



:UNIVERISTY of MESSINA:
PROFESSOR: ARMANDO RUGGERI
Course: Web Programming
Created by: Shehryar Raja
Matricula: 523115

Contents :

- 1. Introduction**
- 2. Project requirement**
- 3. Website overview**
- 4. Explanation of codes**
- 5. Conclusion**

Introduction:

Online pet food purchases are becoming more and more popular in today's environment. Consumers seek a simple online method to buy pet supplies without having to leave their houses. The aim of this project is to develop a website that facilitates pet owners' search and purchase of various pet food types, including treats, wet food, and dry food. We want pet owners to have a pleasant and simple internet buying experience.

This document outlines the steps we took to create our website for pet food. Everything is included, including constructing and testing the website in addition to organizing and designing it. Presenting the website's functionalities, development process, and features is the aim.

Project Requirements:

The website requires both front-end as well as back-end. The languages in which the website has been made and designed include HTML, CSS, JavaScript, PHP, and MySQL.

Front-end requirements:

the user interface, or UI:

The website should have an intuitive, contemporary, and clean layout. Links, menus, and buttons should be simple to locate and use. Important components like the shopping cart, product photos, and navigation bar should be easily seen and understood.

Product Listings:

Pet food goods should be viewable by users, together with information about each item such as pricing, descriptions, and an "Add to Cart" button.

Bar of Search:

Users should be able to easily locate certain goods by using the search box. Based on user input, it needs to deliver results instantly.

Cart of goods:

The quantity of products a user has added needs to be shown on the shopping cart symbol. It should be possible for users to see, modify, and remove goods from their carts.

Form of Contact:

Users should be able to send messages to the website's customer care using a contact form. There should be fields for the following: message, topic, phone, email, and name.

Back-end specifications (the inner workings of a system)

Database Administration:

Product information (name, description, price, and photos) and user account information (login details, cart items, etc.) should be managed by a database.

Additionally, orders and client inquiries via the contact form should be stored in the database.

Management of Sessions:

In order to enable users to add products even when they aren't signed in, their shopping carts must be kept in the session. Until checkout or session expiry, cart data should remain active.

User Verification:

Users should be able to sign up, log in, and log out of their accounts on the website. Users should have secure access using a session-based login mechanism.

Product Administration:

Products should be able to be added, edited, and removed from the database by administrators. By doing this, the product catalog is kept current.

Features of a Shopping Cart:

Things must be able to be added by users to the cart, and the things ought to be tracked by the back end. Real-time item counts should be displayed in the dynamically updated cart.

The website has to compute the total cost of the products in the cart and get ready for the user to check out.

Functionality of Search:

In response to user input, a back-end system should manage search queries and retrieve pertinent goods from the database.

Integration of AJAX:

AJAX, or asynchronous JavaScript and XML, should be used by the website to do tasks like adding items to the cart without refreshing the page. A more seamless user experience results from this.

Safety:

To secure user data, the website should utilize security features like password hashing. To stop typical vulnerabilities like SQL injection and cross-site scripting (XSS), input validation should be included.

Notifications by email:

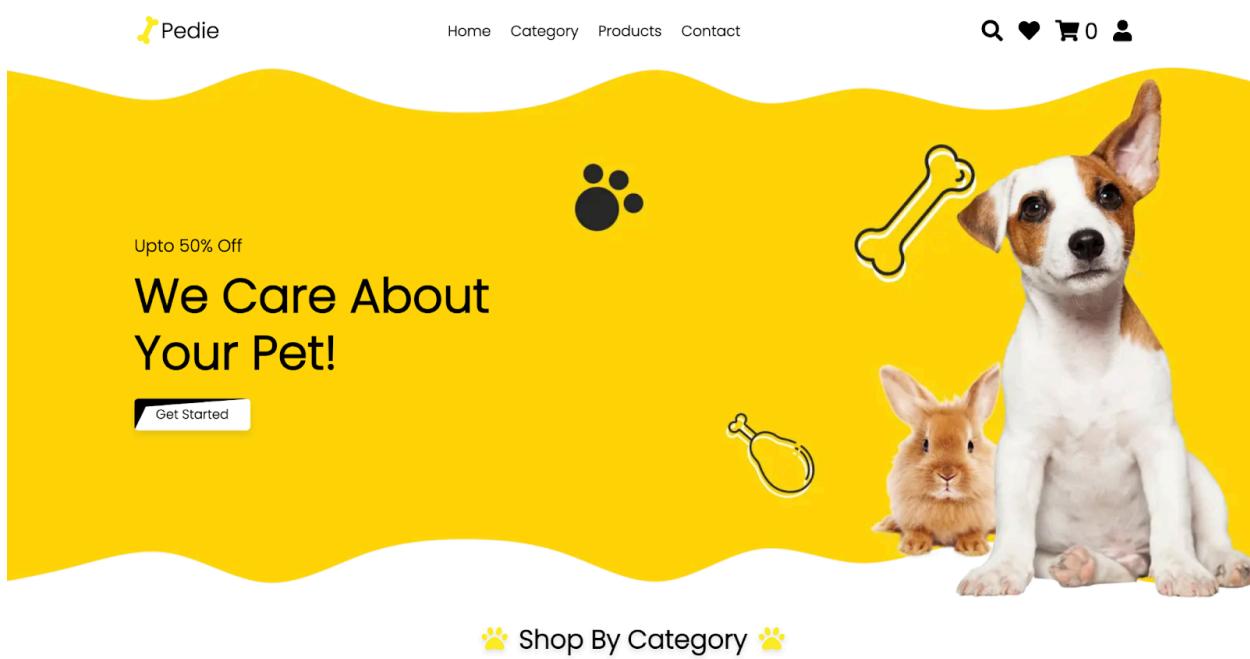
Users should get confirmation emails from the system after completing tasks like creating an account, changing their password, and confirming an order.

Error Resolution:

When anything goes wrong, the back end should manage problems gracefully and give users informative error messages (e.g., if a product is out of stock or there's an issue with their account).

Website Overview:

The website is called Pedie (pet food shop). Basically, this is the front and starting view of the website, also called the home page or main page of the website.



Next we have a category section in which we have divided the food and products based on their categories.

[Shop By Category](#)



These are the products that are available in our store.

 Our Cat Food Products 



Dry Cat Food
Dry Cat Food

\$19.99

 Add To Cart



Wet Cat Food - Tuna Flavor
Wet Cat Food - Tuna Flavor

\$15.99

 Add To Cart



Grain-Free Cat Food
Grain-Free Cat Food

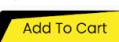
\$22.99

 Add To Cart



Organic Cat Food
Organic Cat Food

\$24.99

 Add To Cart

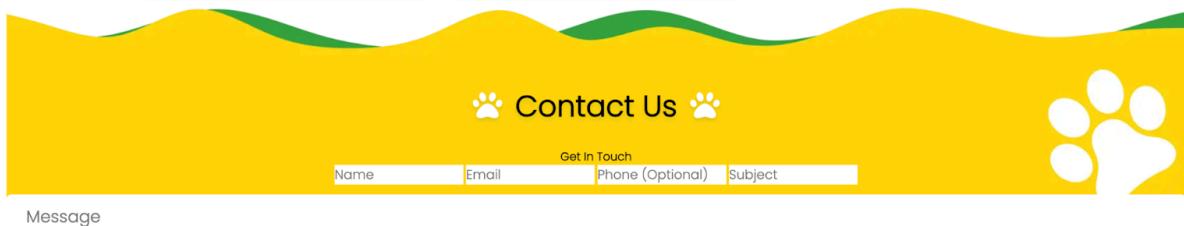


Cat Treats - Chicken Bites
Cat Treats - Chicken Bites

\$9.99

 Add To Cart

Then we have the contact form for the customers so they can write their inquiries in the contact section. It can be regarding orders regarding the products or any information for the shop. They can simply fill in their information and type the message below in the text section.

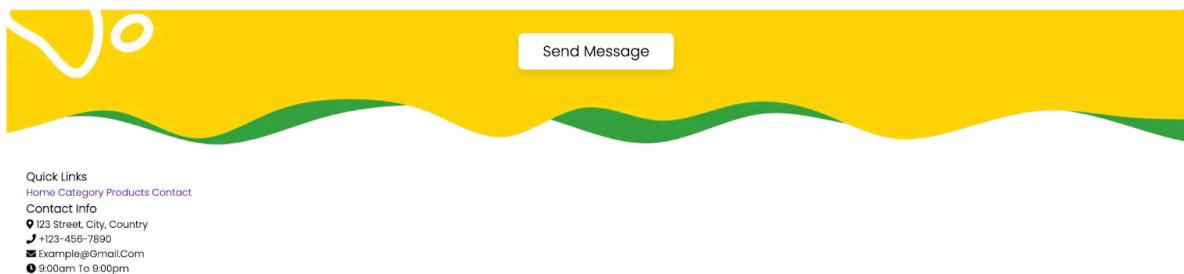


 Contact Us 

Get in Touch

Name Email Phone (Optional) Subject

Message



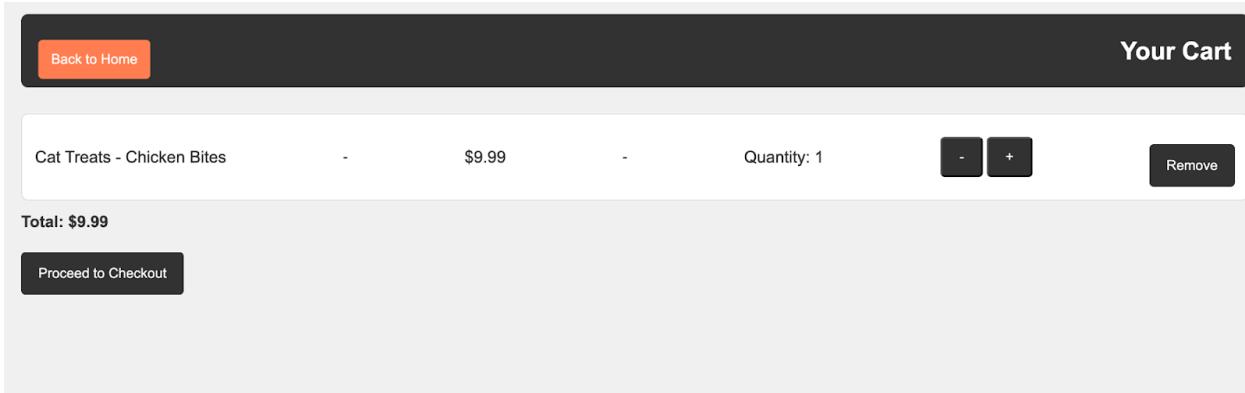


Quick Links
[Home](#) [Category](#) [Products](#) [Contact](#)

Contact Info
📍 123 Street, City, Country
📞 +123-456-7890
✉️ Example@gmail.com
⌚ 9:00am To 9:00pm

Add to Cart option:

The users can add products in the add to cart option. From here they can buy the products.



After selecting the items, users can click on proceed to checkout, and they can pay, and the order will be submitted.

The user will enter cart details here and proceed to pay.

The screenshot shows a "Checkout" page. At the top left is a "Your Cart" section showing "Cat Treats - Chicken Bites - 1 x \$9.99". To the right is the word "Checkout". Below the cart summary, it says "Total Amount: \$9.99". There are two main sections: "Customer Information" and "Payment Information". The "Customer Information" section contains fields for "Name" (with a placeholder), "Email" (with a placeholder), and "Address" (with a large text area). The "Payment Information" section contains fields for "Card Number" (with a placeholder) and "Card Expiry (MM/YY)" (with a placeholder).

Checkout

[Back to Cart](#)

Your Cart

Cat Treats - Chicken Bites - 1 x \$9.99

Total Amount: \$9.99

Customer Information

Name:

Email:

Address:

Payment Information

Card Number:

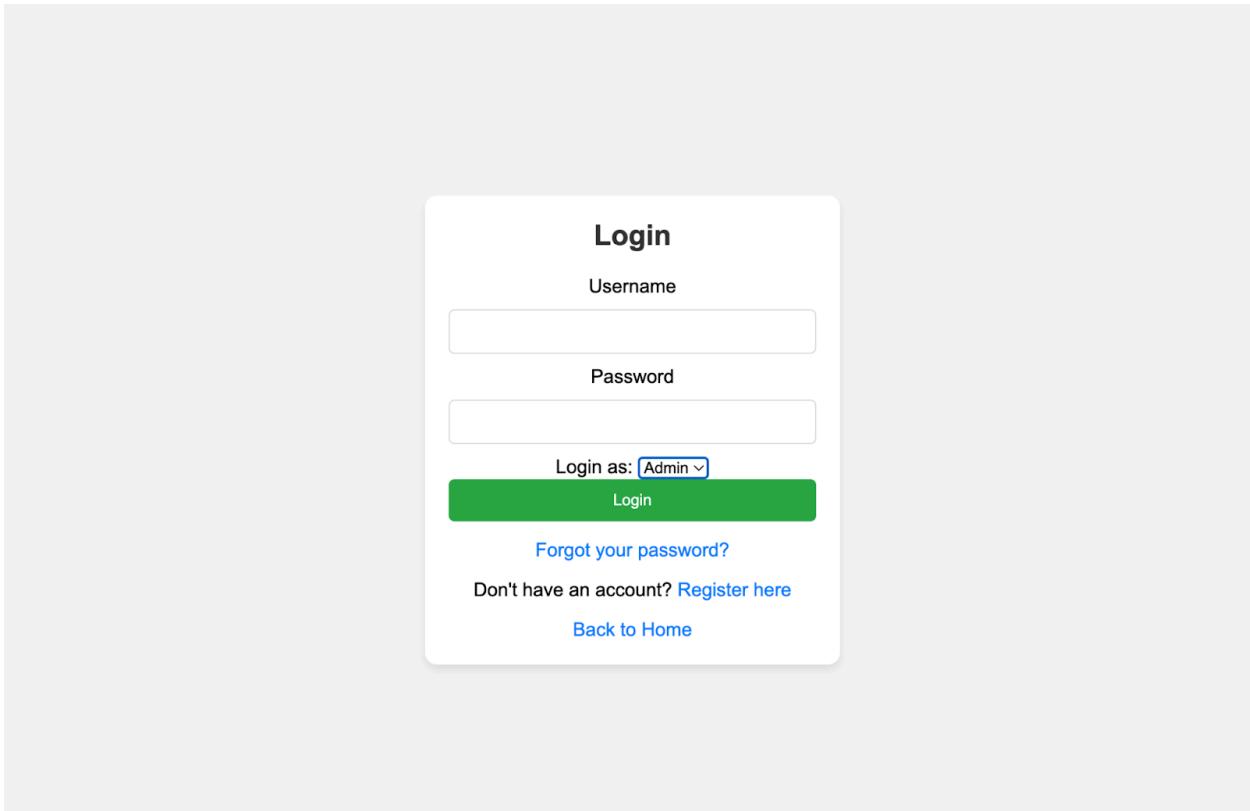
User Log-in:

This is a user-login interface. The user can log in to their account and see their profile; they can add or remove orders; and they can change the passwords of their accounts. It's like a user dashboard.

The image shows a user login form titled "Login". The form has two input fields: "Username" and "Password", both represented by empty text input boxes. Below these fields is a dropdown menu labeled "Login as:" with the option "User" selected. A large green button labeled "Login" is positioned below the dropdown. At the bottom of the form, there are three links: "Forgot your password?", "Don't have an account? [Register here](#)", and "Back to Home".

Admin Log in:

Same as user log-in, this is an admin dashboard where admins can login and see details about orders and customers. Admin can add new products, and admin can remove them as well. Admin can perform several different operations as well.



User registration: if a user is not registered, he or she can register first to log in to their accounts.

Register

[Back to Previous Page](#)

Username:

Email:

Password:

Confirm Password:

Register

Already have an account? [Login here](#)

The Forgot password option is also added so the users can change the password if they forget their password.

Forgot Password

Enter your email

[Request Reset Link](#)

[Back to Previous Page](#)

This is the admin dashboard. As we can say here, we can see the details about the orders done, and in the second box, the customer contacts.

Welcome, Admin!

Logged in as: sherry

Manage Orders

ID	Customer Name	Email	Address	Total Amount	Payment Status	Created At
3	vsad	shery@bhabs	sdsad	143.23	Paid	2024-08-23 14:21:09
4	sherry	sdhasjdas@sadasbd	corso cavour	0.00	Paid	2024-08-23 14:22:03
5	ali	shery@bhabs	corso cavour	24.08	Paid	2024-08-23 14:23:06
6	azan	aza@gama	asjabs	237.94	Paid	2024-08-23 14:33:38

Customer Contacts

ID	Name	Email	Phone	Subject	Message	Created At
1	sami	sami@gmail			i want to buy cat	2024-08-24 14:56:07
2	SHEHRYAR	bsdbdb@gmail.com			swwdc	2024-08-28 13:44:31
3	SHEHRYAR	bsdbdb@gmail.com	2342543453	jksahdkh	swwdc	2024-08-28 13:49:30
4	SHEHRYAR	shery@abbsba	1234455	food	i want to buy food.	2024-08-28 14:53:52
5	SHEHRYAR	shery@abbsba	1234455	food	i want to buy food.	2024-08-28 14:53:57

4	SHEHRYAR	shery@abbsba	1234455	food	i want to buy food.	2024-08-28 14:53:52
5	SHEHRYAR	shery@abbsba	1234455	food	i want to buy food.	2024-08-28 14:53:57
6	SHEHRYAR	shery@abbsba	1234455	food	i want to buy food.	2024-08-28 14:58:41
7	SHEHRYAR	shery@abbsba	1234455	food	i want to buy food.	2024-08-28 14:59:48
8	SHEHRYAR	shery@abbsba	1234455	food	i want to buy food.	2024-08-28 14:59:59
9	SHEHRYAR	shery@abbsba	1234455	food	i want to buy food.	2024-08-28 15:00:44
10	SHEHRYAR	shery@abbsba	1234455	food	i want to buy food.	2024-08-28 15:03:57
11	SHEHRYAR	shery@abbsba	1234455	food	i want to buy food.	2024-08-28 15:04:07
12	SHEHRYAR	shery@abbsba	1234455	food	i want to buy food.	2024-08-28 15:04:16
14	SHEHRYAR	shery@abbsba	1234455	food	i want to buypablo cat	2024-08-28 15:05:00
15	SHEHRYAR	shery@abbsba	1234455	food	i want to buypab	2024-08-28 15:05:20

Admin Actions

[Add New Product](#)

[View All Users](#)

[Logout](#)

Then we go to two options: one is to add new products so the user can add new products their pricing in the shop, and he or she can delete the products as well.

The second option is to view all users. The people who have registered on the website as customers.

Manage Products

Add New Product

Product Name:

Product Price:

[Add Product](#)

Current Products

ID	Name	Price	Actions
1	Pablo Cat Food	8.99	<input type="text" value="Pablo Cat Food"/> <input type="text" value="8,99"/> Update Delete
2	dog food	5.99	<input type="text" value="dog food"/> <input type="text" value="5,99"/> Update Delete

2	dog food	5.99	<input type="text" value="5,99"/> Update Delete
3	parrot food	6.99	<input type="text" value="parrot food"/> <input type="text" value="6,99"/> Update Delete
4	fish food	7.99	<input type="text" value="fish food"/> <input type="text" value="7,99"/> Update Delete
5	chicken food	9.99	<input type="text" value="chicken food"/> <input type="text" value="9,99"/> Update Delete

[Back to Dashboard](#)

Here we have all user details.

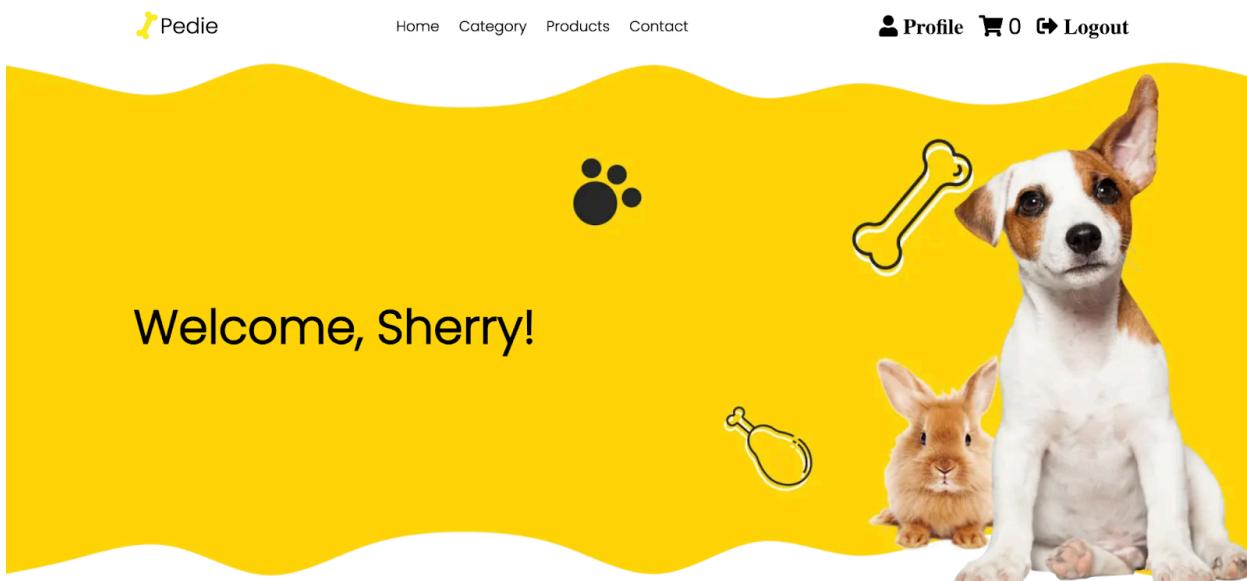
All Users

Manage Users

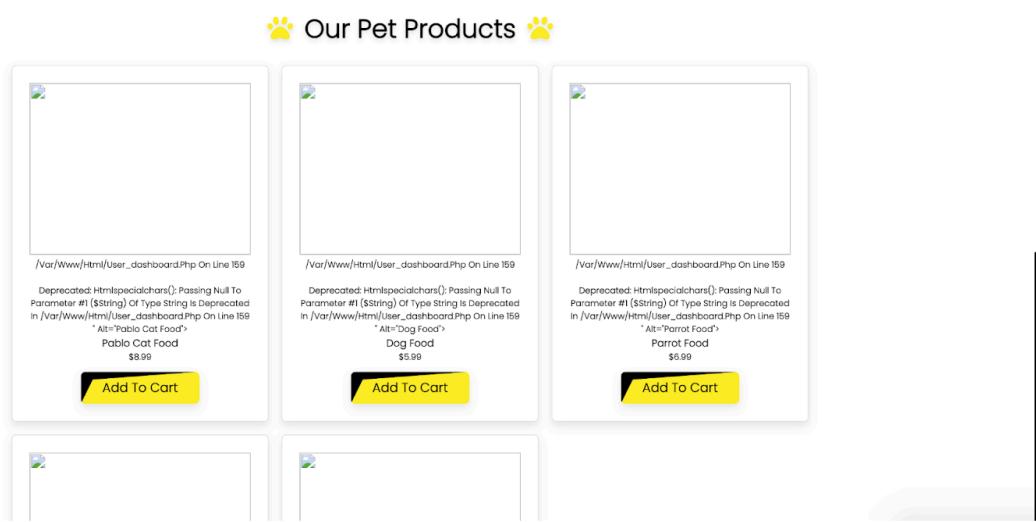
ID	Username	Email	Actions
2	Sami	sami@gmail.com	<button>Edit</button> <button>Delete</button>
4	sherry	sherry@gmail.com	<button>Edit</button> <button>Delete</button>

[Back to Dashboard](#)

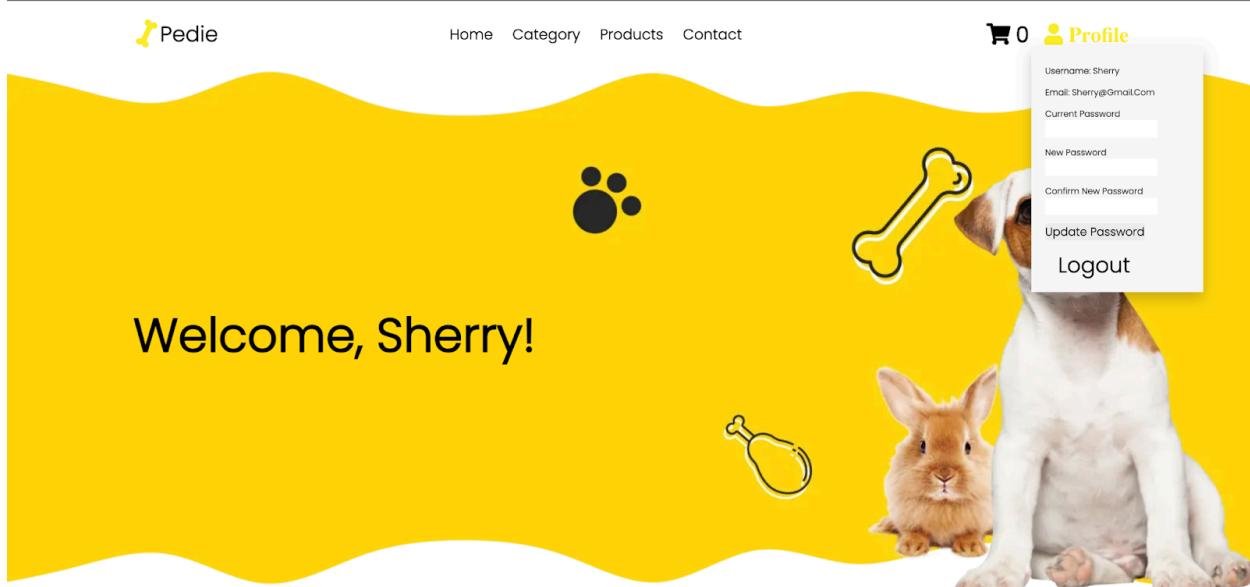
This is the interface when the user logs in to their accounts. There are three options on the top right: one is profile, the second is the cart option, and the third is log-out. In the profile section, the user can see its details; the password can be changed as well; and in the cart option, the user can buy the products that they have already added to their cart.



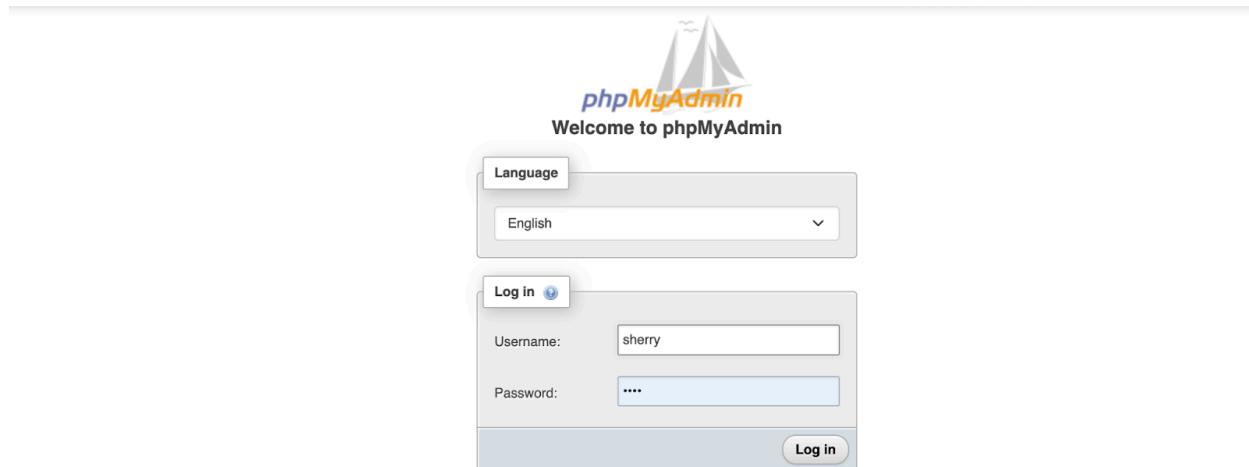
The user can add products from here as well.



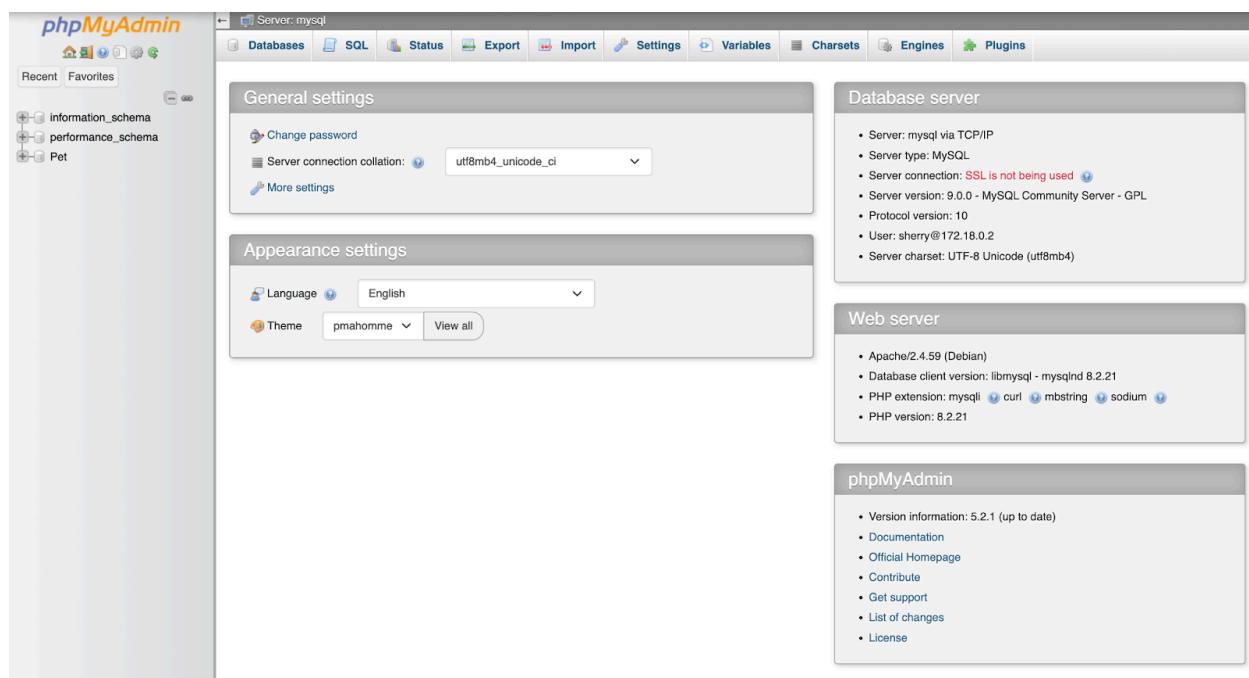
This is the profile section. First is the user name, then email, and after that, change the password in case the user forgot or wants to change for security purposes.



This is my phpMyAdmin login: once I login into this, I will get access to my database of the website.



The screenshot shows the phpMyAdmin login interface. At the top center is the phpMyAdmin logo featuring a sailboat icon and the text "phpMyAdmin". Below it is the message "Welcome to phpMyAdmin". There are two main sections: "Language" and "Log in". The "Language" section has a dropdown menu set to "English". The "Log in" section contains fields for "Username" (set to "sherry") and "Password" (represented by four dots). A "Log in" button is located at the bottom right of the log in form.



The screenshot shows the phpMyAdmin configuration interface. The top navigation bar includes links for "Databases", "SQL", "Status", "Export", "Import", "Settings", "Variables", "Charsets", "Engines", and "Plugins". On the left sidebar, there are "Recent" and "Favorites" sections, and a list of databases: "information_schema", "performance_schema", and "Pet". The main content area is divided into several panels: "General settings" (with "Change password", "Server connection collation" set to "utf8mb4_unicode_ci", and a "More settings" link); "Database server" (listing MySQL server details like version 9.0.0, user sherry@172.18.0.2, and charset UTF-8 Unicode); "Appearance settings" (showing "Language" set to "English" and "Theme" set to "pmauhomme"); "Web server" (listing PHP and MySQL versions, and extensions like mysqli, curl, mbstring, and sodium); and a "phpMyAdmin" footer panel with links for version information, documentation, and support.

Here are all the tables in the database.

The screenshot shows the phpMyAdmin configuration interface. On the left, a sidebar lists databases: information_schema, performance_schema, Pet (which contains New, admins, cart, contacts, orders, order_items, password_resets, products, and users). The main area has several tabs: General settings, Appearance settings, Database server, Web server, and phpMyAdmin. The General settings tab shows a 'Server connection collation' dropdown set to utf8mb4_unicode_ci. The Appearance settings tab shows the language as English and theme as pnahomme. The Database server tab provides detailed server information: Server: mysql via TCP/IP, Server type: MySQL, Server connection: SSL is not being used, Server version: 9.0.0 - MySQL Community Server - GPL, Protocol version: 10, User: sherry@172.18.0.2, and Server charset: UTF-8 Unicode (utf8mb4). The Web server tab shows Apache/2.4.59 (Debian), Database client version: libmysql - mysqld 8.2.21, PHP extension: mysqli, curl, mbstring, sodium, and PHP version: 8.2.21. The phpMyAdmin tab lists version information: Version information: 5.2.1 (up to date), Documentation, Official Homepage, Contribute, Get support, List of changes, and License.

Codes Explanation:

```
src > index.php
 1  <?php
 2  session_start(); // Start the session to manage cart items
 3  require('dbconnection.php');
 4  ?>
 5  <!DOCTYPE html>
 6  <html lang="en">
 7  <head>
 8      <meta charset="UTF-8">
 9      <meta name="viewport" content="width=device-width, initial-scale=1.0">
10      <title>Pet Food</title>
11
12      <!-- font awesome cdn link -->
13      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">
14
15      <!-- custom css file link -->
16      <link rel="stylesheet" href="style.css">
17
18  </head>
19  <body>
20
21  <!-- header section starts -->
22
23  <header class="header">
24
25      <section class="flex">
26
27          <div id="menu" class="fas fa-bars"></div>
28
29          <a href="#" class="logo"><i class="fas fa-bone"></i>pedie</a>
30
31          <nav class="navbar">
32              <a href="#home">home</a>
33              <a href="#category">category</a>
34              <a href="#products">products</a>
35              <a href="#contact">contact</a>
36          </nav>
```

This is an index.php file; it has complete coding about the main page of the website, and the code is written in HTML.

```
src > index.php
23  <header class="header">
25      <section class="flex">
31          <nav class="navbar">
32              <a href="#category">category</a>
34          <a href="#products">products</a>
35          <a href="#contact">contact</a>
36      </nav>
37
38      <div class="icons">
39          <i class="fas fa-search" id="search-icon"></i>
40          <a href="#" class="fas fa-heart"></a>
41          <!-- Updated cart icon with session-based cart count -->
42          <a href="cart.php" class="fas fa-shopping-cart">
43              <span id="cart-count"><?php echo isset($_SESSION['cart']) ? count($_SESSION['cart']) : 0; ?></span>
44          </a>
45          <a href="login.php" class="fas fa-user"></a>
46      </div>
47
48      <form action="" id="search-box">
49          <input type="search" id="search" placeholder="search here">
50          <label for="search" class="fas fa-search"></label>
51      </form>
52
53  </section>
54
55 </header>
56
57 <!-- header section ends -->
58
59 <!-- home section starts -->
60
61 <div class="home" id="home">
62
63     <section class="flex">
64         <div class="content">
65             <span>upto 50% off</span>
66         </div>
67     </section>
68
69 </div>
```

This is a CSS file. It is used for designing the website.

```
src > style.css > ::selection
1  @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300&display=swap');
2
3  :root{
4      --yellow:#fcee26;
5      --black:#000000;
6      --white:#ffffff;
7  } /* define css variable for colours */
8
9  *{
10      font-family: 'Poppins', sans-serif;
11      margin:0; padding:0;
12      box-sizing: border-box;
13      outline: none; border:none;
14      text-decoration: none;
15      text-transform: capitalize;
16      transition: .3s cubic-bezier(.38,1.15,.7,1.12);
17  }
18
19  *::selection{
20      background:var(--black);
21      color:var(--white);
22  }
23
24  html{
25      font-size: 62.5%;
26      overflow-x: hidden;
27  }
28
29  html::-webkit-scrollbar{
30      width:1.3rem;
31  }
32
33  html::-webkit-scrollbar-track{
34      background:transparent;
35  }
```

This is the dbconnection.php file that connects our website with the database. The username and password are set for the admin login in the database.

```
src > dbconnection.php
1  <?php
2  // Database connection parameters
3  $host = 'mysql';           // Database server name, e.g., 'localhost' or 'mysql' for Docker
4  $user = 'sherry';          // Your MySQL username
5  $password = 'root';        // Your MySQL password
6  $db = 'Pet';              // Your database name
7
8  // Create connection
9  $conn = mysqli_connect($host, $user, $password, $db);
10
11 // Check connection
12 if (!$conn) {
13     die("Connection failed: " . mysqli_connect_error());
14 }
15
16 // Optionally log successful connection in a file instead
17 // file_put_contents('log.txt', 'Connected successfully to the database!', FILE_APPEND);
18 ?>
19
```

This script improves the user experience by adding animations and toggling features for search, menu, and smooth scrolling.

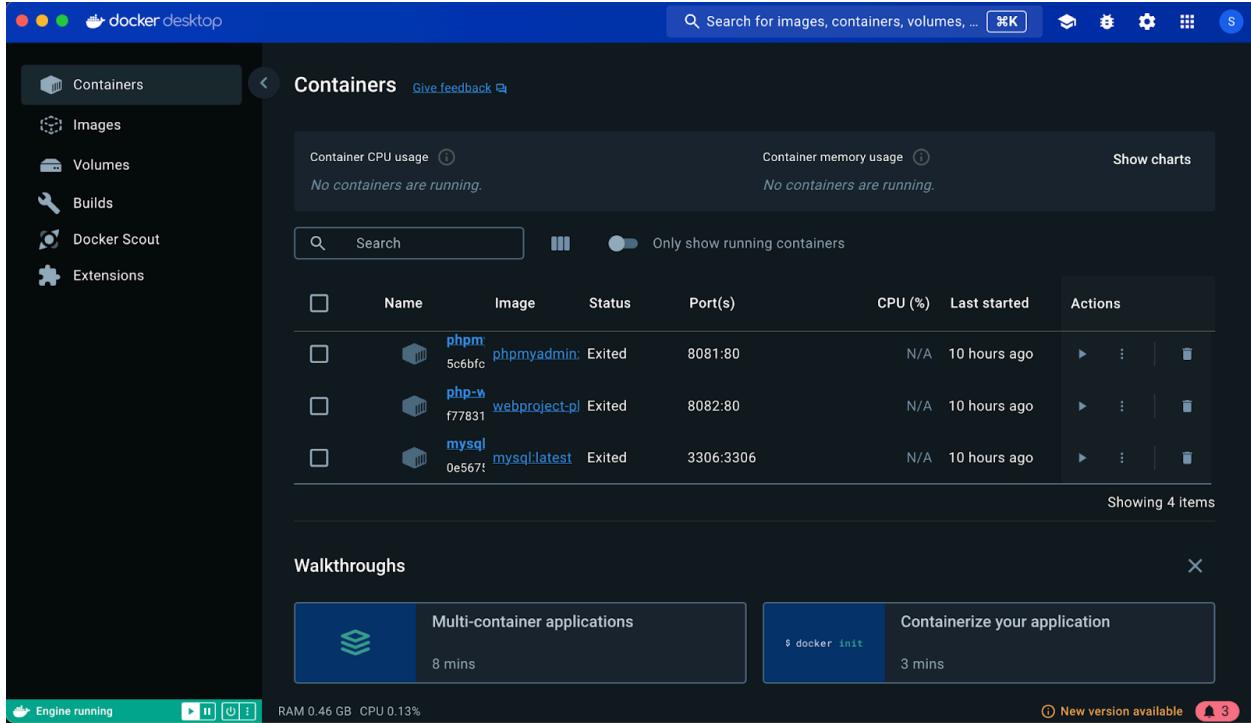
```
src > script.js > ...
1  $(document).ready(function(){
2
3      $('#search-icon').click(function(){
4          $(this).toggleClass('fa-times');
5          $('#search-box').toggleClass('active');
6      });
7
8      $('#menu').click(function(){
9          $(this).toggleClass('fa-times');
10         $('.navbar').toggleClass('nav-toggle');
11     });
12
13     $(window).on('scroll load',function(){
14
15         $('#search-icon').removeClass('fa-times');
16         $('#search-box').removeClass('active');
17
18         $('#menu').removeClass('fa-times');
19         $('.navbar').removeClass('nav-toggle');
20
21         if($(window).scrollTop() > 0){
22             $('header').addClass('sticky');
23         }else{
24             $('header').removeClass('sticky');
25         }
26
27     });
28
29     // smooth scrolling
30
31     $('a[href*="#"]').on('click',function(e){
32
33         e.preventDefault();
34
35         $('html, body').animate({
36
```

This file helps you easily run a web app with a MySQL database and phpMyAdmin to manage it locally.

```
❶ docker-compose.yaml
1   version: '3.8'
2
3   services:
4     mysql:
5       image: mysql:latest
6       container_name: mysql-wp
7       environment:
8         MYSQL_ROOT_PASSWORD: root
9         MYSQL_DATABASE: Pet
10        MYSQL_USER: sherry
11        MYSQL_PASSWORD: root
12       ports:
13         - "3306:3306"
14
15       volumes:
16         - ./data:/var/lib/mysql
17
18     php:
19       build:
20         context: .
21         dockerfile: Dockerfile
22       container_name: php-wp
23       ports:
24         - "8082:80"
25       volumes:
26         - ./src:/var/www/html
27       depends_on:
28         - mysql
29
30     phpmyadmin:
31       image: phpmyadmin:latest
32       container_name: phpmyadmin-wp
33       environment:
34         PMA_HOST: mysql
35         MYSQL_ROOT_PASSWORD: root_password
36       ports:
```

Docker is a tool that helps developers run applications in isolated environments called **containers**. These containers package

everything the app needs to run (code, libraries, etc.), making it work the same way on any system.



Conclusion:

To sum up, I used HTML, CSS, JavaScript, PHP, MySQL, and jQuery to create a pet food website. Docker containers were also used to provide a consistent and effective development environment. The website has all the necessary features, such as a contact form, an add-to-cart option, password change capability, and admin and user login. Users and administrators may enjoy a smooth and user-friendly experience as customers explore, order, and receive items with ease.

