COMP3322 Modern Technologies on World Wide Web

Assignment Three

Total 16 points

Deadline: 23:59 April 18, 2025

Overview

You are going to develop a Music Streaming application using JavaScript, PHP, and MySQL database. Once users are logged into the application, they can browse the list of songs under different genres and listen to their favourite songs. The application will automatically log out users once their login session has exceeded the expiration duration.

Objectives

1. A learning activity to support ILO 1 and ILO 2.
2. To practice using JavaScript, AJAX, PHP, session control, and MySQL to create a dynamic interactive web application.

Specification

You must develop the application using the course's LAMP docker containers.

You will develop two PHP programs – index.php and file.php, along with their respective CSS stylesheet (look.css) and JavaScript file (handle.js). It is assumed that all files, including the images and music, are located within a folder in the public_html directory of the Docker web server. You can use a different name for the folder; in this document, we name the folder to 2425B-ASS3.

```
2425B-ASS3
├── [Images]
├── [Music]
├── file.php
├── handle.js
├── index.php
└── look.css
```

Here is a quick overview of the four program files.

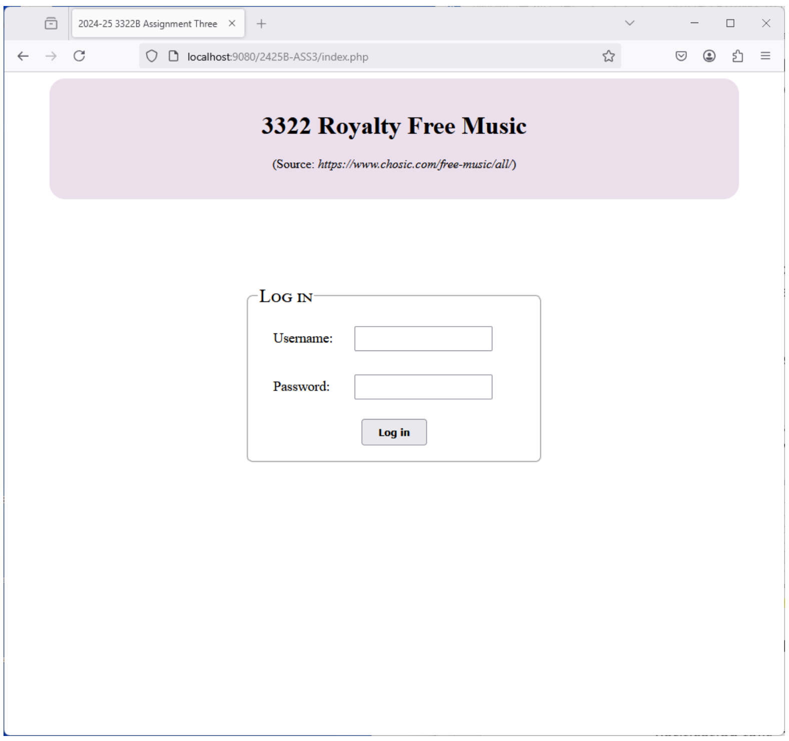
- index.**php** – a webpage for users to log on to the system, select and play the music, and find the list of music under a specific genre; this program should implement session control.
- file.**php** – a PHP program for supporting the client's music streaming request; this program should also implement session control.
- look.**css** – the styling file contains all CSS style rules for the index.php page.
- handle.**js** – the scripting file contains all JavaScript code to support the dynamic features of the index.php page.

You are **not allowed to use external libraries** for this assignment.

INDEX.PHP

Implement the index.php program to handle the GET /2425B-ASS3/index.php, POST /2425B-ASS3/index.php, and GET /2425B-ASS3/index.php?search=xxxxxxxx requests. This PHP program also has logic to handle session control. Feel free to choose a different folder name if you prefer; '2425B-ASS3' is provided as an example for your reference.

1. Users access the application by sending a GET request to /2425B-ASS3/index.php. The page always shows the login form when there is no active/authenticated session.


2024-25 3322B Assignment Three
localhost:9080/2425B-ASS3/index.php

3322 Royalty Free Music

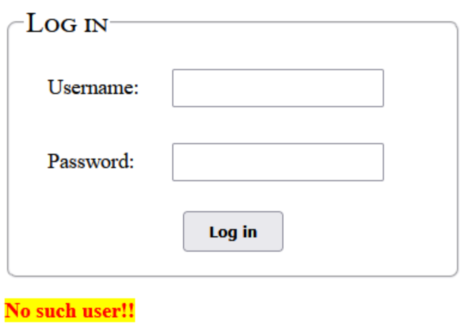
(Source: *https://www.chosic.com/free-music/all/*)

Log in
Username:
Password:
Log in

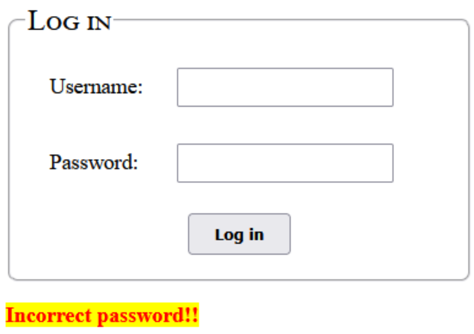
The page should contain a header with the title "3322 Royalty Free Music" and a quote to the source: https://www.chosic.com/free-music/all/. Users can click on this URL to load this target web page in another browser tag. Below the header is a form that contains a "Username" field, a "Password" field, and a "Log in" button.

Users enter the required fields in the login form for authentication. Once the users press enter or click the "Log in" button, the program should send a POST request to /2425B-ASS3/index.php with the message body carries the parameters: username={username} and password={password}, where {username} and {password} are the values entered by the users in the required fields.

If successful authenticated, the program should establish a session and redirect the browser back to the index.php page. In the case of authentication failure, the program should display the login form again, accompanied by a notification message below the form to inform users of the issue. The message should be **generated by PHP code** and send back to the client together with the form.

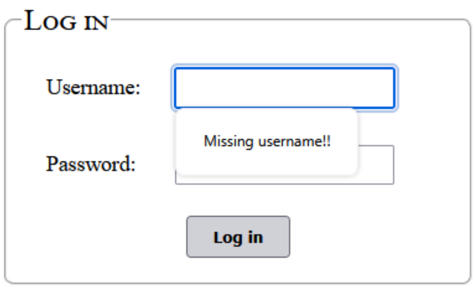


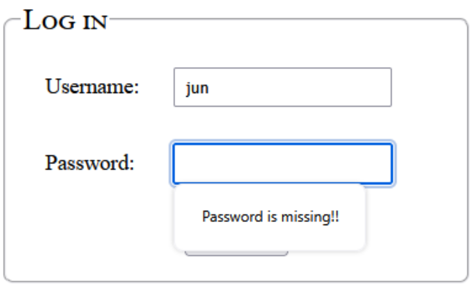
No such user!!



Incorrect password!!

The program makes use of HTML attributes and JavaScript to perform client-side validation on the form data. The program should check whether the Username and Password fields have the inputs **before submitting** the data to the server. If it detects missing input, it should display an alert on the corresponding input field.





Please handle this checking by JavaScript.

2. User accounts
You should create a table in the database (db3322) in our docker container, to store user account information. Here is a reference SQL statement in creating the **account** table.

```
CREATE TABLE `account` (
  `id` smallint NOT NULL AUTO_INCREMENT,
  `username` varchar(60) NOT NULL,
  `password` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
);
```

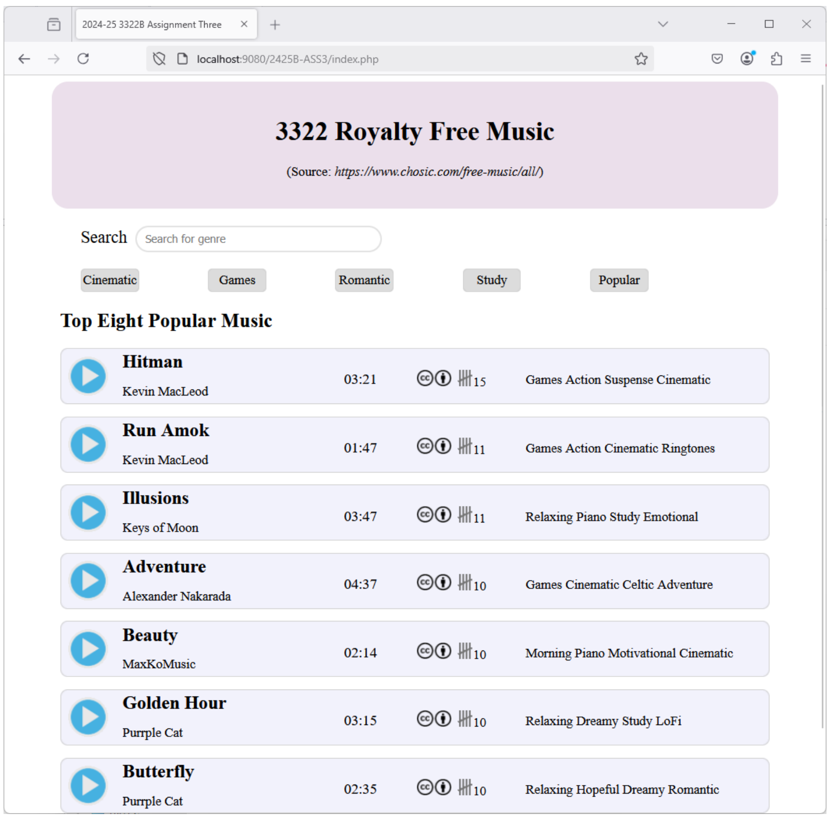
When grading your assignment, we would expect the application is using the above table schema with the table name – account.

3. When the user successfully authenticated and logged in to the application, the system should create a new session for this user and trigger the browser to issue another GET /2425B-ASS3/index.php HTTP request. Instead of showing a log in form, the application should show a list of music.

Here is the look of the application after successful authentication. It should contain:

- The same header block.
- A search bar for users to search for music under a genre.
- A few buttons that allow users to quickly search for music under the genre specified by the button.
- The list of top eight popular music, which are ranked by the number of play counts and listed in reverse order. Before the list of music, there is a title line that provides descriptive information about the list.
- For each music entry, it should show:
    - A button for playing or pausing the music.
    - The title of the music and the artist.
    - The total duration of the music (in mm:ss format).
    - The licence & attribution icon and the play count.
    - The list of genres to which the music may be categorized.

[Server-side rendering] All the HTML content should be generated at the server-side before sending it to the client.


2024-25 3322B Assignment Three
localhost:9080/2425B-ASS3/index.php

3322 Royalty Free Music

(Source: *https://www.chosic.com/free-music/all/*)

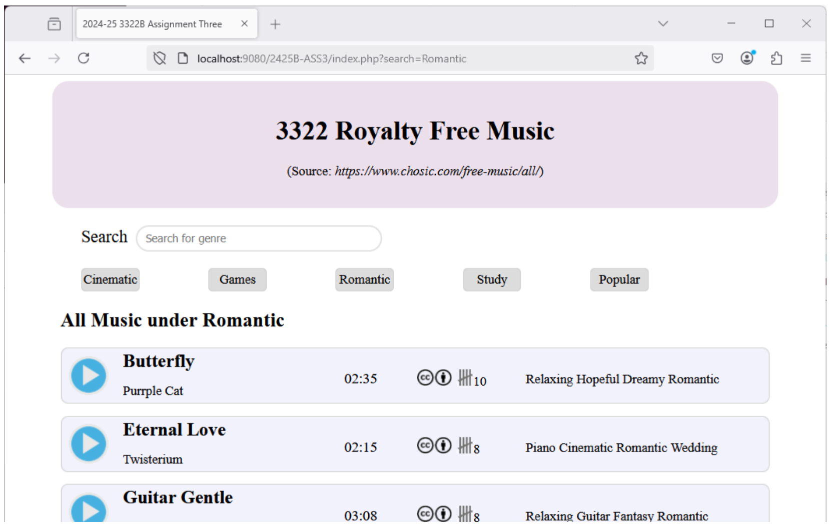
Search [Search for genre]

Cinematic | Games | Romantic | Study | Popular

Top Eight Popular Music

Hitman Kevin MacLeod	03:21	15	Games Action Suspense Cinematic
Run Amok Kevin MacLeod	01:47	11	Games Action Cinematic Ringtones
Illusions Keys of Moon	03:47	11	Relaxing Piano Study Emotional
Adventure Alexander Nakarada	04:37	10	Games Cinematic Celtic Adventure
Beauty MaxKoMusic	02:14	10	Morning Piano Motivational Cinematic
Golden Hour Purrple Cat	03:15	10	Relaxing Dreamy Study LoFi
Butterfly Purrple Cat	02:35	10	Relaxing Hopeful Dreamy Romantic

4. Users can use the search bar or genre buttons to find music within a specific genre. The application will initiate a search when a user enters a keyword in the search bar and presses the 'Enter' key, or when a user clicks one of the genre buttons. Upon pressing 'Enter' to begin the search, the program will clear the search bar and execute the search. However, if the user presses 'Enter' on an empty search bar, the program will not conduct the search.


2024-25 3322B Assignment Three
localhost:9080/2425B-ASS3/index.php?search=Romantic

3322 Royalty Free Music

(Source: *https://www.chosic.com/free-music/all/*)

Search [Search for genre]

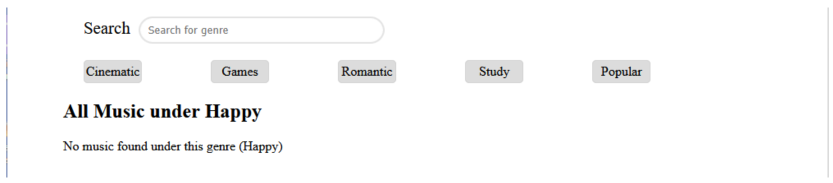
Cinematic | Games | Romantic | Study | Popular

All Music under Romantic

Butterfly Purrple Cat	02:35	10	Relaxing Hopeful Dreamy Romantic
Eternal Love Twisterium	02:15	8	Piano Cinematic Romantic Wedding
Guitar Gentle	03:08	8	Relaxing Guitar Fantasy Romantic

All searches are handled by the server. When initiating a search, the application sends a GET request to /2425B-ASS3/index.php with a query string containing search=xxxxxxxx, where ‘xxxxxxxx’ represents the genre being searched. For instance, to search for music in the Romantic genre, the application sends /2425B-ASS3/index.php?search=Romantic to the server. The only exception is the Popular button, which prompts the application to revert to the default list of the top eight popular songs. In this case, the application should send /2425B-ASS3/index.php to retrieve the top eight popular tracks.

The list of music should be ordered in reverse order of the play counts, and the title line should be updated accordingly. If a search yields no results for a particular genre, the message "No music found under this genre (xxxxxxx)" should be shown.



All Music under Happy

No music found under this genre (Happy)

FILE.PHP

When a user clicks on the Play button ▶, this would trigger the **audio element** to start playing the corresponding music. When playing a song from the beginning, the browser should send a GET request to the server using the path /2425B-ASS3/file.php?musid=XXXXXXXX, where XXXXXXXX represents the music ID.

Implement the file.php program to handle the GET /2425B-ASS3/file.php?musid=XXXXXXXX request from the browser. This PHP program should also contain logic to handle session control. Every song within the system is assigned a unique music ID, serving as a key in the Music database table. All songs are contained within the Music folder, **concealed from users** to prevent direct access to its file path. Your implementation must ensure that the actual file path of the music files remains undisclosed.

5. Music database table
   To support this Music Streaming Application, we need to set up a table in the db3322 database to store all information about the songs. Please download two files - the Music.csv file and the Music.zip file.

   The Music.csv file contains all the music data including the ID, the title, the artist, the length, the path, the filename, the play count, and the tags (which contains the list of genres the song belonged to). We can import the Music.csv file to the db3322 database using the PHPMyAdmin program and create the database table called **Music**. Below shows the first four rows of the CSV file.

_id	Title	Artist	Length	License	Path	Filename	Pcount	Tags
IDCKB4050	Adventure	Alexander Nakarada	04:37	Creative Commons CC BY 4.0	Music	Adventure.mp3	2	Games Cinematic Celtic Adventure
IDCKB4051	Cyberpunk Computer Game | IDRA	Alex-Productions	03:38	Creative Commons CC BY 4.0	Music	Cyberpunk-computer-game-idra.mp3	2	Games Action Electronic Trailer
IDCKB4052	Hitman	Kevin MacLeod	03:21	Creative Commons CC BY 4.0	Music	Hitman.mp3	3	Games Action Suspense Cinematic
IDCKB4053	Run Amok	Kevin MacLeod	01:47	Creative Commons CC BY 4.0	Music	Run-Amok.mp3	8	Games Action Cinematic Ringtones

   You should not change the data except the Pcount (play count) field.

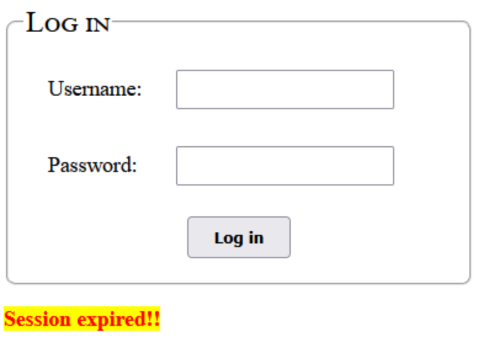
The Music.zip file contains all the songs. Uncompress the file in your assignment folder (e.g., 2425B-ASS3) and create the Music folder. As this folder should be hidden/concealed from users, we may rename this folder to other name when performing the grading.

6. The application should provide the following dynamic behaviour.
   - When the user clicks the play button to start playing the music **from the beginning**, a GET request is sent to the server to stream the music song with the URL: /2425B-ASS3/file.php?musid=XXXXXXXX. The play button ▶ will be replaced by the pause button ⏸.
   - When the user clicks the pause button, the music will pause, and the pause button will be replaced by the play button. You should **implement a mechanism** to alert the user that the song is in a **paused state**.
   - To resume playback, the user clicks the play button, which is currently in the paused state, to continue playing the music. The play button will be replaced by the pause button.
   - When the music comes to an end, the program should switch back to the play button.

7. Update the play count
   - **Every time** when the file.php program receives a GET request to the URL: /2425B-ASS3/file.php?musid=XXXXXXXX, it should inclement the play count ('Pcount') of the music with ID XXXXXXXX by one.
   - Whenever the user plays that music from the beginning, the play count should be increased. This includes the user plays the same song again.
   - It is **not a requirement** of the program to immediately show the updated play count value after starting the music. Instead, we anticipate the updated value to be shown when the **program reloads** or **loads a new music list** following a search.

SESSION CONTROL

8. Once the user has logged in successfully, an active session will be created, **lasting for 300 seconds**. During this period, the user can utilize the search service and play songs. Upon session **expiration**, any subsequent search actions or play button clicks by the user should redirect the browser to the index.php page, **displaying the notification message**: "Session expired!!".



Session expired!!

The notification message should be generated by PHP code and send back to the client together with the form.

9. Both the programs, index.php and file.php, contain the logic for session control and management. In the case of an active session, index.php should display the page with the music list. Conversely, if there is no active session, index.php should show the login form. For file.php,

when an active session is detected, it should handle music streaming requests. On the other hand, if there is no active session, file.php should respond with a **status code of 401** to signify unauthorized access and reject the request.

10. Session control occurs when the user initiates playback by clicking the play button, triggering an HTTP GET request to the file.php program. On the other hand, session control is not triggered when the user clicks the play button to resume playback or the pause button to pause it.

LOOK.CSS AND HANDLE.JS

The look.css file contains all the style rules for the index.php page. The handle.js file contains all the JavaScript code for the interactive and dynamic behavior of the Music application.

Resources

You can download the file resource_ASS3.zip from (http://i.cs.hku.hk/~atctam/c3322/resource_ASS3.zip). This file contains the following files:

- Music.zip – there are 24 songs downloaded from https://www.chosic.com/free-music/all/.
- Music.csv – A CSV file for importing all music data to the db3322 database.
- CC4.png, count.png, pause.png, and play.png – four image files

Testing platform

We shall run the server program in the LAMP container set and use Chrome to test the programs. The expected screen widths for testing are: 1000px ≤ vw ≤ 1500px.

Submission

Please finish this assignment before Friday April 18 23:59. Submit the following files:

1. index.php
2. file.php
3. look.css
4. handle.js
5. GenAI usage report (if applicable)
6. Any files that are necessary for testing the application

Both index.php and file.php are the primary service endpoints visible in this application. You have the option to introduce an additional PHP file that contains shared PHP code utilized by both index.php and file.php programs. This approach aids in minimizing maintenance effort.

Grading Policy

Points	Criteria
4.0	The Login page and the Music page ▪ Display the login form when there is no active session.

	▪ Manage the authentication process and establish an active session when successful authentication. ▪ Display the login form with a notification message in case of authentication failure or session expiration. ▪ Show the music page when there is an active session. ▪ Process search requests and deliver accurate results.
4.0	Access music files ▪ Return a status code of 401 for unauthorized access when there is no active session or incorrect access. ▪ When there is an active session and a valid music ID, locate the music file, stream the music, and update the play count for each playback.
2.5	Session expiration ▪ Session expires after 300 seconds. ▪ Any user activity after session expiration, except for resuming/pausing playback, will result in showing the login form with a notification message.
3.5	JavaScript control ▪ Conduct form validation. ▪ Capture and process user search requests from the search bar and genre buttons. ▪ Manage music playback functions/events, including play, pause, resume, and end.
2.0	CSS styling ▪ Apply appropriate styling for the login page and the music page
-1.0	Not using index.php as the Web app main page
-4.0	Using any external JavaScript and CSS libraries

Plagiarism

Plagiarism is a very serious academic offence. Students should understand what constitutes plagiarism, the consequences of committing an offence of plagiarism, and how to avoid it. ***Please note that we may request you to explain to us how your program is functioning as well as we may also make use of software tools to detect software plagiarism.***

Using GenAI

You are allowed to use Generative AI to explore various coding techniques and examples to complete your assignment. Feel free to divide the work into subtasks and employ GenAI to generate code snippets for better understanding. If you are using GenAI, you SHOULD include a report to clearly state the use of GenAI, including:

- The GenAI models you used.
- The prompts/questions you had with the GenAI and the responses. If you have adopted those code snippets, you SHOULD clearly indicate to us.

Should we suspect plagiarism in your submission, it would be unacceptable to justify this by stating you used GenAI for coding guidance and its output merely mirrors that of other students. While GenAI can be a valuable tool for exploring different techniques, we often need to apply adjustments to the provided code snippet to tailor it to our specific tasks.