

Speech-Based Search Engine Project

1. Introduction:

The traditional method of searching involves typing queries into a search engine. However, with the rise of voice technology, there's an increasing demand for systems that allow users to interact via speech. This project introduces a speech-based search engine that enables users to search the web using voice commands.

2. Objective:

To develop a system that:

- Takes voice input from the user.
- Converts it into a text-based query.
- Searches the internet using the query.
- Displays or reads out the search results.

3. Technologies Used:

- Programming Language: Python
- Speech Recognition: `speech_recognition`, Google API
- Search Engine Access: `googlesearch-python`
- Text-to-Speech (TTS): `pyttsx3` or `gTTS`
- GUI (optional): Tkinter or Flask

4. System Architecture:

Voice Input -> Speech to Text -> Query Processing -> Web Search API -> Display Results

5. How It Works:

1. User speaks a query through the microphone.
2. Speech is captured and converted to text.
3. The text is passed to a search engine API (like Google).
4. Search results are fetched and displayed (or read aloud).

6. Sample Code:

```
```python

import speech_recognition as sr

from googlesearch import search

import pyttsx3

def speak(text):

 engine = pyttsx3.init()

 engine.say(text)

 engine.runAndWait()

def recognize_speech():

 r = sr.Recognizer()

 with sr.Microphone() as source:

 print("Please speak your search query:")

 audio = r.listen(source)

 try:

 query = r.recognize_google(audio)

 print("You said:", query)

 return query

 except:
```

```

 print("Sorry, could not recognize your voice.")

 return ""

def search_online(query):

 print("\nTop 5 Search Results:")

 for i, result in enumerate(search(query, num_results=5), start=1):

 print(f"{i}. {result}")

 speak("Here are the top search results for your query.")

if __name__ == "__main__":

 query = recognize_speech()

 if query:

 search_online(query)

'''

```

## 7. Applications:

- Assistive technology for visually impaired users.
- Hands-free search in smart devices.
- Voice-controlled educational tools.

## 8. Future Enhancements:

- Multilingual voice search support.
- Personalized search using AI.
- Offline search capabilities with local databases.
- Mobile application version (Android/iOS).

## 9. Conclusion:

The speech-based search engine bridges the gap between humans and machines by enabling natural voice-based interaction. It promotes accessibility and efficiency in retrieving information from the internet.