F28WP Web Programming

Coursework 2 - Assignments Guidelines

(25% of total course mark)

Submission Deadline: **Monday 5th December at 18:00 Hours**

Learning Objective

The aim of this coursework is to demonstrate that you are able to implement the concepts discussed in class and described in the lecture notes. It should also show your knowledge of technology, such as PHP/Python, CSS3, HTML5, JavaScript, AJAX, databases (MySQL) and web services and; your ability of using them to create attractive and useful websites.

General Guidelines

This is a lab-oriented coursework, composed by five lab assignments and one "*website*." The five assignments must be completed during the lab sessions. Your work will be assessed and marked individually. There are no groups in Coursework 2. During each lab, you will complete an assignment that will be marked at the end of the lab. You cannot use external software libraries, web frameworks or open source code in the first three assignments. However, you can use jQuery, JS libraries and open source code to develop your "*website"*. You are expected to do most of the development work required for your "*website*", outside timetabled labs. You must demonstrate a working version of your "*website*" on Friday 2nd December 2016. Your user interfaces should use JavaScript, CSS3, HTML5 and responsive web design (RWD) techniques. Your server-side scripts should always validate the data entered by the user and manage sessions using cookies stored in your database.

Lab tasks, Dates and Marks

Tasks	Due Dates	Mark (%)
Assignment 1	Friday 28th October 2016	2
Assignment 2	Friday 4th November 2016	2
Assignment 3	Friday 11th November 2016	2
Assignment 4	Friday 18th November 2016	2
Assignment 5	Friday 25th November 2016	2
Website demo	Lab - Friday 2nd December 2016	6
Website submission	VISION – Monday 5th December 2016	9

Assignments: Description and Rubrics

LAB 1 - Assignment 1 (2% mark)

1.1 Database Exercise:

Create a table called '*my_movies*'. This table should contain the five fields below:

- movieID (unique numeric values only, automatically increased)
- movieGenre (alphabetic values, indexed)
- movieTitle (alphanumeric values, no longer than 200 characters)
- movieType (a value to indicate whether the movie is in English or not)
- movieRating (numeric values between 0 and 10)

Insert a couple of different movies to your '*my_movies*' table.

Note: Students can use the MySQL database from the command line of *anubis* or from https://www.macs.hw.ac.uk/phpMyAdmin.

1.2. Programming Exercise:

Using Python or PHP, write a script called either *movies.py* or *movies.php*. The script should connect to your database and then display the phrase "The movie of today is [MOVIE_TITLE]!" on the browser. **The scripts of your assignments and website must be installed on and run from http://www2.macs.hw.ac.uk/~[YOUR_USERNAME]/**

1.3 The *Website* Exercise:

Let the lecturer or lab helper know the type of website and the core functionality that you have decided to develop for the coursework. Don't forget to show your '*my_movies*' table as well as your "*movies*" webpage to the lecturer or helper to receive your marks and feedback.

Rubric:

Exercise 1.1: 0.70% if student can create a MySQL table and insert data into it.

Exercise 1.2: 0.80% if student can create a webpage using PHP (Python) and MySQL.

Exercise 1.3: 0.50% if student has selected the *website* & core functionality for their CW.

Note: Students who have not taken F27WD (Web Design and Databases) can work with the lecturer during Lab 1. They must complete and email the exercises of Lab 1 by Thursday 4th Nov 2016.

LAB 2 - Assignment 2 (2%)

2.1 Create the '*my_users*' table to store the user IDs, usernames and passwords of hypothetical movie buffs. Populate this table with the details of four users.

2.2 Create the '*users_movies*' mapping table with two columns only. The first column will store the user IDs of your movie buffs (from your '*my_users*' table). The second column will store movie IDs (from your '*my_movies*' table). Populate this table with at least six records.

2.3 Create a secure small webpage that allows users to sign-in by entering their username and password.

2.4 If the provided username and the password match a record in your '*my_users*' table, the script will start a session and display the '*Welcome* [USERNAME]!' webpage with a link to sign-out. Otherwise, it will display the sign-in page with a "Not Found" message.

Rubric:

Exercise 2.1: 0.20% if it was successfully completed.

Exercise 2.2: 0.20% when successfully completed.

Exercise 2.3: 1.00% if user can sign in and data is properly validated against hackers.

Exercise 2.4: 0.60% if 'sessions' is implemented and pages display the expected messages.

LAB 3 - Assignment 3 (2%)

3.1 Write a script that, once the user has signed in, displays a list with all the movies of the user. The list should include the title, URL, genre, action, poster and rating of each movie. The URL will be a link to the official movie trailer on YouTube. Leave the 'action' and 'poster' columns blank. You will implement their content and functionality in Lab 4.

3.2. Implement an option to allow the user to add more movies to her/his list of movies.

3.3. Show the alpha version of your *website* to your lab helper or lecturer. It should give a clear idea of how "your core functionality" will be implemented.

Rubric:

Exercise 3.1: 0.90% if the webpage displays the user's movies (+0.10% for RWD)

Exercise 3.2: 0.60% if user can add more movies to her/his list of movies.

Exercise 3.3: 0.40% if *website* works and shows "core functionality" as work in progress.

LAB 4 - Assignment 4 (2%)

4.1 Working on the webpage that displays the list with all the movies of the user, add a Delete button in each row (movie) in the 'action' column.

4.2 When the user hits the Delete button, your script will only remove the selected movie from the list of movies of the signed in user. If the movie is successfully removed, your script will replace the Delete button with the message: "Removed". Otherwise, the button should be replaced with the message: "Sorry".

4.3 Make sure that when a movie is removed, the “Removed” and “Sorry” messages are displayed in place of the Delete button WITHOUT REFRESHING THE WEBPAGE.

Rubric:

Exercise 4.1: 0.30% if each movie has a Delete button that when used displays a warning.

Exercise 4.2: 0.80% if a movie can be successfully deleted using its Delete button.

Exercise 4.3: 0.90% if student has made a good implementation of AJAX.

Note: In this assignment you can use any external software, JavaScript libraries, web frameworks and open source code.

LAB 5 - Assignment 5 (2%)

5.1 Add the ‘View’ button in each row in the ‘poster’ column. Behind the scenes, this button will use the free OMDb web services (API), which allows websites to search and retrieve movie posters, if available. For example this is the ‘API call’ to search for movies that have the keywords Home and Alone in their titles: http://www.omdbapi.com/?S=home+alone

5.2 When the user clicks on the 'View' button, your script will work ‘behind the scenes’ to create the ‘API call’ to invoke the OMDb API, execute the API call and parse the JSON string produced by the API to extract the URL of the movie poster.

5.3. If the movie poster is found in the JSON string and successfully extracted, your script will replace the View button with a small image of the poster. Otherwise, the button should be replaced with the message: “poster not found”.

Rubric:

Exercise 5.1: 0.30% if each movie has a View button pointing to the URL of the OMDb API.

Exercise 5.2: 1.00% if script can fetch data from OMDb API and extract poster’s URLs.

Exercise 5.3: 0.70% if functionality was implemented using AJAX correctly.

Note: In this assignment you can use any external software, JavaScript libraries, web frameworks and open source code.

The *website* (15%)

You are going to create a small website. This website can be a Q&A or a portal. During the first lab you will have the opportunity of selecting the type of website you would like to create. For this coursework, you only need to develop one of the core functionalities of your chosen website. Further information will be provided in our second lecture (week 7).

Q&A: A Q&A site is a type of online community or collaborative media. For this coursework, a Q&A is an online place where individuals with common interests can ask questions, share knowledge, and find solutions. Q&As exist to help people to find solutions. They bring

unskilled and expert users together. An example of the type of Q&A we would like you to create is a simplified version of StackOverflow (http://stackoverflow.com/tour).

Portal: A portal is an all-in-one website. It is a single point of access to information. It brings information together from diverse but related sources, making their access quicker, easy and convenient. A portal combines information from different sources into a single user interface but each source gets its dedicated area on the portal website. Users have the option to personalise the content of their portal. For example https://my.in.gov is a governmental portal and http://www.newsnow.co.uk is a news portal.

Website Demo (6%)

Every student will have the opportunity of presenting their *website*. The presentations and their assessments will be organised as follows:

- 12 groups of 5 students will be formed.
- You assess 4 other students' demos.
- You demo to the other 4 students, a lab helper and the lecturer.
- You must enter your assessments of 4 other students' assignments.
- A lab helper will simultaneously attend three demos and give their mark if needed.
- You have six minutes to demo your website.
- You are advised to rehearse your presentation before the demo day.

Your website will be assessed in three aspects:

- Functionality: Your website works without errors and its expected core functionality has been implemented. (2%)
- Design: You have used responsive design techniques, your website is attractive and its components (text, images and colours) are harmoniously laid out. (2%)
- Usability: Your website is easy to use, intuitive and the functionality can be accomplished quickly and effectively. (2%)

You are going to receive feedback on your work as well as marks for your demo during and after your presentation. You will receive full marks if you have designed and created an error-free, attractive, intuitive and useful website.

Website Submission (9%)

Your submission will be assessed in three aspects:

- Code Quality: You have used web programming best practices, implemented web script security recommendations and used MVC as much as possible. (2%)
- Interoperability: You have efficiently used JavaScript libraries (e.g. jQuery), AJAX, JSON and web services (e.g. REST) and API technologies. (2%)
- Report: Your have clearly described your work and included all the required items in your report (2%)
- Overall functionality, design and User Experience of your website (3%)

--.--

Submission Guidelines

1. Submit your coursework through VISION (https://vision.hw.ac.uk):
Use the following links to submit your work:

Web Programming F28WP_2016-2017 » Assessments » Coursework 2 Report Submission

2. Your submission must contain two files:

- A one-page PDF document including the following:
  - Your full name and Student code.
  - The URLs of your website located at http://www2.macs.hw.ac.uk/~[YOUR_USERNAME]
  - A small paragraph describing the core functionality you implemented in your website.
  - One statement describing how you have used AJAX, JSON, Responsive Web Design, web security, MVC and web service (APIs). Choose one of them, the one that you think was more relevant to your work.
  - References to your code to help us to verify your descriptions.
  - Additional comments to help us help you when we mark your coursework.

- A ZIP archive containing the following files:
  - All your source scripts and files used by your website, and organised in directories as advised in class.
  - A simple text file listing the names of your MySQL tables (We will use this list to check your database and run your website)

3. The student must work wholly on their own to write the software code of this coursework. However, students are permitted to discuss their ideas for the coursework with classmates and the lecturer and lab assistants.
4. Coursework reports must be written in a student's own words and any code in their coursework must be their own code. If some text or code in the coursework has been taken from other sources, these sources must be properly acknowledged.
5. Failure to reference work that has been obtained from other sources or to copy code of another student is plagiarism and if detected, this will be reported to the School's Discipline Committee. If a student is found guilty of plagiarism, the penalty could involve voiding the course.
6. Students must never give hard or soft copies of their coursework reports or code to another student. Students must always refuse any request from another student for a copy of their report and/or code.
7. Sharing a coursework report and/or code with another student is collusion, and if detected, this will be reported to the School's Discipline Committee. If a student is found guilty of collusion, the penalty could involve voiding the course.

8. **The final deadline for the coursework submission is Monday 5th December 2016 at 18:00 o'clock.** Late submissions will be penalised at the standard school rate, unless explicit permission to submit late is sought and granted before the due submission date and time. Permission to submit late will not be obtainable without manifest good reasons. Late completion of lab assignments are penalised at the rate of 15% per day.

Please contact me at S.Chumbe@hw.ac.uk if you have any question or need further guidance.