# **ShopAllMarket E-commerce Website**

# 

# *A Comprehensive Overview of Features, Structure, Code Implementation, and Development*

# 

# Authors: Awa Afo & Hassan Niass Date: November 2024

# 

# Website Link: https://awaafo.github.io/ShopAllMarket/

# 

# Project Description: ShopAllMarket is an e-commerce platform offering a smooth shopping experience with product browsing, secure user authentication, and an evolving product catalog. Additional features are in development.

**Table of Content**

| **Section** | **Subsections** |
| --- | --- |
| **Project Overview** | - Introduction  - Key Features  - Current Limitations |
| **Technologies Used** | - Frontend Stack  - Backend Stack |
| **Features** | - User Authentication  - Authentication Flow  - Endpoints  - Product Catalog  - Endpoints  - Shopping Cart  - Current Status  - Unimplemented Features |
| **Database Design** | - Schema Design  - Users  - Products  - Carts  - Orders |
| **To-Do List (Unimplemented Features)** | - Payment Gateway Integration  - Full Cart Management  - Admin Panel |
| **Known Issues** | - Cart Management Limitations  - Missing Admin Panel  - Payment Gateway Integration |
| **Future Improvements** | - Payment Gateway Integration  - Admin Panel Development  - Enhanced Cart and Checkout Flow  - Mobile Application Expansion |
| **Screenshots and Descriptions** | - Homepage  - Product Catalog  - Product Details Page  - User Authentication Pages  - Shopping Cart Page |
| **Appendices** | - Key Code Snippets  - References |
| **Conclusion** |  |
| **Glossary** |  |

## **Project Overview**

**ShopAllMarket** is an e-commerce platform designed to provide users with a seamless online shopping experience. The platform enables customers to browse a wide range of products, register and log in to manage their accounts, and interact with the product catalog. However, the project is still under development, and some features are yet to be fully implemented.

### **Key Features**

1. **User Authentication**: Secure login and registration for users to create and manage accounts.
2. **Product Catalog**: Users can browse a wide variety of products categorized for easy navigation.
3. **Shopping Cart**: Users can add items to their cart, view product details, and proceed to checkout.

### **Unimplemented Features**

* **Payment Gateway Integration**: Integration with a payment gateway (e.g., Stripe or PayPal) is pending.
* **View Cart Functionality**: Users can add products to the cart, but viewing or managing the cart is not yet functional.
* **Admin Panel**: A dashboard for managing products, orders, and user data is not yet developed.

## **2. Folder structure**

Root Directory: ShopAllMarket/

| **Folder/File** | **Description** |
| --- | --- |
| **ShopAllMarket/** | Root directory of the project. |
| **css/** | Folder for CSS files. |
| ├── bootstrap.css | Bootstrap framework stylesheet. |
| ├── flexslider.css | FlexSlider plugin stylesheet. |
| ├── jquery-ui.css | jQuery UI framework stylesheet. |
| ├── pignose.layerslider.css | Pignose Layer Slider plugin stylesheet. |
| └── style.css | Custom stylesheet for the website. |
| **fonts/** | Folder for font files. |
| ├── glyphicons-halflings-regular.eot | Glyphicons font file (EOT format). |
| ├── glyphicons-halflings-regular.svg | Glyphicons font file (SVG format). |
| ├── glyphicons-halflings-regular.ttf | Glyphicons font file (TTF format). |
| ├── glyphicons-halflings-regular.woff | Glyphicons font file (WOFF format). |
| └── glyphicons-halflings-regular.woff2 | Glyphicons font file (WOFF2 format). |
| **images/** | Folder for image assets used on the website. |
| └── [image files] | Placeholder for actual image files. |
| **js/** | Folder for JavaScript files. |
| ├── bootstrap-3.1.1.min.js | Bootstrap JavaScript library. |
| ├── easyResponsiveTabs.js | Plugin for responsive tabs. |
| ├── imagezoom.js | JavaScript for image zoom functionality. |
| ├── jquery-2.1.4.min.js | Minified jQuery library. |
| ├── jquery-ui.js | jQuery UI JavaScript library. |
| ├── jquery.easing.min.js | JavaScript for easing effects. |
| └── jquery.flexslider.js | FlexSlider plugin JavaScript. |
| **HTML Files** | Static HTML files for the website pages. |
| ├── index.html | Homepage of the website. |
| ├── contact.html | Contact page. |
| ├── electronics.html | Electronics category page. |
| ├── mens.html | Men's products page. |
| ├── womens.html | Women's products page. |
| └── single.html | Product details page. |

## **3.Features**

### **3.1 User Authentication**

* Users can register for a new account and log in to the system using secure credentials.
* JWT is used to generate tokens for authenticating user requests. Sessions are maintained until the token expires or the user logs out.

**Authentication Flow**:

1. Users register with an email and password.
2. Upon successful registration, a JWT token is issued.
3. Users log in with their credentials to receive a token for authenticated requests.

**Endpoints**:

* POST /auth/register: Registers a new user.
* POST /auth/login: Logs in an existing user.

### **3.2 Product Catalog**

* Displays all available products for users to browse. Products are categorized for easy navigation (e.g., electronics, clothing).

**Endpoints**:

* GET /products: Fetches all products.
* GET /products/:id: Fetches details of a specific product by its ID.

**3.3 Shopping Cart**

* Users can add products to their shopping cart. Current limitations include the inability to view or manage cart items.

**Unimplemented Features**:

* **View Cart**: Users cannot view cart items or edit quantities.
* **Cart Management**: Users cannot remove items or proceed to checkout.

## **4. Database Design**

The database will be structured to manage products, users, orders, and carts efficiently.

### **Collections:**

1. **Users**:
   * **Schema**: username, email, password, role, orderHistory
   * **Purpose**: Store user details and their order history.
2. **Products**:
   * **Schema**: name, description, price, category, imageUrl, stockQuantity
   * **Purpose**: Store product details, which are displayed in the product catalog.
3. **Carts**:
   * **Schema**: userId, productId, quantity, totalPrice
   * **Purpose**: Store items added to a user's cart. It allows modifications (e.g., adding/removing products).
4. **Orders**:
   * **Schema**: userId, products, totalPrice, status, orderDate
   * **Purpose**: Store orders after the checkout process. Each order references products and their quantities.

## **5.To-Do List**

### **Unimplemented Features**

#### **5.1 Payment Gateway Integration**

* A third-party payment gateway (e.g., Stripe or PayPal) will be integrated for secure transactions.

**Steps to Implement**:

1. Set up a payment service account.
2. Integrate the payment service API with the backend.
3. Design a frontend checkout form.

### **5.2 View Cart Functionality**

* Enable users to view and manage their shopping carts.

**Steps to Implement**:

1. Create a Cart component on the frontend.
2. Add functionality to update quantities and remove items.
3. Store cart data in the user’s session or database.

### **5.3 Admin Panel**

* Develop a dashboard for product and order management.

**Steps to Implement**:

1. Design an admin interface.
2. Implement role-based access control.
3. Add features for managing products and users.

## **6. Known Issues**

* **Cart Management**: The cart currently only allows adding items but lacks the ability to view or manage them.
* **Admin Panel**: Admin functionalities like product management and order tracking are not yet implemented.
* **Payment Gateway**: The payment gateway has not been integrated, so users are unable to make purchases.

## **7. Future Improvements**

1. **Payment Gateway Integration**: Secure payment processing.
2. **Admin Panel**: Enhanced product and order management.
3. **Cart and Checkout Enhancements**: Improved cart management and checkout flow.
4. **Mobile Application**: Develop a mobile app for broader accessibility.

### **8. Screenshots and Descriptions**

This section provides visual examples of the **ShopAllMarket** website, showcasing its key features and functionality. Each screenshot is accompanied by a brief description to help users and developers understand the design and purpose of each page.

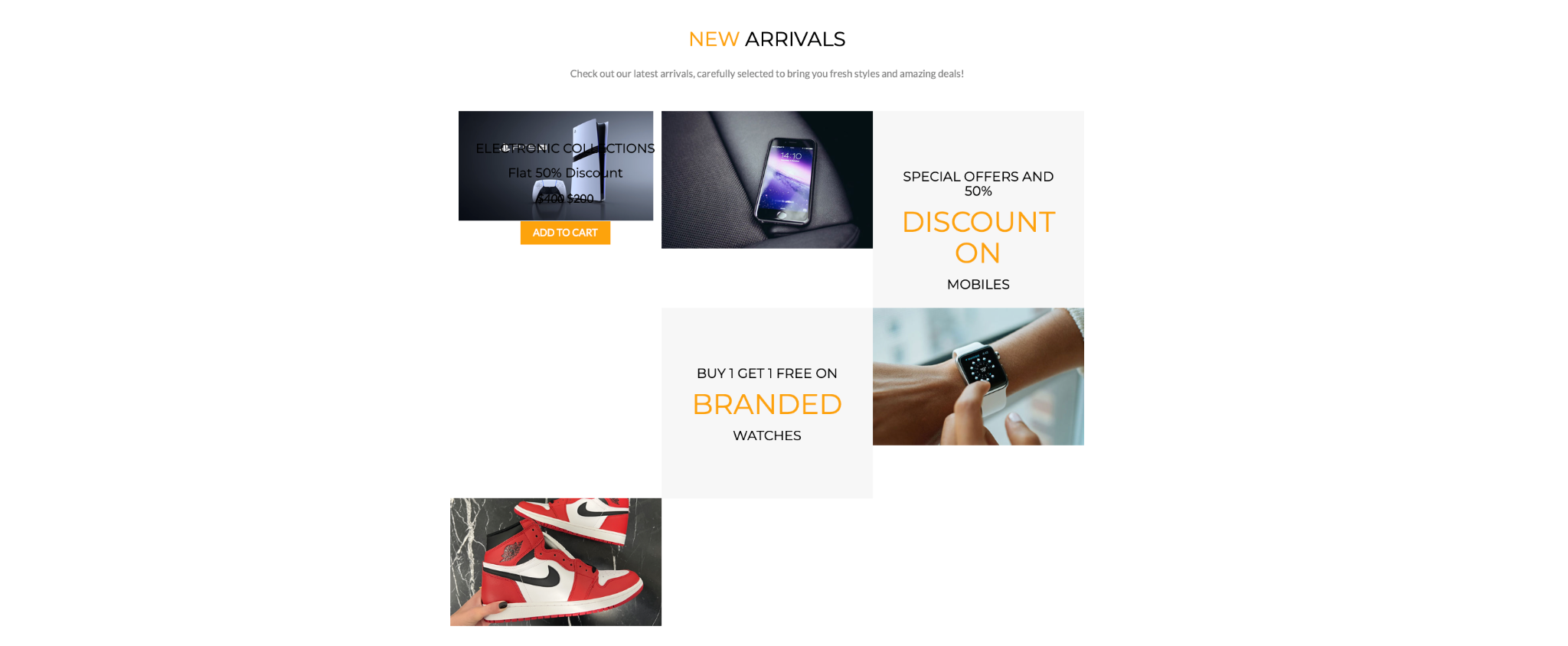
#### **Homepage**

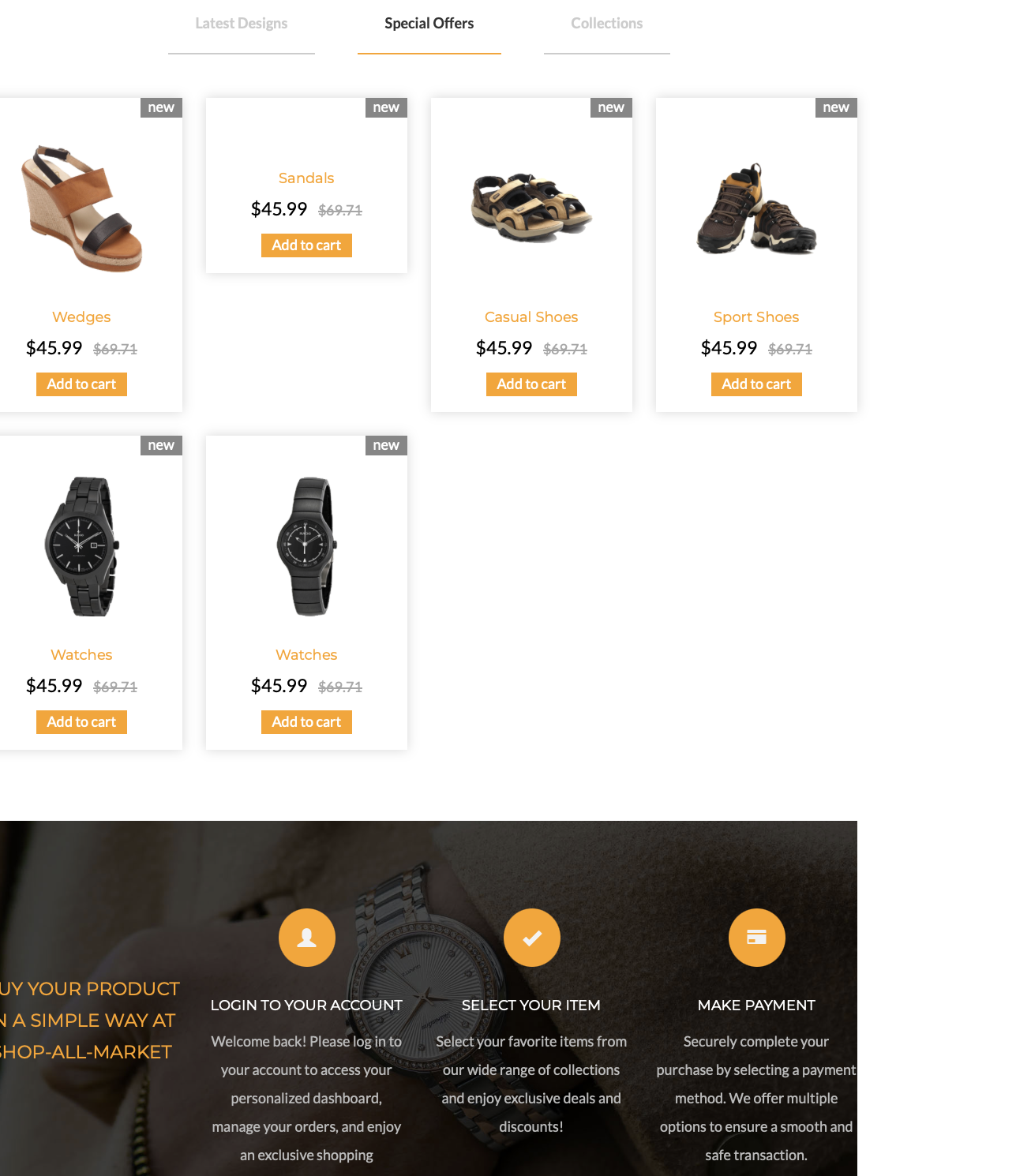
**Description**:  
The homepage introduces users to the ShopAllMarket platform. It features a prominent banner, quick links to popular categories, and a search bar for easy navigation. Designed with a modern, responsive layout, the homepage ensures an optimal shopping experience on all devices.

**Screenshot**:  
**

#### **Product Catalog**

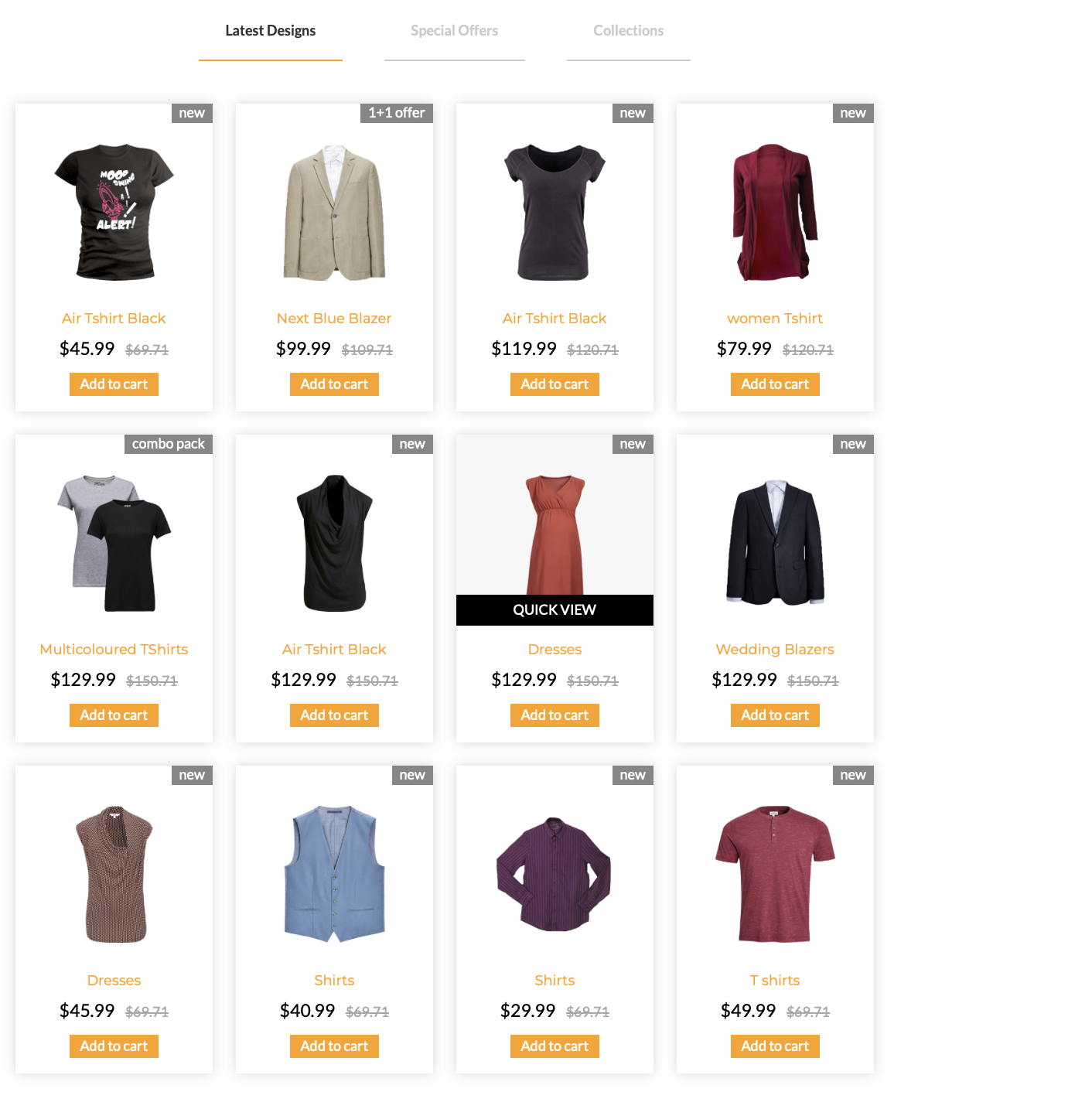
**Description**:  
This page displays the full product catalog with categories such as electronics, clothing, and home appliances. Users can browse products, view thumbnails, and sort or filter items by price, category, or availability.

**Screenshot:**



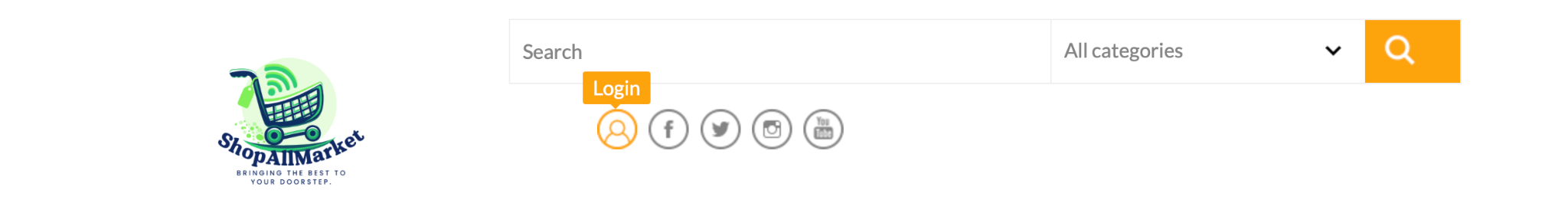
#### **Product Details Page**

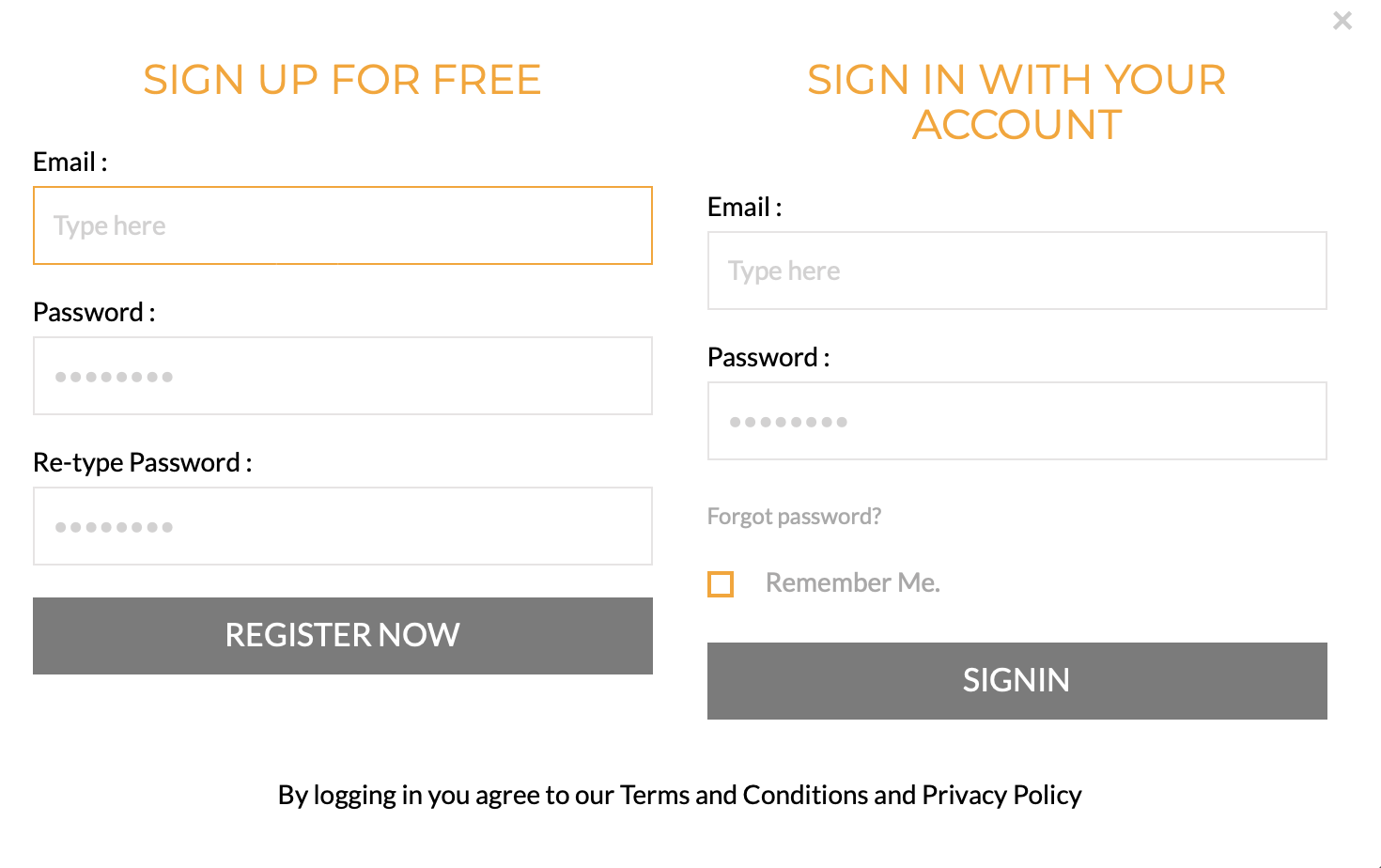
**Description**:  
The product details page provides a detailed view of an individual product, including its description, price, availability, and a high-resolution image. Users can add items to their shopping cart directly from this page.

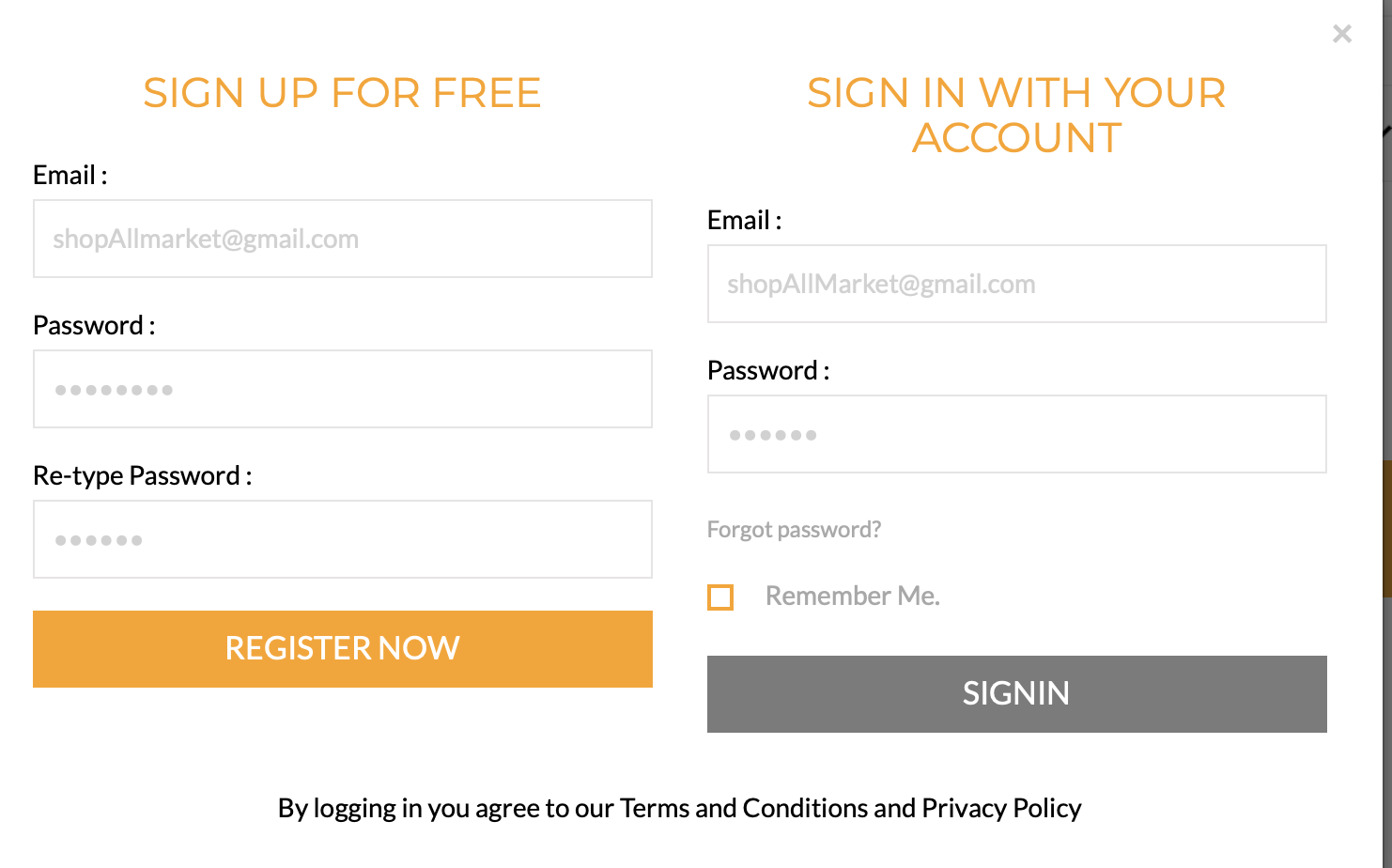
**Screenshot**:  


#### **User Authentication (Login and Registration)**

**Description**:  
The login and registration pages allow users to securely access their accounts. Features include form validation, JWT token generation for session management, and user feedback for errors or successful authentication.

**Screenshot**:  
**





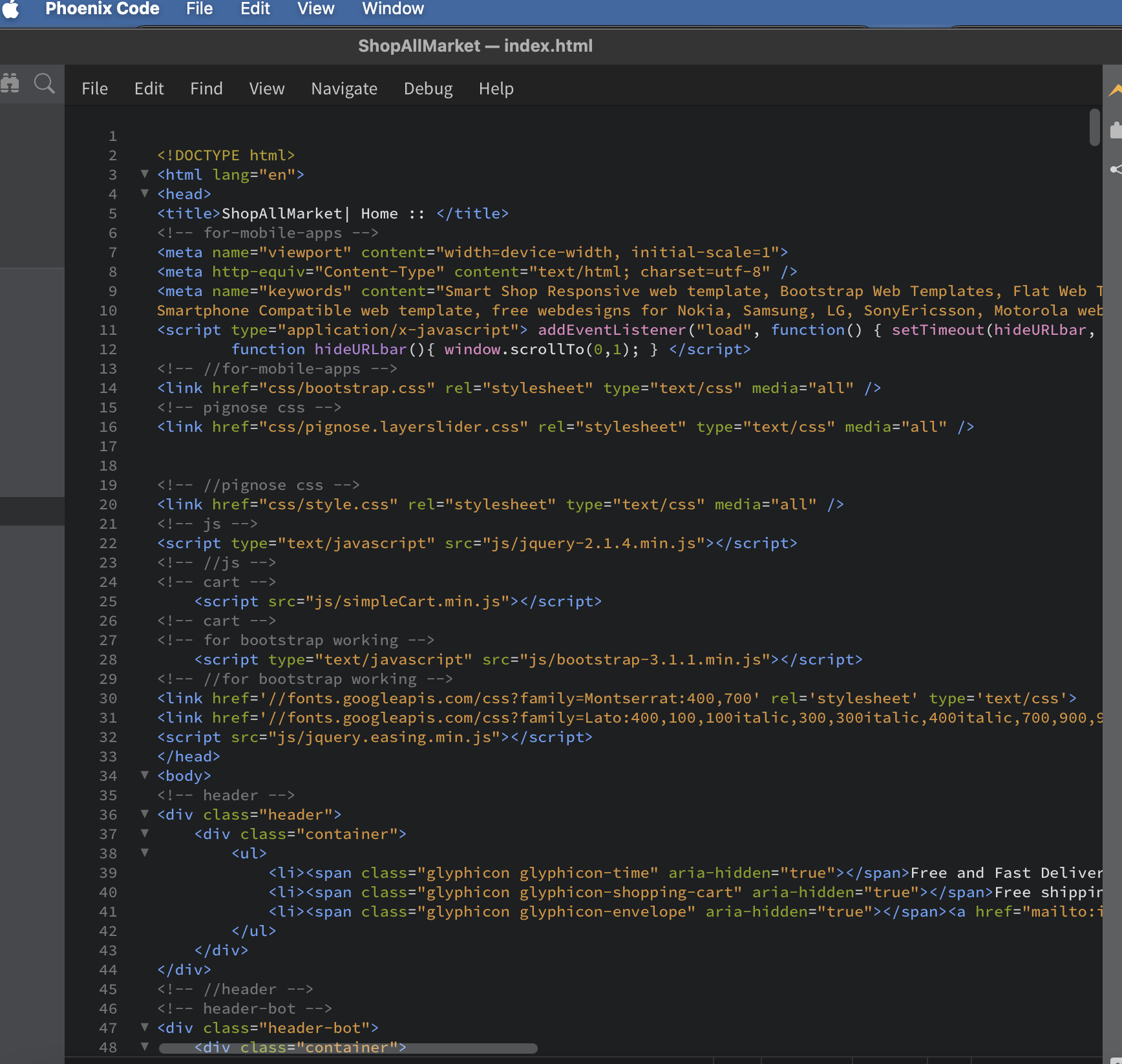
### **9. Appendices**

#### **Code & References**

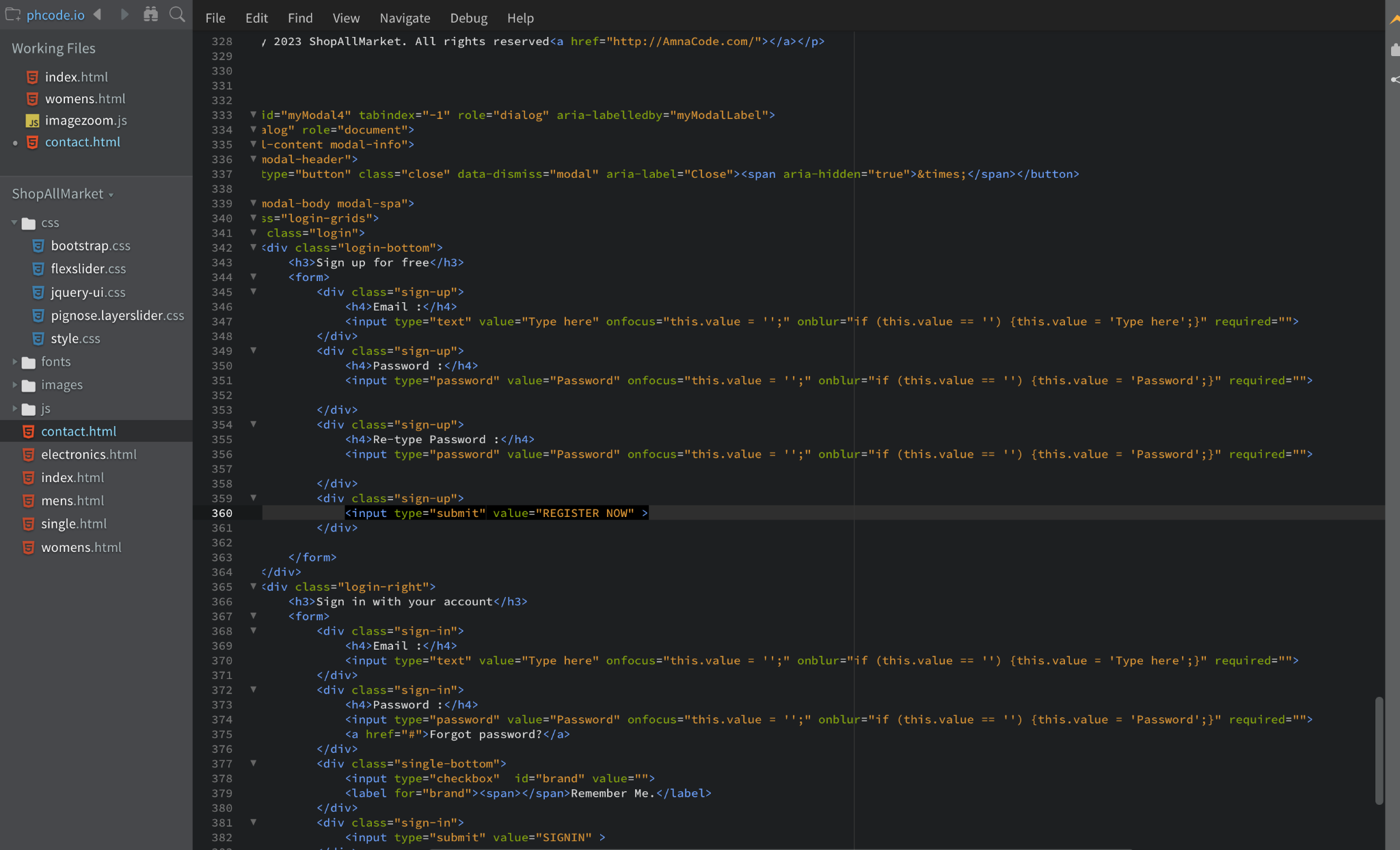
This section provides supplementary materials, including relevant code snippets and references for the ShopAllMarket e-commerce platform.

#### **Key Code Snippets**

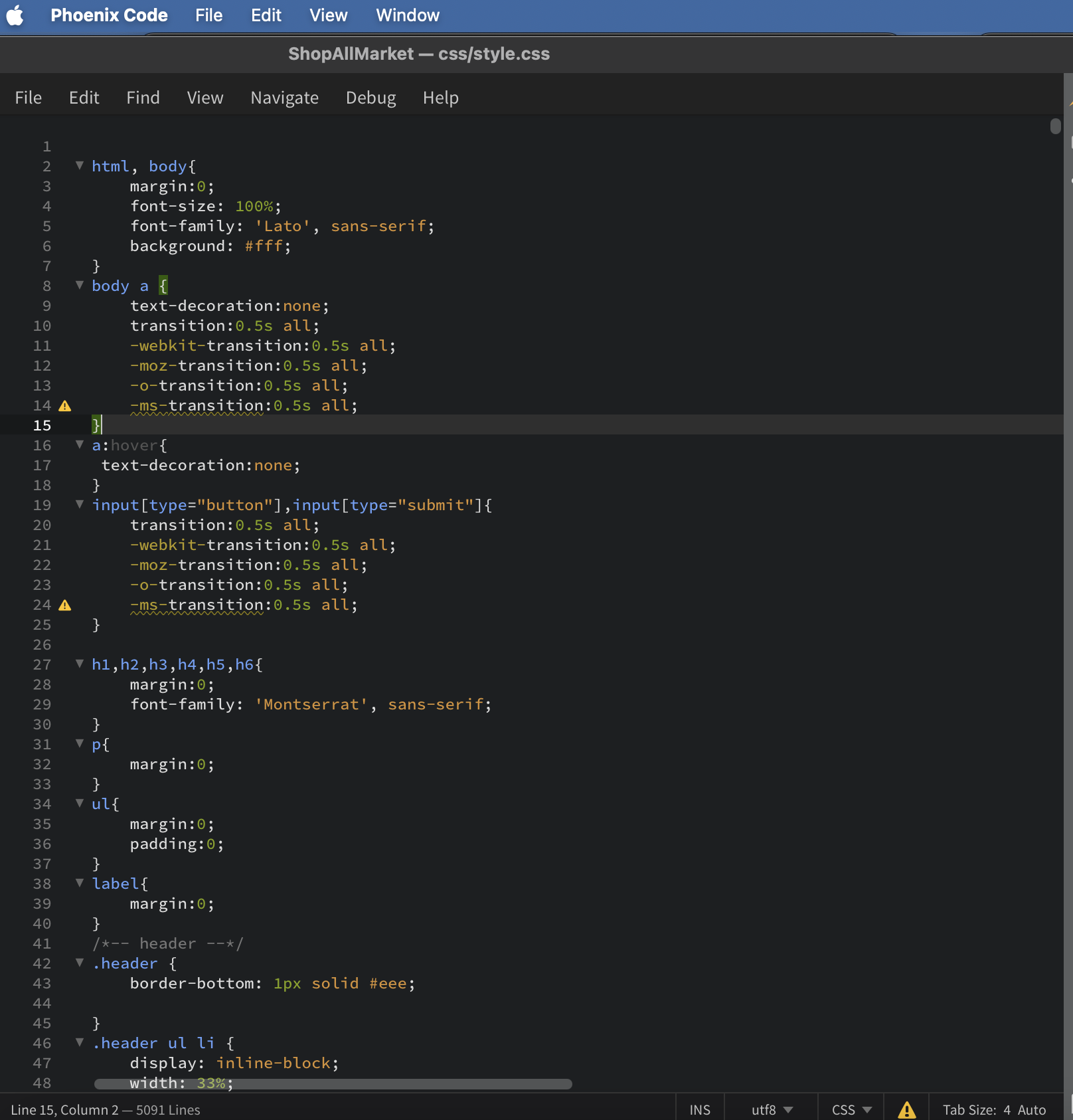
1. **HTML Structure**The homepage (index.html) utilizes semantic HTML to ensure accessibility and SEO optimization. Below is a snippet of the homepage structure:



**Register /Login code**



1. **CSS Styling**The styles.css file centralizes styling for consistent design. Here’s an example:



## **10. Conclusion**

The **ShopAllMarket** project is still under development, with several features missing or in progress. Currently, the user authentication, product catalog, and basic shopping cart functionality are in place. The next steps include integrating the payment gateway, improving the cart management system, and building an admin panel for product and order management.

#### **11. Glossary**

* **ShopAllMarket**: An e-commerce platform designed for a seamless shopping experience, featuring a product catalog and user authentication.
* **HTML (Hypertext Markup Language)**: The foundational markup language for structuring web pages.
* **CSS (Cascading Style Sheets)**: A language used to style the HTML structure, including layouts, colors, and fonts.
* **Node.js**: A runtime for executing JavaScript on the server side.
* **Express.js**: A Node.js framework used to create API endpoints and handle backend logic.
* **JWT (JSON Web Token)**: A token standard used for secure user authentication and session management.
* **MongoDB**: A NoSQL database used to store data like user accounts and product details.
* **Responsive Design**: An approach to web design that ensures the platform is usable on various devices, including desktops and mobile phones.
* **Semantic HTML**: HTML that conveys clear meaning and enhances accessibility and SEO, using elements like <header>, <main>, and <footer>.