

WEB DEVELOPMENT

BARIAS, LAIZA MAE B.

BSIT-3C

08-26-2024

LAB 1: Identify and document the purpose of key directories and files in the laravel project.

app/

- **Purpose:** Contains the core application code. This includes the models, controllers, and other classes that make up the application.

Sub-directories:

Http/- Handles the incoming web requests and responses.

Models/- Contains the application's data models, which represent the structure and relationships of the data in the database.

Providers/- Contains service providers that are used to configure and bootstrap the various components of the application.

bootstrap/

- **Purpose:** Contains files for bootstrapping the Laravel framework.

Sub-directories:

cache/- Used to store framework-generated files that improve performance by reducing the need to recompile or regenerate them every time the application runs.

config/

- **Purpose:** Contains configuration files for various services and application settings.

database/

- **Purpose:** Contains files related to database management.

Sub-directories:

factories/- Houses factory classes used to generate fake data/model instances in tests.

migrations/- Contains migration files that define the structure of the database.

seeders/- Contains seed classes to populate the database with initial data.

File:

database.sqlite- SQLite database file used for local development or testing, allowing the Laravel application to store and retrieve data without needing a full database server.

public/

- **Purpose:** The entry point for web requests. It contains the index.php file, which initializes the Laravel application and serves static assets like CSS, JavaScript, and images.

resources/

- **Purpose:** Houses all views, raw assets (CSS, JS), and language files.

Sub-directories:

views/- Blade template files for rendering HTML.

js/- JavaScript files.

css/- CSS files.

routes/

- **Purpose:** Defines all routes for the application. Laravel automatically loads these files.

Files:

web.php- Contains routes for web interfaces.

console.php- Defines console-based routes.

storage/

- **Purpose:** Stores logs, compiled Blade templates, file-based sessions, and other files generated by the application.

Sub-directories:

app/- Stores application-specific files/ files uploaded by the user.

framework/- Stores framework-generated files, including sessions and cache.

logs/- Contains application logs.

tests/

- **Purpose:** Contains automated tests for the application.

Sub-directories:

Feature/- Feature tests for testing HTTP requests and responses.

Unit/- Unit tests for testing individual components.

vendor/

- **Purpose:** Contains third-party packages and dependencies managed by Composer. Laravel itself is included here.

Key File:

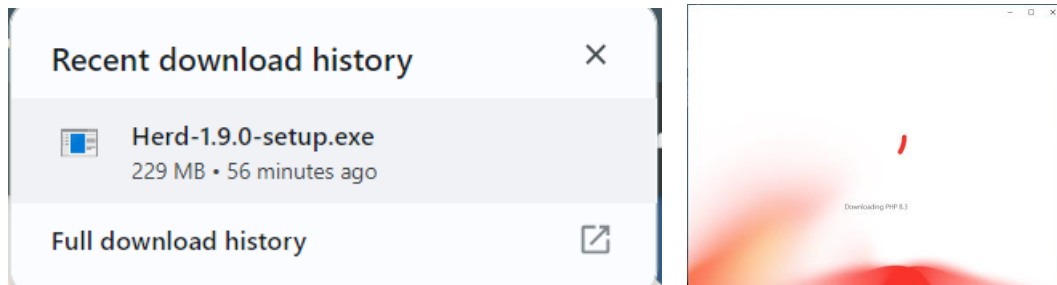
autoload.php- Autoloader for project dependencies.

Key Files in the Root Directory:

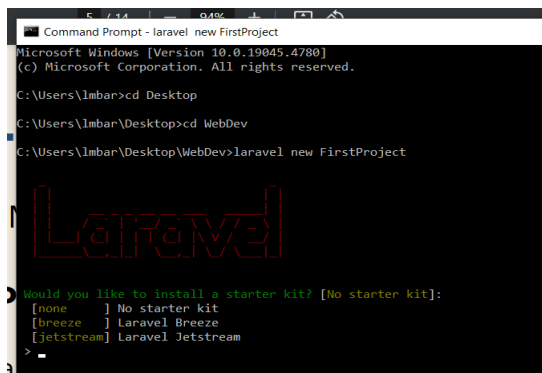
- **.editorconfig**- Standardizes coding styles across different editors and IDEs by specifying formatting rules.
- **.env**- Stores environment-specific configuration variables like database credentials and application keys.
- **artisan**- Provides a command-line interface for managing various aspects of a Laravel application.
- **composer.json**- Defines PHP dependencies and project configuration for Composer, the PHP package manager.
- **composer.lock**- Locks the versions of dependencies to ensure consistent installations across environments.
- **package.json**- Manages JavaScript dependencies and scripts for the frontend of the application.
- **phpunit.xml**- Configures PHPUnit settings for running tests in the Laravel application.
- **vite.config.js**- Configures Vite for building and bundling frontend assets like JavaScript and CSS.

DOCUMENTATION

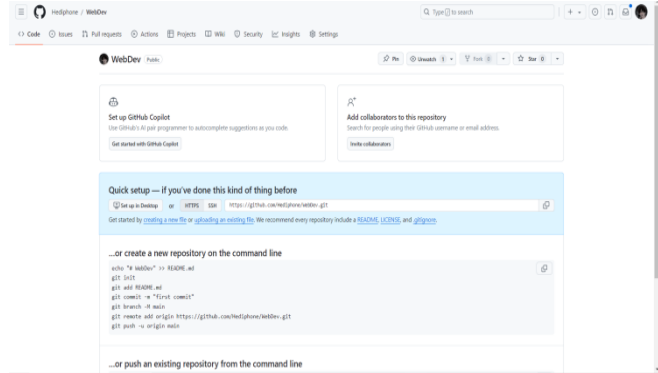
1. INSTALL LARAVEL ALONG WITH ITS DEPENDENCIES.



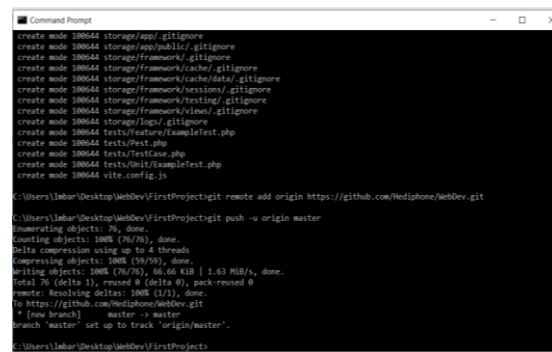
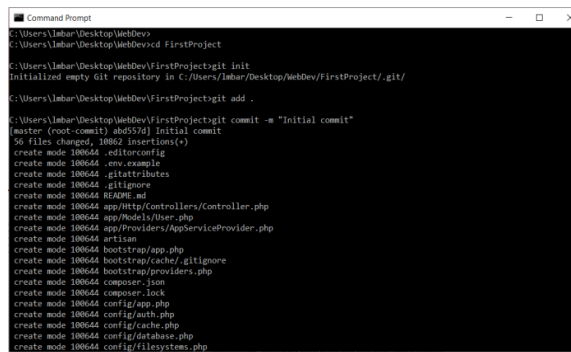
2. CREATE A LARAVEL PROJECT AND PUT IT IN A GIT REPOSITORY



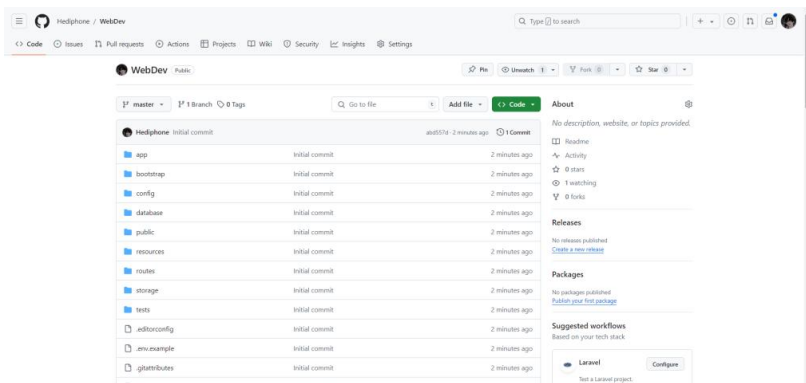
Create Laravel Project (FirstProject)



Create Repository



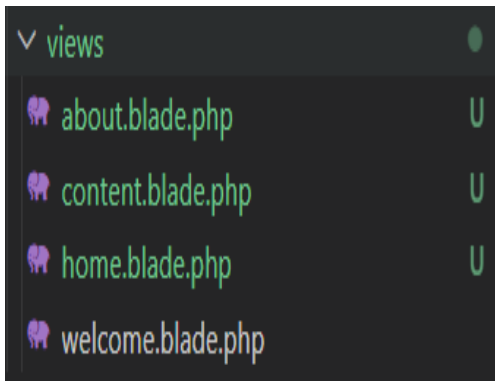
Add Project to
Git Repository



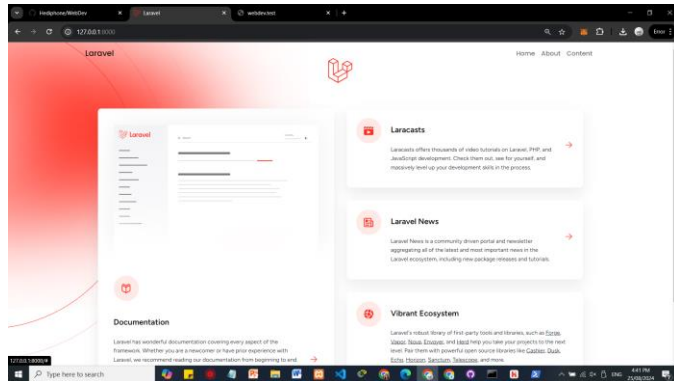
3. SET UP THE .ENV FILE TO INCLUDE DATABASE CONNECTION DETAILS

```
21
22 DB_CONNECTION=sqlite
23 DB_DATABASE=C:\Users\lmba\Desktop\WebDev\FirstProject\database\database.sqlite
24 DB_FOREIGN_KEYS=true
25 # DB_HOST=127.0.0.1
26 # DB_PORT=3306
27 # DB_DATABASE=laravel
28 # DB_USERNAME=root
29 # DB_PASSWORD=
30
```

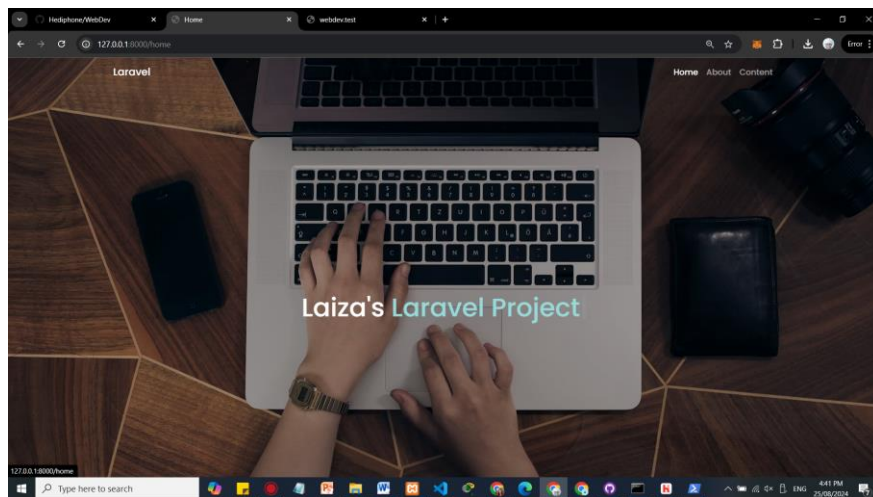
4. CREATE AT LEAST 3 VIEWS (HOMEPAGE, ABOUT, CONTENT)



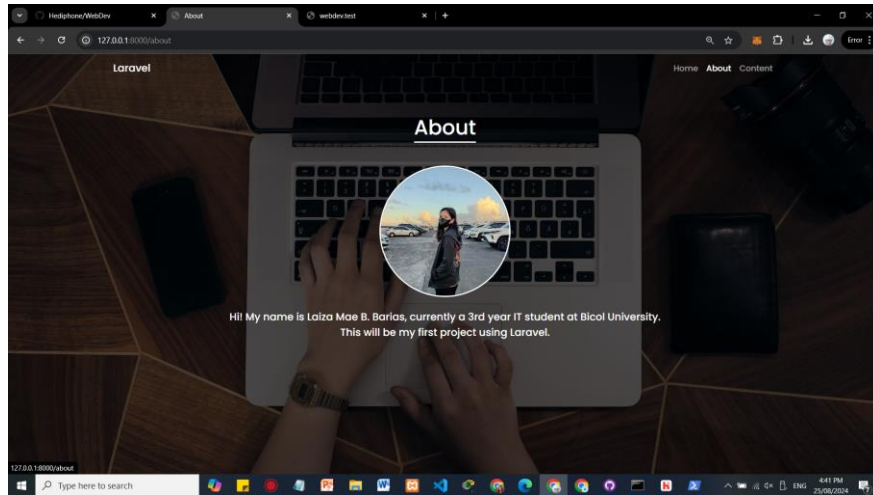
Create Views



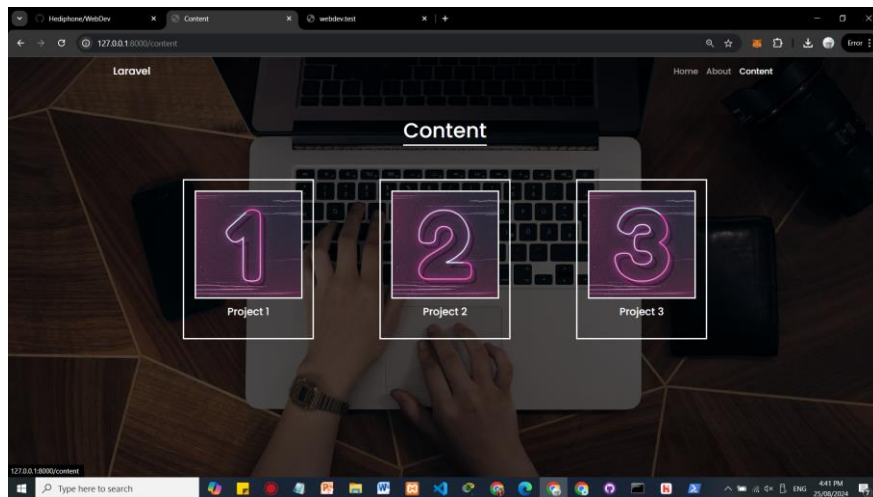
Laravel (welcome.blade.php)



Home (home.blade.php)



About (about.blade.php)



Content (content.blade.php)

5. CREATE AND RUN ROUTES

```

welcome.blade.php 1M, 1M      web.php 1M, 1M  X
FirstProject > routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  Route::get('/', function () {
6      return view('welcome');
7  });
8
9  Route::get('/home', function () {
10     return view('home');
11 });
12
13 Route::get('/about', function () {
14     return view('about');
15 });
16
17 Route::get('/content', function () {
18     return view('content');
19 });

```