COMP 3512
Assignment #1: Data-Driven PHP *(version 1)*

Due Tuesday October 20, 2015 at midnightish

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 Photo case study from the textbook. You will be building upon the visual design used in Chapter 8 Project 3, Chapter 9 Project 2, and Chapter 11 Project 2.

Submitting

Put your assignment in a folder named assign1_*yourlogin* (e.g., assign1_fsmi9876). Put all resources used by your assignment into this folder. Copy this folder to the submit drive in B215 **and** to the normal university-wide submit drive. You will lose marks if you do not follow these submission instructions.

Grading

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

Visual Design and Styling	5%
Programming Design	15%
Functionality (follows requirements)	80%

Data Files

You have been provided with the necessary SQL script that you will need to import into MySQL (easiest to use PHPMyAdmin). For additional information about the database schema, see Figure 11.25 in Chapter 11. You might also want to examine more carefully the code samples in section 11.7 of Chapter 11.

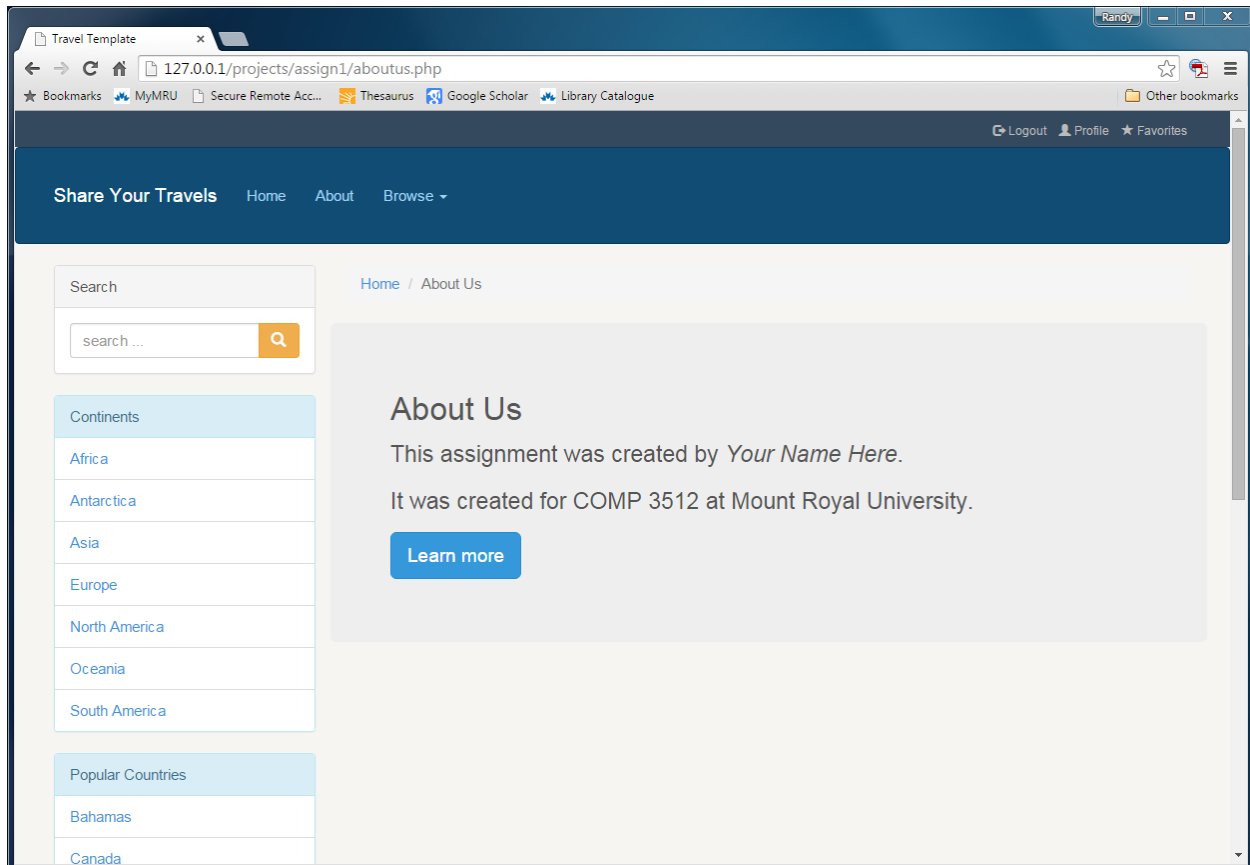Various image files have also been provided for you.

Requirements

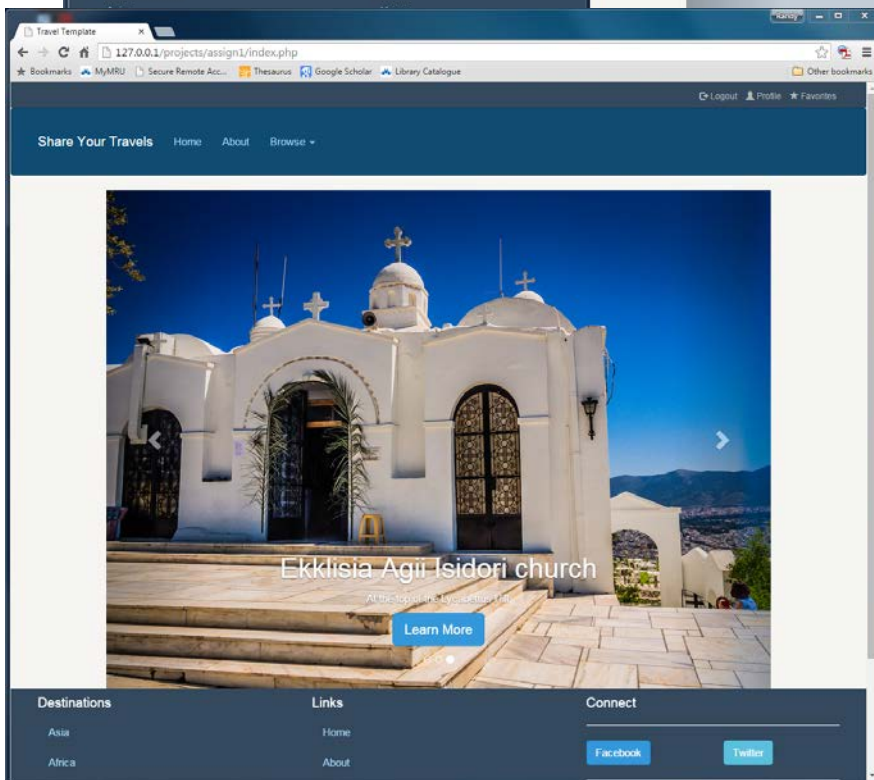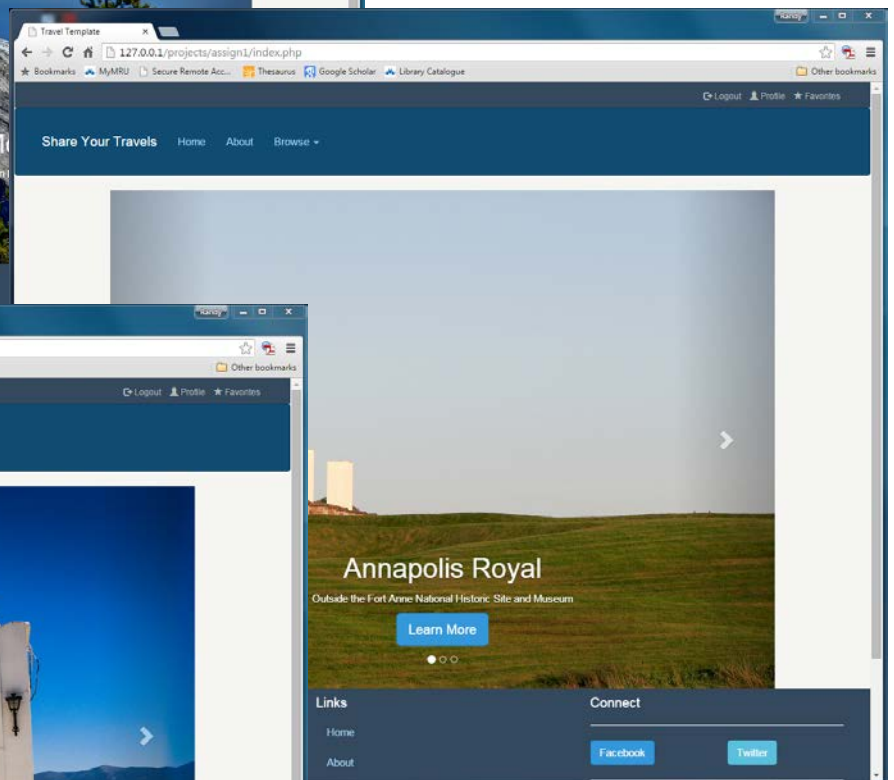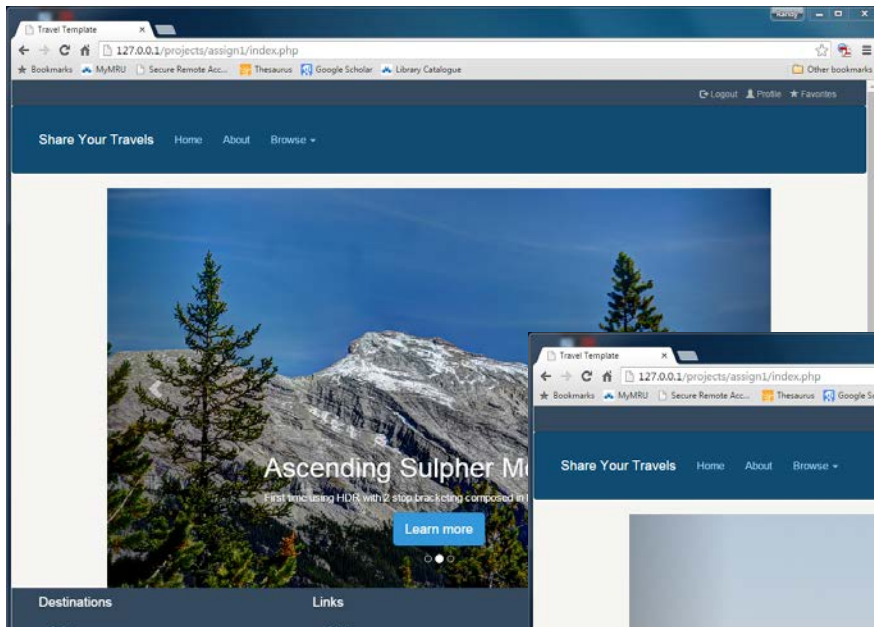This assignment consists of 12 pages with the following functionality:

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. Look at the examples on the Bootstrap web site to help you understand how the make use of it. You will be using Bootstrap on the other assignments this semester so you might as well start learning it now.

2. You will be using the same layout and design as Chapter 8 Project 3 and Chapter 9 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.

4. All pages **must** have six links: to `index.php`, `aboutus.php`, `browse.php,` `browse-countries.php,` `browse-images.php,` `browse-posts`, and `browse-user.php` pages. Add these links to the Browse submenu in the header and to the relevant breadcrumb links. Notice as well the link expectations for the header and left menu area; these must be consistent on all pages.

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 15%] 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.

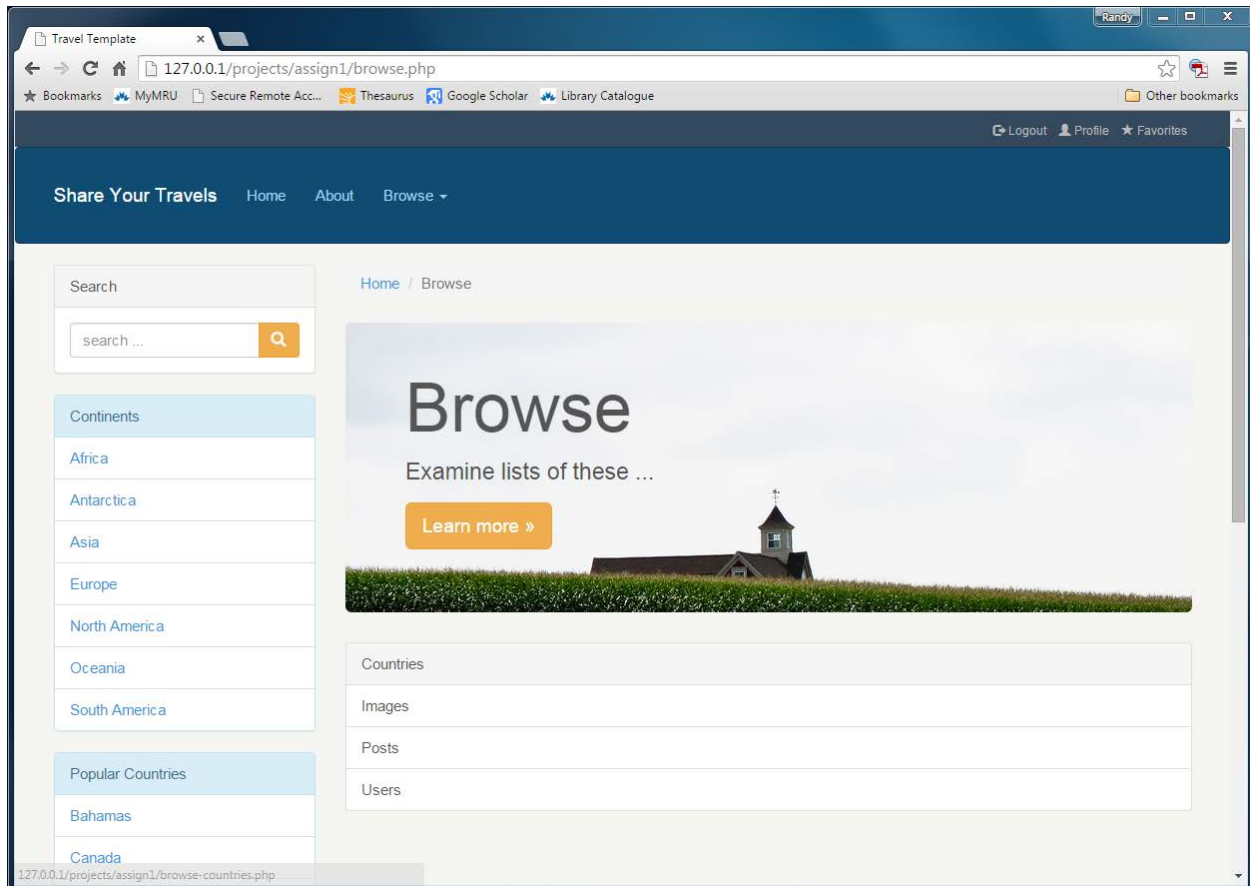   I would also strongly recommend creating the following pages in the order provided below.

6. This page must be named `aboutus.php`. It should have your name. You can use the bootstrap jumbotron component. The "Learn more" link can just be a dummy link. For information about the continents and popular countries lists, see Project 2 write up in Chapter 11. Notice the sort order of the continents and countries lists. The links for the continents can be dummy links. The countries should be links to `single-country.php` with the appropriate query string. The breadcrumb Home link should be to `index.php`. See number 4 above for information about the menu links.
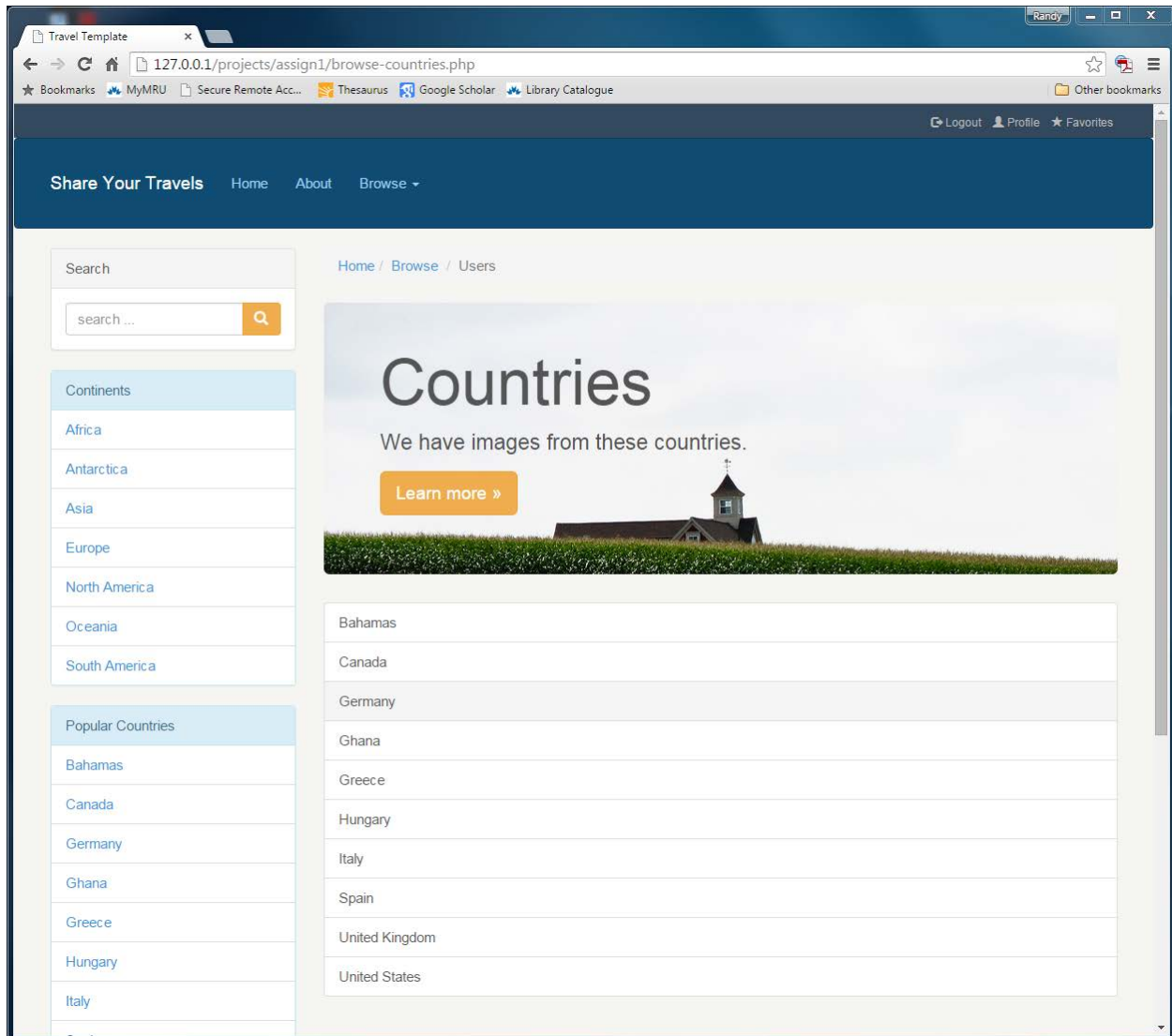
7. This page must be named `index.php` and must be the destination for any Home links in your site. It should use a Bootstrap carousel component (in Javascript section of documentation … no Javascript programming is actually needed). Each slide of the carousal should show a travel image and its title and description. You can use whatever image you'd like for the carousel slides. The text and links for the slides can be hard-coded. The "Learn more" button should be a link to `single-image.php` with the appropriate querystring. This page can also be completed last.
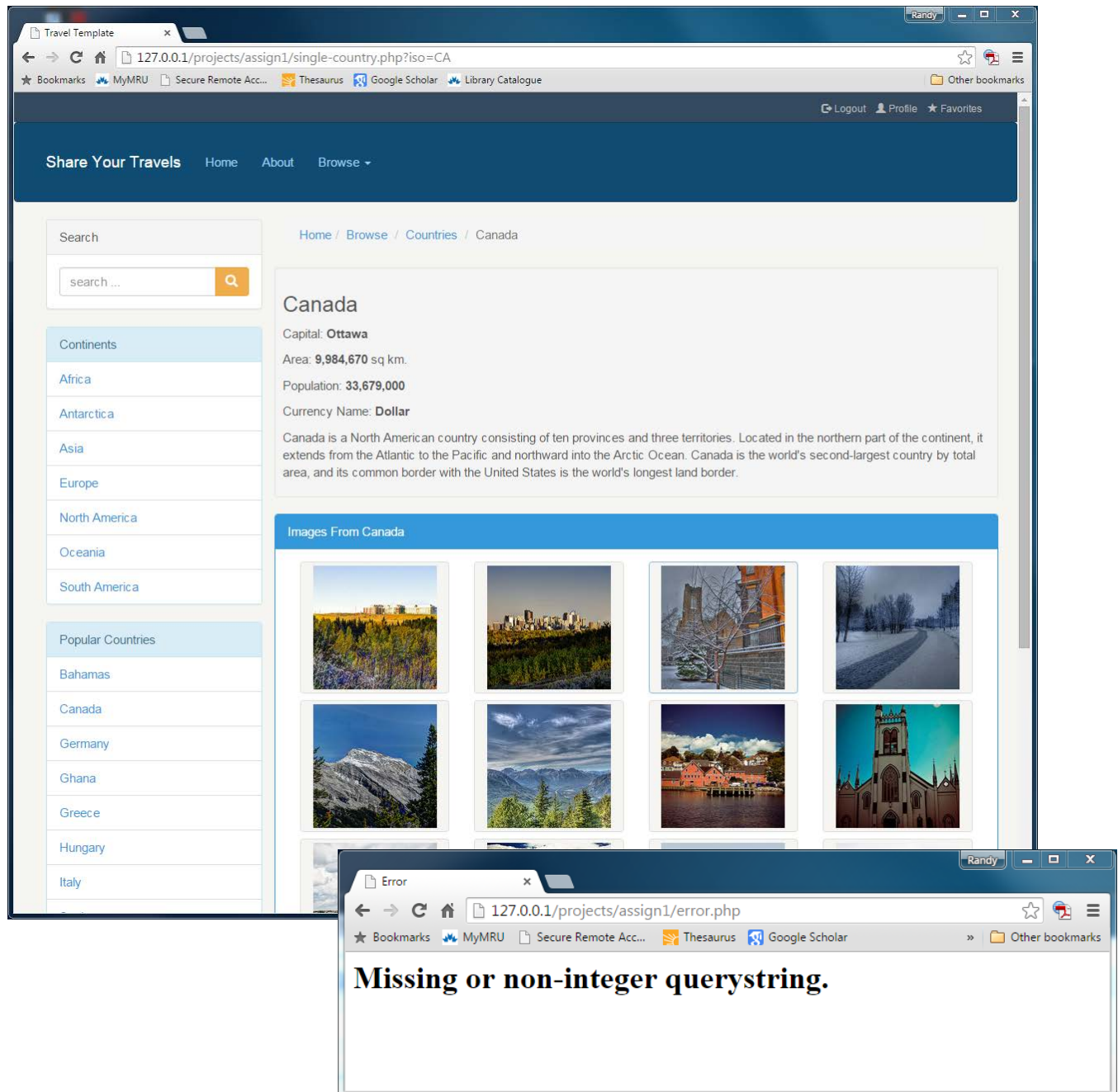
8. This page must be named `browse.php`. It must display links to `browse_countries.php`, `browse_images.php`, `browse_posts.php`, and `browse_users.php`. You can use the bootstrap list group component to simply create your list of links. The "Learn more" link can just be a dummy link. The jumbotron layout here is the same as that used in Project 3 in Chapter 8.
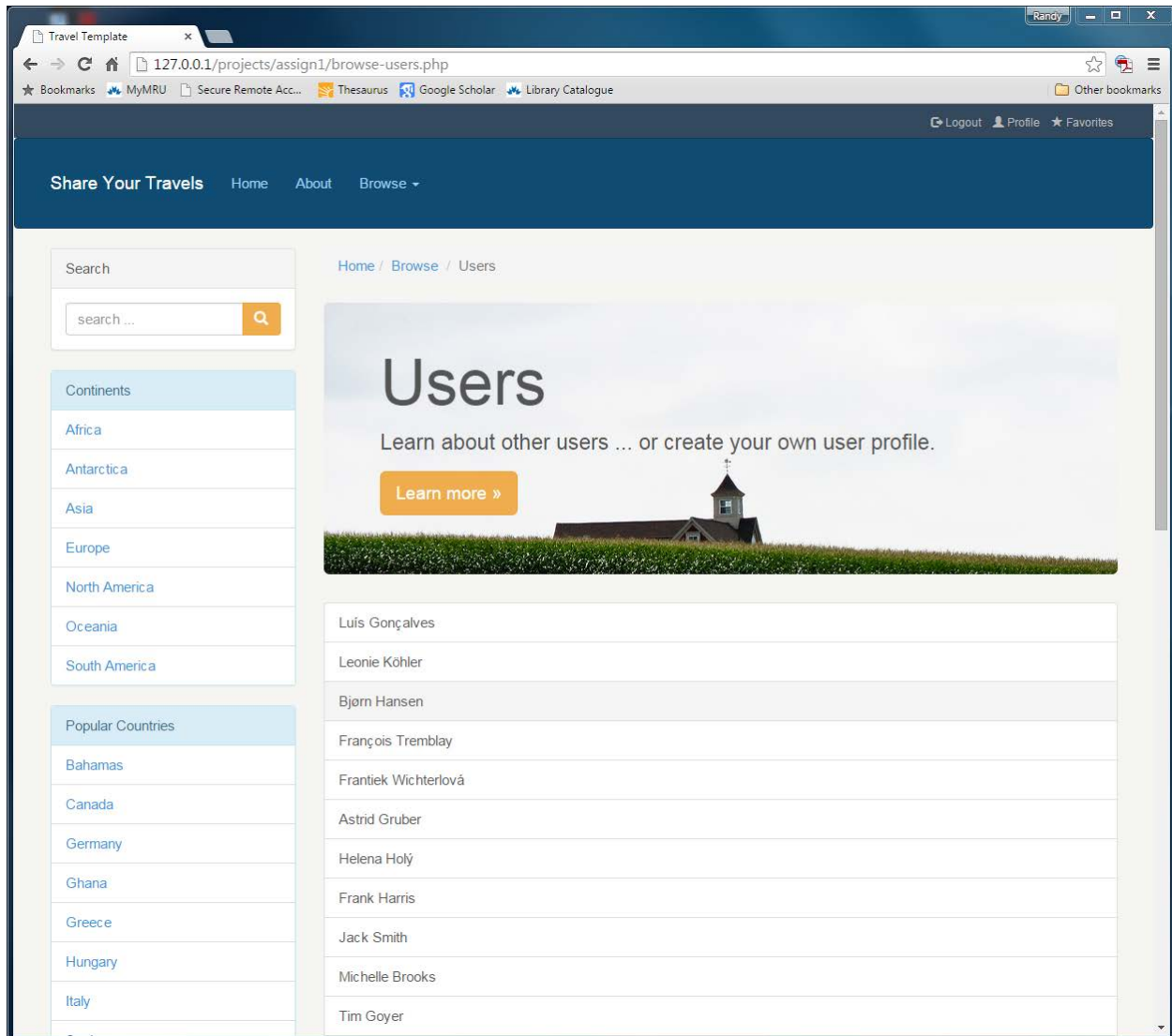
9. This page must be named `browse-countries.php`. You can use the bootstrap list group component to simply create your list of links. The countries in the list should be links to `single-country.php` with the appropriate querystring. The Browse breadcrumb link should be to `browse.php`.
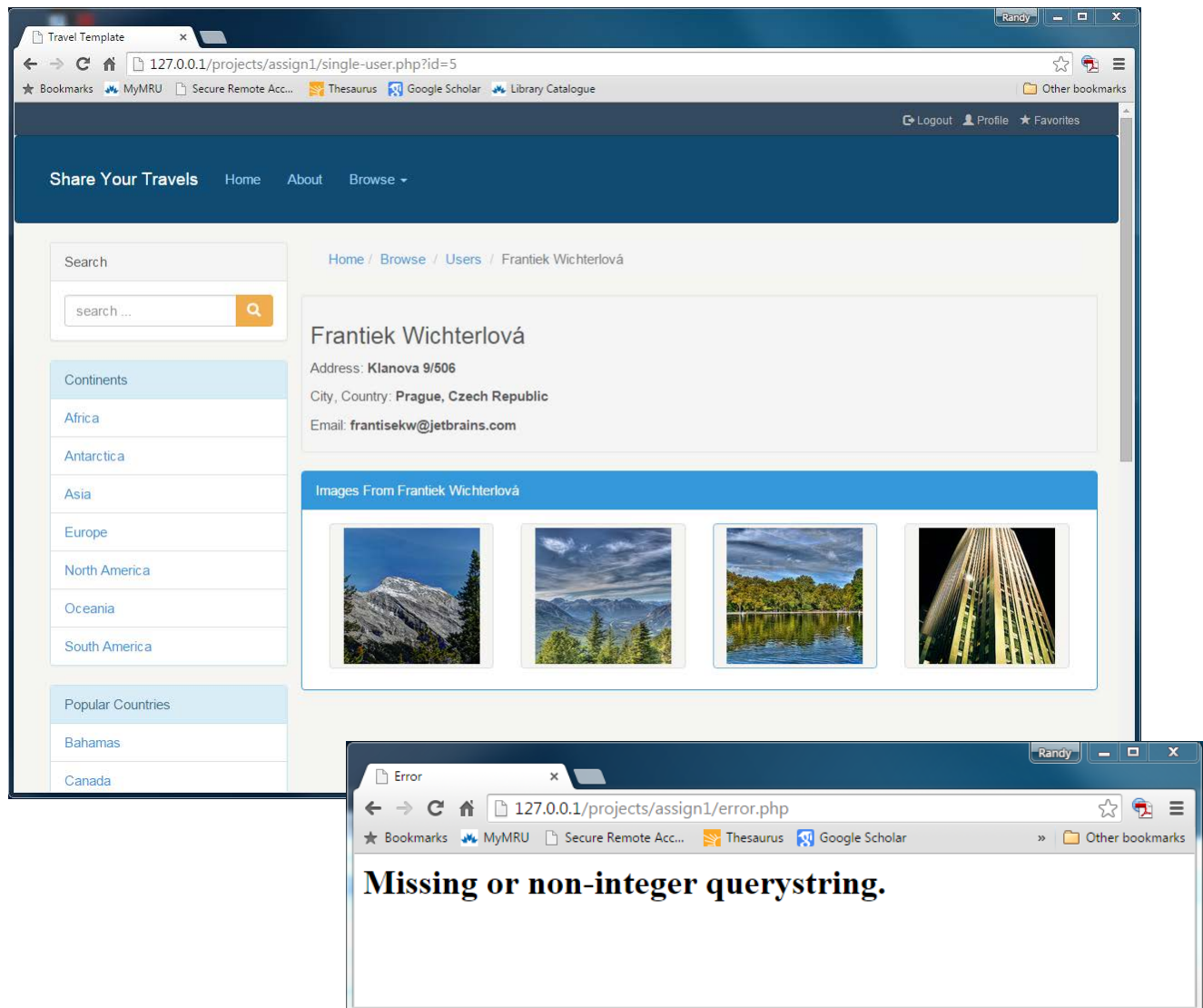
10. This page must be named `single-country.php`. It should display information for the specified country (specified via a query string parameter) from the `GeoCountry` table. The images panel should show the images for that country (obtained from the `TravelImage` and `TravelImageDetails` tables). Each 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-medium` images). You should generalize this functionality as you will need to display lists of images for other pages as well. The Countries breadcrumb link should be to `browse-countries.php`. This page should handle a missing query string parameter by redirecting to a page called `error.php` (don't worry about styling this error page).
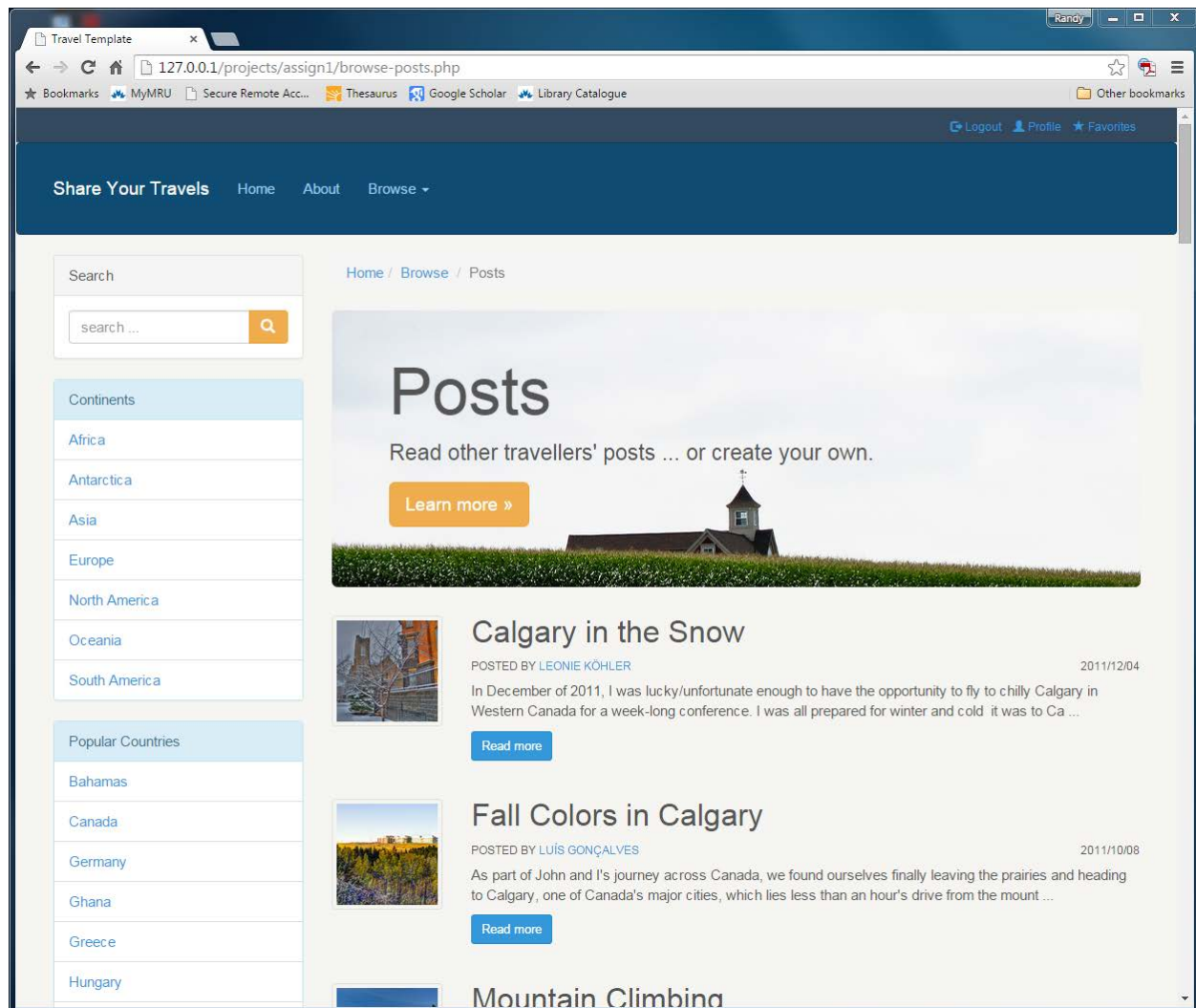
11. This page must be named `browse-users.php`. You can use the bootstrap list group component to simply create your list of links. The countries in the list should be links to `single-user.php` with the appropriate querystring. You will need to use the PHP `utf8_encode()` function to properly display some of the foreign characters in the names.
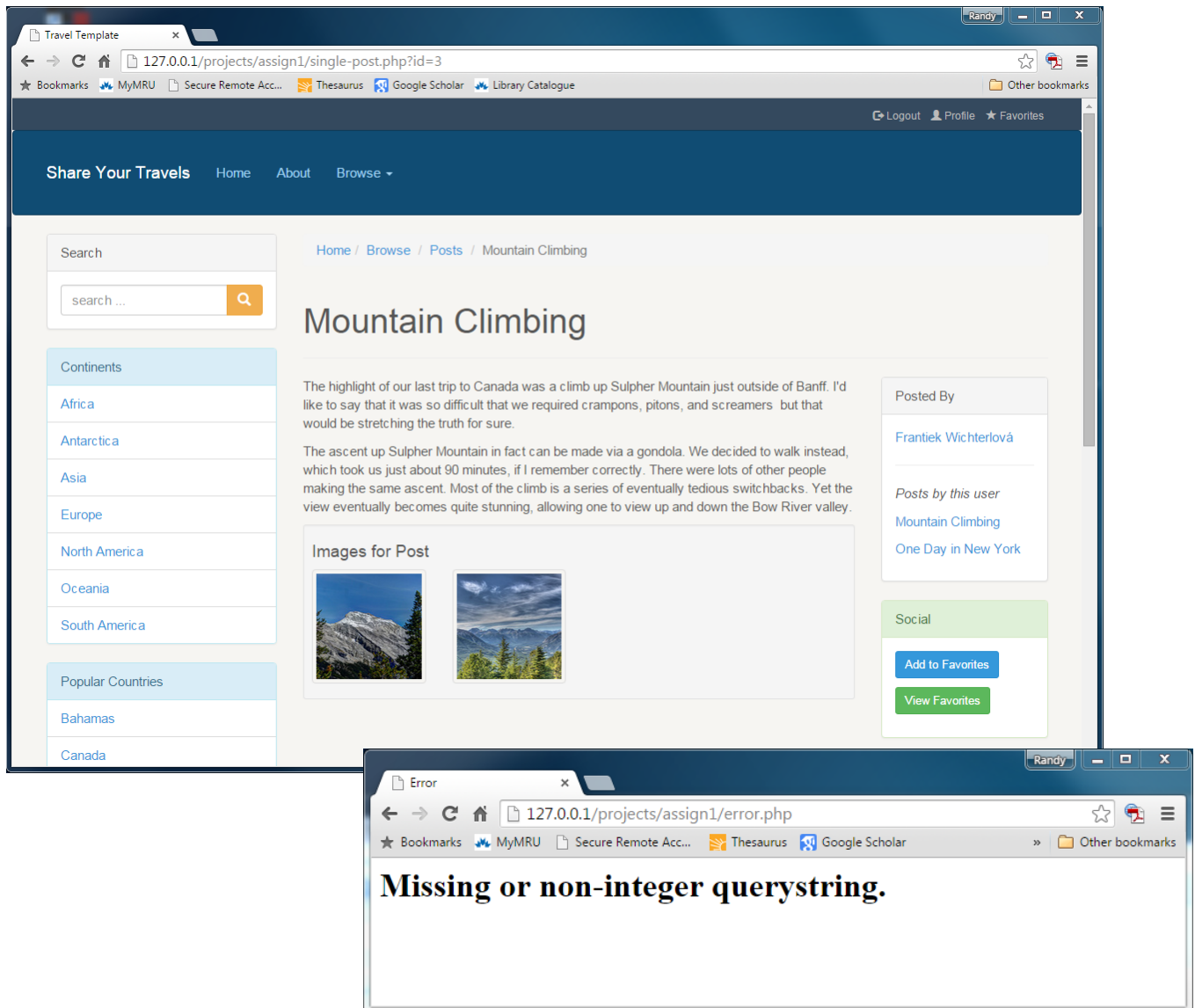
12. This page must be named `single-user.php`. It must display the information about a single user (passed in via a query string parameter). Be sure to use the PHP `utf8_encode()` function to properly display some of the foreign characters in the names and address. The images panel should show the images for that user. The Users breadcrumb link should be to `browse-users.php`. This page should handle a missing or a non-integer query string parameter by redirecting to a page called `error.php`

13. This page must be named `browse-posts.php`. You already have the layout for this page in Chapter 8, Project 3. It must display the MainPostImage for each post, as well as the title, the user name, the post date, and an excerpt (i.e., the first 200 characters) of the post message. The read more link will be for `single-post.php` with the appropriate querystring. Be sure to use the PHP `utf8_encode()` function to properly display some of the foreign characters in the user names and excerpt.
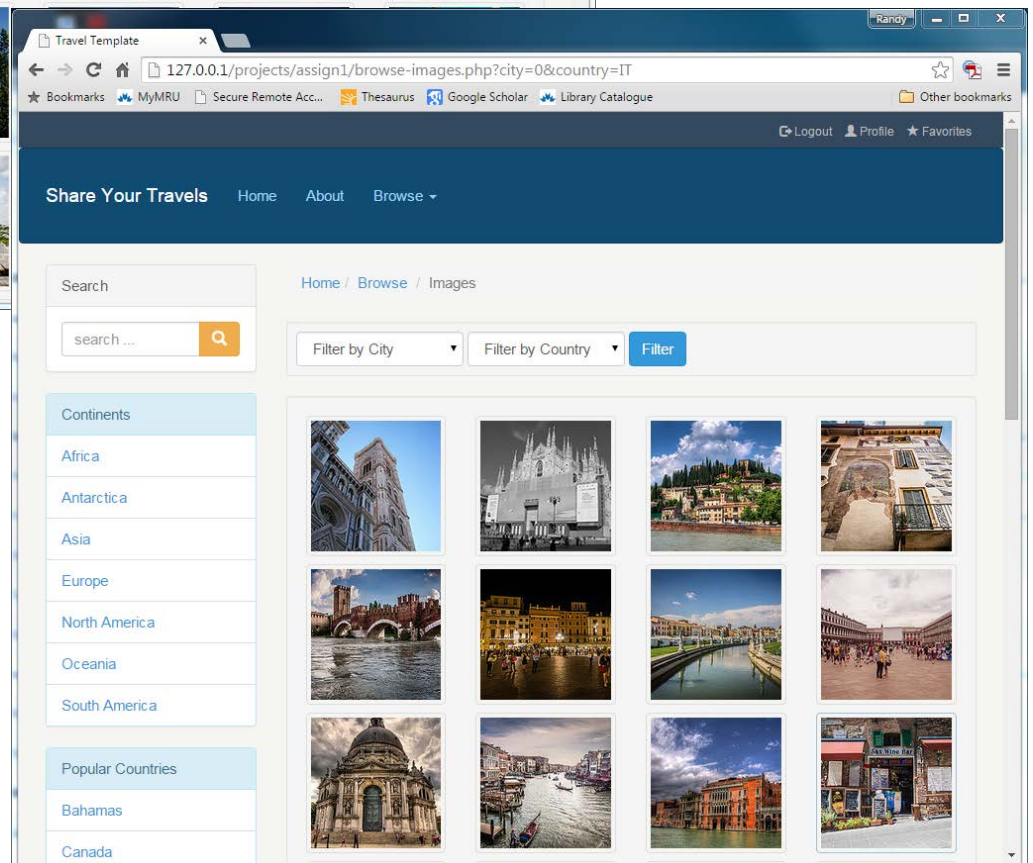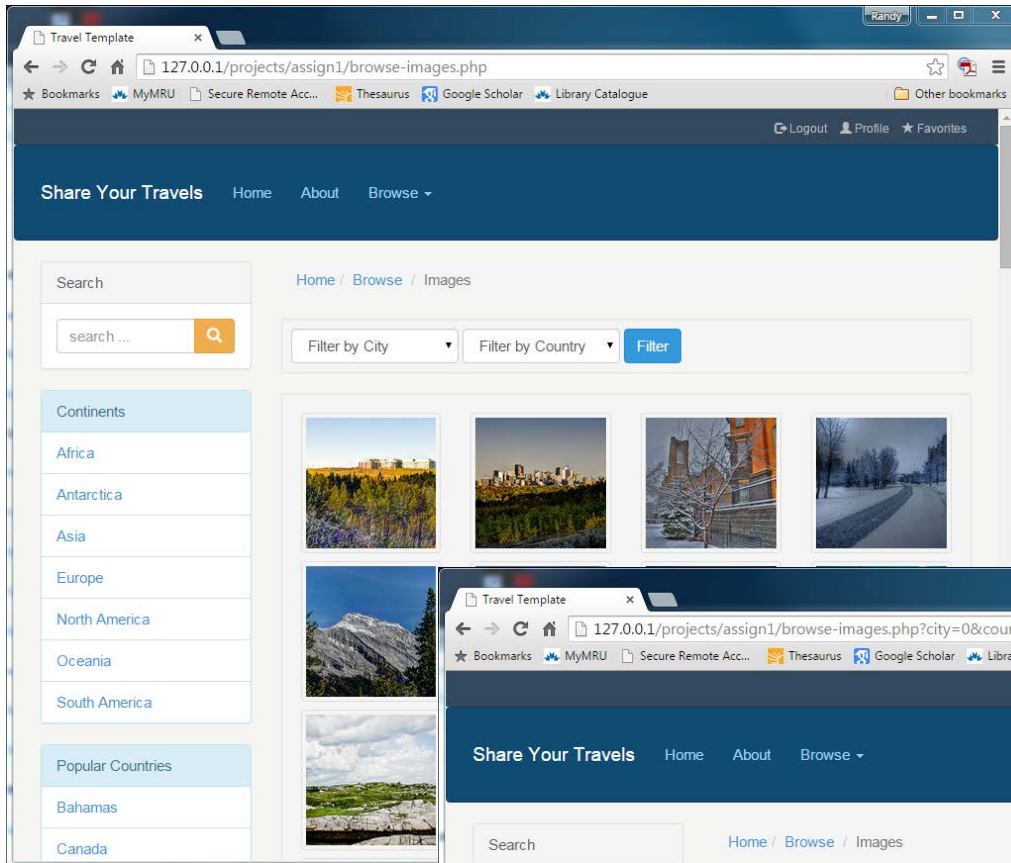
14. This page must be named `single-post.php`. It must display the information about a single travel post (specified via the `id` passed in via a query string parameter). It must display the post title and message, as well as the images for the post. It must also display the user name for the post (which should be a link to the `single-user.php` page with the appropriate querystring). It should also display a list of the other posts by the same user; each post name should be a link to `single-post.php` with the appropriate querystring). Be sure to use the PHP `utf8_encode()` function to properly display some of the foreign characters in the user name. The Posts breadcrumb link should be to `browse-posts.php`. This page should handle a missing or a non-integer query string parameter by redirecting to a page called `error.php`
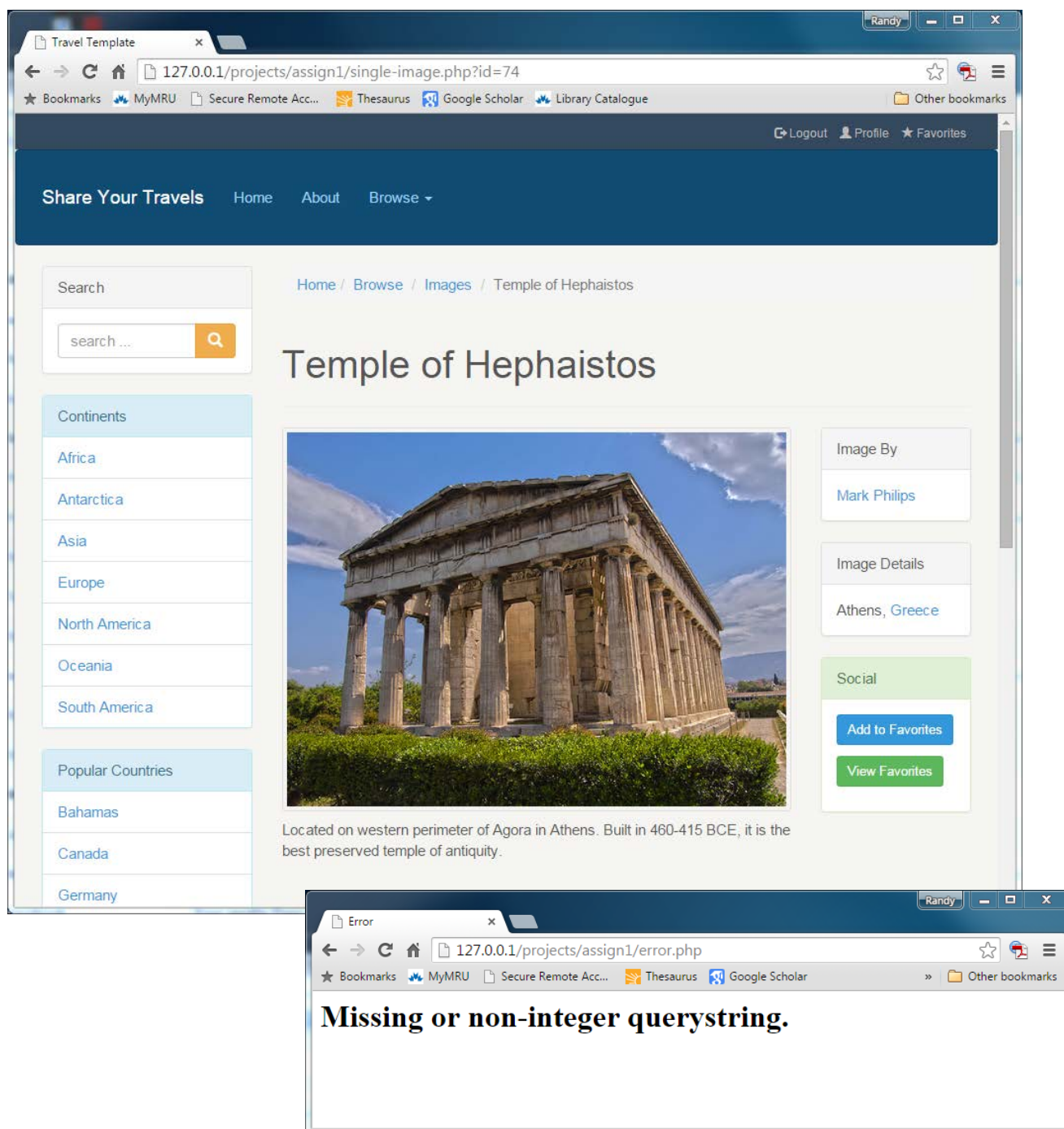
15. This page must be named `browse-images.php`. This page displays multiple travel images. Each of the images shown must be links with the appropriate query string to the `travel-image.php` page. You already have this layout in Chapter 11 Project 2.

Initially, this page must display all the travel images in the table. The user should be able to filter the list by specifying the country or city in the two drop-down lists (populated from the GeoCountries and GeoCities tables). 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 either a city or a country but not both.

16. 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 9, 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. The Images breadcrumb link should be to `browse-images.php`. Be sure to use the PHP `utf8_encode()` function to properly display some of the foreign characters in the user name.

The image must be a link to the large version. Use the Bootstrap Modal dialog (in Javascript section of documentation) to display the larger version. This page should handle a missing or invalid query string `id` parameter by redirecting to a page called `error.php`.

Preliminary Marking Guide COMP 3512 Assignment #1

/ 5 **Visual Design and Styling**
- Text and Color Formatting + CSS usage + consistency
- Used Bootstrap appropriately
- All pages follow same visual design

/ 15 **Programming Design and Styling**
- reduces markup and code duplication
- minimizes code in the markup

/ 10 **Navigation and Home and About pages**
- correct links + appropriate content

/ 3 **browse.php**
- appropriate links

/ 5 **browse-countries.php**
- appropriate links

/ 8 **single-country.php**
- correct data
- handle querystring errors
- list of travel images for that country

/ 5 **browse-users.php**
- appropriate links

/ 8 **single-user.php**
- correct data
- handle querystring errors
- list of travel images for that user

/ 8 **browse-posts.php**
- appropriate links and data (excerpt, main image, date)
- utf8 encoding

/ 8 **single-post.php**
- correct data (user name link, other posts)
- handle querystring errors
- list of travel images for that post

/ 17 **browse-images.php**
- show images with appropriate links
- filtering works properly

/ 8 **single-image.php**
- correct data, hyperlink with proper querystring id
- uses modal dialog
- handle querystring errors

_____

/ 100 Total Mark