

LAYOUT FILE

- The layout file (layout.blade.php) acts as a master template that defines the overall structure and appearance of the web pages. This file contains common elements that are shared across multiple views, such as the header, footer, and styling links. By centralizing these elements in a single layout file, it ensures consistency across the application and avoids repetition.
- **Meta Tags and Stylesheets:** For managing responsiveness and aesthetics.
- **Header:** Displayed on pages other than the login page, with navigation links to key sections (Home, Dog Breeds, About, Contact Us).
- **Footer:** Displayed on non-login pages, providing copyright information.
- **Main Content Section:** A dynamic area where individual views will inject their unique content using `@yield('content')`

This layout file is extended by specific view files using the `@extends` directive, allowing those views to inject content into the defined `@yield` sections (such as title and content).

```
Layout.blade.php X
C:\Users\PC\Herd\ProjTest\resources\views\Components> Layout.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>@yield('title')</title>
8     <link rel="stylesheet" href="{{ asset('style.css') }}">
9     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
10
11 </head>
12 <body>
13     <!-- Header -->
14     @unless (Request::is('/'))
15     <header>
16         <nav>
17             <div class="logo-container" style="display: flex; align-items: center;">
18                 <a href="index.html" class="logo">
19                     
20                 </a>
21                 <span class="logo-text">Hi, {{ session('username', 'guest') }}!</span>
22             </div>
23             <div class="nav-links">
24                 <a href="http://127.0.0.1:8000/home">Home</a>
25                 <a href="http://127.0.0.1:8000/content">Dog Breeds</a>
26                 <a href="http://127.0.0.1:8000/about">About Me</a>
27                 <a href="http://127.0.0.1:8000/contact">Contact Us</a>
28             </div>
29         </nav>
30     </header>
31     <!-- End Header -->
32 @endunless
33
34 <!-- Main Content Section -->
35 <main>
36     @yield('content')
37 </main>
38
39 @unless (Request::is('/'))
40 <footer>
41     <p>&copy; 2024 Pawsome Breeds. All rights reserved.</p>
42 </footer>
43 @endunless
44 </body>
45 </html>
```

```
Layout.blade.php X
C:\Users\PC\Herd\ProjTest\resources\views\Components> Layout.blade.php
11 <body>
12 <header>
13     <div class="logo-container" style="display: flex; align-items: center;">
14         <div class="nav-links">
15             <a href="http://127.0.0.1:8000/contact">Contact Us</a>
16         </div>
17     </div>
18 </header>
19 <!-- End Header -->
20 @endunless
21
22 <!-- Main Content Section -->
23 <main>
24     @yield('content')
25 </main>
26
27 @unless (Request::is('/'))
28 <footer>
29     <p>&copy; 2024 Pawsome Breeds. All rights reserved.</p>
30 </footer>
31 @endunless
32 </body>
33 </html>
```

ROUTES

- The routes are defined in the web.php file using the Route facade. Each route specifies a URL pattern and returns a corresponding view. For example:

Login Route (/): Displays the login page when users access the root URL (/).

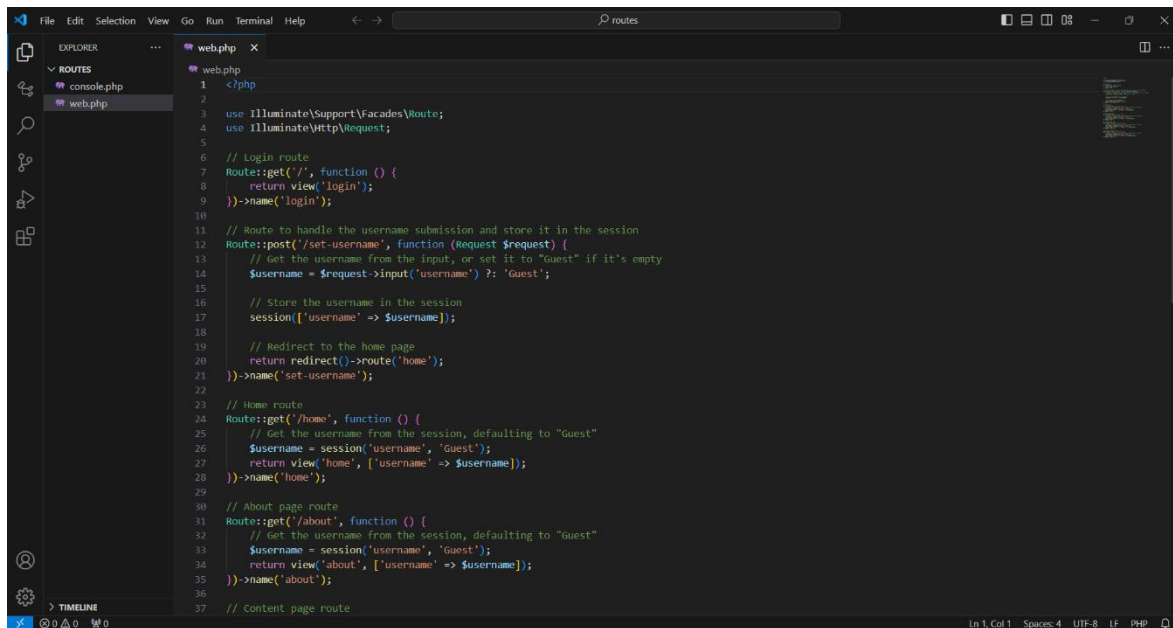
Set Username Route (/set-username): Handles form submissions from the login page, stores the username in the session, and redirects users to the home page.

Home Route (/home): Retrieves the username from the session and displays the home page view.

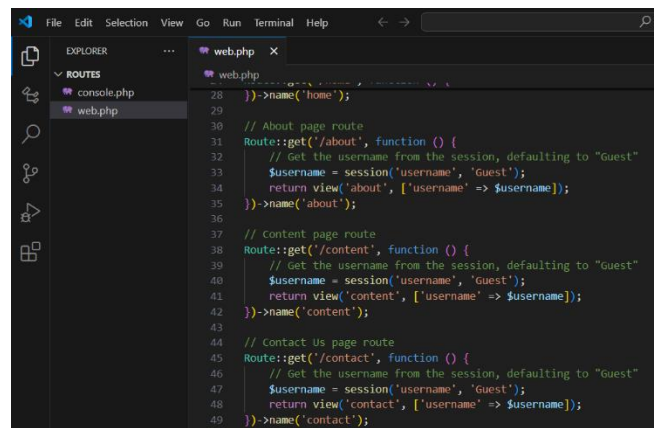
About Route (/about): Similar to the home route but serves the About Us page.

Content Route (/content): Displays a list of dog breeds, extending the layout for this specific content.

Contact Route (/contact): Displays the Contact Us page.



```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use Illuminate\Http\Request;
5
6 // Login route
7 Route::get('/', function () {
8     return view('login');
9 })->name('login');
10
11 // Route to handle the username submission and store it in the session
12 Route::post('/set-username', function (Request $request) {
13     // Get the username from the input, or set it to "Guest" if it's empty
14     $username = $request->input('username') ?: 'Guest';
15
16     // Store the username in the session
17     session(['username' => $username]);
18
19     // Redirect to the home page
20     return redirect()->route('home');
21 })->name('set-username');
22
23 // Home route
24 Route::get('/home', function () {
25     // Get the username from the session, defaulting to "Guest"
26     $username = session('username', 'Guest');
27     return view('home', ['username' => $username]);
28 })->name('home');
29
30 // About page route
31 Route::get('/about', function () {
32     // Get the username from the session, defaulting to "Guest"
33     $username = session('username', 'Guest');
34     return view('about', ['username' => $username]);
35 })->name('about');
36
37 // Content page route
```



```
28 })->name('home');
29
30 // About page route
31 Route::get('/about', function () {
32     // Get the username from the session, defaulting to "Guest"
33     $username = session('username', 'Guest');
34     return view('about', ['username' => $username]);
35 })->name('about');
36
37 // Content page route
38 Route::get('/content', function () {
39     // Get the username from the session, defaulting to "Guest"
40     $username = session('username', 'Guest');
41     return view('content', ['username' => $username]);
42 })->name('content');
43
44 // Contact Us page route
45 Route::get('/contact', function () {
46     // Get the username from the session, defaulting to "Guest"
47     $username = session('username', 'Guest');
48     return view('contact', ['username' => $username]);
49 })->name('contact');
```

VIEWS

- Each view file (like home.blade.php, about.blade.php, content.blade.php, and contact.blade.php) extends the layout file by using the `@extends('Components.layout')` directive. This ensures that all these pages share the same base layout (header, footer, and styles).

Inserting Specific Content: These view files use the `@section` directive to provide content for the layout's `@yield` sections. For example:

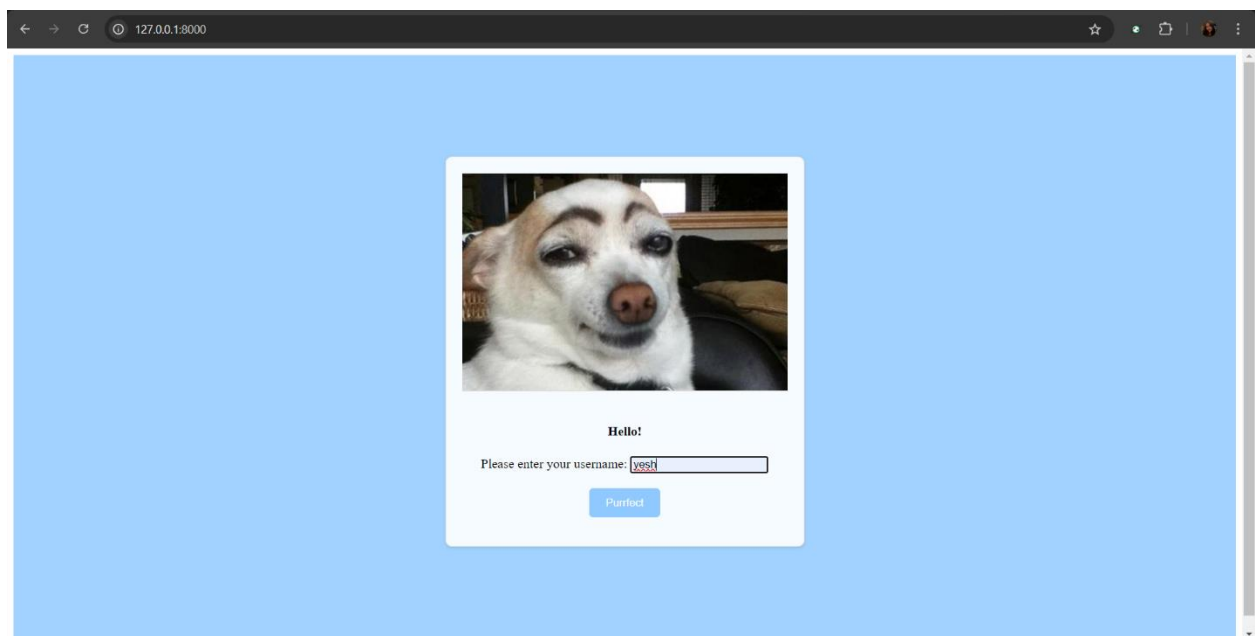
`@section('title', 'Username - Home')` defines the title for the home page.

`@section('content')` wraps the unique content for that view, which gets placed where

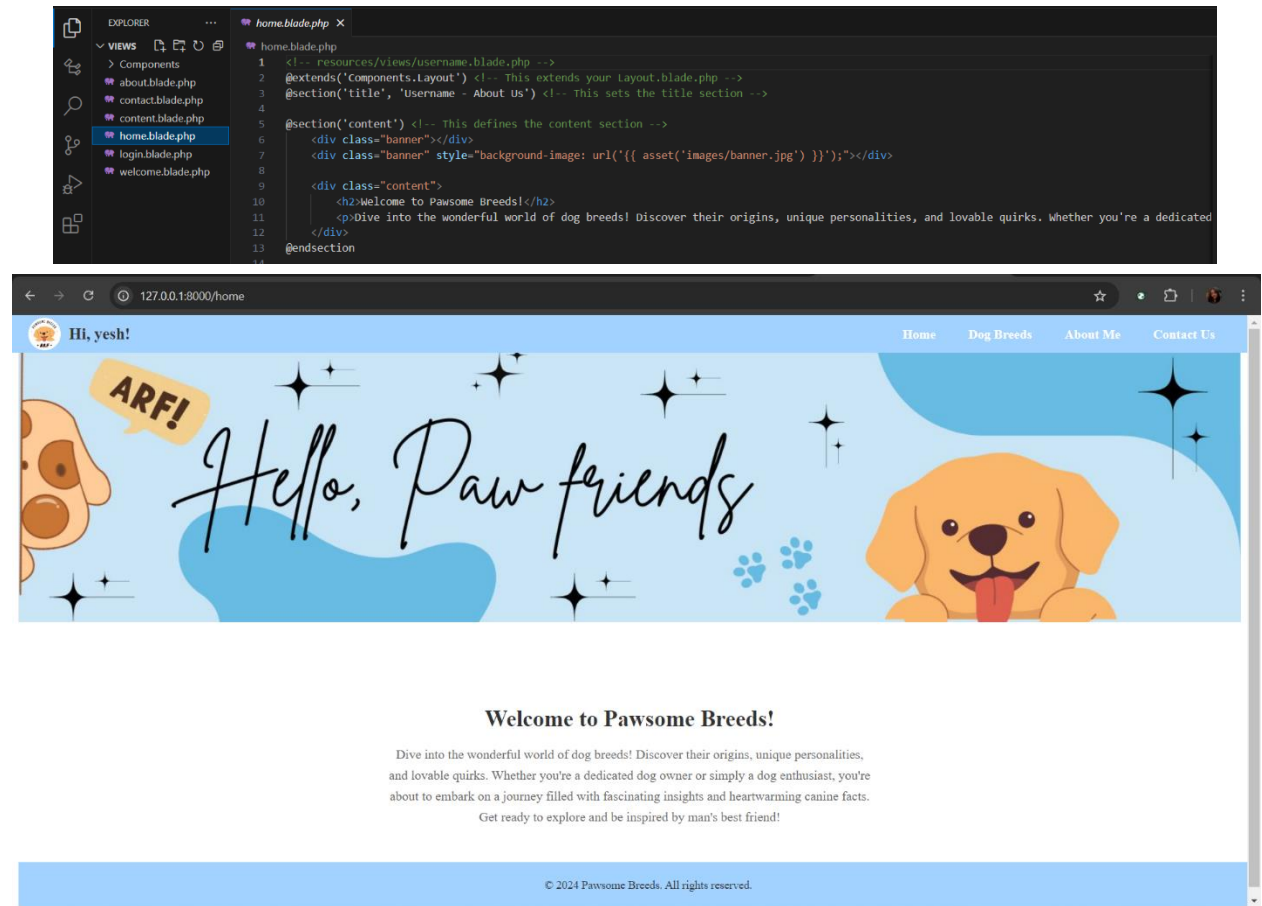
`@yield('content')` is defined in the layout file.

Login View

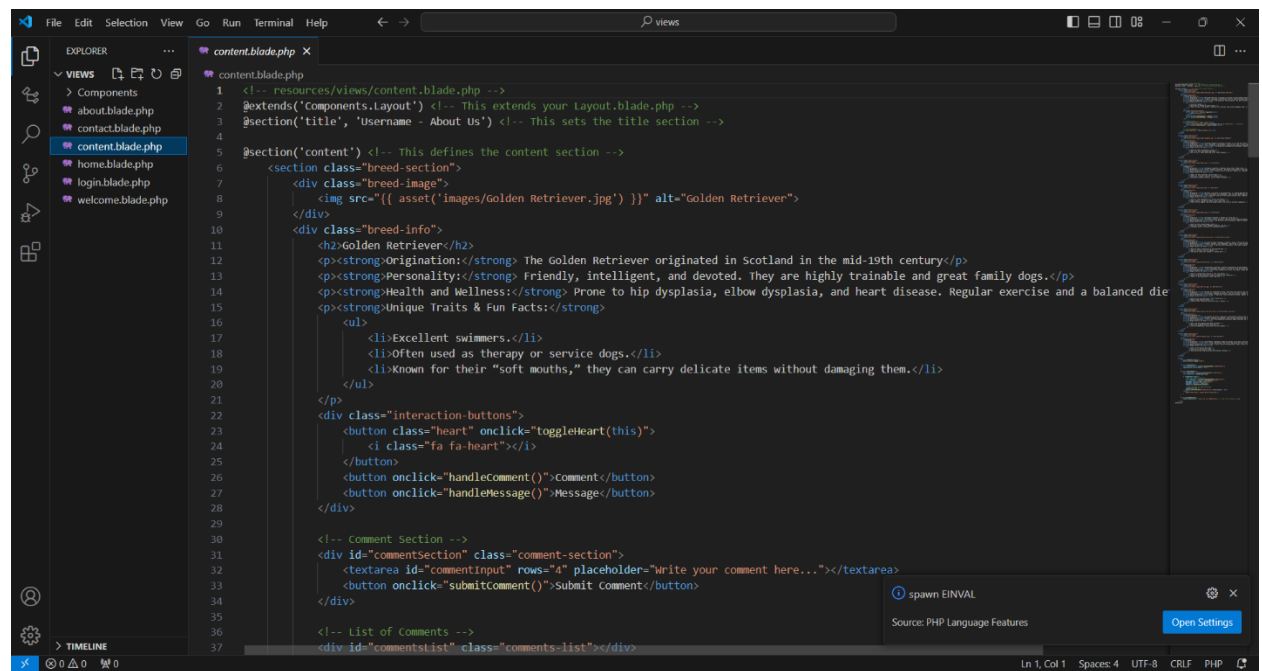
```
EXPLORER
├── VIEWS
│   ├── Components
│   │   ├── about.blade.php
│   │   ├── contact.blade.php
│   │   ├── content.blade.php
│   │   ├── home.blade.php
│   │   └── login.blade.php
│   └── welcome.blade.php
└── login.blade.php
    1 @extends('Components.Layout')
    2 @section('title', 'Username - Log in')
    3
    4 @section('content')
    5     <main class="login-page">
    6         <div class="card">
    7             
    8             <div class="card-body">
    9                 <div class="center">
   10                     <h4 class="card-title">Hello!</h4>
   11                     <form action="{{ route('set-username') }}" method="POST">
   12                         @csrf <!-- Important for security in POST requests -->
   13                         <label for="username">Please enter your username:</label>
   14                         <input type="text" id="username" name="username" value="{{ old('username') }}" required pattern="[a-zA-Z]+" title="Please enter a valid username">
   15                         <br><br>
   16                         <button type="submit" class="btn-primary">Purrfect</button>
   17                     </form>
   18                 </div>
   19             </div>
   20         </div>
   21     </main>
   22 @endsection
```



Home View



Content View



File Edit Selection View Go Run Terminal Help

views

EXPLORER

content.blade.php

content.blade.php

5 @section('content') <!-- This defines the content section -->

212 <script>

213 function toggleHeart(button) {

214 button.classList.toggle('liked');

215 }

216

217 function handleComment() {

218 const commentSection = document.getElementById('commentSection');

219 commentSection.style.display = 'block';

220 }

221

222 function submitComment() {

223 const commentInput = document.getElementById('commentInput');

224 const commentText = commentInput.value;

225

226 if (commentText.trim()) {

227 // Add the comment to the list

228 const commentList = document.getElementById('commentsList');

229 const newComment = document.createElement('div');

230 newComment.classList.add('comment-item');

231 newComment.innerText = commentText;

232 commentList.appendChild(newComment);

233

234 // Clear the input and hide the section

235 commentInput.value = '';

236 document.getElementById('commentSection').style.display = 'none';

237 } else {

238 alert('Please enter a comment before submitting.');

239 }

240 }

241

242 function handleMessage() {

243 window.location.href = 'http://127.0.0.1:8000/contact'; // Link to the "contact us" page

244 }

245 </script>


246 @endsection

247

127.0.0.1:8000/content

Hi, Nino!

Home Dog Breeds About Me Contact Us



Golden Retriever

Origination: The Golden Retriever originated in Scotland in the mid-19th century

Personality: Friendly, intelligent, and devoted. They are highly trainable and great family dogs.


Health and Wellness: Prone to hip dysplasia, elbow dysplasia, and heart disease. Regular exercise and a balanced diet are essential.

Unique Traits & Fun Facts:

- Excellent swimmers.
- Often used as therapy or service dogs.
- Known for their "soft mouths," they can carry delicate items without damaging them.

Comment

Message



German Shepherd

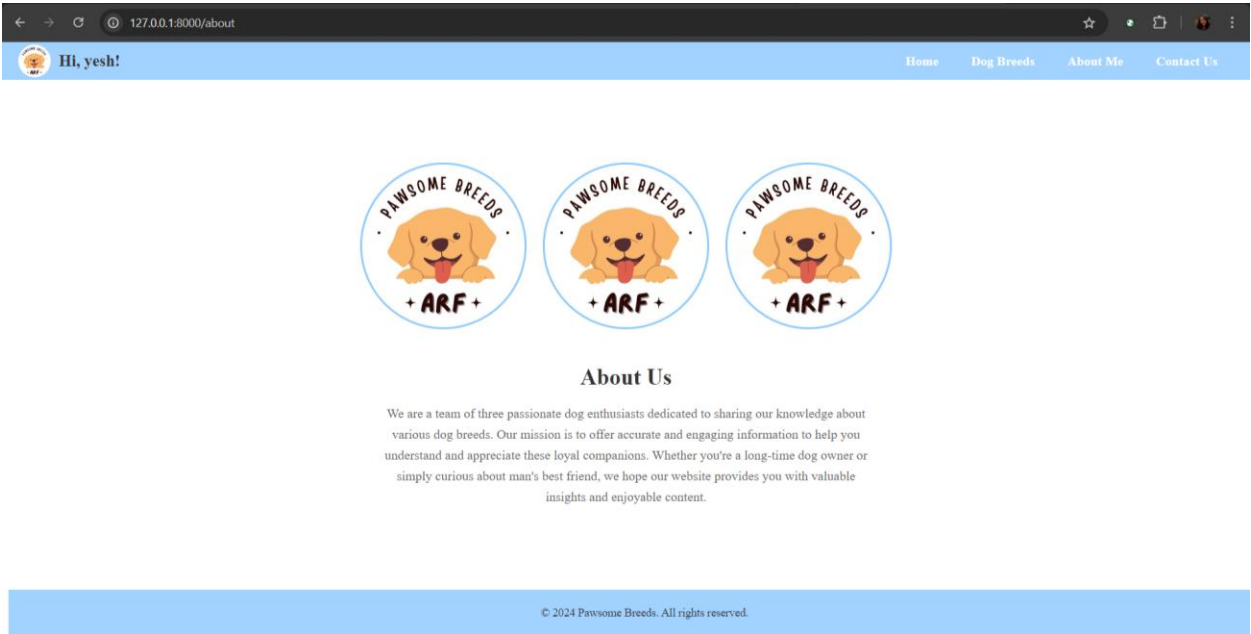
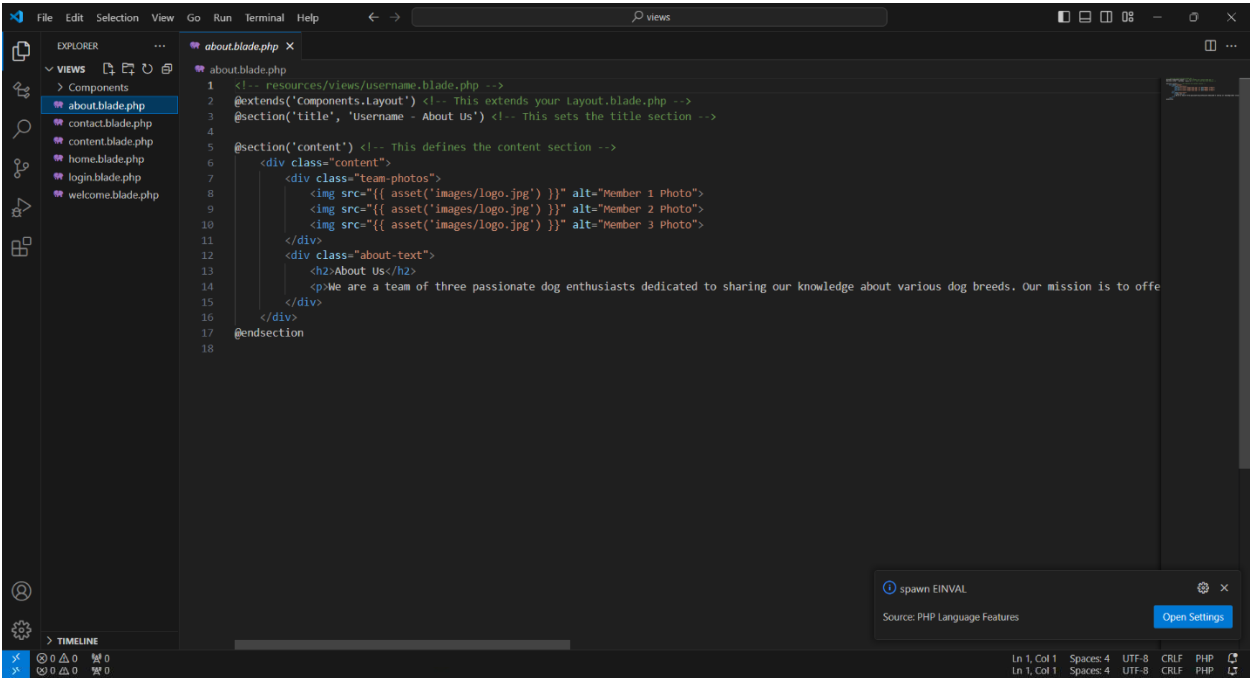
Origination: The German Shepherd originated in Germany in the late 19th century.

Personality: Loyal, courageous, and confident. They are often used in police and military roles.

Health and Wellness: Prone to hip dysplasia and degenerative myelopathy. Regular vet visits and joint care are important.

© 2024 Pawsoome Breeds. All rights reserved.

About View



Contact View

contact.blade.php

1 <!-- resources/views/contact.blade.php -->

2 @extends('components.layout') <!-- This extends your layout.blade.php -->

3 @section('title', 'Username - Contact') <!-- This sets the title section -->

4

5 @section('content') <!-- This defines the content section -->

6 <section class="contact">

7 <div class="content">

8 <h2 style="margin-top: 3cm;">Contact Us</h2>

9 <p style="color:rgb(76, 75, 75);margin-top: 20px; padding-top: 20px">Share a little about yourself, and we'll be in touch with the pawso

10 </div>

11 <div class="container">

12 <div class="contactForm">

13 <form id="contactForm">

14 <div class="inputBox">

15 Full Name

16 <input type="text" name="fullName" required>

17 </div>

18 <div class="inputBox">

19 Email

20 <input type="email" name="email" required>

21 </div>

22 <div class="inputBox">

23 Message

24 <textarea name="message" required></textarea>

25 </div>

26 <div class="inputBox_submit">

27 <input type="submit" value="Send Message">

28 </div>

29 <div id="responseMessage">Message sent!</div>

30 </form>

31 </div>

32 </div>

33 </section>

34

35 <script>

36 document.getElementById('contactForm').addEventListener('submit', function(event) {

37 event.preventDefault();

spawn EINVAL

Source: PHP Language Features

Open Settings

127.0.0.1:8000/contact

Hi, yesh!

Home Dog Breeds About Me Contact Us

Contact Us

Share a little about yourself, and we'll be in touch with the pawsome response soon

Full Name

Email

Message

Send Message

© 2024 Pawsome Breeds. All rights reserved.

Difference Between @yield and {{\$slot}}

- **@yield:** Used in layout files to define placeholders for sections of content. Child views use @section to provide content for these placeholders. Example: @yield('content') in a layout will be replaced by content from the child view's @section('content').
- **{{\$slot}}:** Used in Blade components to define a placeholder where content from the parent view is inserted. Example: In <x-alert>, content passed inside will be injected into the {{\$slot}} placeholder within the component.

@yield works with sections in layouts, while {{\$slot}} is specific to reusable Blade components.

Challenges Faced and Resolutions

1. **Dynamic Content Insertion:** There was an issue with the username not being consistently stored and displayed across different pages after login.
Resolution: The problem was solved by using the session() function in both the layout and individual views to store and retrieve the username consistently.
2. **Conditional Display of the Header and Footer:** The header and footer needed to be hidden on the login page, which caused difficulties in dynamically displaying them based on the route.
Resolution: The @unless(Request::is('/')) directive was used to conditionally hide the header and footer on the login page.