



RMIT International University Vietnam

Assignment Cover Page

Subject Code:	COSC2430
Subject Name:	Web Programming
Location & Campus (SGS or HN) where you study:	SGS
Title of Assignment:	Full-stack Web Application
Teachers Name:	Tom Huynh
Group Name:	Group 8
Group Members (names and id numbers):	1. s3974735 – Tran Manh Cuong 2. s3965528 – Truong Quang Bao Loc 3. s3978467 – Nguyen Van Khai 4. s3979259 – Truong Tuong Hao 5. s3978742 – Huynh Nguyen Thien Minh
Assignment due date:	20 May 2023
Date of Submission:	20 May 2023
Declaration of Authorship	We declare that in submitting all work for this assessment, we have read, understood and agreed to the content and expectations of Assessment Declaration as specified in https://www.rmit.edu.vn/students/my-studies/assessment-and-exams/assessment
Consent to Use:	We give RMIT University permission to use our work as an exemplar and for showcase/exhibition display.

Table of Contents

I. Introduction	1
II. Project Description.....	1
1. Register Page	1
2. Login Page	2
3. Customer Page	2
4. Vendor Page.....	3
5. Shipper Page	4
6. Account Detail Page	4
7. Static Page.....	5
III. Implementation Details	5
IV. Conclusion	11
V. Peer Evaluation.....	12
VI. Reference	12
VII. Appendix.....	13



I. Introduction

In this project, we have developed a full-stack e-commerce website using HTML, CSS, JavaScript, NodeJS, ExpressJS, MongoDB, and Bootstrap. We create an e-commerce system similar to Lazada. Vendors can create their businesses, add products, and assign those products to the specific distribution hub. Customers can access the website to find and order the products they want. Then, shippers will receive the orders from their distribution hub and deliver the products to customers. Our website has 7 types of webpages: Register Page, Login Page, Customer Page, Vendor Page, Shipper Page, Account Detail Page, and Static page. For each page, we tried our best to create a user-friendly interface and achieve all the functional requirements.

II. Project Description

1. Register Page

There are many purposes that users have when visiting our website. Whether it's to buy something, sell products, or look for a job as a shipper, they must have an account firsthand to use the website. There are three different register pages with different purposes, one for daily customers, one for vendors who want to sell some products, and one for the job as a shipper. For the customer page, the information form includes a username, profile picture, full name of the user, address, email, and password field. On the page for vendors, there are additional fields to fill in, such as business name and business address. For shippers, they must choose a distribution hub to register successfully.

There are requirements in some fields that must be satisfied to register successfully. The password field must include at least an uppercase and lowercase letter, numbers, and special characters and must be at least 8 characters long. For the username field, it must include uppercase and lowercase letters and some numbers if users want. If users enter the wrong format on a field and try to click the register button, the website will display a message telling them to enter the correct format for it. Some fields such as username, business address, and business name are unique, which means users cannot choose an already existing name. A message will display if the database detects the information on the field already exists and will demand users to go back to the register



pages to change it. When they register successfully, the information on the registration form will be sent to the remote database to store and finish the account creation process. For the password, it will be hashed before storing for encryption and security purposes.

2. Login Page

In every e-commercial website, a login page is indispensable, so we designed this page quite strictly, we divided this page into a layout with 2 parts which were a left-side background image and the right-corner login form. To give the website a decent visual appearance, we decided to also implement JavaScript to make several transitions and animation for those lines, and buttons for the website and we did not forget to provide an ‘Aladin’ logo picture that was posited on the top of the login form. There are two interactive buttons which are the main ‘Login’ button and the ‘Register Now’ button. There is countless lesson that we acquired from this project, we learned to use CSS in an efficient way and more engaged to apply CSS in needed scenarios, we can use CSS to make the website responsive with mobile devices, as well as gained skill in HTML code, we can code faster and unlikely to find a hint on ChatGPT or browsing on the internet.

Furthermore, we studied JavaScript more than before, so it provided us with experience in using JavaScript to animate some elements. CSS and JavaScript code are two of the most vital specification on this page since they helped us to design the website based on the layout pattern of the template as well as decorate each element to be responsive and eye-catching since it had a hover effect and a transition line for the login form and animation effect for the ‘Login’ button.

3. Customer Page

Aladin website offers 5 pages for the customer namely the customer homepage, category page, product detail page, search page, and shopping cart and all of those pages work well together. On the customer homepage, there is a category section that will render the category page if the user clicks it, including Beauty Cosmetics, Electronics, Fashion, Health & Household, Home & Kitchen, and Sports & Outdoors. On the category page, there is a banner and product section including all the products belonging to that category. The website also recommends users some popular products in the hot deal section, and all the products from the vendor will be displayed in this



section. Another feature in the hot deal section is the filter, where users can enter the minimum price and maximum price, then the website will return all the products in that range for users. To view detailed information about a product, Users just need to click directly on the product. Aladin will send users to the product detail page and users can get the description of the product, price, and the product's name. To add the product to the cart users can click on the add to cart button then the product will be added to the item list in the cart and users can finish the order after filling in all the information. The intercommunity component is the navigation bar and the footer. There is a logo, navigation, search bar, shopping cart, and user's avatar on the navigation bar, and users can access the copyright or understand more about Aladin through the About Us section in the footer.

4. Vendor Page

According to the project's requirements, we need to develop an e-commerce website allowing vendors to add a new product and see all the products they have added. We have designed all features and functionalities of the vendor page properly so that users can utilize our website efficiently. On the vendor page, there will be one main page, Products, and an extra page which is the Orders page, two of which users' access through the tabs in the side menu.

When vendors get to the Products page, they will see a table containing a list of added products with basic information, such as Name, Price, Image, Description, and Amount. Moreover, this table has an Action column containing Delete buttons for users to remove products. Meanwhile, an Add-product button is placed right on top of the table, allowing users to access the Product Form. On this page, there will be a form for users to fill in all the essential information about products, then the information will be saved in the database and added to the table.

Additionally, we have designed an extra page, which is the Orders Page connected with the Orders database of the Customers Page. In addition to managing the products, vendors also need to receive orders from customers and work with distribution hubs to deliver to them. We arrange customers' orders in a table including ID, Products, Payment, Total, Receiver, and Address.



5. Shipper Page

To access the shipper page, users must log in to their account or register a new account with the correct format of username and password. After they log in successfully, they will navigate to the shipper home page.

For the shipper page interface, there are 4 sections: sidebar, header, footer, and main content. On the sidebar, users can view their profile picture, their name, and role, and they can also click on the Dashboard to view the active orders or log out of their account. On the header, there is our website logo on the left and some buttons on the right. Shippers can view their notifications and their account details by clicking on those buttons. On the footer, users can navigate to some static pages such as About Us, Privacy, Refund, Terms, and Support.

On the main content, shippers can view the orders from the distribution hub they selected in the registration. For example, a shipper from Tan Son Nhat Cargo Terminal can view all the active orders from this distribution hub. If users want to see the order details, they can click on the “detail” button, and a modal will pop up containing all the order information such as the order’s id, status, item list, total price, receiver, address, and payment method. Shippers can also change the order status from “Active” to “Delivered” or “Canceled” and after they click “Save changes”, the non-active order will not be displayed on the main page anymore and the status of that order in our database will also be changed.

6. Account Detail Page

The Account Detail Page contains several requirements based on the Assignment structure, so we spent a long time completing this part. This page is divided into two sections which are the User Profile Image section and Information Input Section. The avatar is quite important since it’s mentioned to be changeable, this figure allows users to click on the profile to switch profile avatar by choosing their favorite image in their local computer and there also must be a Logout button to lead users back to the Login page. It is necessary to improvise JavaScript to make the function picture change as well as CSS is also required to modify the design of some elements to give it a decent look. For example, the ‘Update’ and ‘Logout’ buttons must be in the contrast color, we used CSS to config those two elements.

7. Static Page

Our static pages are: About Us, Privacy Policies, Terms of Use, Refund Policies, and Support. All these pages were designed with the main orange color and using content from internet sources but also be modified to fit our scenario, we also made an ‘Up-top’ button to increase the browsing experience for users as well, We also implemented ‘Back to Home page’ button and an up top keypress using pure CSS and HTML code, we maintained to configured all these pages to look attractive as much as possible and responsive. Bootstrap was helpful at this point since we did not have much time left, so we make use of it efficiently for each element.

III. Implementation Details

This project was released on April 24 but by May 6 we could be able to start this project because we had to finish the project from other courses. At that time, none of us were familiar with ExpressJS or Back-end development so we decided to develop the Front-end first. We listed all the pages that we had to develop and decided to finish the Front-end within 4 days, the Back-end in 7 days, complete the testing process, record the demonstration video, and write the report in the last 3 days. In the first meeting, together we chose the website names, logo, primary colors, fonts, and icons to ensure consistency between the web pages. We chose “Aladin” as our website name because we wanted to give our customers everything they want, just like the magic lamp. Then, we designed our logo by Canva and chose our primary colors bright orange (#F68713) and white (#FFFFFF) to make it attractive. About fonts, we chose Poppins and Roboto for our main content and Dancing Script for the logo. Finally, we all agreed that we would use the icons from Font Awesome 5 and make our website responsive easily by using Bootstrap. After choosing the style and color for the website, we created GitHub and branches for each team member. We also assign specific pages for each team member to design and develop the Front-end and Back-end. We will explain our developing process in more detail in the following sections.

Firstly, we must build 3 register pages for Customer, Vendor, and Shipper and create the User schema to store the user information. To switch between each register page, there will be a small navigation bar on top of each register form. The navigation bar includes three parts: Customer, Vendor, and Shipper. Clicking on it will direct the



user to the chosen page. To make the pages visually appealing, we look for background images from the website freepik.com, which is an online image bank platform. The logo of the website is from canva.com. On the left of those register pages, there is a registration form for users to create their accounts. In our form, we constrained that the username is required and contains only letters (lower and upper case) and digits, has a length from 8 to 15 characters, the password is required and contains at least one upper case letter, at least one lower case letter, at least one digit, at least one special letter in the set !@#\$%^&* and has a length from 8 to 20 characters. Other fields are also required and have a minimum length of 5 characters. After completing the design of the interface, we continue to make the user details stored in our database. We create the User schema with 3 roles: Customer, Vendor, and Shipper. The username is unique across the whole system. The business name and business address are unique for vendors, and all the attributes in the schema have been constrained to the correct format and are required for every user. To hash passwords, we use “bcryptjs” package to make sure it is secured for users. The hardest thing is how to upload the profile picture and then we can display it on other pages. We spent a lot of time searching for any package that can help us on the internet, and we found “express-fileupload” to be the most suitable package for us. When users upload their profile picture, it will be stored in our database as a BinData object, and we can convert it into a base64 image and display it on the screen. Then, we created error messages for the user if they entered the existing username, business name, or business address.

(Screenshots: Appendix, Figure 1-3)

After creating the register pages and storing the user information in our database successfully, we continue to build the login page. The Login Page is the first page when users access our website. We found the Login template on YouTube since they taught step-by-step how to complete it so that it was easier to learn by step. Firstly, In the body section, we gave it a class, a div tag with a class name ‘img’ which means that we were put in the background image for the website called ‘Wave’ which stands for the orange wave in the background. Then, we created a container to wrap everything up, we found an image on Undraw by searching some topic relevant to an e-commercial shopping

website and we caught a beautiful image, we gave an ‘img’ tag and put it to another background image on the left, after that we created a class “login-content” started by the form tag that contains all input labels which are Username, Password, and main Login button. We also provided a logo on the top of the form and created a “Register Now” button to lead it to a Register Page if the user didn’t create an account. To animate the underline of labels, we also implemented JavaScript code to give two functions named ‘focusFunc’ and ’blurFunc’ that can motivate the underline effect from the middle to both sides. Then, we made this login page functional by checking if the username and password are matched to an existing user in our database, they will be navigated to the specific page related to their role and account detail. We hope that in the future we will have a chance to reuse this Login page and make it more attractive by adding on a ‘Forgot Password’ function or giving it more animation.

(Screenshots: Appendix, Figure 4)

We start working with the front-end of the Customer Page. We referenced some popular e-commerce websites such as Shopee, Lazada, and Amazon and some samples on the theme wagon website. Regarding the responsiveness of the website, we assume a lot of bootstrap elements to reduce the workload as much as we can but to meet our desire, we still applied CSS to the website. After four days of working with the front end, we have a review day and give feedback to the whole website front end and then all the members have one night to improve their part.

Next is the back end of the customer page, we use NodeJS, ExpressJS, and MongoDB to achieve all the functions. With the get method, we got all the products in the database, counted all the categories among products, and user information then rendered it to the homepage customer. The customer homepage has a user id on the path. On the homepage customer, we used a for loop to display all the products and a for loop to display all the categories. On the navigation bar, when a user searches for a product we use the attribute Find to search for that product in the Product database and render it to the search page and apply a for loop to display all the results. In the hot deal section, there is a filter function, we use the post method and when users apply minimum price and maximum price, the server will turn it into an integer and if the

user put it blank the result will be infinity positive or negative then the server will search that price in the database and render it to the homepage customer.

On the category page, with the banner, we use an if statement to recognize the category and apply the exact banner and the heading. On the server side, we use the app.get and find the products with the same category to render it in the category page with a for loop. To access the product detail page, a get method is applied and find the product with the same Id and render it to the product detail page. To access the shopping cart, users must find the shopping cart icon on the navigation bar. We created an object called cart with two attributes products as a list and totalPrice which stores an integer. Whenever users click add to cart there will be a post method to give a request to the body then the server finds and stores it as a variable then that product will be pushed to the products attribute of the cart object and the server adds the product's price to totalPrice. Next, they will render the cart to the shopping cart through the get method whenever users click on the shopping cart icon. Finally, a forEach() function is used to display the product array in the shopping cart. In the shopping cart, users can send the order through the order button. We have made a form with a post method and all attributes have the same name with the attribute in the order schema, if the user fulfills all the required information the server will query and store all that information through its name as a variable, create a new database in the Order schema and save it.

(Screenshots: Appendix, Figure 5-13)

To start developing Vendor Page, we have to begin the front-end, which took us a lot of time to choose a template and design it with HTML and CSS. While looking for a suitable template for this project, we were inspired by a design of a developer named Matfury on dribble.com, so we decided to choose it for our project. However, with the requirements of this project and our limited ability, we designed our website following the template, but with fewer features and pages. During the front-end development, with the utilization of Bootstrap, we were able to complete the front-end in a short while. In addition to HTML and CSS, we also used JavaScript to move between pages and the responsive toggle side menu.



After finishing the front-end, we continued to work with the back-end, which would be harder and more time-consuming. In this phase, we have collaborated with Customer Page and Shipper Page developers to create schemas for orders and products in MongoDB Atlas using Mongoose. We tried to design these schemas with fundamental attributes, according to requirements, and the work seemed easy and smooth at first. After that, the data of orders will be added by the Customer Page developer, and we were responsible for adding product data, so we need to create some routes with Express to add or delete data. In terms of product information, we were using the source from the Amazon store. Moreover, our static HTML file needs to be transferred into the EJS file, so that it will be dynamic. However, when we found out that images could be uploaded and saved in the database, we couldn't retrieve and display them on the website. Therefore, we have to spend many days fixing this problem by finding tutorials on Google or YouTube, after that we can solve this by using the library "express-fileupload" so that we can rewrite the path of images to display them. Additionally, our products are saved in the same database, so we need to filter the products by vendor's username using the if-else expression in the EJS file, which will just display the products of that vendor. Finally, the Orders table will display the data based on the database of orders from customers.

(Screenshots: Appendix, Figure 14-15)

The next page is the shipper page. At first, we designed the shipper page interface with HTML, CSS, and JavaScript because, at that time, we didn't get used to NodeJS and ExpressJS. This page contains a sidebar, a header, a footer, and the main content section. We copied the sidebar, header, and footer from Bootstrap and modified it a little bit by changing the primary color, adding a user image and information area, and changing the footer content. Then, we also design the main content section and the order detail modal by using Bootstrap and CSS, so that the whole page is responsive on many different devices. After we practice and get used to ExpressJS and MongoDB, we convert the HTML file into an EJS file. On this page, we used the User database and Order database to show the user and order information. We also used the conditional statement and for loop to make the main content section only display the

active order from the user's distribution hub. We use another for loop to create forms as modals for shippers to update the order's status. When users change the status of the order to "Delivered" or "Canceled", that order will not be displayed on the main page because we made a conditional statement that the main page only displays the order whose status is active.

(Screenshots: Appendix, Figure 16-17)

Then, we created account detail pages for Customers, vendors, and Shippers. We found templates on the internet by searching on YouTube and we choose a template with primary color that matches our project, then we start to work on it by finding similar elements on Bootstrap, we collected very useful sources on Bootstrap because the form was automatically responsive and attractive as well. However, the above-mentioned requirements are not attached to bootstrap templates so we self-coding, we added an 'img' class, below is a label named 'filetag' that contains the avatar image, then we created a file type with the id 'filetag' to connect it with JavaScript, at this point we add that filetag with 'addEventListener' function to change that image, then we created a variable name reader to attached it with new File Reader which will contain the new image. In addition, we created the Update button for users to update their account detail and a log-out button for users if they want to log out of their accounts. In the future, we may redesign a different logo that is suitable for this page since we used the main color for the background but accidentally the color was duplicated with the logo.

(Screenshots: Appendix, Figure 18-20)

Finally, we build some static pages such as About Us, Privacy, Refund, Terms, and Support. "About Us" page serves as an overview of the Aladin website, showcases what we can do, our vision and the reason to choose us. It shows our commitment to quality, a wide selection of goods for customers and exceptional customer service (*Screenshots, Figure...*). "Privacy" page states all the privacy practices of Aladin. First is the information that they collect, such as information related to user's accounts, business accounts and shipper accounts. Next is the statement of how they will use that information, along with Data security, third-party sharing, and cookies policy.

“Refund” page informs our customers about how our company deals with refunds or returns of the product that we are selling; this is also a set of rules and procedures that a business company will apply for every section of buy and sell. “Terms” page was made to emphasize the rules and regulations for visitors using our business’s website. It also plays a role in protecting our business by limiting liability as well as intellectual property rights, user behavior, and liability. “Support” page is a self-serve, web-providing customer service tool to help customers quickly when they are in trouble with buying or selling items. Support portals allow customers to find data and request assistance or find solutions for issues while using our website.

(Screenshots: Appendix, Figure 21-25)

After completing all the pages and functions of our website, we started to validate the HTML and CSS format of the website. There were many errors, but we spent a whole day solving most of them. However, there were some problems that we could not solve. Firstly, on the customer homepage, we put the product’s category name on the path for users to navigate to the specific category page, but there was a space in the category name, so it causes errors, and we did not know how to solve it. The second error is on the vendor homepage. We put the button tag inside the anchor tag to make that button functional, but when we put the anchor tag inside the button tag, the button has no function. The last error that we could not fix was in the shipper page, we used a for loop to make the order details, so those order keys were duplicated. If we delete it, we cannot display all the active orders.

(Screenshots: Appendix, Figure 26-40)

IV. Conclusion

Overall, we tried our best to complete a full-stack e-commerce website that achieved all the project requirements. After this project, we have gained a lot of experience and learned so many things. We know how the e-commerce website system works, how to make a connection between Front-end and Back-end in a website application and get used to trendy technology such as NodeJS, ExpressJS, and MongoDB. Moreover, we have significantly improved our research and teamwork skills, which are essential for our future careers. Finally, we hoped that in

the future, we will have an opportunity to continue developing this website and make it popular like Lazada, Shopee, and Amazon.

V. Peer Evaluation

Name	Role	Contribution	Tasks given
Tran Manh Cuong	Project Manager	20%	Manage the project workflow, develop the login page, register page, shipper page, develop shipper page's function, manage the database, record the demonstration video, write report.
Truong Quang Bao Loc	Technical Leader	20%	Develop the customer pages, develop customer page's function, manage the database, record the demonstration video, write report.
Nguyen Van Khai	Technical Leader	20%	Develop the vendor page, create extra features, develop shipper page's function, manage the database, record the demonstration video, write report.
Truong Tuong Hao	Developer	20%	Develop the login page, account detail page, static pages, record the demonstration video, write report.
Huynh Nguyen Thien Minh	Developer	20%	Develop the register pages, static pages, record the demonstration video, write report.

VI. Reference

- Bootstrap: <https://getbootstrap.com/docs/5.3/examples/>
- Dribbble: <https://dribbble.com/shots/21148180-Cargo-Delivery-SaaS-Finance-Page> , <https://dribbble.com/shots/16856809-Normaland-Customer-Service-Landing-Page-Tablet-Preview>

- My receptionist: <https://myreceptionist.com/>
- Nike terms of use:
[https://agreementservice.svs.nike.com/vn/en_gb/rest/agreement?agreementType=t
ermsOfUse&uxId=com.nike&country=VN&language=en&requestType=redirect](https://agreementservice.svs.nike.com/vn/en_gb/rest/agreement?agreementType=t ermsOfUse&uxId=com.nike&country=VN&language=en&requestType=redirect)
- Shopee: [https://shopee.vn/?utm_campaign=-&utm_content=----
&utm_medium=affiliates&utm_source=an_17348440051&utm_term=99dqj39roi
c7](https://shopee.vn/?utm_campaign=-&utm_content=----&utm_medium=affiliates&utm_source=an_17348440051&utm_term=99dqj39roi_c7)
- Coding Nepal: <https://www.youtube.com/watch?v=eeHqZeJ9Vqc>

VII. Appendix

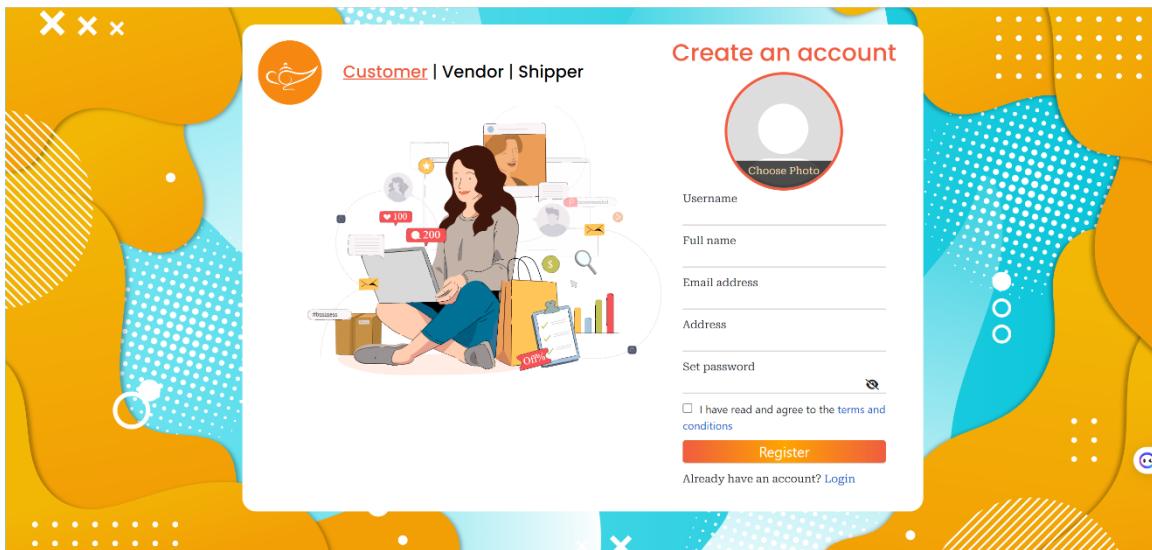


Figure 1: Register Page - Customer

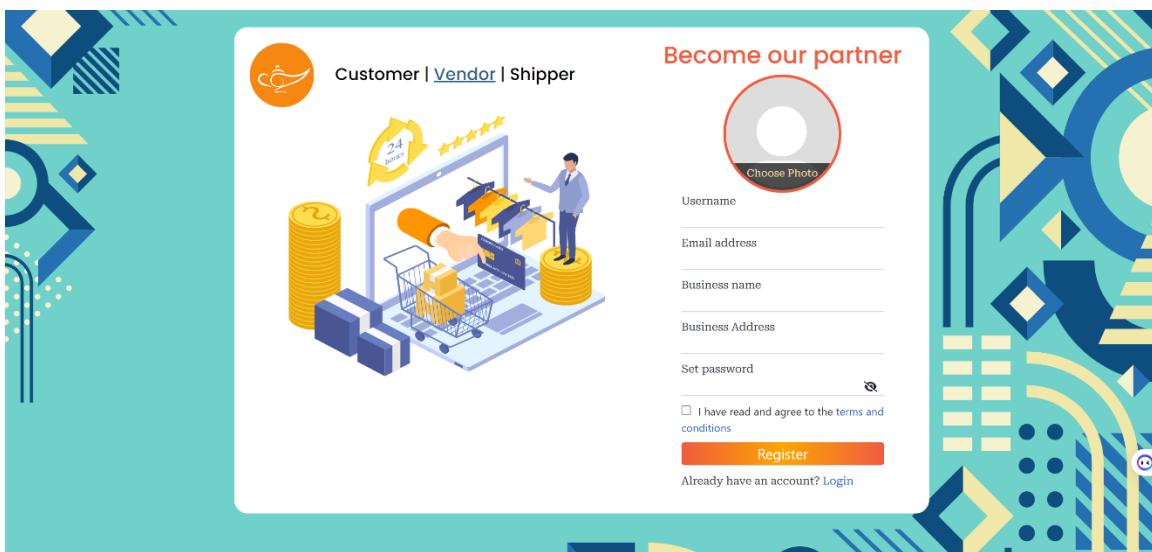


Figure 2: Register Page – Vendor

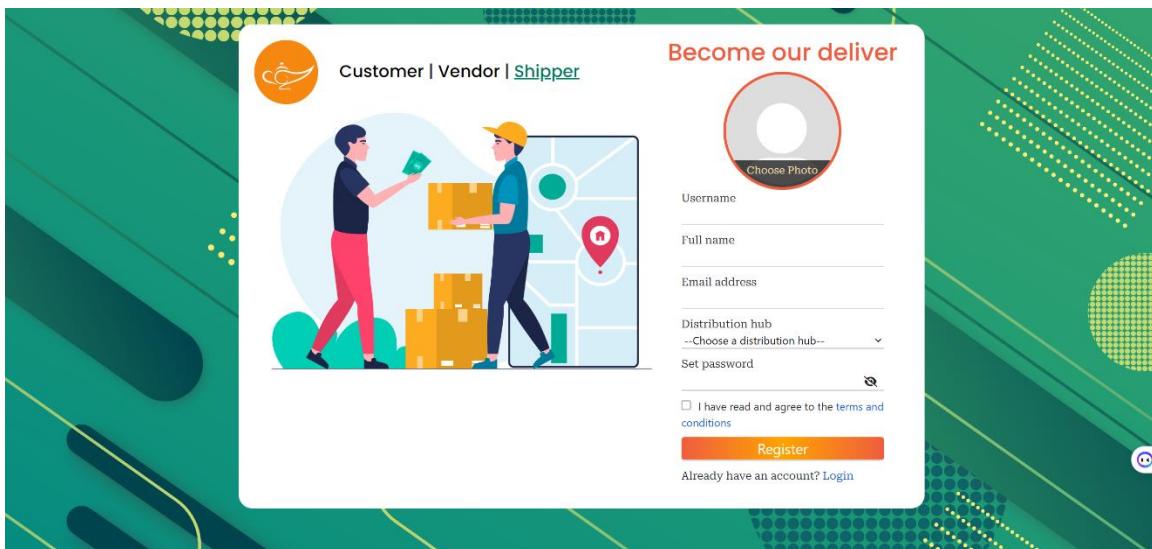


Figure 3: Register Page - Shipper

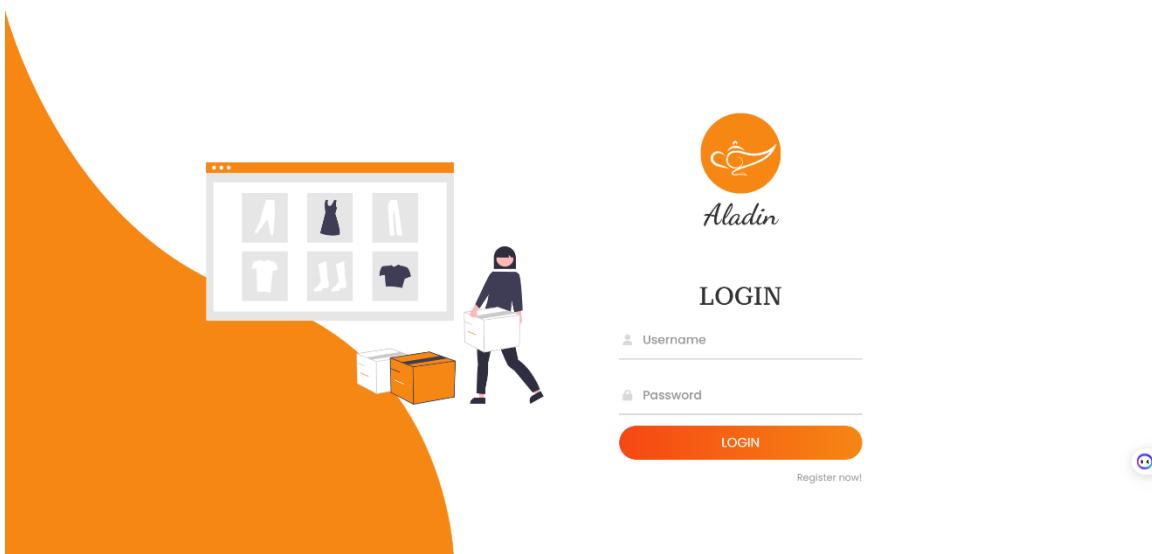


Figure 4: Login Page

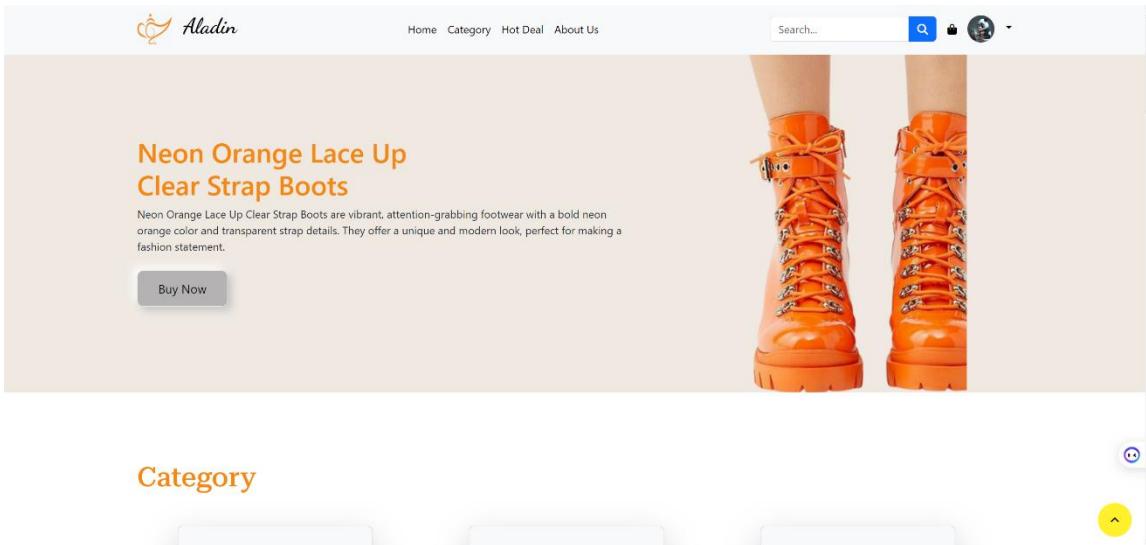


Figure 5: Customer Homepage

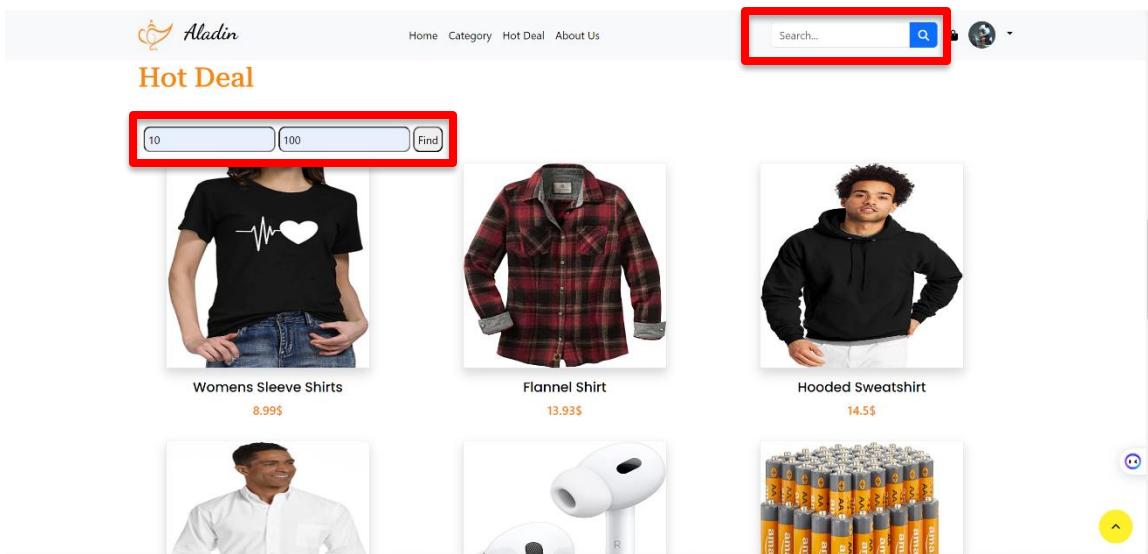


Figure 6: Filter and Search Functions

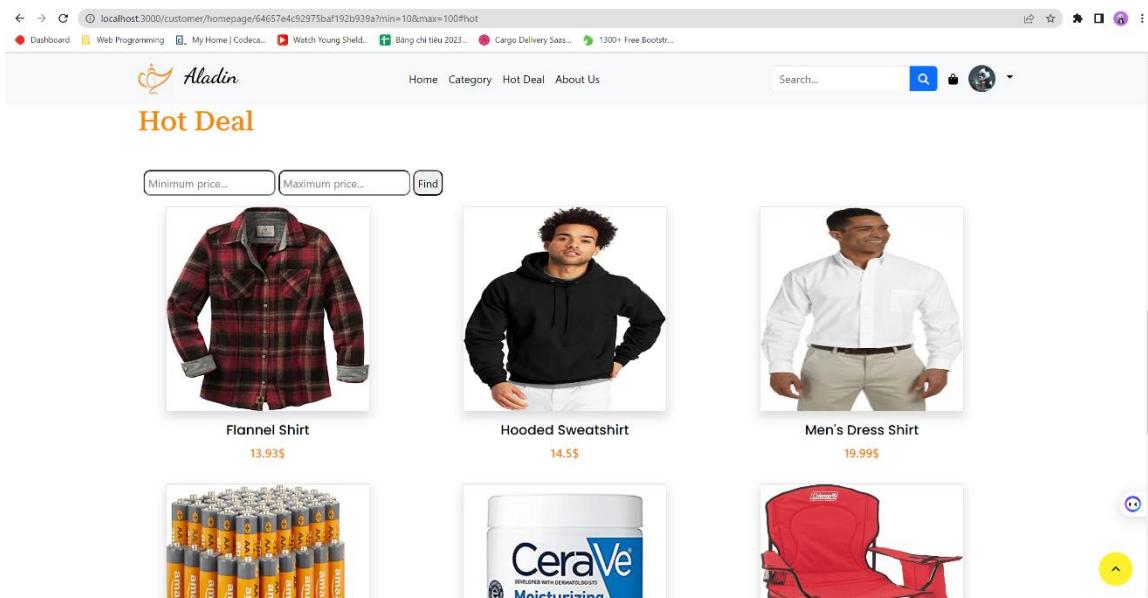


Figure 7: Filter Result



 Aladin

Home Category Hot Deal About Us

Search...   

Results



Womens Sleeve Shirts
8.99\$



Flannel Shirt
13.93\$



Men's Dress Shirt
19.99\$

^

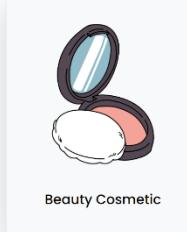
Figure 8: Search Result

 Aladin

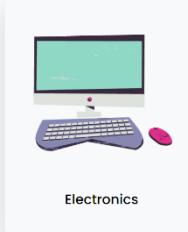
Home Category Hot Deal About Us

Search...   

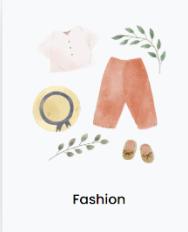
Category



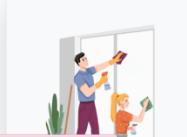
Beauty Cosmetic



Electronics



Fashion



Household



Groceries



Sports

^

Figure 9: Categories

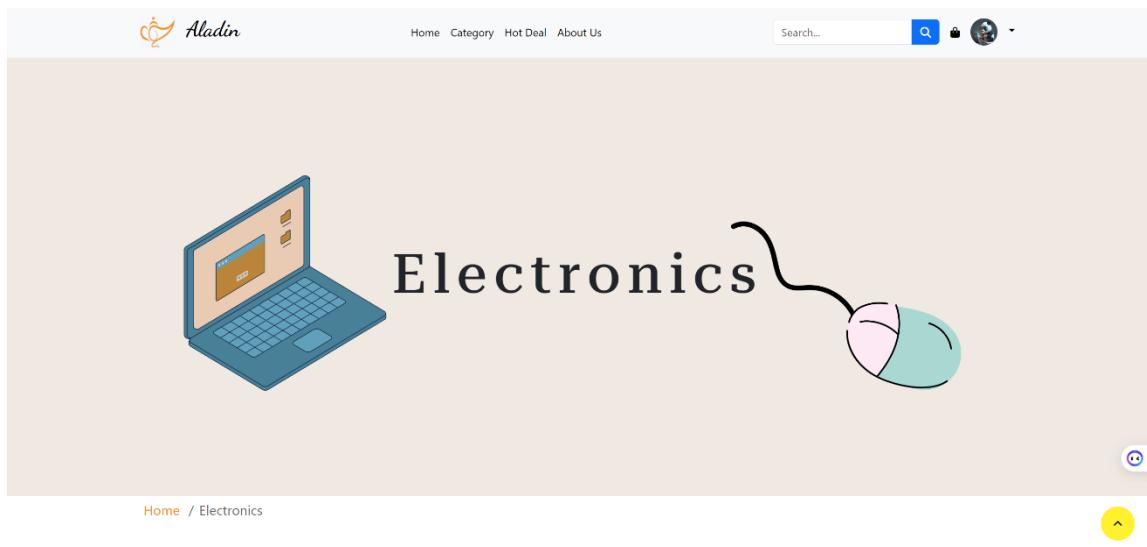


Figure 10: Category Page – Banner

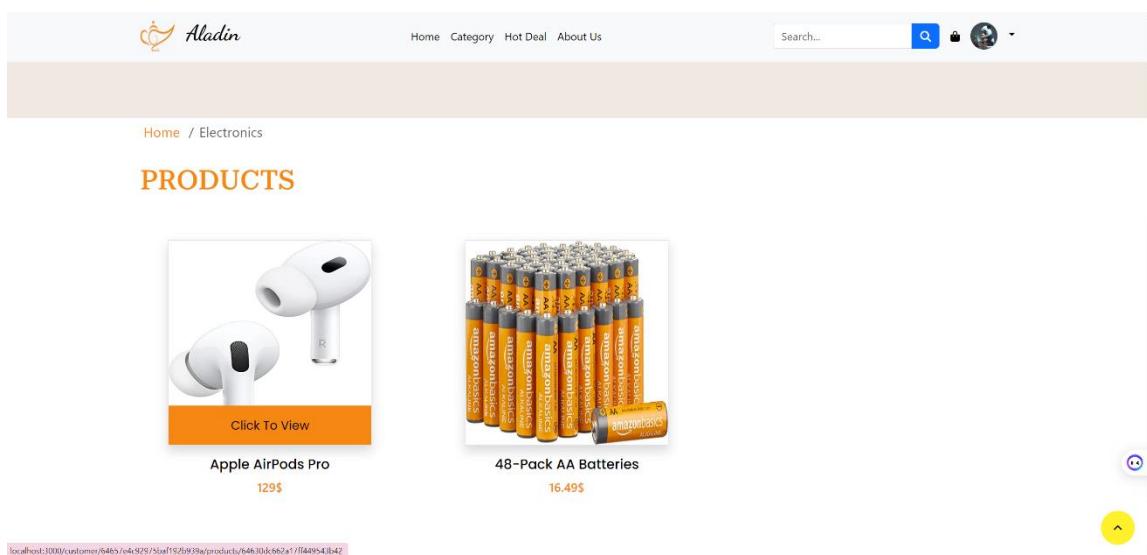


Figure 11: Category Page - Products



The screenshot shows a product detail page for Apple AirPods Pro. At the top left is the Aladin logo. The main image displays two white AirPods Pro units. To the right, the product name "Apple AirPods Pro" is shown with a 5-star rating. A large orange button indicates the price as "129\$". Below the price is a "Product Description" section with a brief summary. An "Add to cart" button is located at the bottom left of the product card.

Figure 12: Product Detail Page

The screenshot shows a shopping cart page. At the top left is the Aladin logo. The main area is titled "My Shopping Cart". It lists items: "Apple AirPods Pro" at 129 \$ and "48-Pack AA Batteries" at 16.49 \$. The total price is listed as 145.49 \$. Below the cart summary, there are fields for "Receiver" (Truong Loc), "Address" (123 Ton Duc Thang), and "Payment Method" (Cash on Delivery (COD)). A red "Order" button is positioned at the bottom right of the cart summary area. The footer contains links for About Us, Privacy, Refund, Terms, Support, and a copyright notice: Copyright © Aladin, Inc.

Figure 13: Shopping Cart



Hello, khainguyen

Earning \$0

Products

Add Product +

Name	Image	Price	Description	Category	Amount	Action
Womens Sleeve Shirts		\$8.99	Women's cute casual short sleeve tops, women's summer-fall tee shirt easily pair with others. For example, shorts, long pants, jeans, dress, leggings, boots and etc. Such a beautiful and graceful tee shirt will make you be more fashionable and attractive! The perfect top for spring, summer, autumn, and winter.	Fashion	120	Delete X
Flannel Shirt		\$13.93	Let loose in our timeless flannel, featuring a classic design with chambray lined collar and cuffs. Made of soft brushed mid-weight cotton flannel, this yarn-dyed shirt is just right and very comfortable. Wear it alone or layer up during those brisk fall days. Pairs perfectly with our Womens Quilted Vest.	Fashion	122	Delete X

View Your Account

Aladin

Figure 14: Vendor Homepage

Hello, khainguyen

Earning \$0

Products

Orders 12

Product Form

Add your products here

Fill in the product information

Name

Price

Select Category

Description

Amount

Upload Your Product's Image

Figure 15: Add New Product Form



The screenshot shows the Aladin E-commerce Website - Shipper Homepage. The interface has an orange header bar with the Aladin logo and a user profile for Tran Manh Cuong. Below the header is a sidebar with a dashboard link and a log out button. The main content area displays a table of active orders:

Invoice	Receiver	Payment Method	Item List	Address	Status
#INV-0001	Tran Manh Cuong	Internet Banking	Fitness Tracker Non-Scratch Sponges Wine Fridge	123 Nguyen Van Linh	Active
#INV-0002	Truong Quang Bao Loc	Internet Banking	Flannel Shirt Hooded Sweatshirt	456 Nguyen Van Linh	Active

A sidebar on the right shows the Tan Son Nhat Cargo Terminal with two active status entries. At the bottom of the page are links for About Us, Privacy, Refund, Terms, and Support.

Figure 16: Shipper Homepage

The screenshot shows an 'Invoice Details' modal window. The modal contains the following information:

- Order No.**: 64630aa9027f597958cb108
- Distribution Hub**: Tan Son Nhat Cargo Terminal
- Date**: Thu Sep 28 2023 07:00:00 GMT+0700 (Indochina Time)
- Status**: Active
- Item list** (partial list):
 - Fitness Tracker: 99.95\$
 - Non-Scratch Sponges: 3.23\$
 - Wine Fridge: 317.23\$
- Total Price**: 420.41\$
- Receiver**: Tran Manh Cuong
- Address**: 123 Nguyen Van Linh
- Payment Method**: Internet Banking

At the bottom of the modal are 'Close' and 'Save changes' buttons.

Figure 17: Order Detail Modal



ACCOUNT DETAIL



Username <input type="text" value="emsignouc"/>	Full Name <input type="text" value="Tran Manh Cuong"/>
Email <input type="text" value="emsignouc0407@gmail.com"/>	Address <input type="text" value="33 Nguyen Huu Tho"/>
<input type="button" value="Update"/>	<input type="button" value="Logout"/>



Figure 18: Customer Account Detail Page

ACCOUNT DETAIL



Username <input type="text" value="khainguyen123"/>	Business Name <input type="text" value="khainguyen"/>
Business Address <input type="text" value="100 Nguyen Van Linh"/>	Email <input type="text" value="nvkhail05@gmail.com"/>
<input type="button" value="Update"/>	<input type="button" value="Logout"/>



Figure 19: Vendor Account Detail Page



ACCOUNT DETAIL



Username <input type="text" value="emsignouc1234"/>	Full Name <input type="text" value="Tran Manh Cuong"/>
Email <input type="text" value="emsignouc0407@gmail.com"/>	Distribution Hub <input type="text" value="Tan Son Nhat Cargo Terminal"/>
<input type="button" value="Update"/>	<input type="button" value="Logout"/>

Figure 20: Shipper Account Detail Page



[« Back To Homepage](#)

Aladin

About Us

Welcome to Aladin!

At Aladin, we believe in making online shopping an effortless and enjoyable experience. We are an e-commerce platform dedicated to providing you with a wide range of high-quality products at affordable prices. Whether you're searching for trendy fashion, cutting-edge electronics, or unique home decor, we've got you covered.

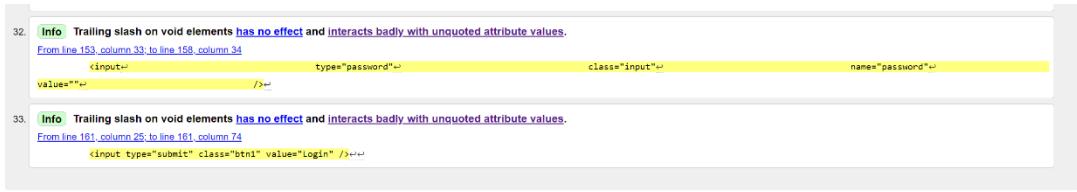
Our mission is to connect consumers with top-notch products from around the world, delivered right to your doorstep. We understand the importance of convenience and accessibility, which is why we've built an intuitive online platform that allows you to explore, select, and purchase your desired items with ease.

Why Choose Us?

- Quality Assurance: We are committed to offering products that meet the highest standards of quality.
- Wide Selection: We take pride in offering a diverse selection of products to cater to every individual's taste and preference. From the latest fashion trends to state-of-the-art gadgets, we have something for everyone.
- Exceptional Customer Service: We strive to provide outstanding customer service that exceeds your expectations. Our friendly and knowledgeable support team is here to assist you with any inquiries, concerns, or feedback you may have. We are dedicated to resolving any issues promptly and ensuring your satisfaction.

Thank you for choosing Aladin. We are thrilled to have you here, and we look forward to providing you with an exceptional online shopping experience.

Figure 21: About Us Page



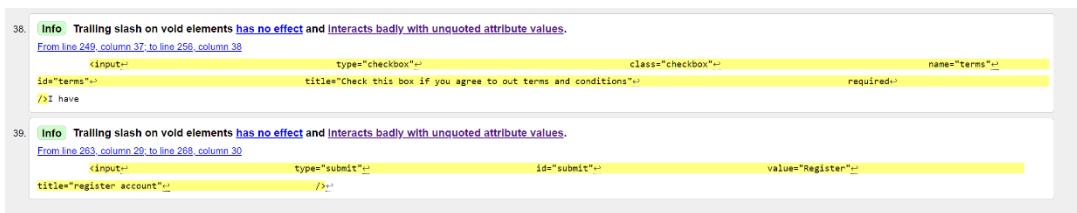
Source

```

1. <!-- RMIT University Vietnam-->
2. <!-- Course: COSC2438 Web Programming-->
3. Semester: 2023A
4. Assessment: Assignment 2
5. Author:<br>
6. 1. Tran Hanh Cuong (#3974735)<br>
7. 2. Truong Quang Bao Loc (#3965528)<br>
8. 3. Nguyen Van Khai (#3979259)<br>
9. 4. Truong Tuong Hao (#3979259)<br>
10. 5. Huynh Nguyen Thien Minh (#3978742)<br>
11. ...
12. --><!DOCTYPE html>
13. <html lang="en">
14.   <head>
15.     <meta charset="UTF-8" />
16.     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
17.     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
18.     <title>Login Now!</title>
19.   </head>
20.   <body>
21.     <link rel="stylesheet" href="/css/reset.css" />
22.     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KX94CHF1Lc+nY2dmCuHq09rCG1rCGa5gtU4mk92HdYeM/SXH301p5Ily+dN9+nJ02" crossorigin="anonymous">
23.     ...
24.     <link rel="stylesheet" href="/css/login-account.css" />
25.     ...
26.   </body>
27. </html>
28. ...
29. ...

```

Figure 26: Login Page Validation Result



Source

```

1. <!-- RMIT University Vietnam-->
2. <!-- Course: COSC2438 Web Programming-->
3. Semester: 2023A
4. Assessment: Assignment 2
5. Author:<br>
6. 1. Tran Hanh Cuong (#3974735)<br>
7. 2. Truong Quang Bao Loc (#3965528)<br>
8. 3. Nguyen Van Khai (#3979259)<br>
9. 4. Truong Tuong Hao (#3979259)<br>
10. 5. Huynh Nguyen Thien Minh (#3978742)<br>
11. ...
12. --><!DOCTYPE html>
13. <html lang="en">
14.   <head>
15.     <meta charset="UTF-8" />
16.     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
17.     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
18.     <title>Register Now!</title>
19.   </head>
20.   <body>
21.     <!-- CSS -->
22.     <link rel="stylesheet" href="/css/reset.css" />
23.     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KX94CHF1Lc+nY2dmCuHq09rCG1rCGa5gtU4mk92HdYeM/SXH301p5Ily+dN9+nJ02" crossorigin="anonymous">
24.     ...
25.     <link rel="stylesheet" href="/css/register-account.css" />
26.     ...

```

Figure 27: Register Page Validation Result

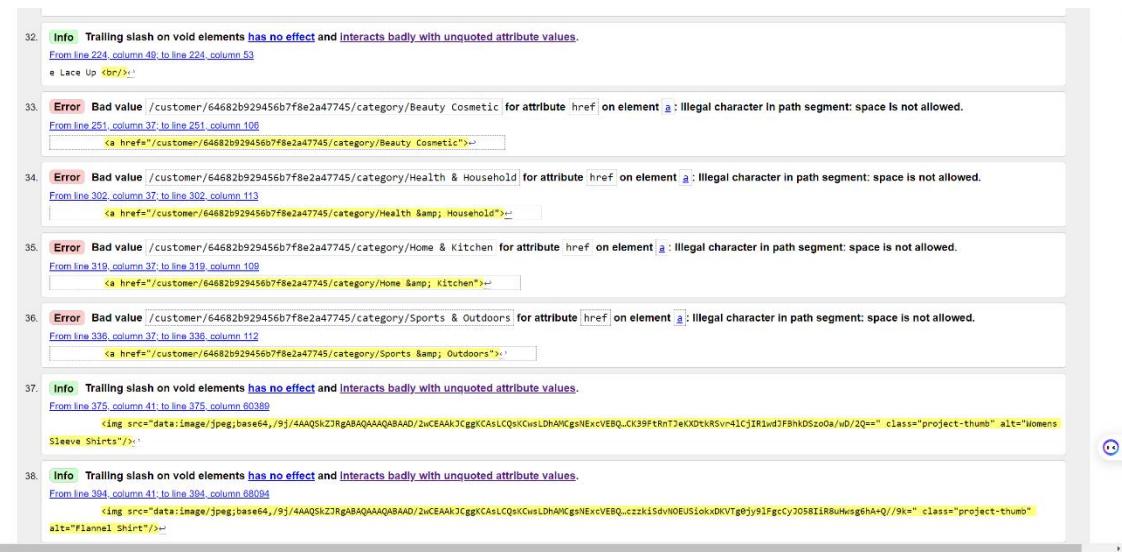
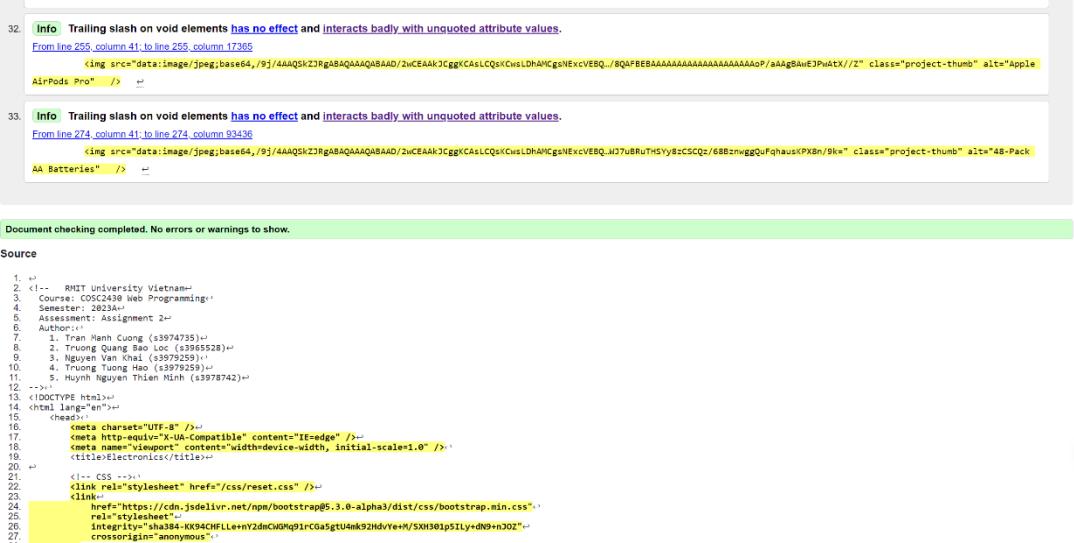


Figure 28: Customer Homepage Validation Result



The screenshot shows a validation report for a Category Page. It lists two errors (32 and 33) and a note at the bottom stating "Document checking completed. No errors or warnings to show." The errors are similar to those in Figure 28, involving trailing slashes on void elements. The source code shown at the bottom is the HTML code for the category page, which includes meta tags for mobile compatibility and a Bootstrap CSS link.

```

1. <!-- RMIT University Vietnam-->
2. <!-- Course: COSC2939 Web Programming-->
3. <!-- Semester: 2/2018 -->
4. <!-- Assessment: Assignment 2-->
5. <!-- Authors: -->
6. <!-- 1. Tran Hanh Chung (s3974735) -->
7. <!-- 2. Truong Quang Bao Loc (s3965528) -->
8. <!-- 3. Nguyen Van Khai (s3979259) -->
9. <!-- 4. Truong Tuong Hao (s3979259) -->
10. <!-- 5. Huynh Nguyen Thien Minh (s3978742) -->
11. -->` 
12. <!DOCTYPE html>
13. <html lang="en">
14. <head>
15.   <meta charset="UTF-8" />
16.   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
17.   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
18.   <title>Electronics</title>
19. </head>
20. <body>
21.   <!-- CSS -->
22.   <link rel="stylesheet" href="/css/reset.css" />
23.   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">
24.     integrity="sha384-KX94CHILLe+nY2dmCM0hq91rCGa5gtU4mk92MdYe=r/SXH301p5ILy+dN9+nJ0Z"-
25.     crossorigin="anonymous">
26. 
27. 
```

Figure 29: Category Page Validation Result



Figure 32 shows the validation results for the Vendor Home Page. The results are listed in a table with numbered rows from 30 to 36. Rows 30, 31, 32, 34, 35, and 36 are marked as errors, while rows 33 and 37 are marked as info.

30.	Error The element button must not appear as a descendant of the a element. From line 222, column 168; to line 222, column 192
31.	Info Trailing slash on void elements has no effect and interacts badly with unquoted attribute values. From line 232, column 57; to line 232, column 60088
32.	Error The element button must not appear as a descendant of the a element. From line 240, column 168; to line 240, column 192
33.	Info Trailing slash on void elements has no effect and interacts badly with unquoted attribute values. From line 250, column 57; to line 258, column 34512
34.	Error The element button must not appear as a descendant of the a element. From line 258, column 168; to line 258, column 192
35.	Info Trailing slash on void elements has no effect and interacts badly with unquoted attribute values. From line 268, column 57; to line 268, column 36732
36.	Error The element button must not appear as a descendant of the a element. From line 276, column 168; to line 276, column 192

Figure 32: Vendor Home Page Validation Result

Figure 33 shows the validation results for the Shipper Home Page. The results are listed in a table with numbered rows from 35 to 40. Rows 35, 36, 37, 39, and 40 are marked as errors, while rows 38 and 40 are marked as warnings.

35.	Error Duplicate ID order-detail. From line 581, column 57; to line 587, column 79
36.	Warning The first occurrence of ID order-detail was here. From line 289, column 57; to line 295, column 79
37.	Error Duplicate ID status. From line 598, column 81; to line 598, column 149
38.	Warning The first occurrence of ID status was here. From line 326, column 81; to line 326, column 149
39.	Error Duplicate ID order-detail. From line 802, column 57; to line 808, column 79
40.	Warning The first occurrence of ID order-detail was here. From line 289, column 57; to line 295, column 79

Figure 33: Shipper Home Page Validation Result



Document checking completed. No errors or warnings to show.

Source

```

1. <!-- RMIT University Vietnam-->
2. Course: COSC2430 Web Programming</>
3. Semester: 2023A</>
4. Assessment: Assignment 2</>
5. Authors:
6. 1. Tran Hanh Cuong (s3974735)</>
7. 2. Truong Quang Bao Loc (s3965528)</>
8. 3. Tran Ngan Lan (s3974737)</>
9. 4. Truong Tuong Hao (s3979259)</>
10. 5. Huynh Nguyen Thien Minh (s3978742)</>
11. <br>
12. <></>
13. <!DOCTYPE html>
14. <html lang="en">
15.   <head>
16.     <meta charset="UTF-8">
17.     <title>Account Detail Page</title>
18.     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
19.     <link rel="stylesheet" type="text/css" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
20.     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
21.   </head>
22.   <body>
23.     <link rel="stylesheet" href="css/reset.css" />
24.     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KKK4CHFLLennY2dmCWQ9irCGa5gtU4mk92HdvYeM/SXH301p5ILy+dN9+jN0Z" crossorigin="anonymous">
25.   </body>
26. </html>

```

Figure 34: Account Detail Page Validation Result

Document checking completed. No errors or warnings to show.

Source

```

1. <!-- RMIT University Vietnam-->
2. Course: COSC2430 Web Programming</>
3. Semester: 2023A</>
4. Assessment: Assignment 2</>
5. Authors:
6. 1. Tran Hanh Cuong (s3974735)</>
7. 2. Truong Quang Bao Loc (s3965528)</>
8. 3. Tran Ngan Lan (s3974737)</>
9. 4. Truong Tuong Hao (s3979259)</>
10. 5. Huynh Nguyen Thien Minh (s3978742)</>
11. <br>
12. <></>
13. <!DOCTYPE html>
14. <html lang="en">
15.   <head>
16.     <meta charset="UTF-8" />
17.     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
18.     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
19.     <title>About Us</title>
20.   </head>
21.   <body>
22.     <link rel="stylesheet" href="css/reset.css" />
23.     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KKK4CHFLLennY2dmCWQ9irCGa5gtU4mk92HdvYeM/SXH301p5ILy+dN9+jN0Z" crossorigin="anonymous">
24.   </body>
25. </html>

```

Figure 35: About Us Page Validation Result



Document checking completed. No errors or warnings to show.

Source

```

1. <!-- RMIT University Vietnam-->
2. <!-- Course: COSC438 Web Programming-->
3. Semester: 2023A<br>
4. Assessment: Assignment 2<br>
5. Author:<br>
6. 1. Tran Hanh Cuong (#3974735)<br>
7. 2. Truong Quang Bao Loc (#3965528)<br>
8. 3. Nguyen Van Khai (#3979259)<br>
9. 4. Truong Tuong Hao (#3979259)<br>
10. 5. Huynh Nguyen Thien Ninh (#3978742)<br>
11. --><!DOCTYPE html>
12. <html lang="en">
13.   <head>
14.     <meta charset="UTF-8" />
15.     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
16.     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
17.     <title>Privacy Policy</title>
18.   </head>
19.   <body>
20.     <!-- CSS -->
21.     <link rel="stylesheet" href="/css/reset.css" />
22.     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KX94CHFLLe+nY2dmCuHdQ9rG1rC5qF92HveM9fDj0HwvYe=M/SKH30ip5ILy+dN9+nJ02" crossorigin="anonymous" />
23.   </body>
24. </html>
25. 
```

Figure 36: Privacy Policy Page Validation Result

Document checking completed. No errors or warnings to show.

Source

```

1. <!-- RMIT University Vietnam-->
2. <!-- Course: COSC438 Web Programming-->
3. Semester: 2023A<br>
4. Assessment: Assignment 2<br>
5. Author:<br>
6. 1. Tran Hanh Cuong (#3974735)<br>
7. 2. Truong Quang Bao Loc (#3965528)<br>
8. 3. Nguyen Van Khai (#3979259)<br>
9. 4. Truong Tuong Hao (#3979259)<br>
10. 5. Huynh Nguyen Thien Ninh (#3978742)<br>
11. --><!DOCTYPE html>
12. <html lang="en">
13.   <head>
14.     <meta charset="UTF-8" />
15.     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
16.     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
17.     <title>Refund Policy</title>
18.   </head>
19.   <body>
20.     <!-- CSS -->
21.     <link rel="stylesheet" href="/css/refund-terms-support.css" />
22.     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KX94CHFLLe+nY2dmCuHdQ9rG1rC5qF92HveM9fDj0HwvYe=M/SKH30ip5ILy+dN9+nJ02" crossorigin="anonymous" />
23.   </body>
24. </html>
25. 
```

Figure 37: Refund Policy Page Validation Result

Document checking completed. No errors or warnings to show.

Source

```

1. <!-- RMIT University Vietnam-->
2. <!-- Course: COSC4340 Web Programming-->
3. Semester: 2023A
4. Assessment: Assignment 2
5. Author:<br>
6. 1. Tran Hanh Cuong (<a href="mailto:s3974735@gmail.com">s3974735</a>)
7. 2. Truong Quang Bao Loc (<a href="mailto:s3965528@gmail.com">s3965528</a>)
8. 3. Nguyen Van Khai (<a href="mailto:s3979259@gmail.com">s3979259</a>)
9. 4. Truong Tuong Hao (<a href="mailto:s3979259@gmail.com">s3979259</a>)
10. 5. Huynh Nguyen Thien Ninh (<a href="mailto:s3978742@gmail.com">s3978742</a>)
11. -->
12. <!-- CSS -->
13. <link rel="stylesheet" href="/css/refund-terms-support.css" />
14. <!-- JS -->
15. <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" integrity="sha384-KX9dIYL1+q92HvO5X0y3QD4fZI002d8J46t9Qj4ZoGhWwQD43d3Dw6F0ZwWZLW&lt;!-- crossorigin="anonymous" -->
16. <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-KX9dIYL1+q92HvO5X0y3QD4fZI002d8J46t9Qj4ZoGhWwQD43d3Dw6F0ZwWZLW&lt;!-- crossorigin="anonymous" -->
17. </body>
18. 
```

Figure 38: Terms of Use Page Validation Result

Document checking completed. No errors or warnings to show.

Source

```

1. <!-- RMIT University Vietnam-->
2. <!-- Course: COSC4340 Web Programming-->
3. Semester: 2023A
4. Assessment: Assignment 2
5. Author:<br>
6. 1. Tran Hanh Cuong (<a href="mailto:s3974735@gmail.com">s3974735</a>)
7. 2. Truong Quang Bao Loc (<a href="mailto:s3965528@gmail.com">s3965528</a>)
8. 3. Nguyen Van Khai (<a href="mailto:s3979259@gmail.com">s3979259</a>)
9. 4. Truong Tuong Hao (<a href="mailto:s3979259@gmail.com">s3979259</a>)
10. 5. Huynh Nguyen Thien Ninh (<a href="mailto:s3978742@gmail.com">s3978742</a>)
11. -->
12. <!-- CSS -->
13. <link rel="stylesheet" href="/css/refund-terms-support.css" />
14. <!-- JS -->
15. <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" integrity="sha384-KX9dIYL1+q92HvO5X0y3QD4fZI002d8J46t9Qj4ZoGhWwQD43d3Dw6F0ZwWZLW&lt;!-- crossorigin="anonymous" -->
16. <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-KX9dIYL1+q92HvO5X0y3QD4fZI002d8J46t9Qj4ZoGhWwQD43d3Dw6F0ZwWZLW&lt;!-- crossorigin="anonymous" -->
17. </body>
18. 
```

Figure 39: Support Page Validation Result



The W3C CSS Validation Service
W3C CSS Validator results for TextArea (CSS level 3 + SVG)

Jump to: Validated CSS

Congratulations! No Error Found.

This document validates as [CSS level 3 + SVG!](#)

To show your readers that you've taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the XHTML you could use to add this icon to your Web page:

```
<p>  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
  </a>
</p>
```



```
<p>  <a href="http://jigsaw.w3.org/css-validator/check/referer">
    
  </a>
</p>
```

(close the img tag with > instead of /> if using HTML <= 4.01)

The W3C validators rely on community support for hosting and development.
[Donate](#) and help us build better tools for a better web.

Figure 40: CSS Validation Result