Fresh Cart

Authors :

- Ahmed Hamdy
- Zyad Hesham
- Karim Adel
- Omar Nashaat
- Mohamed Atef
- Gamal Ahmed

Date : 5/8/2024

Fresh Cart - Table of Contents

1. **Introduction**
2. **Goals**
3. **Requirements**
   - 3.1 Functional Requirements
   - 3.2 Non-Functional Requirements
4. **Architectural Design**
5. **Detailed Design (UML Diagrams)**
   - 5.1 Activity Diagram
   - 5.2 Class Diagram
   - 5.3 Object Constraint Language (OCL) Diagrams (Optional)
   - 5.4 Sequence Diagram
   - 5.5 Entity-Relationship Diagram (ERD)
6. **User Interface Design**
7. **Security Considerations**
8. **Testing Strategy**

Introduction:

This document outlines the Software Design Document (SDD) for Fresh Cart, an e-commerce website enabling users to purchase various products, including clothes, electronics, and food

Our goal is :

1. Develop a user-friendly and secure e-commerce platform.
2. Offer a wide range of products across different categories.
3. Provide a seamless shopping experience for users.
4. Facilitate secure online transactions.

Requirements

Functional Requirements

- **User Management:**
  - User registration and login.
  - User profile management (address, payment information).
- **Product Management:**
  - Add, edit, and delete products.
  - Manage product categories and subcategories.
  - Upload product images and descriptions.
- **Inventory Management:**
  - Track product stock levels.
  - Update inventory upon purchase and returns.
- **Shopping Cart:**
  - Add products to cart.
  - Modify cart quantities.
  - View and manage cart contents.
- **Checkout Process:**
  - Secure payment processing (credit card, etc.).
  - Order confirmation with order details.
  - Shipping options and cost calculation.
- **Order Management:**
  - Track order status (pending, shipped, delivered).
  - User order history and management.
  - Order cancellation (within a specific timeframe).
- **Search and Filter:**
  - Search for products by keyword or category.
  - Filter products by price, brand, and other attributes.
- **Reviews and Ratings:**
  - Users can leave reviews and ratings on products.
  - Admin can manage and moderate reviews.

Non-Functional Requirements

- **Performance:**
  - The website should load quickly and handle concurrent user requests efficiently.
- **Security:**
  - Implement secure user authentication and authorization.
  - Protect sensitive information (payment details) with encryption.
  - Prevent unauthorized access and data breaches.
- **Scalability:**
  - The architecture should accommodate future growth in products, users, and transactions.
- **Availability:**
  - The website should be highly available with minimal downtime.
- **Usability:**
  - Provide a user-friendly and intuitive interface for all types of users.
  - Make the website accessible for users with disabilities.

Architectural Design

Fresh Cart will utilize a three-tier architecture:

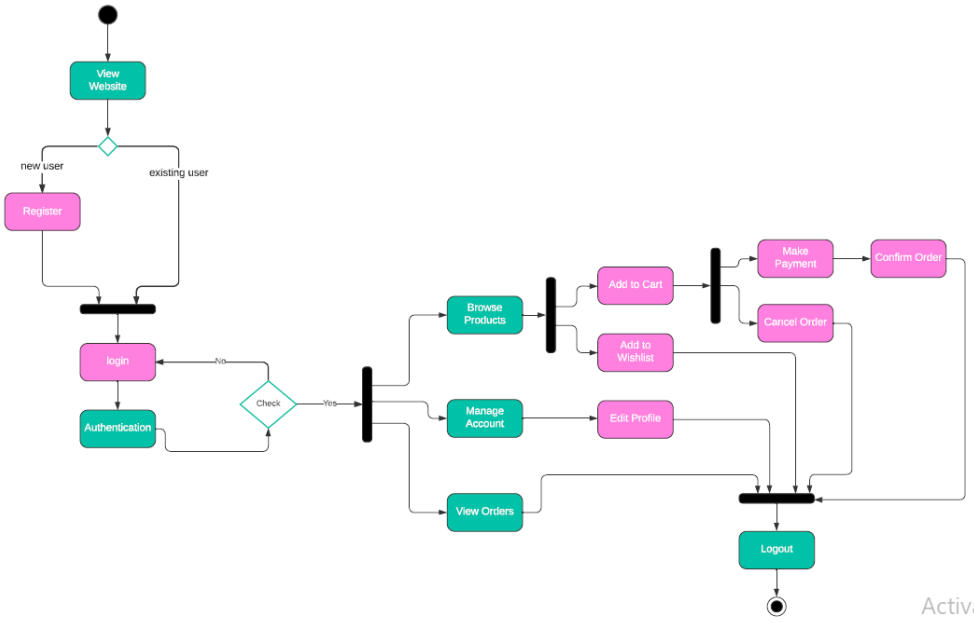
- **Presentation Layer:** The user interface (UI) built with HTML, CSS, and JavaScript for user interaction.
- **Business Logic Layer:** The server-side application logic handling tasks like product management, shopping cart functionality, and order processing. This layer may use a framework like Python's Django or Node.js' Express.
- **Data Access Layer:** The database layer responsible for storing and retrieving data related to users, products, orders, etc. This could be a relational database like MySQL or PostgreSQL.

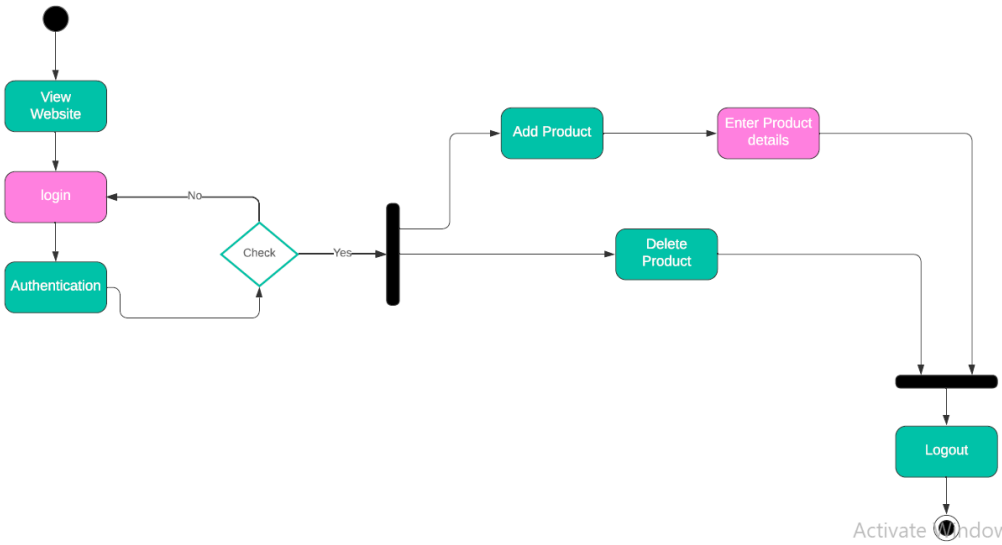
UML Diagrams:

1- Activity Diagram: This will depict the overall workflow of a user placing an order, including browsing products, adding to cart, checkout, and payment processing

Admin Activity v.1.pdf

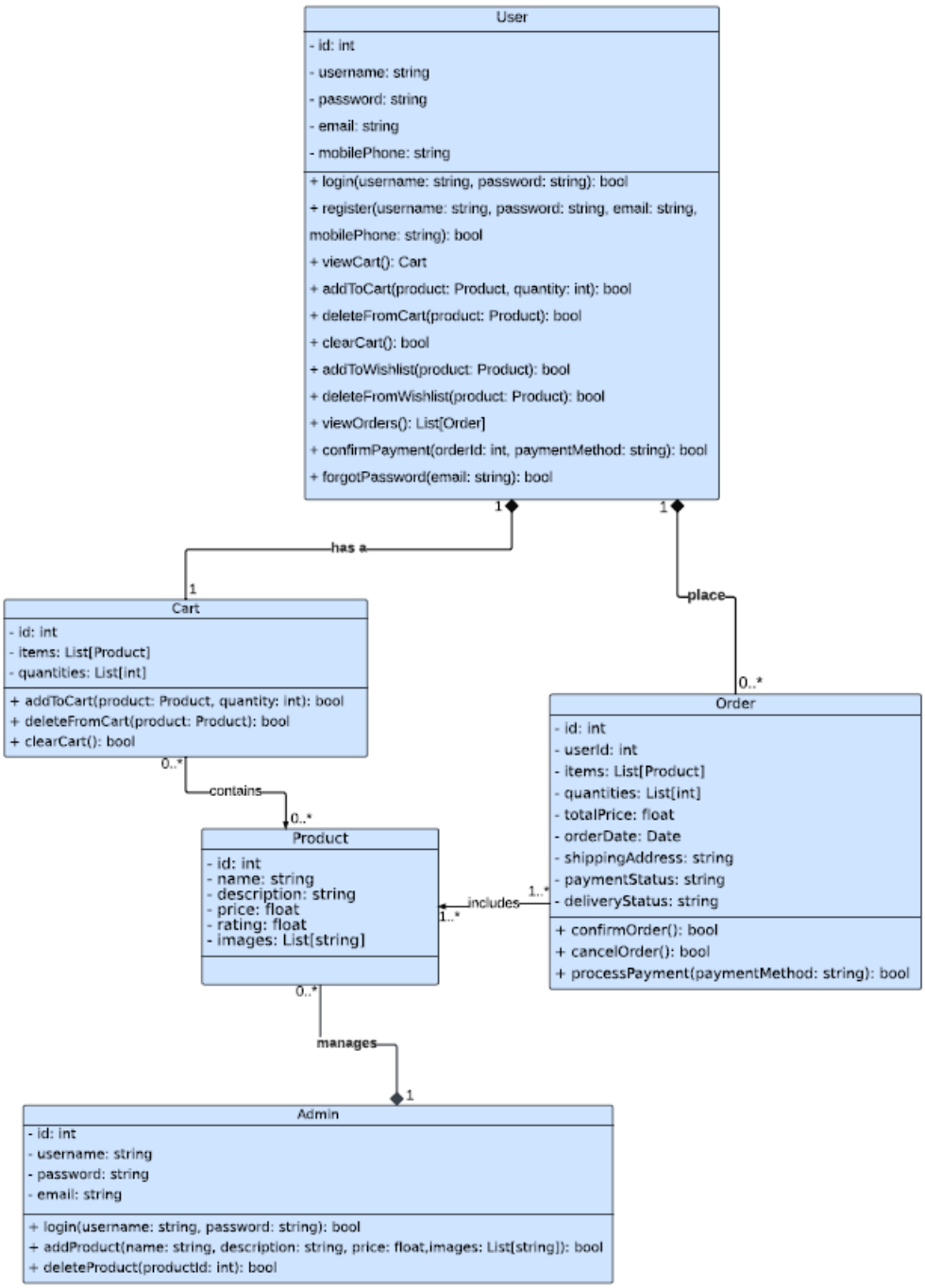
User Activity v.1.pdf





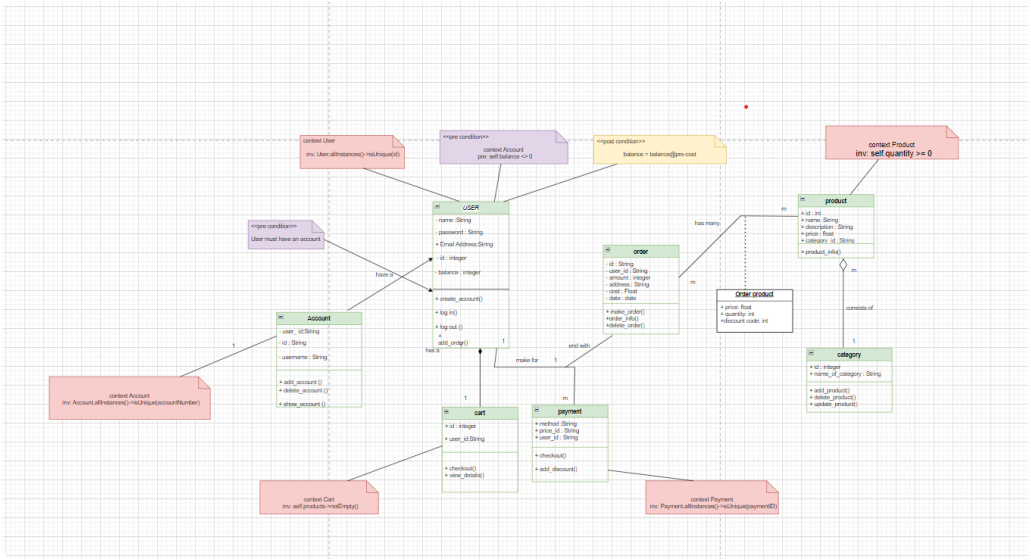
2- **Class Diagram:** This will define the classes involved in the system, such as User, Product, Order, Shopping Cart, etc., with their attributes and methods.

e commerce class diagram.pdf

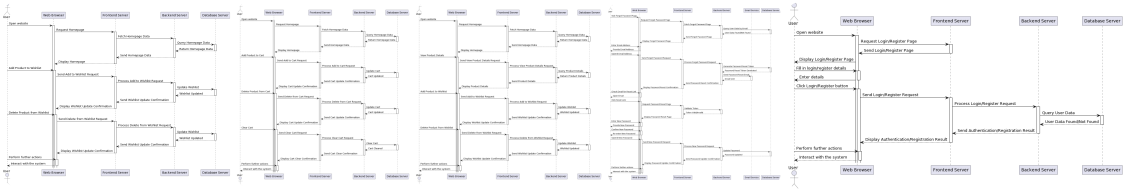


3- Object Constraint Language (OCL) Diagrams: These can be used to specify constraints on the system's data model, such as ensuring a user has a valid email address or a product has a positive quantity in stock.

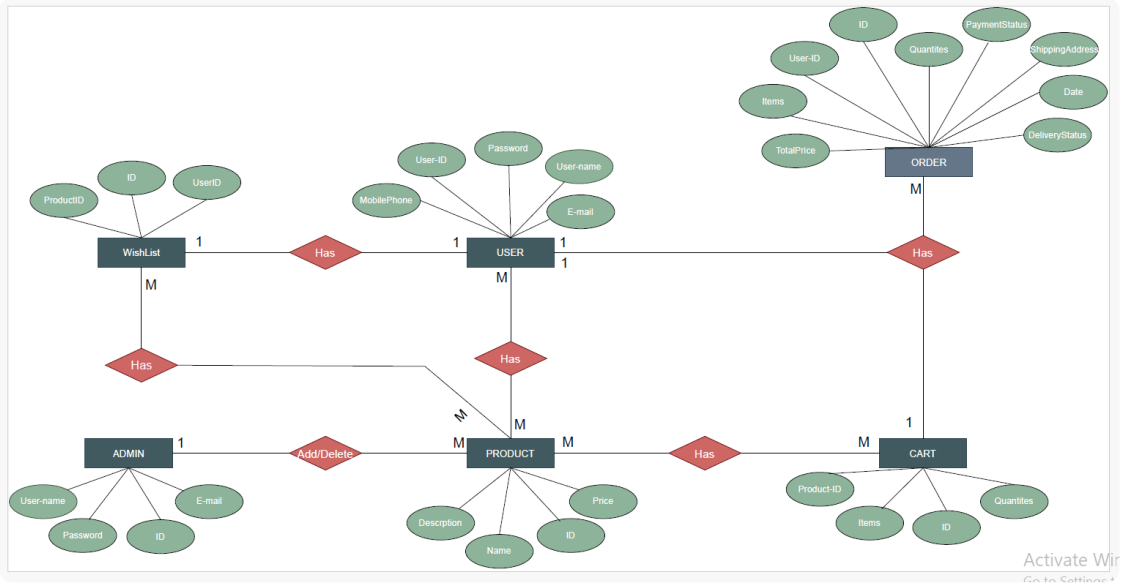
Ocl.drawio



3- Sequence Diagram: This will illustrate the interaction between objects during a specific scenario, such as a user adding an item to the shopping cart.

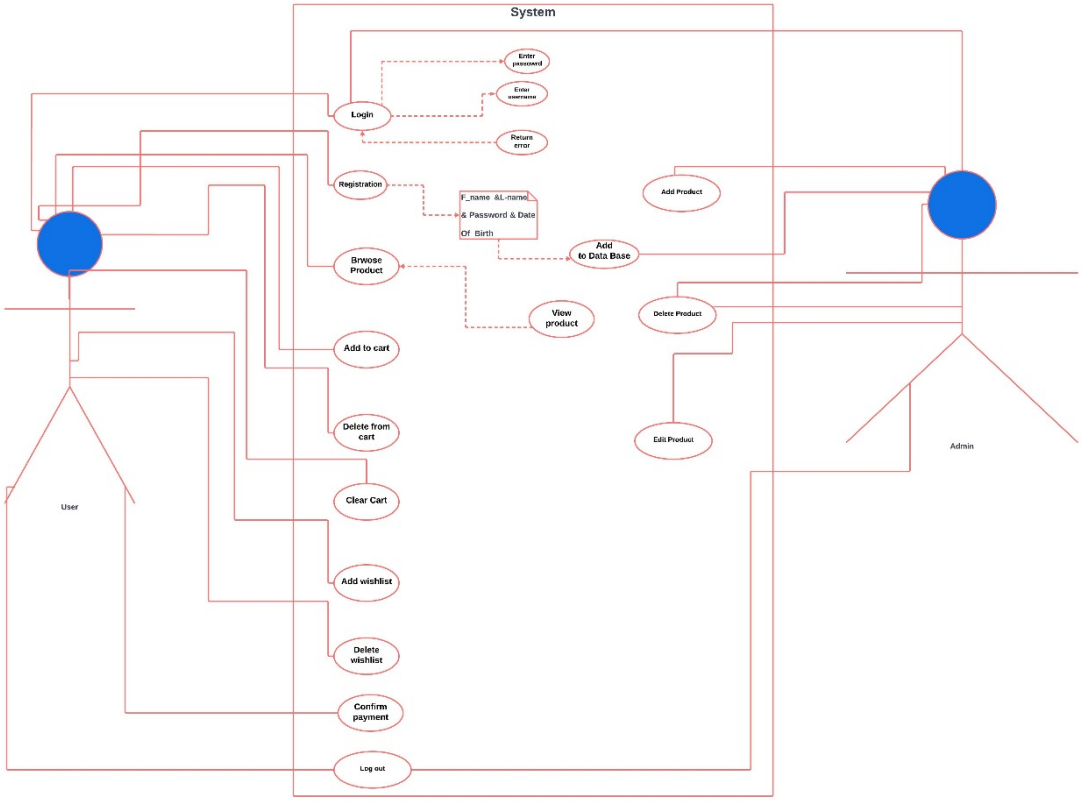


4- **Entity-Relationship Diagram (ERD):** This will represent the relationships between entities in the database, like users, products, orders, and how they are linked.



E-commerce.drawio (2).pdf

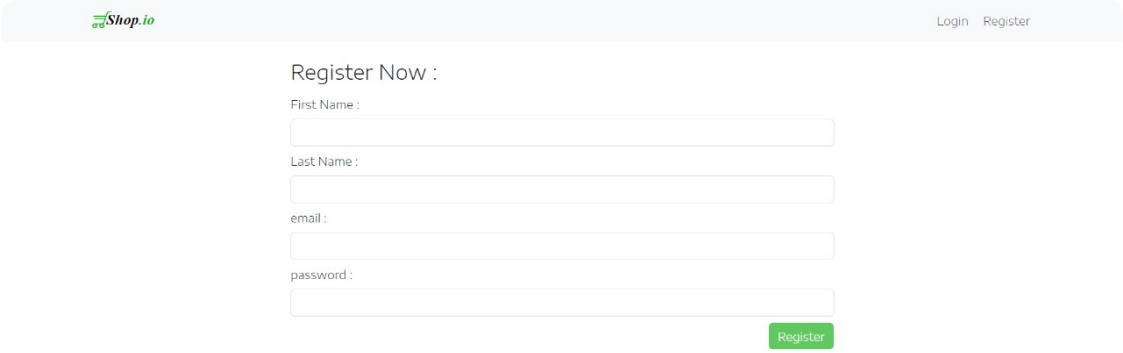
5- Use-case diagram

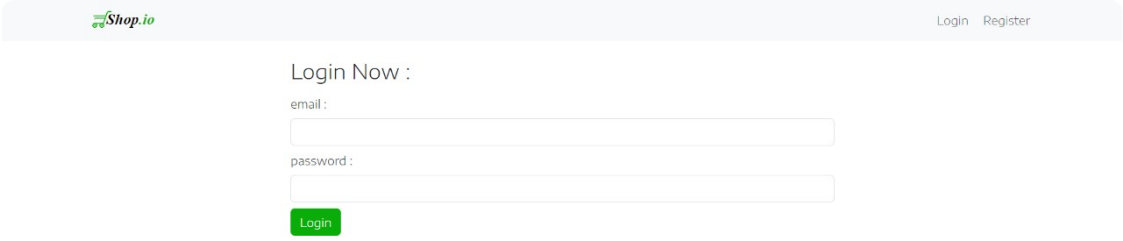


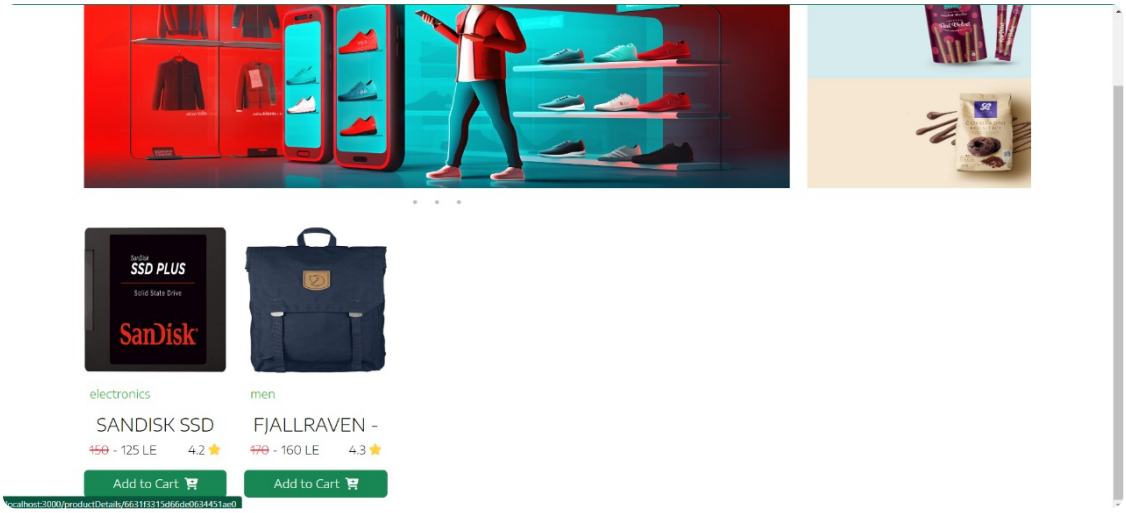
User Interface Design

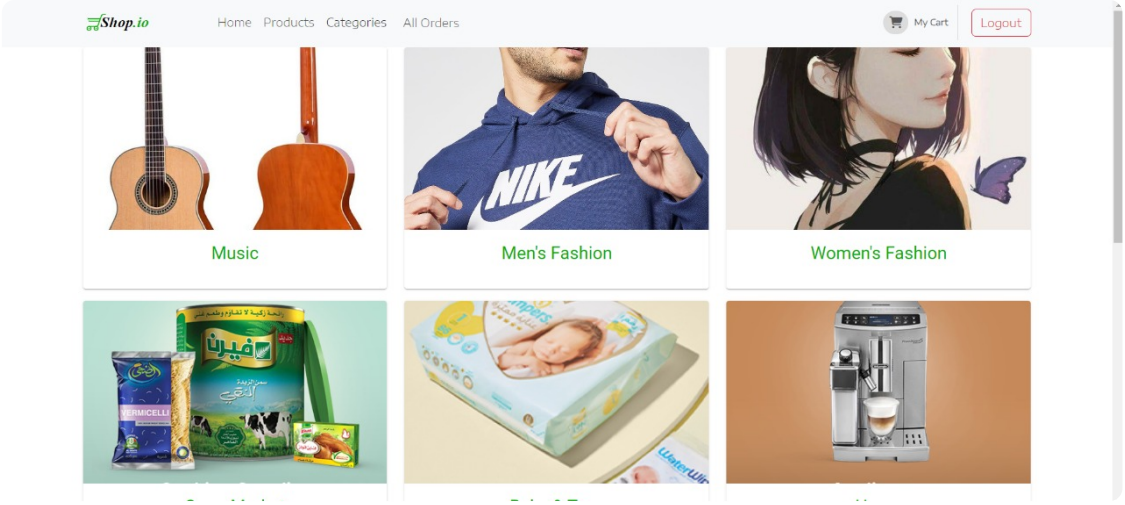
The UI should be:

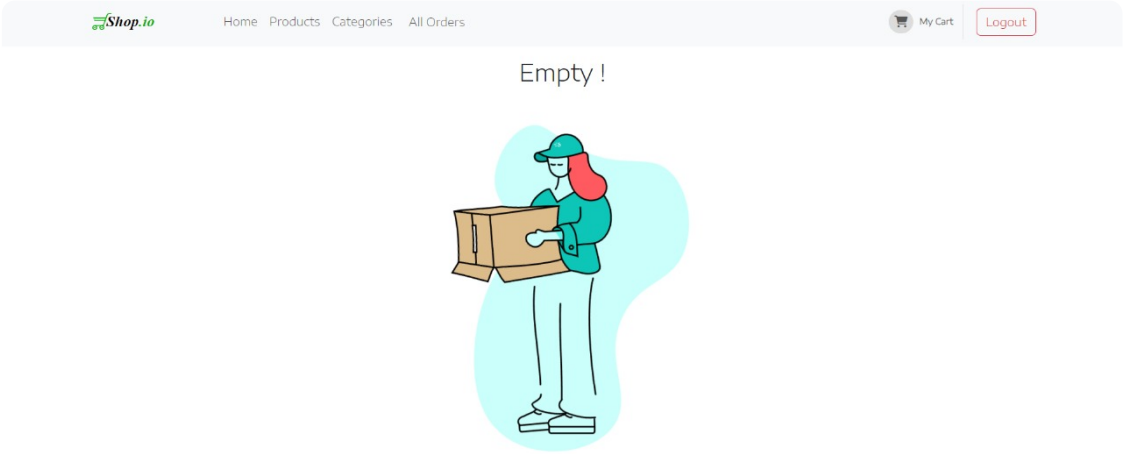
- Clean and uncluttered with a focus on product visibility.
- Easy to navigate with clear menus and search options.
- Responsive for optimal viewing on various devices (desktop, mobile, tablet).
- Accessible with features for users with disabilities.






SSD PLUS
SanDisk
electronics
SANDISK SSD
150 - 125 LE
4.2
Add to Cart
men
FJALLRAVEN -
170 - 160 LE
4.3
Add to Cart


Shop.io
Home
Products
Categories
All Orders
My Cart
Logout
Music
Men's Fashion
Women's Fashion


Shop.io
Home
Products
Categories
All Orders
My Cart
Logout
Empty !

Security Considerations :

- Secure user authentication (hashed passwords) and authorization.
- Secure communication using HTTPS with encryption.
- Implement security measures against common threats like SQL injection and cross-site scripting (XSS).
- Regularly update software and libraries to address vulnerabilities.

Testing Strategy

- **Unit Testing:** Test individual functionalities of the application logic.
- **Integration Testing:** Ensure different modules of the application work together seamlessly.
- **Functional Testing:** Verify that all features of the website meet the specified requirements.
- **Usability Testing:** Evaluate user experience with real users to identify any usability issues.
- **Security Testing:** Conduct penetration testing to identify and address potential security vulnerabilities.
- **Performance Testing:** Analyze website performance under load to ensure responsiveness and scalability.
- **Compatibility Testing:** Test the website across different browsers, operating systems, and devices.
- **Regression Testing:** Re-run critical tests after code changes to ensure existing functionalities remain intact.

THANK YOU