

Table of Contents

[Test Number 1 – Search User Input 2](#_Toc2090507)

[Test Number 2 - Search User Input (retest) 3](#_Toc2090508)

[Test number 3 – Get all songs from Database 5](#_Toc2090509)

[Test number 4 – Get all playlists from Database 7](#_Toc2090510)

[Test number 5 – Get all playlists from Database (retest) 8](#_Toc2090511)

[Test Number 6 – Create playlist 9](#_Toc2090512)

[Test Number 7 – Create playlist (retest) 10](#_Toc2090513)

[Test Number 8 – Add a song to current playlist 12](#_Toc2090514)

[Test Number 9 – Music controls 15](#_Toc2090515)

[Test Number 10 – Shuffle 16](#_Toc2090516)

[Test Number 11 – Idle Mode 16](#_Toc2090517)

## Test Number 1 – Search User Input

Test description

Testing the search input bar will produce a call to the back-end server so the data base can be search based on user input.

Test Tools

UAT testing

Console log

Input - “ed”

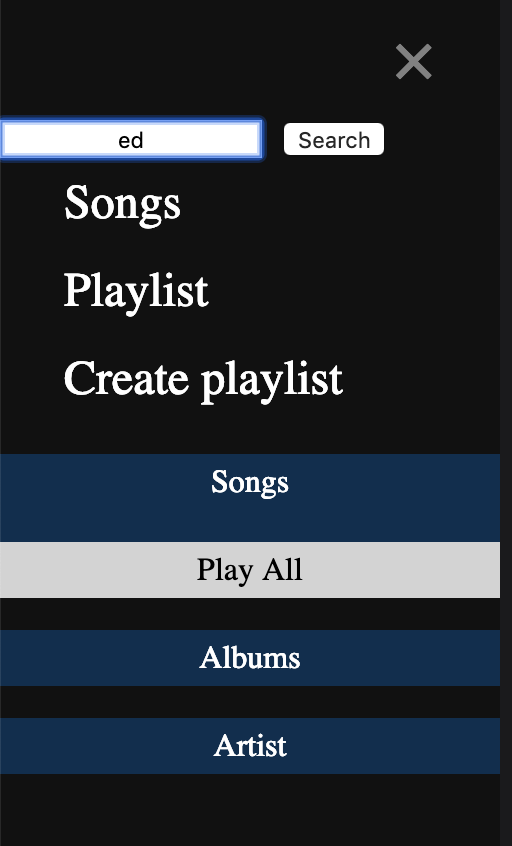
Expected outcome

Expected outcome was a list of songs by artists named ed.

Results of test

No results.

As the image below shows text input from front-end returns 0 results from the data base.



Test outcome – Failed.

## Test Number 2 - Search User Input (retest)

Test description

Repeat of test 1

Test Tools

UAT testing

SQLITE db browser

Console log

Input - “ed”

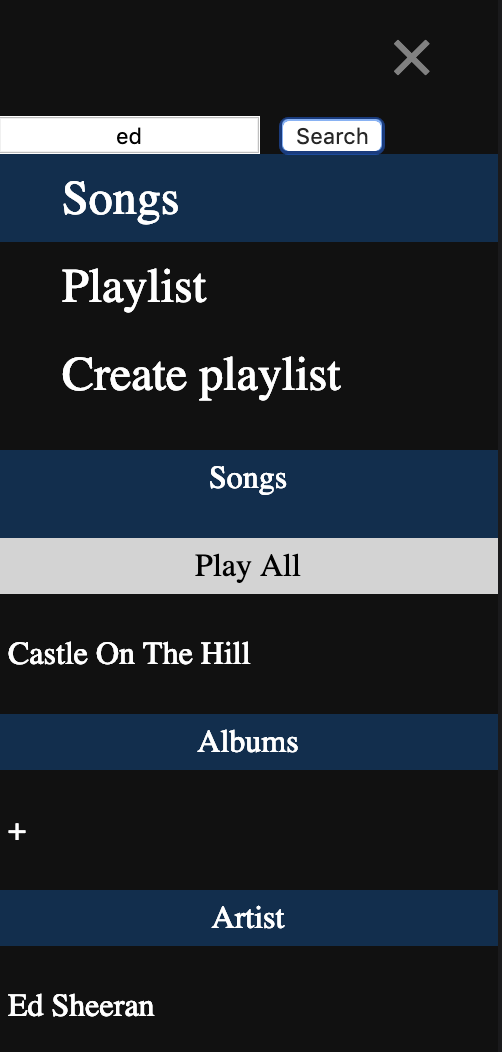
Expected outcome

Expected outcome was a list of songs by artists named ed.

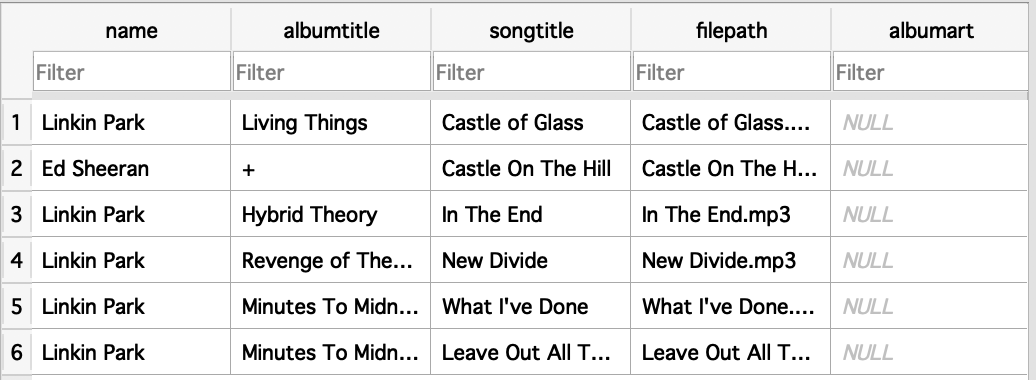
Results of test

Returned one song and album liked to song

As you can see from the image below front-end search returns result from database



This is the current view created in the SQLITE db you can see that the only result for “ed” would be the single Ed Sheeran song in the data base.



Test Outcome - Pass

## Test number 3 – Get all songs from Database

Test description

Return a list of all songs from the database

Test Tools

UAT testing

SQLITE db browser

Console log

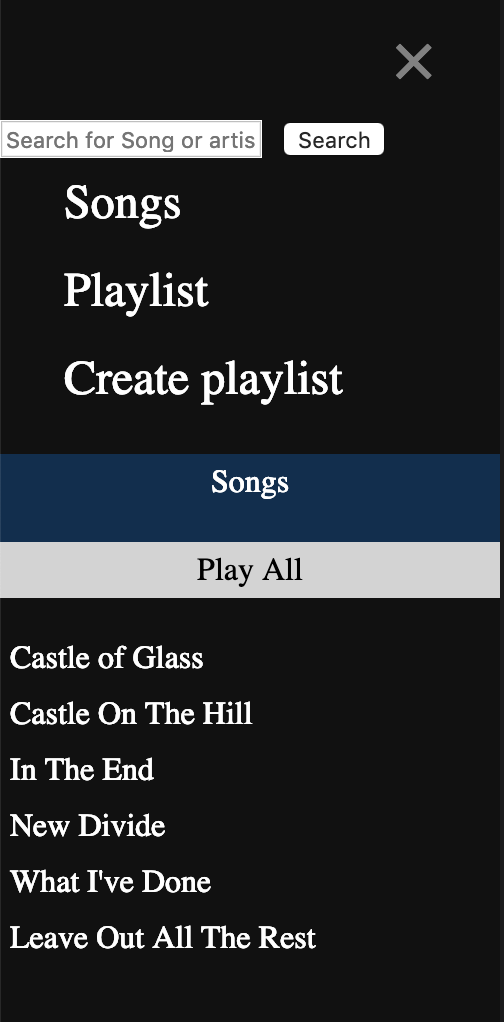
Input - songs button click in side nav

Expected outcome

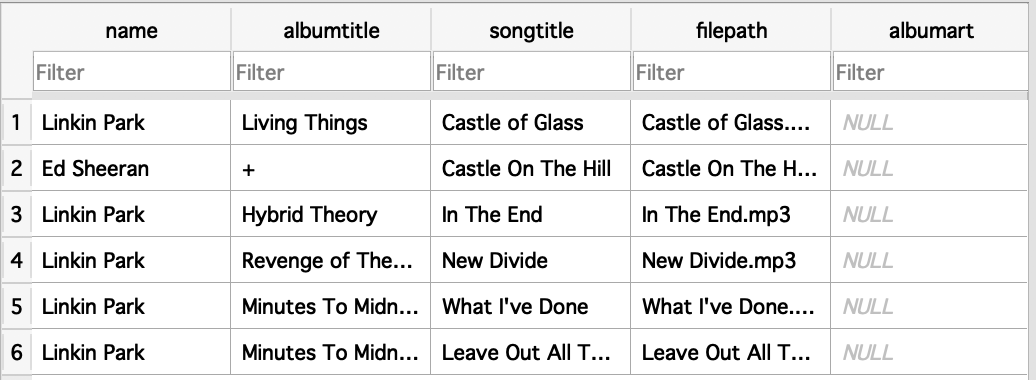
To return and display a full list of songs from the data base

Results of test

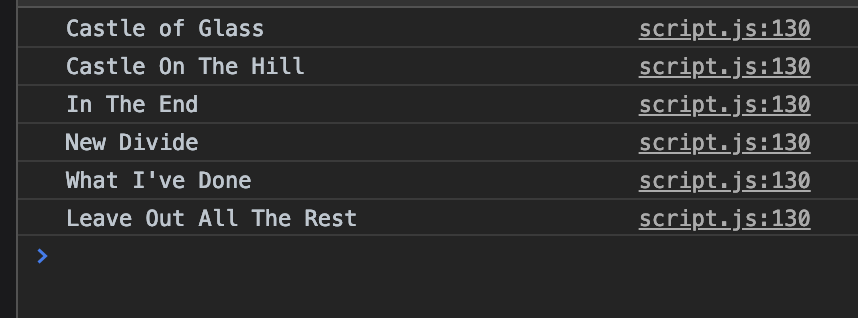
From the image below, you can see that when the songs button is triggered it will display a full list of songs that are currently in the database.



Below again is the image of the view used on the db as you can see in the third Colum all the songs titles that exist on the data base.



Here is a copy of the console log of each song that is applied to the section after being retrieved for the db.



Test outcome – Pass

## Test number 4 – Get all playlists from Database

Test description

Return all playlist from the database and display on the front-end playlist section

Test Tools

UAT testing

SQLITE DB Browser

Console log / Server Side

Input - Playlist button click

Expected outcome

Display all playlists currently in the db

Results of test

Playlist section displays no results.

Test outcome – Fail

## Test number 5 – Get all playlists from Database (retest)

Test description

Retest of test 4

This time I repositioned the database call to look at just the playlist table and not to pull the data for this section from the cross-reference table.

Test Tools

UAT testing

SQLITE DB Browser

Console log / Server Side

Input - Playlist button click

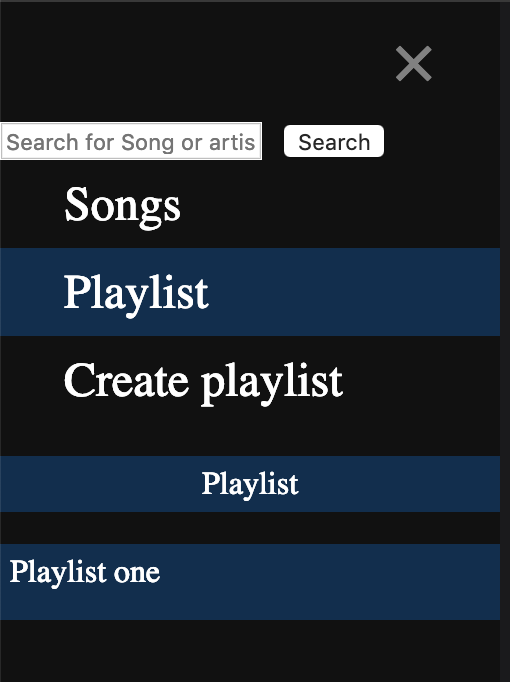
Expected outcome

Display all playlists currently in the db

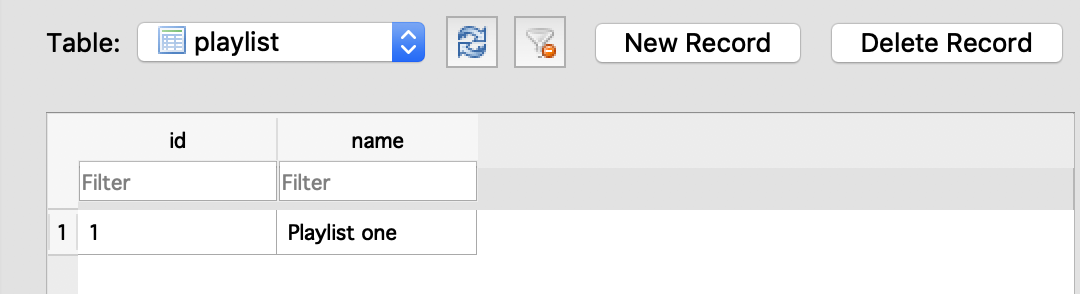
Results of test

Playlist section shows the one playlist currently in the db

As you can see from the image below it has returned the current playlist in the db



Here is the data that the playlist is currently being pulled from.



Test outcome – Pass

## Test Number 6 – Create playlist

Test description

Create a new playlist

Test Tools

UAT testing

SQLITE DB Browser

Console log / Server Side

Input – create playlist button click – Playlist name

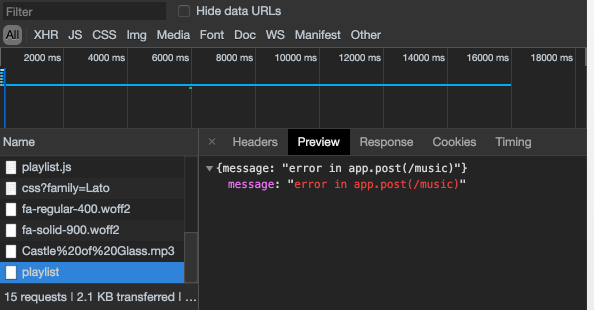
Expected outcome

To Create and push a new playlist name to the db

Results of test

Error from inserting

As you can see from the image below there is an error with the insert statement posting the data back to the db.



Test outcome – Fail

## Test Number 7 – Create playlist (retest)

Test description

Create a new playlist

Test Tools

UAT testing

SQLITE DB Browser

Console log / Server Side

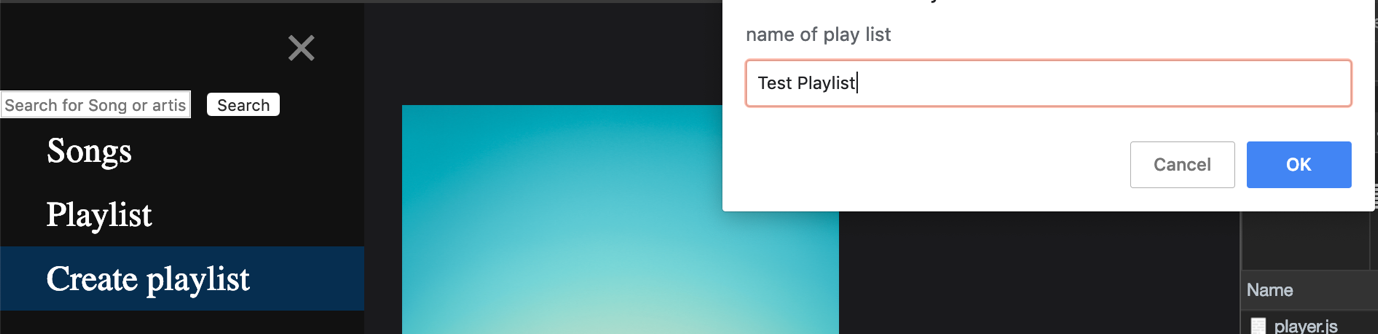
Input – create playlist button click – Playlist name

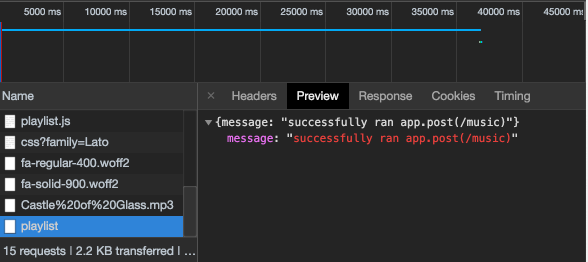
Expected outcome

To Create and push a new playlist name to the db

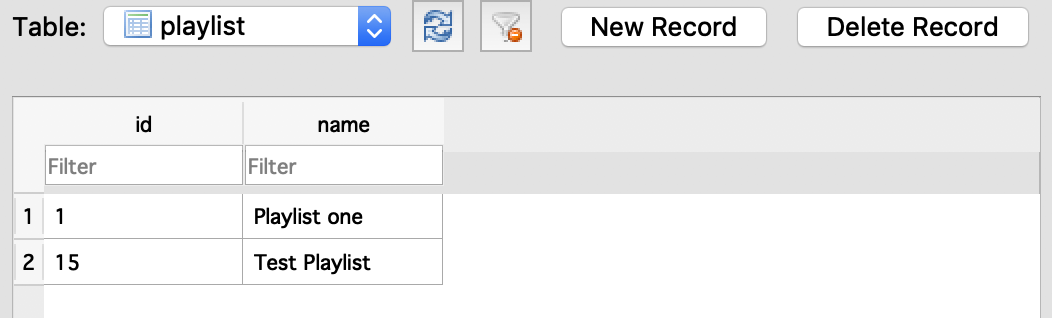
Results of test

Successfully created a new playlist and posted it to the db.





As you can see the new playlist is currently included in the db



Test outcome – Pass

## Test Number 8 – Add a song to current playlist

Test description

Add a song to the current playlist

Test Tools

UAT testing

SQLITE DB Browser

Console log / Server Side

Input – Click on song to add to playlist

Expected outcome

To push have the selected song added to the playlist

Results of test

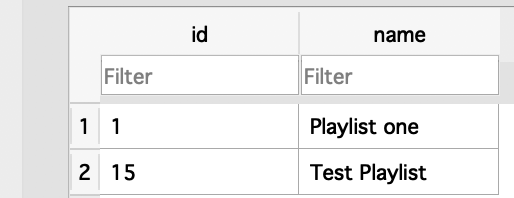
Successfully added song to playlist

Adding song Castle on the hill to test Playlist

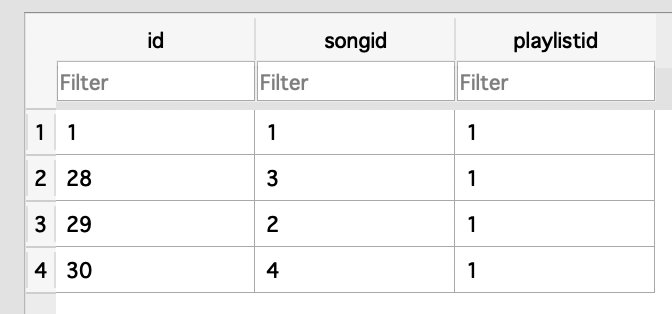


Currently playlist “test playlist” has no songs,

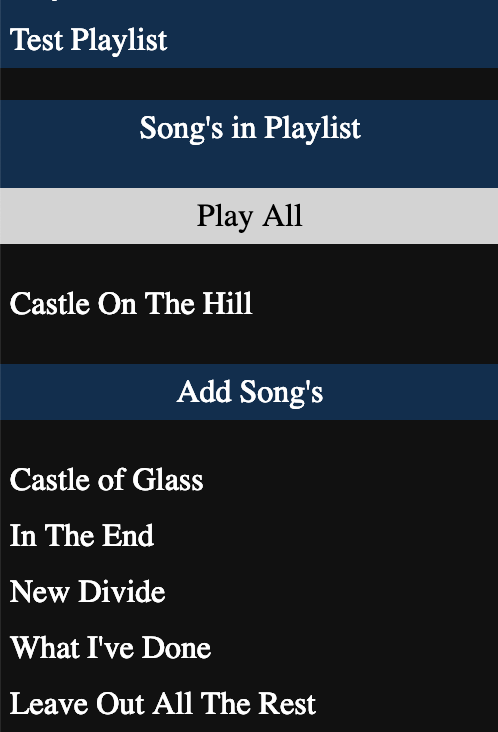
“Test Playlist” id number is 15

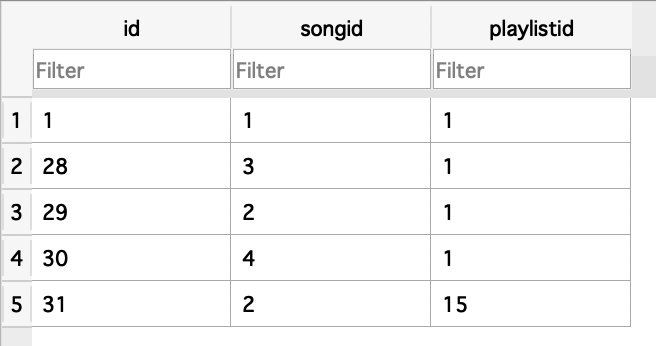


When checked against the playlistxref table you can see that no song id’s are allocated to playlist id 15



After clicking on the song to add you can see that it has been added to the playlist and is now represented in the database with its playlist id.





Test outcome – Pass

## Test Number 9 – Music controls

Test description

Test of all music controls play/pause, next track and previous track.

Test Tools

UAT testing

Input – click ok play/pause, clock next and previous track buttons

Expected outcome

Play/pause – to play track, pause track and resume playing again.

Next track – to play the next track in the playlist

Previous track - to play the previous track in the playlist

Results of test

Play/pause – successfully plays and pauses a track and will resume from pause position

Next track – when a playlist or all songs are playing will move to the next track in the list

Previous track – when a playlist or all songs are playing will go back to a pervious track in the list.

Test outcome – Pass

## Test Number 10 – Shuffle

Test description

Randomised the play order of a current playlist

Test Tools

UAT testing

Input – click ok shuffle button

Expected outcome

Playlist should order should be randomised

Results of test

When playing a playlist or all songs and the shuffle button is active the playlist order is randomised.

Test outcome – Pass

## Test Number 11 – Idle Mode

Test description

To see if the playback of a track is still active when device is in idle mode

Test Tools

Dev Machine

Input – idle mode set to 60 seconds (Minimum allowed on the machine)

Expected outcome

Song should continue to play even when machine is in idle mode

Results of test

Track plays while the dev machine is in idle mode.

Test outcome – Pass