

THE PURPOSE OF KEY DIRECTORIES AND FILES IN THE LARAVEL PROJECT.

1. **'app/'**: Contains the core code of application.

Key Subdirectories/Files:

- a. **'Http/'**: Contains controllers, middleware, and request handling.
- b. **'Models/'**: Houses your Eloquent models (database entities).
- c. **'Providers/'**: Service providers to register your application services and configuration.

2. **'bootstrap/'**: Initializes and configures the Laravel framework.

Key Subdirectories/Files:

- a. **'cache/'**: Stores cached files for performance.
- b. **'app.php'**: Boots up the framework.
- c. **'providers.php'**: Caches service providers for faster loading.

3. **'config/'**: Contains all of the application's configuration files, such as `'app.php'`, `'database.php'`, and `'mail.php'` and many more. This is also where you configure settings like timezone, app name, database connections, mail services, and other application settings.

4. **'database/'**: Contains database-related files and configurations.

Key Subdirectories/ Files:

- a. **'factories/'**: Generates sample data for models.
- b. **'migrations/'**: Manages database schema changes.
- c. **'seeders/'**: Populates the database with data.
- d. **'gitignore'**: Excludes certain database files (like SQLite) from version control.
- e. **'database.sqlite'**: A local SQLite database file used for development and testing

5. **'public/'**: The web server's root directory. It contains publicly accessible files such as HTML, images, CSS, JavaScript, and your app's entry point (`'index.php'`).

6. **'resources/'**: Contains all the frontend resources like views, language files, and uncompiled assets (CSS, JavaScript).

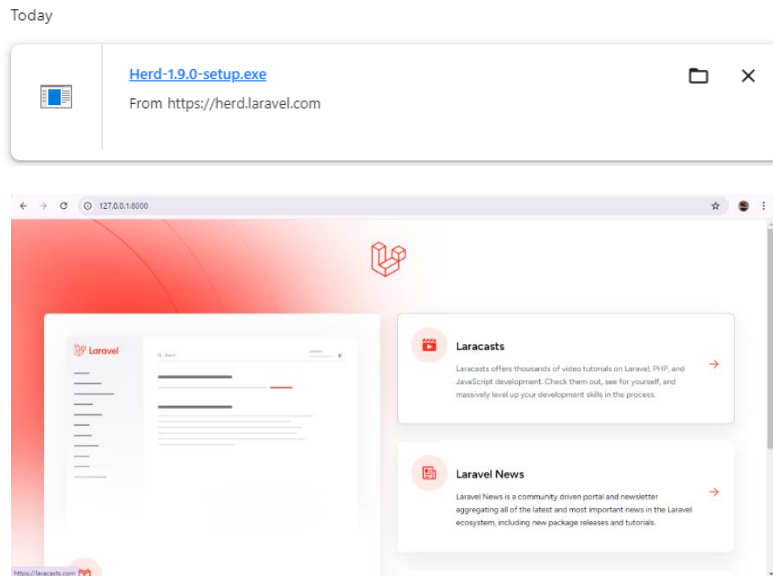
Key Subdirectories/Files:

- a. **'views/'**: Stores Blade template files that represent the HTML structure of your app.

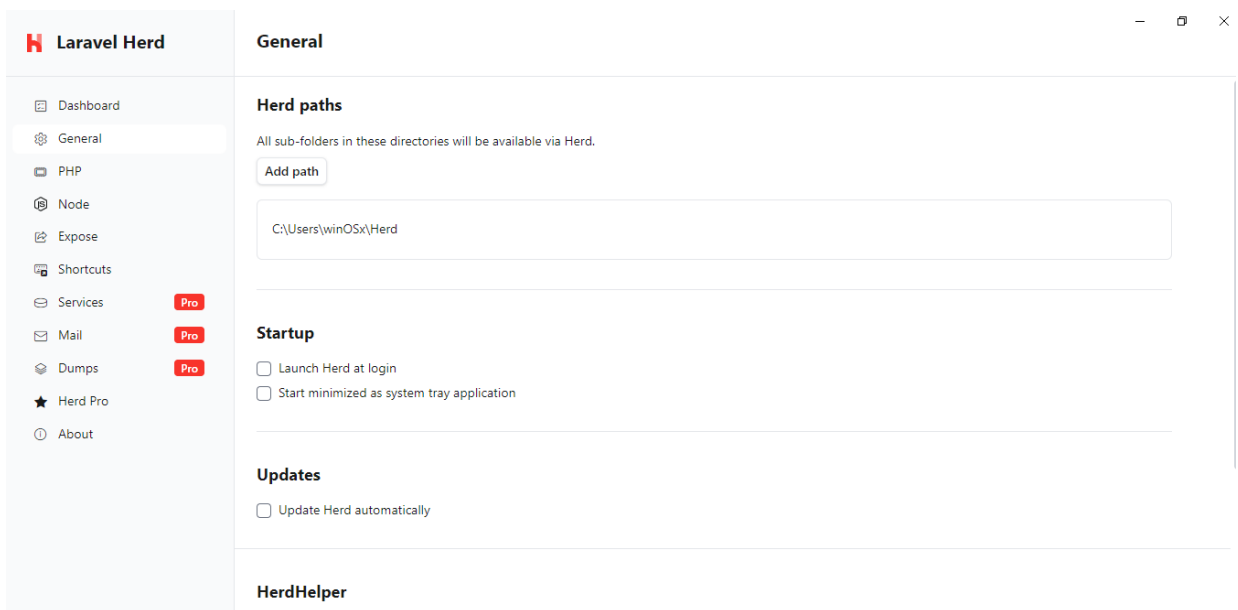
- b. `'css/'` and `'js/'`: Holds the uncompiled assets such as CSS and JavaScript before they're processed.
- 7. **`'routes/'`**: Defines all the routes for the application. These routes tell Laravel which controller and action to invoke when a particular URL is accessed.
Key Subdirectories/Files:
 - a. `'web.php'`: Contains routes for web (browser) requests.
 - b. `'console.php'`: Defines all console-based routes.
- 8. **`'storage/'`**: Contains files generated by the application such as logs, cached views, and file uploads.
Key Subdirectories/Files:
 - a. `'app/'`: Storage for files that are generated by the app.
 - b. `'framework/'`: Caches and temporary files used by the framework.
 - c. `'logs/'`: Stores log files generated by the application.
- 9. **`'tests/'`**: Contains the test files for the application.
Key Subdirectories/Files:
 - a. `'Feature/'`: Contains tests for larger application features.
 - b. `'Unit/'`: Contains tests for individual units of the application.
 - c. `'pest.php'`: Configuration file for Pest, a testing framework used in Laravel.
 - d. `'TestCase.php'`: Base test class that other test classes extend, providing shared setup and configuration for tests.
- 10. **`'vendor/'`**: Contains all the composer dependencies of the project. Laravel's core files and other third-party libraries are placed here.

DOCUMENTATION OF EACH STEP (FROM INSTALLATION TO NAVIGATION OF THE PAGES)

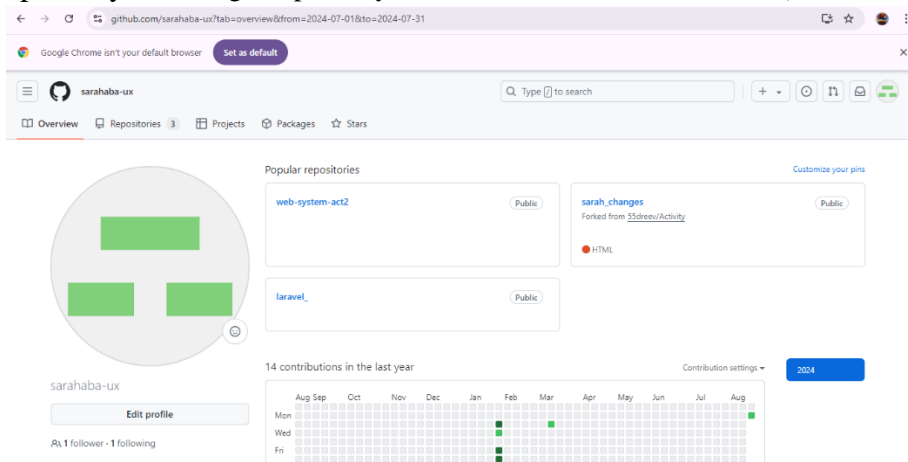
- Installation of Laravel



- Configuration



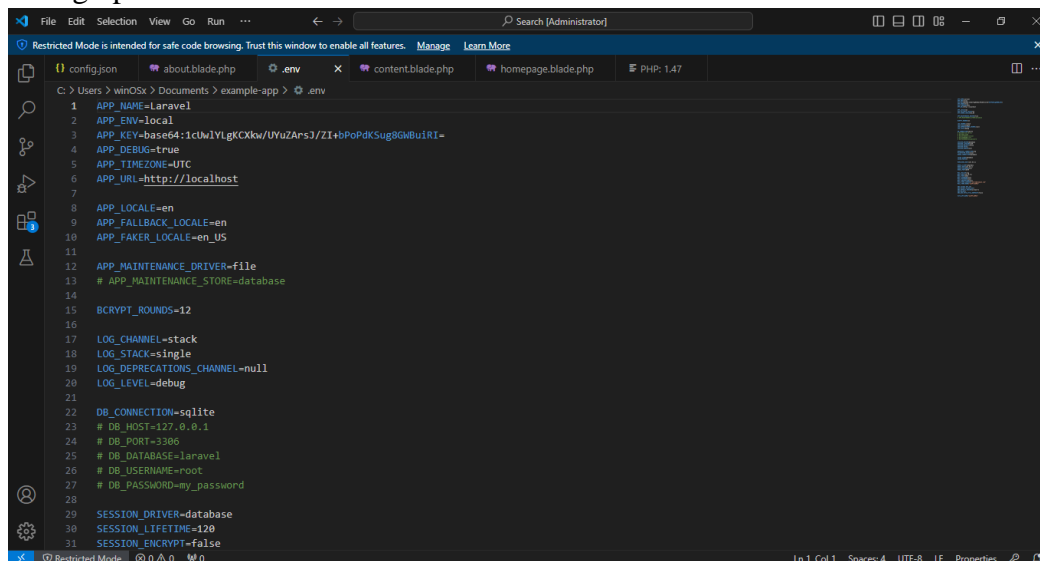
- Creating a Laravel Project and Putting it in a Git Repository (+Adding Remote Repository: Creating a repository on GitHub and add it as a remote)







```
cd projectName
cd Documents
cd projectName
git init
git add .
git commit -m "Initial commit"
git remote add origin <https://github.com/sarahaba-ux/laravel_~>
git remote add origin <github.com/sarahaba-ux/laravel>
git remote add origin https://github.com/sarahaba-ux/laravel.git
git push -u origin master
git remote remove origin
git remote add origin https://github.com/sarahaba-ux/laravel.git
git remote -v
git push -u origin master
git remote remove origin
git remote add origin https://github.com/sarahaba-ux/laravel_.git
git remote -v
git push -u origin master
php artisan key:generate
php artisan migrate
php artisan serve
php -v
php -v
php -v
resources/views/about.blade.php

win05x@DESKTOP-8PT848S MINGW64 ~ (sarah)
$
```

- Setting up the .env file to include database connection details



- Creating Views (Homepage, About, Content)

This PC > Local Disk (C:) > Users > winOSx > Documents > projectName > resources > views				
Name	Date modified	Type	Size	
 about.blade	19/08/2024 10:33 pm	PHP Source File	4 KB	
 content.blade	19/08/2024 6:07 pm	PHP Source File	1 KB	
 homepage.blade	19/08/2024 6:08 pm	PHP Source File	1 KB	
 welcome.blade	16/07/2024 7:39 am	PHP Source File	36 KB	

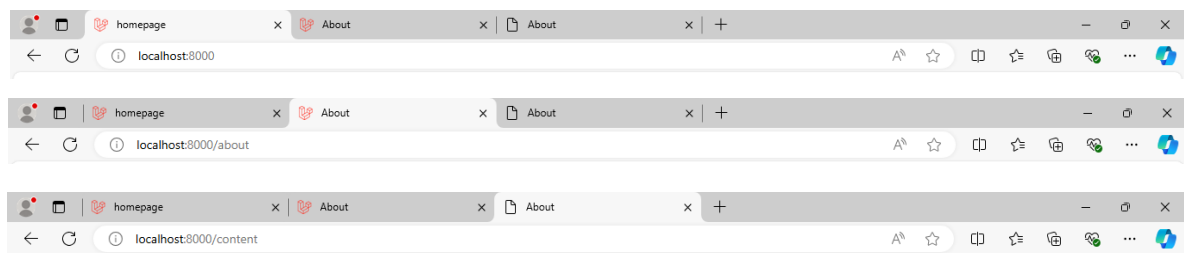
- Create and Run Routes

```
web.php x PHP: 1.50
C: > Users > winOSx > Documents > projectName > routes > web.php
1  {?php
2
3  use Illuminate\Support\Facades\Route;
4
5  Route::get('/', function () {
6      return view('homepage');
7  });
8
9  Route::get('/about', function () {
10     return view('about');
11 });
12
13 Route::get('/content', function () {
14     return view('content');
15 });
16
```

- Start the Laravel Server

```
php artisan serve
```

- Test Routes (Visiting URLs in my browser)
localhost:8000 or 127.0.0.1:8000



✚ Editing the Routes (web.php) to build connections

- The `->name(...)` method is a way to give a route a specific name, making URL generation, redirection, and code maintenance easier and more flexible.

```
web.php x about.blade.php content.blade.php
C: > Users > winOSx > Documents > projectName > routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  Route::get('/', function () {
6      return view('homepage');
7  }->name('homepage');
8
9  Route::get('/about', function () {
10     return view('about');
11 }->name('about');
12
13 Route::get('/content', function () {
14     return view('content');
15 }->name('content');
```

✚ Navigating the Pages: The navigation bar on my homepage that allow me to navigate to the Home, About, and Content pages.

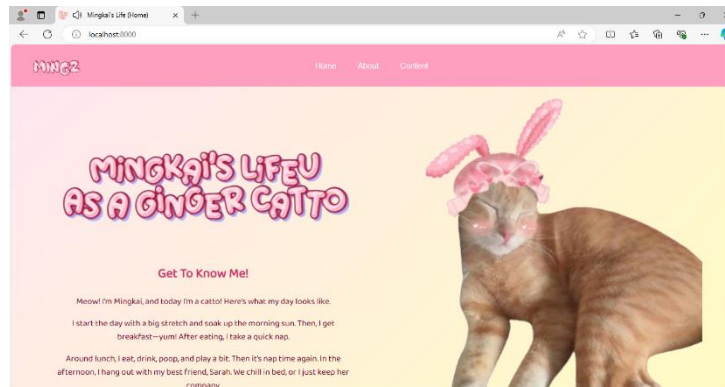
```
<nav>
<ul>
<li><a href="{{ route('homepage') }}" class="button">Home</a></li>
<li><a href="{{ route('about') }}" class="button">About</a></li>
<li><a href="{{ route('content') }}" class="button">Content</a></li>
</ul>
</nav>
```

• Buttons

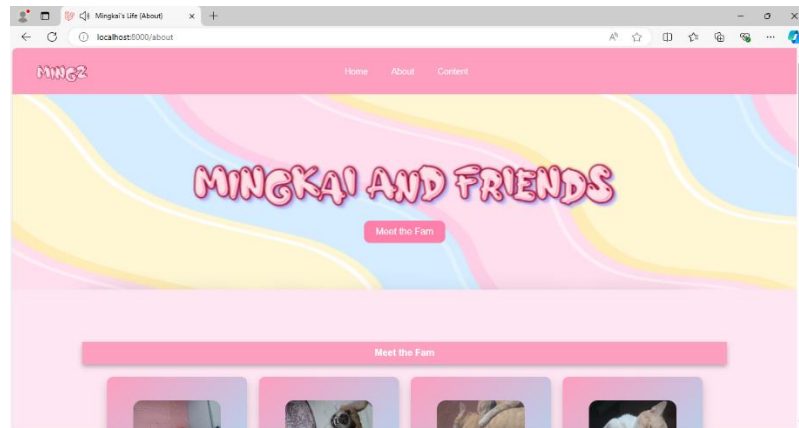


✚ Final Output for:

- Home



○ About



○ Content

