National College of Ireland

FeelGood App

1st May 2024

Computing

Digital Business Transformation

2023/2024

Jack Nicholson
x20444072
x20444072@student.ncirl.ie

FeelinG

Technical Report

Contents

Executive Summary

This technical report describes the creation and assessment of FeelinG, a programme that creates playlists for users based on their feelings. The project makes use of contemporary technologies including OpenAI's GPT-4, Spotify API, Vite, and Vue.js. The main goal was to develop a customised music experience that increases user engagement by suggesting songs that fit the user's mood at the time.

Important features include mood analysis with OpenAI, playlist creation via Spotify API, and user input for emotions. User input, performance assessments, and usage data gathering were all part of the testing process, which showed excellent user satisfaction and efficient system operation.

1.0 Introduction

1.1. Background

Since I was a kid, I have always had a great love for music. so, when Dr. Francis Sheridan , my project supervisor said that our final year project should be based on something we love, I thought something music based would be the best option. Music is especially important for people as emotions can be powerfully expressed through music. It can express a lot of different feelings, such as happiness and joy as well as sadness and grief. People frequently use music as a tool to process and communicate their emotions. Music plays a significant role in shaping cultural identities. It is often tied to specific regions, traditions, and historical contexts, and it helps preserve and transmit cultural heritage from one generation to the next.

Music is a vital part of our culture and our daily existence. In many ways it might uplift, heal, connect, entertain, and enhance our lives in so many ways.

1.2. Aims

I am going to build an application where you enter your feelings, your location and why you feel that way.

So, for example, I would enter: I'm feeling happy and motivated, working out. So, from here the application will give you a genre of music to best suit your mood. It will then say " The genre of music for you is: Motivational workout music. Why not listen to: Power by Kanye West. This helps other people broaden their taste of music and listen to genres they may have never thought of listening to before.

I will apply what I have learned throughout my academic career with things like project management, business analysis, programming & coding, use innovation and try being creative. Use my project management skills that I acquired in my academic career to conduct original research. Develop strong time management skills as I will need to juggle multiple tasks and deadlines during the project. Learn how to accurately identify and define problems or research questions, which is an important skill in not just my career but many career fields. Use my critical thinking skills by evaluating different solutions or approaches, considering the strengths and weaknesses of each, and making informed decisions.

1.3. Technology

Some of the technologies used in my project are; Vue.js, NPM, Spotify API's, Vite and Open AI.

Vue.js: A progressive JavaScript framework for creating user interfaces . Because of its incremental adoption architecture, you can utilise as much or as little of it as necessary.

Key characteristics include component-based architecture, flexibility, simplicity, and reactive data binding.

Use Case for Project: I used Vue.js to create the application's front end, which allows users to enter their emotions. Because of its reactive data binding, the user interface is guaranteed to refresh instantly in reaction to user input.

NPM is a JavaScript package manager that you may use to share, install, and maintain dependencies within your projects. Some of the packages I used are Vite (which builds the project), Pinia for global state management, and spotify-web-api

Key Features: Scripts for task automation, simple project dependency management, and access to a vast registry of JavaScript packages.

Use Case for Project: All of the JavaScript libraries and packages required for the project, such as Vue.js for HTTP request processing, and any Spotify or OpenAI SDKs, can be managed with NPM.

Developers can communicate programmatically with Spotify's music streaming service by using its APIs.

Key Features: Make and manage playlists, search for music, get details about songs, artists, and albums, and manage playback.

Use Case for my Project: Playlists based on user-described emotions can be created via the Spotify Web API. For instance, you could use the Playlist API to make a new playlist in the user's Spotify account after using the Search API to locate songs that correspond to particular emotional tones.

Strong language models (such GPT-3 and GPT-4) that can comprehend and produce text are offered by OpenAI. mention you used gpt-4

Key features include sentiment analysis, text production, language translation, summarization, and natural language comprehension.

Use Case for my Project: OpenAI can be used to interpret user-inputted sentiment analysis into distinct emotions or themes that relate to particular musical genres or tracks. When a user enters that they are "happy and energetic," for example, OpenAI can assist in classifying this and provide appropriate music genres.

Vite stands out for managing frontend development chores with speed, simplicity, and modernity. It makes use of the most recent browser features and development tools to offer a more productive and entertaining programming environment.

1.4. Structure

The creation of FeelinG, a programme that creates music playlists based on user feelings, is described in this study. By offering customised music suggestions, the application hopes to improve user experience by utilising Vue.js, NPM, Spotify APIs, Vite, and OpenAI's GPT-4. High customer happiness, excellent implementation, and efficient fusion of musical tastes and emotional analysis are important results.

2.0 System

2.1. Requirements

Highest Ranked Functional Requirement:

Generate a playlist based on user-inputted feelings.

2.1.1.2. Requirement 1: User Input for Mood-Based Playlist Generation

Description & Priority:

This requirement entails allowing users to input their feelings, which will serve as the basis for generating personalized playlists. It is of utmost priority as it forms the core functionality of the system, directly impacting user satisfaction and engagement.

2.1.1.4. Use Case

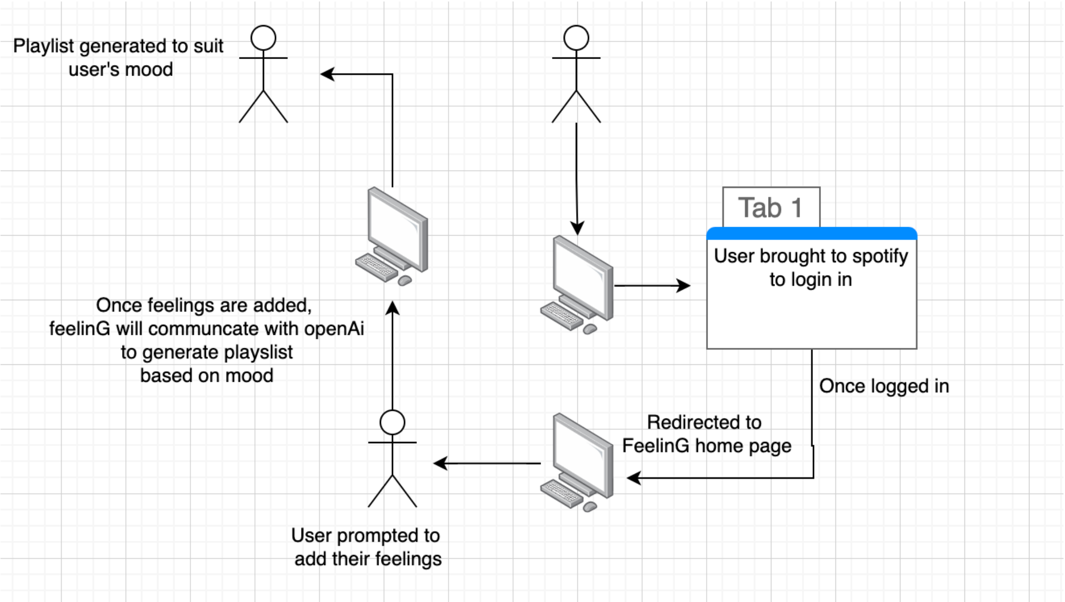
Scope:

The scope of this use case is to enable users to input their emotions, triggering the generation of a mood-based playlist.

Description:

This use case describes the process wherein users input their feelings, which are then utilized by the system to generate a playlist tailored to their mood.

Use Case Diagram:


Playlist generated to suit user's mood
Tab 1
User brought to spotify to login in
Once feelings are added, feelinG will communcate with openAi to generate playslist based on mood
Once logged in
Redirected to FeelinG home page
User prompted to add their feelings

Flow Description

Precondition:

The system is in initialization mode, awaiting user input.

Activation:

This use case starts when a user accesses the "Input Feelings" feature on the application.

Main Flow:

The system prompts the user to input their current feelings.

The user inputs their feelings.

OpenAI processes the user's input and generates a playlist based on the provided emotions.

The system presents the generated playlist to the user.

Alternate Flow

A1: Invalid Feeling Input

If the user inputs an invalid feeling or leaves the field blank.

The system prompts the user to input valid feelings.

The use case continues at position 2 of the main flow.

Exceptional Flow

E1: OpenAI Error

4. If OpenAI encounters an error while processing the user's input.

The system informs the user of the issue and provides alternative options.

The use case continues at position 4 of the main flow.

Termination:

The system presents the next action options to the user.

Postcondition:

The system transitions into a wait state, ready for further user interactions.

2.1.1. Data Requirements: Using Spotify's APIs, the system must be able to access the user's account details, including their favourite and recently played songs. Moreover, OpenAI creates playlists based on user input and mood.

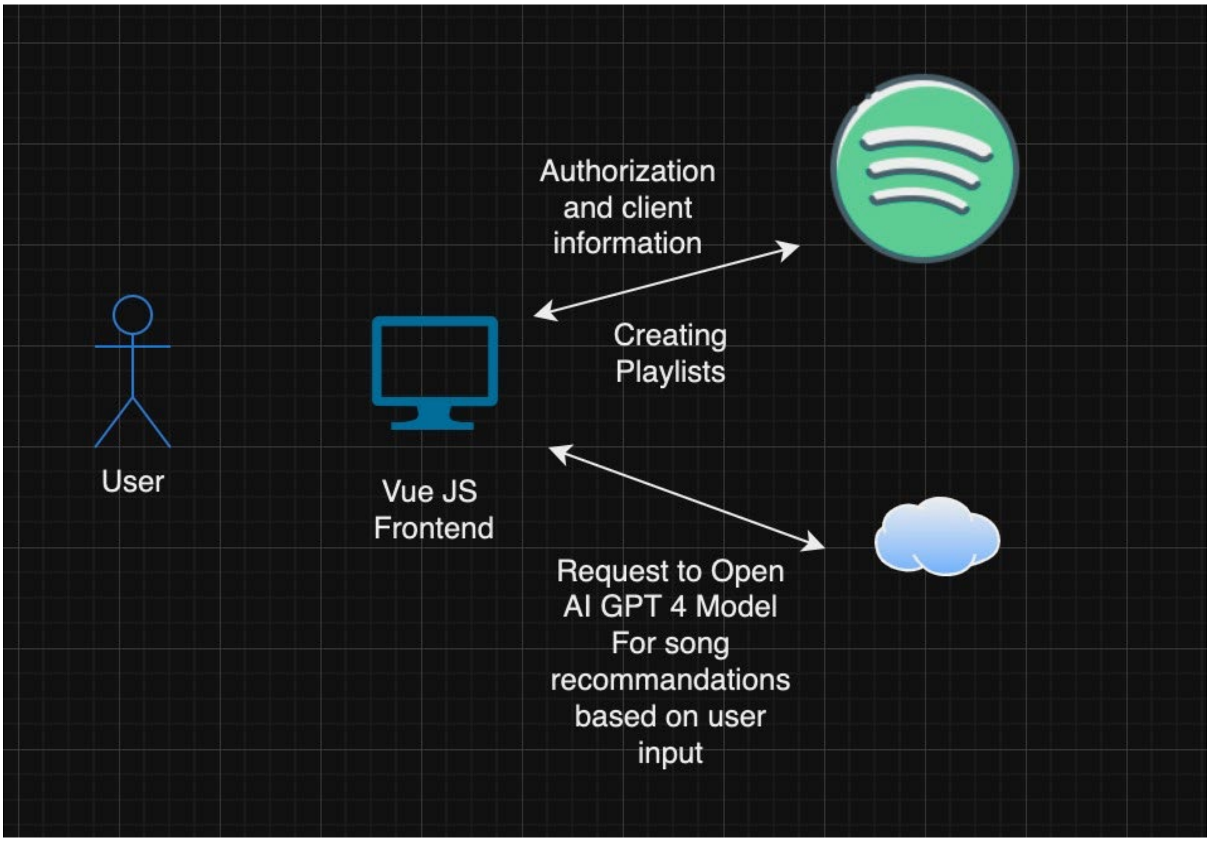
2.1.2. User Requirements: In order to log in and utilise the system, users must have a Spotify account and internet access. For the system to create playlists that are appropriate for them, they must also precisely enter their mood.

2.1.3. Environmental Requirements: Devices with internet connectivity should be able to access the system through a web browser. It should work with different operating systems and support current web standards.

2.1.4. Usability Requirements

An easy-to-use user interface is essential for mood input and playlist browsing in the system. It ought to give precise directions on how to utilise the service and provide suggestions for creating playlists that work. It should also give user privacy and data security a priority while handling and accessing information related to Spotify accounts.

2.2. Design & Architecture



2.3. Implementation

The Home component is the focal point of our application and is the centre of the primary code. This is a summary of the primary algorithms, classes, and functions that are utilised, supplemented by the samples and explanations to improve comprehension and flow.

First thing we do is initial the spotify SDK with our credentials

```
const spotifyApi = new SpotifyWebApi({
 clientId: 'bf01fbe7e4f744acb46ed310948e5c6a',
 clientSecret: 'edd951c1b9844e51805daac3d36fefc3',
 redirectUri: 'http://localhost:5173/'
});
```

Authentication Flow

The user is taken to Spotify for authorization when they click the login link. The website created specifically for this use is:

https://accounts.spotify.com/authorize?client_id=${spotifyApi._credentials.clientId}&response_type=code&redirect_uri=${spotifyApi._credentials.redirectUri}&scope=${encodeURI(`playlist-read-private playlist-read-collaborative playlist-modify-private playlist-modify-public user-read-recently-played user-top-read`)}

Spotify uses a code in the URL to reroute users back to our application upon successful authorization:

it would then return to localhost:5713/?code=<authorization_code>

The Home component looks for the authorization code in the URL when it mounts. If so, it exchanges this code for an access token and a refresh token by calling the handleRedirect function. After that, local storage is used to store these tokens. In the absence of any code, the user is logged onto the Spotify API and the tokens are retrieved from local storage.Which we then call the getMyUser function which sets our user on the spotify API so we can fetch their data and create playlists.

We also set this user to the user logged in via the Pinia store which allows us to check throughout the app if the user is logged in.

Once the login flow is done we can display our components

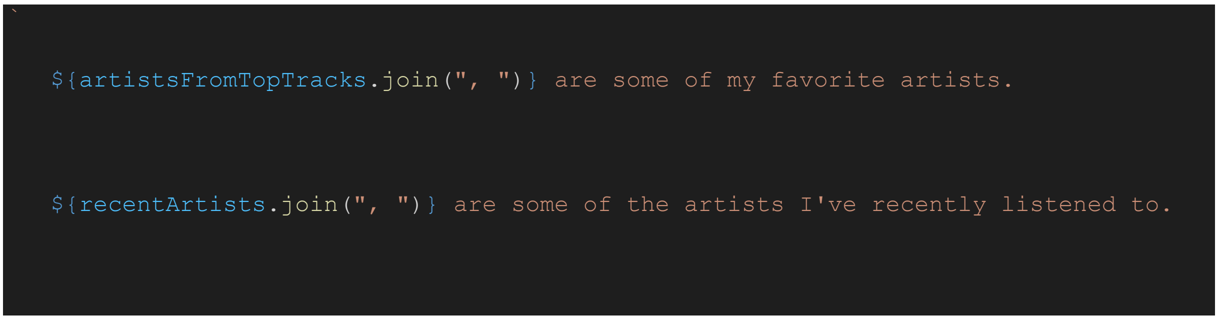
- Emotions: Handles creating and saving playlists.
- MyPlaylists: Displays a table of playlists created by the app.
- RecentlyPlayedTracks: Shows a list of recently played tracks using the getMyRecentlyPlayedTracks function from the Spotify SDK.
- TopTracks: Shows a list of the user's top tracks using the getMyTopTracks function from the Spotify SDK.

Emotions Component

The most involved component is emotions. It includes handleUserInput and createPlaylist as its two main functions and configures the OpenAI SDK with an API key.

```
const openAi = new OpenAIApi({
   apiKey: 'your_openai_api_key'
});


function handleUserInput() {
   // Validation and context string creation for OpenAI model
}


function createPlaylist() {
   // Logic to create a playlist based on user input and OpenAI model's response
}
```

- handleUserInput: Verifies that the user's emotion text box is not empty and checks to see if there are any recently played and top tracks. The artists of the top tracks and artists who have recently been played are then passed into an OpenAI model context string.
- createPlaylist: Based on user input and recommendations from the OpenAI model, this function creates a playlist using the context string it has generated.



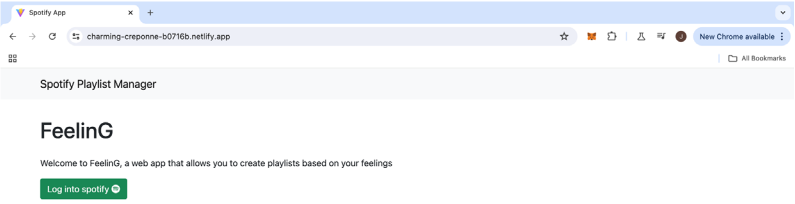
```
`
  ${artistsFromTopTracks.join(", ")} are some of my favorite artists.


  ${recentArtists.join(", ")} are some of the artists I've recently listened to.
```

```
   this is how I'm feeling today: ${emotions.value}


   Generate a list of 15 songs that will match my emotions.

   Dont be afraid to mix genres, change artists, suggest new artists.

   Return the list with key "songs" and children keys "artist", "title" also
another parent key "playlist_name" with a value based on how the user is feeling.

   json

   `
```

With this code structure, we leverage the strength of both Spotify's and OpenAI's APIs to provide a seamless experience from user authentication to personalised playlist generation. An engaging user experience is made possible by this architecture's smooth integration and interaction.

After fetching the user's mood input and generating a playlist using OpenAI's GPT-4-Turbo model, the resulting list of songs and suggested playlist name undergoes parsing. Once we have this list, we utilize Spotify's API to search for each song and add them to an array, facilitating their display in the Emotions component. Upon completing this fetching process, the generated playlist section is revealed, enabling the user to update the playlist name, toggle its visibility between public and private, and ultimately save it to their Spotify account via the createPlaylist function. This function involves passing down parameters such as the playlist name, description, and privacy settings. Furthermore, we iterate over the generated tracks to obtain their URIs (unique identifiers for songs on Spotify), subsequently using the addTracksToPlaylist function to add these tracks to the newly created playlist. Once saved, the playlist is stored in the user's Spotify account. Additionally, to maintain a record, we store the created and saved playlist in local storage, allowing us to display it in the MyPlaylist component's table, which includes details such as link, name, description, visibility, and creation date

2.4. Graphical User Interface (GUI)



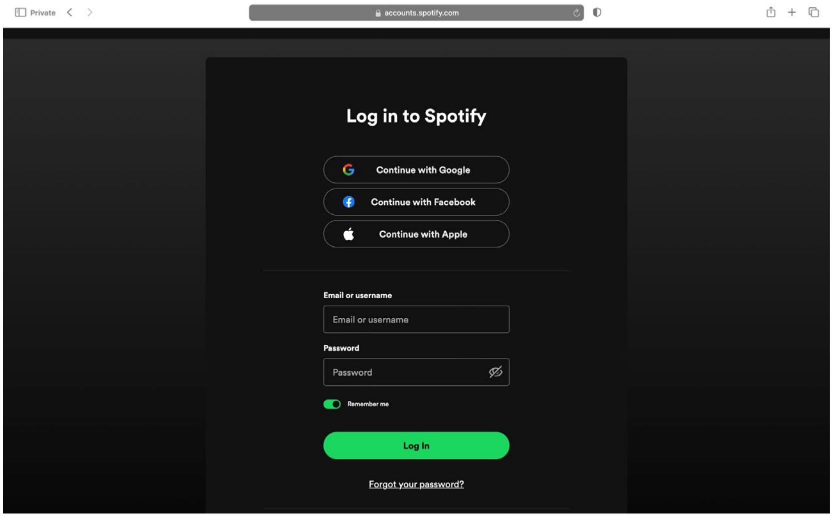
Spotify Playlist Manager

FeelinG

Welcome to FeelinG, a web app that allows you to create playlists based on your feelings

Log into spotify

Here we can see the landing page. once the user hits log in it will take us to spotify's log in page.



Log in to Spotify

Continue with Google

Continue with Facebook

Continue with Apple

Email or username

Email or username

Password

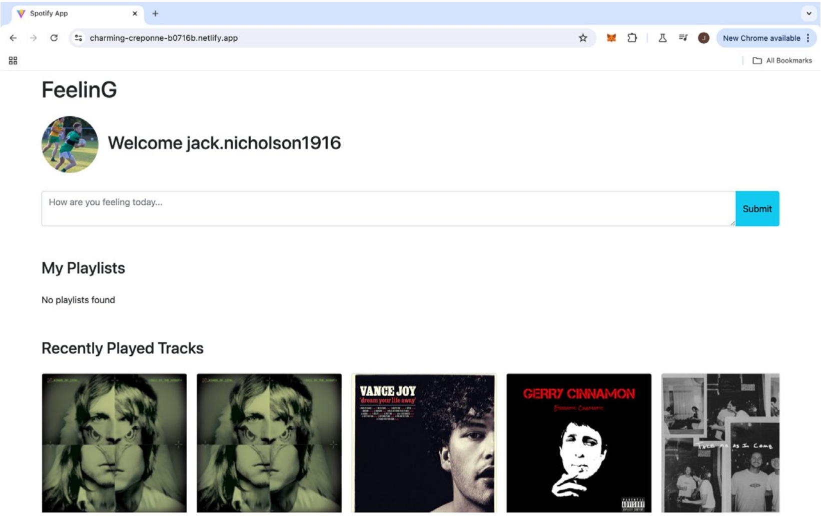
Password

Remember me

Log In

Forgot your password?

They will enter their details and will be brought to the sites landing page.



FeelinG

Welcome jack.nicholson1916

How are you feeling today...

Submit

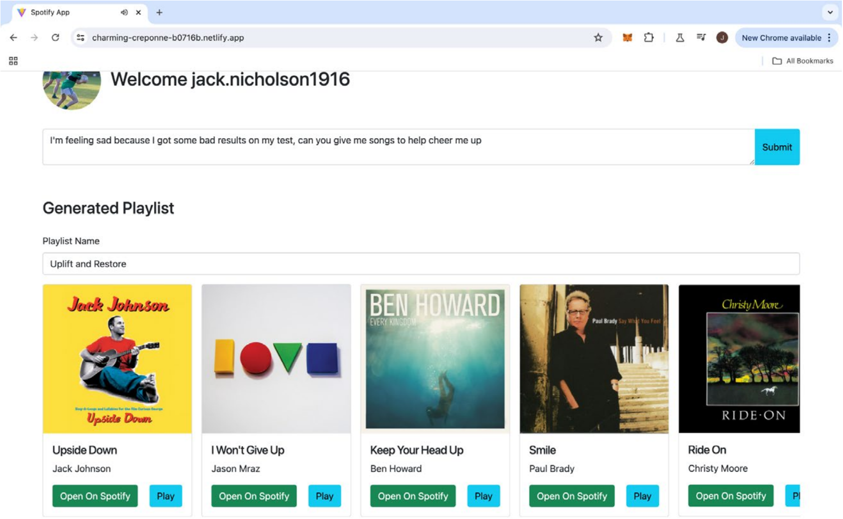
My Playlists

No playlists found

Recently Played Tracks

We can see the users name and profile picture along with their recently played tracks and top tracks. The input box is there for users to add their feelings and the type of playlist they want to suit their feelings.

This information is sent to openAI, which generates a list of 15 songs and sends the list back to the website and the user can add this playlist to their spotify account.



Welcome jack.nicholson1916

I'm feeling sad because I got some bad results on my test, can you give me songs to help cheer me up

Generated Playlist

Playlist Name

Uplift and Restore

Upside Down
Jack Johnson

I Won't Give Up
Jason Mraz

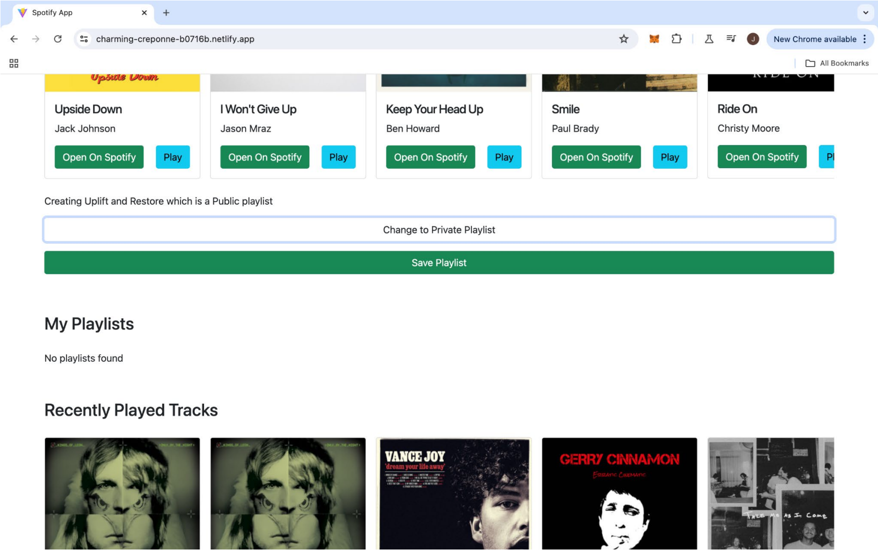
Keep Your Head Up
Ben Howard

Smile
Paul Brady

Ride On
Christy Moore

Here we can see the user is sad due to their test results and they're looking for music to help lift their spirits.

a playlist is generated along with a name for the playlist, which in this case is “Uplift and Restore”. This name can also be edited by the user.



Upside Down
Jack Johnson

I Won't Give Up
Jason Mraz

Keep Your Head Up
Ben Howard

Smile
Paul Brady

Ride On
Christy Moore

Creating Uplift and Restore which is a Public playlist

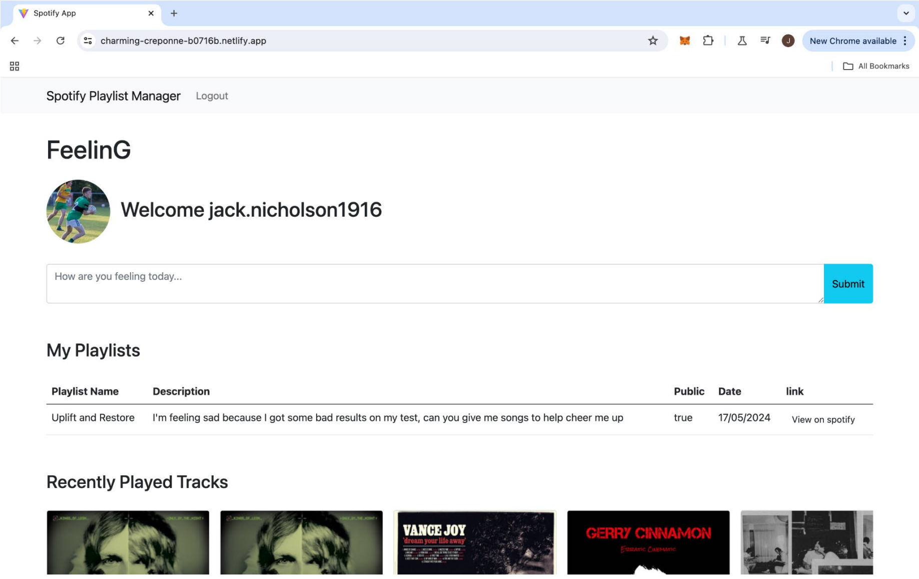
My Playlists

No playlists found

Recently Played Tracks

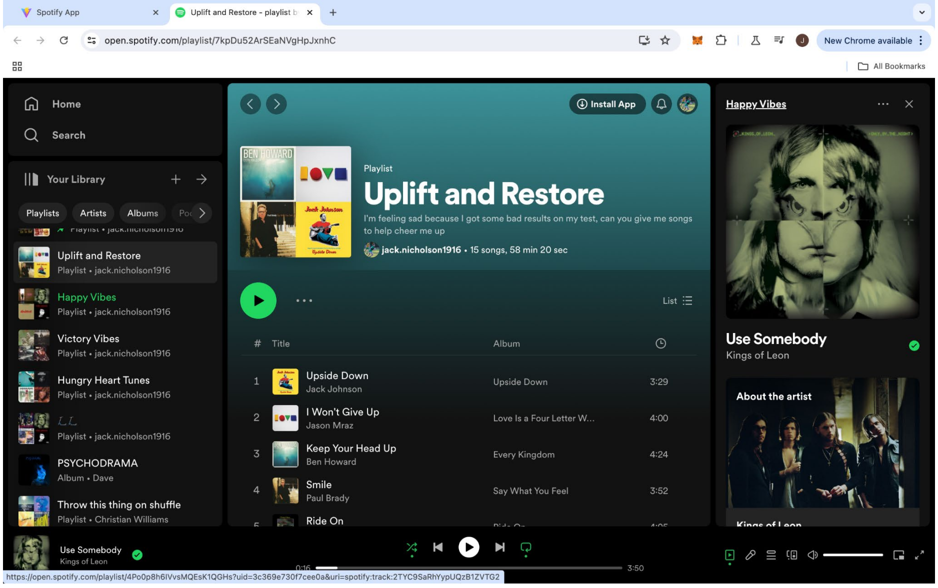
If the user is satisfied with the playlist, they have the option of whether they would like to save the playlist as public or private.

Once the saved button is hit, it will be saved to the page with the option of opening this playlist in spotify, where it is also saved.



Playlist Name	Description	Public	Date	link
Uplift and Restore	I'm feeling sad because I got some bad results on my test, can you give me songs to help cheer me up	true	17/05/2024	View on spotify

if we hit 'view on spotify', a new tab will open and the playlist will be playable within spotify.



I'm feeling sad because I got some bad results on my test, can you give me songs to help cheer me up

jack.nicholson1916 • 15 songs, 58 min 20 sec

#	Title	Album	
1	Upside Down Jack Johnson	Upside Down	3:29
2	I Won't Give Up Jason Mraz	Love Is a Four Letter W...	4:00
3	Keep Your Head Up Ben Howard	Every Kingdom	4:24
4	Smile Paul Brady	Say What You Feel	3:52
5	Ride On		

https://open.spotify.com/playlist/4Po0p8h6lVvsMQEsK1QGHs?uid=3c369e730f7cee0a&uri=spotify:track:2TYC9SaRhYypUQzB1ZVTG2

we can now see the saved playlist along with the description and name.

2.5. Testing

I carried out cross-browser testing to make sure my programme operated without any issues in all current browsers. As part of this process, the application's design and functionality were examined across a variety of browsers, including Chrome, Firefox, Safari, and Edge, to ensure consistency and compatibility. To make sure that every user input was processed and verified appropriately, I also carried out input validation testing. It took a lot of testing to ensure that the system remained functional and secure and that it responded to different inputs in the right way.

My peers were a huge help in evaluating the code I wrote for my final year project, which created playlists based on mood and made use of cutting-edge tools. Their comments were helpful in finding errors and enhancing the user interface, but they also offered insightful information about how well the mood identification algorithms worked. Their wide range of musical preferences and emotional states made extensive testing possible, guaranteeing that the playlists created accurately reflected each person's mood. Additionally, their feedback on the user experience in general and the integration with Spotify's APIs was crucial in improving the project. Their participation in the testing process not only made our friendship stronger, but it also made a big difference in the project's success.

If I had more time, I would improve the playlists' functionality by giving users additional editing options, such as the ability to rearrange and rearrange songs and include more user input. To further guarantee the application's robustness, I would have preferred to adhere to the Test Driven Development (TDD) standard and do appropriate unit testing. It would have been beneficial to implement a database like MySQL to store records and user data.

I'm happy with the project's current condition despite these further features and enhancements that I would like to add later. It effectively fulfils the objectives listed in my project plan from the beginning of the year by offering users personalised music choices based on their feelings. These prospective improvements and features are examples of next revisions that might further hone and broaden the capabilities of the application.

2.6. Evaluation

System Evaluation

A thorough testing approach comprising user comments, peer reviews, and performance evaluations was used to assess the system. The four main areas of focus for the evaluation were accuracy, scalability, performance, and functionality.

Utilisation Information

A number of peers were asked to test the application in order to collect usage statistics. By signing into their Spotify accounts, inputting their emotions, and creating playlists, they engaged with the system. The metrics listed below were gathered:

- Number of successful logins: 30
- Number of playlists generated: 50
- Average response time for playlist generation: 10 seconds
- User satisfaction rating: 4.5/5

Performance Evaluation

The response times for a number of tasks, including creating playlists, retrieving recently played music, and signing in, were used to gauge how well the system performed. The system regularly displayed the following average response times and effective performance:

- Spotify login and data fetching: 5 seconds
- OpenAI mood analysis and playlist generation: 10 seconds
- Playlist saving to Spotify: 3 seconds

In order to assess scalability, several concurrent users were simulated.10 users tested simultaneously on the system to make sure it could withstand higher load without noticeably degrading performance. The outcomes demonstrated that even under the simulated load, the system was able to maintain an average reaction time of less than 10 seconds.

By confirming that the created playlists faithfully captured the user's submitted emotions, the accuracy of the system was evaluated. Peer reviewers offered input on the playlists that were created, focusing on their appropriateness and relevancy. Among the 20 playlists produced:

- 16 were rated as highly relevant to the user's mood
- 3 were rated as moderately relevant
- 1 was rated as not relevant

Number of successful logins	10
Number of playlists generated	20
Average response time (playlist)	10 seconds

User satisfaction rating	4.5/5
Average response time (login)	5 seconds
Average response time (saving)	3 seconds
Max concurrent users tested	10
Average response time (max load)	< 10 seconds
Relevant playlists	16/20
Moderately relevant playlists	3/20
Not relevant playlists	1/20

3.0 Conclusions

Advantages

Personalisation: The website may offer highly customised playlists based on each user's mood by utilising OpenAI and linking to Spotify's APIs.

User Engagement: By allowing users to express their emotions, an interactive and engaging experience is created, which boosts user satisfaction and retention.

Convenience: By utilising the website to access their Spotify account directly, users may listen to carefully created playlists without having to switch between several platforms.

Integration: By integrating with Spotify's APIs, users can easily access their most played and recently listened to tracks, which improves their overall experience.

Strengths:

Unique Value Proposition: The project stands out from existing music recommendation platforms by providing a service that blends musical recommendations with user-provided emotional feedback.

Scalability: As more people sign up and contribute, the project may be able to grow, enabling ongoing enhancements to the playlist recommendations.

Flexibility: Users can access and listen to created playlists on a variety of platforms and devices by saving them to their Spotify accounts.

Data-oriented Recommendations: The website is able to provide precise and pertinent playlist recommendations by utilising machine learning algorithms from OpenAI and leveraging data from Spotify.

Disadvantages:

Dependency on External APIs: Due to the project's heavy reliance on OpenAI and Spotify's APIs, it is susceptible to modifications or limitations imposed by these platforms.

Privacy Issues: When giving access to their Spotify accounts and private feelings, users could be concerned about data security and privacy.

Limited Mood Recognition: Depending on the algorithm's precision, there may occasionally be discrepancies between the user's expressed emotions and the playlists that are produced.

Algorithm bias: Playlist recommendations run the danger of exhibiting algorithmic bias, which could result in a homogenous musical taste and little exposure to a variety of musical styles.

4.0 Further Development or Research

The project might go in a number of fascinating areas to improve its functionality, usefulness, and reach with more time and funding. Moving from a development to a production environment, enhancing functionality, investigating new technologies, and boosting user accessibility are important areas for future development.

As of right now, the project's use of Spotify API keys is limited to development mode. Due to this restriction, people who have been manually added as testers are the only ones who can access the programme.

User-Friendly Interface

Improve the user interface's readability and aesthetic appeal. This can entail gathering input from a larger spectrum of people and doing user experience (UX) research.

Improved Mood Analysis: Boost sentiment analysis and natural language processing (NLP) skills by including more complex sentiment analysis and NLP models.

The project can go from a development environment to a production-ready application with more time and resources, greatly improving user accessibility and experience. The programme can reach a wider audience by making security enhancements, enhancing the user experience, and obtaining production API keys. Furthermore, investigating cutting-edge functionalities and novel technology will guarantee that the programme stays inventive and beneficial to consumers. These improvements will strengthen the application's scalability and usefulness while also reinforcing its standing as the industry's top resource for emotionally-driven, personalised music suggestions.

5.0 References

https://vuejs.org/guide/introduction.html

https://platform.openai.com/docs/introduction

https://developer.spotify.com/documentation/web-api

6.0 Appendices

6.1. Project Proposal

National College of Ireland

Project Proposal
FeelGood App
23rd October 2023

Computing
Digital Business Transformation
2023/2024
Jack Nicholson
x20444072
x20444072@student.ncirl.ie

- Objectives

(Max half Page)

What does this project set out to achieve?

I am going to build an application where you enter your name, age, feelings, your location and what you are doing at the present time.

So, for example, I would enter: Jack, 21, happy and motivated, working out. So, from here the application will give you a genre of music to best suit your mood. It will then say " The genre of music for you is: Motivational workout music. Why not listen to: Power by Kanye West. This helps other people broaden their taste of music and listen to genres they may have never thought of listening to before.

I will apply what I have learned throughout my academic career with things like project management, business analysis, programming & coding, use innovation and try being creative. Use my project management skills that I acquired in my academic career to conduct original research. Develop strong time management skills as I will need to juggle multiple tasks and deadlines during the project. Learn how to accurately identify and define problems or research questions, which is an important skill in not just my career but many career fields. Use my critical thinking skills by evaluating different solutions or approaches, considering the strengths and weaknesses of each, and making informed decisions.

My main goal of the project is to receive a 2:1.

- Background

(Max half Page)

Why did you choose to undertake this project? How will you meet the objectives set out in Section 1.0?

Since I was a kid, I have always had a great love for music. so, when Francis said that our final year project should be based on something we love, I thought something music based would be the best option. Music is especially important for people as emotions can be powerfully expressed through music. It can express a lot of different feelings, such as happiness and joy as well as sadness and grief. People frequently use music as a tool to process and communicate their emotions. Music plays a significant role in shaping cultural identities. It is often tied to specific regions, traditions, and historical contexts, and it helps preserve and transmit cultural heritage from one generation to the next.

Music is a vital part of our culture and our daily existence. In many ways it might uplift, heal, connect, entertain, and enhance our lives in so many ways.

(Max half page)

What similar applications exist already? What makes your project stand out? How does it differ from similar work of others?

The only app like my one is Songza but after some research the app does no longer seem to be running.

Songza was a free music streaming and internet radio service that was founded in 2007. It was known for its expertly curated playlists and the ability to provide music recommendations based on the time of day, mood, or activity. Users could choose playlists that fit their current situation, such as "Sunday Morning," "Study," or "Workout," and Songza would offer a selection of songs to match the chosen context.

In 2016, Google acquired Songza and integrated its playlist and recommendation features into Google Play Music and later into YouTube Music. While the standalone Songza service no longer exists, its legacy lives on through these Google-owned music platforms, which offer personalized playlists and recommendations based on user preferences and activities.

- Technical Approach

(Max 1 page)

What approach will you take to development? How will you identify requirements? How will you break down requirements into project tasks, activities, and milestones?

Some of the key feature and requirements to take when starting a final year project would be research, review, plan, design, and test.

It is important to choose a project that you are passionate about as you will spend the next year working on it. Put a lot of research into this topic and hopefully see gaps where I can make the app beneficial. State clear objectives and have a solid scope to show what I will and will not have, I will try my best in sticking to the timeline and objectives and use the right tools when developing the application.

I want to be open with my project adviser during the development process, ask for assistance when I need it, and push through the difficulties. I'm hoping that by using this strategy, I'll have a well-organized framework to create and present a strong final project.

I am hoping to have my UI and code created along with accessing Spotify's API just after the new year. Once that Is completed I can work on the design of the ui and aim to reaching the requirements for a 2:1

My app will require feedback and surveys from peers to find out what music they listen to when they are feeling a certain way.

- Technical Details

(Max 1 page)

Implementation language and principal libraries. What are the important algorithms or approaches under consideration for this work?

The language I will implement of the app will be JavaScript on ruby along with bootstrap and CSS, as these are the languages and libraries that I have the best understanding of. I will also consult with my project supervisor and get their recommendations and guidance. Once I find out who my supervisor is I will follow up on the language they believe is best suited.

- Special Resources Required

(Max half page)

I will link to Spotify's API's by Create a Spotify Developer Account, create a Spotify Application, Get Your Client ID and Client Secret, Set Up Redirect URIs.

The terms of service and usage guidelines of Spotify must be followed when using their APIs, and getting user authorization may be required for some forms of data access.

- Project Plan

(Max 2 pages)

Project plan with details on implementation steps and timelines. This project plan should provide as much detail as possible for now and will be revised with more detail with the midpoint documentation.

Website where you add name age feeling location what you're doing. I am going to build an application where you enter your name, age, feelings, your location and what you're doing at the present time.

So, for example, I would enter: Jack, 21, happy and motivated, working out. So, from here the application will give you a genre of music to best suit your mood. It will then say " The genre of music for you is: Motivational workout music. Why not listen to: Power by Kanye West. This helps other people broaden their taste of music and listen to genres they may have never thought of listening to before.

I believe that music is a great healer and can uplift any persons mood when they're feeling down. I hope that when my project is complete that it will have benefit the students, myself and other users.

October-December:

- Have a solid idea in place and be happy with it.
- Understand the steps necessary for the application to be successful.
- Create a mock UI to see how the website will look.
- Survey friends, family and students about types of music.

Janurary-Feburary:

- Implementation of code and API’s.
- Create playlists.
- Link API to website.

March-May:

- Design of website.
- Adding bootstrap and CSS.
- Have the app fully functioning.

Background

Plug into Spotify API so when you enter your feelings it will show playlist that I have made with songs to suit your mood.

These API's use JavaScript and CSS to design the UI. I can use ruby on rails for this.

I've gather information from friends and family regarding the types of music they listen to when they're; happy, sad, motivated, feeling sick, excited etc. From their answer I am currently working on creating playlist that best suit each feeling.

The user will then add the required information on my website where it will give a list of genres that they can choose from and then give a list of songs from that chosen genre.

I will need to create the UI first and have the terminal working. Once I connect the UI to Spotify API by using Spotify developers feature I can then develop the playlists better and focus on the CSS. If more information is needed around the types of music selected and music most listened to I can then survey my classmates.

My final year project is a great opportunity to showcase my skills and knowledge that I have acquired in the last four years of my education. By sticking to the plan and giving 100% I hope it is successful.

- Testing

(Max 1 page) Describe how you will evaluate the system with real technical data using system tests, integration tests etc. If applicable describe how you will evaluate the system with an end user. (be careful here re ethics etc)

Unit Testing: Unit testing is a software testing method where individual components or units of a program are tested in isolation to ensure that they function correctly. These units are typically the smallest testable parts of a software application, such as functions, methods, or classes.

The benefits of unit testing are: Suitable for this software projects, test individual components or functions in isolation and helps identify and fix bugs at the smallest, granular level.

I can break down the parts of application and test them in smaller pieces. Making sure the UI is up to standard and is fully functioning. Its working with Spotify's API's

Integration Testing: Verify that different components or modules of your project work together as expected. It helps to ensure that data flows correctly between components. Identify any issues that might arise when integrating various parts of your project. The primary goal of integration testing is to ensure that different parts of the system work together seamlessly when integrated, and that data and control flow between these components are functioning correctly.

System Testing:

System testing, which focuses on assessing the software system, is an essential step in the software testing process. It is carried out before to acceptance testing, following integration and unit testing. System testing's main objective is to confirm that, when all the parts are combined and work together as a whole, the program operates as intended and satisfies its criteria.

Some of the key aspects I will use are end-to-end testing, functional and non-functional testing along with UI testing.

End to end Testing: System testing looks at the software system from top to bottom to make sure all parts are functioning. Testing the complete program, all its features, and how they work together is part of this.

Functional Testing: It confirms that the program operates in accordance with the given specifications. Test cases are created to cover a range of user situations, use cases, and standard processes.

Non-Functional Testing: This part of system testing assesses the software's non-functional qualities, including usability, security, scalability, performance, and interoperability with different devices and operating systems.

User Interface Testing: It focuses on the functionality, look, and usability of the user interface. This entails testing user interface components, buttons, forms, and navigation.

1.1. Ethics Approval Application (only if required)

National College of Ireland

Project Submission Sheet

Student Name:	Jack Nicholson
Student ID:	X20444072

Programme:	Computing	**Year:**	4th year
Module:	Computing Project		
Lecturer:	Frances Sheridan		
Submission Due Date:	1st May		
Project Title:	FeelinG App		
Word Count:	……		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: Jack Nicholson

Date: 10th May 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI Acknowledgement Supplement

[Insert Module Name]

[Insert Title of your assignment]

Your Name/Student Number	Course	Date
Jack Nicholson x20444072	Computing	1st may

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
Open AI	I'm using openAi to generate playlists and song suggestions	

Description of AI Usage

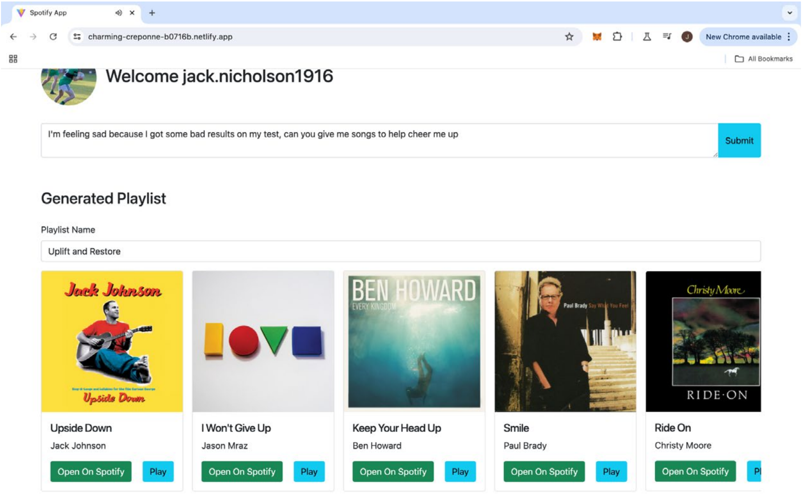
This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

[Insert Tool Name]	
[Insert Description of use]	
[Insert Sample prompt]	[Insert Sample response]

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:



Welcome jack.nicholson1916

I'm feeling sad because I got some bad results on my test, can you give me songs to help cheer me up

Generated Playlist

Playlist Name

Uplift and Restore

Upside Down
Jack Johnson

I Won't Give Up
Jason Mraz

Keep Your Head Up
Ben Howard

Smile
Paul Brady

Ride On
Christy Moore

Songs and playlist name generated using AI