

Course Project

CS 348 - Spring 2020

Supplementary Material II:

PHP & MySQL Sample Application on Google Cloud Platform (GCP)

This sample application works with Google Cloud Platform (GCP) (<https://cloud.google.com/>). GCP offers multiple choices of platforms and database systems. This document presents a simple PHP application on GCP's App Engine (GAE) together with MySQL.PHP (<https://www.php.net/>) is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. MySQL (<https://www.mysql.com/>) is one of the most popular database system used with PHP.

If this setting is chosen for your project, you may consider web as the user interface and in Milestone 0, you should get familiar with the following basics of PHP and MySQL on GCP.

- Creating a database system instance and a database on MySQL.
- Creating a PHP based application on GCP's App Engine.
- Establishing a Connection from your application that connects to your database.
- Processing SQL Statements with PHP.
- Displaying query results on the web with PHP.
- Retrieving and modifying values in the database using PHP.

Get Started

TO use the services, you need to register for an account. You may check free products via <https://cloud.google.com/free/> (New customers will also get \$300 to spend on GCP products during their first 12 months.) This course also provides \$50 student coupon (information will be sent later separately). Coupon can be retrieved with domain @uwaterloo.ca.

To support the sample application, there are 2 main components:

- a) Google App Engine to host your PHP script and to serve the landing page to the user;
- b) Cloud SQL to host your database.

Here are the main steps:

1. Go to the CONSOLE of GPC and **create a new project** with name "cs348demo" and id"cs348demo-spring"¹. More information about this project can be found on the dashboard. The project name would translate into your site name, cs348demo-spring.appspot.com. You can choose other project names, but remember to update all the corresponding files. Double check if the project is linked to your billing account. If not, link it using account management under "BILLING" to ensure your service are available.
2. Select project "cs348demo" as your working project. Under this project, **create a new database system instance**. Go to SQL under STORAGE, and click "CREATE INSTANCE". Choose MySQL and create a MySQL Second Generation instance. You need to fill in the following information
 - Instance ID: "cs348demo-db"
 - Root password: "password"
 - etc.

¹The project number and project ID are unique across Google Cloud. If another user owns a project ID for their project, you won't be able to use the same project ID. So you can not create the id exactly the same. <https://cloud.google.com/resource-manager/docs/creating-managing-projects>

It takes a few minutes to create the database system instance. After the instance is created, find the instance connection name, “cs348demo-spring:asia-east2:cs348demo-db” for this example. You will need these information for the PHP codes to make connection later.

3. On this database, you can create a database by clicking “create database” under MASTER INSTANCE. You can populate the database with several approaches, e.g. using google cloud shell <https://cloud.google.com/sql/docs/mysql/quickstart>.

- First open google cloud shell, and then enter

```
$ gcloud sql connect cs348demo-db -user=root
```

- Enter your root password. You should see the mysql prompt.
- Follow the instructions in <https://cloud.google.com/sql/docs/mysql/quickstart>, you will create a database name “guestbook”. This database has a table “entries”. We will use these examples for the remaining codes.

4. You need to **enable the Cloud SQL Admin API** so that your application can communicate with the database. To do this, go to APIs & Services, search for “cloud sql admin API”. Click “enable” button.
5. Under the same project “cs348demo”, go to APP Engine under COMPUTE and **create an app** by clicking “Create app”. Choose “PHP” for language and “Flexible” for environment. You should see “Your App Engine app has successfully been created”. You can check the application browser by clicking "Storage -> Browser" under STORAGE. You will find “cs348demo.appspot.com”.
6. After successfully creating an App Engine, you can work with the Google Cloud shell or download the Cloud SDK and install it into your computer. This Cloud shell/SDK was needed to deploy your project codes into GCP.

Download sample code in the Cloud shell/SDK (Thanks Prof. Xi He):

```
$ git clone https://github.com/heqianjane88/cs348demo.git
```

Before you deploy the code, you need to prepare the following items:

- Update the names (of project, database system instance, database, user if different from our example) and password in the downloaded files.
- Deploy the application by typing

```
$ cd cs348demo
$ gcloud app deploy
```

- Open the site cs348demo-spring.appspot.com/ in your browser. You will see “Connected successfully first guest (I got here!) second guest (Me too!) 2” if everything works fine.

The deployment will take quite a while (15 mins). Hence, before deploying your code, you could test the code on localhost:8080 with the web preview on the cloud shell by working on the following items:

- Update the names (of project, database system instance, database, user if different from our example) and password in the downloaded files.
- Update the line in index.php for \$conn to `$conn = new mysqli("127.0.0.1", $username, $password, $dbname, 3306);`
- Run composer.json in the same directory to get all files for localhost:

```
$ composer install
```

- Install the proxy in the root directory following the instructions in <https://cloud.google.com/sql/docs/mysql/connect-admin-proxy>, which can be summarize into these few points. First, download the proxy and make the proxy executable. You will have a executable file `cloud_sql_proxy` in the root directory. Next, start the proxy in the Cloud Shell by running

```
$ cd ~
$ wget https://dl.google.com/cloudsql/cloud_sql_proxy.linux.amd64 -O
cloud_sql_proxy
$ chmod +x cloud_sql_proxy
$ ./cloud_sql_proxy -instances=cs348demo-spring:asia-east2:cs348demo-db=tc:3306
&
```

- Then switch back to the directory `cs348demo` and run the code on the local host.

```
$ cd cs348demo
$ php -S localhost:8080
```

The Web Preview button can be found on the top right of the Cloud Shell taskbar <https://cloud.google.com/shell/docs/using-web-preview>. Click on it and select “Preview on port 8080”. The browser will then show the preview of the `index.php` content.

Notes for Project

The sample code shows a toy example to start the project. The data management application should be relatively substantial, but not too enormous. Several project ideas are described at the end of the main project document, but you are encouraged to come up with your own. As web will be considered as the user interface for this application, the project team needs to ensure the interface support a rich set of interactions between users and the database (e.g. enter a value, delete an entry, etc.) The interface can cover a rich set of queries and also handles exceptions (e.g. invalid input, etc.)

You are free to choose other applications other than PHP, for the user interface.

Here are several good tutorials and links for your reference:

- Deploying PHP Website to Google App Engine:
<https://medium.com/@kerion7/creating-a-simple-php-application-on-google-app-engine-c75>
<https://www.youtube.com/watch?v=QIWBxx0mE00&feature=youtu.be>.
- MySQL + Google Cloud Platform: Java Setup MySQL
<https://www.youtube.com/watch?v=dZ1wyxF0GAI>.