**User Guide**

**Overview**

*This guide provides a description of the technologies used to build our website, blending important features with key concepts used in our programming.*

**DATABASE**

A database consists of logically related persistent data organised by entities, fields and records in tables that a computer program can quickly retrieve.

MYSQL is a database management system with features to update data; modified and managed by executing SQL’s query language statements directly into MySQL’s command line, or using phpMyAdmin’s interface aiding simplicity by producing SQL code for creating and managing a database schema.

Our database schema consists of three tables:

* ***Song***
* ***Artist:***One record can be associated with many records in the ‘Song’ table. A one-to-many relationship.
* ***Customer:*** Users in our registration page generate data

**CREATING A TABLE**

SQL’s CREATE TABLE statement creates a new database table. Features include:

* Using appropriate **data types** and **lengths** ensuring **efficient** memory storage.
* All our fields require a value therefore ‘**NOT NULL’.**
* **Auto Increment** selects the next available number when a record is created, important for our ‘Customer’ table since this unique **Primary** **key** information is concealed from users populating this data.

**INSERTING DATA**

Importing CSV formatted files using PHPMyAdmin’sinterface is efficient for large amounts of data.

**PRIMARY KEYS**

Inform a RDBMS that specified fields could uniquely identify a single table row.

**FOREIGN KEYS**

Only need to be assigned when two or more tables are to be joined.  Applied to the many-side of the relationship to guarantee query results from both tables contain correct data; we assigned a foreign key to our Song table.

**INDEXING**

Used to speed search operations and provide linkage for foreign key constraints.

**FOREIGN KEY CONSTRAINTS**

Created on foreign key ‘Artist ID’ to prevent database users uploading invalid content. We point the ‘Song table’ foreign key to the ‘Artist table’ where it exists as a primary key.

**JOINING TABLES**

Allows a single database to be spread across several tables offering **redundancy.** Combining tables using an SQL **SELECT** query with a **LEFT JOIN** and linking the primary and foreign keys displays all the records in the left table; and matched records from the right table where the values are the same in both.

**<pre>***FROM song LEFT JOIN artist*

*ON song.artist\_ID = artist.artist\_ID***</pre>**

--------------------------------------------------------------------------------------

*Bringing SQL and PHP together we look at how we connect to a database to send/retrieve data from an HTML page and describing key concepts.*

**PHP**

‘PHP Hypertext Preprocessor’: embeds into HTML code; adds, deletes, modifies, and retrieves data in a database; and displays the content on a webpage.

HTML, CSS and JavaScript create ***static*** web pages which always produce the ***same*** content.

The server presents their files to the browser that interprets them and displays the content. ***PHP*** contrasts as its content is produced **dynamically**; meaning the data is processed in the server and delivered to the browser as an HTML file as-and-when the browser requests the information.

**PHP INCLUDE**

Afunction useful for websites containing multiple pages with intended duplicate reusable content.  Duplicated code is replaced with include statements whilst containing the reusable content in separate PHP files.  The browser sends both PHP files to the server where they are combined into single file.  Consequently enabling amendments to be performed from one file as opposed to making changes in multiple files.

**VARIABLE**

A script needs to remember specific pieces of data that is recalled. This data is stored temporarily in a variable that can vary its value each time a script is run.

**RETRIEVING DATA**

**PHP** and **MySQLi functions** enable database access and retrieval of data using functions and variables for efficiently handling lengthy code.  For example the **define()** function contains the database security whilst the variable $r carries the result of the database connection with the returning SELECT query values obtained from the database.

**<pre>$r = @mysqli\_query ($dbc, $q);</pre>**

The two-dimensional result is converted to an **associative array** with **named key/value pairs** to enable PHP to understand and display the information. Each arrayrepresents a database row ideal for keeping data together without having to name each item as an individual variable.

**<pre>mysqli\_fetch\_array($r, MYSQLI\_ASSOC))</pre>**

**WHILE LOOP DISPLAYS RESULT**

A **while loop** sets a condition, executing a **statement** should the condition be true. The condition is checked and repeated until the condition returns **false**.  Consequently each fetched row/item is accessed one-by-one displaying the **key’s value** until there are no more items to display.

**ADDING DATA TO A DATABASE**

Users add their credentials to our database using the**HTML Form window** from our Registration page.  On clicking a **submit** button the form-data is sent for ***processing*** to a PHP file via the ***POST method***.

POST method transfers collected form-data to **$\_POST** where items are stored as an associative array in key/value pairs enabling the database to accept the two-dimensional data.  Additionally, the web page displays a message indicating a success or failure in connecting to the database to store the customer’s details as persistent data.

--------------------------------------------------------------------------------------

*This section brings together AppLab and the fundamentals of JavaScript used to code the embedded applications within our web pages*.

**JAVASCRIPT WITH APPLAB**

Code.org provide a visual web-based programming-learning environment named AppLab as a tool for designing applications with coding blocks or the underlying JavaScript text code. App data is exchanged using JSON format and apps can be shared with web pages by embedding the link to the JavaScript code within HTML.

Common JavaScript concepts used in our Playlist and Poll app:

* **Indexed array**: Zero-based ordered collection of values wrapped-up in one named variable that can be accessed by referring to its index number**.**
* **OnEvent**: Interactions create events and events trigger code; our events are clicked buttons.
* **Function:** Run a segment of code whenever the function is called.
* **join():**  A method we use to join all array items into a string and display the text.
* **.length:** Returns the length of an array.
* **randomNumber():**  Returns a random number from a closed range.
* **setText():** Displays text indicated in the parentheses.
* **If / else:** A Boolean true condition executes the IF statement, otherwise executes the ELSE statement
* **readRecords:** Reads records that match specified terms returning as an indexed array to the callback function.
* **updateRecord:** Updates records in the data storage table.
* **drawChartFromRecords:** Displays a chart retrieving persistent data from the specified data table.

JavaScript encompasses many of the classical rules of programming allowing you to create useful functionality to make web pages more interactive with its case sensitive script for the browser to react in response to user behaviour.