

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://awaaf.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	<ul style="list-style-type: none">- Introduction- Key Features- Current Limitations
Technologies Used	<ul style="list-style-type: none">- Frontend Stack- Backend Stack
Features	<ul style="list-style-type: none">- User Authentication- Authentication Flow- Endpoints- Product Catalog- Endpoints- Shopping Cart- Current Status- Unimplemented Features
Database Design	<ul style="list-style-type: none">- Schema Design- Users- Products- Carts- Orders
To-Do List (Unimplemented Features)	<ul style="list-style-type: none">- Payment Gateway Integration- Full Cart Management- Admin Panel
Known Issues	<ul style="list-style-type: none">- Cart Management Limitations- Missing Admin Panel- Payment Gateway Integration
Future Improvements	<ul style="list-style-type: none">- Payment Gateway Integration- Admin Panel Development- Enhanced Cart and Checkout Flow- Mobile Application Expansion
Screenshots and Descriptions	<ul style="list-style-type: none">- Homepage- Product Catalog- Product Details Page- User Authentication Pages- Shopping Cart Page
Appendices	<ul style="list-style-type: none">- Key Code Snippets- References
Conclusion	
Glossary	

1. 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.

<code>└── style.css</code>	Custom stylesheet for the website.
<code>fonts/</code>	Folder for font files.
<code> └──</code> <code> glyphicons-halflings-regular.eot</code>	Glyphicon font file (EOT format).
<code> └──</code> <code> glyphicons-halflings-regular.svg</code>	Glyphicon font file (SVG format).
<code> └──</code> <code> glyphicons-halflings-regular.ttf</code>	Glyphicon font file (TTF format).
<code> └──</code> <code> glyphicons-halflings-regular.woff</code>	Glyphicon font file (WOFF format).
<code> └──</code> <code> glyphicons-halflings-regular.woff2</code>	Glyphicon font file (WOFF2 format).
<code>images/</code>	Folder for image assets used on the website.
<code> └── [image files]</code>	Placeholder for actual image files.
<code>js/</code>	Folder for JavaScript files.
<code> └── bootstrap-3.1.1.min.js</code>	Bootstrap JavaScript library.
<code> └── easyResponsiveTabs.js</code>	Plugin for responsive tabs.
<code> └── imagezoom.js</code>	JavaScript for image zoom functionality.
<code> └── jquery-2.1.4.min.js</code>	Minified jQuery library.
<code> └── jquery-ui.js</code>	jQuery UI JavaScript library.
<code> └── jquery.easing.min.js</code>	JavaScript for easing effects.
<code> └── jquery.flexslider.js</code>	FlexSlider plugin JavaScript.
HTML Files	Static HTML files for the website pages.
<code> └── index.html</code>	Homepage of the website.
<code> └── contact.html</code>	Contact page.

└── <code>electronics.html</code>	Electronics category page.
└── <code>mens.html</code>	Men's products page.
└── <code>womens.html</code>	Women's products page.
└── <code>single.html</code>	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

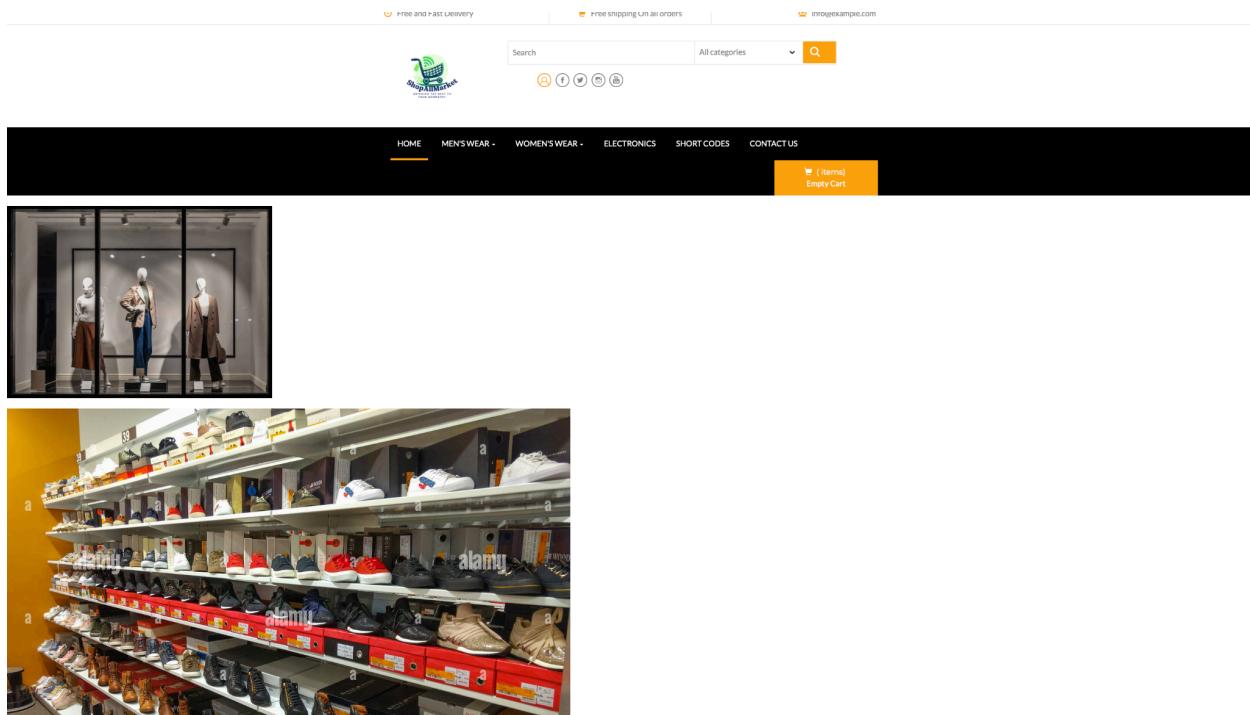
This section provides visual examples of the **ShopAllMarket** website, showcasing its key features and functionality. Each screenshot is accompanied by a brief description to help users and developers understand the design and purpose of each page.

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:

The screenshot displays a mobile application interface for an e-commerce platform. At the top, there is a banner titled "NEW ARRIVALS" with the subtext "Check out our latest arrivals, carefully selected to bring you fresh styles and amazing deals!". Below this, there are three promotional sections: "ELHOPRONIC COLLECTIONS" (Flat 50% Discount), "SPECIAL OFFERS AND 50% DISCOUNT ON MOBILES", and "BUY 1 GET 1 FREE ON BRANDED WATCHES".

The main content area features a navigation bar with tabs: "Latest Designs" (selected), "Special Offers" (highlighted in orange), and "Collections". Below the tabs, there are two rows of product cards. The first row includes "Wedges" (Sandals) at \$45.99, "Casual Shoes" at \$45.99, and "Sport Shoes" at \$45.99. The second row includes "Watches" at \$45.99. Each card has an "Add to cart" button.

At the bottom, there is a dark-themed dashboard section titled "BUY YOUR PRODUCT IN A SIMPLE WAY AT SHOP-ALL-MARKET". It features four circular icons: a user profile, a checkmark, a credit card, and a gear. Below each icon is a corresponding text block: "LOGIN TO YOUR ACCOUNT" (Welcome back! Please log in to your account to access your personalized dashboard, manage your orders, and enjoy an exclusive shopping experience.), "SELECT YOUR ITEM" (Select your favorite items from our wide range of collections and enjoy exclusive deals and discounts!), and "MAKE PAYMENT" (Securely complete your purchase by selecting a payment method. We offer multiple options to ensure a smooth and safe transaction.).

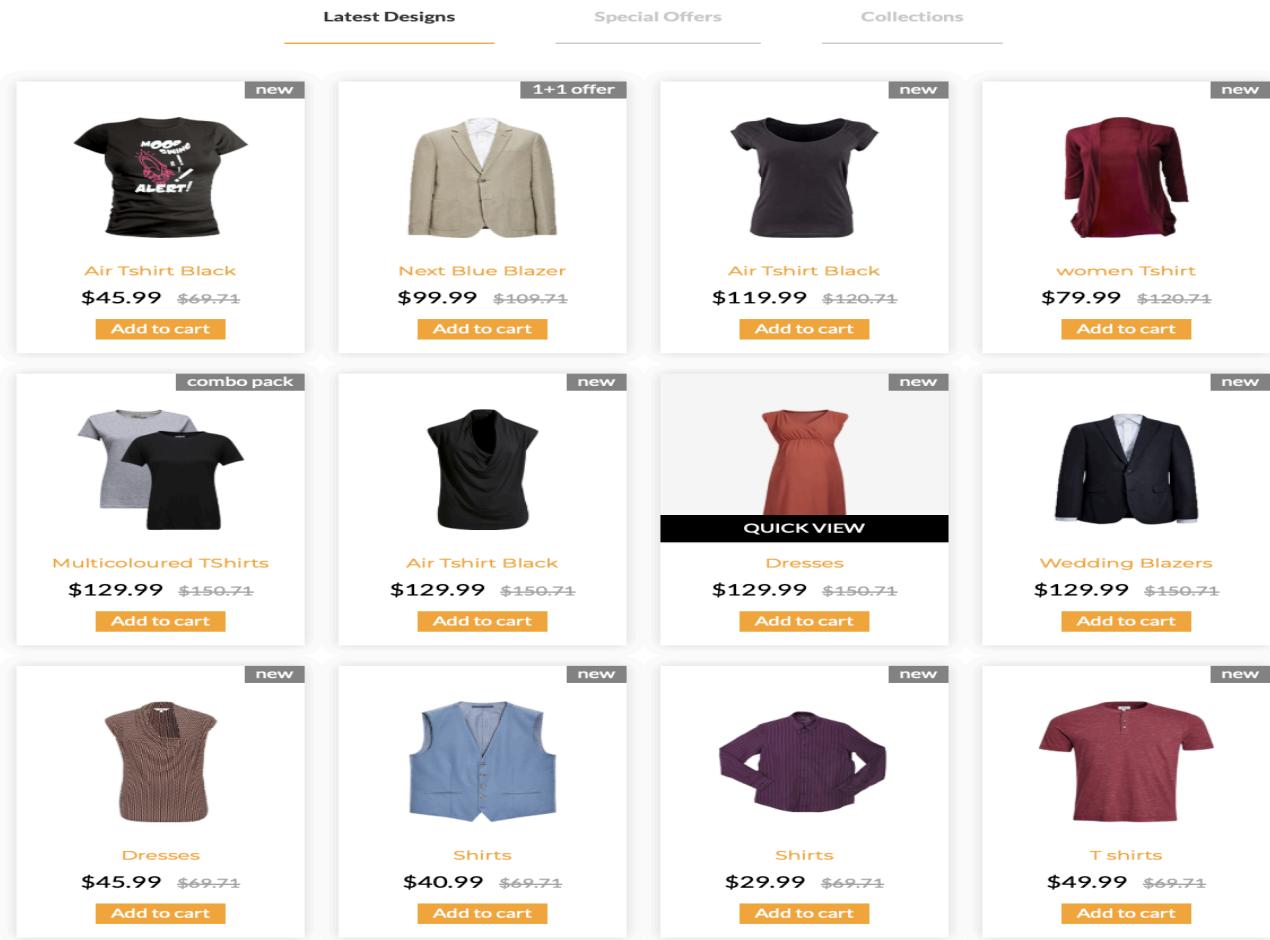
Product Details Page

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:



Search All categories

SIGN UP FOR FREE

Email :

Type here

Password :

.....

Re-type Password :

.....

REGISTER NOW

SIGN IN WITH YOUR ACCOUNT

Email :

Type here

Password :

.....

Forgot password?

Remember Me.

SIGNIN

By logging in you agree to our [Terms and Conditions](#) and [Privacy Policy](#)

SIGN UP FOR FREE

Email :

shopAllmarket@gmail.com

Password :

.....

Re-type Password :

.....

REGISTER NOW

SIGN IN WITH YOUR ACCOUNT

Email :

shopAllMarket@gmail.com

Password :

.....

Forgot password?

Remember Me.

SIGNIN

By logging in you agree to our [Terms and Conditions](#) and [Privacy Policy](#)

9. Appendices

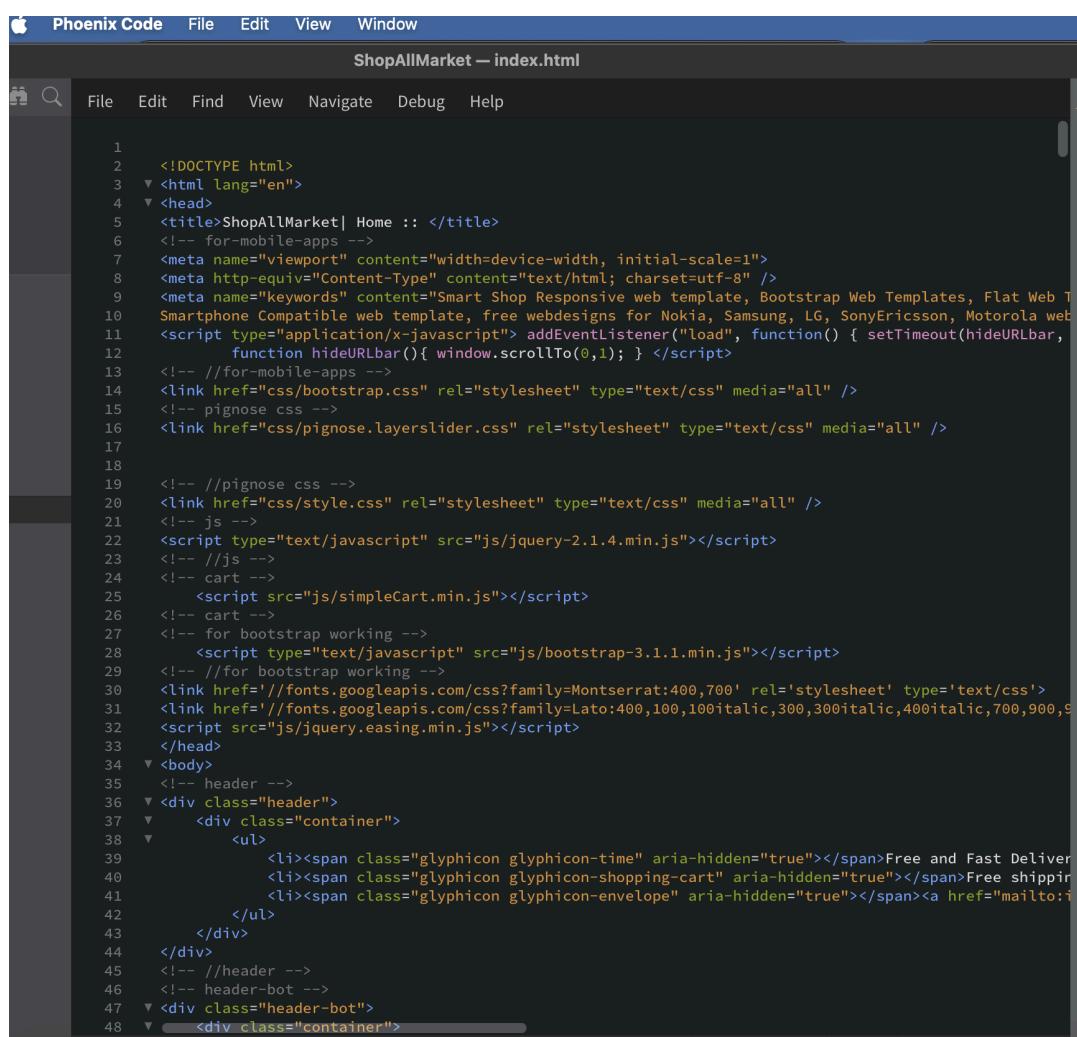
Code & References

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:



The screenshot shows the Phoenix Code editor interface with the file `ShopAllMarket – index.html` open. The code is a semantic HTML structure for a homepage, featuring meta tags for viewport, content-type, and keywords, as well as links to CSS files for Bootstrap and Pignose sliders. It also includes scripts for jQuery, simpleCart.js, and Bootstrap. The code is color-coded for syntax highlighting.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <title>ShopAllMarket | Home :: </title>
5  <!-- for-mobile-apps -->
6  <meta name="viewport" content="width=device-width, initial-scale=1">
7  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8  <meta name="keywords" content="Smart Shop Responsive web template, Bootstrap Web Templates, Flat Web T
9  Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG, SonyEricsson, Motorola web
10 <script type="application/x-javascript"> addEventListener("load", function() { setTimeout(function() {
11   hideURLbar();
12 }, 0); } ) </script>
13 <!-- //for-mobile-apps -->
14 <link href="css/bootstrap.css" rel="stylesheet" type="text/css" media="all" />
15 <!-- pignose css -->
16 <link href="css/pignose.layerslider.css" rel="stylesheet" type="text/css" media="all" />
17
18
19 <!-- //pignose css -->
20 <link href="css/style.css" rel="stylesheet" type="text/css" media="all" />
21 <!-- js -->
22 <script type="text/javascript" src="js/jquery-2.1.4.min.js"></script>
23 <!-- //js -->
24 <!-- cart -->
25 <script src="js/simpleCart.min.js"></script>
26 <!-- cart -->
27 <!-- for bootstrap working -->
28 <script type="text/javascript" src="js/bootstrap-3.1.1.min.js"></script>
29 <!-- //for bootstrap working -->
30 <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet" type="text/css">
31 <link href="https://fonts.googleapis.com/css?family=Lato:400,100,100italic,300,300italic,400italic,700,900,900italic" rel="stylesheet" type="text/css">
32 <script src="js/jquery.easing.min.js"></script>
33 </head>
34 <body>
35 <!-- header -->
36 <div class="header">
37 <div class="container">
38 <ul>
39   <li><span class="glyphicon glyphicon-time" aria-hidden="true"></span>Free and Fast Delivery
40   <li><span class="glyphicon glyphicon-shopping-cart" aria-hidden="true"></span>Free shipping
41   <li><span class="glyphicon glyphicon-envelope" aria-hidden="true"></span><a href="mailto:info@shopallmarket.com">Email us</a>
42 </ul>
43 </div>
44 </div>
45 <!-- //header -->
46 <!-- header-bot -->
47 <div class="header-bot">
48 <div class="container">
```

Register /Login code

```
File Edit Find View Navigate Debug Help
Working Files
  index.html
  womens.html
  imagezoom.js
  contact.html

ShopAllMarket .
  CSS
    bootstrap.css
    flexslider.css
    jquery-ui.css
    pignose.layerslider.css
    style.css
  fonts
  images
  js
    contact.html
    electronics.html
    index.html
    mens.html
    single.html
    womens.html

328  / 2023 ShopAllMarket. All rights reserved<a href="http://AmnaCode.com/"></a></p>
329
330
331
332
333  <!-- Modal -->
334  <div id="myModal4" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
335    <div class="modal-dialog">
336      <div class="modal-content">
337        <div class="modal-header">
338          <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
339        <div class="modal-body modal-spa">
340          <div class="login-grid">
341            <div class="login">
342              <div class="login-bottom">
343                <h3>Sign up for free</h3>
344                <form>
345                  <div class="sign-up">
346                    <h4>Email :</h4>
347                    <input type="text" value="Type here" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Type here';}" required="">
348                  </div>
349                  <div class="sign-up">
350                    <h4>Password :</h4>
351                    <input type="password" value="Password" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Password';}" required="">
352                  </div>
353                  </div>
354                  <div class="sign-up">
355                    <h4>Re-type Password :</h4>
356                    <input type="password" value="Password" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Password';}" required="">
357                  </div>
358                  <div class="sign-up">
359                    <input type="submit" value="REGISTER NOW" >
360                  </div>
361                </form>
362              </div>
363            </div>
364          </div>
365          <div class="login-right">
366            <h3>Sign in with your account</h3>
367            <form>
368              <div class="sign-in">
369                <h4>Email :</h4>
370                <input type="text" value="Type here" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Type here';}" required="">
371              </div>
372              <div class="sign-in">
373                <h4>Password :</h4>
374                <input type="password" value="Password" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'Password';}" required="">
375                <a href="#">Forgot password?</a>
376              </div>
377              <div class="single-bottom">
378                <input type="checkbox" id="brand" value="" checked="">
379                <label for="brand"><span>Remember Me.</span></label>
380              </div>
381              <div class="sign-in">
382                <input type="submit" value="SIGNIN" >
383              </div>
384            </form>
385          </div>
386        </div>
387      </div>
388    </div>
389  </div>
390
```

2. CSS Styling

The `styles.css` file centralizes styling for consistent design. Here's an example:

The screenshot shows the Phoenix Code editor interface. The title bar reads "Phoenix Code" and "ShopAllMarket — css/style.css". The menu bar includes "File", "Edit", "View", and "Window". The main area displays a CSS code editor with the following content:

```
1 2  ▼ html, body{  
3      margin:0;  
4      font-size: 100%;  
5      font-family: 'Lato', sans-serif;  
6      background: #fff;  
7  }  
8  ▼ body a {  
9      text-decoration:none;  
10     transition:0.5s all;  
11     -webkit-transition:0.5s all;  
12     -moz-transition:0.5s all;  
13     -o-transition:0.5s all;  
14     -ms-transition:0.5s all;  
15  }  
16  ▼ a:hover{  
17      text-decoration:none;  
18  }  
19  ▼ input[type="button"],input[type="submit"]{  
20      transition:0.5s all;  
21      -webkit-transition:0.5s all;  
22      -moz-transition:0.5s all;  
23      -o-transition:0.5s all;  
24     -ms-transition:0.5s all;  
25  }  
26  
27  ▼ h1,h2,h3,h4,h5,h6{  
28      margin:0;  
29      font-family: 'Montserrat', sans-serif;  
30  }  
31  ▼ p{  
32      margin:0;  
33  }  
34  ▼ ul{  
35      margin:0;  
36      padding:0;  
37  }  
38  ▼ label{  
39      margin:0;  
40  }  
41  /*-- header --*/  
42  ▼ .header {  
43      border-bottom: 1px solid #eee;  
44  }  
45  ▼ .header ul li {  
46      display: inline-block;  
47      width: 33%;
```

The status bar at the bottom shows "Line 15, Column 2 — 5091 Lines" and icons for INS, utf8, CSS, and Tab Size: 4 Auto.

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>`.