CSE 3241 Introduction to Database Systems

Individual Project using PHP w/ MySQL

Goal of this assignment:

Learn how to write a simple web application using PHP that interfaces with MySQL and Apache. This project is worth 10% of your total grade.

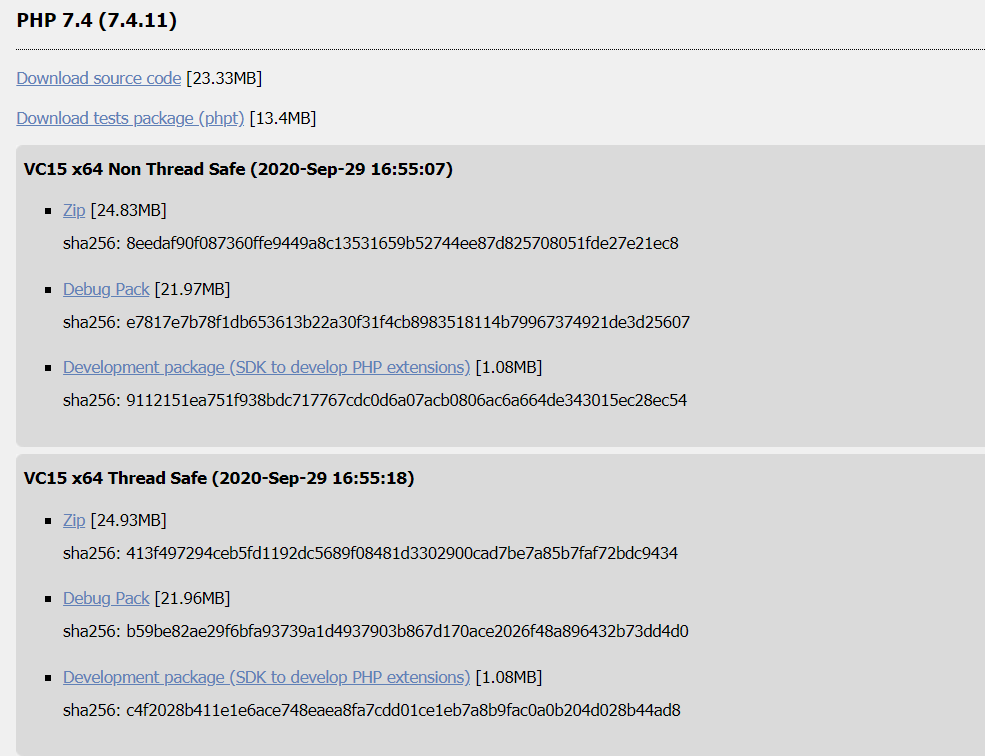
Prerequisites:

PHP and Apache are successfully installed and running.

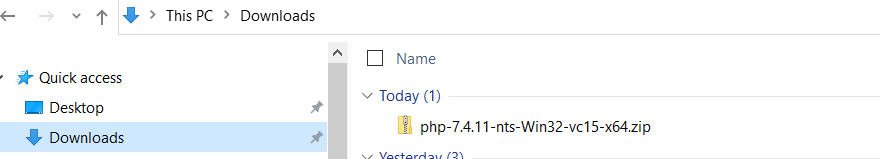
***For Windows Users:***

Task 1: Download and install PHP on your PC

You can download the PHP from <http://windows.php.net>. Select the ‘Downloads’ menu on the top. The current version at the time this lab assignment was written is 7.4.11. My laptop is 64-bit so I downloaded the x64 Thread Safe version. My previously installed version was 7.3.8.



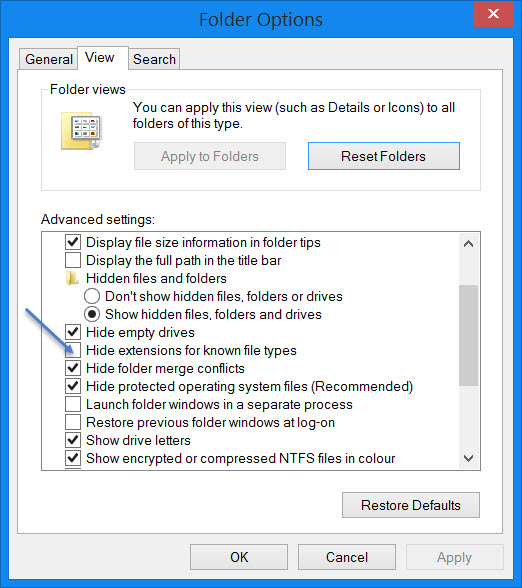
After the download, you can find the downloaded file in ‘Downloads’ folder.



Unzip the zip file (if you have winzip) or extract all to ‘C:\php’. *Note if you install the php files under C:/Program Files or other system folders, PHP may not have access to these directories and therefore, will not be able to access your php scripts.*

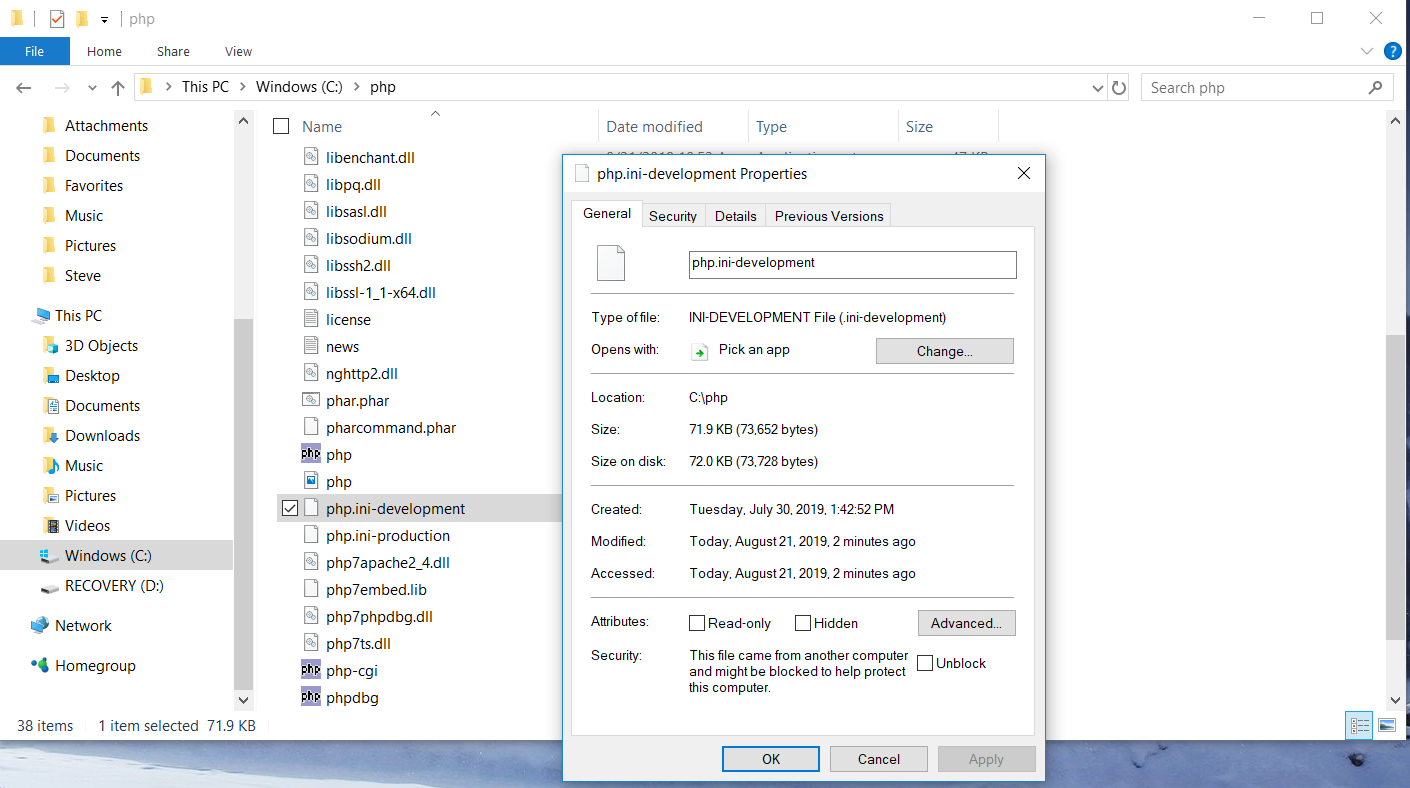
Before we configure the PHP, let us turn off the ‘Hide extensions for known file types’ to avoid headaches. This step is optional though. Open Control Panel > Appearance and Personalization.  Now, click on **Folder Options** or**File Explorer Option**, as it is now called > View tab. In this tab, under Advanced Settings, you will see the option **Hide extensions for known file types.**Uncheck this option and click on Apply and OK.

**Windows 10** users may also search for **File Explorer Options** in Start search box and open this box

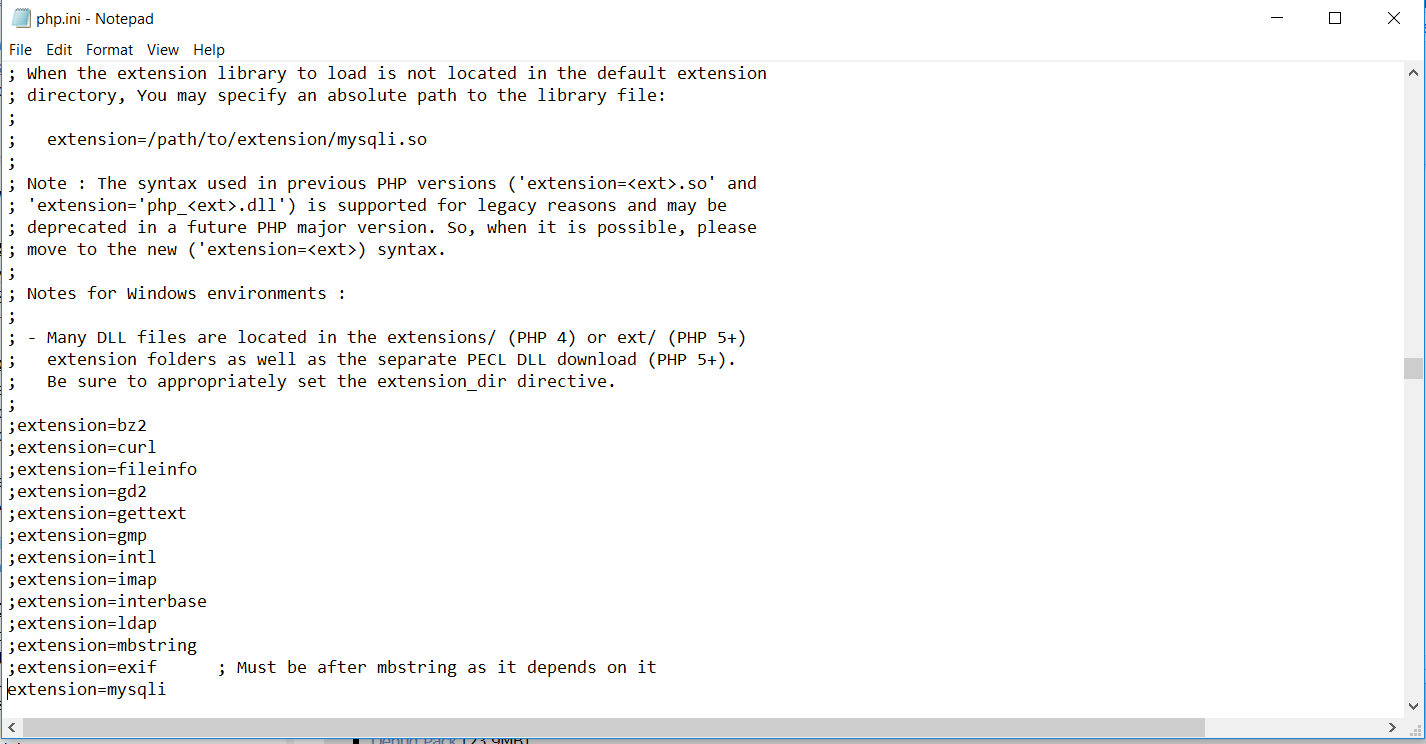


Task 2: Configure your PHP

Configuring PHP is simple since we only use the very basic features of it. Open php.ini-development and save a copy as php.ini.



Inside php.ini, remove the ‘;’ in front of “extension=mysqli”.



Task 3: Create a sample web page

In your PHP root directory (i.e. C:\php), create an index.php file. This will be the default file when PHP searches a .php file in the root directory.

// Here is what goes in your index.php file.

<html>

<head>

<title>My first PHP</title>

<body>

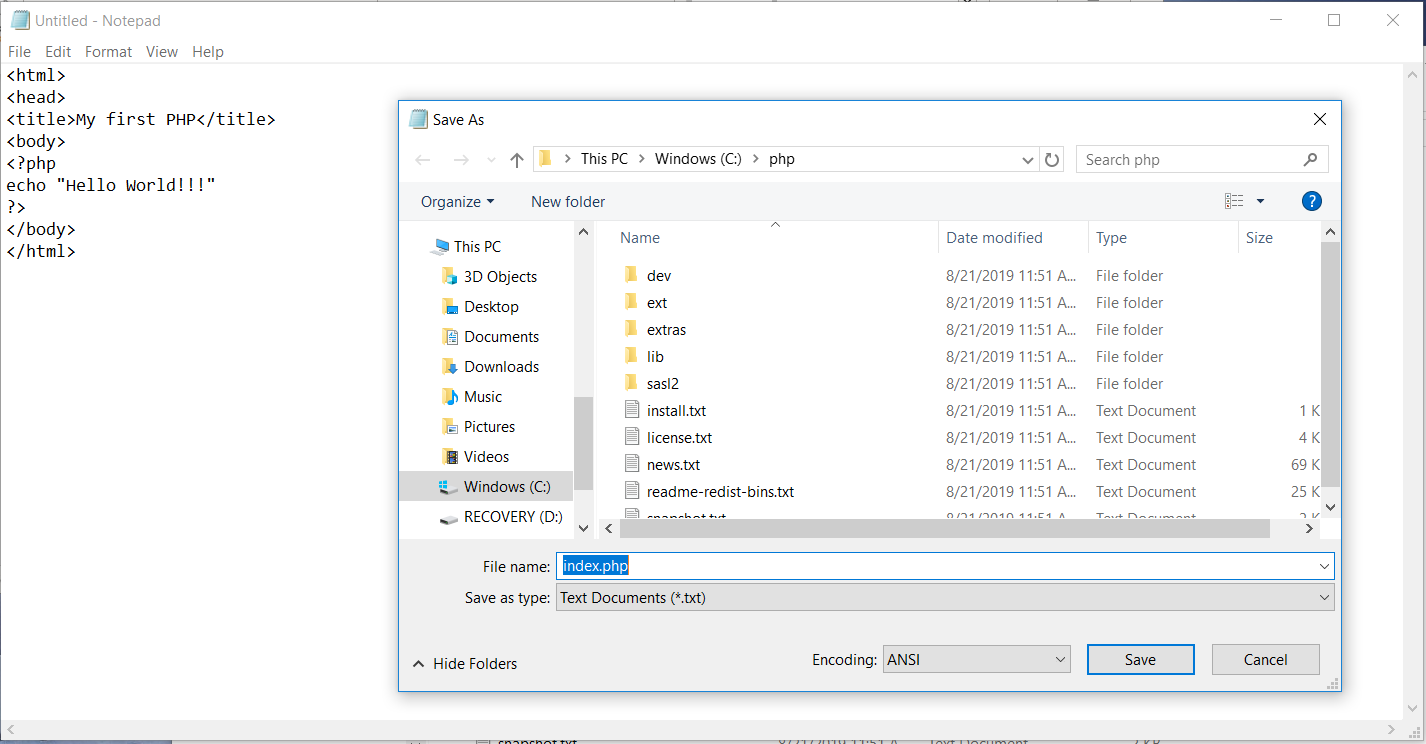
<?Php

Echo “Hello World!!!”

?>

</body>

</html>



Task 4: Test your PHP

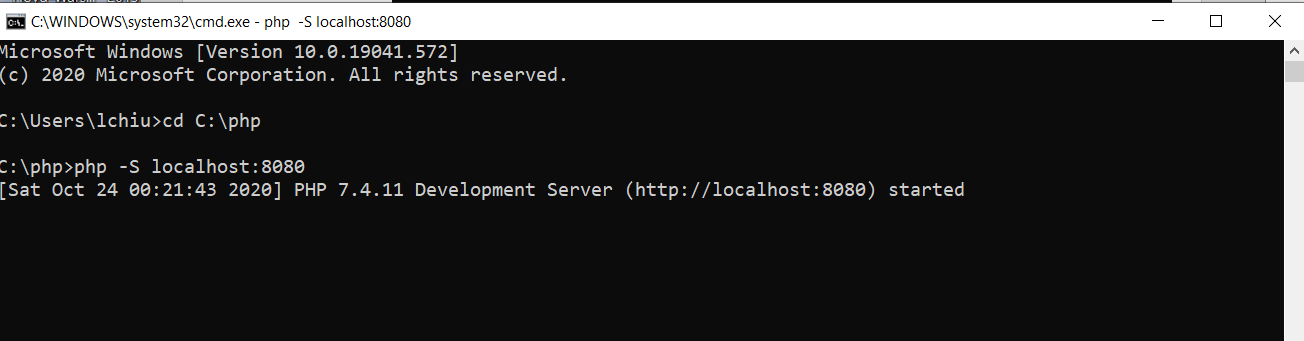
// Start up your PHP with Apache webserver

Enter ‘cmd’ in the Windows search box and follow the steps below,

* go to your php root directory, i.e. cd C:\php
* run the php.exe, i.e. php -S localhost:8080

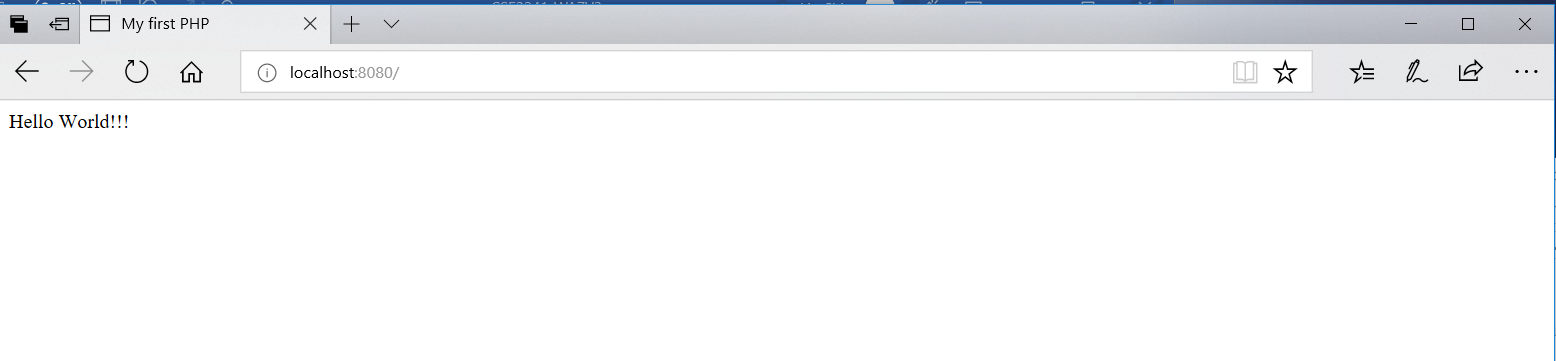
*// if your root directory is not where php.exe is located, you need to use the ‘-t’ option. For example, if you run PHP other than under its root directory (for example, C:\tmp), you will need to specify the entire path to the php.exe and its root directory with “-t C:\php” as in “C:\php\php -S localhost:8080 -t C:\php”.*

You should see something shown below in your cmd window,



* On your browser, enter the URL //localhost:8080

You should see some webpage similar to what is shown below. This means your php is installed correctly and the Apache server came with it is working.

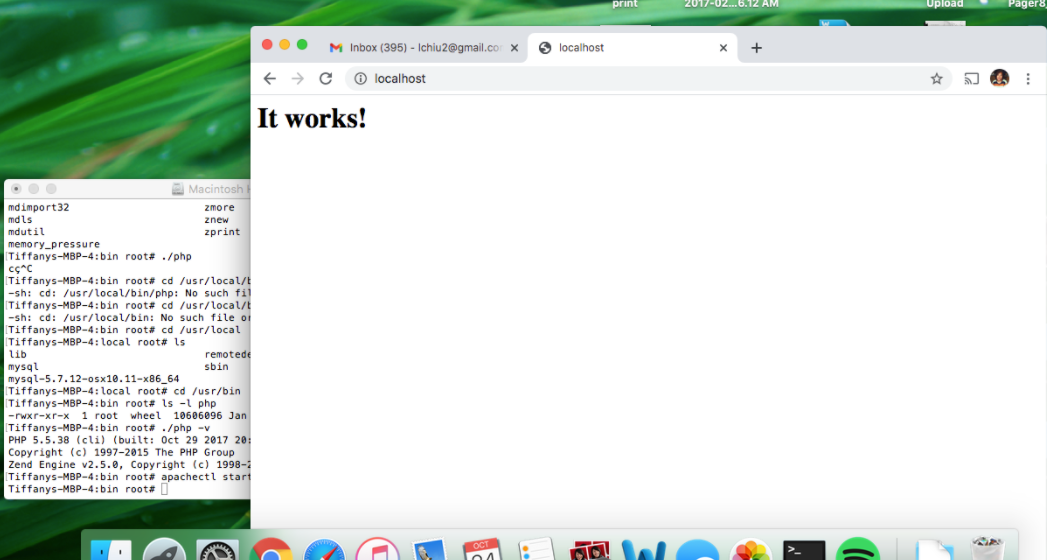


***For MacOS Users:***

MacOS comes with PHP and Apache already loaded.

Task 1: Start your Apache on the Mac

Become root by ‘sudo su –‘ and then run ‘apachectl start’ (‘apachectl stop’ to stop it.) To test it, open the browser and enter ‘http://localhost/’ and if it works, you will see



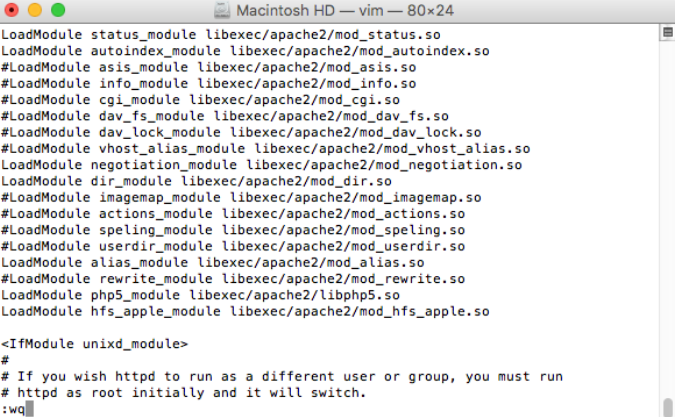
Task 2: Configure your PHP

Uncomment out the line ‘#LoadModule php5\_module libexec/apache2/libphp5.so’ in httpd.conf. To do that, cd to the directory where that file is located. Save a copy before you make changes.

cd /etc/apache2

cp httpd.conf httpd.conf.save

vi httpd.conf



Task 3: Test your PHP

Create the index.php file (shown below) in the php document root directory.

<html>

<head>

<title>My first PHP</title>

<body>

<? Php

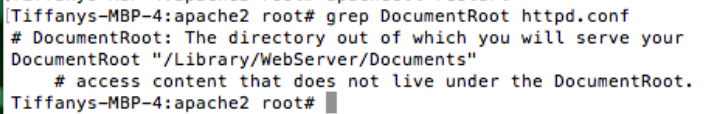
Echo “Hello World!!!”

?>

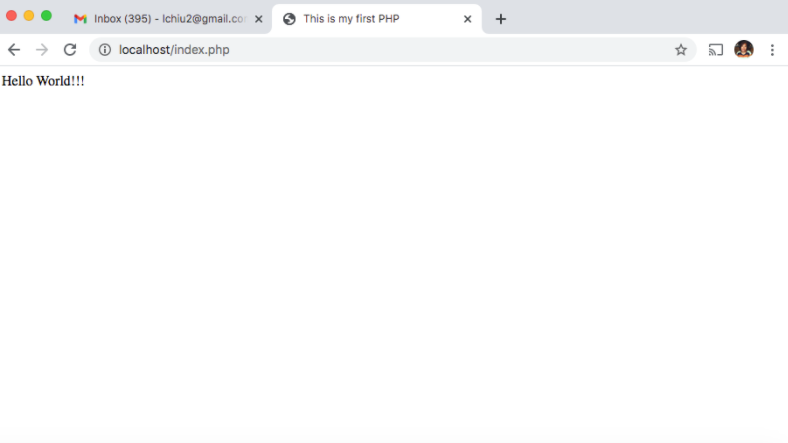
</body>

</html>

To find out where the document root directory is, run , ‘grep DocumentRoot httpd.conf’. By the default, it is /Library/WebServer/Documents.



Enter ‘localhost/index.php’ into your browser.



**Individual Project – CSE3241 (7022) AU20**

You need to submit the PHP file(s) so they can be checked by the grader or me. Please name your landing page as index.php. Make sure all your php files are from the same directory. No absolute path used in your code; otherwise, they will not work in my environment. Please stick to the table names and column names listed in this document for the same reason explained above. If your database name is not ‘COVID’, please point it out in a ‘readme.txt’ document.

COVID Technologies sells software products to its customers. Sometimes the software is bundled with hardware equipment but not all the time. Only the servers need to be tracked for this project. After some successful years of business, its CEO found out the company is lacking the tool to track sales to further their business in selling warranties, software upgrades and hardware replacements after a new sale. You are hired to develop such tool that will allow the user (i.e. salespersons) to enter the information on a purchase order into the database. The database schema has been designed by a senior employee in the company and your job is to write a web application for the user to enter the information into the database.

Database schema for this project is defined as follows,

CUSTOMER

|  |  |  |  |
| --- | --- | --- | --- |
| cusID | cusName | contactName | contactNo |
| 11001 | Clairs | James Smith | 2123546000 |
| 11002 | StarBucks | Jenny Will | 5134445000 |

*CusID is an identifier to uniquely identify each customer company.*

HARDWARE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| machineID | manufacturer | model | vendor | EOL |
| m0001 | HP | DL380 | NewTech | 2025-10-30 |
| m0002 | SUN | NS2 | ServerDepot | 2027-01-01 |

*MachineID is an identifier to uniquely identify each type of server and its model. (EOL = end of line or end of life)*

APP

|  |  |  |  |
| --- | --- | --- | --- |
| appID | appName | Rel | EOL |
| a000120 | SalesManager | 2.0 | NULL |
| a000121 | SalesManager | 2.1 | NULL |
| a000211 | Primo | 11.0 | 2023-06-30 |

*AppID is an identifier to uniquely identify each software release. Some applications do not have EOL set yet and therefore, contains a NULL value.*

CUSENV (Customer’s Server(s) Information)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cusID | sysNo | machineID | purDate | Support | OS | Web | Java | PHP |
| 11001 | 1 | m0001 | 2017-01-31 | 2022-12-21 | Centos7 | Tomcat7 | 8 | 5.5 |
| 11001 | 2 | m0001 | 2018-03-21 | 2023-12-21 | Centos8 | Tomcat7 | 8 | 7.1 |
| 11002 | 1 | m0002 | 2017-9-30 | NULL | Redhat6.2 | Apache2.4.2 | 7 | NULL |

*CusID and machineID are foreign keys. A customer can purchase more than one system, each of which will have its own server and OS version, identified by a sysNo.*

CUSAPP (Customer’s Applications Purchased)

|  |  |  |  |
| --- | --- | --- | --- |
| cusID | appID | purDate | Support |
| 11001 | a000120 | 2019-06-01 | 2021-06-01 |
| 11001 | a000211 | 2019-10-31 | 2021-10-31 |
| 11002 | a000120 | 2018-04-15 | NULL |

*CusID and appID are foreign keys. A customer can purchase more than one application. This project does not track which application runs on which hardware.*

Each purchase order contains the following information:

* Customer Name (cusName), contact person (contactName), contact phone number (contactNo).
* Date of the purchase (all products purchased on this order will use this date as the start of the warranty contract if purchased.)
* Hardware (optional)
  1. Server manufacturer (required)
  2. Model (required)
  3. OS (required)
  4. Support contract end date (optional)
  5. Web server (optional)
  6. Java version (optional)
  7. PHP version (optional)

*// If the customer purchases the server from COVID Technologies, there is no need to track vendor and EOL information because they are already stored in the HARDWARE table.*

* Software (at least one or more on the same purchase order)
  1. Application name (required)
  2. Application release (required)
  3. Application support contract end date (optional)

Your web application (referred to as ‘the program’ from this point on) needs to meet the following requirements: For each purchase order,

1. Prompt the user for the customer name, if it does not already exist in CUSTOMER, create one for that customer. Use the largest cusID + 1 as the new cusID. If it already exists, retrieve its cusID.
2. Ask the user if hardware is included on this purchase order. If no, skip to the next step. Otherwise, find out the internal machineID representing the server manufacturer and model, and create one entry in the CUSENV table. If this is a new customer, use ‘1’ for sysNo, otherwise, use the highest sysNo +1 as the new sysNo. If the machine information cannot be found in HARDWARE, exit the program.
3. Ask the user to enter the applications purchased one at a time. For each application, create one entry in the CUSAPP table. If the application does not exist in the APP table, exit the program.

Something to think about: When the program terminated because of errors, should all updates be rolled back?

Rubrics for this project: (points are out of a total of 100)

|  |  |  |
| --- | --- | --- |
| Capabilities | Points deducted | Bonus points added |
| Not able to add new customer | -5 |  |
| Not able to retrieve cusID of existing customers | -5 |  |
| Not able to retrieve machineID from HARDWARE | -5 |  |
| Not able to retrieve appID from APP | -5 |  |
| Web Application does not allow users to enter all needed information | -10 |  |
| Web Application does not maintain referential integrity in the tables. | -10 |  |
| Failed to insert new entries in CUSENV | -20 |  |
| Failed to insert new entries in CUSAPP | -20 |  |
| Rollback changes when error occurred |  | +5 |
| Use a dropdown list with NEW for selecting customers |  | +5 |
| User dropdown list for machine or application |  | +5 |