**Installation Manual**

This installation manual will explain how to install the scheduler system onto their a hosting server. It is also possible to do it onto a shared server, meaning the website would be accessible via the internet, but this requires two things: a domain name and a shared server capable of running Laravel, such as https://www.fortrabbit.com/. Both costing a certain amount of money. However, the installation manual will only show the administrator how to install the project locally on the computer.The following installation is performed on Windows 10, but it is also possible to do it on other OS.

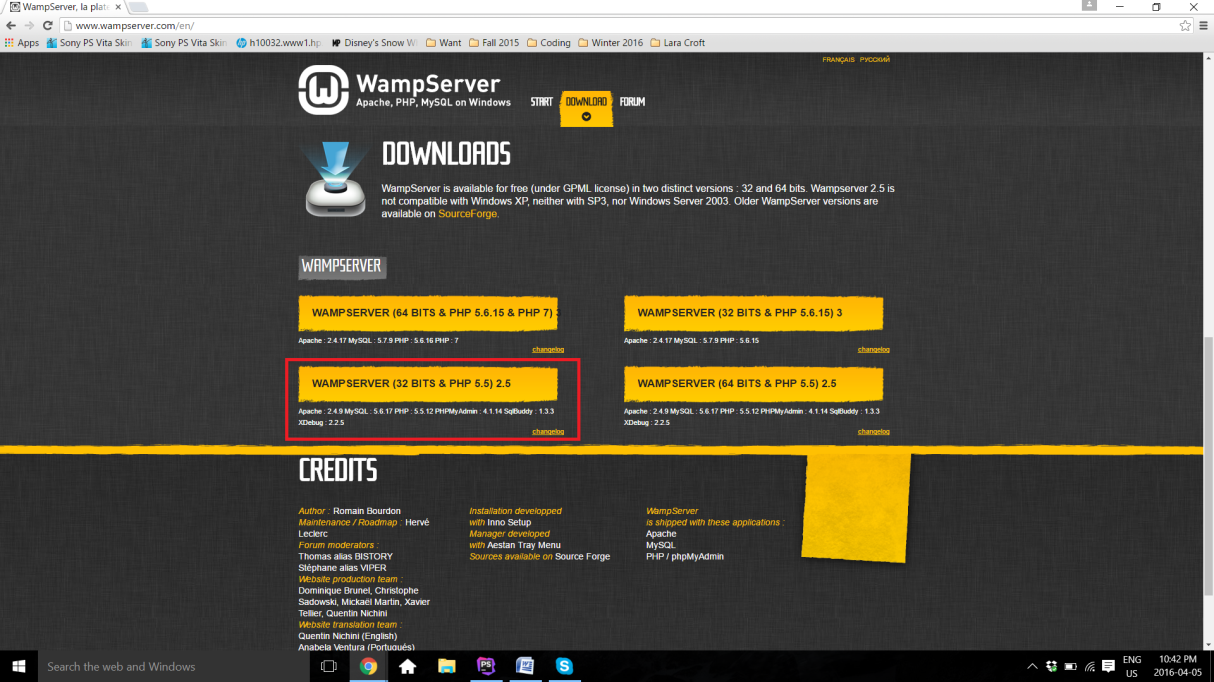
4 software are mainly required, plus the scheduler project:

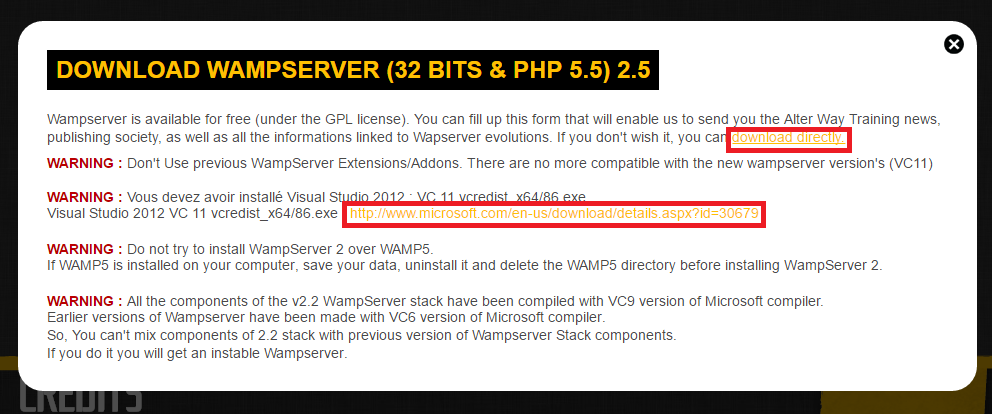
* Wamp
* Visual Studio
* Composer
* MySQL
* Scheduler Project from GitHub

Step 1: WampServer Installation

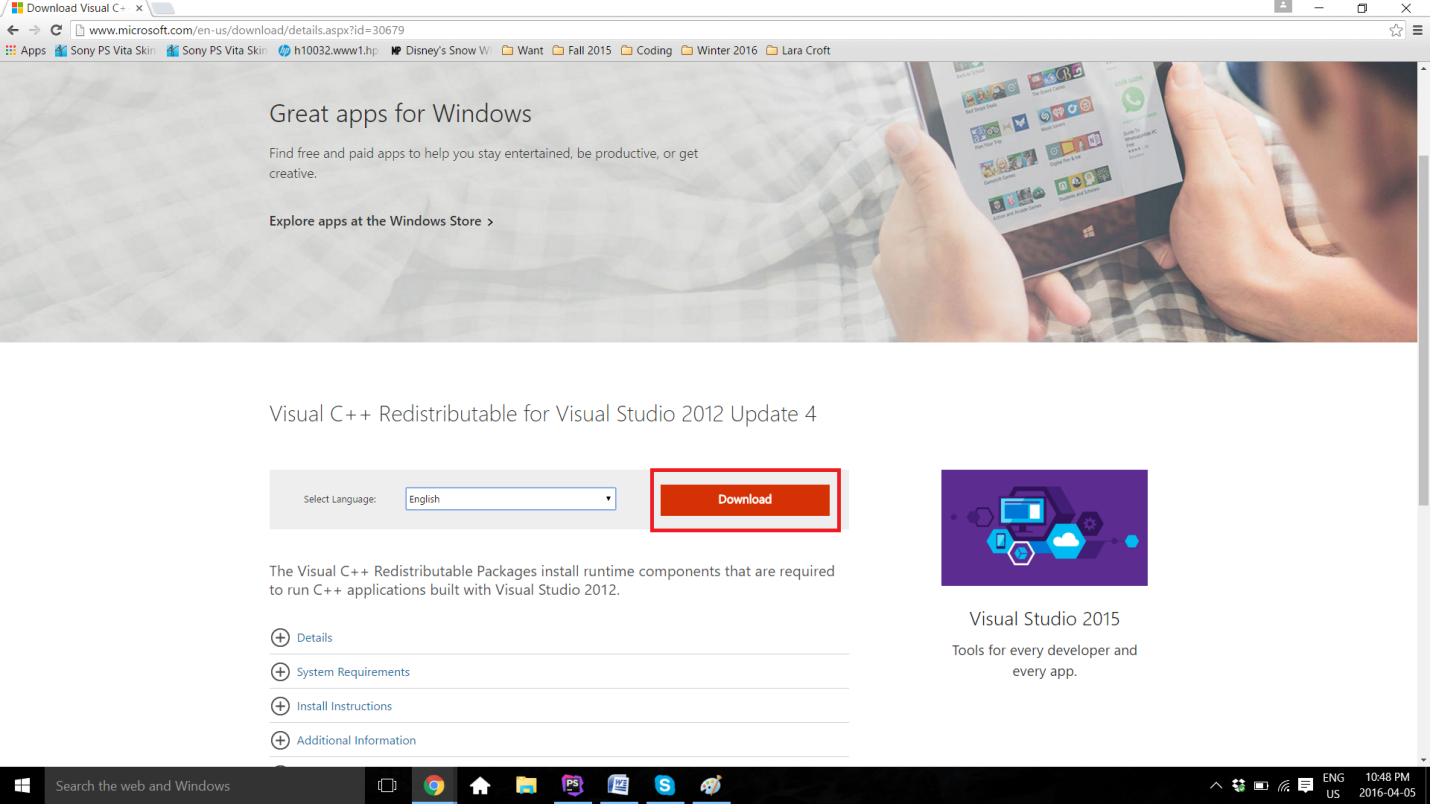
The first step consist of downloading and installing WampServer. WampServer contains all the needed softwares to run a php program on the local host: PHP, Apache and MySQL. It is preferable to download WampServer 2.5, containing especially Apache 2.4.9, MySQL 5.6.17, PHP 5.5.12. The scheduler has been coded in PHP 5.5 therefore, it is prefered to use that version in case syntax modifications have been done in the newer versions such as PHP 7. Furthermore, this package works both on 32-bit and 64-bit computers. The website to this download is http://www.wampserver.com/en/.

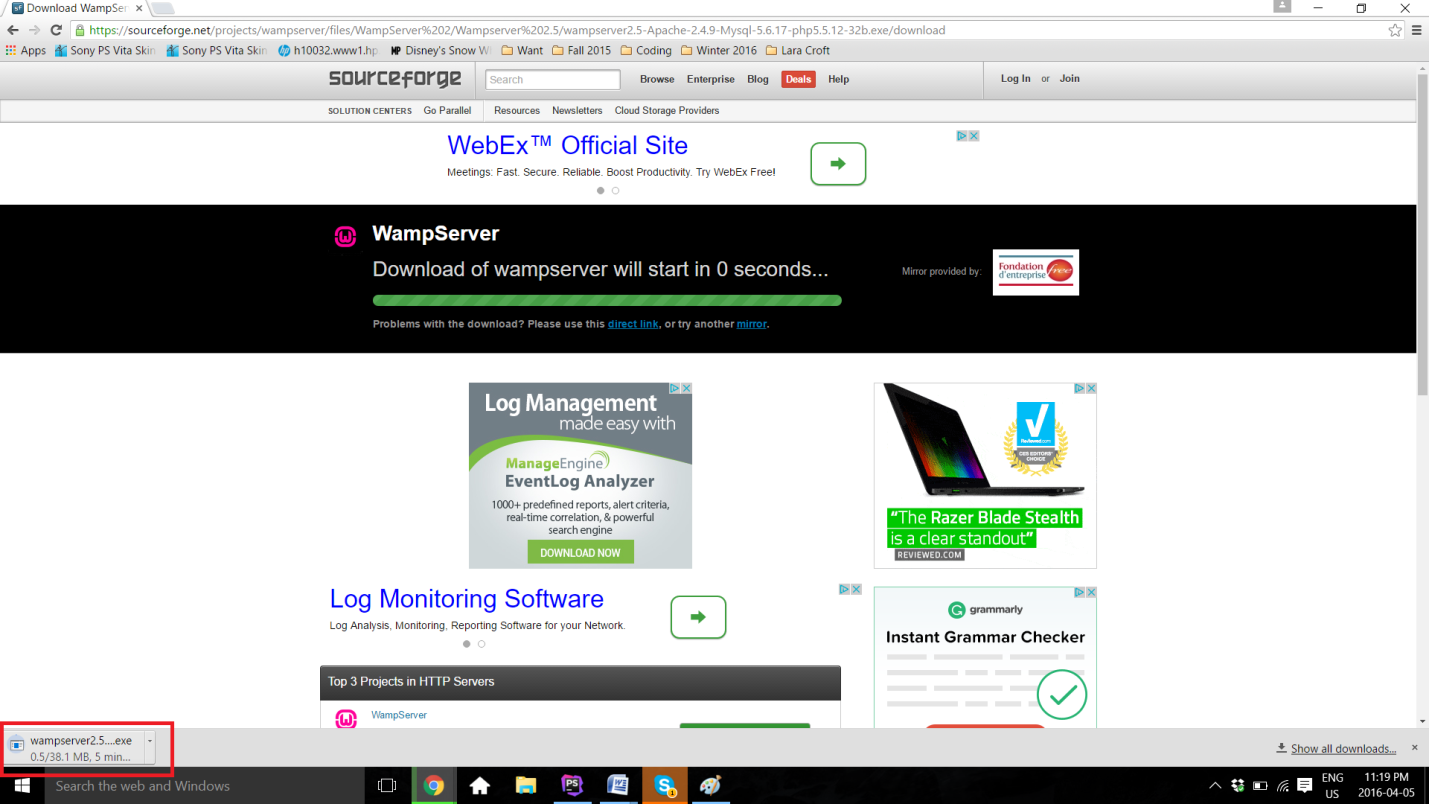
**Note:** Another alternative to WampServer is XAMPP. The only difference between both is that XAMPP is offered on multiple operating systems, such as iOS and Linux, in addition to Windows. WampServer is only offered for Windows.



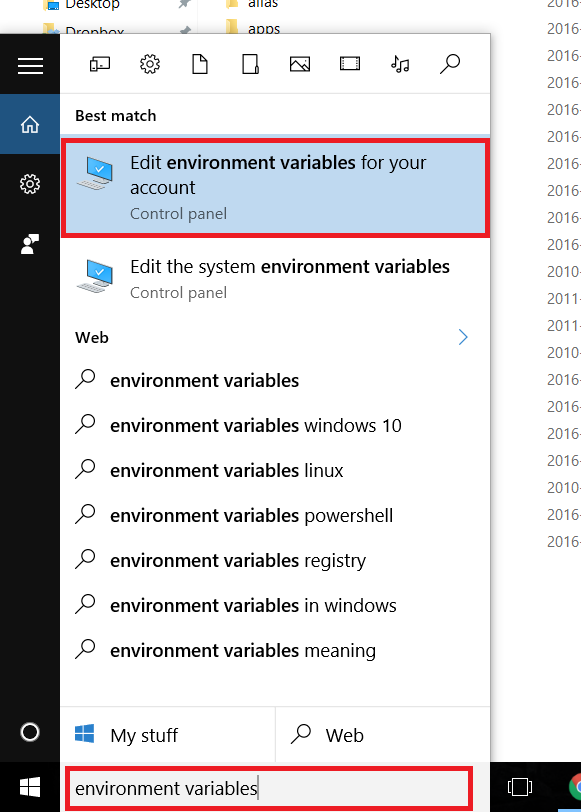


As shown here, there is a couple of warnings regarding WampServer. It is especially important to pay attention to second **warning**. In order for WampServer to work, Visual Studio is required. This is because of Apache, which works along with visual studio. If Visual Studio 2012 is not installed on the computer, the administrator should click on the link found in the same warning. This open Microsoft download page for the correct version of Visual Studio. Once downloaded, Visual Studio can be installed. No special modification is required, therefore it is only necessary to follow the instructions displayed on Visual Studio installation window and click on **Next** a couple of times until the installation has begun, and proceeded successfully. It is important to note that there is a paid version of Visual Studio, but it is not required as the free version works equally for the purpose of Apache.

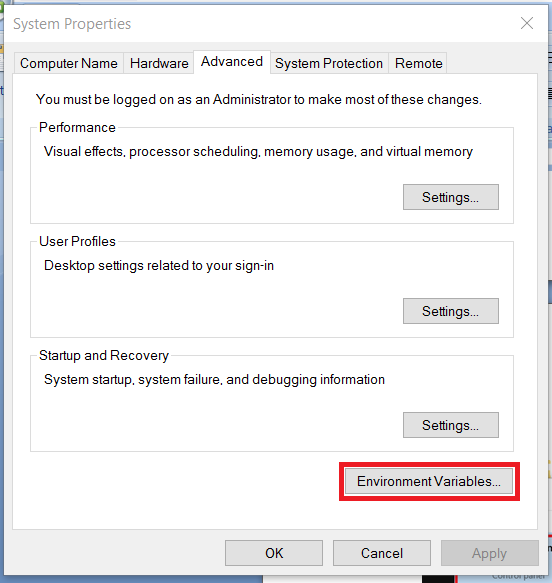
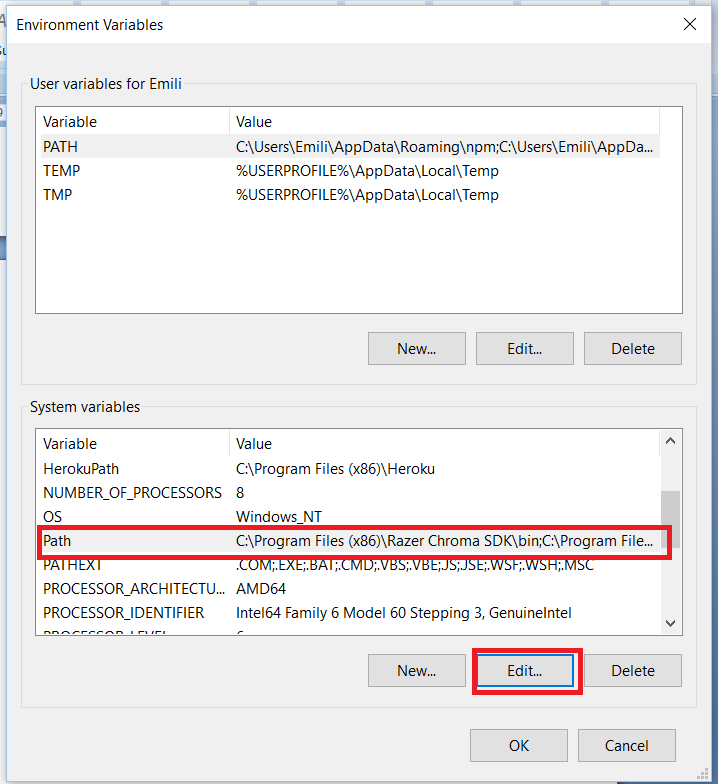


Once Visual Studio is set, the administrator can now go back to WampServer website and click **download directly**. A new page from the website SourceForge will open, containing the file. A few seconds has to pass for the download to start, and the download will begin. On Google Chrome, the download will be shown on the bottom left of the web page. After the download is completed, it is only sufficient to click on the file and the installation will begin. Otherwise, as any other downloads, the file has to be located to wherever downloaded files are usually saved (very often, it is in the **Download** file). 

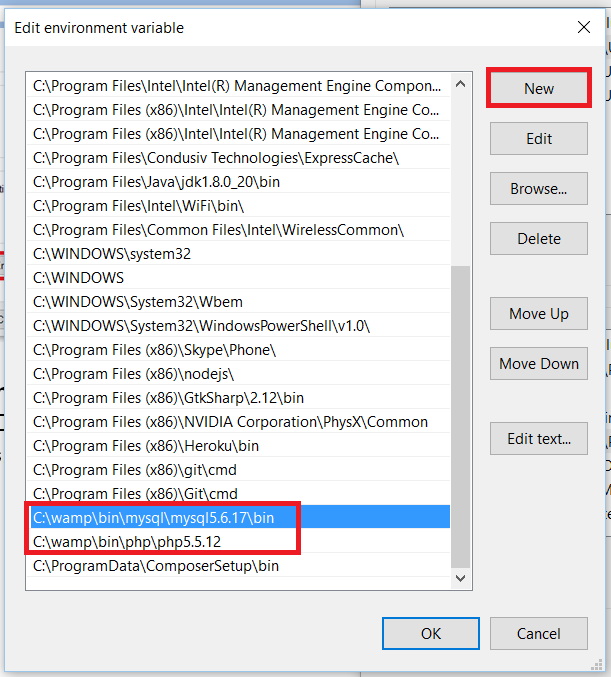
Once again, installing WampServer does not require any particular specifications or changes on the automatic procedure. The only thing required from the administrator is to keep track of where the software is saved. In this example, the software has been saved under **C:\wamp**.

Step 2: Create the PHP and MySQL Path in the Environment Variables.

This step enables to Command Prompt to have access to the PHP and MySQL, therefore being able to manipulate the launch the software from there. First, it is possible to find the Environment Variables by typing the name in the **Search** available on the computer.

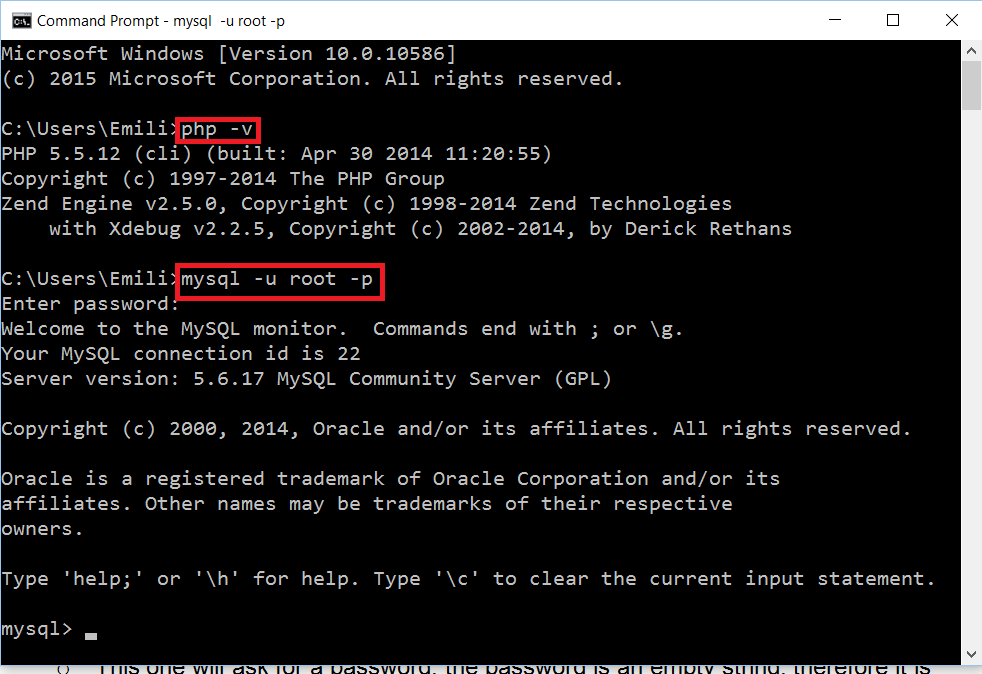
 

The control panel of the **System Properties** opens, under the **Advanced** section. **Environment Variables..** can be find there. Once clicked, another window called **Environment Variables** will open. In the **System variables**, **Path** has to be located, and then edited. Two new variables have to be added. The first one is the path to the Wamp MySQL bin file. The file has to be located into the computer, as it is **C:\wamp\bin\mysql\mysql5.6.17\bin** in this case. The second one is the path to the Wamp php file, as it is **C:\wamp\bin\php\php5.5.12**. Once everything is set, the administrator has to click **OK** on all three windows, therefore closing the **Control Panel**.

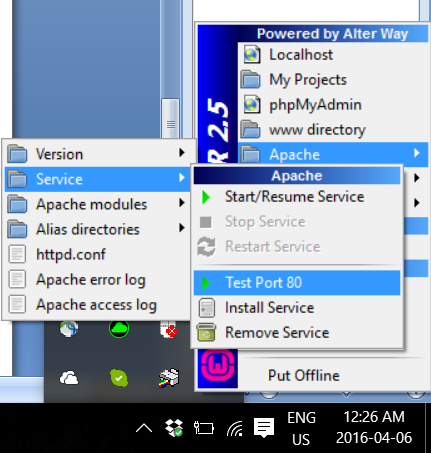
****

To make sure the path worked, the following commands have to be typed on the Command Prompt and the display should be similar to the following image, without any error message:

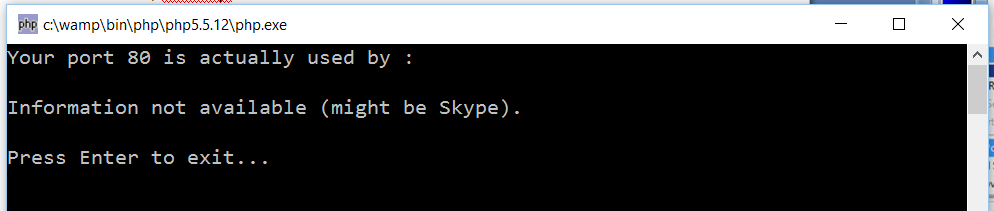
* **php -v**
* **mysql -u root -p** 
  + This one will ask for a password, the password is an empty string, therefore it is only necessary to click on enter.

****

Step 3: Turn on WampServer and test connection

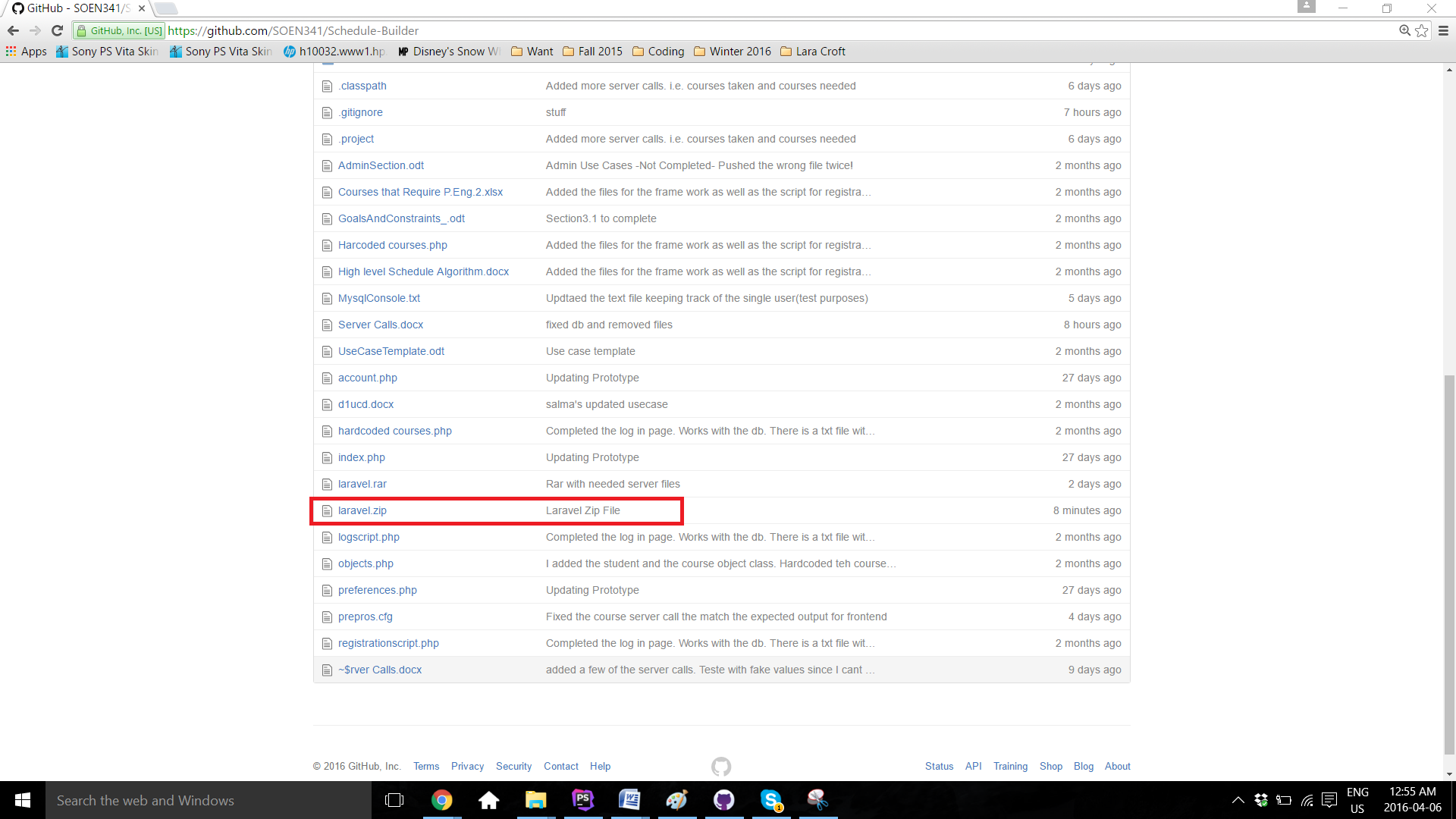
WampServer, as well for XAMPP, uses **port 80** for Apache. First, Wamp has to be turned on. If the color does not turn green after clicking **Start All Services** and the color is stuck at orange. It is most likely because **port 80** is used by another application. It is possible to test the connection by following this path on the Wamp application: **Apache > Service > Test Port 80**.

This is going to display an error if the port is really not available for Apache. As shown here, the error is usually caused by Skype being on at the same time, as both Apache and Skype use the same port. If it is the case, Skype has to be completely closed. It is then only sufficient to **Restart All Services** on Wamp, and the icon should turn green.

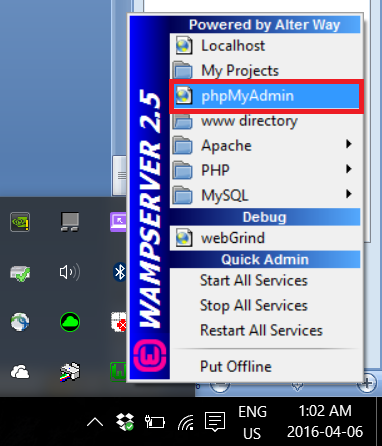


Step 4: Download the project from GitHub

The project has to be now retrieved from GitHub. This can be done from the following link: https://github.com/SOEN341/Schedule-Builder. As it is a zip file, the following has to be unzipped. The file can then be placed at any desired location.



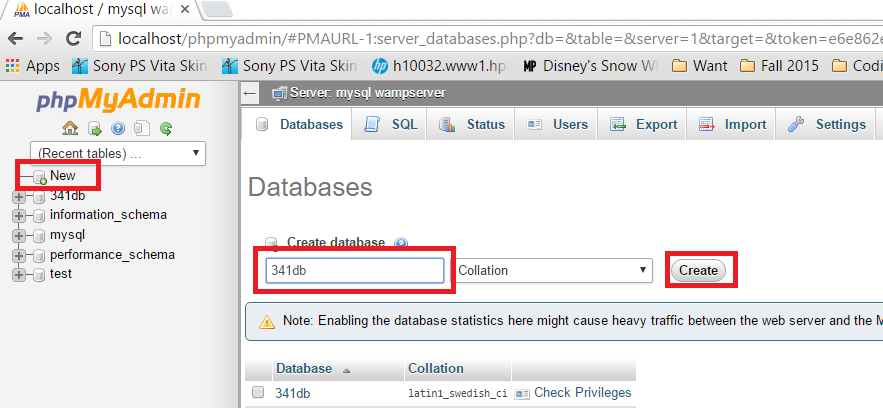
Step 5: Set Database

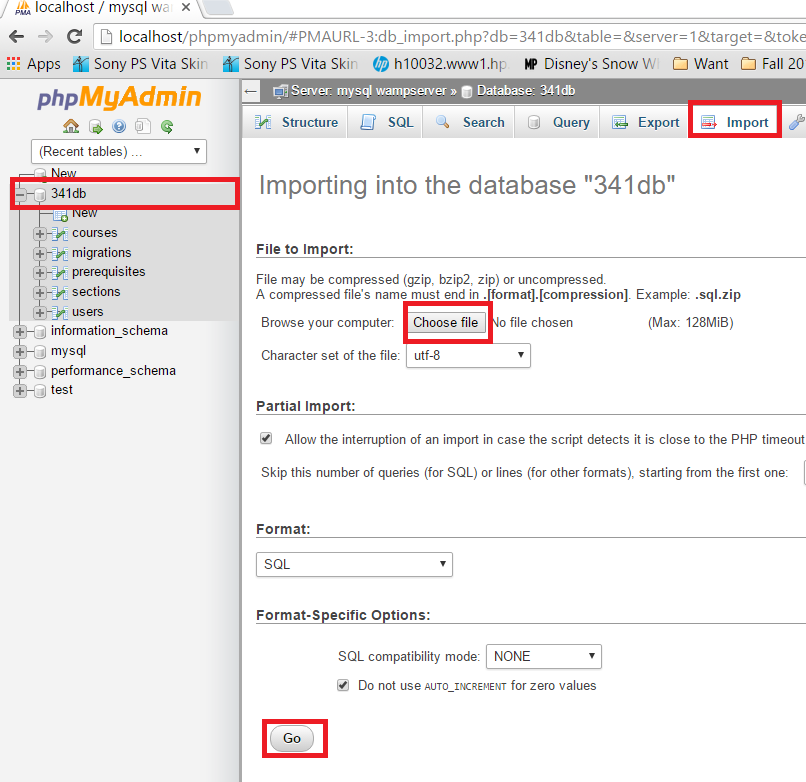


In order for the software to work, it has to be connected to the Database. First,

From the **Laravel** file downloaded form GitHub, there is a file called **finaldbfile.sql**. This is the entire database of the project, containing all courses, sections, users information. This database has to be uploaded on MySQL on Wamp. To do this, **phpMyAdmin** has to be opened on Wamp.

**phpMyAdmin** will then open on the default browser. First, the database has to be created, by clicking on **New**. The name of the database is **341db**.

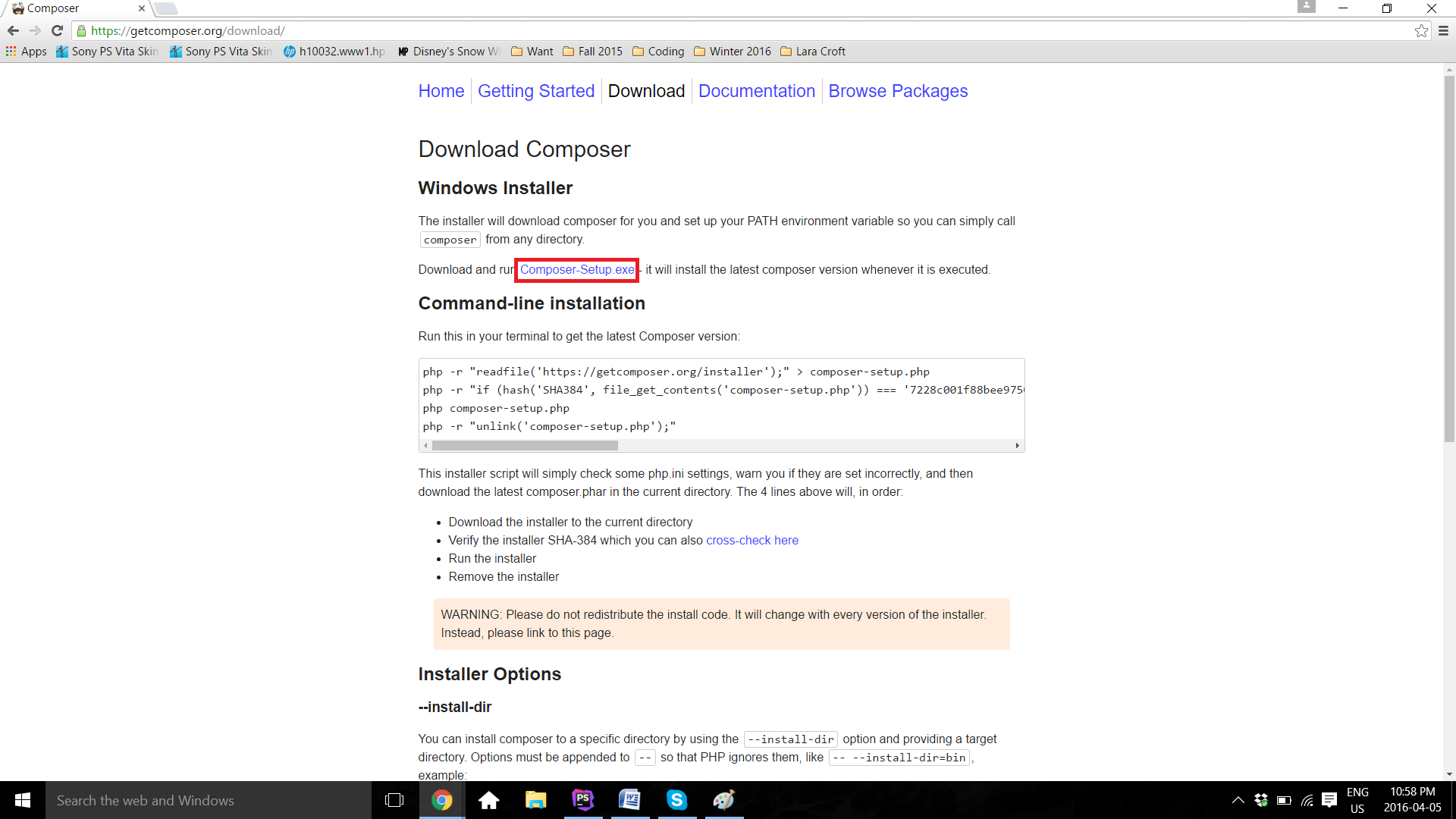




Afterwards, **finaldbfile.sql** has to be imported onto the same **341db** database. On **phpMyAdmin**, 341db has to be selected, then **Import**, then **Choose file**. Navigate to the laravel file and select **finaldbfile.sql**. Once the upload done, this section can be finalized by clicking **Go**. A green confirmation message will display, saying the import has been successful.

Step 6: Download and Install Compositor

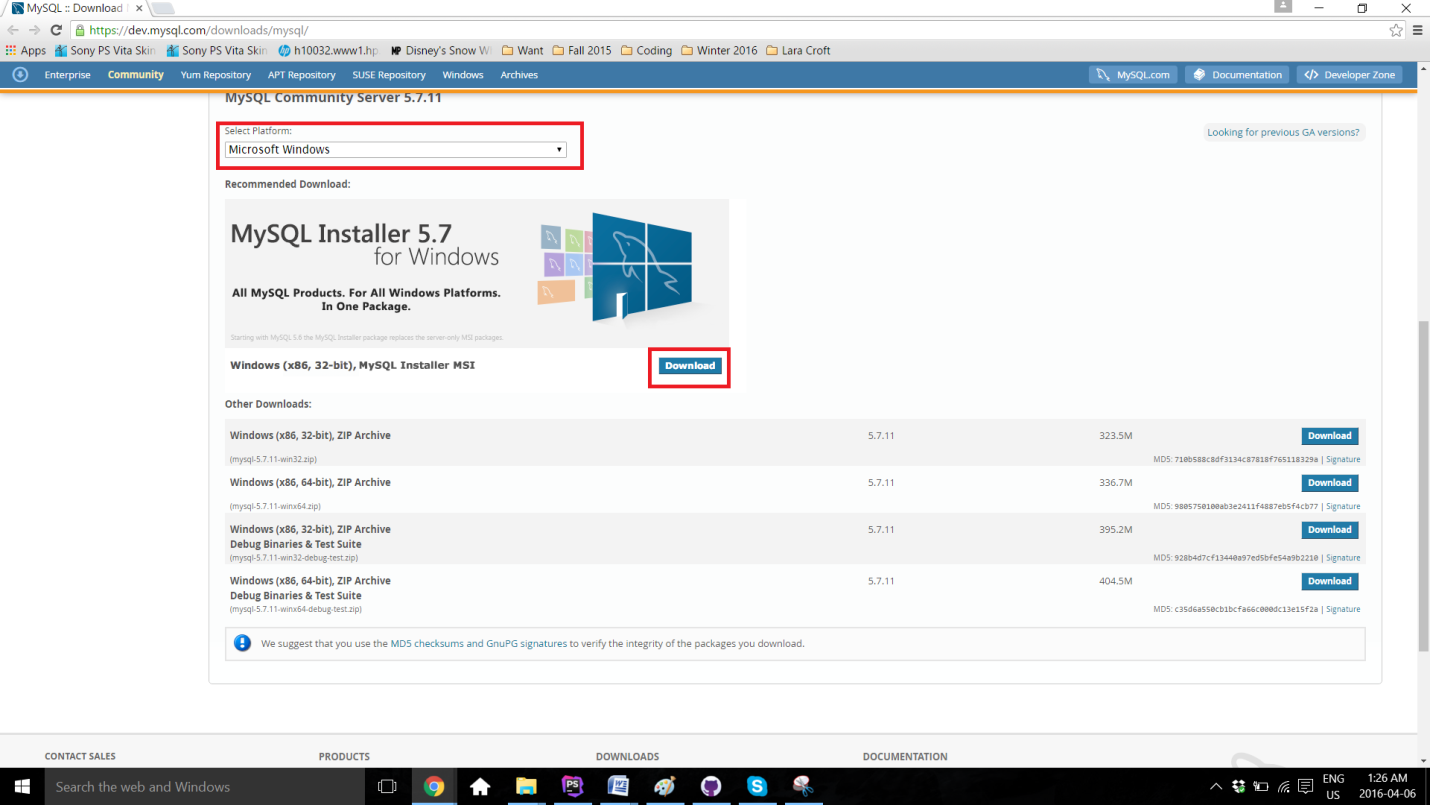
Compositor is a dependices manager for PHP. This means that Compositor will handle all the needed libraries and files, in order to ease the development of the project. It can be download from https://getcomposer.org/download/. It is once again only necessary to follow the installation instructions. However, it is important keep track of where the Compositor gets the php.exe file from. It has to be from Wamp. If it is not the case, relocate the path to the **php.exe** inside of Wamp. The path is **C:\wamp\bin\php\php5.5.12\php.exe** for this case.



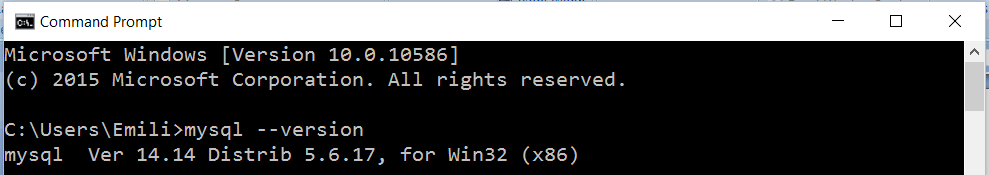
Step 7: Download and Install MySQL

Even though MySQL has been downloaded using Wamp, there seem to be some connection problem from Wamp if the software is not directly inside its www directory. To solve this problem, the original MySQL software is installed again, which helps to install MySQL correctly accross the whole system. This allows the project to see the available MySQL throughout the system.

The software can be downloaded from https://dev.mysql.com/downloads/mysql/. The download can be found under **Community Server**. The desired OS can be picked (Windows in this case). **MySQL Installer 5.7 for Windows** is the desired version.

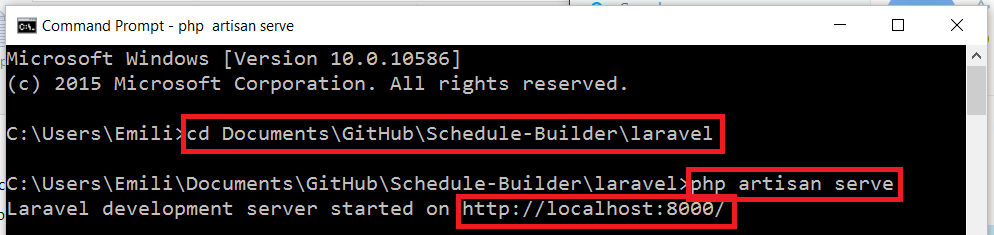


As any other past installation, it is only necessary to follow the instructions provided by the installer. Once installed, it is recommeded to test if MySQL is setup correctly, by typing **mysql --version** on **Command Prompt**.



Step 8: Run Scheduler Project

Once everything is set up and Wamp is turned on and green, the project can now run. To do so, **Command Prompt** is used. The first command to type is **cd \*project location\***. To navigate backward in folders, the command **cd..** does the trick. Once to the file of the project, the last command is executed: **php artisan serve**. The output **Laravel development server started on http://localhost:8000/** should display.



Now, it is only necessary to open a browser, and type **localhost:8000** in the URL bar.

