WEB DEVELOPMENT

**LAB1**

**BONGAT, MAE ANNE B. BSIT-3C 08/26/2024**

* Identify and document the purpose of key directories and files in the Laravel project.

1. **`app/`**

* Contains the core code of the application.
  + Subdirectories and Files
    - `Console/`: Contains console commands for Artisan CLI.
    - `Exceptions/`: Contains application-specific exception classes.
    - `Http/`: Contains controllers, middleware, and form requests.
    - `Models/`: Contains Eloquent models representing database tables.
    - `Providers/`: Contains service providers which are used to bind services into the service container.

1. **`bootstrap/`**

* Contains files for bootstrapping the application.
  + Subdirectories and Files-
    - `app.php`: The file that initializes the Laravel application.
    - `cache/`: Contains cache files for faster application bootstrapping.

1. **`config/`**

* Contains configuration files for the application.
  + Files:
    - `app.php`: Configuration for the application’s general settings.
    - `database.php`: Database connection settings.
    - `mail.php`: Configuration for mail settings.
    - `queue.php`: Queue configuration settings.

1. **`database/`**

* Contains database-related files.
  + Subdirectories and Files-
    - `migrations/`: Contains migration files for managing database schema changes.
    - `factories/`: Contains model factory files for seeding the database.
    - `seeders/`: Contains database seed classes for populating the database.

1. **`public/`**

* Contains publicly accessible files, including assets.
  + Files and Subdirectories-
    - `index.php`: The entry point for all requests into the application.
    - `css/`: Contains CSS files.
    - `js/`: Contains JavaScript files.
    - `images/`: Contains image files.

1. **`resources/`**

* Contains resources for the application.
  + Subdirectories and Files-
    - `views/`: Contains Blade templates used to render HTML.
    - `lang/`: Contains language files for localization.
    - `sass/`: Contains SASS files for styling (if using Laravel Mix).

1. **`routes/`**

* Contains route definitions.
  + Files-
    - `web.php`: Routes for web requests.
    - `api.php`: Routes for API requests.
    - `console.php`: Routes for Artisan commands.
    - `channels.php`: Channels for broadcasting.

1. **`storage/`**

* Contains files generated by the application, such as logs and cached files.
  + Subdirectories-
    - `app/`: Application files and generated files.
    - `framework/`: Framework-generated files (e.g., cache, sessions).
    - `logs/`: Application log files.

1. **`tests/`**

* Contains test files.
  + Subdirectories and Files-
    - -`Feature/`: Contains feature tests.
    - `Unit/`: Contains unit tests.

1. **`vendor/`**

* Contains third-party packages and Laravel framework files.

\*\*This directory is created by Composer and should not be modified directly.

1. **`.env`**

* Contains environment-specific configuration settings (e.g., database credentials, application key).

\*\*This file should not be committed to version control and is typically included in `.gitignore`.

1. **`composer.json`**

* Defines project dependencies and scripts.

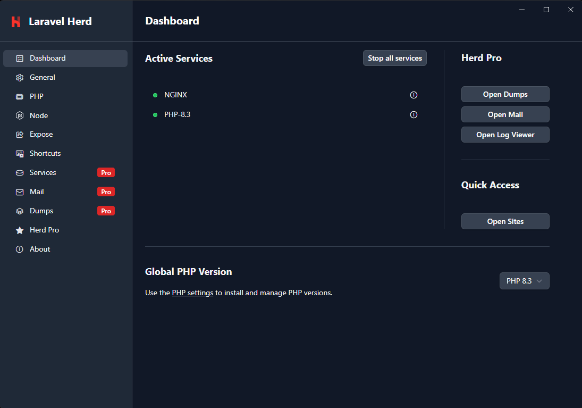
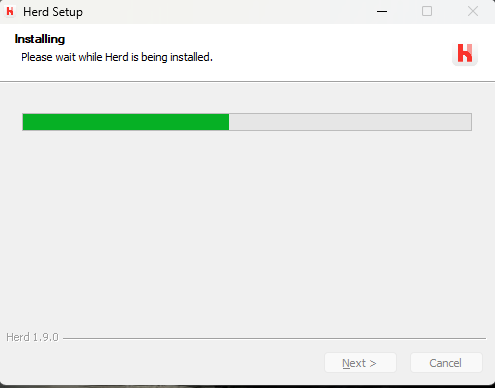
\*\*Used by Composer to manage PHP package dependencies.

1. **`artisan`**

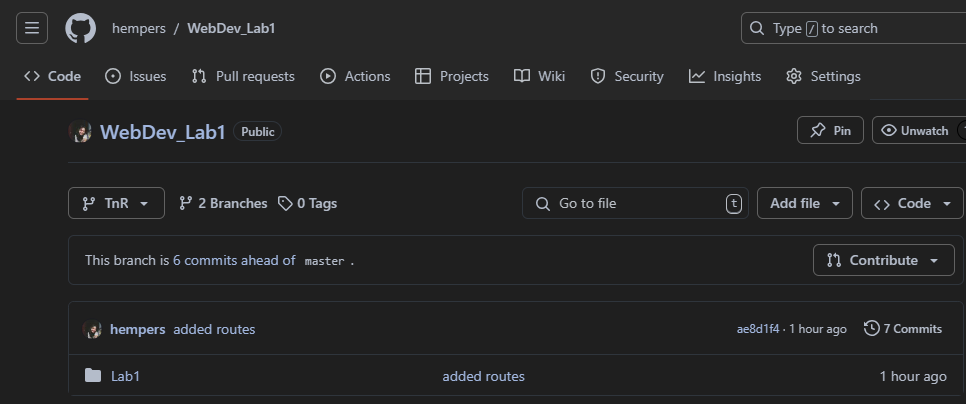
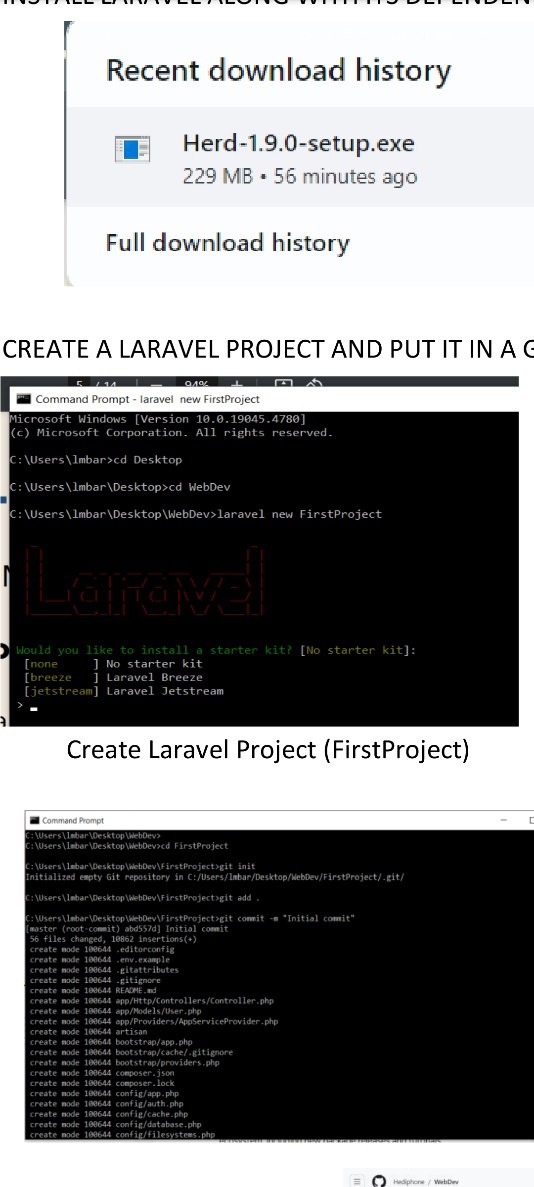
* The command-line interface for Laravel. It provides commands for tasks like running migrations, generating code, and more.

**DOCUMENTATIONS**

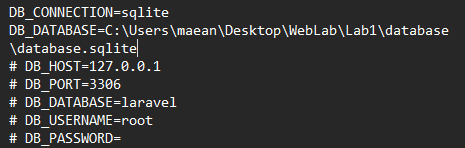
1. Install Laravel along with its dependencies.



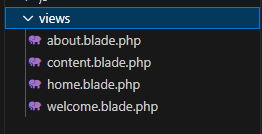
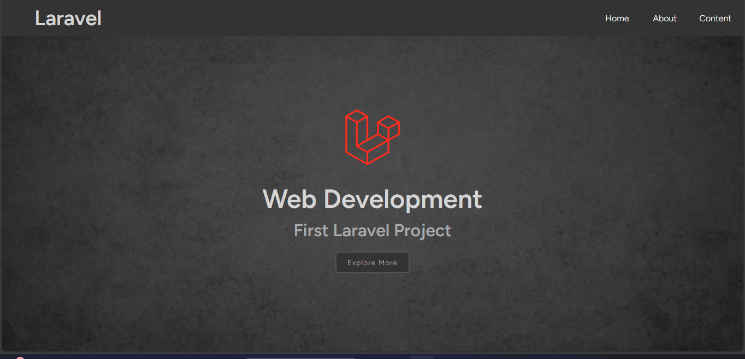
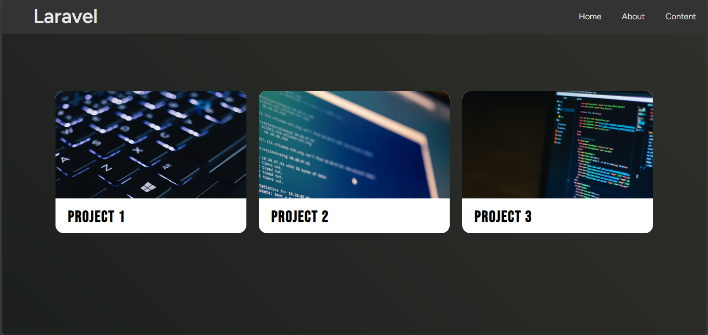
1. Create a Laravel project and put it in a Git repository

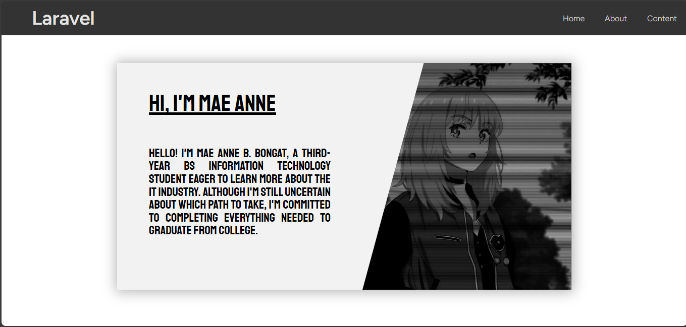


1. Set up the .env file to include database connection details



1. Create at least 3 views (homepage, about, content)





1. Create and run routes

