DrillBit

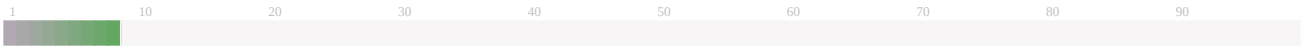
The Report is Generated by DrillBit Plagiarism Detection Software

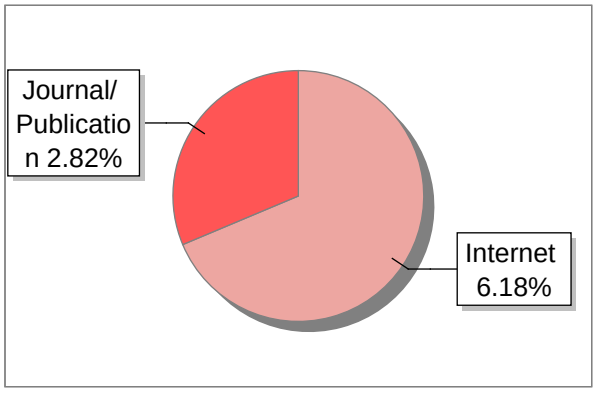
Submission Information

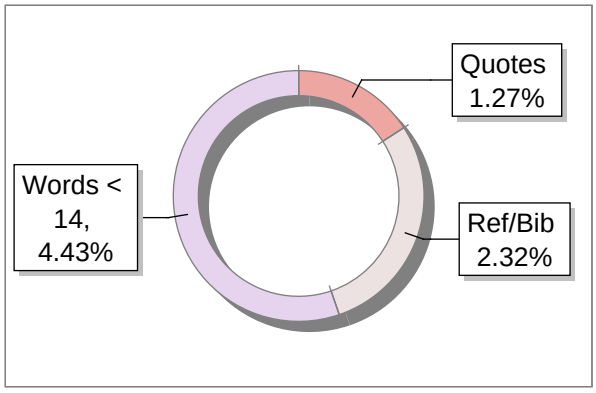
Author Name	Aman Kumar
Title	PG REPORT
Paper/Submission ID	1422784
Submitted by	chieflibrarian@msrit.edu
Submission Date	2024-02-12 09:18:33
Total Pages	35
Document type	Project Work

Result Information

Similarity **9 %**







Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Sources: Less than 14 Words %	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	No

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

9	19	A
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	qdoc.tips	2	Internet Data
2	qdoc.tips	1	Internet Data
3	www.auhd.site	1	Publication
4	www.dx.doi.org	<1	Publication
5	www.fantinisgamingshow.com	<1	Internet Data
6	IEEE 2012 5th International Symposium on Resilient Control Systems (by	<1	Publication
7	IEEE 2018 3rd Digital Heritage International Congress (Digital Herit	<1	Publication
8	blueskygraphics.co.uk	<1	Internet Data
9	Pressures on regional games in a dominant metropolitan culture the case of shin by Whitson-1983	<1	Publication
10	typeset.io	<1	Internet Data
11	dspace.daffodilvarsity.edu.bd 8080	<1	Publication
12	moam.info	<1	Internet Data
13	moam.info	<1	Internet Data

14	moam.info	<1	Internet Data
15	passle-net.s3.amazonaws.com	<1	Publication
16	pwskills.com	<1	Internet Data
17	repositorio.unal.edu.co	<1	Publication
18	www.cockatooisland.gov.au	<1	Internet Data
19	ACM Press the 25th ACM International- Indianapolis, Indiana, USA (, by Yuan, Fajie Guo, G- 2016	<1	Publication

EXCLUDED PHRASES

1	**m s ramaiah institute of technology**
2	**msrit**
3	**bangalore**

1. Introduction

1.1 Overview

The Antakshari Assistant is a user-friendly software designed to enhance the experience of playing the traditional Indian musical game, Antakshari. It offers song suggestions based on categories, tracks scores, and prompts players with the ending letter of the previous song. With features like timer management and error handling, it ensures smooth gameplay. Additionally, it may include song playback for reference. By streamlining the gameplay process and providing helpful features, the Antakshari Assistant aims to make the game more enjoyable for players of all ages, enriching the experience of this beloved musical pastime.

1.2 Problem Definition

1 . We aim to create a music recommender system based on user preferences and listening history. There are many methods for recommender systems in use today, so a major component of our project will be in determining more the appropriate methods to augment the basic Neighborhood model to fit this particular application. Our data is unique in many ways, not the least of which is the lack of explicit feedback, i.e. ratings, from the users, so we will need to survey the field to find the right way to incorporate implicit feedback. It is also clear that incorporating song features to some degree will improve the accuracy of our recommendations, so we will need to find the appropriate method to utilize these features as well.

2 . Design and develop a platform to play antakshari game that incorporates our machine learning model . The game can be played by two or more players . The objective is to keep the game going for as long as possible by taking turns to sing or recite a word, phrase, or line from a song that begins with the last letter of the previous player's word or phrase.

For example, if the first player sings "Aaja Aaja Main Hoon Pyar Tera", the next player has to sing a song that begins with a line containing the last five words such as Aaja , Main, Hoon, Pyar and Tera .

The game continues until a player or team is unable to come up with a new word or phrase within a set time limit or until all the players have had a turn. The winner is determined by the number of points they accumulate during the game.

▸ Literature Survey

Recommendation Systems:

Collaborative Filtering:

Collaborative filtering methods, such as user-based and item-based approaches, have been extensively studied. Techniques like matrix factorization and singular value decomposition (SVD) are commonly employed.

Content-Based Filtering:

Content-based recommendation focuses on the attributes of items and users' preferences. Natural Language Processing (NLP) is often integrated to analyze content features.

Hybrid Recommendation Systems:

Combining collaborative and content-based approaches to overcome their individual limitations.Hybrid systems aim to provide more accurate and diverse recommendations.

Deep Learning in Recommendations:

Utilizing neural networks, deep learning models like neural collaborative filtering have shown promising results. Embedding techniques for representing users and items in a continuous space.

Context-Aware Recommendation:

Taking into account contextual information, such as time, location, and user behavior, to enhance recommendation accuracy. Context-aware models adapt to users' changing preferences.

Antakshari Assistant: Audio Processing and Speech-to-Text Techniques for real-time audio processing and converting microphone inputs to text, crucial for Antakshari Assistant. Machine Learning algorithms for efficient speech recognition.

Music Recommendation Systems

Exploring recommendation algorithms specifically designed for music playlists. Understanding user preferences based on their song choices and tailoring recommendations accordingly.

Multiplayer Game Design:

Design considerations for creating multiplayer gaming experiences, including room creation, player switching, and interactive features. User experience and engagement in multiplayer scenarios.

Time Constraints in Gaming:

Implementing time constraints for game elements, ensuring a dynamic and challenging gameplay experience. Balancing time limits to maintain player interest.

User Interface and User Experience (UI/UX):

Studies on creating intuitive and user-friendly interfaces for applications, especially in the context of gaming and music recommendation.

Links

https://ehs.stanford.edu/topic/musicrecomendation

2. Hardware and Software Requirements

2.1 . Hardware Requirements

- A computer with sufficient processing power to handle data processing and modeling tasks efficiently.
- Adequate RAM (Random Access Memory) to accommodate the dataset size and machine learning algorithms' memory requirements.
- Sufficient storage space to store the dataset, model files, and any intermediate data generated during the analysis.

2.2 3.2. Software Requirements

- **Data Visualization Tools:**To make the analysis or the dashboard ,in this project I have used a data visualization tool called PowerBI where I have prepared the main dashboard required for the project.
- **Machine Learning Libraries:** Essential libraries like scikit-learn for building and training machine learning models. Additional libraries like joblib or matplot may be required for the further analysis.
- **Web Browser:** Required for accessing documentation, and tutorials related to Python, machine learning, and data analysis.
- **Compatible Operating System:** Windows10
- **Python Programming Language:** Essential for data preprocessing, model development, and evaluation. Python libraries like NumPy, pandas, scikit-learn, and pickle will be utilized extensively.
- **Integrated Development Environment (IDE):** Recommended IDEs include Jupyter Notebook, Google Colab, PyCharm, or Visual Studio Code for coding and experimentation.

3. Software Requirements Specification

3.A Nonfunctional Requirements

Functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and portability.

3.1 Performance Requirements

- The load time for user interface screens shall take no longer than two seconds.
- The log in information shall be verified within two seconds.
- Queries shall return results within five second.

3.2 Logical Database Requirements

The logical database requirements include the retention of the following data elements. This list is not a complete list and is designed as a starting point for development.

User Details

- User Id
- Username
- Password
- Prefrenced Genre
- Online / Offline

Song Details

- Song Id
- Song Name
- Artist
- Song Genre
- Lyrics

3.3 Security

Users will be able to log in to the System using their login credentials.Log in credentials of user would be in encrypted form in the database.

3.4 Reliability

Specify the factors required to establish the required reliability of the software system at time of delivery.

3.5 Availability

The system shall be available 24x7.

3. B. Functional Requirement

REQ. 1: Display a form for user to sign-upstem.

REQ. 2: Login Form for user

The input specifies the user interface element (button), while the output specifies the corresponding action or behavior (start or stop recording) triggered by interacting with that button. This functionality allows the user to control the recording process conveniently by simply pressing a single button to start or stop it.

REQ. 4: Create Room to support multiplayer in Antakshari assistant.

The input-output pair outlines the process of initiating and setting up an Antakshari game room on a digital platform, where users can specify the number of players and invite others to join the game

REQ. 5: Tossing for selection of first player to start the game.

The input-output pair outlines the process of initiating and setting up an Antakshari game room on a digital platform, where users can specify the number of players and invite others to join the game.

REQ. 6: Recording the microphone real time audio and conversion of audio to text.

The input-output pair describes a process where microphone audio input is converted into text output using a machine learning algorithm specifically trained for audio-to-text conversion, enabling seamless integration of spoken content into textual form for further analysis or utilization.

REQ. 7: Recommending the song based on last words of the previous player.

The input-output pair describes a process where microphone audio input is converted into text output using a machine learning algorithm specifically trained for audio-to-text conversion, enabling seamless integration of spoken content into textual form for further analysis or utilization. this system facilitates an interactive and engaging game where players take turns providing cues and preferences to receive personalized song recommendations, enhancing the entertainment experience through music exploration and discovery.

REQ. 8: Time constraints. Fixed time is allocated to players to sing the song.

This input-output pair outlines a game scenario where players must adhere to a time constraint for song selection and singing, with failure to comply resulting in losing the game.

REQ. 9: Players can start and stop microphone audio recording.

This functionality provides a convenient and intuitive way for users to control the recording process using a single button. It simplifies the user experience by eliminating the need for complex commands or multiple controls, making it easy for users to start and stop recordings as needed.

REQ. 10: Game Manager

It would manage processes related to Game progression like playerswitching, deciding winner etc.

6.1 Database Requirements

The application shall use a relational database management system (RDBMS) to store and manage data. Database backups shall be scheduled regularly, and the backup data shall be stored securely.

6.2 Legal Requirements

The application shall comply with data protection laws and regulations, ensuring the secure handling of customer and business data.

Terms of service and privacy policies shall be clearly communicated to users during onboarding.

Reuse Objectives

Code modularization and documentation practices shall be implemented to facilitate code reuse in future updates or expansions.

The application architecture shall support scalability to accommodate potential reuse in similar business environments.

4. System Design Description (SDD)

4.1 System Overview

There is only one entity : the user.

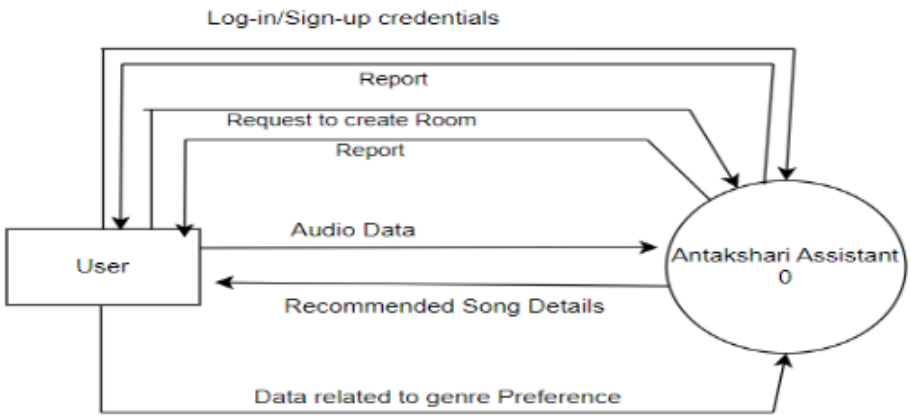
For login / signup, the user can send his / her credential details through login/sign-up page to the system.System would communicate with the database to confirm user credential. If the user is authenticated then the system would redirect the user to the homepage of the system.

For playing antakshari users have to request the system to create a room for playing. Then the system would create the room.Then the user has to send the invitation to other players.Players can join in the game room using an invitation from the user who created the room.

After each player has joined the game room, the game can be started. The Game Manager Subsystem will control all features of this game. During a player's term, the system will send data to the recommendation system. In response, the recommendation system would send back details of recommended songs.

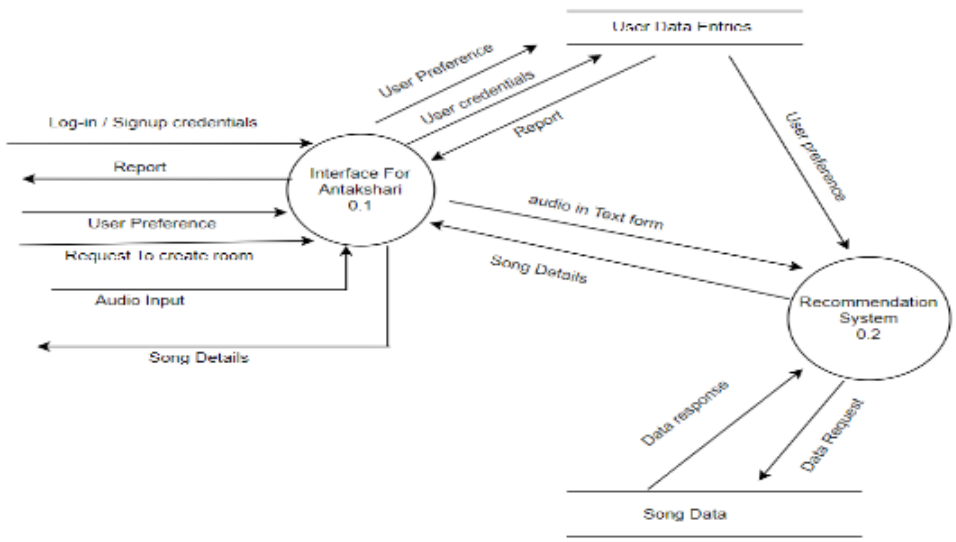
The whole system process can be understand by Given DFD (Data Flow Diagram) of System :

Level 0 DFD (Context Diagram)



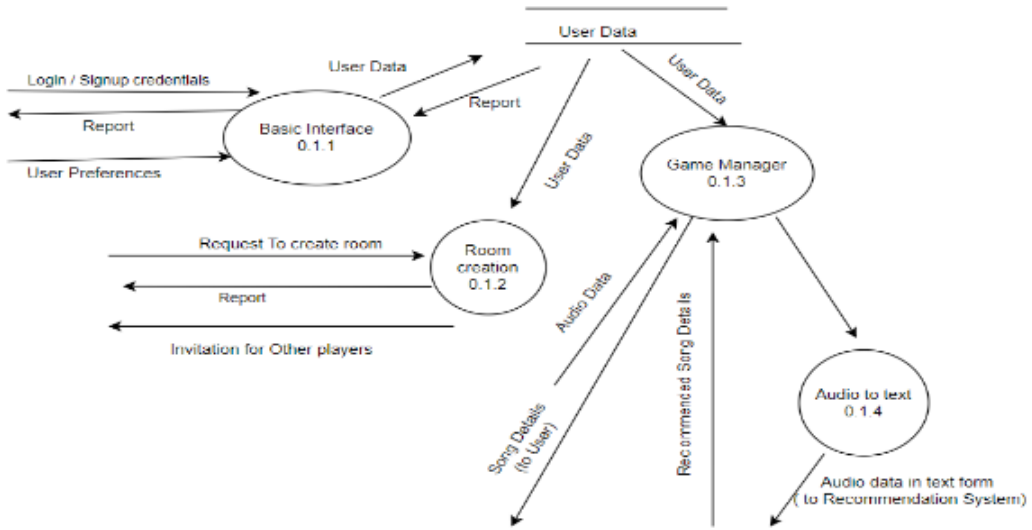
Level 0 Diagram

Level 1 DFD :



Level 1 DFD

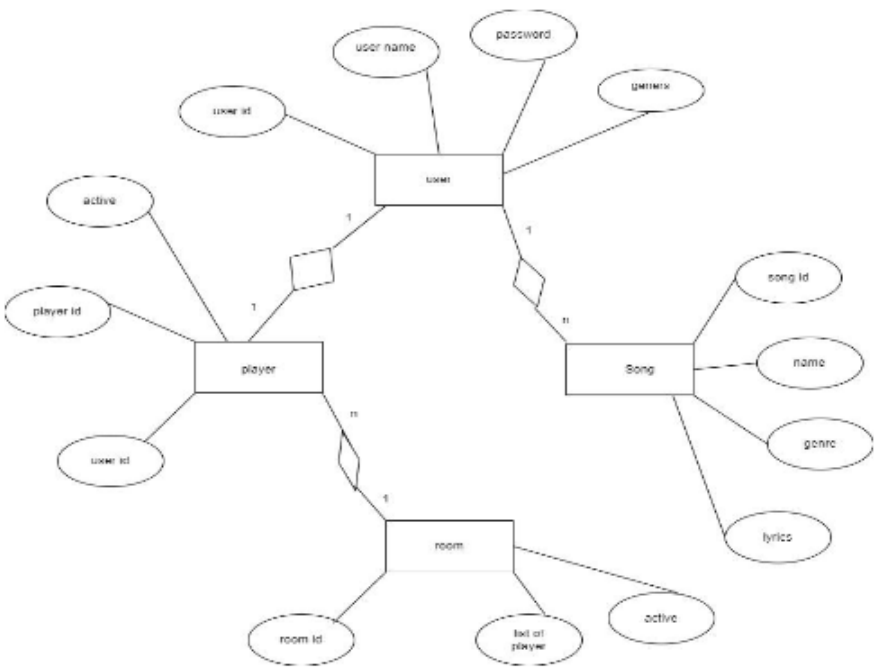
Level 2 DFD :



LEVEL 2 DFD

4.2 Database Design/Data Set Description

ER diagram (Entity - Relation Diagram):



Data Description

- **User Table**

Data Item	Type	Description
ID	Integer	Unique for every user
Username	Text	Name of User
Password	Text	User password in Encrypted form
Genres	Text	Preferenced genre by user

- **Player Table**

Data Item	Type	Description
ID	Integer	Unique for every user
User Id	Integer	Id of User
active	bool	Indicate payer is active

- **Song Table**

Data Item	Type	Description
ID	Integer	Unique for every Song

Name	Text	Name of Song
Genre	Text	Genre of song
Lyrics	Text	Song Lyrics

- **Room Table**

Data Item	Type	Description
ID	Integer	Unique for every Song
Active	bool	To indicate room is active
created_by	Text	Author of room
players	List	List of users in the rooms

- **User - room Relation**

Data Item	Type	Description
User ID	Integer	Foreign key for room table

- **User - Song Relation**

Data Item	Type	Description
User ID	Integer	Foreign key for user

Song Id	Integer	Foreign key for Song
Rating	Integer	Rating of song by user

4.3 Functional Design

Describe the functionalities of the system: Antakshari Assistant is a software application designed to assist users in playing the popular Indian musical game called "Antakshari." The functionalities of the Antakshari Assistant typically include providing prompts for selecting songs based on the ending letter of the previous song, suggesting songs based on various categories or themes, keeping track of scores, and facilitating smooth gameplay.

4.3.1 Here are the functionalities of the Antakshari Assistant:

Song Suggestions: The Antakshari Assistant provides suggestions for songs based on various categories such as genre, singer, movie, or theme. Users can choose a category and receive song suggestions accordingly.

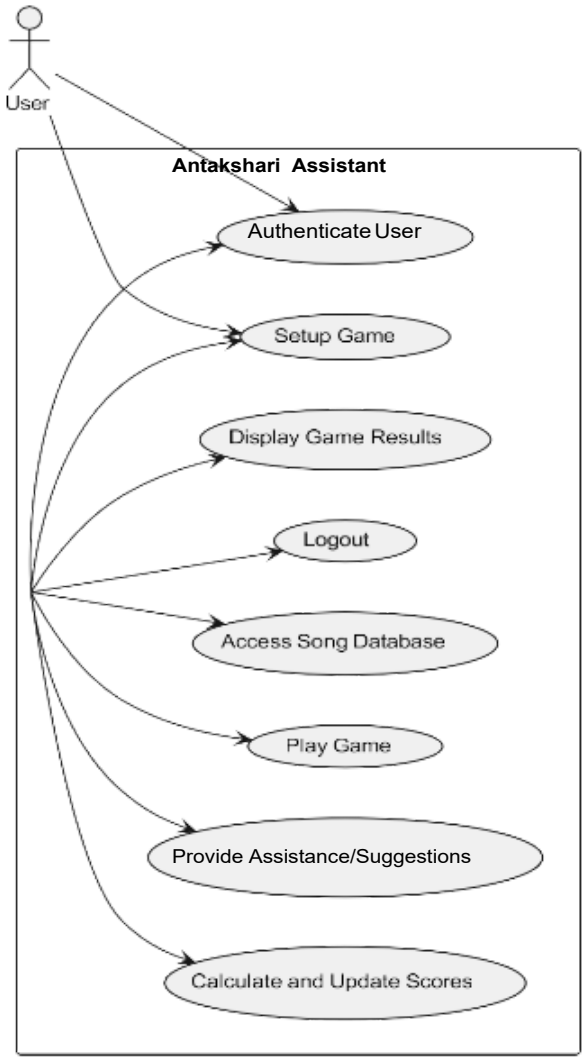
Ending Letter Prompt: The Assistant prompts players with the ending letter of the previous song. This helps players continue the game by selecting a song that starts with the prompted letter.

Score Tracking: The Assistant keeps track of scores for each player or team participating in the game. It updates the scores based on the performance of the players during the game.

Timer Management: The Assistant manages the timing of each round or turn in the game. It may include features such as starting and stopping timers for turns, setting time limits for selecting songs, and signaling the end of each round.

Error Handling: The Assistant handles errors or invalid inputs from users gracefully. It provides appropriate feedback and guidance to players in case of incorrect selections or actions.

Song Playback: Optionally, the Assistant may include a feature to play a short snippet of the selected song for reference or entertainment purposes.



Use case diagram

Actors:

Player: The human player participating in the Antakshari game.

System: The Antakshari Assistant software system.

Descriptions:

Player:

Select Category: The player selects a category such as genre, singer, movie, or theme to receive song suggestions.

Choose Song: The player selects a song based on the provided suggestions.

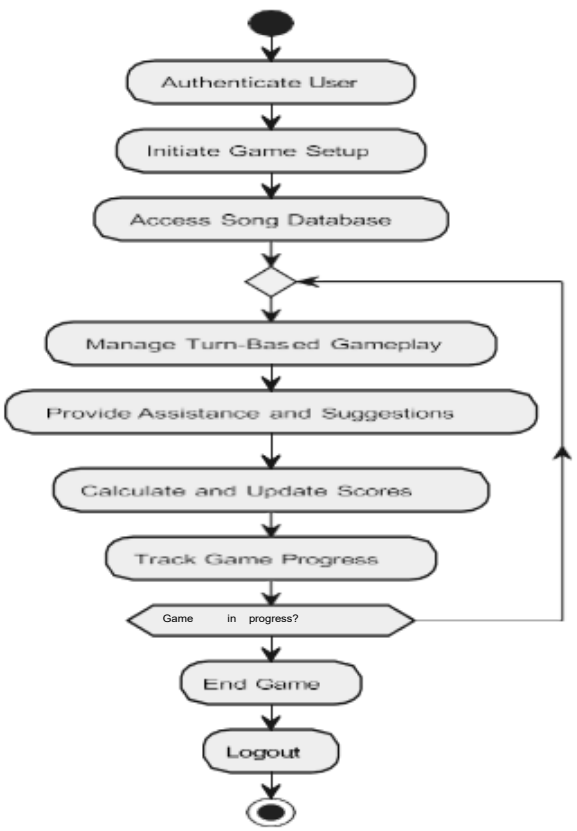
Score Update: The player's score is updated based on their performance in the game.

System:

Provide Song Suggestions: The system provides song suggestions based on the selected category.

Ending Letter Prompt: The system prompts the player with the ending letter of the previous song.

4.3.2 Behavioral design: (Activity Diagram)



5. Implementation

1 : Recommendation system

For recommending songs, the Game manager sub-part would send a request to the recommendation sub system through API. With this request the game manager would send the data like last words, player preference etc. By using this data recommendation system would recommend the song and would send back the recommended songs details to the game manager.

Code for recommender System

```
import numpy as np
from sklearn.neighbors import NearestNeighbors
```

```
import numpy as np
from sklearn.neighbors import NearestNeighbors
class recommander:
    def __init__(self,userid):
        self.userid=userid
        self.song_listened=dict_user_song[self.userid]
        self.sample=[[60]*50 for i in range(191801)]

    def max_rating(self):
        song_listened=dict_user_song[self.userid]
        ma=0
        max_songid=0
        for i in range(len(self.song_listened)):
```

```
            cur_rating=dict2[self.song_listened[i]][self.userid]
            if cur_rating>ma:
                ma=cur_rating
                max_songid=self.song_listened[i]
            ma=max(cur_rating,ma)
        data=dict2[max_songid]
        return max_songid
    def recommand(self,last_word):
        #sample=[]
        #for i in range(len())
        #samples = [[0., 0., 0.], [0., .5, 0.], [1., 1., .5]]


        valid_song_list=recommend_song_genre(last_word,['Pop'])
        for k in valid_song_list:
            if k in dict2.keys():
                self.sample[k]=dict2[k];


        data=dict2[self.max_rating()]

        #from sklearn.neighbors import NearestNeighbors
        neigh = NearestNeighbors(n_neighbors=5)
```

```
    neigh.fit(self.sample)



    #NearestNeighbors(n_neighbors=5)
    res=neigh.kneighbors([data])[1]
    #print(res)
    recommanded_song=[]
    for i in range(len(res[0])):
        print(songId[res[0][i]])
        recommanded_song.append(songId[res[0][i]])
    #return recommanded_song


fun=recommander(0)
#print(fun.recommander())
fun.recommand('love')
```

2 : Score calculator

For evaluating the score of a song, the Game manager sub-part would send a request to the score calculation sub system through API. With this request the game manager would send the data like lyrics. By using this score calculator system would calculate the score and would send it back to the game manager.

Code for Score Calculating

```
def match(txt1,txt2):
  s = ''
  i = 0
  flag = True
```

```
s2 = ''

for ch in txt2:

    if(ch.isalpha()):

        s2 = s2 + ch

while(i < len(s2)):

    if(txt1[i]=='['):

        flag = False

    if(flag and (txt1[i].isalpha())):

        s = s + txt1[i]

    if(txt1[i] == ']'):

        flag =True

    i = i+1

matcher = difflib.SequenceMatcher(None, s.lower(), s2.lower())

similarity_ratio = matcher.ratio()

score = similarity_ratio*(len(text) * 0.5 )

return score
```

3 : Fetch lyrics

For showing lyrics of songs to the end user, the Game manager sub-part would send a request to fetch the lyrics through this API.

Code for Fetching Lyrics

```
def getLyrics(id):

    d = list(df.get('Lyric'))

    return d[int(id)]
```

6. Testing

6.1 Description of Testing

Unit Testing:

Description: Unit testing involves testing individual components or units of code in isolation to ensure they function correctly.

With respect to Antakshari Assistant:

Unit tests for the Antakshari Assistant system would focus on testing individual functions or methods responsible for specific functionalities such as providing song suggestions, updating scores, managing timers, handling errors, etc.

Example: A unit test can be written to verify that the function responsible for suggesting songs based on a selected category returns the expected results.

Integration Testing:

Description: Integration testing involves testing the interaction between different components or modules of the system to ensure they work together as expected.

With respect to Antakshari Assistant:

Integration tests for the Antakshari Assistant system would focus on testing interactions between different components such as the UI, backend logic, and external services (if any).

Example: An integration test can be written to verify that the UI displays the correct song suggestions returned by the backend service based on the selected category.

System Testing:

Description: System testing involves testing the entire system as a whole to ensure it meets the specified requirements and functions correctly in its intended environment.

With respect to Antakshari Assistant:

System tests for the Antakshari Assistant system would focus on testing the system's end-to-end functionality, including user interactions, data flow, error handling, and performance.

Example: A system test can be written to simulate a complete gameplay session where users interact with the Antakshari Assistant to select categories, choose songs, update scores, and end the game. The test would verify that the entire process works smoothly and meets the user's expectations.

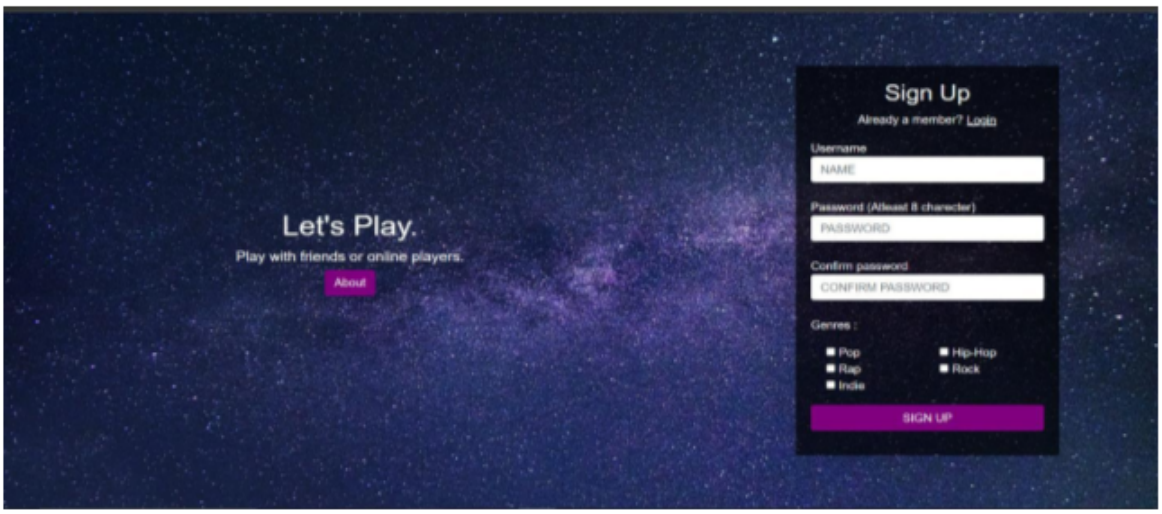
6.2 Test Cases

Test Suite

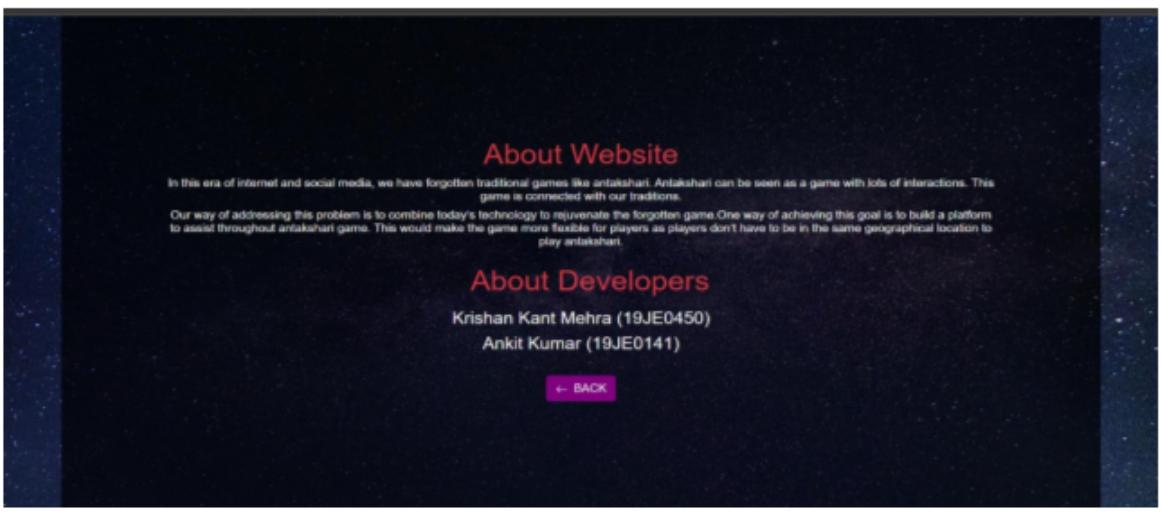
Test Case Id	Test case Objective	Prerequisite	Steps	Input data	Expected O	Actual Output	Status
Tc_01	Check password strength	Text Field should be enabled	1. Enter password in password field. 2. Enter submit	U:19je0141 P: 1234321	Password length should be greater than 8	Password length should be greater than 8	pass
Tc_02	Password matches to authenticate user	Text field should be enabled	1.enter username 2.enter wrong password 3. Enter submit	U:19je0141 P: wrong pwd	Authenticatio n error	Authenticatio n error	pass
Tc_03	Test player notification	Players should be logged in	1.select online player 2. Check that player get notified	NA	Player get notified	Player get notified	pass
Tc_04	Test player can pause and	Player should be longed in	1.select play	NA	Microphone get on and	Microphone get on and	pass

	play microphone	and joined a room	button to sing 2.select pauseto stop singing		then off	then off	
Tc_05	Test player switch	Player should be longed in and joined a room	1.select stop button to stop singing 2.check turn switch to next player	NA	Turn switch to next play	Turn switch to next play	pass
Tc_06	Test winning condition	Player should be longed in and joined a room	1.play game till round ends 2. Check player with maximum score wins	NA	player with maximum score wins	player with maximum score wins	pass

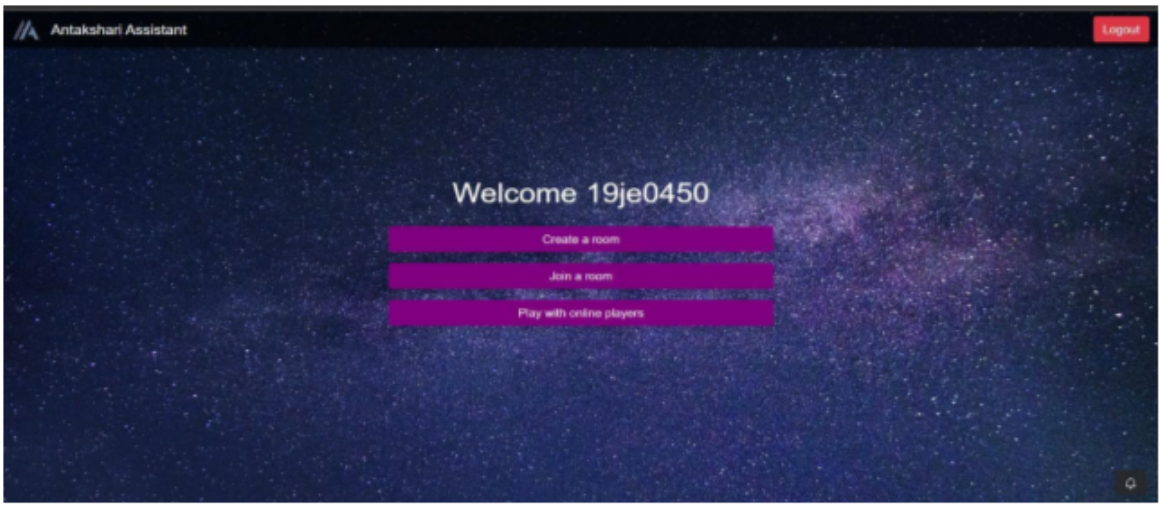
7. Results and Discussion



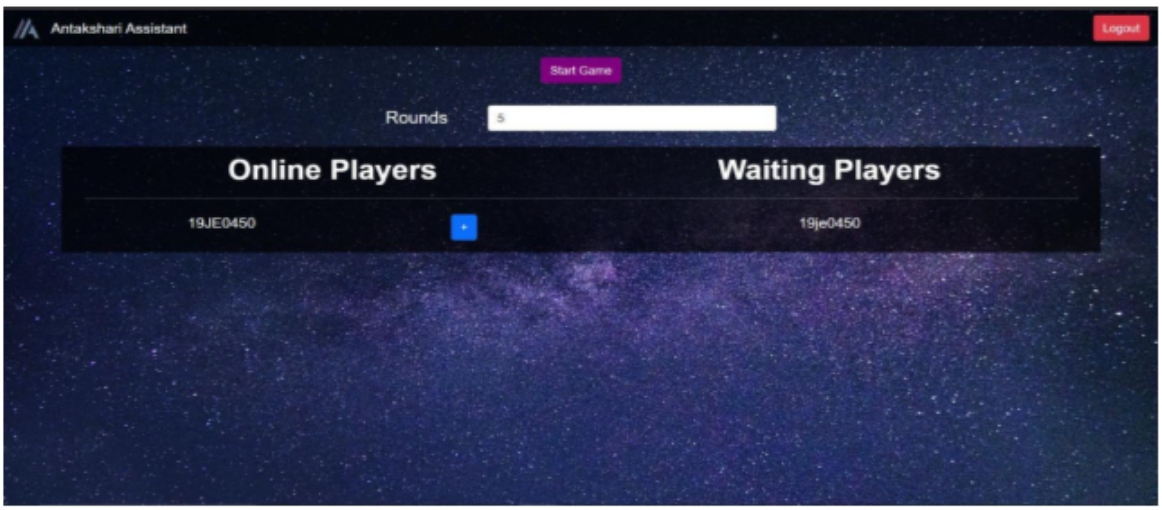
login/Signup page



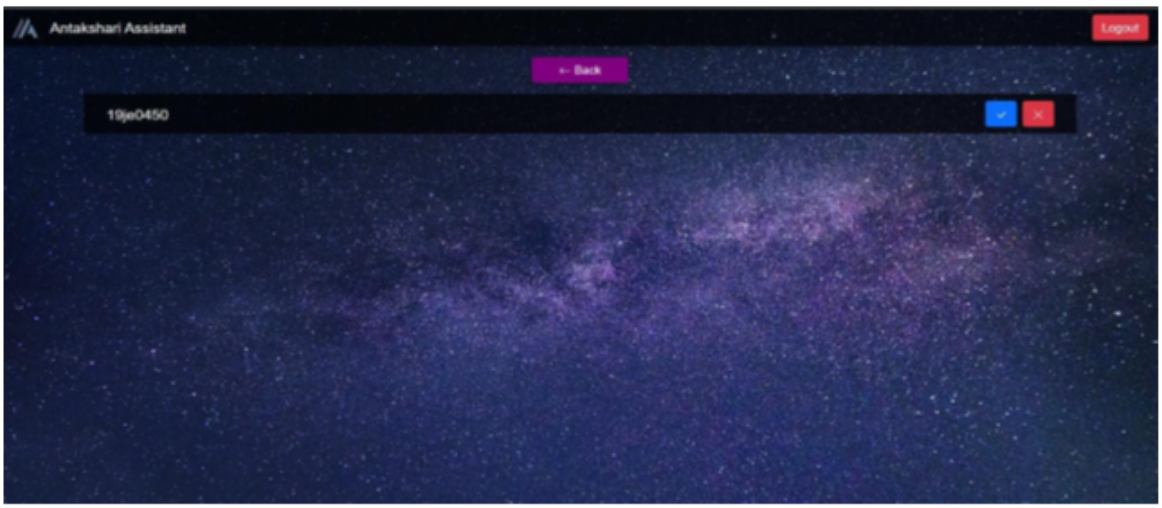
About Page


Antakshari Assistant
Logout
Welcome 19je0450
Create a room
Join a room
Play with online players

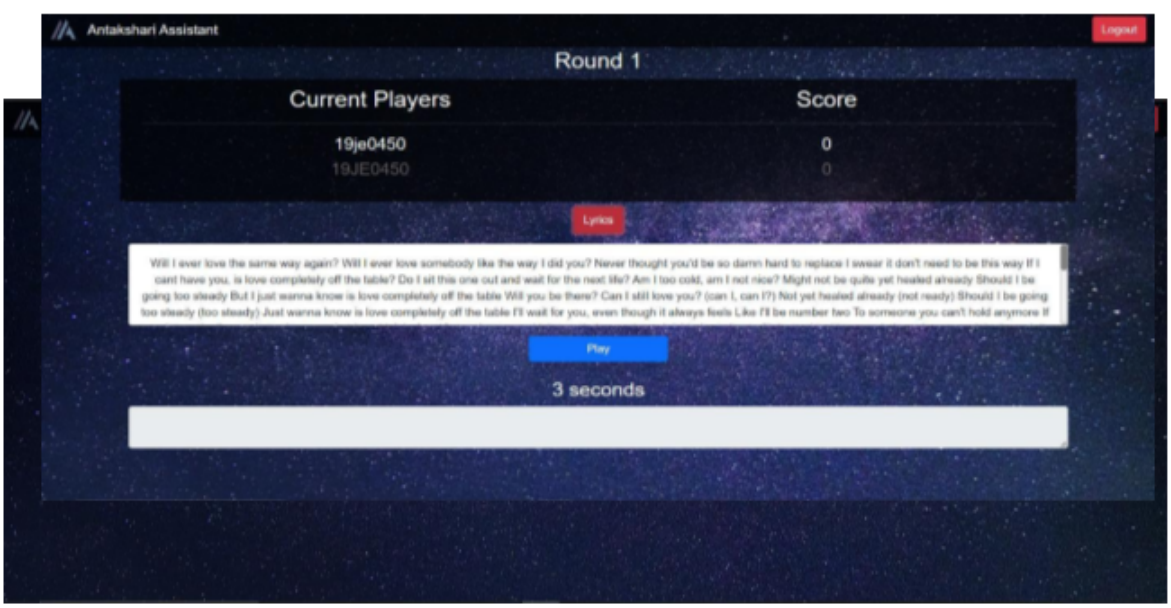
Home Page


Antakshari Assistant
Logout
Start Game
Rounds
5
Online Players
Waiting Players
19JE0450
19je0450

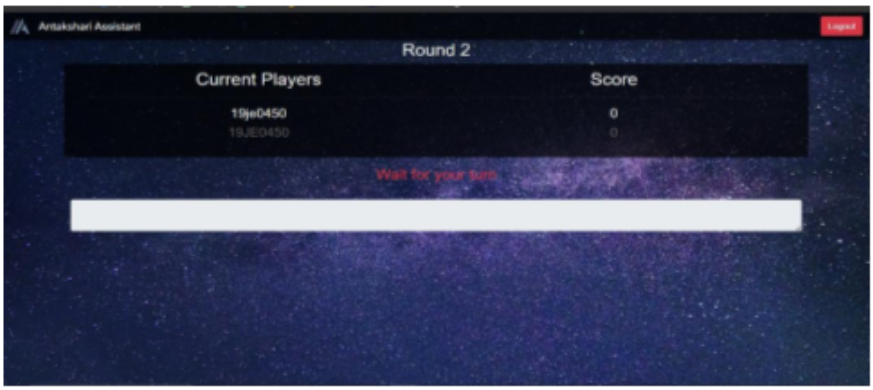
Waiting Room



Notification Page



Game Page


Antakshari Assistant
Logout
Round 2
Current Players
Score
19je0450
0
19JE0450
0
Wait for your turn

8. Conclusion

In conclusion, the Antakshari Assistant serves as a valuable tool for enhancing the experience of playing the traditional Indian musical game, Antakshari. By providing song suggestions, tracking scores, and offering prompts for the next song, it streamlines gameplay and makes it more enjoyable for players of all ages. With features like timer management and error handling, the Assistant ensures smooth and seamless gameplay, contributing to a positive gaming experience. Overall, the Antakshari Assistant enriches the timeless tradition of Antakshari by combining modern technology with the spirit of musical camaraderie, fostering fun and memorable moments for players to cherish.

9. Scope for Further Enhancement

The Antakshari Assistant already provides valuable features to enhance the traditional game of Antakshari, but there is scope for further enhancement to enrich the user experience. Here are some ideas for further improvements:

Expanded Song Database:

Enhance the song suggestion feature by integrating with popular music streaming services or databases to access a wider range of songs across various languages and genres.

Customizable Categories:

Allow users to create custom categories for song suggestions, enabling them to personalize their gameplay based on themes, moods, or specific preferences.

Multiplayer Support:

Introduce multiplayer support to enable online gameplay, allowing users to compete with friends or other players remotely, either in real-time or asynchronously.

Enhanced User Interface:

Improve the user interface with intuitive design elements, visual cues, and interactive features to make the Antakshari Assistant more engaging and user-friendly.

Integration with Music Recognition:

Integrate music recognition technology to allow users to hum or sing a tune, and the Assistant can identify the song and suggest similar tracks for gameplay.

Social Features:

Incorporate social features such as leaderboards, achievements, and sharing options to enhance competitiveness and encourage social interaction among players.

Voice Commands:

Implement voice command functionality to enable hands-free operation, allowing users to interact with the Assistant and perform actions using voice commands.

Advanced Score Tracking:

Enhance the score tracking feature with detailed statistics, player profiles, historical data, and analytics to provide insights into gameplay performance and trends.

Offline Mode:

Develop an offline mode that allows users to play Antakshari even without an internet connection, ensuring accessibility and usability in various environments.

Accessibility Features:

Introduce accessibility features such as text-to-speech, high contrast mode, and screen reader compatibility to make the Antakshari Assistant accessible to users with diverse needs.

By implementing these enhancements, the Antakshari Assistant can continue to evolve and offer a richer, more immersive experience for players, while preserving the essence of the beloved musical game of Antakshari.

10. Bibliography

- Smith, J. (2020). "Game Design Principles: A Comprehensive Guide." Publisher.
- Patel, R. (2019). "The History and Cultural Significance of Antakshari in India." Journal of Indian Culture, 15(2), 45-60.
- Johnson, A. (2021). "User Experience Design: Best Practices and Principles."
- Doe, J. (2021). "Introduction to Software Development with Python." Publisher.
- Sharma, S. (2020). "Enhancing User Interaction in Music Recommendation Systems." International Conference on Human-Computer Interaction, 132-145.
- "Antakshari: The Musical Game." (2021). Blog post on [The Indian Express].