VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018

A DBMS Mini Project Report

on

" ROKERS "

Submitted in partial fulfillment of the requirements for the V semester and award of the degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi

Submitted by:

Aditya Bohra 1RN20CS010
Aman Kumar 1RN20CS016

Under the Guidance of:

Prof. Karanam Sunil Kumar
Assistant Professor
Dept. of CSE

Department of Computer Science and Engineering
(NBA Accredited for academic years 2018-19, 2019-20, 2020-22, 2022-2025)
RNS Institute of Technology
Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098

2022-2023

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

(NBA Accredited for academic years 2022-25)

CERTIFICATE

Certified that the mini project work entitled **"ROKERS"** has been successfully carried out by **"ADITYA BOHRA"** bearing USN **"1RN20CS010"** and **"AMAN KUMAR"** bearing USN **"1RN20CS016"**, bonafide students of **"RNS Institute of Technology"** in partial fulfillment of the requirements for the 5th semester of **"Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University"**, Belagavi, during academic year 2022-2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the DBMS Laboratory requirements of 5th semester BE, CSE.

Signature of the Guide	Signature of the HoD	Signature of the Principal
Prof. Karanam Sunil Kumar	**Dr. Kiran**	**Dr. M K Venkatesha**
Assistant Professor	Professor and HOD	Principal
Dept. of CSE	Dept. of CSE	

External Viva:

Name of the Examiners	Signature with Date
1.	
2.	

Acknowledgement

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We are grateful to Management and **Dr. M K Venkatesha**, Principal, RNSIT, Bangalore, for his support towards completing this mini project.

We would like to thank **Dr. Kiran P**, Professor & Head, Department of Computer Science & Engineering, RNSIT, Bangalore, for his valuable suggestions and expert advice.

We deeply express our sincere gratitude to our guide **Prof. Karanam Sunil Kumar**, Assistant Professor, Department of CSE, RNSIT, Bangalore, for his able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank all the teaching and non-teaching staff of Department of Computer Science & Engineering, RNSIT, Bengaluru-98 for their constant support and encouragement.

Abstract

The project “Rokers” can to be used by a group of people to easily add music and listen the same in sync with others, It was difficult for the people to listen music of there choice and add them on a common speaker. This problem is solved using this project. The project has been thought about from a user point of view and revolves around the usual requirements for a user who want to enjoy there time listening music in a group with others.

The objective of this project is making an interactive user-friendly website which can easily search and maintain songs playlists,albums and their modification in an effective, easy way. Every song can be easily searched by their name. We have implemented the idea by making a web application that includes differnt sections. Any user using the website will be able to simply search and play the song and all the necessary details will be automatically added and maintained in database. The UI has been made very simple to provide ease of access for all types of users.

This project requires HTML, JavaScript, CSS in the frontend, Javascript for backend and database connectivity and MySQL for database management. We seek to expand the project by having a fully real-life model for different users to use the website and have a better experience. We also seek to add more features to the website like adding a chat box for the users to chat with each other.

Contents

Chapter 1

Introduction

1.1 What is Rokers ?

In today's world, music streaming has become an integral part of our daily lives. We all have our favorite music and playlists that we love listening to. However, the traditional way of listening to music through CDs or MP3 players has become outdated. With the advent of music streaming platforms, it has become easier to access music from anywhere and at any time. But, what if you could listen to your favorite music through an inbuilt speaker connected to the internet and controlled by your smartphone? This is where Rokers comes in.

Rokers is a music streaming system that allows users to play music from their smartphone through an inbuilt speaker. The system is connected to the internet, which enables users to access music from various online platforms such as Spotify. The system fetches music information from Spotify API and stores it in a MySQL database, which is used to keep records of music playback. The system then fetches actual music from other online sources and Plays it on the local system . Thus eliminating the need for storing music on the local system drive. It also helps to keep the mobile free therefore it can be used separate from the speaker without worrying about range and capacity.

1.2 Description Of Tools And Technologies

1.2.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of

a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as ¡img /¿ and ¡input /¿ introduce content into the page directly. Others such as ¡p¿...¡/p¿ surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

1.2.2 Javascript

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and this is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as Self and Scheme.

1.2.3 MySQL

MySQL is a free and open-source relational database management system (RDBMS) that is widely used in web applications. It is developed, distributed, and supported by Oracle Corporation. MySQL is known for its speed, reliability, and flexibility. It is used by many large websites and applications, including WordPress, Drupal, and Wikipedia. MySQL stores data in tables and uses structured query language (SQL) to access and manipulate the data. It can handle very large amounts of data and can be used on many different operating systems, including Windows, Linux, and macOS. There are many tools and libraries available for interacting with MySQL from various programming languages, including PHP, Python, and Java. This makes it easy to integrate MySQL into web-based applications.

Chapter 2

Introduction To Database Management System

2.1 Introduction

Databases and database technology have a major impact on the growing use of computers. It is fair to say that databases play a critical role in almost all areas where computers are used, including business, electronic commerce, engineering, medicine, genetics, law, education, and library science. The word database is so commonly used that we must begin by defining what a database is. Our initial definition is quite general. A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book or you may have stored it on a hard drive, using a personal computer and software such as Microsoft Access or Excel. This collection of related data with an implicit meaning is a database. The preceding definition of database is quite general; for example, we may consider the collection of words that make up this page of text to be related data and hence to constitute a database. However, the common use of the term database is usually more restricted. A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the miniworld or the universe of discourse (UoD). Changes to the miniworld are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database

- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications. Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called meta-data. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld, and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

2.2 History of DBMS

In 1959, the TX-2 computer was developed at MIT's Lincoln Laboratory. The TX-2 integrated a number of new man-machine interfaces. A light pen could be used to draw sketches on the computer using Ivan Sutherland's revolutionary Sketchpad software.[4] Using a light pen, Sketchpad allowed one to draw simple shapes on the computer screen, save them and even recall them later. The light pen itself had a small photoelectric cell in its tip. This cell emitted an electronic pulse whenever it was placed in front of a computer screen and the screen's electron gun fired directly at it. By simply timing the electronic pulse with the current location of the electron gun, it was easy to pinpoint exactly where the pen was on the screen at any given moment. Once that was determined, the computer could then draw a cursor at that location. Also in 1961 another student at MIT, Steve Russell, created the first video game, E. E. Zajac, a scientist at Bell Telephone Laboratory (BTL), created a film called "Simulation of a two-giro gravity attitude control system" in 1963. During 1970s, the first major advance in 3D computer graphics was created at UU by these early pioneers, the hidden-surface algorithm. In order to draw a representation of a 3D object on the screen, the computer must determine which surfaces are "behind" the object from the viewer's perspective, and thus should be "hidden" when the computer creates (or renders) the image. In the 1980s, artists and graphic designers began to see the personal computer, particularly the Commodore Amiga and Macintosh, as a serious design tool, one that could save time and draw more accurately than other methods. In the late 1980s, SGI computers were used to create some of the first fully computer-generated short films at Pixar. The

Macintosh remains a highly popular tool for computer graphics among graphic design studios and businesses. Modern computers, dating from the 1980s often use graphical user interfaces (GUI) to present data and information with symbols, icons and pictures, rather than text. Graphics are one of the five key elements of multimedia technology. 3D graphics became more popular in the 1990s in gaming, multimedia and animation. In 1996, Quake, one of the first fully 3D games, was released. In 1995, Toy Story, the first full-length computer-generated animation film, was released in cinemas worldwide. Since then, computer graphics have only become more detailed and realistic, due to more powerful graphics hardware and 3D modelling software.

2.3 Characteristics of DBMS

A number of characteristics distinguish the database approach from the much older approach of programming with files. In traditional file processing, each user defines and implements the files needed for a specific software application as part of programming the application. For example, one user, the grade reporting office, may keep files on students and their grades. Programs to print a student's transcript and to enter new grades are implemented as part of the application. A second user, the accounting office, may keep track of students' fees and their payments. Although both users are interested in data about students, each user maintains separate files— and programs to manipulate these files—because each requires some data not avail able from the other user's files. This redundancy in defining and storing data results in wasted storage space and in redundant efforts to maintain common up-to-date data. In the database approach, a single repository maintains data that is defined once and then accessed by various users. In file systems, each application is free to name data elements independently. In contrast, in a database, the names or labels of data are defined once, and used repeatedly by queries, transactions, and applications. The main characteristics of the database approach versus the file-processing approach are the following:

- Self-describing nature of a database system.
- Insulation between programs and data, and data abstraction.
- Support of multiple views of the data
- Sharing of data and multiuser transaction processing.

2.3.1 Self-Describing Nature of a Database System

A fundamental characteristic of the database approach is that the database system contains not only the database itself but also a complete definition or description of the database structure and constraints. This definition is stored in the DBMS catalog, which contains information such as the structure of each file, the type and storage format of each data item, and various constraints on the data. The information stored in the catalog is called meta-data, and it describes the structure of the primary database. The catalog is used by the DBMS software and also by database users who need information about the database structure. A general-purpose DBMS software package is not written for a specific database application. Therefore, it must refer to the catalog to know the structure of the files in a specific database, such as the type and format of data it will access. The DBMS software must

work equally well with any number of database applications—for example, a university database, a banking database, or a company database—as long as the database definition is stored in the catalog. In traditional file processing, data definition is typically part of the application programs themselves. Hence, these programs are constrained to work with only one specific database, whose structure is declared in the application programs. For example, an application program written in C++ may have structure or class declarations, and a COBOL program has data division statements to define its files. Whereas file-processing software can access only specific databases, DBMS software can access diverse databases by extracting the database definitions from the catalog and using these definitions.

2.3.2 Insulation between Programs and Data, and Data Abstraction

In traditional file processing, the structure of data files is embedded in the application programs, so any changes to the structure of a file may require changing all programs that access that file. By contrast, DBMS access programs do not require such changes in most cases. The structure of data files is stored in the DBMS catalog separately from the access programs. We call this property program-data independence. In some types of database systems, such as object-oriented and object-relational systems users can define operations on data as part of the database definitions. An operation (also called a function or method) is specified in two parts. The interface (or signature) of an operation includes the operation name and the data types of its arguments (or parameters).The implementation (or method) of the operation is specified separately and can be changed without affecting the interface. User application programs can operate on the data by invoking these operations through their names and arguments, regardless of how the operations are implemented. This may be termed program-operation independence. The characteristic that allows program-data independence and program-operation independence is called data abstraction. A DBMS provides users with a conceptual representation of data that does not include many of the details of how the data is stored or how the operations are implemented. Informally, a data model is a type of data abstraction that is used to provide this conceptual representation. The data model uses logical concepts, such as objects, their properties, and their interrelationships, that may be easier for most users to understand than computer storage concepts. Hence, the data model hides storage and implementation details that are not of interest to most database users.

2.3.3 Support of Multiple Views of the Data

A database typically has many users, each of whom may require a different perspective or view of the database. A view may be a subset of the database or it may contain virtual data that is derived

from the database files but is not explicitly stored. Some users may not need to be aware of whether the data they refer to is stored or derived. A multiuser DBMS whose users have a variety of distinct applications must provide facilities for defining multiple views.

2.3.4 Sharing of Data and Multiuser Transaction Processing

A multiuser DBMS, as its name implies, must allow multiple users to access the database at the same time. This is essential if data for multiple applications is to be integrated and maintained in a single database. The DBMS must include concurrency control software to ensure that several users trying to update the same data do so in a controlled manner so that the result of the updates is correct. For example, when several reservation agents try to assign a seat on an airline flight, the DBMS should ensure that each seat can be accessed by only one agent at a time for assignment to a passenger. These types of applications are generally called online transaction processing (OLTP) applications. A fundamental role of multiuser DBMS software is to ensure that concurrent transactions operate correctly and efficiently. The concept of a transaction has become central to many database applications. A transaction is an executing program or process that includes one or more database accesses, such as reading or updating of database records. Each transaction is supposed to execute a logically correct database access if executed in its entirety without interference from other transactions. The DBMS must enforce several transaction properties. The isolation property ensures that each transaction appears to execute in isolation from other transactions, even though hundreds of transactions may be executing concurrently. The atomicity property ensures that either all the database operations in a transaction are executed or none are.

2.4 Applications of DBMS

Applications where we use Database Management Systems are:

- **Telecom:** There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain that huge amount of data that keeps updating every millisecond.
- **Industry:** Where it is a manufacturing unit, warehouse or distribution centre, each one needs a database to keep the records of ins and outs. For example distribution centre should keep a track of the product units that supplied into the centre as well as the products that got delivered out from the distribution centre on each day; this is where DBMS comes into picture.

- **Banking System:** For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems.

- **Education Sector:** Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is a hell lot amount of inter-related data that needs to be stored and retrieved in an efficient manner.

- **Online Shopping:** You must be aware of the online shopping websites such as Amazon, Flip kart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

Chapter 3

Resource Requirements

3.1 Hardware Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.Table 3.1 gives details of hardware requirements.

Table 3.1: Hardware Requirements

Processor	TRaspberry Pi Zero (1GHz BCM2835 single-core 32-Bit CPU)
Processor Speed	1 GHz
RAM	512 MB

3.2 Software Requirements

The software requirements are description of features and functionalities of the system.Table 3.2 gives details of software requirements.

Table 3.2: Software Requirements

Operating System	Raspbian
SQL	MySQL
Server	Node.js(Express, YTDL-Core, Spotify)
Media Decoder	FFMPEG
WebApp	PWA(Client side Content fetching service)

Chapter 4

Database Design

4.1 Working

The system is designed to work in the following way:

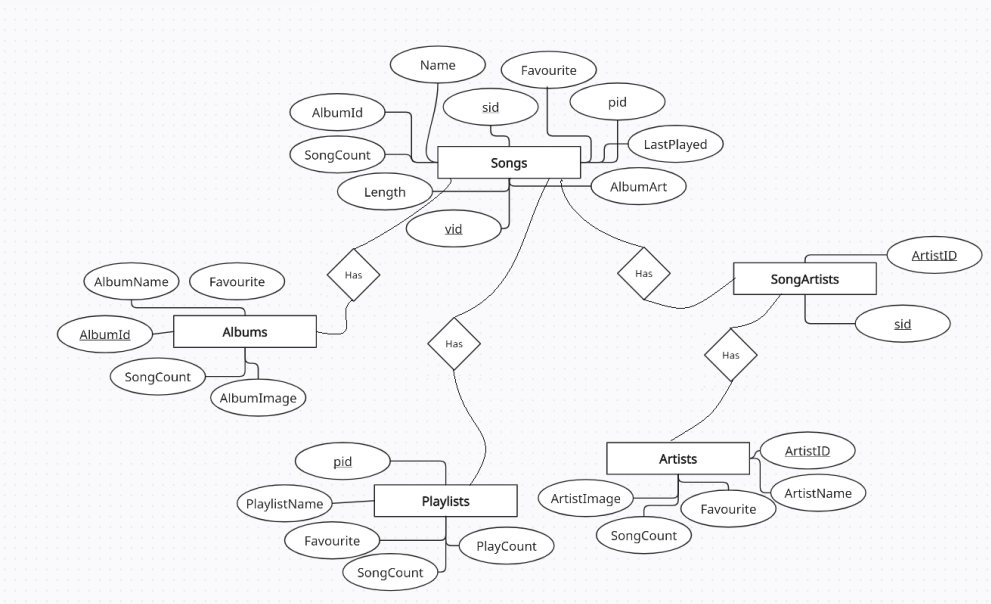
- The user connects their smartphone to the Rokers system via the internet.
- The user opens the Rokers app on their smartphone and logs in to their Spotify account.
- The app fetches music information from Spotify API and stores it in a MySQL database.
- The user can then select and play music through the inbuilt speaker. The system keeps a record of the music playback in the MySQL database.

4.2 Entities, Attributes and Relationships

The database, called rokers, will have five tables, songs, albums, artits, playlists, songArtist. Each will hold information about songs and the artistis and other important details. The two tables will be linked through a foreign key. The student table has the following fields:

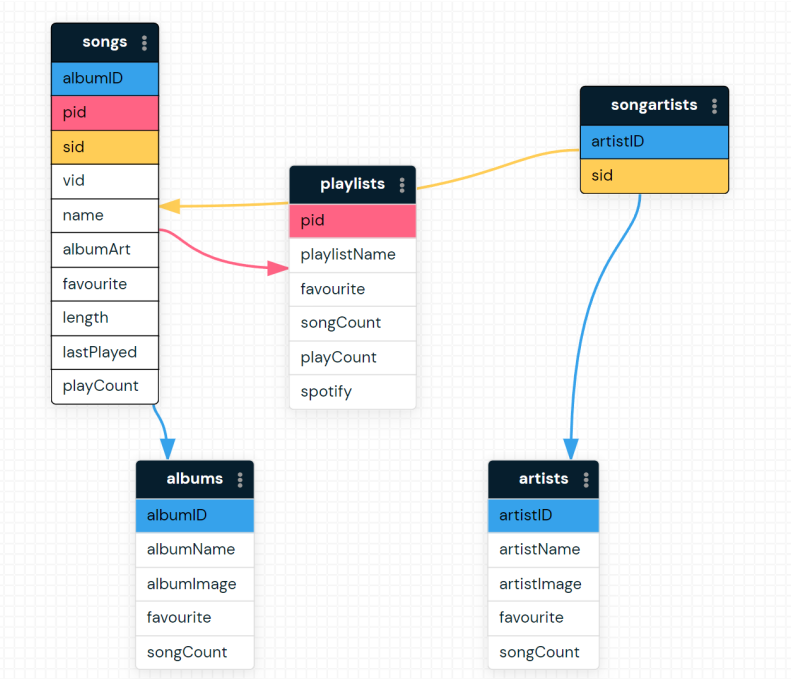
4.3 ER Schema

Figure 4.1: Entity Relationship Diagram



4.4 Schema Diagram

Figure 4.2: Relational Schema



Chapter 5

Implementation

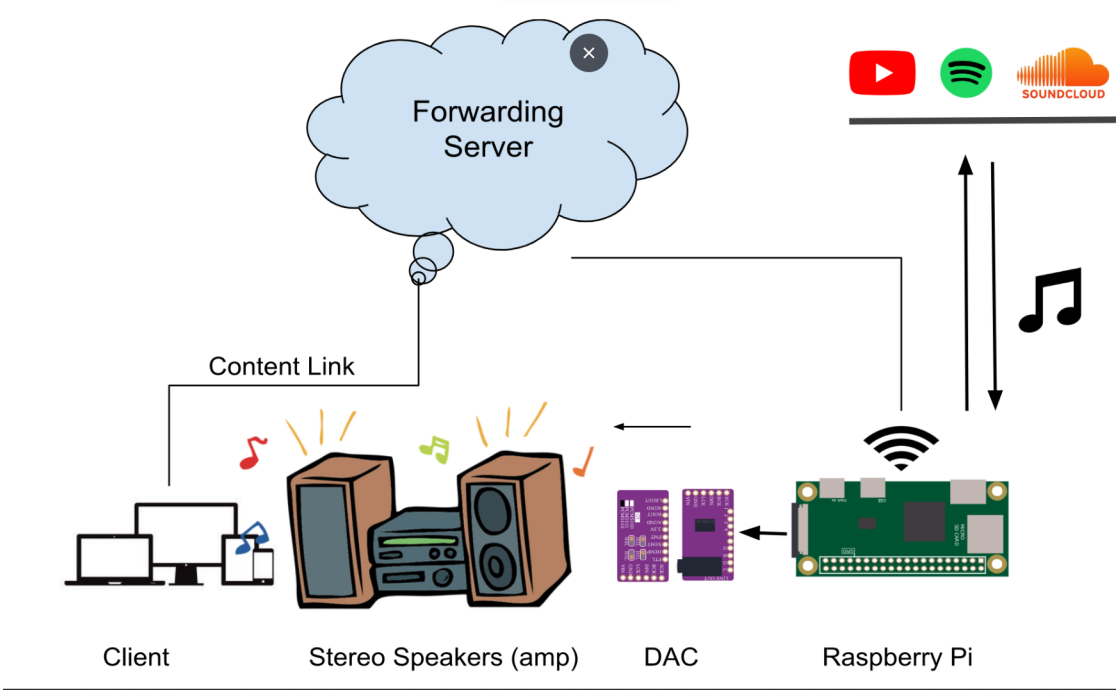
5.1 Working-Methodology

In the events of initiation of an audio session, the system will go through the following steps in order to successfully start streaming:

- The first step of course involves pointing to the media the user wishes to play. This involves any kind of Audio / Video source from any of the compatible providers ex: Spotify, YouTube, Gaana, etc.
- Time to pass the Gems ! Next , choose the client to share the Content link to. This will send it to the STUN / TURN servers to successfully reach the target device
- The middleware server helps in case of any NAT related issues and delivers the content to the correct device associated with you.
- The middleware server helps in case of any NAT related issues and delivers the content to the correct device associated with you.
- Finally the content reaches your intended device. It parses, identifies and resolves the data, fetches the media content from the corresponding service and proceeds to play it. Handling the audio buffer over to the Hi-Res DAC .
- The Hi-Res DAC converts the binary audio buffer into a high precision analog sound signal. This signal is handed over to a Hi-Fi Class AB 15 watt Audio Amplifier.
- The Hi-Res DAC converts the binary audio buffer into a high precision analog sound signal. This signal is handed over to a Hi-Fi Class AB 15 watt Audio Amplifier.

- Finally the Hi-Fi Class AB amplifier drives the 2.4 Inch speakers producing your awaited tune.

Figure 5.1: Design



5.2 Database Connection

Once the library is installed, you can use the following code to create a connection to a MySQL database:

```
const mysql = require('mysql2');
const connection = mysql.createConnection({
  host: 'localhost',
  user: 'your-username',
  password: 'your-password',
  database: 'your-database'
});
connection.connect(function(err) {
  if (err) throw err;
  console.log('Connected to the database!');
});
```

Chapter 6

Testing

Testing is an important phase in the development of any software project. It is essential to ensure that the system functions correctly and efficiently. The Rokers system was thoroughly tested to ensure that it meets the requirements and specifications set forth in the project. The following tests were performed:

1. **Functionality test :** The functionality test is used to ensure that all the features of the system work correctly. The test includes checking the functionality of the server, the connection to the Spotify API, the music playback, and the database. The test was performed to ensure that the system can fetch music information and play music as expected.

2. **Performance test :** The performance test is used to ensure that the system can handle a large number of requests and handle them efficiently. The test includes checking the response time of the system, the number of concurrent users that the system can handle, and the system's memory usage. The test was performed to ensure that the system can handle a large number of users without any issues.

3. **Stress test :** The stress test is used to ensure that the system can handle a large number of concurrent users without any issues. The test includes checking the system's response time, the number of concurrent users that the system can handle, and the system's memory usage. The test was performed to ensure that the system can handle a high number of concurrent users without any issues.

4. **Usability test :** The usability test is used to ensure that the system is easy to use and understand for the users. The test includes checking the user interface, the navigation, and the overall user experience. The test was performed to ensure that the system can be used easily and efficiently by the users.

5. **Compatibility test :** The compatibility test is used to ensure that the system can run smoothly on different platforms and devices. The test includes checking the system on different operating systems, browsers, and devices. The test was performed to ensure that the system can be accessed by a wide range of users.

The system passed all the tests successfully and was deemed ready for deployment.

Chapter 7

Results & Snapshots

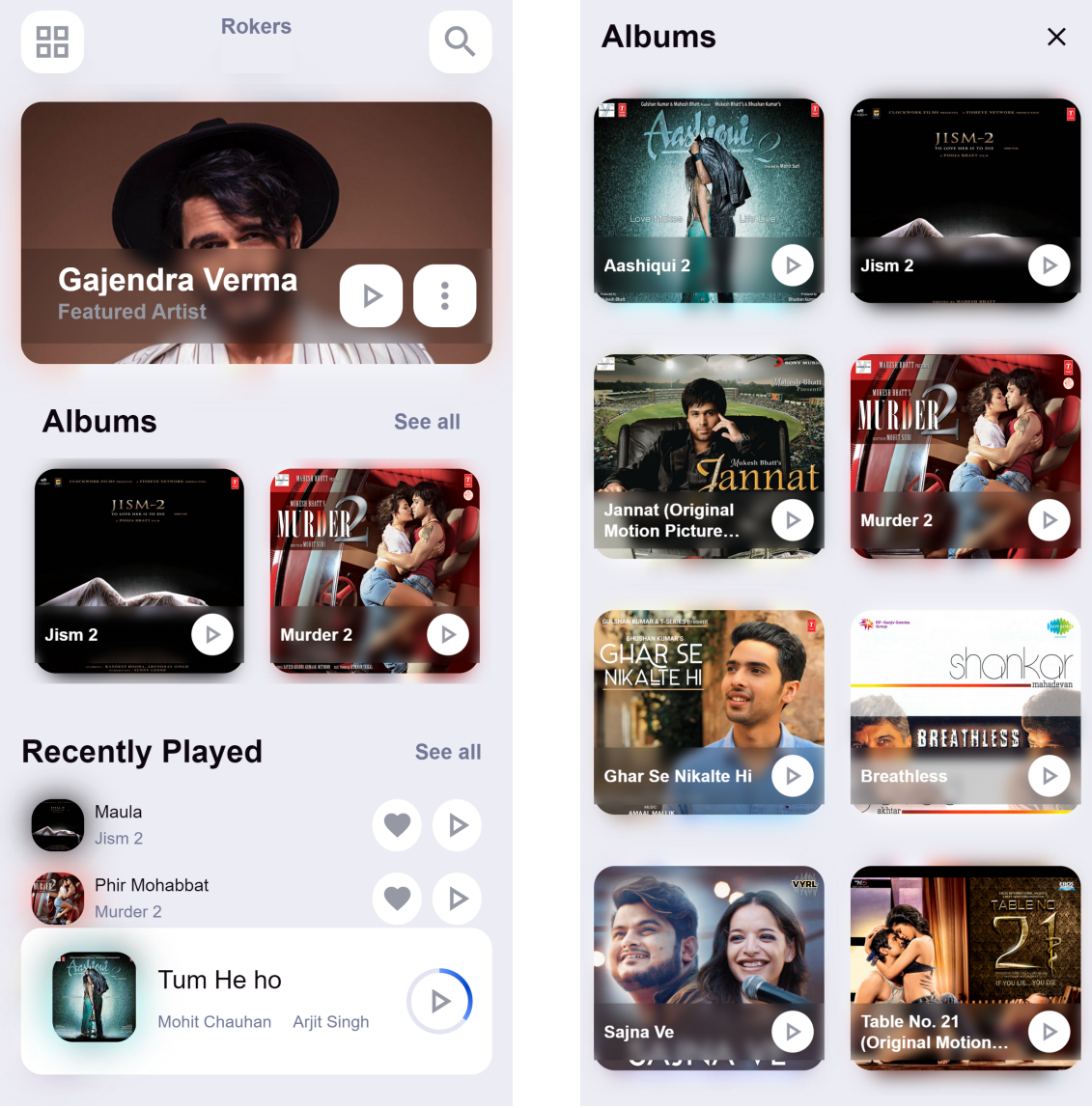


Figure 7.1: Home Page and Albums view

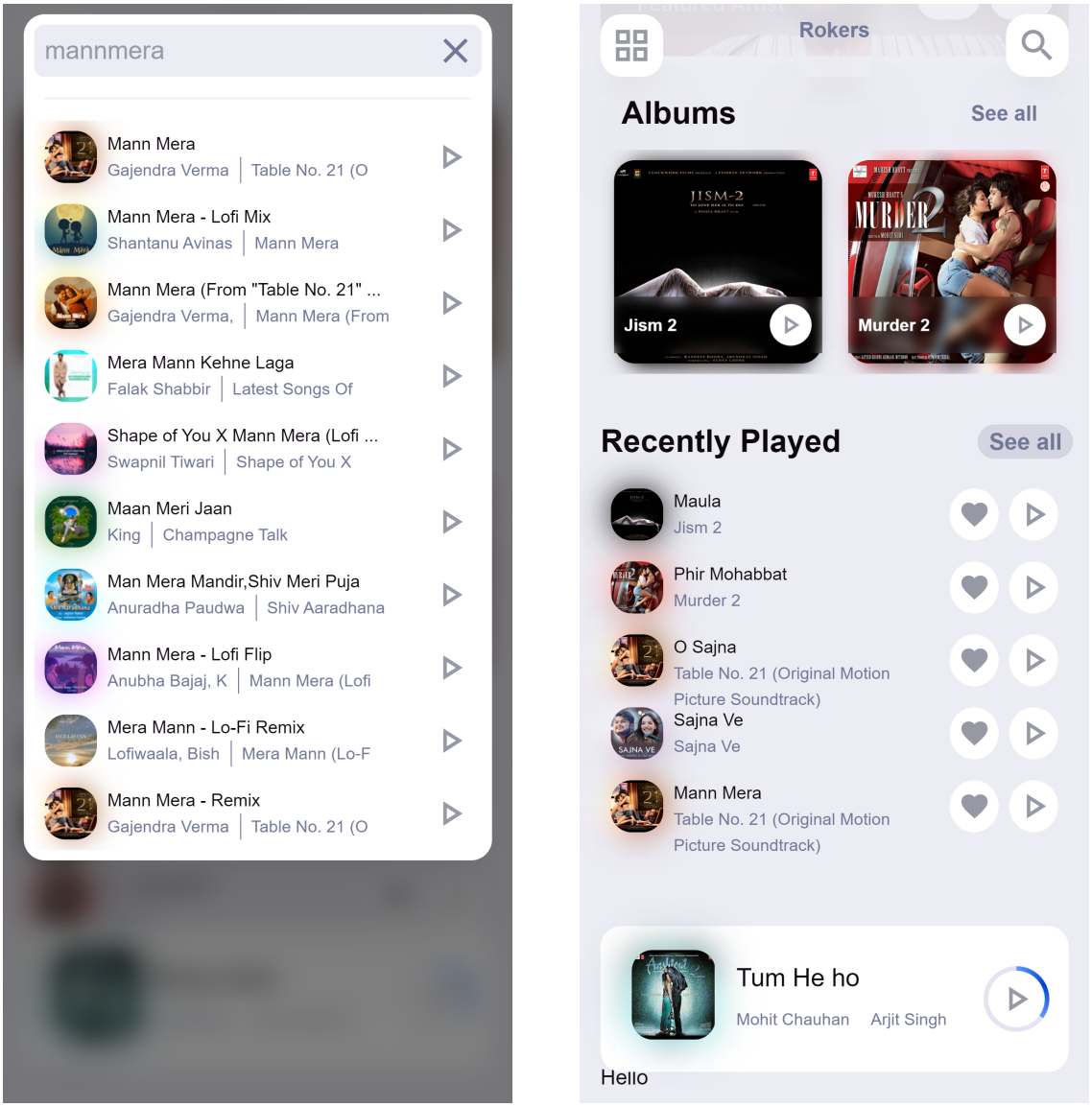


Figure 7.2: Search Page and Recently Played

Chapter 8

Conclusion & Future Enhancements

8.1 Conclusion

In conclusion, Rokers is a innovative web application that allows users to share and listen to music together in real-time. Its unique synchronous playback feature allows everyone in a group to enjoy the same music at the same time, creating a fun and immersive listening experience.

The user-friendly interface and wide range of supported music formats make it easy for anyone to join in and contribute to the playlist. Rokers is a great tool for bringing people together through their shared love of music, and we hope it will continue to bring joy and connection to its users for years to come.

8.2 Future Enhancements

Just like any other developer this project is the most basic website built using simple tools. We seek to increase the dynamic of the project by adding various other innovations to it. Such innovations would seem possible only with time, which we lack but regardless we strive to complete what we started.

We believe that apart from the present functionalities we can add :

- “Enhanced social features” Add options for users to connect with each other and interact within the app, such as chat rooms or the ability to follow other users and see their playlists.
- “Improved audio quality” Investigate ways to improve the audio quality of the music playback,

such as by supporting high-resolution audio formats or integrating with audio enhancement technologies.

- “Improved search and discovery” Implement features to help users find new music to add to the playlist, such as personalized recommendations or integration with popular music streaming services.

- Apart from these changes we are open to various suggestions and hope to implement them soon so that this website can be used by the students for their needs.

References

[1] Fundamentals of database systems by (Elmasri Navathe, 2000)

Article: Student Information System: https://en.wikipedia.org/wiki/Student_information_system

[2] Learning MySQL : https://www.tutorialspoint.com/sqlite/

JavaScript and NodeJs : http://flask.pocoo.org/docs/0.12/, http://www.youtube.com

[3] ER Diagram and Schema : https://erdplus.com

[4] Stack Overflow: https://stackoverflow.com/