

COMP 4513

Assignment #1:

Data-Driven PHP and JavaScript

Version 2.0 (additions in yellow).

Due Tuesday February 23, 2016 at 11am

Overview

This multi-part group assignment will be a part review/refresh of PHP, and part new exploration of JavaScript. Unlike assignments from me in previous years, this one will be in two parts: both will be due at the same time but the first part should be completed before the second. Groups can contain three or four people. Please don't ask to be a group of five, eight, eleven, or any other number larger than four.

Submitting

You will not be submitting your source code. Instead, I will mark your code from a GitHub repository. I will mark the functionality by visiting some type of live server. Thus, you will submit your assignment by emailing me a short message consisting of the group member names, the GitHub repository URL, and the web address where I will be able to find your working assignment.

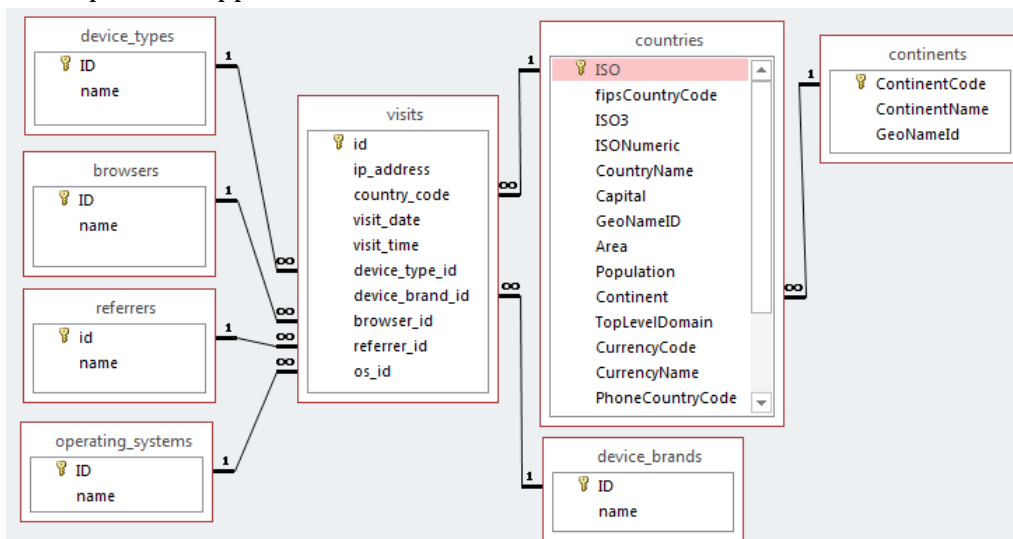
Grading

The grade for this assignment (both parts) will be broken down as follows:

Visual Design and Styling	15%
Programming Design	15%
Functionality (follows requirements)	70%

Data Files

You have been provided with the necessary SQL script that you will need to import into MySQL. It represents some analytic information for some web site. I have also included a Microsoft Access version if you want to use that to experiment with queries (**notice updated diagram below**). The SQL script is quite large so you may have to modify some of your environment's defaults. Alternately, you can import the zipped version instead.



Overview

In this assignment you will be creating an **Admin Dashboard** for the data in the visits database. An admin dashboard is a set of user interface elements and menus that give the user quick access to information. Usually, admin dashboards will contain information such as messages, notifications, graphical stats, entry count, analytics, schedules, and so on.

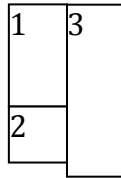
Requirements

1. You will NOT use Bootstrap!

Instead I would like you to use a “lite” CSS framework that is based on Google Material Design principles. For a quick introduction to Google Material, visit <http://materialdesignblog.com/making-use-of-googles-material-design-guide/>. To learn more about Google Material, please read through <https://www.google.com/design/spec/material-design/introduction.html>.

Some examples of “lite” implementations of Material include Materialize CSS or Material Design Lite. To find out more, visit <http://www.sitepoint.com/top-5-material-design-frameworks-use-2015/>. Although you will not be using bootstrap, you might want to do some web searches for “admin templates” to get some ideas about the basic visual design of a typical admin dashboard.

2. To begin, you are going to want to create a working data access layer in PHP. I have provided the code from Chapter 14 in the book from which you can begin. Your table gateway class for the visits table is going to need the most functionality. It will need the ability to filter and count by country code, browser, device type, device brand, and referrer. It will also need the ability to filter by date: for instance, visits in the past month, or between date ranges.
3. To demonstrate this functionality, create a page using your admin design that displays three material cards. Cards one and two should be in the same layout column; the third card should be “besides” the other two:



In the first card, it should display a table of browser names and their percentage of the visits (note the diagram shows the rough look of the site not its exact dimensions or scale).

In the second card, it should provide a drop-down list of device brands. When the user selects a brand, it should display in the same card a count of how many visits there have been for that brand.

In the third card, it should provide a drop-down list of continents. When the user selects a continent, it should display in the same card a material table of the countries from that continent and a count of how many visits from that country.

This page should be the home page for the assignment.

Everything on the next pages of this assignment is also new.

4. You are going to need to create a series of web services that provide an HTTP interface to your PHP data access layer. These services will all return JSON data and it will be up to you to determine the exact API. Chapter 17, pages 756-774 provides guidance for the construction and consumption of web services.

For inspiration, you can try consuming one of the sample web services on my domain. Try entering the following into your browser:

<http://www.randyconnolly.com/funwebdev/services/travel/serviceTravelCountries.php>

This returns a JSON array of all the countries with images in the Travel sample database. Now change it to:

<http://www.randyconnolly.com/funwebdev/services/travel/serviceTravelCountries.php?continent=NA>

This returns a JSON array of all the countries in North America. Try instead the parameters `iso=CA` (just a single country, Canada) or `begins=aus` (countries whose name begins with AUS).

PHP web services differ from regular web pages in that they don't return HTML, but return XML or JSON. As such, you will need to change the response header using the PHP `header()` function:

```
// Tell the browser to expect JSON rather than HTML
header('Content-type: application/json');
// needed for javascript clients from another domain
header("Access-Control-Allow-Origin: *");
```

5. To demonstrate the functionality of the web services, you are going to create a new separate page that has the same functionality as step 3 above. Note that this is a new page; it doesn't replace the previous one. Rather than generating the tables and lists in PHP, this page will use JavaScript **and** JQuery to generate them. The data will come from your web services created in step 4. Unlike in step 3, the event handling for the drop-down lists will now happen completely in JavaScript/JQuery AJAX. You will probably want to use the JQuery `$.get()` function to request data from your web services.

You must display some type of loading animation while the data is being retrieved. Find some type of loading animation image if your google material framework doesn't already have one and then show or hide it via JavaScript. Do **not** use a third-party JQuery plugin for this loading animation.

6. Add an about us page with the group member names, the project's github URL, and links to any third-party resources used.
7. Ensure your site's navigation system makes it easy for me to find the various pages.

8. You are going to create a page named visit browser. This will provide the user with a way of viewing subsets of the big visit data set. The page will consist of two cards: a filter card (which allows the user to specify filters) and a data card (which displays the data).

Because the data can get quite large (and I don't want you to worry in this assignment about paginating the results), use the MySQL LIMIT keyword to limit the number of returned rows to 100.

In the data card, display in a table the following info: visit date, visit time, ip address, and country name. Data should be sorted by visit date and time. You might need to also include the visit id in a hidden element.

For each data item, add an icon button at the beginning or end of the row (use an icon that visually indicates to the user that clicking on it will view more information). Clicking on this button should display the complete data record for the clicked row (i.e., visit date, visit time, ip address, country name, display device type name, device brand name, browser name, referrer name, and operating system name) in a modal dialog that appears "above" the page. This dialog should have some type of close button to make it disappear. Your google material template might already have that functionality. If not, feel free to use some other third-party modal dialog plugin (be sure to properly cite it in your documentation).

In the filter card, allow the user to filter the displayed visit information using filters. Country name, device type, device brand, browser name, referrer name, and operating system name should all be filters as drop-down lists. Country should be an auto suggest text box that uses some third-party JQuery autocomplete plugin (see pages 676-8 for example ... be sure to properly cite it in your documentation). The visit data in the data card should change automatically when a filter is changed (be sure to display a loading animation while getting the data).

9. You must create a new page that displays your web services data within some charts. This page is going to have three charts on it (each chart should be within its own “card”). You are going to use Google Charts for these charts. The Google Charts Javascript objects and functions do all the hard work for you. What you will have to do is convert your JSON data into the format expected by Google’s chart objects. For all of these charts, display some type of loading animation while the data is being retrieved.

Chart 1 – Area Chart

The Visits Per Month area line chart (see below... your data will be different) must show daily web site visit counts for a given month in 2016. Provide a drop-down list of months; when the user selects a different month, regenerate the chart. Make the default starting month January.



Chart 2 – Geo Chart

The Visits in a Month Geo Chart (see below... your data will be different) should display a geographic representation of the visit data for a given month in 2016. Provide a drop-down list of months; when the user selects a different month, regenerate the chart. Make the default starting month January.

There is a lot of data here with many countries represented; to make it more manageable visually, show only those countries from which there was 10 or more visits during the month. The countries should be color-coded to indicate the relative number of visits; also, the visit count and country name is visible, either as a label or tooltip for each country. *Both of these two requirements are actually handled automatically by Google Charts.*

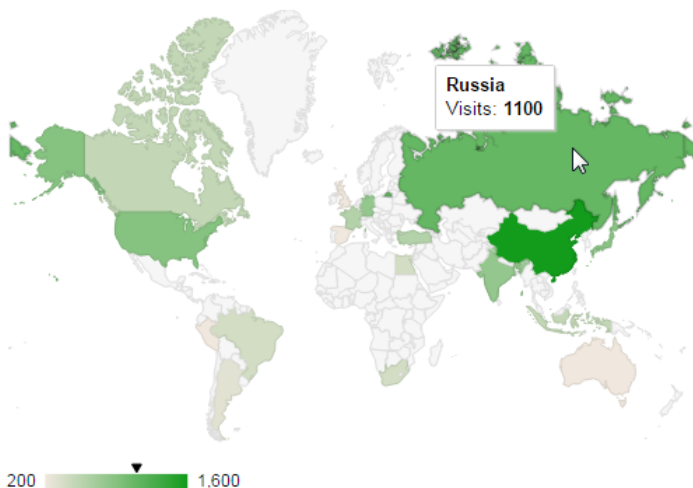
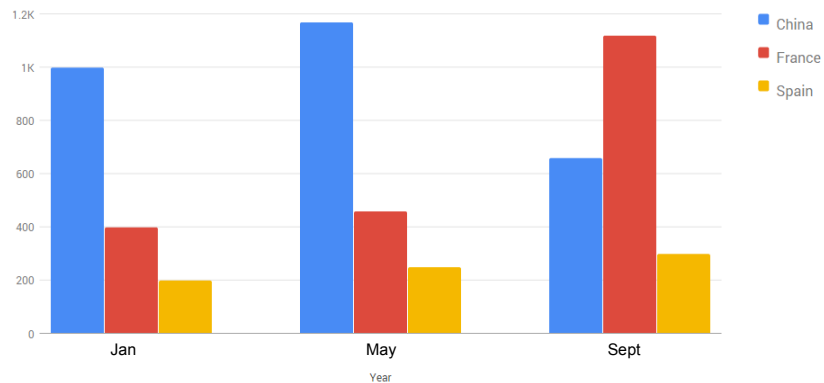


Chart 3 – Column Chart

Display visit data in a column chart. Your page should provide three drop-down lists that allow the user to select a country. Only show the top ten countries (sorted by name) in terms of # of visits (they are the same countries for each of these months) in these lists. Add a Chart It button (it should only become enabled once the user has selected three countries) that retrieves the relevant site data for the three selected countries and displays the totals for the months of Jan, May, and Sept. There should be a Switch button that switches the axis of the chart (see below ... your data will be different).

Site Visits
2016



Site Visits
2016

