

ASSIGNMENT 2 FRONT SHEET

Qualification	BTEC Level 5 HND Diploma in Computing		
Unit number and title	10: Website Design & Development		
Submission date		Date Received 1st submission	
Re-submission Date		Date Received 2nd submission	
Student Name	Nguyễn Việt Hà	Student ID	GCH200035
Class	GCH0903	Assessor name	Nguyễn Đình Trần Long
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	

Grading grid

P5	P6	P7	M4	M5	D2	D3

⚙ **Summative Feedback:**

⚙ **Resubmission Feedback:**

Grade:

Assessor Signature:

Date:

Signature & Date:

Table of content:

Contents

I. Web design	5
1.1 Client and user requirement analysis	5
1.2 Case diagram	6
1.3 Database diagram	8
1.4 Site map	10
1.5 Wireframe	10
II. Web implementation	26
1. Web principles, standards and guidelines	26
2. In my website	35
3. Functional screens shot of multipage website	43
4. Some screenshots related to error handling	74
III. Web testing	75

Table of figure:

Figure 1: Case diagram	7
Figure 2: Database diagram	8
Figure 3: Site map	10
Figure 4: Customer Home Wireframe	11
Figure 5: Customer product wireframe	12
Figure 6: Product detail wireframe	13
Figure 7: Customer category wireframe	14
Figure 8: Admin login wireframe	15
Figure 9: Admin Home wireframe	16
Figure 10: View product wireframe	17
Figure 11: Update product wireframe	18
Figure 12: Add Product wireframe	19
Figure 13: View category wireframe	20
Figure 14: Update category wireframe	21
Figure 15: Add category wireframe	22
Figure 16: View user wireframe	24
Figure 17: Add user wireframe	25
Figure 18: Customer homepage	36
Figure 19: Admin homepage	36
Figure 20: Home Image	37
Figure 21: Product Image	38
Figure 22: Customer Product List	40
Figure 23: Admin Product List	41
Figure 24: Admin User List	42
Figure 25: Admin Category List	42
Figure 26: Customer Category List	42
Figure 27: Customer Navigation Bar	43
Figure 28: Admin Navigation Bar	43

I. Web design

1.1 Client and user requirement analysis

- Customers can view the product list.
- Customers can view the categories list.
- Customer can see the product detail.
- Admin can login to view, add, update or delete products, categories and users.
- Admin can login and logout of their account.

1.2 Case diagram

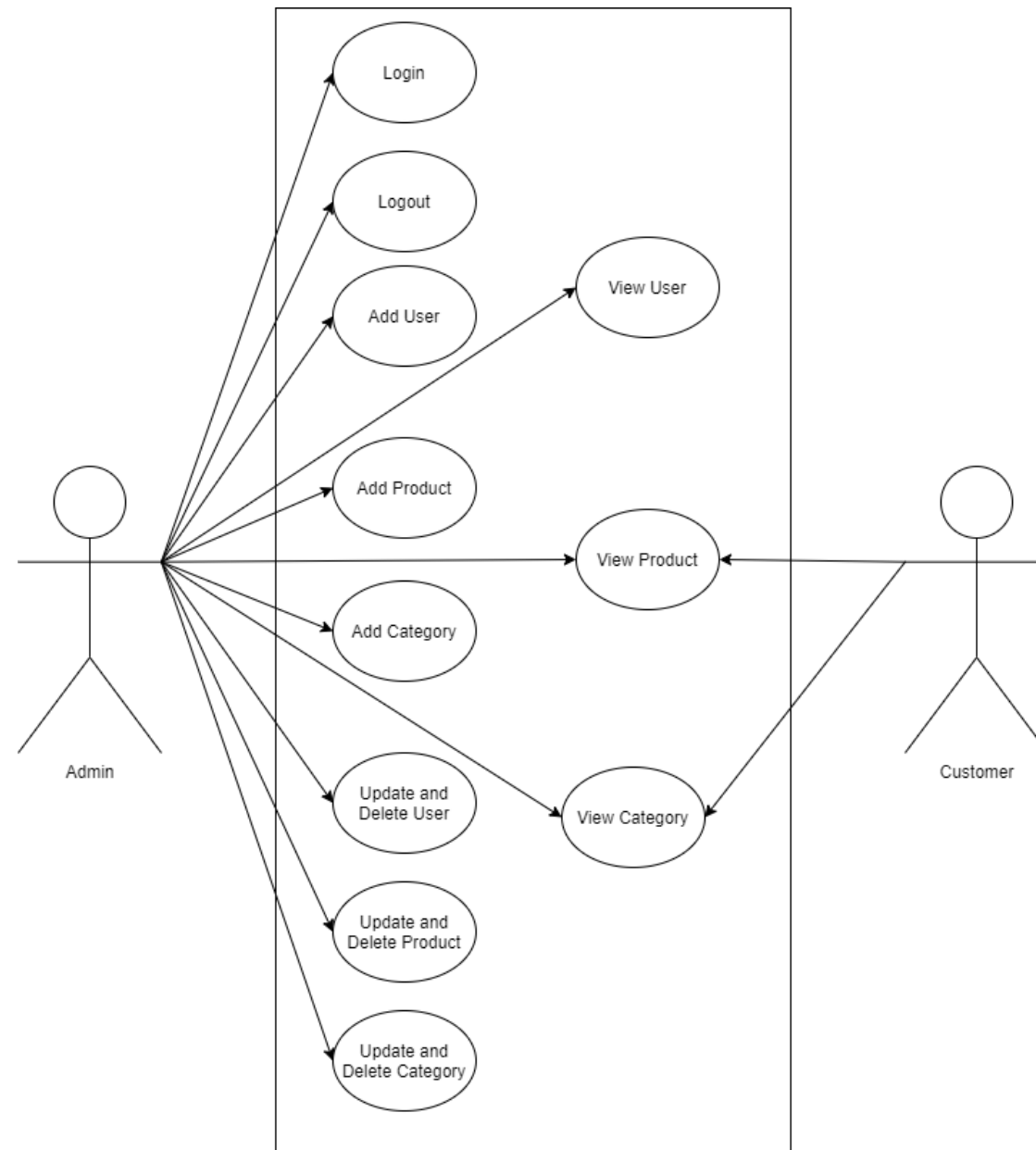


Figure 1: Case diagram

When visiting the website, the case diagram explains the positions of the user and the administrator. The admin has access to the admin website interfaces and can introduce new users, products, categories, and logout as a user in the admin job. Both the admin and the user can see the categories and items, but only the admin can edit and uninstall them from their own interface. Present accounts can be modified as well as deleted by the administrator. The user will also look at the items that fit their category by looking at the category info.

1.3 Database diagram

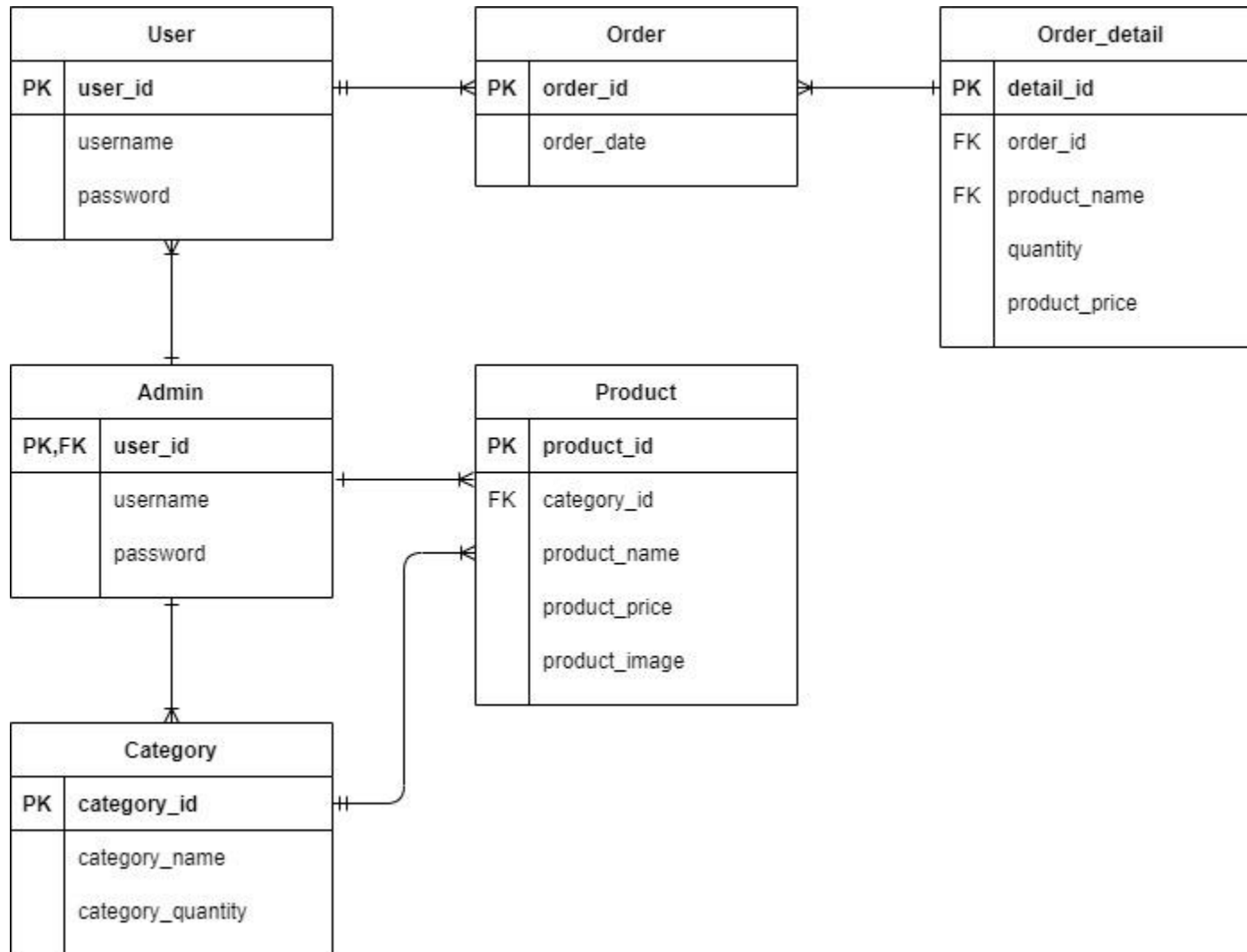


Figure 2: Database diagram

- The primary key for the user entity is `user_id`, which is the most important piece of information for identifying a user.

- Since admin accounts are similar to user_id with the same detail information, the admin entity has user id as a primary key and also a foreign key from the user entity.
- To classify the category entity, it has its own category_id.
- To make searching simpler, each product has its own product_id. To classify goods in one category to those in another, the commodity entity has a foreigner key called category_id.
- Each order has its own unique identifier call order_id.
- The detail_id of the order detail is used to describe it. The order_id foreign key in order details is used to find the instructions that are found in the detail.
- An admin can create one or more accounts, and one admin can create one user account, so the admin-user relationship is MANY TO ONE.
- One admin can create one or more categories, and one admin can create one category account, so the admin-category relationship is MANY TO ONE.
- An admin can create one or more goods, and one admin can create one product account, so the admin-product relationship is MANY TO ONE.
- An order can be taken by one or more users, but an order can only be taken by one user, so the user-order relationship is MANY TO ONE.
- A category can have one or more products within it, but a product can only belong to one category, so the category-product relationship is MANY TO ONE.
- One or more orders can be listed in a single order detail, and one order can be listed in several order details. As a result, the order-order detail relationship is ONE TO MANY.

1.4 Site map

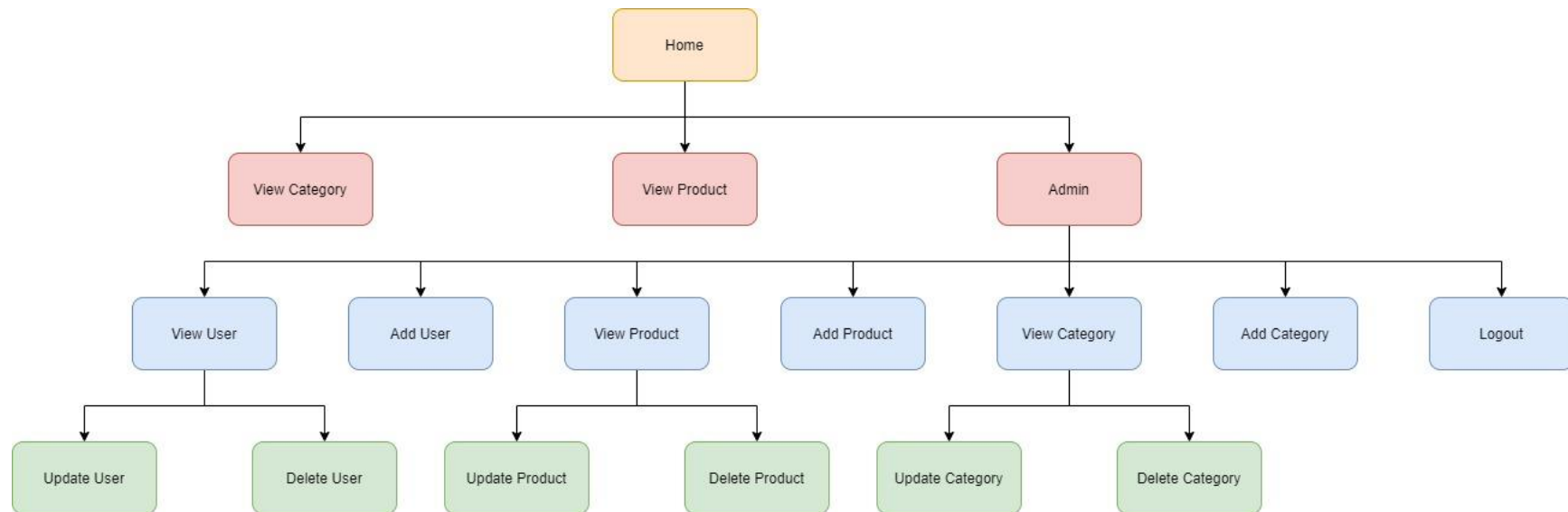


Figure 3: Site map

In the site map mentioned above, every user will first see the home page of the website. In the customer home page, user can select 2 options “View Category” and “View Product” to see all categories and products. But if the user login to the admin account, they will see a different home page that includes “View User” to see all the account, “Add User” to add more accounts, “View Product” to see all products, “Add Product” to add in more products, “View Category” to see all categories, “Add Category” to add in more categories and “Logout” to logout off the admin’s account. And in each “View User/Product/Category” in the admin home page, there are 2 options named “Update User” and “Delete User” for the “View User” page, “Update Product” and “Delete Product” for the “View Product” page, “Update Category” and “Delete Category” for the “View Category” page.

1.5 Wireframe

1.5.1 Customer home Wireframe

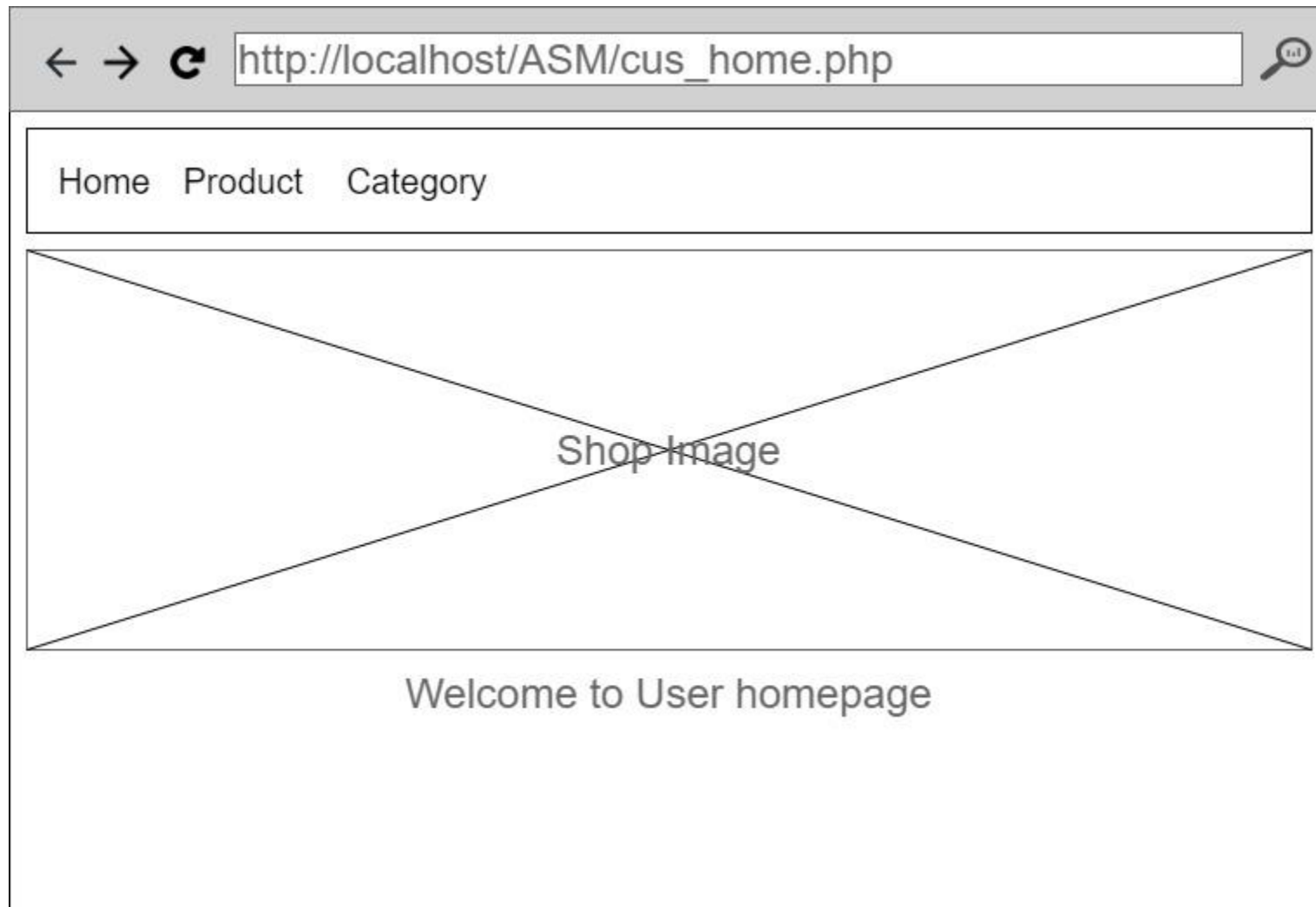


Figure 4: Customer Home Wireframe

When user enter the customer homepage, they will see 3 options on the top navigation bar and the shop image with the message that said "Welcome to User homepage".

1.5.2 Customer product webpage

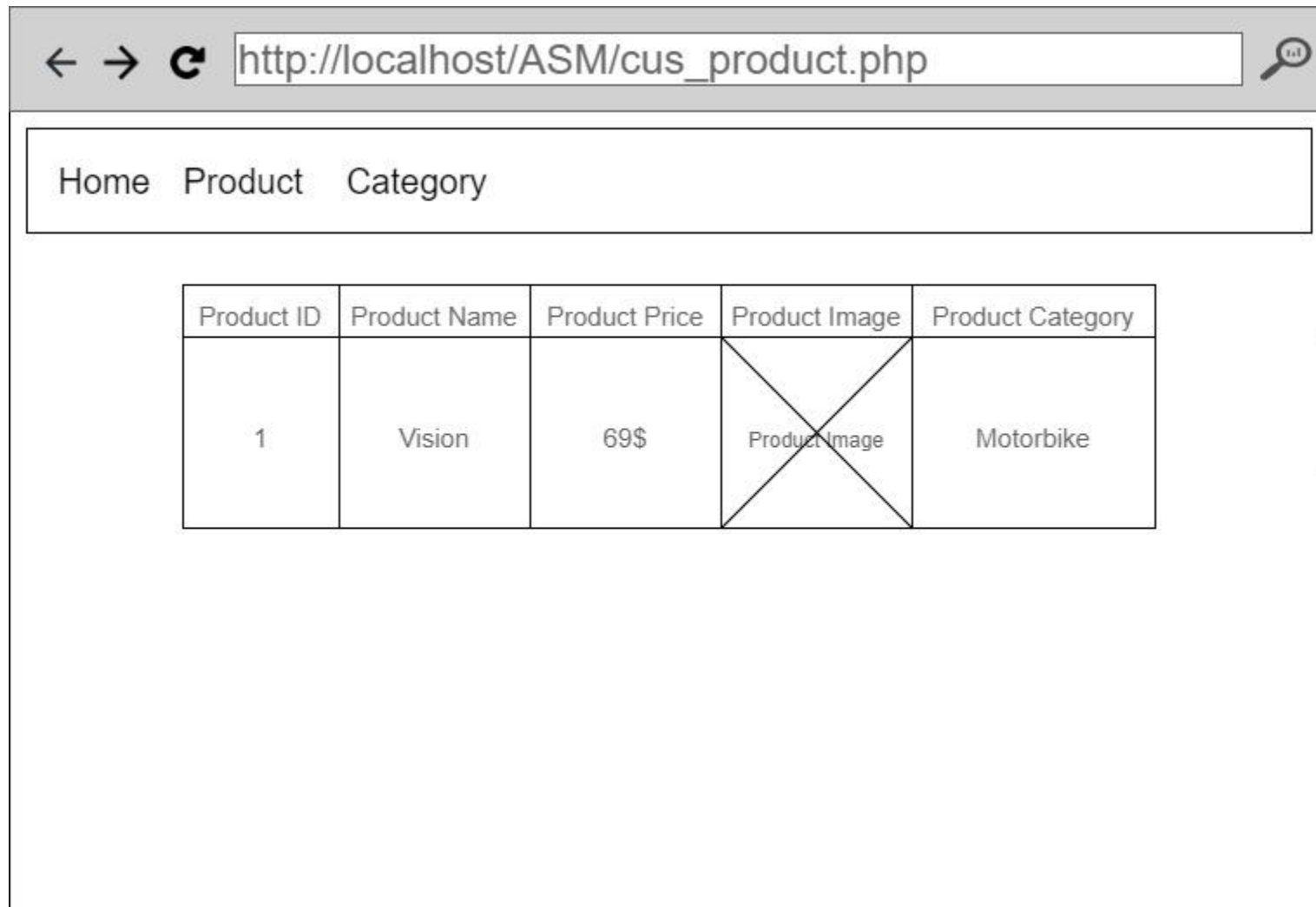


Figure 5: Customer product wireframe

The user will see all the products and its information in the table in the center of the page by choosing the “Product” option on the navigation bar above.

1.5.3 Customer product detail webpage



Figure 6: Product detail wireframe

If the user click on the product's image in the "Product" option, they will see up close the product detail including the product image, the product name and the product price.

1.5.4 Customer category webpage

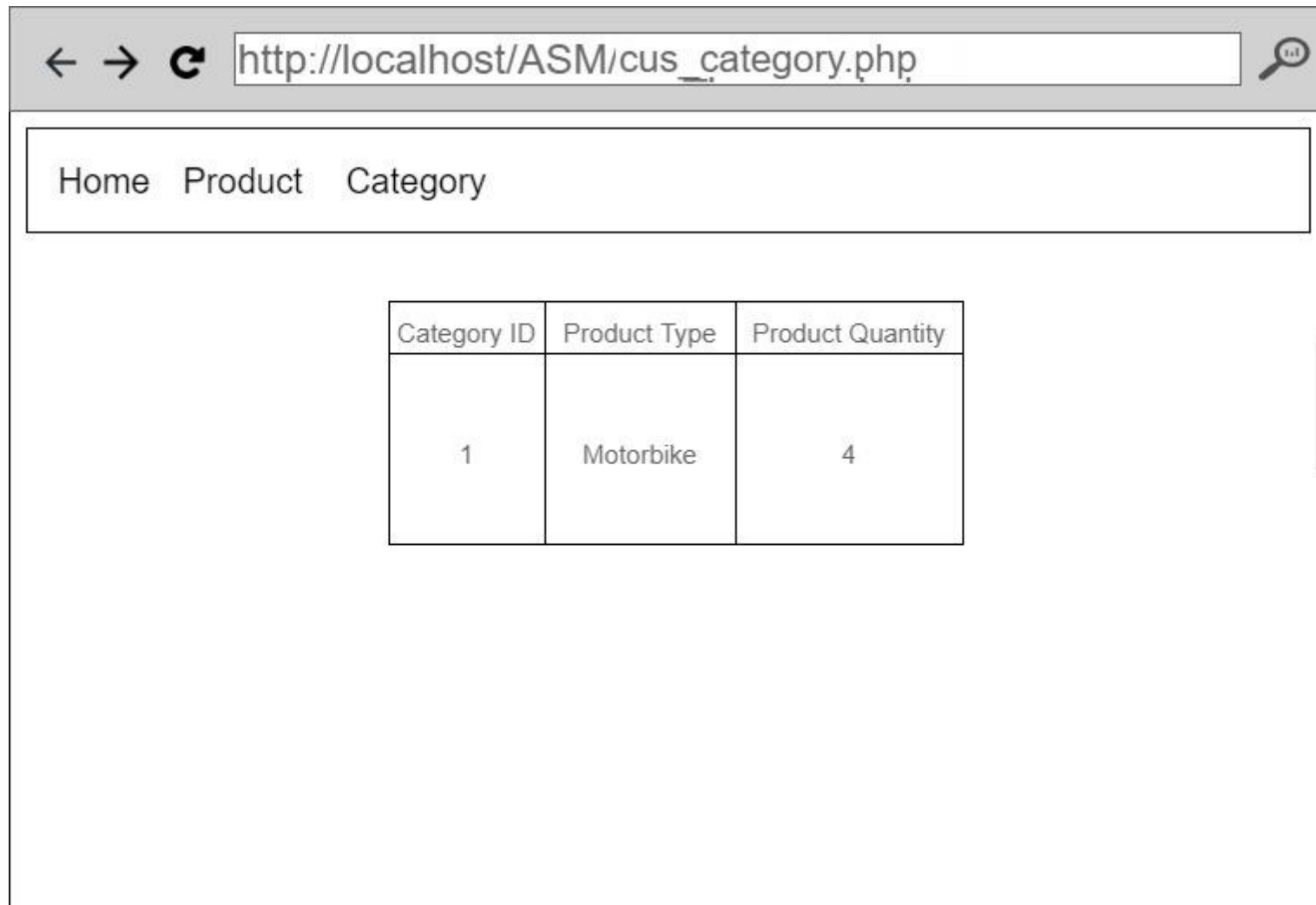



Figure 7: Customer category wireframe

The user will see all the category and its information in the table in the center of the page by choosing the “Category” option on the navigation bar above.

1.5.5 Admin login webpage



The image shows a wireframe of an admin login page within a web browser window. The browser's address bar at the top contains the URL `http://localhost/ASM/login.php`. The main content area of the browser displays a centered login form. The form has a title "Log in" at the top. Below the title, there are two input fields: one for "Username:" and one for "Password:". At the bottom of the form is a dark button labeled "Log in".

Figure 8: Admin login wireframe

To access this page, admin must use another link to log in to the admin's account because the "Log in" option will not be shown in the navigation bar in order to prevent the user from figuring out the username and password of the admin's account randomly.

1.5.6 Admin home Wireframe

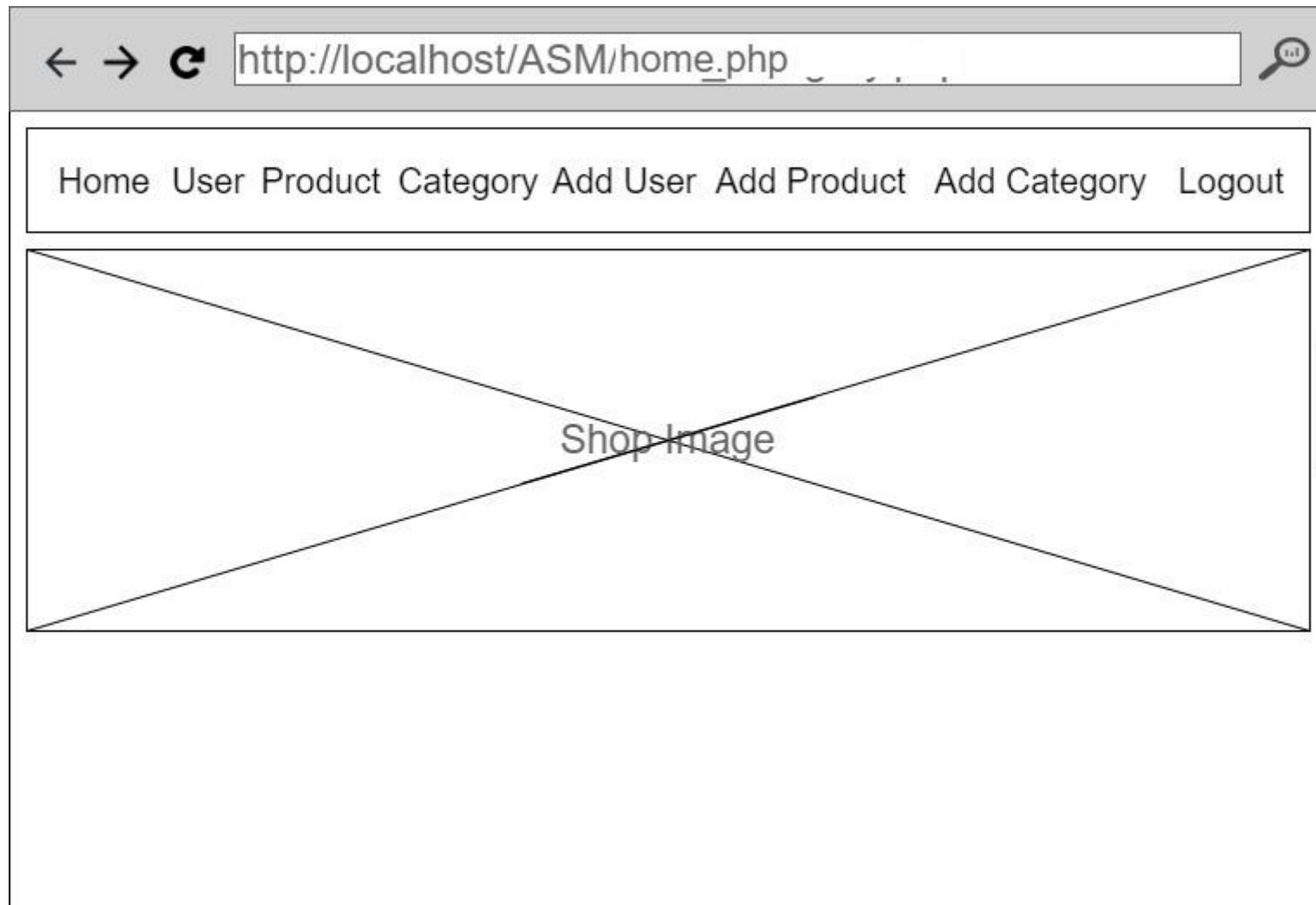


Figure 9: Admin Home wireframe

After logging in to the admin's account, the user will see that there are more options than the customer's homepage in the navigation bar. There are a total of 8 options for the admin to choose from, which are "Home", "User", "Product", "Category", "Add User", "Add Product", "Add Category" and "Logout" to log out of the admin's account.

1.5.7 Admin product webpage

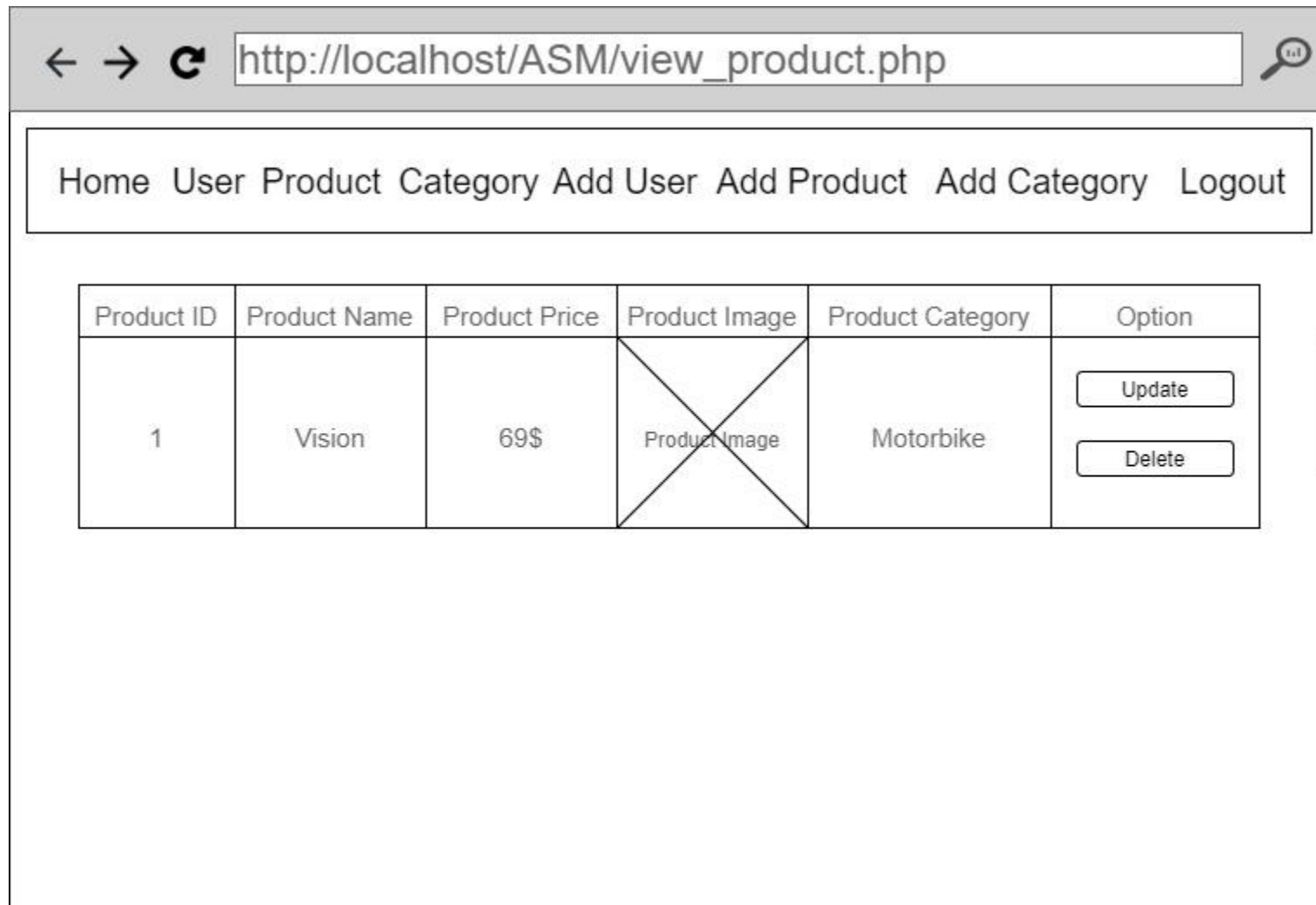


Figure 10: View product wireframe

The admin product webpage is almost the same as the customer product webpage. But in the admin webpage, there are 2 option that admin can choose from to update or delete the product by simply select “Update” button to update or “Delete” button to delete.

1.5.8 Update product webpage


← → ↻ http://localhost/ASM/update_product.php 🔍

Home User Product Category Add User Add Product Add Category Logout

Update Product

Name:

Price:

Image: 

Category:

Figure 11: Update product wireframe

If the user chooses “Update” option on a specific row, all the that product detail will appear on the screen including the current image for the user to edit. When a user clicks the “Delete” button, the localhost bar displays options for the user to validate delete or not.

1.5.9 Add product webpage

← → ↻ http://localhost/ASM/add_product.php 🔍

Home User Product Category Add User Add Product Add Category Logout

Add Product

Name:

Price:

Image:

Category:

Figure 12: Add Product wireframe

If the admin choose the “Add Product” option on the navigation bar, the Add Product webpage will pop up like the picture above. The page will allow user to input new Product Name, Product Price, Product Image, Product Category to add in the “Product” option by selecting the “Add” option when finish filling up the product information.

1.5.10 Admin Category webpage

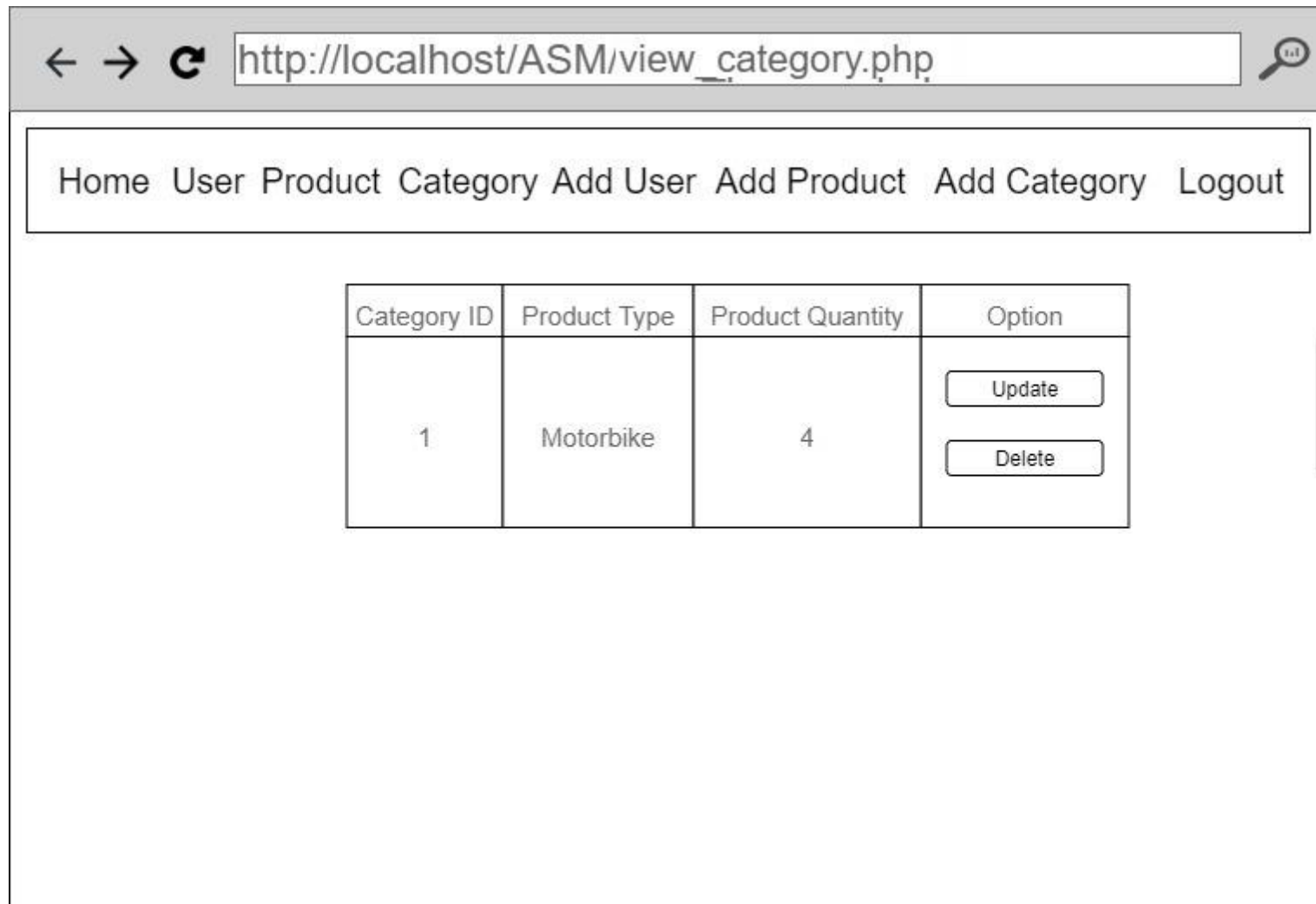


Figure 13: View category wireframe

By selecting the “Category” option on the navigation bar, Admin will see all category and its detail along with 2 options “Update” and “Delete”.

1.5.11 Update Category webpage

← → ↻ http://localhost/ASM/update_category.php 🔍

Home User Product Category Add User Add Product Add Category Logout

Update Category

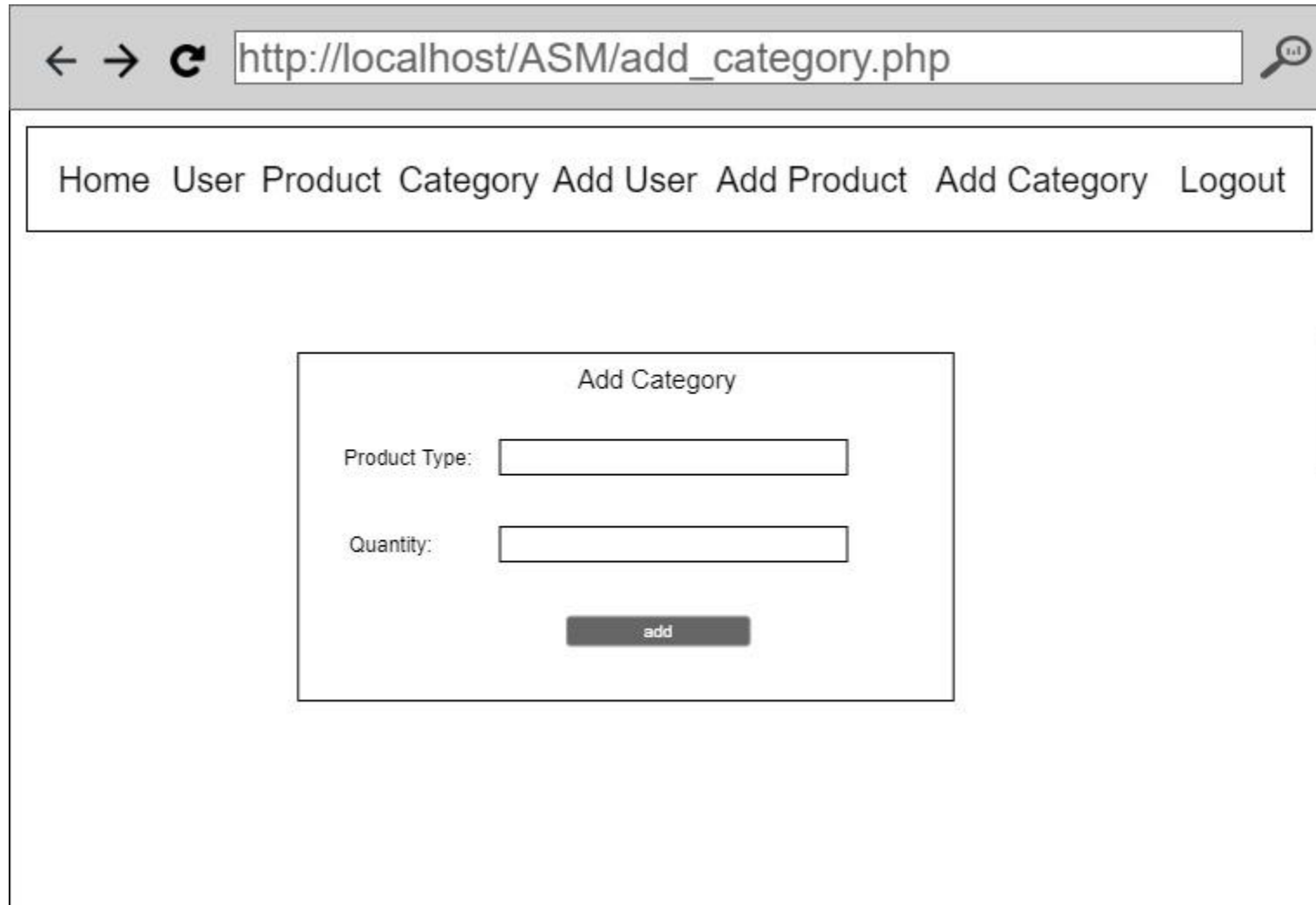
Product Type:

Quantity:

Figure 14: Update category wireframe

If the user chooses “Update” option on a specific row, all the that category detail will appear on the screen for the user to edit by pressing the “Update” button when finish editing. When a user clicks the “Delete” button, the localhost bar displays options for the user to validate delete or not.

1.5.12 Add Category webpage



The wireframe shows a web browser window with the address bar containing `http://localhost/ASM/add_category.php`. The browser has navigation buttons (back, forward, refresh) and a search icon. Below the address bar is a navigation menu with the following links: Home, User, Product, Category, Add User, Add Product, Add Category, and Logout. The main content area features a central box titled "Add Category". Inside this box, there are two input fields: "Product Type:" and "Quantity:". Below these fields is a dark grey button labeled "add".

← → ↻ `http://localhost/ASM/add_category.php` 🔍

Home User Product Category Add User Add Product Add Category Logout

Add Category

Product Type:

Quantity:

add

Figure 15: Add category wireframe

If the admin chooses the “Add Category” option on the navigation bar, the Add Product webpage will pop up like the picture above. The page will allow user to input new Product type, the quantity to add in the “Category” option by selecting the “Add” option when finish filling up the product information.

1.5.13 Admin User webpage

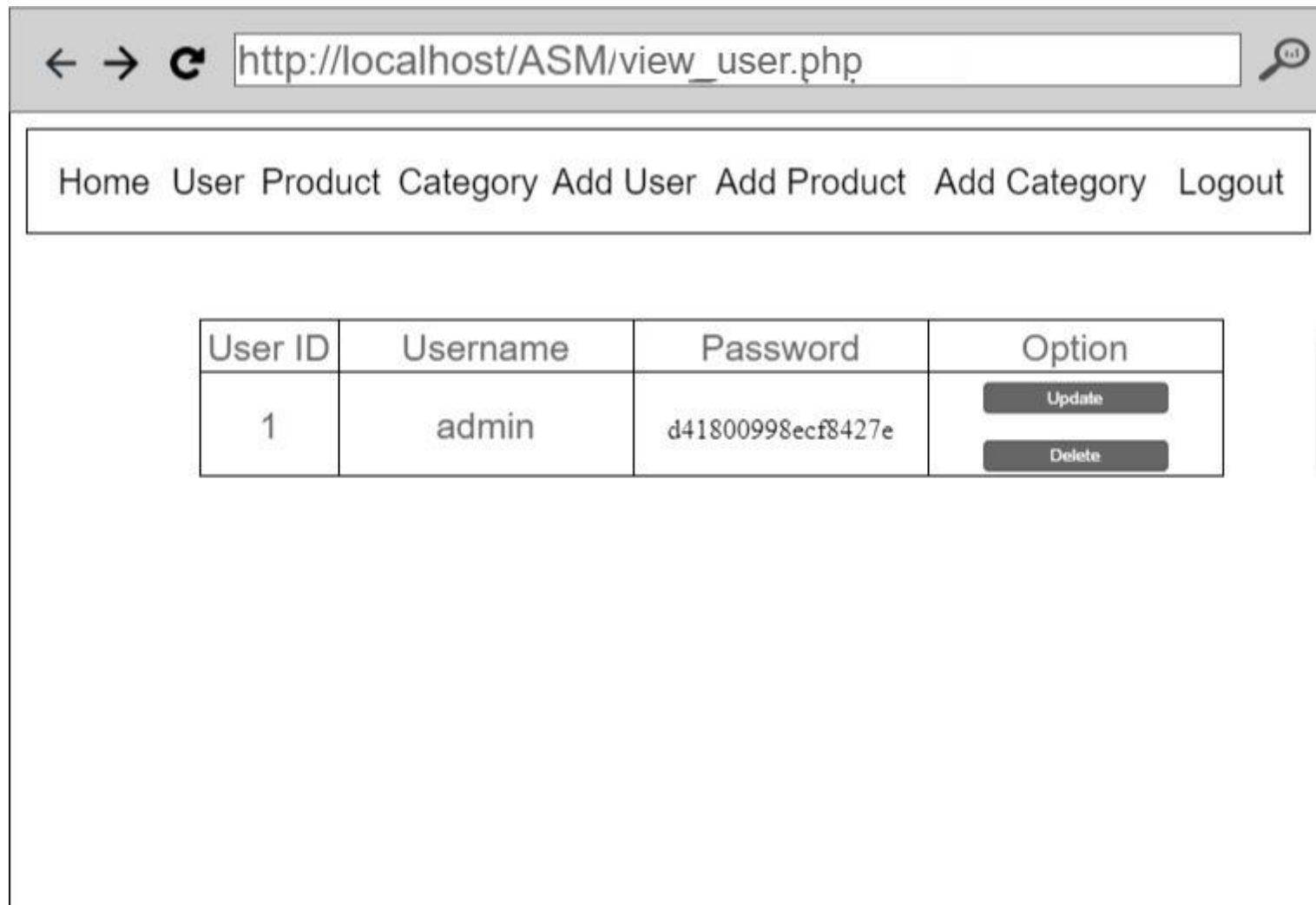


Figure 16: Admin user wireframe

If the user choose the “User” option, all the account will be display including User ID, Username, Password. In the admin user webpage, there are 2 option that admin can choose from to update or delete the product by simply select “Update” button to update or “Delete” button to delete.

1.5.14 Add user webpage

The image shows a wireframe of a web browser window. The address bar contains the URL `http://localhost/ASM/add_user.php`. The browser's navigation buttons (back, forward, refresh) are visible on the left. The page has a horizontal navigation menu with the following links: Home, User, Product, Category, Add User, Add Product, Add Category, and Logout. The 'Add User' link is currently selected. The main content area features a form titled 'Add User'. This form contains three input fields: 'username:', 'password', and 'Confirm Password:'. Below these fields is a dark grey button labeled 'add'.

Figure 17: Add user wireframe

If the user chooses to add a user, the website will redirect to the Add user tab, where the administrator will fill in the necessary information. After clicking the ADD button, the localhost will display a message indicating whether or not the user has been entered. If your feedback is effective, the website will redirect you to the user list tab.

II. Web implementation

1. Web principles, standards and guidelines

Web principle: Website design encompasses not only how a website looks and behaves, but also how it functions and reacts. As web designers work on a website, they aim to make it highly sensitive, functional, simple, and useful in addition to those elements that add visual appeal. Designers adopt those standards in order to produce a highly accessible and functional website. These principles serve as guidelines or standard points to remember. The below are some of the main concepts of good web design (Luenendonk, 2020):

- **Highly intuitive:** The first rule or theory of website usability states that a web page's configuration should be extremely intuitive and easy to follow so that users do not have to worry about which way to go. It must be self-explanatory in a straightforward manner. Keep the navigation intuitive and quick, and don't let any question marks or questions pop up. This improves the website's accessibility while still making it even more entertaining. Visitors should not have to wonder how to get from point A to point B because the structure is free of cognitive load.
- **Visual hierarchy:** A visual hierarchy is the next concept that leads to the creation of a successful and efficient website. The order or pattern in which our body travels and perceives the objects it sees is referred to as visual hierarchy. When it comes to a web page, the visual hierarchy is the order in which our eyes travel from one topic/content/block to the next. When creating a web page, a planner must first determine the relevance of the different topics and then arrange them so that users see the most interesting first and then move on to the others in a hierarchical order. There are two methods for creating a visual hierarchy, which are as follows:
 - **Size hierarchy:** Size hierarchy, as the term implies, is a hierarchy in which the most important content or image on a website is shown in the largest size, followed by the second most important content or image in the second largest size, and so on. The size differences should be so that a tourist can see the objects in order of priority, and the pecking order of things should be evident.
 - **Content hierarchy:** Aside from using a hierarchy of height, which is one of the most effective ways to construct an order of priority, you can also use a hierarchy of content to implement this concept. In a hierarchical order, you should arrange content such that the human mind first goes to the most important content, such as the business's mission or meaning, and then proceeds to the less important information blocks.
- **Accessibility:** The accessibility of a web page or website is another crucial concept that must not be overlooked when creating it. When a visitor comes to the website, he or she should be able to quickly access all of the content. This ensures that the

text must be readable, the colors should not be too bright, and the backdrop should not overwhelm the material, among other things. You should take any of the steps below to make the website more available to everyone:

- **Typefaces:** Make sure you choose a font style and size that is understandable for anyone and isn't too complicated for anyone to understand. Fonts such as Verdana, Times New Roman, Arial, and others are basic fonts that almost everyone can read online. Similarly, the optimal font size is 16 px, but you can be a bit more creative about it.
- **Color:** In terms of the user interface, the colour scheme and contrast should be well-considered and capable of creating visual unity and balance. To make it easier to read, it's best to use contrasting colors for the backdrop and written text. To make the effect easier on the brain, use a darker text color and a brighter background hue. Extra-bright shades should be used with caution.
- **Images:** It is a smart idea to choose and position the appropriate images on your web pages in order to properly engage with your target audiences. Make sure the pictures are of good quality and appropriate for your needs.
- **Hick's law:** According to Hick's rule, "any additional option lengthens the time taken to make a decision." This rule applies not only to web design but to a variety of other contexts and settings. For example, if you go to a restaurant and are given a large number of food options to choose from, you would take longer to make a decision. When it comes to web design, the more choices you offer your users, the more difficult it would be to use and navigate your site. This suggests that in order to have a decent user interface, we must reduce the amount of options available. To assist higher revenue and improved overall profit, distracting alternatives must be removed.
- **Fitt's law:** Fitt's rule is another law that serves as a major standard of web design. The time it takes to get to a target is determined by the height of the target as well as the distance between them, according to this rule. This assumes that the bigger the object or target and the smaller it is in space, the easier it would be to approach or meet it. This rule can be easily integrated into web design, and it can significantly improve the web design. This does not imply that the larger the target, the greater, but rather that the usefulness factor of a target follows a curve rather than a straight line. Users will be more inspired and enthused if you adapt this law to your web design.
- **Communication and content:** Everyone who comes to your website is searching for some kind of information or material, so it's critical that you engage with them directly and in an entertaining way. Your data should be persuasive, easy to read, and understand. Communication encompasses not just the delivery of written content, but also the use of photographs, infographics, and other forms of media such as videos and audio recordings. The idea of 'Visible text' is taken into account in web design. Visible language, according to this definition, is the text that people display on screens and is based on three basic concepts. The following is a list of them:
 - **Organize:** The first theory of recognizable language or efficient communication is to have a simple and consistent interface or form to your website's users. The architecture, which involves the separation of text, continuity,

- navigation, and visual appeal, are all essential aspects of concept organisation. You must organise and arrange the blocks of information in an easy-to-process manner in order for your website to interact effectively.
- **Economize:** This idea entails getting more done for fewer. This suggests that you can strive to convey as much detail as possible by using less visual elements. The template or layout must be clutter-free, easy to understand, and have a high level of consistency. It needs to emphasize what's important while still distinguishing various components.
 - **Communicate:** The web design's user interface must allow users to interpret it in the most simple way possible. Readability, legibility, colour, texture, and views must all be balanced.
 - **White space and simple design:** A basic design is a design that works. A traveler does not want to see complexity on your website, and white space is one of the most critical features of a simplistic design. White space is all that web designers can embrace. White space not only reduces the cognitive burden on the page, but it also makes it easy for users to understand the information presented. White space aids in the division of a web page into many distinct sections or regions, making information processing easier. When dealing with dynamic hierarchical systems, using a whitespace solution is often preferable. Other elements that can be seen as part of a basic template are the following:
 - **Grid-based layout:** To prevent a cluttered website structure or presentation, use a grid-based interface of content separated into columns, boxes, and various pages.
 - **F-pattern design:** The human eye scans screens in a 'F' pattern, which is a reality. As a result, it is a smart idea to design a web page or website in such a manner that it complements the visitors' normal reading behaviour.
 - **Conventional designs:** Traditional designs aren't necessarily repetitive, and they can really perform well in terms of guest response and likeability. They instill a sense of confidence, dependability, and brand reputation.
 - **Regular testing:** Another web design philosophy that both programmers and website owners would remember is TETO (Test Early and Test Often). Conducting usability checks on a regular basis can yield valuable information and insights into a variety of issues and complexities relating to a website's architecture or design elements. What happens is that websites often run into problems, and by failing to test them frequently, they can generate issues that drive tourists away. Upgrades and enhancements are needed on a regular basis for websites to sustain user traffic and customer interest.

Web standards: The formal, non-proprietary standards and other technical specifications that specify and represent features of the World Wide Web are referred to as Web standards. In recent years, the word has been more closely aligned with a pattern of endorsing a series of structured best practices for developing web pages, as well as a web design and development philosophy that incorporates such practices. Web standards were created to safeguard the web environment, ensuring that it remains available, secure, and usable to everyone. enclosing the internet in a security bubble and disbanding the idea of having to create websites for individual browsers

(Dickens, 2019).

According to Dickens (2019), Web standards are a set of interconnected requirements and requirements that regulate various facets of the Internet, not just the World Wide Web. Even if they aren't specifically related to the internet, those specifications have an effect on the creation and management of web pages and web services. Interoperability, reliability, and compatibility of web pages and websites are also important factors to consider.

Web standards consist of the following (Dickens, 2019):

- HTML/XHTML, Cascading Style Sheets (CSS), file formats like Portable Network Graphics (PNG) and Scalable Vector Graphics (SVG), and connectivity innovations like WAI-ARIA are all recommended by the World Wide Web Consortium (W3C).
- The Web Hypertext Application Technology Working Group (WHATWG) publishes protocols and "living standards" such as the HTML Living Standard, DOM Standard, Encoding Standard, and URL Standard.
- Ecma International (formerly ECMA) publishes standards such as JavaScript (also known as ECMAScript) and JavaScript Object Notation (JSON).
- JPEG, for example, is a standard issued by the International Organization for Standardization (ISO).

We will see the importance of providing rules for the World Wide Web because we understand the history of standards and when they were adopted. So why is it so important that we keep contributing to Web Standards. Here are a few examples (Dickens, 2019):

- **Keeping the web free and accessible to all:** Browser developers would make choices about what should and shouldn't be aspects on the internet if the Web Standards group didn't exist. This could result in the internet being a monopolized asset, with only the biggest players having a voice in what happens next.
- **Helping make source code simpler, reducing development and maintenance time:** It became more difficult to produce content that would be served in the same way through different browsers as more browsers were released and browser developers started to diversify their approaches. This increased the amount of work needed to create a completely compatible website, including bloating web page source code. Today, we also have to add [X script] so that this runs on [X web browser], but it would be a lot different if Web Standards weren't in place.
- **Allowing for backward compatibility and validation:** Web guidelines have laid the groundwork for new websites to work with existing browser versions if they adhere to them. This concept of backward compatibility is critical for maintaining online

accessibility. It doesn't guarantee that older browsers can view the content precisely the way you want it to, but it helps ensure that the form of the web document is recognized and presented appropriately.

- **Helping maintain better SEO (Search Engine Optimization):** Another big secret advantage (at the time when Web Standards were first introduced) was that a Web Standards compliant website was more search engine discoverable. When Google became a big player in the search engine community in the early 2000s, this became much more apparent.
- **Creating a pool of common knowledge:** Web standards provide an environment where there are a set of rules that any developer can obey, learn, and become familiar with. In principle, this assumes that one developer could create a standard-compliant website, while another developer could easily start up where the first left off. Standards, in fact, provide the basis for this, but the concept is highly reliant on developers writing well-documented code.

Web guidelines:

- **Design process and evaluation:** The report concludes that you can only provide valuable material on your website when it comes to the design process and assessment. Some excessive content on your page can cause it to perform poorly when users visit it. You should also try collecting information between exchanges with real customers during the design process to get a better understanding of what the users want. Pay attention to what they have to say. You'll be able to support their wishes if you meet their standards. Knowing what they expect when they visit a website is half the fight. Participating in this phase with them can be a very satisfying experience for both you and the customer. You can also set usability targets and outline how you want to achieve them. According to research, websites that do not appear in the top 30 results when keywords are searched are rarely viewed. Instead of relying on the guidance of a single developer, you should try partnering with a group of them. This way, you can look at your website design from a variety of perspectives in order to appeal to a wide range of people.
- **Optimizing the user experience:** When it comes to improving customer experience, there is a lot to consider. Pop-ups and other unwelcome windows are one feature of a website that can be aggravating to consumers. Making the website look professional will also help the users like it. They would most likely quit if they do not believe the website is trustworthy. Maintaining a coherent workflow series is a perfect way to improve customer experience. It assists consumers in comprehending and becoming more familiar with specific moves. Make sure the customers are aware of how long they would have to wait by displaying their wait time whether it is more than a minute or two. If a person is forced to wait for an extended period of time, they may get frustrated. Other approaches to improve the user interface include making crucial information pages easy to print, including commonly asked questions in a prominent location, and being very descriptive about words that may come easily to you but may annoy people. As all of the customers can be young or unfamiliar, make sure to provide them with assistance on how to navigate your website.

- **Accessibility:** Make your website universally open. Websites must be open to anyone, including people with disabilities, according to Section 504 of the Rehabilitation Act. You will do this by incorporating assistive devices into the website. Make sure that all photographs, videos, diagrams, and other media have text. You will do this by creating a separate website for people with disabilities that is text-only. For anyone who suffer from epilepsy, be mindful that the website does not have screen flicker. Be sure the page's content can be viewed without the use of style sheets, which can cause issues with assistive technologies. Mark the various sections of the website to make it easy to navigate. Ensure the plugins and applets are accessible to all users by checking them. As a web designer, it's your responsibility to make sure that anyone, regardless of skill or inability, can quickly navigate through your website. This is necessary not only to attract clients, but also to ensure that you are in line with Section 504.
- **Hardware and Software:** The hardware and software of your website are the subject of the fourth significant research-based recommendation. There are several browsers available, and website designers can strive to make their sites compliant with the most common ones, such as Internet Explorer, Google Chrome, Firefox, and Safari. Learn what users want in terms of fonts, text size, and other browser options. If you can, aim to make at least 95% of your users happy while building your website. Not only should the website be compliant with browsers, but you should also think of the most popular operating systems that your customers will use. According to research, Windows XP is the most common operating system, followed by Windows 2000, Windows 98, and Mac. You should also consider usage levels when designing the website. Make sure your customers don't have to wait too long for your website to load fully. The specification of your monitor's screen resolution is also essential for usability. Consider the size of 1024x768 pixels, this is what the majority of citizens use.
- **The home page:** One of the most critical aspects of the website would be the home page. This is the first page the guests see. They could go somewhere else if they don't like what they see. Make sure your homepage is easy to find—include all of the most relevant links, and encourage visitors to stay and learn more about your company. Many users would just look at the homepage before deciding whether or not they want to dig further into the content. You should also think about what kind of content you want to bring on the homepage and how much of it you want to put there. When a person first sees a lot of text, they will get confused. It's also necessary to avoid cramming too many photos into the page, as this can be distracting. The aim or motto of your website or organization is one of the most important items to have on your homepage. If you're going to make some updates to your website that people have become used to, make sure to announce it on your homepage so they know what's going on.
- **Page layout:** There are some relevant rules to consider when it comes to the layout of the website's pages. Remember one of the first rules: just include what is absolutely required on your website. This suggests that you can avoid clogging up your pages with unnecessary data or photos. According to the report, a website with less clutter is easier to browse than one with a lot of material. Important information that you want the customers to see right away should be placed in the page's top middle. The

comparative data you show should be shown next to each other to make it easier to see. Allow your users to compare details without having to go back and forth between sites. Use a scale that ranges from extreme to low in value. The most valuable information should be at the top, and the least important information should be at the bottom. By aligning all of the things on a list, you can make it look more appealing. Additionally, use acceptable page and line lengths, avoid using scroll stoppers, and limit the amount of white space you use.

- **Navigation:** The back button is one of the most often used buttons on a website. Users like exploring, but they still want to ensure that the material they've used before is easily accessible. Check to see how the back button works. In reality, use the various buttons and links on your website on a regular basis. Make an effort not to perplex the customer. Make sure the tab labels are clear enough that the reader knows where they are and where they are going. One recommendation to ensure that consumers can get support if they need it is to use sitemaps. Using "glosses" is another simple way to ensure that the customers know where they're headed. These are the pop-ups that appear when you hover your mouse over a connection. If you click, it will tell you where you will land. Tabs should be clearly labeled and placed at the top of the list. If a user needs to visit another website, it would be simple to do so. Even, keep in mind that literature indicates that breadcrumb navigation is rarely used.
- **Scrolling and paging:** This can be a make-or-break attribute on a website for users when it comes to browsing and paging rules. Horizontal scrolling is one thing that study has shown can be avoided in web design. This prevents people from seeing any of the details they need on a single tab. When it comes to the website, you have complete power over the scroll. You can speed up navigation by highlighting or expanding the size of important things on your pages. Slower scrolling is recommended because there is a lot of text on the website so that users can appreciate what they are reading. You should make your scrollbar work quicker with pictures or tiny volumes of text. You should think about using paging to improve the accessibility of your website. Instead of searching to find content, they may use links to navigate to other sites that include the critical information they need. If you need a lot more detail, make more pages that are shorter to minimize the need for scrolling.
- **Headings, titles and labels:** Consider using informative headings, names, and logos to aid the visitors in navigating your web. While you don't want to use a lot of terms in your headings, names, or logos, the ones you do use should be significant. It's also crucial that they accurately represent the detail your reader would encounter while visiting the page on your website. Making the headings or logos stand out would also pique users' interest in what they'll find. This would make them want to press on them more often. When it comes to using headings on your website, think about using the proper hierarchy. This implies that you can use acceptable heading tags wherever possible. It would also improve your productivity if you have simple choices for your users to select from. If you can visually highlight relevant headings, readers will be able to locate the details they're looking for. Your users may become dissatisfied with your website and leave if you do not follow these required guidelines for headings, names, and labels.

- **Links:** These critical rules should be followed when it comes to the accessibility of links on the website. Instead of using the phrase "click here" with all of your page's links, get creative. Make it clear to the customer what they will be learning as they press. Also, double-check that the links you're using are important to the content on the website. Be sure the connections work before you use them. It can be aggravating for a customer to be searching for details but the connection does not work. If necessary, repeat essential connections, but don't go overboard. Image connections are less helpful than text links, so don't use too many of them. Be clear with letting users know they can press in any way you want. When users are unsure of what to expect or where to go for more details, it can be frustrating. Instead of using the mouse over method of scrolling, try pointing and clicking. Labeling internal and external connections, using acceptable connection lengths, and letting the user know if there are sections of a picture they can click on for a link are all good criteria.
- **Text appearance:** Text is an essential component of the website, and there are certain rules to follow when it comes to the forms, sizes, and placement of texts. When designing a website, it's necessary to keep font styling consistent in general. When selecting background and text colors, bear in mind the greater contrast between the two makes it easier to decipher. In the study that was done, using black text on a plain backdrop made it easier for consumers to understand. Text objects such as hours, dates, phone numbers, and addresses should all be formatted consistently. This will make it easier for your reader to comprehend and notice the details. Remember to capitalize appropriately. If you use CAPS Lock so much, it may seem as if you are swearing at your guests. When it comes to bold and italic font, use them only when necessary. Its aim is to draw attention to essential facts, but if you use it too often, it loses its usefulness. Use such features sparingly when you want to draw users' interest. Using 12 point or greater when it comes to scale.
- **Lists:** You might want to provide short details in the form of lists to your customers. Lists are an excellent way to teach them what they need to do without overwhelming them with email. Lists are visually appealing, and the reader isn't as distracted as they may be when reading several paragraphs of detail. Ascertain that your reader understands the purpose of the list. Use a simple heading or title so they can find what they're looking for quickly. The aim of lists is for the reader to search them easily to find answers to any questions they might have. Place the most interesting things at the top of the list so that if readers get tired of reading, they can still get the most important information. When making a list, never use 0 as a total. This could invalidate the data. On each line, capitalize the first letter of the first word. If you usually capitalize proper nouns in the rest of the text, this is the only exception.
- **Screen-Based controls (widgets):** Fields for entering content, drop down boxes, push buttons, and icons are some of the most popular widgets or screen-based controls that your users can encounter on websites. When it comes to using widgets for compatibility, there are a lot of rules to follow. Make sure widgets are clearly named so people understand what they're for. Widgets can be used sparingly. Although they are useful tools for assisting users in navigating your website, you must be careful not to overuse them. Users can become overwhelmed and leave your site as a result of this. Reduce the amount of data that users

must join. If users have to insert the same data over and over, they can get irritated. So that visitors know what to do, use the same data entry form on the platform. For ease of use, remember to immediately position the cursor in the first data field. To assist people, use at least two radio keys. Auto tabbing is common with users because it allows them to jump from one box to the next without having to switch between the mouse and keyboard.

- **Graphics, images and multimedia:** You should also have continuity in mind when it comes to graphics, photographs, and media guidelines. These features are beneficial to have on your website, but you don't want to annoy your users with them. Graphics that are appropriate for your site will greatly improve its usability. On the website, have the picture logo so that the user will recognize it. Remember that multiple photos can take longer to load, so don't overburden your page and make your user bored as they wait. When doing usability research, one thing to check is whether graphics, photographs, and multimedia are conveying the correct message to consumers. If not, they aren't needed. Give each picture a descriptive mark to help you remember what it's about. Keep photos that don't fill the whole backdrop to a minimum. This makes it impossible to see important text. Any icons you use do not look too much like banner advertising, since people are more likely to miss them, according to analysis. These features are appropriate for your website, please just use them when absolutely necessary.
- **Writing web content:** One of the most critical facets of making a website is the content. The material contains all of the information you determine is necessary to provide the customers with the information they need. With too much or too little content, users may become bored or irritated, so make sure everything on your website has a reason and does not overload. It's important that you choose terms and expressions that your guests are likely to recognize. Keep in mind how you're trying to meet, and use language they'll understand. No matter who you're writing with, avoid using words with a large vocabulary so people will get bored or frustrated. If you do use expert terms related to your website's subject, make sure to describe them for your visitors. Define any terms you believe your guests may need to comprehend in order to grasp your message. Be sure to write in the active voice and in the affirmative while giving orders.
- **Content organization:** To ensure a positive user experience and the accessibility of your website, organization is essential. How simple it is for your users to locate crucial details can depend on how you organise your content. Your visitors will be able to browse content and just concentrate on what they need if you build your website with simple headings and explanations. Make sure only the detail you need is visible. Get rid of something that isn't really necessary. You don't have to say the same thing on every page, either. It is appropriate to repeat very valuable facts once or twice, but no more. Put the most critical information at the top of the list, in the middle. When the page opens, this will be one of the first places the users look. It's important to present it in a number of ways to guarantee that it reaches a wide range of people. Although some people prefer to read content, others may prefer to learn about your website through graphics or audio files.
- **Search:** The opportunity to browse for information throughout the website is a very useful tool that people enjoy. There are some instructions used in this. It's important to note to have a search bar on each page on the website by using a search toolbar.

This would make it much easier for them to discover what they're searching for. The more specific you can make your quest, the better. You can also configure each search tool and look for keywords through the entire platform. Users can not realize how capitalizing search words impacts the information returned while browsing, so be cautious. Make sure you can find both variations of a word or sentence. You should work out what kind of vocabulary your guests may use and build your searches around those terms. You should provide tips to assist people in finding what they need to make the quest process simpler. To make the procedure easier, templates can be used. This feature on your website will assist users in quickly locating the information they need.

- **Usability testing:** Testing the website's usability is critical to maintaining its efficiency and performance. There are several different types of usability tests you can run, but it's important that you choose the one that can provide you with the most useful information on how your website is doing. Using actual subjects that have no connection to the business is one of the easiest ways to get research done. You'll want to hear from people who aren't afraid to tell it as it is. Obtaining both qualitative and quantitative data, rather than just one kind of data, may be useful. Often, keep in mind that once you've done your trial, it's a good idea to make the necessary corrections and then retest your website. The more you repeat this procedure, the more successful your website can become. As part of your data, you can also consider getting user feedback. This will provide you with a wealth of knowledge. Using programmed assessment techniques on your website and reviewing the results on a regular basis will also be beneficial.

2. In my website

- a. The home page: One of the most critical aspects of the website would be the home page. This is the first page the guests see. They could go somewhere else if they don't like what they see. Make sure your homepage is easy to find—include all of the most relevant links, and encourage visitors to stay and learn more about your company. Many users would just look at the homepage before deciding whether or not they want to dig further into the content. You should also think about what kind of content you want to bring on the homepage and how much of it you want to put there. When a person first sees a lot of text, they will get confused. It's also necessary to avoid cramming too many photos into the page, as this can be distracting. The aim or motto of your website or organization is one of the most important items to have on your homepage. If you're going to make some updates to your website that people have become used to, make sure to announce it on your homepage so they know what's going on.



Welcome to User homepage

Figure 18: Customer homepage

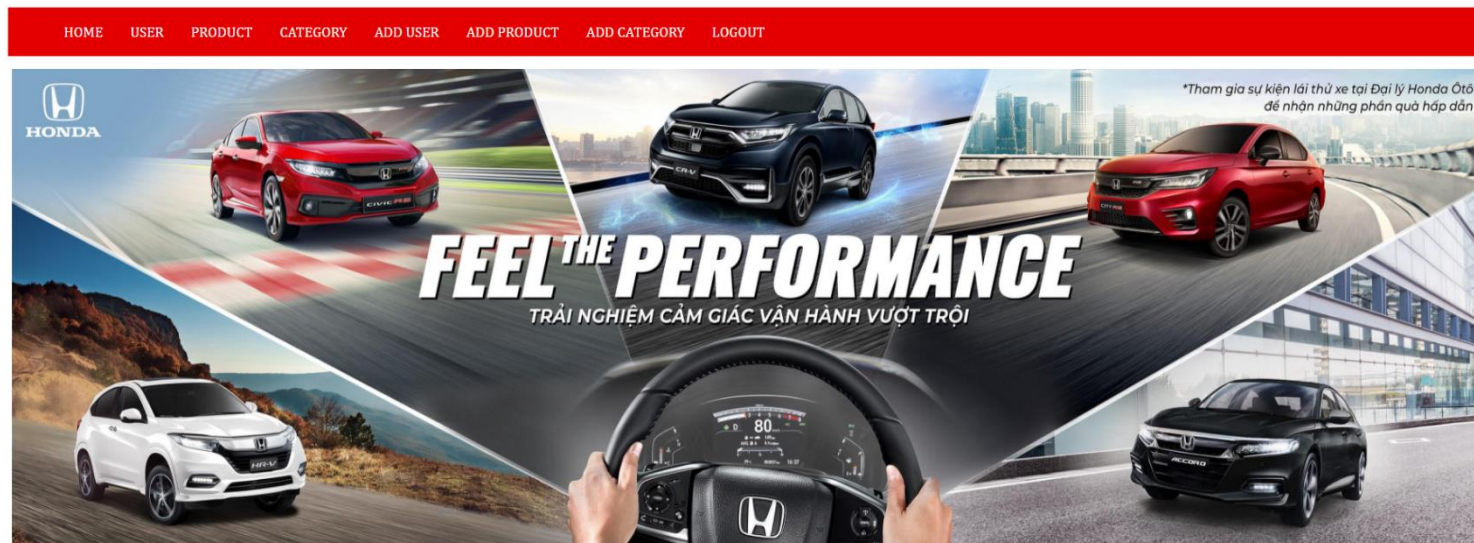


Figure 19: Admin homepage

- b. Graphics, images and multimedia: You should also have continuity in mind when it comes to graphics, photographs, and media guidelines. These features are beneficial to have on your website, but you don't want to annoy your users with them. Graphics that are appropriate for your site will greatly improve its usability. On the website, have the picture logo so that the user will recognize it. Remember that multiple photos can take longer to load, so don't overburden your page and make your user bored as they wait. When doing usability research, one thing to check is whether graphics, photographs, and multimedia are conveying the correct message to consumers. If not, they aren't needed. Give each picture a descriptive mark to help you remember what it's about. Keep photos that don't fill the whole backdrop to a minimum. This makes it impossible to see important text. Any icons you use do not look too much like banner advertising, since people are more likely to miss them, according to analysis. These features are appropriate for your website, please just use them when absolutely necessary.



Figure 20: Home Image



Figure 21: Product Image

- c. **Lists:** You might want to provide short details in the form of lists to your customers. Lists are an excellent way to teach them what they need to do without overwhelming them with email. Lists are visually appealing, and the reader isn't as distracted as they may be when reading several paragraphs of detail. Ascertain that your reader understands the purpose of the list. Use a simple heading or title so they can find what they're looking for quickly. The aim of lists is for the reader to search them easily to find answers to any questions they might have. Place the most interesting things at the top of the list so that if readers get tired of reading, they can still get the most important information. When making a list, never use 0 as a total. This could invalidate the data. On each line, capitalize the first letter of the first word. If you usually capitalize proper nouns in the rest of the text, this is the only exception.







Product ID	Product Name	Product Price	Product Image	Product Category
1	Honda Vision	30.000.000\$		motorbike
2	Honda Airblade	45.000.000\$		motorbike
3	Honda SH	100.000.000\$		motorbike
4	Mercedes C300	1.969.000.000\$		car
5	Vinfast A2.0	1.000.000.000\$		car
6	BMX GT1	2.100.000\$		bicycle

Figure 22: Customer Product List





Product ID	Product Name	Product Price	Product Image	Product Category	Options
1	Honda Vision	30.000.000\$		motorbike	<div>UPDATE</div> <div>DELETE</div>
2	Honda Airblade	45.000.000\$		motorbike	<div>UPDATE</div> <div>DELETE</div>
3	Honda SH	100.000.000\$		motorbike	<div>UPDATE</div> <div>DELETE</div>
4	Mercedes C300	1.969.000.000\$		car	<div>UPDATE</div> <div>DELETE</div>
5	Vinfast A2.0	1.000.000.000\$		car	<div>UPDATE</div> <div>DELETE</div>
6	BMX GT1	2.100.000\$		bicycle	<div>UPDATE</div> <div>DELETE</div>

Figure 23: Admin Product List

User ID	Username	Password	Options
1	admin	d41d8cd98f00b204e9800998ecf8427e	<input type="button" value="UPDATE"/> <input type="button" value="DELETE"/>
2	admins	e10adc3949ba59abbe56e057f20f883e	<input type="button" value="UPDATE"/> <input type="button" value="DELETE"/>
3	admin	202cb962ac59075b964b07152d234b70	<input type="button" value="UPDATE"/> <input type="button" value="DELETE"/>

Figure 24: Admin User List

Category ID	Product Type	Product Quantity	Options
1	car	3	<input type="button" value="UPDATE"/> <input type="button" value="DELETE"/>
2	motorbike	3	<input type="button" value="UPDATE"/> <input type="button" value="DELETE"/>
3	bicycle	1	<input type="button" value="UPDATE"/> <input type="button" value="DELETE"/>
4	Tank	0	<input type="button" value="UPDATE"/> <input type="button" value="DELETE"/>

Figure 25: Admin Category List

Category ID	Product Type	Product Quantity
1	car	3
2	motorbike	3
3	bicycle	1
4	Tank	0

Figure 26: Customer Category List

- d. **Navigation:** The back button is one of the most often used buttons on a website. Users like exploring, but they still want to ensure that the material they've used before is easily accessible. Check to see how the back button works. In reality, use the various buttons and links on your website on a regular basis. Make an effort not to perplex the customer. Make sure the tab labels are clear enough that the reader knows where they are and where they are going. One recommendation to ensure that consumers can

get support if they need it is to use sitemaps. Using "glosses" is another simple way to ensure that the customers know where they're headed. These are the pop-ups that appear when you hover your mouse over a connection. If you click, it will tell you where you will land. Tabs should be clearly labeled and placed at the top of the list. If a user needs to visit another website, it would be simple to do so. Even, keep in mind that literature indicates that breadcrumb navigation is rarely used.

HOME PRODUCT CATEGORY

Figure 27: Customer Navigation Bar

HOME USER PRODUCT CATEGORY ADD USER ADD PRODUCT ADD CATEGORY LOGOUT

Figure 28: Admin Navigation Bar

3. Functional screens shot of multipage website

a. Customer's site

- Homepage

HOME PRODUCT CATEGORY



Welcome to User homepage




This is the homepage of the customer website.

Customer homepage code:

```
htdocs > ASM > 🐼 cus_home.php > ...
1  <?php
2  require_once "cus_header.php";
3  session_start();
4  ?>
5      <center>
6          
7          <h1>Wealcome to User homepage</h1>
8      </center>
9
10
11
```

As you can see from the first 4 lines of code, the function `require_once` will call in the “cus_header.php” to create the navigation bar that you see in the homepage. Secondly, the last 4 rows are used to create the home image.

Customer header code:

```
htdocs > ASM >  cus_header.php >  html >  body
1  <?php
2  require_once "functions.php";
3  ?>
4  <!DOCTYPE html>
5  <html lang="en">
6  <head>
7      <meta charset="UTF-8">
8      <meta http-equiv="X-UA-Compatible" content="IE=edge">
9      <meta name="viewport" content="width=device-width, initial-scale=1.0">
10     <title>Web administration</title>
11     <link rel="stylesheet" href="style_admin.css">
12 </head>
13 <body>
14     <nav>
15         <ul>
16             <li><a href="cus_home.php">Home</a></li>
17
18             <li><a href="cus_product.php">Product</a></li>
19             <li><a href="cus_category.php">Category</a></li>
20
21         </ul>
22     </nav>
23 </body>
24 </html>
```

In the first 3 rows, there is one function called `require_once` to call in “function.php”. Next is from rows 4 to 12 used to design the color and the font of the navigation bar. Last 13 rows are used to show all the options that the user can choose and direct them to the link of each option.

Functions code:

```
htdocs > ASM > functions.php > query
1  <?php
2  require_once "db.php";
3  function query ($sql) {
4      global $connection;
5      return $connection->query($sql);
6  }
7  function encrypt ($pass) {
8      return md5($pass);
9  }
```







This is used to encrypt the password for the admin webpage and call in “db.php” to connect to the database.

DB code:

```
htdocs > ASM > db.php > ...
1  <?php
2  $host_name = "localhost";
3  $sql_username = "root";
4  $sql_password = "root";
5  $db_name = "asm";
6  $db_port = "3306";
7  $connection = new mysqli($host_name,$sql_username,$sql_password,$db_name,$db_port);
8  ?>
```

This is used to connect to the admin DB in order to call in all the information and store in more information.



- Customer product page

Product ID	Product Name	Product Price	Product Image	Product Category
1	Honda Vision	30.000.000\$		motorbike
2	Honda Airblade	45.000.000\$		motorbike
3	Honda SH	100.000.000\$		motorbike
4	Mercedes C300	1.969.000.000\$		car
5	Vinfast A2.0	1.000.000.000\$		car
6	BMX GT1	2.100.000\$		bicycle

This is the customer product page.

Customer product code:

```

htdocs > ASM >  cus_product.php >  center
1  <?php
2  require_once "cus_header.php";
3  ?>
4  <center>
5      <table border="1">
6          <tr>
7              <th>Product ID</th>
8              <th>Product Name</th>
9              <th>Product Price</th>
10             <th>Product Image</th>
11             <th>Product Category</th>
12         </tr>
13         <?php
14         $sql = "SELECT * FROM product";
15         $run = query($sql);
16         while ($product = mysqli_fetch_array($run)) {
17             ?>
18             <tr>
19                 <td><?= $product[0] ?> </td>
20                 <td><?= $product[1] ?> </td>
21                 <td><?= $product[2] ?> </td>
22                 <td>
23                     <a href="product_detail.php?productID=<?= $product[0] ?>">
24                         
25                     </a>
26                 </td>
27                 <?php
28                 $sql1 = "SELECT product_type FROM category WHERE category_id = '$product[4]'";
29                 $run1 = query($sql1);
30                 $cls = mysqli_fetch_array($run1);
31                 ?>
32                 <td> <?= $cls[0] ?> </td>
33             </tr>
34         <?php
35         } ?>
36     </table>
37 </center>

```


The first 3 rows are used to create the customer navigation bar. From row 4 to 39 is to create the table that you see from the customer product image and call in all the information from the product database. In row 24 to 26, there are codes to direct the user to the product detail if the user clicks on the image.

- Product detail page:



This is the product detail page on the customer's site.

Product detail code:

```

htdocs > ASM > product_detail.php > html > head > style > td
1  <?php
2  require_once "cus_header.php";
3  session_start();
4  ?>
5  <?php
6  require_once "functions.php";
7  $id = $_GET['productID'];
8  $sql = "SELECT * FROM product WHERE product_id = '$id'";
9  $run = query($sql);
10 $std = mysqli_fetch_array($run);
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>Document</title>
19     <style>
20         img {
21             border-radius: 20%;
22             margin-top: 50px;
23         }
24         td {
25             padding: 20px;
26             color:red;
27         }
28         body {
29             background-color: white;
30         }
31     </style>
32 </head>
33 <body>
34     <center>
35         <table>
36             <tr>
37                 <td>
38                     
40                 </td>
41                 <td>
42                     <h1>Name: <?=$std['product_name'] ?> </h1>
43                     <h1>Price: <?=$std['product_price'] ?> </h1>
44                 </td>
45             </tr>
46         </table>
47     </center>
48 </body>
49 </html>

```

The first 4 rows are also used to call in the navigation bar for customer. The next 7 rows are used to call in the image's specific information from the sql server. The last 37 rows is to design the webpage.



- Customer category page:

HOME PRODUCT CATEGORY

Category ID	Product Type	Product Quantity
1	car	3
2	motorbike	3
3	bicycle	1
4	tank	0

This is the customer category page.

Customer category code:

```
htdocs > ASM >  cus_category.php >  center
1  <?php
2  require_once "cus_header.php";
3
4  $sql = "SELECT * FROM category";
5
6  $run = query($sql);
7  ?>
8  <center>
9  <table border="1">
10     <tr>
11         <th>Category ID</th>
12         <th>Product Type</th>
13         <th>Product Quantity</th>
14     </tr>
15     <?php
16     while ($category = mysqli_fetch_array($run)) {
17         ?>
18         <tr>
19             <td><?= $category[0] ?></td>
20             <td><?= $category[1] ?></td>
21             <td><?= $category[2] ?></td>
22         </tr>
23     <?php
24     }
25     ?>
26 </table>
27 </center>
```

The “require_once” code from row 2 is to call in the customer header. In line 4 and 6, the function is used to call in all the information from category database in sql. The last 20 rows are used to design the table and add in all the information of the category list.

b. Admin's site

- Admin login page:

Login

This is the login page for admins.

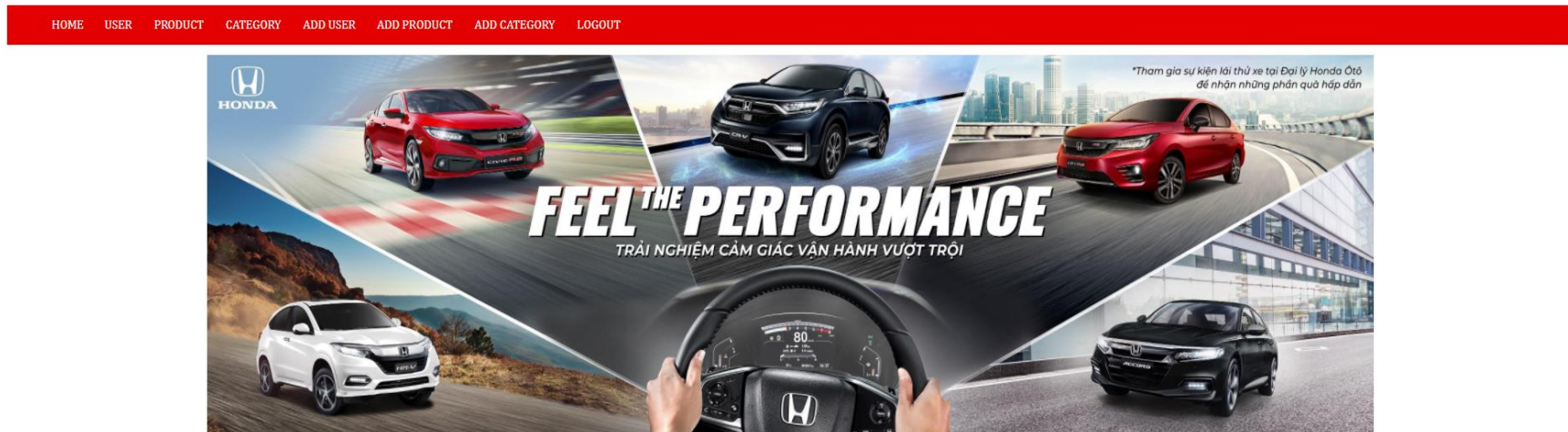
Admin login code:

```
<?php
session_start();
require_once "db.php";
if (isset($_POST['login'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];
    $pass = md5($password);
    $sql = "SELECT * FROM user WHERE user_name = '$username' AND password = '$pass'";
    $run = $connection->query($sql);
    $check = mysqli_fetch_array($run);
    if (is_array($check)) {
        $_SESSION['username'] = $username;
        $_SESSION['password'] = $pass;
        ?>
        <script>
            alert("Login succeed !");
            window.location.href = "home.php";
        </script>
    }
    <?php } else { ?>
    <script>
```

```
        alert("Login failed !");
        window.location.href = "";
    </script>
<?php }
}
else {
?>
<!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</title>
    <link rel="stylesheet" href="style_admin.css">
</head>
<body>
<center>
    <form action="" method="post">
        <fieldset>
            <legend>Login</legend>
            <input type="text" name="username" placeholder="Enter username here" id="" required>
            <br> <br>
            <input type="password" name="password" id="" placeholder="Enter password here" required>
            <br> <br>
            <input type="submit" value="Login" name="login">
        </fieldset>
    </form>
</center>
</body>
</html>
<?php
}
?>
```

The “require_once” code is used to connect to the user database information. The next 7 rows are used to compare the username and password that the user type in to see if it is match or not. If it is match, then the localhost box will notify the user “Login succeed !”. but if it’s not match, the next 6 rows will notify the user that login has been failed. The next 9 rows are used to design the page style. The last few rows are used to notify the user to fill in the username or password if they have not fill it in and press login.

- User homepage:



This is the homepage for admin interface.

Admin home code:

```
<?php
require_once "header.php";
session_start();
if (isset($_SESSION['username']) && isset($_SESSION['password'])) { ?>
    <center>
        
    </center>
<?php } else { ?>
    <script>
        alert("You are not allowed to access this page. Please login first !");
```

```
    window.location.href = "login.php";  
</script>  
<?php } ?>
```

The “require_once” code is to call in the header for admin interface. The last lines are used to check if the user have login or not yet. If the user have login to the admin’s account, the page will show the homepage for admin. But if the user have not enter in the admin’s account, the page will direct the user to the login webpage.

Admin header code: <?php

```
require_once "functions.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>Web administration</title>  
    <link rel="stylesheet" href="style_admin.css">  
</head>  
<body>  
    <nav>  
        <ul>  
            <li><a href="home.php">Home</a></li>  
            <li><a href="view_user.php">User</a></li>  
            <li><a href="view_product.php">Product</a></li>  
            <li><a href="view_category.php">Category</a></li>  
            <li><a href="add_user.php">Add user</a></li>  
            <li><a href="add_product.php">Add product</a></li>  
            <li><a href="add_category.php">Add category</a></li>  
            <li><a href="logout.php">Logout</a></li>  
        </ul>  
    </nav>  
</body>  
</html>
```


The admin header is almost the same as the customer header but just add in more options to the admin privilege and change the link to that webpage.

- Admin user page:

HOME	USER	PRODUCT	CATEGORY	ADD USER	ADD PRODUCT	ADD CATEGORY	LOGOUT
------	------	---------	----------	----------	-------------	--------------	--------

User ID	Username	Password	Options
1	admin	d41d8cd98f00b204e9800998ecf8427e	UPDATE DELETE
2	admins	e10adc3949ba59abbe56e057f20f883e	UPDATE DELETE
3	admin	202cb962ac59075b964b07152d234b70	UPDATE DELETE

This is the admin user page for admins.

Admin user code:

```
<?php
require_once "header.php";
?>
<center>
<table border="1">
<tr>
<th>User ID</th>
<th>Username</th>
<th>Password</th>
<th>Options</th>
</tr>
<?php
$sql = "SELECT * FROM user";
$execute = query($sql);
while ($user = mysqli_fetch_array($execute)) {
?>
<tr>
<td><?= $user[0] ?></td>
```

```
<td><?= $user[1] ?></td>
<td><?= $user[2] ?></td>
<td class="options">
  <form class="inline" action="update_pass.php" method="post">
    <input type="hidden" name="id" value="<?= $user[0] ?>">
    <input type="submit" value="UPDATE" name="update">
  </form>
  <form class="inline" action="delete_user.php" method="post" onsubmit="return confirm_delete()">
    <input type="hidden" name="id" value="<?= $user[0] ?>">
    <input type="submit" value="DELETE" name="delete">
  </form>
</td>
</tr>
<?php
}
?>
</table>
</center>
<script>
function confirm_delete() {
  var del = confirm("Are you sure to delete this user ?");
  if (del) {
    return true;
  } else {
    return false;
  }
}
</script>
```

The first 3 lines are used to call in the navigation bar for admin. All codes in the <center> </center> are used to create the table and add in the information from the user database. This part of the code is nothing much except for the options code. In the “options” code, there are two functions called “Update” and “Delete”. The “Delete” code is in the last 10 lines and the “Update” button need to call in the “Update_pass.php”.

- Admin Product page:

Product ID	Product Name	Product Price	Product Image	Product Category	Options
1	Honda Vision	30.000.000\$		motorbike	UPDATE DELETE
2	Honda Airblade	45.000.000\$		motorbike	UPDATE DELETE
3	Honda SH	100.000.000\$		motorbike	UPDATE DELETE
4	Mercedes C300	1.969.000.000\$		car	UPDATE DELETE
5	Vinfast A2.0	1.000.000.000\$		car	UPDATE DELETE
6	BMX GT1	2.100.000\$		bicycle	UPDATE DELETE

This is the admin product page.

Admin product code:

```
<?php
require_once "header.php";
?>
<center>
  <table border="1">
    <tr>
      <th>Product ID</th>
      <th>Product Name</th>
      <th>Product Price</th>
```

```

<th>Product Image</th>
<th>Product Category</th>
<th>Options</th>
</tr>
<?php
$sql = "SELECT * FROM product";
$run = query($sql);
while ($product = mysqli_fetch_array($run)) {
    ?>
<tr>
<td><?= $product[0] ?> </td>
<td><?= $product[1] ?> </td>
<td><?= $product[2] ?> </td>
<td>
    <a href="product_detail.php?productID=<?= $product[0] ?>">
        
    </a>
</td>
<?php
    $sql1 = "SELECT product_type FROM category WHERE category_id = '$product[4]'";
    $run1 = query($sql1);
    $cls = mysqli_fetch_array($run1);
    ?>
<td> <?= $cls[0] ?> </td>
<td>
    <form class="inline" action="update_product.php" method="post">
        <input type="hidden" name="id" value="<?= $product[0] ?>">
        <input type="submit" value="UPDATE">
    </form>
    <form class="inline" action="delete_product.php" method="post" onsubmit="return confirm_delete()">
        <input type="hidden" name="id" value="<?= $product[0] ?>">
        <input type="submit" value="DELETE">
    </form>
</td>

```

```

</tr>
<?php
} ?>
</table>
</center>
<script>
function confirm_delete() {
    var del = confirm ("Do you want to delete this product ?");
    if (del) {
        return true;
    } else {
        return false;
    }
}
</script>

```

This code is almost the same as the customer product code apart from the “Options” that have 2 button called “Update” and “Delete”. The “Delete” button code is on the last 10 lines. But for the “Update” button need to call in another function called “update_product.php”.

Update Product page:

HOME USER PRODUCT CATEGORY ADD USER ADD PRODUCT ADD CATEGORY LOGOUT

Update product

Name:

Price:

Image:

không có tệp nào được chọn

Category:

Update product code:

```

<?php

```

```
require_once "header.php";

$id = $_POST['id'];
if ($_POST['change']) {
    $name = $_POST['name'];
    $price = $_POST['price'];
    $image = "";
    $current_image = $_POST['current_image'];
    $category = $_POST['category'];
    if (isset($_FILES['image']) && $_FILES['image']['size'] != 0) {
        $temp_name = $_FILES['image']['tmp_name'];
        $img_name = $_FILES['image']['name'];
        $parts = explode(".", $img_name);
        $extension = end($parts);
        $random = rand(1, 100);
        $image = $name . "_" . $class . "_" . $random . "." . $extension;
        $path = "images/" . $image;
        move_uploaded_file($temp_name, $path);
        $sql1 = "UPDATE product SET product_name = '$name', product_price = '$price',
            product_image = '$image', product_category = '$category'
            WHERE product_id = '$id'";
    } else {
        $sql1 = "UPDATE product SET product_name = '$name', product_price = '$price',
            product_image = '$current_image', product_category = '$category'
            WHERE product_id = '$id'";
    }
    $run1 = query($sql1);
    if ($run1) { ?>
        <script>
            alert ("Update product succeed !");
            window.location.href = "view_product.php";
        </script>
    <?php } else { ?>
        <script>
```

```

        alert ("Update product failed !");
        window.location.href = "";
    </script>
<?php } } else {
    $sql2 = "SELECT * FROM product WHERE product_id = '$id'";
    $run2 = query($sql2);
    $std = mysqli_fetch_array($run2);
    ?>
<center>
    <form action="" method="post" enctype="multipart/form-data">
        <fieldset>
            <legend>Update product</legend>
            Name: <input type="text" name="name" required minlength="1" maxlength="30" value="<?= $std[1] ?>">
            <br><br>
            Price: <input type="text" name="price" required minlength="1" maxlength="30" value="<?= $std[2] ?>">
            <br><br>
            Image:
             <br>
            <input type="file" name="image" accept="image/*">
            <br><br>
            Category:
            <select name="category">
                <?php
                $sql = "SELECT * FROM category";
                $run = query($sql);
                while ($cls = mysqli_fetch_array($run)) {
                    if ($cls['category_id'] == $std['product_category']) { ?>
                        <option selected value="<?= $cls['category_id'] ?>">
                            <?= $cls['product_type'] ?>
                        </option>
                    }
                }
                <?php
            } else { ?>
                <option value="<?= $cls['category_id'] ?>">

```

```

        <?= $cls['product_type'] ?>
    </option>
    <?php }
    }?>
</select>
<br><br>
<input type="hidden" name="id" value="<?= $id ?>">
<input type="hidden" name="current_image" value="<?= $std[3] ?>">
<input type="submit" value="UPDATE" name="change">
</fieldset>
</form>
</center>
<?php } ?>

```

The second line is used to call in function “header.php” for admin. The next 7 lines are used to save all the current data to a value and then fill it in the table for the user to make changes created in code <center> ... </center>. And in line 11 to 27, the codes there are used to keep to current image and place it in another value in order to let the user see the image and let the user have the choice to choose different image or keep the current ones.

- Admin Category page:

HOME USER PRODUCT CATEGORY ADD USER ADD PRODUCT ADD CATEGORY LOGOUT

Category ID	Product Type	Product Quantity	Options
1	car	3	UPDATE DELETE
2	motorbike	3	UPDATE DELETE
3	bicycle	1	UPDATE DELETE
4	Tank	0	UPDATE DELETE

This is the admin category page.

Admin category code:

```

<?php
require_once "header.php";

```



```
$sql = "SELECT * FROM category";

$run = query($sql);
?>
<center>
<table border="1">
  <tr>
    <th>Category ID</th>
    <th>Product Type</th>
    <th>Product Quantity</th>
    <th>Options</th>
  </tr>
  <?php
  while ($category = mysqli_fetch_array($run)) {
    ?>
    <tr>
      <td><?= $category[0] ?></td>
      <td><?= $category[1] ?></td>
      <td><?= $category[2] ?></td>
      <td>
        <form class="inline" category="inline" action="update_category.php" method="post">
          <input type="hidden" name="id" value="<?= $category[0] ?>">
          <input type="submit" value="UPDATE">
        </form>
        <form class="inline" category="inline" action="delete_category.php" method="post" onsubmit="return confirm_delete()">
          <input type="hidden" name="id" value="<?= $category[0] ?>">
          <input type="submit" value="DELETE">
        </form>
      </td>
    </tr>
  <?php
  }
  ?>
</table>
```

```
</center>
<script>
  function confirm_delete() {
    var del = confirm("Do you want to delete this category ?");
    if (del) {
      return true;
    } else {
      return false;
    }
  }
</script>
```

Much like the customer category code, the admin category code has almost the same amount of content as the customer category code but in the admin category code, there are 2 options for the admin to choose from “Update” and “Delete”. The “Delete” button is like the product delete button with the code in the last 10 lines. And for the “Update” button, the author also need to create another function called “update_category.php”.

- Update category page:

HOME USER PRODUCT CATEGORY ADD USER ADD PRODUCT ADD CATEGORY LOGOUT

**UPDATE
CATEGORY**

Product Type:

Quantity:

This is the update category for admin.

Update category code:

```
<?php
require_once "header.php";

$id = $_POST['id'];
if (isset($_POST['update'])) {
    $name = $_POST['name'];
    $quantity = $_POST['quantity'];
    $sql = "UPDATE category SET product_type = '$name', product_quantity = '$quantity' WHERE category_id = '$id'";
    $run = query($sql);
    echo $run;
    if ($run) { ?>
        <script>
            alert ("Update category succeed !");
            window.location.href = "view_category.php";
        </script>
    <?php } else { ?>
        <script>
            alert ("Update category failed !");
            window.location.href = "";
        </script>
    <?php } } else {
        $sql1 = "SELECT * FROM category WHERE category_id = '$id'";
        $run1 = query($sql1);
        $class = mysqli_fetch_array($run1);
        ?>
    <form action="" method="post">
        <fieldset>
            <legend>UPDATE CATEGORY</legend>
            Product Type: <input type="text" name="name" id=""
            minlength="1" maxlength="10" value="<?= $class[1] ?>" required>
            <br><br>
            Quantity: <input type="number" name="quantity" id=""
            min="0" max="30" value="<?= $class[2] ?>" required>
            <br><br>
```

```
<input type="hidden" name="id" value="<?= $id ?>">
<input type="submit" value="UPDATE" name="update">
</fieldset>
</form>
<?php } ?>
```

The code “require_once” in line 2 is to call in the function “header.php” for admin. In the line 6 and 7, there are 2 variables call out to hold the current category information which are call out in the table created in <form>...</form>. And from line 8 to line 28, these codes are used to update the category table with the information that the user have entered in and check if the update have been successful or not.

- Add product page:

HOME USER PRODUCT CATEGORY ADD USER ADD PRODUCT ADD CATEGORY LOGOUT

Add product

Name:

Price:

Image: Không có tệp nào được chọn

Category:

This is the add product page for admin.

Add product code:

```
<?php
require_once "header.php";
if ($_POST['add']) {
    $name = $_POST['name'];
    $price = $_POST['price'];
    $image = "";
    $category = $_POST['category'];

    if (isset($_FILES['image']) && $_FILES['image']['size'] != 0) {
        $temp_name = $_FILES['image']['tmp_name'];
```

```

$img_name = $_FILES['image']['name'];
$parts = explode(".", $img_name);
$extension = end($parts);
$random = rand(1, 100);
$image = $name . "_" . $class . "_" . $random . "." . $extension;
$path = "images/" . $image;
move_uploaded_file($temp_name, $path);
}
$sql1 = "INSERT INTO product (product_name, product_price, product_image, product_category) VALUES ('$name', '$price', '$image', '$category')";
$run1 = query($sql1);
if ($run1) { ?>
    <script>
        alert ("Insert new product succeed !");
        window.location.href = "view_product.php";
    </script>
<?php } else { ?>
    <script>
        alert ("Insert new product failed !");
        window.location.href = "";
    </script>
<?php } } else { ?>
<center>
    <form action="" method="post" enctype="multipart/form-data">
        <fieldset>
            <legend>Add product</legend>
            Name: <input type="text" name="name" required><br><br>
            Price: <input type="text" name="price" required> <br><br>
            Image: <input type="file" name="image" accept="image/*" required> <br><br>
            Category:
            <select name="category">
                <?php
                    $sql = "SELECT * FROM category";
                    $run = query($sql);
                    while ($cls = mysqli_fetch_array($run)) { ?>

```

```
<option value="<?= $cls['category_id']?>">
    <?= $cls['product_type'] ?>
</option>
<?php } ?>
</select>
<br><br>
<input type="submit" value="ADD" name="add">
</fieldset>
</form>
</center>
<?php } ?>
```

In line 2, the code “require_once” is to call in the function “header.php” for the admin users. From line 4 to 7, there are 4 variables call out to hold the new product information which are fill in the table created in <form>...</form>. In line 19, 20 and 21, all of that variable will be add on to the product table by using an sql line code. Line 9 to 18 are used to allows admin to choose a picture from their computer to illustrate the product. From line 22 to 32, the codes are used to check if the variable add on to “product” database is successful or not.

- Add category page:

HOME USER PRODUCT CATEGORY ADD USER ADD PRODUCT ADD CATEGORY LOGOUT

**ADD
CATEGORY**

Product Type:

Quantity:

This is the add category page for admin.

Add category code:

```
<?php
require_once "header.php";

$id = $_POST['id'];
```

```
if (isset($_POST['add'])) {  
    $name = $_POST['name'];  
    $quantity = $_POST['quantity'];  
    $sql = "INSERT INTO category (product_type, product_quantity) VALUES ('$name','$quantity')";  
    $run = query($sql);  
    if ($run) { ?>  
        <script>  
            alert ("Add category succeed !");  
            window.location.href = "view_category.php";  
        </script>  
    <?php } else { ?>  
        <script>  
            alert ("Add category failed !");  
            window.location.href = "";  
        </script>  
    <?php } }else { ?>  
    <center>  
        <form action="" method="post">  
            <fieldset>  
                <legend>ADD CATEGORY</legend>  
                Product Type: <input type="text" name="name" id=""  
                minlength="1" maxlength="10" required>  
                <br><br>  
                Quantity: <input type="number" name="quantity" id=""  
                min="0" max="100" required>  
                <br><br>  
                <input type="submit" value="ADD" name="add">  
            </fieldset>  
        </form>  
    </center>  
    <?php } ?>
```

In the first 2 lines of the code, it called in the function “header.php” for the admin users. In line 4, 6 and 7, there are 3 variables called out in order to hold in all the information that the user entered in and then insert them into the category database by using the sql code

in line 8 and 9. After that, line 10 to line 20 use the if loop to check whether it can be added in or not. Line 21 to 34 is used to create a “Add Category” table for the user to input new category.

- Add user page:



**Add
User**

This is the add user page for admin.

Add user code:

```
<?php
require_once "header.php";

if (isset($_POST['add'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];
    $confirm = $_POST['confirm'];
    $sql = "SELECT user_name FROM user WHERE user_name = '$username'";
    $run = query($sql);
```



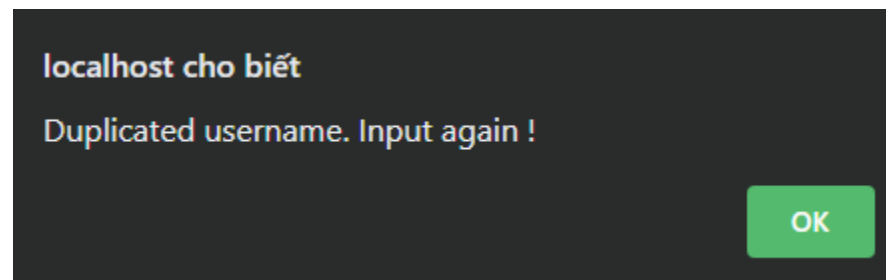
```
$check = mysqli_fetch_array($run);
if (is_array($check)) { ?>
    <script>
        alert("Duplicated username. Input again !");
        window.location.href = "";
    </script>
<?php } else {
    if ($password != $confirm) { ?>
        <script>
            alert("Passwords are not similar. Input again !");
            window.location.href = "";
        </script>
    <?php } else {
        $pass = encrypt($password);
        $sql1 = "INSERT INTO user (user_name, password) VALUES ('$username','$pass')";
        $run1 = $connection->query($sql1);
        if ($run1) { ?>
            <script>
                alert("Add new user succeed !");
                window.location.href = "view_user.php";
            </script>
        <?php } else { ?>
            <script>
                alert("Add new user failed !");
                window.location.href = "";
            </script>
        <?php }
    }
} else {
    ?>
    <form action="" method="post" enctype="multipart/form-data">
        <fieldset>
            <legend>Add User</legend>
```

```
<input type="text" name="username" placeholder="Username" id="" required>
<br><br>
<input type="password" name="password" placeholder="Password" minlength="3" maxlength="20" required>
<br><br>
<input type="password" name="confirm" placeholder="Confirm password" minlength="3" maxlength="20" required>
<br><br>
<input type="submit" value="ADD" name="add">
</fieldset>
</form>
<?php } ?>
```

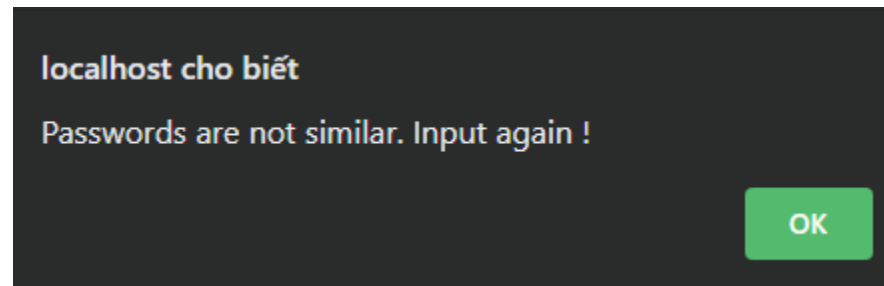
In this code, a navigation bar for admin had been called in in line 2. There are 3 empty variables that the coder called out in order to hold the new information that the user entered in. Next up, the variable will be added in the user database through the code in line 24 and 25. Line 10 to 21 will be used to check if the username was duplicated or not and if the password that the user type in match the password confirmation or not. From line 41 till the end, the coder used `<form></form>` and `<fieldset></fieldset>` to create the table for user to add in the information to the category table.

4. Some screenshots related to error handling

If the user enter a new account that the username was duplicated, the localhost will inform a line that said:



If the user enter a new account but the password did not match the confirm password, the localhost will show:



If the user enter a new account but the password did not long enough, localhost will also announce:

HOME
USER
PRODUCT
CATEGORY
ADD USER
ADD PRODUCT
ADD CATEGORY
LOGOUT

Add User

! Vui lòng kéo dài văn bản này thành 3 ký tự trở lên (bạn hiện đang sử dụng 1 ký tự).

ADD

III. Web testing

No	Test case	Input data	Expected output	Actual output	Evaluation
1	Admin Login	Username, password	Localhost: “Login succeed !”	Localhost: “Login succeed !”	The website run as expected (Pass)
2	Admin Logout	Press “Logout” button after admin logging in	Direct to the “login.php” page	Direct to the “login.php” page	The website run as expected (Pass)
3	Add User	Enter New username and password similar to confirm password	Localhost: “Add new user succeed !”	Localhost: “Add new user succeed !”	The website run as expected (Pass)

4	Delete User	Press “Delete” button in admin user page	Localhost: “Are you sure to delete this user ?”	Localhost: “Are you sure to delete this user ?”	The website run as expected (Pass)
5	Confirm Delete User	Press “OK” when the Localhost ask: “Are you sure to delete this user ?”	Localhost: “Delete user succeed !”	Localhost: “Delete user succeed !”	The website run as expected (Pass)
6	Update password	Press “Update” button in admin user page	Direct to the “update_password.php” page	Direct to the “update_password.php” page	The website did not run as expected (Failed)
7	Add product	Enter Product Name, Price, Image, Category	Localhost: “Insert new product succeed !”	Localhost: “Insert new product succeed !”	The website run as expected (Pass)
8	Update product	Press “Update” button in admin product page	Direct to the “update_product.phpt” page	Direct to the “update_product.phpt” page	The website run as expected (Pass)
8	Confirm Update product	Replace new price	Show new Price in the product table	Show new Price in the product table	The website run as expected (Pass)
9	Delete product	Press “Delete” button in admin product page	Localhost: “Are you sure to delete this product ?”	Localhost: “Are you sure to delete this product ?”	The website run as expected (Pass)
10	Confirm Delete Product	Press “OK” when the Localhost ask: “Are you sure to delete this product ?”	Localhost: “Delete product succeed !”	Localhost: “Delete product succeed !”	The website run as expected (Pass)
11	Add category	Enter Product Type, Quantity	Localhost: “Add category succeed !”	Localhost: “Add category succeed !”	The website run as expected (Pass)

12	Update category	Press “Update” button in admin category page	Direct to the “update_category.php” page	Direct to the “update_category.php” page	The website run as expected (Pass)
13	Confirm Update Category	Replace new Quantity	Show new Quantity in the category table	Show new Quantity in the category table	The website run as expected (Pass)
14	Delete Category	Press “Delete” button in admin category page	Localhost: “Are you sure to delete this category ?”	Localhost: “Are you sure to delete this category ?”	The website run as expected (Pass)
15	Confirm Delete Category	Press “OK” when the Localhost ask: “Are you sure to delete this category ?”	Localhost: “Delete category succeed !”	Localhost: “Delete category succeed !”	The website run as expected (Pass)

IV. Reference:

Dickens, A., 2019. *Web Standards: The What, The Why, And The How* — *Smashing Magazine*. [online] Smashing Magazine. Available at: <<https://www.smashingmagazine.com/2019/01/web-standards-guide/>> [Accessed 9 May 2021].

Luenendonk, M., 2020. *Web Design Principles of Successful Websites*. [online] Cleverism. Available at: <<https://www.cleverism.com/web-design-principles-successful-websites/>> [Accessed 9 May 2021].