

# It Elec 1 - Web Development

Ken Anne J. Acosta  
BSIT 3C

## LABORATORY 1

1. Install Laravel along with its dependencies.

One click PHP development environment.  
Zero dependencies. Zero headaches.

Download for Windows

[Looking for macOS?](#)

Installing

Please wait while Herd is being installed.

Herd 1.10.0

Next > Cancel

Command Prompt - laravel new projectName

- Downloading sebastian/exporter (6.1.3)

- Downloading sebastian/environment (7.2.0)

- Downloading sebastian/diff (6.0.2)

- Downloading sebastian/comparator (6.0.2)

- Downloading sebastian/code-unit (3.0.1)

- Downloading sebastian/cli-parser (3.0.2)

- Downloading phpunit/php-timer (7.0.1)

- Downloading phpunit/php-text-template (4.0.1)

- Downloading phpunit/php-invoker (5.0.1)

- Downloading phpunit/php-file-iterator (5.0.1)

- Downloading theseer/tokenizer (1.2.3)

- Downloading sebastian/lines-of-code (3.0.1)

- Downloading sebastian/complexity (4.0.1)

- Downloading sebastian/code-unit-reverse-lookup (4.0.1)

- Downloading phpunit/php-code-coverage (11.0.6)

- Downloading phar-io/version (3.2.1)

- Downloading phar-io/manifest (2.0.4)

- Downloading myclabs/deep-copy (1.12.0)

- Downloading phpunit/phpunit (11.3.1)

0/107 [>-----] 0%

15/107 [===>-----] 14%

24/107 [=====>-----] 22%

33/107 [=====>-----] 30%

45/107 [======>-----] 42%

57/107 [======>-----] 53%

Command Prompt - laravel new projectName

Installing

Please wait while Herd is being installed.

Herd 1.10.0

Next > Cancel

Command Prompt - laravel new projectName

- Downloading sebastian/exporter (6.1.3)

- Downloading sebastian/environment (7.2.0)

- Downloading sebastian/diff (6.0.2)

- Downloading sebastian/comparator (6.0.2)

- Downloading sebastian/code-unit (3.0.1)

- Downloading sebastian/cli-parser (3.0.2)

- Downloading phpunit/php-timer (7.0.1)

- Downloading phpunit/php-text-template (4.0.1)

- Downloading phpunit/php-invoker (5.0.1)

- Downloading phpunit/php-file-iterator (5.0.1)

- Downloading theseer/tokenizer (1.2.3)

- Downloading sebastian/lines-of-code (3.0.1)

- Downloading sebastian/complexity (4.0.1)

- Downloading sebastian/code-unit-reverse-lookup (4.0.1)

- Downloading phpunit/php-code-coverage (11.0.6)

- Downloading phar-io/version (3.2.1)

- Downloading phar-io/manifest (2.0.4)

- Downloading myclabs/deep-copy (1.12.0)

- Downloading phpunit/phpunit (11.3.1)

0/107 [>-----] 0%

15/107 [===>-----] 14%

24/107 [=====>-----] 22%

33/107 [=====>-----] 30%

45/107 [======>-----] 42%

57/107 [======>-----] 53%

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

Github Repository link : [https://github.com/alyxvkeniii/WebDev\\_lab1.git](https://github.com/alyxvkeniii/WebDev_lab1.git)

alyxvkeniii / WebDev\_lab1

Type to search

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

WebDev\_lab1

Public

Pin

Unwatch

1

master

1 Branch

0 Tags

Go to file

Add file

Code

hempers

uploadfiles

fd4b5eb · 1 hour ago

1 Commit

Laboratory1

uploadfiles

1 hour ago

README

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README

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

DATABASE CONNECTION

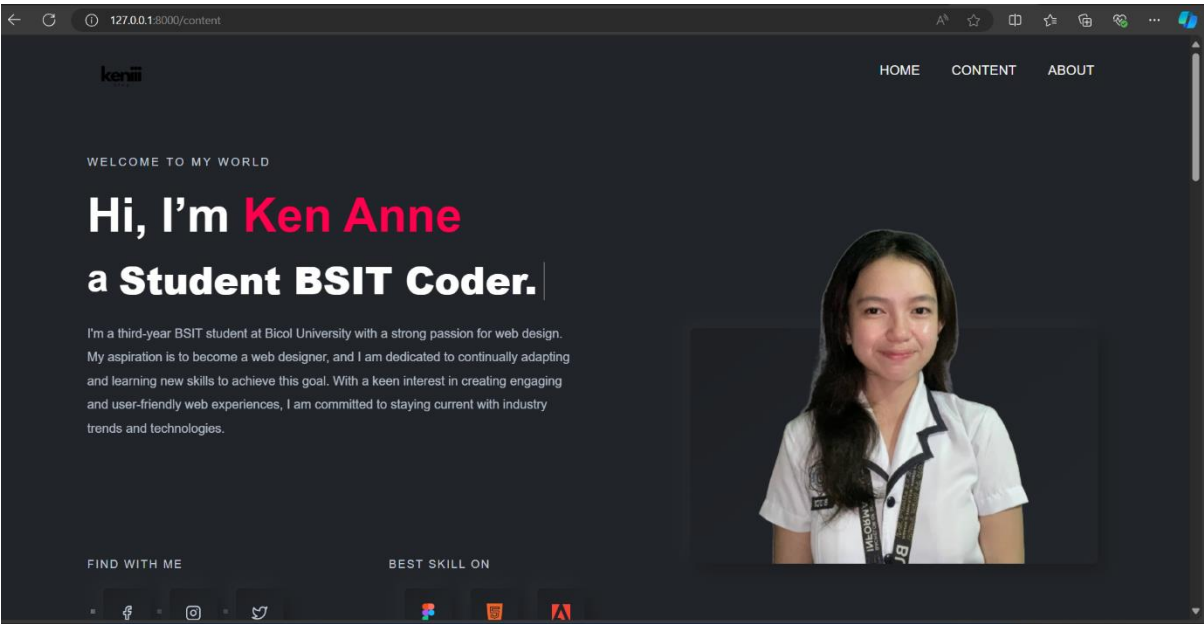
```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laboratory1
DB_USERNAME=root
DB_PASSWORD=
```

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

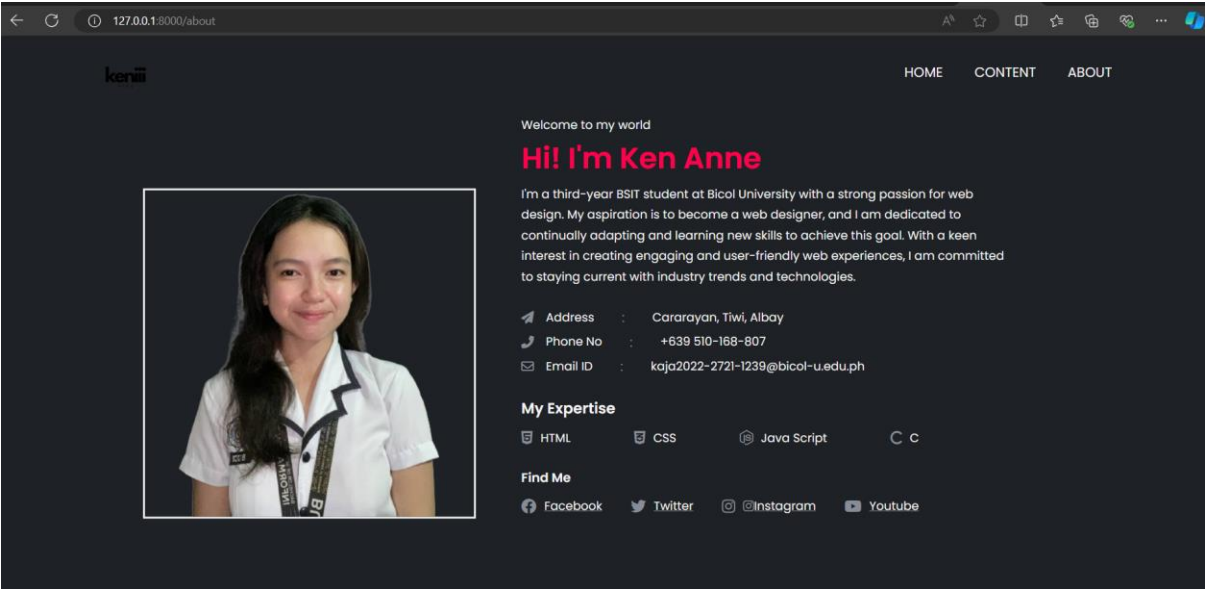
HOMEPAGE



CONTENT



# ABOUT



## 5. Create and run routes

```
Laboratory1 > 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('/about', function () {
10     |   return view('about');
11 });
12
13 Route::get('/content', function () {
14     |   return view('content');
15 });|
```

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

### Key Directories

#### a. `app/`:

Purpose: Contains the core code of your application.

Subdirectories:

    `Http/` : Contains controllers, middleware, and requests.

    `Models/` : Contains Eloquent models representing database tables.

    `Console/` : Contains custom Artisan commands.

    `Providers/` : Contains service providers for binding classes into the service container.

#### b. `bootstrap/`:

Purpose: Contains files that bootstrap the framework and configure the application.

Subdirectories:

    `cache/` : Contains cached files for performance optimization.

c. ``config/``:

Purpose: Contains all the configuration files for the application, such as database, mail, and cache configurations.

d. ``database/``:

Purpose: Contains database-related files.

Subdirectories:

``migrations/``: Contains migration files for creating and modifying database tables.

``factories/``: Contains model factories for generating test data.

``seeds/``: Contains seed classes for populating the database with test data.

f. ``public/``:

Purpose: The web server's document root. Contains assets that are publicly accessible.

Files:

``index.php``: The entry point for the application.

g. ``resources/``:

Purpose: Contains views, raw assets, and language files.

Subdirectories:

``views/``: Contains Blade template files for rendering HTML.

``lang/``: Contains localization files.

``js/``: Contains JavaScript files.

h. ``routes/``:

Purpose: Contains route definitions for the application.

Files:

``web.php``: Defines routes for web requests.

``console.php``: Defines console commands routes.

i. ``storage/``:

Purpose: Contains logs, file uploads, and other files generated by the application.

Subdirectories:

``app/``: For application-specific files.

``framework/``: Contains framework-generated files like cache and sessions.

``logs/``: Contains log files.

j. ``tests/``:

Purpose: Contains test files for the application.

Subdirectories:

``Feature/``: Contains feature tests.

``Unit/``: Contains unit tests.

k. ``vendor/``:

Purpose: Contains third-party packages and Laravel's internal packages. Managed by Composer.

## Key Files

1. ``env``:

Purpose: Contains environment-specific configuration, such as database credentials and application settings.

2. ``artisan``:

Purpose: The command-line interface for Laravel, used to run Artisan commands.

3. ``composer.json``:

Purpose: Manages PHP dependencies and autoloading.

4. ``package.json``:

Purpose: Manages JavaScript dependencies and scripts.

5. ``server.php``:

Purpose: Used by Laravel's built-in server for handling requests.

6. ``README.md``:

Purpose: Provides information about the project, including setup instructions and usage.