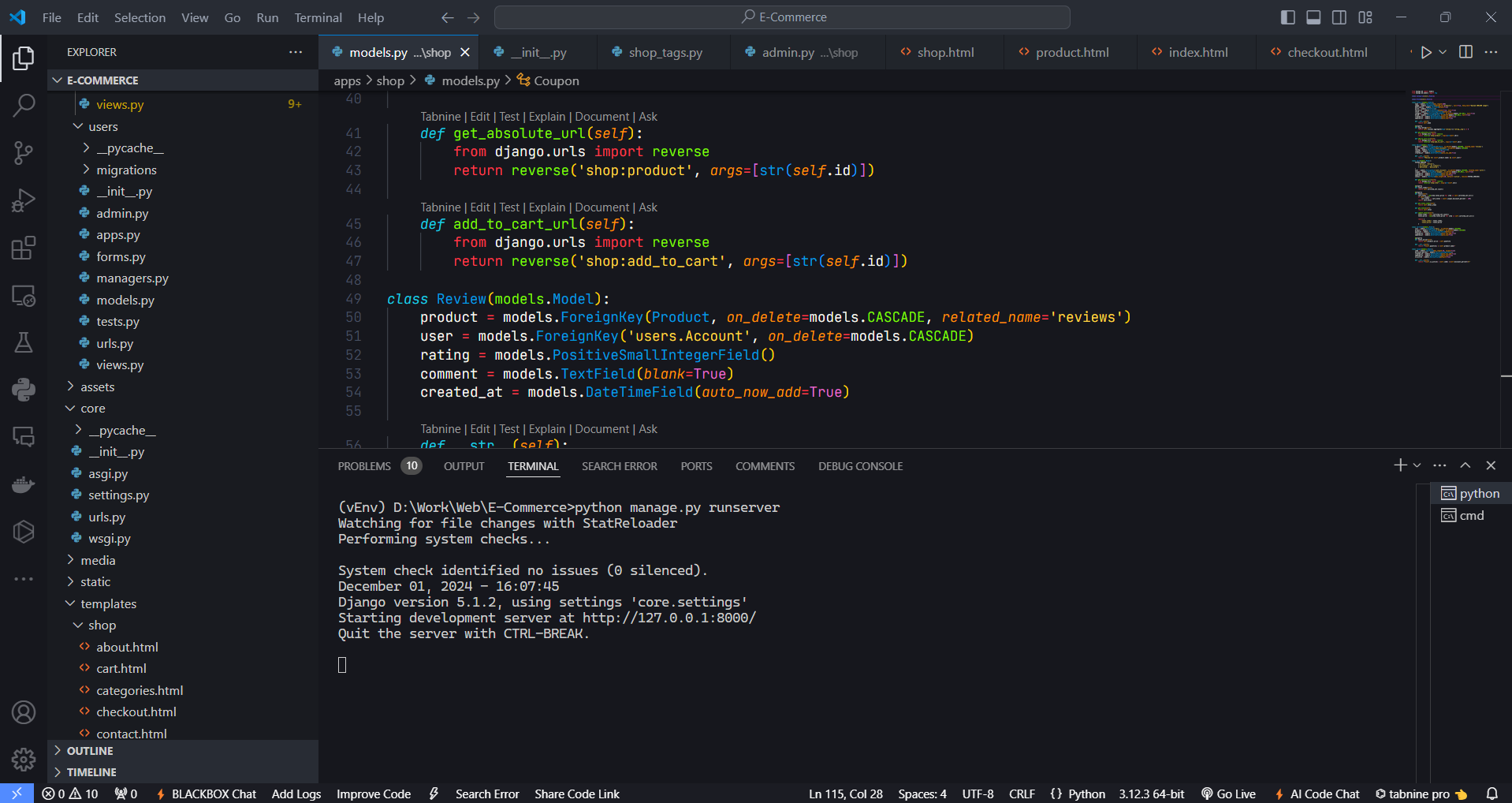
****

The updated Use Case Diagram for Phase 3 reflects the enhanced functionality of the e-commerce system, incorporating both normal and error cases. Users can perform standard actions such as browsing products, adding items to the cart, managing addresses, applying coupons, tracking orders, and writing reviews for purchased products. During checkout, the system handles payment errors through the "Coupon Error Handling" use case, which extends the apply coupon process and includes logging errors for administrative monitoring. An Admin actor has been added to oversee logged errors, ensuring effective issue resolution. This diagram highlights the system's robustness in managing both user interactions and error scenarios.

**Test Cases**

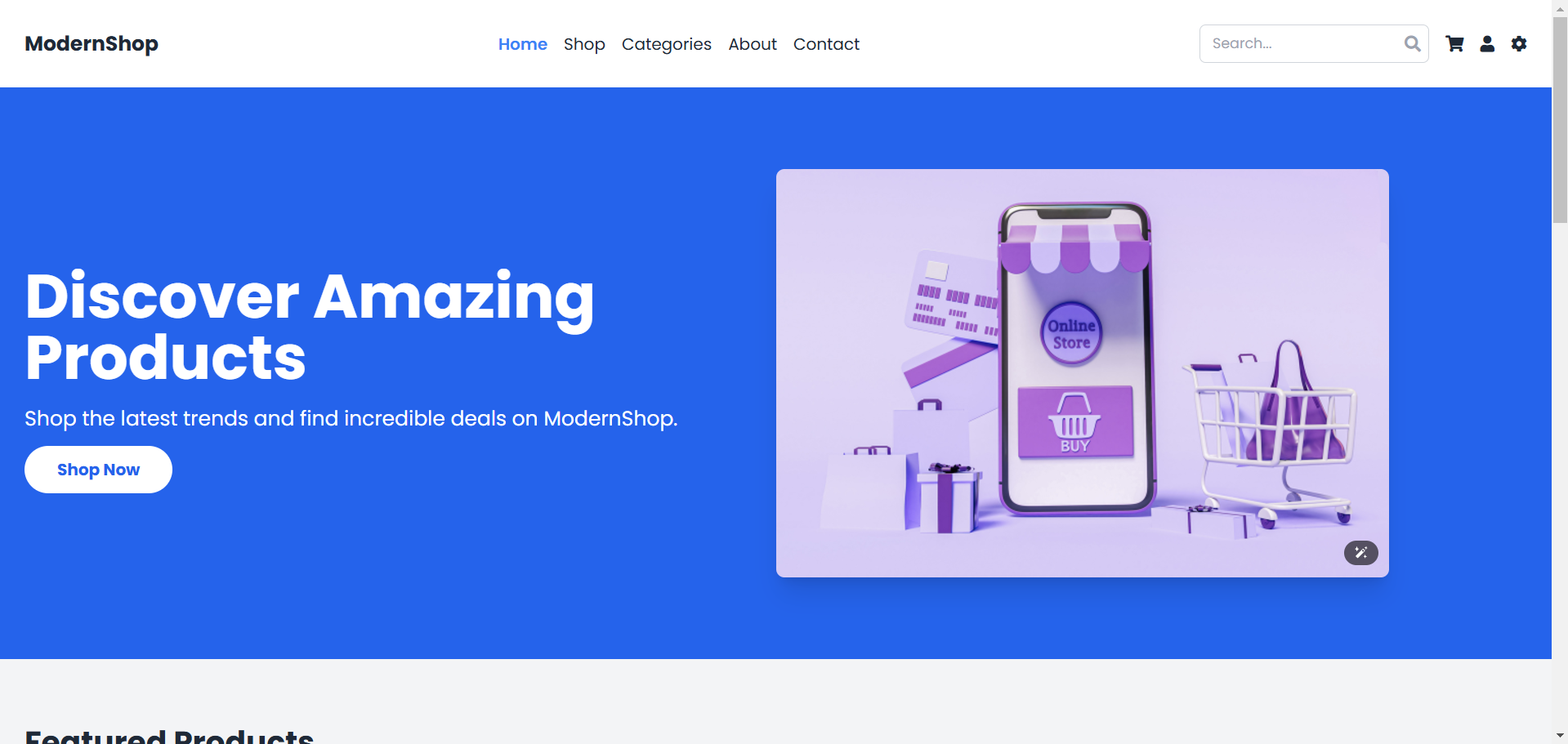
**Phase 1: Core Feature Development**

1. **Project Setup Test**



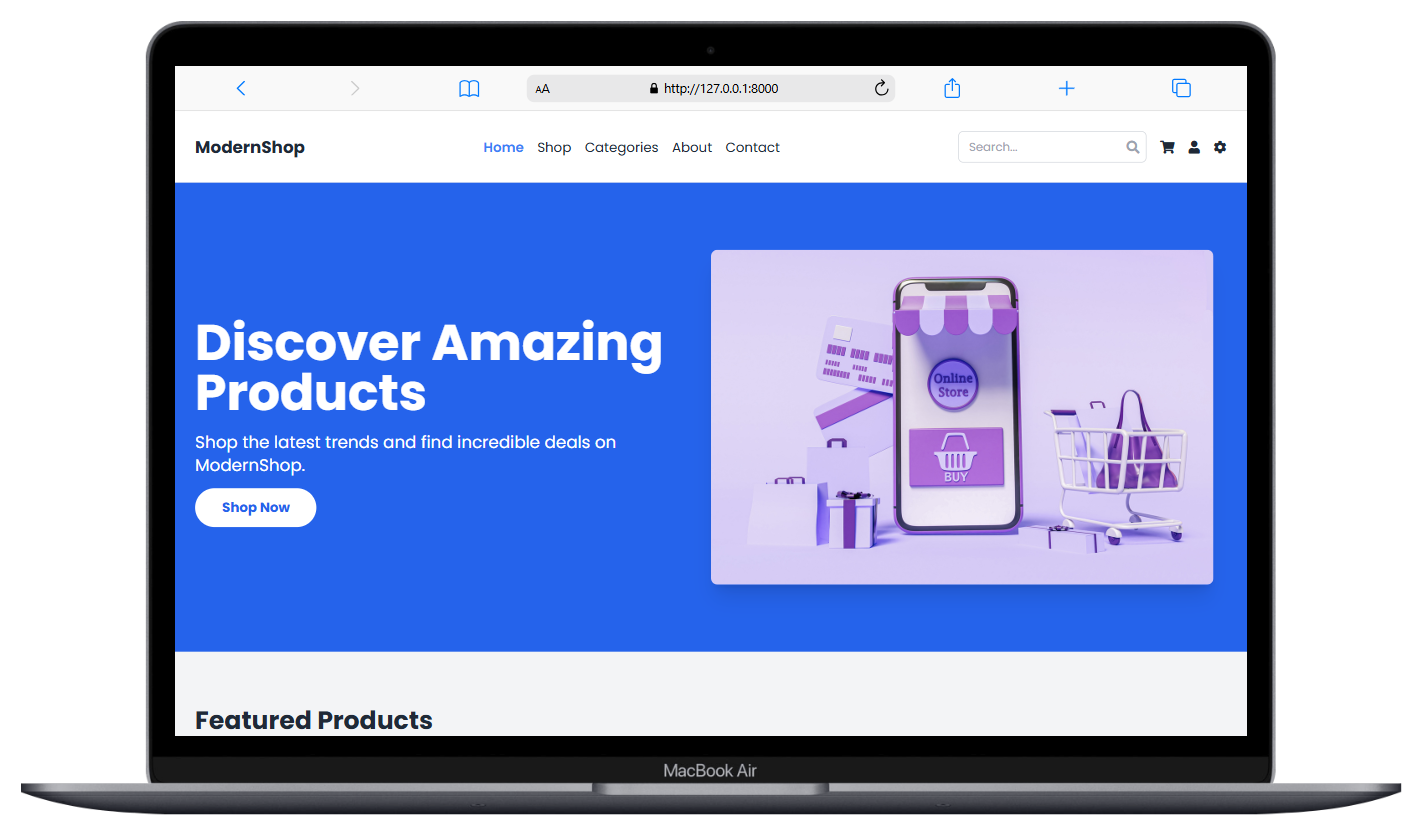
* + **Description**: Verifies the initial setup of the project, including cloning the repository, booting the server, and accessing the application.
  + **Inputs**: Repository URL and server start command.
  + **Expected Output**: The application runs without errors and is accessible at http://localhost:8000.

1. **User Interface Design Test**



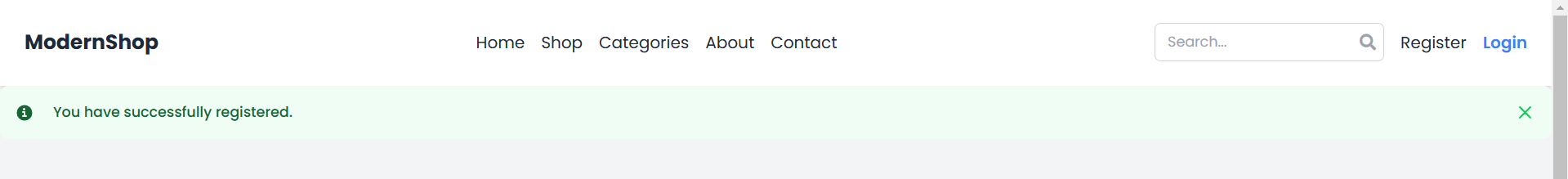
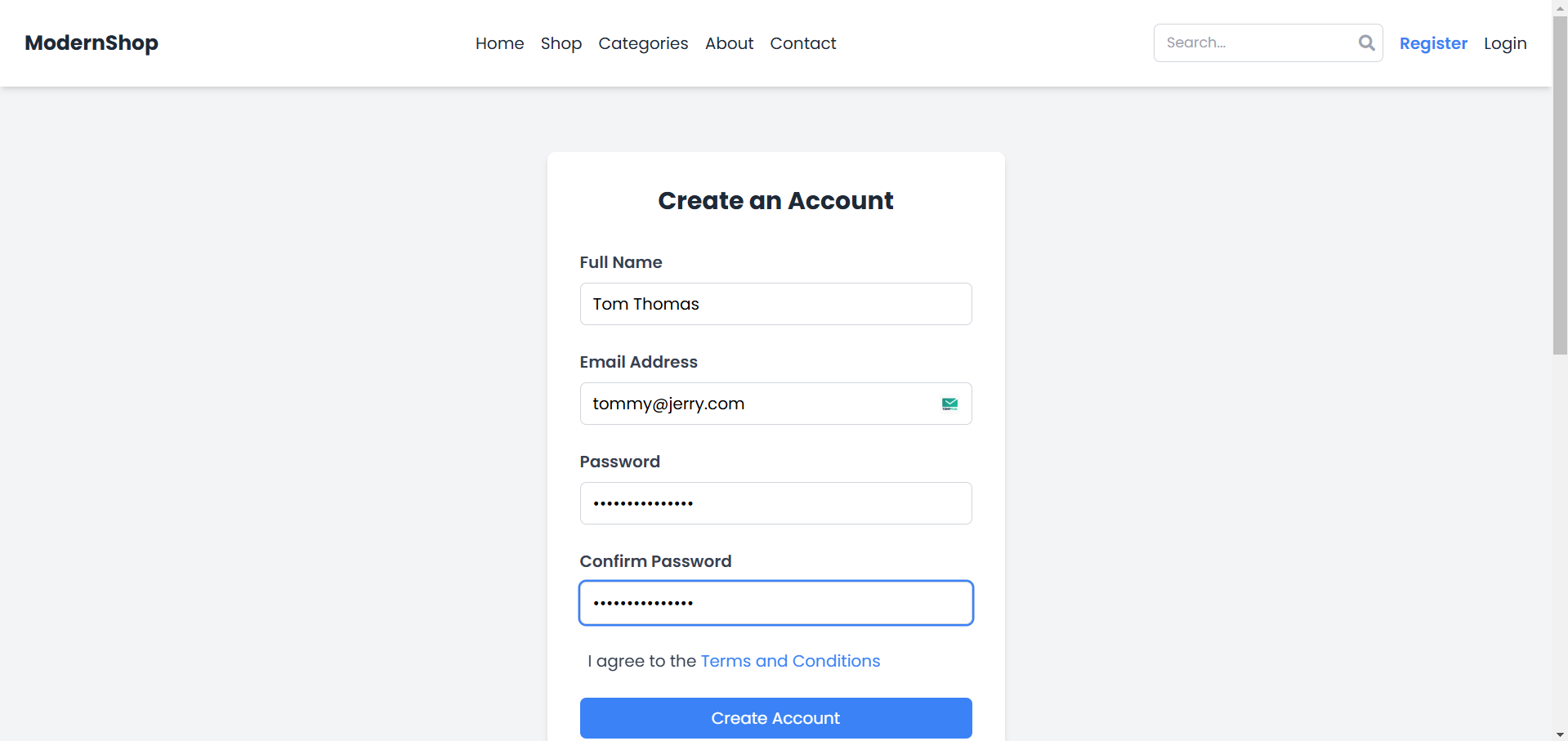
* + **Description**: Ensures the consistency of the application's look and feel across all pages.
  + **Inputs**: Navigation through different pages of the application.
  + **Expected Output**: Uniform style, easy navigation, and proper layout on all pages.

1. **Large Screen Responsiveness Test**



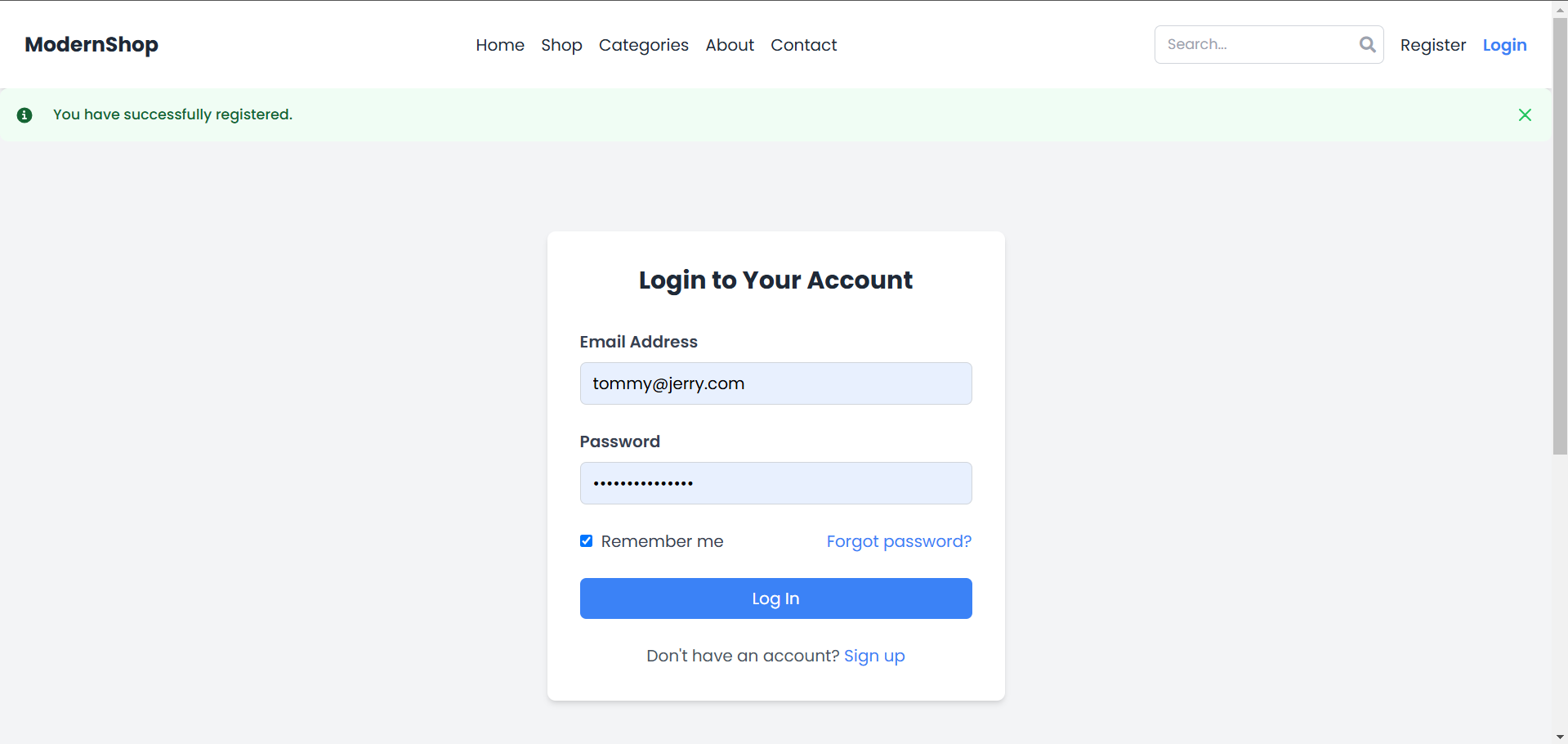
* + **Description**: Tests the application's performance on large screens (e.g., desktop and tablet).
  + **Inputs**: Screen resolutions (1920×1080, 1366×1024).
  + **Expected Output**: Content fits the screen without horizontal scrolling, and all buttons are clickable.

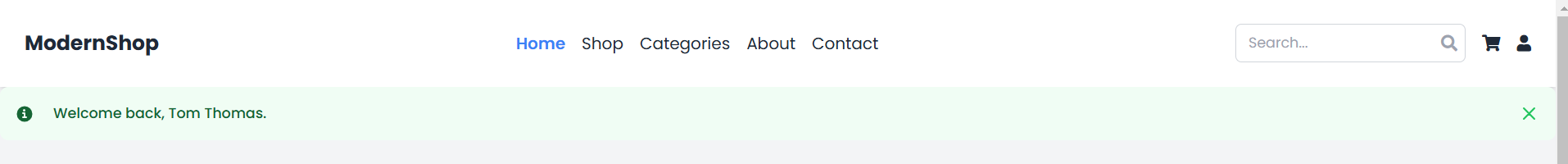
1. **User Registration Test**



* + **Description**: Validates the signup process for new users.
  + **Inputs**: User details (e.g., email, password, username).
  + **Expected Output**: Successful account creation, redirection to the correct page, and ability to log in with new credentials.

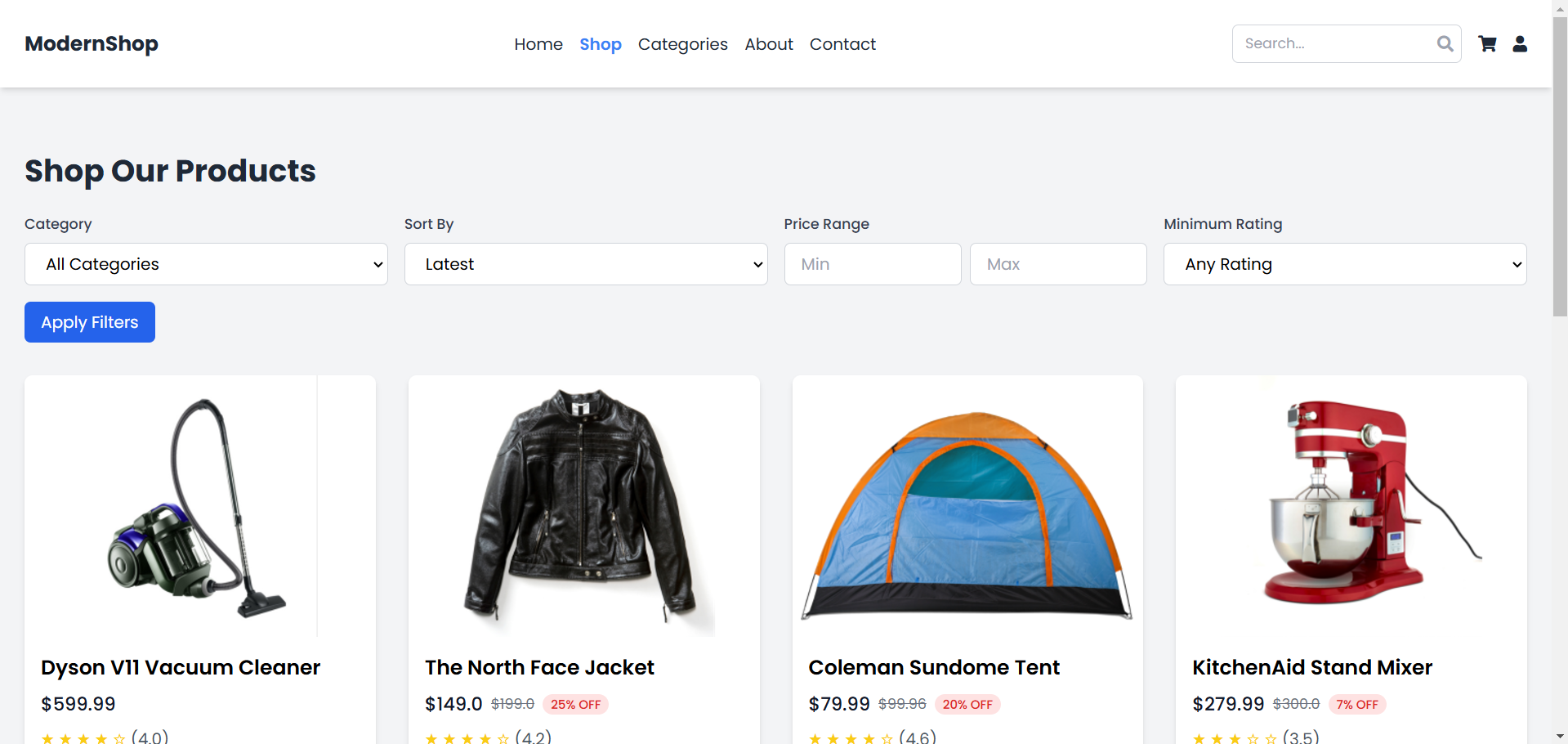
1. **User Login Test**





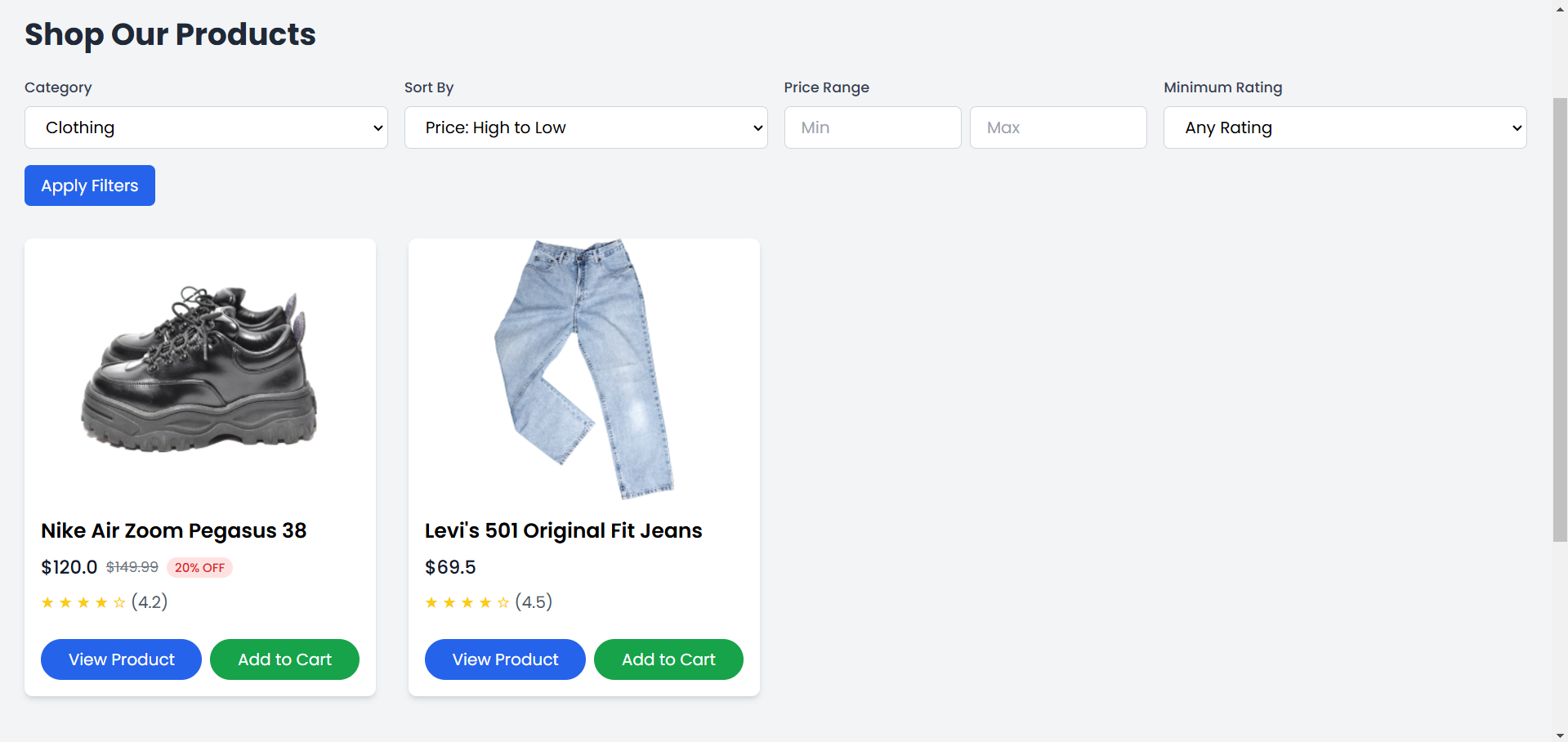
* + **Description**: Ensures that users can log in with valid credentials.
  + **Inputs**: Valid username and password.
  + **Expected Output**: Successful authentication, redirection to the user dashboard, and display of user-specific content.

1. **Product Addition Test (Admin)**
   * **Description**: Verifies that admins can add new products through the admin interface.
   * **Inputs**: Product details (e.g., name, price, category).
   * **Expected Output**: The product is added to both the admin panel and the main site product list.
2. **Product Listing Test**



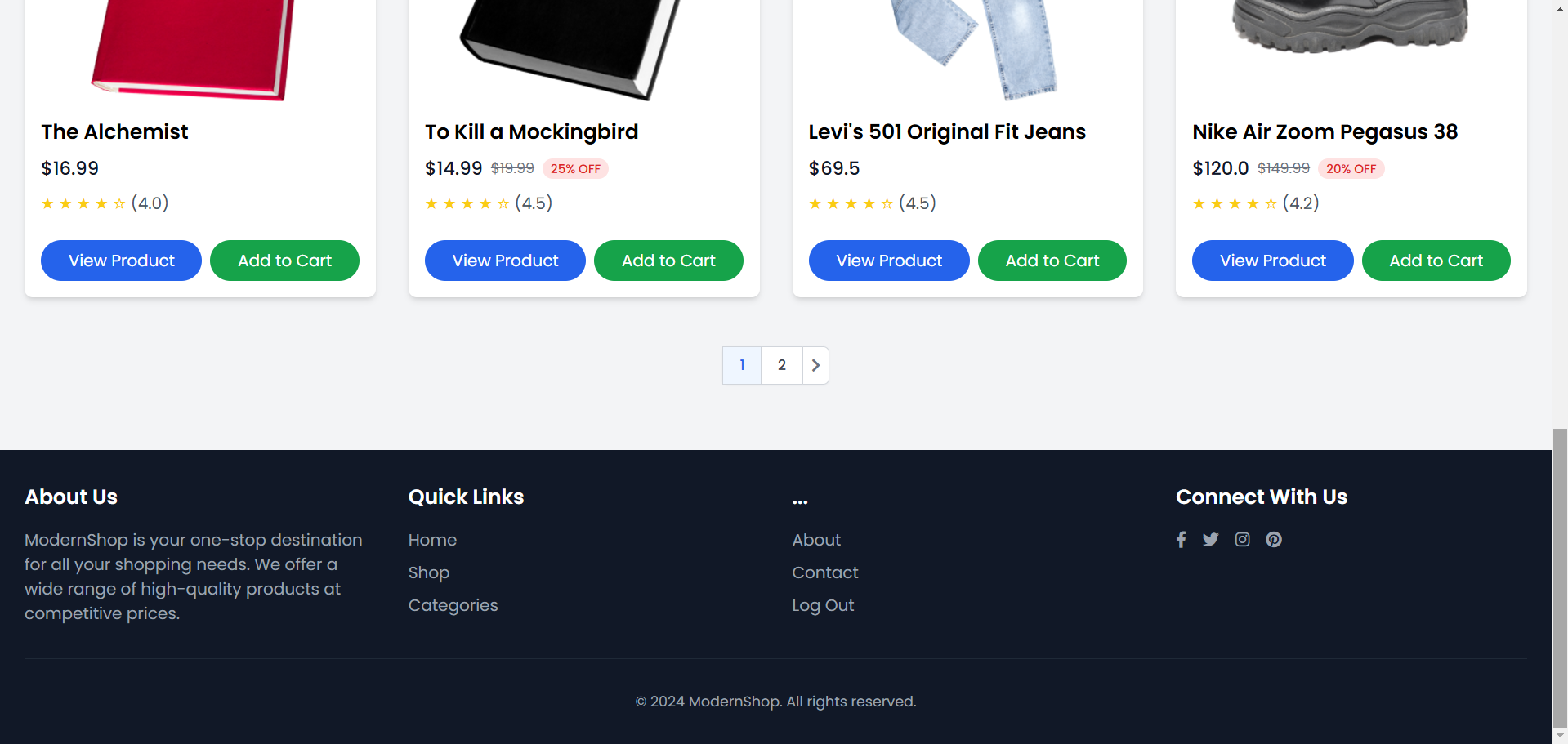
* + **Description**: Ensures that the product index page displays products correctly.
  + **Inputs**: Existing product data.
  + **Expected Output**: Proper display of product images, names, prices, and availability.

1. **Product Filtering Test**



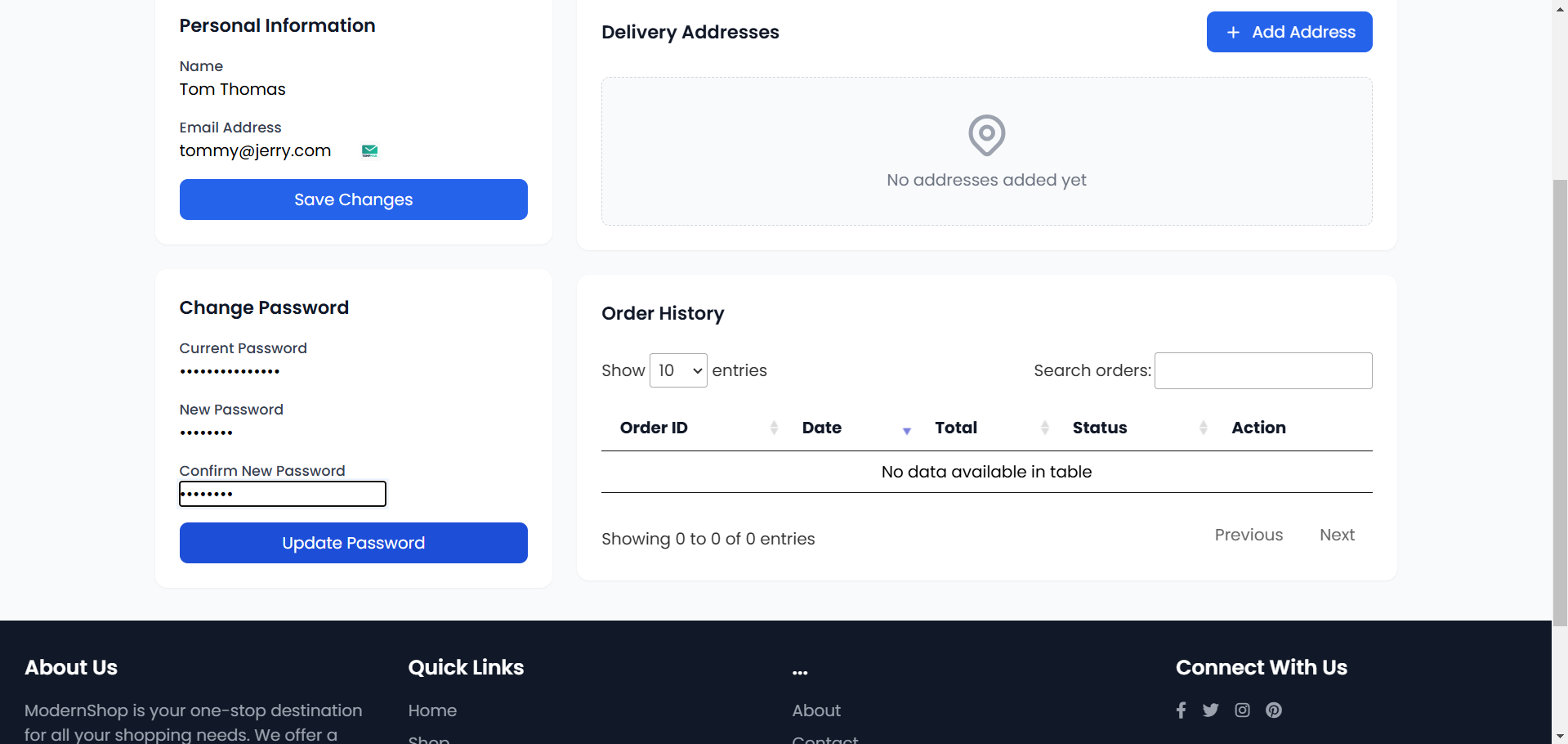
* + **Description**: Tests the functionality of filtering products based on criteria (e.g., category, price).
  + **Inputs**: Filter options.
  + **Expected Output**: Accurate results for individual and combined filters.

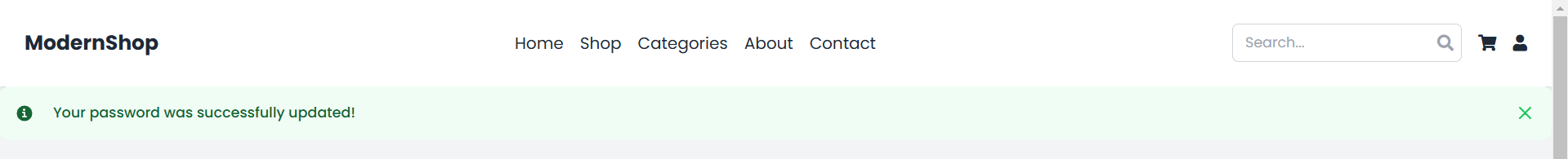
1. **Product Pagination Test**



* + **Description**: Ensures proper pagination of product listings.
  + **Inputs**: Number of products per page.
  + **Expected Output**: Correct display of products per page and smooth page transitions.

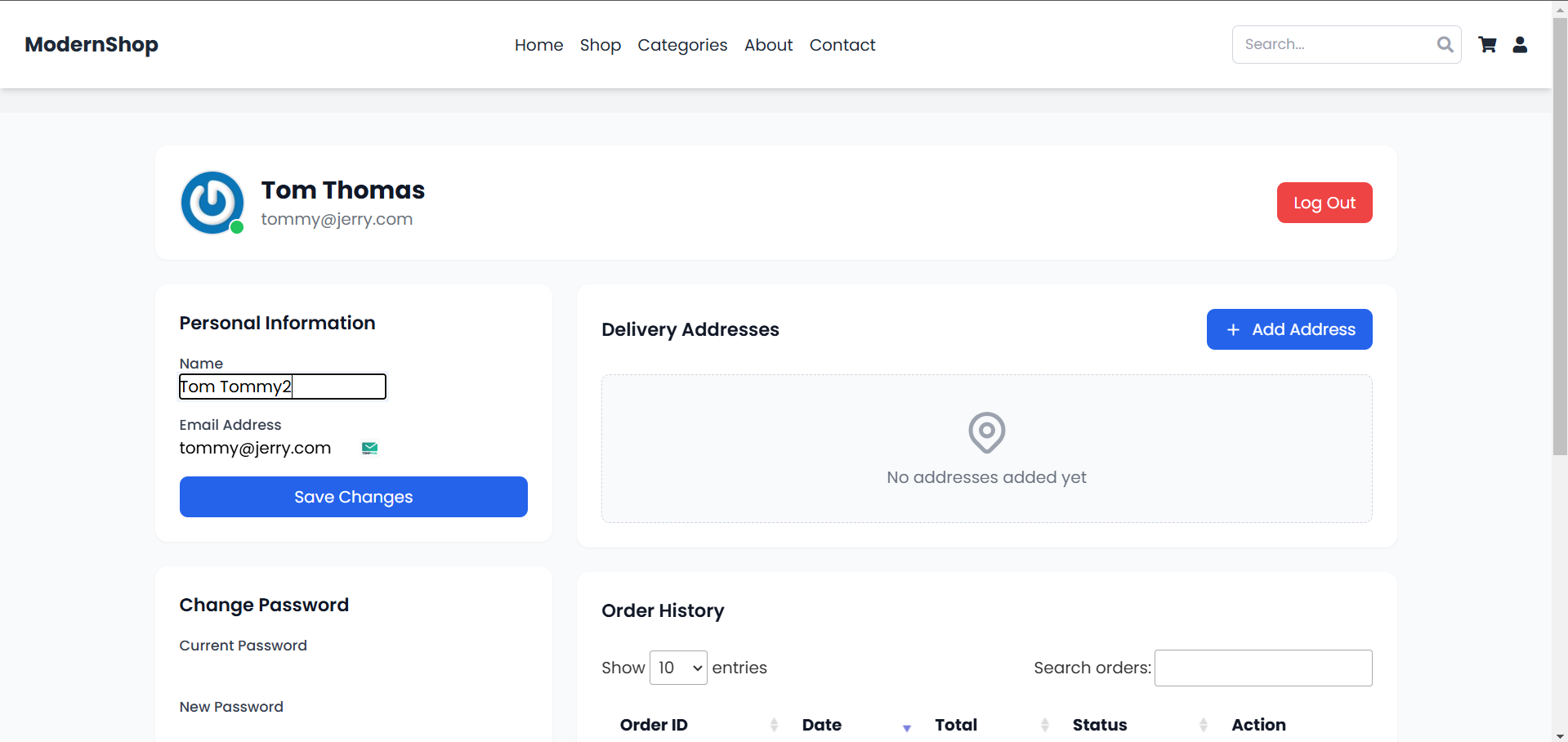
1. **User Password Change Test**

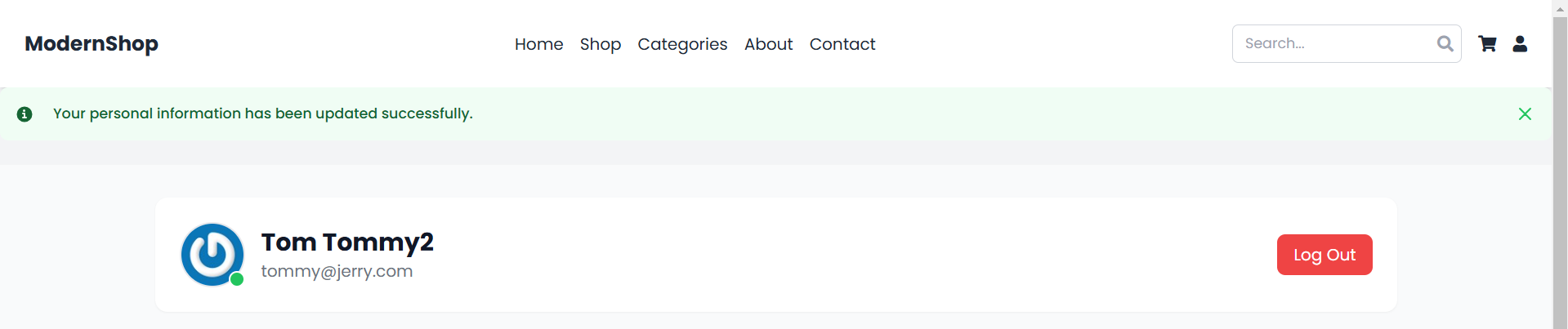




* + **Description**: Tests the functionality of updating user passwords.
  + **Inputs**: Old password, new password.
  + **Expected Output**: Successful password update and ability to log in with the new password.

1. **User Profile Update Test**

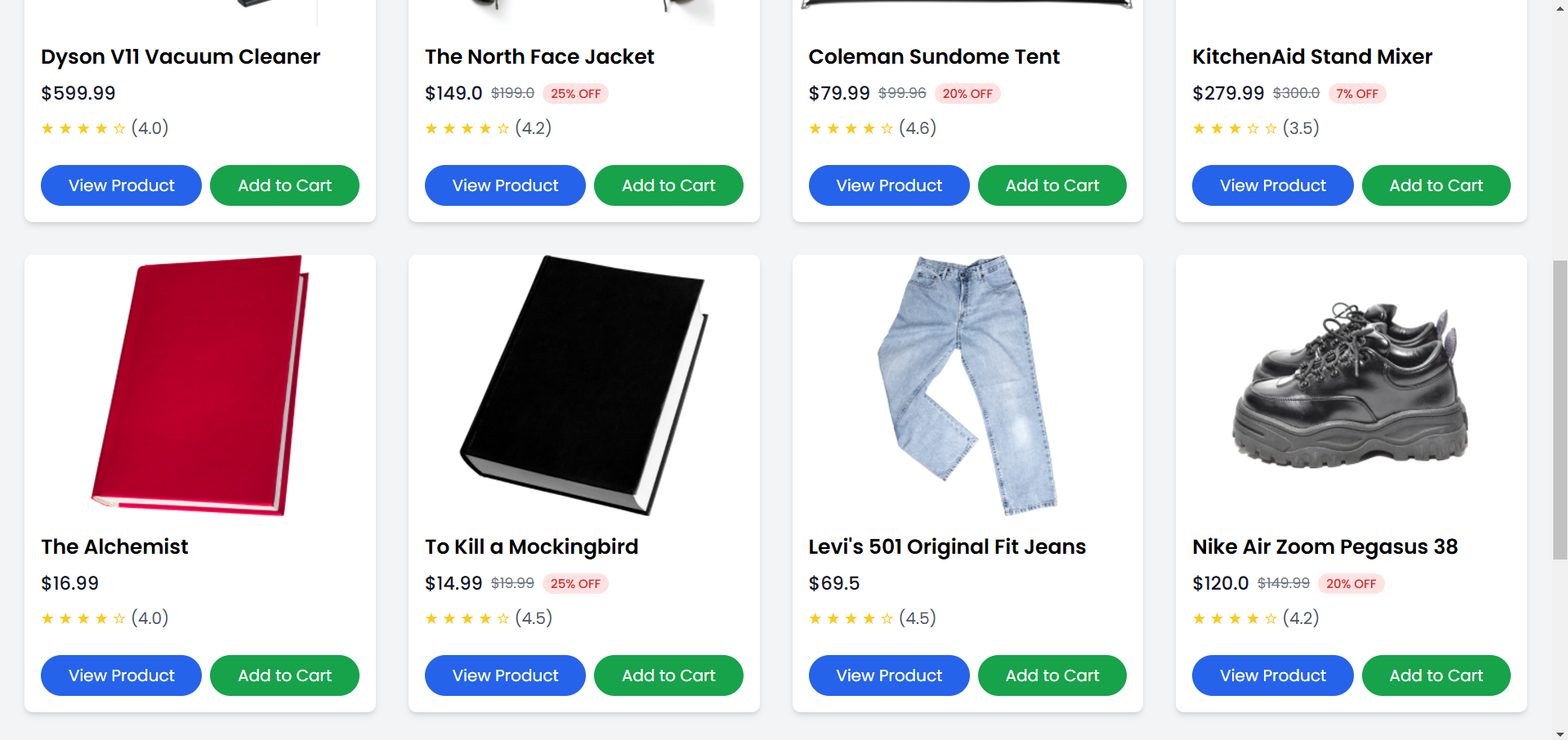




* + **Description**: Verifies that users can edit their personal details.
  + **Inputs**: Updated user information (e.g., name, email).
  + **Expected Output**: Successful update of user profile details.

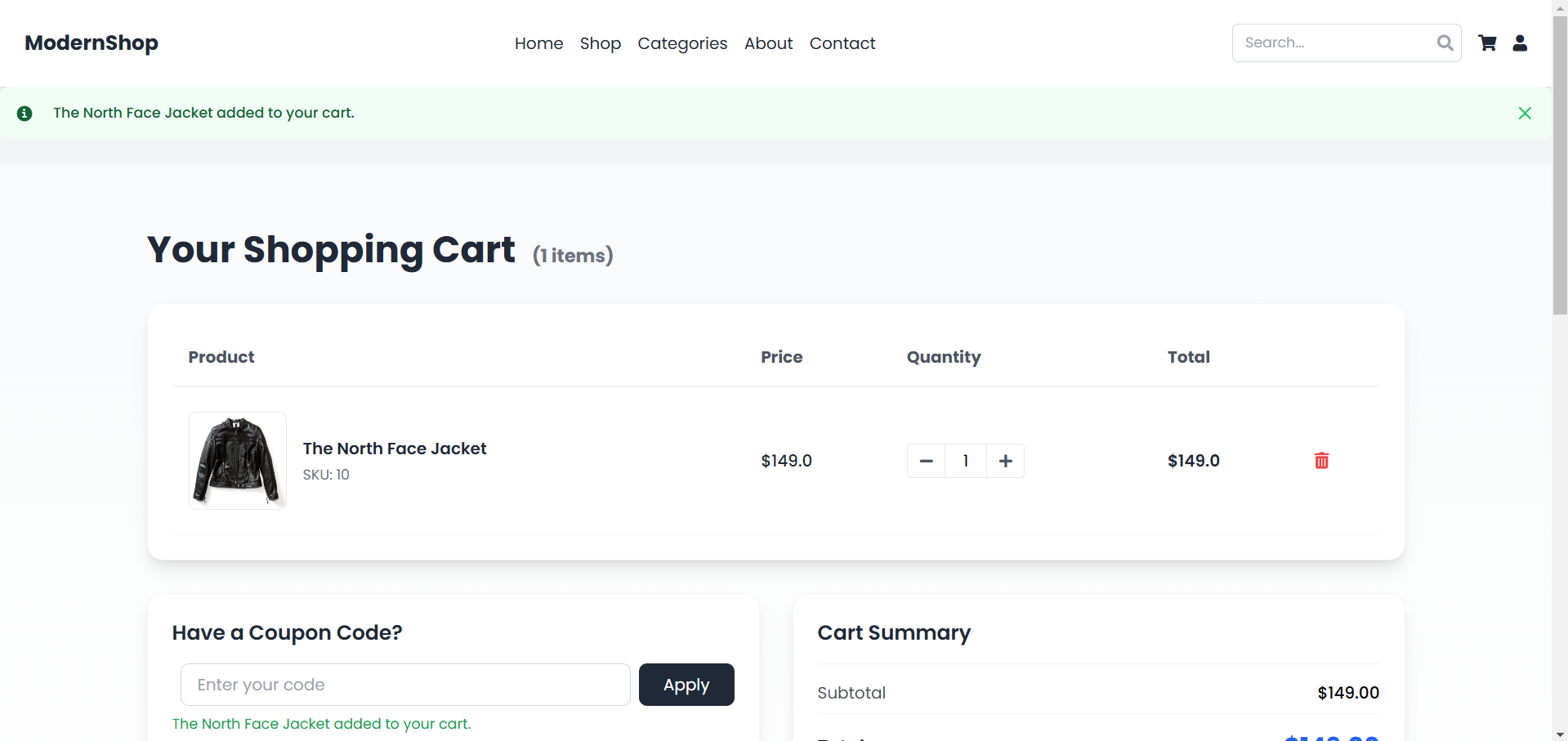
**Phase 2: Core Feature Expansion**

1. **Browse Products**



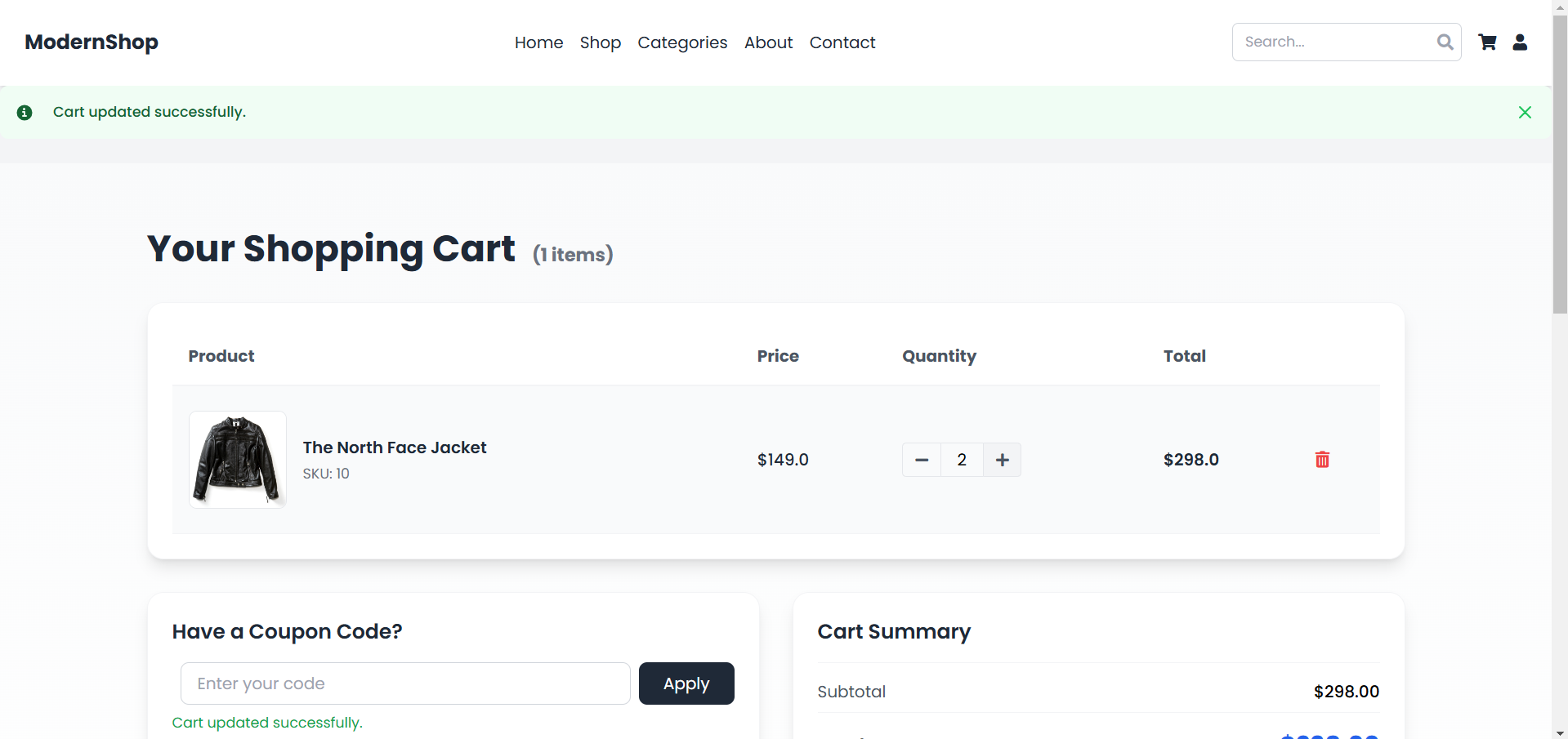
* + **Description**: Tests the ability to view a list of products with details like name, price, and availability.
  + **Inputs**: Navigation to the product browsing page.
  + **Expected Output**: A list of products is displayed with accurate details.

1. **Add to Cart**



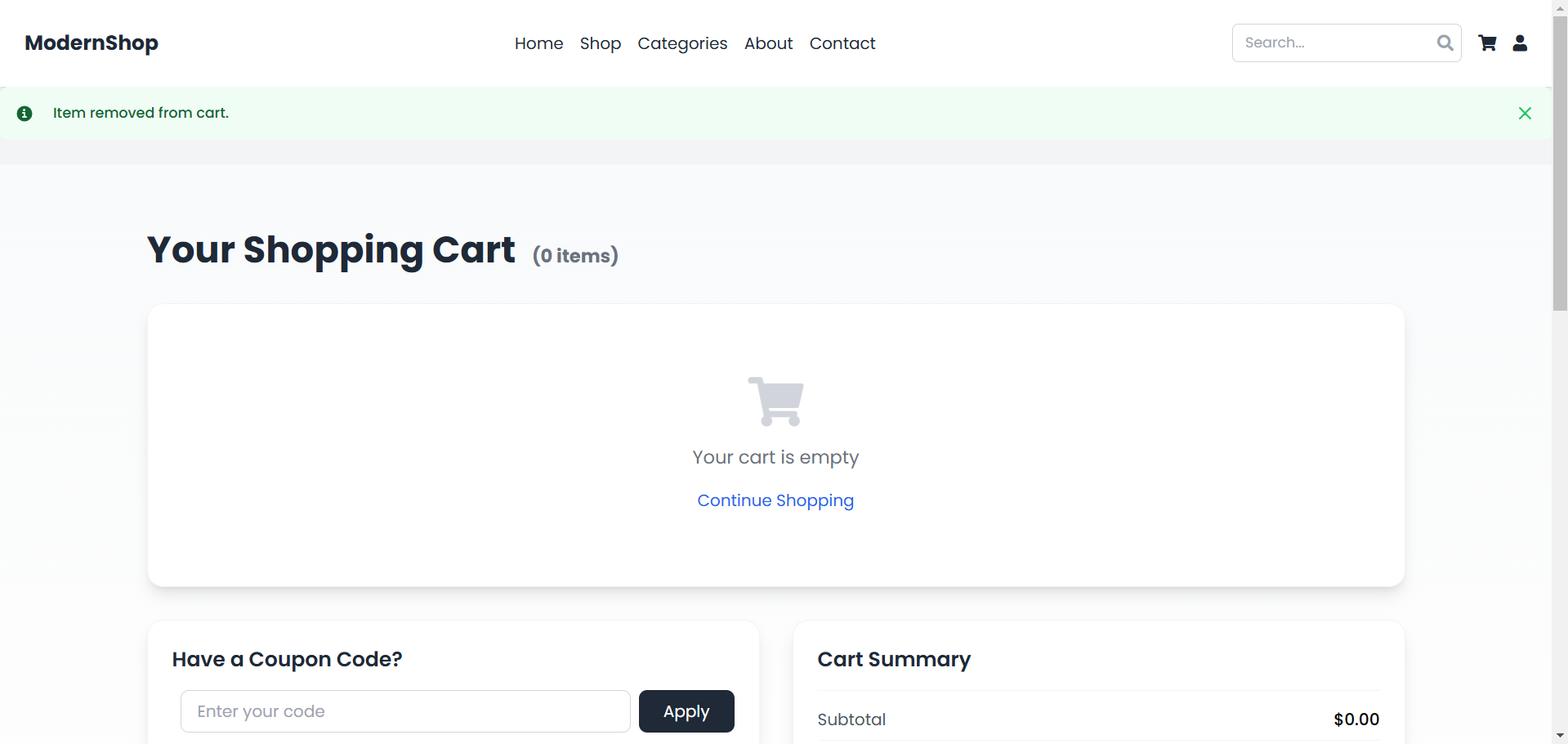
* + **Description**: Ensures that selected products are added to the shopping cart.
  + **Inputs**: Product selection and "Add to Cart" action.
  + **Expected Output**: The product is added to the cart, and the cart count updates.

1. **Update Cart Quantity**



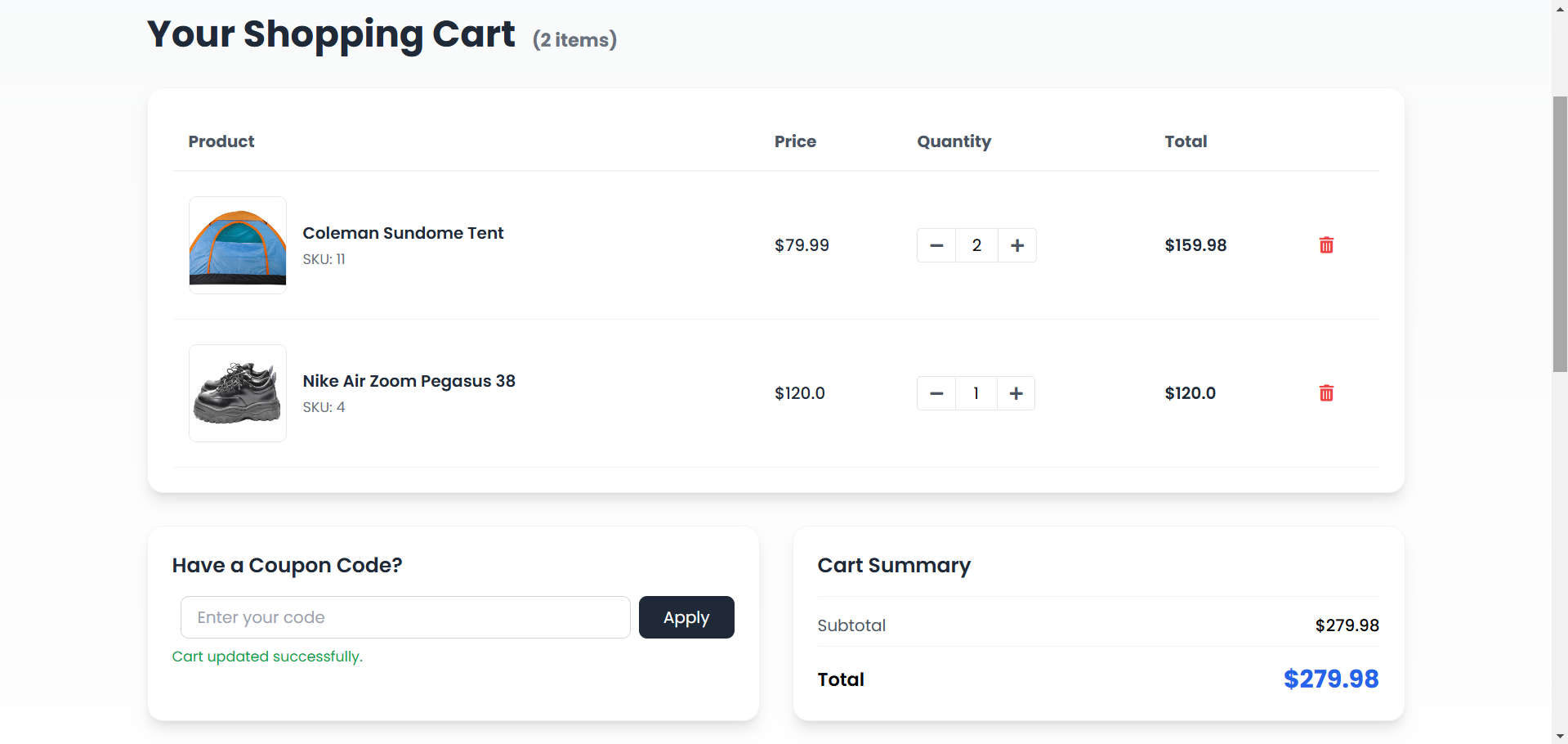
* + **Description**: Verifies that users can modify the quantity of items in their cart.
  + **Inputs**: Updated item quantity.
  + **Expected Output**: The cart reflects the updated quantity, and the total price adjusts accordingly.

1. **Remove from Cart**



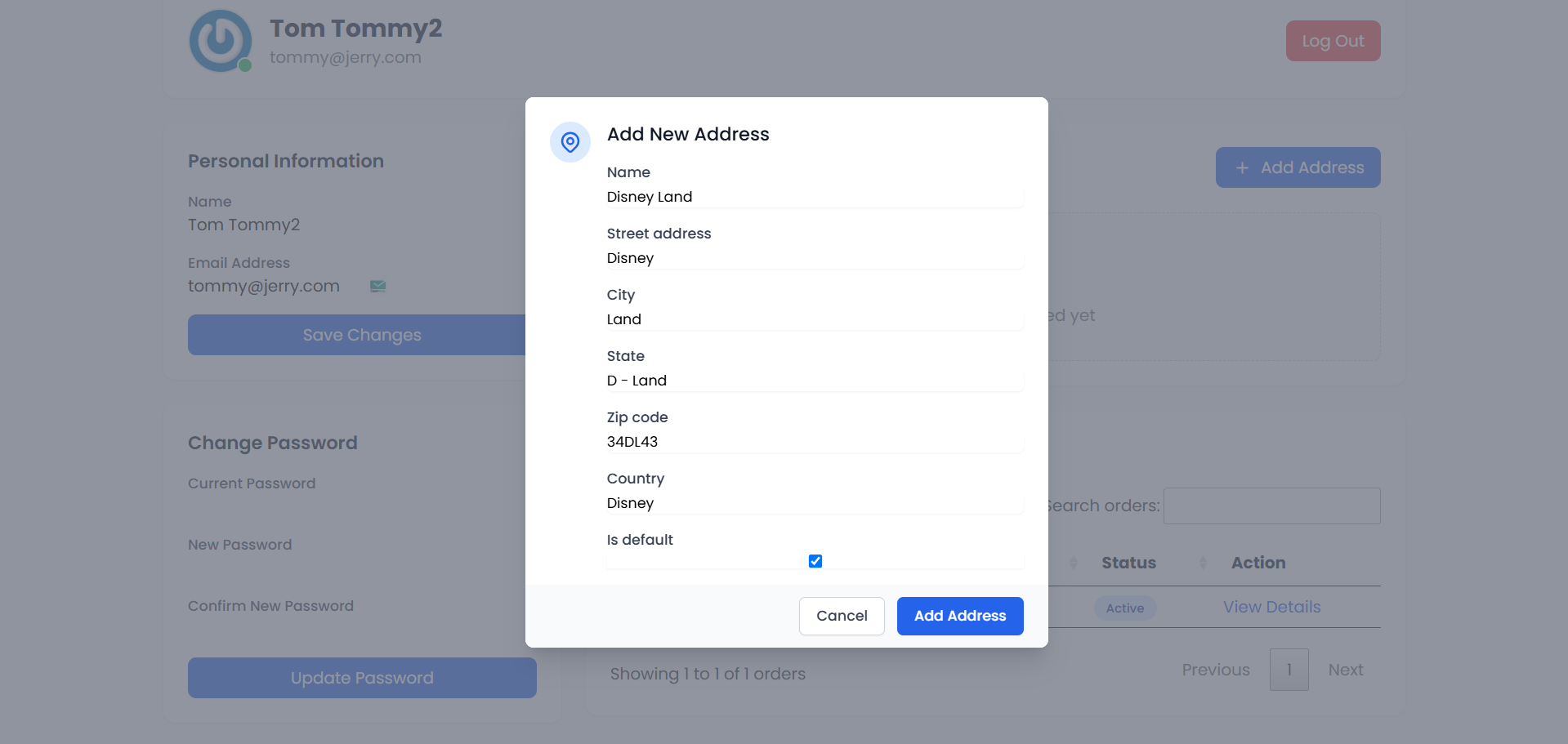
* + **Description**: Tests the ability to remove items from the shopping cart.
  + **Inputs**: Selection of an item and "Remove" action.
  + **Expected Output**: The item is removed, and the cart count decreases.

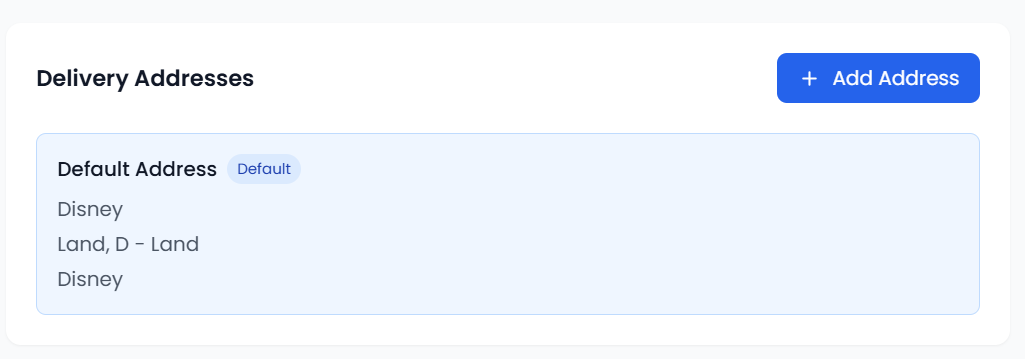
1. **Compute Total**



* + **Description**: Ensures that the total cost at checkout includes item prices, taxes, and shipping fees.
  + **Inputs**: Items in the cart.
  + **Expected Output**: The total reflects accurate calculations.

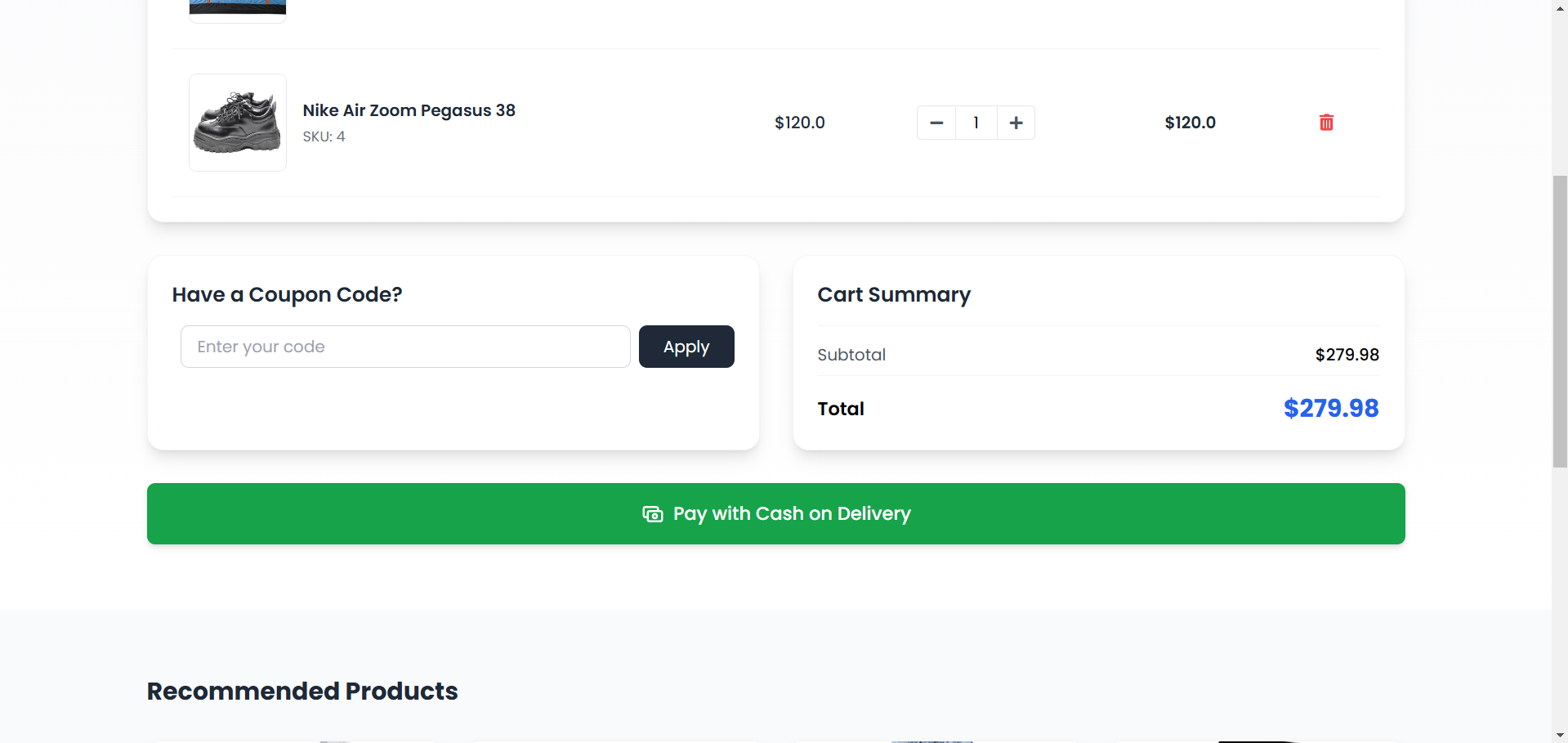
1. **Save Address**

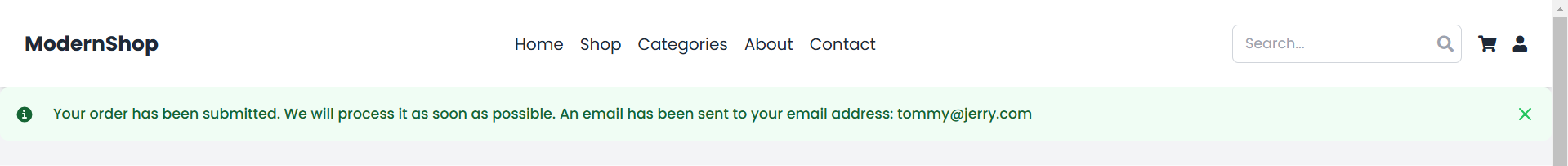




* + **Description**: Verifies that users can save new shipping addresses.
  + **Inputs**: Address details.
  + **Expected Output**: The address is saved and appears in the user's address book.

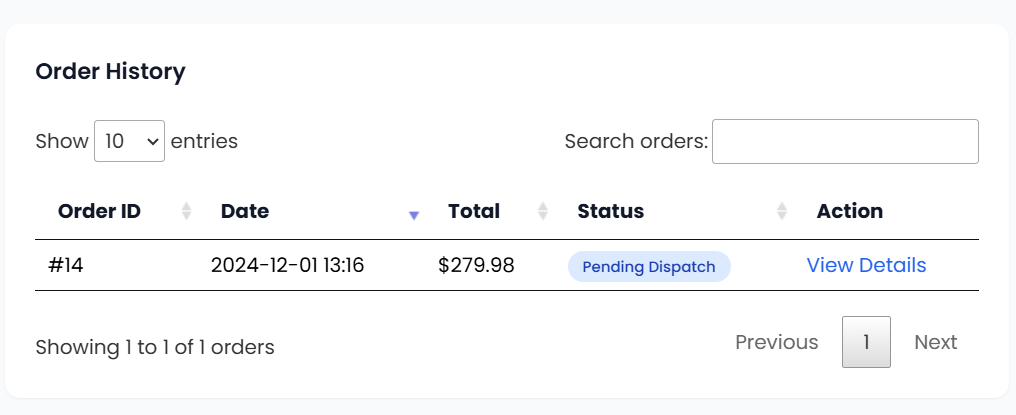
1. **Payment Processed Successfully**





* + **Description**: Tests successful payment with pay on delivery details.
  + **Inputs**: Valid payment details.
  + **Expected Output**: Payment is processed, and an order confirmation is sent to the user.

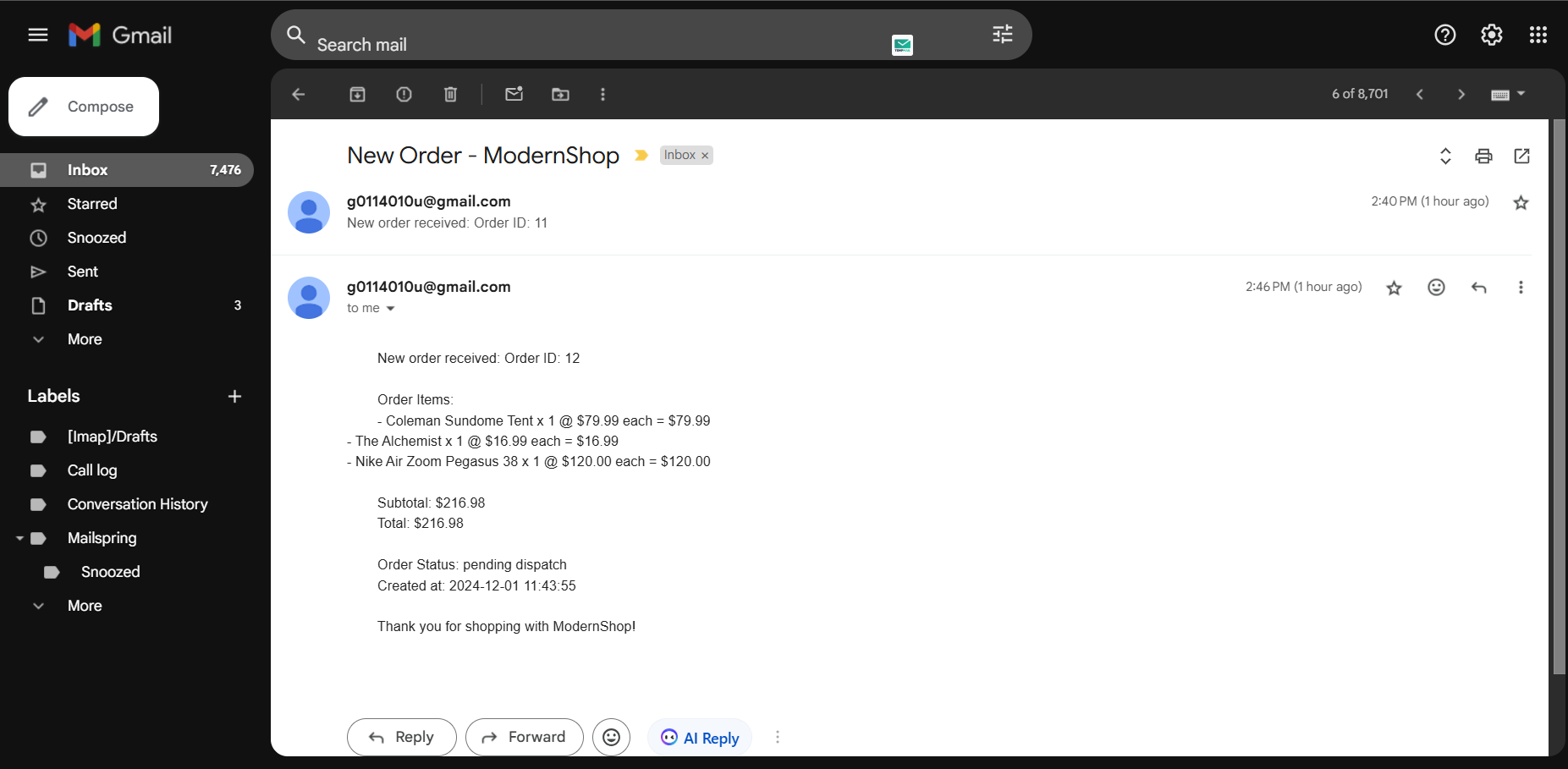
1. **Track Order**



* + **Description**: Verifies the ability to track orders after purchase.
  + **Inputs**: Valid order ID.
  + **Expected Output**: The order status is displayed accurately.

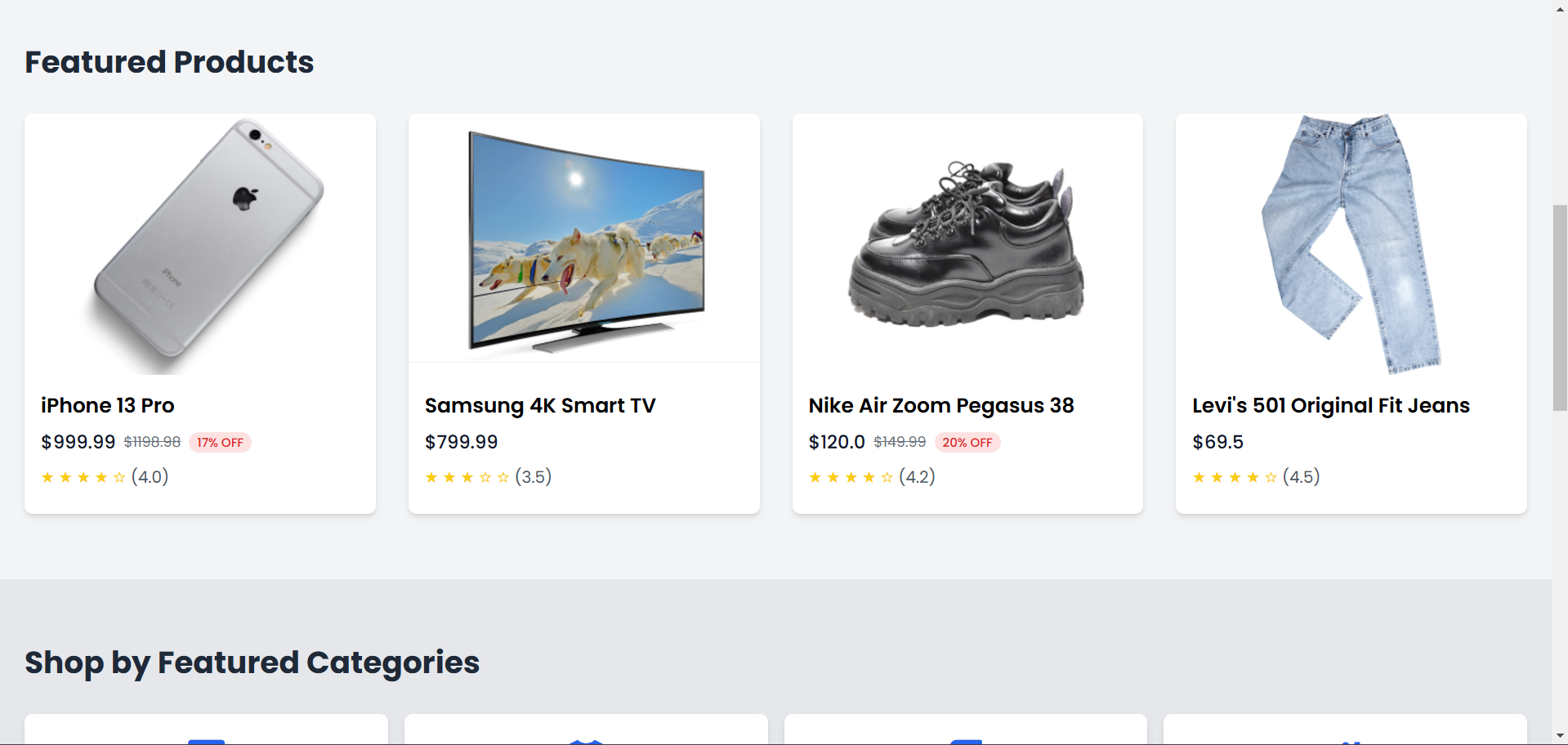
**Phase 3: Secondary Feature Development**

1. **Notification and Email Functionality**



* + **Description**: Tests the system's ability to send notifications and emails (e.g., order updates).
  + **Inputs**: User email and notification triggers (e.g., order confirmation).
  + **Expected Output**: Notifications and emails are sent successfully.

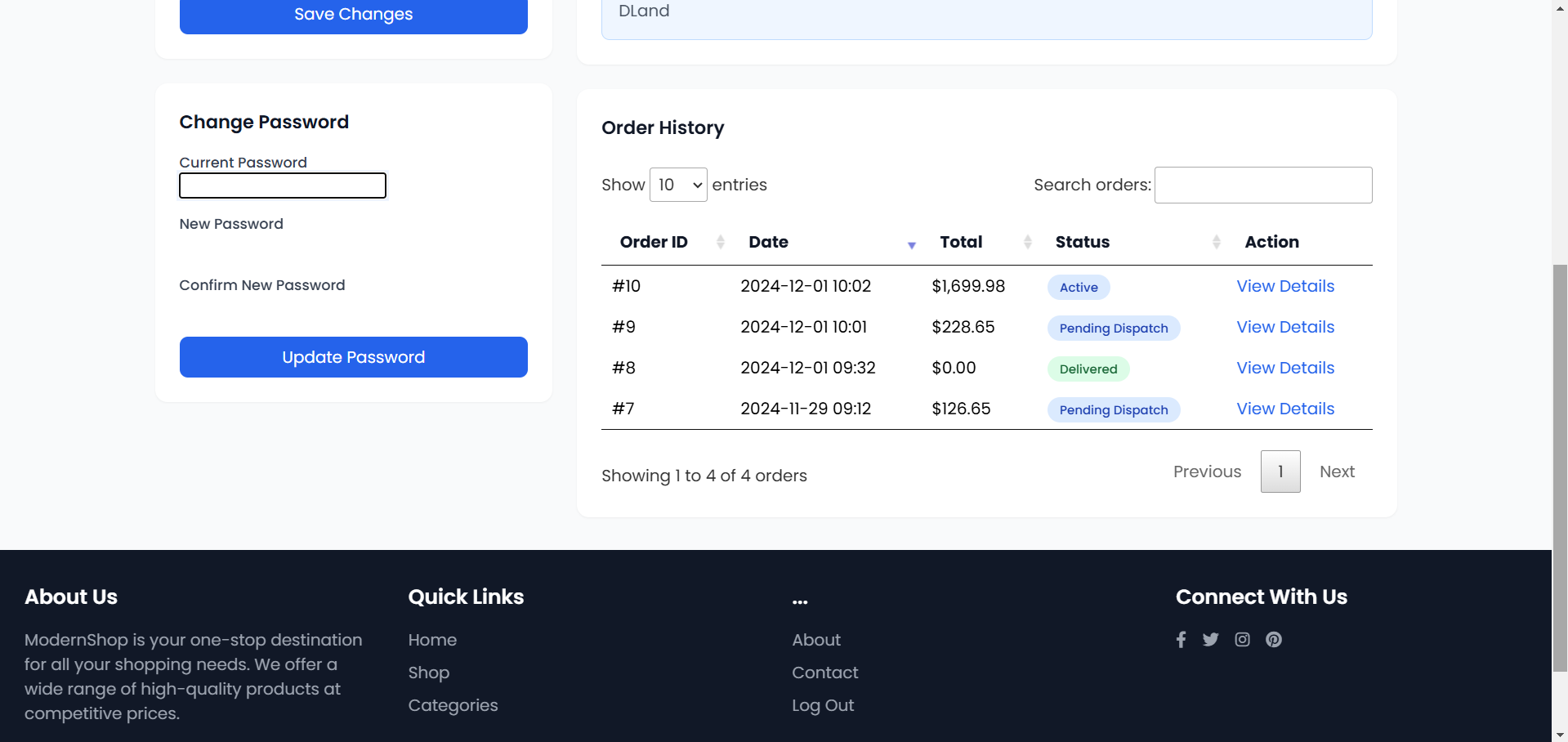
1. **Discount and Coupons Functionality**





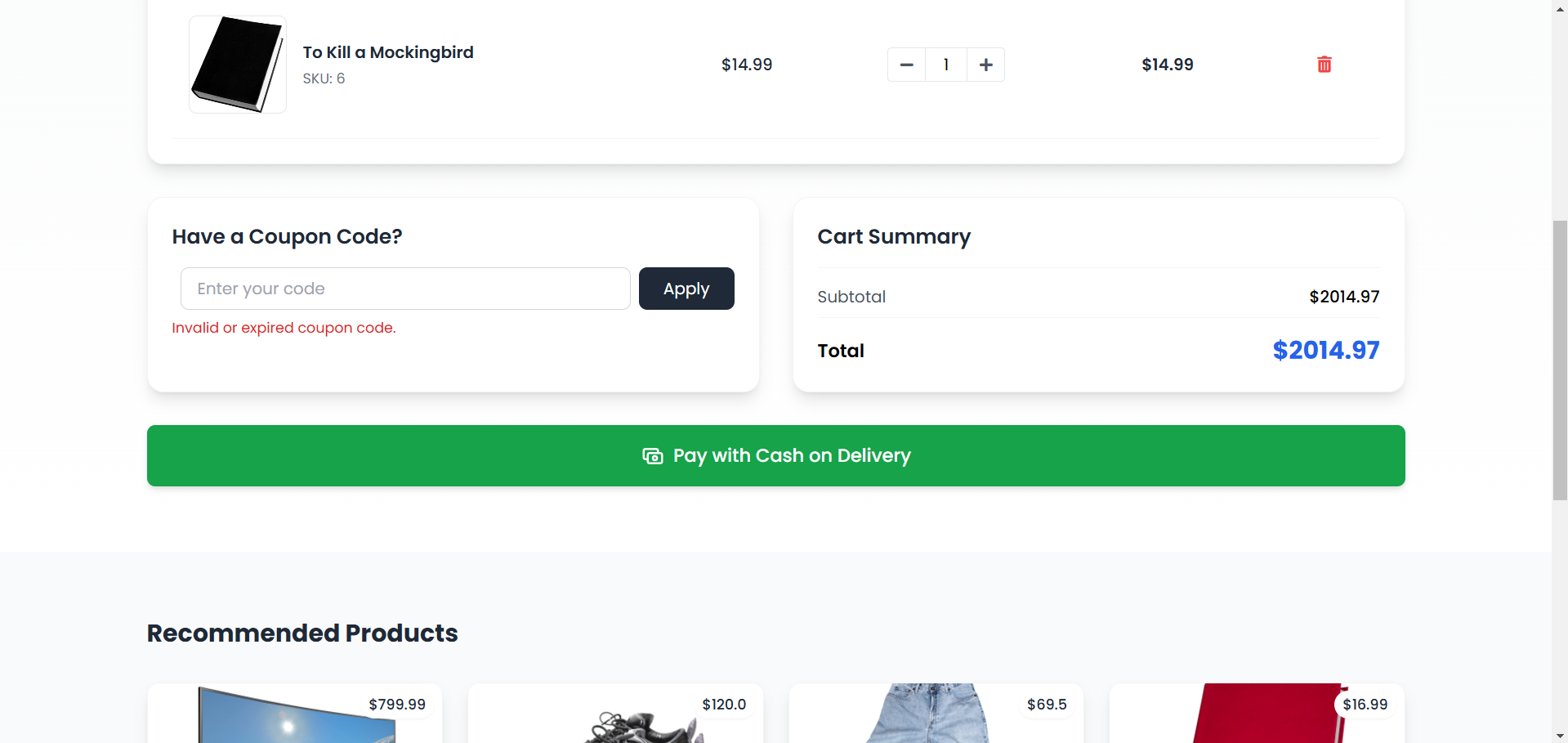
* + **Description**: Verifies the application of valid and invalid coupons during checkout.
  + **Inputs**: Valid and invalid coupon codes.
  + **Expected Output**: Valid coupons apply discounts, while invalid coupons return error messages.

1. **Order History Functionality**



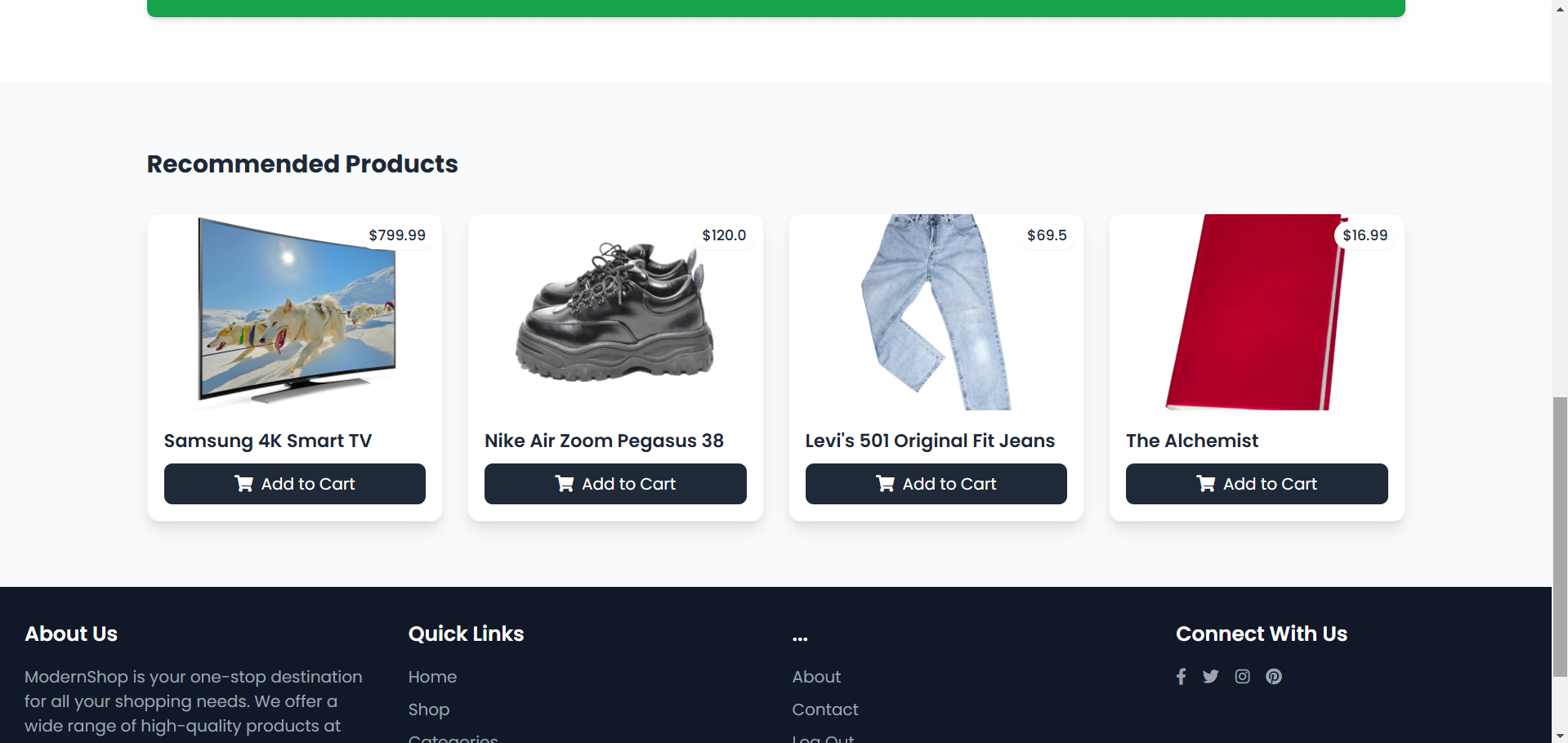
* + **Description**: Ensures that users can view their past orders.
  + **Inputs**: User account with order history.
  + **Expected Output**: Accurate display of past orders with details like items, dates, and status.

1. **Data Validation and Security**



* + **Description**: Tests input validation and security measures (e.g., Invalid coupon code prevention).
  + **Inputs**: Valid and malicious inputs.
  + **Expected Output**: Valid inputs are processed, and malicious inputs are rejected.

1. **Finalize Recommendation Engine**



* + **Description**: Verifies that the recommendation engine provides personalized product suggestions.
  + **Inputs**: User browsing and purchase history.
  + **Expected Output**: Relevant product recommendations are displayed.

**User Manual**

This user manual is designed to guide end-users through the features and functionalities of the program. Each section explains how to navigate and use the system effectively, ensuring a seamless experience. Screenshots can be added alongside the instructions to provide visual clarity.

**1. Getting Started**

1. **Accessing the Application**
   * Open your web browser and go to the application URL: Visit the address http://localhost:8000, or the link given during the live session.
   * You will be redirected to the home page where you can either choose to view products or select a login link to view more.
2. **User Registration**
   * Click on the "Sign Up" button located at the top-right corner of the homepage.
   * Fill in the registration form with your details (e.g., email, username, password).
   * Click "Register." After successful registration, you will be redirected to the login page.
3. **User Login**
   * Click on the "Login" button at the top-right corner.
   * Enter your registered email and password, then click "Log In."
   * Upon successful login, you will be redirected to your dashboard.

**2. Browsing Products**

1. **Viewing Products**
   * Navigate to the "Products" page using the menu bar.
   * Browse the list of available products, each displaying details such as name, price, and availability.
2. **Filtering Products**
   * More specifically, the filter options are located on the left of the page and allow selecting some criteria such as the category, price level, etc.
   * When selecting filters on the toolbar, the information displayed on the product list will change accordingly.
3. **Pagination**
   * If the products are split over several pages, there will be navigation controls located near the bottom of the page.

**3. Shopping Cart**

1. **Adding Items to Cart**
   * On the product page, click the "Add to Cart" button for the desired item.
   * A confirmation message will appear, and the cart count in the top-right corner will update.
2. **Viewing and Editing Cart**
   * Click on the "Cart" icon in the top-right corner to view your shopping cart.
   * To update the quantity of an item, adjust the number in the quantity field and click "Update."
   * To delete it, click the button labeled “Remove” close to the product information.
3. **Checking Out**
   * Once satisfied with your cart, click the "Pay with Cash on Delivery" button.
   * Ensure you had entered your shipping address.
   * Apply a coupon code if available, then click "Place Order."

**4. Order Management**

1. **Order Tracking**
   * After placing an order, go to the "Order History" page from your dashboard.
   * Select the order you want to track. The system will display the current status (e.g., "Processing," "Shipped," "Delivered").
2. **Order History**
   * You can see a record of all the orders that you have placed, the details being order date, products purchased and amount spend.

**5. Notifications and Emails**

1. **Notifications**
   * Stay updated on your order status and other activities via notifications displayed on your dashboard.
2. **Emails**
   * You will receive email updates for actions such as order confirmations, shipping updates, and promotional offers.

**6. Discounts and Coupons**

1. **Applying Coupons**
   * During checkout, enter a valid coupon code in the "Apply Coupon" field.
   * Click "Apply," and the discount will be reflected in the total price.
   * If the coupon is invalid, an error message will appear.

**7. User Profile Management**

1. **Updating Profile Information**
   * Go to "My Profile" from the dashboard.
   * Edit your personal details (e.g., name, email, password) and click "Save Changes."
2. **Changing Password**
   * In the "My Profile" section, navigate to the "Change Password" tab.
   * Enter your current password and the new password, then click "Update Password."

**8. Recommendation Engine**

1. **Personalized Recommendations**
   * On a product’s page, view product recommendations based on your browsing and purchase history.
   * Click on a recommended product to view its details or add it to your cart.

**9. Data Validation and Security**

1. **Secure Inputs**
   * The system ensures that all inputs are validated. For example, ensure that form fields are filled out correctly.
   * If an input is invalid, an error message will guide you to correct it.
2. **Account Security**
   * Your account is protected by secure authentication methods. Always use a strong password and avoid sharing your credentials.

**Setup**

1. **Clone the Repository**:
   * Use Git to clone the repository to your local machine and navigate to the project directory.
2. **Create a Virtual Environment (Optional but Recommended)**:
   * Create and activate a virtual environment to manage dependencies separately from your system Python installation.
3. **Install Dependencies**:
   * Install the required Python packages using pip install -r requirements.txt.

**Run Program**

1. **Apply Database Migrations**:
   * Run python manage.py migrate to set up your database schema.
2. **Create a Superuser (Optional)**:
   * If needed, create an admin superuser using python manage.py createsuperuser.
3. **Start the Development Server**:
   * Start the server with python manage.py runserver. By default, it runs on http://localhost:8000.
4. **Access the Application**:
   * Open http://localhost:8000 in your browser to view the application. Use http://localhost:8000/admin to access the admin interface.

**Features Implemented and Limitations**

The program successfully implements core features, including user authentication, database integration, and essential functionalities such as customized recommendations. These features work as intended and have been tested for basic usability and performance.

However, there are a few limitations. Some advanced features, such as advanced search options were not completed due to time constraints. Additionally, minor bugs were identified during peer review sessions, such as occasional crashes during bulk operations.

Feedback from peer reviews highlighted the need for better error handling, improved user interface design, and enhanced performance when handling large datasets.

**Plans for the Next Phase**

In the next phase, we plan to address the existing limitations by:

1. Fixing known bugs and improving error handling mechanisms.
2. Enhancing the user interface for a more seamless and intuitive experience.
3. Optimizing performance, particularly for specific scenarios, e.g., "large data processing".
4. Incorporating additional feedback from peer reviews.

These improvements aim to make the program more robust, user-friendly, and feature-complete in the next development cycle.

**Reflection**

This project has accomplished several key objectives, including the successful implementation of core features. These achievements demonstrate a solid foundation for the program and its potential for further development.

What went well was the seamless integration of the main features and the overall stability of the program during testing. It was during peer review sessions that group collaboration offered beneficial information for making modifications to some aspects of the project.

More time control could have been exercised in order to complete all the features which are planned for implementation. Likewise, further distinctions and a deeper analysis could improve the speed and efficiency of the tool as well. Therefore, to help avoid such issues in future they will have to be dealt with in order to enhance the progress of the development process.

**Member Contribution Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Member Name** | **Contribution Description** | **Overall Contribution (%)** | **Note (if applicable)** |
| Harini | Enhanced the authentication module and addressed security bugs. | 16% | Conducted a second round of security reviews. |
| Osama | Updated the database schema to support new features and optimized queries further. | 15% | Performed database migration testing. |
| Nirupama | Redesigned parts of the user interface based on peer feedback and improved accessibility. | 13% | Led UI/UX feedback sessions. |
| Niharika | Integrated additional third-party APIs and worked on payment error handling. | 14% | Updated API documentation. |
| Jaswanth | Developed advanced application logic for new features and improved modularity. | 15% | Oversaw the integration of new components. |
| Rajsekhar | Enhanced debugging tools, fixed known bugs, and improved error reporting. | 12% | Monitored and resolved critical errors. |
| Triveni | Expanded test coverage to include new features and edge cases. | 10% | Led automated testing efforts. |
| Sharanya | Updated the project report, including feedback and future plans. | 5% | Finalized report formatting and edits. |