CIS 435-525 Fall 2019 Project 3:

Demonstration Registration with PHP and MySQL

Assigned: 10/31/2019
Due: **11/21/2019**
Total Points: **150**

Assignment

There are 36 students in the web technology class. The students will demonstrate their projects at six different one-hour long time slots. Up to 6 students can give their demonstrations in a given time slot. The professor decides to have a registration webpage to allow students to sign up for one of the time slots. A student visiting the page should be able to submit his/her *UMID*, *first name*, *last name*, *project title*, *email address*, *phone number*, and *book a seat in one of the available time slots*. A student is uniquely identified by his/her UMID.

The submitted data should be stored in a MySQL database which is maintained on a server. The webpage and the server should interact with each other at every step of the registration process. The page should show how many free seats are available in each time slot, announcing and blocking all fully booked time slots. After a student makes a data submission, it should check whether the student has been already registered. If not, the data is stored on the server and the student is notified about her registration. Otherwise, if already registered, the student should be prompted to ensure that she wants to change her registration to the new section (and removed from the current one she is registered for). For example, the time slots may look like the following list:

1. 12/9/19, 6:00 PM – 7:00 PM, 6 seats remaining
2. 12/9/19, 7:00 PM – 8:00 PM, 5 seats remaining
3. 12/9/19, 8:00 PM – 9:00 PM, 3 seats remaining
4. 12/10/19, 6:00 PM – 7:00 PM, 2 seats remaining
5. 12/10/19, 7:00 PM – 8:00 PM, 4 seats remaining
6. 12/10/19, 8:00 PM – 9:00 PM, 0 seats remaining

In addition, you need to write a **separate** webpage that will display the list of students (including their UMIDs, names, project titles, email addresses, phone numbers, and time slots) who are registered, after querying the database.

Requirements

1. For your assignment, you should use HTML, CSS, JavaScript, PHP, and MySQL.
2. The database must fully implement the registration process, maintaining the submitted information. The server and page behavior must meet the requirements listed above.
3. The specific look and feel of the pages as well as the database implementation is left intentionally vague, allowing considerable design freedom on your part. However, the page should have a nice look and the code should satisfy common standards.
4. User inputs must be validated at the server side in PHP. The validation at the client side in JavaScript is optional. The first and last name fields cannot be empty and consist of alpha letters only. UMID must be exactly 8 digits. Email must be a valid email address.

Phone number must be in the form 999-999-9999. *Hints, input validation can be done with PHP filter or regular expressions.*

If there are any errors, your program must **highlight the error input** and keep the correct inputs. You should not ask a user to "go back" to the previous page or input everything again.

Deploy Your Project using Amazon Web Services

Hints:

I suggest that you install XAMPP on your personal computer. Then, you can develop and test your program locally. When all done, you can then deploy it to the AWS.

The following are the instructions on how to create, view, and deploy your web applications with AWS .

Step 1: Create a secret key pair
At the EC2 dashboard under the management console, create a new secret-key pair, and download the private key to a location on your computer.

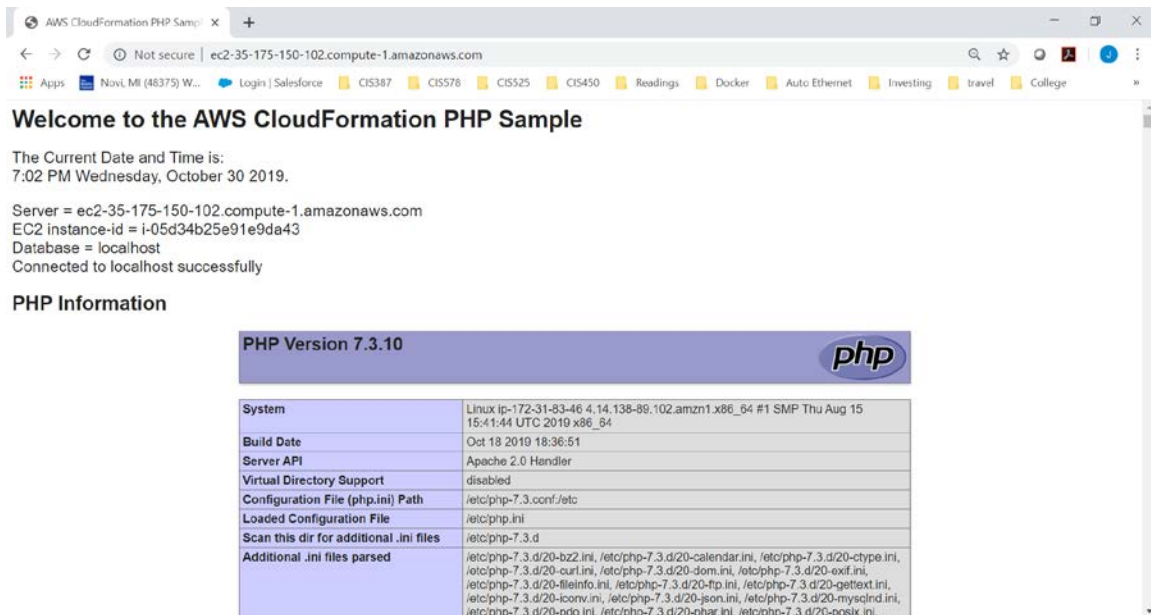**Step 2: Creates a LAMP stack on a single Amazon EC2**

At the CloudFormation dashboard, click the **"Create Stack"** link, and then follow the step by step instructions to create an AWS EC2 with LAMP stack. The acronym LAMP refers to first letters of the four components of a solution stack (Linux, Apache, MySQL, PHP).

Important Tips:
- Please make the following choices and pick the default values for other options:
    - Specify Template: **Upload a template file** (*Project3-AWS-LAMP-template.json* file, which is available on canvas under Files->Projects-P3 folder)
    - Stack name: **Any name (e.g cis435P3)**
    - DBName, DBPassword, DBRootPassword, DBUser: **Any value and take a note of your choice.**
    - KeyName: pick the existing key pair you created at Step 1.
    - Instance type: **t2.micro (default)**

Step 2: Test your LAMP server

After the Stack creation is completed, go to EC2 Console, find the Public DNS (ipv4) of your instance, and copy it to the URL of your web browser. You should see the default page as follows:

Step 3: SSH into your machine

For details on how to SSH into the machine, go here (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html).

- o Log on as "ec2-user"
- o For Windows, PuTTY ssh client is relatively easier to use.

Step 4: Setting file permissions

Apache **httpd** serves files that are kept in a directory called the Apache document root. The Amazon Linux Apache document root is /var/www/html, which by default is owned by root.

To allow the ec2-user account to manipulate files in this directory, you must modify the ownership and permissions of the directory. There are many ways to accomplish this task. Here, you add ec2-user to the apache group, to give the apache group ownership of the /var/www directory and assign write permissions to the group.

Add your user (in this case, ec2-user) to the `apache` group.
```
sudo usermod -a -G apache ec2-user
```

Log out and then log back in again to pick up the new group, and then verify your membership.

Log out (use the exit command or close the terminal window):
```
exit
```

To verify your membership in the `apache` group, reconnect to your instance, and then run the following command:

```
groups
```

Change the group ownership of `/var/www` and its contents to the `apache` group.
```
sudo chown -R ec2-user:apache /var/www
```

To add group write permissions and to set the group ID on future subdirectories, change the directory permissions of `/var/www` and its subdirectories.
```
sudo chmod 2775 /var/www
```

Then:
```
find /var/www -type d -exec sudo chmod 2775 {} \;
```

To add group write permissions, recursively change the file permissions of `/var/www` and its subdirectories:
```
find /var/www -type f -exec sudo chmod 0664 {} \;
```

Now, the ec2-user user can add, delete, and edit files in the Apache document root.

Step 5: Install phpMyAdmin

To install phpMyAdmin, first install the required dependencies.
```
sudo yum install php73-mbstring.x86_64 php73-zip.x86_64 -y
```

Restart Apache.
```
sudo service httpd restart
```

Navigate to the Apache document root at `/var/www/html`.
```
cd /var/www/html
```

Next we need to download the latest release of phpMyAdmin
```
wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-
languages.tar.gz
```

Extract the package.
```
tar -xvzf phpMyAdmin-latest-all-languages.tar.gz
```

Check the files in the current directory and note the name of the extracted phpMyAdmin version.
```
ls
```

Then change the name of the resulting directory to something more manageable. 4.9.1 below, will be changed the version you noted in the last step.
```
mv phpMyAdmin-4.9.1-all-languages phpMyAdmin
```


In a web browser, type the URL of your phpMyAdmin installation. This URL is the public DNS address (or the public IP address) of your instance followed by a forward slash and the name of your installation directory. For example:
http://*my.public.dns.amazonaws.com*/phpMyAdmin

Where *my.public.dns.amazonaws.com* is your own public DNS.

Step 5: Deployment

1. Using FileZilla, copy all your project files into the /var/www/html folder.
2. Using your local phpMyAdmin, **export** your project database into a SQL file.
3. Using phpMyAdmin at your AWS EC2 instance, **import** the database from the SQL file.

How to (S)FTP into your EC2 instance

In this guide, we will be using FileZilla. If you need it, google FileZilla, and download the CLIENT. Not the server.

Step 1: Get your Public DNS address.
Step 2: Find your .pem key that you got when you create your key pair.
Step 3: Open FileZilla.
Step 4: Go to the FileZilla settings, and on the left, click SFTP.
Step 5: Add a new private key. (Your .pem key)
Step 6: At the top in the Quickconnect bar, put your Public DNS in the host, ec2-user, port 22
      (Port 22 is SFTP rather than FTP), and NO PASSWORD.
Step 8: Click Quickconnect.
Step 9: You are done!

What to turn in:
1. Upload all your files as **a single ZIP file** to the Canvas under the "P3" folder. Be sure to include all relevant HTML, CSS, JavaScript, and PHP files, and include an SQL dump of your database including all the relevant table(s) that you are using for this assignment.

2. **Write down the URL of your web site in the comment section**, when submitting.