Autumn 2020: Web Application Assignment

Due by 11:59pm on Friday 5th June 2020 **Assessment Weight: 50%**

A. Requirements

a) ALL instructions given in this document MUST be followed in order to be **eligible** for full marks for the Web Application Assignment. This document has four (4) pages.

b) This assignment is **NOT** a group assignment; collusion, plagiarism, cheating of any kind is not acceptable. As part of your submission you MUST certify that all work submitted is your own. If you cannot honestly certify that the work is your own then do not submit the assignment. Breaches of the Misconduct Rule will be dealt with according to the university policy (see the learning guide for more information).

c) All assignment submissions will be checked for academic misconduct by the use of the MOSS program from Stanford University. Details on MOSS can be obtained from the MOSS web site http://theory.stanford.edu/~aiken/moss/

d) Design the web pages with ease of navigation and operation, attractiveness and accessibility in mind. Images other than those provided in the assignment zip file (if any) may also be used in the assignment.

e) All assignment source files are to be uploaded to the **Assignment1** folder of your TWA web site prior to the due date and time. You may create subfolders within this folder if you wish.
   - Compressed archive files (eg, zip, tar etc) are not acceptable and will not count toward submission requirements.

f) The submission script must then be used to submit your work once all work is on your TWA website. Failure to use this script will result in a non-submission. See section D of this document for submission details.

g) All styling and page layout must be achieved using CSS. The use of Bootstrap or other frameworks is not permitted.

For the problem definition described in section B you must

h) include your authorship details at the top of **each** file in coded comments;

i) **reference** all sources that you used for inspiration of your solution as per Section C of this document;

j) ensure that your web application renders correctly in Chrome and runs correctly from the TWA web server.

B. Web Application Assignment Details

B(i) - Background information and description

Depending on which provider you use and the type of music that you want to stream there are currently multiple music streaming services available such as Spotify, Deezer, Tidal, Amazon Prime Music, Apple Music, PrimePhonic, and YouTube Music.

In this assignment you will create a web-based application for a product similar to Spotify named **24/7Music**.
The 24/7Music web application will allow

- members and non-members to search for songs, artists, albums
- members and non-members to view lists of songs, artists, albums
- members and non-members to play short samples of selected songs
- members to create playlists of their favourite songs

This is a very simplified web application and leaves out aspects that would be required in a real music streaming web application [such as actually streaming the music!!]. Read the Functional Requirements section (**section B(ii)** of this document) to determine what is specifically required for 24/7Music. The MySQL database that supports the functionality of 24/7Music is named 247Music. The 247Music database is described in **section B(iii)** of this document. You will need to design and create the PHP web pages for this assignment.

B(ii) – Functional Requirements

The 24/7Music web application **must**

a) be coded using HTML 5, CSS, JavaScript, and PHP **as necessary**. Note: all files described below must be PHP files to achieve the server-side functionality.

b) provide easy-to-use navigation for the user as described in the following page descriptions.

c) provide the following page content and functionality for each page as described.

Search Page (search.php)

Page purpose/description:
This page is a search page to find songs, artists, or albums. The search page can be used by both non-members and members.

Page content:
The page will have the following content:

1. Navigation to the other pages of 24/7Music ***as appropriate*** including a ***Log Out*** link (if the user is logged in) or a ***Log In*** link (if the user is not logged in).
2. member username (if the user is logged in)
3. membership category (if the user is logged in)
4. An HTML form (the search form) which contains
   - a text box to enter the keyword(s) for the search (mandatory field)
   - a submit button

   As indicated above the search text box is mandatory. That is, a search may be performed only if a search keyword has been entered.
5. A 'search results' section that lists songs, artists, albums that match the keyword(s) entered in the form.

Page functionality:
The Search page will

1. validate that a search keyword has been entered in the search form (using client-side JavaScript). Do not allow the form to submit unless a search keyword has been entered. Any characters may be entered.
2. use **postback** for the search form,
3. sanitise the submitted keyword(s) to avoid possible sql injection then search the 247Music database for records that contain the keyword(s) entered in the search form. The search should check for matches with song titles, artist names, and album names. The search may find zero, one or many matches depending upon the search term(s) entered. The output from the search depends on the number of matches found:
   a. **One or more records found:** generate a summary list of matching songs, artists, and albums underneath the search form. The user will be able to choose the match they want to view more detail about from this list (see point 4 below) by clicking on the match. The summary list should be **displayed in groups** of songs, artists, albums showing:
      i. The name of the song, artist, album that has matched with the search keyword(s)
      ii. For all songs that match, also list the Artist name
      iii. For all albums that match, also list the Artist name
      iv. A thumbnail image for the match

   b. **Zero records found:** display an appropriate message underneath the search form instead of a list of results
4. enable the user to choose a song, artist, album match for viewing/playing from the search results. To do this, the name of the song, name of the artist, or name of the album in the search results list is to be a **hypertext link.** When clicked, the hypertext link loads the Play page (**play.php**) which displays more specific details for the chosen match. Details of the Play page are given below.

General Page Notes:
The design of this page is open to your creativity but it must appropriately display all required information in a clearly readable, well organised and usable format.

Play Page (play.php)

Page purpose/description:
The purpose of this page is to display to the user the content of the **chosen** match. The content displayed will vary depending upon the type of match chosen by the user. The chosen match could be for an artist, album, playlist, or a song. The user may make the choice of match on the **Search** page (as described on the previous page of this document), on the **Play** page (as described below in point 4), or on the **Playlist** page (as described on the next page of this document). The **Play** page can be used by both non-members and members.

Page content and functionality:
The page will:

1. include navigation to the other pages of 24/7Music ***as appropriate*** including a ***Log Out*** link (if the user is logged in) or a ***Log In*** link (if the user is not logged in).
2. display the member username (if the user is logged in)
3. display the membership category (if the user is logged in), and
4. display content depending on the type of **chosen** match as follows:
   a. if the chosen match is an **Artist**, display the name of the artist and the thumbnail image (if one is available). Then display a list of all albums by this artist. For each album, display the album name, the year the album was released and the thumbnail image (if one is available). Each album name should be a hypertext link that will open the **Play** page so that the **Play** page may then display the details of the chosen album (see point 4b)**.**
   b. if the chosen match is an **Album**, display the album thumbnail image (if one is available), album title, artist name, and a list of all songs from the album. For each song from the album display the song title, and song length. Each song title should be a hypertext link that will open the **Play** page so that the **Play** page may then display the details of the chosen song (see point 4d).
   c. If the chosen match is a **Playlist**, display the playlist name and a list of all songs in the playlist. For each song in the playlist display the song title, artist name, and song length. Each song title should be a

hypertext link that will open the **Play** page so that the **Play** page may then display the details of the chosen song (see point 4d).

d. if the chosen match is a **Song**, display the song title, album title, song length, a link to 'add to playlist' (only available to logged in users), an iframe to play a sample of the song using the spotify track id (see details below)

Playing a sample of a chosen song:
To be able to play a preview of the chosen song use the following code to embed the Spotify iframe within the page

```
<iframe src="https://open.spotify.com/embed/track/##########" width="300"
height="380" frameborder="0" allowtransparency="true" allow="encrypted-
media"></iframe>
```

where ########## represents the `spotify_track` value from the `track` table in the database.

General Page Notes:
The design of this page is open to your creativity but it must appropriately display all required information in a clearly readable, well organised and usable format.

Playlist Page (playlist.php)

Page purpose/description:
Playlists are collections of songs. Playlists can only be created and accessed by logged in members. The page enables logged in members to create new playlists, add songs to their playlists, play songs from their playlists. Songs cannot be removed from playlists. Playlists cannot be deleted. Playlists belong to a specific member; logged in members cannot access any other member's playlist(s). Non-members cannot access this page and hence cannot create or change playlists.

Page content and functionality:
The page will:

1. include navigation to the other pages of 24/7Music ***as appropriate*** including a ***Log Out*** link,
2. display the member username for the logged in member,
3. display the membership category for the logged in member,
4. include a postback form to enable the logged in member to create a **new** playlist. The playlist name will need to be sanitised and validated on the server-side before being inserted into the database. Playlist names may only contain alphanumeric characters and spaces,
5. include a postback form to enable the member to add a song to a chosen playlist,
6. display a list of all playlist names (that belong to the logged in member). Each playlist name is to be a hypertext link that will open the **Play** page so that songs from the chosen playlist may be viewed as described in the **Play** page.

General Page Notes:
The design of this page is open to your creativity but it must appropriately display all required information in a clearly readable, well organised and usable format.

Member Login Page (login.php)

Page purpose/description:
The purpose of this page is to provide a login facility for members/subscribers of 24/7Music. Some of the functionality described in previous pages is only available to users after they have logged in.

Page content and functionality:
The page will:

1. include navigation to the other pages of 24/7Music ***as appropriate***,
2. include a postback login form which contains
   - a text box to capture the **member username**
   - a password box to capture the **member password**
   - a submit/log in button

   When the form is submitted by the user the page will need to **authenticate** the member credentials (**`username`** and **`password`**) as supplied by the member in the login form against the records in the **`membership`** table of the 247Music database.

   The page will allow or deny access to the subsequent pages of 24/7Music depending on the result of authentication as follows:

   a. **Successful login attempt**: Successful authentication should automatically redirect the member to the Search page.

b. **Failed login attempt:** Unsuccessful authentication should automatically redirect to the login page so that the user may try to login again. An appropriate error message must be displayed to the user if the login attempt fails. The message display must be implemented using appropriate PHP server-side code; it is not to be a JavaScript alert (or any other browser side component). The message should be displayed in an appropriate location on the login page in a suitable colour.

3. A link to the Sign Up page (the Sign Up page is not implemented in this version of the web application)
4. The provision for displaying error messages as detailed below

Member Login Credentials:
The passwords that are stored in the `membership` table of the 247Music database are encrypted using the sha256 algorithm (the passwords are not salted to decrease the complexity of implementation). A list of member `usernames` and (plain text decrypted) `passwords` can be found in the **Member Credentials** document.

Log Out Page (logout.php)

The purpose of this page is to log the member off the web application and automatically return them to the Search page. The Log Out page must not display anything to the user. The page should remove all session variables and end the session (if sessions were used). It should then automatically redirect to the Search page. This page can be accessed from any of the pages as detailed above.

B(iii) – Database Description

To connect to the ***247Music*** database use the following in your php script

```
$dbConn = new mysqli("localhost", "TWA_student", "TWA_2020_Autumn", "247Music");
if($dbConn->connect_error) {
  die("Failed to connect to database " . $dbConn->connect_error);
}
```

Notes

- The tables within the ***247Music*** database have already been populated with some data. Use the supplied `allTables.php` script to view the data (make sure you use the connection information as indicated above).
- Tables in the ***247Music*** database are described in the **247Music Data Dictionary** document.
- A list of user names and [plain text decrypted] passwords can be found in the **Member Credentials** document.

C. Referencing

Referencing must follow the guidelines given in Section 2.5.2 of the TWA Learning Guide.

D. Submission Instructions

To submit your Assignment, you must do the following by the due date and time specified on page 1 of this document.

1. Upload all your assignment files in the **assignment1** folder in your TWA web site on the TWA server

2. Run the submission script located at

   `http://twaaut.cdms.westernsydney.edu.au/submit/submit.asp`

   As part of the submission, you will be prompted for your TWA website username and password. You will then be asked to read the WSU policy on misconduct and certify that work submitted by you is your own work. This action will be logged in a database for future reference and is deemed to be evidence that you claim that your work is original. Next, you will need to select from a drop down list the Assessment you are submitting, eg, **Assignment 1**, and click the ”Submit Assessment” button. The web page will then display a listing of the files you have submitted along with a receipt number. You should print this page for proof of submission.

E. Marking Criteria and Standards

The marking criteria and standards for the Web Application Assignment are published in Section 2.5.2 of the Learning guide and will be used to assess your assignment submission according to the specific weightings identified in the table below:

Code Functionality/Correctness	60%	Web Page Design	25%
Form Design	10%	Code Readability	5%