BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)
Bull Temple Road, Basavanagudi, Bangalore – 560019

An AAT report on

"SPEECH RECOGNITION"

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

BACHELOR OF ENGINEERING IN
INFORMATION SCIENCE AND ENGINEERING

By

Varun N, 1BM19IS176
Thousif Pasha, 1BM19IS169
Vishal L Prasad, 1BM19IS180
Hiral Honnoor Ali, 1BM20IS406

Under the guidance of

Shobana T S, Assistant

Professor

Department of Information Science and Engineering

2021-22

BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)
Bull Temple Road, Basavanagudi, Bangalore – 560019

Department of Information Science and Engineering

CERTIFICATE

This is to certify that the project entitled "**SPEECH RECOGNITION**" is a bonafide work carried out by **Varun N (1BM19IS170), Thousif Pasha (1BM19IS169), Vishal L Prasad (1BM19IS180) & Hiral Honnoor Ali (1BM20IS406)** in partial fulfillment for the award of degree of Bachelor of Engineering in **Information Science and Engineering** from **Visvesvaraya Technological University, Belagavi** during the year **2021-2022**. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering Degree.

Signature of the Faculty
Name and Designation

Signature of the HOD
Name and Designation

Table of Contents

ABSTRACT

The report represents project work of developing an Interactive Voice Response (IVR) system with the capability of Speech Recognition. It is a system where the user can interact with his/her voice and on the basis of the user's voice input the system performs particular action as a response to the voice command. It is an approach to make use of the voice of the user to recognize the given command instead of using keyboard or other form of input. The project is helpful for various large organizations where there are many call queries or for telecommunications companies or other business ventures where user queries through their voice online or via cellular connections. The system provides an application environment whereby users are prompted to input specific voice commands and the given voice input is being processed for further task accomplishment as per the request.

Making use of speech as a input signal significantly increases user experience. Unlike traditional touch-tone or key-press based IVR system the system being developed is capable to take a real time voice input from a user. Speech recognition applications are becoming more and more useful nowadays. With growth in the needs for embedded computing and the demand for emerging embedded platforms, it is required that the speech recognition systems (SRS) are available on them too. This project is a simple approach to developing a system where the speech recognition is embedded within another system to automate task using speech as input command.

INTRODUCTION

1.1 Background

Speech probably is the most efficient and natural way to communicate with each other. Thus, being the best way of communication, it could also be a useful interface to communicate with machines and systems like the IVR system. The Interactive Voice Response (IVR) system along with the speech recognition technology can play an efficient role in providing easy and efficient customer/user service. If properly implemented it can increase the user satisfaction and offer new services. Speech Recognition has now begun to dominate the market technology and is pushing away the traditional way of using hectic interfaces such as keyboards and mouse as input source to computer systems. Voice command based applications will make life easier due to the fact that people will get easy and fast access to information. Therefore the popularity of automatic speech recognition systems has greatly increased.

The work of speech recognition further helps in establishing an easy way of communication between interactive response systems and users/ customers i.e. as a part of post processing of the speech recognizing process we can accomplish some computational task with such a system making voice input as a trigger to do some task within the system. The IVR with AVR (Automatic Voice Recognition) allows callers to obtain information and perform transactions simply by speaking naturally. Recognition of free-form conversation is not yet a reality, and speech recognition has sometimes been over-hyped in the past. However, speech recognition technology is now proving itself commercially viable in a number of customer service applications.Early speaker-dependent dictation systems had to be trained to understand the speech of one specific user.

Now, speaker-independent recognition technologies allow IVR systems to interpret the speech of many users. Today’s speech enabled IVR applications serve large numbers of unknown callers without prior training of the system. Many factors are driving the emergence of speech as the IVR user interface of choice for today and tomorrow. The first is reducing labor costs. The cost of employing live customer service agents is rising at the same time that organizations are facing increased pressure to reduce the cost of serving customers. When an automated call-processing solution must be employed, a speech-enabled IVR application increases caller acceptance because it provides the friendliest and fastest self-service alternative to speaking with a customer service agent.

1.2 Problem Statement

Most of the work done till today in the field of IVR systems has been primarily focused on the input mechanisms based on the keyboard or touch pad. In such cases it is tedious to provide the input command every time through typing of texts. This way of providing input to the computer system may be enhanced if we could provide direct speech input instead of typing. This enables fast interaction between the system and user and therefore increases overall satisfaction of the customers. This also increases the speed of access of the information from the system. Furthermore, the English language has been widely implemented in IVR systems. This has created difficulty for people while interacting with the system. Thus by implementing the Nepali voice commands it is easier to interact and provide the input to the system.

The major focus of the project being developed is the use of direct Nepali voice command for the interactive voice response system without need of typing which then further can be applicable to real world applications like call centers, customer support systems and other several organization inquiry systems.

1.3 Objectives

Project objectives:

The prime objective of the project being proposed is to design and build a system that a basic user can interact so that she/he can make use of voice commands to deal with system i.e. making a system that has capability of recognizing the isolated speaker words and process the request to forward the given task.

The typical objectives are listed below:

- To make use of domain specific models and algorithms in field.
- To develop an interactive voice response system along with speech.
- To understand the basics of speech processing.
- To get knowledge on various speech recognition approaches.
- To get insights on speech responsive application development.

Academic objectives:

Academically, the project is primarily focused on fulfilling the discipline of an engineering student as a computer engineer working on a project and gain experience as a team throughout the different phases of a project.

Some Typical academic objectives of the project are :

- To design and complete a functional project that integrates various course.
- To develop various skills related to project management like team work.
- To get hands-on experience of working in a project as a team work.
- To learn about and become familiar with the professional engineering practices.

1.4 Scope

The Interactive Voice Response (IVR) system is becoming more popular day by day. Such systems make it more convenient in interaction with the user and computer system and hence help in easy accomplishment of several tasks. The IVR system serves as a bridge between people and computer systems.

The IVR system with Automatic Voice Recognition (AVR) can make it more convenient for the users if they can command the system through their voice and this makes such system applicable in vast areas.The current system is being built for the desktop computers and won't be implemented in actual phone devices due limitation of time and research. However in future the project work can be further enhanced and has huge scope and potential for future implementations in several areas.

Some of the applicable areas are discussed as follows:

1.Organizational inquiry desk: The system can be used in several organizations for easy information access regarding the organization using the voice command.

2.Automatic speech to text conversion: The proposed system deals only with the isolated words detection but with further enhancement in algorithms it can be used in speech to text conversion application.

3.Speech controlled applications: we can make use of speech recognition of Nepali words in performing the task of new developed applications and hence make more user friendly applications.

4.Speech control in embedded systems: using speech recognition technology several tasks can be controlled using voice commands in embedded systems this further increases the automation of works and hence can be very beneficial in industrial process automation.

5.Application for Disabled people: Disabled people are another part of the population that can use speech recognition programs. It is especially useful for people who can't use their hands, from the profoundly physically impaired to people with mild repetitive stress injuries i.e. persons who might require helpers to control their environments.

SYSTEM REQUIREMENT

Hardware Requirement

- i3 Processor Based Computer or higher
- Memory: 1 GB RAM
- Hard Drive: 50 GB
- Monitor

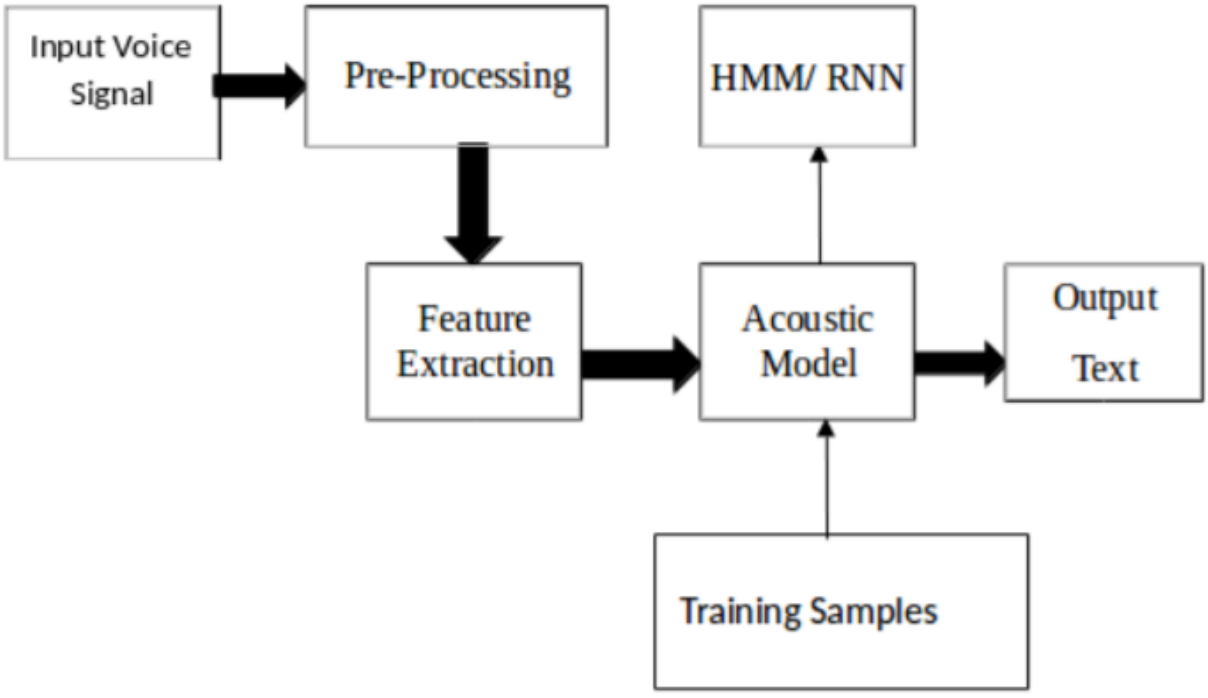
Software Requirement

- Windows 7 or higher and Python

System Design and Flow Diagram

Speech Recognition System:

The idea behind speech recognition is to provide a means to transcribe spoken words into written text. There exist many approaches to achieve this goal. The most simple technique is to build a model for every word that needs to be recognized.Speech signal primarily conveys the words or message being spoken. Area of speech recognition is concerned with determining the underlying meaning in the utterance. Success in speech recognition depends on extracting and modeling the speech dependent characteristics which can effectively distinguish one word from another. The system is a collection of several modules as shown in Figure 1.



.

Figure 1: Speech Recognition System

METHODOLOGY

Methodology overview

Before beginning the project work we went through several researches. Through the research we learnt about several useful methodologies so that we can approach the project development phase. Going through several papers and books we selected a suitable approach that fulfilled our needs as per our domain system. the overall outline of the project development methodology. The preliminary stage began with the collection of sample voice signals from different speakers using a recording software tool called audacity. The recorded voice signal consists of external noises which were depreciated using the noise removal technique. After that from each recorded signal training data samples were generated using a split module which makes the data samples suitable for feature extraction. Using those samples, a specific feature extraction technique is used to extract unique feature vectors. In our case we used MFCC for feature extraction purposes.

The extracted features were used to train the prediction model. In our project we have used two approaches for prediction purposes which are HMM and RNN approaches respectively. The details regarding the methodology are discussed in the sections below.

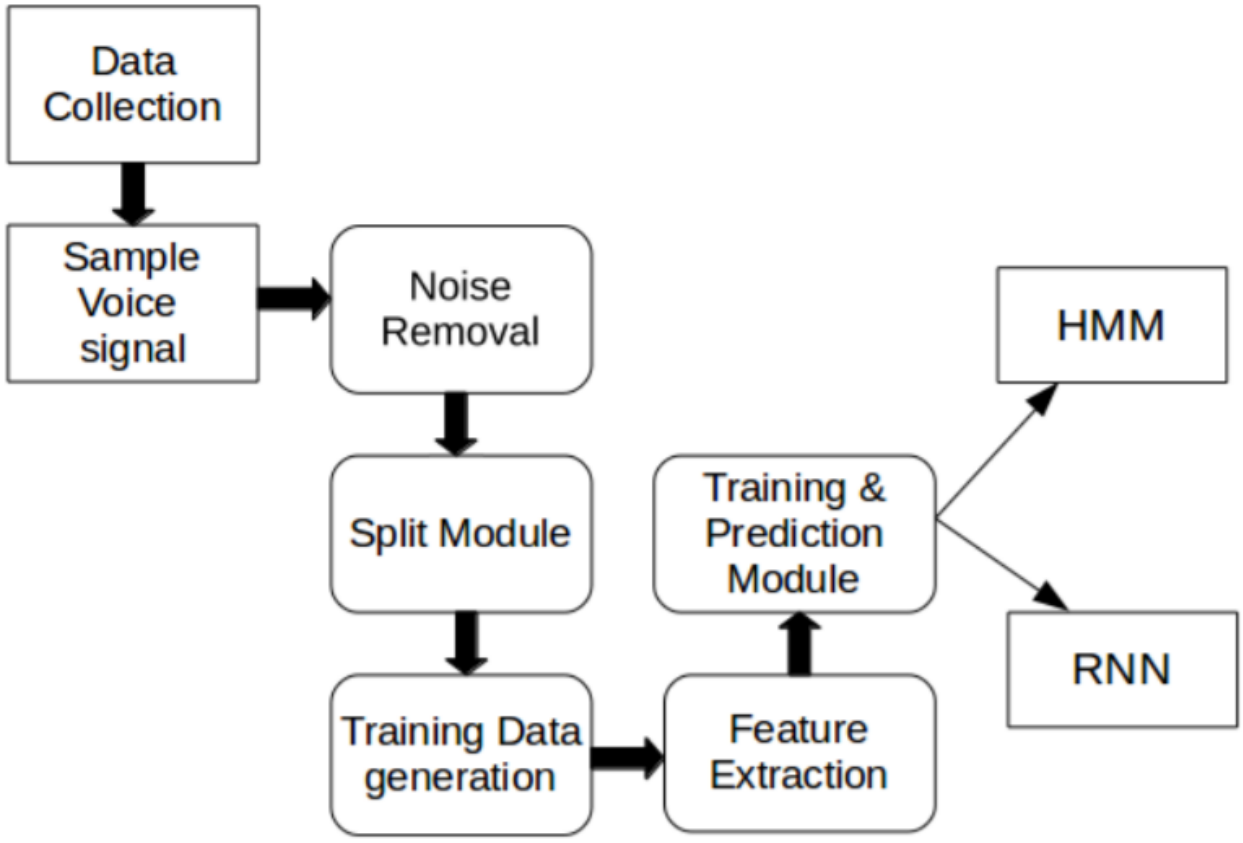


Figure 2: Outline of the project methodology

System Design

Use Case Diagram

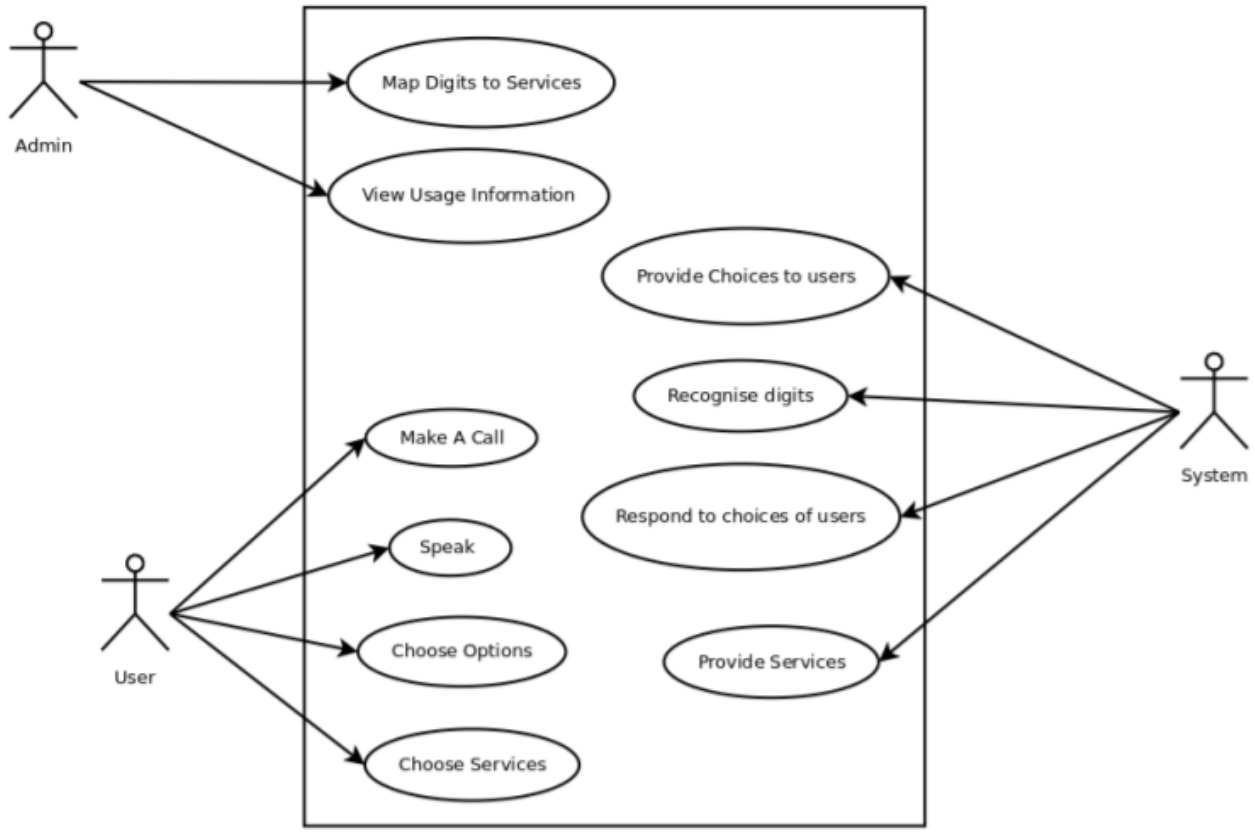


Figure 3: Use Case Diagram

Sequence Diagram

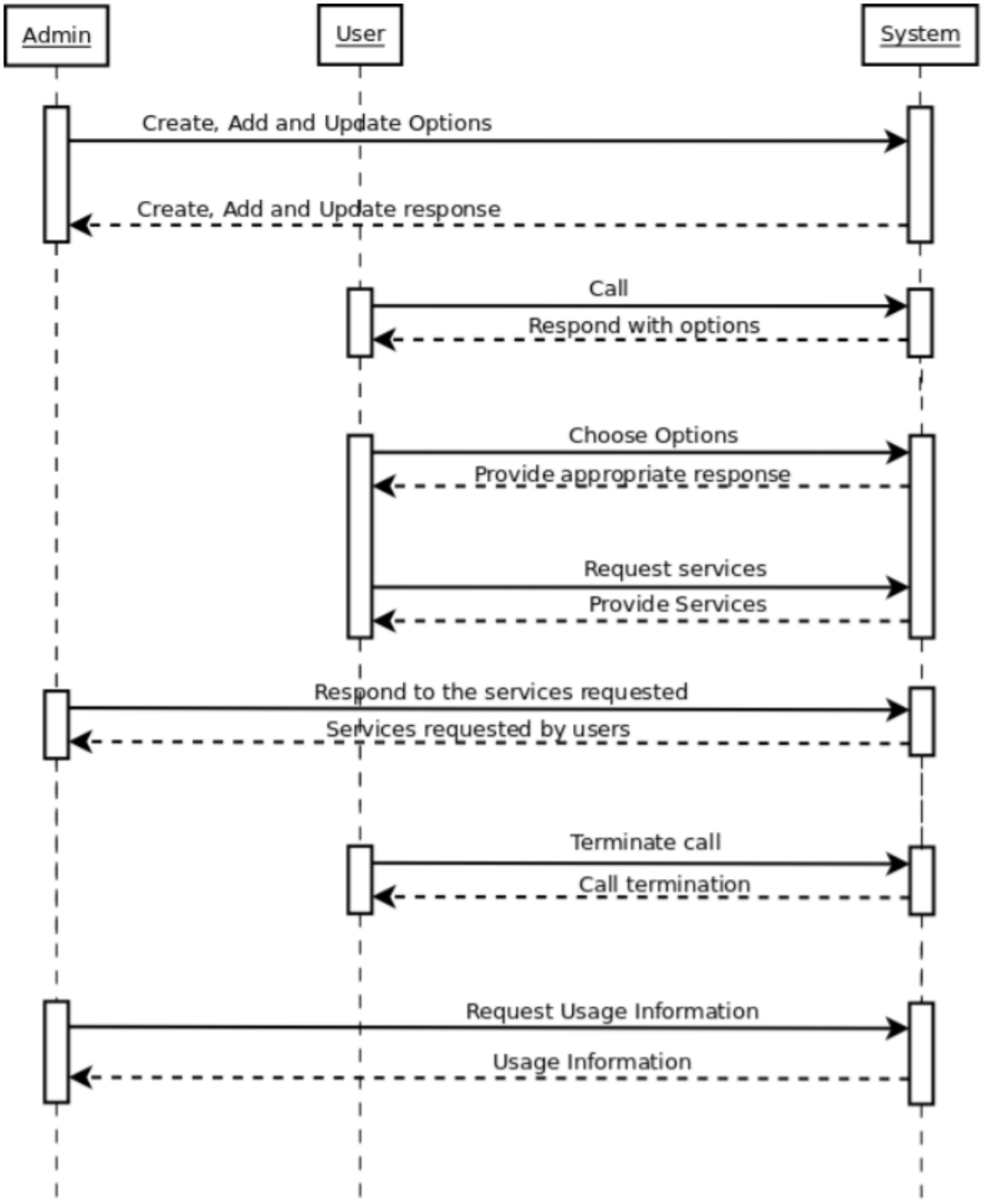


Figure 4: Sequence Diagram

Data Flow Diagram

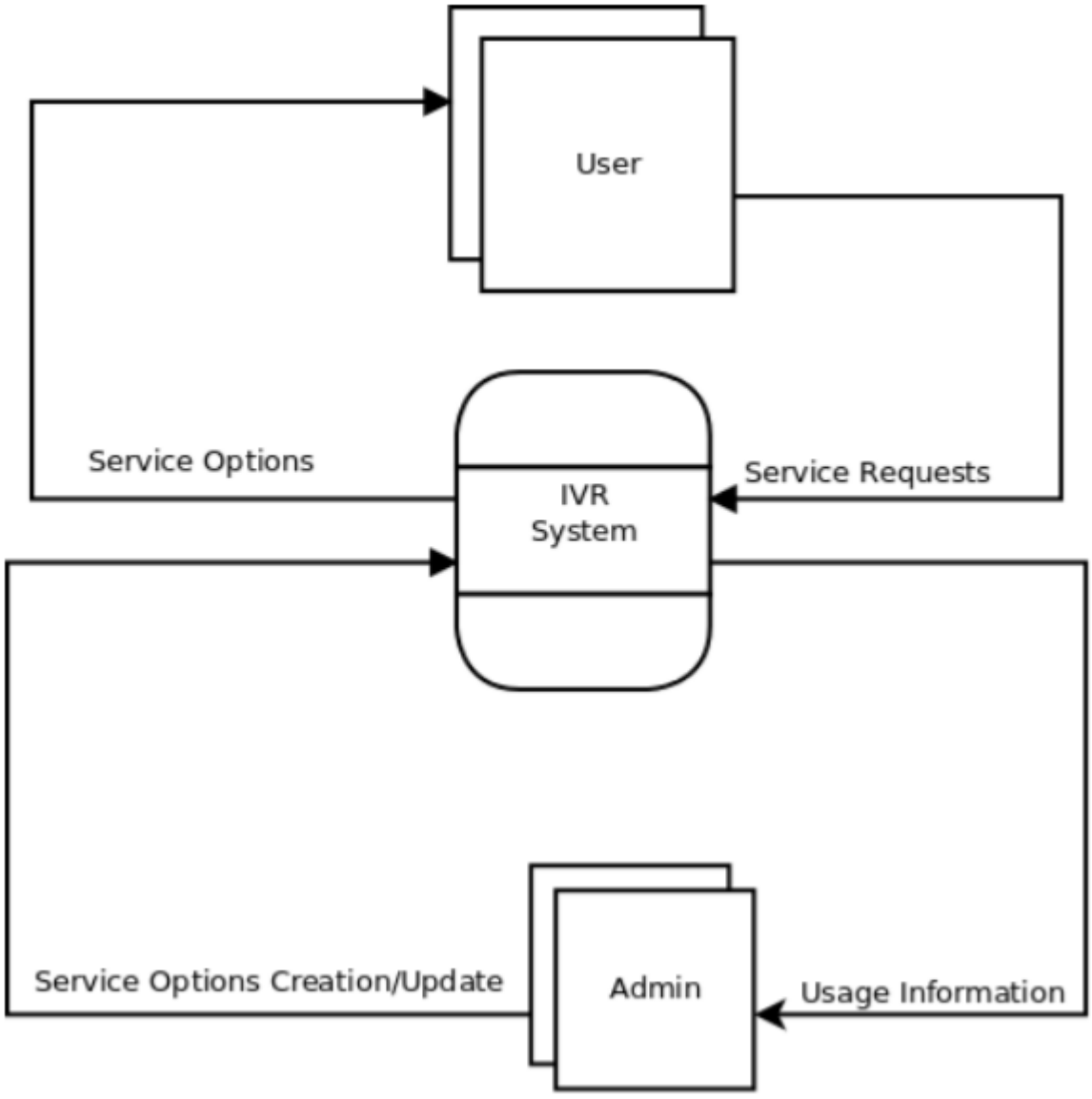


Figure 5: Data Flow Diagram

IMPLEMENTATION

```
import pyttsx3 as p
import speech_recognition as sr
from selenium import webdriver
import randfacts
#from YT_auto import *
import requests
import pyjokes
import datetime
import time



engine =p.init()
rate=engine.getProperty('rate')
engine.setProperty('rate',130)
voices=engine.getProperty('voices')
engine.setProperty('voices',voices[0].id)



#print(voices)
#print(rate)
#engine.say("hell Rehan  my name is Ali")
#engine.say("Hello there i am your Siri")
#engine.runAndWait()

def speak(text):
    engine.say(text)
    engine.runAndWait()
```

```
def wishMe():
    hour = int(datetime.datetime.now().hour)
    if hour>= 0 and hour<12:
        speak("Good Morning Sir !")

    elif hour>= 12 and hour<18:
        speak("Good Afternoon Sir !")

    else:
        speak("Good Evening Sir !")



#url="https://official-joke-api.appspot.com/random_joke"
#json_data=requests.get(url).json()

#arr=["",""]
#arr[0]=json_data["setup"]
#arr[1]=json_data["punchline"]

#def joke():
#    return arr

# class camera():
#     def detect_face():
#
cascPath=os.path.dirname(cv2.__file__)+"/data/haarcascade_frontalface_
default.xml"
#     faceCascade = cv2.CascadeClassifier(cascPath)
#     video_capture = cv2.VideoCapture(0)

#     while True:
#         # Capture frame-by-frame
#         ret, frames = video_capture.read()
```

```
#         gray = cv2.cvtColor(frames, cv2.COLOR_BGR2GRAY)

#         faces = faceCascade.detectMultiScale(
#                 gray,
#                 scaleFactor=1.1,
#                 minNeighbors=5,
#                 minSize=(30, 30),
#                 flags=cv2.CASCADE_SCALE_IMAGE
#                 )

#         # Draw a rectangle around the faces
#         for (x, y, w, h) in faces:
#               cv2.rectangle(frames, (x, y), (x+w, y+h), (0, 255, 0),
2)

#         # Display the resulting frame
#         cv2.imshow('Video', frames)
#         speak("detecting face")
#         print("Detecting face.....")
#         time.sleep(10)
#         pyautogui.press('q')
#         if cv2.waitKey(1) & 0xFF == ord('q'):
#             break
#     video_capture.release()
#     cv2.destroyAllWindows()

def calculator():
        global query

        if 'add' in query or 'edi' in query:
            speak('Enter a number')
            a = float(input("Enter a number:"))
```

```
            speak('Enter another number to add')
            b = float(input("Enter another number to add:"))
            c = a+b
            print(f"{a} + {b} = {c}")
             speak(f'The addition of {a} and {b} is {c}. Your answer is
{c}')

            speak('Do you want to do another calculation?')
            query = takeCommand().lower()
            if 'y' in query:
                speak('ok which calculation you want to do?')
                query = takeCommand().lower()
                calculator()
            else:
                speak('ok')

        elif 'sub' in query:
            speak('Enter a number')
            a = float(input("Enter a number:"))
            speak('Enter another number to subtract')
            b = float(input("Enter another number to subtract:"))
            c = a-b
            print(f"{a} - {b} = {c}")
             speak(f'The subtraction of {a} and {b} is {c}. Your answer
is {c}')

            speak('Do you want to do another calculation?')
            query = takeCommand().lower()
            if 'y' in query:
                speak('ok which calculation you want to do?')
                query = takeCommand().lower()
                calculator()
            else:
```

```
                speak('ok')


        elif 'mod' in query:
            speak('Enter a number')
            a = float(input("Enter a number:"))
            speak('Enter another number')
            b = float(input("Enter another number:"))
            c = a%b
            print(f"{a} % {b} = {c}")
               speak(f'The modular division of {a} and {b} is equal to
{c}. Your answer is {c}')


            speak('Do you want to do another calculation?')
            query = takeCommand().lower()
            if 'y' in query:
                speak('ok which calculation you want to do?')
                query = takeCommand().lower()
                calculator()
            else:
                speak('ok')


        elif 'div' in query:
            speak('Enter a number as dividend')
            a = float(input("Enter a number:"))
            speak('Enter another number as divisor')
            b = float(input("Enter another number as divisor:"))
            c = a/b
            print(f"{a} / {b} = {c}")
             speak(f'{a} divided by {b} is equal to {c}. Your answer is
{c}')


            speak('Do you want to do another calculation?')
            query = takeCommand().lower()
```

```
            if 'y' in query:
                speak('ok which calculation you want to do?')
                query = takeCommand().lower()
                calculator()
            else:
                speak('ok')
        elif 'multi' in query:
            speak('Enter a number')
            a = float(input("Enter a number:"))
            speak('Enter another number to multiply')
            b = float(input("Enter another number to multiply:"))
            c = a*b
            print(f"{a} x {b} = {c}")
               speak(f'The multiplication of {a} and {b} is {c}. Your
answer is {c}')

            speak('Do you want to do another calculation?')
            query = takeCommand().lower()
            if 'y' in query:
                speak('ok which calculation you want to do?')
                query = takeCommand().lower()
                calculator()
            else:
                speak('ok')
        elif 'square root' in query:
            speak('Enter a number to find its sqare root')
            a = float(input("Enter a number:"))
            c = a**(1/2)
            print(f"Square root of {a} = {c}")
            speak(f'Square root of {a} is {c}. Your answer is {c}')

            speak('Do you want to do another calculation?')
            query = takeCommand().lower()
```

```
            if 'y' in query:
                speak('ok which calculation you want to do?')
                query = takeCommand().lower()
                calculator()
            else:
                speak('ok')
        elif 'square' in query:
            speak('Enter a number to find its sqare')
            a = float(input("Enter a number:"))
            c = a**2
            print(f"{a} x {a} = {c}")
            speak(f'Square of {a} is {c}. Your answer is {c}')

            speak('Do you want to do another calculation?')
            query = takeCommand().lower()
            if 'y' in query:
                speak('ok which calculation you want to do?')
                query = takeCommand().lower()
                calculator()
            else:
                speak('ok')
        elif 'cube root' in query:
            speak('Enter a number to find its cube root')
            a = float(input("Enter a number:"))
            c = a**(1/3)
            print(f"Cube root of {a} = {c}")
            speak(f'Cube root of {a} is {c}. Your answer is {c}')

            speak('Do you want to do another calculation?')
            query = takeCommand().lower()
            if 'y' in query:
                speak('ok which calculation you want to do?')
                query = takeCommand().lower()
```

```
                calculator()
            else:
                speak('ok')
        elif 'cube' in query:
            speak('Enter a number to find its sqare')
            a = float(input("Enter a number:"))
            c = a**3
            print(f"{a} x {a} x {a} = {c}")
            speak(f'Cube of {a} is {c}. Your answer is {c}')

            speak('Do you want to do another calculation?')
            query = takeCommand().lower()
            if 'y' in query:
                speak('ok which calculation you want to do?')
                query = takeCommand().lower()
                calculator()
            else:
                speak('ok')

        elif 'fact' in query:
            try:
                  n = int(input('Enter the number whose factorial you
want to find:'))
                fact = 1
                for i in range(1,n+1):
                    fact = fact*i
                print(f"{n}! = {fact}")
                  speak(f'{n} factorial is equal to {fact}. Your answer
is {fact}.')
                speak('Do you want to do another calculation?')
                query = takeCommand().lower()
                if 'y' in query:
                    speak('ok which calculation you want to do?')
```

```
                query = takeCommand().lower()
                calculator()
            else:
                speak('ok')
        except Exception as e:
            #print(e)
            speak('I unable to calculate its factorial.')
            speak('Do you want to do another calculation?')
            query = takeCommand().lower()
            if 'y' in query:
                speak('ok which calculation you want to do?')
                query = takeCommand().lower()
                calculator()
            else:
                speak('ok')

    elif 'power' in query or 'raise' in query:
        speak('Enter a number whose power you want to raised')
          a = float(input("Enter a number whose power to be raised
:"))
        speak(f'Enter a raised power to {a}')
        b = float(input(f"Enter a raised power to {a}:"))
        c = a**b
        print(f"{a} ^ {b} = {c}")
          speak(f'{a} raise to the power {b} = {c}. Your answer is
{c}')

        speak('Do you want to do another calculation?')
        query = takeCommand().lower()
        if 'y' in query:
            speak('ok which calculation you want to do?')
            query = takeCommand().lower()
            calculator()
```

```
            else:
                speak('ok')



        elif 'percent' in query:
                  speak('Enter a number whose percentage you want to
calculate')
             a = float(input("Enter a number whose percentage you want
to calculate :"))
            speak(f'How many percent of {a} you want to calculate?')
               b = float(input(f"Enter how many percentage of {a} you
want to calculate:"))
            c = (a*b)/100
            print(f"{b} % of {a} is {c}")
            speak(f'{b} percent of {a} is {c}. Your answer is {c}')


            speak('Do you want to do another calculation?')
            query = takeCommand().lower()
            if 'y' in query:
                speak('ok which calculation you want to do?')
                query = takeCommand().lower()
                calculator()
            else:
                speak('ok')


        elif 'interest' in query:
            speak('Enter the principal value or amount')
            p = float(input("Enter the principal value (P):"))
            speak('Enter the rate of interest per year')
                r = float(input("Enter the rate of interest per year
(%):"))
            speak('Enter the time in months')
            t = int(input("Enter the time (in months):"))
```

```
            interest = (p*r*t)/1200
            sint = round(interest)
            fv = round(p + interest)
            print(f"Interest = {interest}")
             print(f"The total amount accured, principal plus interest, from simple interest on a principal of {p} at a rate of {r}% per year for {t} months is {p + interest}.")
                speak(f'interest is {sint}. The total amount accured, principal plus interest, from simple interest on a principal of {p} at a rate of {r}% per year for {t} months is {fv}')

            speak('Do you want to do another calculation?')
            query = takeCommand().lower()
            if 'y' in query:
                speak('ok which calculation you want to do?')
                query = takeCommand().lower()
                calculator()
            else:
                speak('ok')




class music():
    def __init__(self):
                                                    self.driver      = webdriver.Chrome(executable_path='F:/py/chromedriver.exe')
    def play(self,query):
        self.query=query

self.driver.get(url="https://www.youtube.com/results?search_query="+query)
```

```
                                                                video       =
self.driver.find_element_by_xpath('//*[@id="dismissible"]')
        video.click()


class infow():
    def __init__(self):
                                                            self.driver      =
webdriver.Chrome(executable_path='F:/py/chromedriver.exe')


    def get_info(self,query):
        self.query=query
        self.driver.get(url="https://www.wikipedia.org")
                                                               search       =
self.driver.find_element_by_xpath('//*[@id="searchInput"]')
        search.click()
        search.send_keys(query)
                                                                enter       =
self.driver.find_element_by_xpath('//*[@id="search-form"]/fieldset/but
ton/i')
        enter.click()



r=sr.Recognizer()
wishMe()
h="Hello sir i am your voice assistant How are you?"
print(h)
speak(h)



with sr.Microphone() as source:
    r.enery_threshould=10000
    r.adjust_for_ambient_noise(source,1.2)
    print("listening....")
    audio = r.listen(source)
```

```
    text = r.recognize_google(audio)
    print(text)

if "what" and "about" and "you" in text:
    speak("i am also having a good day sir")
h2="what can i do for you?"
print(h2)
speak(h2)



with sr.Microphone() as source:
    r.enery_threshould=10000
    r.adjust_for_ambient_noise(source,1.2)
    print("listening...")
    audio = r.listen(source)
    text2 = r.recognize_google(audio)
    print(text2)

if "information" in text2:

    h3="you need information related to with topic"
    print(h3)
    speak(h3)

    with sr.Microphone() as source:
        r.enery_threshould=10000
        r.adjust_for_ambient_noise(source,1.2)
        print("listening...")
        audio = r.listen(source)
        infor = r.recognize_google(audio)
        print(infor)
    speak("searching {} in wikipedia".format(infor))
    assist=infow()
```

```
    assist.get_info(infor)

elif 'video' and 'play' in text2:
    h4="you want me to play which video?"
    print(h4)
    speak(h4)

    with sr.Microphone() as source:
        r.enery_threshould=10000
        r.adjust_for_ambient_noise(source,1.2)
        print("listening...")
        audio = r.listen(source)
        vid = r.recognize_google(audio)
        print(vid)
    speak("playing {} in youtube".format(vid))
    assist=music()
    assist.play(vid)




elif "fact" or "facts" in text2:
    speak("Sure sir,")
    x=randfacts.getFact()
    print(x)
    speak("Did you know that, "+x)



elif "joke" or "jokes" in text2:
    speak("sure sir, get ready for some chukles")
    #arr=joke()
    #print(arr[0])
    #speak(arr[0])
    #print(arr[1])
```

```
    #speak(arr[1])
    My_joke = pyjokes.get_joke(language="en", category="neutral")
    print(My_joke)

elif "send message " in text2:
                  # You need to create an account on Twilio to use this
service
                account_sid = 'Account Sid key'
                auth_token = 'Auth token'
                client = Client(account_sid, auth_token)

                message = client.messages \
                                .create(
                                    body = takeCommand(),
                                    from_='Sender No',
                                    to ='Receiver No'
                                )

                print(message.sid)



elif "calculator" in text2:
    with sr.Microphone() as source:
        r.enery_threshould=10000
        r.adjust_for_ambient_noise(source,1.2)
        print("listening...")
        assist=cal()
        assist.calculator(vid)

elif 'amazon' in text2:
            webbrowser.open('https://www.amazon.com')
            time.sleep(10)
elif 'flipkart' in text2:
```

```
            webbrowser.open('https://www.flipkart.com')
            time.sleep(10)
```

TEST RESULT

```
PS E:\5th Sem\Python\Project> python -u "e:\5th Sem\Python\Project\voice_Assistant.py"
Hello sir i am your voice assistant How are you?
listening....
I am fine what about you
what can i do for you?
listening...
can you can you give information
you need information related to with topic
```

Fig 6: Introduction Page

```
PS E:\5th Sem\Python\Project> python -u "e:\5th Sem\Python\Project\voice_Assistant.py"
Hello sir i am your voice assistant How are you?
listening....
I am fine what about you
what can i do for you?
listening...
can you do information
you need information related to with topic
listening...
python
```

Fig 7: Information Page

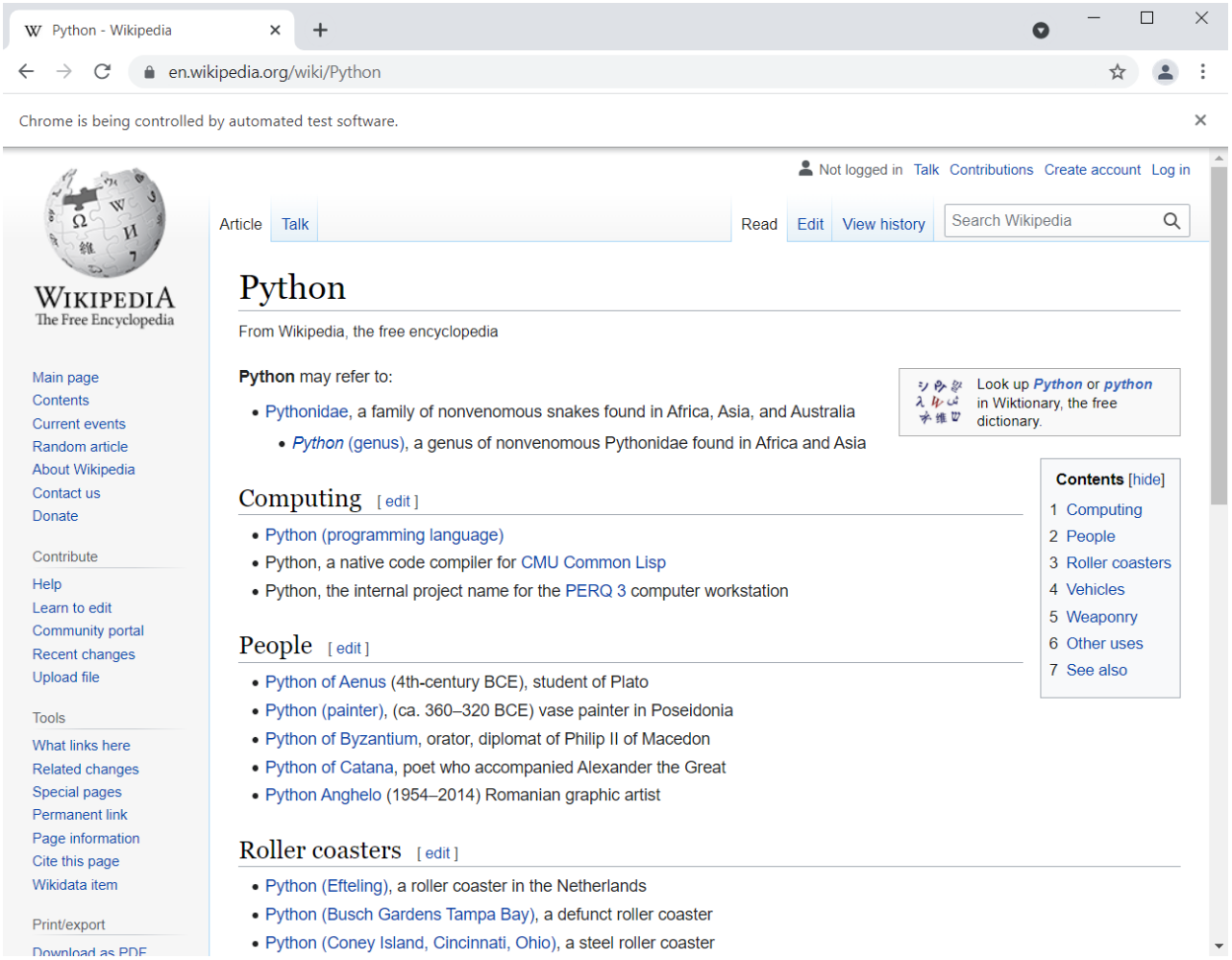


Fig 8: Wikipedia Page for Python

```
PS E:\5th Sem\Python\Project> python -u "e:\5th Sem\Python\Project\voice_Assistant.py"
Hello sir i am your voice assistant How are you?
listening....
I am fine what about you
what can i do for you?
listening...
can you play one video
you want me to play which video?
listening...
believer
```

Fig 9: Video Player Page

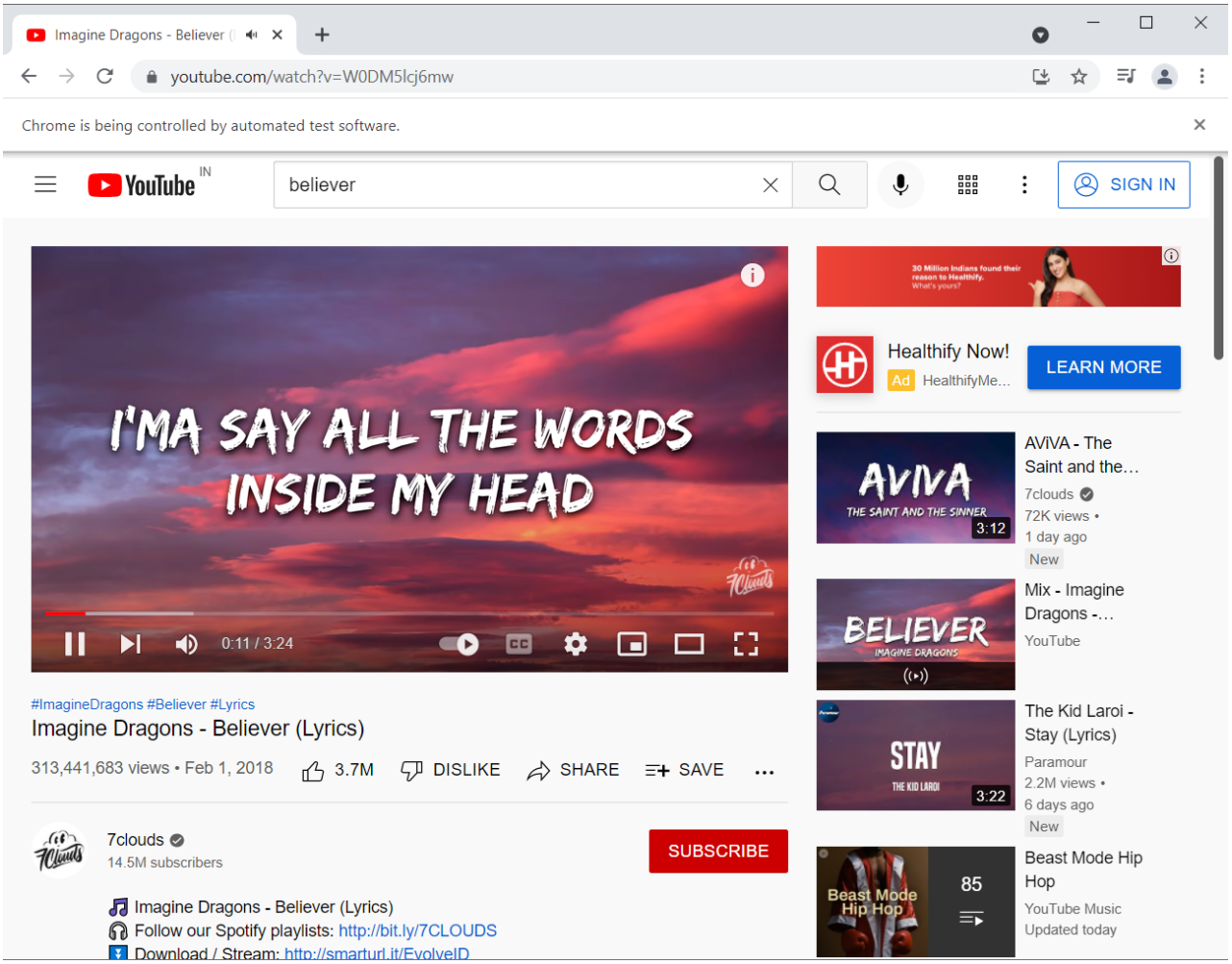


Fig 10: Believer Video Player

```
PS E:\5th Sem\Python\Project> python -u "e:\5th Sem\Python\Project\voice_Assistant.py"
Hello sir i am your voice assistant How are you?
listening....
I am fine what about you
what can i do for you?
listening...
tell me a joke
e:\5th Sem\Python\Project\voice_Assistant.py:421: DeprecationWarning: getFact is deprecated. Please use get_fact
  x=randfacts.getFact()
Koalas have human-like fingerprints.
PS E:\5th Sem\Python\Project>
```

Fig 11: Joke Page

```
PS E:\5th Sem\Python\Project> python -u "e:\5th Sem\Python\Project\voice_Assistant.py"
Hello sir i am your voice assistant How are you?
listening....
I am fine what about you
what can i do for you?
listening...
please can you tell fact
e:\5th Sem\Python\Project\voice_Assistant.py:421: DeprecationWarning: getFact is deprecated. Please use get_fact
  x=randfacts.getFact()
In 1938, Cliquot Club ginger ale was the first soft drink to be canned
PS E:\5th Sem\Python\Project>
```

Fig 12: Fact Page

CONCLUSION

Speech Recognition has become very important in today's world. With the advancements in technology and improvements in recognition algorithms, speech has become one of the primary source of input for many applications. Speech is the most efficient and natural way of communication.

So, it is intuitive that speech recognition systems have found applications in various fields. Interactive Voice Response (IVR) systems are one of the prominent systems that have a huge potential for use of voice signals as input to the system. With this in mind, we presented an idea for the development of an IVR system with Automatic Speech Recognition (ASR). The initial objective of the project was to develop a system capable of recognizing voice signals in Nepali Language input to the IVR system.

Throughout the course of the development phase, various limitations and obstacles were encountered which prompted us to develop the system capable of recognizing words corresponding to the digits of the Nepali Language. For this, we researched various methods of speech recognition and used the findings of these researches to develop the system.

REFERENCES

[1] Christopher Olah. Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[2] Andrej Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks. http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

[3] Md Salam, Dzulkifli Mohamad, and Sheikh Salleh. Malay isolated speech recognition using neural network: A work in finding number of hidden nodes and learning parameters. The International Arab Journal of Information Technology, 8, 2011.

[4] Hidden Markov model. In Wikipedia. https://en.wikipedia.org/wiki/Hidden_Markov_model.

[5] Markov model. In Wikipedia. https://en.wikipedia.org/wiki/Markov_model.