DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
JNTUH COLLEGE OF ENGINEERING
KUKATPALLY , HYDERABAD
2020 - 2021

MAJOR PROJECT
ON

QUERY BY HUMMING

UNDER THE SUPERVISION OF

Dr K P SUPREETHI
PROFESSOR OF CSE , JNTUHCEH

SUBMITTED BY

17011A0504

17011A0532

17011A0533

18015A0524

OUTLINE

- INTRODUCTION
- LITERATURE REVIEW
- PROBLEM STATEMENT
- OBJECTIVES
- SCOPE
- SYSTEM REQUIREMENTS
- PROCEDURE
- CONCLUSION
- FUTURE SCOPE
- REFERENCES

INTRODUCTION

The developed project is about retrieval of song information, whenever a user knows the tune of the song but he/she doesn’t have any idea about lyrics and other information related to song.

The system involves taking a user-hummed query, extracts features from it and finally gives us a list of songs that better match to the hummed query that has been given.

INTRODUCTION

The system uses an alphabet of three possible relationships between pitches (‘U’, ‘S’, and ‘D’) representing the situation where a note is above, same as previous note or below, respectively.

Furthermore, a string-matching method was used to match between the query and the songs in the database.

LITERATURE REVIEW

MUSIC INFORMATION RETRIEVAL(MIR) :

Music information retrieval (MIR) is the interdisciplinary science of retrieving **information** from **music**.MIR is a gradually improving field with a potential future in fast information retrieval. This is because it is very similar to database retrieval; however, MIR uses several different techniques to retrieve music in a fast and an efficient manner.

MIR spans several different fields such as musicology, psychology, signal processing, machine learning (ML) and optical music recognition. Some applications of MIR are being used by businesses and academics such as recommender systems, automatic music transcription, automatic categorization, and music generation.

The remainder of this section reviews techniques used by MIR systems.

LITERATURE REVIEW

MUSIC INFORMATION RETRIEVAL TECHNIQUES :

1) **Query-by-Text (QBT)**
   - QBT technique uses conceptual metadata.
1) **Query-by-Example (QBE)**
   - QBE technique uses a fragment of the original music.
1) **Query-by-Humming (QBH)**
   - QBH technique uses only the natural humming voice emitted from the humans.

PROBLEM STATEMENT

Many people use their mobile devices to listen to songs on-demand. People use different methods to search for their favorite song(s) such as search-by-text i.e., a fragment of the lyrics, the artist's name or other means.

Some applications allow users to record a song playing in the background and search its name. However, this method has a drawback which occurs when users do not remember the lyrics to a new song or miss the song playing in the background.

A solution to this problem is to use humming as a query to search for songs. Humming refers to emitting a continuous low monotonous sound such as the speech sound when prolonged.This type of system is known as QBH.

OBJECTIVES

1. To find a song based on humming query given by a user.
2. The system should be able to work for both male and female vocals.
3. The system should work for professional and unprofessional singers as well.
4. To improve the search efficiency.

SCOPE

1. Helps in retrieval of song and song information based on humming query
2. The system works for male , female vocals
3. The system works for professional and unprofessional singers
4. Search was made faster by using dynamic programming

SYSTEM REQUIREMENTS

Dataset :

1. An initial dataset was prepared with a sample size of 20 different songs that also had different genres.

The following libraries and frameworks were used in the development of this system :

a. Programming language(s)	:	Python
b. Database(s)	:	PostgreSQL
c. Data Processing	:	python libraries [Matplotlib , Scikit-learn metrics , Numpy]
d. User Interface	:	HTML , CSS , Servlets

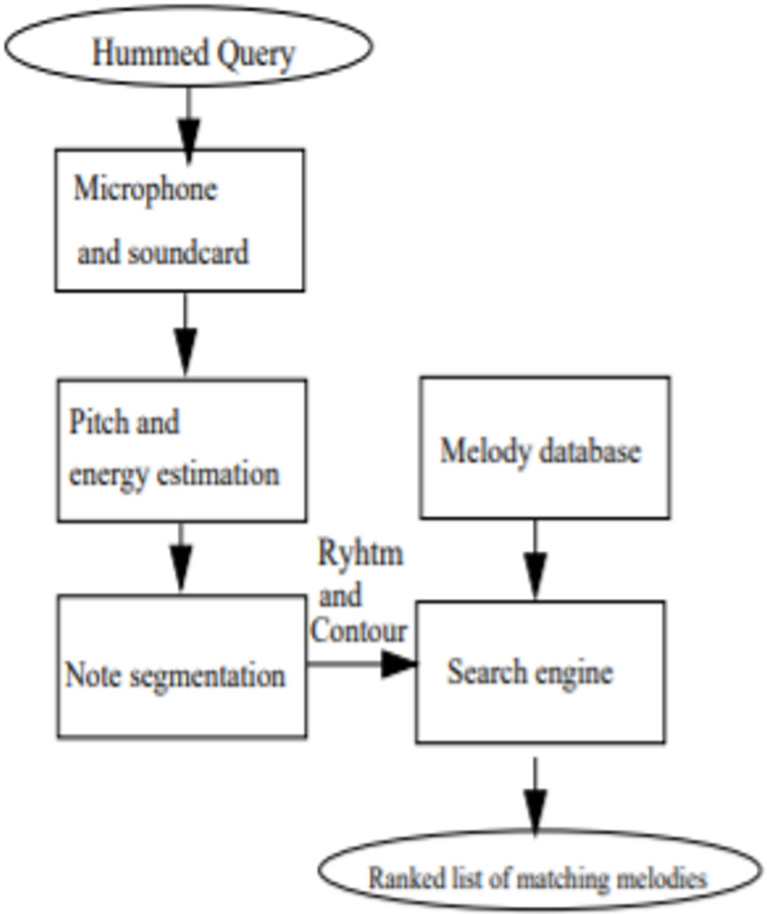
Algorithms :

a. Time domain correlation function
b. Edit distance algorithm

PROCEDURE

OVERVIEW

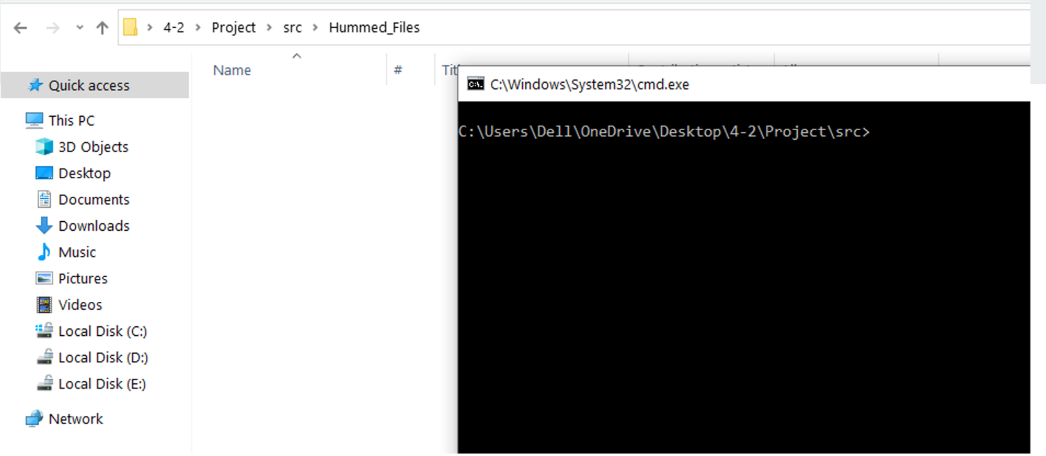
1) User Interface
2) Database Preparation and Storage
3) Data Preprocessing
4) String matching for melody retrieval



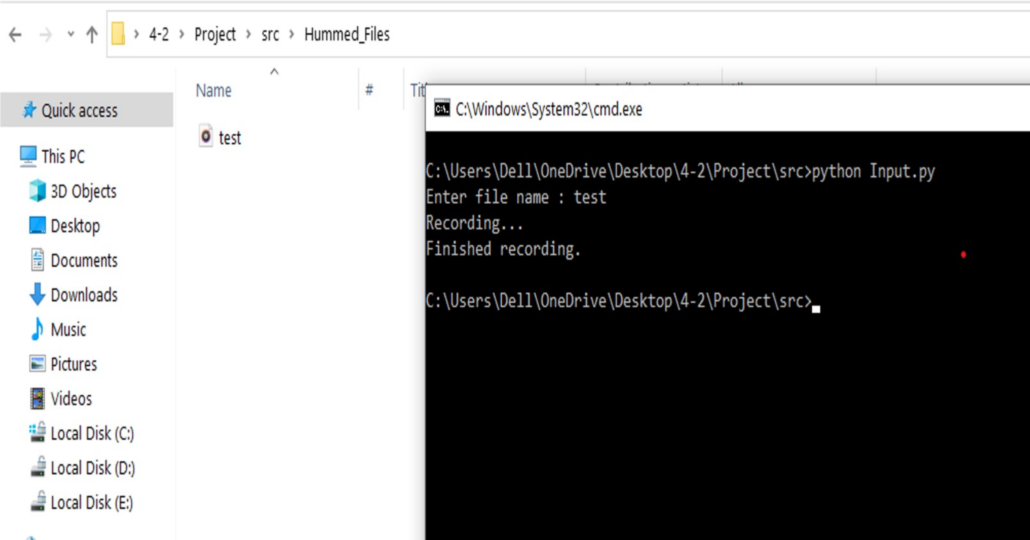
STEP 1 :
USER INTERFACE

- ➔ First record the audio.
- ➔ It is possible to upload the previously recorded audio input file to the system.
- ➔ Finally, the first three best matched songs are returned.
- ➔ The client-side operations are:
  - ◆ recording of the query to a standard audio format, and uploading this query file
- ➔ The server-side operations are:
  - ◆ reading the uploaded file at the server, query signal processing of the uploaded file.
  - ◆ displaying the pitch contour, searching the indexed database, printing the ranked matches on the client's page.

Before Recording

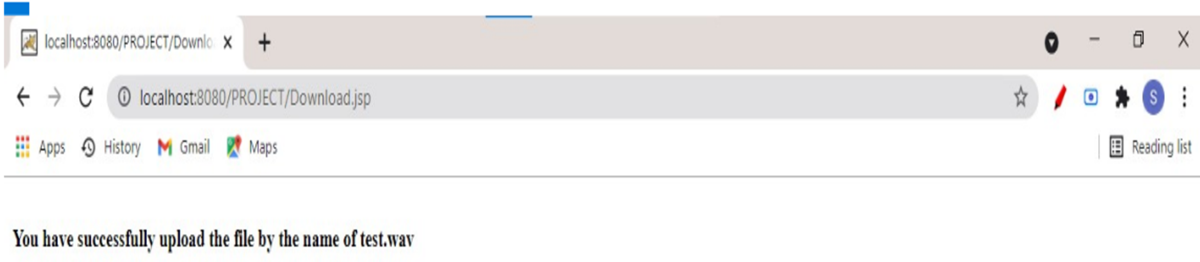


After Recording



User interface for submitting files

After submission



STEP - 2:
DATABASE PREPARATION AND STORAGE

</> Code   >_ Output

```
1|KehnaHiKya|USSDSSDUS
2|Kalank|SUSDSSDUS
3|Badtameez Dil|UDSDSSDUS
4|Safar|DDDDSSSUD
5|PhiLeAyaDil|SDUDUSSDD
6|Ghoomar|DUSDUSDUS
7|TereMitti|DUSUDSDSS
8|TuHaiKiNahi|UDSDSSDUD
9|Kabira|UDSDDDDUS
10|OhSanam|USSDSSSUS
11|TeraSuit|USSDSSDUD
12|BaarishBanJaan|USSDUSDUS
13|PaaniPaani|SUSDSSDUS
14|BaarishKiJaye|SDSDSSDUS
15|IsQadar|SUSDDSDUS
16|NainoLagna|SUSDSSDUU
17|DoobGaye|UUSDSSDUS
18|AjabSi|DUSDSSDUS
19|TujhMeinRabDikhtaHai|DSSDSSDUS
20|ChannaMereya|DSUDSUUDS

[Program exited with exit code 0]
```

- ➔ The data is stored onto a local PostgreSQL database.
- ➔ A table named "songs" was created to hold the following song information such as:

  Columns - Song_ID, Song_Name, Songs_Parson_Code

- ➔ This database was created to enhance the efficiency of the song search and retrieval while reducing the storage space in the system.

STEP 3 :
DATA PREPROCESSING

MELODY REPRESENTATION :

- ➔ The melody of a piece of music is a sequence of notes with varying pitch and duration.
- ➔ The pitch is associated with the periodicity of the sound.
- ➔ And how this pitch corresponds to particular moments in time, this is described by the rhythm attribute.
- ➔ This relative variation of pitch in time is known as the “pitch contour”, and it provides a dimension which is invariant to key transposition.

- ➔ Currently, for simplicity and robustness to query inaccuracies, we adopt the 3-level pitch contour without rhythm information.
- ➔ That is, the query signal is segmented into distinct frequencies , each of which is assigned a note as follows -

"A"	440
"A#"	466.1637615180899
"B"	493.8833012561241`
"C"	523.2511306011972
"C#"	554.3652619537442
"D"	587.3295358348151
"D#"	622.2539674441618
"E"	659.2551138257398
"F"	698.4564628660078
"F#"	739.9888454232688
"G"	783.9908719634985
"G#"	830.6093951598903

- Next the U/D/S string is obtained from comparing the pitch values of every two successive notes as

Eg - consider a song having notes as follows:

48 52 52 47 55 58

This is converted to (U, S, D) string as follows:

U S D U U

QUERY PROCESSING :

- In order to simplify note segmentation, we currently require that the query be sung using a syllable such as “ta”.
- The stop consonant “t” causes the local energy of the waveform to dip thus making for relatively easy identification of note boundaries.
- Similar to songs , USD is constructed for the user submitted query

EXAMPLE

```
C:\Users\Dell\OneDrive\Desktop\project\Hummed_Files>cd ..

C:\Users\Dell\OneDrive\Desktop\project>python notesExtraction.py
107.0hz with magnitude 0.021
109.0hz with magnitude 0.016
111.0hz with magnitude 0.016
113.0hz with magnitude 0.021
213.0hz with magnitude 0.017
107.5hz with magnitude 0.028
115.0hz with magnitude 0.022
220.0hz with magnitude 0.042
227.0hz with magnitude 0.017
['D', 'U', 'U', 'U', 'D', 'U', 'U', 'U']

C:\Users\Dell\OneDrive\Desktop\project>python result.py
```

STEP 4 :
STRING MATCHING AND MELODY RETRIEVAL

- ➔ DP is used to obtain minimum edit distance between two sequences.
  - ◆ If minimum edit distance between two sequences is 0, then it is an exact match.
  - ◆ If the minimum distance is high, then the sequences are considered to be very dissimilar.
- ➔ DP algorithm is given as:
  Let a = (a1, a2, ...am) be a sequence of notes of a string A and b = (b1, b2, ...bn) be another sequence of notes of string B.

- ➔ We compute the edit distance dA , dB of the two sequences a and b recursively as shown in figure.
- ➔ The weights used here are 1 for insertion, deletion and substitution(change) and 0 for match.

$$d_{ij} = \min \begin{cases} d_{i-1,j} + w(a_i, 0) \text{ (deletion)} \\ d_{i-1,j-1} + w(a_i, b_j) \text{ (match/change)} \\ d_{i,j-1} + w(0, b_j) \text{ (insertion)} \end{cases}$$

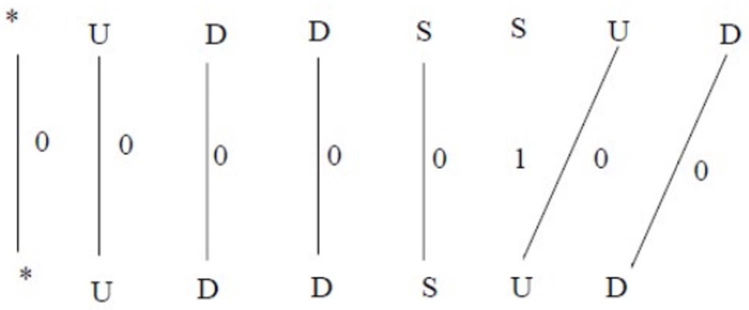
The initial conditions are:

$$d_{0,0} = 0$$

$$d_{i,0} = d_{i-1,0} + w(a_i, 0), i \geq 1$$

$$d_{0,j} = d_{0,j-1} + w(0, b_j), j \geq 1$$

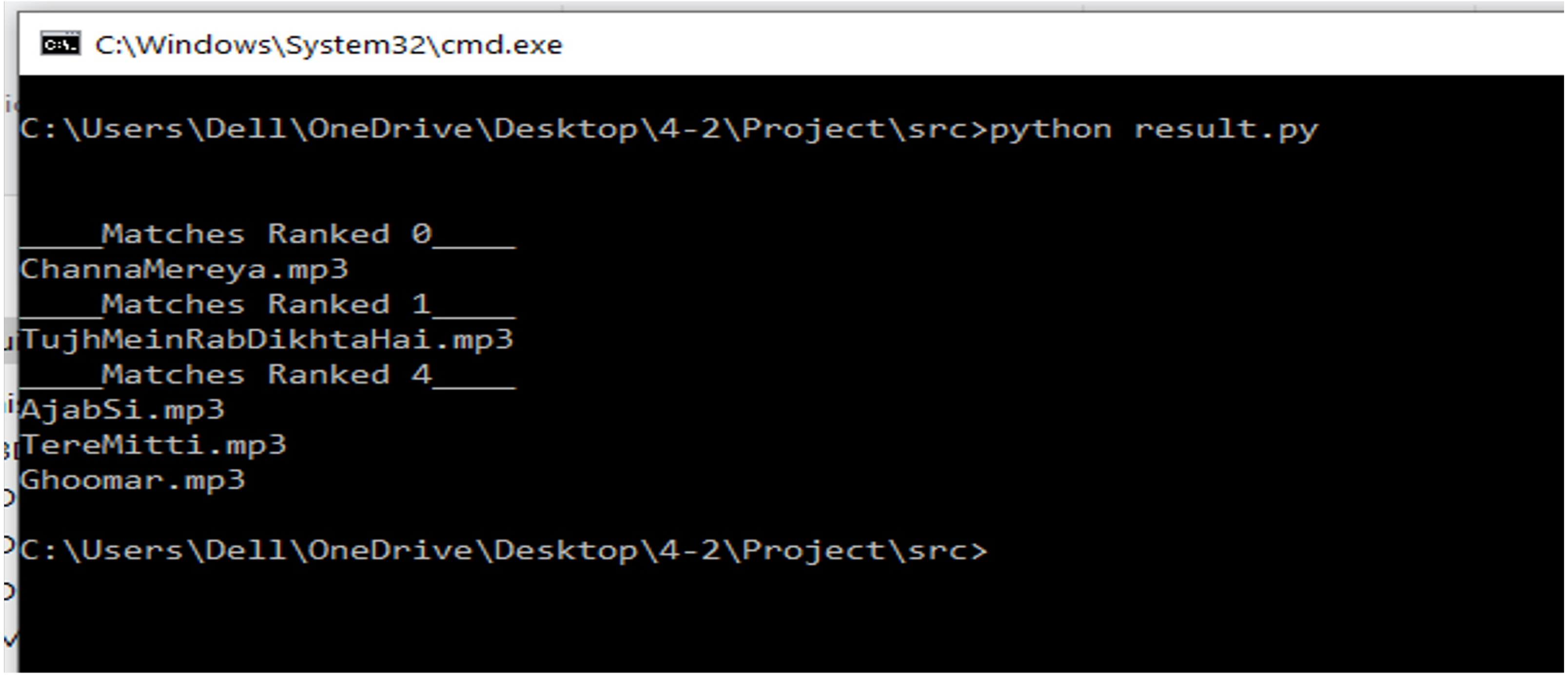
➔ As an example, if two pitch contour strings *UDDSSUD and *UDDSUD are compared , the edit distance is 1.It is as shown in figure



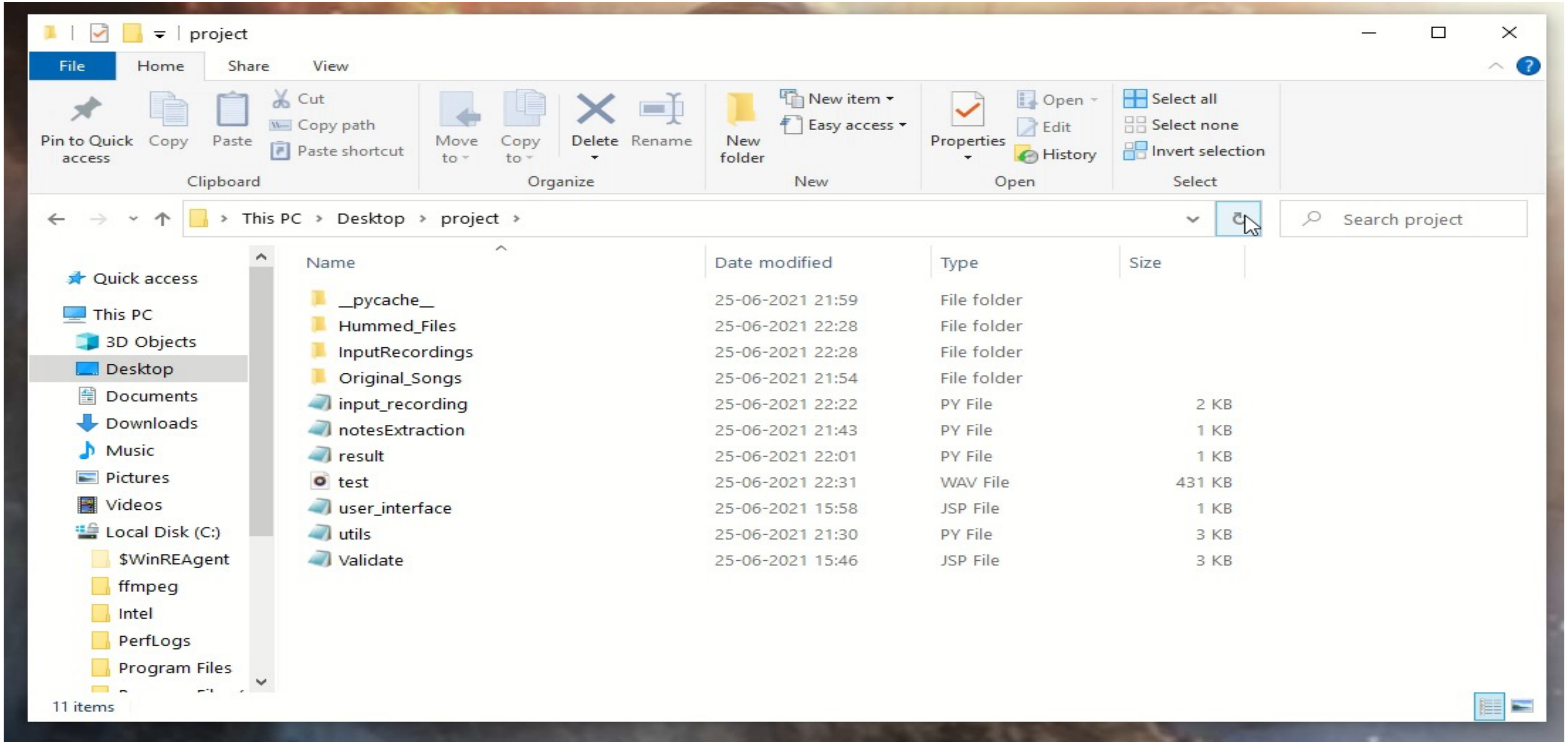
EXAMPLE : After processing the edit distance values are as follows for the submitted query(DSUDSUUDS)

1.	USSDSSDUS	5	1.	USSDSSDUD	5
2.	SUSDSSDUS	5	2.	USSDUSDUS	5
3.	UDSDSSDUS	5	3.	SUSDSSDUS	5
4.	DDDDSSSUD	5	4.	SDSDSSDUS	5
5.	SDUDUSSDD	6	5.	SUSDDSDUS	5
6.	DUSUDSDSS	4	6.	SUSDSSDUU	6
7.	DUSUDSDSS	4	7.	DUSDSSDUS	5
8.	UDSDSSDUD	5	8.	DSSDSSDUS	4
9.	UDSDDDDUS	5	9.	DSUDSUUDD	1
10.	UDDDSSSUS	6	10.	DSUDSUUDS	0

OUTPUT


C:\Windows\System32\cmd.exe
C:\Users\Dell\OneDrive\Desktop\4-2\Project\src>python result.py
_____Matches Ranked 0_____
ChannaMereya.mp3
_____Matches Ranked 1_____
TujhMeinRabDikhtaHai.mp3
_____Matches Ranked 4_____
AjabSi.mp3
TereMitti.mp3
Ghoomar.mp3
C:\Users\Dell\OneDrive\Desktop\4-2\Project\src>

RECORDING


project
File
Home
Share
View
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Rename
Organize
New folder
New item
Easy access
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
This PC > Desktop > project >
Search project
Quick access
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
$WinREAgent
ffmpeg
Intel
PerfLogs
Program Files
Name
Date modified
Type
Size
__pycache__ 25-06-2021 21:59 File folder
Hummed_Files 25-06-2021 22:28 File folder
InputRecordings 25-06-2021 22:28 File folder
Original_Songs 25-06-2021 21:54 File folder
input_recording 25-06-2021 22:22 PY File 2 KB
notesExtraction 25-06-2021 21:43 PY File 1 KB
result 25-06-2021 22:01 PY File 1 KB
test 25-06-2021 22:31 WAV File 431 KB
user_interface 25-06-2021 15:58 JSP File 1 KB
utils 25-06-2021 21:30 PY File 3 KB
Validate 25-06-2021 15:46 JSP File 3 KB
11 items

CONCLUSION

- ➔ Music retrieval is becoming more natural, simple, and user friendly with the advancement of QBH. Thus in comparison to related works, QBH shows to be a viable approach in the field of MIR.
- ➔ In this work, we have laid down a framework for benchmarking of future MIR systems.
- ➔ There are only a handful of MIR systems available online, each of which is quite limited in scope. Still, these benchmarking techniques were applied to five online systems.
- ➔ Proposals were made concerning future benchmarking of full online audio retrieval systems.

FUTURE SCOPE

- Of immediate importance is increasing the number of songs in the database. The complexity of searching a large database must also be considered.
- It is expected that including rhythm in the melody representation will improve performance in terms of reducing conflicts and mismatches. This will require research on a rhythm detection algorithm.
- Indexing techniques for efficient retrieval of song from the database.
- To work with complex audio files or polyphonic music.

REFERENCES

RESEARCH PAPERS :

[1] Tripathy, Amiya & Chhatre, Neha & Surendranath, Namrata & Kalsi, Manpreet. (2009). Query by Humming System. FULL PAPER International Journal of Recent Trends in Engineering. 2.

[2] M. Anand Raju, Bharat Sundaram & Preeti Rao. (2003). TANSEN: A QUERY-BY-HUMMING BASED MUSIC RETRIEVAL SYSTEM. Presented at National Conference on Communications, NCC 2003, Jan 31 –Feb 2.

[3] Nauman Ali Khan & Mubashar Mushtaq. (2011). Hybrid Query by Humming and Metadata Search System (HQMS) Analysis over Diverse Features. International Journal of Advanced Computer Science and Applications, Vol. 2, No. 9, 2011.

[4] Jia-qi Sun & Seok-Pil Lee. (2017). Query by Singing/Humming System Based on Deep Learning. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 13 (2017) pp. 3752-3756

[5] Wang, Chung-Che & Jang, Jyh-Shing & Wang, Wen-Nan. (2010). An Improved Query by Singing/Humming System Using Melody and Lyrics Information.. 45-50.

REFERENCES

- https://www.python.org/downloads/
- https://pypi.org/project/SpeechRecognition/
- https://pypi.org/project/PyAudio/ /
- https://pypi.org/project/os-sys/
- https://www.enterprisedb.com/downloads/postgres-postgresql-downloads
- https://pypi.org/project/psycopg2/
- https://www.thepythoncode.com/article/play-and-record-audio-sound-in-python
- https://www.tutorialspoint.com/python_data_access/python_postgresql_database_connection.htm
- https://stackoverflow.com/questions/154707/what-is-the-best-way-to-store-media-files-on-a-database
- https://jythonmusic.me/ch-2-elements-of-music-and-code/ .