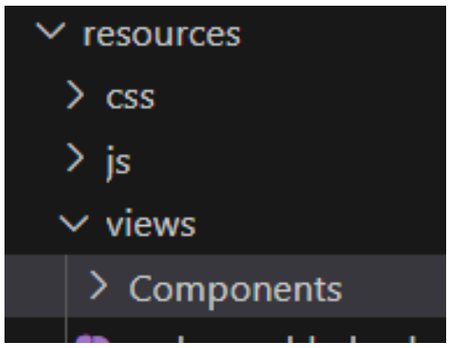
PART 1: CREATING A LAYOUT FILE

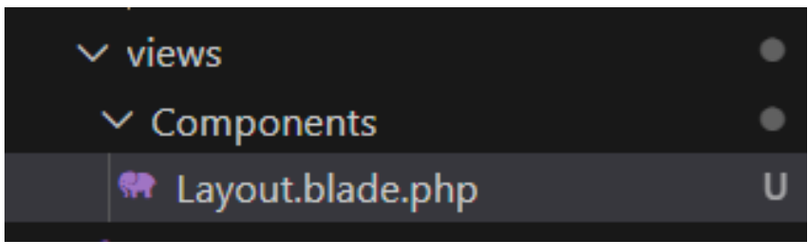
- **In the resources/views directory, create a new folder named "Components"**



EXPLANATION:

Laravel treats files in this folder as layout files templates. Laravel components are like the reusable parts of the website. This makes it easier to keep your site organized and easy to make changes.

- **Inside the "Components" folder, create a file named Layout.blade.php.**



EXPLANATION:

The purpose of layout.blade.php in the components folder is to define a reusable webpage structure that includes common elements like headers, footers and navigational links.

- **Define the basic HTML structure in Layout.blade.php:**

CODE (HTML):

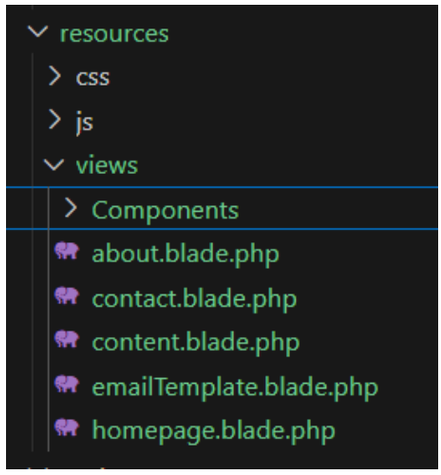
```
Layout.blade.php U
resources > views > Components > Layout.blade.php > html > body > div.content
<!DOCTYPE html>
<html lang="en">

    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Mingkai's Life (Website)</title>

        <!-- CSS Styles goes here -->
        <link rel="stylesheet" href="/css/main.css"> <!-- Default CSS -->
        @yield('additional-styles')
    </head>

    <body>

        <!-- HEADER -->
        <header>

            <div class="logo">
                <img src="/css/assets/images/logo.png" alt="Logo">
            </div>

            <nav>
                <ul>
                    <li><a href="{{ url('/') }}" class="button">Home</a></li>
                    <li><a href="{{ url('/about') }}" class="button">About</a></li>
                    <li><a href="{{ url('/content') }}"class="button">Content</a></li>
                    <li><a href="{{ url('/contact') }}" class="button">Contact</a></li>
                </ul>
            </nav>
        </header>
        <!-- END OF HEADER -->

        <!-- CONTENT -->
        <div class="content">
            <!-- Page content goes here -->
            @yield ('content')
        </div>
        <!-- END OF CONTENT -->

        <!-- FOOTER -->
        <footer>
            <p>&copy; Mingkai's Life Website.</p>
        </footer>
        <!-- END OF FOOTER -->
    </body>
</html>
```

EXPLANATION:

This HTML code defines a webpage layout template. It includes a header with a logo and navigation links, a main content area where page-specific content will be inserted, and a footer. The `@yield('additional-styles')` directive allows for additional CSS styles to be added by extending views. The `@yield('content')` directive is where the unique content of each page will be displayed. This layout provides a consistent structure for different pages on the website, making it easy to maintain a uniform design.

PART 2: CREATING VIEWS

- **In the resources/views directory, create 3 new blade files.**



EXPLANATION:

We created a 4 blade files including the homepage, about, content, and contact of the website.

Note: The `emailTemplate` is used to format the email for the Contact form to ensure it works correctly.

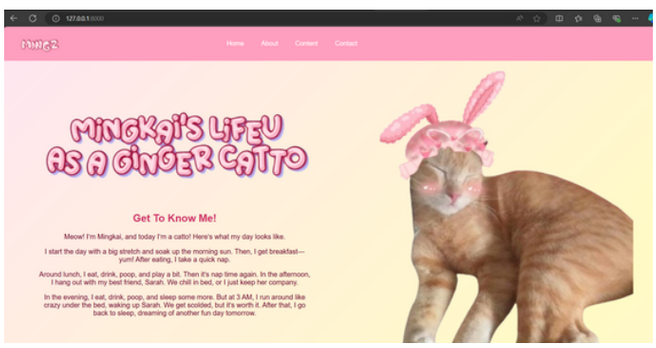
- **Each view should extend the layout file and include page-specific content.**

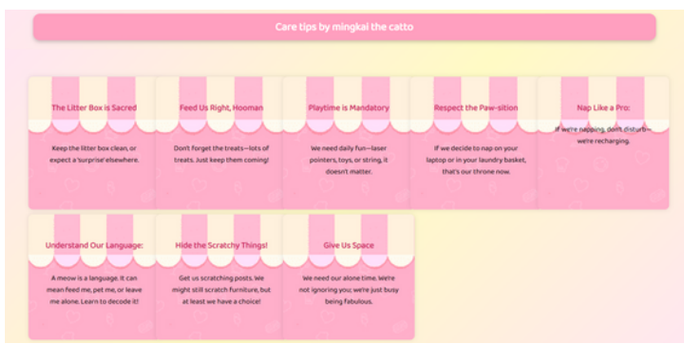
HOMEPAGE - MINGKAI

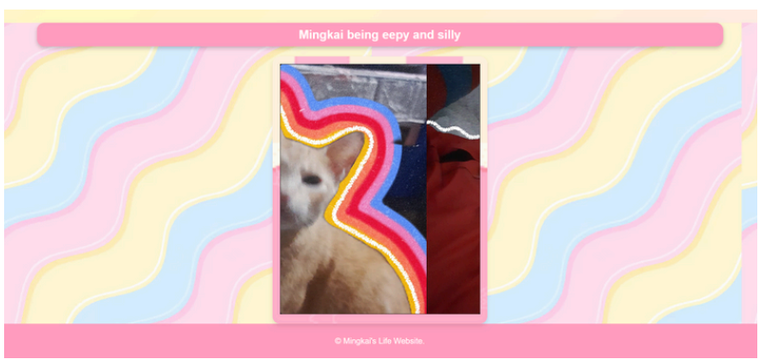
CODE (HTML):

```
homepage.blade.php U
resources > views > homepage.blade.php > ...
@extends('Components.layout')

@section('additional-styles')
<link rel="stylesheet" href="/css/home.css">
@endsection

@section('content')
        <!--SECTION OF GET TO KNOW MINGKAI-->
        <section id="image-section">···
        </section>
        <!--END SECTION OF GET TO KNOW MINGKAI-->

        <!--SECTION OF MINGKAI CARE TIPS-->
        <div style="background: linear-gradient(135deg, #ffe6f3, #fffbc5, #ffe6f3);" class="text-container cute-font">···
        </div>
        <!--END SECTION OF MINGKAI CARE TIPS-->

        <!-- Video Section -->
        <section id="video-section" class="text-container cute-font" style="background-image: url('/css/assets/images/pattern.jpg'); ">···
        </section>
        <!-- END Video Section -->···
        <audio id="background-music" src="/css/assets/audio/mizutama.mp3" preload="auto" loop></audio>···
        <script>···
        </script>
@endsection
```

OUTPUT:







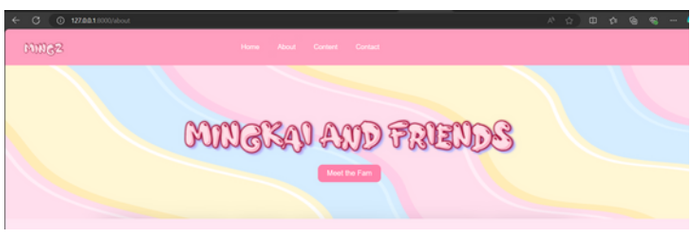
EXPLANATION:

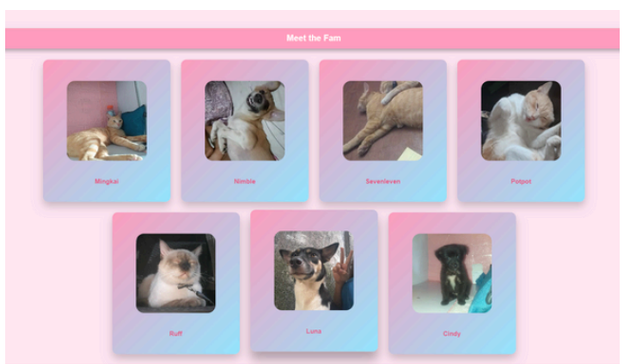
Within the layout file, we use `@yield('content')` to specify where the content from child view/in this homepage view will be inserted. In homepage blade view, we use `@extends('Components.layout')` to indicate that this view should use the `layout.blade.php` file from the `Components` folder as its base layout. We then use `@section('additional-styles')` to add any extra CSS specific to in this view, which will be included in the layout's `<head>` section. Finally, we use `@section('content')` to define the main content of the page, which will be inserted into the placeholder defined by `@yield('content')` in the layout.

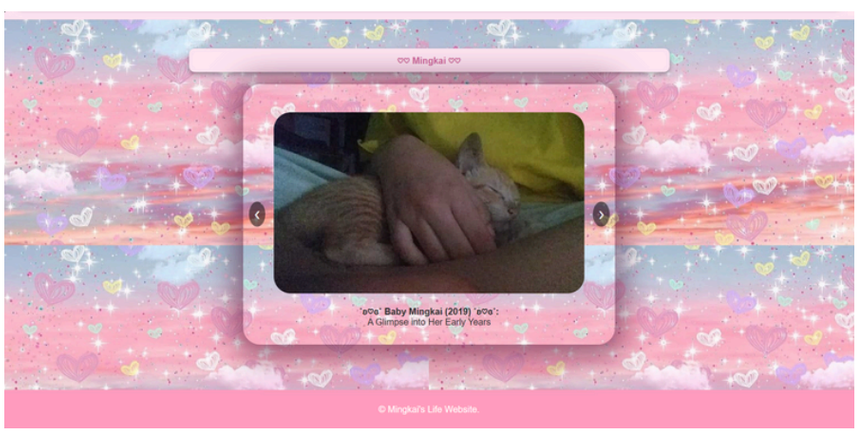
ABOUT - MINGKAI

CODE (HTML):

OUTPUT:







EXPLANATION:

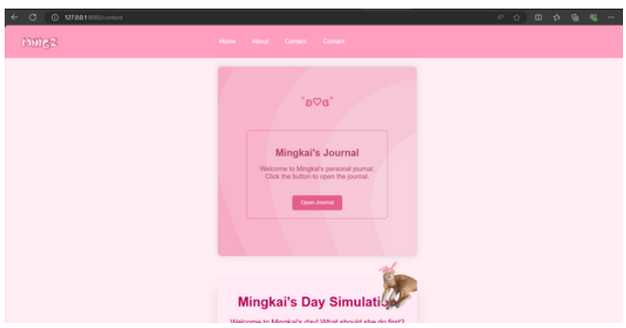
Within the layout file, we use `@yield('content')` to specify where the content from child view/in this about page view will be inserted. In about blade view, we use `@extends('Components.layout')` to indicate that this view should use the `layout.blade.php` file from the `Components` folder as its base layout. We then use `@section('additional-styles')` to add any extra CSS specific to in this view, which will be included in the layout's `<head>` section. Finally, we use `@section('content')` to define the main content of the page, which will be inserted into the placeholder defined by `@yield('content')` in the layout.

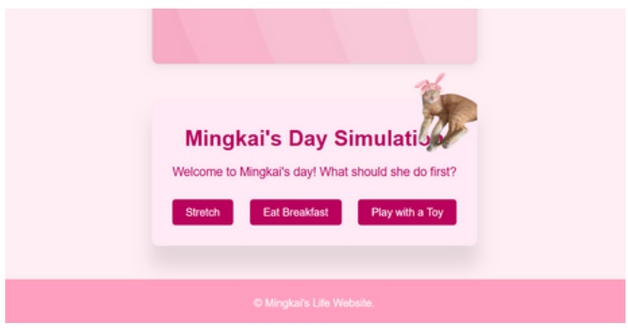
CONTENT - MINGKAI

CODE (HTML):

```
content.blade.php U
resources > views > content.blade.php > ...
@extends('Components.layout')

@section('additional-styles')
<link rel="stylesheet" href="/css/content.css">
@endsection

@section('content')
        <!-- SECTION OF MINGKAI JOURNAL -->
        <div class="journal-container">···
        </div>
        <!-- END SECTION OF MINGKAI JOURNAL -->

        <!-- SECTION OF MINGKAI SIMULATION -->
        <div class="simulation-container">···
        </div>
        <!-- END SECTION OF MINGKAI SIMULATION -->

        <!-- BACKGROUND MUSIC -->
        <audio id="background-music" src="/css/assets/audio/meowfey.mp3" preload="auto" loop></audio>

        <!-- JAVASCRIPT -->
        <script>···
        </script>
@endsection
```

OUTPUT:





EXPLANATION:

Within the layout file, we use `@yield('content')` to specify where the content from child view/in this homepage view will be inserted. In homepage blade view, we use `@extends('Components.layout')` to indicate that this view should use the `layout.blade.php` file from the `Components` folder as its base layout. We then use `@section('additional-styles')` to add any extra CSS specific to in this view, which will be included in the layout's `<head>` section. Finally, we use `@section('content')` to define the main content of the page, which will be inserted into the placeholder defined by `@yield('content')` in the layout.

CONTACT - MINGKAI

CODE (HTML):

```
contact.blade.php U
resources > views > contact.blade.php > ...
@extends('Components.layout')

@section('additional-styles')
<link rel="stylesheet" href="/css/contact.css">
@endsection

@section('content')
        <!--SECTION OF FORM  -->···
        <!--END OF SECTION OF FORM  -->

        <!-- BACKGROUND MUSIC -->
        <audio id="background-music" src="/css/assets/audio/Cats_sped_up.mp3" preload="auto" loop></audio>

        <!-- JAVASCRIPT for (Background Music and Contact form) -->
        <script>···
        </script>
@endsection
```

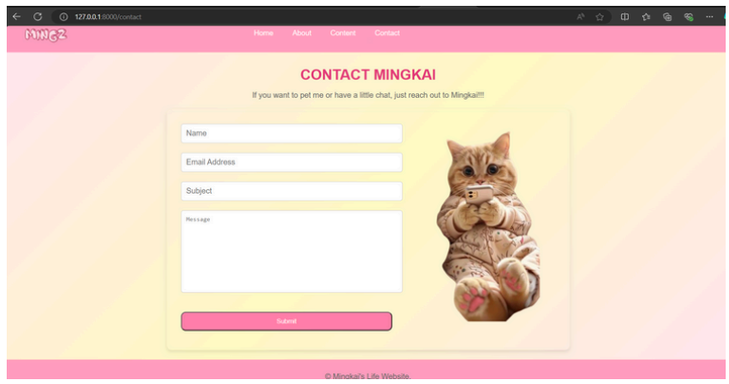
```
emailTemplate.blade.php U
resources > views > emailTemplate.blade.php > ...
@extends('Components.layout')

@section('content')
    <!--SECTION OF FORM  -->
        <p><b>Name: </b>{{ $dataReceived['name'] }}</p>
        <p><b>Email: </b>{{ $dataReceived['email'] }}</p>
        <p><b>Subject: </b>{{ $dataReceived['subject'] }}</p>
        <p><b>Message: </b>{{ $dataReceived['message'] }}</p>
    <!--END OF SECTION OF FORM  -->
@endsection
```

```
web.php M
ContactMe.php
app > Mail > ContactMe.php > ContactMe > __construct
<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;
use Illuminate\Mail\Mailable;
use Illuminate\Queue\SerializesModels;

class ContactMe extends Mailable
{
    use Queueable, SerializesModels;

    public $dataReceived;

    /**
     * Create a new message instance.
     *
     * @param array $data
     */
    public function __construct($data)
    {
        $this->dataReceived = $data;
    }

    /**
     * Build the message.
     *
     * @return $this
     */
    public function build()
    {
        return $this->view('emailTemplate');
    }
}
```

.env

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=mingkai103019@gmail.com
MAIL_PASSWORD=iyluffhncjvcomzx
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=mingkai103019@gmail.com
MAIL_FROM_NAME="${APP_NAME}"
```

OUTPUT:



EXPLANATION:

The `contact.blade.php` file contains a form that lets users send a message. It includes fields for their name, email, subject, and message. The `@csrf` directive adds a CSRF token to protect against cross-site request forgery attacks.

The `ContactMe` class defines how the email will be sent. It uses Laravel's `Mailable` class to create the email with the provided data and specifies the `emailTemplate` view to format the email content.

In `web.php`, the GET route displays the contact form when a user visits `/contact`. The POST route processes form submissions by collecting the submitted data, sending it via email using the `ContactMe` class, and then redirects the user back to the contact page with a success message.

The `.env` file is configured to send emails using Gmail's SMTP server. It includes mingkai Gmail address, password, and server details. When an email is sent from your application, it will appear to come from your Gmail address, and Gmail's servers will handle the sending securely.

PART 3: UPDATING ROUTES

- **In routes/web.php, define routes to return the views:**

CODE (ROUTE):

web.php M

routes > web.php > ...

```php
<?php

use Illuminate\Support\Facades\Route;
use Illuminate\Support\Facades\Mail;
use App\Mail\ContactMe;

Route::get('/', function () {
    return view('homepage');
});

Route::get('/about', function () {
    return view('about');
});

Route::get('/content', function () {
    return view('content');
});

Route::get('/contact', function () {
    return view('contact');
});

Route::post('/contact', function () {
    $data = request()->all();
    Mail::to('mingkai103019@gmail.com')->send(new ContactMe($data));
    return redirect('/contact')->with('flash', 'Message Sent Successfully');
});
```

EXPLANATION:

- *Routes Definitions:*

- It defines simple routes for the homepage ('/'), about page ('/about'), content page ('/content'), and contact page ('/contact'). When users visit these pages, the corresponding view is displayed.

- *Contact Form Submission:*

- When someone submits the contact form, the code collects the form data.

- It sends an email to 'mingkai103019@gmail.com' using the ContactMe class.

- After the email is sent, the user is redirected back to the contact page with a message saying "Message Sent Successfully."

PART 4: EXPLANATION

- **The purpose of the layout file and how it is used.**

  The purpose of layout.blade.php in the components folder is to define a reusable webpage structure that includes common elements like headers, footers and navigational links.

- **How each view file extends the layout and inserts specific content.**

  1. *Create a Layout File*:
     - *The layout file is stored in the Components folder inside resources/views and defines the common structure for all pages, including elements like the header, footer, and navigation links. It uses placeholders (@yield'content') where specific content will be inserted.*
  2. *Extend the Layout*:
     - *Each view file extends this layout by using @extends('Components.layout'). This means the view will use the layout's structure and styling.*
  3. *Insert Specific Content*:
     - *Within each view file, content is inserted into the layout's placeholders using @section('content') defines the main content that will be displayed in the layout's content area.*

- **Explain any challenges you faced and how you resolved them.**

  Challenge 1: Double Content of our web
  What Happened:
  - *We used both @section and <x-layout> in the views. This caused the content to appear twice on the web pages.*

  How we Fixed It:
  - *We made sure to use only one set of directives correctly. We used @extends in the views to inherit the layout, and @section in the views to add content. The layout file used @yield to show this content in the right place. This fixed the issue with duplicate content.*

  Challenge 2: Different CSS Styles in every Blade View
  What Happened:
  - *We weren't sure how to apply different styles for each page since the header is only in one place which is in the layout. If we used that every blade view, we have the same css style in which we don't want to happen.*

  How We Fixed It:
  - *We added a @yield('styles') section in the layout for page-specific styles. In each view, we used @section('styles') to include the CSS needed for that page. This way, each page gets its own styles without affecting the others.*

- **Explore the difference between {{$slot}} and @yield**

{{$slot}}
- *Context: Used in Blade component views.*
- *Purpose: Represents the content passed into a component.*
- *Usage: When you create a Blade component and want to insert content into that component, you use {{$slot}} in the component's view. The content between the component tags in the parent view will be placed where {{$slot}} is used in the component.*

Example:

```blade
<div class="card">
    {{ $slot }}
</div>
```

```blade
<x-card>
    <h2>Card Title</h2>
    <p>Some content for the card.</p>
</x-card>
```

@yield
- *Context: Used in Blade layouts.*
- *Purpose: Defines a section of content that can be filled in by child views.*
- *Usage: When creating a layout file, you define sections using @yield. Child views can then extend this layout and provide content for those sections using @section.*

Example:

```blade
<!DOCTYPE html>
<html>
<head>
    <title>Mingkai Website</title>
    @yield('additional-styles')
</head>
<body>
    @yield('content')
</body>
</html>
```

```blade
@extends('Components.layout')

@section('additional-styles')

@endsection

@section('content')
    <h1>Welcome to Mingkai Page</h1>
@endsection
```