

## Match Melody - Group Portfolio

Emilia Skoins, Yizhou Zhang, Tianxin Wang, Xiangyu Sha, Rebecca Wright, Cameron McCurdy

## Report

### Introduction

The purpose of this report is to present a synopsis of the entire project experience. The report consists of the following sections: detailing the roles and responsibilities of each team member, an application overview, achievements and the evaluation of our software system. It will also highlight future potential developments and our commitment to the professional BSC standards. To begin with, a general overview of the website is provided.

'Match Melody' is a music discovery website that allows users to listen to new music according to their personal preferences. The website uses a combination of client-side languages to implement a simple yet appealing user interface. Once users log in with their existing Spotify account, the website utilises the Spotify API to retrieve the users top tracks from their music library and provides personalised song recommendations, based on certain track characteristics. The website also offers interactivity through function keys such as play/pause button, skip button and the heart-like button, which enables users to add currently playing songs to their Spotify liked playlist.

### Members and Roles

We will discuss each team members participation to the overall software project:

**Cameron McCurdy** - Cameron has previous experience in front-end website development and his familiarity with HTML, CSS, and JavaScript was an essential base for our project. He undertook most of the front-end development for the website and created the overall user interface. He was also involved in the creation of the music recommendation algorithm.

**Emilia Skoins** - Emilia worked in close collaboration with Cameron for the user interface design. She also played a crucial role in recording the video presentation for our software demonstration.

**Tianxin Wang** - Tianxin has previous website development experience and is familiar with database building and management. She was responsible for implementing the Spotify API for accessing the Spotify user database and retrieving user data. She was also involved in developing some of the back-end functionality for the website.

**Yizhou Zhang** - Yizhou is very knowledgeable about back-end development for websites. She contributed to the project by implementing website functions, such as making the website buttons clickable and ensuring that they redirect to the correct webpages. She also worked closely with Tianxin for the Spotify API and user database.

**Rebecca Wright** - Rebecca took a leadership role throughout the project and ensured to guide the team through each stage of the production process. She was highly involved with the documentation of the project and took on most of the planning and organisation. She was also involved with the music recommendation algorithm and video presentation.

**Xiangyu Sha** - Xiangyu was involved in both the front-end and back-end development of the website. Her main contributions include enabling the user to log in to the 'Match Melody' website with their Spotify account and developing some algorithms. She also assisted Cameron with the website deployment, to connect our website to the departmental server.

The specific roles and tasks performed by each team member, in different stages of the project, are shown in the following table (sorted by first name initial):

Task	Major Roles	Minor Roles
Decide on a project	Everyone	
Discuss site features and research	Everyone	
Requirement Analysis documentation	Everyone	
Front end - basic UI structure	Cameron	Emilia
Research on Spotify API	Rebecca, Tianxin, Yizhou	Xiangyu
Get tokens and JSON data of Spotify API	Tianxin, Yizhou	-
Front end - music player page	Cameron	Yizhou
Website deployment	Cameron, Xiangyu	-
Connect Spotify API to the website	Everyone	
Login with Spotify API	Xiangyu	Emilia, Rebecca
Back end - like button implementation + insert songs	Tianxin, Yizhou	-
Back end - random songs algorithm	Tianxin, Xiangyu, Yizhou	-
Front end - social page	Cameron	-
Back end - recommendation algorithm	Cameron, Rebecca	-
Testing	Xiangyu, Yizhou	Remaining team members
Complete final website	Everyone	
Slides and user manuals	Rebecca	-
Video presentation	Emilia, Rebecca	Remaining team members
Design documentation	Everyone	
Team portfolio documentation	Everyone	

Note that the major role column indicates the primary responsibility of that team member for a particular task, while the minor role column indicates any additional responsibilities or contributions that other team members made. As we had overlapping responsibilities in the project, introducing roles in this way can help to clarify the level of responsibility for each team member in each task.

The initial design and preparatory work was discussed and completed by all team members. Apart from those documentation tasks, the team was split up into sub-teams in different phases of the project to make the workflow more efficient.

During the front-end development phase, Cameron and Emilia worked closely together on the basic UI structure, and Cameron and Yizhou worked together on the music player page. For the Spotify API calls, Tianxin and Yizhou formed a sub-team to gain access to the Spotify API and enable users to add songs to their Spotify liked playlist. In the back-end development phase, Tianxin, Xiangyu and Yizhou collaborated on back-end functions such as the heart-like button implementation and random songs algorithm. Cameron and Rebecca worked together on the creation of the music recommendation algorithm. Finally, Emilia and Rebecca collaborated on the video presentation for the demonstration. Overall, the team communicated effectively and worked collaboratively to ensure the success of the project.

## **Application Overview**

Our website's aim is to bring people together over a shared interest in music. By utilising the music recommendation algorithm created, users can discover new songs catered to their personal tastes and introduce themselves to new genres and artists.

The intended users of our system are young people with a passion for music. This user model is based on our own team members, who have been brought together by our shared interest in music and talking about our favourite songs and artists. However, we believe that our user base can expand to people of all ages as our music recommendation algorithm can be tailored for all music tastes and the website is simple to use for anyone. To access the 'Match Melody' music player and have full access to the website's functions, any user must log in with a verified Spotify account. This could deter some potential users from our website as they may not have a Spotify account and may not want to create one especially.

For the reasons of use, a user may use 'Match Melody' other over music recommendation programs as they feel that our website's song recommendations are more personal to their tastes. Also, the website is easily accessible and has a clear and comprehensive layout, making the website appealing to people of all age ranges and needs.

## Achievements

The team members worked diligently throughout the project to ensure that the requirements and specifications decided were met. A few alterations had to be made along the way however, the team are still very satisfied with the outcome of the final website created.

The biggest change from our initial plan was that we were unable to provide full functionality to the social media page of the website. This was mainly due to the time constraints that were imposed. As a group we set deadlines that preceded the determined module deadlines, to ensure that all work was completed and checked over as a group, before being submitted. We decided to prioritise the creation of the music recommendation algorithm and website music player over the social media page, which was a secondary feature. The social media page would require a larger workload than first expected and with the remaining time left after the completion of other features, the webpage and its functions were unable to be implemented in time.

On the other hand, we feel that the work that was produced compensated for these changes. The team worked hard to ensure that the overall look and functionality of the website was finished to an exceptional standard. The user interface is visually appealing but without losing any clarity or accessibility. With distinct navigation buttons and links, the website is easy to navigate around and access the features. The creation of the music recommendation algorithm was the main feature of the website, and so it was imperative that it worked as we had planned. We used the Spotify API to obtain a user's top tracks from their personal Spotify library, and from this extracted certain track attributes which were used to curate a recommendation of songs. These songs are then played through 'Match Melody's music player. As a team we feel that the creation of this algorithm is the biggest success of the project.

Another requirement that was achieved was in the testing and evaluation of the project as a whole. We used a variety of different ways to test our website and its algorithm such as compatibility testing, performance testing and functional testing. Test users were also invited to try out our website and, through a survey, give us feedback regarding their experience. We received positive feedback from this, with the majority of test users agreeing that the songs that were recommended to them through the music player matched their personal tastes. From this we can determine that we created a successful final product.

## Evaluation

The website that we created, 'Match Melody' meets most of the aims and objectives that we set out to achieve. The main functions of the website, such as logging in with a Spotify account and playing recommended songs through the music player were implemented in the final product. However, due to limited time, the feature of the social media page was not able to be implemented with full functionality.

We evaluated our website with the following strategies: accuracy of the recommendation algorithm, quality of the website, website performance, user experience and user satisfaction.

According to the evaluation results, the website is stable and fully functional on most browsers. JMeter and PageSpeed Insights were used to evaluate the website performance considering the loading time, interaction time, visual stability and response times. All performance attributes were within the normal range and according to the results of the survey, the users are mostly satisfied with the performance [1].

A/B testing was conducted with one group using the random song algorithm and the second group using the recommended music algorithm. It was concluded that our recommendation algorithm performs well at curating a personalised tracklist to each individual user.

In our preliminary design, we intended to set up our own database to store user information. However during development, we discovered that utilising the Spotify data catalogue would be more convenient and secure. Before logging in to the website, users must accept the Spotify user agreement to licence their information and data so that our website can obtain the data legally. Because of this, we can determine that using the Spotify data catalogue in our website is a safe option. Through Spotify approval, 'Match Melody' is also licensed to be able to play millions of songs without having to worry about any copyright or storage problems. Furthermore, logging in with a Spotify account saves the users from any work such as creating their liked songs list on our website all over again. For those users who do not initially have a Spotify account created, it could be inconvenient for them as they will need to create one and utilise it first before they can use 'Match Melody' effectively.

Our website was completed within the timeframe that was set. However, some tasks, such as the basic UI structure, connecting the website to the API and recommendation algorithm, were finalised later than planned which created further delays for other tasks. The team members compensated for these delays by completing further tasks in a quicker time.

All team members worked well together during the development of the website. Everyone was highly involved in the process and contributed towards the document preparation and making suggestions for improvement. Overall, the team agrees that this software engineering project gave us an opportunity to learn how to cooperate with others and practise our self-learning, problem-solving and coding abilities.

## Future Developments

There are many ways in which we could provide further enhancements to our website.

Currently, testing has taken place in all major browsers and on a range of screen sizes, to ensure our website is displayed correctly on devices such as laptops or computers. However, there is currently no support for mobile devices and the website does not display in a usable way on small screens. Adding this feature would increase the outreach of 'Match Melody', making it accessible to more users. One way we could implement this feature is with responsive web design using CSS. Media queries could be used to make changes to the appearance based on the dimensions of the browser window.

Aside from the music player feature of the website, we intended to implement a social media page allowing users to share music and make friends based on similar music tastes. Due to

time constraints, only the front end design of this feature was developed. Adding this feature would greatly increase the engagement with the website due to the increased functionality. To implement the social media page, we would use a database as discussed in the earlier stages of the project. It would require tables for storing user posts, user activity and user followers and following. Since the user has already logged in with their Spotify account, it would make sense to tie their profile on this page to their account. The team discussed the potential issue of requiring the user to agree to further terms and conditions in order to use this feature, such as collecting and storing user data.

'Match Melody' takes a 'one size fits all' approach to music recommendation. However, it may be useful for some users to have finer control over the parameters used in the algorithm. The website could feature an advanced mode for the music player, where the user would be able to adjust the priority of the parameters used, as well as adjust the amount of randomness added. More randomness within the parameters could allow for more variation from the user's typical music taste, which may be a useful feature for music discovery. This feature would be turned off by default, in order to not overwhelm the user with options. This would maintain the intended approach of the application to provide a simple and intuitive user experience.

The recommendation system currently uses content-based filtering, taking into account the listening history and preference of the user. One possible addition to the system would be to introduce collaborative filtering. This would allow the system to recommend the liked songs of other users with a similar music taste. Our implementation of this feature would be an optional addition, as some users may prefer to discover completely new music which nobody in their friendship circle listens to. Furthermore, this feature may work well together with the social page, as it could make friend suggestions based on users with similar music tastes. One issue with implementing this feature is that it would require a substantial user base in order to function properly. For this reason this feature would be added after the initial launch of the website.

## Professional Issues

As our degree programme is accredited by BCS, the Chartered Institute for IT, the software project we created was required to meet specific codes of conduct. We were made aware of this code of conduct from the beginning of our project, as it was discussed in the initial Requirement Analysis document. Due to this each team member could ensure that BCS practices were used throughout the building of the project. The BCS code of conduct comprises of four key principles: Public Interest, Professional Competence and Integrity, Duty to Relevant Authority and Duty to the Profession [2].

Public Interest means to work together in your team to solve problems and uphold fair and professional standards. This standard was upheld throughout our project to ensure that each team member was treated equally and any problems which occurred could be discussed and tackled together as a team. It was important to consider this standard at each stage, to ensure that no member was treated less favourably than another or had an unequal workload.

Professional Competence and Integrity means to develop your skills as you work, but not to claim a level of competence that you do not possess. Throughout this whole project each team member developed new skills in both technical and interpersonal terms. We made sure to only undertake work that we thought would be achievable and told each other honestly about our own personal skill levels.

Duty to Relevant Authority means to work with care and diligence while taking personal and professional responsibility. As a team we made sure to follow each task diligently, which was set out by the module requirements, and that all deadlines were reached in time. Due to this we could ensure that no conflict of interest would arise with a relevant authority.

Duty to Profession means to support your colleagues and uphold all BCS standards. Each member of the team was supported and their opinions respected, in order to create an encouraging environment for each member to develop and thrive. Also, as ambassadors for the University of Liverpool and the IT industry, we ensured to uphold the good reputation and standard set by the BCS so that our project could be a worthy contribution to the system.

## Bibliography

- [1] "Make your web pages fast on all devices," *PageSpeed Insights*. [Online]. Available at: <https://pagespeed.web.dev/>.
- [2] BCS (2022). BCS Code of Conduct | BCS. [online] www.bcs.org. Available at: <https://www.bcs.org/membership-and-registrations/become-a-member/bcs-code-of-conduct/>.

## **Test Documentation**

# **Contents**

**1 Test Objectives and Scope**

**2 Testing Strategy**

**3 Functional Testing**

    3.1 Test Cases

**4 API Testing**

    4.1 Test Cases

**5 Compatibility Testing**

**6 Performance Testing**

**7 Recommendation Algorithm Testing**

**8 Bug Report**

**9 Regression Test Plan**

**10 Bibliography**

## **1 Test Objectives and Scope**

**Test Objectives:** The website functions properly, the user interface is easy to use, and the recommendation algorithm is accurate.

**Scope of Testing:** Home page, navigation bar, user login, play songs, switch to next song, add the liked song, the accuracy of recommendation algorithm, website compatibility and website performance.

## **2 Testing Strategy**

**Test Types:** Functional testing, API testing, compatibility testing and performance testing

**Testing Methods:** Manual testing, automated testing and A/B testing

**Testing Environment:**

- Operating system: MacOS, Windows
- Browsers: Chrome, Safari, Microsoft Edge

## **3 Functional Testing**

### **3.1 Test Cases**

**Test Case Number:** TC1

**Test Case Name:** Homepage Display

**Test Steps:**

1. Open the homepage of the website
2. Check if the homepage is loaded correctly
3. Check that the homepage contains the correct content

**Expected Result:** The homepage loads correctly and contains the correct content

**Actual Result:** Test passed

**Test Case Number:** TC2

**Test Case Name:** Page Jump

**Test Steps:**

1. Open the homepage of the website
2. Click on any of the tabs in the navigation bar or the "Get Started" button
3. Check if the page in the selected tab is redirected to the selected page

**Expected Result:** The pages within each tab are loaded and accessed correctly and contain the corresponding content and layout

**Actual Result:** Test passed

**Test Case Number:** TC3

**Test Case Name:** User Login

**Test Steps:**

1. Access the login page
2. Enter your email address and password correctly, or use other social media software to authorise the corresponding account information
3. Click on "Login" to access the authorisation screen
4. Click on "Agree to permission" to authorise the website to access the user's information
5. Check that the user has successfully logged in and is redirected to the music player screen

**Expected Result:** User successfully logged in

**Actual Result:** Test passed

**Test Case Number:** TC4

**Test Case Name:** Displaying Song Information

**Test Steps:**

1. Go to the "player" page
2. Check that the song information on the page is complete, i.e. that it contains the duration of the song and the corresponding album artwork.

**Expected Result:** The song is displayed correctly, the information is complete and the album artwork is available on the page

**Actual Result:** Test passed

**Test Case Number:** TC5

**Test Case Name:** Play and Pause Song

**Test Steps:**

1. Access the "player" page
2. Click the play/pause button
3. Check if the song is played/paused successfully

**Expected Result:** The song is successfully played/paused

**Actual Result:** Test passed

**Test Case Number:** TC6

**Test Case Name:** Switch to the Next Song

**Test Steps:**

1. Access the "player" page
2. Click on the "arrow" button
3. Check if the song is switched to the next song

**Expected Result:** Switch to the next song information and album image, page information is complete and song function is normal

**Actual Result:** Test passed

**Test Case Number:** TC7

**Test Case Name:** "Heart" Button State Change

**Test Steps:**

1. Open the "player" page
2. Click on the "heart" button next to the picture of the song album
3. Check if the "heart" button turns into a red heart

**Expected Result:** The “player” page loads correctly and the colour of heart changes to red once clicked

**Actual Result:** Test passed

**Test Case Number:** TC8

**Test Case Name:** Song Added to User’s Liked List

**Test Steps:**

1. Click on the "heart" button on the "player" page
2. Check if the song is added to the user's Spotify liked list

**Expected Result:** By clicking on the "heart" button, the song is added to the user's Spotify liked list

**Actual Result:** Test passed

## 4 API Testing

**Test tool:** Web Console [1]

### 4.1 Test Case

**Test Case Number:** TC9

**Test Case Name:** Request an Access Token

**Test Steps:**

1. Go to the Spotify developer dashboard
2. Use developer mode, get the Client\_ID, and add redirect URI
3. Send a POST request to the token endpoint URI.

**Expected Result:** Status Code is 200, get the JSON data of the corresponding response:

```
{  
    "access_token":  
        "BQDBKJ5eo5jxbtpWjVOj7ryS84khybFpP_1TqzV7uV-T_m0cTfwvdn5BnBSKPxKgEb11",  
    "token_type": "Bearer",  
    "expires_in": 3600  
}
```

**Actual Result:** Test passed

**Test Case Number:** TC10

**Test Case Name:** Connect Web Playback SDK

**Test Steps:**

1. Embed the SDK
2. Initialise the player
3. Use the addListener method to listen and subscribe to 'ready', 'not\_ready' or some error events

**Expected Result:** The console hits "Ready with Device ID" and no error message is sent

**Actual Result:** Test passed

**Test Case Number:** TC11

**Test Case Name:** Read the User's Playlist

**Test Steps:**

1. Create a new URLSearchParams object and add the client\_ID, response\_type, redirect\_URI and scope parameters to it. The scope parameter is a list of permissions requested from the user and contains user-top-read.
2. Set the endpoint, limit, and offset parameters and use the "GET" request data

**Expected Result:** Status Code is 200, and the JSON data from the response contains: collaborative, description, external\_URLS, href, ID, images, name, owner, public, snapshot\_ID, tracks, type, URI, and the values inside the data are normal.

**Actual Result:** Test passed

**Test Case Number:** TC12

**Test Case Name:** Request Track Data

**Test Steps:**

1. Set the endpoint, corresponding to the Spotify track ID and market
2. Use the "GET" request to get Spotify catalogue information for multiple tracks based on their Spotify IDs.

**Expected Result:** Status Code is 200, the JSON data of the response contains: album, artists, available\_markets, disc\_number, duration\_ms, explicit, external\_IDs, external\_URLs, is\_playable URLs, isPlayable, linked\_from, restrictions, name, popularity, preview\_URL, track\_number, type, URI, and all values inside the data are normal.

**Actual Result:** Test passed

## 5 Compatibility Testing

**Test Environment:**

- Operating system: MacOS, Windows
- Browsers: Chrome, Safari, Microsoft Edge

**Test Steps:**

1. Use the same browser for a different operating system or a different browser for the same operating system to access the website
2. Check that the web pages function correctly and that the pages are complete

**Expected Result:** The website works correctly regardless of the operating system and browser, with no missing pages

**Actual Result:** Test passed

## 6 Performance Testing

**Testing Content:** Based on user experience and website performance, website loading time, website interaction time, website visual stability and website response time

**Testing Tool:** PageSpeed Insights website [2]

**Test Results:**

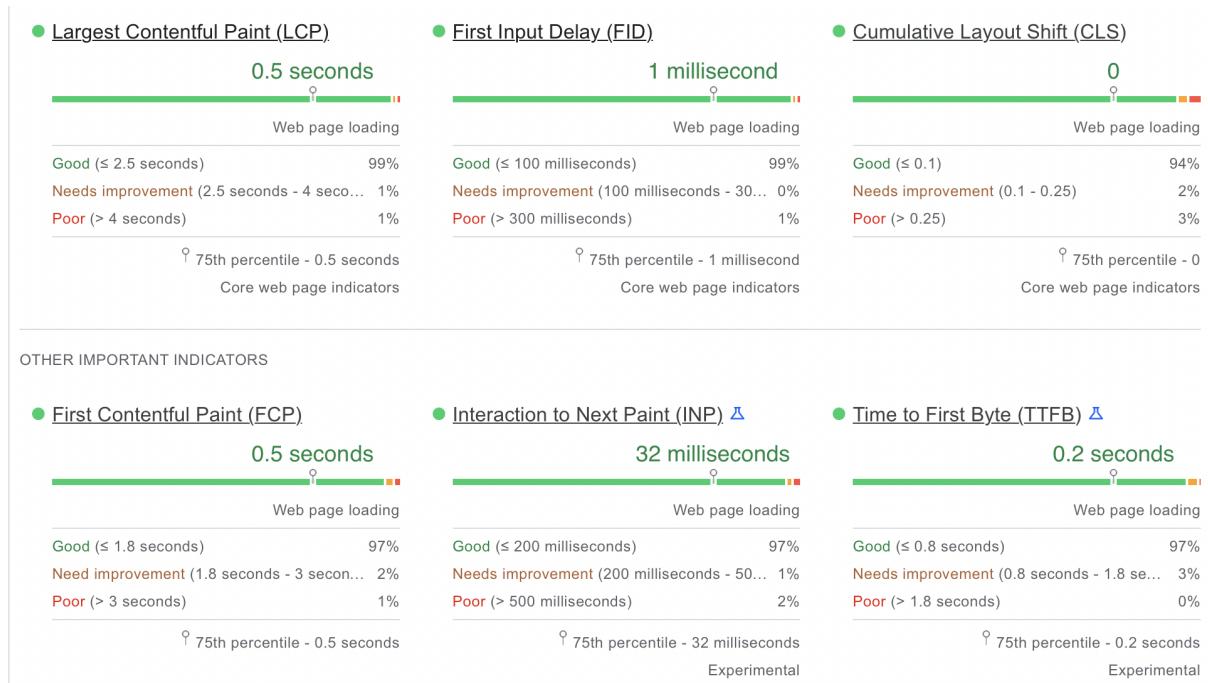


Figure 6.1 - Performance Test

In Figure 1, the LCP is the maximum content drawing time for the website to complete. Since pages are loaded in stages, the time taken for each element to complete loading is recorded and reported once a larger element has completed rendering. Generally, the time taken within 2.5 seconds is appropriate, with 2% of our elements on the website loading in excess of 2.5 seconds and most elements loading in less than 0.5 seconds.

The FID measures the time it takes for a user to first interact with our site, wait for the site to respond and actually start processing. 1% of the sections exceed 300 milliseconds, and the rest are within 100 milliseconds, with the 75th percentile at 1 millisecond which is within a good time range.

CLS is an important user-centred measure of visual stability and improved user experience. This metric is used to measure whether the content in a web page shifts abnormally, causing users to mistouch it. This accident is caused by time lags when loading resources or differences from the initial page when rendering the page. In order to provide a good user experience within the website, this should be kept below 0.1. 5% of our website is outside this range, but the overall value of 0 is in a more reasonable position.

FCP measures the time it takes for a user first to see any content on our website. This factor affects the user's experience of the website, with 3% of our website exceptions being over the recommended 1.8 seconds, possibly due to some larger CSS causing the FCP section to take longer, but overall our website is in a good position.

INP assesses responsiveness by measuring the user's interaction with different parts of the website page and the page's response to it, recording the longest interaction time. 3% of the site's page interactions exceeded a good response time, probably because our website was requesting the Spotify API via "GET" to return some response values, and the Spotify API call and response section may have been key to this 3% INP.

TTFB is the time between the first byte of the request and the first byte of the response, this value represents the response speed of the web server. The influencing factor for this value could be that the infrastructure of the hosting server is not capable enough to handle the high traffic load. In our case, this influence could be due to the fact that our site borrows the Spotify API for redirection, resulting in the 3% portion being over 0.8 seconds. It could also be due to the fact that our site uses the JQuery request API multiple times to establish and connect, resulting in a partial timeout.

## 7 Recommendation Algorithm Testing

**Test Content:** Testing the accuracy of the site's recommendation algorithm

**Testing Method:** A/B testing

**Test Objective:** To improve the accuracy of the recommendation algorithm and to increase the percentage of users who like the recommended songs

**Experiment Design:** Whether or not to use the Content-based Filtering recommendation algorithm as a variable, and set the rest as constants

**Randomised Grouping:** Ten users were randomly and equally divided into two groups, one as the experimental group, which used the Content-based Filtering recommendation algorithm, and one as the control group, which used only the randomised algorithm.

**Testing Procedure:** The JSON data from the users' top tracks was obtained via a Spotify API "GET" request. Two algorithms were used to generate 50 songs based on the user's top tracks. The experimental group used the Content-based Filtering algorithm to obtain the JSON data of the 50 songs and generate a playlist, scoring the algorithm based on the genre, artist, album and other characteristics of the songs in the generated list. Similarly, the control group used a randomised algorithm to obtain 50 songs and score them against each other. The accuracy of the recommendation algorithm was ultimately judged by the score, on a scale from 1 to 10, with higher scores indicating more accurate algorithms.

**Analysis of Data:** For genre, the average score was 6 for the experimental group and 3 for the control group. For the album, 7 was the experimental group's score and 4 was the control group's score. By comparing the relevance of the top tracks to the artist in the generated songs, the experimental group's score was 7 and the control group's score was 3.

**Conclusion:** Based on the results of the genre, album and artist scores of the songs generated using the two algorithms, it can be found that the Content-based Filtering recommendation algorithm scores are generally higher and therefore the recommendation algorithm is more accurate.

## 8 Bug Report

This bug report gives an overview of bugs identified during the testing phase of the project, detailing the information of bugs that have been fixed and not been fixed.

- B01-B08: Bugs that have all been fixed and satisfied our requirements.
- B09: The unfixed bug ('Fail to Log Out'), which can be considered as a system limitation of our project

**Bug ID:** B01

**Bug Title:** Incomplete Page Display

**Description:** The home page elements are not displayed as expected because a large icon 'Match Melody' is missing from the top left corner of the home page.

**Bug Type:** GUI

**Steps to Reproduce:**

1. Open the homepage of the website
2. Check if the homepage is loaded correctly
3. Check if the homepage contains the correct content and layout

**Expected Result:** The homepage loads correctly and contains the correct content

**Actual Result:** A large icon 'Match Melody' is missing from the top left corner of the homepage

**Priority:** Low

**Severity:** Minor

**Status:** Fixed

**Bug ID:** B02

**Bug Title:** Incompatibility Issues

**Description:** The same pages look different in different browsers and operating systems.

**Bug Type:** Compatibility

**Steps to Reproduce:**

1. Use the same browser for a different operating system or a different browser for the same operating system to access the website.
2. Check that the web pages function correctly and that the pages are complete

**Expected Result:** The website works correctly and consistently on every different operating system and browser, with a consistent UI and layout.

**Actual Result:** The original version displayed fine on Mac OS Safari, but on Chrome and Windows there was typographical overflow and the border size was not consistent.

**Priority:** High

**Severity:** Major

**Status:** Fixed

**Bug ID:** B03

**Bug Title:** Fail to Retrieve Songs from Spotify API

**Description:** The song data in the Spotify database is not called successfully. This means that when you go to the music player page, no songs are displayed or played

**Bug Type:** Functionality

**Steps to Reproduce:**

1. Open the homepage of our website
2. Click on the “Get Started” button
3. Click on the arrow button on the next page (the page which shows ‘Discovery starts here’) to go to the music player page

4. Check if the music player page displays and plays the songs from the Spotify music database

**Expected Result:** Songs are displayed and played successfully.

**Actual Result:** There are no songs displayed and played on the music player page.

**Priority:** High

**Severity:** Critical

**Status:** Fixed

**Bug ID:** B04

**Bug Title:** Fail to Switch to the Next Song

**Description:** The button on the music player page that is used to switch to the next song does not work.

**Bug Type:** Functionality

**Steps to Reproduce:**

1. Open the homepage of our website
2. Click on the “Get Started” button
3. Click on the arrow button on the next page (the page which shows ‘Discovery starts here’) to go to the music player page
4. When music is playing on the music player page, click on that arrow button on the right to switch songs
5. Check if the song has been switched to the next one

**Expected Result:** The song has been switched to the next song successfully.

**Actual Result:** The button cannot work to switch to the next one. There is no response when clicking on that button.

**Priority:** High

**Severity:** Major

**Status:** Fixed

**Bug ID:** B05

**Bug Title:** Inconsistencies Between the Song Information and the Actual Song

**Description:** The song being played does not match its album cover and the corresponding background gradient colour

**Bug Type:** Functionality and GUI

**Steps to Reproduce:**

1. Open the homepage of our website
2. Click on the "Get Started" button
3. Click on the arrow button on the next page (the page which shows 'Discovery starts here') to go to the music player page
4. When music is playing on the music player page, click on that arrow button on the right to switch songs
5. Check if the album cover and its background gradient colour displayed correspond to the right song being played when it is playing successfully as well.

**Expected Result:** The album cover and its background gradient colour displayed correspond to the right song being played.

**Actual Result:** The album cover and its background gradient colour does not match or the album cover just does not show.

**Priority:** Medium

**Severity:** Minor

**Status:** Fixed

**Bug ID:** B06

**Bug Title:** Fail to Change the "Heart" Button State

**Description:** The "Heart" button on the music player page does not change the state, which means it fails to turn the empty heart-like image into a red heart.

**Bug Type:** Functionality

**Steps to Reproduce:**

1. Open the homepage of our website
2. Click on the “Get Started” button
3. Click on the arrow button on the next page (the page which shows ‘Discovery starts here’) to go to the music player page
4. When music is playing on the music player page, click on that ‘heart’ button on the left.
5. Check if the white empty heart-like image has turned into a red heart.

**Expected Result:** The white empty heart-like image has turned into a red heart successfully.

**Actual Result:** The white empty heart-like image has not turned into a red heart. There is no response when clicking on that button.

**Priority:** Medium

**Severity:** Minor

**Status:** Fixed

**Bug ID:** B07

**Bug Title:** Fail to Add Songs to Spotify ‘Liked’ Playlist

**Description:** When the "Heart" button on the music player page changes the state to a red heart, it fails to add songs to the Spotify ‘Liked’ playlist.

**Bug Type:** Functionality

**Steps to Reproduce:**

1. Open the homepage of our website
2. Click on the “Get Started” button
3. Click on the arrow button on the next page (the page which shows ‘Discovery starts here’) to go to the music player page
4. When music is playing on the music player page, click on that ‘heart’ button on the left.
5. When the state of the ‘heart’ button has changed, check if the song has been added to the Spotify ‘Liked’ playlist.

**Expected Result:** The song has been added to the Spotify ‘Liked’ playlist successfully.

**Actual Result:** The song has not been added to the Spotify ‘Liked’ playlist.

**Priority:** High

**Severity:** Critical

**Status:** Fixed

**Bug ID:** B08

**Bug title:** Fail to Make Music Recommendations

**Description:** The songs being played on our website are not based on the user's music taste, which means we fail to make personalised music recommendations.

**Bug Type:** Functionality

**Steps to Reproduce:**

1. Open the homepage of our website
2. Click on the “Get Started” button
3. Click on the arrow button on the next page (the page which shows ‘Discovery starts here’) to go to the music player page
4. When music is playing on the music player page, click the arrow button several times.
5. Check if the songs being played are based on the user's music taste

**Expected Result:** The songs played to match the user's tastes.

**Actual Result:** The songs played do not match the user's tastes. They are still just random songs we retrieved from the Spotify database.

**Priority:** High

**Severity:** Critical

**Status:** Fixed

**Bug ID:** B09

**Bug title:** Fail to Log Out

**Description:** When users click on the logout button in the top right corner of the navigation bar, they only jump to the homepage and do not actually log the user out of their Spotify account.

**Bug Type:** Functionality

**Steps to Reproduce:**

1. Open the homepage of our website
2. Login to our site with the existing Spotify account
3. Click on the logout button in the top right corner of the navigation bar
4. Check if the user has logged out

**Expected Result:** The user has logged out successfully.

**Actual Result:** The user only jumps to the homepage and does not actually log out of their Spotify account.

**Priority:** Low

**Severity:** Low

**Status:** Unfixed

## 9 Regression Test Plan

**Test Scope:** Entire website

**Test Methods:** Manual testing and automated testing

**Test Cases:** All test cases

**Test Environment:**

- Operating system: MacOS, Windows
- Browsers: Chrome, Safari, Microsoft Edge

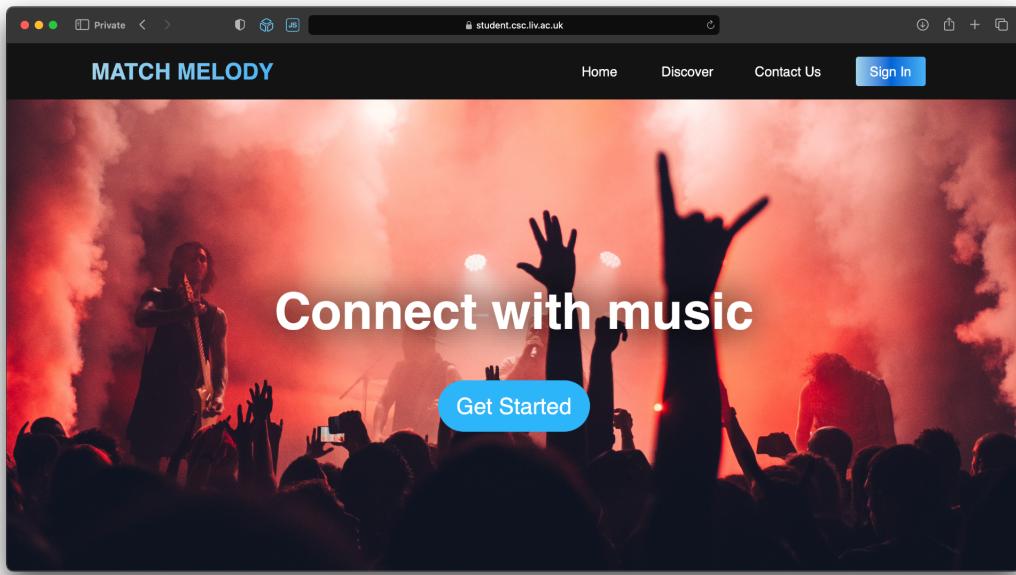
**Steps in the Regression Test Plan:**

1. Retesting against previous bug reports to ensure that issues have been resolved
2. Testing of additional functionality to ensure that the new functionality has not introduced new defects
3. Retesting of previously tested features to ensure that changes have not introduced new defects
4. Testing of performance to ensure that the site is working as expected

## 10 Bibliography

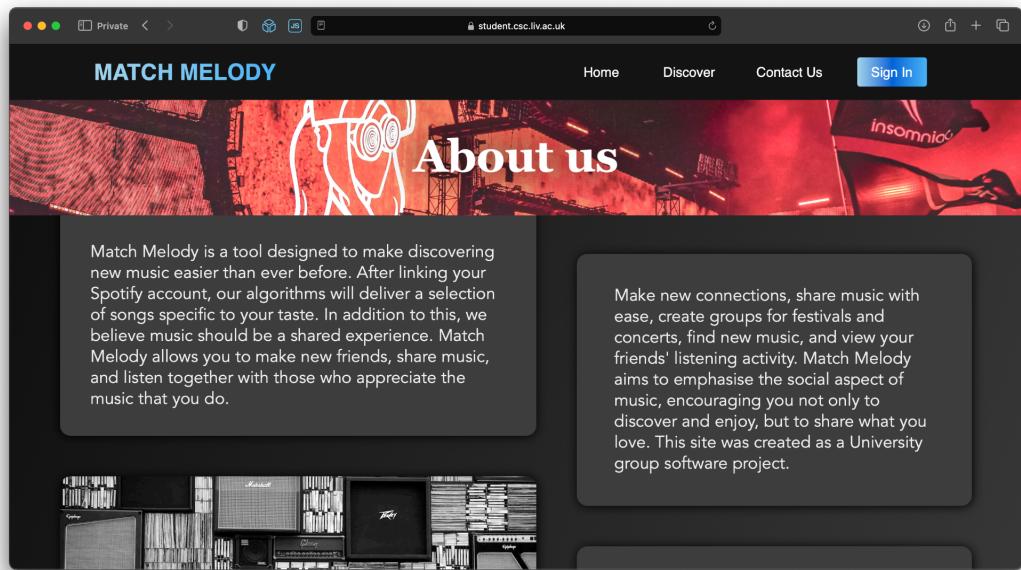
- [1] developer.spotify.com. (n.d.). Web API | Spotify for Developers. [Online] Available at: <https://developer.spotify.com/documentation/web-api/>.
- [2] "Make your web pages fast on all devices," *PageSpeed Insights*. [Online]. Available at: <https://pagespeed.web.dev/>.

### 'Match Melody' Website Screenshots



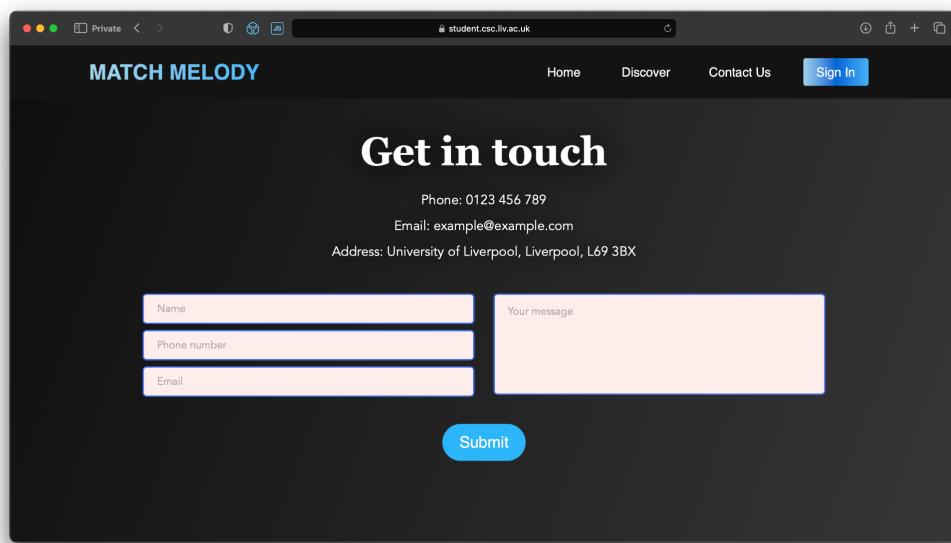
**Figure 1.1 - 'Match Melody' Homepage**

This is the homepage of our website, 'Match Melody'. As this is the first page which greets the user upon entering our website, it was important to make it visually appealing and eye-catching. To do this we used a bold, red image which pictures fans at a festival and topped this with a white text slogan, 'Connect with music'. From this page a user can sign in using the 'Get Started' or 'Sign In' button or visit the 'Discover' or 'Contact Us' pages.



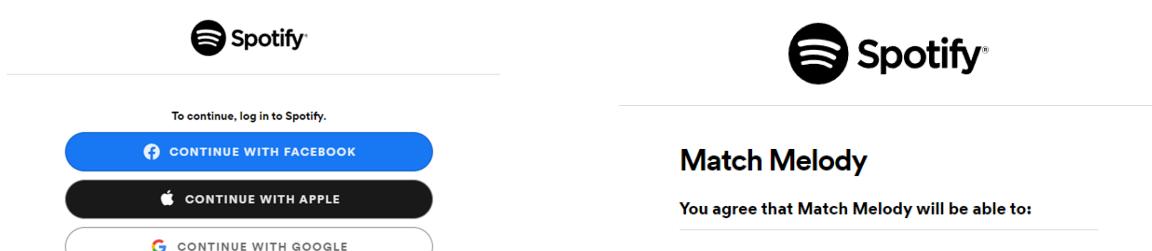
**Figure 1.2 - ‘Match Melody’ Discover Page**

This is the ‘Discover’ page of the website which documents what the website can do and how and why it was created. The user can look through this information to see if the website is suited towards them before they sign up. From this page a user can sign in using the ‘Sign In’ button or visit the ‘Home’ or ‘Contact Us’ pages.



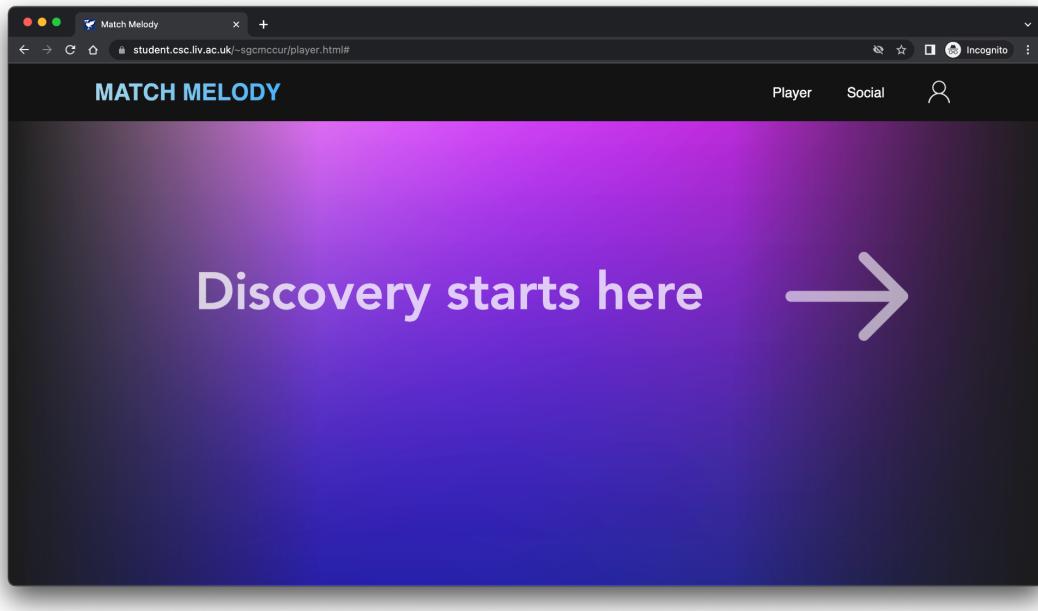
**Figure 1.3 - ‘Match Melody’ Contact Us Page**

This is the ‘Contact Us’ page of the website which contains the details of how to contact us by phone, email, address or via the online enquiry form. The user can use the information contained here to contact us with any queries or problems they may come across while using the website. From this page a user can sign in using the ‘Sign In’ button or visit the ‘Home’ or ‘Discover’ pages.



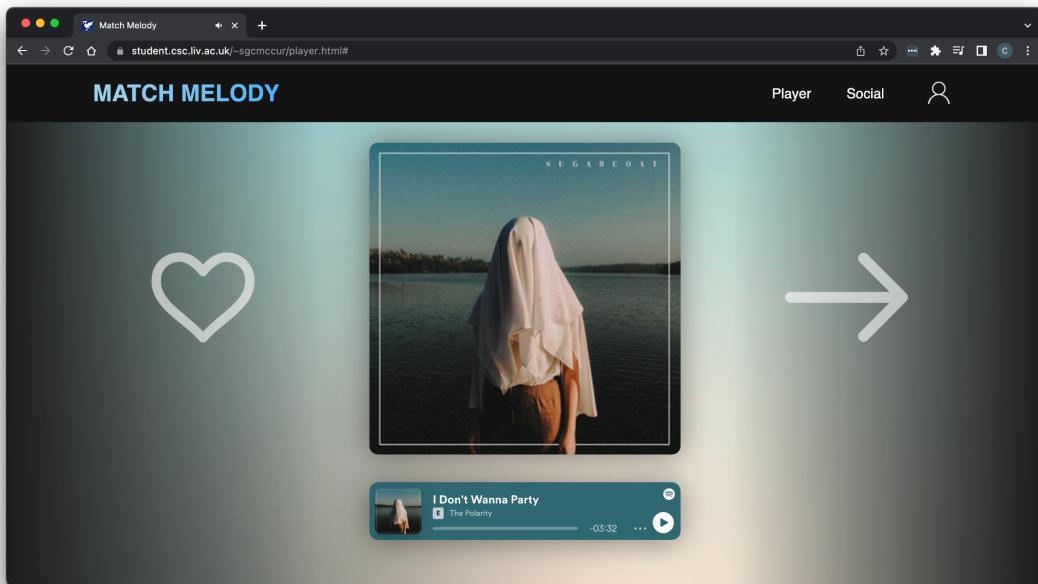
**Figure 1.4 - Spotify Login Page**

Upon clicking the ‘Sign In’ button the user will be directed to the Spotify Login page shown in Figure 1.4. A user must log in with a valid Spotify account to progress further. After successful login validation, the user must then accept certain permissions such as allowing ‘Match Melody’ to view their Spotify account data. Once agreed they can progress to the main features of the website.



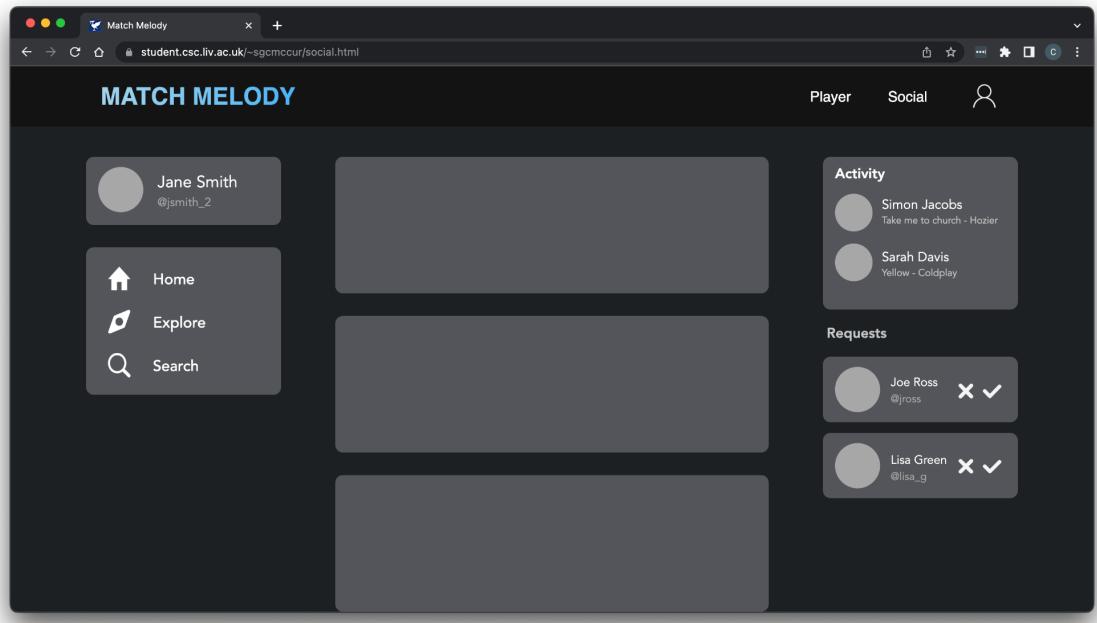
**Figure 1.6 - ‘Match Melody’ Player Page**

This is one of the main features of the ‘Match Melody’ website, the ‘Player’ page. This is the initial page which first greets the user. They can press the arrow to listen to their recommended music or click the ‘Social’ or ‘My Account’ icon.



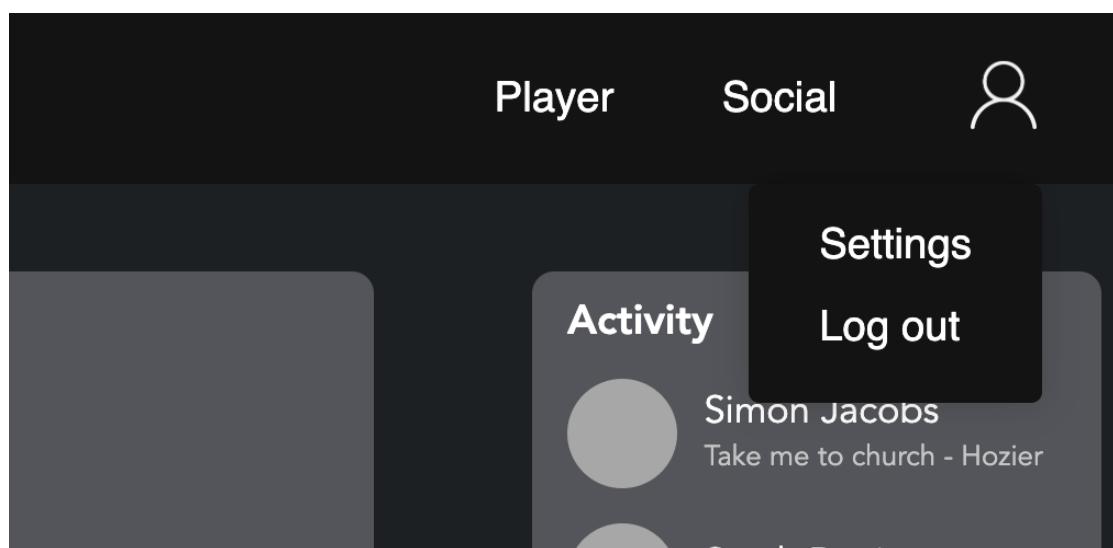
**Figure 1.7 - ‘Match Melody’ Player Page (Music Recommendation)**

This is the music player page which plays the users recommended songs based on their previous Spotify listening history and various track attributes. It is a visually appealing player which features the album artwork and a colour changing background which changes to match the mood and colour scheme of each song. The user can play or pause the song with the media player, add the current song to their 'Liked' playlist by clicking the 'Heart' button or listen to the next song by clicking the 'arrow'. From this page the user can also access the 'Social' page or 'My Account' icon.



**Figure 1.8 - 'Match Melody' Social Page**

This is the social media aspect of the website. As pictured, this part of the website was not able to be completed in time so does not have full functionality. However the initial layout of the page is displayed and we would have liked to implement functionality such as sharing music activity on the users feed and being able to add other 'Match Melody' users as friends. From this page the user can also access the 'Player' page or 'My Account' icon.



***Figure 1.9 - ‘Match Melody’ My Account Drop Down Button***

When the ‘My Account’ icon is clicked, the account settings of the website are shown. The user can access their ‘Settings’ or ‘Log Out’ of the website. When the user logs out they are redirected back to the ‘Match Melody’ homepage.