**THE PROJECT PLAN**

The developers will be grouped into **frontend team** and **backend team.** The frontend team will handle the user interface part of the project while the backend team will handle the server-side part of the project. The required languages/frameworks for the frontend team are **HTML, CSS, BOOTSTRAP, JAVASCRIPT, AJAX.** The backend team on the other hand are going to make use of **PHP, MYSQL.**

In order to enforce good standards, we will be making use of the **CodeIgniter** framework. The framework is efficient enough and it will save us time! This mainly concerns the backend team but the frontend team also needs to have a basic idea of **where and where not to place codes.** Read the **Project Structure** section to get more info.

**INSTALLATIONS (The Recommended Tools)**

Below is the list of recommended software that you should have installed in order to actively participate in the project. Both teams are required to have these tools (at least some of it):

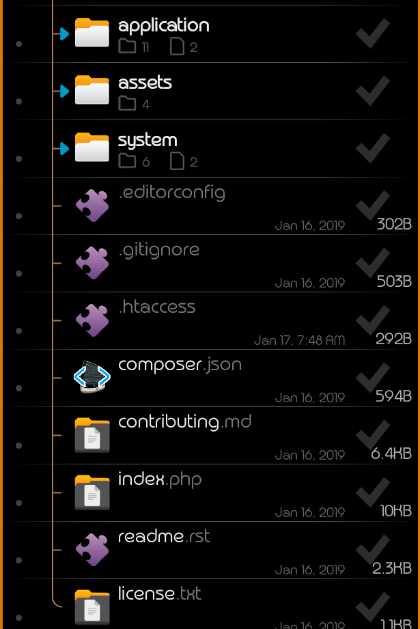
* XAMPP – You need a server to locally work on the site. XAMPP is an ideal tool for this purpose.
* A WEB IDE – You probably have this already. For your convenience, make sure your IDE supports VCS natively. This will be useful when working with GitHub.
* BROWSERS – You should be able to test the website on more than one browser.

**THE PROJECT STRUCTURE**

Since we are using a standard framework, the project structure is pretty much standard. Every file type has its folder where it must be placed, you cannot just place them anywhere.

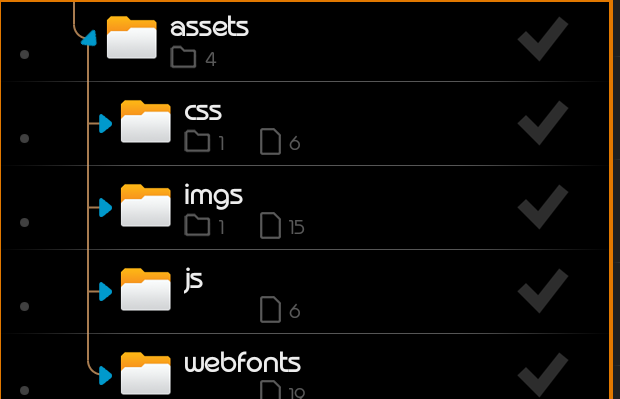
If you have ever get stuck, you can use this document as a reference. You can also contact me anytime. And don’t worry, I’ll send the current starting structure of the project to the group so you can see it, experiment and ask questions before the project development starts proper.

**THE PROJECT ROOT**

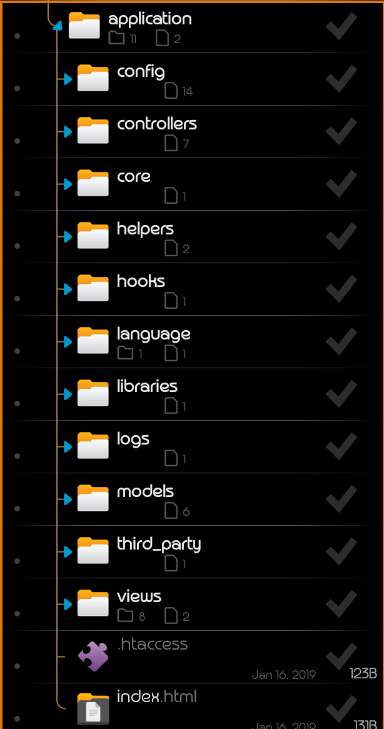


The image above shows exactly how the project root directory looks like. You will be working with just the **application** and **assets** folders. The **system** folder should be left the way it is, **NEVER SAVE ANY FILE THERE**.

The frontend team will primarily use the **assets** folder to store assets. Inside this folder, additional folders are provided to guide you on where to save a specific asset file. For example, if you are to create a new CSS file, you will save it inside the **css** folder that is located inside the **assets** folder. Study the image below to get the idea.



The **application** folder is what you will be working with most of the time. It contains a wealth of several other folders (scroll down to see the image below). Inside the **application** folder, the three most important folders are **controllers, models** and **views.** Among these folders, the frontend team are allowed to modify only the **views** folder, the rest are for the backend team to work with. The **views** folder is where you will save your HTML contents. Your HTML files must be saved with **.php** extension and not **.html**. Please always remember this!



**CREATING PAGES**

When you are to create a page, all you need to worry about is **the specific contents of that page**. Since the **header** and **footer** are almost similar in all pages, you don’t need to recreate them for every page, it will be added automatically for you.

For example, if I want to create a login page, I can start writing the HTML/CSS for the form right away, I don’t have to worry about the toolbar and footer links.

The header and the footer will be provided by default and will be located at **application/views/templates**. The frontend team can make changes to them as they see fit.

This concludes the documentation for now. Please read through and understand. We’re black belts so I don’t expect too much confusions. However, I’m always available if you need any help. **And as for those that have never used a framework like codeigniter before, consider this an opportunity to learn something new!**