Project 1 Part C

Due Tuesday 11/03/2015 by 11:00pm

Late Submission Policy

To accommodate the emergencies that students may encounter, each team has 4-day grace period for late submission. The grace period can be used for any part of the project in the unit of one day. For example, a student may use 1-day grace period for Project 1A and 2-day grace period for Project 2B. Any single project part may not be more than 2 days late. Note that even if a team submits a project 12 hours late, they would need to use a full day grace period to avoid late penalty. If your project is submitted late, we will automatically use the available days in your grace period unless you specifically mention otherwise in the README file.

The final part of the Movie Database project is very open-ended and some students may want to get started early.

In the last part you are to (finally) create a fully functioning Movie Database system accessed by users exclusively through a Web interface. The functionality of your Movie Database system is quite flexible, although we expect all students to implement some baseline capabilities:

Four input pages:

- Page I1: A page that lets users to add actor and/or director information. e.g. Here are some "reasonable" names: Chu-Cheng Hsieh, J'son Lee, etc.
- Page I2: A page that lets users to add movie information.
- Page I3: A page that lets users to add comments to movies.
- Page I4: A page that lets users to add "actor to movie" relation(s).
- Page I5: A page that lets users to add "director to movie" relation(s).

Two browsing pages:

- Page B1: A page that shows actor information.
    - Show links to the movies that the actor was in.
- Page B2: A page that shows movie information.
    - Show links to the actors/actresses that were in this movie.

    - Show the average score of the movie based on user feedbacks.
    - Show all user comments.
    - Contain "Add Comment" button which links to Page I3 where users can add comments.

One search page:

- Page S1: A page that lets users search for an actor/actress/movie through a keyword search interface. (For actor/actress, you should examine first/last name, and for movie, you should examine title.)

  Your search page **should** support **multi-word** search, such as "Tom Hanks". For multi-word search, interpret space as "AND" relation. That is, return all items that contain both words, "Tom" and "Hanks". Since the search page is for actor/actress/movie, so if there was a movie named "I love Tom Hanks!", it should be returned. As for the output, you should sort them in a way that users could find an item easily.

A demo site is available here. This page is available strictly to give you an idea of the basic requirements, and is *not* meant to guide your choice of style or user interface in any way. Please be creative, and do not simply mimic the UI of the demo site.

Important Notes:

- "Pages" mentioned above are units of conceptual functions that you need to provide, not distinct "files". It is OK to implement multiple "pages" as a single php page. For example, it is OK to combine S1 and B2 by adding a search box to a page that shows movie information. You can also use multiple php pages to implement a single "page" in our spec.
- While the functionality of your Movie Database system is quite open-ended, the interface itself is *extremely* open-ended. This class is not a user interface class and *you can certainly get full credit for a solid system* with simple input boxes, menus, and/or radio buttons, and simple HTML output tables as long as your web site is reasonably intuitive to use. That is, the user should be able to navigate through your web site without too much difficulty. Having said this, we welcome much snazzier interfaces, like something resembling IMDb.com itself, or perhaps even something better!
- It is okay to embed some javascript or CSS to make your website look good, but please make sure your work is **completely self-contained**. Namely, all

javascript/CSS files related to your website should be incorporated in your submission.

- You may assume that the users are not malicious. They do not intentionally perform anything bad, such as an SQL injection. However, any PHP/database errors due to simple data entry errors or bad input values should be managed gracefully, i.e., it should be possible for users to continue interacting with the system, and the database should not get corruptted.
- You need to use the same set of tables (and only them) that you created in Part B of this project. You should use the database CS143 by connecting with username "cs143" and empty password.
- Please ensure that your work does not use more than 20MB space.

Although the project interface is very open-ended, we have to make the basic features of your system accessible in a typical browser environment. More precisely, you must make sure your system work with the most recent vesion of the Mozilla Firefox browser without any additional plugins or extensions. If you feel this restriction will somehow limit yourself too much, you must get "preapproval" from the TA to use a different browser environment. Send an email message to your TA telling him precisely what browser environment you wish to use for your project and why it is necessary. When the preapproval process is not followed, projects that have problems on Mozilla Firefox may lose points, possibly all points if we cannot run your project at all.

We may select a small number of Movie Database systems to be demonstrated to the class at time to be determined. Students will not receive extra credit, but they will receive extra recognition. The criteria for selection will be some combination of beyond-the-basics functionality and a good user interface.

Hints on implementation

- Please note that when your PHP scripts inserts a new tuple, your program will likely need to lookup either MaxPersonID or MaxMovieID table, assign a unique ID to the new actor/movie, and increase the MaxPersonID or MaxMovieID table value.
- In adding user reviews to a movie, you may need to obtain the current timestamp. You can do it either in your PHP code or in MySQL itself. The MySQL date and time functions page explains how you can obtain the current timestamp in MySQL.
- Note that your php pages can be accessed directly through a URL like http://guestip/~cs143/mypage.php?param1=value1&param2=value2&par

am3=value3 where mypage.php is the name of your php page, and the string after the question mark contains parameters passed to the php page just like in a "GET" method. So you can*embed a clickable link to the page* generated by your php code using hyperlinks of the above format, with appropriate parameters that instruct your PHP code to execute the right query.

Your Project 1B

- Note that your project 1C is dependent on your project 1B. To make sure you can get 100% score, please make sure your project 1B is working correctly and fix any existing bugs. As part of project 1C submission, you will have to submit the relevant files of your (revised) project 1B again.

Demonstrate your website

By now you should already have a working website, and in this section you are going to prove that your site is working correctly. To prove it, you are asked to use Seleniumproject, specificly Selenuim IDE.

What is Selenium IDE?

Selenium IDE is designed for automating browser tests. It is essentially a Firefox broswer plug-in. Primarily it is for automating web applications for testing purposes. However, here we use Selenium as an "screen action recorder". That is to say, Selenum allows you to record a sequence of steps and a TA could **replay** it.

Install Selenum IDE

1. Install Firefox (if you don't have it). Note that you should install Firefox to your operating system (windows/mac), but **NOT** to the virtual machine (ubuntu). Your ubuntu image (installed in the virtual machine) is just a web server and you need a client (which is your current operating system) to try your website. (Please download the latest released version.)
2. Download Selenium IDE: Link (Please download the latest released version.) In your Firefox browser, when you click the download link which is showed as a version number, your Firefox will pop up a window to let you install the plug-in.
3. If you don't know how to install an xpi...

Using Selenium (A Video Tutorial by Chu-Cheng)

Using selenium, create the following five test case files:

1. `t1.html`: use your search page to search for **Tom Hanks**, and it should return just ONE match (actor). Then you should click the actor "Tom Hanks" in your search result (it should be *a link pointing to a page that displays information about Tom Hanks*).
2. `t2.html`: use your web pages to create:
    - Actor: **Happy Jack, Male, born on 1994-10-10, alive now** (note: first name: **Happy** & last name: **Jack**).
    - Director: **Mary Queen, Female, born on 1977-02-27, died on 2012-12-31** (note: first name: **Mary** & last name: **Queen**).
3. `t3.html`: create a fake movie, title:**CS143**; company:**UCLA** ; year:**2013**; director:**Mary Queen**; genre: **Action** and **Horror**; MPAA: **G**; "Happy Jack" acted as role **TA** in this movie.
4. `t4.html`: use your review page to create two reviews: (1) content: **very good** and score: **5** (2) content: **excellent** and score: **4**. And then use your search page to search**CS143**, and your test case should stop at the page where we can see the average is 4.5, two comments your wrote, and other movie info.
5. `t5.html`: user your search page to search keyword **santa** (note: all letters in lowercase in this test case), and you test case should stop at the search result page, which should contain 7 actors (Actor: Toni Santagata(1940-12-09), Actor: Yusuke Santamaria(1971-03-12), Actor: Carlos Santana(1947-07-20), Actor: Henrique Santana(1924-05-07), Actor: Merlin Santana(1976-03-14), Actor: Peter Santana(1966-03-29), Actor: Carolina Santangelo(1986-08-05)) and 3 movies (Movie: Santa Claws(1996), Movie: Santa Fe(1997), Movie: Santa with Muscles(1996)).

Submission Instruction

Preparing Your Submission

Please create **a folder named as your UID**, and create "`sql`", "`www`", and "`testcase`" folders within your UID folder. Put your `create.sql` and `load.sql` files from project 1B into the `sql` folder, all your web site source files into the `www` folder, your selenium test case files into the `testcase` folder, and your README file into the UID folder. Compress your UID folder into a single zip file called "P1C.zip". In summary, your zip file should have the following structure.

```
P1C.zip
 |
```

```
 +- Folder named with Your UID, like "904200000" (without quotes)
     |
     +- readme.txt
     |
     +- team.txt
     |
     +- sql
         |
         +- create.sql
         |
         +- load.sql
     |
     +- www (all your website files are put under this folder, files listed are
just examples, you can choose other names)
         |
         +- index.php
         |
         +- search.php
          |
          +- others.php
          :
     |
     +- testcase
          |
          +- t1.html
         |
          +- t2.html
          |
          +- t3.html
          |
          +- t4.html
          |
          +- t5.html
```

Please note that the **file names are case sensitive**, so you should use the exact same cases for the file names. (For team work, use the submitter's UID to name the folder)

- `readme.txt`: A README file, with a detailed explanation of which of the project criteria you met, as well as any additional features you have implemented.
    - If you worked as a team with a partner, briefly explain how you split the work and collaborated in your README file. Also briefly explain what aspect you feel you can improve in a team setting for better collaboration.

- **`team.txt`**: A plain-text file (no word or PDF, please) that contains the UID(s) of every member of your team. If you work alone, just write your UID (e.g. 904200000).
  If you work with a partner, write both UIDs separated by a comma (e.g. 904200000, 904200001). **Do not include any other content in this file!**
- `sql/`: a folder which contains your (revised) `create.sql` and `load.sql` files from project 1B.
  - Please make sure that when you run **mysql CS143 < ./create.sql** and **mysql CS143 < ./load.sql** in this directory, there is no error.
- `www/`: a folder which contains all the source code and other supplementary files of your web site
  - After loading the database, the TA should be able to simply copy the files in this directory to ${HOME}/www and run your web site without any issues.
  - Your web site should use the database CS143 by connecting with username "cs143" and empty password.
  - Remember to use relative URLs in your machine/port references. You may be get zero point for code with faulty URL links that don't work during grading.
- `testcase/`: a folder which contains your five selenium test case files, named as `t1.html` through `t5.html`.
  - Make sure that your web server (virtual machine) is accessed through **localhost:1438** in your test case files. If you assigned a different address and/or port to your virtual machine for any reason, make sure you use an editor to replace your address to localhost:1438 in the `t?.html` files.
  - To ensure your recorded test cases work, **always replay your recorded test cases before you submit them**. If your test cases cannot run correctly, you will get as low as zero. If two persons work as a team, it may be a good idea to have one person record the test cases and have the other replay them.

Testing of Your Submission

Grading is a difficult and time-consuming process, and file naming and packaging convention is very important to test your submission without any error. In order to help you ensure the correct packaging of your submission, you can test your packaging by downloading this test script. In essence, this script unzips your

submission to a temporary directory and tests whether or not you have included all your submission files. The script for Project 1C can be executed like:

```
cs143@cs143:~$ ./p1c_test <Your UID>
```

(Put your P1C.zip file in the same directory with this test script, you may need to use "chmod +x p1c_test" if there is a permission error).

You **MUST** test your submission using the script before your final submission to minimize the chance of an unexpected error during grading. Again, significant points may be deducted if the grader encounters an error during grading.When evertything runs properly, you will see an output similar to the following from this script:

```
Check File Successfully. Please upload your P1C.zip file to CCLE.
```

Submitting Your Zip File

Visit the Project 1C submission page and submit **only your "PIC.zip" file** electronically by the deadline. In order to accomodate the last minite snafu during submission, you will have 30-minute window after the deadline to finish your submission process. That is, as long as you start your submission before the deadline and complete within 30 minutes after the deadline, we won't deduct your grade period without any penalty.

Grading:

- (10%) We can navigate your website using Firefox without any errors, and your websites contain all assigned pages.
- (20%) All your Selenium test cases point to a web server with address: **localhost:1438**, and your test cases complete without encountering any error/warning.
- (50%) Each test case is 10% if it acts as expected (correct result). Please follow "What to submit" precisely, especially all the values (case sensitive) you are asked to enter.
- (20%) 2 additional test cases designed by our grader/TA.

Notes:
(1) Do not record your screen with any other software. We only accept Selenium test cases.

(2) If your Selenium test cases have problems because you did not follow the spec, you may lose significant points.
(3) Start your project early and use our discussion forum, asking for clarifications if any part of this spec looks unclear to you.