

**VOVOCA -  
Stream, Upload & Download Royalty Free  
Music**

Major project report submitted in partial fulfilment of  
the requirement for the degree of **Bachelor of  
Technology**

in

**Computer Science and Engineering**

by

**Anshul Trivedi (181410)**

Under the supervision of

**Dr. Vipul Kumar Sharma**



Department of Computer Science  
Engineering and Information  
Technology,  
Jaypee University of Information Technology,  
Waknaghat, 173234, Himachal Pradesh, INDIA

# CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled "VOVOCA" in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Wagnaghat is an authentic record of my own work carried out over a period from December 2021 to March 2022 under the supervision of **Dr. Vipul Kumar Sharma** of Computer Science And Technology.

The matter embodied in the report has not been submitted for the award of any other degree or diploma

Anshul Trivedi (181410)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Supervisor Name: Dr. Vipul Kumar Sharma

Designation: Assistant Professor (Grade-II)

Department Name: Computer Science and Engineering

Dated: 12<sup>th</sup> May 2022

# ACKNOWLEDGEMENT

Firstly, we would like to express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully

We are really grateful and wish our profound indebtedness to Dr. Vivek Kumar Sehgal, Professor & HOD of the Department of CSE & IT, Jaypee University of Information Technology, Wanknaghat.

Deep Knowledge & keen interest of my supervisor in the field of “Data Science” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would also generously welcome each one of those individuals who have helped us straightforwardly or in a roundabout way in making this project a win.

In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Regards,

Anshul Trivedi (181410)

CSE | JUIT

# Table of Contents

CANDIDATE'S DECLARATION

ACKNOWLEDGEMENT

ABSTRACT

CHAPTER 1: INTRODUCTION

- INTRODUCTION
- OBJECTIVE
- METHODOLOGY
- PROBLEM STATEMENT
- ORGANIZATION

CHAPTER 2: LITERATURE SURVEY

CHAPTER 3: SYSTEM DEVELOPMENT

- RESOURCES
- STEPS
- E-R DIAGRAM

CHAPTER 4: PERFORMANCE ANALYSIS

- ANALYSIS TARGET VARIABLES
- DATA FLOW DIAGRAM

CHAPTER 5: CONCLUSION

FUTURE SCOPE

REFERENCES

APPENDICES

# ABSTRACT

Since March, 2020 we have been confined to our homes due to the impact of the COVID-19 pandemic. Music has helped the majority of us to cope with the lockdown, it helped us to relax, escape and raise our mood. Because of the prominence of streaming services, people are listening to more music from more artists than ever before.

We have also seen a significant rise in the number of music streaming platforms. However, there is no platform for streaming and downloading royalty free music. There are thousands of non-copyright tracks available on the internet but there is no centralized place where freelancers and students can go to get their right music for their projects without paying any royalty. The need for royalty free music extends to but is not limited to freelance projects and advertising agencies.

The main purpose of our application is that users can stream their favorite music and even download it hassle-free so they can use that audio in whichever way they want without worrying about copyright issues.

Consumers' preferred manner of listening to music has shifted to music streaming services. Streaming services provide users with limitless access to vast music libraries. These services store music on a server that users may access via their computers and mobile devices. mobile phones Users may listen to any music they choose while connected to the internet. It may then be played by choosing it on the application. On a lot of services, the Customers can select to download the song and listen to it offline. This implies that the music is saved on the consumer's device in the streaming service's memory app for service. The user does not own any of the music in the collection, comparable to how individuals borrow books from a library. Users are also unable to access the individual MP3 songs contained inside the catalogue, preventing them from making copies for personal use. Users may stream their favourite musicians on their own devices with a monthly membership, whether paid or free. Streaming technology, whether for music or video, has been steadily improving over the last 10 years. Rather of utilizing the main website, most streaming services enable consumers to download a free application that allows them to access the whole service

# Chapter 01-Introduction

## 1.1 Introduction:

As the creative copyrights laws are getting more stringent and complicated with time, the need for royalty-free, non-copyright sounds becomes more urgent than ever. Some of the major use cases of non-copyright sounds can be for college projects, freelance projects, video making, advertisement industry. This tells us that there is a market for this kind of music that needs to be tapped to its full potential. The problem with the currently existing market players that provide NCR sounds like YouTube is that they are not centralized. One can find thousands of NCR sounds but there is no filtering, therefore, it can be considered as a dump of sounds where you have to search through each of them along with plenty of other irrelevant music videos. There is no centralized marketplace for NCR sounds.

Music streaming apps have already established and proved their business models. The statistics help us show how cassettes have become a thing of the past and people have moved on to streaming apps and websites. As of late 2020, Spotify had 345 million users, moreover, that is just 33% of the music streaming market.

The music streaming market is already reaching new highs but there is still a lot to be discovered. Consumers' preferred manner of listening to music has shifted to music streaming services. Streaming services provide users with limitless access to vast music libraries. These services store music on a server that users may access via their computers and mobile devices. mobile phones Users may listen to any music they choose while connected to the internet. It may then be played by choosing it on the application. On a lot of services, the Customers can select to download the song and listen to it offline. This implies that the music is saved on the consumer's device in the streaming service's memory app for service. The user does not own any of the music in the collection, comparable to how individuals borrow books from a library. Users are also unable to access the individual MP3 songs contained inside the catalogue, preventing them from making copies for personal use. Users may stream their favorite musicians on their own devices with a monthly membership, whether paid or free. Streaming technology, whether for music or video, has been steadily

improving over the last 10 years. Rather of utilizing the main website, most streaming services enable consumers to download a free application that allows them to access the whole service. Because many customers have frequent access to a smart phone with internet, music streaming has become increasingly popular. As a result, the rise of music streaming services has been reliant on the technology that has accompanied it. The music business and its artists now have a new way of sharing and distributing music to people all over the world thanks to music streaming.

## 1.2 Objective:

The objective of this web app is to try to centralize the needs of thousands of freelancers and students. Our web app aims to become the one stop destination for NCR sounds and royalty-free music. We don't want people to spend their time searching for the right music for their project amongst thousands of irrelevant tracks. The users will be able to upload and download music from the web app itself. We endeavor to save their time so that they can focus on what actually matters, which is, the content and quality of their projects. The music list's lack of a search box is unsatisfactory. The users will be able to upload and download music from the web app itself. We endeavor to save their time time so that they can focus on what actually matters, which is, the content and quality of their projects.

Users will feel more at ease and relaxed because it will pay more attention to the features that are frequently used by users, excluding some rarely used features that consume a large amount of system processors, resulting in a music player that is lightweight, simple, and has powerful basic features. As a result, the app will leverage the search bar on the right side of the screen, as well as rapid scrolling utilizing alphabets, to allow users to swiftly sift through hundreds of songs to locate the ones they wish to listen. Because many developers recognize that modern city dwellers are stressed, they have seized the business opportunity, and a plethora of comparable applications have developed on the market.

However, present music players blindly seek a flashy design and numerous functions, resulting in excessive CPU and memory use on customers' mobile phones. For most ordinary users, however, such vast and numerous features are irrelevant. As a result, this project is devoted with the goal of improving performance and simplifying user demands.

People in modern culture live in a fast-paced world where pressure is always there. Because of the widespread usage of mobile phones, music has evolved into a daily spiritual necessity, and every mobile phone must include a music player. To balance stress and happiness, an application such as Vovoca is utilized. It is with people at all times, in all places, such as when they are on the bus or exercising. The application is developed to allow users to listen to music in a more convenient and pleasant manner. Furthermore, it is capable of playing music without interruption from commercials or offline.

The motivation behind this project was the team's common love for music. However, the idea for this web app stemmed when a friend of ours went looking for NCR sound for his freelance video project. We realized there is no go-to place on the internet where one can find good quality royalty-free music. The only usable site was YouTube but there was no way to filter it by category. We had to go through hundreds of tracks to find the right music. This solidified the need of the idea, on further discussing with our peers, we realized we were on track to solve a problem for thousands of students.

## 1.3 Methodology:

Our full-stack web application has CRUD functionality and is built on PENN stack. The PENN stack consists of the PostgreSQL, Express JS, Next JS and Node JS. The user interface has been made using the Next JS framework.

1. **PostgreSQL:** PostgreSQL is an advanced object-relational database management system that uses Structured Query Language (SQL) in addition to its own procedural language, PL/pgSQL.
2. **JavaScript:** It is a lightweight, interpreted programming language. Mostly well known as a scripting language, it is also used in non-browser environments like Node JS & Express JS.

The project aims to implement a scalable web app with the following features:



- 1) **SignUp** - Users will be able to create their accounts on the web app using their email.
- 2) **Login**- Users who have signed up will be able to access their dashboard and login to the web app using their set passwords.
- 3) **Stream music**- Authenticated users will be able to stream the available music on our web app.
- 4) **Filter through categories**- Users will also have the option to filter through different categories of music like Pop, Rock, Jazz etc.
- 5) **Upload Music**- Authenticated users will be able to upload their own music to the web app.  
**Download Music**- Users will be able to download the available music directly to their devices to enjoy a seamless offline experience.

## 1.4 Problem Statement:

Because of the severe rivalry among music player programs, several developers attempted to add a plethora of features, advertisements, and content to their individual music players in order to maintain and attract new users. This tendency has made it more difficult for customers to access material from their music player, which also makes filtering the content they want more difficult. The music player will get even more bloated as the program evolves and the number of features increases, making the user experience less fluid. According to Mehul (2018), consumers become annoyed and upset if they wait a long time for a response from a mobile application, and 48% of users never return to the same app.

When consumers keep adding new music to their playlist, the complexity of filtering the songs they wish to listen to increases. After hundreds of songs have been added to the playlist, the user may only search for songs by repeatedly swiping up or down. It is easy to miss the songs that the user wishes to filter if the material is not properly examined, and then to repeat the process until the desired result is obtained. As a result, users have an exceedingly negative experience.

## 1.5 Organization:

The project report is broken down into 5 sections. The first chapter covers the background and motivation for the proposed application, the problem statement and aims to answer the issue statement, the recommended technique or research, and the highlighting of successful proposed applications. Chapter 2 illustrates the literature survey of the project from which we took the references. The system development chapter includes the site map, use case diagram, activity diagram, and system wireframe, which is the proposed application's user interface. Software design approach, tools, requirements, system performance specifications, and timescales are discussed in Chapter 4. The fifth chapter concludes the implementation, project evaluation, benefits and future scope of the project.

# Chapter 02- Literature Survey

## 2.1 Review on existing application

### Review on YY Music:

YY Music is a free music player that can play any song on YouTube, play music in the background, and free download songs to local storage. It has an extensive music collection that includes practically all songs in English, Chinese, Cantonese, Japanese, and even Korean. It performs exceptionally well in terms of functionality and user-application interaction.

The 'Discover' section of YY Music's site provides random lists and popularity rankings of numerous music categories, such as rank, top 100, recent songs, and weekly 20. This function allows users to instantly click and listen to the current music when they are confused about which songs to choose or are weary of listening to old tunes.

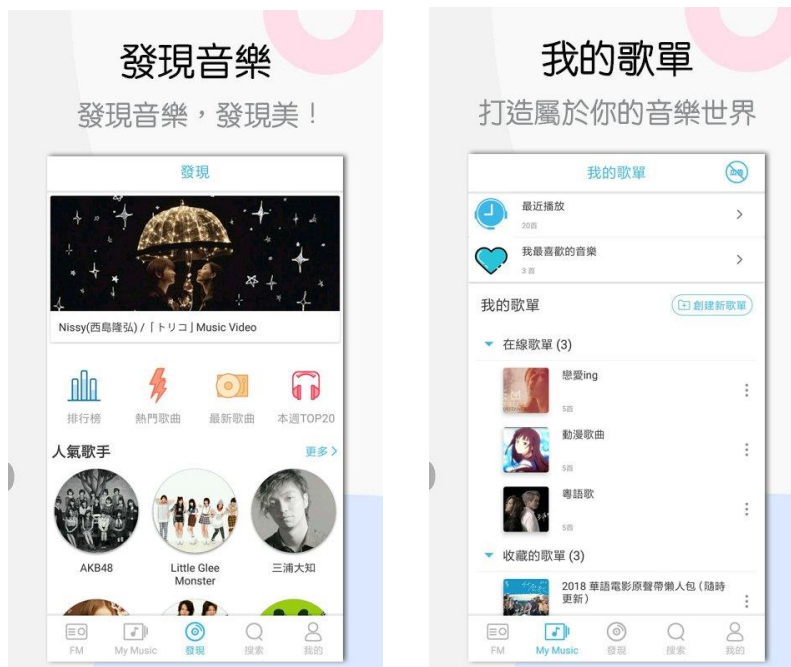


Figure 1

Furthermore, the discovery page will provide a selection of prominent musicians and playlists, allowing visitors to listen to their favorite artists' music rapidly. When the user selects the playlist, all of the songs are shown for the user's convenience. If the user is pleased with all of the songs in the playlist, just click "Play All" in the upper right corner to begin listening to them all.

## Review on Spotify:

Spotify is a music streaming service that provides customers with other, legal methods to listen to music. Spotify offers both free and premium memberships that allow users to create personalized radio stations and playlists. This music streaming service allows users to socialize through music by allowing them to share songs on social media platforms like Facebook and Twitter (Spotify, n.d.).

Spotify's target demographic is mostly made up of early adopters and opinion leaders, such as tech-savvy music fans, musicians, and record companies. 50 percent of Spotify users are between the ages of 18 and 34, and 24 percent come from households with an annual income of around \$100,000.

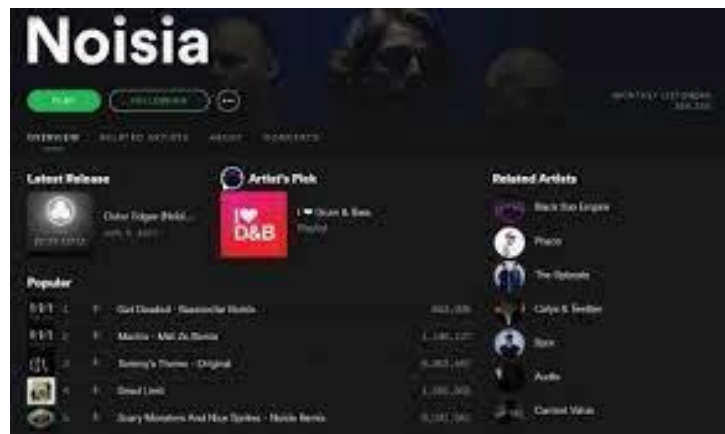
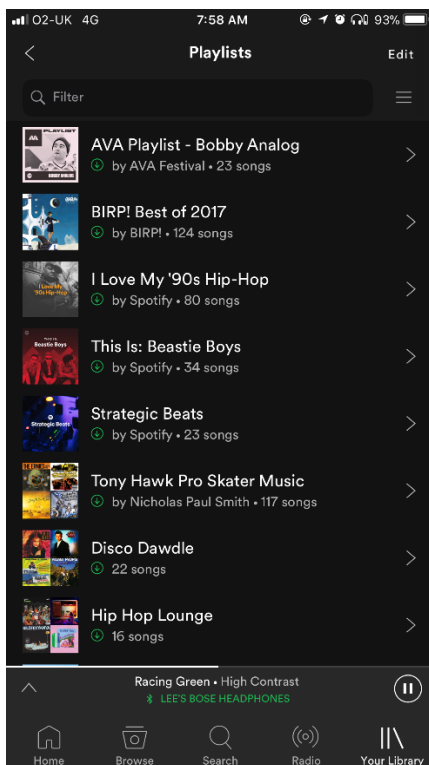


Figure 2

Because of the amount of ease, it provides for respondents, we chose a web-based survey over the numerous other options for obtaining quantitative data:

rather of holding a series of group meetings to distribute pen and paper,

Using a web-based collecting mechanism, the survey may be completed from anywhere. Furthermore, using an online survey method removes hours of tiresome data entering and processing.

## Review on Soundcloud:

Communication and Information In language instruction, technology plays a critical role. People are becoming increasingly involved in internet activities as their lives progress.

They build a virtual environment that fosters innovation and expands networking. Virtual environments are areas where individuals may find, trade, and share information, thoughts, and ideas over the internet (Parker & Berry, 2010).

The usage of technology is no longer restricted to a select number of students. According to Pim (2013), various case studies show that the application of technology is well-developed and inventive. Some language learners are exposed to the use of technology from an early age. Furthermore, elementary instructors assist young kids in using digital play to learn languages.

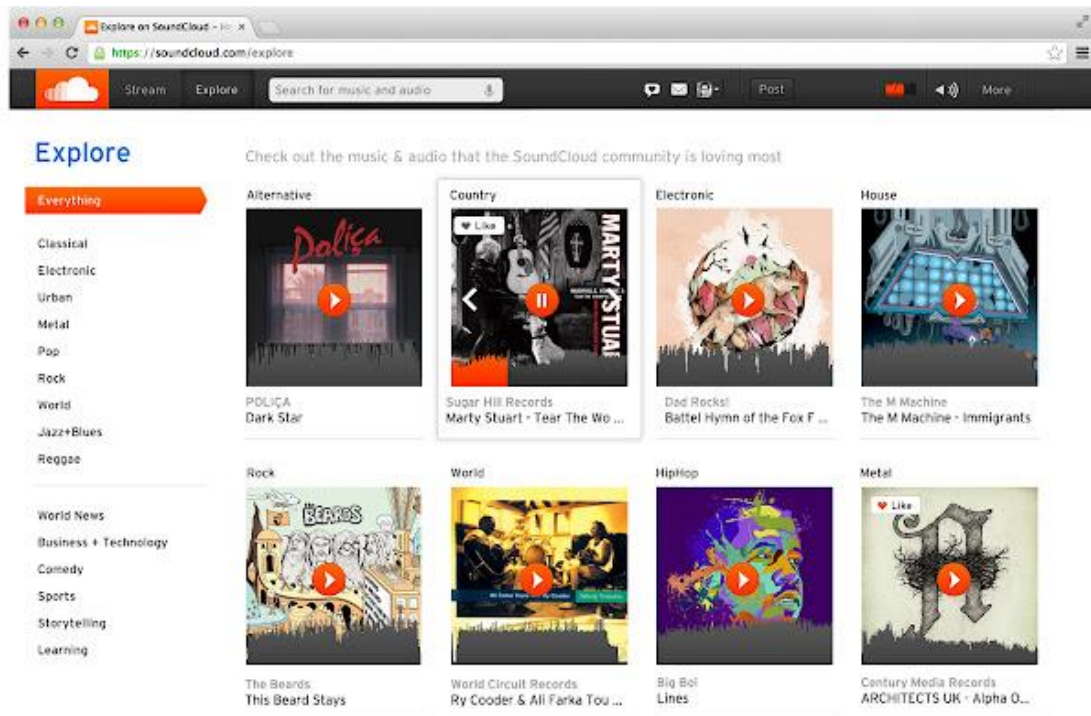


Figure 3

This descriptive qualitative study was to examine closely how SoundCloud was used to facilitate students in sharing their book review in order to let the other students listened and left comments. Here, the students completed sequence procedures in accomplishing the task. It began from preparing, selecting and reading their favorite books. They read for two weeks and prepared some notes, which were related to the content of the book. Once they finished, they recorded the review and posted on their SoundCloud.

The internet provides unrestricted access to information, expertise, and the creation of virtual environments. It allows individuals to interact and socialise with one another anytime and wherever they choose (Alwia, Mahirb, & Ismaila, 2014). Social media has recently been the center of attention for many individuals all around the world. People may connect with one another without any boundaries because to the fantastic features and services.

### Review on Apple Music:

On iPod Touches, iPhones, and iPads, the Apple Music app is the default music player. It has parts of Apple's previous free iTunes Radio service (which is still available and free), as well as iTunes Match (which is still troublesome), and even a hint to Ping, the company's unsuccessful social music effort that ended in 2012. It also incorporates Beats Music, the

streaming service it bought in 2014 along with the rest of the Beats headphone company. Apple Music, however, falls short on navigation and user experience with all of those services crammed into one app, which is surprising considering Apple's long-standing commitment on outstanding design. The service's integration of users' own music libraries also has several serious flaws.

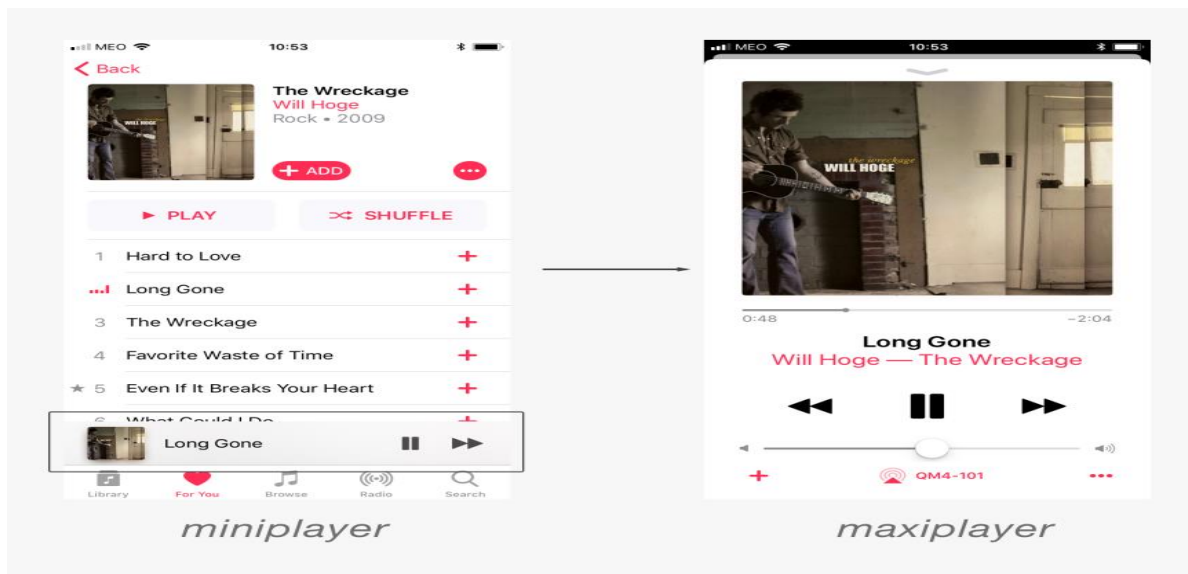


Figure 4

These difficulties prevent Apple Music from being a clear Spotify competitor at launch. Apple Music, on the other hand, is well worth a try thanks to its generous 3-month free trial period. Because Apple Music is software rather than hardware, we anticipate to see tweaks, corrections, and potential enhancements in the future.

If you're coming from the old iTunes Music app, you'll notice that the new Apple Music app looks chevalier. Instead of tabs for artists, songs, albums, and radio, Apple Music is categorized by the kind of music you wish to listen to: recommended music, new music, live or Internet radio, or your purchased tunes. The app's bottom-level tabs make it easy to navigate between sections.

# Chapter 03 - System Development

The project has been made keeping in mind the needs of students and music enthusiasts. For the same reason we decided to keep our app completely free to use. Additionally, every single piece of music uploaded to and downloaded from our web app is completely royalty free and non-copyright. Therefore, students can use it for their college projects and download them for a seamless offline experience without worrying about copyright infringement. With the rising cases of creative copyright infringements, our web app solves a major problem for college students. However, keeping in mind the needs for expansion, we might require a paid server for hosting. Currently, we have planned two revenue models:

- 1) Ad revenue:** We will integrate Google Ads with our web app, thereby getting a continuous income. This will help us keep this service free and accessible to everyone.
- 2) Featured Tracks:** Users will be able to get their uploaded tracks to move to a featured list for a meagre amount. The featured list will be an entirely different section and serve as a promoted page. It has not been designed yet, but can easily be integrated with our current structure.

## Technical Feasibility

The project takes advantage of the latest versions of popular open source frameworks and libraries with tremendous community support. In our testing, we were unable to find any critical bugs that hinder the user experience. However, we'll continue to develop the web app further and add more features with time. The entire project is properly documented and has been written following the standard coding practice

## Legal Feasibility

The aim of the project is to provide good quality royalty free music to college students and freelancers. Being aware of the stringent creative copyright laws in place, we have meticulously designed the privacy policy of our web app. Our privacy policy is present on the sign-up page, it clearly states that we'll not be liable to any legal action in case of any creative commons feud. The users have to agree to the privacy policy to take the full



advantage of our web app.

## Requirements

### Functional Requirements:

The web app can be majorly divided into the following :

- **User Authentication:** The sign-up functionality takes advantage of custom authentication. The user's email and passwords are stored in the admin table of our PostgreSQL database.
- **Streaming:** The app has an integrated music player with a basic play, pause and skip functionality.
- **Filter:** The customised filter is present on the user dashboard where the user can filter through categories like pop and rock.
- **Upload Music:** The web app has a basic uploader that takes basic audio format files like mp3, aac etc. as input.
- **Download Music:** The users will be able to download the tracks in major audio formats.
- **Accessibility:** The project has been hosted on Vercel at <https://final-project2021.vercel.app>

### Non-Functional Requirements:

The web app has the following non-functional requirements:

- 1) **Security:** User data is stored securely in the database. The passwords are encrypted using B-crypt.
- 2) **Performance:** The app has been properly structured and no performance bottlenecks. The app received a performance score of 91 from Google Lighthouse. In our functional testing, we concluded that our web app does not cause any network or cpu throttling issues..
- 3) **Portability:** The web app has been built keeping responsiveness in mind, therefore, it can easily be run on mobile devices.
- 4) **Scalability:** The web app is completely scalable. A simple upgrade to paid

servers will add the ability to interact with thousands of users at the same time.

- 5) **Network:** The api deployed on Heroku Dyno can handle 5 simultaneous requests with its free version.

The most popular music streaming platforms like Spotify and Youtube seem to have a major problem, it is that you have to be a premium user to get an ad-free experience. Consequently, it gets difficult for students and freelancers to upload or download music because of copyright issues. Therefore, we have seen a lot of times that YouTube videos have muted audio whenever there is a track from another artist. The copyright strikes can even lead to the termination of their channel. So, the first problem we are trying to solve is to provide royalty-free non-copyright music to students and freelancers.

Also, at the start of their journey, musicians are not sure about the taste of the audience. There is no way to test if your creation is loved by one and all. This is another problem we are trying to solve. Musicians can upload their music to our web app and according to the trending page, they can get a decent idea if their music is making people move. They will become more aware and connected to their audience and have a solid way for feedback.

After starting the journey, most musicians find it difficult to scale. Their tracks might be very good but people haven't heard of them. When people upload their music on our web app, users can share it across platforms and use them in their freelance projects or YouTube videos hassle-free. We can all agree that something that is better than marketing is free-marketing.

However, our USP is that we have centralized the needs of thousands of people. All the creators, freelancers and artists require audio tracks for their projects but finding the right music for your project is as difficult as finding a needle in a haystack. There are millions of tracks available on the internet. YouTube has thousands of NCR sounds but you cannot specifically search and filter through them according to their needs. Even if you find the right one, sometimes they are copyrighted and your mistake can cost you the entire project. Our web app is run by the community, for the community. There are no hidden copyrights,

the users explicitly agree to our privacy policy which states that by uploading the track on the website, you grant permission to everyone to use a part of the entirety of this track. So, no copyright infringements will be coming to bite you after your project goes live.

Lastly, the download feature is present on all the streaming platforms but what's unique about ours is that the track is entirely yours. While other popular services require you to be logged in and use the app while you're listening to your downloaded music offline, our web app allows you to download once and use it anywhere, on your mobile phone, your laptops, your iPod etc. The problems prevalent in the current platforms that we are aiming to solve are listed below:

1. **Cost effective:** All the music streaming platforms require premium subscription for an ad-free experience and additional features. Our web app just requires you to login to unlock all the features.
2. **Upload Music:** Streaming platforms require you to have a developer account and take certain fees for hosting your music whereas with our web app you can just login and start uploading tracks free of cost.
3. **Download music:** Most of the music streaming platforms require you to be a premium member to download music and limit the usage of offline music to their app, our web app lets you download royalty free music without any cost and limits.
4. **Royalty free music:** We provide royalty free music so the audio files are not copyrighted. The creator at any point of time can change the album art, track name and category. Our web app helps users forget about copyright infringement and royalties.

# Algorithm of the problems of the Project:

## *1. Signup-*

1. Start
2. Enter Username
3. Enter Email id
4. Enter Password
5. Confirm Password
6. Click on Create account
7. If(parameters\_filled\_correctly) "Username Added"  
Else  
"Invalid Email"
8. End

## *2. Login-*

1. Start
2. Enter Email Id
3. Enter Password
4. If (user\_found) "You are logged in"  
Else  
"User Not Found" .
5. End

## *3. Upload-*

1. Click on Upload Music
2. Select a valid audio track
3. Click on upload
4. If (file\_size <= 5 Mb) "File Uploaded to Database:  
Else  
"Something went wrong"
5. End

**4. Filter-**

1. Select multiple categories
2. Click on search
3. Finds relevant tracks with selected categories and returns them on a new page.
4. End

**5. Trending Page-**

1. Find the download counts of each track from the database.
2. Sort them according to the highest number

End

# TECHNOLOGY:

## What is Node.js?

Node.js is a server-side platform based on the JavaScript Engine in Google Chrome (V8 Engine). Ryan Dahl created Node.js in 2009, and the most recent version is v0.10.36. The official documentation for Node.js provides the following definition:

Node.js is a framework for quickly creating fast and scalable network applications based on Chrome's JavaScript engine. Node.js is lightweight and efficient because to its event-driven, non-blocking I/O approach, which is ideal for data-intensive real-time applications that operate across multiple devices.

Node.js is a cross-platform runtime environment for creating server-side and networking applications that is free source. Node.js apps are written in JavaScript and run on OS X, Microsoft Windows, and Linux using the Node.js runtime.

Node.js also comes with a large library of JavaScript modules, which greatly facilitates the creation of web applications with it.

## Features of Node.js:

The features that make Node.js the primary choice of software architects are listed below.

**Event-driven and asynchronous** The Node.js library's APIs are all asynchronous, or non-blocking. It basically implies that a Node.js server never waits for data from an API. After accessing an API, the server moves on to the next one, and a notification system in Node.js called Events assists the server in receiving a response from the previous API request.

Node.js library is highly quick in code execution since it is built on Google Chrome's V8 JavaScript Engine.

Node.js employs a single threaded paradigm with event looping, making it very scalable. In contrast to typical servers that establish restricted threads to process requests, the event mechanism allows the server to reply in a non-blocking manner and

making it more scalable. Node.js makes use of a single threaded application that can handle a considerably higher number of requests than traditional servers like Apache HTTP Server.

There is no data buffering in Node.js apps. The data is simply produced in chunks by these apps.

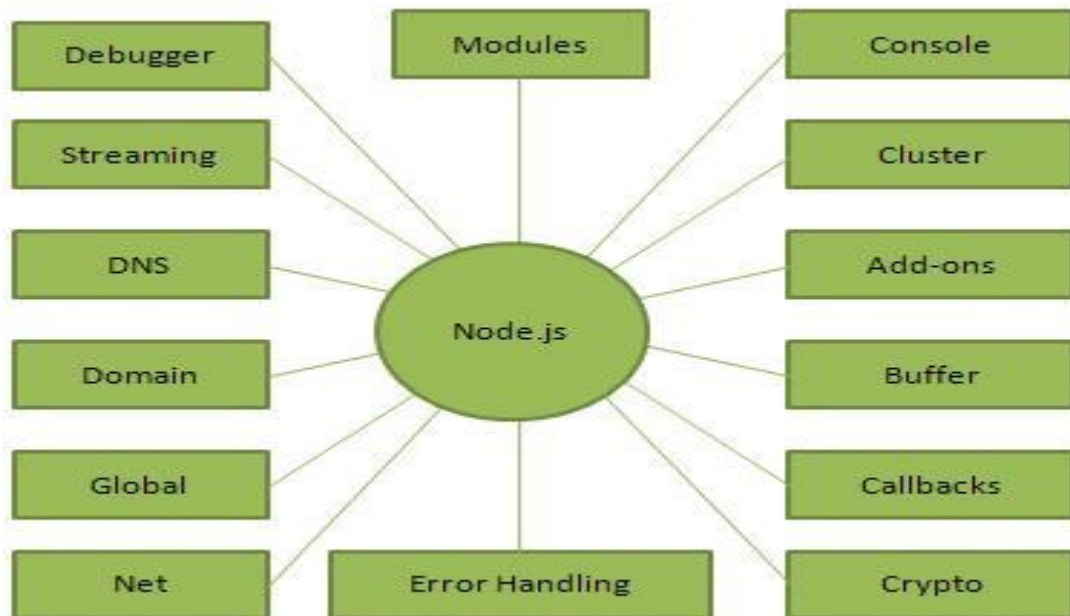


Figure 5

## What is Express?

Express is a lightweight framework that runs on top of Node.js' web server capability, simplifying the APIs and adding useful additional features. It helps you structure your application's functionality using middleware and routing; it offers useful utilities to Node.js' HTTP objects; and it makes dynamic HTTP objects easier to render.

Express is a component of the MEAN stack, a JavaScript full stack solution for developing quick, dependable, and maintainable production online applications.

MongoDB (Database)

ExpressJS is a server-side scripting language (Web Framework)

AngularJS is a web application framework (Front-end Framework)

## NodeJS (Application Server)

The most common minimalistic framework is Express. It is based on NodeJS's built-in HTTP module, which allows for simple and straightforward interaction between frontend and backend logic through API, as well as the organisation of business logic in a pleasing manner. It's a lot more versatile, and we can use it for both web and android. It also has a straightforward error handling technique.

The following are the basic stages to building an express app. Setting up the environment with module installation, developing an application, starting the web server, and performing basic interactions with the server are all covered here. Must have a fundamental understanding of how to utilise the node package manager for simple tasks, a basic understanding of how to install dependencies and modules using the terminal, a basic understanding of how a web application works, and a strong understanding of ES6.

## Introducing Node:

Node (or Node.js) is an open-source, cross-platform runtime environment that allows JavaScript developers to construct a variety of server-side tools and apps. The runtime is designed to be used outside of a browser (i.e. running directly on a computer or server OS). As a result, the environment drops support for browser-specific JavaScript APIs in favour of more standard OS APIs like HTTP and file system libraries.

Node provides a variety of advantages in terms of web server development:

- Outstanding performance! Node was created to improve online application performance and scalability, and it's a solid answer for many typical web-development issues (e.g. real-time web applications).
- Because code is written in "plain old JavaScript," you'll spend less time dealing with "context shift" between languages while building both client-side and server-side code (e.g. Python, PHP, etc.) TypeScript, Coffee Script, Clojure Script, Scala, Live Script, and many more new and popular languages compile/convert to JavaScript, so you may use them as well.



- The node package manager (NPM) gives you access to thousands of reusable packages. It also offers the greatest dependency resolution in the industry and can automate the majority of the build process.
- Node.js is lightweight. Microsoft Windows, macOS, Linux, Solaris, FreeBSD, OpenBSD, WebOS, and Nonstops OS are all supported. Furthermore, many web hosting companies support it, frequently providing dedicated infrastructure and documentation for hosting Node sites.
- It features a vibrant third-party ecosystem and development community, with many individuals eager to assist.

## Introducing Express:

The most popular Node web framework is Express, which also serves as the foundation for a number of other prominent Node web frameworks. • Write handlers for requests with different HTTP verbs at different URL paths using the techniques provided (routes).

- Integrate with "view" rendering engines to provide replies by populating templates with data.
- Configure typical web application settings such as the connection port and the location of templates used to produce the answer.
- At any step in the request handling pipeline, add extra request processing "middleware."

Despite the fact that Express is a rather simple framework, developers have written suitable middleware packages to solve practically any web development issue. Cookies, sessions, user logins, URL parameters, POST data, security headers, and many more topics are covered by libraries. At Express Middleware, you may discover a list of middleware packages maintained by the Express team (along with a list of some popular 3rd party packages).

## What is PostgreSQL?

PostgreSQL is an open source database management solution for businesses. For flexibility and SQL compliance, it supports both SQL and JSON for relational and non-relational queries. Advanced data types and performance optimization features that are only accessible in pricey commercial databases like Oracle and SQL Server are supported by PostgreSQL. Postgres is another name for it.

Here's a quick rundown of PostgreSQL's history:

- INGRES was created in 1977
- Michael Stonebraker and his colleagues created Postgres in 1986
- Support for real ACID and PL/pgSQL was added in 1990
- Postgres95 was released in 1995
- Postgres95 was re-released as PostgreSQL 6.0 in 1996
- MVCC, GUC, Join syntax Controls, and Procedural Language Loader were added in 1998-2001
- Version 7.2 to 8.2: Included features like Schema support,
- PostgreSQL 8.4 was released in 2009; PostgreSQL 9.0 was released in 2010; NYCPUG (New York City PostgreSQL User Group) became a member of PgUS (United States PostgreSQL Association) in 2013; PGconf was held in 2014.

## Advantages of PostgreSQL:

The following are the primary PostgreSQL advantages and benefits:

- As a LAMP stack alternative, PostgreSQL can operate dynamic webpages and web apps.
- Write-ahead logging in PostgreSQL makes it a highly fault-tolerant database.
- The source code for PostgreSQL is open source and publicly available. This gives you the flexibility to utilise, alter, and apply it as needed for your organisation.
- Geographic objects are supported by PostgreSQL, allowing it to be used for location-based services and geographic information systems.
- PostgreSQL may be used as a geographical data store for location-based applications and geographic information systems since it supports geographic objects.
- Postgres requires less training because it is simple to use.

- PostgreSQL is easy to maintain and administer in both embedded and corporate environments.

## Disadvantages of PostgreSQL:

The following are PostgreSQL's drawbacks and limitations:

Postgres is not owned by a single company. Despite being fully featured and comparable to other DBMS systems, it has had difficulty getting its name out there.

Because PostgreSQL takes more work for performance improvements than MySQL since it emphasises on compatibility.

Many open source programmes support MySQL, while PostgreSQL may not.

It is slower than MySQL in terms of performance.

## Applications of PostgreSQL:

Following are the popular applications of PostgreSQL:

### **Financial Services**

PostgreSQL is an excellent database management system for the financial sector. Furthermore, it is completely ACID compliant, making it an excellent alternative for OLTP (Online Transaction Processing). It also has the ability to do database analytics. It is compatible with mathematical applications such as Matlab and R.

### **Government GIS information**

#### **PostGIS**

PostgreSQL has a robust GIS named "PostGIS." Hundreds of functions for processing geometric data in various forms are included in this module. PostGIS adheres to several industry standards. Furthermore, the Open-Source community provides the simplest technique to manage Geodata by using both QGIS and GeoServer.

## **Manufacturing**

PostgreSQL is being used by industrial businesses to speed up their entire business processes. Using this open-source DBMS as a storage backend also helps businesses enhance supply chain performance. It enables businesses to lower their operating costs.

## **NoSQL and web technologies**

Scalability is an important consideration if your website must handle hundreds or even thousands of requests per second. Postgre is the best option in this case.

Django, Node.js, Hibernate, PHP, and other current web frameworks are compatible with PostgreSQL. It also has replication features, allowing you to expand out as many database servers as needed.

## **Scientific evidence**

If you're working on a study or scientific endeavour, you'll need to create terabytes of data. As a result, it's critical to manage things as efficiently as possible. PostgreSQL has excellent analytical capabilities and a robust SQL engine for this. This allows you to easily handle a vast volume of data.

## **MySQL vs PostgreSQL**

- The fundamental distinction between MySQL and PostgreSQL is as follows:
- MySQL has released its source code under the GNU General Public License and other private agreements.
- The PostgreSQL License governs the distribution of PostgreSQL.
- Oracle Corporation currently owns it and provides many premium editions.
- It's open-source and free software.
- As a result, you will never have to pay for this service.
- Only when utilising the NDB and InnoDB Cluster Storage engines is MySQL ACID compliant.

- PostgreSQL complies with all ACID requirements.
- MySQL does excellent in OLAP and OLTP systems where read performance is the most essential factor.
- PostgreSQL performs well in systems that require complicated queries to be executed.

## Introduction to HTML

HyperText Markup Language stands for HyperText Markup Language. It's a markup language that's used to create web pages. HTML is a markup language that combines hypertext with markup. The link between web pages is defined as hypertext. The text document within the tag that specifies the structure of web pages is defined using a markup language. This language is used to annotate (add notes to) text so that a computer can comprehend it and alter it appropriately. Human-readable markup languages include HTML and others. Tags are used in the language to specify what type of text processing is required.

HTML is a markup language that the browser uses to transform text, pictures, and other material so that it may be displayed in the desired format.

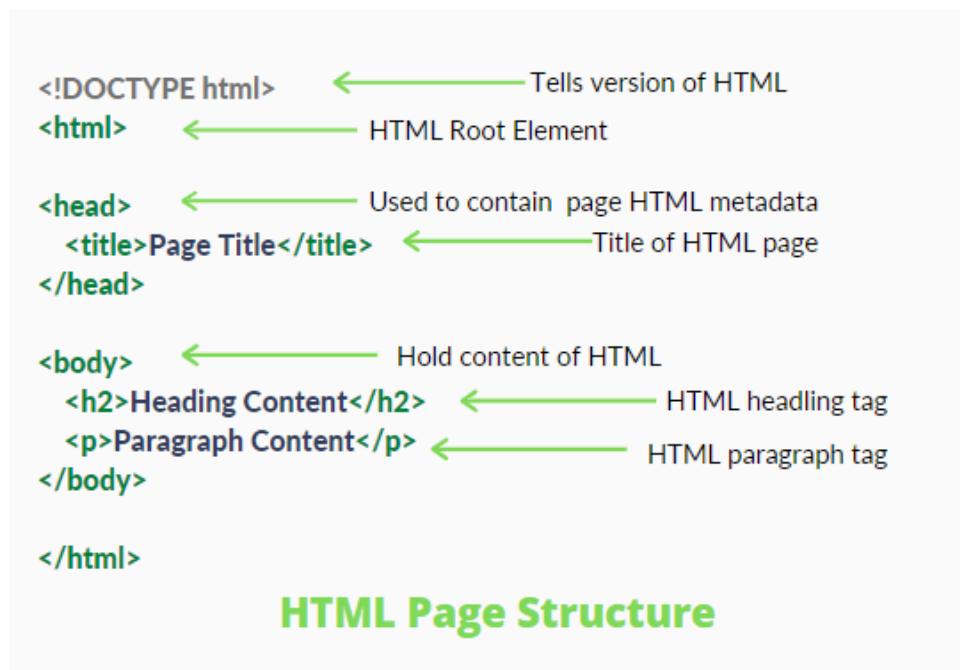


Figure 6

HTML has the following advantages:

- It is simple to learn and use.
- It is platform agnostic.
- A web page can include images, videos, and audio.
- The text can include hypertext.
- It's a type of markup language.

Why bother learning HTML?

- It's a straightforward markup language. It is simple to implement.
- It's used to build websites.
- Assists with the development of web programming principles.
- Advance your professional career.
- HTML is commonly used to create webpages.
- It is compatible with all browsers.
- It can work with other languages such as CSS and JavaScript.
- HTML is only capable of creating static web pages. Other languages must be used for dynamic web sites.
- To develop a simple web page, a considerable amount of code must be written.
- The security feature is insufficient.

CSS, or Cascading Style Sheets, is a basic design language designed to make the process of creating web pages presentable easier.

The style and feel of a web page is handled by CSS. You may use CSS to change the text colour, font style, paragraph spacing, how columns are scaled and laid out, what background pictures or colours are used, layout designs, display variants for different devices and screen sizes, and a range of other effects.

CSS is simple to learn and comprehend, but it gives you a lot of power over how an HTML document looks. CSS is frequently used in conjunction with the markup languages HTML or XHTML.

## Advantages of CSS

- CSS saves time by allowing you to develop CSS once and then reuse it across various HTML pages. Each HTML element has its own style that you may apply to as many Web pages as you desire.
- Pages load quicker since you don't have to create HTML tag attributes every time you use CSS. Simply define a single CSS rule for a tag and apply it to all instances of that tag. As a result, less code implies quicker downloads.
- Nicer styles than HTML CSS provides a far larger set of characteristics than HTML, so you may give your HTML page a lot better appearance than HTML attributes.
- Multi-Device Compatibility Style sheets enable content to be optimised for several devices. Different versions of a website can be displayed for portable devices such as PDAs and mobile phones, or for printing, utilising the same HTML content.
- International web standards HTML attributes are now deprecated, and CSS is suggested instead. To make HTML sites compatible with future browsers, it's a good idea to start incorporating CSS in all of them.

## CSS is created and maintained by who?

CSS is developed and maintained by the CSS Working Group of the World Wide Web Consortium. The CSS Working Group produces specification papers. A specification becomes a recommendation when it has been reviewed and officially ratified by W3C members.

Because the W3C has no influence over how the language is implemented, these certified standards are referred to as recommendations. That software is created by independent businesses and groups.

JavaScript is a scripting language that is lightweight, cross-platform, and interpreted. It is well-known for web page creation, but it is widely used in numerous non-browser applications. JavaScript may be used to create both client-side and server-side applications. A standard library of objects, including as Array, Date, and Math, as well as a basic set of language components, such as operators, control structures, and statements, are included in JavaScript.

- Client-side: It provides objects for managing a browser's Document Object Model (DOM). Client-side extensions, for example, might allow an application to add items on an HTML form and respond to user actions like mouse clicks, form input, and page navigation. AngularJS, ReactJS, VueJS, and many more client-side frameworks are useful.

- Server-side: It provides objects that are required to run JavaScript on a server. For example, if server-side extensions allow an application to interface with a database and give information continuity from one programme invocation to the next, or do server-side file operations. Node.js is a handy framework that is currently the most popular.

There are two ways to include JavaScript in your HTML file:

- Internal JS: We may write the code for JavaScript straight into our HTML file.

JavaScript was conceived in 1995 by Brendan Eich while working at Netscape as a developer. It was initially called LiveScript, but it was changed. The JavaScript programming language, unlike most others, has no concept of input or output. It is intended to be used as a scripting language in a host environment, and the host environment is responsible for providing communication methods with the outside world. The browser is the most popular host environment.

JavaScript features include: According to a recent Stack Overflow poll, JavaScript is the most used programming language on the planet.

JavaScript is capable of so much more now that browser technology has advanced and JavaScript has migrated into the server with Node.js and other frameworks.

Here are a few things that we can do with JavaScript:

JavaScript has the following limitations:



- **Performance:** JavaScript does not have the same degree of performance as many older languages, therefore a complicated application developed in JavaScript would be sluggish. However, because JavaScript is used to accomplish simple operations in a browser, speed is not a major concern.
- **Complexity:** In order to learn a scripting language, programmers must have a complete understanding of all programming ideas, basic language objects, client and server-side objects; otherwise, writing complex scripts in JavaScript will be challenging.
- **Weak error handling and type checking:** It is a weakly typed language since the variable's data type does not need to be specified. As a result, compile does not conduct any type checking.

JavaScript's uses include:

- **Web development:** incorporating interactivity and behaviour into static websites In 1995, JavaScript was created to accomplish this. That is really simple to implement using AngularJS.
- **Online Apps:** As browser technology has advanced, a language is now necessary to construct effective web applications. We merely need to click and drag the mouse to explore a map on Google Maps. All detailed views are accessible with a single click, thanks to JavaScript. It makes use of Application Programming Interfaces (APIs), which provide the code more capability. This is when Electron and React come in handy.
- **Server Applications:** JavaScript made its way from client to server with the aid of Node.js, and Node.js is the most powerful on the server side.
- **Games:** In addition to webpages, JavaScript can be used to create fun games. JavaScript is very popular in game creation due to its mix of JavaScript and HTML 5. It includes the EaseJS package, which allows you to deal with rich visuals.
- **Smartwatches:** JavaScript is utilised in a wide variety of devices and apps. It offers the PebbleJS library, which is utilised in wearable apps. This framework is useful for apps that rely on the internet to function.

- **Art:** Artists and designers may draw on an HTML 5 canvas using JavaScript, and the p5.js package can be used to make sound more effective.
- **Machine Learning:** This JavaScript ml5.js library may be used for machine learning in web development.

React-based framework Next.js. It offers the ability to create stunning Web applications for a variety of platforms, including Windows, Linux, and Mac. If you have limited experience with react and want to learn more about the ecosystem, you should be familiar with the Next.js framework.

Let's start with a quick overview of Next.js.

**Introduction:** React, webpack, and babel are all used in Next.js. It's a fantastic tool for building online applications, and it's well-known for server-side rendering. Zeit is the creator of Next.js. Next.js is simple to learn and switch for developers who are familiar with HTML, CSS, Java Script, and React.

**Characteristics:**

- **Hot Code Reloading:** When changes in the code are saved, the application is instantly reloaded.
- **Automatic Code Splitting:** This functionality bundles and serves each page with every import in the code. It signifies that no extra code is put onto the website.
- **Ecosystem Compatibility:** JavaScript, Node, and react are all supported.
- **Rendering on the Server:** Render react components on the server before delivering HTML to the client.
- **Styled-JSX:** With Styled-JSX, you can create CSS right within JavaScript code.

## Music classification and machine learning

### Librosa's Feature Extraction

When I initially started researching music information retrieval, the process of retrieving information about music and its aural content via audio signal processing, it appeared like a difficult endeavour that would need a high level of technical

knowledge. It absolutely does; in fact, there is an entire discipline dedicated to the endeavour. Fortunately, the makers of the Librosa Python package have made this procedure a lot easier. This module allows Python users to quickly extract information from any mp3 file they feed it.

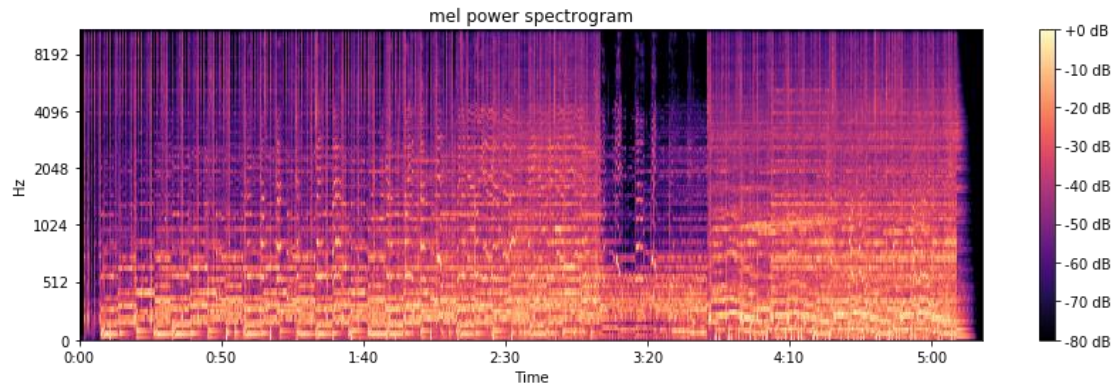


Figure 7

## Use-Case Diagram:

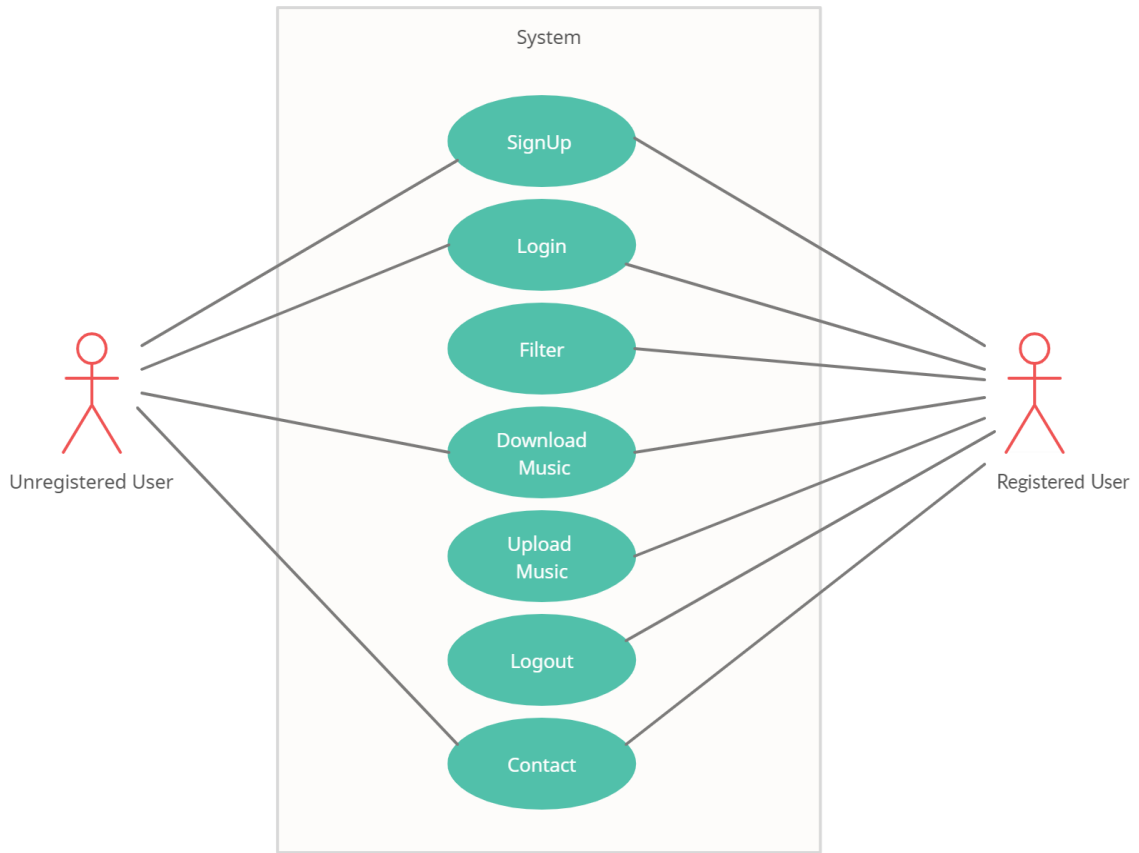


Figure 8

## Data Flow Diagram:

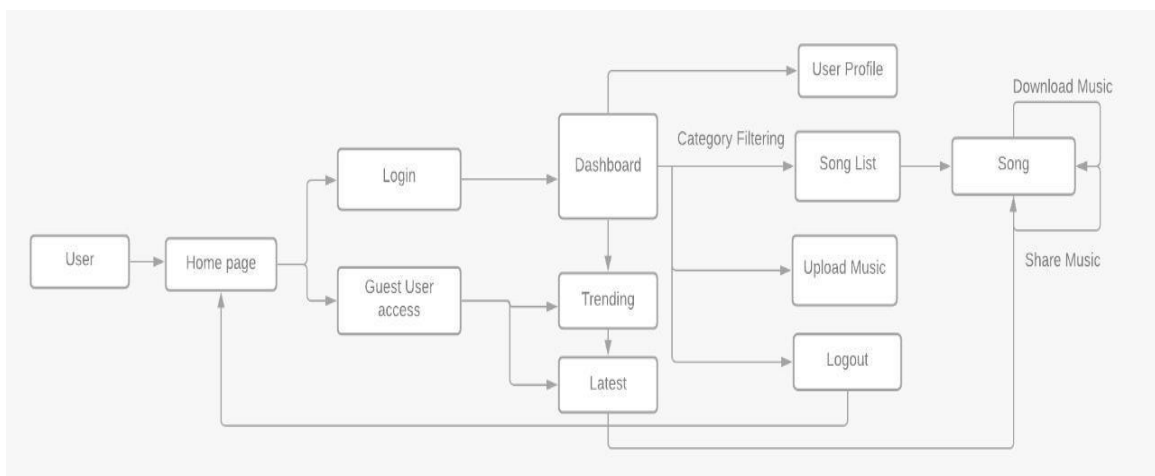


Figure 9

# Chapter 4 – Performance Analysis

## Critical Remarks of previous works

Application	Strength	Weaknesses
YY Music	Able to download music on YouTube	Annoying ads keep pop up during using
Xiami Music	Lightweight application and emphasize the experience of listening to music	Lack of hierarchy in some user interface
JOOX	With useful features such as My Karaoke	A large number of features require VIP privileges

**Table 2-2-1-1 Strength and Weaknesses of reviewed application**

Figure 10

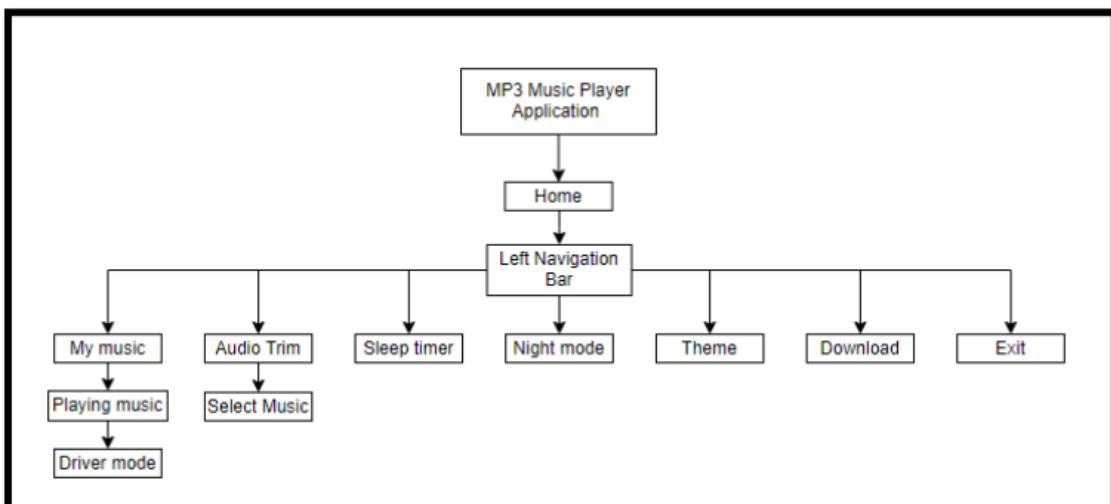


Figure 11

Use case name: Listen to music	ID: UC001
Actor: User	
Description: User able listen to music on this application	
Trigger: 1. User select any song from the playlist 2. User click the play button to start playing the music	
Precondition: 1. The user should import the song from their mobile device into the music player 2. The user should add the song into a playlist by search music online feature	
Normal flow of events: 1. User select a song from the playlist 2. The song playing until the end	
Alternate / Exceptional flows: 1a. The application will display a message "No local music" if the playlist does not have any song	

Figure 12

Use case name: Progress bar	ID: UC002
Actor: User	
Description: The progress bar shows the progress of the song	
Trigger: 1. The progress bar becomes active when the user starts playing the song	
Precondition: 1. The song must in playing status	
Normal flow of events: 1. Song is playing 2. The progress bar will show the duration of the song 3. The progress bar will be finished if the song is finished playing	
Alternate / Exceptional flows: 1a. The progress bar will reset if user play other song 2a. If the user drags the progress bar to left or right, the song's progress will changes 3a. The progress bar will reset if finished playing	

Figure 13

## Model-Based Testing for Application Analysis

Figure depicts our strategy to analysing mobile apps using model-based testing, which is separated into two primary steps: (a) test case development and execution, and (b) runtime verification activities. Executable test cases will be produced automatically in the first step using a model of probable user interactions with the application under test. After that, each test case will be run on a real mobile device. The execution of test cases will be examined in the second step to see if they behave appropriately according to user-defined attributes.

User behaviour modelling, test case creation, and test case execution are the three substeps of the first stage. The user provides a model of the application (or apps) to be examined in the first substep. The model is built from the perspective of a user engaging with the

software. Instead of producing random inputs, modelling user behaviour is critical for developing test cases that correlate to genuine applications of the programme. The capacity for automatic exploration of Spin model checker is used to generate test cases from this model in the following substep.

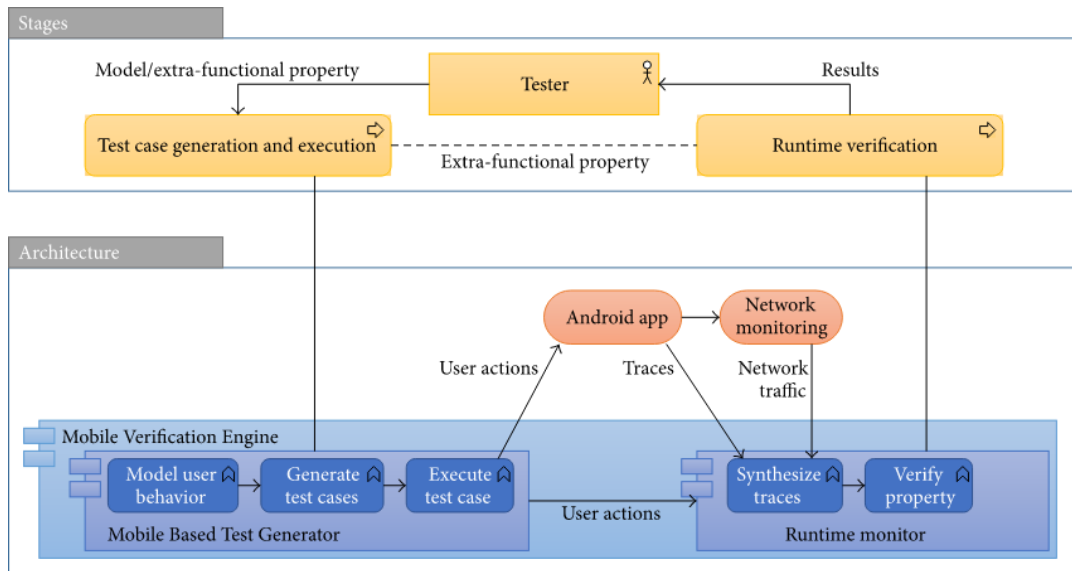


Figure 14

In the second stage, the test case executions will be analyzed according to properties given by the user. This stage is divided into two substeps: trace synthesis and property verification. A trace is an ordered sequence of states, that is, data gathered from the test case execution at different time instants. The trace is built from data provided by multiple sources at runtime, such as the actions performed by the user, energy consumption measured by an external power analyzer, or network traffic statistics. In the last substep, each of these traces is analyzed at runtime by the Spin model checker to check the extra-functional properties given by the user; for example, in our case, to check whether network traffic follows a given expected pattern. The main components of this architecture are further detailed in the following sections.

### Analysis of Network Traffic

After we've created the test cases, we can go on to the network traffic analysis we want to do on the Spotify app. We started by looking at the app's network traffic. We discovered that it downloads music from Spotify servers using HTTP connections and utilises the User-Agent field to identify itself as "Spotify-Unknown." While Spotify utilises HTTP for a

variety of applications, GET requests with the syntax "/audio/id>" might be used to identify song requests. Each music is downloaded in segments via a sequence of HTTP GET requests delivered immediately after the user pushes play. Only network data packets from these audio streams are considered in the following discussion.

This is the known mean value of the distribution of  $\mu$ -measurements for all songs that display typical behaviour, and  $\sigma$  is the standard deviation of the same distribution. The interval is specified as the time between the user action that starts playing a song in the test case and the user action that starts a new song title search, which marks the conclusion of the test scenario. The interval formula in (2) checks whether the continuous variable rate of a given network traffic variable falls between a band around the mean in a normal distribution and a width of two standard deviations, i.e., a significant confidence interval of 95.45% of the songs that exhibit typical behaviour. The runtime monitor may identify and supply these events and variables; in particular, these measurements can be taken at regular intervals using the tcpdump programme remotely on an Android mobile and so included in the enhanced execution trace.

## Model-Based Testing for Application Analysis

Our model-based approach to mobile application analysis

Based testing may be separated into two basic parts, as shown in Figure: (a) test case creation and execution, and (b) runtime verification activities. Executable test cases will be produced automatically in the first step using a model of probable user interactions with the application under test. Each created test case will be run on a real mobile device. The execution of test cases will be examined in the second step to see if they behave appropriately according to user-defined attributes.

User behaviour modelling, test case creation, and test case execution are the three substeps of the first stage. The user provides a model of the application (or apps) to be examined in the first substep.

The model is built from the perspective of a user who interacts with the programme. Instead of producing random inputs, modelling user behaviour is critical for developing test cases that correlate to genuine applications of the programme. The capacity for autonomous exploration of Spin model checker is used in the following substep to automatically build



test cases from this model. Each test case generated by the model checker is then transformed into Android-compatible Java code. These test cases are run one by one in an actual mobile device in the final stage.

# CHAPTER 5 – CONCLUSION

## Discussion on previous Results achieved:

The advantages of having a website from which you can upload and download royalty free music are truly limitless. It creates a sense of excitement and enthusiasm in current age music creators to show their work to the world and learn from their audience and other fellow creators.

At the time of writing this report, we had 30 users registered on our application. In addition to that, we had 24 tracks uploaded. In total, tracks were downloaded more than 20 times.

The project is for the students and freelancers who want their projects to be hassle free and take advantage of royalty free, non-copyright music.

Furthermore, the fact that it is completely ad-free engages a user, whether a creator or a non-creator, in exploring new music and new talents from all around the world.

## The Pros:

- Many instances, the sound quality of streaming music will be significantly superior than that of tracks on CDs.
- Out-of-print recordings will be available to subscribers.
- Streaming saves the user's hard disc space
- Customers may choose from a large variety of songs while using streaming services.
- Subscribers may make their own playlists with ease.
- Customers have easier access to new artists and genres of music.
- Customers are informed about new releases from their favourite musicians through streaming services.
- New artists have a higher chance of being discovered
- Subscribers may utilise their playlist to help new releases become success stories.
- New artists have a better opportunity of profiting from their work.

## The Cons:

- Subscribers do not possess a copy of the music they've listened to on the service.
- Big corporations and technology businesses benefit the most from music streaming services.
- Many music streaming services collect information about their users.
- It is impossible to digitally recreate music that has been streamed.
- As a result of streaming, established artists lose out on prospective profits.
- To keep afloat, most music streaming services rely on advertisements.
- "Free" music streaming services sometimes include concealed membership costs.
- Streaming service music catalogues are frequently unorganised.
- Subscribers never see or appreciate the time and effort that goes into making an unique record cover.
- Because established musicians have so much competition, emerging artists will have a tougher difficulty being discovered.
- Music that has been digitally duplicated for streaming reasons is frequently distorted during the remastering process.
- It's possible that new artists may have to wait years to see a return on their investment.

## **Streaming Music Challenges (for the Music Industry):**

- Difficult to store/organize - extensive digital infrastructure is required
- Processing and aggregating micro cent payments from digital service providers is difficult for computers.
- Unauthorized usage is difficult to control and punish due to the fluctuating value of streams
- Artist contracts from the past do not cover this form of revenue.

## **Pros of internet music listening:**

1. You may save a lot of storage space by listening to music online because nothing is kept locally.
2. It also means that no matter where you are in the globe, as long as you have an internet connection, you have immediate access to any information you desire.

3. You receive first access to new releases since content is made accessible digitally long before preorders arrive at your door.
4. It's a lot easier to share online music with friends and family since more people can share an account or have shared access to a service, which is made further easier by features like sharing and embedding.
5. You may listen to as much music as you like without having to buy a whole CD when you listen to music online (on some platforms).

#### Cons

6. Listening to music online frequently entails dealing with some form of promotion, most commonly advertisements, which can be annoying at times.
7. Most music streaming services require a membership in order to avoid the aforementioned advertisements. In certain circumstances, such as Tidal, you will require a membership to access the content at all.
8. You have no access to any of your online saved libraries or music-playing services if you don't have access to the internet.

### **Offline music listening**

#### Pros:

9. The fact that you don't require an internet connection is the primary advantage of listening to music offline. You can listen to your music even if the WiFi is down if you have it stored locally.
10. In an ironic twist, the emergence of all the internet streaming choices, which also provide an "offline" capability, has made listening offline simpler. This permits music to be downloaded and saved locally in case the Internet is unavailable. This is not a novel notion; using a YouTube converter, you can accomplish much the same thing with a YouTube video. You may convert movies to MP3 files using services like <https://ytmp3.com/> by simply copying a URL and hitting a button.
11. Offline audio files can also be transmitted between music-playing devices that aren't connected to the internet, such as music players and some older MP3 players.

12. Don't forget about vinyl. This may be a very unique experience if you have a nice record player."

#### Cons

13. Offline music takes up storage space and is difficult to manage as the collection expands and the amount of open storage space decreases.

14. Offline material may be untagged or improperly tagged, causing your local music player to display incorrect artist, album, or track information.

15. With the music business increasingly going toward streaming, it might be difficult to find a good offline music player that is still regularly updated and has an interface and layout that is up to date.

Sometimes content creators especially the YouTubers face the problem of getting their video removed just because they used a copyrighted song in their video and many other platforms do the same.

So, the main purpose of our application is that users can stream their favourite music and even download it to their devices to use that audio in whichever way they want hassle-free because there are no copyright issues anywhere.

We wanted to keep the app royalty-free and ad free. So, as developers, we are not getting any sponsors or earning from any ads and that is our biggest limitation. As developers, all we care about is to keep our application

ad-free. This project depends on the community and word of mouth to get more and more users to use our web app. For the financial feasibility of the project, we'll eventually have to succumb to the revenue models mentioned [here](#). That would ultimately mean to have our users pay a certain amount and we might become the very problem that we were aiming to solve.

## **FUTURE WORK:**

### **SETTING THE STAGE (ACT I)**

Change, upheaval, and transformation are all speeding rapidly. That is the narrative of M&E in our technologically advanced society. The music business is no different. Consider TikTok, which was one of the biggest stories in the business last year. TikTok was just released a few years ago, following the acquisition of Musical.ly in 2017. As 2019 began, most in the M&E business in the United States rejected TikTok. The majority of TikTok's target Gen Z demographic in the United States did as well. Look at it now, only a year later, as we enter 2020. TikTok is a big new social media force that is no longer limited to music. That's how quickly things happen. Gen Z's current craze is TikTok.

### **THE STATE OF MUSIC TODAY AND WHERE IT'S GOING (ACT – II)**

The music industry's days of despair and gloom are definitely ended as we enter 2020. The total worldwide recorded music ecosystem continues to win big, so happy days are back. Despite the fact that official 2019 data have yet to be released, 2019 is likely to be the fifth consecutive year of double-digit (or near-double-digit) increase following decades of eviscerating declines. Of course, this does not imply that all participants are content. Many artists and labels continue to criticise streaming's growing dominance, despite the fact that it is driving profits even higher. The global recorded music market is predicted to reach \$20-\$21 billion by the end of 2019. According to Goldman Sachs, today's figures will more than treble to \$45 billion by 2030.

#### **The Current State Of Music**

Streaming is the main force in recorded music, and its influence is growing every year. I guess Steve Jobs was wrong when he famously (and perhaps arrogantly) said, "The subscription model of buying music is bankrupt" and "the Second Coming" won't save it. "It arrived just the same," as a Grinch-defying Dr. Seuss (a completely different type of brilliance) put it. Streaming currently accounts for 80% of recorded music income in the United States. In a world where we can now access 60 million songs ad-free for less than \$10 a month, the distinction between Steve Jobs' version of music "ownership"

(downloads) and "rental" (streaming) doesn't really matter.

## Tomorrow's Music Situation

As Goldman Sachs points out, music's new "good old days" will accelerate in the coming years, with worldwide recorded music profits more than tripling in the next ten years.

Increased streaming and globalization are revenue drivers, but so are new technologies and form factors. Our new AI-powered home helpers, such as Alexa and Siri, make our lives more enjoyable. We may just sit on our couches and request the music we want.

Wearables, another quietly significant new force, speeds things even more. Earbuds alone generate billions of dollars in revenue. Apple's wearables division has announced \$10.1 billion in revenue, with Air Pod sales topping the way. Another revolutionary form factor that points the way is Bose Audio Sunglasses. The music business is now cleverly borrowing a page from the playbook of other M&E sectors in terms of rapidly creating new forms of music engagement and revenue. Expect more gamification to invade your music experiences, thanks to its "free to play" business model. Consumers can participate for "free" (supported by ads). A healthy section, on the other hand, will pay spontaneously as they participate. Tencent Music was a pioneer in this field, and others may learn from them. Its auditory universe is littered with virtual tip jars.

We've merely scratched the surface of the Internet's long-promised music utopia, a world where meaningful real-time direct artist-to-fan and fan-to-fan involvement is possible. In the years ahead, more music experiences will focus on music's intrinsically communal and tribal features - on audiences with comparable artist and song interests and enthusiasm. After all, devoted fans will gladly spend practically anything to get closer to their favourite musicians. According to a new poll by Thinkwell, over half of American people aged 18 to 64 are prepared to spend \$1,000 for such an encounter. Fans also want to interact with other fans. Consider Tinder for music enthusiasts in this vein. Use your phones to find like-minded fans in your area. Swipe left or right after that. That is the method taken by one Los Angeles-based startup, which is still in stealth mode.

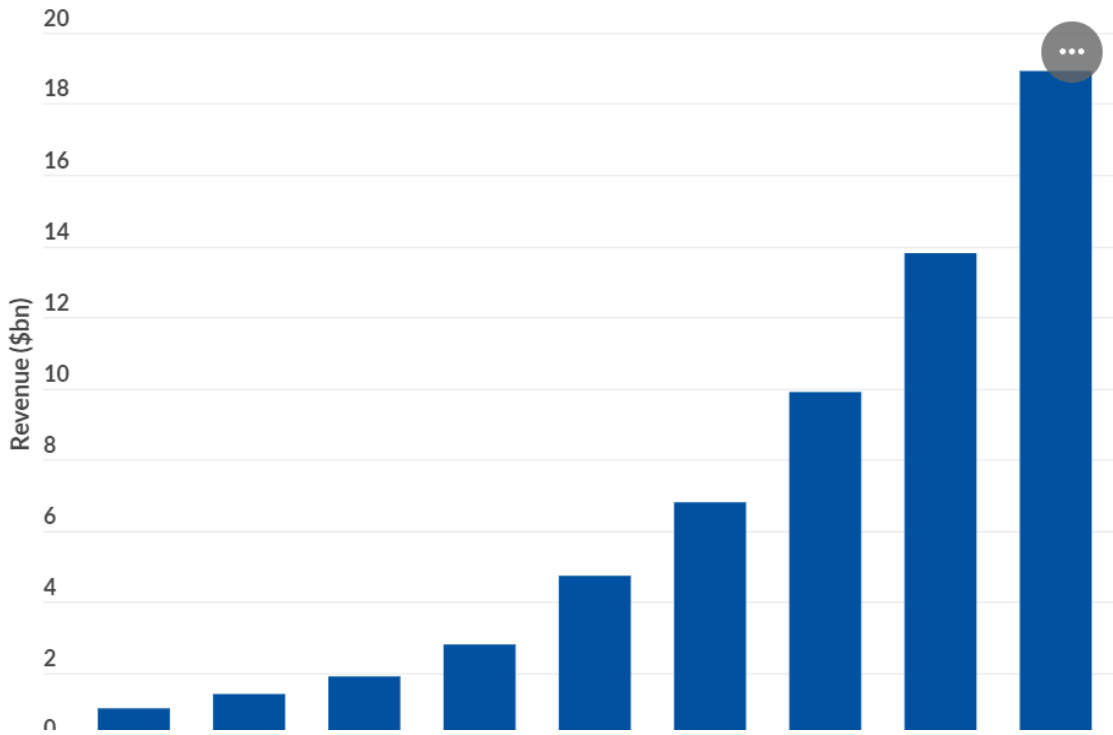


Figure 15



### Global music streaming revenue

Year	Revenue
2012	\$1 billion
2013	\$1.4 billion
2014	\$1.9 billion
2015	\$2.8 billion
2016	\$4.7 billion
2017	\$6.8 billion
2018	\$9.9 billion
2019	\$13.8 billion
2020	\$18.9 billion

### Music streaming subscribers 2020

App	Subscribers
Spotify	165 million
Apple Music	88 million
Amazon Music	70 million
YouTube Premium	50 million
Deezer	7 million
Pandora	6 million
Tidal	3 million

Figure 16

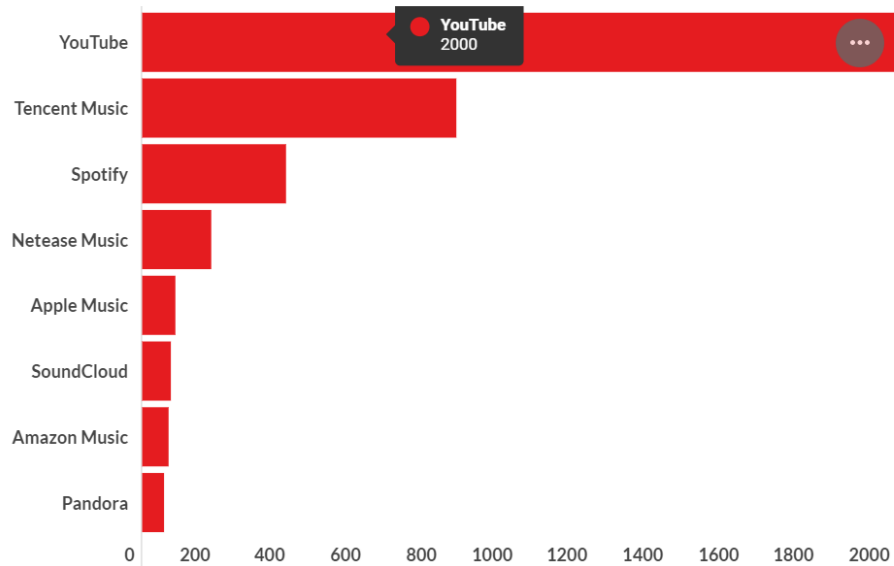


Figure 17

Our future work will include integrating Machine Learning in our app for better playlist and track recommendations based on the user's preference once we have enough user data. We are also planning to build a better User Interface and more cleaner design so the user does not have to go around searching for their favourite music they can create different playlists of their own.

We also aim to add user statistics using which users will be able to look at the number of people currently listening to their songs, and the number of times their track was downloaded.

We are also planning to build a separate review section so the user can give ratings to the music creators and support them by giving them ratings so the music creators can learn more about the music taste of their audience and create music accordingly to engage more with their audience and other music creators.

## References:

1. M. Kamalzadeh, D. Baur, and T. Miller, "A survey on music listening and management behaviours," in Proceedings of the 13th International Symposium on Music Information Retrieval (ISMIR), 2012.

"The Million Musical Tweets Dataset: What Can We Learn From Microblogs," D. Hauger, M. Schedl, A. Kosir, and M. Tkalcic, Proc. of the 14th International Society for Music Information Retrieval Conf. (ISMIR), 2013.

"#nowplaying music dataset: Extracting listening behaviour from twitter," Proc. of the 1st ACM Intl. Workshop on Internet-Scale Multimedia Management, pp. 21-26, 2014. E. Zangerle, M. Piehl, W. Gassler, and G. Specht, "#nowplaying music dataset: Extracting listening behaviour from twitter," Proc. of the 1st ACM Int

"Exploiting twitter's collective knowledge for music recommendations," Proc. of the 2nd Workshop on Making Sense of Microposts (#MSM2012): Big things come in tiny packages, pp. 14-17, E. Zangerle, W. Gassler, and G. Specht, 2012.

M. Schedl and D. Schnitzer, "Location-aware music artist recommendation", Proc. of the 20th Intl. Conf. on MultiMedia Modeling (MMM), 2014.

- M. Piehl, E. Zangerle and G. Specht, "Towards a context-aware music recommendation approach: What is hidden in the playlist name?", *Proc. of the 15th IEEE Intl. Conf. on Data Mining Workshops (ICDM)*, pp. 1360-1365, 2015.
- M. Alghoniemy and A. H. Tewfik, "A network flow model for playlist generation", *Proc. of the Intl. Conf. on Multimedia and Expo (ICME)*, pp. 329-332, 2001.
- J.-J. Aucouturier and F. Pachet, "Scaling up music playlist generation", *Proc. of the Intl. Conf. on Multimedia and Expo (ICME)*, pp. 105-108, 2002.
- S. Pauws and B. Eggen, "PATS: Realization and user evaluation of an automatic playlist generator", *Proc. of the 3rd Intl. Symposium on Music Information Retrieval (ISMIR)*, 2002.
- B. Logan, "Content-Based Playlist Generation: Exploratory Experiments", *Proc. of the 3rd Intl. Symposium on Music Information Retrieval (ISMIR)*, 2002.
- E. Pampalk, T. Pohle and G. Widmer, "Dynamic playlist generation based on skipping behavior", *Proc. of the 6th Intl. Symposium on Music Information retrieval's (ISMIR)*, pp. 634-637, 2005.
- A. Flexer, D. Schnitzer, M. Gasser and G. Widmer, "Playlist generation using start and end songs", *Proc. of the 9th Intl. Symposium on Music Information Retrieval (ISMIR)*, pp. 173-178, 2008.
- M. Slaney and W. White, "Measuring playlist diversity for recommendation systems", *Proc. of the 1st ACM workshop on Audio and music computing multimedia*, pp. 77-82, 2006.
- Qs. J. Cunningham, M. Jones and S. Jones, "Organizing digital music for use: an examination of personal music collections", *Proc. of the 5th Intl. Symposium on Music Information Retrieval (ISMIR)*, 2004.
- M. Piehl, E. Zangerle and G. Specht, "Combining Spotify and Twitter Data for Generating a Recent and Public Dataset for Music Recommendation", *Proc. of the 26nd Workshop Grundlagen von Datenbanken*, 2014.

## Appendices:

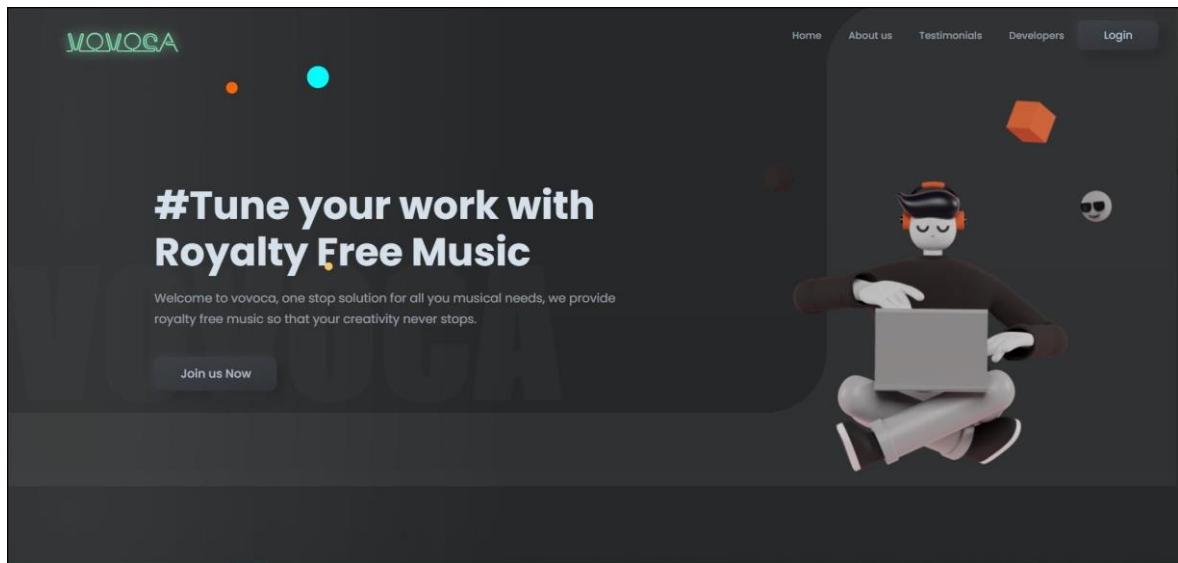


Figure 18

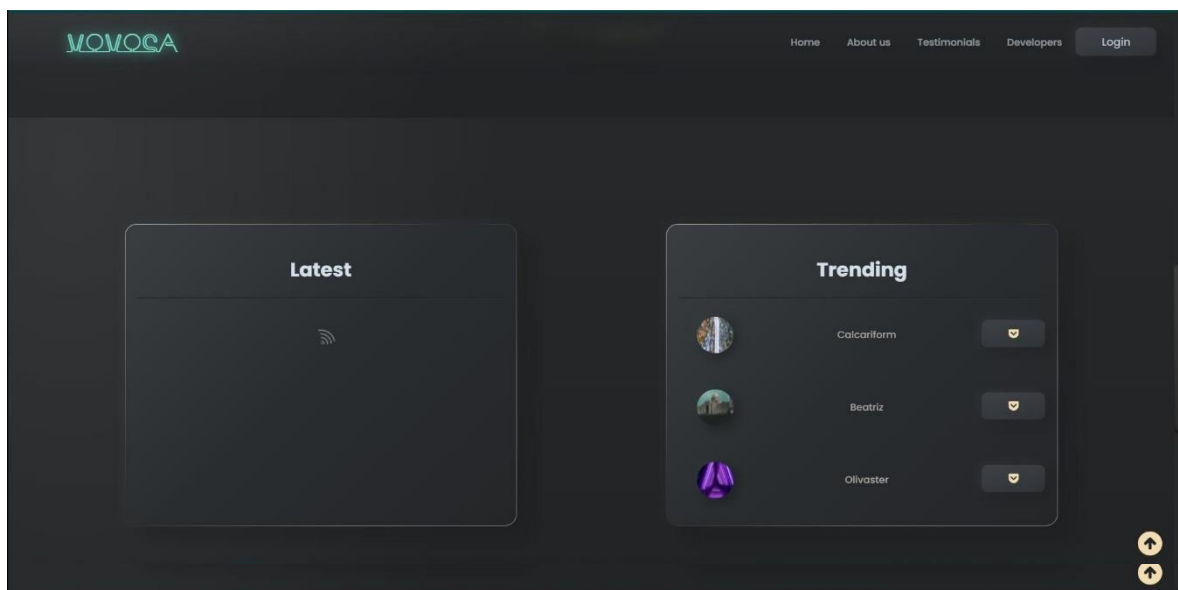


Figure 19

```
import axios from "axios";  
  
import cookie from "js-cookie";  
  
import { toast } from "react-toastify";  
  
import {
```

```

GET_MUSIC,

LOADING,

GET_MUSIC_SINGLE,

UPLOADED_MUSIC,

GET_LATEST_MUSIC,

GET_TRENDING_MUSIC,

CATEGORISED_MUSIC,

RESET_CATEGORISED_MUSIC,

SET_ERROR,

} from "./type";

import setAuthToken from "./utils/setAuthToken";

export const getAllMusic = (page) => async (dispatch) => {

  try {

    dispatch({ type: LOADING });

    const res = await axios.get(

      `https://vovoca.herokuapp.com/api/music/?page=${page}&limit=5`

    );

    dispatch({

      type: GET_MUSIC,

      payload: res.data,

    });
  }
}

```

```

    } catch (error) {

      console.log(error.message);

    }
  };

export const getSingleMusic = (id) => async (dispatch) => {

  try {

    dispatch({ type: LOADING });

    const res = await
    axios.get(`https://vovoca.herokuapp.com/api/music/${id}`);

    dispatch({

      type: GET_MUSIC_SINGLE,

      payload: res.data,

    });

  } catch (error) {

    console.log(error);

  }
};

export const getTrendingMusic = () => async (dispatch) => {

  try {

    dispatch({ type: LOADING });

```

```

const res = await axios.get(
  `https://vovoca.herokuapp.com/api/music/trending`
);

dispatch({
  type: GET_TRENDING_MUSIC,
  payload: res.data,
});
} catch (error) {
  console.log(error);
}
};

```

```

export const getLatestMusic = () => async (dispatch) => {
  try {
    dispatch({ type: LOADING });

    const res = await axios.get(
      `https://vovoca.herokuapp.com/api/music/latest`
    );

    dispatch({
      type: GET_LATEST_MUSIC,
      payload: res.data,

```

```

    });

  } catch (error) {

    console.log(error);

  }

};

export const getUploadedMusic = (page = 1) => async (dispatch) => {

  try {

    setAuthToken(cookie.get("token"));

    const res = await axios.get(

      `https://vovoca.herokuapp.com/api/admin/uploaded?page=${page}`

    );

    dispatch({

      type: UPLOADED_MUSIC,

      payload: res.data,

    });

  } catch (error) {

    dispatch({ type: SET_ERROR });

    console.log(error);

  }

};

```



```

export const getCategorisedMusic = (selected_category, page) => async (
  dispatch
) => {
  try {
    const res = await axios.get(
      `https://vovoca.herokuapp.com/api/music/?category=${selected_category.join(
        "+"
      )}&page=${page}`
    );
    dispatch({
      type: CATEGORISED_MUSIC,
      payload: res.data,
    });
  } catch (error) {
    console.log(error);
  }
};

export const resetCategorisedMusic = () => async (dispatch) => {

```

```
dispatch({  
  type: RESET_CATEGORISED_MUSIC,  
});  
};
```

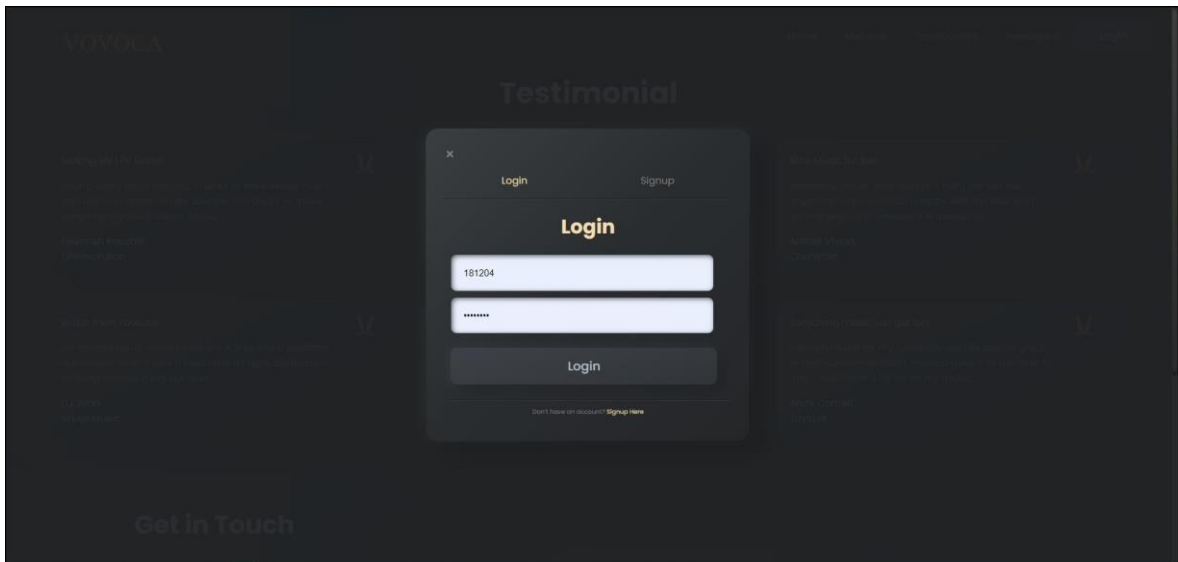
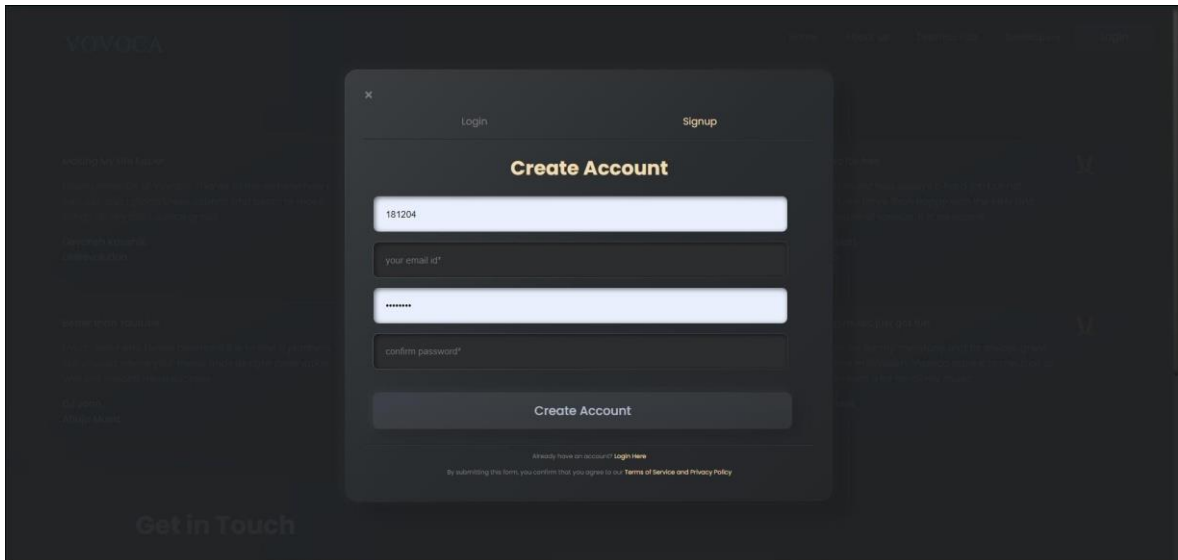


Figure 20

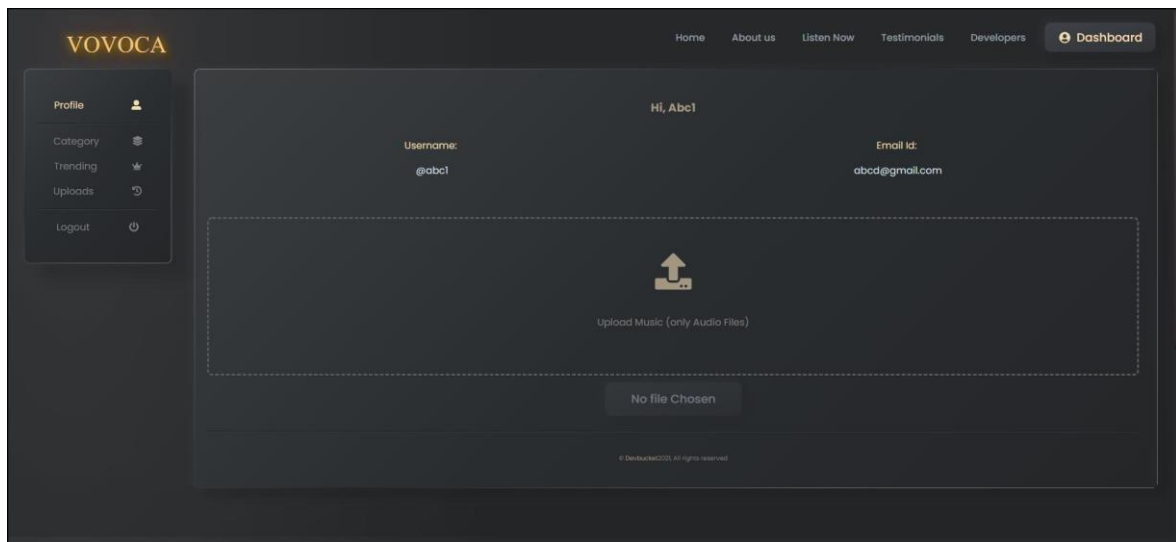


Figure 21

Dependencies:

```
{  
  
  "name": "frontend",  
  
  "version": "0.1.0",  
  
  "private": true,  
  
  "scripts": {  
  
    "dev": "next dev -p 3001",  
  
    "build": "next build",  
  
    "start": "next start"  
  
  },  
  
  "dependencies": {  
  
    "axios": "^0.21.1",  
  
    "js-cookie": "^2.2.1",  
  
    "netlify": "^6.1.16",  
  
  }  
}
```

```
"next": "10.0.9",  
"nextjs-redirect": "^5.0.2",  
"react": "17.0.2",  
"react-dom": "17.0.2",  
"react-h5-audio-player": "^3.6.2",  
"react-icons": "^4.2.0",  
"react-js-pagination": "^3.0.3",  
"react-modal": "^3.12.1",  
"react-paginate": "^7.1.2",  
"react-redux": "^7.2.3",  
"react-responsive": "^8.2.0",  
"react-responsive-modal": "^6.0.1",  
"react-spinners-kit": "^1.9.1",  
"react-toastify": "^7.0.3",  
"redux": "^4.0.5",  
"redux-devtools-extension": "^2.13.9",  
"redux-thunk": "^2.3.0"  
}  
}
```

## Music Actions:

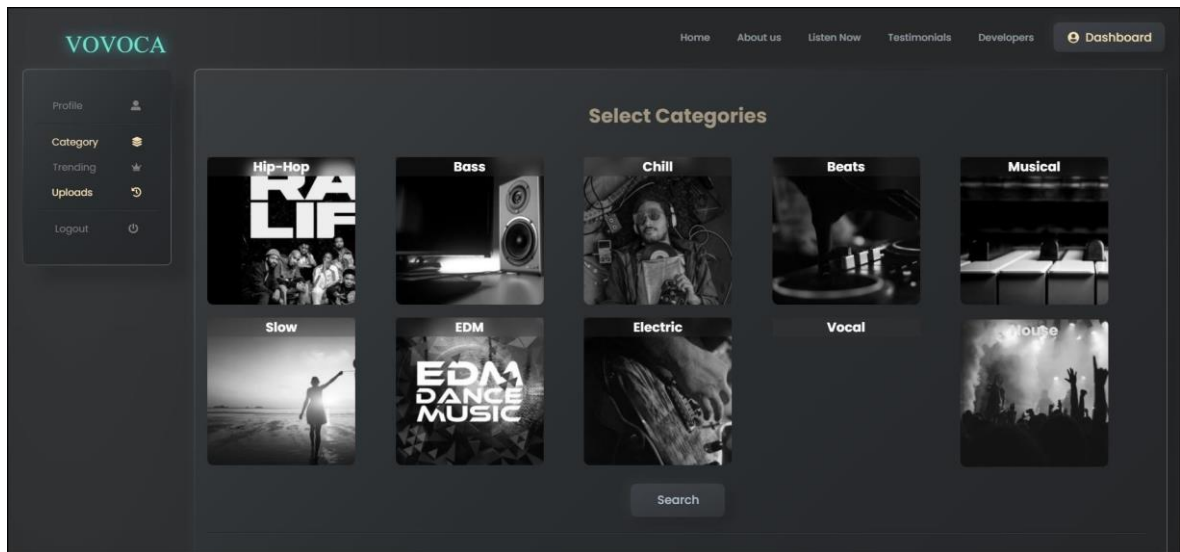


Figure 22

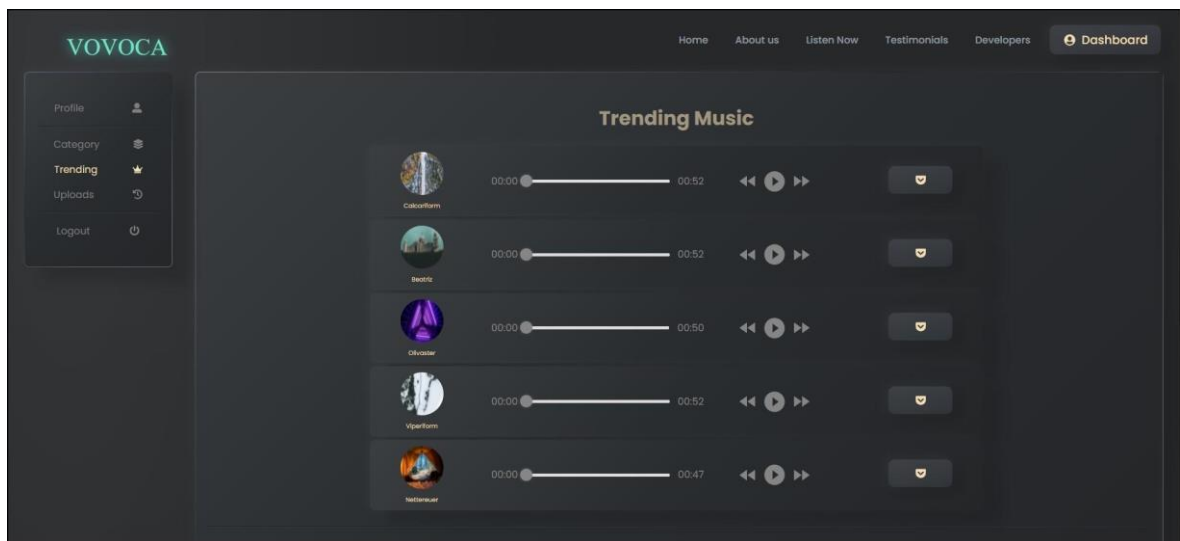


Figure 23

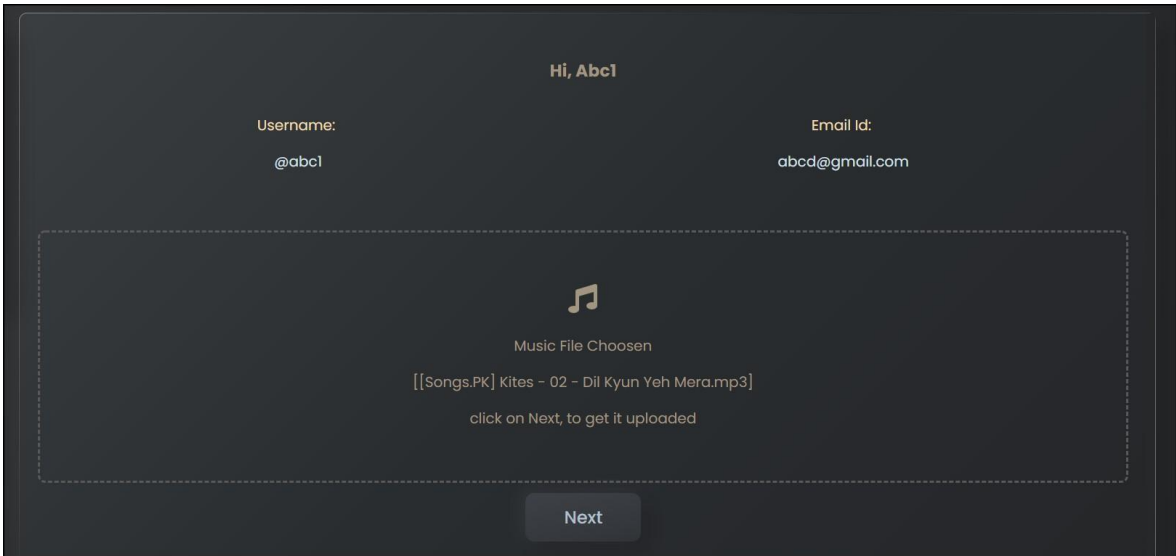


Figure 24

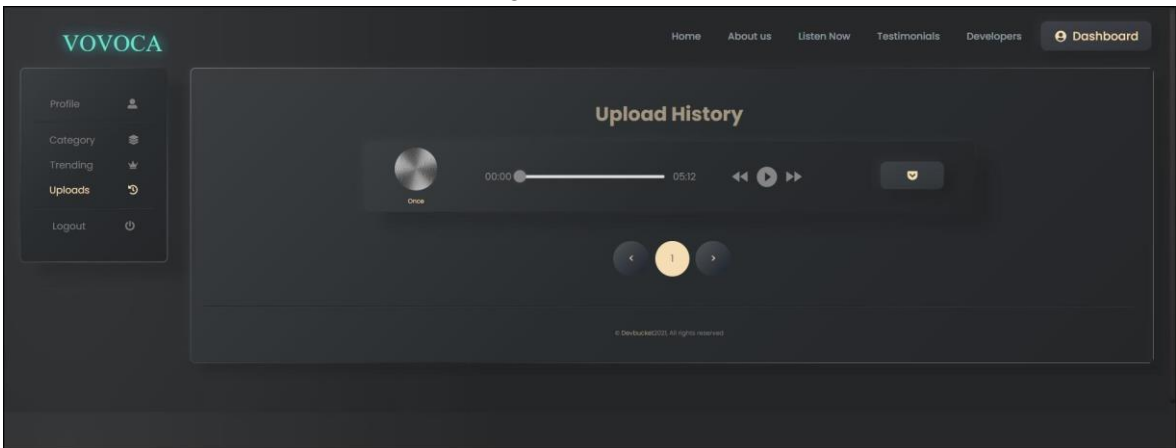


Figure 25

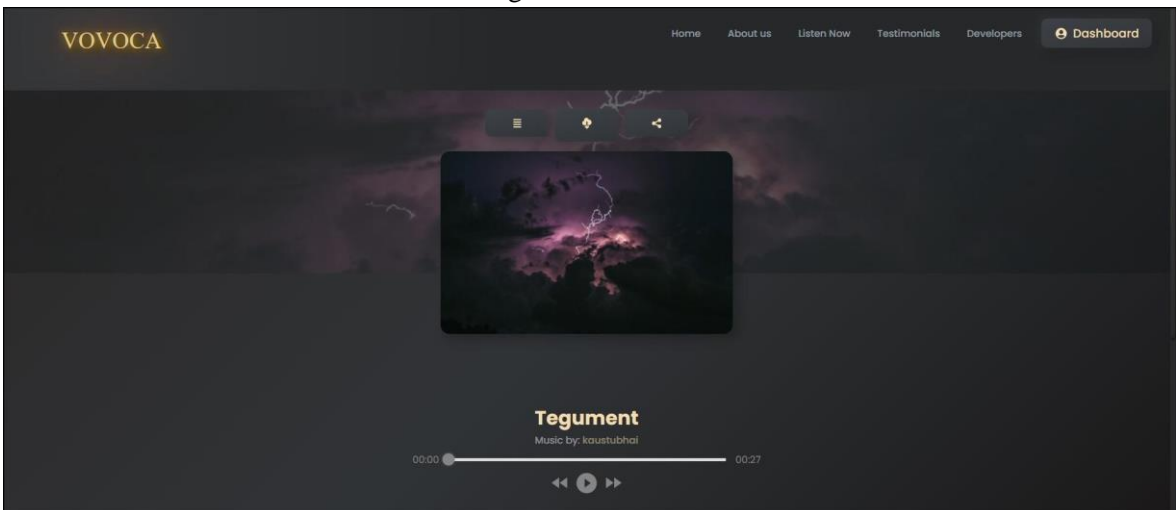


Figure 26

```

import axios from "axios";

import {
  USER_LOGIN,
  LOADING,
  USER_LOGIN_FAILED,
  LOAD_USER,
  FETCH_USER,
} from "./type";

import setAuthToken from "./utils/setAuthToken";

import cookie from "js-cookie";

export const loginUser = (email, password) => async (dispatch) => {
  const config = {
    headers: {
      "Content-Type": "Application/json",
    },
  };

  const body = JSON.stringify({ email, password });

  try {
    dispatch({ type: LOADING });

    const res = await axios.post(
      "https://vovoca.herokuapp.com/api/admin/login",

```

```

    body,
    config
  );
  dispatch({
    type: USER_LOGIN,
    payload: res.data,
  });
  // dispatch({ type: USER_LOGIN_FAILED, payload: null });
  dispatch({ type: LOAD_USER, payload: res.data });
  return true;
} catch (err) {
  dispatch({ type: USER_LOGIN_FAILED, payload: err?.response.data });
  return false;
}
};

```

```

export const registerUser = async (username, email, password) => {
  const config = {
    headers: {
      "Content-Type": "Application/json",
    },
  };
};

```



```

// request Body

const body = JSON.stringify({ username, email, password });

try {

  const res = await axios.post(

    "https://vovoca.herokuapp.com/api/admin/register",

    body,

    config

  );

  return res.data;

} catch (err) {

  console.error(err);

  return err?.response.data;

}

};

export const getUserDetails = () => async (dispatch) => {

  if (cookie.get("token")) {

    setAuthToken(cookie.get("token"));

    const res = await axios.get("https://vovoca.herokuapp.com/api/admin");

    dispatch({

      type: FETCH_USER,

```

```

payload: res.data,

});

}

};

```

Admin Table from the Database containing current users.

	<b>_id</b> [PK] uuid	<b>username</b> character varying (150)	<b>email</b> character varying (150)	<b>password</b> character varying (150)
1	126706b0-99...	kaustubhai	kaustubh229@gmail.com	\$2a\$08\$9d.1MmUJQzDAu2d...
2	982357be-04...	varun_vns	varunsingh00712@gmail.com	\$2a\$08\$gtFP8dHYv6HCKufn...
3	fa193be6-43...	Vovoca-Mix	admin@vovoca.com	\$2a\$08\$NCTH4rZU1E6p9Gkb...
4	fa829d3b-d6...	varunkk	aka39@outlook.com	\$2a\$08\$6e5FDno.1IY1keJkC...
5	c9291b35-3a...	varunke	akarr39@outlook.com	\$2a\$08\$C10X3ZPSlvggpGpE...
6	669069ac-1b...	teet	test@test.com	\$2a\$08\$NvCAUNYSkygd0e7d...
7	1c6f53db-d4...	VV	DDun99@outlook.com	\$2a\$08\$CkKwG7BzxYc5alDD5...
8	97177da9-a7...	gryffindorito	ananyashukla2110@gmail.com	\$2a\$08\$TOfLihhRK3Je5ePlhq...
9	a75addc3-40...	Phalanxia	gryffindorito	\$2a\$08\$dCcogHf/bVW0.OI5Q...
10	2954a420-dd...	Prakharpandey	prakhar@pandey.com	\$2a\$08\$b.mlStXzRQlBmvfzD...
11	5f9de28c-4b...	arpit0891	arpit.mishra@getwork.org	\$2a\$08\$KdTwAXOXUUIUqx69...
12	868c7151-fc...	mav3rick	amolikvivan@gmail.com	\$2a\$08\$PAmcew0v27aAUBn...
13	0afa2535-61...	xyz	nandan3101@gmail.com	\$2a\$08\$5fDn.gjsRd7pg7U1p...
14	e5303ae7-47...	xayem63652@0pppp.com	xayem63652@0pppp.com	\$2a\$08\$TNMBSAZyWQ4i0Fo...
15	129e1e9b-80...	Prakharbh	prakharbhardwaj784@gmail.c...	\$2a\$08\$0Fp4eYiMK17Pe27f...
16	af16122a-78...	kagayaki	1030229578@qq.com	\$2a\$08\$b8m8/heEd2.aWG9Kn...

Figure 28

## Tracks currently present in the database.

Data Output		Explain	Messages	Notifications				
<u>_id</u>	<u>name</u>	<u>audiobuffer</u>	<u>downloadcount</u>	<u>tags</u>	<u>image</u>	<u>timestamps</u>	<u>createdby</u>	
[PK] uuid	character varying (150)	bytea	integer	text[]	character varying (250)	timestamp without time zone	uuid	
1	04876926-93...	crevice	[binary data]	2	{electric,...	https://images.pexels.com/ph...	2021-04-14 10:36:21.401768	fa193be6-43b0-4...
2	403f62ca-7fb...	divvs	[binary data]	0	{edm}	https://images.pexels.com/ph...	2021-05-09 16:43:01.355023	26c4b553-5b2b-...
3	f1ba7bf5-b2...	tegument	[binary data]	21	{chill,mu...	https://images.unsplash.com...	2021-04-03 17:32:51.643875	126706b0-9989-...
4	7528bdf8-b8...	virtualism	[binary data]	20	{slow,ed...	https://images.unsplash.com...	2021-04-03 17:33:37.141374	126706b0-9989-...
5	388c9b6c-48...	Epidemic	[binary data]	0	{hip-hop,...	https://images.pexels.com/ph...	2021-05-14 09:35:00.390476	fa193be6-43b0-4...
6	62115fd9-95...	rhythmometer	[binary data]	24	{slow,ed...	https://images.unsplash.com...	2021-04-03 17:34:15.827742	126706b0-9989-...
7	cf74cff3-9d9...	autology	[binary data]	19	{beats,b...	https://images.unsplash.com...	2021-04-03 17:38:02.958953	982357be-04fe-4...
8	dea0adc8-38...	viperiform	[binary data]	55	{beats,b...	https://images.unsplash.com...	2021-04-03 17:38:20.487139	982357be-04fe-4...
9	6334eb55-92...	Red velvet	[binary data]	0	{slow,mu...	https://images.pexels.com/ph...	2021-05-14 09:39:17.955174	fa193be6-43b0-4...
10	3dd66f45-06...	nimonic	[binary data]	26	{beats,b...	https://images.unsplash.com...	2021-04-04 18:26:33.170818	982357be-04fe-4...
11	e5fad9fa-808...	Beatriz	[binary data]	59	{beats,el...	https://images.unsplash.com...	2021-04-05 19:33:21.804527	982357be-04fe-4...
12	7a2f9539-84...	grumschoses	[binary data]	1	{beats,b...	https://images.pexels.com/ph...	2021-04-07 11:18:42.078019	fa193be6-43b0-4...
13	02bf40f-930...	tintinnabulum	[binary data]	6	{house,e...	https://images.unsplash.com...	2021-04-03 17:36:18.236129	fa193be6-43b0-4...
14	cea62825-7c...	physitheism	[binary data]	11	{house,e...	https://images.unsplash.com...	2021-04-03 17:35:55.87371	fa193be6-43b0-4...
15	fb5a4420-f7e...	digitiform	[binary data]	17	{slow,ed...	https://images.unsplash.com...	2021-04-03 17:33:54.732963	fa193be6-43b0-4...
16	abd5ff4c-a99...	troika	[binary data]	25	{house,b...	https://images.unsplash.com...	2021-04-03 17:31:41.991267	fa193be6-43b0-4...
17	617fa527-23...	waveson	[binary data]	44	{beats,b...	https://images.unsplash.com...	2021-04-03 17:38:53.034612	fa193be6-43b0-4...
18	788d93cb-40...	oilvaster	[binary data]	56	{beats,b...	https://images.unsplash.com...	2021-04-03 17:39:46.302814	fa193be6-43b0-4...
19	ab104803-26...	auspong	[binary data]	23	{beats,hi...	https://images.unsplash.com...	2021-04-05 18:45:06.783513	982357be-04fe-4...
20	045d10c6-02...	deballecut	[binary data]	22	{slow,mu...	https://images.unsplash.com...	2021-04-03 17:33:11.401399	fa193be6-43b0-4...
21	006e97fd-b3...	callipygous	[binary data]	13	{chill,mu...	https://images.unsplash.com...	2021-04-03 17:32:23.771953	126706b0-9989-...
22	8f313468-bf...	calcariform	[binary data]	70	{beats,b...	https://images.unsplash.com...	2021-04-03 17:38:34.855389	fa193be6-43b0-4...

Figure 29