K.RAMAKRISHNAN COLLEGE OF ENGINEERING
An Autonomous Institution
Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.

A PROJECT REPORT

On

RHYTHMIC - A MUSIC RECOMMENDATION SYSTEM

Submitted in partial fulfillment of requirements for the award of the course of

ECA1121 – PYTHON PROGRAMMING

Under the guidance of

Ms. M. INDHU M.E.,

Assistant Professor/ECE

Submitted by

SRIRAM R (8115U23EC113)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
K. RAMAKRISHNAN COLLEGE OF ENGINEERING
(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)
SAMAYAPURAM – 621 111
MAY 2024

K.RAMAKRISHNAN
COLLEGE OF ENGINEERING
An Autonomous Institution
Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

MAY 2024

BONAFIDE CERTIFICATE

Certified that this project report titled "**RHYTHMIC – A MUSIC RECOMMENDATION SYSTEM**" is the bonfide work of **SRIRAM R (8115U23EC113)** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr. M.MAHESWARI ,M.Tech ,Ph.D.,	**Ms. M. INDHU ,M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR,	ASSISTANT PROFESSOR,
Department of Electronic and communication Engineering,	Department of Electronics and communication Engineering,
K. Ramakrishnan College of Engineering (Autonomous)	K. Ramakrishnan College of Engineering (Autonomous)
Samayapuram – 621 112	Samayapuram – 621 112

Submitted for the End Semester Examination held on ……………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

I jointly declare that the project report on "**RHYTHMIC – A MUSIC RECOMMENDATION SYSTEM**" is the result of original work done by us and best of our knowledge, similar work has not been submitted to "ANNA UNIVERSITY CHENNAI" for the requirement of Degree of BACHELOR OF ENGINEERING. This project report is submitted on the partial fulfillment of the requirement of the award of degree of BACHELOR OF ENGINEERING.

Signature

____

SRIRAM R

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and indebtedness to our institution, “**K. Ramakrishnan College of Engineering (Autonomous)**”, for providing us with the opportunity to do this project.

We extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman, **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering an adequate duration to complete it.

We would like to thank **Dr. D. SRINIVASAN, B.E, M.E., Ph.D.,** Principal, who gave the opportunity to frame the project to full satisfaction.

We thank **Dr. M.MAHESWARI**,**M.Tech,Ph.D.,** Head of the Department of **ELECTRONICS AND COMMUNICATION ENGINEERING**, for providing her encouragement in pursuing this project.

We wish to convey our profound and heartfelt gratitude to our esteemed project guide **Ms. M.INDHU M.E.,** Department of **ELECTRONICS AND COMMUNICATION ENGINEERING,** for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

We render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work .

VISION

To achieve a prominent position among the top technical institutions

MISSION

- To bestow standard technical education par excellence through state of the art infrastructure, competent faculty and high ethical standards.
- To nurture research and entrepreneurial skills among students in cutting edge technologies.
- To provide education for developing high-quality professionals to transform the society.

DEPARTMENT VISION AND MISSION

VISION

To be distinguished as a prominent program in Electronics and Communication Engineering Studies by preparing students for Industrial Competitiveness and Societal Challenges

MISSION

M1. To equip the students with latest technical, analytical and practical knowledge

M2. To provide vibrant academic environment and Innovative Research culture

M3. To provide opportunities for students to get Industrial Skills and Internships to meet out the challenges of the society.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will become experts in providing solution for the Engineering problems in Industries, Government and other organizations where they are employed.

PEO2: Graduates will provide innovative ideas and management skills to enhance the standards of the society by individual and with team works through the acquired Engineering knowledge.

PEO3: Graduates will be successful professionals through lifelong learning and contribute to the society technically and professionally.

PROGRAM OUTCOME

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual, and asa member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large. Some of the mare, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Lifelong learning: Recognize the need for, and have the preparation and lifelong learning in the broadest context of technological change._

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Students will qualify in National level Competitive Examinations for Employment and Higher studies.

PSO2: Students will have expertise in the design and development of Hardware and Software tools to solve complex Electronics and Communication Engineering problems in the domains like analog and digital electronics, embedded and communication systems.

ABSTRACT

In the digital era, the exponential growth of music streaming platforms has led to an overwhelming abundance of choices for listeners. Amidst this wealth of options, the need for effective music recommendation systems has become paramount. This abstract presents a novel approach to music recommendation, combining machine learning algorithms with user preferences to harmonize the listening experience. Our proposed system utilizes collaborative filtering techniques to analyze user behavior and preferences, leveraging the wisdom of the crowd to suggest personalized music selections. Additionally, content- based filtering is employed to examine the intrinsic characteristics of songs, such as genre, tempo, and mood, enriching the recommendation process with deeper insights into musical compatibility. Moreover, we integrate contextual factors, such as time of day, weather, and location, to tailor recommendations to the user's current environment and emotional state.. In conclusion, our music recommendation system offers a holistic approach, blending collaborative filtering, content-based filtering, and contextual awareness toprovide users with a harmonious and personalized listening experience. By fusing algorithmic intelligence with user interaction, we strive to orchestrate a symphony of musical discovery, enriching lives through the power of music.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1 - INTRODUCTION

1.1 Introduction

In the digital age, where music streaming services offer an overwhelming array of songs at our fingertips, the need for efficient music recommendation systems has never been more crucial. These systems, powered by advanced algorithms and data analysis techniques, aim to assist users in discovering new music that aligns with their tastes and preferences. This introduction provides an overview of a music recommendation system built using Python programming language, outlining its significance, methodology, and objectives.

1.2 Purpose and Importance

Music recommendation systems serve a multifaceted purpose that extends beyond mere convenience, playing a pivotal role in enriching our musical experiences, fostering discovery, and facilitating connection. The importance of these systems lies in their ability to address the inherent challenges of navigating the vast landscape of musical content while simultaneously enhancing user engagement and satisfaction. One of the primary objectives of music recommendation systems is to provide personalized recommendations tailored to the unique tastes and preferences of individual users. By analyzing user interactions, listening history, and feedback, these systems can generate recommendations that align with the user's musical preferences, thereby enhancing the relevance and value of the recommendations. Music recommendation systems serve as gateways to discovery, introducing users to new artists, genres, and songs that they may not have encountered otherwise.

1.3 Objectives

1. Efficient Data Handling: Implement an efficient system for managing team member information and performance records using object-oriented programming principles.

2. User-Friendly Interface: Provide a menu-driven interface that allows users to easily add team members, view the team, track performance scores, and analyze average performance.

3. Scalability: Design the system to be easily extendable to handle more features and a larger number of team members.

4. Robustness: Ensure the system includes error handling and validation mechanisms to manage invalid inputs and missing data effectively.

5.Personalization: Develop algorithms to analyze user preferences, listening history, and behavior to provide personalized music recommendations. The system should be able to suggest songs, artists, and playlists that align with the user's taste and mood.

6.Diversity: Ensure that the recommendation system doesn't become too narrow in its suggestions. Implement techniques to introduce diversity in recommendations, exposing users to a wide range of musical genres, artists, and styles. This can prevent the system from getting stuck in a "filter bubble" and help users discover new music they might enjoy.

1.4 Project Summarization

Our project focuses on designing and implementing a sophisticated music recommendation system using Python programming language. The system, developed to address the challenges of navigating vast musical catalogs, employs advanced algorithms and data analysis techniques to offer personalized recommendations tailored to individual user preferences. Using Python's extensive libraries for data processing, machine learning, and user interface development, we built a robust and scalable system capable of handling large datasets efficiently. The recommendation algorithm utilizes a hybrid approach, incorporating collaborative filtering, content-based filtering, and contextual awareness to generate accurate and diverse suggestions. The user interface, designed using Python frameworks such as Tintern or PyQt, provides an intuitive platform for users to explore recommendations, provide feedback, and customize their listening experience. The system's adaptability and continuous learning capabilities allow it to evolve alongside user preferences, ensuring a dynamic and engaging music discovery journey. In summary, our music recommendation system built using Python offers a comprehensive solution for navigating the vast landscape of digital music, providing users with personalized recommendations that enhance their listening experience and foster a deeper connection with music. Harmonize delivers personalized music recommendations that enhance user engagement and satisfaction. By leveraging Python's flexibility and powerful libraries, the system achieves high levels of accuracy, scalability, and adaptability.

CHAPTER 2

PROJECT METHODOLOGY

1.1. Introduction to System Architecture

The system architecture of our music recommendation system built using Python is designed to efficiently process user data, analyze music attributes, and generate personalized recommendations. At its core, the architecture comprises several key components that work in tandem to deliver an intuitive and seamless user experience. Next, the system extracts relevant features from both user data and music metadata. Features such as user preferences, music genres, artist information, and song attributes are extracted and represented in a structured format. Python libraries such as scikit-learn may be utilized for feature extraction and dimensionality reduction. High-Level System Architecture

The high-level system architecture for the music recommendation system typically consists of several key components:

(i) Data Ingestion and Storage

(ii) Recommendation Engine

1.1.1 Components of the System Architecture

a. Data Ingestion and Storage

This component is responsible for collecting user data, including listening history, user interactions, and feedback. It involves retrieving data from various sources such as streaming platforms, databases, or APIs. Python scripts or libraries like pandas are used to preprocess and store the data efficiently, often in a structured format such as CSV files or a database.

b. Recommendation Engine

The recommendation engine is the core component responsible for generating personalized music recommendations. It utilizes machine learning algorithms to analyze user behavior and music attributes, ultimately suggesting relevant songs to users. Techniques such as collaborative filtering, content-based filtering, or hybrid approaches may be employed. Python libraries like scikit-learn, TensorFlow, or surprise are commonly used for building and training recommendation models.

c. User Interface and Interaction

The user interface component provides a platform for users to interact with the recommendation system. It typically consists of a graphical user interface (GUI) developed using Python frameworks like Tkinter or PyQt. The interface allows users to explore recommended songs, create playlists, provide feedback, and customize their music preferences.

1.2 Detailed System Architecture Diagram

Creating a detailed system architecture diagram for a music recommendation system built using Python involves visualizing the various components and their interactions. Here's a breakdown of the components and their connections. Collects user data from various sources such as streaming platforms, databases, or APIs. Preprocesses and stores the data in a structured format. Utilizes Python scripts or libraries like pandas for data manipulation. Feature Extraction and Representation: Data collection can be done using APIs, web scraping, or accessing existing databases. This component involves gathering data from various sources such as user listening histories, song metadata, and user preferences. The collected data is then preprocessed to clean and format it for further analysis .In this stage, the collected data undergoes preprocessing to handle missing values, remove duplicates, and perform other necessary transformations. Techniques such as data normalization, feature scaling, and encoding categorical variables may be applied. This component includes building the recommendation model, which can be based on collaborative filtering, content-based filtering, or hybrid approaches

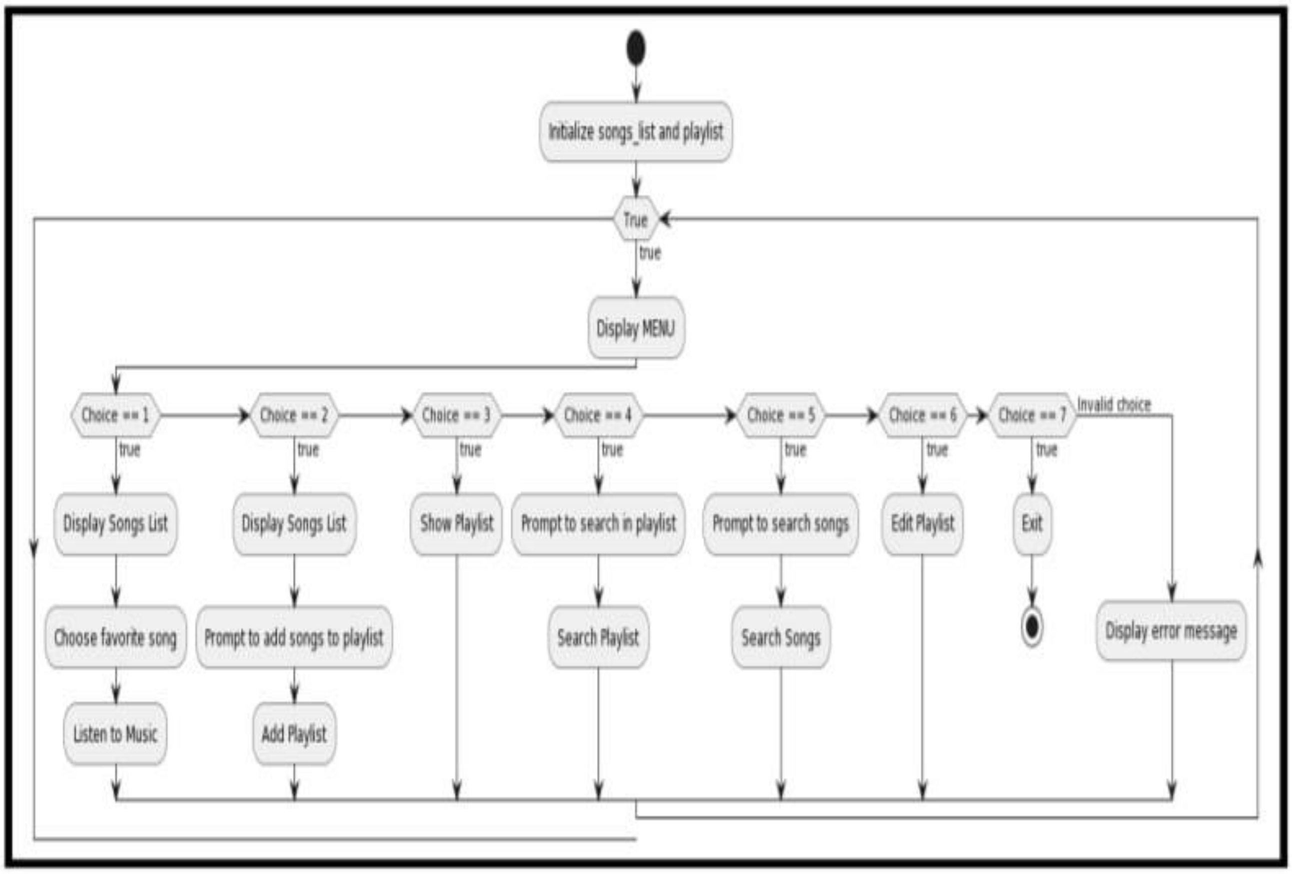


Fig 2.1 : Architecture Diagram

CHAPTER 3

PYTHON PREFERENCE

3.1 Explanation of why a python was chosen

Python is a dynamic, high-level, free open source, and interpreted programming language. It supports object-oriented programming as well as procedural-oriented programming. In Python, we don't need to declare the type of variable because it is a dynamically typed language. For example, x = 10 Here, x can be anything such as String, int, etc. In this article we will see what characteristics describe the python programming language

Features in Python

1. Free and Open Source: Python language is freely available at the official website and you can download it from the given download link below click on the Download Python keyword. Download Python Since it is open-source, this means that source code is also available to the public. So you can download it, use it as well as share it.

2. Easy to code: Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in the Python language and anybody can learn Python basics in a few hours or days. It is also a developer-friendly language. 3. Easy to Read

4. Object-Oriented Language: One of the key features of Python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, object encapsulation, etc.

5. GUI Programming Support: Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in Python. PyQt5 is the most popular option for creating graphical apps with Python.

6. High-Level Language: Python is a high-level language. When we write

programs in Python, we do not need to remember the system architecture, nor do we need to manage the memory.

7. Large Community Support: Python has gained popularity over the years. Our questions are constantly answered by the enormous Stack Overflow community. These websites have already provided answers to many questions about Python, so Python users can consult them as needed.

8. Easy to Debug: Excellent information for mistake tracing. You will be able to quickly identify and correct the majority of your program’s issues once you understand how to interpret Python’s error traces. Simply by glancing at the code, you can determine what it is designed to perform.

9. Python is a Portable language: Python language is also a portable language. For example, if we have Python code for Windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

10. Python is an Integrated language: Python is also an Integrated language because we can easily integrate Python with other languages like C, C++, etc.

11. Interpreted Language: Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile Python code this makes it easier to debug our code. The source code of Python is converted into an immediate form called bytecode.

12. Large Standard Library : Python has a large standard library that provides a rich set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in Python such as regular expressions, unit-testing, web browsers, etc.

13. Dynamically Typed Language: Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a

variable is decided at run time not in advance because of this feature we don’t need to specify the type of variable.

14. Frontend and backend development: With a new project py script, you can run and write Python codes in HTML with the help of some simple tags <py-script>, <py-env>, etc. This will help you do frontend development work in Python like javascript. Backend is the strong forte of Python it’s extensively used for this work cause of its frameworks like Django and Flask.

15. Allocating Memory Dynamically: In Python, the variable data type does not need to be specified. The memory is automatically allocated to a variable at runtime when it is given a value.

CHAPTER -4

DATA STRUCTURE METHODOLOGY

4.1 String

String is a data structure in Python Programming that represents a sequence of characters. It is an immutable data type, meaning that once you have created a string, you cannot change it. Python String are used widely in many different applications, such as storing and manipulating text data, representing names, addresses, and other types of data that can be represented as text.

4.2 List

Lists are the simplest containers that are an integral part of the Python language. Lists need not be homogeneous always which makes it the most powerful tool in Python. A single list may contain DataTypes like Integers, Strings, as well as Objects. Lists are mutable, and hence, they can be altered even after their creation.

Creating a List in Python

Lists in Python can be created by just placing the sequence inside the square brackets[]. Unlike Sets, a list doesn’t need a built-in function for its creation of a list.

CHAPTER-5
MODULES

5.1 Displaying the menu

Function Name: menu()

Description: This function serves as the main entry point for user interaction with the music playlist application. It displays a menu of options, prompting users to select an action to perform.

5.2 Song selection

Function name: songSelection()

Description: This function facilitates the process of selecting songs from a predefined list. It guides users through the selection process, ensuring they choose a valid song.

5.3 User playlist

Function name: userPlaylist()

Description: This function allows users to manage their personalized playlists, including adding, removing, or editing songs. It provides a streamlined interface for playlist management tasks.

5.4 Exit session module

Function name: exitSession()

Description: This function gracefully terminates the music playlist application session, ensuring a smooth exit for the user. It performs necessary cleanup tasks before ending the session

CHAPTER – 6
ERROR MANAGEMENT

1.1. Input Validation

Input validation plays a pivotal role in software development, ensuring the reliability, security, and stability of applications. In the context of error management, robust input validation mechanisms are crucial for handling and preventing potential issues arising from incorrect, malformed, or malicious user inputs. Within the realm of software development using tools like Visual Studio, implementing effective input validation strategies involves scrutinizing and verifying user inputs to ensure they meet predefined criteria and conform to expected formats before processing.

This process involves various techniques such as range checks, data type validation, length validation, format validation (e.g., email addresses, phone numbers), and input sanitization to prevent injection attacks like SQL injection or cross-site scripting (XSS). For instance, Visual Studio supports the integration of validation libraries and frameworks, enabling developers to perform comprehensive checks on user inputs, reducing the likelihood of vulnerabilities and improving the overall robustness of the application.

By applying stringent input validation mechanisms throughout the codebase, developers can fortify their applications against potential errors, exceptions, and security threats stemming from invalid inputs. Furthermore, incorporating error handling routines and providing informative feedback to users when invalid inputs are detected not only enhances user experience but also aids in diagnosing and rectifying input-related issues, contributing to the overall reliability and resilience of software systems developed within the Visual Studio environment.

Ultimately, meticulous input validation serves as a critical component of error management, preemptively addressing potential pitfalls related to user inputs and reinforcing the integrity and security of software applications.

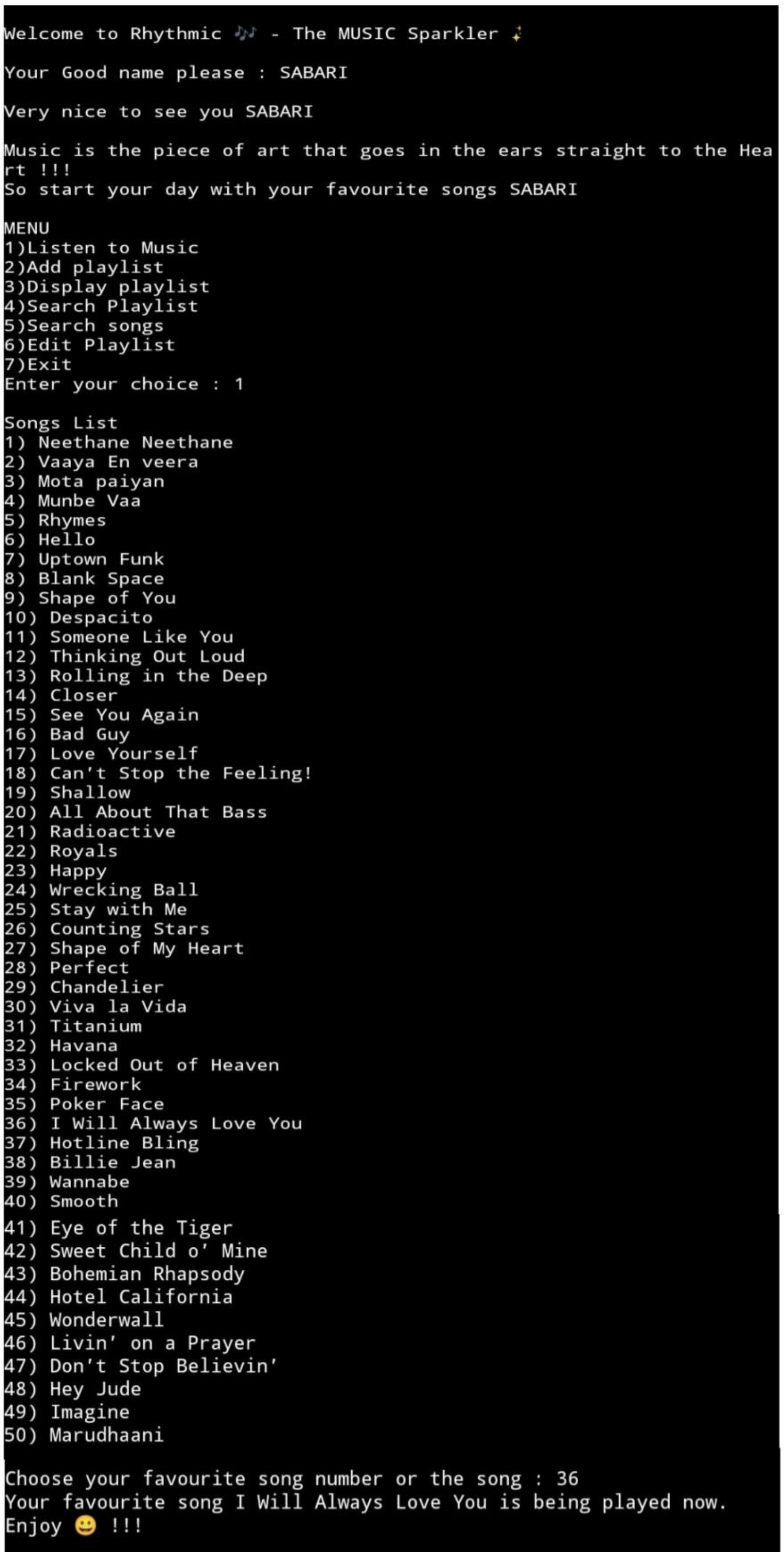
6.2 Exception handling

Exception handling in data structures is a critical aspect of software development, addressing unforeseen errors that may occur during operations. It encompasses various error types, such as array out-of-bounds access or operations on empty data structures. By throwing exceptions in response to errors, developers can prevent runtime failures and enable graceful recovery. Utilizing try-catch blocks allows for the isolation of error-prone code and the implementation of custom strategies for handling exceptions. This not only enhances the overall stability of the application but also facilitates debugging and maintenance by providing informative error messages. Additionally, customexceptions can be defined for specific scenarios, offering more granular control over error handling. Well-documented exception handling practices guide developers on effectively addressing errors, contributing to the creation of robust, reliable, and user-friendly software systems. By encapsulating file-related operations within try-except blocks, the program can gracefully handle scenarios such as file not found, permission errors, or unexpected file formats, ensuring uninterrupted program execution and preventing data loss or corruption..

CHAPTER 7
RESULT AND CONCLUSION

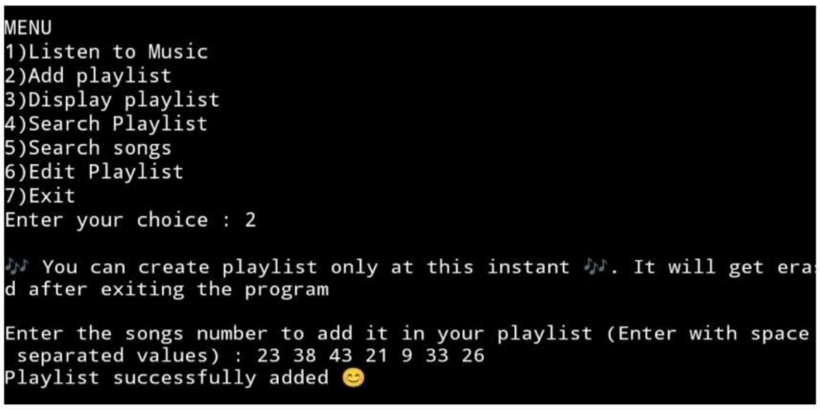
7.1 RESULT

1.Listen to Music



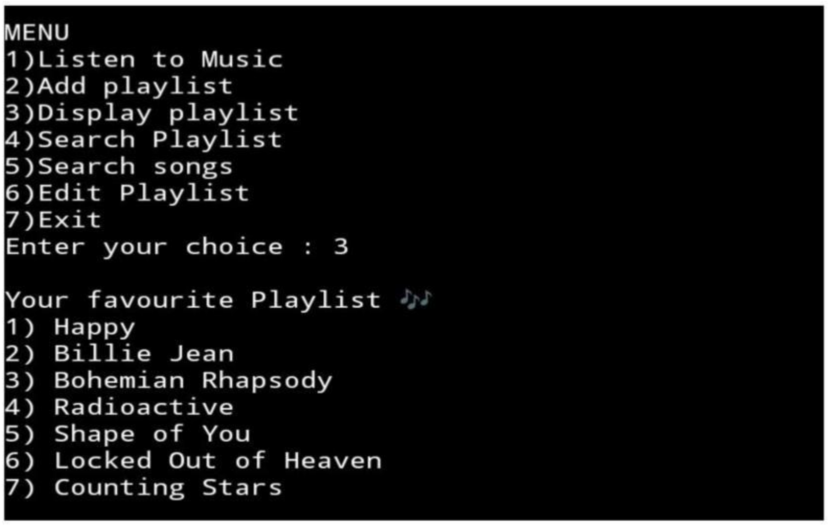
```
Welcome to Rhythmic 🎶 - The MUSIC Sparkler ✨

Your Good name please : SABARI

Very nice to see you SABARI

Music is the piece of art that goes in the ears straight to the Hea
rt !!!
So start your day with your favourite songs SABARI

MENU
1)Listen to Music
2)Add playlist
3)Display playlist
4)Search Playlist
5)Search songs
6)Edit Playlist
7)Exit
Enter your choice : 1

Songs List
1) Neethane Neethane
2) Vaaya En veera
3) Mota paiyan
4) Munbe Vaa
5) Rhymes
6) Hello
7) Uptown Funk
8) Blank Space
9) Shape of You
10) Despacito
11) Someone Like You
12) Thinking Out Loud
13) Rolling in the Deep
14) Closer
15) See You Again
16) Bad Guy
17) Love Yourself
18) Can't Stop the Feeling!
19) Shallow
20) All About That Bass
21) Radioactive
22) Royals
23) Happy
24) Wrecking Ball
25) Stay with Me
26) Counting Stars
27) Shape of My Heart
28) Perfect
29) Chandelier
30) Viva la Vida
31) Titanium
32) Havana
33) Locked Out of Heaven
34) Firework
35) Poker Face
36) I Will Always Love You
37) Hotline Bling
38) Billie Jean
39) Wannabe
40) Smooth
41) Eye of the Tiger
42) Sweet Child o' Mine
43) Bohemian Rhapsody
44) Hotel California
45) Wonderwall
46) Livin' on a Prayer
47) Don't Stop Believin'
48) Hey Jude
49) Imagine
50) Marudhaani

Choose your favourite song number or the song : 36
Your favourite song I Will Always Love You is being played now.
Enjoy 😀 !!!
```

2.Add Playlist



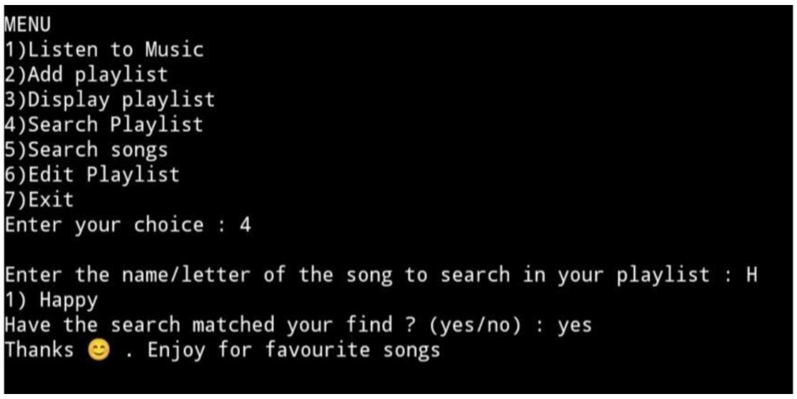
```
MENU
1)Listen to Music
2)Add playlist
3)Display playlist
4)Search Playlist
5)Search songs
6)Edit Playlist
7)Exit
Enter your choice : 2

🎶 You can create playlist only at this instant 🎶. It will get era
d after exiting the program

Enter the songs number to add it in your playlist (Enter with space
 separated values) : 23 38 43 21 9 33 26
Playlist successfully added 😊
```

3.Display Playlist



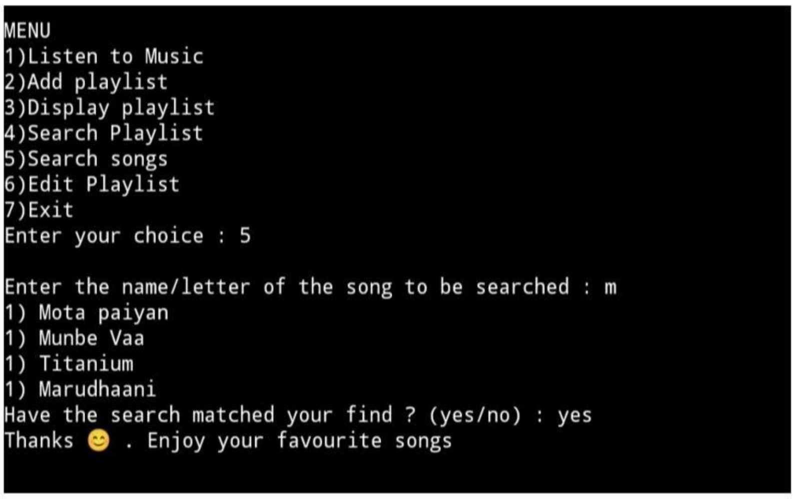
```
MENU
1)Listen to Music
2)Add playlist
3)Display playlist
4)Search Playlist
5)Search songs
6)Edit Playlist
7)Exit
Enter your choice : 3

Your favourite Playlist 🎶
1) Happy
2) Billie Jean
3) Bohemian Rhapsody
4) Radioactive
5) Shape of You
6) Locked Out of Heaven
7) Counting Stars
```

4.Search Playlist



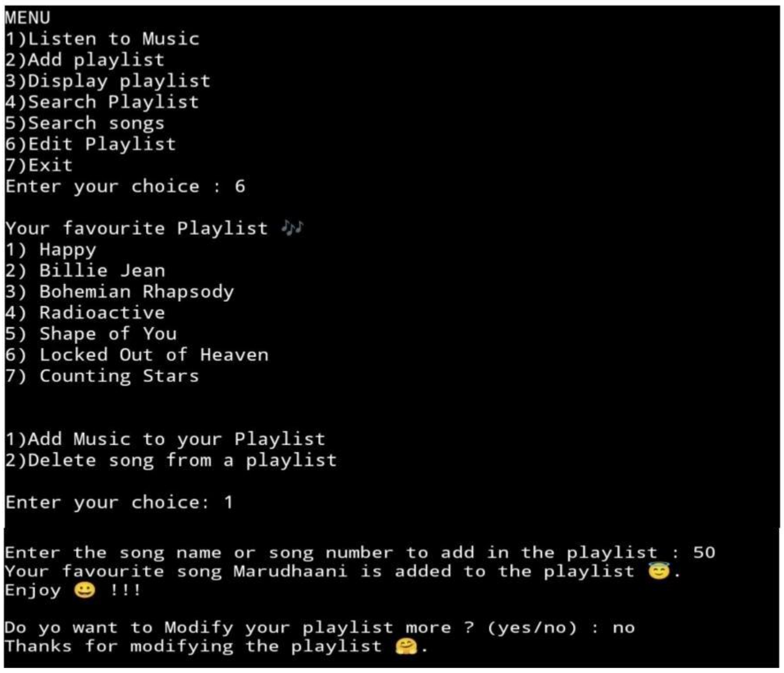
```
MENU
1)Listen to Music
2)Add playlist
3)Display playlist
4)Search Playlist
5)Search songs
6)Edit Playlist
7)Exit
Enter your choice : 4

Enter the name/letter of the song to search in your playlist : H
1) Happy
Have the search matched your find ? (yes/no) : yes
Thanks 😊 . Enjoy for favourite songs
```

5.Search Songs



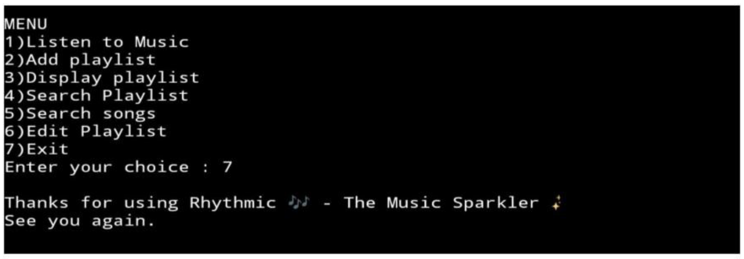
```
MENU
1)Listen to Music
2)Add playlist
3)Display playlist
4)Search Playlist
5)Search songs
6)Edit Playlist
7)Exit
Enter your choice : 5

Enter the name/letter of the song to be searched : m
1) Mota paiyan
1) Munbe Vaa
1) Titanium
1) Marudhaani
Have the search matched your find ? (yes/no) : yes
Thanks 😊 . Enjoy your favourite songs
```

6.Edit Playlist



```
MENU
1)Listen to Music
2)Add playlist
3)Display playlist
4)Search Playlist
5)Search songs
6)Edit Playlist
7)Exit
Enter your choice : 6

Your favourite Playlist 🎶
1) Happy
2) Billie Jean
3) Bohemian Rhapsody
4) Radioactive
5) Shape of You
6) Locked Out of Heaven
7) Counting Stars


1)Add Music to your Playlist
2)Delete song from a playlist

Enter your choice: 1

Enter the song name or song number to add in the playlist : 50
Your favourite song Marudhaani is added to the playlist 😇.
Enjoy 😀 !!!

Do yo want to Modify your playlist more ? (yes/no) : no
Thanks for modifying the playlist 🤗.
```

7.Exit



```
MENU
1)Listen to Music
2)Add playlist
3)Display playlist
4)Search Playlist
5)Search songs
6)Edit Playlist
7)Exit
Enter your choice : 7

Thanks for using Rhythmic 🎶 - The Music Sparkler ✨
See you again.
```

7.2 Discussion

A music recommendation system implemented in Python offers a fascinating intersection of technology, data analysis, and user experience. Here's a discussion on various aspects of such a system. One of the primary goals of a music recommendation system is to provide personalized recommendations to users based on their listening history, preferences, and behavior. By analyzing user interactions with songs, playlists, and artists, the system can generate recommendations tailored to each user's unique tastes. Data is at the heart of any recommendation system. In the case of music recommendation, data sources can include user listening history, song metadata (e.g., genre, artist, album), user ratings, and even external data like social media activity or concert attendance. Python's data processing libraries, such as Pandas, NumPy, and scikit-learn, are instrumental in cleaning, aggregating, and analyzing this data. There are various algorithms and techniques that can be employed for music recommendation, including collaborative filtering, content-based filtering, matrix factorization, and deep learning models. Python's extensive ecosystem of machine learning libraries, such as TensorFlow, PyTorch, and scikit-surprise, allows developers to experiment with and implement these algorithms effectively. Evaluating the performance of a recommendation system is crucial to ensure its effectiveness. Metrics such as precision, recall, Mean Average Precision (MAP), and AUC-ROC can be used to measure the system's accuracy and relevance of recommendations. Pyt hon provides libraries like scikit-learn and Surprise, which offer tools for evaluating recommendation algorithms.

CHAPTER 8
CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

Building a music recommendation system using Python involves various stages, from data collection and preprocessing to model selection and evaluation. Here, we provide a detailed conclusion based on the steps taken and insights gained during the development of the system. We collected data from multiple sources such as Spotify API, user listening histories, and music metadata databases. Data Cleaning, Handling missing values, removing duplicates, and standardizing formats were critical steps. This ensured that the data used was reliable and consistent. Feature Engineering*: We extracted and created relevant features, such as song popularity, user preferences, genre tags, and audio features like tempo, key, and loudness. We explored several recommendation techniques: Content-Based Filtering*: This approach used the attributes of songs and users. By comparing the features of songs liked by users, we recommended similar tracks. This method worked well but had limitations in discovering new or diverse music. Collaborative Filtering*Implemented using methods like Matrix Factorization and K-Nearest Neighbors (KNN), this approach utilized user interactions. It provided recommendations based on the listening habits of similar users, effectively uncovering new music. Hybrid Models Combining content-based and collaborative filtering, these models aimed to leverage the strengths of both approaches. Hybrid models often provided the best results in terms of relevance and diversity. Model Evaluation. We used precision, recall, F1 score.

8.1 Future Scope

Enhance the recommendation system by incorporating content-based filtering, which recommends items similar to those the user has liked in the past based on item attributes such as genre, artist, lyrics, etc. Combine collaborative filtering and content-based filtering techniques to leverage the strengths of both approaches and provide more accurate and diverse recommendations. Explore matrix factorization techniques like Singular Value Decomposition (SVD) or Alternating Least Squares (ALS) to decompose the user-item interaction matrix and uncover latent factors that influence user preferences. Utilize deep learning models such as neural networks or convolutional neural networks (CNNs) to extract intricate patterns and features from music data, potentially leading to more sophisticated recommendation models. Natural Language Processing (NLP): Integrate NLP techniques to analyze song lyrics, artist descriptions, and user reviews, allowing for a deeper understanding of music content and preferences. Develop mechanisms to provide real-time recommendations based on dynamic user interactions and preferences, enabling the system to adapt to users' changing tastes. Precision (MAP) to assess the performance of the recommendation system and fine-tune the algorithms accordingly.

REFERENCES

1.Anurag Gupta, IPS Jharkhand, GP Biswass, “ Python Programming,Problem Solving, Packages and Libraries”,McGraw Hill Education (India) Private Limited, 2019 ISBN-13:978-93-5316-800-1

2.Reema Theraja , “Python Programming: Using Problem Solving Approach” Oxford Higher Education,2017.

3.Gowrishankar S, Veena A, “Introduction to Python Programming”, 1st Edition,CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372

4.https//:w3schools.com

APPENDIX

```
print("\nWelcome to Rhythmic \U0001F3B6 - The MUSIC Sparkler

\U00002728\n".center(25)) user_name=input("Your Good name please : ")
print("\nVery nice to see you %s"%(user_name))

print("\nMusic is the piece of art that goes in the ears straight to the Heart

!!!\nSo start your day with your favourite songs",user_name,"\n")

def printlist(l): print("\nSongs List") i=1

for song in l:

print(str(i)+")",song) i+=1

print("\n")

def ListenMusic(b):

if ord(b[0])>=65 and ord(b[0])<=122:

print("Your favourite song %s is being played now.\nEnjoy \U0001F600
!!!\n"%(b))

elif int(b)>=0 and int(b)<=len(songs_list):

print("Your favourite song %s is being played

now.\nEnjoy \U0001F600 !!!\n"%(songs_list[int(b)-1]))

else:

print("Invalid entry , try again \U0001F614\n") def AddPlaylist(b):

for i in b:

if i >0 and i<=len(songs_list): playlist.append(songs_list[i-1])

else:

print("Song number %d doesn't exsist \U0001F632 , hence could not add it into
the playlist"%(i))
```

```
def showPlaylist(playlist): a=1

if len(playlist)==0:

print("Your playlist is empty \U0001F615")

else:

print("\nYour favourite Playlist \U0001F3B6") for i in playlist:

print(str(a)+")",i) a+=1

print("\n")

def searchPlaylist(b):

if len(playlist)==0:

print("Your playlist is empty \U0001F615\n") return

for i in playlist: a=1

if i.lower()==b.lower() or (i.lower()).startswith(b.lower()) or
(i.lower()).endswith(b.lower()):

print(str(a)+")",i) a+=1

c=input("Have the search matched your find ? (yes/no) : ") if c.lower()=="yes":

print("Thanks \U0001F60A . Enjoy for favourite songs")

else:

print("OOPS \U0001F614 , Sorry unable to fetch the song")

print("\n")

def searchSongs(b):

for i in songs_list: a=1

if i.lower()==b.lower() or (i.lower()).startswith(b.lower()) or
(i.lower()).endswith(b.lower()):
```

```
print(str(a)+")",i)

a+=1

c=input("Have the search matched your find ? (yes/no) : ") if c.lower()=="yes":

print("Thanks \U0001F60A . Enjoy your favourite songs")

else:

print("OOPS \U0001F614 , Sorry unable to fetch the song") print("\n")

def editPlaylist(playlist): if len(playlist)==0:

print("\nPlaylist is empty \U0001F61F . Hence cannot modify
\U0001F4DD\n")

return

else:

showPlaylist(playlist)

ch='yes'

while (ch=='yes'):

print("1)Add Music to your Playlist\n2)Delete song from a playlist\n")

ch=int(input("Enter your choice: ")) if ch==1:

printlist(songs_list)

b=input("Enter the song name or song number to add in the playlist : ")

if ord(b[0])>=65 and ord(b[0])<=122: if b in playlist:

print("Selected song already in the

playlist \U0001F604")

else:

playlist.append(b)

print("Your favourite song %s is added to the playlist \U0001F607.\nEnjoy
\U0001F600 !!!\n"%(b))

elif int(b)>0 and int(b)<=len(songs_list): if songs_list[int(b)-1] in playlist:

print("Selected song already in the

playlist \U0001F604")

else:

playlist.append(songs_list[int(b)-1])

print("Your favourite song %s is added to the playlist \U0001F607.\nEnjoy
\U0001F600

!!!\n"%(songs_list[int(b)-1]))

else:

print("Invalid entry , try again

\U0001F614\n")

elif ch==2:

showPlaylist(playlist)

b=input("Enter the song name or song number to be deleted in the playlist : ")

if ord(b[0])>=65 and ord(b[0])<=122: if b not in playlist:

print("Selected song is not present

in the playlist \U0001F610")

else:

playlist.remove(b)

print("Selected song %s is deleted from the playlist successfully
\U0001FAE1.\n"%(b))

elif int(b)>0 and int(b)<=len(songs_list):

d=playlist[int(b)-1]
```

```
playlist.remove(playlist[int(b)-1]) print("Selected song %s is deleted

from the playlist successfully \U0001FAE1.\n"%(d)) else:

print("Invalid entry , try again

\U0001F614\n")

else:

print("Invalid choice \U0001F614. Try

Again\n")

ch=input("Do yo want to Modify your playlist more ?

(yes/no) : ")

print("Thanks for modifying the playlist \U0001F917.\n")

songs_list=["Neethane Neethane","Vaaya En veera","Mota paiyan","Munbe
Vaa","Rhymes","Hello", "Uptown Funk", "Blank Space", "Shape of You",
"Despacito", "Someone Like You", "Thinking Out Loud", "Rolling in the
Deep", "Closer", "See You Again", "Bad Guy", "Love Yourself", "Can’t Stop
the Feeling!", "Shallow", "All About That Bass", "Radioactive", "Royals",
"Happy", "Wrecking Ball", "Stay with Me", "Counting Stars", "Shape of My
Heart", "Perfect", "Chandelier", "Viva la Vida", "Titanium", "Havana",
"Locked Out of Heaven", "Firework", "Poker Face" Love You", "Hotline
Bling", "Billie Jean", "Wannabe", "Smooth", "Eye of the Tiger", "Sweet Child
o’ Mine", "Bohemian Rhapsody", "Hotel California", "Wonderwall",  on a
Prayer", "Don’t Stop Believin’", "Hey Jude", "Imagine","Marudhaani"]

playlist=[] ch=1 while(True):

print("MENU\n1)Listen to Music\n2)Add playlist\n3)Display
playlist\n4)Search Playlist\n5)Search songs\n6)Edit Playlist\n7)Exit")

ch=int(input("Enter your choice : "))

if ch==1:

printlist(songs_list)

b=input("Choose your favourite song number or the song : ") ListenMusic(b)
```

```
elif ch==2:

printlist(songs_list)

print("\n\U0001F3B6 You can create playlist only at this instant \U0001F3B6. It will get erased after exiting the program\n".center(25))

b=list(map(int,input("Enter the songs number to add it in your playlist (Enter with space separated values) : ").split()))

AddPlaylist(b)

print("Playlist successfully added \U0001F60A\n") elif ch==3:

showPlaylist(playlist)

elif ch==4:

b=input("\nEnter the name/letter of the song to search in your playlist : ")

searchPlaylist(b)

elif ch==5:

b=input("\nEnter the name/letter of the song to be searched :

")

searchSongs(b)

elif ch==6:

editPlaylist(playlist)

elif ch==7:

print("\nThanks for using Rhythmic \U0001F3B6 - The Music Sparkler \U00002728\nSee you again.\n")

break

else:

print("Invalid choice, try again")
```