1. Setup Laravel

For this sample app, we used the latest stable version of Laravel (v8.74.0 as of this writing).

a. Clone the github repo:

Git clone https://github.com/path-to/sample-app.git

*If you’re not familiar with git, you can read Github’s guide here: https://docs.github.com/en/get-started/quickstart/hello-world*

b. Install required Laravel dependencies:

Within the project directory, run the following command from your command line interface:

composer install

*More info can be found at https://laravel.com/docs/8.x/installation*

You should now have the app scaffolding like this:

Graphical user interface

Description automatically generated with medium confidence

We will explain the helpers, controllers, and models in detail later.

c. Prepare your Laravel configuration file:

rename .env.example to .env

d. Create an application key

Within the project directory, run the following command from your command line interface:

php artisan key:generate

This will create the required application key (necessary for all encrypted cookies, such as session variables) and insert it into your .env configuration file.

e. Configure your application

Your .env file should look like this:

Table

Description automatically generated

Modify the database connection section to match your database credentials.

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=**contosocoffee**

DB\_USERNAME=**root**

DB\_PASSWORD=**root**

MYSQL\_ATTR\_SSL\_CA=DigiCertGlobalRootCA.crt.pem

f. Create the required database tables

Within the project directory, run the following command from your command line interface:

php artisan migrate

g. seed the database with sample data

Within the project directory, run the following command from your command line interface:

php artisan db:seed

2. Configure Your Environment

**Windows**

???

**Mac**

For the Mac, we used MAMP for our local server environment. In this example, we’re using the built-in latest stable version of PHP (v8.0.8 as of this writing).

Create a new host…

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Note: Make sure you set the document root to the “public” directory.

Graphical user interface, text, application

Description automatically generated

Your local server should be running. If all goes well, you should see a page like this:

Graphical user interface, website

Description automatically generated

If you don’t have your database configured, the sample app will use included JSON data and display a warning on the site.

Graphical user interface, website

Description automatically generated



3. Application Details

As you navigate the sample app, you’ll notice a cart widget on every page. This is built via an app helper (/app/Helpers/AppHelper.php) so it can be accessed anywhere in the app.

The Product controller provides the functionality for the categories and items.

Graphical user interface

Description automatically generated with low confidenceGraphical user interface

Description automatically generatedA collage of food

Description automatically generated with medium confidence

The Cart controller provides all the functionality needed to add items to your cart, process your order, and display a receipt.

Graphical user interface, text, application

Description automatically generated

Since we are trying to keep it as simple as possible, we only have a few fields on the checkout process. And to avoid adding any javascript complexities for requiring fields, we use the User information if any field is left blank.

Graphical user interface, text, application, email

Description automatically generated

Delivery time is based on a random number plus the longest “cook time” of the items in your order. We store the delivery time in a session variable so we can display the time remaining, even if the user reloads the page.