

Coursework 1 Report

Richard McMaster

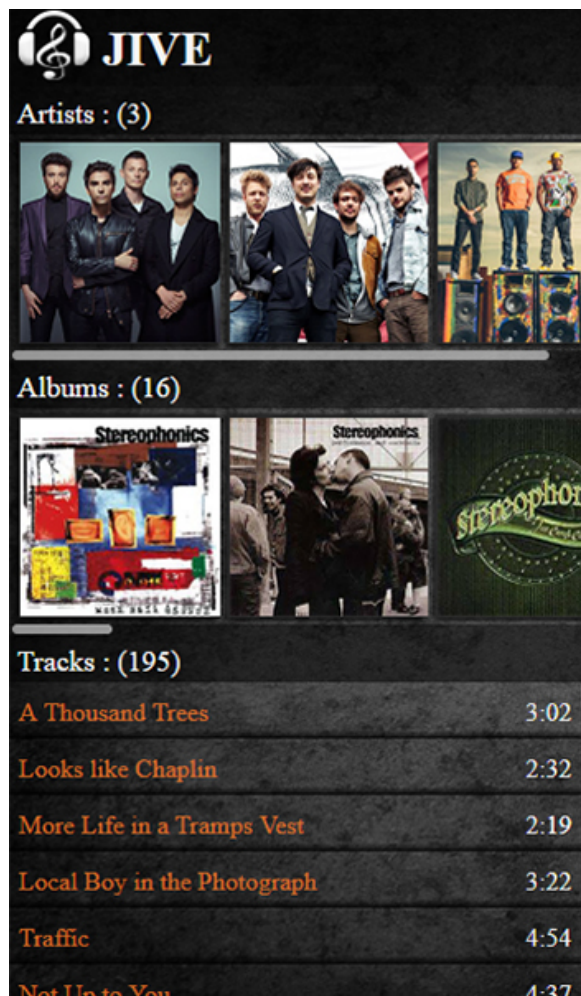
40280868@napier.ac.uk

Edinburgh Napier University

Module : SET09103

JIVE Music App

Introduction



JIVE – Home screen

JIVE is an app that could potentially, with enough data, be used to view the discography of any artist, view their albums, tracks on those albums and then view track videos along with lyrics for that track.

I have developed this by creating my own CSS, json files and using some images I downloaded then resized. Doing it the hard way, this app was almost completely developed using vim, I used adobe dreamweaver when designing certain aspects of the CSS such as gradients which you can't visualize using vim, I also used it

to create a json file for the tracks lyrics too but at least 90% of the work was using vim.

The app currently only has full track information (videos and lyrics) for the first album...
Stereophonics – Word gets around

Design

When designing the app, I had the ShowBox app in mind, specifically the icon/thumbnail approach to navigation so I attempted recreating something similar. (see appendix A.)

I thought of the logical sequence of how a user would navigate a discography which would be...

Artists → Albums → Tracks → Track Information

I also created relative json files to hold information for each part applying the following structures...

Artists...

{artist_id, artist_name, genre, image}

Albums...

{album_id, artist_id, artist, album_name, genre, released, image}

(linked back to artists through artist_id)

Tracks...

{album_id, tracks [track_no, name, length, link]}

(linked back to albums through album_id)

Track Information... (lyrics)

{album_id, track, lyrics}

(linked back to tracks through album_id and track number)

I created templates/pages for artists, albums, and tracks to follow the logical sequence of navigation. There is also a home page which technically, is a summary of all the artists, albums and tracks and a way of navigating

directly to the track player for example without having to go through the navigation sequence, saving the user a potential three clicks.

When it comes to hierarchy and the routing of the app, I think I may have made a mistake. Each template such as artist, album and tracks are routed as “/artist/” or “/tracks/” for example. I think now that I should have done it different and it should have followed hierarchy and it should have been “artist/album/tracks” for tracks.

Enhancements

There are a lot of enhancements I would like to have made given more time...

Keyword search

I originally had intended to add a search textbox in right hand side of the header. The search would not only search artist, album and tracks names but it could also search lyrics. This along with an autofill function on the form means the user could start typing a line out of a song and potential matches appear. The link would then take the user to that track player.

Personal Lists

By the use of logins and cookies, it could be developed to allow users to create lists specific to them such as personal playlists so when the user returned, they can continue their playlist or see a list of their favourite albums.

Preferred Track versions

As each track video is linked to youtube through an embedded url... (*see appendix C*)

There is potential to let signed in users see other versions of the track on youtube and set another one they prefer. This might be quite useful to some users maybe like myself who prefer some live versions of certain tracks. I'm not sure how I would approach this if I was to

develop it but off the top of my head, I think another json file with the structure...

preferred_versions.json...

```
{user_id,[album_id, track_id,pREFERRED_url]}
```

This way, as the track player was opened, the app could check if there is a result in preferred_version based on the album_id and track_id. If there is no result, show the default url path but if there is, switch out the url path for the preferred one in the result and this would work.

Visual appearance

I would liked to have had more time to improve the visual design of the app. Some elements need to be adjusted such as the sub-header on the page has to definition from the main header. The main header should be prominent and visually stand out from everything else. This could be achieved by applying a box-shadow, making the element darker or a slightly different gradient in CSS.

In this case, each element has the color black but the alpha value is set to a different value depending on what it is. In this case, `rgba(0,0,0,0.80)`. If the sub-header was dropped to 0.60, this would lighten it up and define it from the main header and give it the correct appearance in my opinion.

The logo would also need to be changed, this was quickly put together with the intention of refining or replacing but ran out of time.

Critical Evaluation

Display/CSS

The pages that only display thumbnails are fine, they look well on mobile, tablet and desktop devices. It is mainly the tracks page. In mobile, if there is a track selected, it simply swaps out the top half from the album cover to the video.

It also swaps out the bottom half from the track list to the lyrics (*see appendix B and C*).

On mobile this works well but on tablets and more desktops, this doesn't look good at all. Increasing font sizes and div elements can only go so far when you there is limited information to show. My solution would be to display both the track list and the track player on the larger resolutions side by side to fill up the space. It would also enhance the user experience on larger screens as the user could go to the next track on the album without having to go back to the track list screen.

JSON files

In some ways the way the json files are structured make sense but doing it again, I think I could have combined the albums and tracks json files into one, possibly with artist details too using nested arrays more. If I did do that then it would make sense to go all the way, include the lyrics and hold an individual json file per artist.

Post methods

To pass values between pages, it has been done by passing url values, `artist=1` for example. This should be done by a form post method to protect/hide data and is something I would change. The pages also rely on there being a value passed and I never got around to putting in an if statement for it. If the artists page is requested but has no `artist_id`, it should display all artists.

Good points

I think it looks good and mainly just using tints of black, it has turned out well visually. I believe, given a few weeks, a few tweaks to CSS and enough data behind it, it could potentially be good enough to be launched.

Personal Evaluation

Before this project I had never seen vim or python so the first few days were really frustrating for me, it still is. I was trying work out how it all worked, remembering shell commands and thinking I knew what I was doing only to have to start again. This was my own fault as I dived straight into the project head first without reading on further into the workbook. I spent a long time trying to work out how you included custom CSS files, trying everything I could think of to find out in the end, it had to be in a folder called "static". Not having much experience with git repositories, it took me a little time to work out how that worked but I understand it a lot more now.

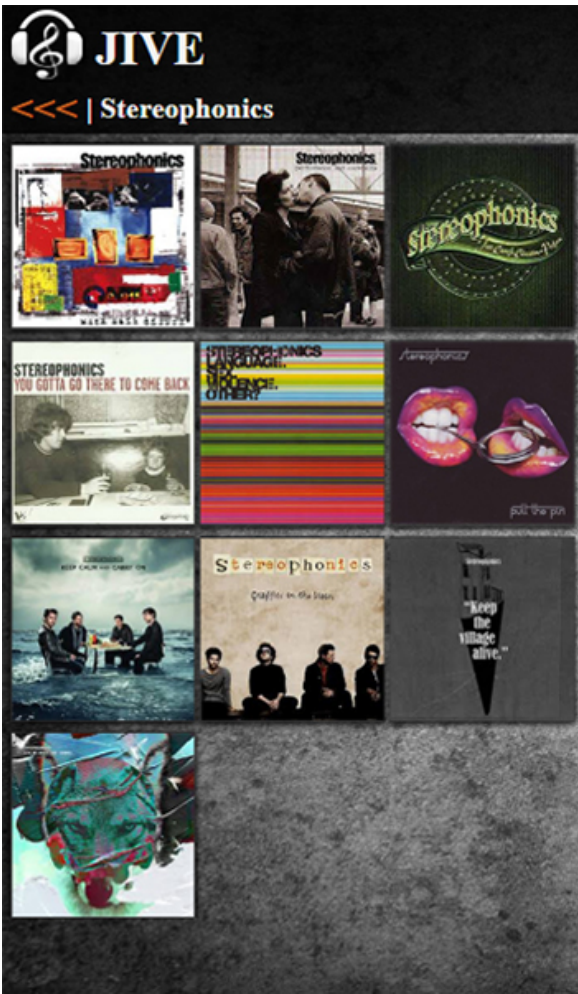
I think on this project I felt the pressure of the deadline more as I didn't know anything about python or vim. This resulted in the app being developed more out of panic than a planned approach hence issues such as app routing and possibly json structures. Confident with jQuery/AJAX and PHP, I constantly was thinking about how I would do it with them and went to google to find out how if statements and for loops were written in python.

At the start I couldn't get my head around it but now that I have worked on this project, I understand it a bit more but I'm not as confident as I am with javascript yet. I was surprised at how effective some simple functions could be in comparison to using JQuery/AJAX and PHP, specifically retrieving and working with data. There is a lot less coding to do the same thing in comparison and I think once I am confident with python, I may prefer it to javascript in the future.

In the end, this last week or so has been hard with a few late nights, but I do feel I have learnt a lot and I am looking forward to the next project with more optimism than I had for this one.

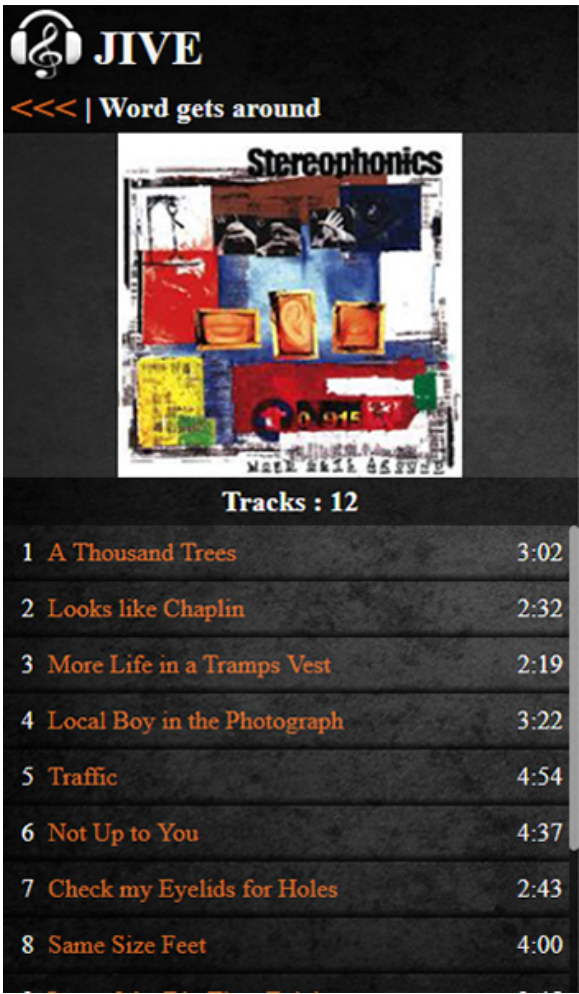
Appendices

A.



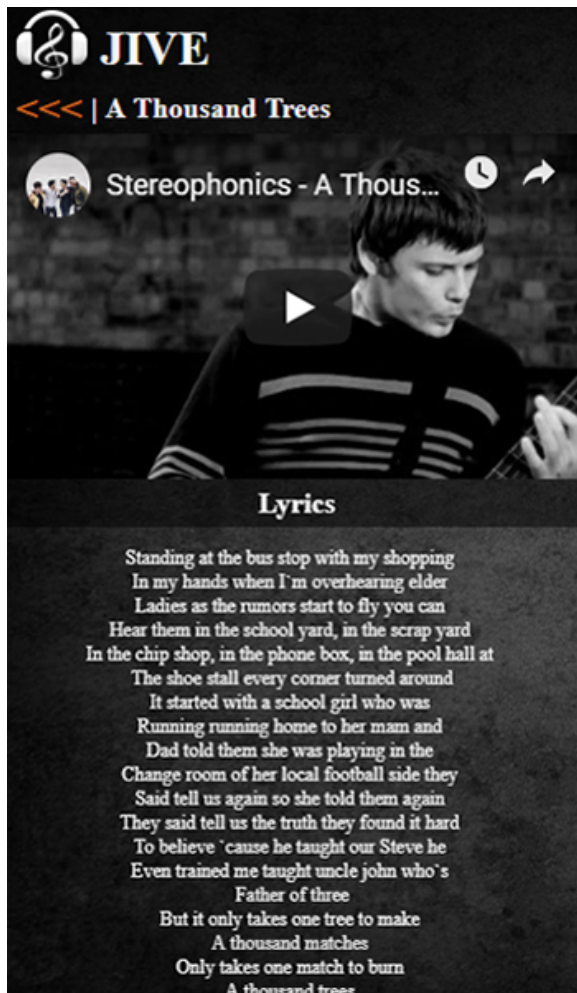
JIVE – Albums page

B.



JIVE – Tracks page

C.



The screenshot shows a music player interface. At the top left is a logo with a treble clef and the word "JIVE". Below it is a navigation bar with three orange arrows pointing left and the text "A Thousand Trees". The main area features a black and white photo of a man playing a guitar, with a large play button in the center. Above the photo is a circular profile picture and the text "Stereophonics - A Thous...". To the right of the photo are icons for a clock and a share button. Below the photo is a section titled "Lyrics" containing the following text:

Standing at the bus stop with my shopping
In my hands when I'm overhearing elder
Ladies as the rumors start to fly you can
Hear them in the school yard, in the scrap yard
In the chip shop, in the phone box, in the pool hall at
The shoe stall every corner turned around
It started with a school girl who was
Running running home to her mazz and
Dad told them she was playing in the
Change room of her local football side they
Said tell us again so she told them again
They said tell us the truth they found it hard
To believe 'cause he taught our Steve he
Even trained me taught uncle john who's
Father of three
But it only takes one tree to make
A thousand matches
Only takes one match to burn
A thousand trees

JIVE – Track player