**John Grier**

**University of Arizona Global Campus**

**CST499 Capstone for Computer Software Technology**

**Instructor: Joe Rangitsch**

**May 31, 2024**

**Explain how to run a PHP file in XAMPP.**

Install the XAMPP server on your MAC OS. After you start your Apache and MySQL, click the start button of the XAMPP control panel.

If it displays an error message while starting it, go to the config file and change the port number. Then it will start, and the bottom green color will appear.

After you start the server, check the Apache port number. Then, go to any of your web browsers, type "localhost/filename" into the URL box, hit the enter button, and the index file for your project will be executed.

• Apache and MySQL must be started before you can run or write a PHP program.

• After starting both servers, you have to write a program I use Sublime Text.

• Once it has been written, save it under "index.php."

• Then copy that file index.php to htdocs.

• Type the following URL into your browser: http://localhost/index.php

• Hit your 'run' button in your code, and watch it run in the browser.

**These HTML5 tags are primarily used for head, title, meta, link, script, div, nav, h1, ul, and user interface.**

Head – This tag refers to the ‘head’ part of the page. The head provides descriptive information about the document, such as the title of the document, any style sheets or other JavaScript files it uses and other kinds of meta information that search engines and other programs make use of.

Title – This tag fills in what will be the label used for the tab or window of the browser when it is displaying this page.

Meta – This tag inserts meta information such as the declaration that the character encoding for the document is UTF-8.

Link – ‘The HTML <link> tag defines a relationship between the current document and an external resource. The <link> tag specified in the HTML document head element and is used to include a resource, such as a style sheet, in the document. The <link> element also defines other meta-information about the document such as the relationship to the linked document.’ (Quackit.com)

Script – ‘The HTML <script> tag is used for declaring a script (e.g JavaScript) within your HTML document.’ (Quackit.com)

Div – this creates a container which can function as a logical grouping of content and which may contain other HTML elements, including text, as well as other containers.

Nav– ‘The HTML <nav> tag indicates a section of a HTML document that contains navigation links to other parts of the document (or other documents). All web sites have sections designed for navigational links so that the user can navigate the site. These links should be inside a <nav> tag.’ (Source: Quackit.com)

H1– this makes the text a header of the largest size on the page, and it will be differentiated from the body text of the page, which will be smaller, more compact, and nested below the header text. It will usually be centrally aligned on the page.

Ul – this designates an unordered list. List items will have bullets rather than numbered entries

Ui – there is no such a tag.

**My registration page is shown in the screenshot below.**

A screenshot of a computer

Description automatically generatedA screenshot of a registration page

Description automatically generated

A screenshot of a computer

Description automatically generated

**In this section, I explain how to create the registration page and save the user's information in the database.**

FirstName, Last Name, Address, Email, and Phone are now stored in the database.

Password, salary, SSN, and phone number.

 To make the website look neat, add a div class="container">.

H2>Register</h2 identifies the user's page as the right one.

Using the form, create an action and method. Based on the action, the script is run.

After the form is submitted, the post method is used.

I then created labels and input for all columns.

I have created a submit button

An already-used function can be entered

**Here is a screenshot of the original PHP source code for the registration page.**

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

**Discuss the MySQL database functions that you used and the steps you took to create the database connection custom class.**

Step1: Creating New Database

1) Go to new and enter the new to create the database in the left-hand column.

2) You can create a new database by entering the name (in this case, employee portal). Click on the Create button when it is complete.

Step 2: As soon as I had created my database, I then created a new table to store our data. To create a new table, I found my cursor on the object palate and line will be appeared, I selected the table. Then tab grow and highlighted all the fields. Input: Create TABLE [tbluser] ( [CustomerID] INTEGER IDENTITY PRIMARY KEY,

[FirstName] C(50) NOT NULL, [LastName] C(50) NOT NULL, [Email] C(50), [Phone] C(50), [Address] C(100), [City] C(50), [State] C(50), [Zip code] C(10), [Pass] C(30) );

Select Go From ribbon to create table.

My new table will be created.

The first (Id)

The second one (email)

The password (password)

The first name of (firstname)

The last name of (lastname)

The address (address)

The telephone number (phone).

Income (Salary)

The Social Security Number (SSN)

Step 3: Creating the SQL table:

1. As written above I was able to fill in the names within each column on the name line.

2. After that, identify the datatype of the given data which that will be inserted to the database. Here I mentioned ID , phone, salary & SSN as at all four integers .remaining all column of the SQL table I kept VARCHAR because it will be storing a string.

3. Inside the column length/ values, I wrote down the length which is the biggest among my dataset that going be in the table

4. At the end, I check the auto increment check box and make sure that the ID will be incremented automatically whenever you add new data.

**Screenshot of the table after saving the user information in the database.**

A screenshot of a computer

Description automatically generated

**Below is my example of how I use SQL script. I have added FirstName, LastName, Address, Email, Phone number, Salary, SSN and Password to be stored in the database.**

-- phpMyAdmin SQL Dump

-- version 5.2.1

-- https://www.phpmyadmin.net/

--

-- Host: localhost

-- Generation Time: May 31, 2024, at 07:19 PM

-- Server version: 10.4.28-MariaDB

-- PHP Version: 8.2.4

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `employeeportalwk4`

--

-- --------------------------------------------------------

--

-- Table structure for table `profile`

--

CREATE TABLE `profile` (

`id` int(11) NOT NULL,

`email` varchar(100) NOT NULL,

`pwd` varchar(255) NOT NULL,

`firstName` varchar(100) NOT NULL,

`lastName` varchar(100) NOT NULL,

`address` varchar(255) NOT NULL,

`phone` int(100) NOT NULL,

`salary` int(255) NOT NULL,

`SSN` int(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `profile`

--

INSERT INTO `profile` (`id`, `email`, `pwd`, `firstName`, `lastName`, `address`, `phone`, `salary`, `SSN`) VALUES

(4, 'joseph.rangitsch@faculty.uagc.edu, 'Jrang321', 'Joseph', 'Rangitsch', '180 S. Arizona Ave. Suite 301', 5555555555, 286000, 55555555),

--

-- Indexes for dumped tables

--

--

-- Indexes for table `profile`

--

ALTER TABLE `profile`

ADD PRIMARY KEY (`id`);

COMMIT;

As part of the login form, there are input fields for usernames and passwords, as well as a POST method. Once the login form is submitted, a database check is required to determine if the user already exists. In the login.php file, we check if the user exists and redirect the user to the destination if the user is present. Otherwise, the user will be required to enter the credentials. Upon changing to another destination, a new session will be created.

***A screenshot of a login page

Description automatically generated***

***A login form with black text

Description automatically generated***

**A screen shot of a computer

Description automatically generated**

**Screenshots of the “profile.php” page layout.**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Screenshot of the profile page after retrieving the following information for the employee: email, password, first Name, last Name, address, phone, salary, SSN.**

A screenshot of a computer

Description automatically generated

**Screenshot of the “profile.php” page source code.**

**A screen shot of a computer

Description automatically generated**

Whenever a user logs in to the site, the welcome page, which is already present in the footer of all the different pages, is used. When the user logs in, they will be redirected to this site, welcome.php. The logout link must be used by creating a new PHP file titled logout.php, just as we created login.php and welcome.php. The script will be initiated by launching the session, since it is the main script. Once the file is created, we can begin writing the code. Once the script is completed, the session will be executed. When we are logged in, we must redirect the users to the login page by unsetting the session and destroying the session.

A screen shot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated