

ASSIGNMENT 2 FRONT SHEET

Qualification	BTEC Level 5 HSSND Diploma in Computing		
Unit number and title	Unit 10: Website Design & Development		
Submission date	27-08-2022	Date Received 1st submission	
Re-submission Date		Date Received 2nd submission	
Student Name	Nguyen Tuan Kiet	Student ID	GDD210002
Class	GCD1002	Assessor name	Phyo Min Tun
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	Kiet Tuan Nguyen

Grading grid

P5	P6	P7	M4	M5	D2	D3

Summative Feedback:

Resubmission Feedback:

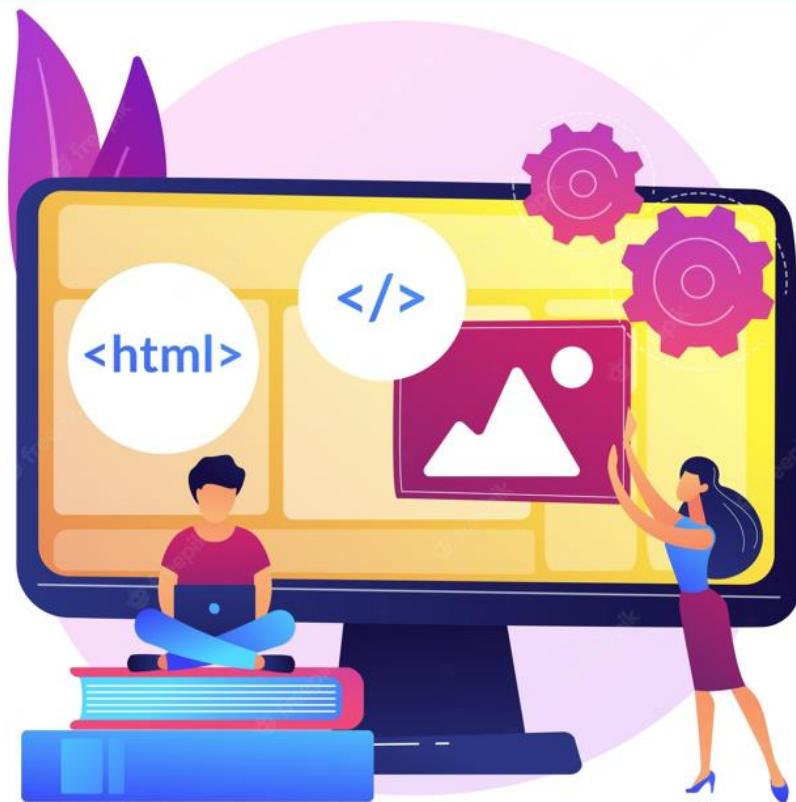
Grade:

Assessor Signature:

Date:

Lecturer Signature:

Website Design and Development



Assignment 2: Website Design & Development.

First and foremost, the first report would not have been accomplished without the assistance of friends and others. I'd want to convey my heartfelt gratitude to **Mr. Phyoe Min Tun**, who made helpful feedback to help me complete the assignment more efficiently.

Furthermore, I would like to express my sincere thanks to the authors for providing me with some important material that helped me create a better assignment report.

Table of Contents

Table of Figures	5
List of Tables.....	7
Chapter 1: Create a design document for a branded, multipage website supported with medium fidelity wireframes and a full set of client and user requirements (P5).....	8
1. Website design principles.....	8
2. Website standards and guidelines.....	10
3. User's Requirements	13
<i>Customer.</i>	13
<i>Manager</i> 13	
4. Use Case Diagram	14
5. Site map of development.....	15
6. Wireframes for each proposed page	17
<i>Wireframes for user session</i>	17
<i>Wireframes for admin session</i>	26
Chapter 2: Use your design document with appropriate principles, standards and guidelines to produce a branded, multipage website supported with realistic content (P6)	29
1. Tools and techniques used to design website	29
<i>Diagram.net</i>	29
<i>Visual Studio Code</i>	30
2. Screenshots of the webpages and its source code	32
<i>User session</i>	32
<i>Admin session</i>	65
<i>CSS and JavaScript</i>	81
3. Features of the website.....	83
<i>Registering for a new user</i>	83
<i>Login with email and password</i>	86
<i>CRUD features</i>	88
Chapter 3: Create a suitable Test Plan identifying key performance area and use it to review the functionality and performance of your website. User Experience (UX) and User Interface (UI). (P7)	95
1. Test plan template	95
2. Test case template	96
3. Screenshot in running.....	98

<i>Test case 1: Registration (Pass).....</i>	98
<i>Test case 2: Log in for user session (Pass)</i>	99
<i>Test case 3: Log in for admin session (Pass)</i>	100
<i>Test case 4: Log out (Pass)</i>	101
<i>Test case 5: View customer information on admin session (Pass).....</i>	103
<i>Test case 6: Update product information on admin session (Pass)</i>	104
<i>Test case 7: Delete product from cart (Pass).....</i>	106
<i>Test case 8: Delete user information on admin session (Pass).....</i>	109
<i>Test case 9: Register with different password and confirm password. (Pass).....</i>	112
<i>Test case 10: Log in with incorrect password (Pass).....</i>	114
Chapter 4: Compare and contrast the multipage website created to the design document. (M4).....	116
Chapter 5: Evaluate the Quality Assurance (QA) process and review how it was implemented during your design and development stages. (M5)	124
1. Identify	124
2. How to do quality assurance: Complete process (Clarify which steps had applied in web development with evidence)	125
Chapter 6: Critically evaluate the design and development process against your design document and analyse any technical challenges. (D2)	128
1. Website evaluation	128
2. The technical challenges	128
Chapter 7: Critically evaluate the results of your Test Plan and include a review of the overall success of your multipage website; use this evaluation to explain any areas of success and provide justified recommendations for areas that require improvement (D3)	130
1. Test result evaluation	130
2. Providing justified recommendations of improvement	130
3. Advantage and disadvantage of web project	131
References	132
Appendix	133
Skill Assessment for HTML.....	133
Skill Assessment for CSS.....	133
The Website on 000WebHost.....	134

Table of Figures

Figure 1: C1.1_Illustration_Website design principles	9
Figure 2: C1.2_Illustration_Web Design Standards.....	12
Figure 3: C1.4_Use Case Diagram	14
Figure 4: C1.5_Site map_User session.....	15
Figure 5: C1.5_Site map_Admin session	16
Figure 6: C1.6_Wireframe_User session_“Register” page	17
Figure 7: C1.6_Wireframe_User session_“Log In” page.....	18
Figure 8: C1.6_Wireframe_User session_“Home” page.....	19
Figure 9: C1.6_Wireframe_User session_“About” page	20
Figure 10: C1.6_Wireframe_User session_“Shop” page	21
Figure 11: C1.6_Wireframe_User session_“Contact” page	22
Figure 12: C1.6_Wireframe_User session_“Search” page.....	23
Figure 13: C1.6_Wireframe_User session_“Cart” page.....	23
Figure 14: C1.6_Wireframe_User session_“Check Out” page	24
Figure 15: C1.6_Wireframe_User session_“Order” page	25
Figure 16: C1.6_Wireframe_Admin session_“Home” page	26
Figure 17: C1.6_Wireframe_Admin session_“Products” page	27
Figure 18: C1.6_Wireframe_Admin session_“Order” page.....	28
Figure 19: C1.6_Wireframe_Admin session_“User” page	28
Figure 20: C1.6_Wireframe_Admin session_“Message” page	29
Figure 21: C2.1_Diagram.net_Wireframes	30
Figure 22: C2.1_Visual Studio Code_IntelliSense feature.....	31
Figure 23: C2.2_User Session_User Interface_“Register” page	32
Figure 24: C2.2_User Session_Source Code_“Register” page	34
Figure 25: C2.2_User Session_User Interface_“Log In” page.....	34
Figure 26: C2.2_User Session_Source Code_“Log In” page	36
Figure 27: C2.2_User Session_User Interface_“Home” page.....	37
Figure 28: C2.2_User Session_Source Code_“Home” page.....	40
Figure 29: C2.2_User Session_User Interface_“About” page	41
Figure 30: C2.2_User Session_Source Code_“About” page.....	43
Figure 31: C2.2_User Session_User Interface_“Shop” page	44
Figure 32: C2.2_User Session_Source Code_“Shop” page	47
Figure 33: C2.2_User Session_User Interface_“Contact” page	47
Figure 34: C2.2_User Session_Source Code_“Contact” page	49
Figure 35: C2.2_User Session_User Interface_“Search” page.....	50
Figure 36: C2.2_User Session_Source Code_“Search” page.....	53
Figure 37: C2.2_User Session_User Interface_“Cart” page.....	53
Figure 38: C2.2_User Session_Source Code_“Cart” page.....	56
Figure 39: C2.2_User Session_User Interface_“Check Out” page	57

Figure 40: C2.2_User Session_Source Code_“Check Out” page	61
Figure 41: C2.2_User Session_User Interface_“Order” page	62
Figure 42: C2.2_User Session_Source Code_“Order” page	64
Figure 43: C2.2_Admin Session_User Interface_“Home” page	65
Figure 44: C2.2_Admin Session_Source Code_“Home” page	68
Figure 45: C2.2_Admin Session_User Interface_“Product” page	69
Figure 46: C2.2_Admin Session_Source Code_“Product” page.....	73
Figure 47: C2.2_Admin Session_User Interface_“Message” page.....	74
Figure 48: C2.2_Admin Session_Source Code_“Message” page	76
Figure 49: C2.2_Admin Session_User Interface_“Order” page.....	76
Figure 50: C2.2_Admin Session_Source Code_“Order” page.....	79
Figure 51: C2.2_Admin Session_User Interface_“User” page	79
Figure 52: C2.2_Admin Session_Source Code_“User” page	81
Figure 53: C2.2_CSS and JavaScript_admin_script.js	82
Figure 54: C2.2_CSS and JavaScript_script.js	83
Figure 55: C2.3_Register.....	84
Figure 56: C2.3_Register_Fill out.....	85
Figure 57: C2.3_Register_Duplicate with the previous email.....	85
Figure 58: C2.3_Register_No password match.....	86
Figure 59: C2.3_Login_Incorrect email or password.....	87
Figure 60: C2.3_Login_Incorrect email	88
Figure 61: C2.3 CRUD feature_ Create action.....	89
Figure 62: C2.3 CRUD feature_ Create action.....	89
Figure 63: C2.3 CRUD feature_ Read action.....	90
Figure 64: C2.3 CRUD feature_ Read action.....	90
Figure 65: C2.3 CRUD feature_ Read action.....	91
Figure 66: C2.3 CRUD feature_ Update action	92
Figure 67: C2.3 CRUD feature_ Update action	93
Figure 68: C2.3 CRUD feature_Delete action.....	93
Figure 69: C2.3 CRUD feature_Delete action.....	94
Figure 70: C3.3_Test case 1_Registration.....	98
Figure 71: C3.3_Test case 1_Checking data on “users” table.....	99
Figure 72: C3.3_Test case 2_Log in for user session.....	99
Figure 73: C3.3_Test case 2_Home page for user session.....	100
Figure 74: C3.3_Test case 3_Log in for admin session.	101
Figure 75: C3.3_Test case 3_Home page for admin session.....	101
Figure 76: C3.3_Test case 4_Log out.....	102
Figure 77: C3.3_Test case 4_Log out.....	102
Figure 78: C3.3_Test case 5_View all user information	103
Figure 79: C3.3_Test case 5_Checking data on “users” table.....	104
Figure 80: C3.3_Test case 6_Before the update	105
Figure 81: C3.3_Test case 6 Updating	105

Figure 82: C3.3_Test case 6_After the update	105
Figure 83: C3.3_Test case 6_Checking data on “products” table.	106
Figure 84: C3.3_Test case 7_Before removing the product from the cart	107
Figure 85: C3.3_Test case 7_While removing product from cart	108
Figure 86: C3.3_Test case 7_After removing the product from the cart.....	108
Figure 87: C3.3_Test case 7_Checking data on “cart” table.	109
Figure 88: C3.3_Test case 8_Before deleting user information	110
Figure 89: C3.3_Test case 8_While deleting user information	110
Figure 90: C3.3_Test case 8_After deleting user information	111
Figure 91: C3.3_Test case 8_Checking data on “cart” table.	111
Figure 92: C3.3_Test case 9_Register with different password and confirm password.....	112
Figure 93: C3.3_Test case 9_Message “Confirm password not matched!”	113
Figure 94: C3.3_Test case 9_Checking data on “users” table.....	113
Figure 95: C3.3_Test case 10_Log in with incorrect password	114
Figure 96: C3.3_Test case 10_Message “Incorrect email or password!”	115
Figure 97: C5.2_Plan_Use Case Diagram.....	126
Figure 98: C5.2_Do_Visual Studio Code.....	127
Figure 99: C5.2_Act_My Highland Coffee Website.....	127
Figure 100: Appendix _Skill Assessment_ HTML	133
Figure 101: Appendix _Skill Assessment_ CSS	133

List of Tables

Table 1: C3.1_Test plan template.....	95
Table 2: C3.2_Test case template.....	96
Table 3: C4_Comparison table of wireframe and actual web page.....	116
Table 4: C.3_Web project advantage and disadvantage table	131

Chapter 1: Create a design document for a branded, multipage website supported with medium fidelity wireframes and a full set of client and user requirements (P5)

1. Website design principles

The success of any website or online company is entirely dependent on the design of their website. The majority of your potential clients or visitors will always examine your website first, followed by a visit to your actual store. As a result, it is critical that your website design be flawless and relevant to your company's needs. Web design is more important for leads or conversions than you may realize. In truth, website design does not always refer to how it appears or feels, but rather to how it functions. A basic website with a well-structured design is sufficient to create leads or conversions.

A good website design requires a great deal of competence in the web design industry, as well as an experienced person who understands design approaches and web design concepts. The design ideas that we will cover will assist you in creating a flawless web design to produce more leads or conversions for your organization.

- Simplicity: Over-designing is ineffective. Having too many items on your website may divert your visitors' attention away from the core objective of your site. Simplicity is essential in producing effective, clear, and up-to-date designs that will make your website appealing and bring people to other pages of your website. In essence, this will assist you in regularly and successfully generating leads or conversions for your website.
- Color Schemes and Images: A correct color combination on your website can assist attract visitors, however a poor color selection will quickly deter people from your website. To get the ideal design for your website, you must choose at least three to four color combinations. The same logic applies to graphics; you must choose images based on the color scheme and content of each page.
- Communication: The primary goal of website visitors is to obtain information. If your website can speak with its users and offer them with all of the information they want, they will spend more time on it. This will assist you in increasing the number of leads or conversions for your internet business. As a result, always strive to organize the content effectively, including links, bullet points, headers, and subheadings.
- Consistency: Web design consistency is critical for your business. Concentrate on your website design and develop uniformity throughout all pages of your website. It is

obvious that the font style, design, size, subheadings, headers, and button styles must be consistent throughout the website. Complete all of these details ahead of time, including your website information.

- Rapid loading: Nobody loves websites that take too long to load. So, make sure you optimize your pictures, CSS, JS, and other On-Page variables to help you raise the loading speed of your website, and HTTPS will also help you enhance the loading speed of your website.
- Easy to Navigate: According to research, users prefer to stay on websites with simple navigation. To develop successful navigation, you must first create a good logical page structure, such as breadcrumbs, and then design clickable buttons across your whole site design.
- Typography and Readability: No matter how gorgeous your website is, the text still reigns supreme since the website's content provides the required information to users. Because search engine crawlers always take this information into account, it becomes a part of SEO activity. You should maintain your website material, as well as keywords, meta titles, descriptions, and other crucial SEO aspects, legible for your visitors. Always consider easy-to-read typefaces while building your website, which includes the body of information, headers, and subheadings.

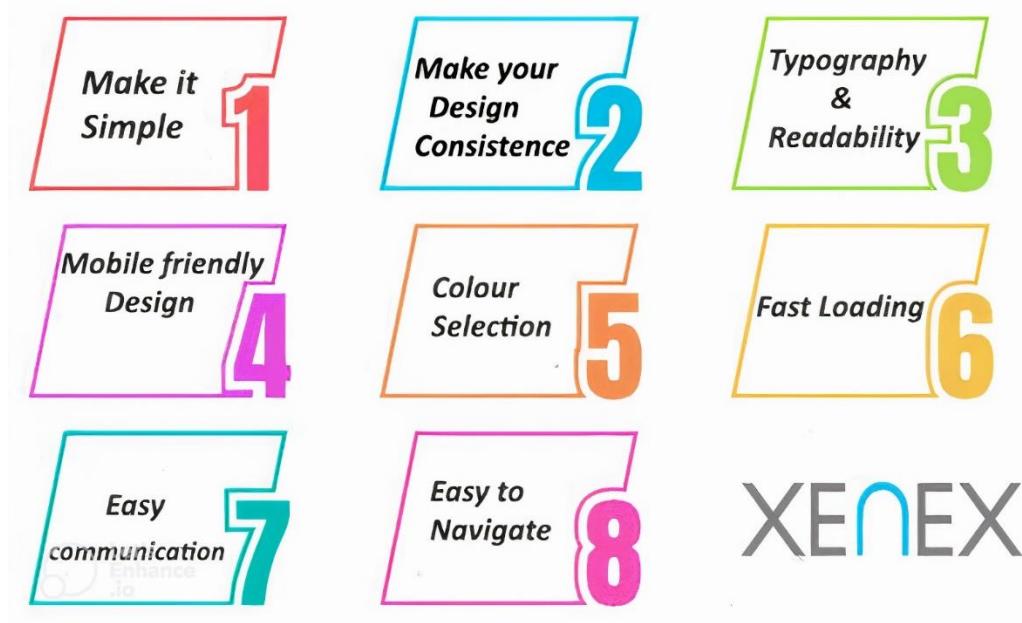


Figure 1: C1.1_Illustration_Website design principles

2. Website standards and guidelines

To tackle this issue, we constructed a checklist of 14 web design features and then examined 500 marketing websites to see how common these standards are. We adopt the following limits based on suggestions from an earlier NN Group article:

The same design technique is used by 80% or more of websites.

- Convention: 50-79% of websites have the same design strategy.
- Confusion: 49% of websites conform, yet no particular design concept is dominant.

Logo in the top left

Approximately 100% of the websites examined featured a clickable logo in the upper left corner of each page. That is an industry standard!

Contact in top right

In the upper right corner of each page, about 44% have a contact button or link. This location, while highly popular and regarded best practices, cannot be deemed standard.

Main navigation across the top

The main navigation was situated in the header at the top of each page on around 88% of the websites, making horizontal top-level navigation a web design standard.

Home page slideshow

A home page slideshow (also known as a carousel) with a revolving series of images and messages is used by about 32% of websites.

Value proposition high up on the home page

Approximately 80% of marketing websites have an explicit value proposition prominently displayed on the front page. As a result, most websites explain their worth to visitors "above the fold." The remaining companies lacked a distinct value proposition. There is no standard pixel height for browsers, as any web designer can tell you. As a result, there is no fold. However, some design components display high on sites and are typically accessible to the majority of visitors without the need for scrolling.

Call to Action high up on the home page

Approximately 78% of the websites contained visible calls to action. The proportion fell below our criteria; it is unquestionably a convention.

Search feature in the header

A search tool in the header is present on around 54% of websites. Approximately 50% of all marketing websites lack a search option that shows "globally" on every page as a link, symbol, or search box.

Signup box in the footer

In the footer of around 24% of websites, users may sign up for and subscribe to email updates. So, this is a typical area to collect email addresses, but it is neither a rule or a standard.

Social media icons in the footer

Around 72% of websites have social networking website buttons in the bottom. This effectively makes them a regular design feature.

Responsive design

Approximately 68% of websites are mobile-friendly thanks to responsive web design. This provides visitors with an excellent experience regardless of device - phone, tablet, or desktop.

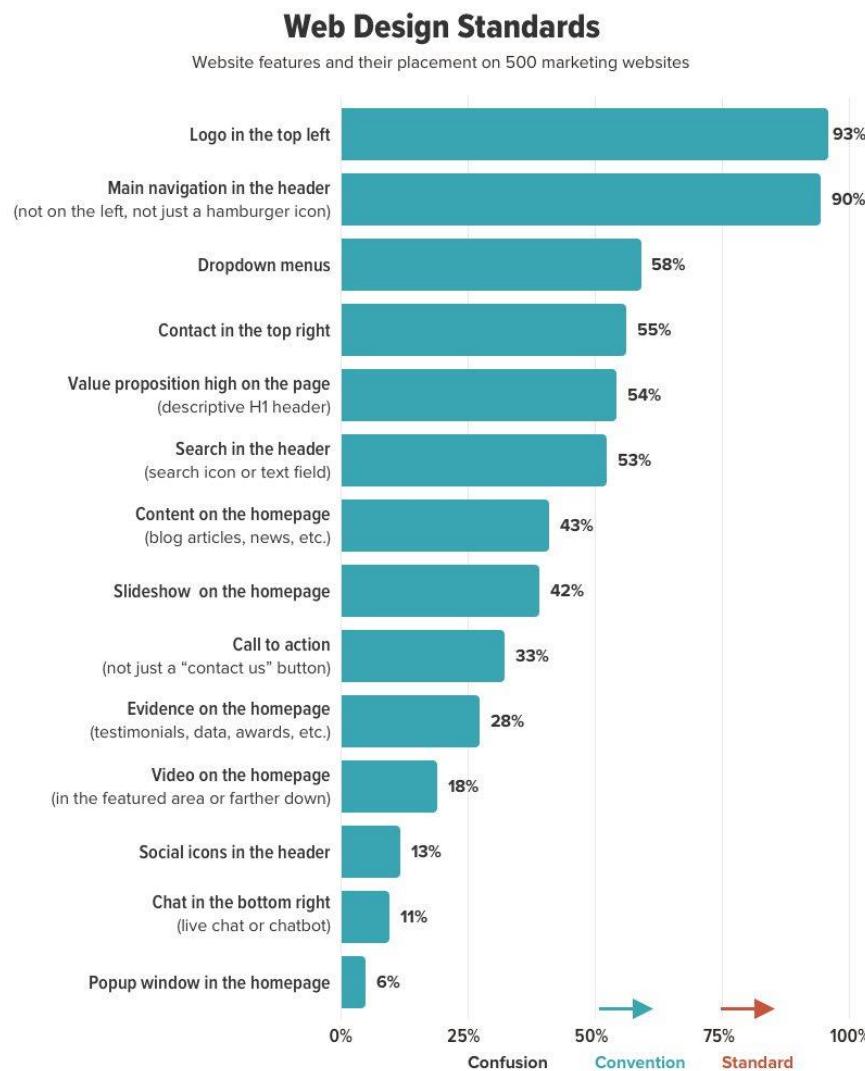


Figure 2: C1.2_Illustration_Web Design Standards

Moreover, entirely developed processes will be following the structure of WCAG 2.0 Guideline that

Web content must be:

- **Perceivable:** Users must be able to perceive information and user interface (UI) components.
- **Operable:** Users must be able to interact with UI components.
- **Understandable:** Users must be able to comprehend information about the UI and its functionality.

- **Robust:** Content must be strong enough to be reliably understood by a wide range of user agents, including assistive devices.

3. User's Requirements

The first stage in designing any e-Commerce application is to conduct user requirements to generate a list of features to include in the application. This is a critical input for determining the application's capabilities.

Customer.

- As a customer, I want to be able to access my Highland Coffee Website from any Web browser that supports HTML and cookies.
- As a client, I'd want to be able to discover products by their names, which is quite helpful for customers when there are a lot of products on the site.
- As a customer, I'd want to set up a secure user account so that I may log in and make purchases on the website.
- As a consumer, I'd want to add the things I'd like to buy to my shopping cart.
- As a consumer, I'd want to have up-to-date shop information so that I may contact them as soon as possible if necessary.

Manager

- As a manager, I'd want to make some changes, add some products, or eliminate others.
- As a manager, I'd want to see the user's username, address, phone number, and email address in order to contact them simply.
- As a manager, I'd like to see what things customers have in their shopping carts, so I know what they purchased.
- As a manager, I'd like to know how much each cart costs so that I can track sales.

4. Use Case Diagram

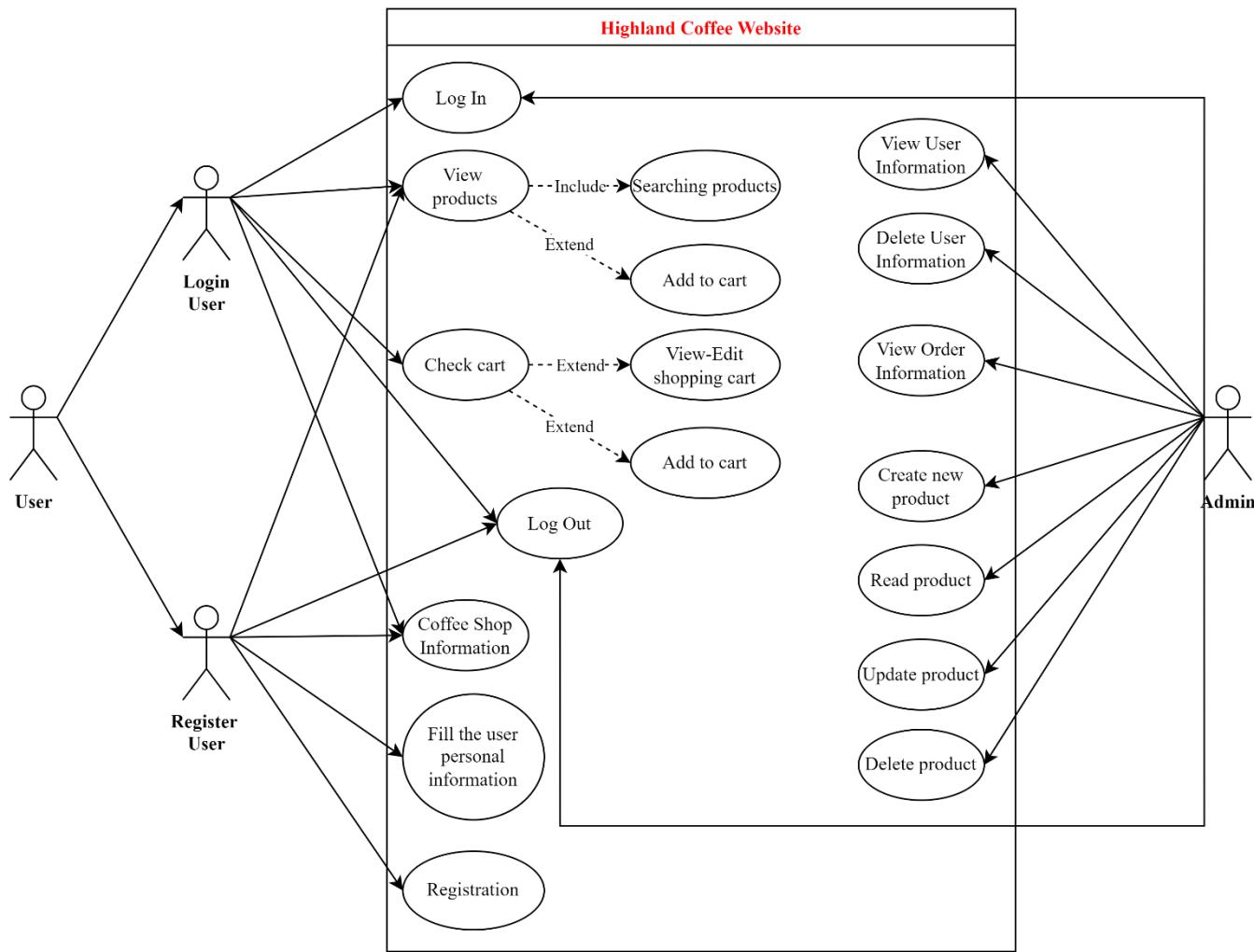


Figure 3: C1.4_Use Case Diagram

This is a **Use Case Diagram** that includes two user roles, customer and manager, as well as nine separate user actions that are categorized by item.

Customers are classified into two types:

- **Customers** who have made an account may shop for and purchase items.
- **New Customers** who do not have an account can explore items and register to buy.

Admin manages customers by viewing order information, deleting user information, creating a new product as well as updating the information of that product, ...

5. Site map of development

The image below is a development sitemap map that I created for the user session.

- In the "Shop" section, I will display all the food and drinks that we currently have so that customers can see what products are still for sale. In addition, the "Add to cart" button in the bottom right corner is created to help customers add an item to the cart more quickly and conveniently.
- There are three subsections in the user account section, including:
 - **Registration:** Customers can use the "Register" function to create an account directly on the website.
 - **Log in:** After having an account, customers with existing accounts can log in and make purchases through the "Log in" function button.
 - **Log out:** Log out of the account with the "Log out" button.
- The "Placed Order" button is created to help customers get more detailed information about their order, including order time, order information, delivery address, payment method, total order value as well as payment status.
- The "Contact" function button was created to help customers contact the admin team for support in the fastest way.

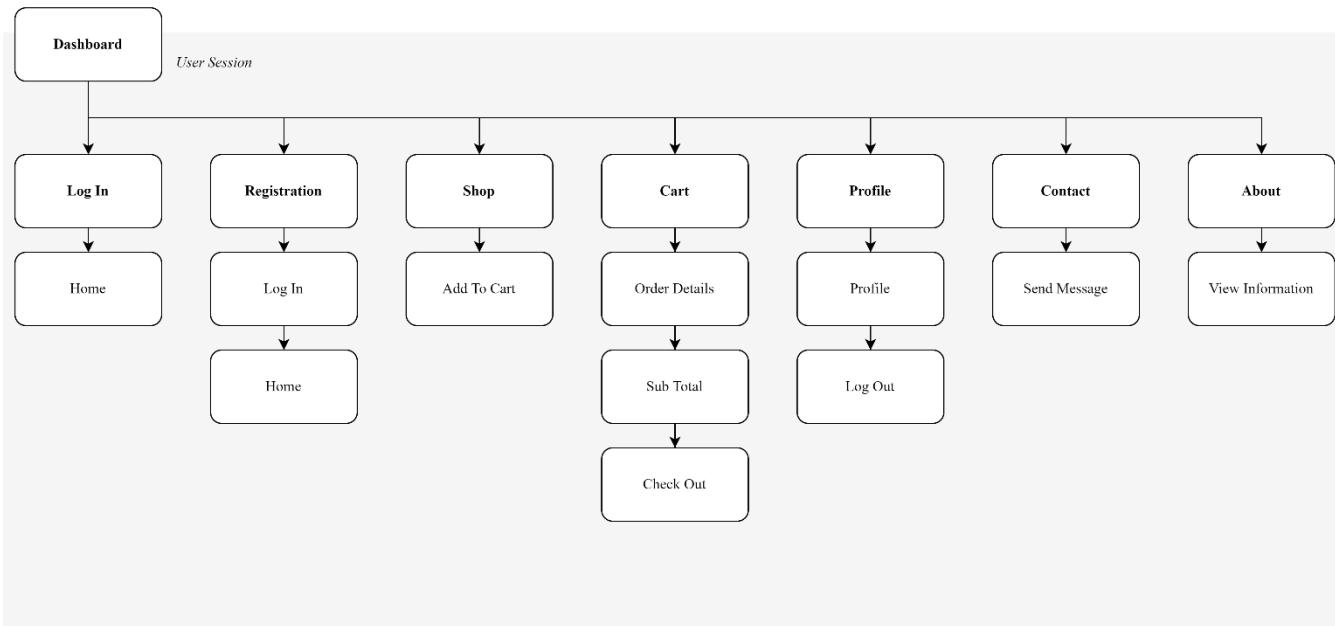


Figure 4: C1.5_Site map_User session

The image below is a development sitemap map that I created for the admin session.

- There are two subsections in the user account section, including:
 - **Log in:** After being provided with an admin account, designated users can log in and support customers through the "Log in" function button.
 - **Log out:** Log out of the account with the "Logout" button.

Note, because it is an admin account, we do not allow other users to register an account for admin in order to secure information for internal employees as well as users.

- Button "Products" function only allows the admin to add products, view product information, update product information as well as delete that product from the website.
- Button "Orders" function allows admin to update customer's order status as well as delete that order.
- The "Users" function button allows the admin to view customer information including their name and email as well as delete their account.
- The "Message" function button allows the admin to view messages from users as well as delete messages after viewing.

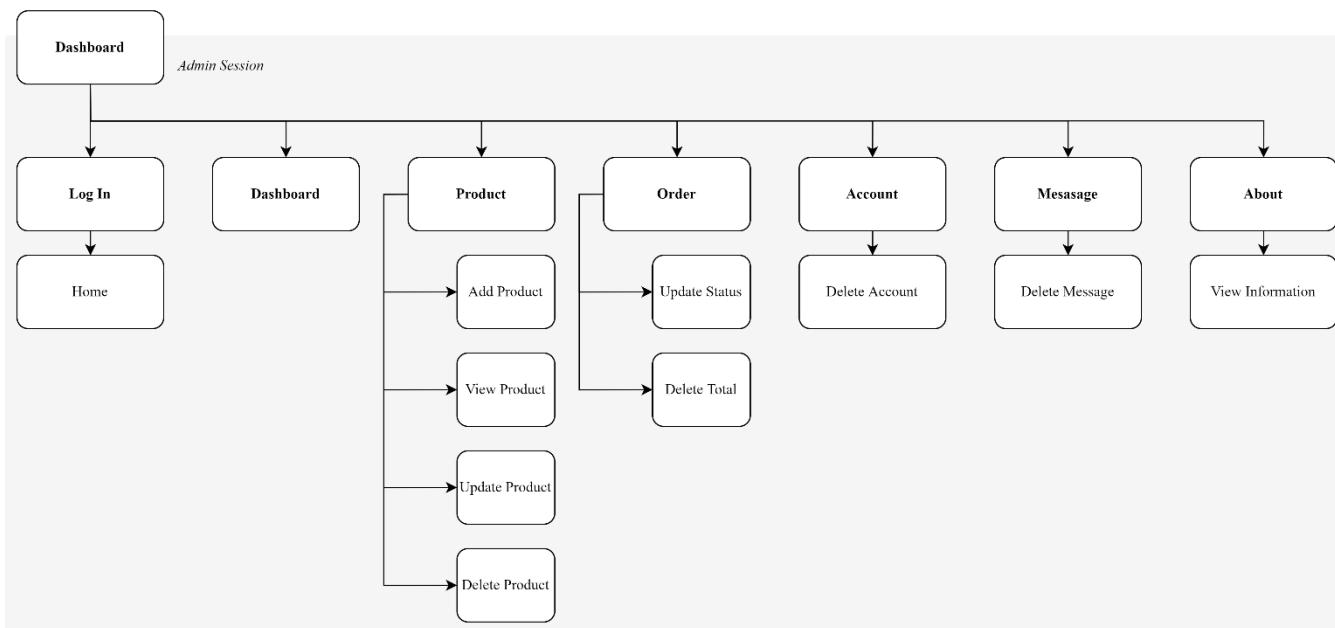


Figure 5: C1.5_Site map_Admin session

6. Wireframes for each proposed page

Wireframes for user session

“Register” page

The wireframe shows a large rectangular frame representing the user session. Inside this frame, there is a smaller rectangular box labeled "REGISTER NOW". This box contains five input fields: "Enter your name", "Enter your email", "Enter your password", "Confirm your password", and "User". Below these fields is a "Register Now" button. At the bottom of the box, there is a link that says "Already have an account? [Login now](#)".

Figure 6: C1.6_Wireframe_User session_“Register” page

“Log In” page

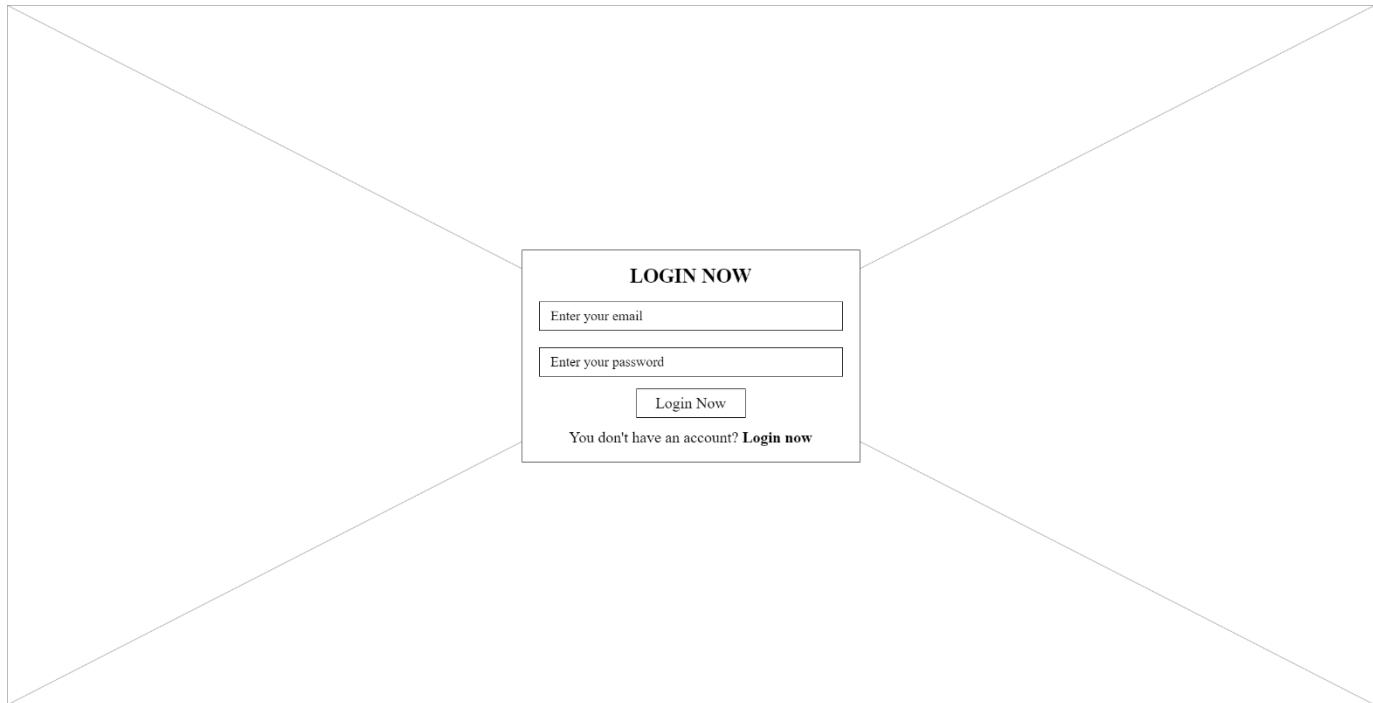


Figure 7: C1.6_Wireframe_User session_“Log In” page

“Home” page

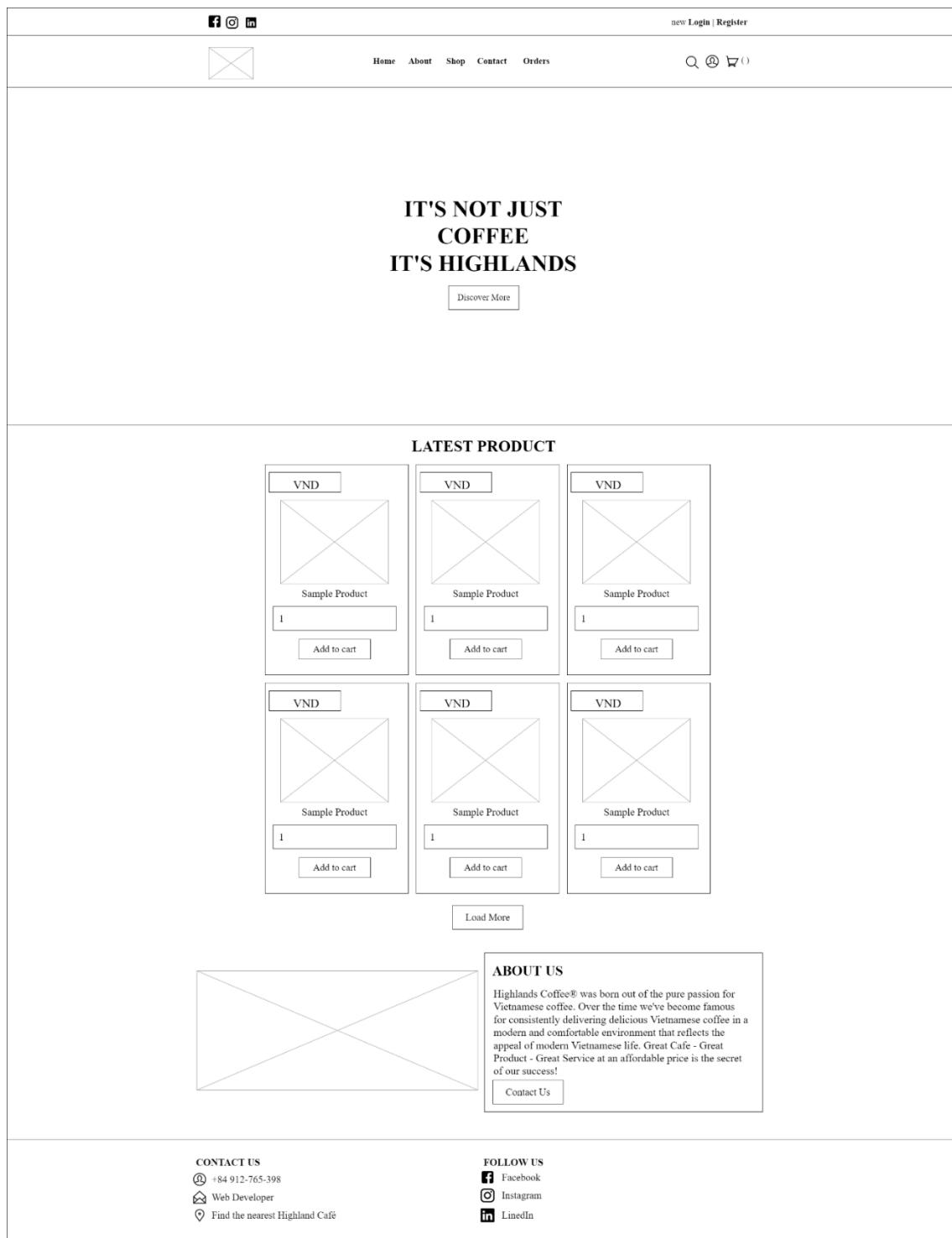
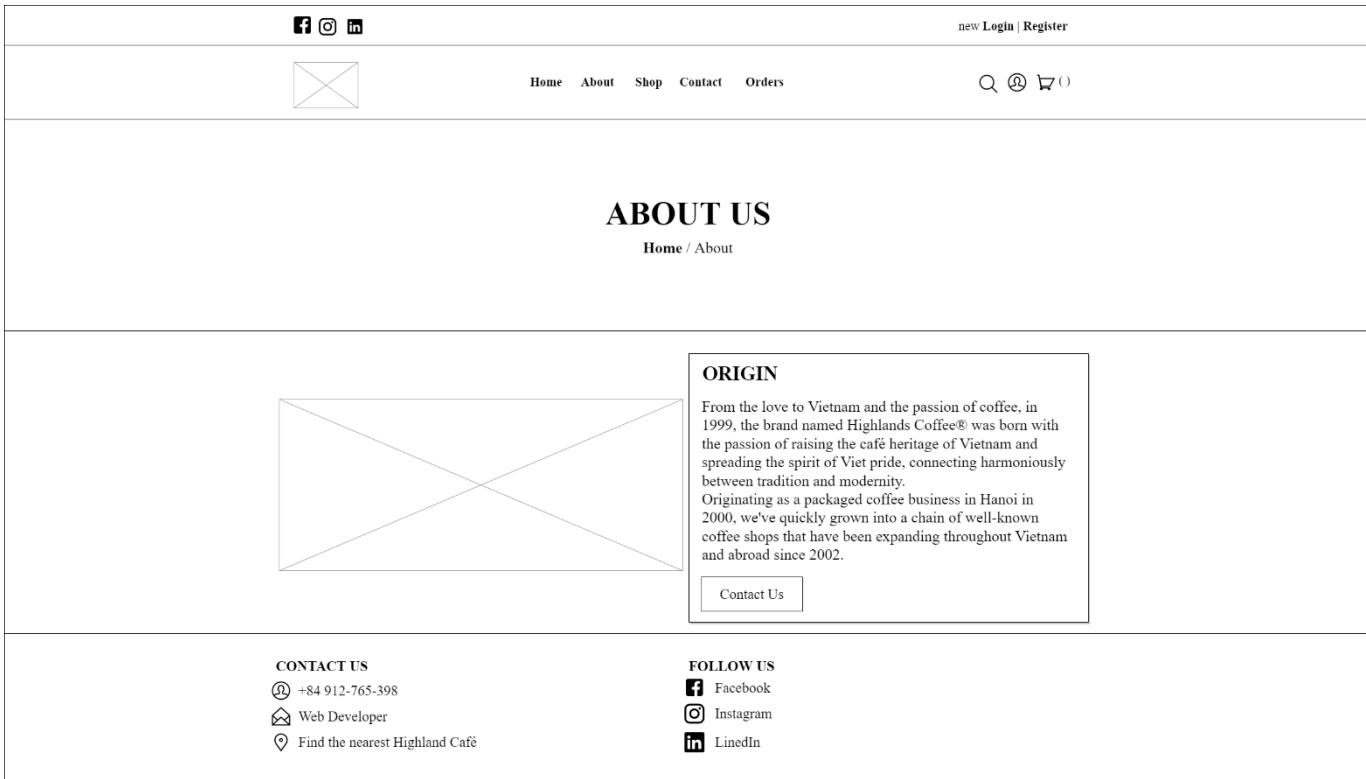


Figure 8: C1.6_Wireframe_User session_ “Home” page

“About” page

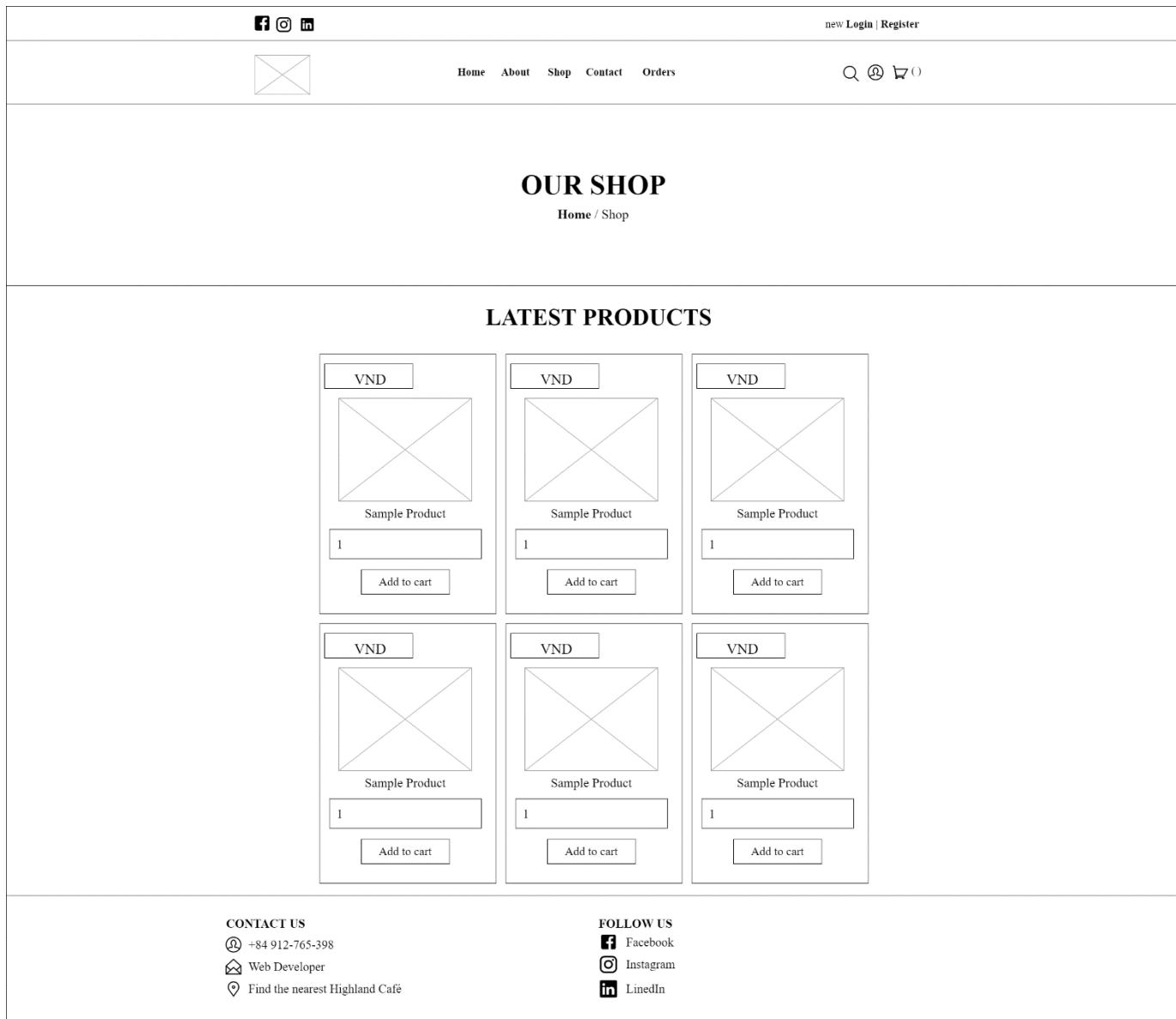


The wireframe illustrates the structure of the 'About' page:

- Header:** Features social media icons (Facebook, Instagram, LinkedIn) on the left, a search bar with a magnifying glass icon on the right, and a navigation menu with links to Home, About, Shop, Contact, and Orders.
- Section Headers:** 'ABOUT US' and 'ORIGIN' are prominently displayed in bold capital letters.
- Content Areas:** The 'ABOUT US' section contains placeholder text and a large rectangular area for an image. The 'ORIGIN' section contains descriptive text about the brand's history and expansion.
- Call-to-Action:** A 'Contact Us' button is located within the 'ORIGIN' section.
- Contact Information:** A 'CONTACT US' section on the left includes icons for phone, email, and location, along with their respective details.
- Social Media:** A 'FOLLOW US' section on the right lists social media links for Facebook, Instagram, and LinkedIn.

Figure 9: C1.6_Wireframe_User session_ “About” page

“Shop” page



The wireframe for the “Shop” page is organized into several sections:

- Header:** Contains social media icons (Facebook, Instagram, LinkedIn), a search bar, and navigation links for Home, About, Shop, Contact, Orders, and a new Login/Register button.
- Main Content Area:** Features a title “OUR SHOP” above a breadcrumb “Home / Shop”. Below this is a section titled “LATEST PRODUCTS” containing six product cards arranged in two rows of three. Each card includes a placeholder image with a large “X”, a price “VND”, a quantity selector (set to “1”), and an “Add to cart” button.
- Footer:** Includes sections for “CONTACT US” (with icons for phone, email, and location) and “FOLLOW US” (with icons for Facebook, Instagram, and LinkedIn).

Figure 10: C1.6_Wireframe_User session_“Shop” page

“Contact” page

<div style="text-align: center;">  </div>	new Login Register
	
Home About Shop Contact Orders Q   	
<h2>CONTACT US</h2> <p>Home / Contact</p>	
<div style="border: 1px solid #ccc; padding: 10px; width: fit-content; margin: auto;"> <p>PLEASE LEAVE A MESSAGE</p> <input type="text" value="Enter your name"/> <input type="text" value="Enter your email"/> <input type="text" value="Enter your number"/> <input type="text" value="Enter your message"/> <input type="button" value="Send Message"/> </div>	
CONTACT US  +84 912-765-398  Web Developer  Find the nearest Highland Café	FOLLOW US  Facebook  Instagram  LinkedIn

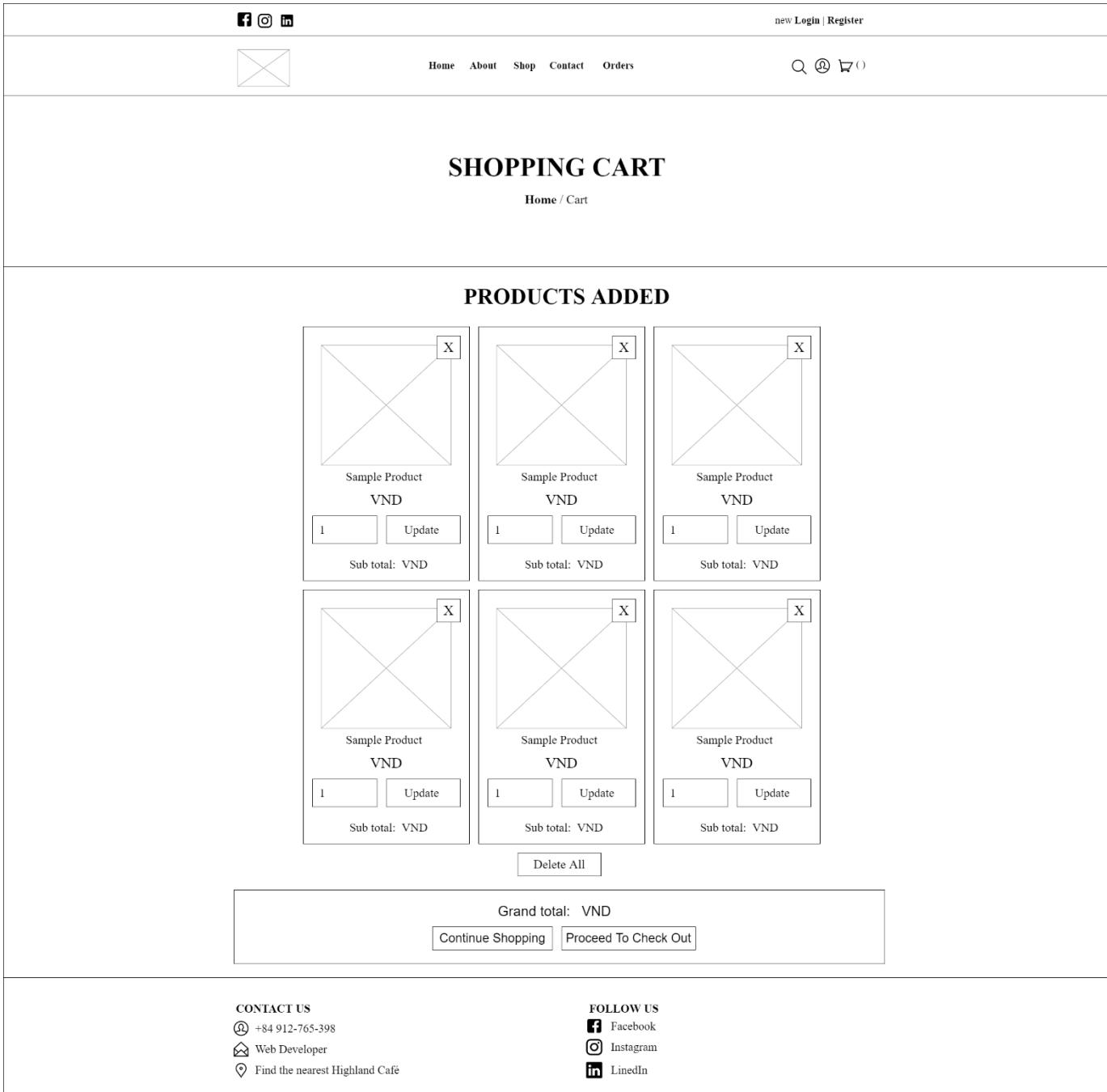
Figure 11: C1.6_Wireframe_User session_ “Contact” page

“Search” page

<div style="text-align: center;">  </div>	new Login Register
	
Home About Shop Contact Orders Q   	
<h2>SEARCH PAGE</h2> <p>Home / Search</p>	
<div style="margin-bottom: 10px;"> <input type="text" value="search products ..."/> Search </div> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin: auto;"> <input type="text" value="Search something!"/> </div>	
CONTACT US  +84 912-765-398  Web Developer  Find the nearest Highland Café	FOLLOW US  Facebook  Instagram  LinkedIn

Figure 12: C1.6_Wireframe_User session_“Search” page

“Cart” page



The wireframe for the “Cart” page is organized into several sections:

- Header:** Features social media icons (Facebook, Instagram, LinkedIn), a search bar, and navigation links for Home, About, Shop, Contact, Orders, and new Login | Register.
- Section Headers:** “SHOPPING CART” and “PRODUCTS ADDED”.
- Product Grid:** A 2x3 grid of product items. Each item is represented by a placeholder image with an “X” button, the text “Sample Product”, “VND”, quantity “1”, “Update” button, and “Sub total: VND”.
- Action Buttons:** A “Delete All” button located below the second row of products.
- Summary Section:** Contains the text “Grand total: VND” and two buttons: “Continue Shopping” and “Proceed To Check Out”.
- Contact Us:** Includes links for “+84 912-765-398”, “Web Developer”, and “Find the nearest Highland Café”.
- Follow Us:** Links to Facebook, Instagram, and LinkedIn.

Figure 13: C1.6_Wireframe_User session_“Cart” page

“Check Out” page

[new Login | Register](#)

[Home](#) [About](#) [Shop](#) [Contact](#) [Orders](#)

CHECK OUT

[Home](#) / Check out

Sample Product (VND X 1)

Grand total: VND

PLACE YOUR ORDER

Your name: <input type="text" value="Enter your name"/>	Your phone number: <input type="text" value="E.g. street name"/>
Your email: <input type="text" value="Enter your email"/>	Payment method: <input type="text" value="Cash on delivery"/> v
Home number: <input type="text" value="E.g. home no."/>	Street name: <input type="text" value="E.g. street name"/>
City: <input type="text" value="E.g. Da Nang"/>	State: <input type="text" value="E.g. Ha Noi"/>
Country: <input type="text" value="E.g. Viet Nam"/>	Pin code: <input type="text" value="E.g. 55000"/>
<input type="button" value="Order Now"/>	

CONTACT US

+84 912-765-398
 Web Developer
 Find the nearest Highland Café

FOLLOW US

Facebook
 Instagram
 LinkedIn

Figure 14: C1.6_Wireframe_User session_“Check Out” page

“Order” page

  	new Login Register									
	Home About Shop Contact Orders    (0)									
<h3>YOUR ORDERS</h3> <p>Home / Orders</p>										
<h3>PLACED ORDERS</h3> <table border="1"><tr><td>Placed on:</td></tr><tr><td>Name:</td></tr><tr><td>Phone number:</td></tr><tr><td>Email:</td></tr><tr><td>Address:</td></tr><tr><td>Payment method:</td></tr><tr><td>Your orders:</td></tr><tr><td>Total price:</td></tr><tr><td>Payment status:</td></tr></table>		Placed on:	Name:	Phone number:	Email:	Address:	Payment method:	Your orders:	Total price:	Payment status:
Placed on:										
Name:										
Phone number:										
Email:										
Address:										
Payment method:										
Your orders:										
Total price:										
Payment status:										
CONTACT US  +84 912-765-398  Web Developer  Find the nearest Highland Café	FOLLOW US  Facebook  Instagram  LinkedIn									

Figure 15: C1.6_Wireframe_User session_ “Order” page

Wireframes for admin session

“Home” page

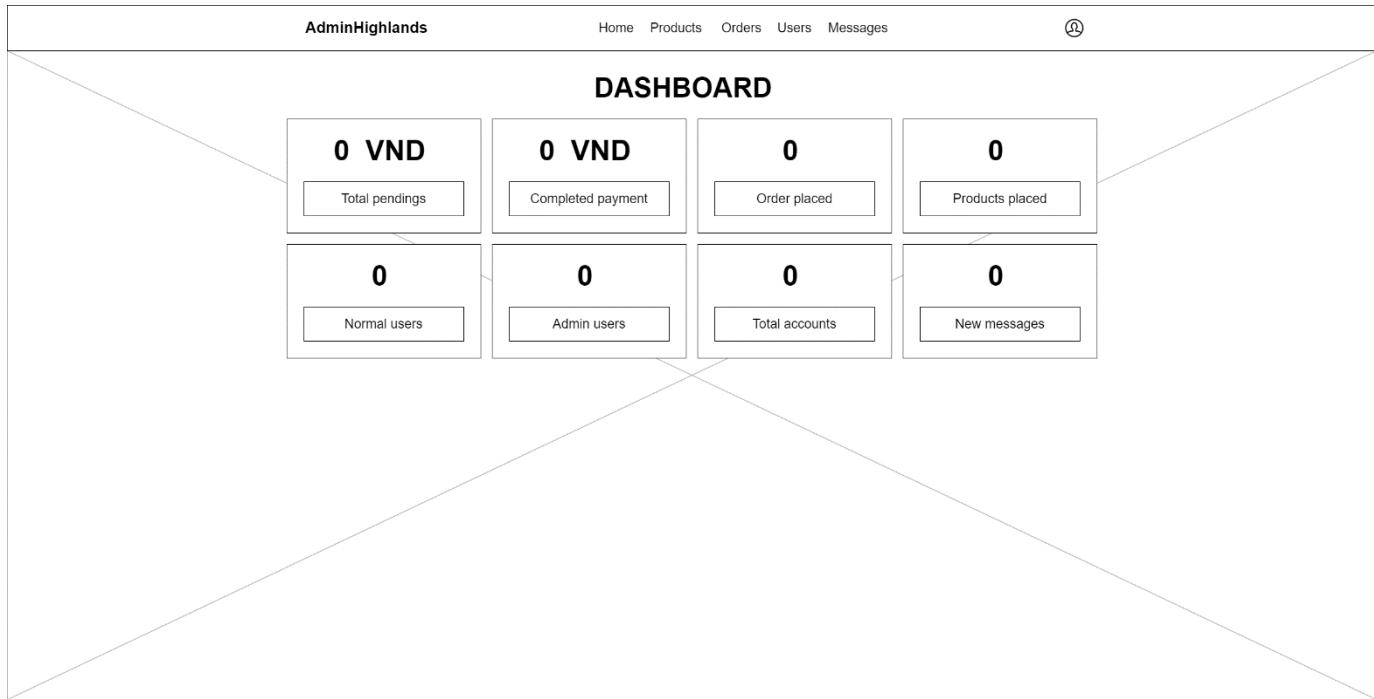
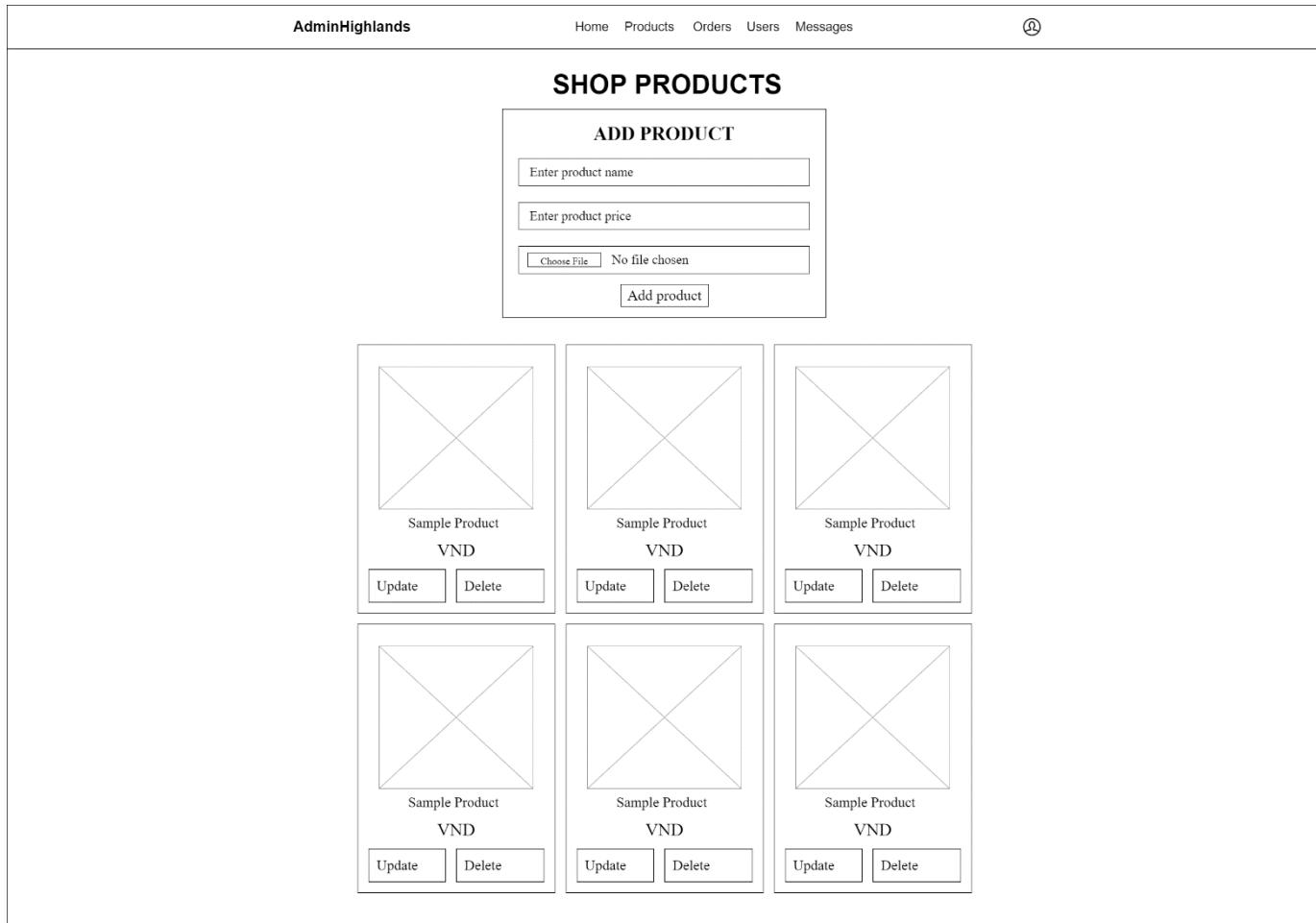


Figure 16: C1.6_Wireframe_Admin session _“Home” page

“Products” page



The wireframe illustrates the 'Products' page structure:

- Header:** AdminHighlands, Home, Products, Orders, Users, Messages, and a user icon.
- Section Header:** SHOP PRODUCTS
- Add Product Form:**
 - Enter product name
 - Enter product price
 - Choose File (No file chosen)
 - Add product button
- Product Grid:** A 2x3 grid of product cards, each featuring a placeholder image (cross), the text "Sample Product", the price "VND", and "Update" and "Delete" buttons.

Figure 17: C1.6_Wireframe_Admin session_“Products” page

“Order” page

AdminHighlands		Home	Products	Orders	Users	Messages	
PLACED ORDERS							
User ID: Placed on: Name: Number: Email: Address: Total products: Total price: Payment method: <input type="button" value="pending"/> <input type="button" value="Update"/> <input type="button" value="Delete"/>							

Figure 18: C1.6_Wireframe_Admin session_ “Order” page

“User” page

AdminHighlands		Home	Products	Orders	Users	Messages	
USER ACCOUNTS							
User ID: Username: Email: User Type: <input type="button" value="Delete User"/>	User ID: Username: Email: User Type: <input type="button" value="Delete User"/>	User ID: Username: Email: User Type: <input type="button" value="Delete User"/>					
User ID: Username: Email: User Type: <input type="button" value="Delete User"/>	User ID: Username: Email: User Type: <input type="button" value="Delete User"/>	User ID: Username: Email: User Type: <input type="button" value="Delete User"/>					

Figure 19: C1.6_Wireframe_Admin session_ “User” page

“Message” page



The wireframe shows a header with 'AdminHighlands' and navigation links for Home, Products, Orders, Users, and Messages. Below the header is a section titled 'MESSAGES' containing three separate message cards. Each card has fields for User ID, Name, Number, Email, and Message, followed by a 'Delete Message' button.

Figure 20: C1.6_Wireframe_Admin session_“Message” page

Chapter 2: Use your design document with appropriate principles, standards and guidelines to produce a branded, multipage website supported with realistic content (P6)

1. Tools and techniques used to design website

Diagram.net

In general, diagrams.net is a free and open-source cross-platform graph drawing software. In this assignment 2, it is used to create diagrams such as sitemap, wireframes and use case diagram (UCD).

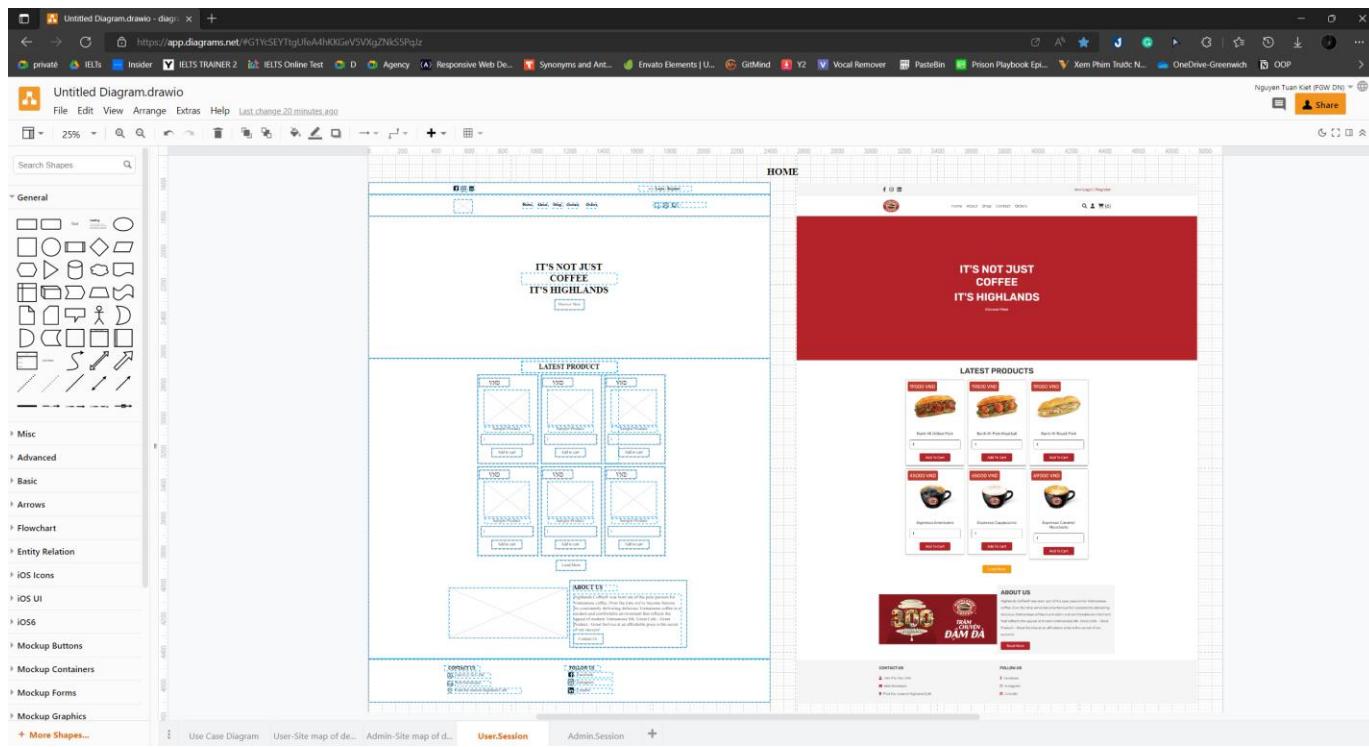
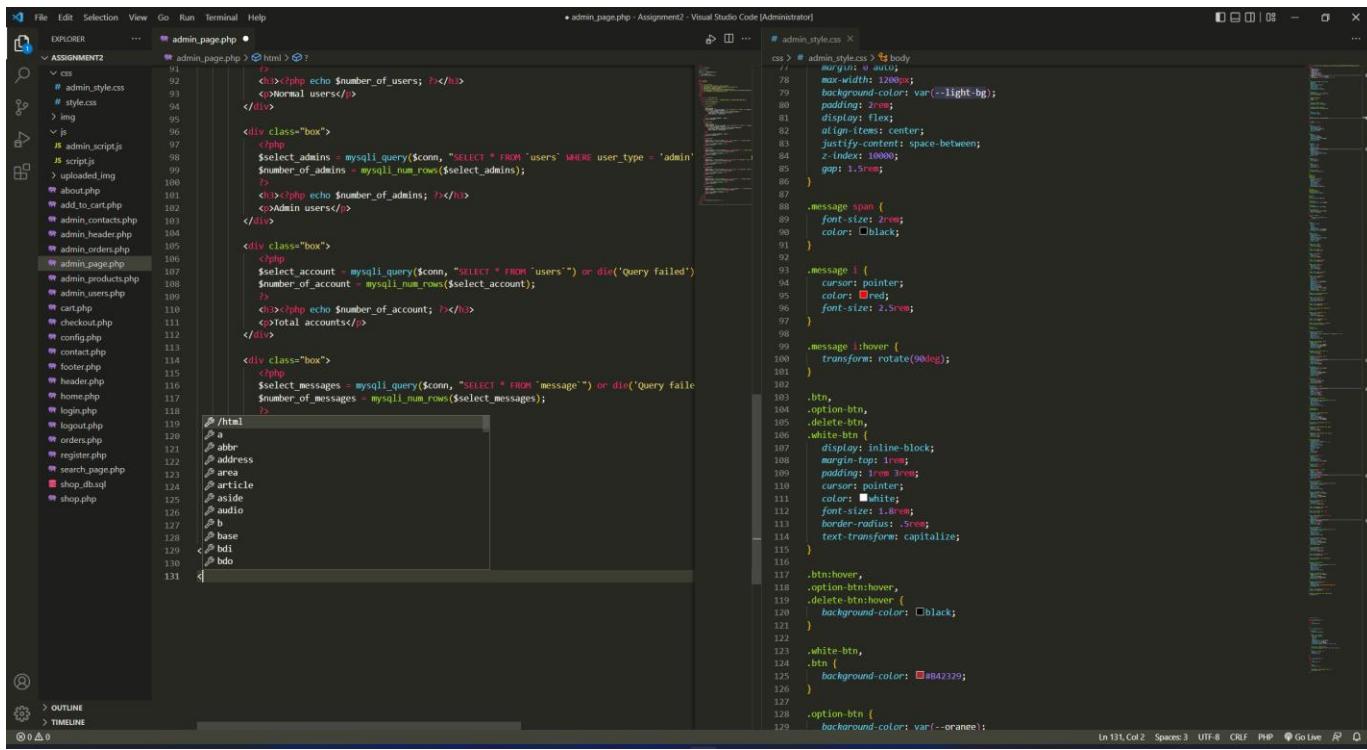


Figure 21: C2.1_Diagram.net_Wireframes

Visual Studio Code

In general, Visual Studio Code is a free coding editor that helps me start coding quickly. Use it to code in any programming language, without switching editors. The reason that I choose to use Visual Studio Code for programming is because Visual Studio Code has support for many programming languages, more specifically in this assignment 2 PHP, JavaScript, and more.

IntelliSense feature is the language service that provides intelligent code completions based on language semantics and an analysis of your source code. If a language service knows possible completions, the IntelliSense suggestions will pop up as you type. If you continue typing characters, the list of members (variables, methods, etc.) is filtered to only include members containing your typed characters. Pressing Tab or Enter will insert the selected member.



The screenshot shows the Visual Studio Code interface with two tabs open: `admin_page.php` and `admin_style.css`. The left pane displays the file structure of the `ASSIGNMENT2` folder, which includes various PHP and CSS files. The right pane shows the code for `admin_page.php` and `admin_style.css`.

```

// admin_page.php - Assignment2 - Visual Studio Code [Administrator]
css > # admin_style.css <body>
  margin: 0 auto;
  max-width: 1200px;
  background-color: var(--light-bg);
  padding: 20px;
  display: flex;
  align-items: center;
  justify-content: space-between;
  z-index: 1000;
  gap: 1.5rem;
}

.message span {
  font-size: 2em;
  color: black;
}

.message i {
  cursor: pointer;
  color: red;
  font-size: 2.5em;
}

.message :hover {
  transform: rotate(9deg);
}

.btn,
.option-btn,
.delete-btn,
.white-btn {
  display: inline-block;
  margin-top: 10px;
  padding: 5px 20px;
  color: white;
  border-radius: 5px;
  font-size: 1.8em;
  border-radius: 5px;
  text-transform: capitalize;
}

.btn:hover,
.option-btn:hover,
.delete-btn:hover {
  background-color: black;
}

.white-btn,
.btn {
  background-color: #8A232B;
}

.option-btn {
  background-color: var(--orange);
}

```

The `admin_page.php` file contains PHP code for displaying user statistics. The `admin_style.css` file defines styles for the application's layout, including a central container with a gap between items and a message component with a red close button.

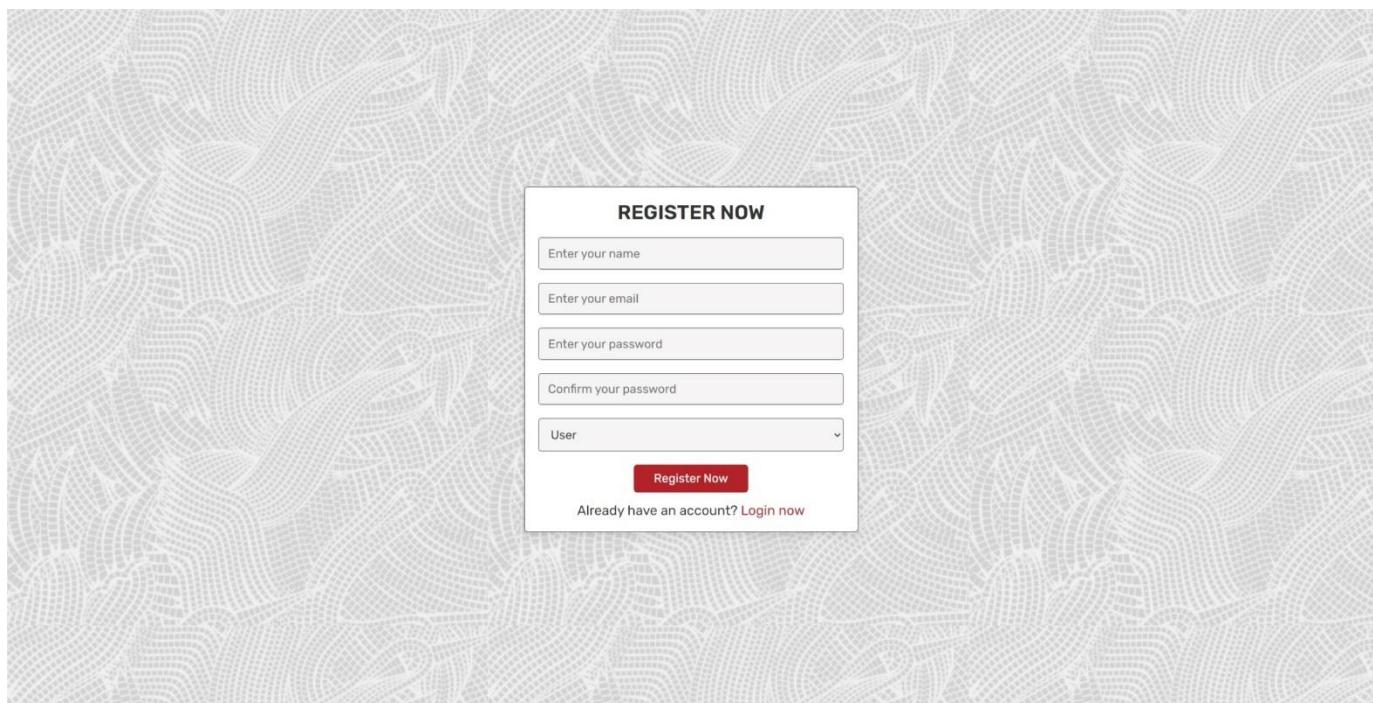
Figure 22: C2.1_Visual Studio Code_IntelliSense feature

2. Screenshots of the webpages and its source code

User session

“Register” page

User Interface of the “Register” page



The screenshot shows a registration form titled "REGISTER NOW". It contains five input fields: "Enter your name", "Enter your email", "Enter your password", and "Confirm your password". Below these is a dropdown menu labeled "User". At the bottom right of the form is a red "Register Now" button and a link "Already have an account? [Login now](#)".

Figure 23: C2.2_User Session_User Interface_“Register” page

Source code of the “Register” page

```
<?php

include 'config.php';
if (isset($_POST['submit'])) {
    $name = mysqli_real_escape_string($conn, $_POST['name']);
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $pass = mysqli_real_escape_string($conn, md5($_POST['password']));
    $cpass = mysqli_real_escape_string($conn, md5($_POST['cpassword']));
    $user_type = $_POST['user_type'];

    $select_users = mysqli_query($conn, "SELECT * FROM `users` WHERE `email` = '$email'");
    or die('Query failed');
    $check_users = mysqli_num_rows($select_users);

    if ($check_users > 0) {
```

```

$message[] = 'User already exist!';
} else {
    if ($pass != $cpass) {
        $message[] = 'Confirm password not matched!';
    } else {
        mysqli_query($conn, "INSERT INTO `users`(`name`, `email`, `password`, `user_type`)
VALUES('$name', '$email', '$cpass', '$user_type')") or die('Query failed');
        $message[] = 'Registered successfully!';
        header('location:login.php');
    }
}
?>
<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Register | Highland Coffee</title>
<link rel="icon" href="img/logo.png" type="image/x-icon">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">
<link rel="stylesheet" href="css/style.css">

</head>

<body>
<?php
if (isset($message)) {
    foreach ($message as $message) {
        echo '
<div class="message">
    <span>' . $message . '</span>
    <i class="fas fa-times" onclick="this.parentElement.remove();"></i>
</div>
';
    }
}
?>

<div class="form-container" style="background-image: url(img/background.page.jpg);">
<form action="" method="post">
    <h3>register now</h3>

```

```

        <input type="text" name="name" placeholder="Enter your name" required class="box">
        <input type="email" name="email" placeholder="Enter your email" required
class="box">
            <input type="password" name="password" placeholder="Enter your password" required
class="box">
                <input type="password" name="cpassword" placeholder="Confirm your password"
required class="box">
                <select name="user_type" class="box">
                    <option value="user">User</option>
                </select>
                <input type="submit" name="submit" value="register now" class="btn">
                <p>Already have an account? <a href="login.php">Login now</a></p>
            </form>
        </div>
    </body>

</html>

```

Figure 24: C2.2_User Session_Source Code_“Register” page

“Log In” page

User Interface of the “Log In” page

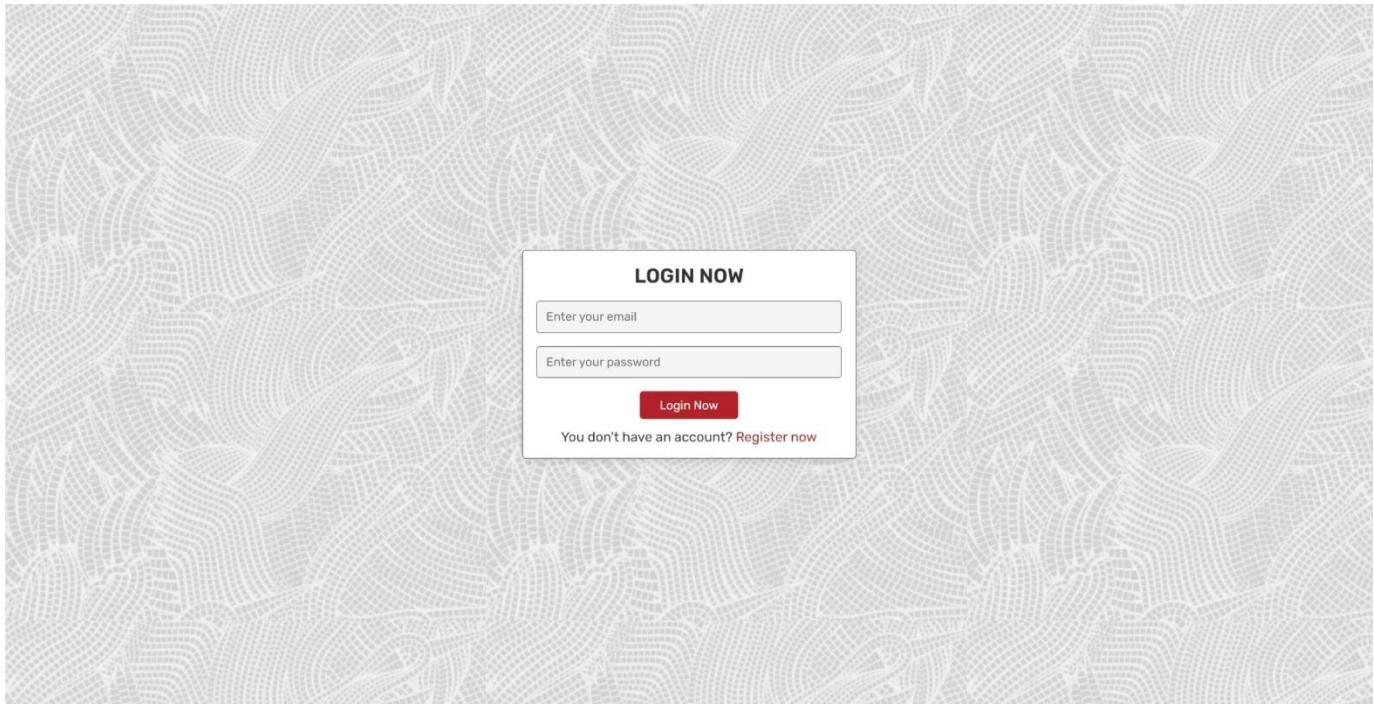


Figure 25: C2.2_User Session_User Interface_“Log In” page

Source code of the “Log In” page

```

<?php

include 'config.php';
session_start();

if (isset($_POST['submit'])) {
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $pass = mysqli_real_escape_string($conn, md5($_POST['password']));

    $select_users = mysqli_query($conn, "SELECT * FROM `users` WHERE email = '$email' AND
password = '$pass'" or die('Query failed'));

    if (mysqli_num_rows($select_users) > 0) {
        $row = mysqli_fetch_assoc($select_users);
        if ($row['user_type'] == 'admin') {
            $_SESSION['admin_name'] = $row['name'];
            $_SESSION['admin_email'] = $row['email'];
            $_SESSION['admin_id'] = $row['id'];
            header('location:admin_page.php');
        } elseif ($row['user_type'] == 'user') {
            $_SESSION['user_name'] = $row['name'];
            $_SESSION['user_email'] = $row['email'];
            $_SESSION['user_id'] = $row['id'];
            header('location:home.php');
        }
    } else {
        $message[] = 'Incorrect email or password!';
    }
}
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login | Highland Coffee</title>
    <link rel="icon" href="img/logo.png" type="image/x-icon">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="css/style.css">
</head>

```

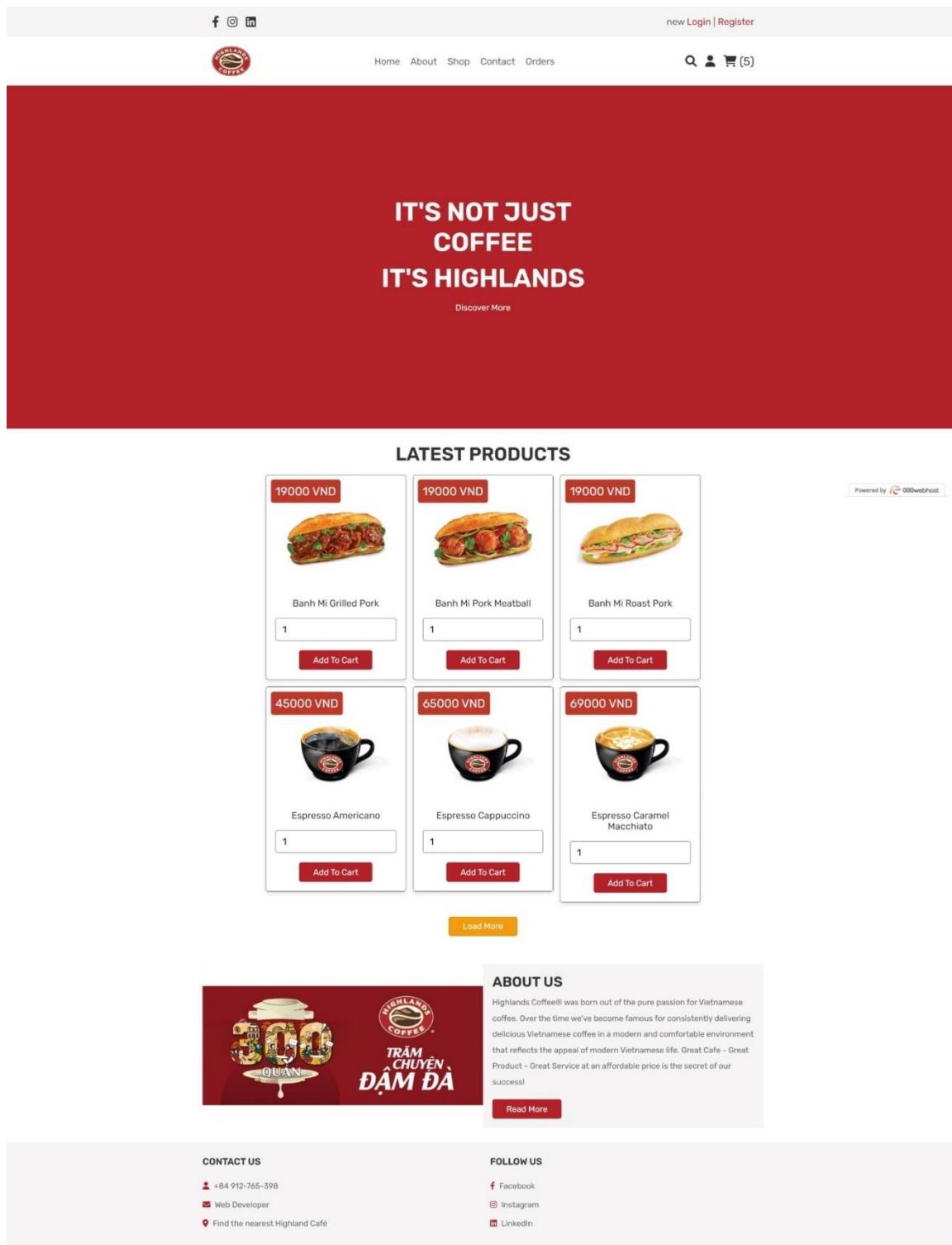
```
<body>
    <?php
    if (isset($message)) {
        foreach ($message as $message) {
            echo '
                <div class="message">
                    <span>' . $message . '</span>
                    <i class="fas fa-times" onclick="this.parentElement.remove();"></i>
                </div>
            ';
        }
    }
    ?>
<div class="form-container" style="background-image: url(img/background.page.jpg);">
    <form action="" method="post">
        <h3>login now</h3>
        <input type="email" name="email" placeholder="Enter your email" required
class="box">
        <input type="password" name="password" placeholder="Enter your password" required
class="box">
        <input type="submit" name="submit" value="login now" class="btn">
        <p>You don't have an account? <a href="register.php">Register now</a></p>
    </form>
</div>
</body>

</html>
```

Figure 26: C2.2_User Session_Source Code_ “Log In” page

“Home” page

User Interface of the “Home” page



new Login | Register

Home About Shop Contact Orders

IT'S NOT JUST
COFFEE
IT'S HIGHLANDS

Discover More

LATEST PRODUCTS

Powered by 

19000 VND  Banh Mi Grilled Pork <input type="text" value="1"/> Add To Cart	19000 VND  Banh Mi Pork Meatball <input type="text" value="1"/> Add To Cart	19000 VND  Banh Mi Roast Pork <input type="text" value="1"/> Add To Cart
45000 VND  Espresso Americano <input type="text" value="1"/> Add To Cart	65000 VND  Espresso Cappuccino <input type="text" value="1"/> Add To Cart	69000 VND  Espresso Caramel Macchiato <input type="text" value="1"/> Add To Cart

Load More

ABOUT US

Highlands Coffee® was born out of the pure passion for Vietnamese coffee. Over the time we've become famous for consistently delivering delicious Vietnamese coffee in a modern and comfortable environment that reflects the appeal of modern Vietnamese life. Great Cafe - Great Product - Great Service at an affordable price is the secret of our success!

[Read More](#)

CONTACT US

- +84 912-765-398
- Web Developer
- Find the nearest Highland Café

FOLLOW US

- Facebook
- Instagram
- LinkedIn

Figure 27: C2.2_User Session_User Interface_“Home” page

Source code of the “Home” page

```

<?php
include 'config.php';
session_start();
$user_id = $_SESSION['user_id'];
if (!isset($user_id)) {
    header('location:login.php');
}
if (isset($_POST['add_to_cart'])) {
    $product_name = $_POST['product_name'];
    $product_price = $_POST['product_price'];
    $product_image = $_POST['product_image'];
    $product_quantity = $_POST['product_quantity'];

    $check_cart_numbers = mysqli_query($conn, "SELECT * FROM `cart` WHERE name =
'$product_name' AND user_id = '$user_id'" ) or die('Query failed');

    if (mysqli_num_rows($check_cart_numbers) > 0) {
        $message[] = 'Already added to cart!';
    } else {
        mysqli_query($conn, "INSERT INTO `cart` (user_id, name, price, quantity, image)
VALUES('$user_id', '$product_name', '$product_price', '$product_quantity',
'$product_image')") or die('Query failed');
        $message[] = 'Product added to cart!';
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home | Highland Coffee</title>
    <link rel="icon" href="img/logo.png" type="image/x-icon">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <?php include 'header.php'; ?>
    <section class="home" style="background-color: #B42329;">
        <div class="content">
            <h3>It's not just Coffee</h3>

```

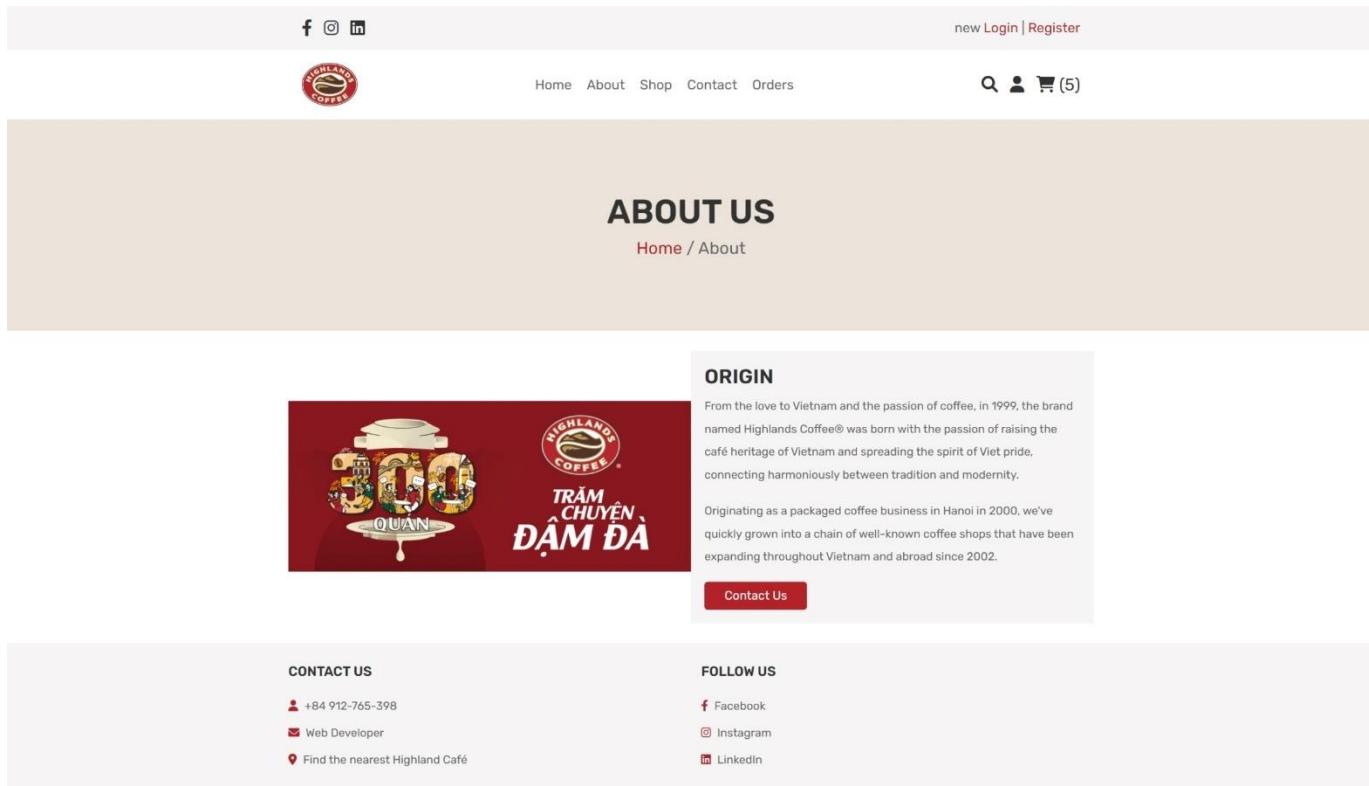
```
<br>
<h3>It's HIGHLANDS</h3>
<a href="about.php" class="white-btn">Discover more</a>
</div>
</section>
<section class="products">
    <h1 class="title">latest products</h1>
    <div class="box-container">
        <?php
            $select_products = mysqli_query($conn, "SELECT * FROM `products` LIMIT 6") or
            die('Query failed');
            if (mysqli_num_rows($select_products) > 0) {
                while ($fetch_products = mysqli_fetch_assoc($select_products)) {
                    ?>
                    <form action="" method="post" class="box">
                        <img class="image" style="max-width: 230px; max-height: 230px;">
                            src="uploaded_img/<?php echo $fetch_products['image']; ?>" alt=""
                        <div class="name">
                            <?php echo $fetch_products['name']; ?>
                        </div>
                        <div class="price">
                            <?php echo $fetch_products['price']; ?> VND
                        </div>
                        <input type="number" min="1" name="product_quantity" value="1"
class="qty">
                            <input type="hidden" name="product_name" value="<?php echo
$fetch_products['name']; ?>">
                            <input type="hidden" name="product_price" value="<?php echo
$fetch_products['price']; ?>">
                            <input type="hidden" name="product_image" value="<?php echo
$fetch_products['image']; ?>">
                            <input type="submit" value="add to cart" name="add_to_cart" class="btn">
                        </form>
                    <?php
                        }
                    } else {
                        echo '<p class="empty">No products added yet!</p>';
                    }
                    ?>
                </div>
                <div class="load-more" style="margin-top: 2rem; text-align:center">
                    <a href="shop.php" class="option-btn">load more</a>
                </div>
            </section>
```

```
<section class="about">
    <div class="flex">
        <div class="image">
            
        </div>
        <div class="content">
            <h3>about us</h3>
            <p>Highlands Coffee® was born out of the pure passion for Vietnamese coffee. Over the time we've become famous for consistently delivering delicious Vietnamese coffee in a modern and comfortable environment that reflects the appeal of modern Vietnamese life. Great Cafe - Great Product - Great Service at an affordable price is the secret of our success!</p>
            <a href="about.php" class="btn">read more</a>
        </div>
    </div>
</section>
<?php include 'footer.php'; ?>
<script src="js/script.js"></script>
</body>
</html>
```

Figure 28: C2.2 _User Session _Source Code _ “Home” page

“About” page

User Interface of the “About” page



The screenshot shows the 'About' page of the Highland Coffee website. At the top, there is a header bar with social media icons (Facebook, Instagram, LinkedIn), a 'new Login | Register' link, the Highland Coffee logo, and a navigation menu with links to Home, About, Shop, Contact, and Orders. A shopping cart icon indicates 5 items. The main content area has a light beige background and features a large title 'ABOUT US' in bold capital letters, followed by a breadcrumb trail 'Home / About'. Below this is a section titled 'ORIGIN' with text about the brand's history and expansion. It includes a graphic showing a traditional Vietnamese coffee shop scene with the text '300 QUAN TRẠM CHUYÊN ĐÂM ĐÀ'. There are also 'Contact Us' and 'Follow Us' sections with links to social media.

Figure 29: C2.2_User Session_User Interface_“About” page

Source code of the “About” page

```
<?php

include 'config.php';

session_start();

$user_id = $_SESSION['user_id'];

if (!isset($user_id)) {
    header('location:login.php');
}

?>

<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>About | Highland Coffee</title>
    <link rel="icon" href="img/logo.png" type="image/x-icon">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="css/style.css">

</head>

<body>
    <?php include 'header.php'; ?>

    <div class="heading">
        <h3>about us</h3>
        <p> <a href="home.php">Home</a> / About </p>
    </div>

    <section class="about">

        <div class="flex">

            <div class="image">
                
            </div>

            <div class="content">
                <h3>ORIGIN</h3>
                <p>From the love to Vietnam and the passion of coffee, in 1999, the brand named Highlands Coffee® was born with the passion of raising the café heritage of Vietnam and spreading the spirit of Viet pride, connecting harmoniously between tradition and modernity.</p>
                <p>Originating as a packaged coffee business in Hanoi in 2000, we've quickly grown into a chain of well-known coffee shops that have been expanding throughout Vietnam and abroad since 2002.</p>
                <a href="contact.php" class="btn">Contact us</a>
            </div>
        </div>
    </section>
```

```
<?php include 'footer.php'; ?>

<script src="js/script.js"></script>

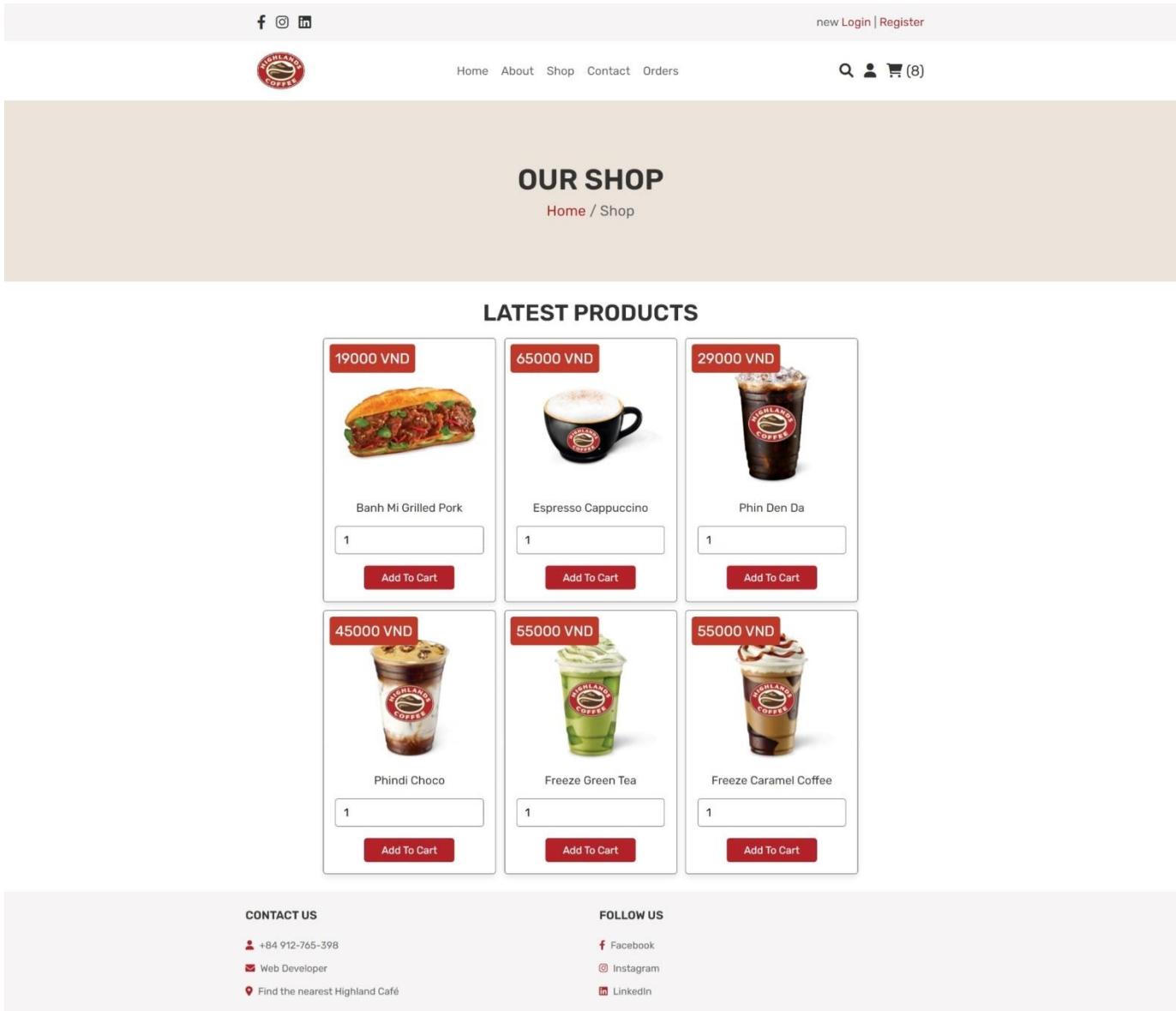
</body>

</html>
```

Figure 30: C2.2_User Session_Source Code_“About” page

“Shop” page

User Interface of the “Shop” page



The screenshot shows the "OUR SHOP" section of the Highland Coffee website. At the top, there are social media links (Facebook, Instagram, LinkedIn) and a navigation bar with Home, About, Shop, Contact, Orders, and a user icon. A search bar and a shopping cart icon (8 items) are also present.

OUR SHOP
[Home](#) / [Shop](#)

LATEST PRODUCTS

Product	Price (VND)	Description	Quantity	Action
Banh Mi Grilled Pork	19000 VND		<input type="text" value="1"/>	Add To Cart
Espresso Cappuccino	65000 VND		<input type="text" value="1"/>	Add To Cart
Phin Den Da	29000 VND		<input type="text" value="1"/>	Add To Cart
Phindi Choco	45000 VND		<input type="text" value="1"/>	Add To Cart
Freeze Green Tea	55000 VND		<input type="text" value="1"/>	Add To Cart
Freeze Caramel Coffee	55000 VND		<input type="text" value="1"/>	Add To Cart

CONTACT US

-  +84 912-765-398
-  Web Developer
-  Find the nearest Highland Café

FOLLOW US

-  Facebook
-  Instagram
-  LinkedIn

Figure 31: C2.2_User Session_User Interface_“Shop” page

Source code of the “Shop” page

```
<?php

include 'config.php';

session_start();

$user_id = $_SESSION['user_id'];
```

```

if (!isset($user_id)) {
    header('location:login.php');
}

if (isset($_POST['add_to_cart'])) {

    $product_name = $_POST['product_name'];
    $product_price = $_POST['product_price'];
    $product_image = $_POST['product_image'];
    $product_quantity = $_POST['product_quantity'];

    $check_cart_numbers = mysqli_query($conn, "SELECT * FROM `cart` WHERE name =
    '$product_name' AND user_id = '$user_id'" or die('Query failed'));

    if (mysqli_num_rows($check_cart_numbers) > 0) {
        $message[] = 'Already added to cart!';
    } else {
        mysqli_query($conn, "INSERT INTO `cart`(user_id, name, price, quantity, image)
VALUES('$user_id', '$product_name', '$product_price', '$product_quantity',
'$product_image')") or die('Query failed');
        $message[] = 'Product added to cart!';
    }
}

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Shop | Highland Coffee</title>
    <link rel="icon" href="img/logo.png" type="image/x-icon">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="css/style.css">

</head>

<body>

    <?php include 'header.php'; ?>

```

```
<div class="heading">
    <h3>our shop</h3>
    <p> <a href="home.php">Home</a> / Shop </p>
</div>

<section class="products">

    <h1 class="title">Latest products</h1>

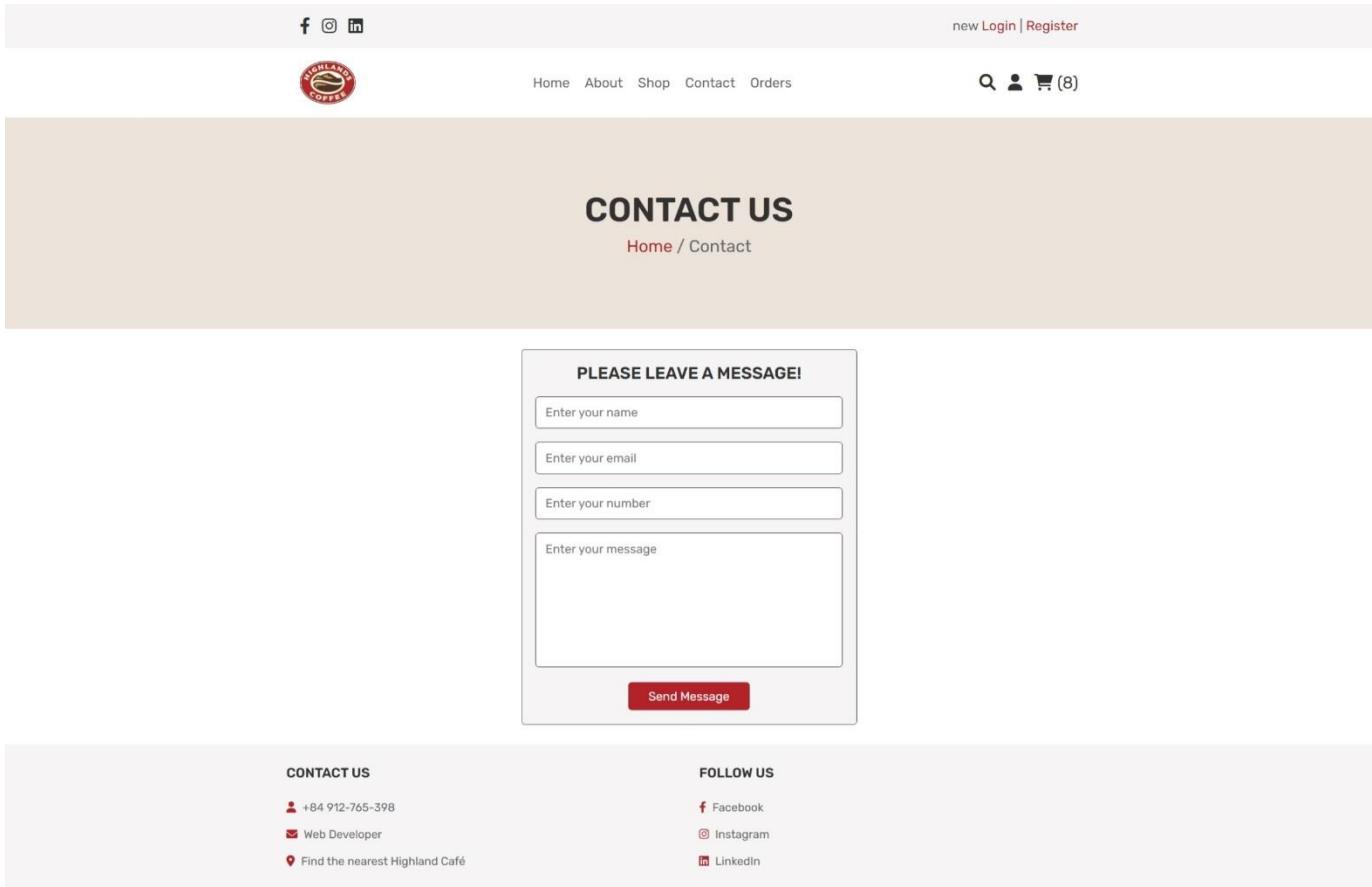
    <div class="box-container">

        <?php
        $select_products = mysqli_query($conn, "SELECT * FROM `products`") or die('Query failed');
        if ($select_products) {
            while ($fetch_products = mysqli_fetch_assoc($select_products)) {
                ?>
                <form action="" method="post" class="box">
                    
                    <div class="name"><?php echo $fetch_products['name']; ?></div>
                    <div class="price"><?php echo $fetch_products['price']; ?> VND</div>
                    <input type="number" min="1" name="product_quantity" value="1" class="qty">
                    <input type="hidden" name="product_name" value="<?php echo $fetch_products['name']; ?>">
                    <input type="hidden" name="product_price" value="<?php echo $fetch_products['price']; ?>">
                    <input type="hidden" name="product_image" value="<?php echo $fetch_products['image']; ?>">
                    <input type="submit" value="add to cart" name="add_to_cart" class="btn">
                </form>
            <?php
            }
        } else {
            echo '<p class="empty">No products added yet!</p>';
        }
        ?>
    </div>
</section>
<?php include 'footer.php'; ?>
<script src="js/script.js"></script>
</body>
</html>
```

Figure 32: C2.2_User Session_Source Code_“Shop” page

“Contact” page

User Interface of the “Contact” page



The screenshot shows the "CONTACT US" page of a website. At the top, there is a header bar with social media icons (Facebook, Instagram, LinkedIn), a "new Login | Register" link, and a logo for "HIGHLAND COFFEE". Below the header, there is a navigation menu with links to Home, About, Shop, Contact, and Orders. On the right side of the header, there is a search icon, a user profile icon, and a shopping cart icon with the number "(8)". The main content area has a light beige background. In the center, the heading "CONTACT US" is displayed in bold capital letters, followed by a breadcrumb trail "Home / Contact". Below this, there is a form titled "PLEASE LEAVE A MESSAGE!" containing four input fields: "Enter your name", "Enter your email", "Enter your number", and a larger "Enter your message" field. A red "Send Message" button is located at the bottom of the form. At the bottom of the page, there are two columns: "CONTACT US" on the left with links to phone number (+84 912-765-398), Web Developer, and Find the nearest Highland Café; and "FOLLOW US" on the right with links to Facebook, Instagram, and LinkedIn.

Figure 33: C2.2_User Session_User Interface_“Contact” page

Source code of the “Contact” page

```
<?php

include 'config.php';

session_start();

$user_id = $_SESSION['user_id'];

if (!isset($user_id)) {
    header('location:login.php');
}
```

```
if (isset($_POST['send'])) {  
  
    $name = mysqli_real_escape_string($conn, $_POST['name']);  
    $email = mysqli_real_escape_string($conn, $_POST['email']);  
    $number = $_POST['number'];  
    $msg = mysqli_real_escape_string($conn, $_POST['message']);  
  
    $select_message = mysqli_query($conn, "SELECT * FROM `message` WHERE name = '$name' AND  
email = '$email' AND number = '$number' AND message = '$msg'") or die('Query failed');  
  
    if (mysqli_num_rows($select_message) > 0) {  
        $message[] = 'Message sent already!';  
    } else {  
        mysqli_query($conn, "INSERT INTO `message` (user_id, name, email, number, message)  
VALUES('$user_id', '$name', '$email', '$number', '$msg')") or die('Query failed');  
        $message[] = 'Message sent successfully!';  
    }  
}  
  
?>  
  
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Contact | Highland Coffee</title>  
    <link rel="icon" href="img/logo.png" type="image/x-icon">  
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">  
    <link rel="stylesheet" href="css/style.css">  
  
</head>  
  
<body>  
    <?php include 'header.php'; ?>  
  
    <div class="heading">  
        <h3>Contact us</h3>  
        <p> <a href="home.php">Home</a> / Contact </p>  
    </div>
```

```
<section class="contact">

    <form action="" method="post">
        <h3>Please leave a message!</h3>
        <input type="text" name="name" required placeholder="Enter your name" class="box">
        <input type="email" name="email" required placeholder="Enter your email" class="box">
        <input type="number" name="number" required placeholder="Enter your number" class="box">
        <textarea name="message" class="box" placeholder="Enter your message" id="" cols="30" rows="10"></textarea>
        <input type="submit" value="send message" name="send" class="btn">
    </form>

</section>

<?php include 'footer.php'; ?>

<script src="js/script.js"></script>

</body>

</html>
```

Figure 34: C2.2_User Session_Source Code_“Contact” page

“Search” page

User Interface of the “Search” page

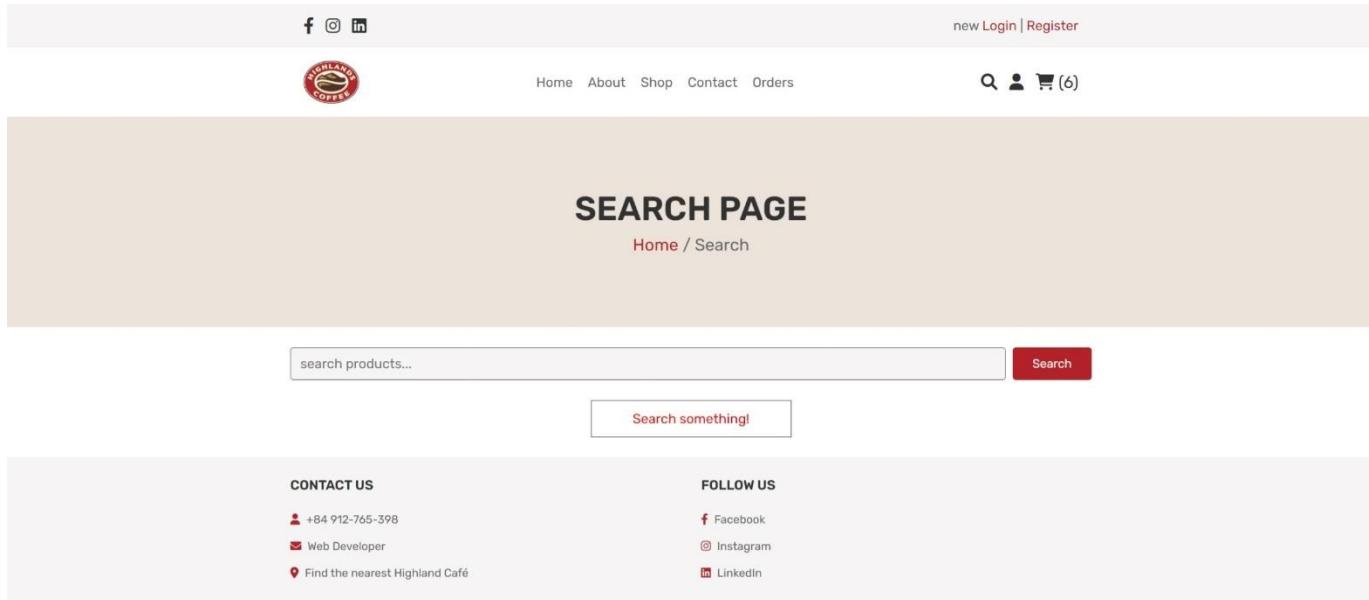


Figure 35: C2.2_User Session_User Interface_“Search” page

Source code of the “Search” page

```
<?php

include 'config.php';

session_start();

$user_id = $_SESSION['user_id'];

if (!isset($user_id)) {
    header('location:login.php');
};

if (isset($_POST['add_to_cart'])) {

    $product_name = $_POST['product_name'];
    $product_price = $_POST['product_price'];
    $product_image = $_POST['product_image'];
    $product_quantity = $_POST['product_quantity'];
}
```

```
$check_cart_numbers = mysqli_query($conn, "SELECT * FROM `cart` WHERE name =
'$product_name' AND user_id = '$user_id'" ) or die('Query failed');

if (mysqli_num_rows($check_cart_numbers) > 0) {
    $message[] = 'Already added to cart!';
} else {
    mysqli_query($conn, "INSERT INTO `cart` (user_id, name, price, quantity, image)
VALUES('$user_id', '$product_name', '$product_price', '$product_quantity',
'$product_image')") or die('Query failed');
    $message[] = 'Product added to cart!';
}
};

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Search page | Highland Coffee</title>
    <link rel="icon" href="img/logo.png" type="image/x-icon">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="css/style.css">

</head>

<body>

<?php include 'header.php'; ?>

<div class="heading">
    <h3>search page</h3>
    <p> <a href="home.php">Home</a> / Search </p>
</div>

<section class="search-form">
    <form action="" method="post">
        <input type="text" name="search" placeholder="search products..." class="box">
        <input type="submit" name="submit" value="search" class="btn">
    </form>
</section>
```

```
<section class="products" style="padding-top: 0;">

    <div class="box-container">
        <?php
            if (isset($_POST['submit'])) {
                $search_item = $_POST['search'];
                $select_products = mysqli_query($conn, "SELECT * FROM `products` WHERE name LIKE
                    '%{$search_item}%'") or die('Query failed');
                if (mysqli_num_rows($select_products) > 0) {
                    while ($fetch_product = mysqli_fetch_assoc($select_products)) {
                        ?>
                        <form action="" method="post" class="box">
                            
                                <div class="name"><?php echo $fetch_product['name']; ?></div>
                                <div class="price"><?php echo $fetch_product['price']; ?> VND</div>
                                <input type="number" class="qty" name="product_quantity" min="1"
                                    value="1">
                                <input type="hidden" name="product_name" value="<?php echo
                                    $fetch_product['name']; ?>">
                                <input type="hidden" name="product_price" value="<?php echo
                                    $fetch_product['price']; ?>">
                                <input type="hidden" name="product_image" value="<?php echo
                                    $fetch_product['image']; ?>">
                                <input type="submit" class="btn" value="add to cart"
                                    name="add_to_cart">
                            </form>
                        <?php
                            }
                        } else {
                            echo '<p class="empty">No result found!</p>';
                        }
                    } else {
                        echo '<p class="empty">Search something!</p>';
                    }
                ?>
            </div>
        </section>

        <?php include 'footer.php'; ?>

        <script src="js/script.js"></script>

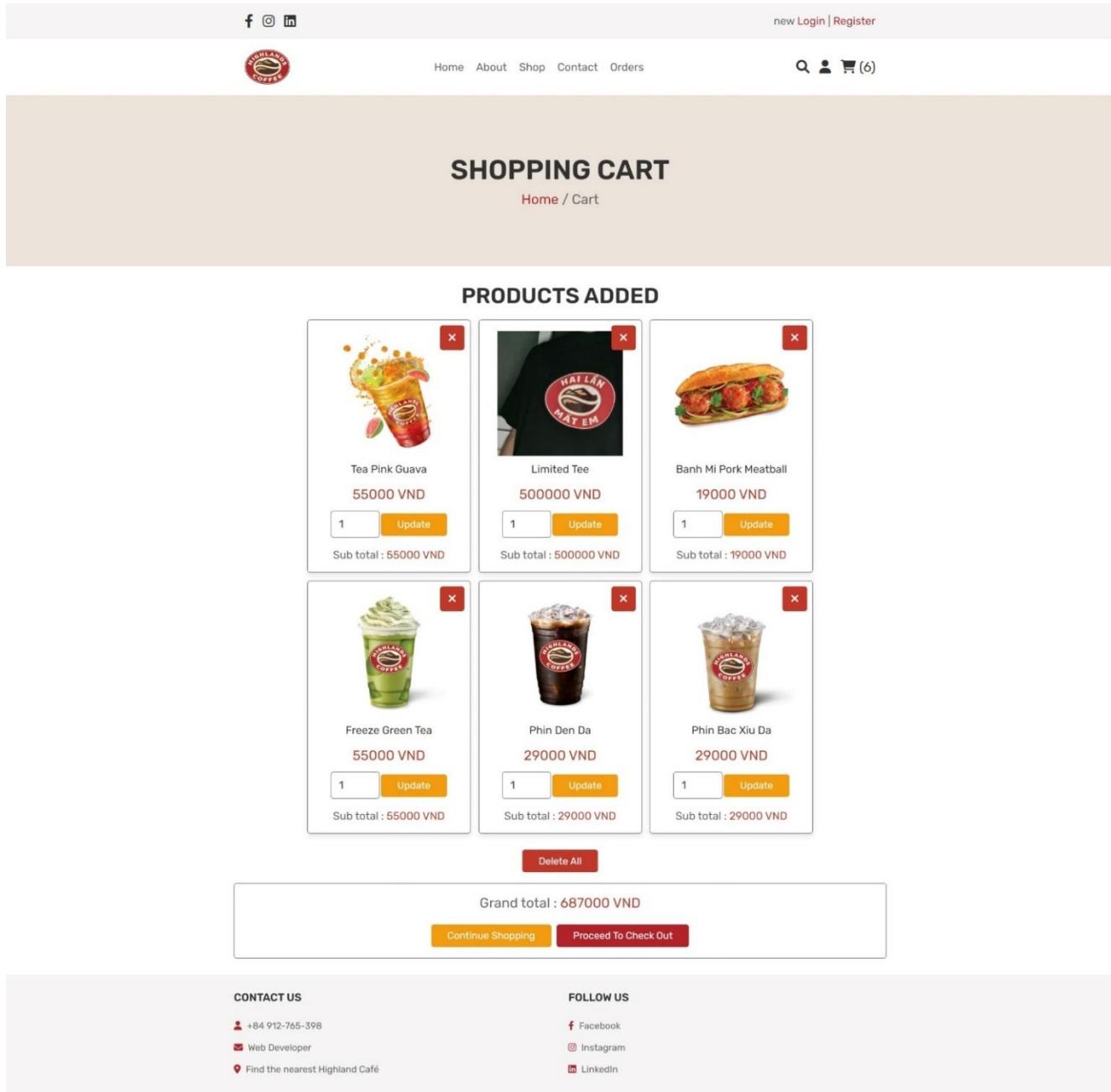
    </body>
```

```
</html>
```

Figure 36: C2.2_User Session_Source Code_“Search” page

“Cart” page

User Interface of the “Cart” page



The screenshot displays the user interface of the "Cart" page for Highland Coffee. At the top, there is a header bar with social media icons (Facebook, Instagram, LinkedIn), a login/register link, and a navigation menu (Home, About, Shop, Contact, Orders). A search bar and a user profile icon are also present.

The main content area is titled "SHOPPING CART" and shows the breadcrumb "Home / Cart". Below this, a section titled "PRODUCTS ADDED" lists six items:

- Tea Pink Guava: 55000 VND (Sub total: 55000 VND)
- Limited Tee: 500000 VND (Sub total: 500000 VND)
- Banh Mi Pork Meatball: 19000 VND (Sub total: 19000 VND)
- Freeze Green Tea: 55000 VND (Sub total: 55000 VND)
- Phin Den Da: 29000 VND (Sub total: 29000 VND)
- Phin Bac Xiu Da: 29000 VND (Sub total: 29000 VND)

Each item has a red "X" button to delete it. Below the products, a "Delete All" button is visible. At the bottom, a summary box shows the "Grand total: 687000 VND" and two buttons: "Continue Shopping" and "Proceed To Check Out".

At the very bottom, there are sections for "CONTACT US" (with links to phone number, email, and location) and "FOLLOW US" (with links to Facebook, Instagram, and LinkedIn).

Figure 37: C2.2_User Session_User Interface_“Cart” page

Source code of the “Cart” page

```

<?php

include 'config.php';

session_start();

$user_id = $_SESSION['user_id'];

if (!isset($user_id)) {
    header('location:login.php');
}

if (isset($_POST['update_cart'])) {
    $cart_id = $_POST['cart_id'];
    $cart_quantity = $_POST['cart_quantity'];
    mysqli_query($conn, "UPDATE `cart` SET quantity = '$cart_quantity' WHERE id =
    '$cart_id'" ) or die('Query failed');
    $message[] = 'Cart quantity updated!';
}

if (isset($_GET['delete'])) {
    $delete_id = $_GET['delete'];
    mysqli_query($conn, "DELETE FROM `cart` WHERE id = '$delete_id'" ) or die('Query failed');
    header('location:cart.php');
}

if (isset($_GET['delete_all'])) {
    mysqli_query($conn, "DELETE FROM `cart` WHERE user_id = '$user_id'" ) or die('Query
failed');
    header('location:cart.php');
}

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cart | Highland Coffee</title>
    <link rel="icon" href="img/logo.png" type="image/x-icon">

```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
<link rel="stylesheet" href="css/style.css">

</head>

<body>

<?php include 'header.php'; ?>

<div class="heading">
    <h3>shopping cart</h3>
    <p> <a href="home.php">Home</a> / Cart </p>
</div>

<section class="shopping-cart">

    <h1 class="title">products added</h1>

    <div class="box-container">
        <?php
            $grand_total = 0;
            $select_cart = mysqli_query($conn, "SELECT * FROM `cart` WHERE user_id = '$user_id'" ) or die('Query failed');
            if (mysqli_num_rows($select_cart) > 0) {
                while ($fetch_cart = mysqli_fetch_assoc($select_cart)) {
                    ?
                    <div class="box">
                        <a href="cart.php?delete=<?php echo $fetch_cart['id']; ?>" class="fas fa-times" onclick="return confirm('Delete this from cart?');"></a>
                        
                        <div class="name"><?php echo $fetch_cart['name']; ?></div>
                        <div class="price"><?php echo $fetch_cart['price']; ?> VND</div>
                        <form action="" method="post">
                            <input type="hidden" name="cart_id" value="<?php echo $fetch_cart['id']; ?>">
                            <input type="number" min="1" name="cart_quantity" value="<?php echo $fetch_cart['quantity']; ?>">
                            <input type="submit" name="update_cart" value="update" class="option-btn">
                        </form>
                        <div class="sub-total"> Sub total : <span><?php echo $sub_total = ($fetch_cart['quantity'] * $fetch_cart['price']); ?> VND</span> </div>
                    </div>
                }
            }
        </?php>
    </div>
</section>
```

```
<?php
    $grand_total += $sub_total;
}
} else {
    echo '<p class="empty">Your cart is empty</p>';
}
?>
</div>

<div style="margin-top: 2rem; text-align:center;">
    <a href="cart.php?delete_all" class="delete-btn <?php echo ($grand_total > 1) ? '' : 'disabled'; ?>" onclick="return confirm('Delete all from cart?');">Delete all</a>
</div>

<div class="cart-total">
    <p>Grand total : <span><?php echo $grand_total; ?> VND</span></p>
    <div class="flex">
        <a href="shop.php" class="option-btn">Continue shopping</a>
        <a href="checkout.php" class="btn <?php echo ($grand_total > 1) ? '' : 'disabled'; ?>">Proceed to check out</a>
    </div>
</div>

</section>

<?php include 'footer.php'; ?>

<script src="js/script.js"></script>

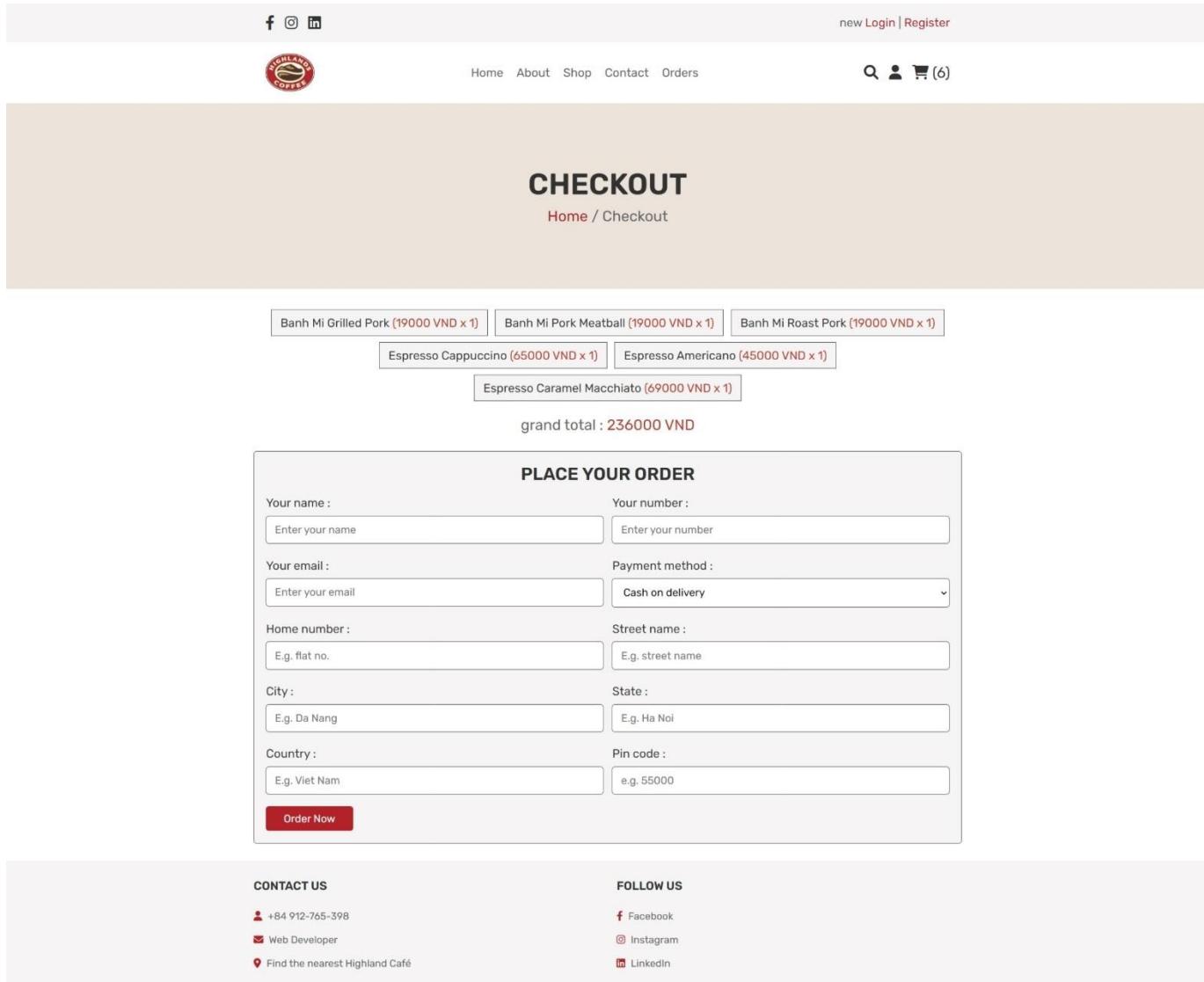
</body>

</html>
```

Figure 38: C2.2_User Session_Source Code_“Cart” page

“Check Out” page

User Interface of the “Check Out” page



The screenshot shows the Highland Coffee website's checkout page. At the top, there are social media links (Facebook, Instagram, LinkedIn) and a login/register link. The header includes the Highland Coffee logo and navigation links for Home, About, Shop, Contact, and Orders. A shopping cart icon indicates 6 items. The main content area has a light beige background. The title "CHECKOUT" is centered above a breadcrumb trail "Home / Checkout". Below this, a summary table lists six items with their names, prices, and quantities. The total grand total is displayed as "grand total : 236000 VND". A large "PLACE YOUR ORDER" form follows, containing fields for personal information and delivery details. At the bottom, there are sections for "CONTACT US" (with phone number, developer info, and nearest cafe location) and "FOLLOW US" (with links to Facebook, Instagram, and LinkedIn).

Figure 39: C2.2_User Session_User Interface_“Check Out” page

Source code of the “Check Out” page

```
<?php

include 'config.php';

session_start();

$user_id = $_SESSION['user_id'];
```

```

if (!isset($user_id)) {
    header('location:login.php');
}

if (isset($_POST['order_btn'])) {

    $name = mysqli_real_escape_string($conn, $_POST['name']);
    $number = $_POST['number'];
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $method = mysqli_real_escape_string($conn, $_POST['method']);
    $address = mysqli_real_escape_string($conn, 'flat no. ' . $_POST['flat'] . ', ' .
    $_POST['street'] . ', ' . $_POST['city'] . ', ' . $_POST['country'] . ' - ' .
    $_POST['pin_code']);
    $placed_on = date('d-M-Y');

    $cart_total = 0;
    $cart_products[] = '';

    $cart_query = mysqli_query($conn, "SELECT * FROM `cart` WHERE user_id = '$user_id'") or
die('Query failed');
    if (mysqli_num_rows($cart_query) > 0) {
        while ($cart_item = mysqli_fetch_assoc($cart_query)) {
            $cart_products[] = $cart_item['name'] . ' (' . $cart_item['quantity'] . ')';
            $sub_total = ($cart_item['price'] * $cart_item['quantity']);
            $cart_total += $sub_total;
        }
    }

    $total_products = implode(', ', $cart_products);

    $order_query = mysqli_query($conn, "SELECT * FROM `orders` WHERE name = '$name' AND
number = '$number' AND email = '$email' AND method = '$method' AND address = '$address' AND
total_products = '$total_products' AND total_price = '$cart_total'") or die('Query failed');

    if ($cart_total == 0) {
        $message[] = 'Your cart is empty';
    } else {
        if (mysqli_num_rows($order_query) > 0) {
            $message[] = 'Order already placed!';
        } else {
            mysqli_query($conn, "INSERT INTO `orders`(user_id, name, number, email, method,
address, total_products, total_price, placed_on) VALUES('$user_id', '$name', '$number',
'$email', '$method', '$address', '$total_products', '$cart_total', '$placed_on')") or
die('Query failed');
            $message[] = 'Order placed successfully!';
        }
    }
}

```

```
        mysqli_query($conn, "DELETE FROM `cart` WHERE user_id = '$user_id'") or die('Query failed');
    }
}
}

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Check out | Highland Coffee</title>
    <link rel="icon" href="img/logo.png" type="image/x-icon">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="css/style.css">

</head>

<body>

<?php include 'header.php'; ?>

<div class="heading">
    <h3>checkout</h3>
    <p> <a href="home.php">Home</a> / Checkout </p>
</div>

<section class="display-order">

    <?php
    $grand_total = 0;
    $select_cart = mysqli_query($conn, "SELECT * FROM `cart` WHERE user_id = '$user_id'") or die('Query failed');
    if ($select_num_rows($select_cart) > 0) {
        while ($fetch_cart = mysqli_fetch_assoc($select_cart)) {
            $total_price = ($fetch_cart['price'] * $fetch_cart['quantity']);
            $grand_total += $total_price;
        }
        <p> <?php echo $fetch_cart['name']; ?> <span>(<?php echo $fetch_cart['price'] . ' VND' . ' x ' . $fetch_cart['quantity']; ?>)</span> </p>
    }
}>
```

```

<?php
    }
} else {
    echo '<p class="empty">Your cart is empty</p>';
}
?>
<div class="grand-total"> grand total : <span> <?php echo $grand_total; ?> VND</span>
</div>

</section>

<section class="checkout">

<form action="" method="post">
    <h3>place your order</h3>
    <div class="flex">
        <div class="inputBox">
            <span>Your name :</span>
            <input type="text" name="name" required placeholder="Enter your name">
        </div>
        <div class="inputBox">
            <span>Your number :</span>
            <input type="number" name="number" required placeholder="Enter your number">
        </div>
        <div class="inputBox">
            <span>Your email :</span>
            <input type="email" name="email" required placeholder="Enter your email">
        </div>
        <div class="inputBox">
            <span>Payment method :</span>
            <select name="method">
                <option value="cash on delivery">Cash on delivery</option>
                <option value="credit card">Credit card</option>
                <option value="paypal">Paypal</option>
            </select>
        </div>
        <div class="inputBox">
            <span>Address line 01 :</span>
            <input type="number" min="0" name="flat" required placeholder="E.g. flat
no.">
        </div>
        <div class="inputBox">
            <span>Address line 01 :</span>
            <input type="text" name="street" required placeholder="E.g. street name">
        </div>

```

```
<div class="inputBox">
    <span>City :</span>
    <input type="text" name="city" required placeholder="E.g. Da Nang">
</div>
<div class="inputBox">
    <span>State :</span>
    <input type="text" name="state" required placeholder="E.g. Ha Noi">
</div>
<div class="inputBox">
    <span>Country :</span>
    <input type="text" name="country" required placeholder="E.g. Viet Nam">
</div>
<div class="inputBox">
    <span>Pin code :</span>
    <input type="number" min="0" name="pin_code" required placeholder="e.g.
55000">
</div>
</div>
<input type="submit" value="order now" class="btn" name="order_btn">
</form>

</section>

<?php include 'footer.php'; ?>

<script src="js/script.js"></script>

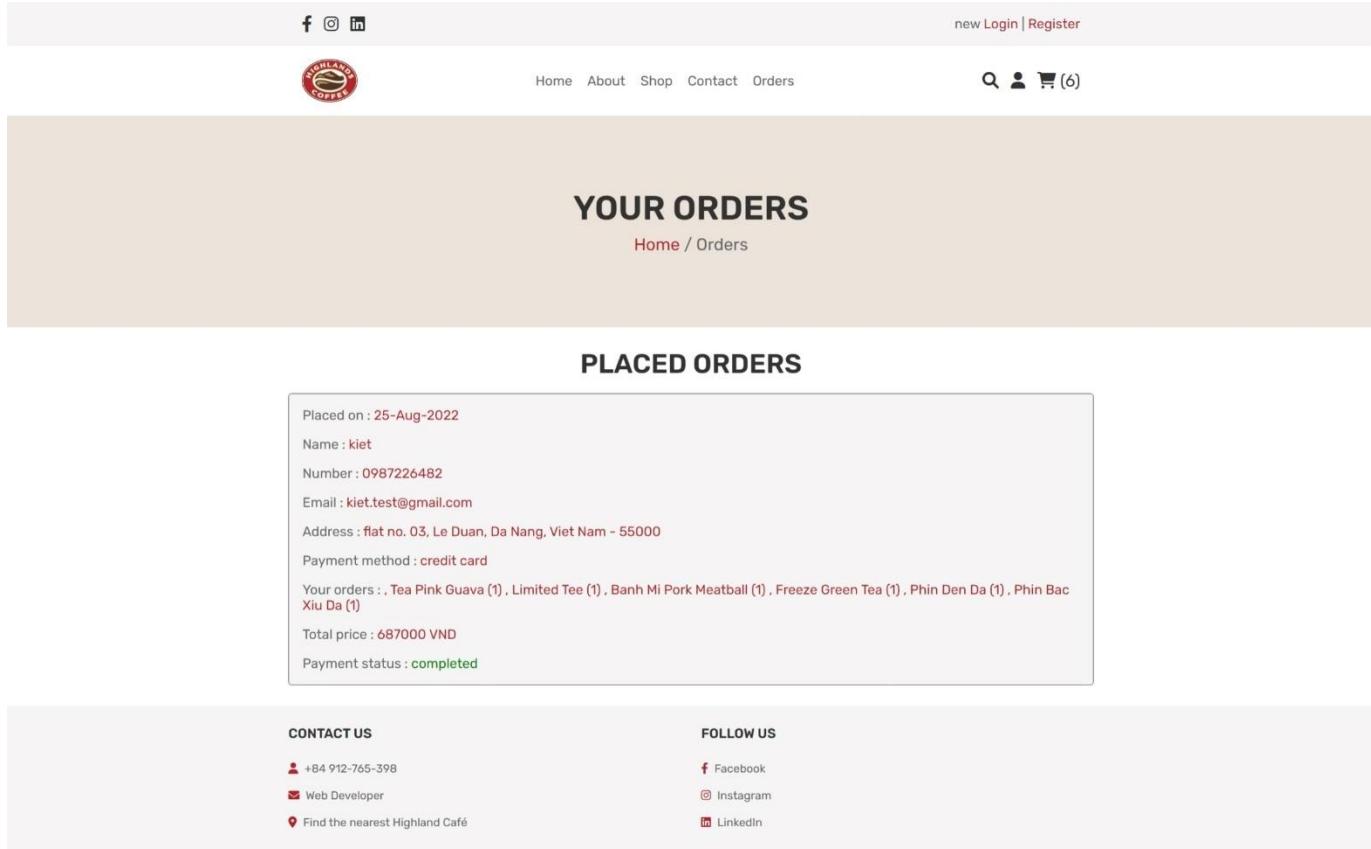
</body>

</html>
```

Figure 40: C2.2_User Session_Source Code_ “Check Out” page

“Order” page

User Interface of the “Order” page



The screenshot shows the 'YOUR ORDERS' section of a website. At the top, there are social media icons (Facebook, Instagram, LinkedIn) and a 'new Login | Register' link. Below the header is a navigation bar with links to Home, About, Shop, Contact, Orders, and a search bar with a user icon and a cart icon showing '(6)'.

The main content area is titled 'YOUR ORDERS' and shows a breadcrumb trail: Home / Orders. Below this, a section titled 'PLACED ORDERS' displays the details of a single order:

- Placed on : 25-Aug-2022
- Name : kiet
- Number : 0987226482
- Email : kiet.test@gmail.com
- Address : flat no. 03, Le Duan, Da Nang, Viet Nam - 55000
- Payment method : credit card
- Your orders : , Tea Pink Guava (1) , Limited Tee (1) , Banh Mi Pork Meatball (1) , Freeze Green Tea (1) , Phin Den Da (1) , Phin Bac Xiu Da (1)
- Total price : 687000 VND
- Payment status : completed

At the bottom, there are sections for 'CONTACT US' (with links to phone number, email, and location) and 'FOLLOW US' (with links to Facebook, Instagram, and LinkedIn).

Figure 41: C2.2_User Session_User Interface_“Order” page

Source code of the “Order” page

```
<?php

include 'config.php';

session_start();

$user_id = $_SESSION['user_id'];

if (!isset($user_id)) {
    header('location:login.php');
}

?>

<!DOCTYPE html>
```

```

<html lang="en">

    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Orders | Highland Coffee</title>
        <link rel="icon" href="img/logo.png" type="image/x-icon">
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
        <link rel="stylesheet" href="css/style.css">

    </head>

    <body>

        <?php include 'header.php'; ?>

        <div class="heading">
            <h3>your orders</h3>
            <p> <a href="home.php">Home</a> / Orders </p>
        </div>

        <section class="placed-orders">

            <h1 class="title">Placed orders</h1>

            <div class="box-container">

                <?php
                    $order_query = mysqli_query($conn, "SELECT * FROM `orders` WHERE user_id = '$user_id'" ) or die('Query failed');
                    if (mysqli_num_rows($order_query) > 0) {
                        while ($fetch_orders = mysqli_fetch_assoc($order_query)) {
                ?>
                        <div class="box">
                            <p> Placed on : <span><?php echo $fetch_orders['placed_on']; ?></span>
                </p>
                            <p> Name : <span><?php echo $fetch_orders['name']; ?></span> </p>
                            <p> Number : <span><?php echo $fetch_orders['number']; ?></span> </p>
                            <p> Email : <span><?php echo $fetch_orders['email']; ?></span> </p>
                            <p> Address : <span><?php echo $fetch_orders['address']; ?></span> </p>
                            <p> Payment method : <span><?php echo $fetch_orders['method']; ?></span>
                </p>
            
```

```
<p> Your orders : <span><?php echo $fetch_orders['total_products']; ?></span> </p>
<p> Total price : <span><?php echo $fetch_orders['total_price']; ?> VND</span> </p>
<p> Payment status : <span style="color:<?php if
($fetch_orders['payment_status'] == 'pending') {
                                echo 'red';
} else {
                                echo 'green';
} ?>;"><?php echo
$fetch_orders['payment_status']; ?></span> </p>
</div>
<?php
    }
} else {
    echo '<p class="empty">No orders placed yet!</p>';
}
?>
</div>

</section>
<?php include 'footer.php'; ?>

<!-- custom js file link --&gt;
&lt;script src="js/script.js"&gt;&lt;/script&gt;

&lt;/body&gt;

&lt;/html&gt;</pre>
```

Figure 42: C2.2_User Session_Source Code_“Order” page

Admin session

“Home” page

User Interface of the “Home” page

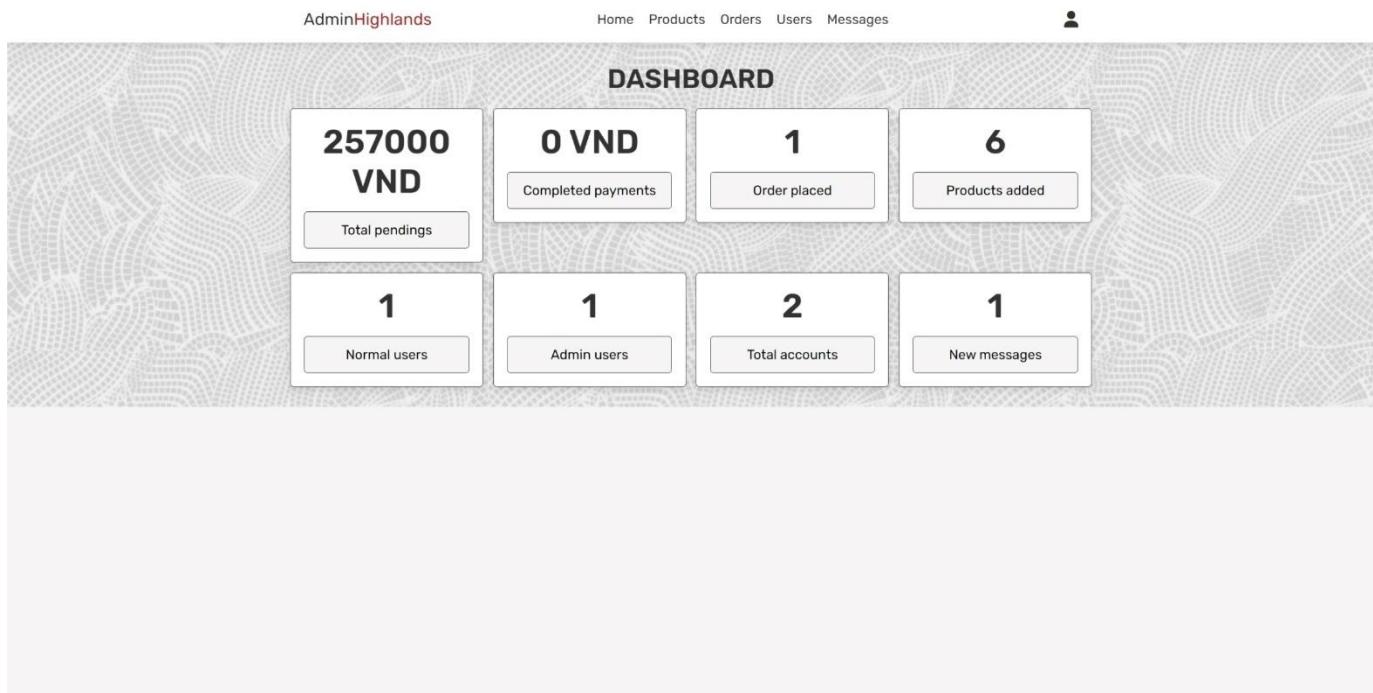


Figure 43: C2.2_Admin Session_User Interface_“Home” page

Source code of the “Home” page

```
<?php

include 'config.php';

session_start();

$admin_id = $_SESSION['admin_id'];

if (!isset($admin_id)) {
    header('location:login.php');
}

?>

<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin.Panel | Highlands Coffee</title>
    <link rel="icon" href="img/logo.admin.png" type="image/x-icon">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="css/admin_style.css">

</head>

<body>

    <?php include 'admin_header.php'; ?>

    <section class="dashboard" style="background-image: url(img/background.page.jpg);">

        <h1 class="title">dashboard</h1>

        <div class="box-container">

            <div class="box">
                <?php
                    $total_pendings = 0;
                    $select_pending = mysqli_query($conn, "SELECT total_price FROM `orders` WHERE payment_status = 'pending'") or die('Query failed');
                    if (mysqli_num_rows($select_pending) > 0) {
                        while ($fetch_pendings = mysqli_fetch_assoc($select_pending)) {
                            $total_price = $fetch_pendings['total_price'];
                            $total_pendings += $total_price;
                        };
                    };
                ?>
                <h3><?php echo $total_pendings; ?> VND</h3>
                <p>Total pendings</p>
            </div>

            <div class="box">
                <?php
                    $total_completed = 0;
                    $select_completed = mysqli_query($conn, "SELECT total_price FROM `orders` WHERE payment_status = 'completed'") or die('Query failed');
                    if (mysqli_num_rows($select_completed) > 0) {
                        while ($fetch_completed = mysqli_fetch_assoc($select_completed)) {
```

```
        $total_price = $fetch_completed['total_price'];
        $total_completed += $total_price;
    };
};

?>
<h3><?php echo $total_completed; ?> VND</h3>
<p>Completed payments</p>
</div>

<div class="box">
<?php
$select_orders = mysqli_query($conn, "SELECT * FROM `orders`") or die('Query
failed');
$number_of_orders = mysqli_num_rows($select_orders);
?>
<h3><?php echo $number_of_orders; ?></h3>
<p>Order placed</p>
</div>

<div class="box">
<?php
$select_products = mysqli_query($conn, "SELECT * FROM `products`") or die('Query
failed');
$number_of_products = mysqli_num_rows($select_products);
?>
<h3><?php echo $number_of_products; ?></h3>
<p>Products added</p>
</div>

<div class="box">
<?php
$select_users = mysqli_query($conn, "SELECT * FROM `users` WHERE user_type =
'user'") or die('Query failed');
$number_of_users = mysqli_num_rows($select_users);
?>
<h3><?php echo $number_of_users; ?></h3>
<p>Normal users</p>
</div>

<div class="box">
<?php
$select_admins = mysqli_query($conn, "SELECT * FROM `users` WHERE user_type =
'admin'") or die('Query failed');
$number_of_admins = mysqli_num_rows($select_admins);
?>
```

```
<h3><?php echo $number_of_admins; ?></h3>
<p>Admin users</p>
</div>

<div class="box">
<?php
$select_account = mysqli_query($conn, "SELECT * FROM `users`") or die('Query
failed');
$number_of_account = mysqli_num_rows($select_account);
?>
<h3><?php echo $number_of_account; ?></h3>
<p>Total accounts</p>
</div>

<div class="box">
<?php
$select_messages = mysqli_query($conn, "SELECT * FROM `message`") or die('Query
failed');
$number_of_messages = mysqli_num_rows($select_messages);
?>
<h3><?php echo $number_of_messages; ?></h3>
<p>New messages</p>
</div>

</div>

</section>

<script src="js/admin_script.js"></script>

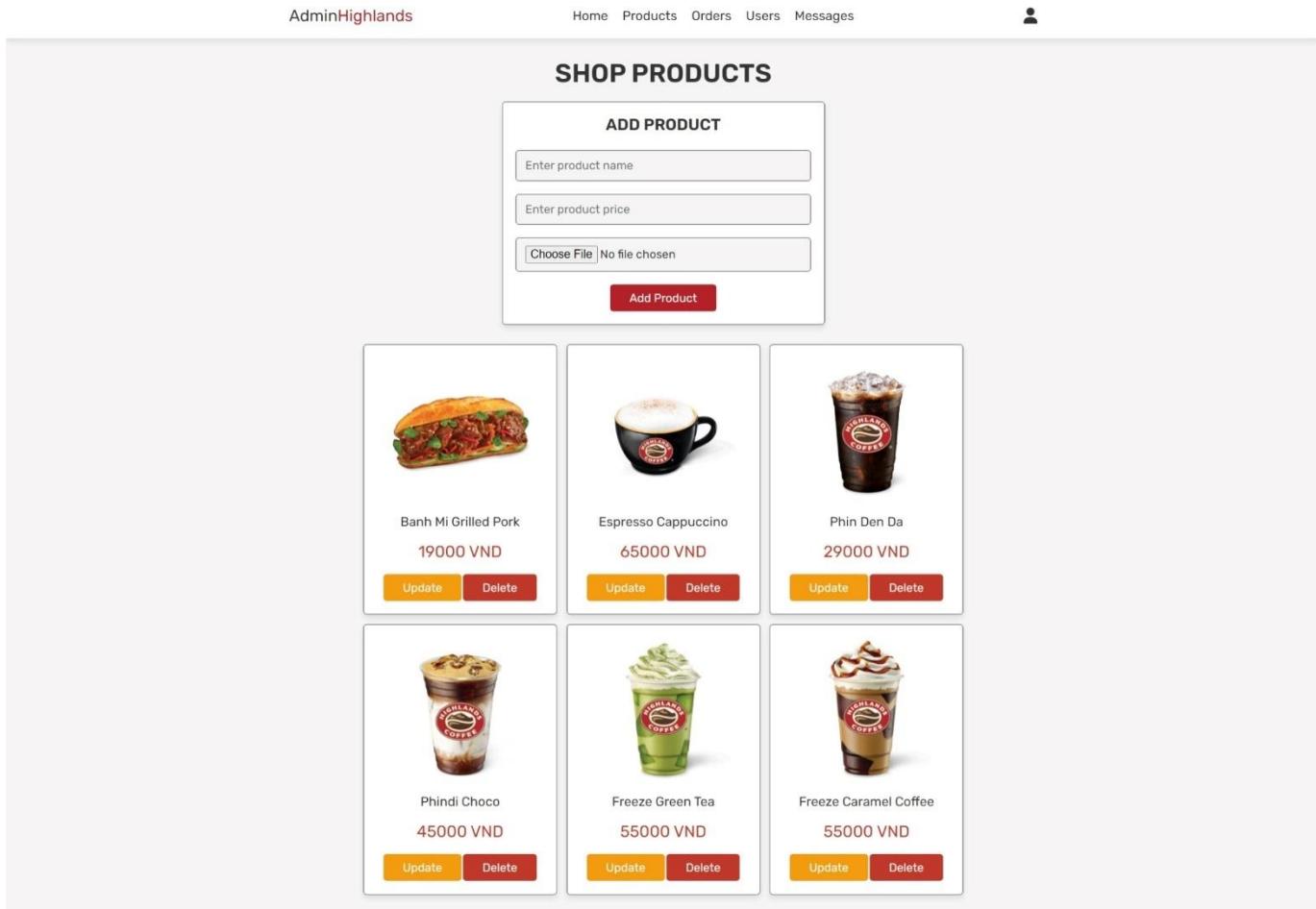
</body>

</html>
```

Figure 44: C2.2_Admin Session_Source Code_“Home” page

“Product” page

User Interface of the “Product” page



The screenshot shows a web-based administration interface for managing products. At the top, there's a navigation bar with links for Home, Products, Orders, Users, and Messages. On the far right, there's a user profile icon. The main content area is titled "SHOP PRODUCTS". On the left, there's a form titled "ADD PRODUCT" with fields for "Enter product name", "Enter product price", and a file upload field ("Choose File"). Below the form is a red "Add Product" button. The main area displays a grid of six product items, each in its own box:

- Banh Mi Grilled Pork**: 19000 VND. Actions: Update (orange), Delete (red).
- Espresso Cappuccino**: 65000 VND. Actions: Update (orange), Delete (red).
- Phin Den Da**: 29000 VND. Actions: Update (orange), Delete (red).
- Phindi Choco**: 45000 VND. Actions: Update (orange), Delete (red).
- Freeze Green Tea**: 55000 VND. Actions: Update (orange), Delete (red).
- Freeze Caramel Coffee**: 55000 VND. Actions: Update (orange), Delete (red).

Figure 45: C2.2_Admin Session_User Interface_“Product” page

Source code of the “Product” page

```
<?php

include 'config.php';

session_start();

$admin_id = $_SESSION['admin_id'];

if (!isset($admin_id)) {
    header('location:login.php');
};

if (isset($_POST['add_product'])) {
```

```

$name = mysqli_real_escape_string($conn, $_POST['name']);
$price = $_POST['price'];
$image = $_FILES['image']['name'];
$image_size = $_FILES['image']['size'];
$image_tmp_name = $_FILES['image']['tmp_name'];
$image_folder = 'uploaded_img/' . $image;

$select_product_name = mysqli_query($conn, "SELECT name FROM `products` WHERE name = '$name'" ) or die('Query failed');

if (mysqli_num_rows($select_product_name) > 0) {
    $message[] = 'Product name already added';
} else {
    $add_product_query = mysqli_query($conn, "INSERT INTO `products` (name, price, image)
VALUES('$name', '$price', '$image')") or die('Query failed');

    if ($add_product_query) {
        if ($image_size > 2000000) {
            $message[] = 'Image size is too large';
        } else {
            move_uploaded_file($image_tmp_name, $image_folder);
            $message[] = 'Product added successfully!';
        }
    } else {
        $message[] = 'Product could not be added!';
    }
}

if (isset($_GET['delete'])) {
    $delete_id = $_GET['delete'];
    $delete_image_query = mysqli_query($conn, "SELECT image FROM `products` WHERE id = '$delete_id'" ) or die('Query failed');
    $fetch_delete_image = mysqli_fetch_assoc($delete_image_query);
    unlink('uploaded_img/' . $fetch_delete_image['image']);
    mysqli_query($conn, "DELETE FROM `products` WHERE id = '$delete_id'" ) or die('Query failed');
    header('location:admin_products.php');
}

if (isset($_POST['update_product'])) {

    $update_p_id = $_POST['update_p_id'];
    $update_name = $_POST['update_name'];
}

```

```

$update_price = $_POST['update_price'];

mysqli_query($conn, "UPDATE `products` SET name = '$update_name', price = '$update_price'
WHERE id = '$update_p_id'" or die('Query failed'));

$update_image = $_FILES['update_image']['name'];
$update_image_tmp_name = $_FILES['update_image']['tmp_name'];
$update_image_size = $_FILES['update_image']['size'];
$update_folder = 'uploaded_img/' . $update_image;
$update_old_image = $_POST['update_old_image'];

if (!empty($update_image)) {
    if ($update_image_size > 2000000) {
        $message[] = 'Image file size is too large';
    } else {
        mysqli_query($conn, "UPDATE `products` SET image = '$update_image' WHERE id =
'$update_p_id'" or die('Query failed'));
        move_uploaded_file($update_image_tmp_name, $update_folder);
        unlink('uploaded_img/' . $update_old_image);
    }
}

header('location:admin_products.php');
}

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin.Products | Highlands Coffee</title>
    <link rel="icon" href="img/logo.admin.png" type="image/x-icon">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="css/admin_style.css">

</head>

<body>

    <?php include 'admin_header.php'; ?>

```

```
<section class="add-products">
    <h1 class="title">shop products</h1>
    <form action="" method="post" enctype="multipart/form-data">
        <h3>add product</h3>
        <input type="text" name="name" class="box" placeholder="Enter product name" required>
        <input type="number" min="0" name="price" class="box" placeholder="Enter product price" required>
        <input type="file" name="image" accept="image/jpg, image/jpeg, image/png" class="box" required>
        <input type="submit" value="add product" name="add_product" class="btn">
    </form>
</section>

<section class="show-products">
    <div class="box-container">
        <?php
            $select_products = mysqli_query($conn, "SELECT * FROM `products`") or die('Query failed');
            if (mysqli_num_rows($select_products) > 0) {
                while ($fetch_products = mysqli_fetch_assoc($select_products)) {
                    ?>
                    <div class="box">
                        
                        <div class="name"><?php echo $fetch_products['name']; ?></div>
                        <div class="price"><?php echo $fetch_products['price']; ?> VND</div>
                        <a href="admin_products.php?update=<?php echo $fetch_products['id']; ?>" class="option-btn">Update</a>
                        <a href="admin_products.php?delete=<?php echo $fetch_products['id']; ?>" class="delete-btn" onclick="return confirm('Delete this product?');">Delete</a>
                    </div>
                }
            } else {
                echo '<p class="empty">No products added yet!</p>';
            }
        ?>
    </div>
</section>

<section class="edit-product-form">
```

```
<?php
if (isset($_GET['update'])) {
    $update_id = $_GET['update'];
    $update_query = mysqli_query($conn, "SELECT * FROM `products` WHERE id =
'$update_id'" ) or die('Query failed');
    if (mysqli_num_rows($update_query) > 0) {
        while ($fetch_update = mysqli_fetch_assoc($update_query)) {
    ?>
        <form action="" method="post" enctype="multipart/form-data">
            <input type="hidden" name="update_p_id" value="<?php echo
$fetch_update['id']; ?>">
            <input type="hidden" name="update_old_image" value="<?php echo
$fetch_update['image']; ?>">
            
            <input type="text" name="update_name" value="<?php echo
$fetch_update['name']; ?>" class="box" required placeholder="Enter product name">
            <input type="number" name="update_price" value="<?php echo
$fetch_update['price']; ?>" min="0" class="box" required placeholder="Enter product price">
            <input type="file" class="box" name="update_image" accept="image/jpg,
image/jpeg, image/png">
            <input type="submit" value="update" name="update_product" class="btn">
            <input type="reset" value="cancel" id="close-update" class="option-btn">
        </form>
    <?php
    }
}
} else {
    echo '<script>document.querySelector(".edit-product-form").style.display =
"none";</script>';
}
?>

</section>

<script src="js/admin_script.js"></script>

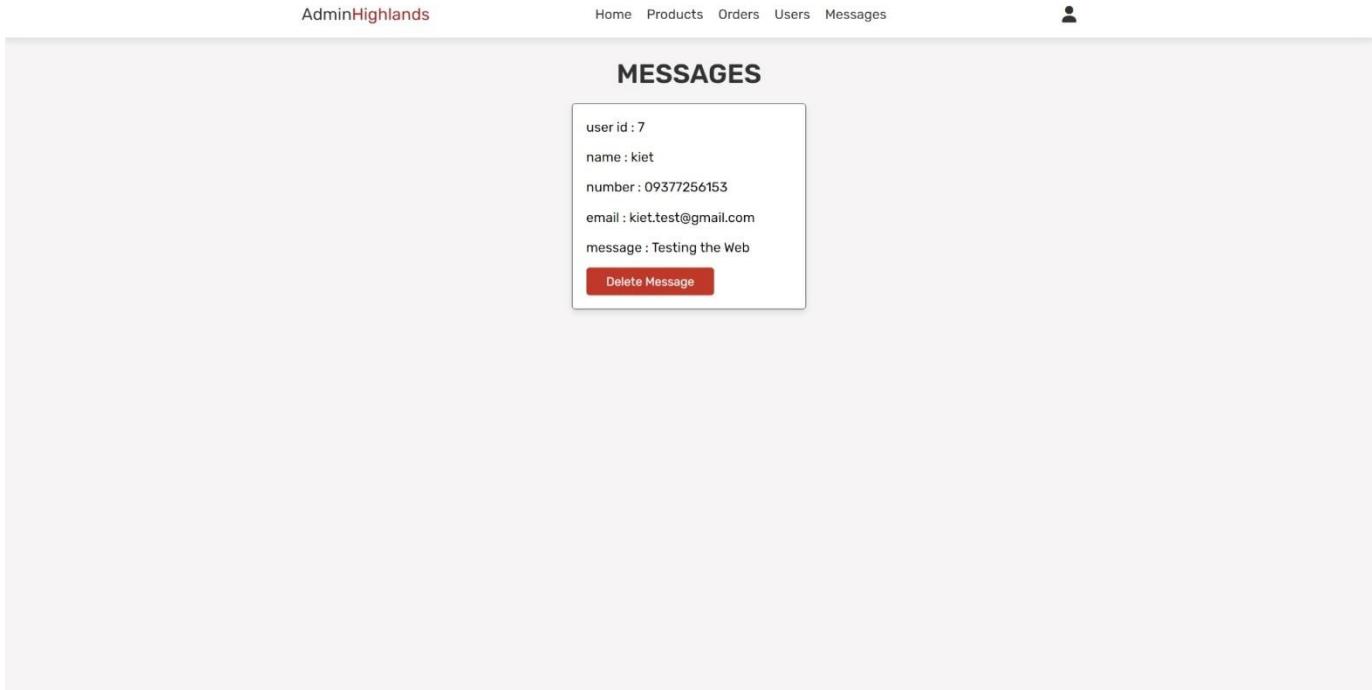
</body>

</html>
```

Figure 46: C2.2_Admin Session_Source Code_“Product” page

“Message” page

User Interface of the “Message” page



The screenshot shows a web application interface for managing messages. At the top, there is a navigation bar with links for Home, Products, Orders, Users, and Messages. On the far right of the navigation bar is a user profile icon. Below the navigation bar, the main content area has a title 'MESSAGES'. Underneath the title, there is a box containing the following data:

```

user id : 7
name : kiet
number : 09377256153
email : kiet.test@gmail.com
message : Testing the Web

```

At the bottom of this data box is a red button labeled 'Delete Message'.

Figure 47: C2.2_Admin Session_User Interface_“Message” page

Source code of the “Message” page

```

<?php

include 'config.php';

session_start();

$admin_id = $_SESSION['admin_id'];

if (!isset($admin_id)) {
    header('location:login.php');
};

if (isset($_GET['delete'])) {
    $delete_id = $_GET['delete'];
    mysqli_query($conn, "DELETE FROM `message` WHERE id = '$delete_id'") or die('Query failed');
    header('location:admin_contacts.php');
}

```

```

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin.Contact | Highlands Coffee</title>
    <link rel="icon" href="img/logo.admin.png" type="image/x-icon">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="css/admin_style.css">
</head>

<body>

<?php include 'admin_header.php'; ?>

<section class="messages">

    <h1 class="title"> Messages </h1>

    <div class="box-container">
        <?php
            $select_message = mysqli_query($conn, "SELECT * FROM `message`") or die('Query failed');
            if (mysqli_num_rows($select_message) > 0) {
                while ($fetch_message = mysqli_fetch_assoc($select_message)) {

                    ?>
                    <div class="box">
                        <p> user id : <span><?php echo $fetch_message['user_id']; ?></span> </p>
                        <p> name : <span><?php echo $fetch_message['name']; ?></span> </p>
                        <p> number : <span><?php echo $fetch_message['number']; ?></span> </p>
                        <p> email : <span><?php echo $fetch_message['email']; ?></span> </p>
                        <p> message : <span><?php echo $fetch_message['message']; ?></span> </p>
                        <a href="admin_contacts.php?delete=<?php echo $fetch_message['id']; ?>" onclick="return confirm('Delete this message?');"
                            class="delete-btn">Delete message</a>
                    </div>
                <?php
                    };
            } else {
                echo '<p class="empty">Right now, you have no messages!</p>';
            }
        </?php>
    </div>
</section>

```

```

        }
    ?>
</div>

</section>

<script src="js/admin_script.js"></script>

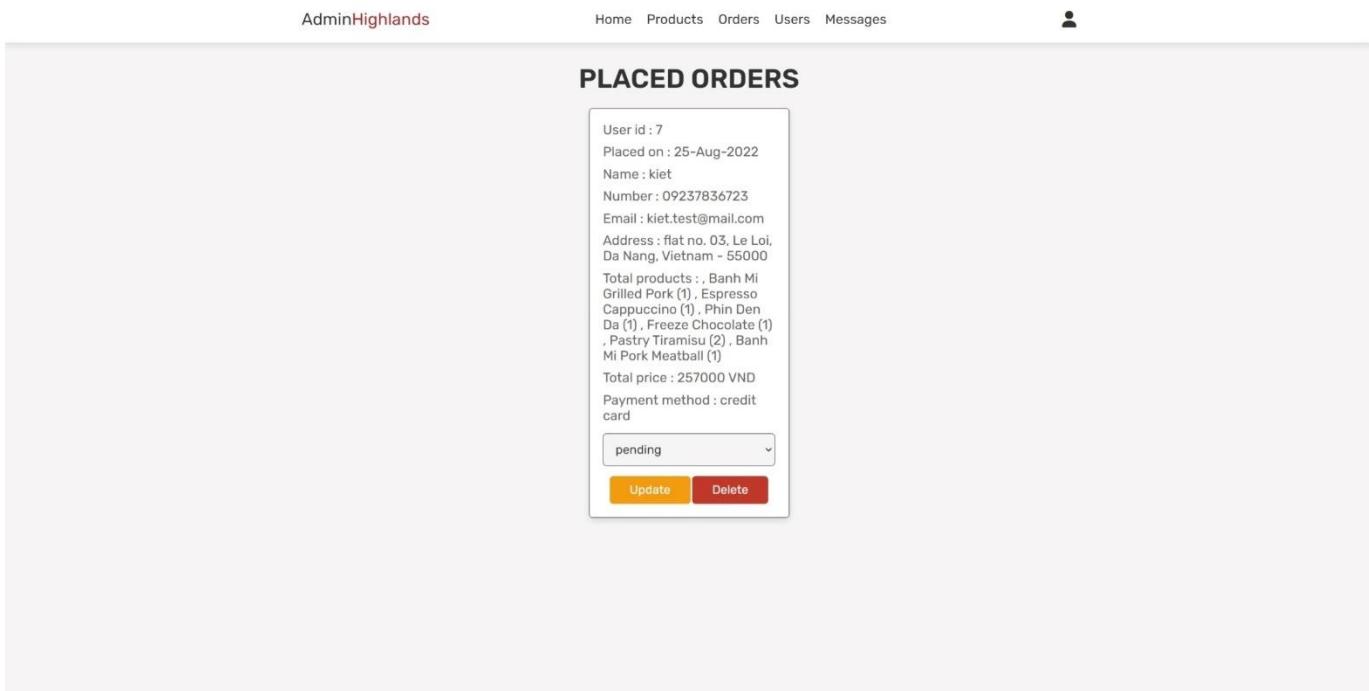
</body>

</html>
```

Figure 48: C2.2_Admin Session_Source Code_“Message” page

“Order” page

User Interface of the “Order” page



The screenshot shows a user interface for managing orders. At the top, there is a navigation bar with links for Home, Products, Orders, Users, and Messages. On the far right, there is a user profile icon. The main content area has a title "PLACED ORDERS". Below the title is a detailed card for a specific order:

- User id : 7
- Placed on : 25-Aug-2022
- Name : kiet
- Number : 09237836723
- Email : kiet.test@mail.com
- Address : flat no. 03, Le Loi, Da Nang, Vietnam - 55000
- Total products : , Banh Mi Grilled Pork (1), Espresso Cappuccino (1), Phin Den Da (1), Freeze Chocolate (1), Pastry Tiramisu (2), Banh Mi Pork Meatball (1)
- Total price : 257000 VND
- Payment method : credit card

Below the card, there is a dropdown menu set to "pending" and two buttons: "Update" and "Delete".

Figure 49: C2.2_Admin Session_User Interface_“Order” page

Source code of the “Order” page

```

<?php

include 'config.php';

session_start();
```

```
$admin_id = $_SESSION['admin_id'];

if (!isset($admin_id)) {
    header('location:login.php');
}

if (isset($_POST['update_order'])) {

    $order_update_id = $_POST['order_id'];
    $update_payment = $_POST['update_payment'];
    mysqli_query($conn, "UPDATE `orders` SET payment_status = '$update_payment' WHERE id =
'$order_update_id'" ) or die('Query failed');
    $message[] = 'Payment status has been updated!';
}

if (isset($_GET['delete'])) {
    $delete_id = $_GET['delete'];
    mysqli_query($conn, "DELETE FROM `orders` WHERE id = '$delete_id'" ) or die('Query
failed');
    header('location:admin_orders.php');
}

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin Orders | Highlands Coffee</title>
    <link rel="icon" href="img/logo.admin.png" type="image/x-icon">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="css/admin_style.css">

</head>

<body>

    <?php include 'admin_header.php'; ?>

        <section class="orders">
```

```
<h1 class="title">Placed orders</h1>

<div class="box-container">
    <?php
        $select_orders = mysqli_query($conn, "SELECT * FROM `orders`") or die('Query failed');
        if (mysqli_num_rows($select_orders) > 0) {
            while ($fetch_orders = mysqli_fetch_assoc($select_orders)) {
                ?>
                <div class="box">
                    <p> User id : <span><?php echo $fetch_orders['user_id']; ?></span> </p>
                    <p> Placed on : <span><?php echo $fetch_orders['placed_on']; ?></span>
                </p>
                    <p> Name : <span><?php echo $fetch_orders['name']; ?></span> </p>
                    <p> Number : <span><?php echo $fetch_orders['number']; ?></span> </p>
                    <p> Email : <span><?php echo $fetch_orders['email']; ?></span> </p>
                    <p> Address : <span><?php echo $fetch_orders['address']; ?></span> </p>
                    <p> Total products : <span><?php echo $fetch_orders['total_products'];
                ?></span> </p>
                    <p> Total price : <span><?php echo $fetch_orders['total_price']; ?>
                VND</span> </p>
                    <p> Payment method : <span><?php echo $fetch_orders['method']; ?></span>
                </p>
                    <form action="" method="post">
                        <input type="hidden" name="order_id" value="<?php echo
$fetch_orders['id']; ?>">
                        <select name="update_payment">
                            <option value="" selected disabled><?php echo
$fetch_orders['payment_status']; ?></option>
                            <option value="pending">Pending ... </option>
                            <option value="completed">Completed</option>
                        </select>
                        <input type="submit" value="update" name="update_order" class="option-
btn">
                        <a href="admin_orders.php?delete=<?php echo $fetch_orders['id']; ?>">
                    onclick="return confirm('Delete this order?');</a>
                        >Delete</a>
                    </form>
                </div>
            <?php
            }
        } else {
            echo '<p class="empty">No orders placed yet!</p>';
        }
    ?>
```

```

</div>

</section>

<script src="js/admin_script.js"></script>

</body>

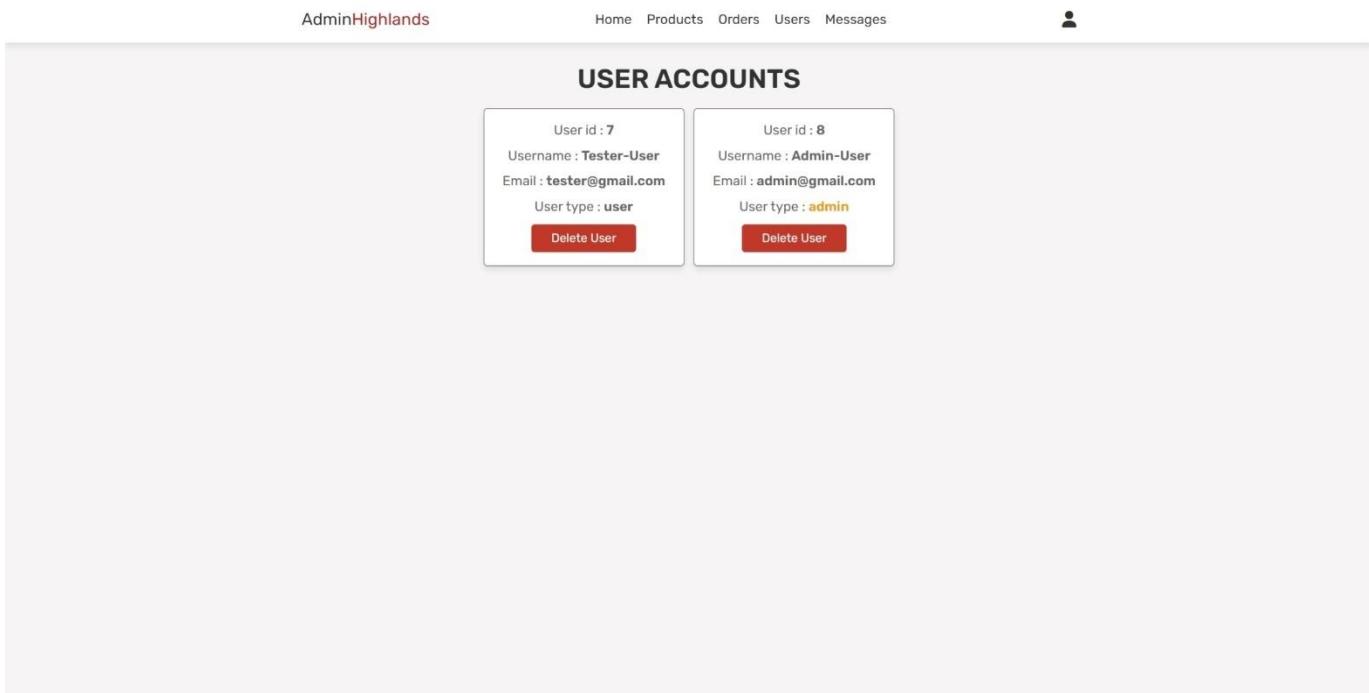
</html>

```

Figure 50: C2.2 Admin Session Source Code “Order” page

“User” page

User Interface of the “User” page



The screenshot shows a web application interface titled "USER ACCOUNTS". At the top, there is a navigation bar with links for "Home", "Products", "Orders", "Users", and "Messages". On the far right of the navigation bar is a user profile icon. The main content area is titled "USER ACCOUNTS" and displays two user profiles in separate boxes:

- User id : 7**
Username : Tester-User
Email : tester@gmail.com
User type : user
[Delete User](#)
- User id : 8**
Username : Admin-User
Email : admin@gmail.com
User type : admin
[Delete User](#)

Figure 51: C2.2 Admin Session User Interface “User” page

Source code of the “User” page

```

<?php

include 'config.php';

session_start();

$admin_id = $_SESSION['admin_id'];

```

```

if (!isset($admin_id)) {
    header('location:login.php');
}

if (isset($_GET['delete'])) {
    $delete_id = $_GET['delete'];
    mysqli_query($conn, "DELETE FROM `users` WHERE id = '$delete_id'" ) or die('Query failed');
    header('location:admin_users.php');
}

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin.Accounts | Highlands Coffee</title>
    <link rel="icon" href="img/logo.admin.png" type="image/x-icon">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <link rel="stylesheet" href="css/admin_style.css">

</head>

<body>

    <?php include 'admin_header.php'; ?>

    <section class="users">

        <h1 class="title"> user accounts </h1>

        <div class="box-container">
            <?php
                $select_users = mysqli_query($conn, "SELECT * FROM `users`") or die('Query failed');
                while ($fetch_users = mysqli_fetch_assoc($select_users)) {
            ?>
            <div class="box">
                <p> User id : <b><?php echo $fetch_users['id']; ?></b> </p>

```

```

<p> Username : <b><?php echo $fetch_users['name']; ?></b> </p>
<p> Email : <b><?php echo $fetch_users['email']; ?></b> </p>
<p> User type : <b style="color:<?php if ($fetch_users['user_type'] ==
'admin') {
                                echo 'var(--orange)'; } ?>"><?php echo $fetch_users['user_type'];
?></b> </p>
        <a href="admin_users.php?delete=<?php echo $fetch_users['id']; ?>"
onclick="return confirm('Delete this user?');" class="delete-btn">Delete user</a>
        </div>
    <?php
    ?>;
    ?>
</div>

</section>
<script src="js/admin_script.js"></script>

</body>

</html>

```

Figure 52: C2.2_Admin Session_Source Code_ “User” page

CSS and JavaScript

CSS files

admin_style.css file

Because of the length of **admin_style.css** file, I will put the link of my source code of that file here which includes all of the folders and files of this project. So that lecturer can check it much easier than check it in this current pdf file.

You can check it through 'css' folder: https://github.com/KieTuaNguyen/Nguyen-Tuan-Kiet-FGW-DN_GDD21002_Assignment-2_1633/tree/main/css

Here is my project source code link: https://github.com/KieTuaNguyen/Nguyen-Tuan-Kiet-FGW-DN_GDD21002_Assignment-2_1633

style.css file

Because of the length of **style.css** file, I will put the link of my source code of that file here which includes all of the folders and files of this project. So that lecturer can check it much easier than check it in this current pdf file.

You can check it through 'css' folder: https://github.com/KieTuaNguyen/Nguyen-Tuan-Kiet-FGW-DN-_GDD21002_Assignment-2_1633/tree/main/css

Here is my project source code link: https://github.com/KieTuaNguyen/Nguyen-Tuan-Kiet-FGW-DN-_GDD21002_Assignment-2_1633

JavaScript files

admin_script.js

```
let navbar = document.querySelector('.header .navbar');
let accountBox = document.querySelector('.header .account-box');

document.querySelector('#menu-btn').onclick = () =>{
    navbar.classList.toggle('active');
    accountBox.classList.remove('active');
}

document.querySelector('#user-btn').onclick = () =>{
    accountBox.classList.toggle('active');
    navbar.classList.remove('active');
}

window.onscroll = () =>{
    navbar.classList.remove('active');
    accountBox.classList.remove('active');
}

document.querySelector('#close-update').onclick = () =>{
    document.querySelector('.edit-product-form').style.display = 'none';
    window.location.href = 'admin_products.php';
}
```

Figure 53: C2.2_CSS and JavaScript_admin_script.js

script.js file

```
let userBox = document.querySelector('.header .header-2 .user-box');

document.querySelector('#user-btn').onclick = () =>{
    userBox.classList.toggle('active');
```

```
    navbar.classList.remove('active');
}

Let navbar = document.querySelector('.header .header-2 .navbar');

document.querySelector('#menu-btn').onclick = () =>{
    navbar.classList.toggle('active');
    userBox.classList.remove('active');
}

window.onscroll = () =>{
    userBox.classList.remove('active');
    navbar.classList.remove('active');

    if(window.scrollY > 60){
        document.querySelector('.header .header-2').classList.add('active');
    }else{
        document.querySelector('.header .header-2').classList.remove('active');
    }
}
```

Figure 54: C2.2_CSS and JavaScript_script.js

3. Features of the website

Registering for a new user

First, for new users who do not have an account, they are required to register an account by clicking "Register now".

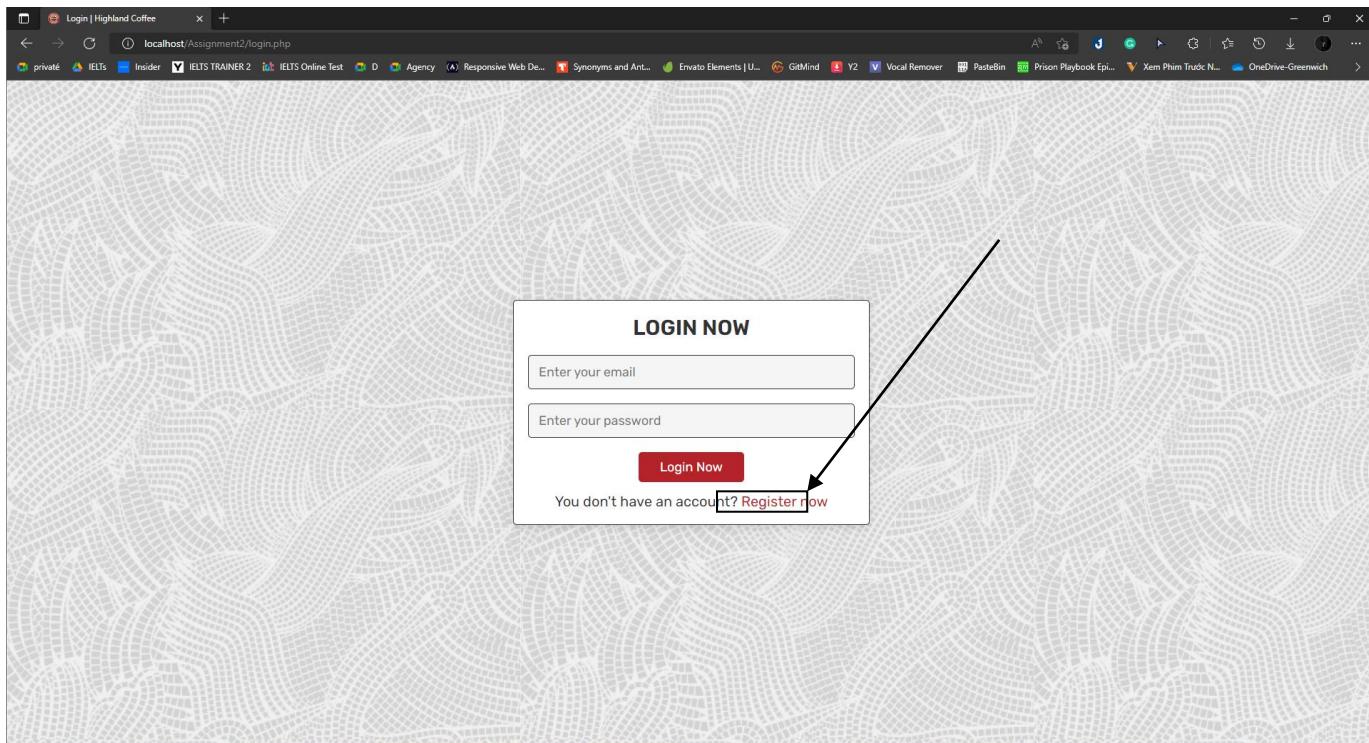


Figure 55: C2.3_Register

After the website forwards the user to the registration page, they are required to fill in all the field information when pressing the "Register now" button.

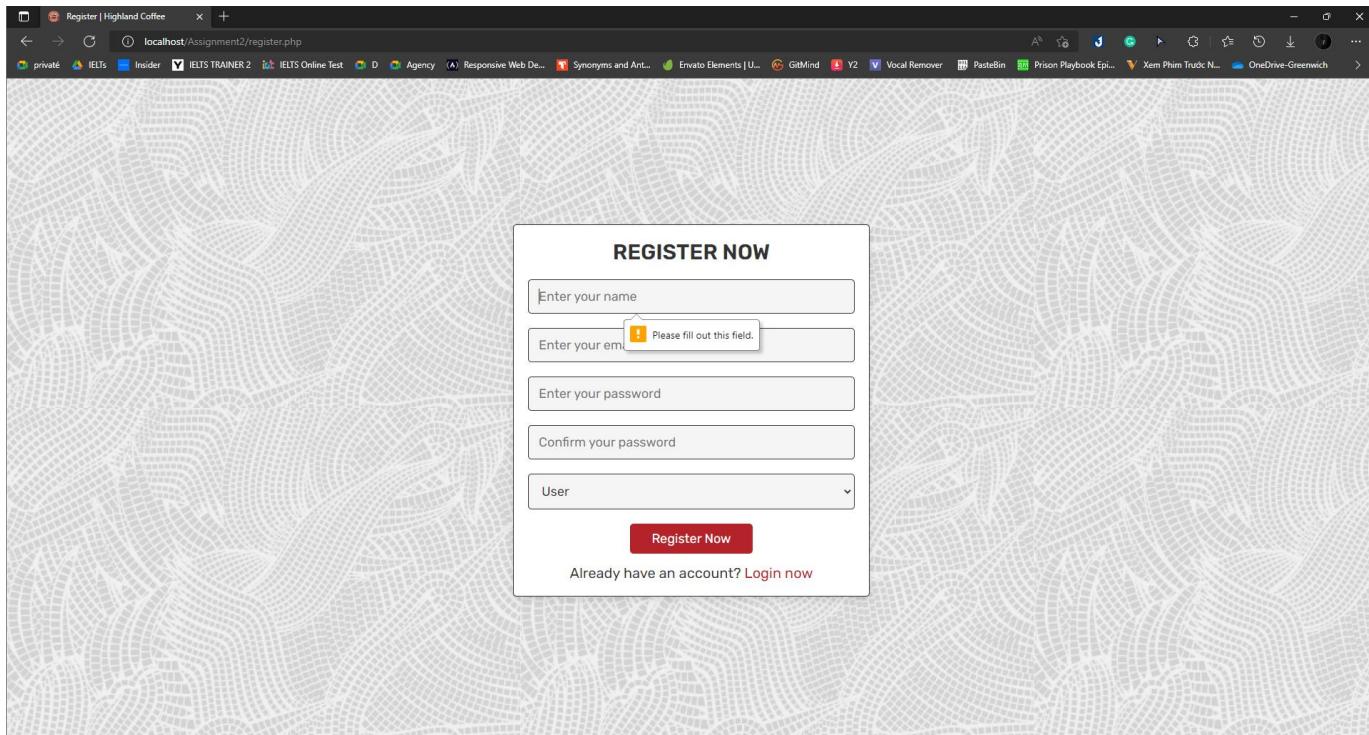


Figure 56: C2.3_Register_Fill out

In case of entering "Email" information that matches the previously registered information, the Website will display a notice board "User already exists!" and the user is required to re-enter valid information.

The screenshot shows a web browser window with a registration form. At the top, a red banner displays the message "User already exist!". Below the banner is a registration form titled "REGISTER NOW". The form contains five input fields: "Enter your name", "Enter your email", "Enter your password", "Confirm your password", and a dropdown menu for "User". At the bottom of the form is a red "Register Now" button. Below the button, a link reads "Already have an account? [Login now](#)".

Figure 57: C2.3_Register_Duplicate with the previous email

In case the user enters a confirmation password that is different from the password entered before, the Website will display a message board "Confirm password not matched!" and the user is required to re-enter valid information.

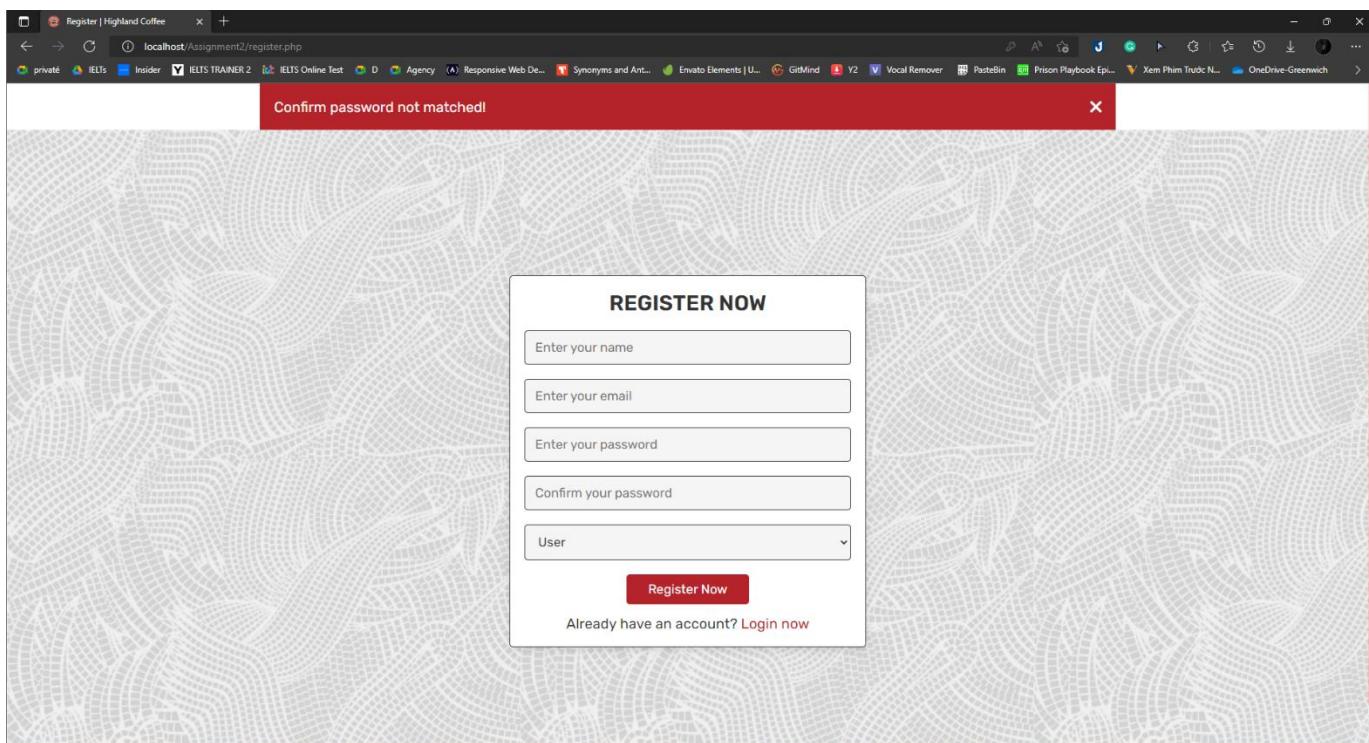


Figure 58: C2.3_Register_No password match

Once valid information has been entered, this means that the user's information is stored in the database in phpMyAdmin when registering with valid credentials, the user will be forwarded to the login page after pressing the "Register now" button.

Login with email and password

In case the user enters the wrong password or email which was previously created when registering an account, the system will display a notice board "Incorrect email or password!" and the user is required to re-enter valid information.

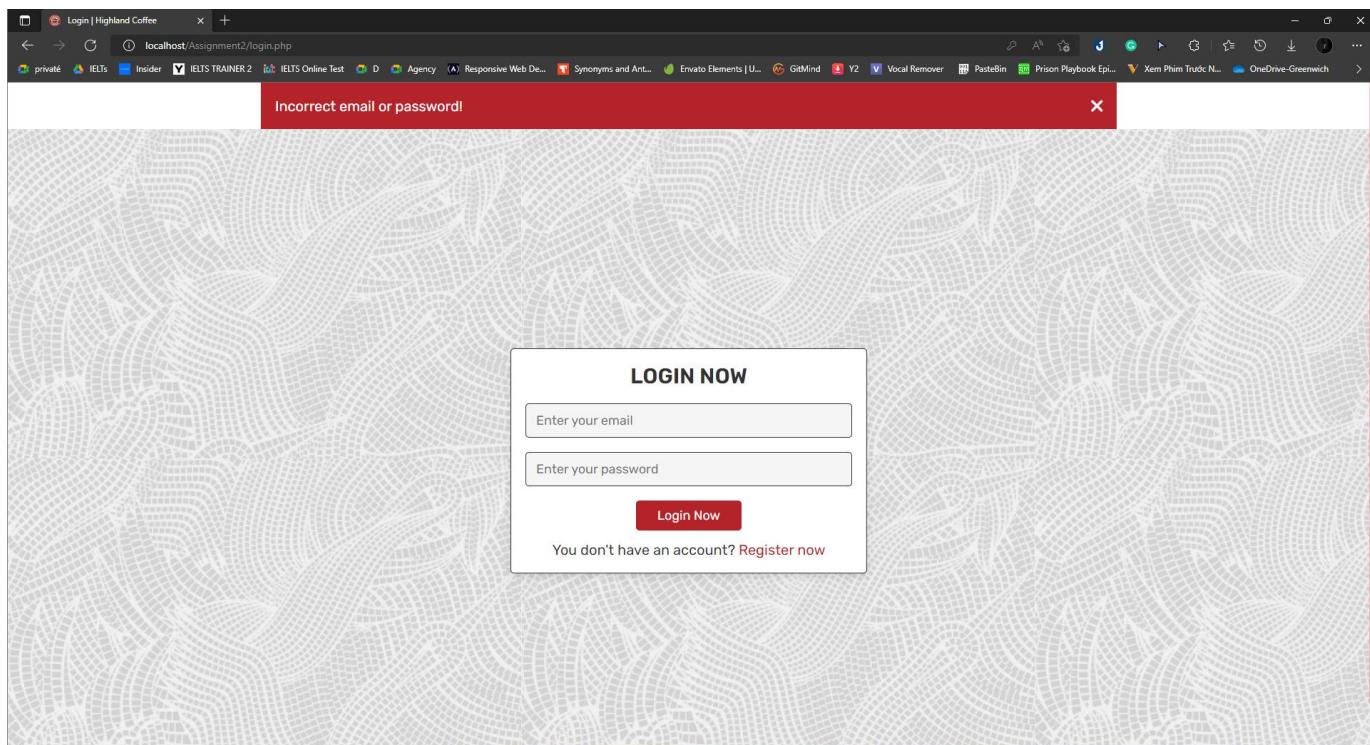


Figure 59: C2.3_Login_Incorrect email or password

In case the user only enters data before the @ character, the system displays a message prompting the user to check the information of the email and the user is required to re-enter valid information.

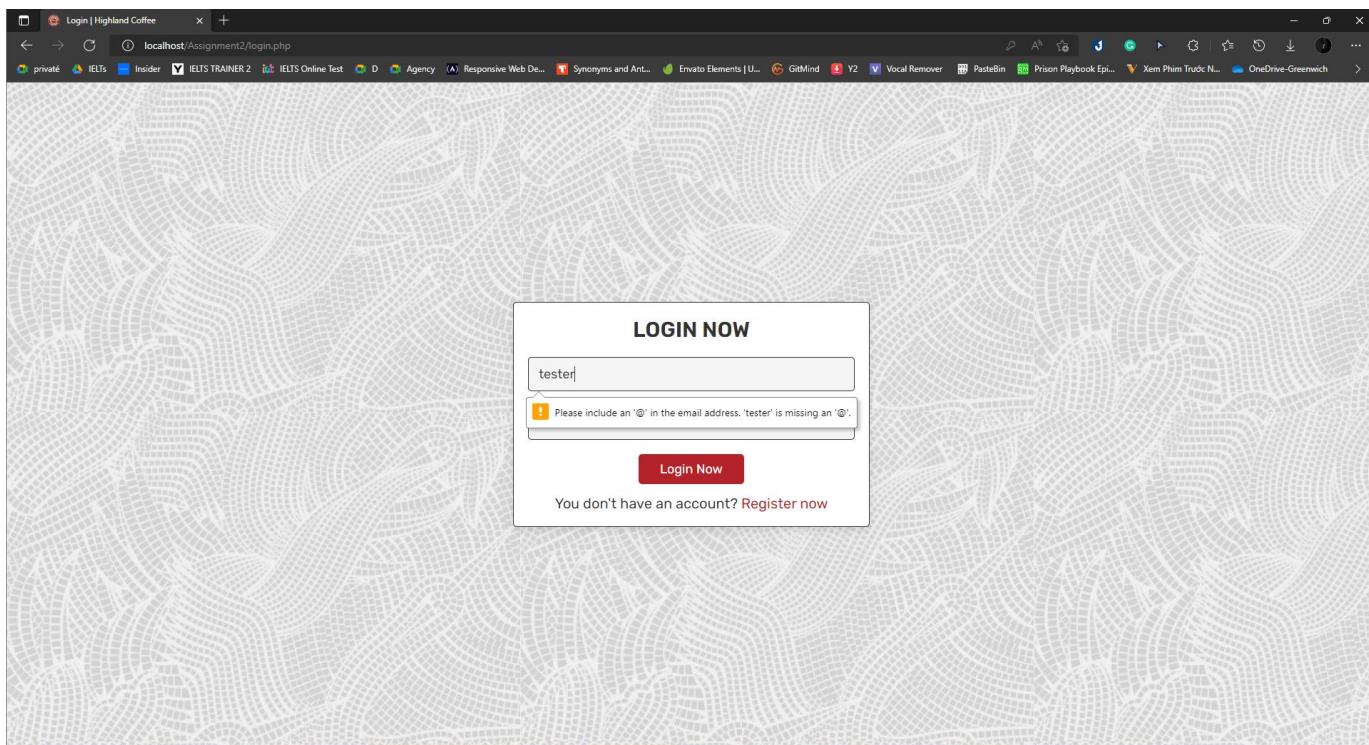


Figure 60: C2.3_Login_Incorrect_email

After entering valid information, after pressing the Login button and entering the correct email and password, the user will be immediately redirected to the "Home" page.

CRUD features

Create action:

All of my registration information, including username, email, phone number, address, and password, is saved in the Customers table of the database when I create a new user. A new account is created in this manner.

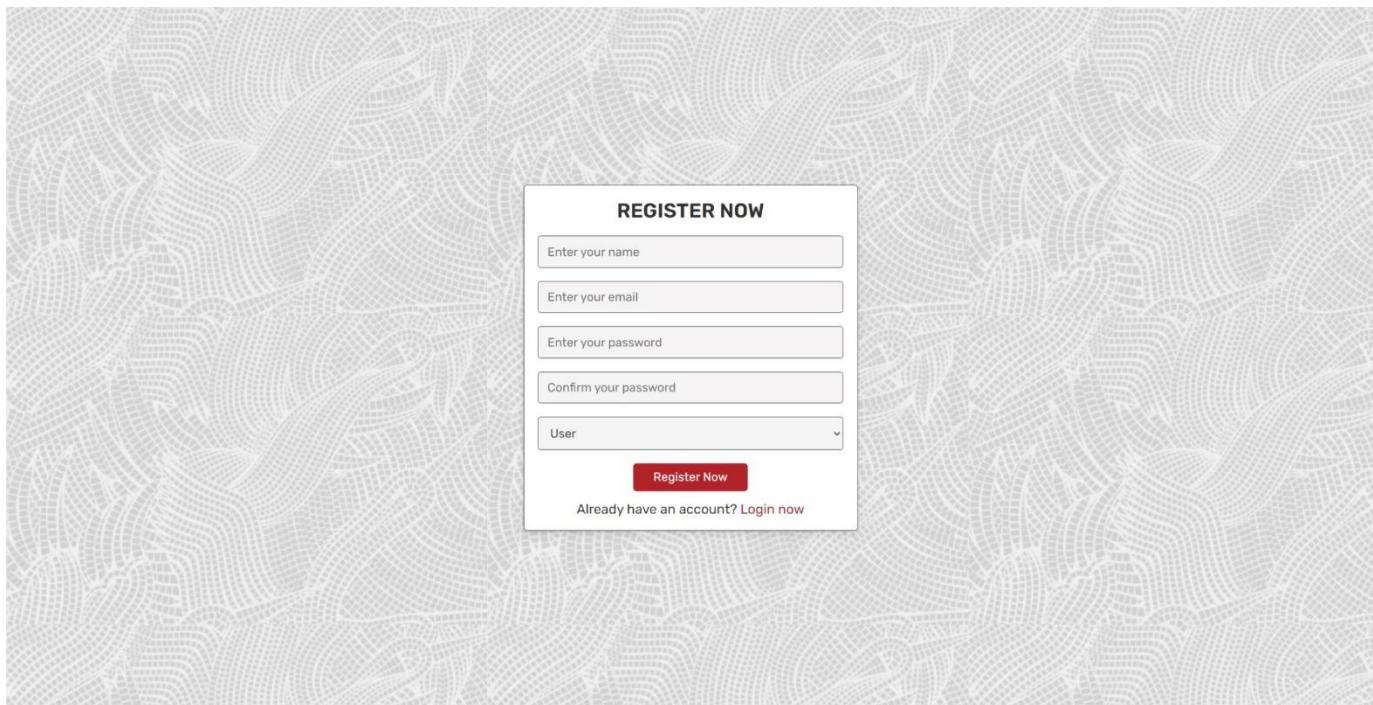


Figure 61: C2.3_CRUD feature_ Create action

	<input type="button" value="←"/>	<input type="button" value="→"/>	<input type="button" value="▼"/>	id	name	email	password	user_type
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Tester-User	tester@gmail.com	f5d1278e8109edd94e1e4197e04873b9	user
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	Admin-User	admin@gmail.com	21232f297a57a5a743894a0e4a801fc3	admin

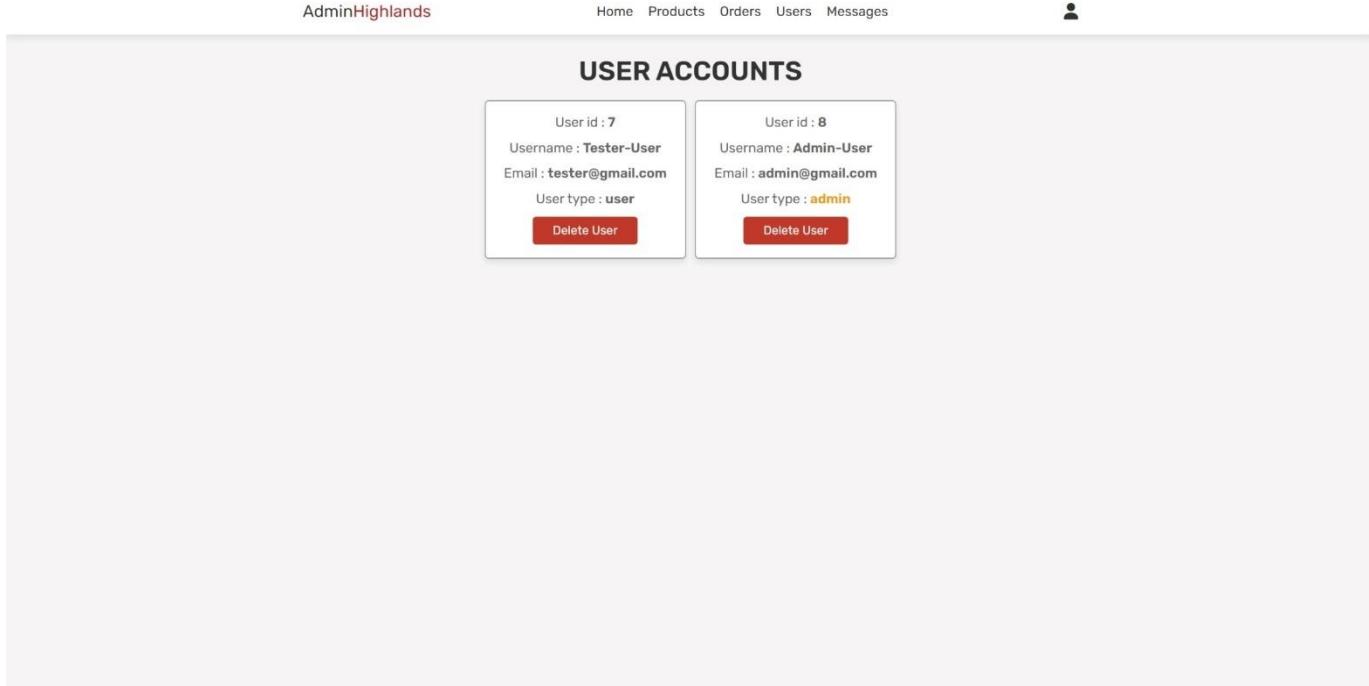
Check all With selected:  Edit  Copy  Delete  Export

Show all | Number of rows: Filter rows: Search this table Sort by key: None

Figure 62: C2.3_CRUD feature_ Create action

Read action:

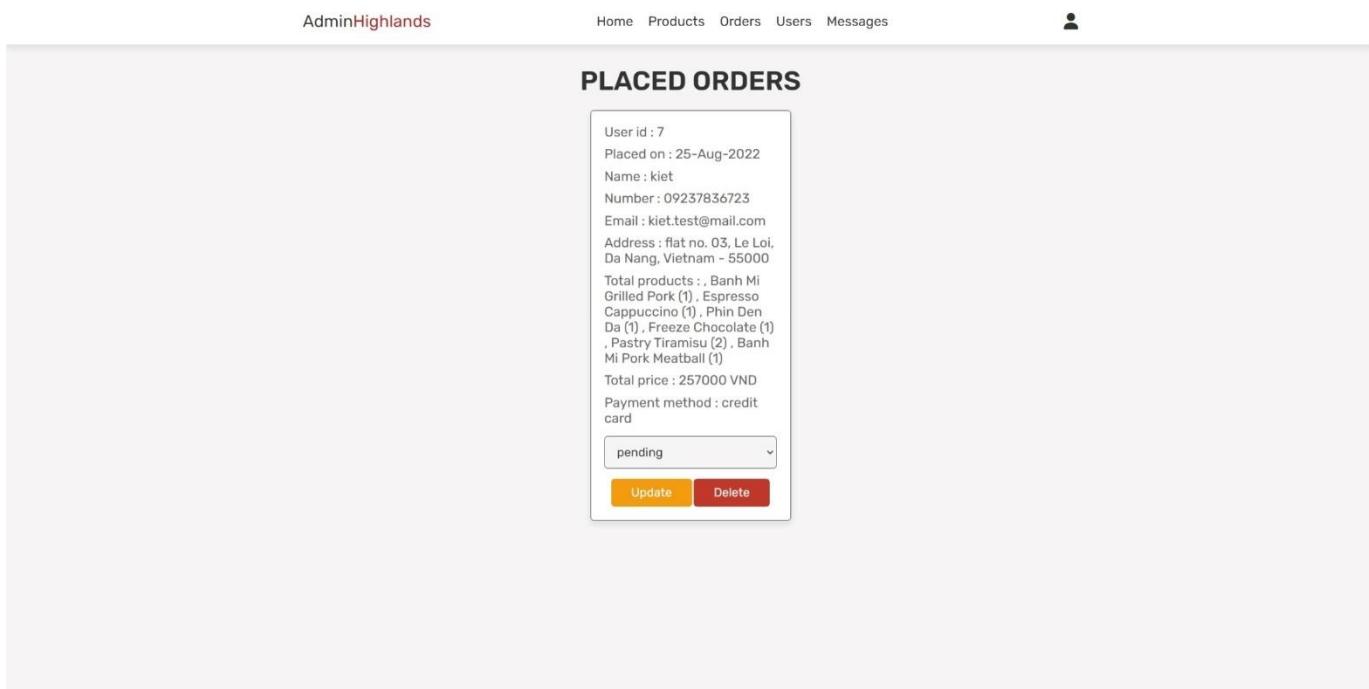
Admin is the person who can read the user's account information as well as view the user's order and message information. "Reading a record" is the term for this.



The screenshot shows a web application interface titled "USER ACCOUNTS". At the top, there is a navigation bar with links: "AdminHighlands", "Home", "Products", "Orders", "Users", "Messages", and a user profile icon. Below the navigation bar, there are two user profiles displayed in separate boxes:

- User id : 7**
Username : Tester-User
Email : tester@gmail.com
User type : user
[Delete User](#)
- User id : 8**
Username : Admin-User
Email : admin@gmail.com
User type : admin
[Delete User](#)

Figure 63: C2.3 CRUD feature_Read action

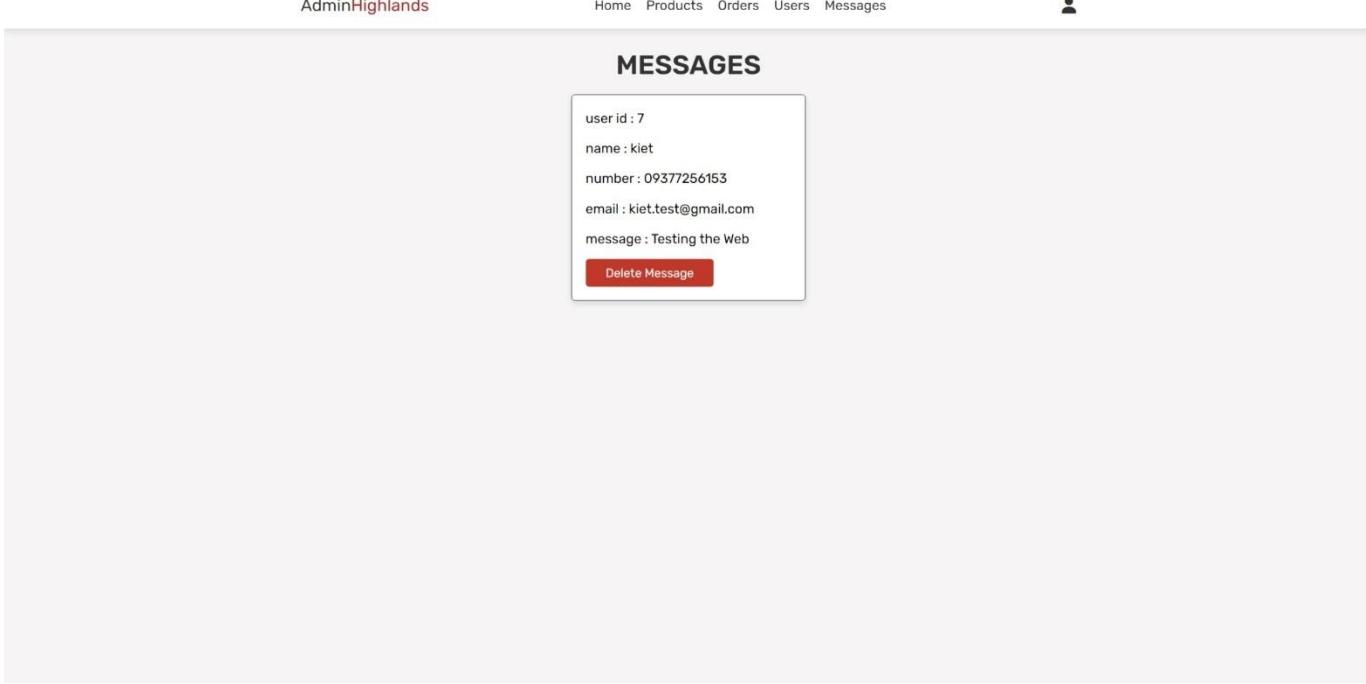


The screenshot shows a web application interface titled "PLACED ORDERS". At the top, there is a navigation bar with links: "AdminHighlands", "Home", "Products", "Orders", "Users", "Messages", and a user profile icon. Below the navigation bar, there is one order detail displayed in a box:

- User id :** 7
Placed on : 25-Aug-2022
Name : kiet
Number : 09237836723
Email : kiet.test@mail.com
Address : flat no. 03, Le Loi, Da Nang, Vietnam - 55000
Total products : Banh Mi Grilled Pork (1), Espresso Cappuccino (1), Phin Den Da (1), Freeze Chocolate (1), Pastry Tiramisu (2), Banh Mi Pork Meatball (1)
Total price : 257000 VND
Payment method : credit card

[Update](#) [Delete](#)

Figure 64: C2.3 CRUD feature_Read action



The screenshot shows a web application interface. At the top, there is a navigation bar with links for Home, Products, Orders, Users, and Messages. On the far right of the navigation bar is a user profile icon. Below the navigation bar, the page title is "MESSAGES". In the center, there is a detailed message card with the following information:
user id : 7
name : kiet
number : 09377256153
email : kiet.test@gmail.com
message : Testing the Web
At the bottom of the message card is a red button labeled "Delete Message".

Figure 65: C2.3 CRUD feature_Read action

Update action:

Admin is the person who can update product information. This is the action I take when I make changes to my profile. In the update section, I will update the item's information such as the product name, product price as well as the image of that product.

AdminHighlands Home Products Orders Users Messages

SHOP PRODUCTS

ADD PRODUCT

Enter product name
Enter product price
Choose File | No file chosen

Add Product

	Espresso Cappuccino	Phin Den Da
Banh Mi Grilled Pork 19000 VND	65000 VND	29000 VND
Update Delete	Update Delete	Update Delete
	Freeze Green Tea	Freeze Caramel Coffee
Phindi Choco 45000 VND	55000 VND	55000 VND
Update Delete	Update Delete	Update Delete

Figure 66: C2.3 CRUD feature _Update action

AdminHighlands Home Products Orders Users Messages

SHOP PRODUCTS

ADD PRODUCT

Enter product name.
Enter product price.

Banh Mi Grilled Pork
19000
Choose File | No file chosen

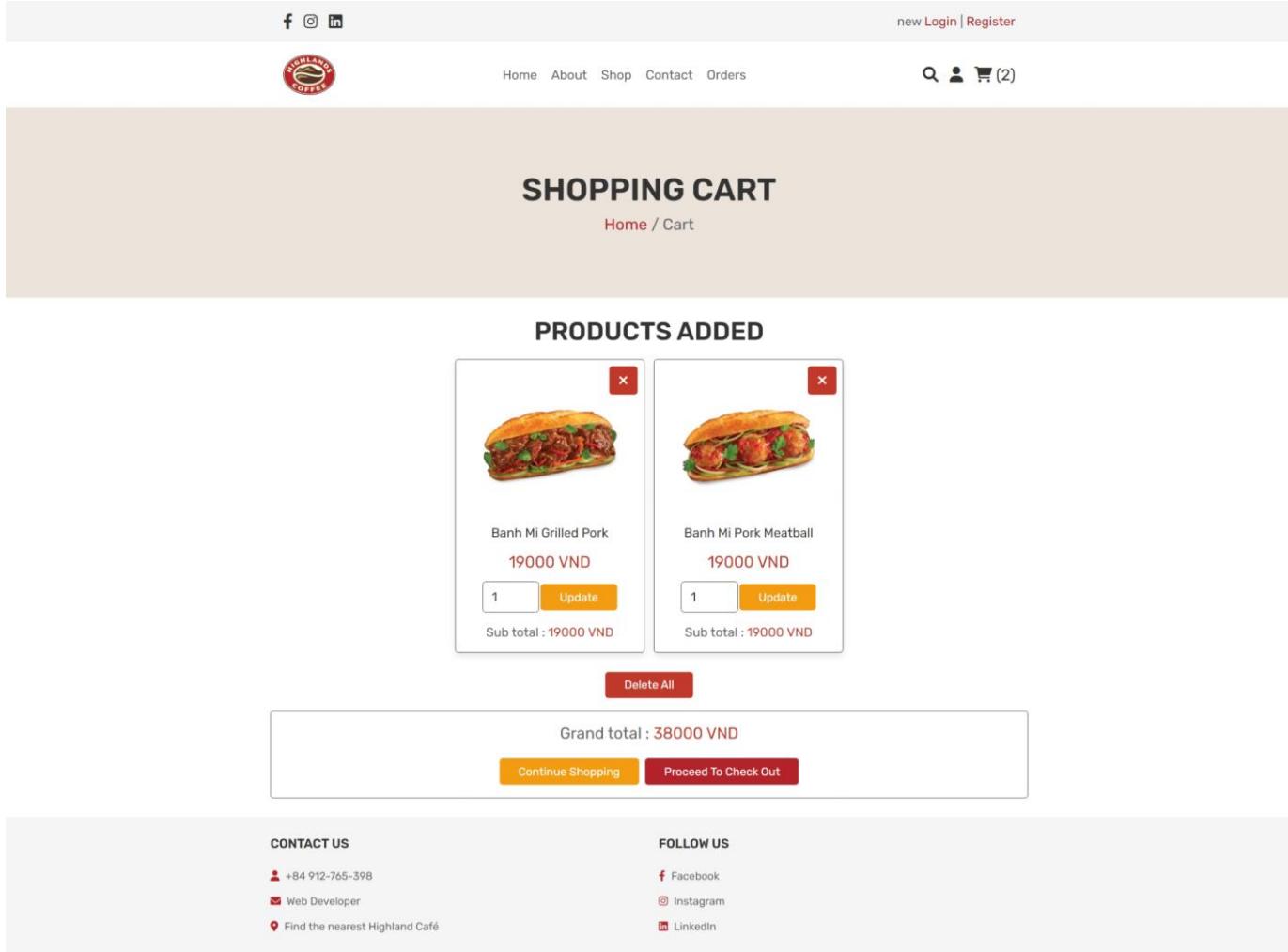
Update Cancel

	Banh Mi Pork Meatball	Banh Mi Roast Pork
Banh Mi Grilled Pork 19000 VND	19000 VND	19000 VND
Update Delete	Update Delete	Update Delete

Figure 67: C2.3_CRUD feature_Update action

Delete action:

After shopping, I press the Cart button to pay. And here is the product I just added. In case I don't want to buy a product, I will press the delete button which is the X icon in the top right corner of that product to delete.



SHOPPING CART
[Home](#) / Cart

PRODUCTS ADDED

 Banh Mi Grilled Pork 19000 VND <input type="button" value="1"/> <input type="button" value="Update"/> Sub total : 19000 VND	 Banh Mi Pork Meatball 19000 VND <input type="button" value="1"/> <input type="button" value="Update"/> Sub total : 19000 VND
--	---

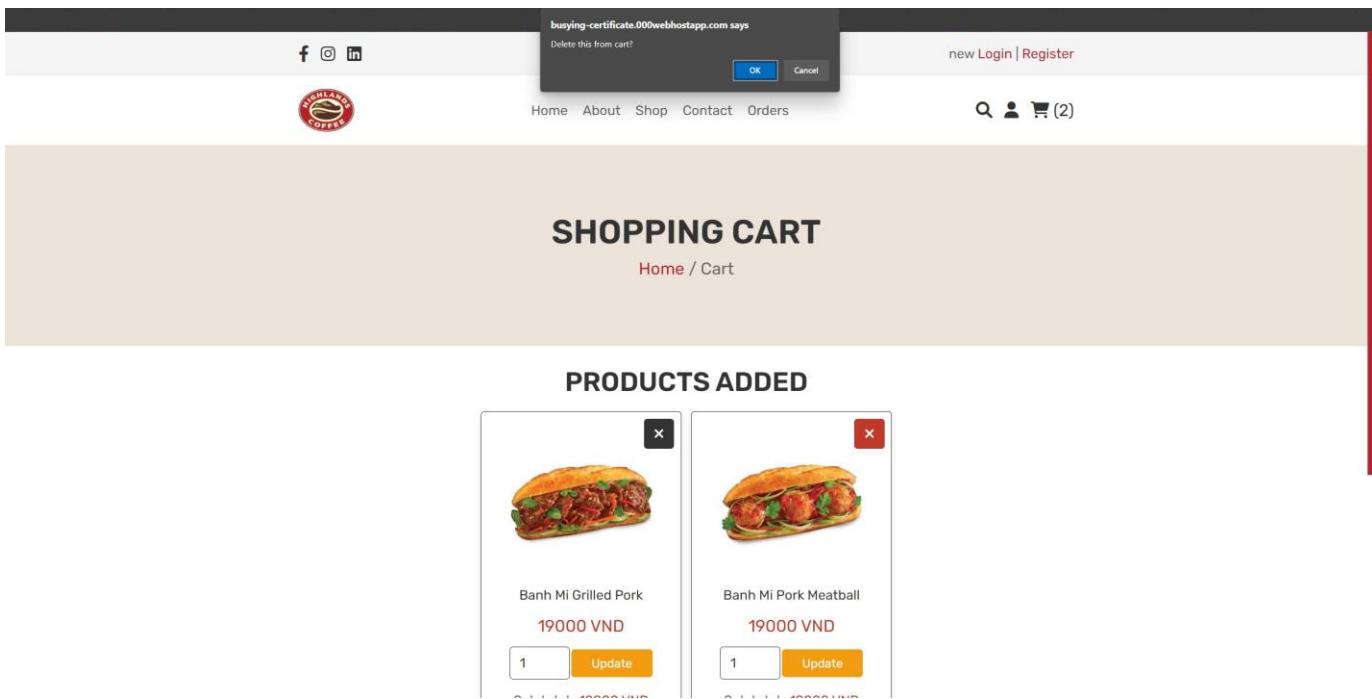
Contact Us

- +84 912-765-398
- Web Developer
- Find the nearest Highland Café

Follow Us

- Facebook
- Instagram
- LinkedIn

Figure 68: C2.3_CRUD feature_Delete action



The screenshot shows a shopping cart page for "BANH MI COFFEE". At the top, there's a delete confirmation dialog box from "buying-certificate.000webhostapp.com says": "Delete this from cart? OK Cancel". The main content area has a light beige background and displays the title "SHOPPING CART" and the breadcrumb "Home / Cart". Below this, a section titled "PRODUCTS ADDED" lists two items:

Product	Quantity	Price	Action
Banh Mi Grilled Pork	1	19000 VND	<input type="button" value="Update"/> <input type="button" value="Delete"/>
Banh Mi Pork Meatball	1	19000 VND	<input type="button" value="Update"/> <input type="button" value="Delete"/>

Figure 69: C2.3 CRUD feature_Delete action

Chapter 3: Create a suitable Test Plan identifying key performance area and use it to review the functionality and performance of your website. User Experience (UX) and User Interface (UI). (P7)

1. Test plan template

Table 1: C3.1_Test plan template

No.	Test case	Action	Day	Expected result
1	Register	Register a new account with the required information.	27-08-2022	Pass
2	Log in for user session	Log in with an existing user account.	27-08-2022	Pass
3	Log in for admin session	Log in with an existing admin account.	27-08-2022	Pass
4	Log out	When the user clicks the logout icon, it will get out of account	27-08-2022	Pass
5	View customer information on admin session	View information of all users and admins currently registered on admin session.	27-08-2022	Pass
6	Update product information on admin session	On the admin session, the user updates the specified product information.	27-08-2022	Pass
7	Delete product from cart	The user proceeds to delete the previously added product in the cart.	27-08-2022	Pass
8	Delete user information on admin session	On the admin session, the admin deletes the user account.	27-08-2022	Pass
9	Register with different password and confirm password.	Enter Password and Confirm Password with two different values.	27-08-2022	Pass
10	Log in with incorrect password	Import incorrect passwords that before that used to register an account.	27-08-2022	Pass

2. Test case template

Table 2: C3.2_Test case template

No.	Test case	Action	Value used	Expected output	Actual output	Expected status	Actual status
1	Registration	Register a new account with the required information.	"Test Case 1", "testcase01@gmail.com", "testcase1", "testcase1"	All information entered in the registration form is displayed correctly in the database.	All information entered in the registration form is displayed correctly in the database.	Pass	Pass
2	Log in for user session	Log in with an existing user account.	"tester@gmail.com", "tester"	Successfully logged in with an existing user account, which matches all existing user information in the database.	Successfully logged in with an existing user account, which matches all existing user information in the database.	Pass	Pass
3	Log in for admin session	Log in with an existing admin account.	"admin@gmail.com", "admin"	Successfully logged in with an existing admin account, which matches all existing user information in the database.	Successfully logged in with an existing admin account, which matches all existing user information in the database.	Pass	Pass
4	Log out	When the user clicks the logout icon, it will get out of account	X	Log out successfully, after reloading the page still stay logged in to continue.	Log out successfully, after reloading the page still stay logged in to continue.	Pass	Pass
5	View customer information on admin session	View information of all users and admins currently registered on admin session.	X	View all customer information in the admin session from the "Users" page, which matches all existing user information in the database.	View all customer information in the admin session from the "Users" page, which matches all existing user information in the database.	Pass	Pass
6	Update	On the admin	"Basic Tee", "550000",	The specified product's	The specified product's	Pass	Pass

	product information on admin session	session, the user updates the specified product information.	"tee.staff.jpg"	information is updated from the current web page as well as from the database.	information is updated from the current web page as well as from the database.		
7	Delete product from cart	The user proceeds to delete the previously added product in the cart.	X	The specified product's information is deleted from the current web page as well as from the database.	The specified product's information is deleted from the current web page as well as from the database.	Pass	Pass
8	Delete user information on admin session	On the admin session, the admin deletes the user account.	id = 3	The specified user's information is deleted from the current web page as well as from the database.	The specified user's information is deleted from the current web page as well as from the database.	Pass	Pass
9	Register with different password and confirm password.	Enter Password and Confirm Password with two different values.	"Test Case 02", "testcase02@gmail.com", " testcase2", "testcase02"	Show "Confirm password not matched!" message. The user must re-enter the correct information if they want to continue registering.	Show "Confirm password not matched!" message. The user must re-enter the correct information if they want to continue registering.	Pass	Pass
10	Log in with incorrect password	Import incorrect passwords that before that used to register an account.	"tester@gmail.com", "testcase03"	Show "Incorrect email or password!" message. The user must re-enter the correct information if they want to continue logging in.	Show "Incorrect email or password!" message. The user must re-enter the correct information if they want to continue logging in.	Pass	Pass

3. Screenshot in running

Test case 1: Registration (Pass)

I will register a new account with the requested information. I will start entering name, gmail, password and confirm password one by one according to the given value from test case template:

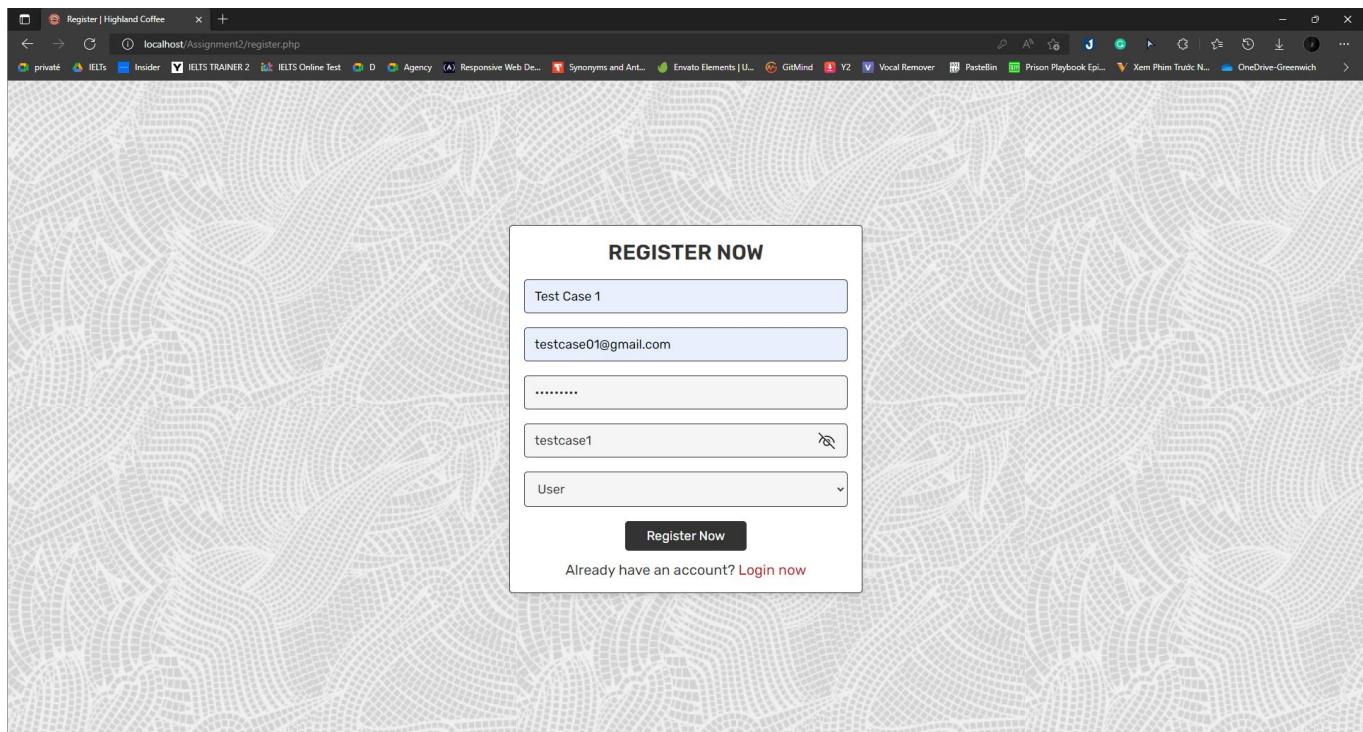
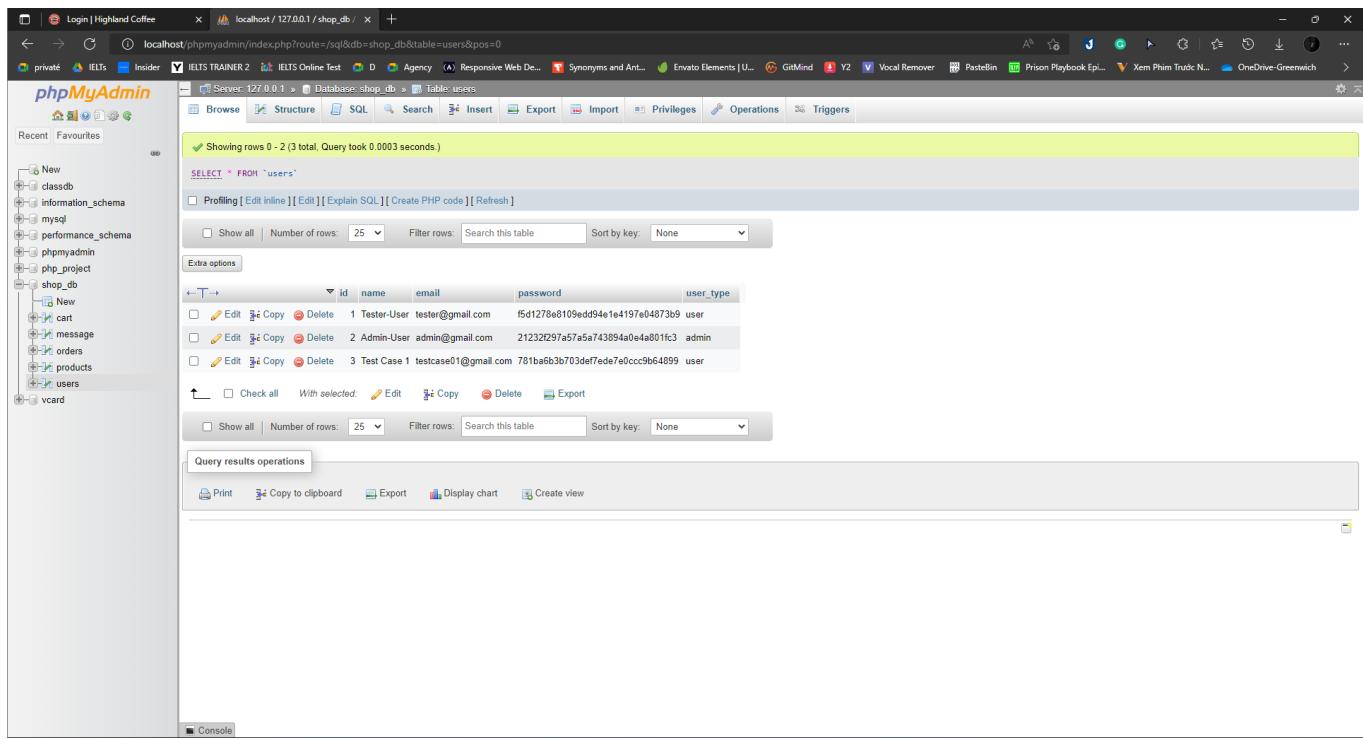


Figure 70: C3.3_Test case 1_Registration

After successfully registered, I press the “Register now” button, I will be taken to the “Log in” page to login with the newly created account.

To make sure that the user's information has been recorded in the database, I will go to phpMyAdmin to check in the "users" table.



The screenshot shows the phpMyAdmin interface connected to a MySQL database named 'shop_db'. The 'users' table is selected. The table structure includes columns: id, name, email, password, and user_type. The data shows three entries:

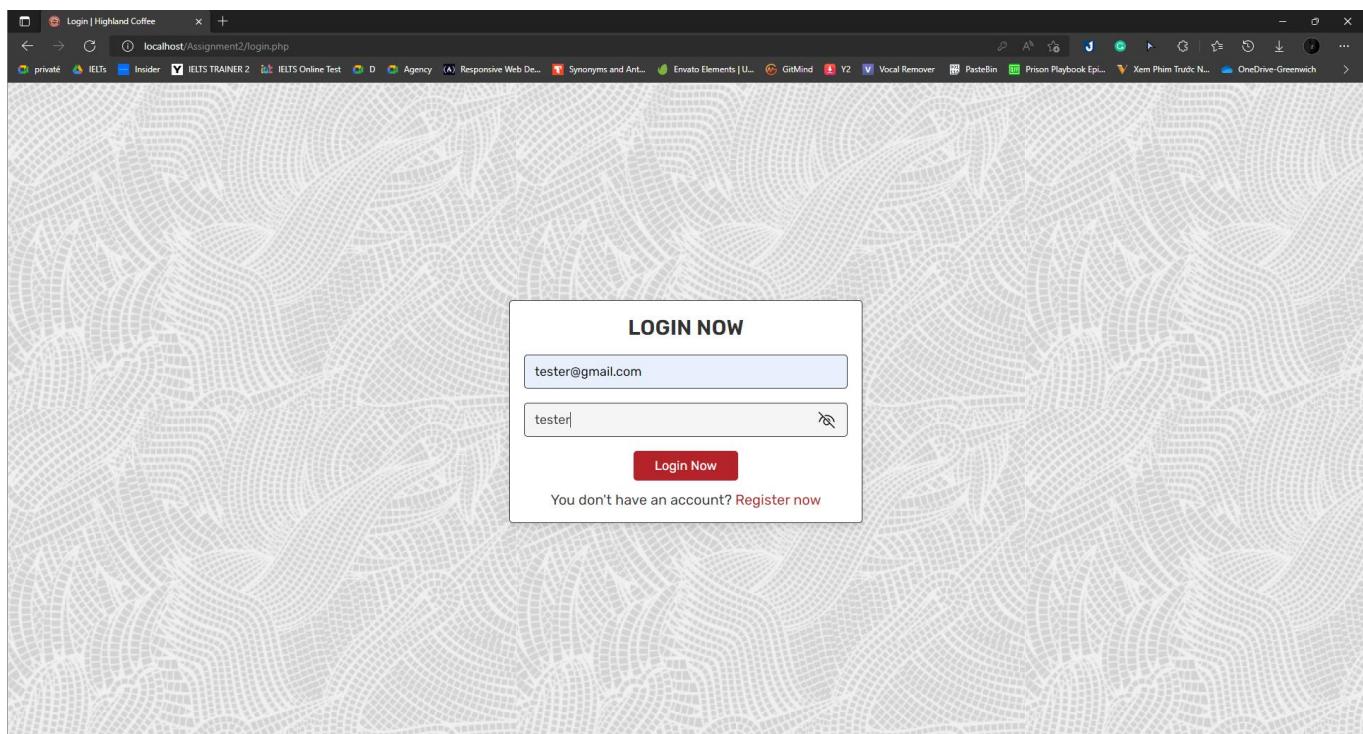
	id	name	email	password	user_type
1	Tester-User	tester@gmail.com	f5d1278e8109ed94e1e4197e04873b9		user
2	Admin-User	admin@gmail.com	21232f297a57a5a743694a0e4a801fc3		admin
3	Test Case 1	testcase01@gmail.com	781ba6b3b703def7ede7e0ccc9b64899		user

Figure 71: C3.3_Test case 1_Checking data on “users” table.

Indeed, all information entered in the registration form is displayed correctly in the database.

Test case 2: Log in for user session (Pass)

I will login with account for user session with given data from test case template:



The screenshot shows a 'LOGIN NOW' form on a website. The form fields are:

- Email: tester@gmail.com
- Password: tester
- Submit button: Login Now
- Link: You don't have an account? [Register now](#)

Figure 72: C3.3_Test case 2_Log in for user session.

After successfully logging in, the system forwarded me to the dashboard page for the user session.

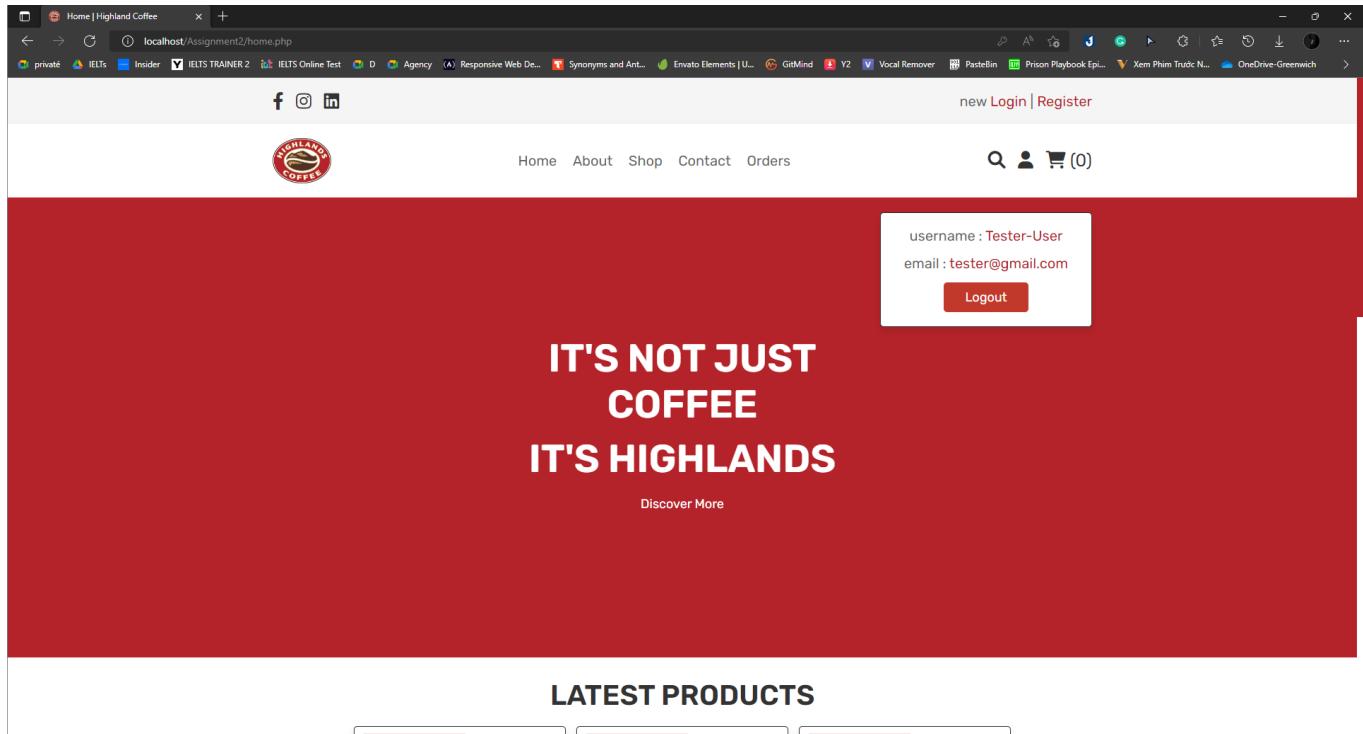


Figure 73: C3.3_Test case 2_Home page for user session.

The final result is that I have successfully logged into the user session.

Test case 3: Log in for admin session (Pass)

I will login with admin session account with given data from test case template:

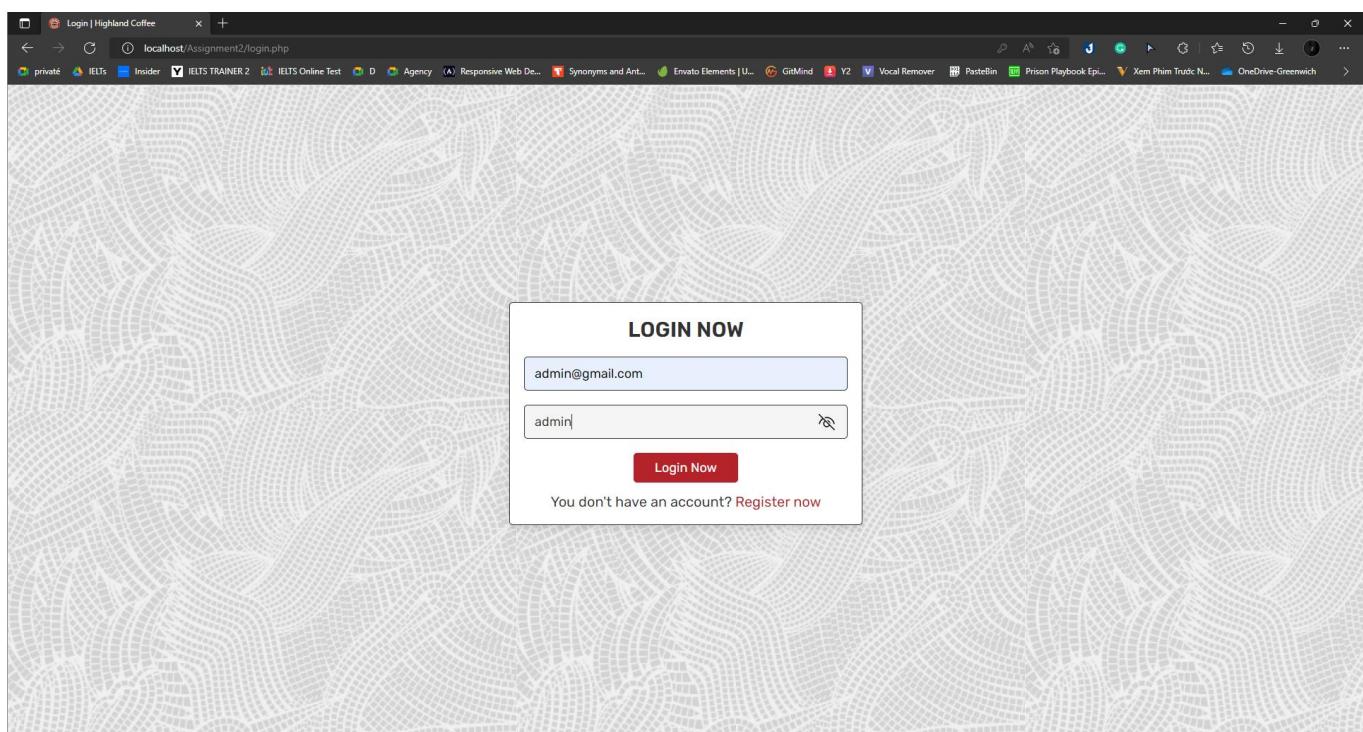


Figure 74: C3.3_Test case 3_Log in for admin session.

After successfully logging in, the system forwarded me to the dashboard for admin session.

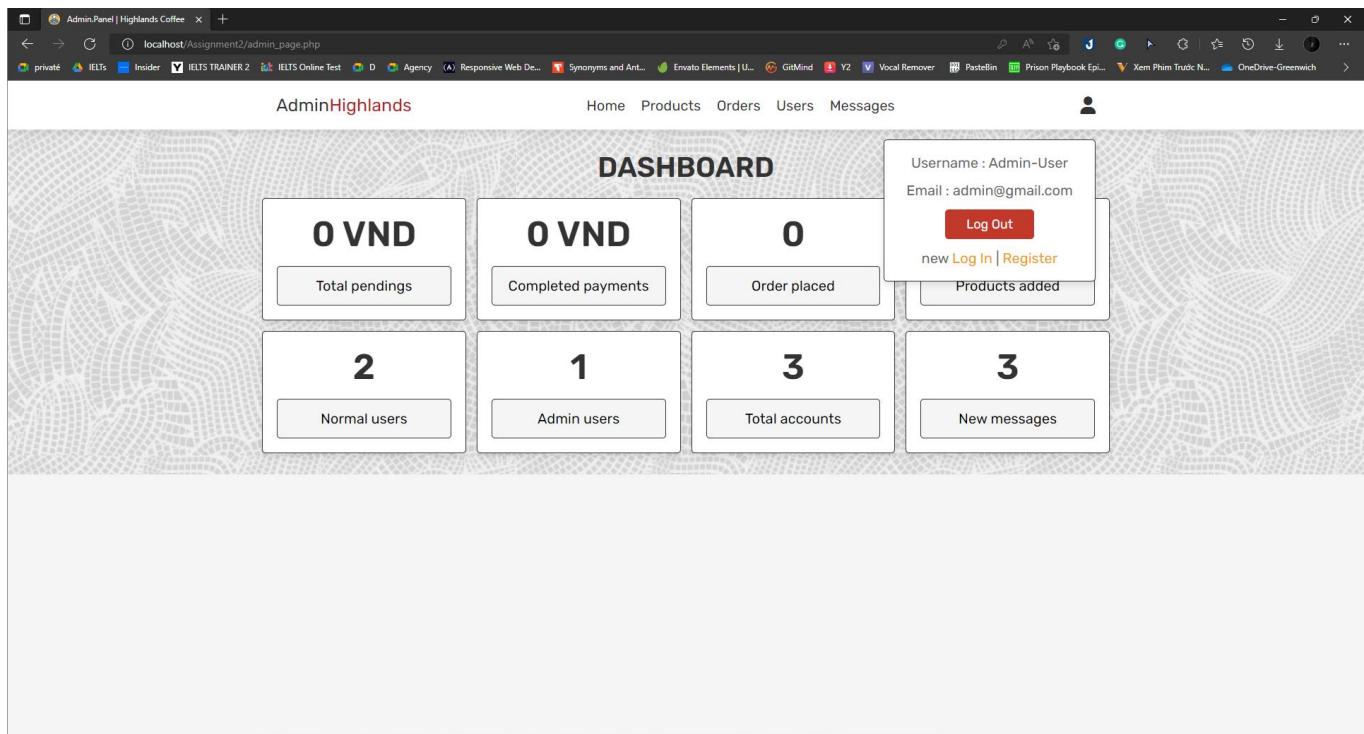


Figure 75: C3.3_Test case 3_Home page for admin session.

The end result was that I was successfully logged into the admin session.

Test case 4: Log out (Pass)

When I click the logout icon, it will get out of account.

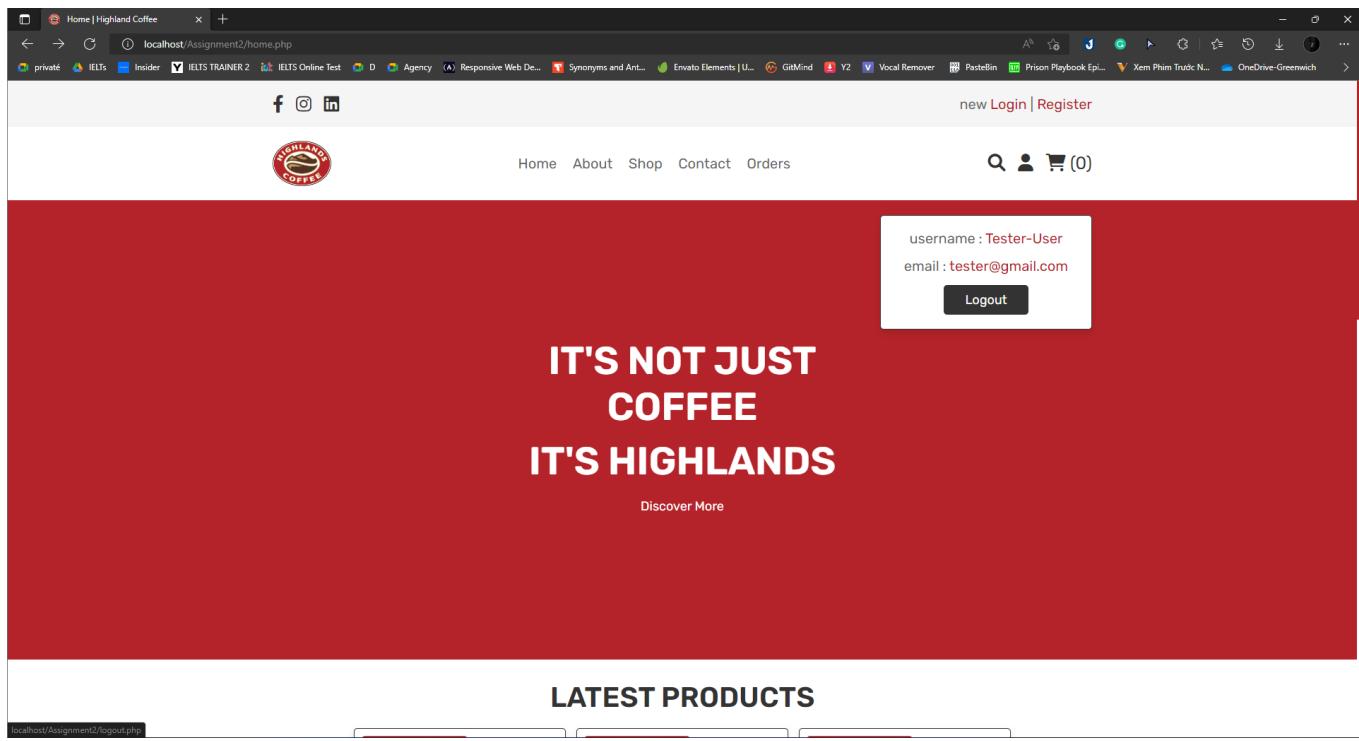


Figure 76: C3.3_Test case 4_Log out

After the account exits the system, the system redirects me to a page to log in.

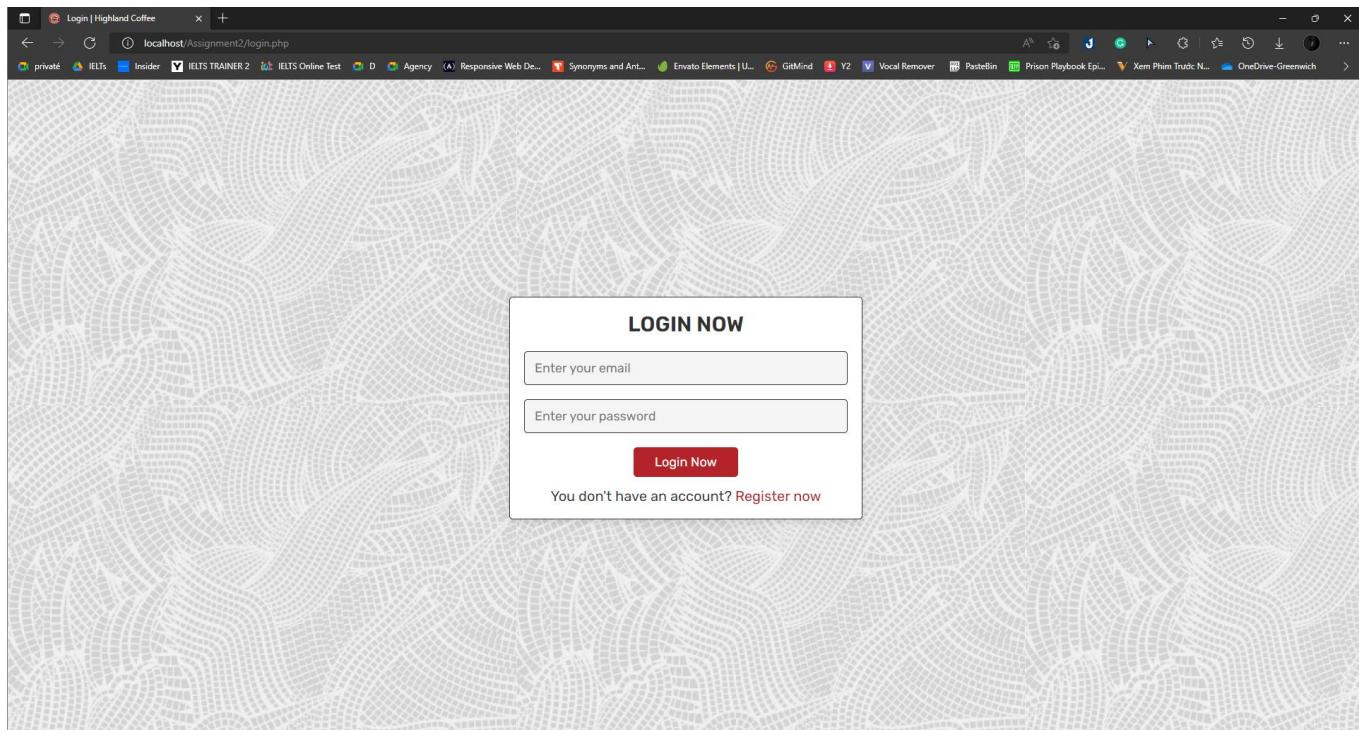


Figure 77: C3.3_Test case 4_Log out

The final result was that I was successfully logged out of the system.

Test case 5: View customer information on admin session (Pass)

I will proceed to view the information of all users and admins currently registered on admin session. After logging in, I will select the “Users” button in the navigation bar.

User id :	Username :	Email :	User type :
1	Tester-User	tester@gmail.com	user
2	Admin-User	admin@gmail.com	admin
3	Test Case 1	testcase01@gmail.com	user

Figure 78: C3.3_Test case 5_View all user information

To make sure that the user's information matches the database, I will go to phpMyAdmin to check in the "users" table.

	id	name	email	password	user_type
<input type="checkbox"/>	1	Tester-User	tester@gmail.com	f5d1278e8109ed94e1e4197e04873b9	user
<input type="checkbox"/>	2	Admin-User	admin@gmail.com	21232f297a575a743894a0e4a801fc3	admin
<input type="checkbox"/>	3	Test Case 1	testcase01@gmail.com	781ba6b3b703de7ede7e0ccc9b64899	user

Figure 79: C3.3 Test case 5 Checking data on “users” table.

Indeed, all user information created is displayed correctly in the database.

Test case 6: Update product information on admin session (Pass)

I will proceed to update the specified product information with the given data from the test case template.

Before the update

The screenshot shows a grid of five product cards. Each card contains an image, the product name, its price in VND, and two buttons: 'Update' and 'Delete'. The products are:

- Tea Peach Lemongrass: 45000 VND
- Tea Cerise Apricot: 55000 VND
- Tea Pink Guava: 55000 VND
- Tea Tropical Mango: 55000 VND
- Signature Tee: 500000 VND

The screenshot shows the 'products' table in phpMyAdmin. The table has columns: id, name, price, and image. The data is as follows:

	id	name	price	image
	26	Pastry Tiramisu	35000	pastry.details.tiramisu.jpg
	27	Tea Golden Lotus (Water Chestnut Topping)	45000	tea.details.chesnut-lotus.jpg
	28	Tea Golden Lotus (Lotus Seed Topping)	45000	tea.details.golden-lotus.jpg
	29	Tea Lychee	45000	tea.details.lychee.jpg
	30	Tea Peach Jelly	45000	tea.details.peach-jelly.jpg
	31	Tea Peach Lemongrass	45000	tea.details.peach-lemongrass.jpg
	32	Tea Cerise Apricot	55000	tea.details.cerise-apricot-tea.jpg
	33	Tea Pink Guava	55000	tea.details.pink-guava-tea.png
	34	Tea Tropical Mango	55000	tea.details.tropical-mango-tea.jpg
	35	Signature Tee	500000	tee.staff.jpg

Figure 80: C3.3_Test case 6_Before the update

Updating

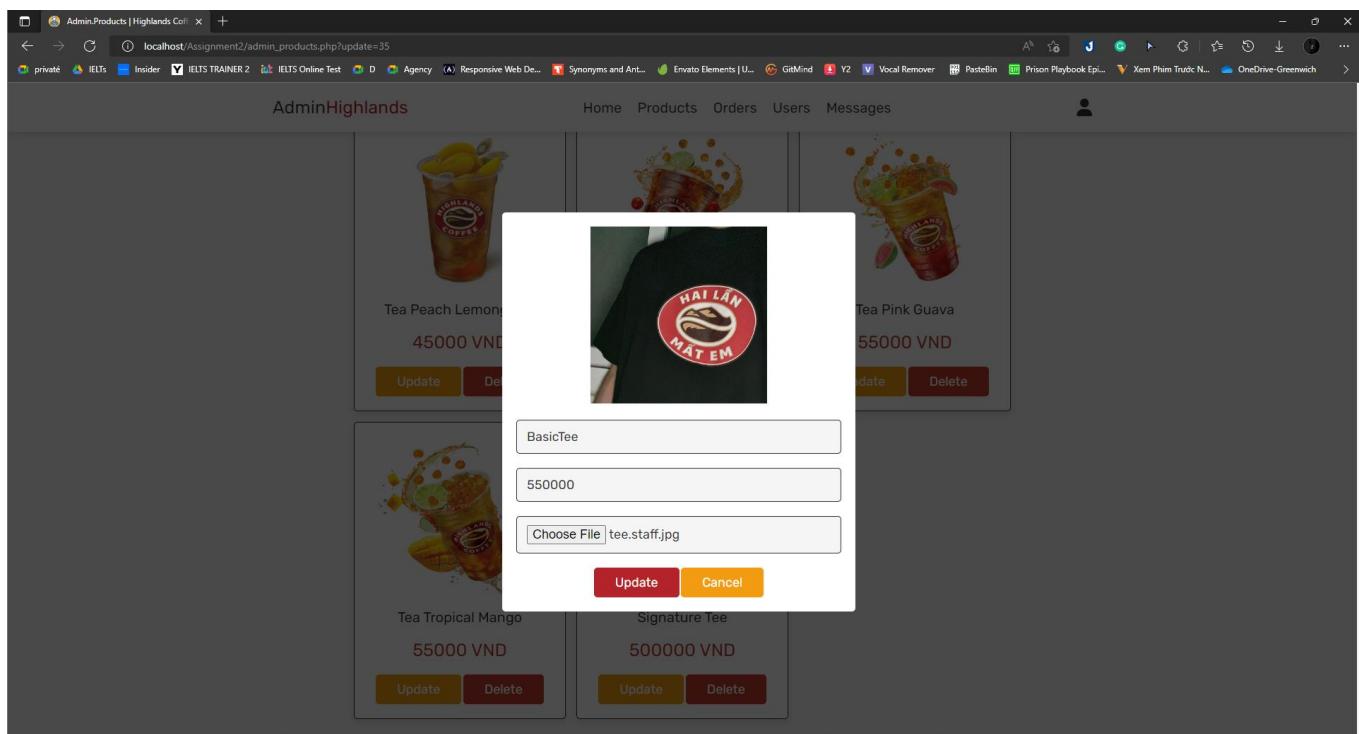


Figure 81: C3.3_Test case 6 Updating

After the update

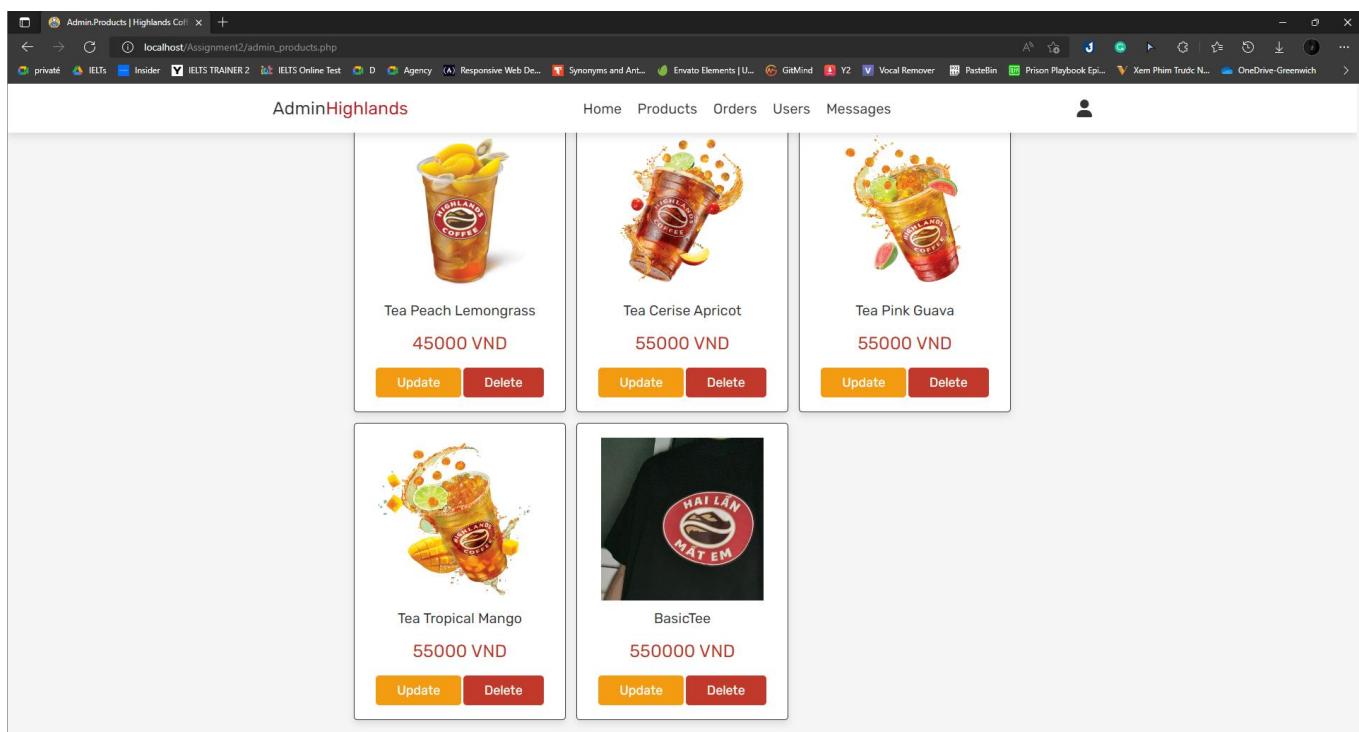


Figure 82: C3.3_Test case 6_After the update

To make sure that the product information after updating matches the database, I will go to phpMyAdmin to check in the "products" table.

	id	name	price	image
<input type="checkbox"/>	26	Pastry Tiramisu	35000	pastry.details.tiramisu.jpg
<input type="checkbox"/>	27	Tea Golden Lotus (Water Chestnut Topping)	45000	tea.details.chestnut-lotus.jpg
<input type="checkbox"/>	28	Tea Golden Lotus (Lotus Seed Topping)	45000	tea.details.golden-lotus.jpg
<input type="checkbox"/>	29	Tea Lychee	45000	tea.details.lychee.jpg
<input type="checkbox"/>	30	Tea Peach Jelly	45000	tea.details.peach-jelly.jpg
<input type="checkbox"/>	31	Tea Peach Lemongrass	45000	tea.details.peach-lemongrass.jpg
<input type="checkbox"/>	32	Tea Cerise Apricot	55000	tea.details.cerise-apricot-tea.jpg
<input type="checkbox"/>	33	Tea Pink Guava	55000	tea.details.pink-guava-tea.png
<input type="checkbox"/>	34	Tea Tropical Mango	55000	tea.details.tropical-mango-tea.jpg
<input type="checkbox"/>	35	BasicTee	650000	tee.staff.jpg

Figure 83: C3.3_Test case 6_Checking data on “products” table.

Indeed, the specified product's information is updated from the current web page as well as from the database.

Test case 7: Delete product from cart (Pass)

I will proceed to delete product from cart.

Before removing the product from the cart

The screenshot shows a shopping cart page with the title "SHOPPING CART" and a breadcrumb navigation "Home / Cart". The page displays three products added to the cart:

- Banh Mi Grilled Pork**: 19000 VND, quantity 1, Update button
- Espresso Cappuccino**: 65000 VND, quantity 1, Update button
- BasicTee**: 550000 VND, quantity 1, Update button

The screenshot shows the "cart" table in the "shop_db" database via phpMyAdmin. The table structure includes columns: id, user_id, name, price, quantity, and image. The data is as follows:

	id	user_id	name	price	quantity	image
1	64	1	Banh Mi Grilled Pork	19000	1	banhmi.details.grilled-pork.jpg
2	65	1	Espresso Cappuccino	65000	1	coffee.details.cappuccino.jpg
3	66	1	BasicTee	550000	1	tee.staff.jpg

Figure 84: C3.3_Test case 7_Before removing the product from the cart

While removing product from cart

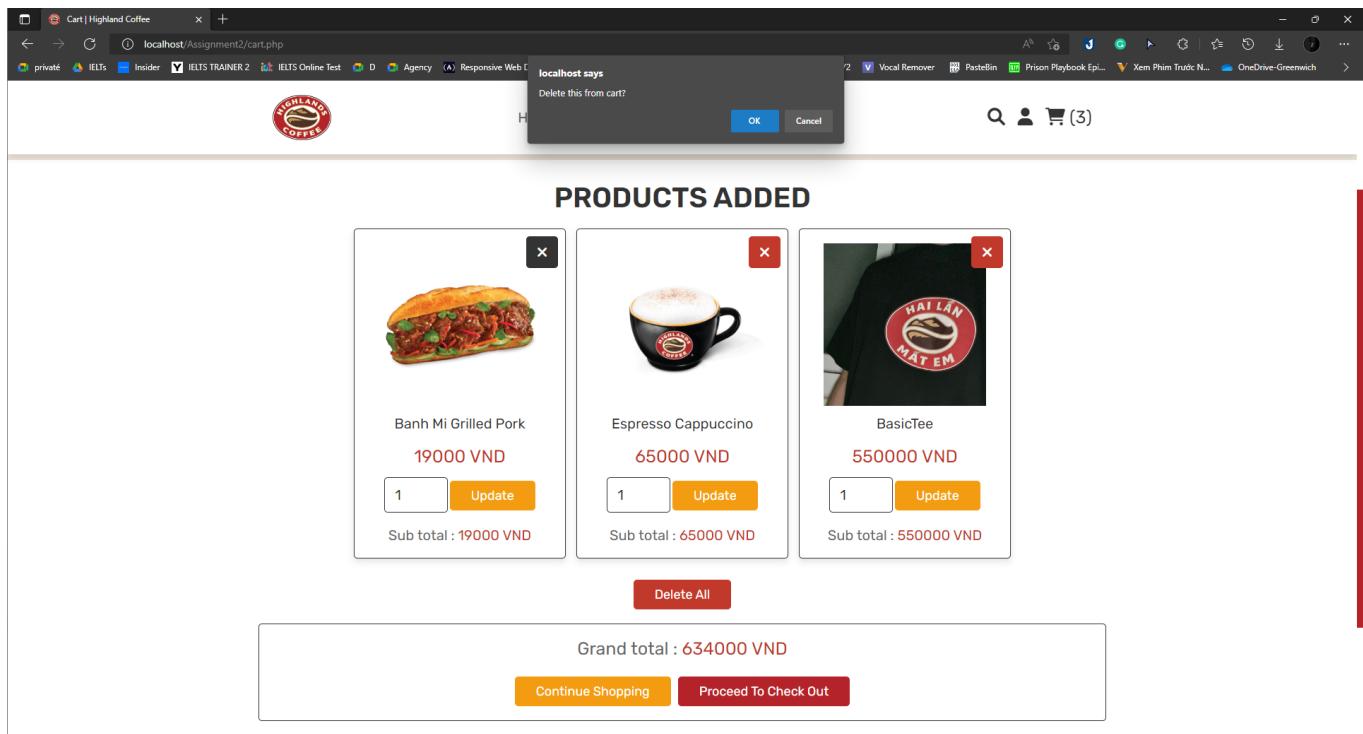


Figure 85: C3.3_Test case 7_While removing product from cart

After removing the product from the cart

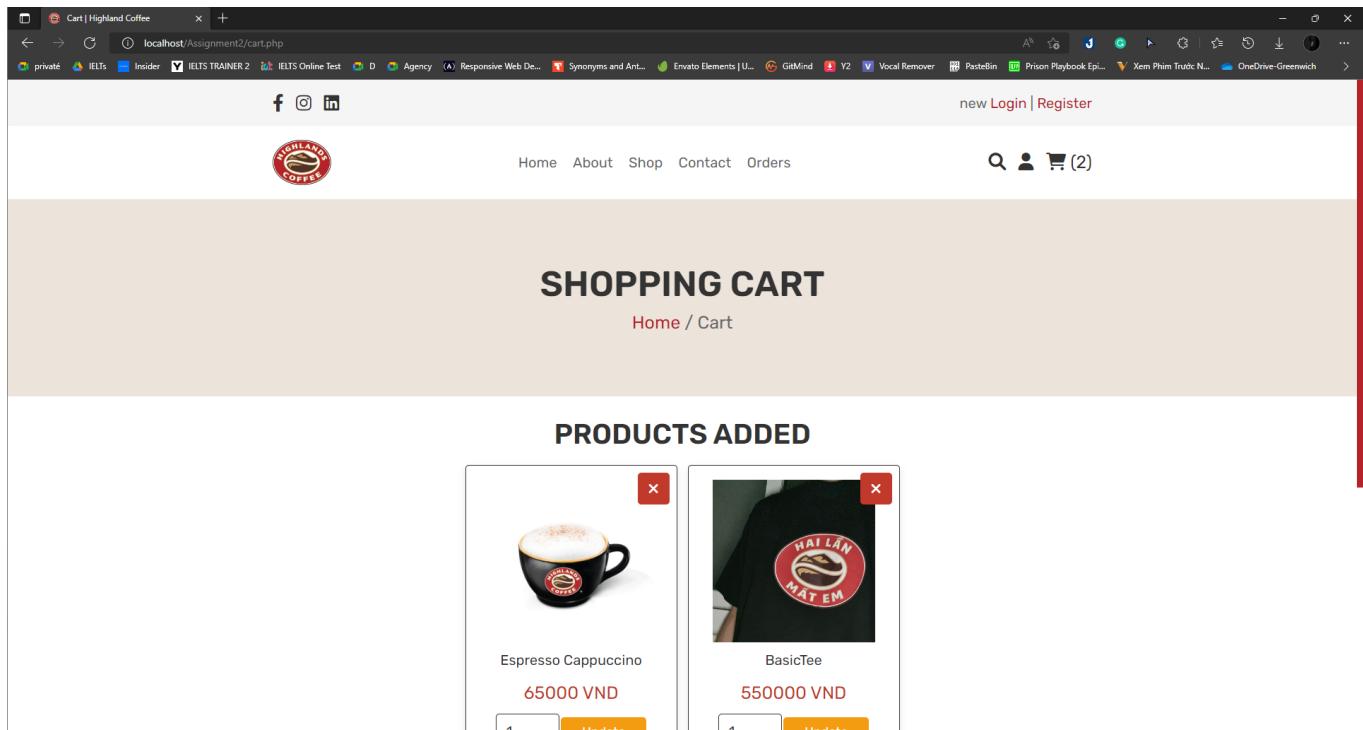
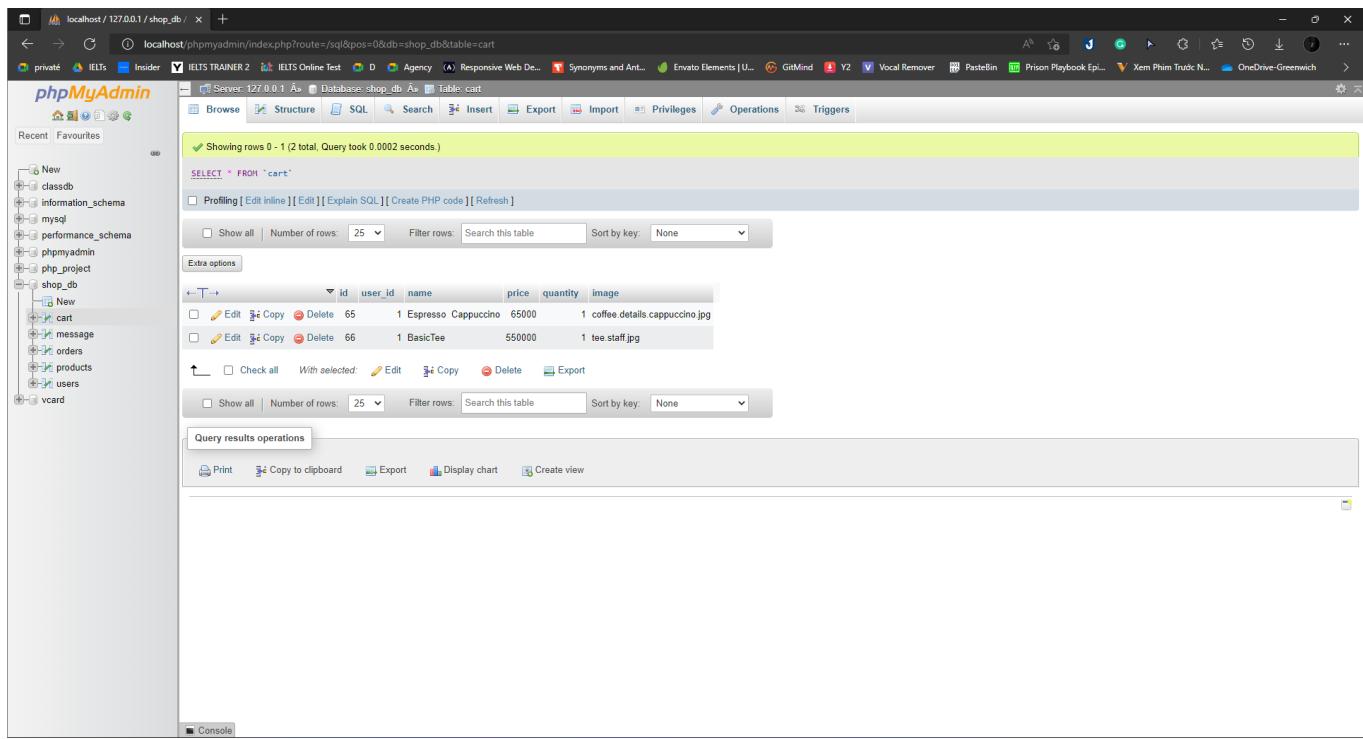


Figure 86: C3.3_Test case 7_After removing the product from the cart

To make sure that the information of the cart after deleting matches the database, I will go to phpMyAdmin to check in the "cart" table.



The screenshot shows the phpMyAdmin interface connected to a MySQL database named 'shop_db'. The 'cart' table is selected. The table has columns: id, user_id, name, price, quantity, and image. Two rows are present:

	id	user_id	name	price	quantity	image
	65	1	Espresso Cappuccino	65000	1	coffee.details.cappuccino.jpg
	66	1	BasicTee	550000	1	tee.staff.jpg

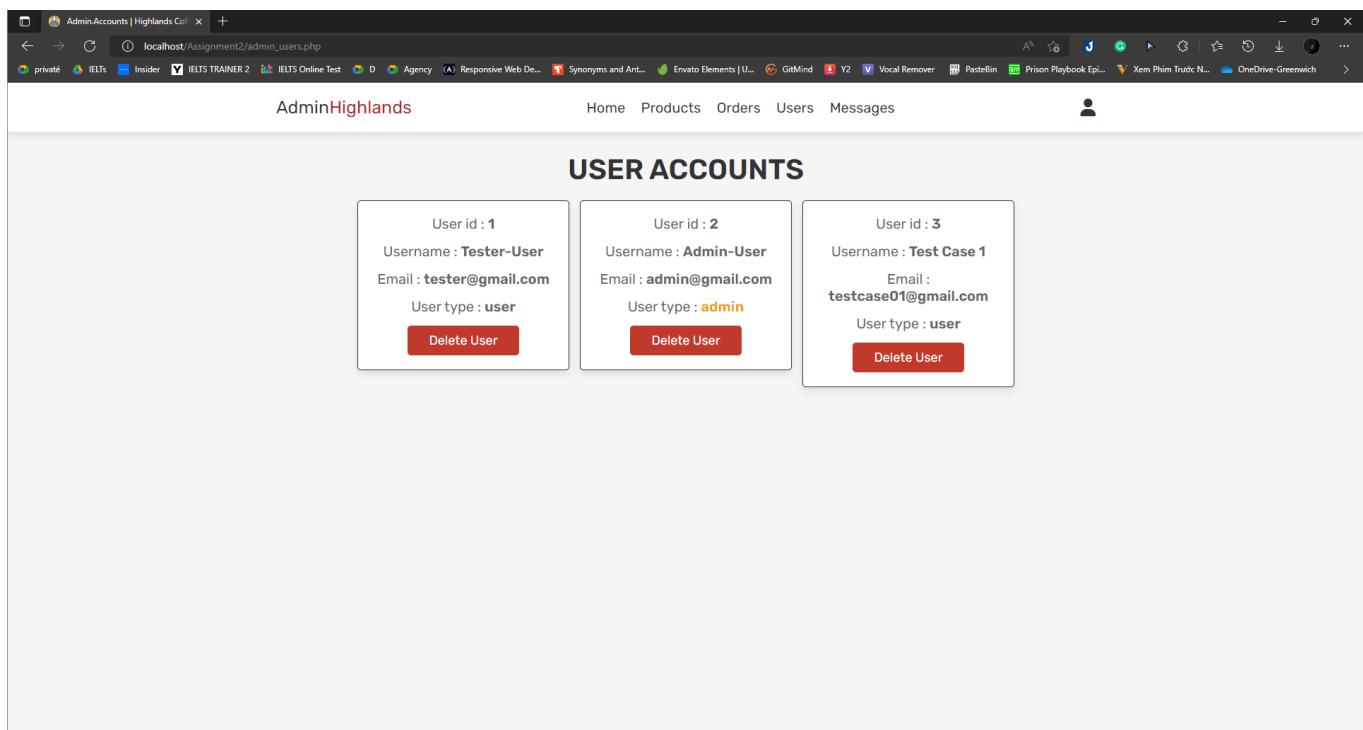
Figure 87: C3.3_Test case 7_Checking data on “cart” table.

Indeed, the item has been removed from the cart from the current web page as well as from the database.

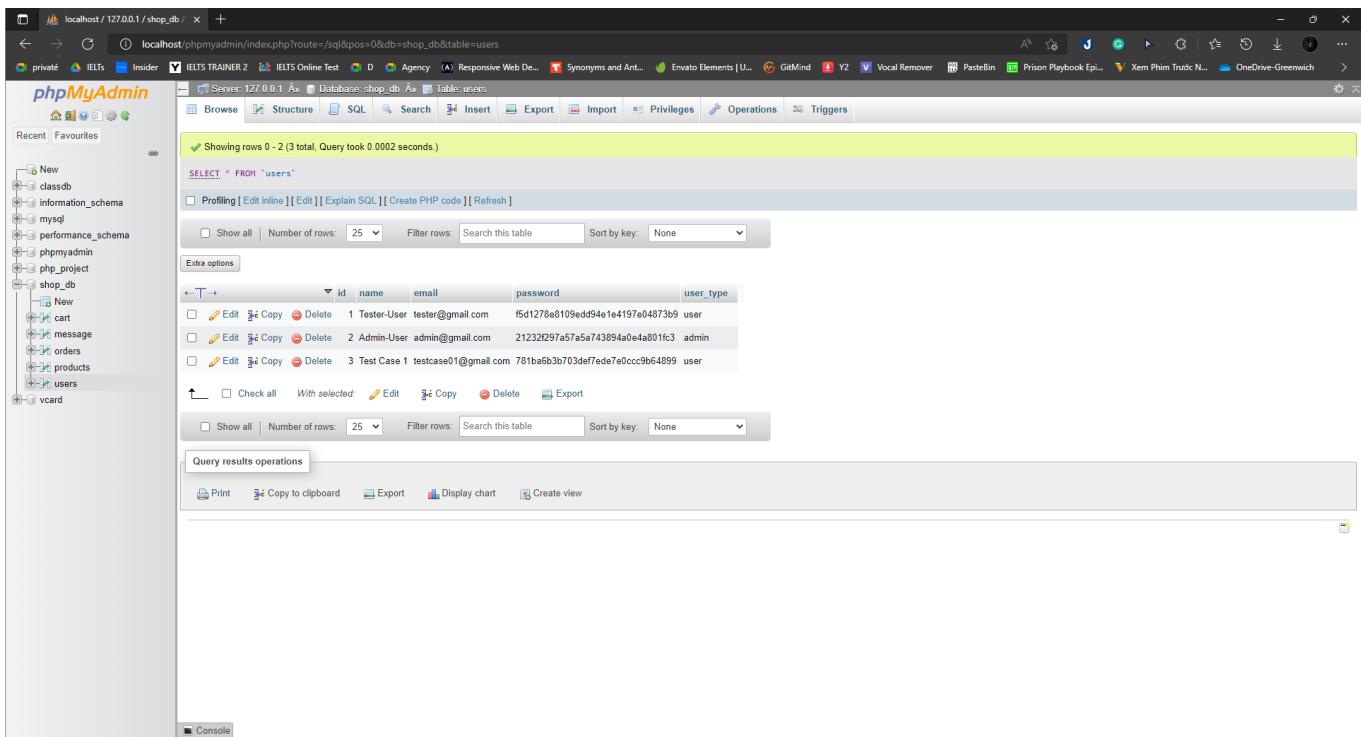
Test case 8: Delete user information on admin session (Pass)

I will proceed to delete user information on the admin session.

Before deleting user information with given data from test case template:



User id :	Username :	Email :	User type :
1	Tester-User	tester@gmail.com	user
2	Admin-User	admin@gmail.com	admin
3	Test Case 1	testcase01@gmail.com	user

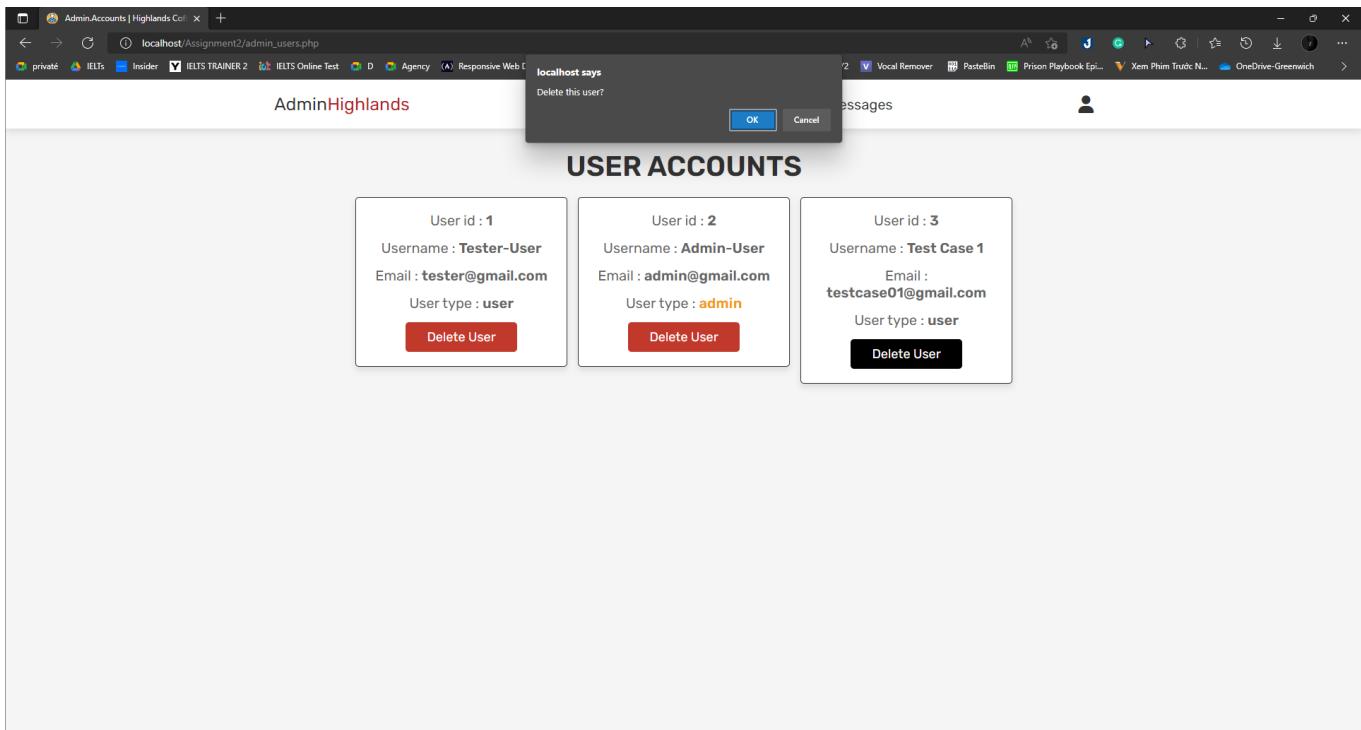


The screenshot shows the phpMyAdmin interface connected to a MySQL database named 'shop_db'. The 'users' table is selected, displaying three rows of data:

	id	name	email	password	user_type
1	1	Tester-User	tester@gmail.com	f5d1278e8109edd94e1e4197e04873b9	user
2	2	Admin-User	admin@gmail.com	21232f297a57a5a743694a0e4a801fc3	admin
3	3	Test Case 1	testcase01@gmail.com	781ba6b3b703def7ede7e0ccc9b64899	user

Figure 88: C3.3_Test case 8_Before deleting user information

While deleting user information



The screenshot shows a web application titled 'AdminHighlands' with a 'USER ACCOUNTS' section. It lists three user accounts:

- User id : 1
Username : Tester-User
Email : tester@gmail.com
User type : user
[Delete User](#)
- User id : 2
Username : Admin-User
Email : admin@gmail.com
User type : admin
[Delete User](#)
- User id : 3
Username : Test Case 1
Email : testcase01@gmail.com
User type : user
[Delete User](#)

A modal dialog box is displayed in the center, asking 'Delete this user?' with 'OK' and 'Cancel' buttons.

Figure 89: C3.3_Test case 8_While deleting user information

After deleting user information

User id :	Username :	Email :	User type :
1	Tester-User	tester@gmail.com	user
2	Admin-User	admin@gmail.com	admin

Figure 90: C3.3_Test case 8_After deleting user information

To make sure that the user's information after deleting matches the database, I will go to phpMyAdmin to check in the "users" table.

id	name	email	password	user_type
1	Tester-User	tester@gmail.com	f5d127e8109edd94e1e4197e04873b9	user
2	Admin-User	admin@gmail.com	21232f297a57a5a743894a0e4a801fc3	admin

Figure 91: C3.3_Test case 8_Checking data on "cart" table.

Indeed, that specified user information has been removed from the system from the current web page as well as from the database.

Test case 9: Register with different password and confirm password. (Pass)

I will proceed to create an account by entering different Password and Confirm Password with given data from test case template:

The screenshot shows a 'REGISTER NOW' form with the following fields filled:

- Name: Test Case 02
- Email: testcase02@gmail.com
- Confirm your password: testcase2 (highlighted in red)
- Password: (empty field)
- User Type: User

A red error message 'Confirm password not matched!' is displayed below the 'Confirm your password' field.

The screenshot shows the same 'REGISTER NOW' form, but the 'Confirm your password' field now contains 'testcase02', matching the value in the 'Password' field. The red highlight and error message are no longer present.

Figure 92: C3.3_Test case 9_Register with different password and confirm password

The system will display a message "Confirm password not matched!". I must re-enter the correct information if I want to continue registering.

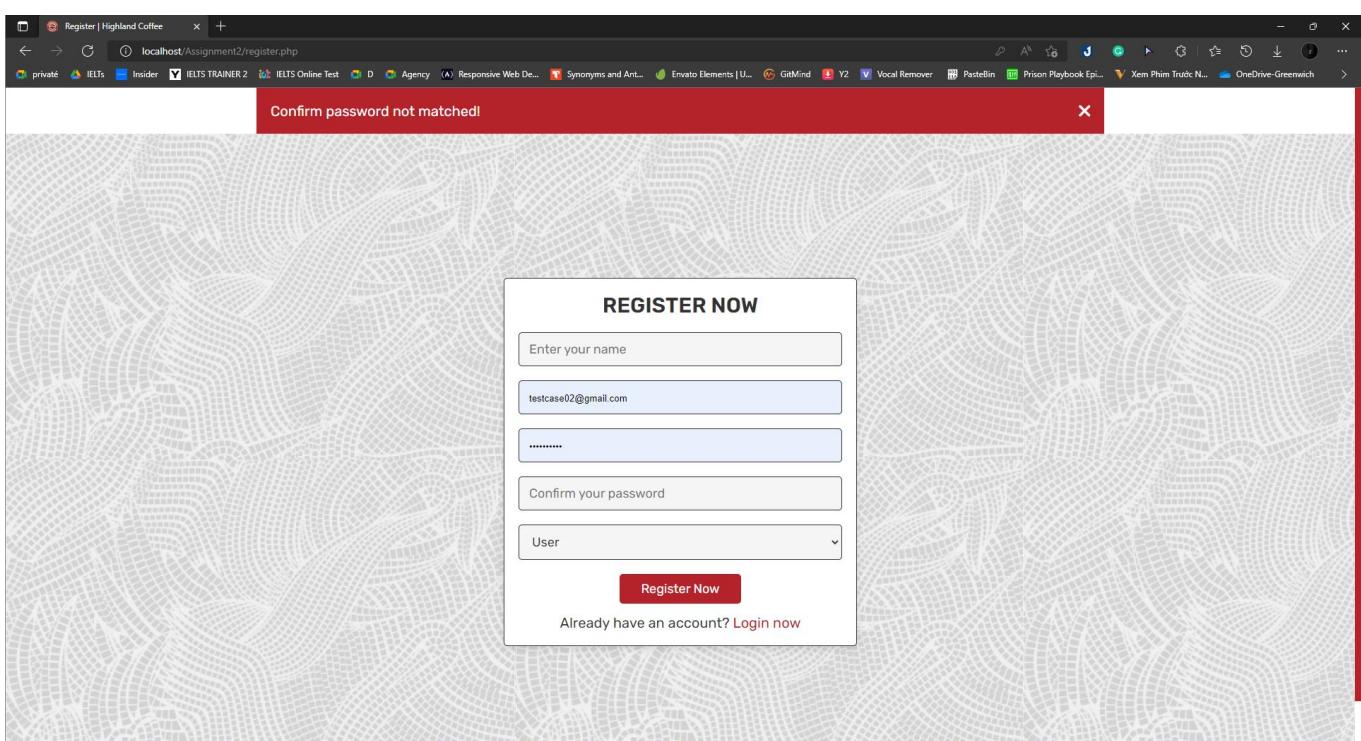


Figure 93: C3.3_Test case 9_Message “Confirm password not matched!”

To make sure that the user's information is not created in the database, I will go to phpMyAdmin to check in the "users" table.

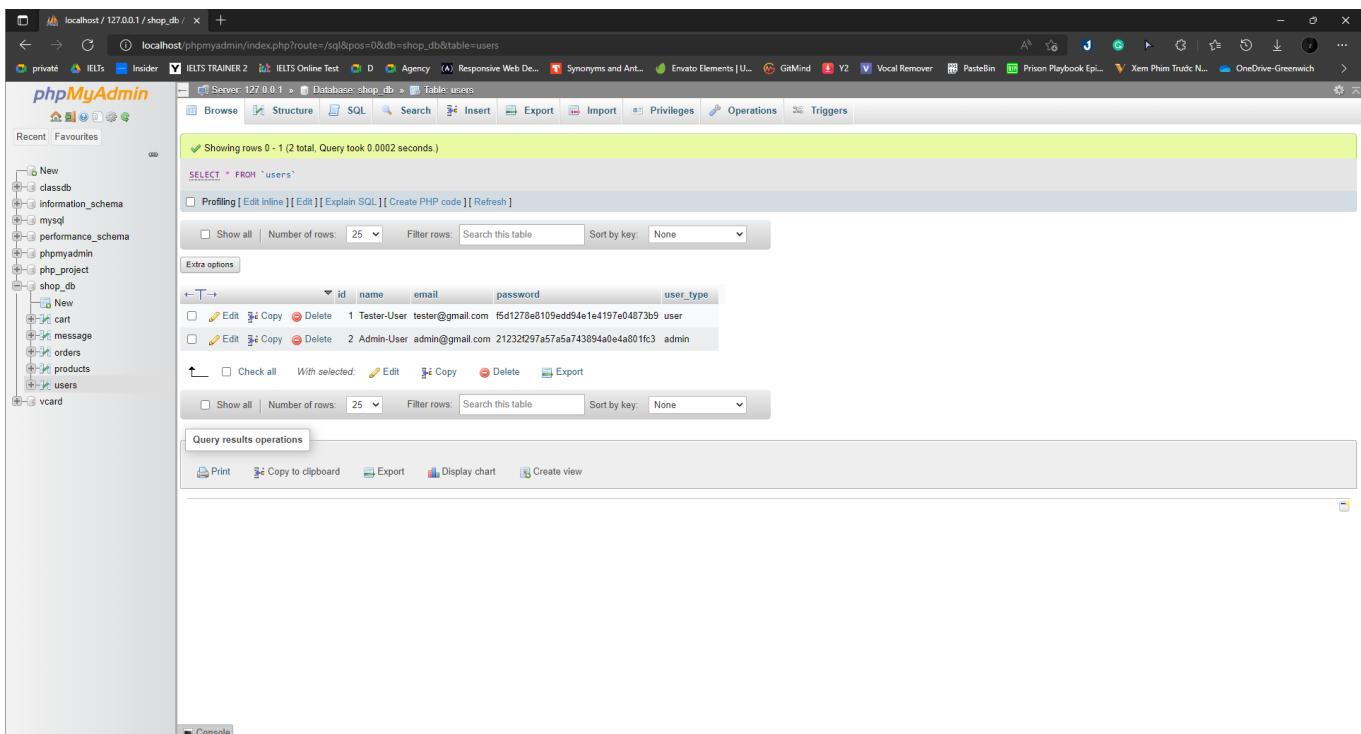


Figure 94: C3.3_Test case 9_Checking data on “users” table.

Indeed, that specified user information is not recorded from the current web page as well as from the database.

Test case 10: Log in with incorrect password (Pass)

With a user session account with email 'tester@gmail.com' and the correct password created earlier is 'tester', I will proceed to login with a wrong password with the given data from the test case template:

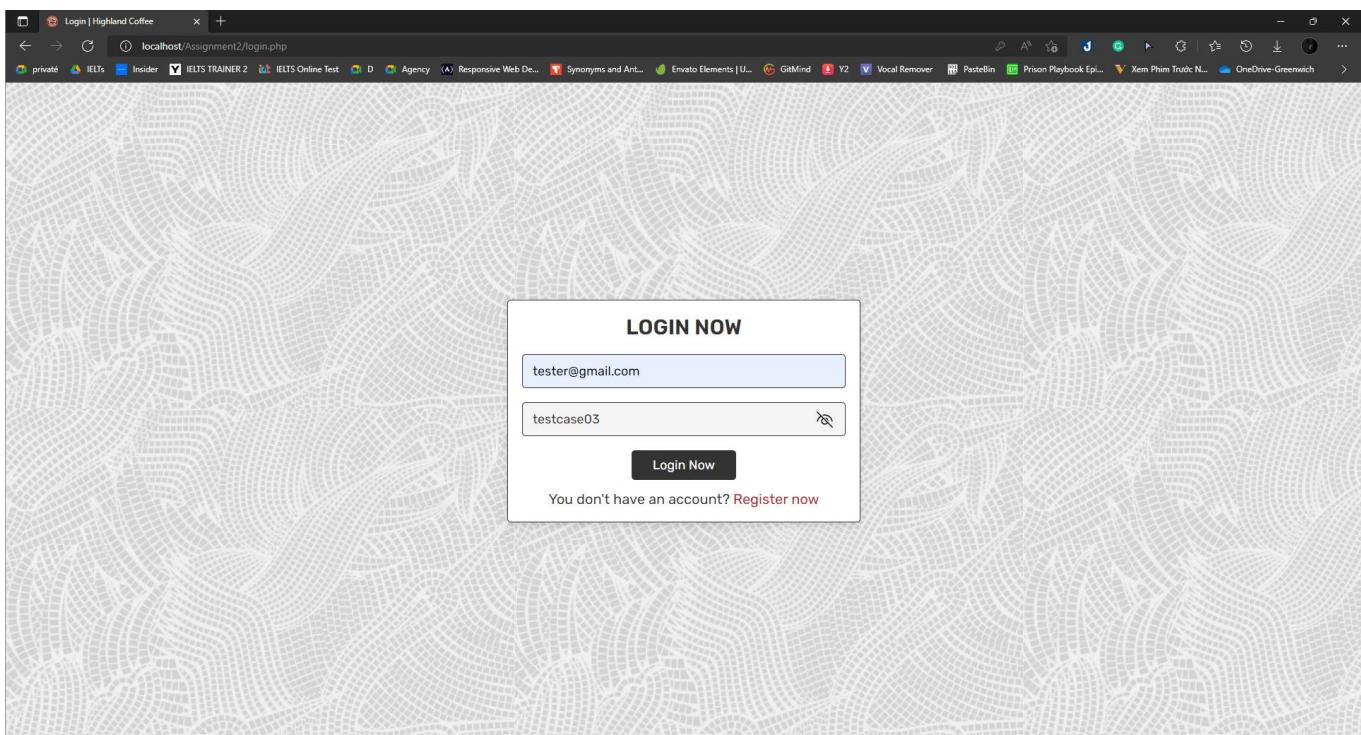


Figure 95: C3.3_Test case 10_Log in with incorrect password

After entering the data, the system will display a message "Incorrect email or password!". I must re-enter the correct information if I want to continue log in.

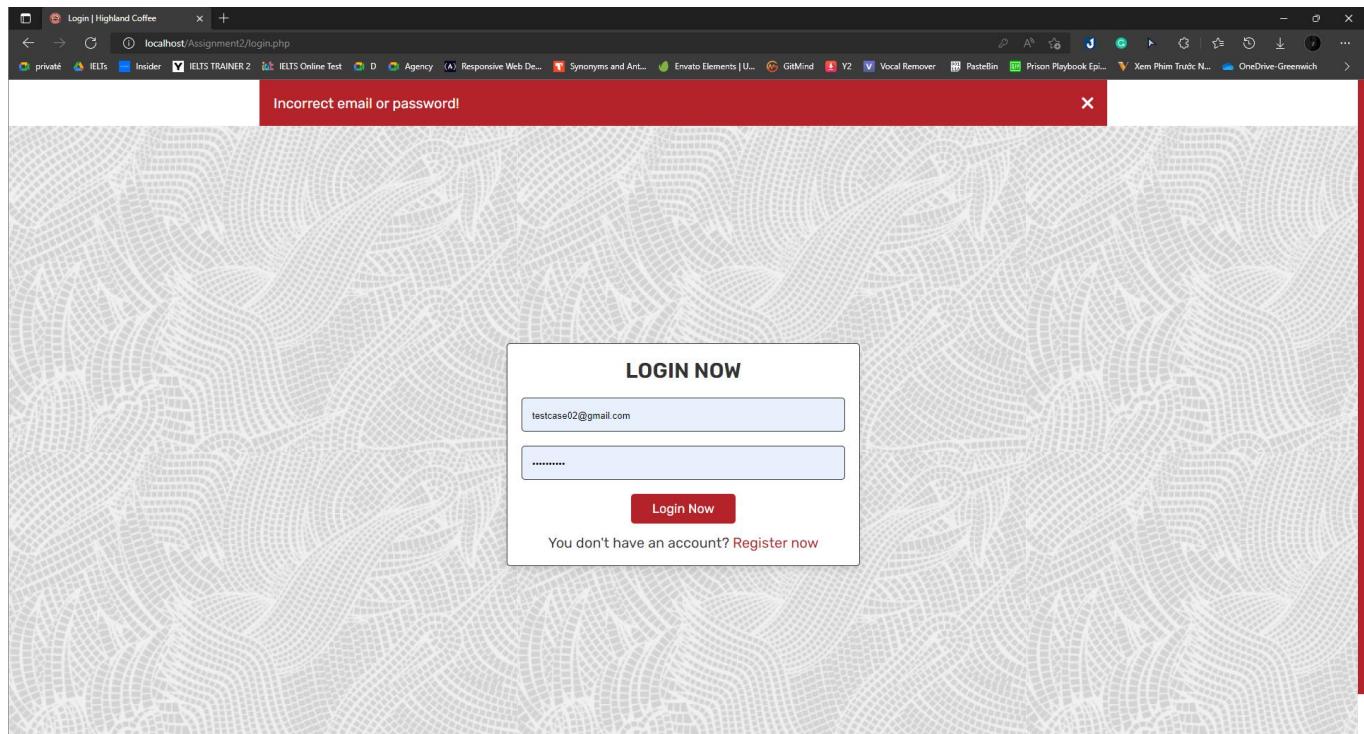
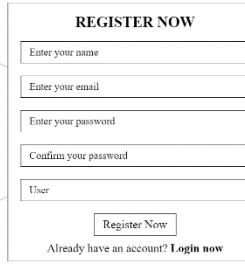


Figure 96: C3.3 _Test case 10 _Message “Incorrect email or password!”

Indeed, I cannot log into the system if I enter the wrong information.

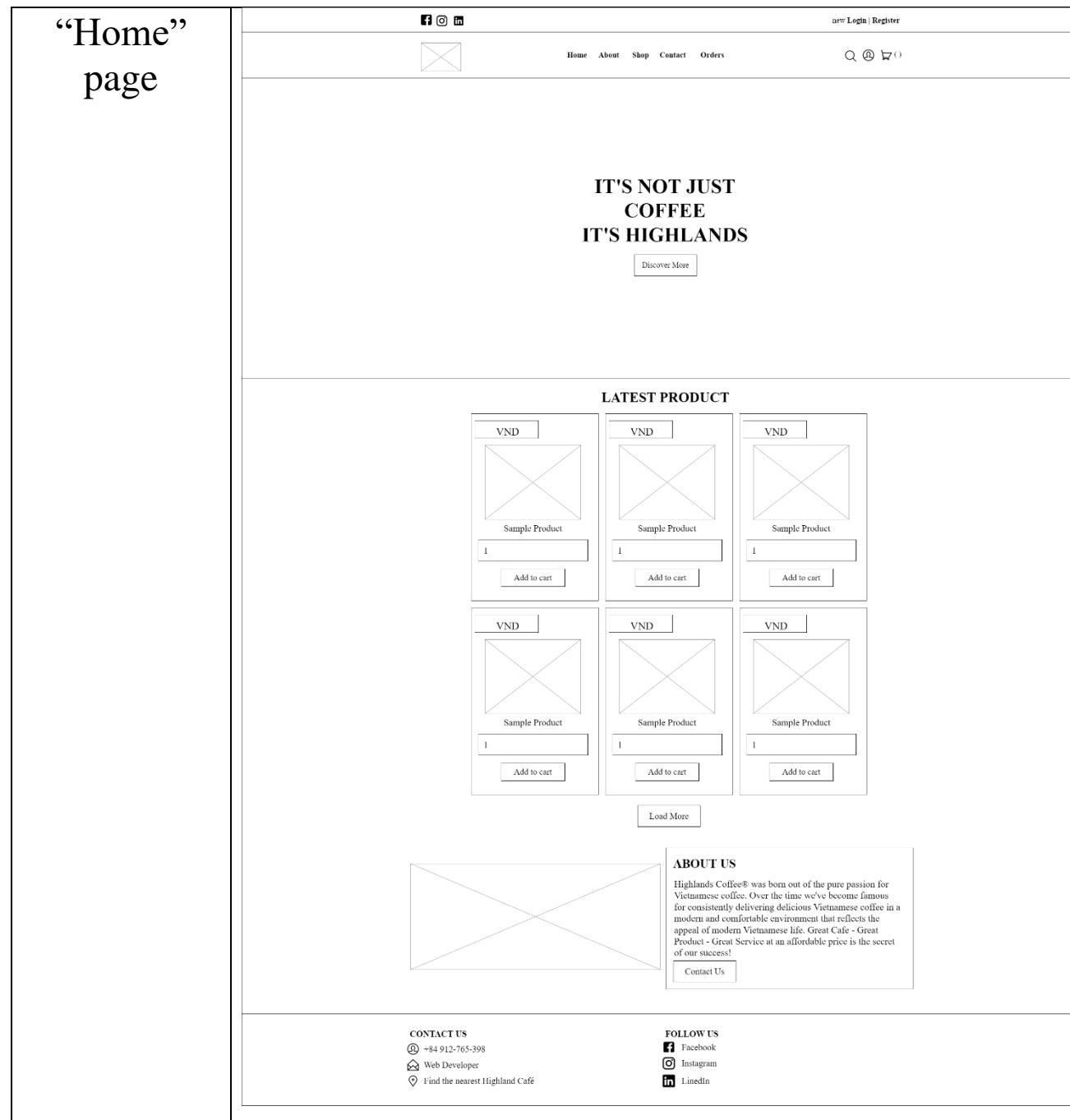
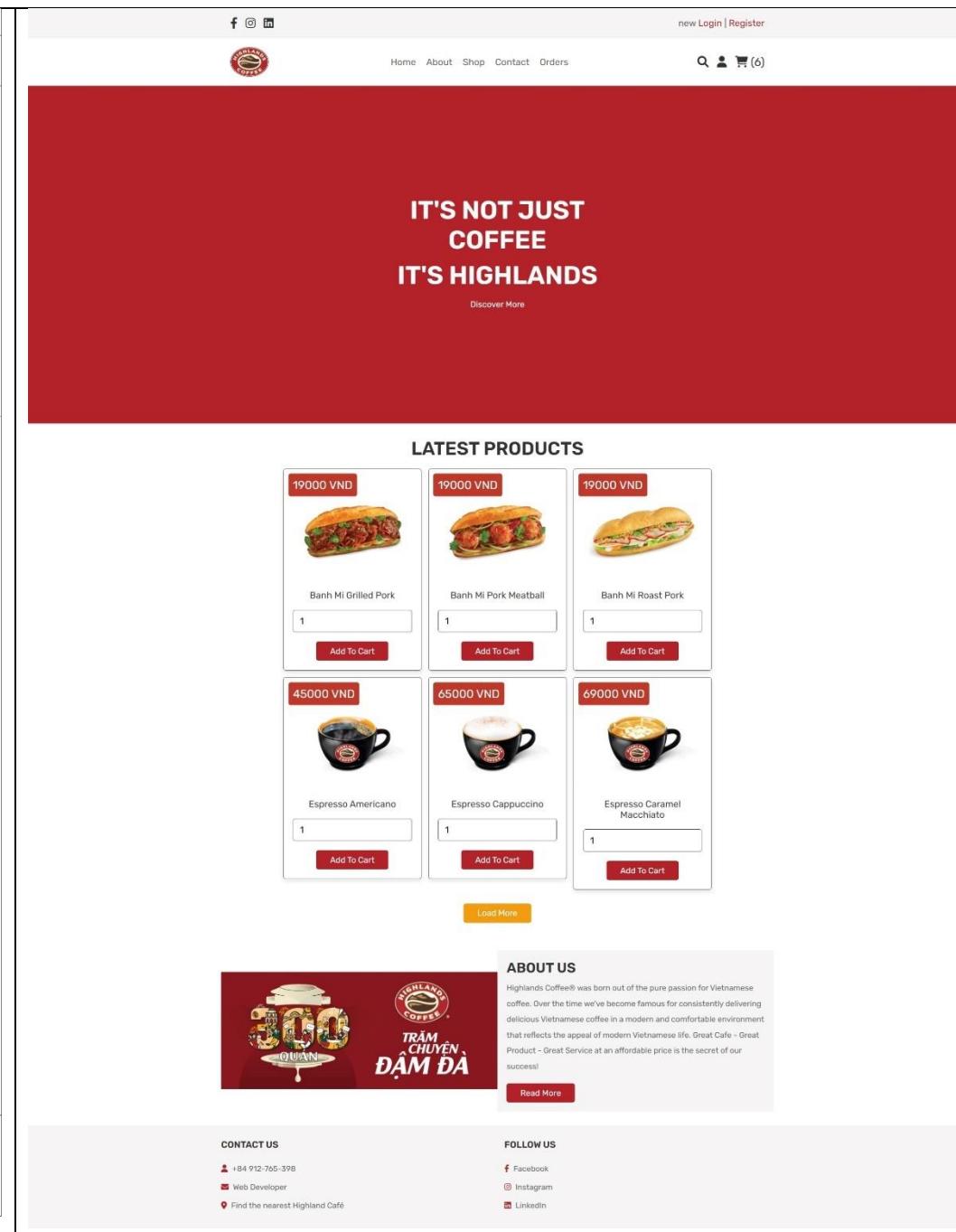
Chapter 4: Compare and contrast the multipage website created to the design document. (M4)

Table 3: C4_Comparison table of wireframe and actual web page

Categories	Wireframe	Web page	Comment	Status
“Register” page			The wireframe is supposed to look exactly like the web page.	Similar
“Log In” page			The wireframe is supposed to look exactly like the web page.	Similar

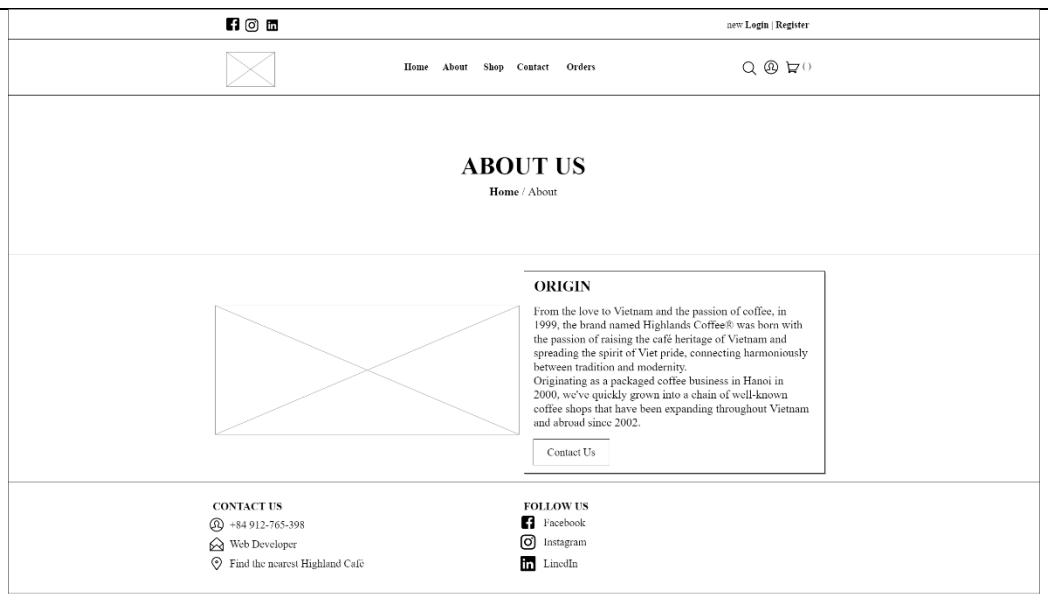
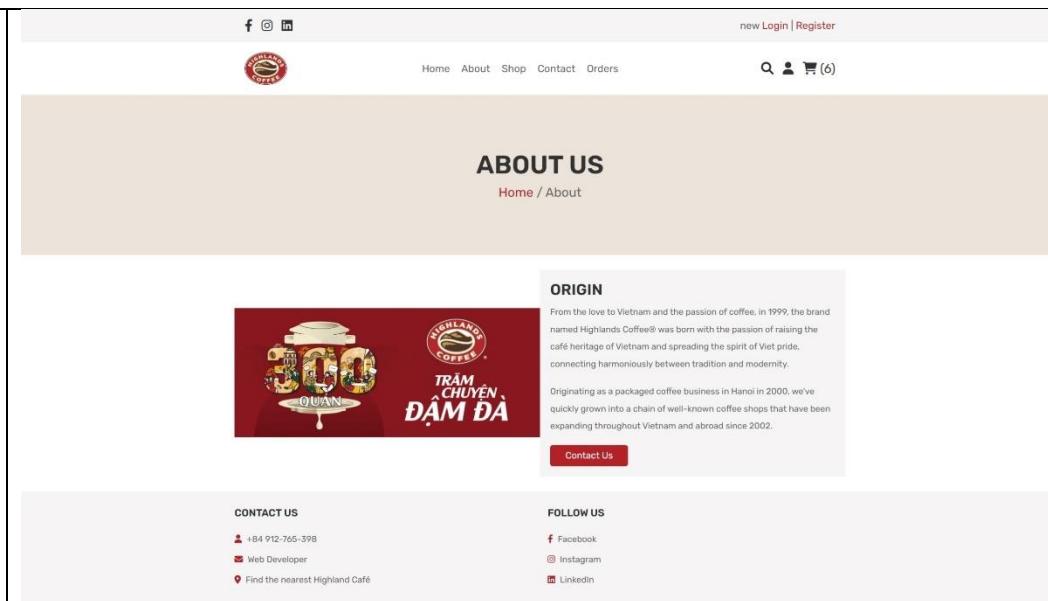
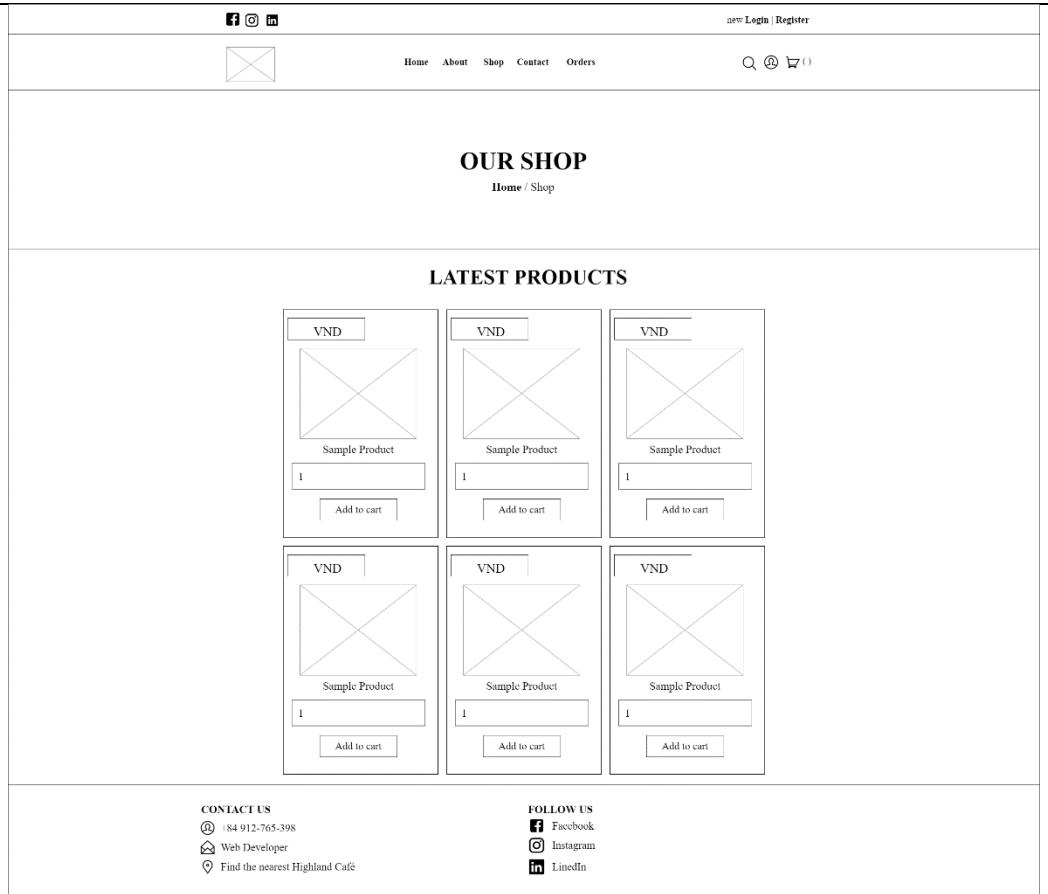
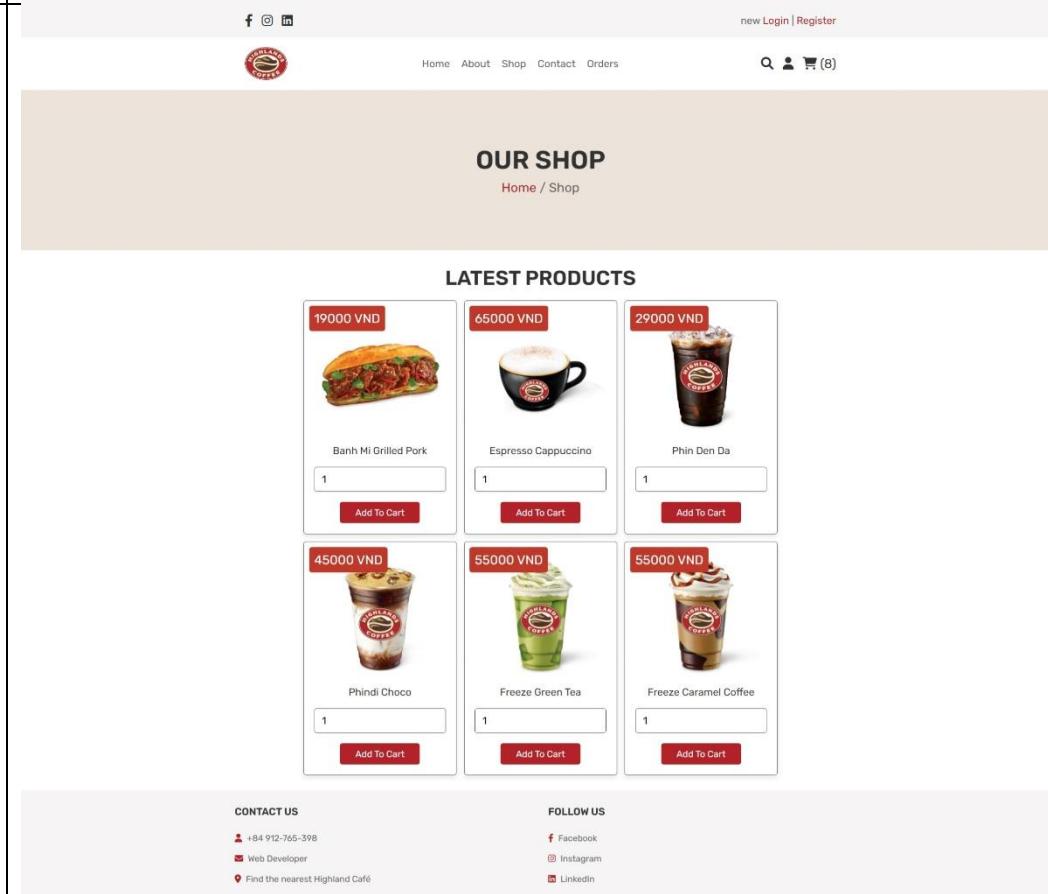
User Session

“Home”
page

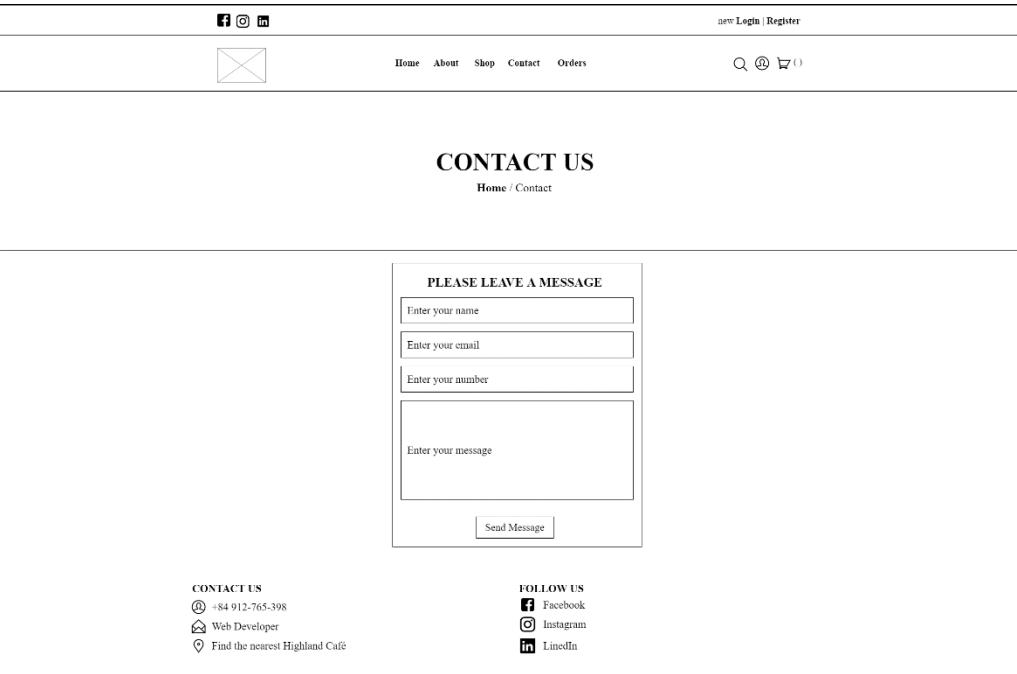



The
wireframe is
supposed to
look exactly
like the web
page.

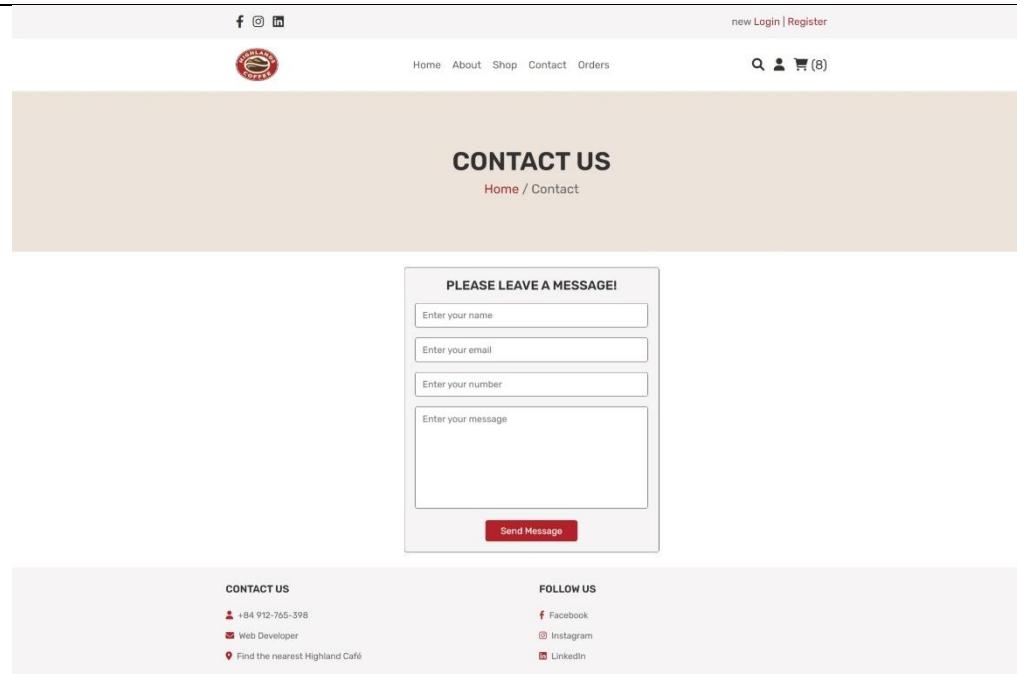
Similar

"About" page			<p>The wireframe is supposed to look exactly like the web page.</p>
"Shop" page			<p>The wireframe is supposed to look exactly like the web page.</p>

“Contact” page



The wireframe shows a contact form titled "PLEASE LEAVE A MESSAGE". It includes fields for "Enter your name", "Enter your email", "Enter your number", and a large area for "Enter your message". A "Send Message" button is at the bottom. The page has a header with social media icons and a navigation bar with links like Home, About, Shop, Contact, Orders, and a search bar.

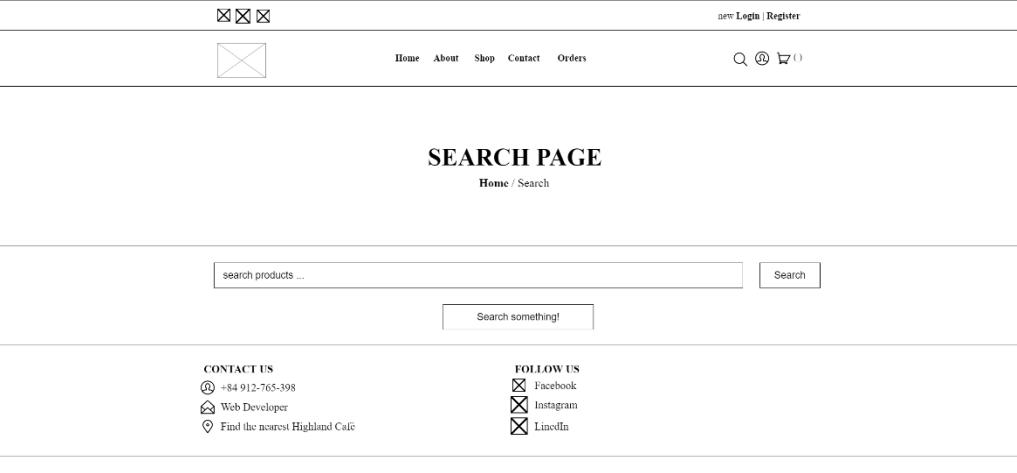


The wireframe shows a contact form titled "PLEASE LEAVE A MESSAGE". It includes fields for "Enter your name", "Enter your email", "Enter your number", and a large area for "Enter your message". A "Send Message" button is at the bottom. The page has a header with social media icons, a navigation bar with links like Home, About, Shop, Contact, Orders, and a search bar showing "(8)".

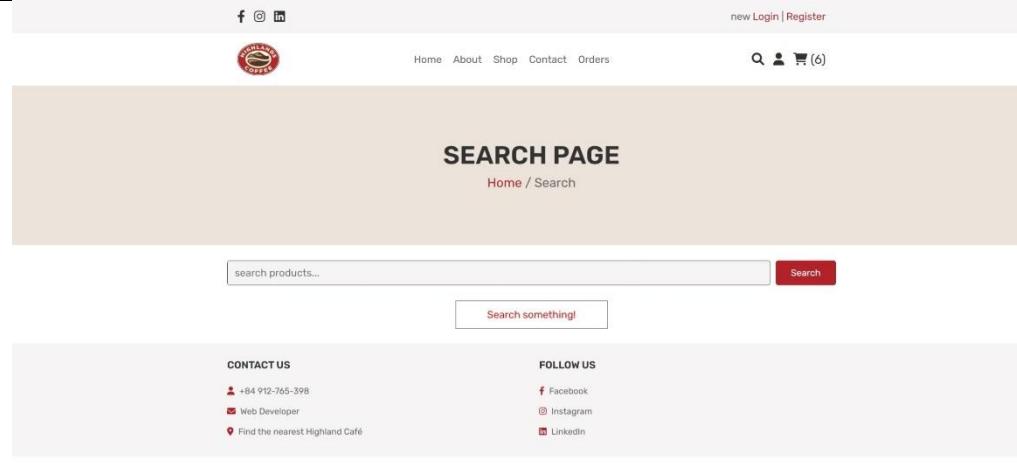
The wireframe is supposed to look exactly like the web page.

Similar

“Search” page



The wireframe shows a search form with a "search products..." input field and a "Search" button. Below it is another search input field with placeholder "Search something!". The page has a header with social media icons and a navigation bar with links like Home, About, Shop, Contact, Orders, and a search bar.

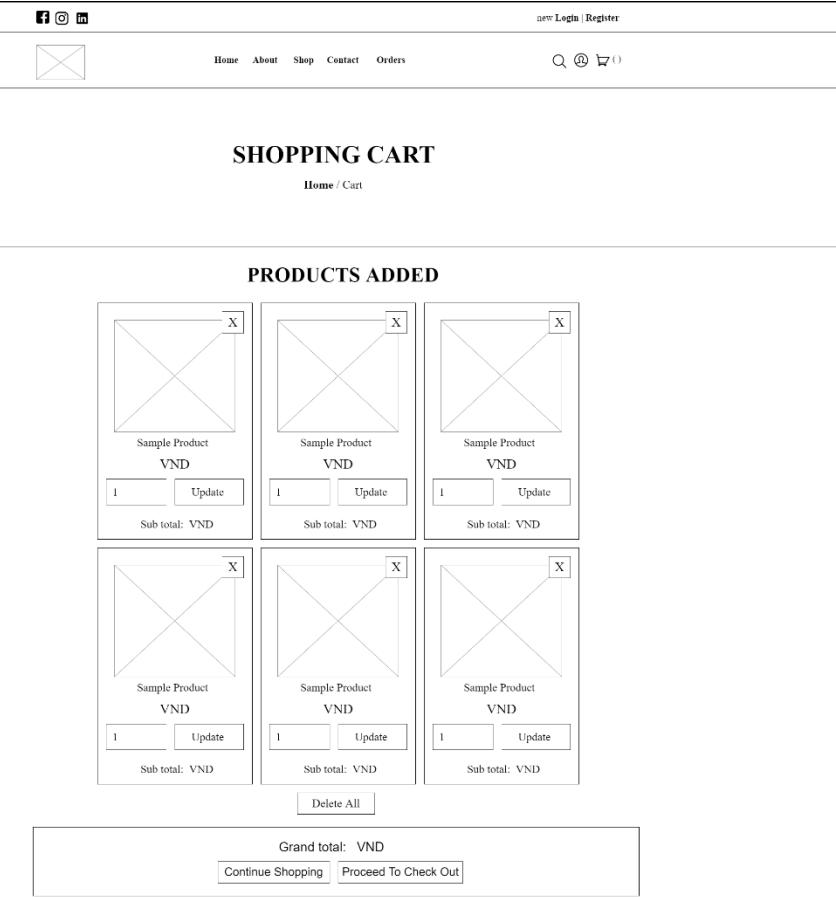


The wireframe shows a search form with a "search products..." input field and a "Search" button. Below it is another search input field with placeholder "Search something!". The page has a header with social media icons, a navigation bar with links like Home, About, Shop, Contact, Orders, and a search bar showing "(6)".

The wireframe is supposed to look exactly like the web page.

Similar

“Cart” page



SHOPPING CART

Home / Cart

PRODUCTS ADDED

Sample Product VND 1 Update Sub total: VND

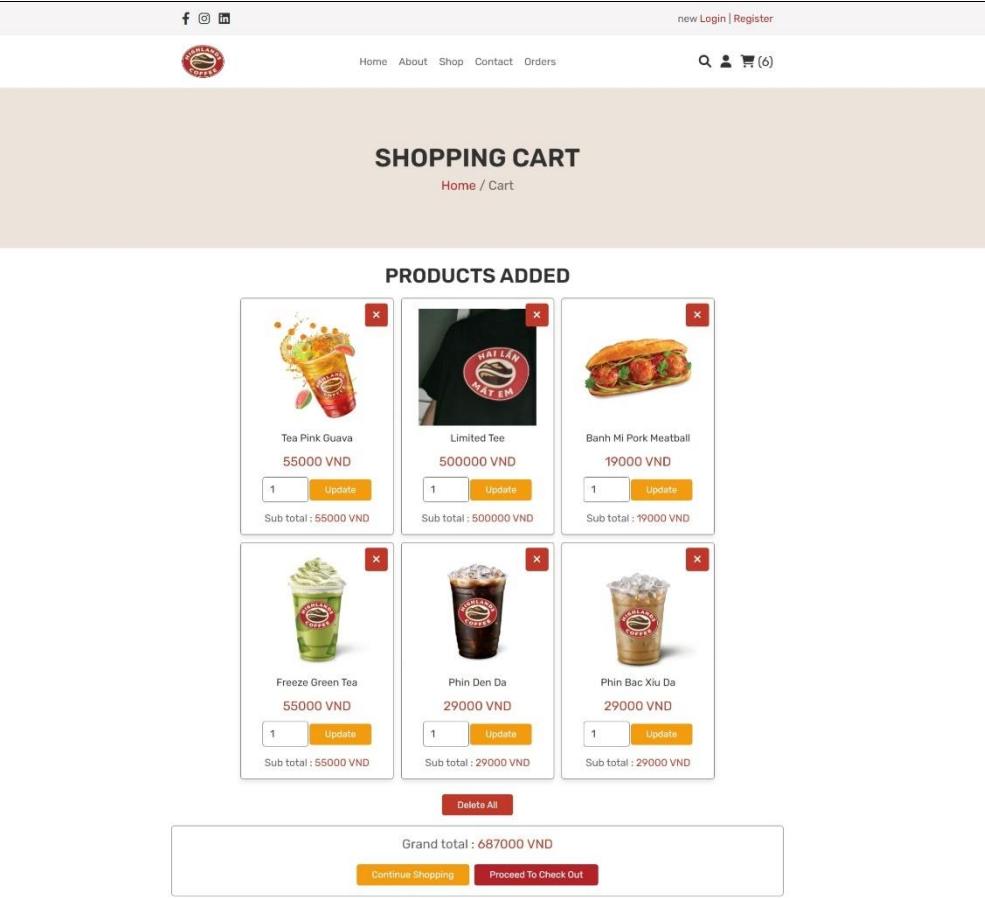
Delete All

Grand total: VND

Continue Shopping Proceed To Check Out

CONTACT US +84 912-765-398 Web Developer Find the nearest Highland Café

FOLLOW US Facebook Instagram LinkedIn



SHOPPING CART

Home / Cart

PRODUCTS ADDED

Tea Pink Guava 55000 VND 1 Update Sub total : 55000 VND

Limited Tee 500000 VND 1 Update Sub total : 500000 VND

Banh Mi Pork Meatball 19000 VND 1 Update Sub total : 19000 VND

Freeze Green Tea 55000 VND 1 Update Sub total : 55000 VND

Phin Den Da 29000 VND 1 Update Sub total : 29000 VND

Phin Bac Xiu Da 29000 VND 1 Update Sub total : 29000 VND

Delete All

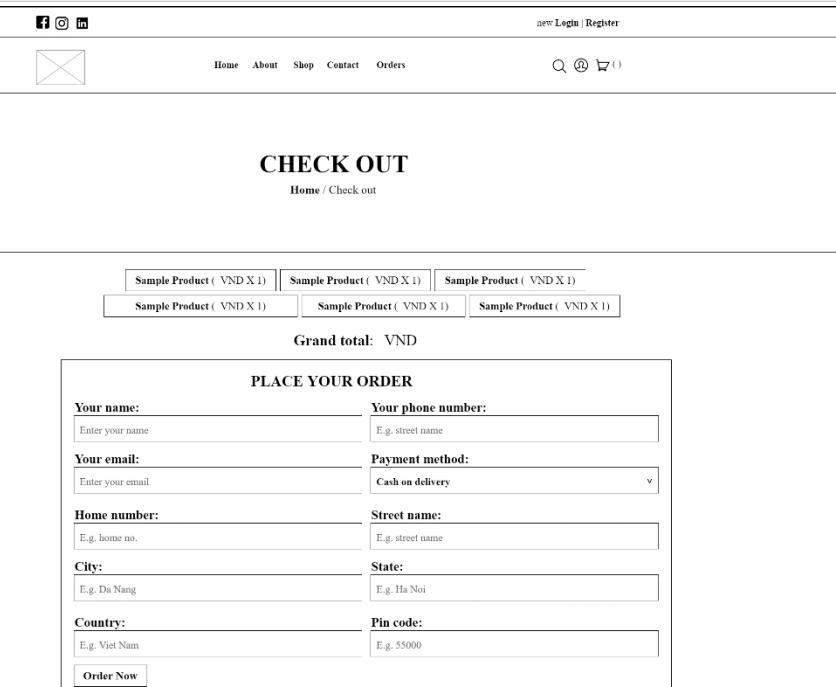
Grand total : 687000 VND

Continue Shopping Proceed To Check Out

CONTACT US +84 912-765-398 Web Developer Find the nearest Highland Café

FOLLOW US Facebook Instagram LinkedIn

“Check Out” page



CHECK OUT

Home / Check out

Sample Product (VND X 1) Sample Product (VND X 1) Sample Product (VND X 1)
 Sample Product (VND X 1) Sample Product (VND X 1) Sample Product (VND X 1)

Grand total: VND

PLACE YOUR ORDER

Your name: Enter your name Your phone number: E.g. street name

Your email: Enter your email Payment method: Cash on delivery

Home number: E.g. home no. Street name: E.g. street name

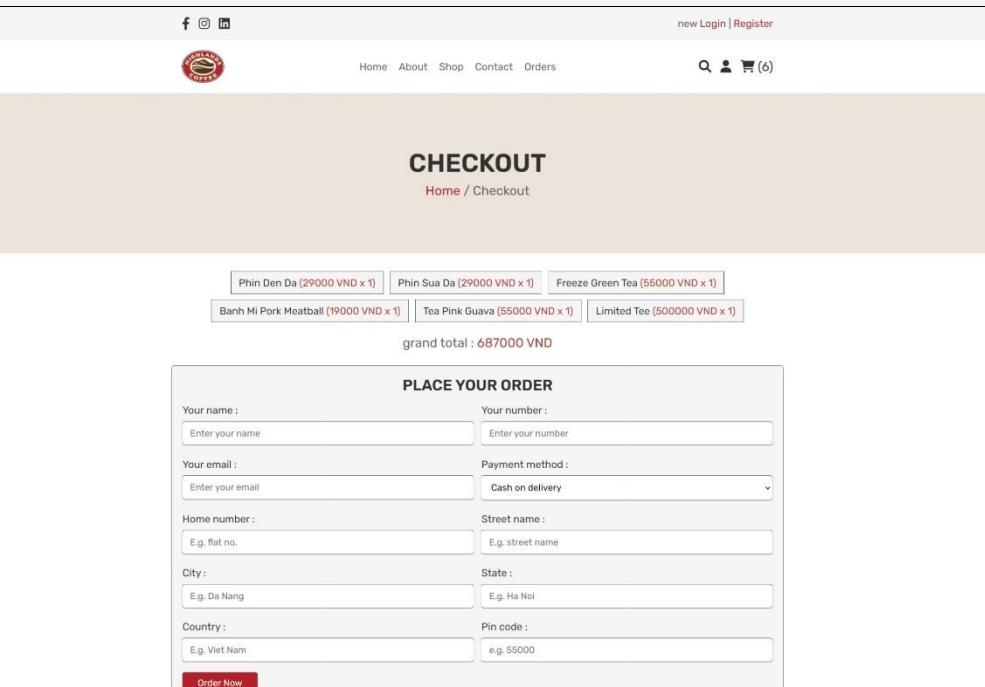
City: E.g. Da Nang State: E.g. Ha Noi

Country: E.g. Viet Nam Pin code: E.g. 55000

Order Now

CONTACT US +84 912-765-398 Web Developer Find the nearest Highland Café

FOLLOW US Facebook Instagram LinkedIn



CHECKOUT

Home / Checkout

Phin Den Da (29000 VND x 1) Phin Sua Da (29000 VND x 1) Freeze Green Tea (55000 VND x 1)
 Banh Mi Pork Meatball (19000 VND x 1) Tea Pink Guava (55000 VND x 1) Limited Tee (500000 VND x 1)

grand total : 687000 VND

PLACE YOUR ORDER

Your name : Enter your name Your number : Enter your number

Your email : Enter your email Payment method : Cash on delivery

Home number : E.g. flat no. Street name : E.g. street name

City : E.g. Da Nang State : E.g. Ha Noi

Country : E.g. Viet Nam Pin code : e.g. 55000

Order Now

CONTACT US +84 912-765-398 Web Developer Find the nearest Highland Café

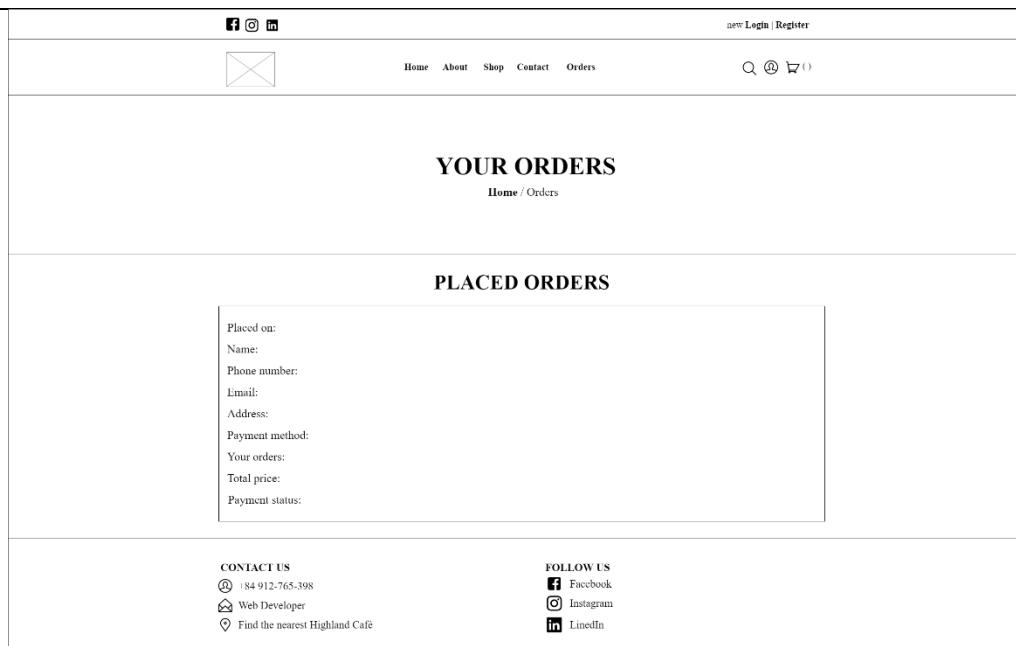
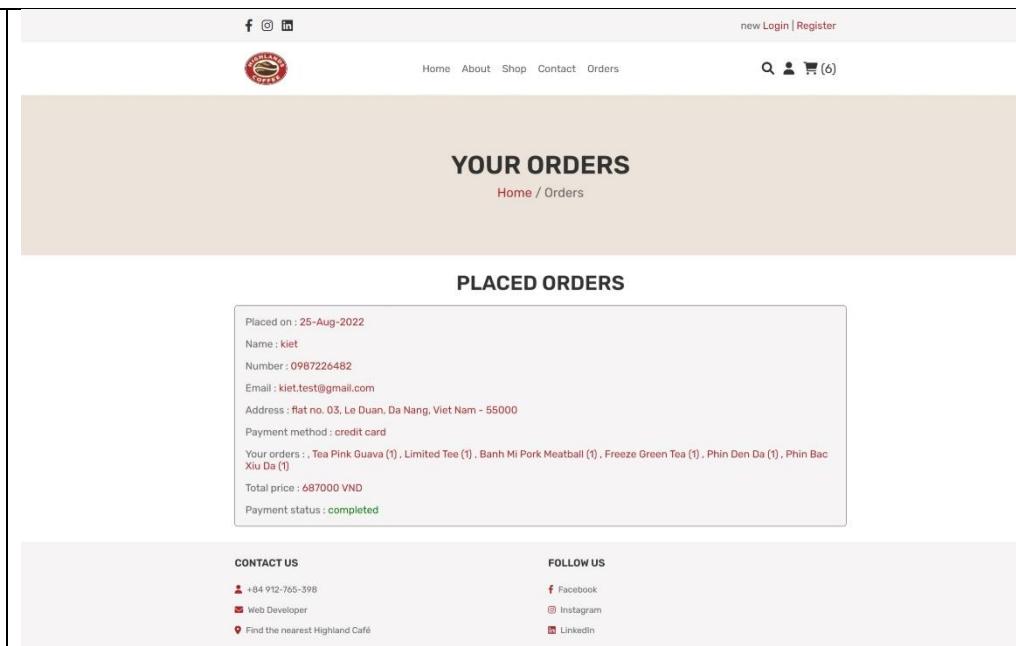
FOLLOW US Facebook Instagram LinkedIn

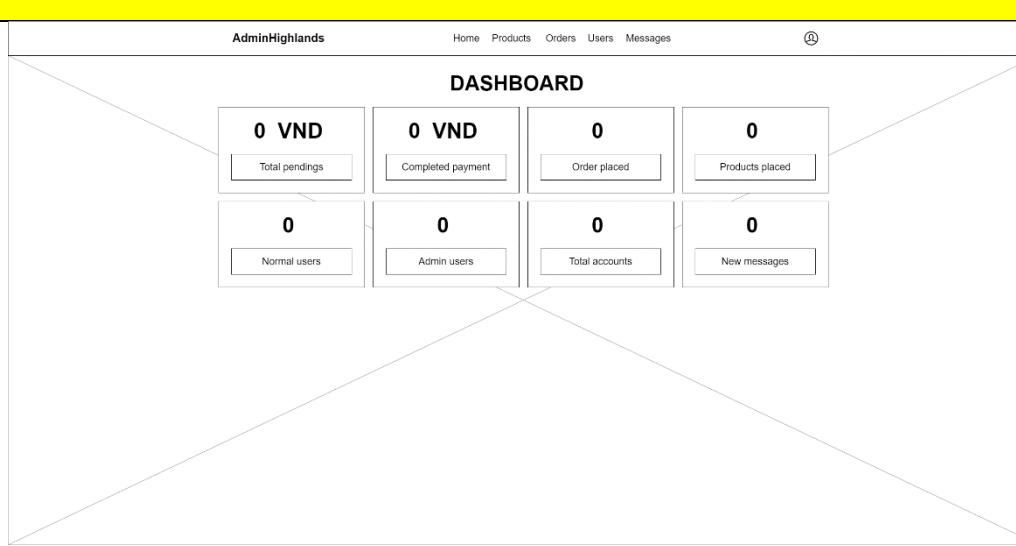
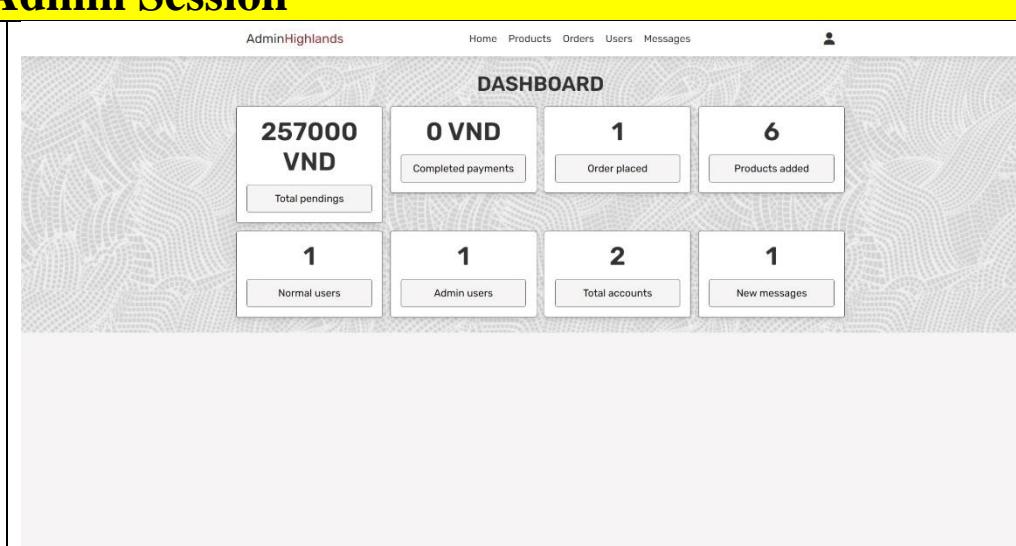
The wireframe is supposed to look exactly like the web page.

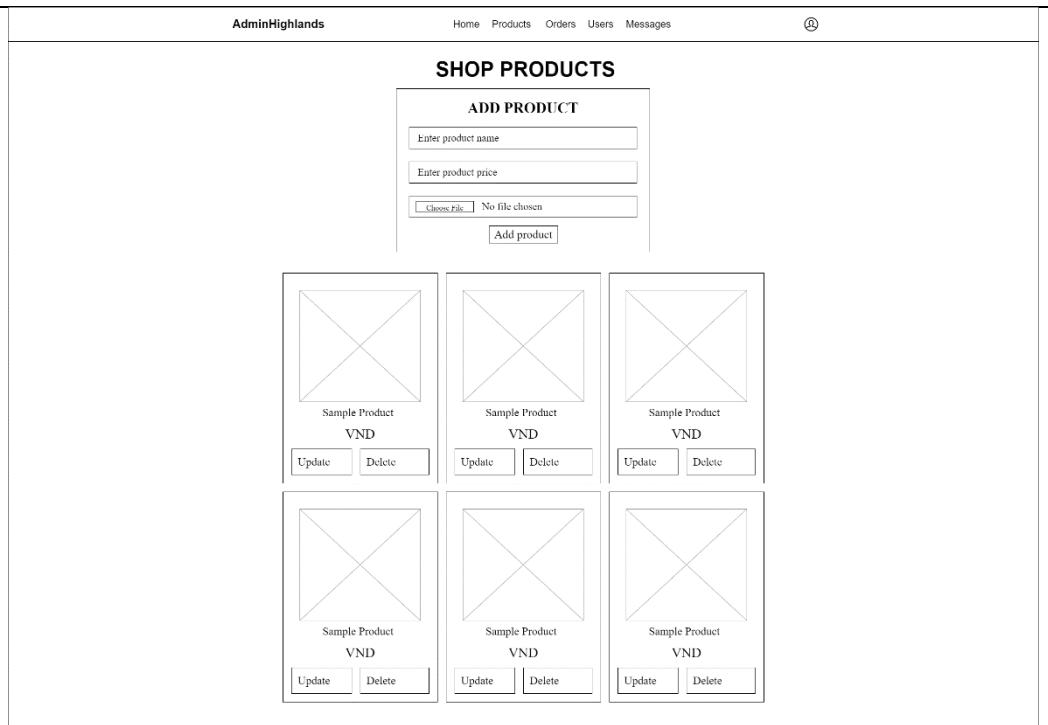
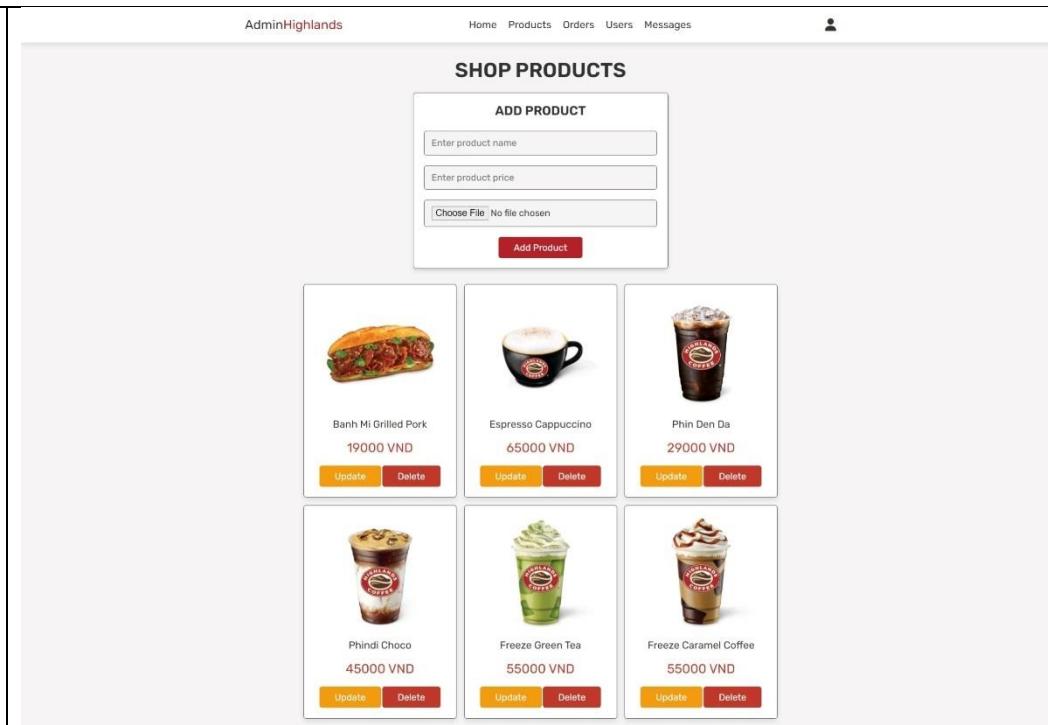
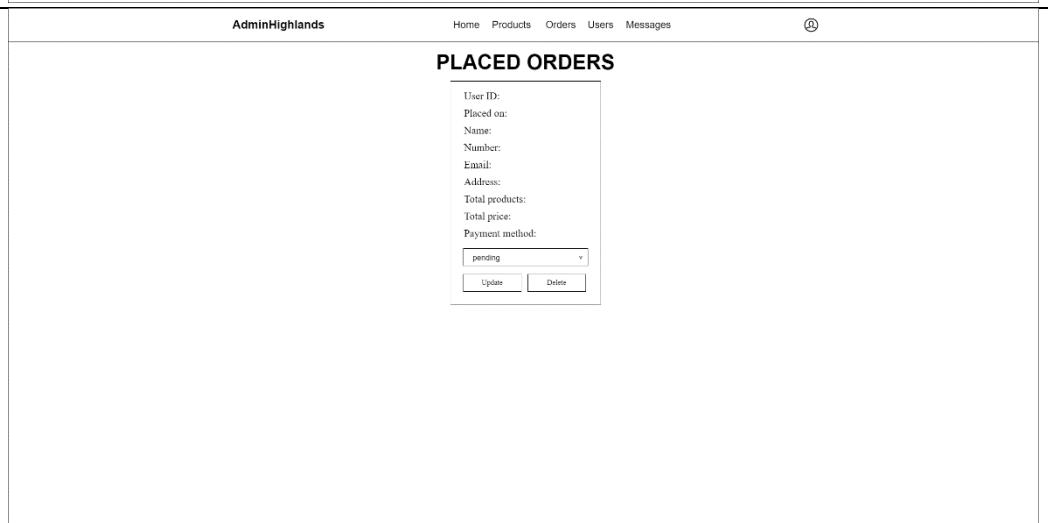
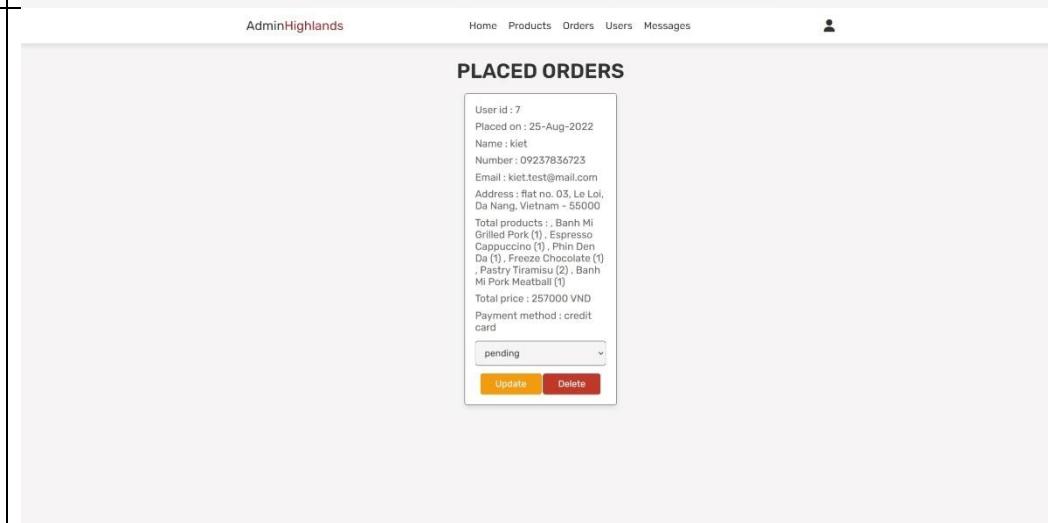
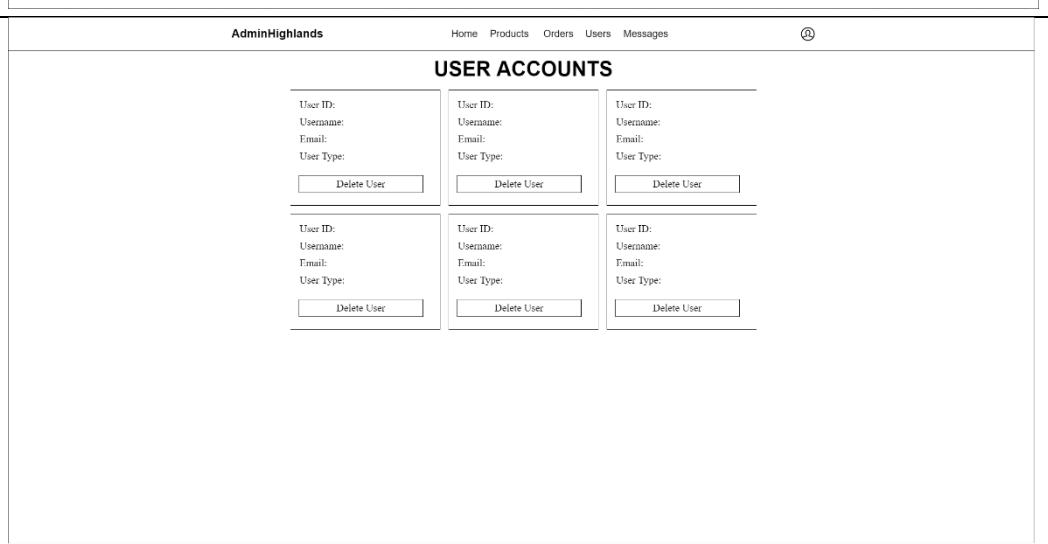
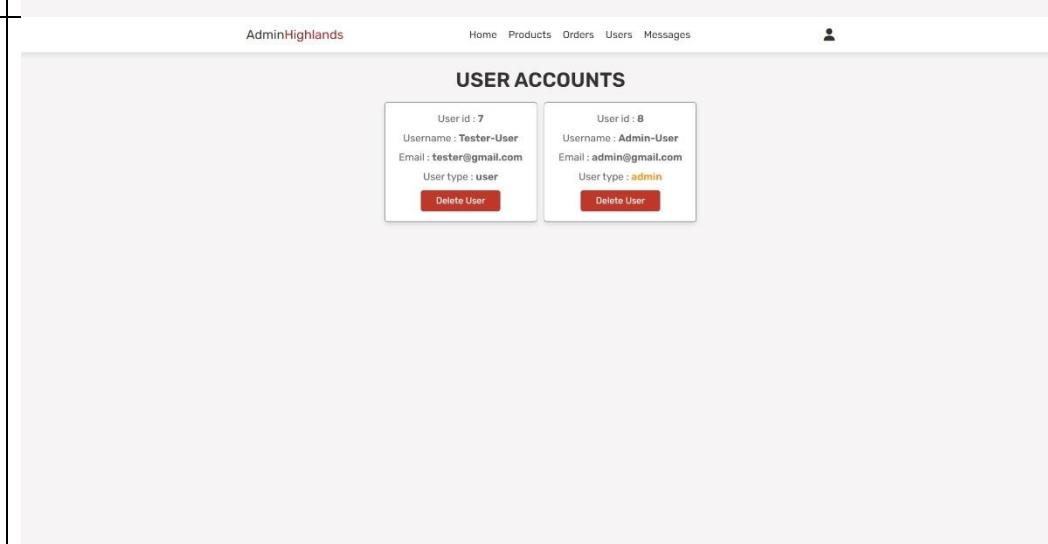
Similar

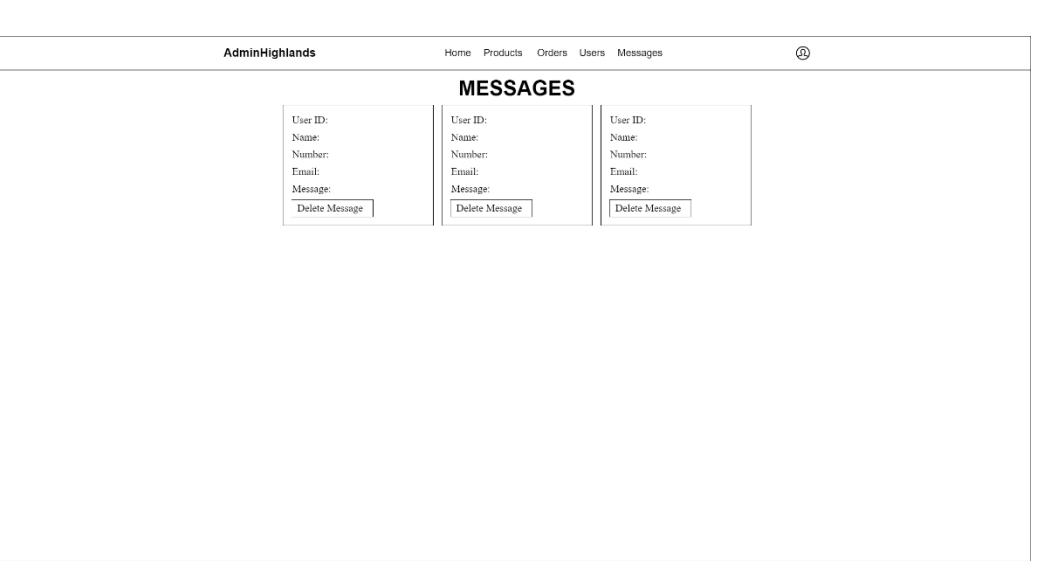
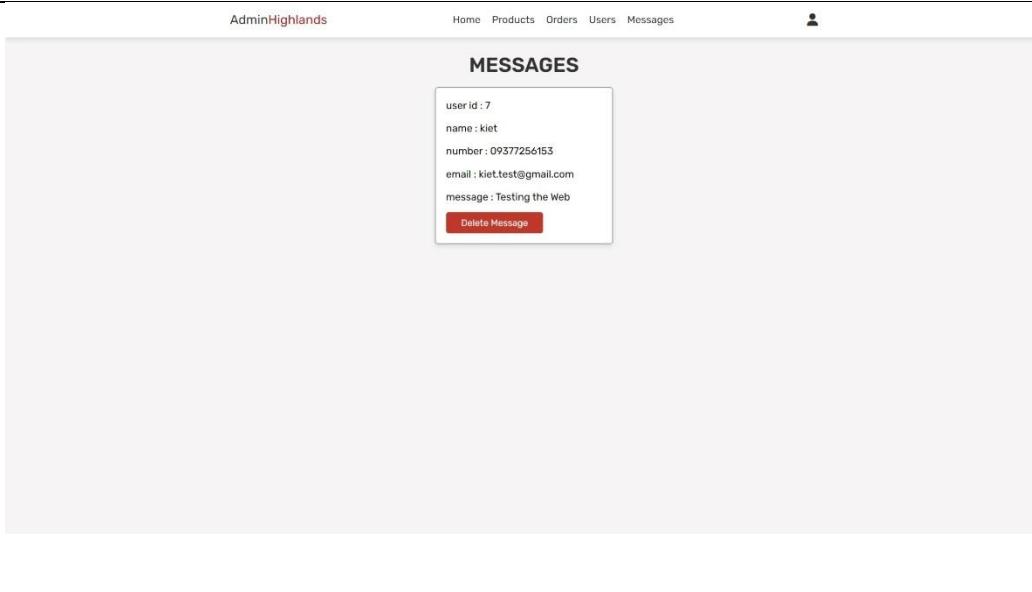
The wireframe is supposed to look exactly like the web page.

Similar

<p>“Order” page</p> 		<p>The wireframe is supposed to look exactly like the web page.</p>
---	---	---

<p>Admin Session</p>		
<p>“Home” page</p> 		<p>The wireframe is supposed to look exactly like the web page.</p>

<p>“Products” page</p>			<p>The wireframe is supposed to look exactly like the web page.</p>
<p>“Order” page</p>			<p>The wireframe is supposed to look exactly like the web page.</p>
<p>“User” page</p>			<p>The wireframe is supposed to look exactly like the web page.</p>

“Message” page 		<p>The wireframe is supposed to look exactly like the web page.</p>	Similar
--	---	---	----------------

In the end, compared to the initial wireframe design, I realized that my website developed most of the basic foundations designed in the wireframe. Besides, my website has some advanced features which are not mentioned in CRUD like:

- Dynamic effect when hover through elements like “Discover More” button on dashboard, “Contact Us” buttons and navbar items and social network link.
- A lot of function page such as "Contact", "About", ...

Chapter 5: Evaluate the Quality Assurance (QA) process and review how it was implemented during your design and development stages. (M5)

1. Identify

Any systematic method of verifying whether a product or service fulfills defined standards are referred to as quality assurance (QA).

Quality assurance is the process of establishing and maintaining standards for producing or manufacturing dependable goods. A quality assurance system is intended to boost consumer confidence and a firm's legitimacy while also enhancing work procedures and efficiency, allowing a company to compete more effectively.

Two principles govern quality assurance:

- Fit for purpose: The product should be appropriate for its intended use.
- Get it right the first time: Errors should be avoided.

Process and phases of quality assurance:

- Analyze the requirements: First, QA engineers examine the website's functional and non-functional needs. They explain to them when required and ensure that they are clear and consistent. This allows the team to produce the required outcome while avoiding future fixes.
- Test planning: QA engineers move on to the planning step based on the information acquired during the analysis. A thorough plan should contain the strategy, scope, budget, timelines, resources, testing kinds, needed software, bug reporting methods, and other details.
- Test design: Quality assurance engineers create test cases that cover all aspects of the project. The examples describe the conditions and processes required to determine whether or not a given feature works effectively. They generate automation scenarios if the project contains automation. They also prepare the testing environment.
- Test execution and bug reporting: At this step, QA engineers run automated scripts or execute defined cases. They record the flaws in bug-tracking software or make automated reports.
- Regression testing: When these defects are repaired or additional relevant functionality is introduced, QA engineers undertake regression testing to ensure that everything works correctly and that the new changes have not changed anything. Later, it should be repeated whenever new features are introduced.

Things to Look for:

- Accuracy to Original Design: Compare the original design that people approved to the final result to confirm that the coding version is accurate to the website design they adore. Keep in mind that while certain changes may have been communicated along the process, they should be aware of them and there should be no shocks.
- Images: Check the photographs on your site to ensure they are picture flawless. Check for any remaining watermarks and make sure your photographs appear right and have visually nice space around them.
- Content: Check that folks have all of the pages they need for their debut and that they have good grammar, spelling, and so on.
- Menus and Links: Click on all of the menu items and links to ensure that they all go where they should and that no pages have missing information. Links may also include PDFs and other papers available for download from the developer's website, as well as email addresses for contact reasons. People may verify if links open in the same window as your site or in a new window as requested.
- Forms for email: Fill out your own email forms to ensure that all of the fields are proper and functional. Then, check your email for the results to confirm that they received the submission that the form was designed to deliver.
- Browser Check: Run your website across many browsers. Older or less popular browsers may have compatibility concerns depending on the amount of design and programming they buy.
- Privacy Policy: Many web developers will utilize a standardized version on your site, but make certain it is present. Check to discover whether they require more explicit phrasing if they are gathering sensitive information.
- Testing: If they have any e-commerce or bespoke system components, make sure they are tested! Try processing payment and ensuring that it reaches their bank account as specified by their third-party payment gateway. Keep in mind that if they make a charge, they will have to reimburse their own money afterward.

2. How to do quality assurance: Complete process (Clarify which steps had applied in web development with evidence)

The PDCA cycle, often known as the Deming cycle, is a specified cycle in quality assurance methods. The Plan-Do-Check-Act cycle is a four-step process for implementing change.

These processes are carried out to ensure that the organization's procedures are regularly examined and improved. Let's take a deeper look at the above-mentioned Quality Assurance Process phases:

- Plan: Highland Coffee's website should plan and define process-related goals, as well as explain the procedures required to provide a high-quality final product. We utilized the Use Case Diagram to outline the development of process-related goals in particular.

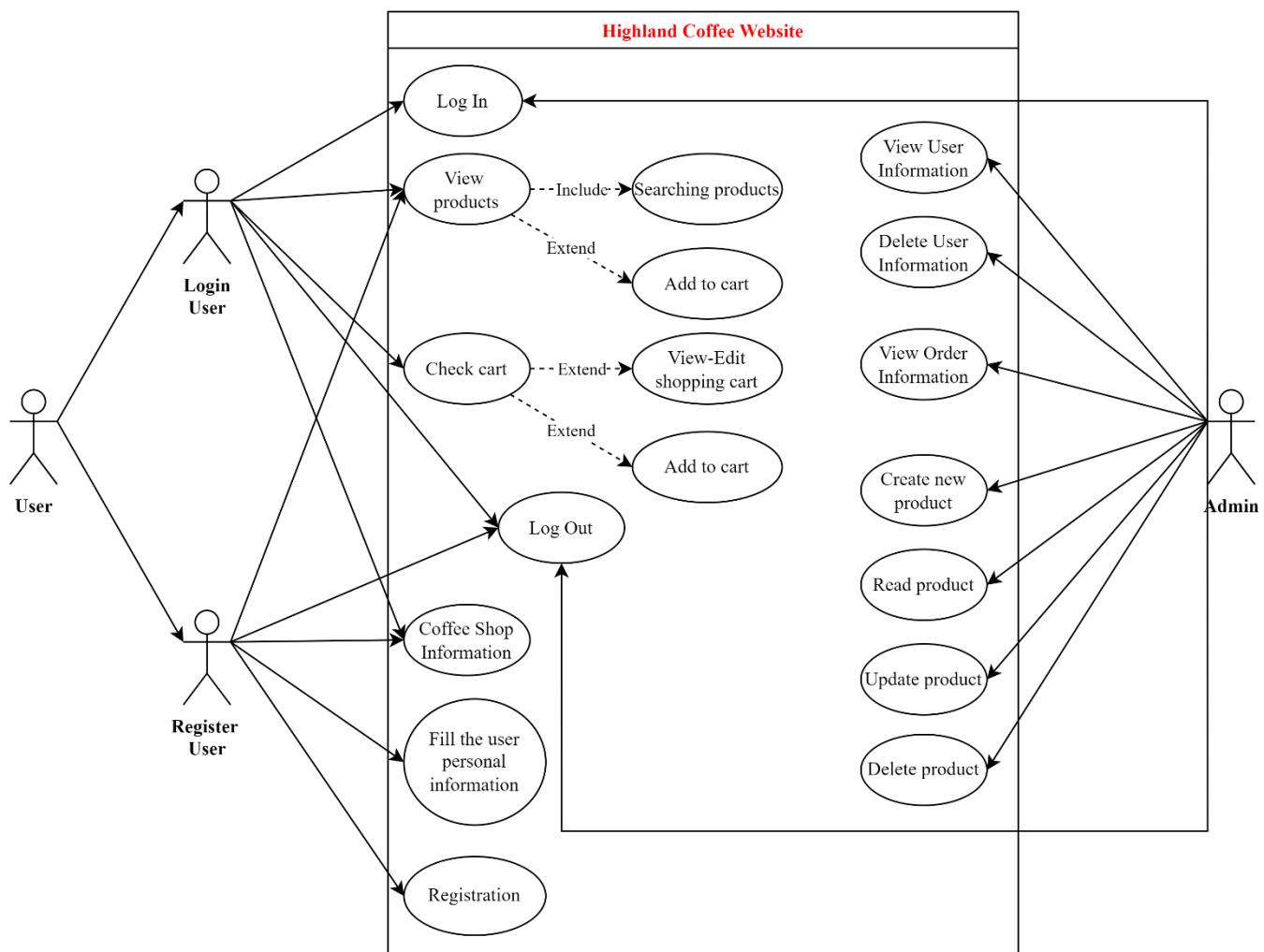
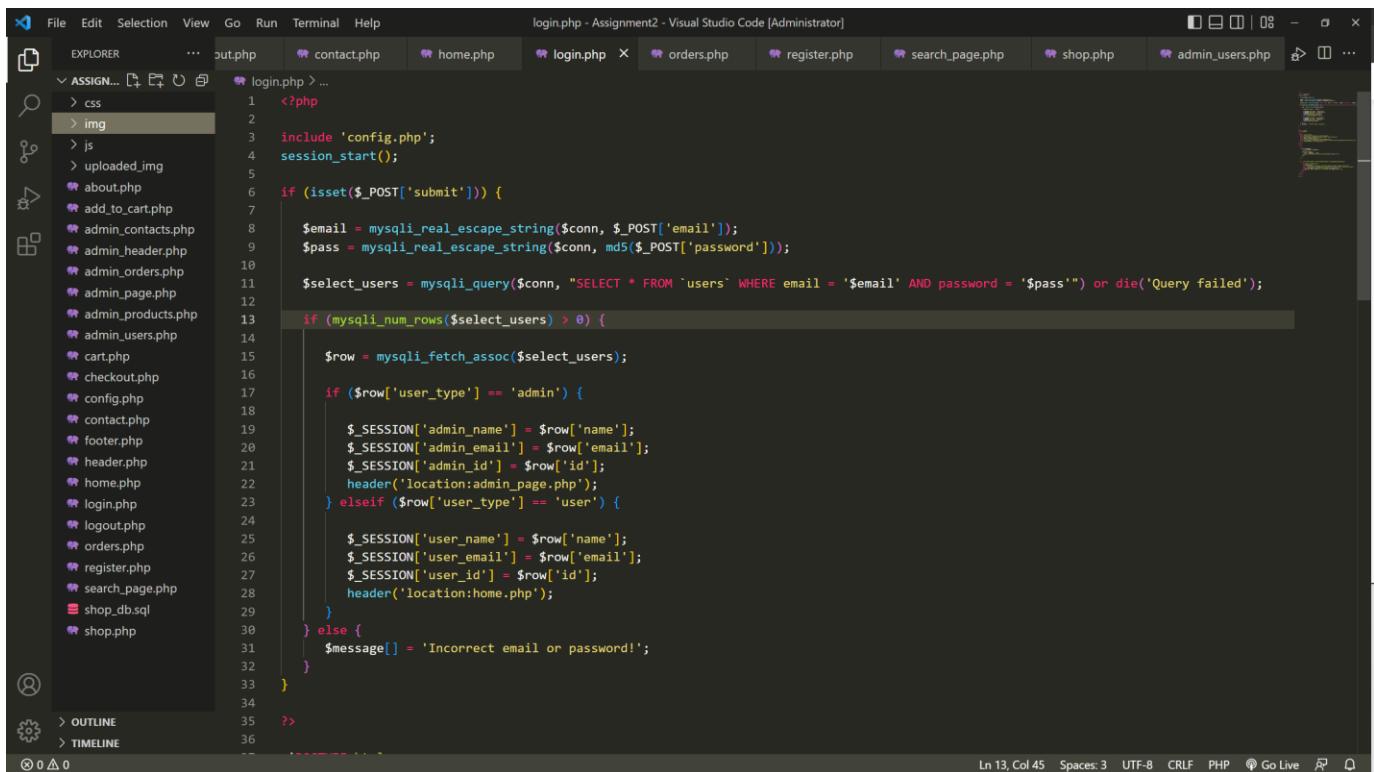


Figure 97: C5.2_Plan_Use Case Diagram

- Do: Process development and testing, as well as "performing" process adjustments. To progress the goals linked to the Highland Coffee website development process, I utilize the Visual Studio Code IDE for coding and debugging.



```

File Edit Selection View Go Run Terminal Help
login.php - Assignment2 - Visual Studio Code [Administrator]
login.php > ...
1 <?php
2
3 include 'config.php';
session_start();
4
5 if (isset($_POST['submit'])) {
6
7     $email = mysqli_real_escape_string($conn, $_POST['email']);
8     $pass = mysqli_real_escape_string($conn, md5($_POST['password']));
9
10    $select_users = mysqli_query($conn, "SELECT * FROM `users` WHERE email = '$email' AND password = '$pass'");
11
12    if (mysqli_num_rows($select_users) > 0) {
13
14        $row = mysqli_fetch_assoc($select_users);
15
16        if ($row['user_type'] == 'admin') {
17
18            $_SESSION['admin_name'] = $row['name'];
19            $_SESSION['admin_email'] = $row['email'];
20            $_SESSION['admin_id'] = $row['id'];
21            header('location:admin_page.php');
22        } elseif ($row['user_type'] == 'user') {
23
24            $_SESSION['user_name'] = $row['name'];
25            $_SESSION['user_email'] = $row['email'];
26            $_SESSION['user_id'] = $row['id'];
27            header('location:home.php');
28        }
29    } else {
30
31        $message[] = 'Incorrect email or password!';
32    }
33}
34
35 ?>

```

Ln 13, Col 45 Spaces: 3 UTF-8 CRLF PHP Go Live

Figure 98: C5.2_Do_Visual Studio Code

- Check:** Processes are monitored, updated, and tested to ensure that they are meeting the intended goals. More particularly, the Test Case stated in section P7 above.
- Act:** A Quality Assurance tester should implement the required changes to enhance the procedures.

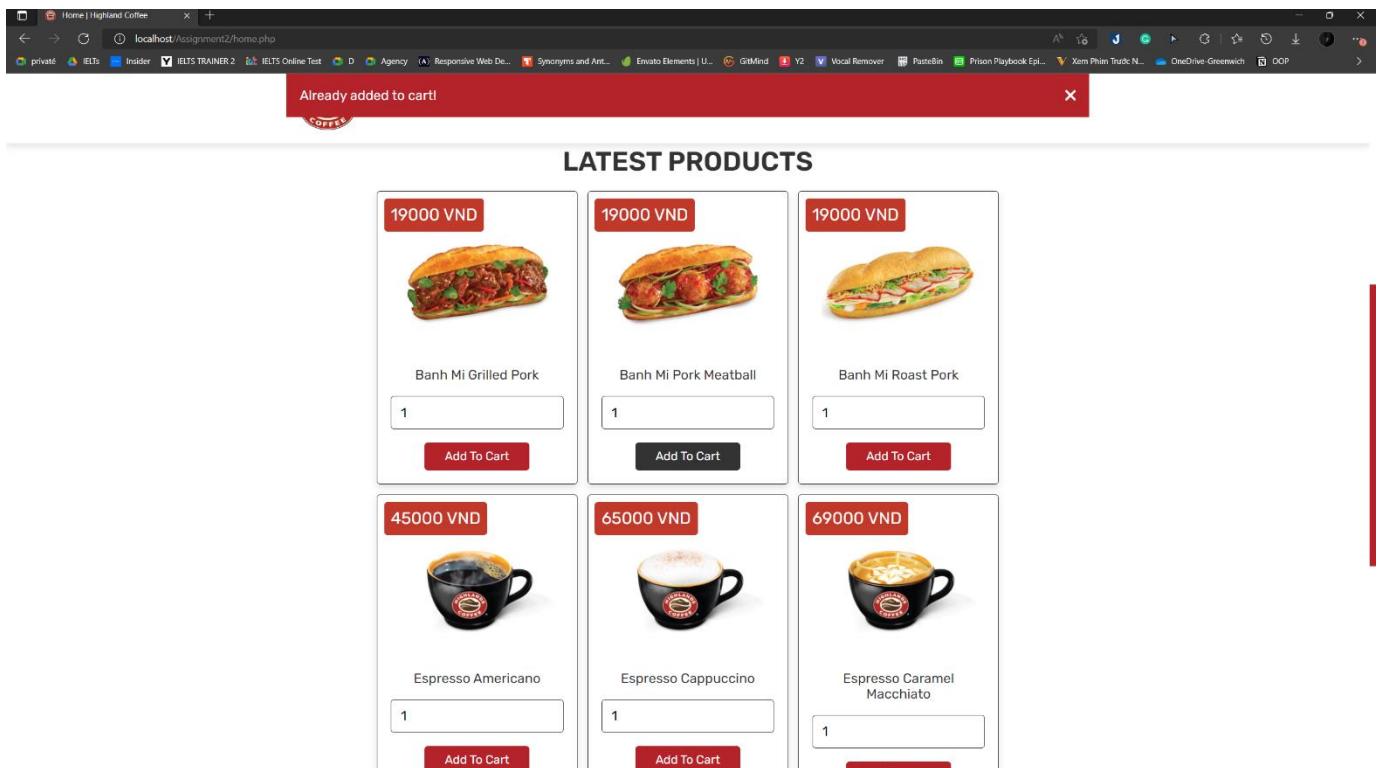


Figure 99: C5.2_Act_My Highland Coffee Website

Chapter 6: Critically evaluate the design and development process against your design document and analyse any technical challenges. (D2)

1. Website evaluation

The following criteria were met:

- The ability to search for a product is an important feature of this website since it helps visitors to get the information they need more quickly. This will improve their overall experience and increase the probability that they will return to the site.
- The user interface (front-end and back-end) was successfully constructed with no display failures, errors, or missing requirements. The user interface also adapts to any screen resolution on any device.
- A back-end database with entity relationships has been created, meeting the 3NF normalization procedure, which eliminates data duplication and keeps the data interesting.
- Basic query capabilities that connect the front-end to the back-end have been established to improve website security and speed, and the Administrator may simply handle the complete data system via the back-end changed pages.

There are still many things that need to be created in order for the websites to become completely functional and effective e-commerce websites that meet all user criteria. This includes the payment feature, which is not available in this project.

However, current Highland Coffee website versions are not fully built to suit all user requirements. While the front-end was reasonably completed, the requisite features that are required to make the website actually operate are not being developed due to a few technological issues. As a result, a longer time period will be required to launch the website in order for it to become a fully effective e-commerce website that meets all user criteria.

2. The technical challenges

Add to cart function:

- The basic idea behind this pattern is to notify the user that an item has been added to the cart and provide them with a link to proceed to the checkout. Although in the Highland Coffee project we have implemented shopping cart functionality, there are still some weak points.

- However, the way to apply this API to the website's source code is still remains a challenge, where the Cart UI needs to be customized to fit the Highland Coffee concept design and payment method.
- At the current version of the website, the Cart function is basically implemented and it still
- needs more updates and development to fully achieve the user requirement, such as adding products, deleting products, and editing product quantities.

Payment function:

- The basic idea behind this model is to help users have many payment options such as cash payments, and credit card payments. In the Highland Coffee project, we have implemented the payment function for users but there are still some weaknesses such as not being able to pay with their credit card completely.
- We are working on payment methods and preparing to roll them out to users in upcoming updates.

Chapter 7: Critically evaluate the results of your Test Plan and include a review of the overall success of your multipage website; use this evaluation to explain any areas of success and provide justified recommendations for areas that require improvement (D3)

1. Test result evaluation

I make the following assessment based on the testing and the case study described above.

Front-end:

- In terms of the front-end, the findings show that I need to start utilizing Wireframe to design ahead of time for the web project, or else I'll have difficulties when the Web code is finished.
- Although the last Case Study focused on the usage of the Flexible Box Module for layout, it does not cover every circumstance that may occur on your website. As a result, I had to use ID classes to override and rectify them. When naming components in CSS and JS, BEM guidelines must be observed.
- Aesthetic considerations, attention to detail, working in line with the original design, the right approach, and SEO content requirements for Web pages are all required for the production of a static web page.

Back-end:

- To execute PHP and SQL on the web server, I use the open-source XAMPP.
- At first, I struggled to adapt to the new programming language PHP. Although linking the database and the website is simple, carrying out the tasks requires thorough and correct study.
- Working with XAMPP has a slight disadvantage in terms of database. If you aren't comfortable changing the database with POSTs, Query is a decent solution. Working with XAMPP requires you to write functions that correlate to the models (Tables) you've defined.

2. Providing justified recommendations of improvement

After the websites evaluation process that also describes the technical challenges that are now yet being accomplished and the improvement recommendations should be:

- Continue the necessary developing functions to make it fully functional: Cart, Payment, etc.

- Continue to develop more Test cases and test them as well as doing alterations to the source code in order to deliver the user fully functional, good design with no bugs or errors e-commerce website.
- Speed up page loading is essential should be improved. Precisely because I am using 000webhost, this will take quite a while to load my website. I am planning to choose a server provider to improve the user experience.
- Compatibility with other types of smart devices such as phones and tablets are a necessity to provide a better user experience. Currently, I only use a little 'responsive' knowledge to develop my website. I'm trying to increase site and mobile compatibility by applying 'responsive' knowledge to my website.
- Finally, after accomplished every requirement component function that declared in the user requirement table and the website became fully developed, the hosting solution as well as domain name solution are needed. The suggested domain name is: www.the-highlandcoffee.vn that can be bought for \$37 per year.

3. Advantage and disadvantage of web project

Table 4: C.3_Web project advantage and disadvantage table

Advantages	Disadvantages
<p>Eye-catching interface, clear images, clear presentation attributes with fully structured of a website like create a new product, read an order detail, update a product information, delete the user.</p> <p>User can add or delete products to cart.</p> <p>User can login or register to the web.</p> <p>The website assigns 2 separate permissions, for the user session as well as the admin session.</p>	<p>The front-end of my website is not compatible with many smart devices such as tablets, phones.</p> <p>The database has a lot of flaws.</p> <p>The site isn't fully compliant with SEO guidelines.</p>

References

ASQ, n.d. *WHAT IS THE PLAN-DO-CHECK-ACT (PDCA) CYCLE?*. [Online]

Available at: <https://asq.org/quality-resources/pdca-cycle>

[Accessed September 2022].

Crestodina, A., 2015. *Web Design Standards vs. Website Best Practices: Our Review of 500 Sites [NEW RESEARCH]*. [Online]

Available at: <https://www.orbitmedia.com/blog/web-design-standards/>

[Accessed September 2022].

Gillis, A., n.d. *What is Quality Assurance?*. [Online]

Available at: <https://www.techtarget.com/searchsoftwarequality/definition/quality-assurance>

[Accessed September 2022].

Hamilton, T., 2022. *What is Quality Assurance(QA)? Process, Methods, Examples.* [Online]

Available at: <https://www.guru99.com/all-about-quality-assurance.html>

[Accessed September 2022].

Kanbanize, n.d. *What is Plan-Do-Check-Act (PDCA) Cycle?*. [Online]

Available at: <https://kanbanize.com/lean-management/improvement/what-is-pdca-cycle>

[Accessed September 2022].

Marianne, 2021. *9 PRINCIPLES OF GOOD WEB DESIGN.* [Online]

Available at: <https://www.feelingpeaky.com/9-principles-of-good-web-design/>

[Accessed September 2022].

WishDesk, 2020. *Quality assurance for your web project: stages, methods, & practices.* [Online]

Available at: <https://wishdesk.com/blog/quality-assurance-stages-steps-tools>

[Accessed September 2022].

Appendix

Skill Assessment for HTML

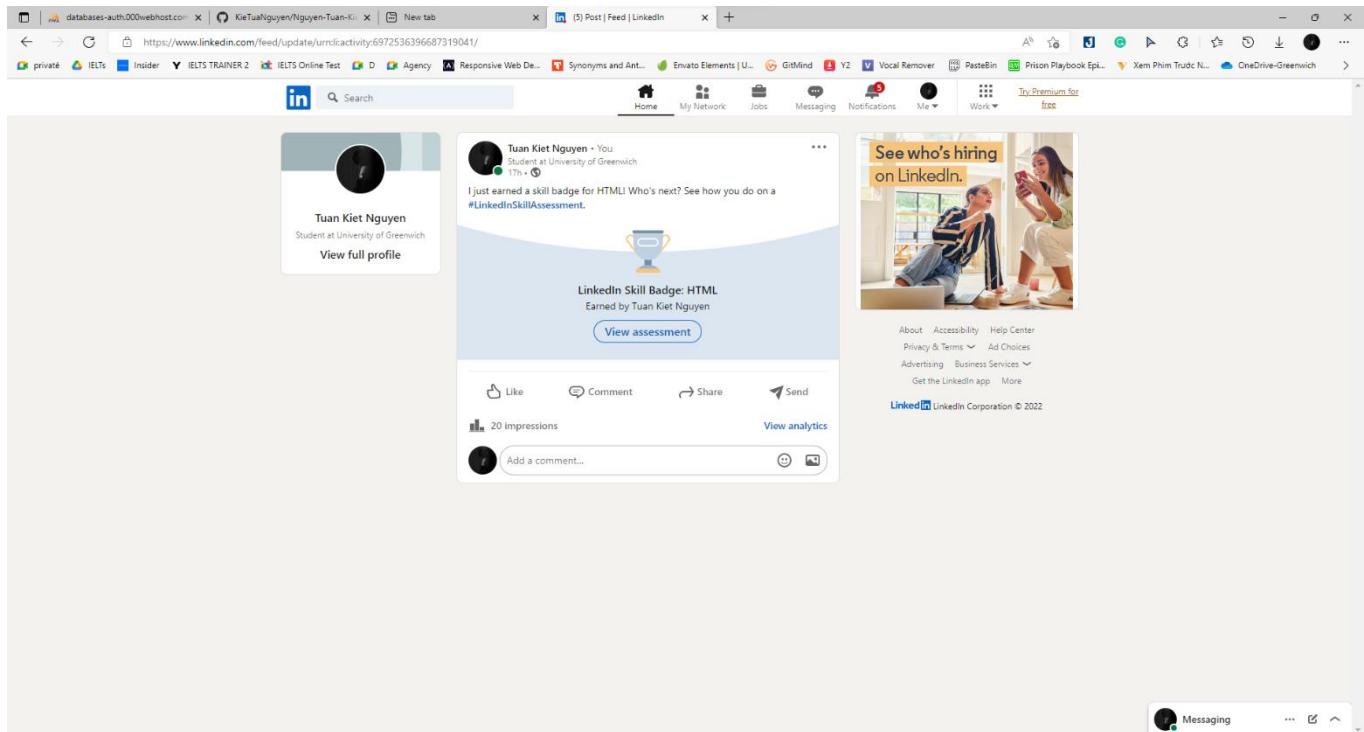


Figure 100: Appendix _Skill Assessment_ HTML

Skill Assessment for CSS

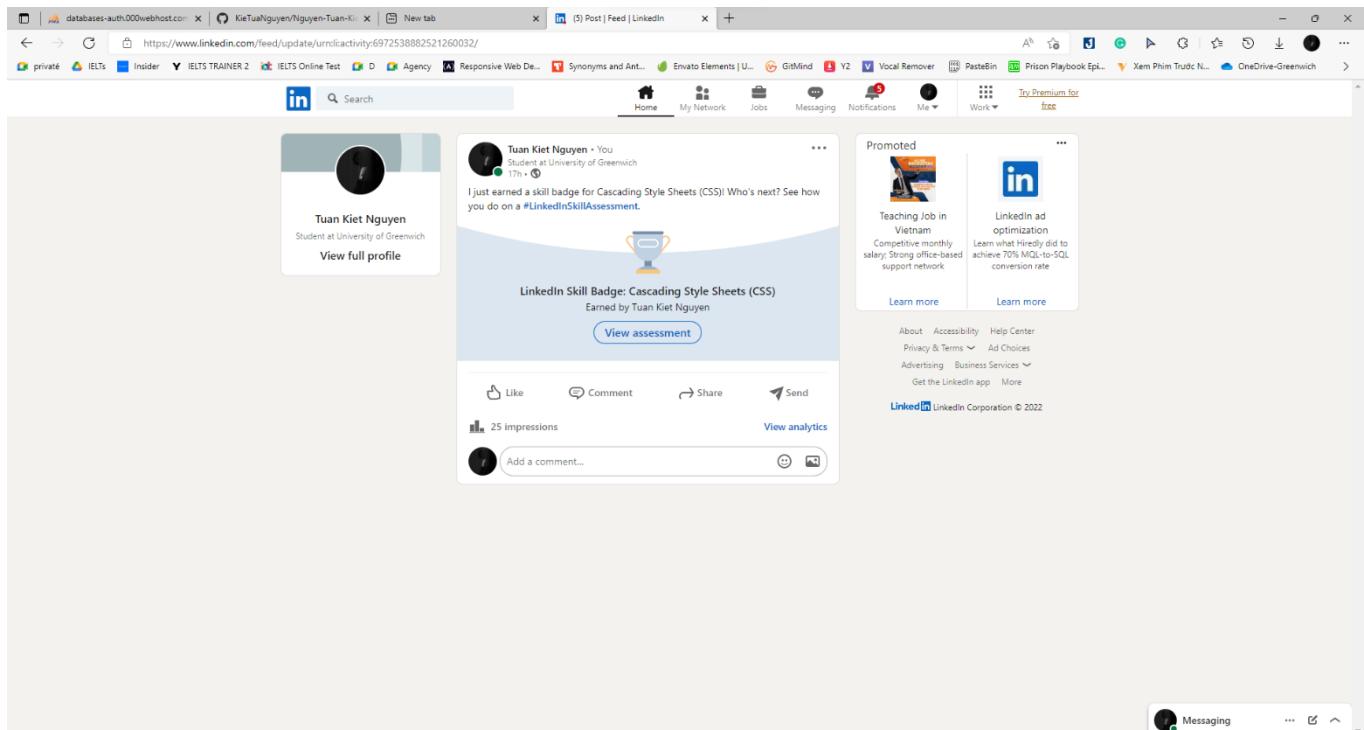


Figure 101: Appendix _Skill Assessment_ CSS

The Website on 000WebHost

My website is now available on 000WebHost:

<https://busying-certificate.000webhostapp.com/Assignment2/home.php>