Lab 4

Objective

- Students can install the XAMPP stack on their laptops.
- Students can write a basic hello world web application using PHP as the server-side programming language.
- Students understand and can host PHP web applications locally using the installed XAMPP stack.

Outcome

- A basic working PHP web application.

Instructions

1. Install XAMPP stack on your laptop.

In lab 1 and lab 2, you installed **CentOS operating system**, **Apache webserver software**, **MariaDB**, and **PHP** on your **Microsoft Azure VM (Remote)**. These together creates an integrated environment for hosting typical web applications. You successfully hosted your **first web application (index.php)** and your **second web application (wordpress)**.

As it turns out, you will be developing most of your applications on your local machine. You probably will only visit your Azure webserver when you are done building locally and want to deploy it so that it is publicly accessible on the internet.

It, therefore, makes sense that you create an environment similar to your Azure webserver for local development. One of the easy ways to do this is to install XAMPP.

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl(not used in this class) bundled together as single software. You installed and configured these constituents of XAMPP as individual components on your Azure VM in previous labs.

Please note down your XAMPP installation path as you will be using it later.

For Windows users, download and install the XAMPP executable from https://www.apachefriends.org/index.html.

macOS users can use the guide here https://medium.com/codespace69/php-xampp-install-xampp-on-macos-and-run-your-php-code-657b614446d9.

Likewise, Linux-Ubuntu users may use the guide here https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-20-04.

2. Your basic PHP web application

Download and unzip the ***lab4*** template on canvas. This template application contains the following files:

a. ***index.html***: contains a simple form with an input field for an Ashesi Student ID.

b. ***style.css***: contains simple styling for the form.

c. ***app.js***: we will use this to perform a basic client-side validation using regular expression, which you learned in the previous class. We will validate the Student ID entered in the input field before submitting it to the server-side programming language (PHP).

d. ***main.php***: PHP is the server-side programming language in this class. A proper introduction to the language will be done by your lecturers in subsequent classes. You can practice PHP using https://www.w3schools.com/php/php_intro.asp.

For this lab, we will print a simple hello world to test our PHP program using this file. Afterward, we will try fetching the Student ID that was submitted with the form on the client-side using this file.

3. Host your web application locally

Hosting your web application in simple language means uploading or putting all necessary files in your Apache webserver **root directory**. Your local Apache webserver root directory is **<your_xampp_installation_path>/htdocs**.

On your Azure VM, the Apache webserver root directory is **/var/www/html**. Please take note of the distinction.

Now put the the unzipped ***lab4*** folder in your local web server root directory.

4. Zip and submit your working web application (lab4 folder) on canvas.

To help solidify your knowledge we recommend you take php, css, html, and JavaScript course on sololearn as an individual initiative.