Introduction to Laravel and MVC

Laravel is a PHP web framework that uses the MVC architecture.

Framework: the underlying system of methods, classes, and files

Architecture: the design pattern that a specific framework implements

M Model

V View

C Controller

The models represent classes, the views represent the methods, and the Control means classes that manipulate the DOM. The MVC is an object-oriented programming Framework. Let now Break down MVC

Model – Deals with databases, ex, if we have users in our application, we can have a user’s model. This class deals with everything related to quiring the database for that specific table of users. When we have a user model, we also have a user table. Most of the time, we have a model and a table for that specific model. Another example is when we have post functionality; in this case, we will also have a post database to keep track of user activity.

View - The view deals with HTML. So everything you see on your application in the browser will be a view that represents our application.

Controller- The middle-man. The controls deal with the views and the model. The controller talks to the model. For example, The controller might ask the model to pull all the last names in the user’s table, and then once the controller receives it, it gives it to the view.

Required applications:

XAMPP - Apache |MYSql|FileZilla|Mercury|Tomcat

Git- Version Control | PHP favors Linux like commands so use bash

Visual Studio Code – Text Editor

Composer- Dependency Manager for PHP (Double check the php target)

Node.js- Dependency Manager for JavaScript

Laravel- PHP framework

Windows – Local Environment Setup

In XAMPP start Apache and MySQL, if you are having issues with Apache this might be caused by you already using the port. If so, change the port and try again. Finally, in git bash, type php -v and double-check the configuration; if you receive a message saying it is not found, restart your computer and try again.

Windows – Local Environment Setup – using MySQL

Start by running Apache and MySQL from XAMPP; next, in your web browser, type local host, which should redirect you to the XAMPP dashboard. From the dashboard, you can navigate to phpMyAdmin. To do this in the command line (using CMD) type the following.

C:\xampp\mysql\bin\mysql -uroot -p

You will then be prompted to enter a password (the default password is an empty space so just hit enter) now you have entered MariaDB. To show databases type the following

show databases;

. In order for Laravel to interact with mysql easer, we should add the path we used to the environment variables.

1. Copy the path
2. In Windows search enter edit the System Environment Variables
3. Click on path
4. Click Edit
5. Click New
6. Enter in the path (C:\xampp\mysql\bin)

Now the following command should work in cmd, allowing Laravel to detect it.

mysql -uroot -p

To create a database, enter the following command –

create database my\_db;

To select a specific database, enter the following command –

use my\_db;

To show the tables inside of the database, enter the following command –

show tables;

Now back in the web browser, if you navigate to phpMyAdmin and refresh, you will now see your new spiffy database!

Windows – Local Environment Setup – Installing Node.js

This course’s primary focus is on the PHP Apache MySQL stack, but facts are facts, there are just some JavaScript frameworks that are mandatory in the creation of modern web applications. Thus we must concatenate JavaScript to our tech stack.

Essentially NodeJS allows us to run JavaScript on our server, but it also comes with another tool that we will need npm or node package manager. Similar to composer it is the dependency Manager for JavaScript. Once you have installed the recommended version of Nodejs, we will now confirm its installation with git bash. Enter the following command into git bash

node -v – this should return the version of Nodejs currently on the machine.

Next, confirm the installation of npm with the following command

npm -v – This should return the version of npm currently installed on the machine.

Windows – Local Environment Setup – Installing Laravel

I will briefly go over the different installation methods for Laravel:

1. Installing Laravel using an installer application
2. Installing Laravel using composer
3. Installing Laravel in a virtual environment using Homestead
4. Installing Laravel using Valet allows for multiple project automation.

For our purposes, we will be using the composer:

In bash, cd into the director, and we will install Laravel. You can either choose your folder or follow along with these instructions.

Enter the following into bash to cd into it - cd ~ /Desktop/code

Once inside the folder, enter the following –

composer create-project laravel/laravel example-app

We are using our PHP dependency manager to create a new Laravel project. Now clear the screen with the following command- clear. Next, enter the following to display the contents of the code file -ls/example-app.

To cover all of our Laravel installation basis, I will demonstrate one additional method to install Laravel Globally on the machine. In bash, clear the screen, next enter the following command –

composer global require laravel/installer

Now whenever I want to create a new larval application, I can type in the following into bash - laravel new new-example-app.

Now open up the example-app in visual studio. Create a new bash terminal in visual studio and enter the following command - php artisan serve.

This will run the application in port 8000 by default and display a template Laravel application; the serve command is a wrapper for the PHP server. To close the server, type ctrl-c .