COMP 3512 Assignment #2

Due Midnight-ish Saturday Nov 18
Version 1.0, Oct 12 2017 [Most recent updated text will be in yellow]

Overview

You can work in groups of three or four for this assignment. It is also possible to work individually, but I do **strongly** discourage it; please talk to me about this if you are planning on working by yourself. This assignment will allow you to apply the concepts and technologies covered in class to a "real" project situation. In this project, you will expand your first assignment.

Groups of four will have additional cases to complete. If working in a group, each member needs to take responsibility for and complete an appropriate amount of the project work. **Be sure to consult the instructor at least one week prior to the due date if your group is experiencing serious problems in this regard.**

I feel foolish saying this in a third-year university course, but it is your responsibility to read all the assignment instructions thoroughly and carefully. If you are unclear about something, ask me. But before you do, read the material in question again!

Beginning

Create a new workspace named *yourlogin*-comp3512-assign2 that doesn't specify a team. To begin, I would recommend you clone one of your group member's first assignment, since that will mean the database is all setup and ready to go. Each group member will have to be given RW access to the workspace via the Share button. Cloud9 allows multiple people to simultaneously edit the same file, so you will have to be careful about overwrites if you have two people editing the same block of code.

Two new tables have been provided that you will have to add to your books database. Once you have started MySql in the terminal (type **mysql-ctl cli**), enter the following commands:

use book;
source users.sql;

GitHub

You will also need to create a private repo on Github for the assignment. Please name the repo the same as your workspace (i.e., *yourlogin*-comp3512-assign2). One member will have to create it, and then add other members as collaborators. Needless to say, this means each group member will also need their own Github account. You will want to push out updates fairly frequently (1-2 times a day when working on it, or more frequently if lots of work is being done). I will examining the commits when marking. You can push your content from Cloud9 to GitHub via the terminal, using the following commands:

git init *[only need to do this one command once for your assignment]*
git add *
git commit –m "Fixed the thingamajib (by randy)" *[alter message and name as appropriate]*
git remote add origin https:... *yourlogin*-comp3512-assign2.git *[specify URL of github repo]*
git push –u origin master *[login using your own individual github credentials]*

For more information about Git and GitHub, read pages 571-577 of textbook (2nd Edition).

Submitting

You will need to share your workspace with me once it is complete. To do so, click the Share button 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. As well, you will need to add me as a collaborator to your GitHub repo when you have completed it.

Grading

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

Usability and visual design	20%
Program design **and** documentation	15%
Features	65%

USE CASE NAME:	**Assignment 1 Functionality**
DESCRIPTION:	Expected base functionality
1.	Your system must have ALL the functionality from assignment 1.

USE CASE NAME:	**Program Design**
DESCRIPTION:	Class Approach
1.	Make use of a class-based infrastructure for this assignment. Chapter 14/17 provides code examples for implementing this infrastructure.

USE CASE NAME:	**Navigation**
DESCRIPTION:	The web site's navigation system.
1.	In the left-side navigation, add a new icon+link named User Profile after the Dashboard link. This link will take the user to the profile.php page (see User Profile use case below), which is a use case for groups of four only.
2.	To your existing top row, add a Logout icon link (see Logout case below).
3.	The Analytics link will go to analytics.php (see Analytics use case below), which is a use case for groups of four only.

USE CASE NAME:	**About Us**
DESCRIPTION:	Display information about this site
1.	The system will display information about this site. Be sure to mention that this site is hypothetical and was created as an assignment for COMP 3512 at Mount Royal University taught by Randy Connolly. Add a link to your GitHub page for the assignment. As well, be sure to list the group member names and roughly what parts of the project each member implemented. If you used any external resources, be sure to cite (or link to) them here.

USE CASE NAME:	**Browse Universities**
DESCRIPTION:	System allows user to browse universities. **This case is only for groups of four**. Groups of three will have to complete this however in assignment 3.
1.	Extend the Browse Universities page from the first assignment by displaying a Google Map (https://developers.google.com/maps/documentation/javascript/) instead of displaying the latitude and longitude values. Display a marker of the university's location on the map. This will require a little bit of JavaScript. As well, your PHP will page will have to generate a few lines of JavaScript to insert the correct latitude and longitude from the database table into the JavaScript. Be sure to set the map zoom sufficiently large so that the maps shows only a few kilometers around the location.

USE CASE NAME:	**Browse Books**
DESCRIPTION:	System allows user to browse books
1.	Extend the Browse Books page from the first assignment by turning each university name in the Adoptions card into a link+querystring to the browse university page (i.e., clicking on the university name will take user to the browse university page and display the correct university in the details section of the page). Your browse university page should already have the functionality to handle such links from the first assignment.

USE CASE NAME:	**Single Book**
DESCRIPTION:	System allows user to view information about a single book
1.	Extend the Single Book page from the first assignment by turning each university name in the Adoptions card into a link+querystring to the browse university page (i.e., clicking on the university name will take user to the browse university page and display the correct university in the details section of the page.
2	Do something similar with the Imprint and Subcategory values. They should be links+querystring to the browse book page (e.g., clicking on the imprint name will take the user to the browse books page and display just the books with the specified imprint). Your browse books page should already have the functionality to handle such links from the first assignment.
3	Clicking on the book cover image should display a larger version of the book cover floating over the rest of the page. The rest of the page should dim/darken until the user clicks again on the large image; the click should also make the large image disappear. This will require some JavaScript and some CSS. Do **not** use some complicated LightBox plugin/code you found on the Internet. Instead, your JavaScript needs to dynamically display an image (either by unhiding an existing large cover image or dynamically adding it to the DOM) and changing the body or some other wrapper element's CSS class so that it dims the entire page (e.g., set background color to black and opacity to 50%).

USE CASE NAME:	**User Profile**
DESCRIPTION:	System allows user to view their profile information. **This case is only for groups of four**.
1.	Display the information from the Users table (FirstName, LastName, Address, City, Region, Country, Postal, Phone, Email) in a sensible, attractive way. In the next assignment, you will likely have to edit this data as well.

USE CASE NAME:	**Login**
DESCRIPTION:	Redirect to login page if not logged in
1.	None of the pages in this assignment should be accessible unless the user first logs in. You will need to check Session state; if the user has not yet logged in, then redirect to a page named login.php. This check should be on every page. I will expect the program design for this repetitive code to be sensible and low on code duplication. You have been provided with two new tables. The first is called **Users** and contains the following fields: UserID, FirstName, LastName, Address, City, Region, Country, Postal, Phone, Email. The other new table is called **UsersLogin** and contains the following fields: UserID, UserName, Password, Salt, State, DateJoined, DateLastModified. The actual password for each user is **abcd1234**, but has been salted and subjected to an MD5 hash function. That means to check for a correct login, you will have to perform two queries: the first to retrieve the Salt field for the entered UserName (e.g., tgoyer@apple.com) from the UserLogin table, and if that user exists, concatenate that salt value and the entered password, and run them through the MD5 function. If the result of that function matches the result in the Password field, then log the user in. The login form needs to be attractive and fit the design of the rest of the site. You may want to make use of some of the MDL components to improve the attractiveness and usability of your form.
2	If a user is successfully logged in, then a few things need to happen. First, you need to save the UserId value for the user in session state. To lower the number of database accesses throughout the site, also save the user's first name, last name, and email in session state. Second, redirect to whatever page was requested prior to the redirect to the login page. That means you need to pass the initially requested page name as a querystring to the login page. If no page was previously requested, then redirect to index.php. Third, for all subsequent pages, while logged in, change the info in the left-side navigation bar to the user's first name, last name, and email. I'll try to also supply a series of replacement portraits to use, but for now, just continue using your own portrait image from assignment 1.
3	If the login was unsuccessful, then redisplay the form with relevant textual error messages and formatting. I expect that you will change the styling of the form elements in some way to make it clearer that the login was unsuccessful. The textual error message should disappear once the user starts typing into the user name or password fields.

USE CASE NAME:	**Logout**
DESCRIPTION:	Allow the user to log out
1.	Add an icon to the top right toolbar that allows the user to logout. Clicking on it will empty out the relevant session state variables and redirect to the login page.

USE CASE NAME:	**Browse Employees**
DESCRIPTION:	System allows user to browse and filter employees
1.	Extend the Browse Employee page from the first assignment by adding an ability to filter the employee list. Your filters should be in its own card but the content of the card should be initially hidden. That is, the user should see that there is a filter card but only the title should be visible. When the user clicks on the filter title, it should open up and the user should be able to see the filter options; clicking on it again will hide the filter options. This hiding/unhiding will require some CSS and JavaScript. Be sure to make it visually clear to the user that the filter title is some type of toggle.
2.	The filter options are as follows: filter by last name and filter by city. The last name filter should be a simple textbox. The user can enter a name, which should match any employee whose last name begins with the search string. The city filter should be a <select> list. It should contain all the unique cities of the employees in the Employees table. When the user clicks the filter button, it should then update the list of displayed employees based on the specified filters. The user should be able to specify either a single filter (name or city) or both (name and city).

USE CASE NAME:	**Simple Search**
DESCRIPTION:	System allows user to search for employees via toolbar
1.	This use case is initiated when a user clicks the search icon on the upper right corner of the page. When the click clicks it, a box should appear directly underneath the toolbar with a textbox and a search button. This will require JavaScript to dynamically hide/unhide an element in response to the click event. The user can enter a search string, and then the page should switch to Browse Employees and filter the employee list by the entered text (i.e., employee last names that begin with the search string).

USE CASE NAME:	**Analytics**
DESCRIPTION:	System allows user to examine analytic information for the site. **This case is only for groups of four**. Groups of three will have to complete a version of this, however, in assignment 3.
1.	This page will contain an Admin Dashboard displaying analytic information for the site (see https://envato.com/blog/admin-dashboard-design-trends/ and http://www.designyourway.net/blog/inspiration/how-to-design-and-code-an-admin-dashboard-like-these-examples/ for more information and examples of admin dashboards). Some partial sample analytic information is contained within the BookVisits and Countries tables. Analytics data can get really large. The BookVisits table contains about 10000 records detailing unique visitors to the Books pages in the month of June 2017.
2.	Your Admin Dashboard will need to display the following information: • A table displaying the top 15 country names and their count. Each BookVisits record contains the CountryCode of the visitor. You will need to do a group query by CountryCode and count them and sort them by this count. • A list displaying the following information: a count of the total number of visits in June, the total number of unique countries the site had visitors from, the total number of employee to-dos in June 2017, and the total number of employee messages in June 2017. These should be formatted as a series of four horizontal boxes; with each containing a relevant icon, the number, and a label describing the number. These should be calculated from the database and not hard-coded. • A table of the top ten adopted books. This table should contain thumbnail image of book cover, title, and a sum of the Quantity in AdoptionBooks. The title should be a link to the Single Book page with the ISBN as a querystring. I will expect this to be designed in a sensible and attractive way that is consistent with the design of the rest of the site.

Guidance

This is substantial assignment and your group will likely need to invest 50-70 hours into it. Some of you will need much less but some groups might need more. I would recommend the following process:

1. This is a data-driven site, so I would recommend beginning by constructing the data access layer. To begin, create data access objects for Books, Employees, Users, and Universities (later create them for addional tables). The architecture provided in chapter 14/17 gives you `GetAll()` and `GetByID()` functionality already. You will need to add additional methods that match the functionality required by the use cases. I wouldn't bother implementing a complex business layer since this assignment is mainly just displaying data. I would recommend using the testdb-classes.php page provided in the labs
2. Retrofit your assignment 1 pages to use this new infrastructure.
3. Add filter functionality to browse-employee.php.
4. Implement the rest of the functionality.
5. Implement the login.

Some of these activities can happen at the same time. **The data access layer will almost certainly need to be done first however**. In past years, some groups get themselves into trouble by assigning the development of the data access layer to one person, who either being a procrastinator or a weaker programmer, completes the task (if at all) too close to the deadline, which means the assignment can't realistically be finished on time. So, be sure to get an early start on the data access layer! **I would recommend everyone in the group implement at least one of the data access objects.**

One Final Bit of Advice

Every year, several students are unable to successfully complete this course and have to take it again. In many cases, the thing that prevents them from achieving success is a low mark on this assignment (and the next one). In many cases, someone in a group will not do the work expected of them in assignment two: this will result in a lower assignment 2 mark. The rest of the group will then decide to drop that person for the third assignment, which almost always means an extremely low mark on assignment 3 for that person, and very likely a D or F in the course. So don't be that person! Don't free ride on the work of the others in your group or let your group down by procrastinating! It is very much in your interest to participate as equally as possible on the work in this assignment.