

COMP 3512

Assignment #1: Data-Driven PHP (*version 1*)

Due Monday February 19, 2017 at midnightish

Changes in **yellow** or **green**

Overview

This assignment provides an opportunity for you to demonstrate your ability to generate dynamic web pages from database content using PHP. In this assignment you will be working with the Travel case study from the textbook.

Beginning

Create a new public workspace. Because other people can potentially find and examine your public workspace, you are going to want to make sure your workspace is private. Unfortunately you only get one private workspace so later in the semester when we work on assignment 2, you will likely need to convert this workspace to a public one.

You can begin by cloning from GitHub or cloning your completed Chapter 14 lab workspace. If you do the former, then you will need to set up MySQL again; if you do the later, you will have to remove the stuff from the lab.

If you wish to begin by Cloning from Github, then in the **Clone from Git** field, enter <https://github.com/rconnolly/comp3512-w2018-assign1.git> and select the PHP template.

If you wish to begin by clone your workspace from the Chapter 14 lab, then simply click the Clone button for that lab workspace. The advantage with that approach (assuming you've done that lab) is your database configuration will already be completed for you. You will however need to download the travel-images from GitHub and manually upload the folder to your new workspace.

We will be using the **travel-small.sql** file to populate our database. The easiest way to do so is to use the Terminal. Once you have started MySql (type **mysql-ctl cli**), enter the following commands (only if you don't have the database set up):

```
create database travel;  
source travel-small.sql;
```

After it completes, run a few test queries, for instance, by entering the following commands:

```
use travel;  
select * from Users;
```

Because MySql maps table names to file system files and because Linux-based operating systems are case sensitive, MySQL table names on cloud9 are also case sensitive. You will only have to import this data once.

Submitting

For me to mark your assignment, you will need to share your workspace with me once it is complete. To do so, click the Share button (it has a gear icon) in the upper-right part of the Cloud9 workspace. Invite me via my email (rconnolly@mtroyal.ca) and be sure to give me RW access (otherwise I can't run anything).

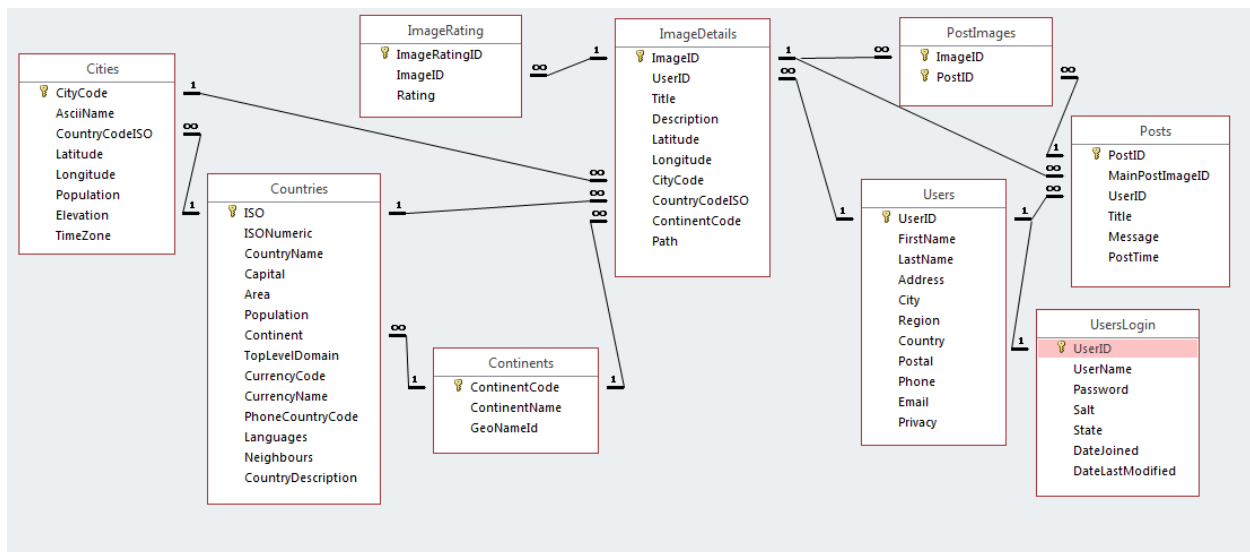
Grading

The grade for this assignment will be broken down as follows:

Visual Design and Styling	10%
Programming Design	20%
Functionality (follows requirements)	70%

Data Files

You have been provided with the necessary SQL script that you will need to import into MySQL (see instructions on first page). A Microsoft Access version of the database has been supplied as well in case you need to want to test run queries or browse the data outside of Cloud9. Images for the travel case have been provided as well. I hope the SQL file I have provided you matches this screen capture: if it doesn't, then run queries in the terminal to determine the correct names!



Requirements

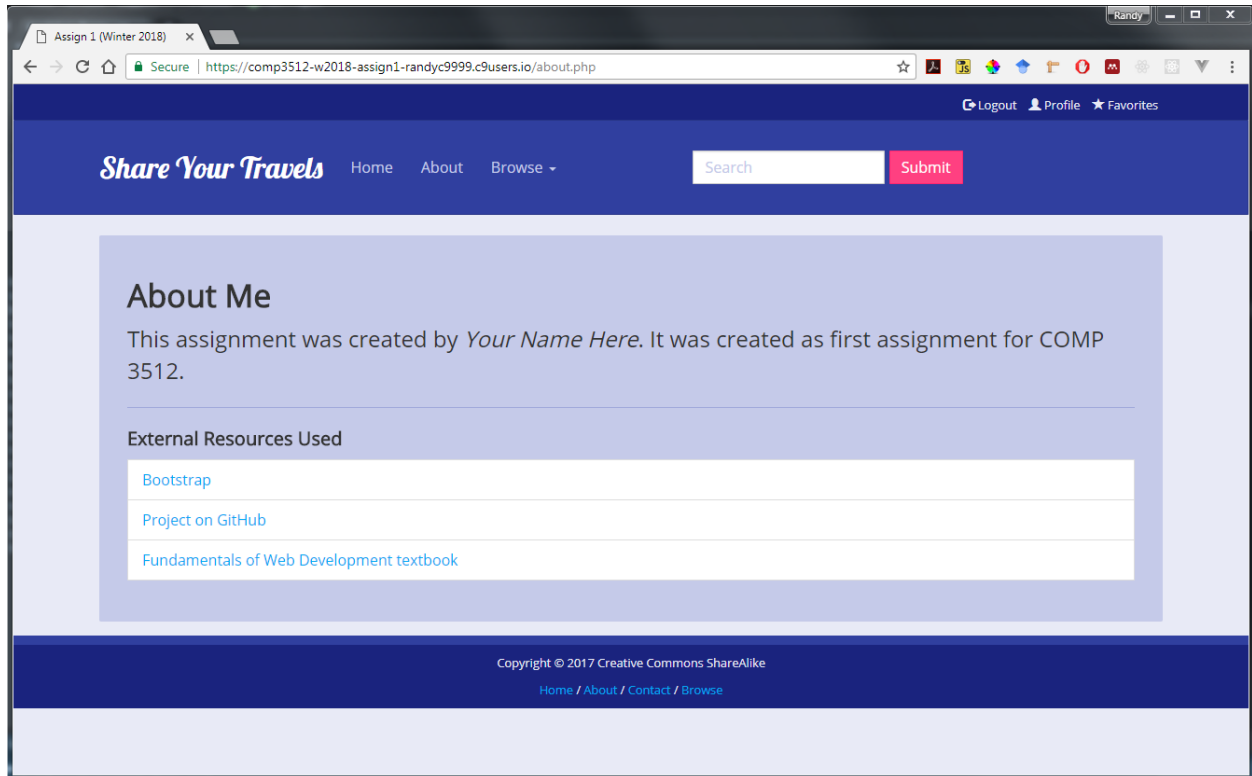
This assignment consists of 9 pages with the following functionality:

1. You must make use of the Bootstrap CSS Framework (<http://getbootstrap.com/>) to style and layout your pages. While this framework does have a little bit of a learning curve, you will find that it makes your life as a web developer significantly easier since you will be able to concentrate on programming and let Bootstrap handle the CSS.
2. You will be using the same layout and design as Chapter 11 Project 3 and Chapter 12 Project 2, so your aim in this assignment is to make sure the rest of your pages “fit” with this design. [visual design worth 5 marks]
3. The site **must** use a single CSS file for any *additional* text and color formatting. You won’t need a lot of additional formatting.
4. All pages **must** have six links: to `index.php`, `aboutus.php`, `browse-countries.php`, `browse-images.php`, and `browse-user.php` pages. Add the browse links to the Browse submenu in the header.
5. You will need to write database access code for most of your pages in this assignment. The programming design component of the grade [worth 20%] will in large part be determined by how you structure this database access. Remember that the preference is for markup that has minimal programming logic within it, and for code that doesn’t have a lot of markup in it.

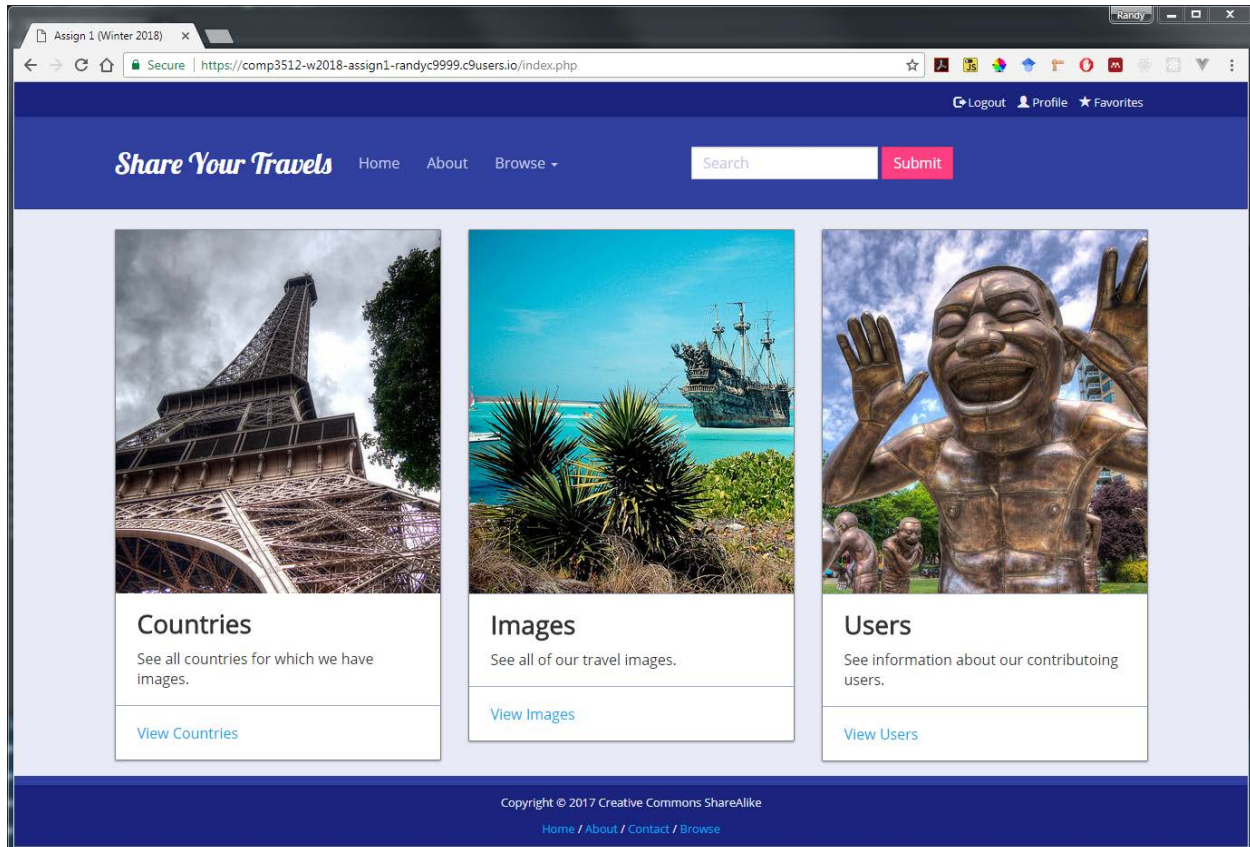
I would also strongly recommend creating the following pages in the order provided below.

- This page must be named `aboutus.php`. It should have your name. You can use the bootstrap jumbotron component. It should have your name, the course name and number, and anything else you'd like to put here. Somewhere on this page, provide a list with links of all resources you are using that you did not create (e.g. bootstrap, images, etc). Try to make it look nice and make it fit with rest of site.

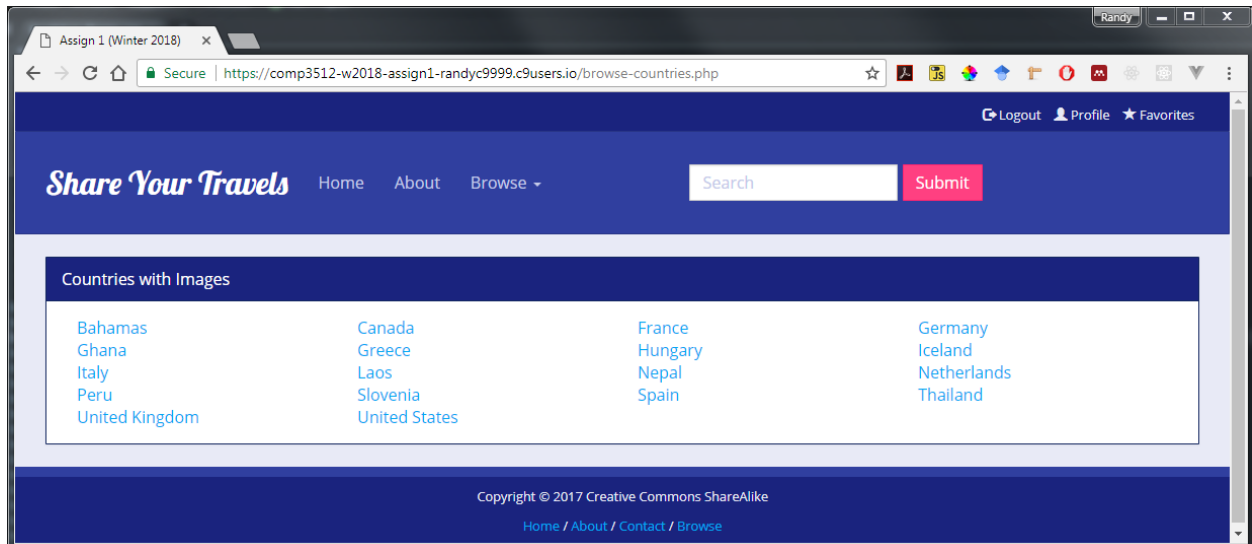
Note: I do not expect that your assignment pages will match these screen captures exactly. They should be close, however, and implement the same functionality and content.



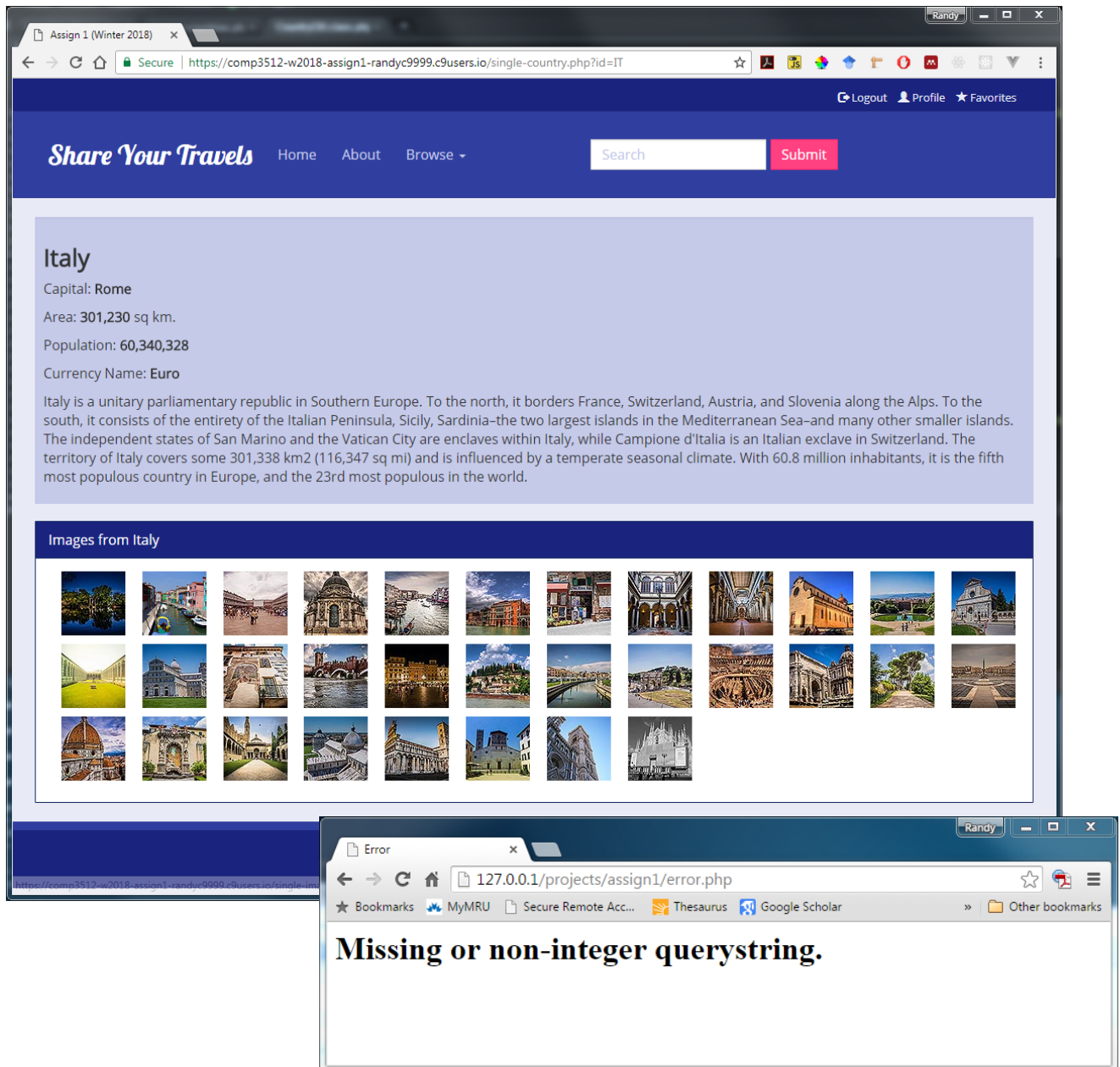
7. This page must be named `index.php` and must be the destination for the Home link in your site. The text and links for the images on the home page can be hard-coded. This page will contain the following links: Browse Countries, Browse Images, and Browse Users. You will have to supply your own custom CSS styling to create the “card” styling shown here (I have supplied these images in the images/misc folder. You could also create this page last.



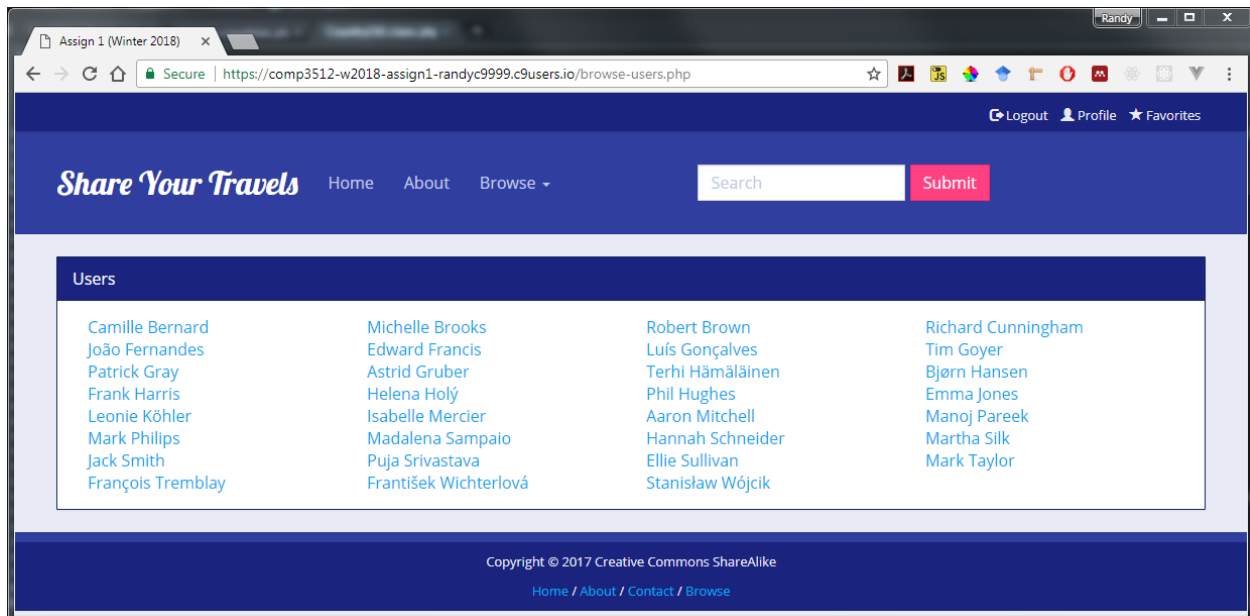
8. This page must be named `browse-countries.php`. The countries in the list should be links to `single-country.php` with the appropriate querystring. Notice that they are sorted by name. The Countries table contains a record for every country in the world. This page should only display those countries for which there is an image. This will require using a GROUP BY sql query.



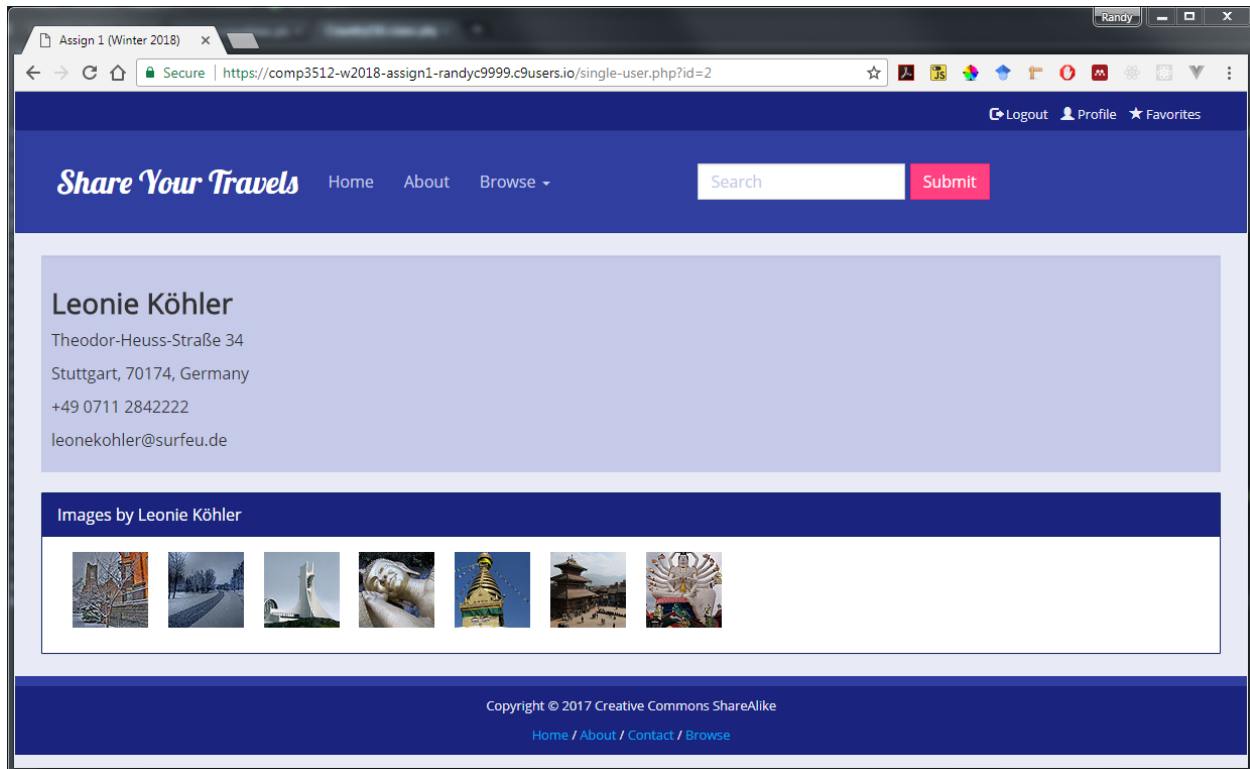
9. This page must be named `single-country.php`. It should display information for the specified country (specified via a query string parameter) from the `Countries` table. The images panel should show the images for that country (obtained from the `ImageDetails` table). Each image should be a link to `single-image.php` with the appropriate querystring (the same will be true in any other page that contains `square-small` images). You should generalize this functionality as you will need to display lists of images for `single-user.php` as well. This page should handle a missing or empty query string parameter by redirecting to a page called `error.php` (don't worry about styling this error page though it would be nice to make it match the styling of the rest of the site).



10. This page must be named `browse-users.php`. The users in the list should be links to `single-user.php` with the appropriate querystring. You may need to use the PHP `utf8_encode()` function to properly display some of the foreign characters in the names (you can achieve the same thing via your connection string). Notice the sort order!

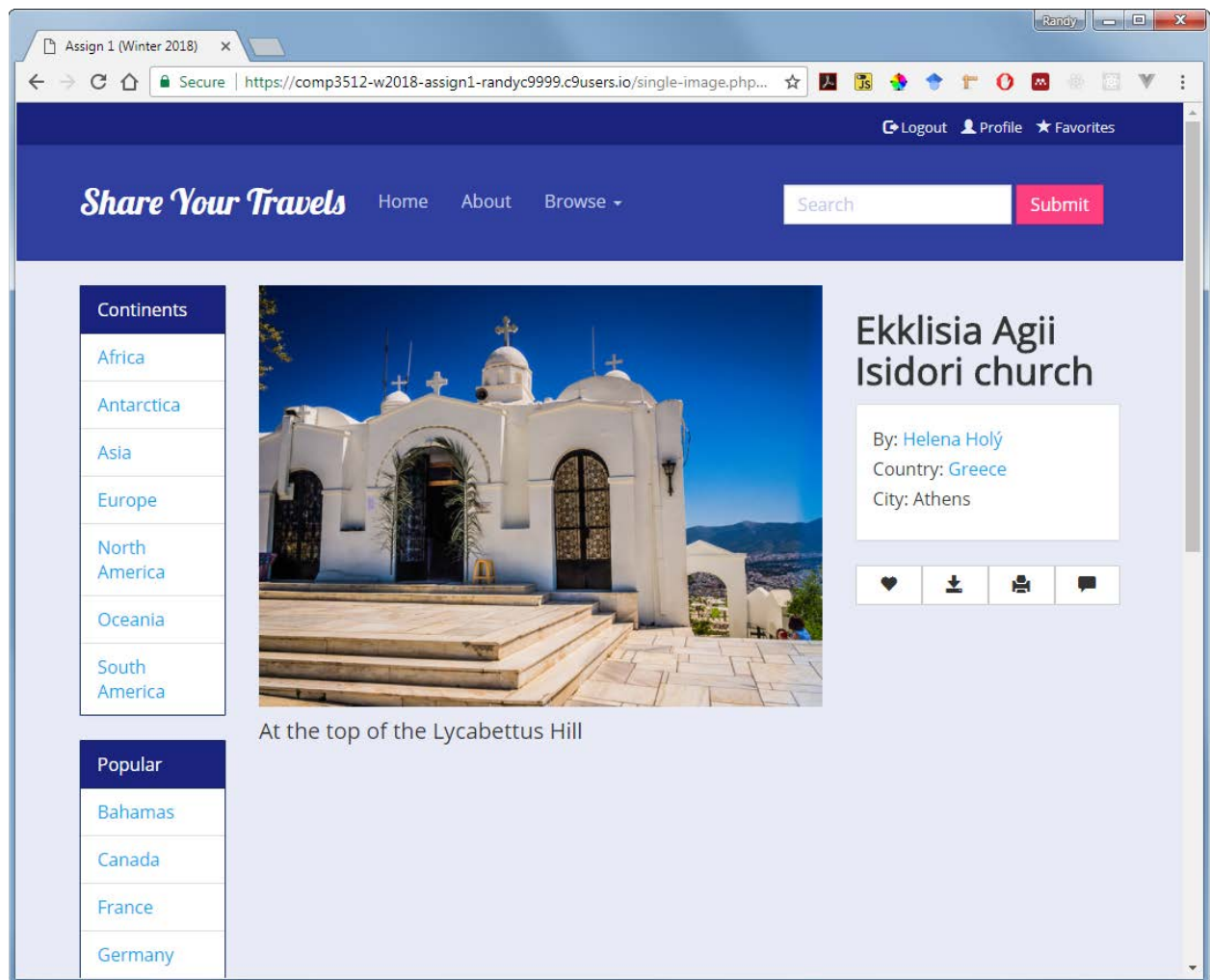


11. This page must be named `single-user.php`. It must display the information about a single user (passed in via a query string parameter). Be sure to use the PHP `utf8_encode()` function to properly display some of the foreign characters in the names and address. The images panel should show the images for that user and have same functionality as the images panel in `single-country.php`. This page should handle a missing or a non-integer query string parameter by redirecting to a page called `error.php`



12. This page must be named `single-image.php`. It must display the information about a single travel image (specified via the `id` passed in via a query string parameter). Its design and functionality is based on Chapter 12, Project 2. The user name should be a link to `single-user.php` with the appropriate querystring. The country name should be a link to `single-country.php` with the appropriate querystring. Be sure to use the PHP `utf8_encode()` function to properly display some of the foreign characters in the data. This page should handle a missing or invalid query string `id` parameter by redirecting to a page called `error.php`.

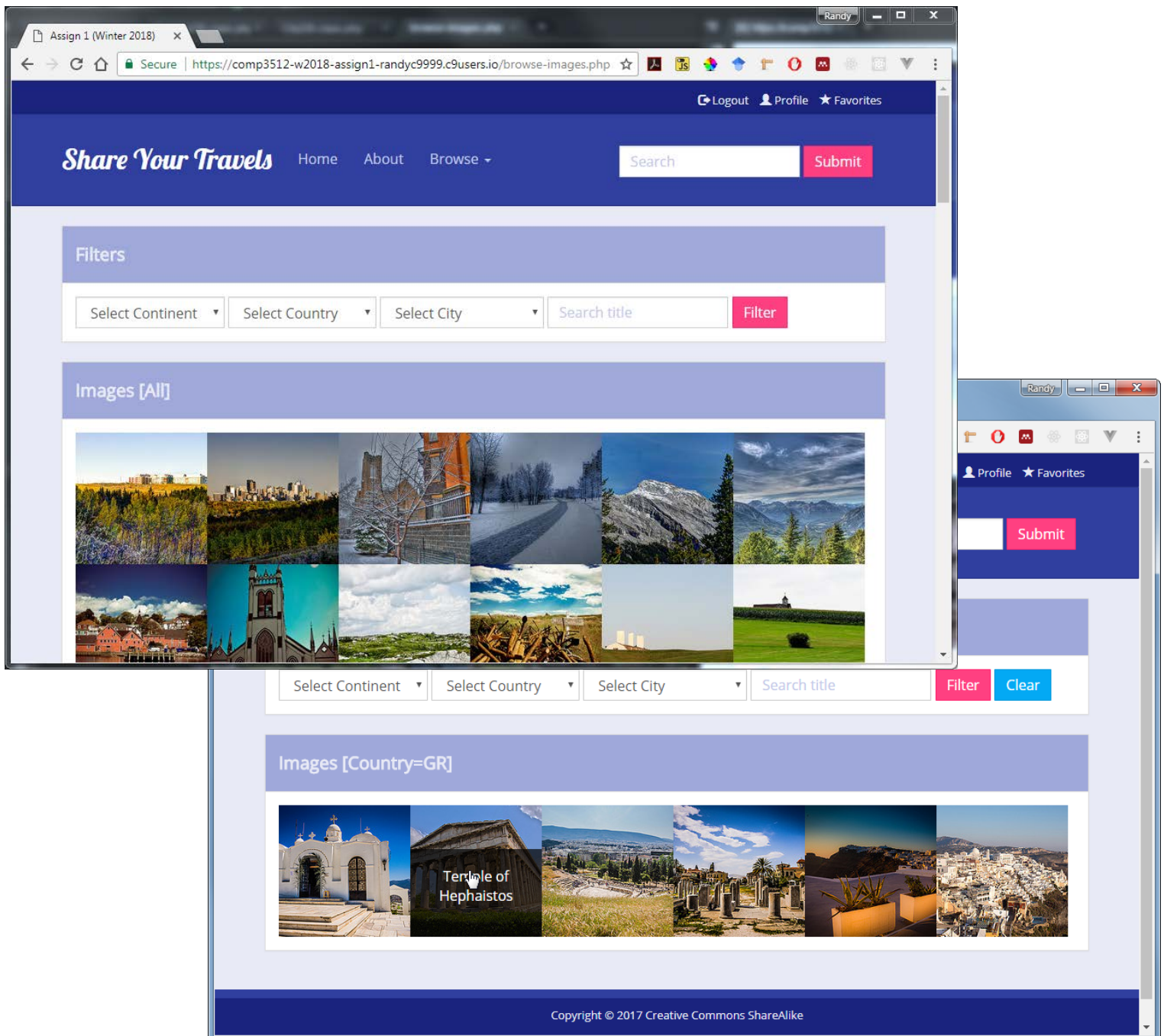
The Continents and Popular Countries (i.e., just those countries that have images) list on the right should be pulled from the Continents and Countries table. Each should be an appropriate link with query string data to `browse-images.php`.



13. This page must be named `browse-images.php`. This page displays multiple travel images. Each of the images shown must be links with the appropriate query string to the `single-image.php` page. You already have this layout in Chapter 14 Project 2.

Initially, this page must display all the images. The user should be able to filter the list by specifying the continent, country, or city in the three drop-down lists (populated from the Continent, Countries and Cities tables). The user should also be able to filter by entering a search string into the "Search Title" text box. Be sure to only show countries and cities with images (hint use SQL group by). To simplify your programming, assume that the user will select only one of these filters (i.e., continent or country or city or title) not multiple ones.

When the user clicks the Filter button, display the matching images and the current filter. The Clear button should reset the drop-down lists and display all the images.



Preliminary Marking Guide COMP 3512 Assignment #1

/ 10	Visual Design and Styling <ul style="list-style-type: none"> - Text and Color Formatting + CSS usage + consistency - Used Bootstrap appropriately - All pages follow same visual design
/ 20	Programming Design and Styling <ul style="list-style-type: none"> - reduces markup and code duplication - minimizes code in the markup - minimizes markup in the code
/ 10	Navigation and Home and About pages <ul style="list-style-type: none"> - correct links + appropriate content
/ 5	browse-countries.php <ul style="list-style-type: none"> - appropriate links
/ 10	single-country.php <ul style="list-style-type: none"> - correct data - handle querystring errors - list of travel images for that country
/ 5	browse-users.php <ul style="list-style-type: none"> - appropriate links
/10	single-user.php <ul style="list-style-type: none"> - correct data - handle querystring errors - list of travel images for that user
/ 27	browse-images.php <ul style="list-style-type: none"> - show images with appropriate links - filtering works properly
/ 13	single-image.php <ul style="list-style-type: none"> - correct data, hyperlink with proper querystring id - handle querystring errors
<hr/>	
/ 100	Total Mark

Note: this is a preliminary marking guide. I tend to revise the marking guide as I mark the first few assignments.