Building Web Applications using MySQL and PHP (W1)

Tutor Marked Assignment (TMA)

Introduction

The TMA carries 25% of the total marks for this module. Its aim is to help confirm that you understand the basics of building web applications with MySQL and PHP. It also ensures that you have mastered the basic skills needed to complete the FMA and provides your tutor with the opportunity to feedback comments on your work.

The completed TMA deliverables should be submitted electronically in the assignment dropbox in Moodle BEFORE the start of session 5.

Completing the TMA

You should work on your TMA after class and during the self-study session scheduled after Session 4. Begin your work early, as the TMA is a substantial task that requires planning and effort to complete satisfactorily. The TMA prepares you for the FMA so you greatly reduce your risk of a poor overall mark by completing and submitting a TMA.

Getting support

Support for the TMA work will be available in class during Session 4.

TMA Specifications

Introduction

You have been asked to build a prototype web application which will use PHP to display data stored in a MySQL database as HTML. You have been provided with the SQL statements required to create and populate the tables for a database of songs and artists which will act as sample data for your application.

The application you build should include the following aspects:

1. All of the *views* of the application should be provided via a single point of entry, saved as *index.php* in your application directory.
2. The default *view* for the application will be a short welcome message (you may use *Lorem Ipsum* text).
3. There will be a second *view* which will display a list of all of the active artists in the database accompanied by a suitable heading. For each artist you must display their name and the number of songs accredited to them. If an artist has no songs in the database, they should not be included in the list. The list should be sorted by the artist name in ascending order.
4. There will be a third view which will display all of the songs in the database accompanied by a suitable heading. For each song you must display the title of the song, the artist name and the duration of the song in the format: *mm:ss* (duration is stored in seconds in the database). The list should be sorted first by artist, then by the song title, both in ascending order.
5. A summary of the songs and artists on the system should also be displayed on each *view*. The summary should inform the user of the total number of songs and the total number of active artists in the system.
6. The user should be able to navigate between the *views* by way of a simple navigation menu. The application name, **W1 Music** should be clearly displayed on each *view*.
7. The HTML and CSS design of the application has not been finalised so you should build your application on the assumption that the HTML which it generates will have to be modified at a later date.
8. The application should also be built with portability in mind so it can easily be deployed to alternative environments and/or locales.
9. All (X)HTML should be semantically appropriate for the context. Feel free to style your solution with CSS as you wish, although any such styling does not gain you extra

credit. All (X)HTML should validate to the specified DOCTYPE with no inline CSS or formatting tags.

Any deviation from these instructions must be agreed with your tutor.

Working towards a Solution - Some Hints!

- Familiarise yourself with the provided database tables and ensure that you can write the queries required to render the *views*, before attempting to execute them in your PHP scripts.
- Plan your application fully before writing any code and think about the functions or classes you will need to implement. Code should not be repeated - marks are awarded for careful use of functions/classes and code re-use.

Deliverables for submission

TMA Solution

You must deploy your solution in your BBK personal web space on the following URL:

http://titan.dcs.bbk.ac.uk/~username/w1tma/index.php

You should also include the full address of your application's home page in your *readme.txt* file. The readme.txt file should also contain full instructions describing how to deploy your application.

Source Files

You must submit all PHP source files, your *readme.txt* file and any MySQL table creation/alteration source files, packaged as an application in a single zip file called **ITSUsername_w1_tma.zip**, replacing ITSUsername with your actual ITS Username. If you fail to submit a zip file containing all of the required files then your work cannot be marked and you will be awarded 0% for this component.

Getting feedback

Feedback on the marked TMA can be downloaded from Moodle and will normally be returned to you within 2 weeks of submission. The feedback on your TMA and any issues that arise can be discussed with your tutor within 2 weeks of the return of the marked TMA.

Backing up files

Always keep a back-up copy of all work submitted for assessment in case of unforeseen submission problems.

Plagiarism

Plagiarism, which is claiming the work of others as your own, is a serious offence and can result in your exclusion from all colleges of the University of London. You should be aware that we use a range of automated tools to spot potential plagiarism in work submitted for assessment. Providing you clearly reference work done by others that you have included in your FMA you will not be penalised.

Criteria for Assessment

You will gain maximum credit for a solution that uses solely your own original code. If you have used any code that is not your own in your solution, or have used other's work as a reference to develop your own code, the source should be fully referenced in code comments (see note about plagiarism above).

The criteria below show the proportion of the marks (out of 100%) that will be awarded for each component of the assignment:

1. **Functionality (30%):** all views correctly rendered; data validated and escaped appropriately; Single Point of Entry implemented.
2. **Application Design (30%):** Separation of concerns; Organisation of files; Portability; Documentation (readme.txt)
3. **Coding Style (26%):** Well commented, readable PHP and SQL sources; Efficient code; Encapsulation of tasks;

4. **HTML Presentation (14%):** Published on correct URL. Valid (X)HTML to specified doctype, with NO inline CSS or formatting tags.

Note: No marks will be awarded for the following:

- The use of functions marked as *deprecated* in the PHP manual
- The use of non-PHP technologies to achieve the required functionality
- Code which suppresses PHP error messages or attempts to alter PHP's configuration in order to suppress error messages.