JARVO: VOICE ASSISTANT

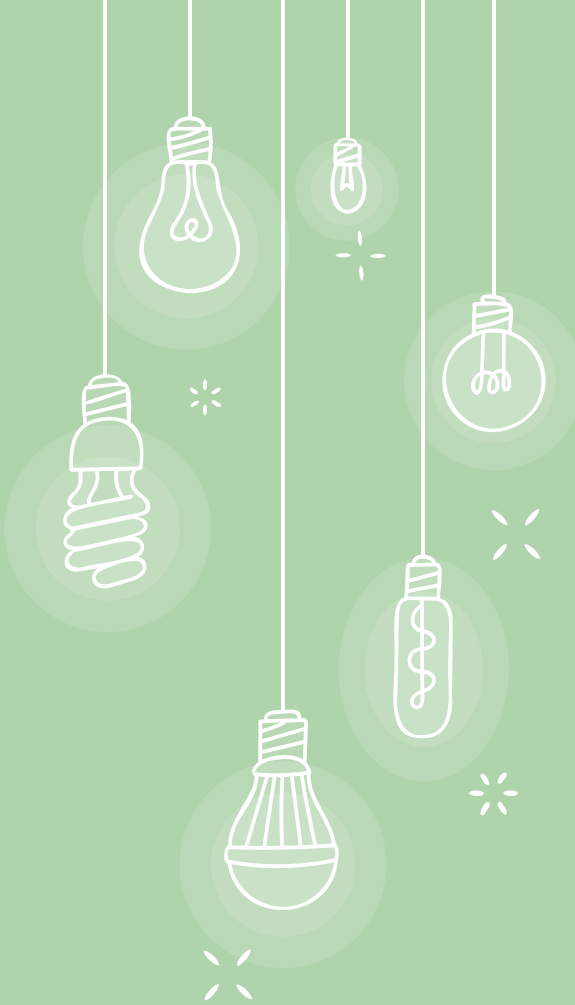
HELLO!

ARNAV SHARMA - E19CSE398

PRANJAL SINGH - E19CSE401

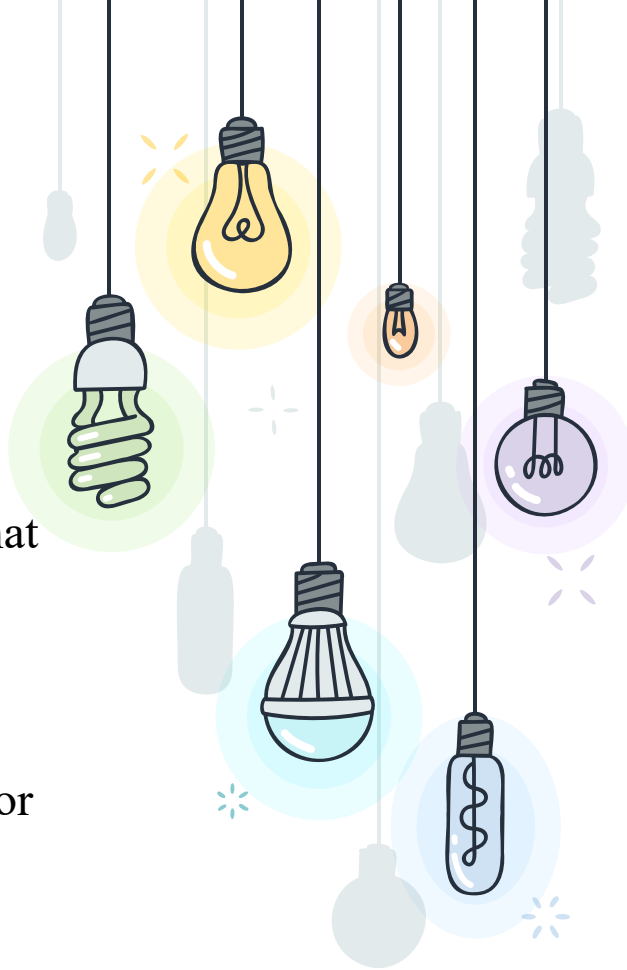
TUSHAR MARWAH – E19CSE399

VATSAL AGARWAL – E19CSE289



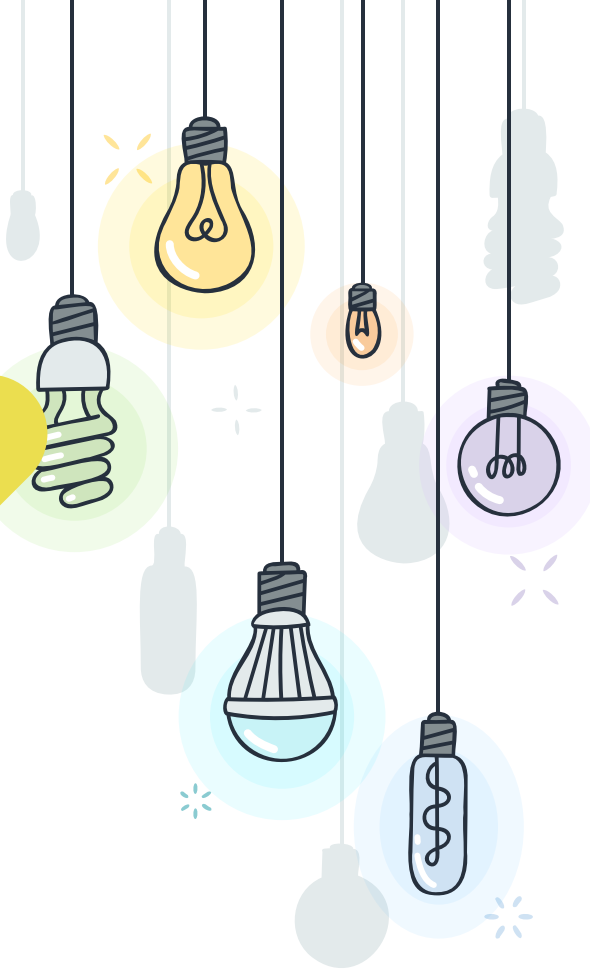
INTRODUCTION

- JARVO: The Voice assistant is an AI assistant that responds to verbal instructions given by the user.
- It allows us a hands-free control over our laptop and that too in a different variety of languages.
- It would also be able to send mails without typing anything, searching on Wikipedia, Google, YouTube, or any other search engine without even opening it.



MOTIVATION

1. It will help the user by reducing typing time
2. Will help your day to day tasks such as sending emails, surfing internet etc. really easy
3. Could be a helping hand for small kids and old persons



SOLUTION

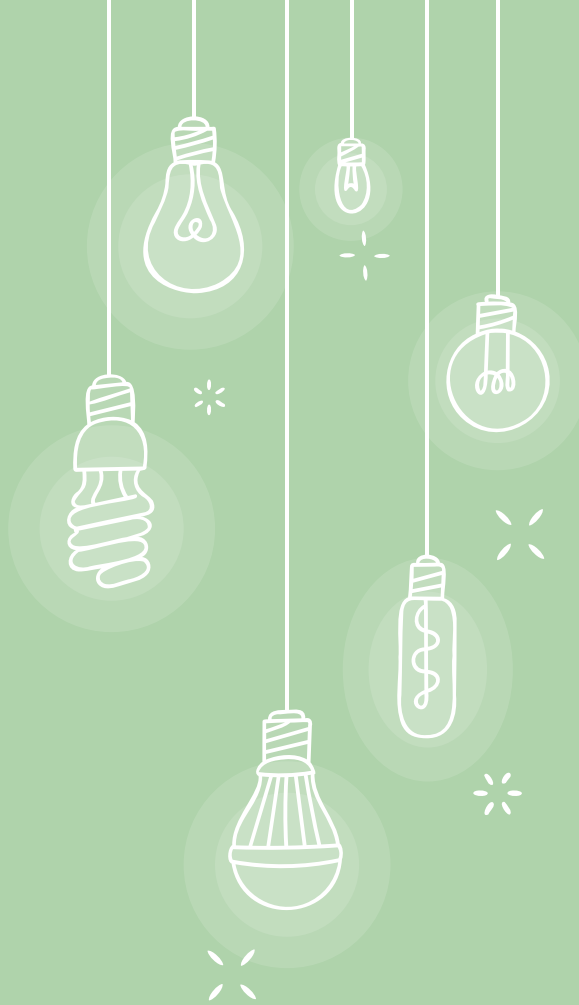
- Voice assistant quickly types the words much faster than a human as soon as they are spoken.
- This would make life easier and will save a lot of time.
- Kids who are in their learning stage will be able to use this voice assistant for much easier ways of learning
- Will surely serve as a bridge between a man and a machine.

Other Algorithms

Speech Recognition in car

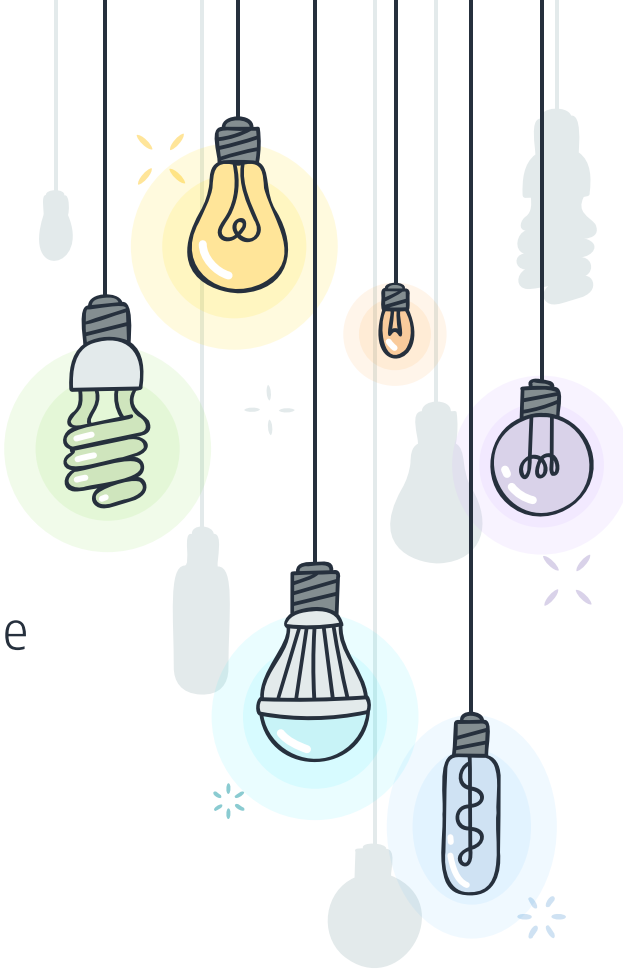
Digital assistant and speech recognition devices make things easier like -

You're in-car speech system will tell you that you need to fill up but also will guide you to the nearest petrol pump or would tell you the petrol pump is far away to reach with amount of fuel left



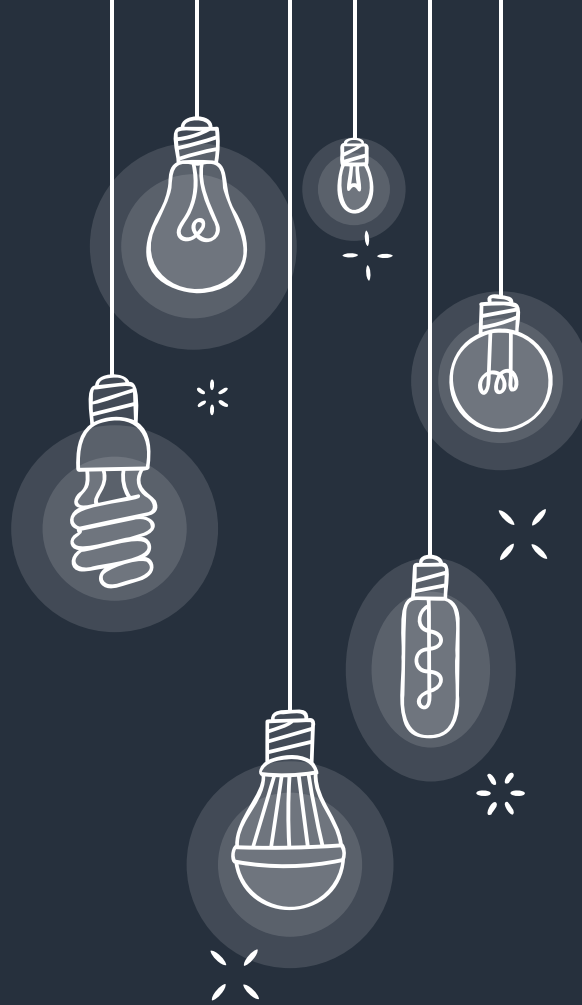
Analysis

- We constructed the wishme() function, which provides greeting functionality based on the time of our A.I. system.
- takeCommand() function takes command from the user.
- We added the ability to send emails, opening YouTube, doing Wikipedia searches etc.

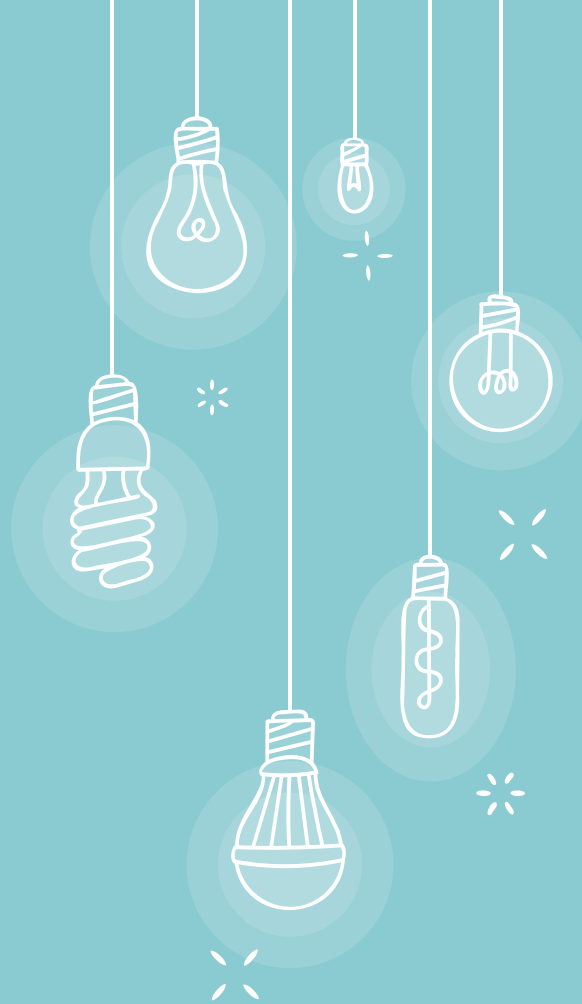


Snapshot of code

```
import pyttsx3
import speech_recognition as sr
import datetime
import wikipedia
import webbrowser
import os
import smtplib

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)


def speak(audio):
    engine.say(audio)
    engine.runAndWait()


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

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

    else:
        speak("Good Evening!")

    speak("I am Jarvo Sir. Please tell me how may I help you")

def takeCommand():


    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
        audio = r.listen(source)

```



```
    try:
        print("Recognizing...")
        query = r.recognize_google(audio, language='en-in')
        print(f"User said: {query}\n")

    except Exception as e:

        print("Say that again please...")
        return "None"
    return query

def sendEmail(to, content):
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.ehlo()
    server.starttls()
    server.login('youremail@gmail.com', 'your-password')
    server.sendmail('youremail@gmail.com', to, content)
    server.close()

if __name__ == "__main__":
    wishMe()
    while True:

        query = takeCommand().lower()


        if 'wikipedia' in query:
            speak('Searching Wikipedia...')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=2)
            speak("According to Wikipedia")
            print(results)
            speak(results)

        elif 'open youtube' in query:
            webbrowser.open("youtube.com")

        elif 'open google' in query:
            webbrowser.open("google.com")
```

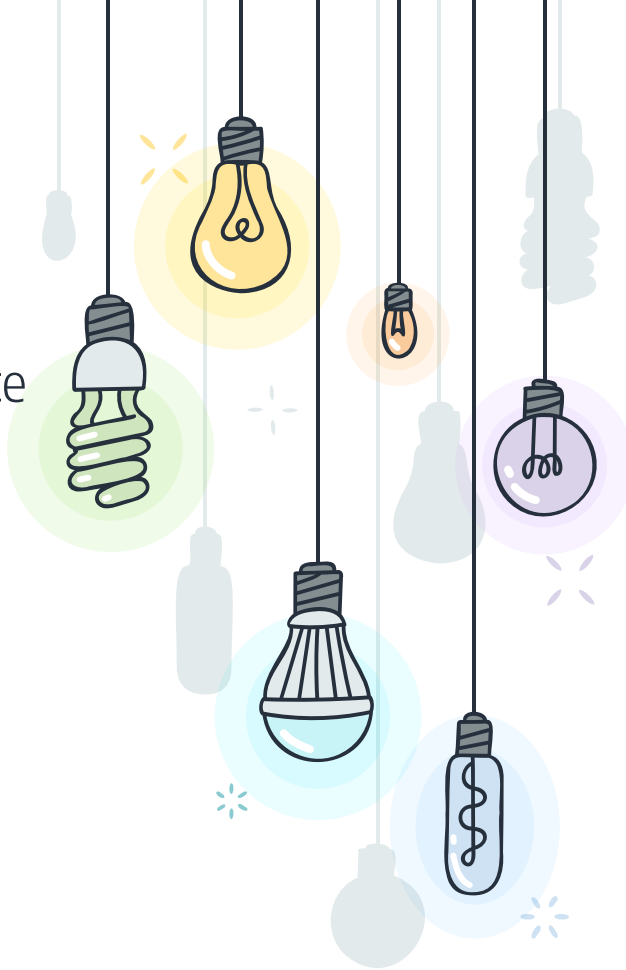


Result

- A simple voice assistant
- It can send emails on your behalf
- It can do Wikipedia searches
- Capable of opening YouTube, Google etc. in a browser.

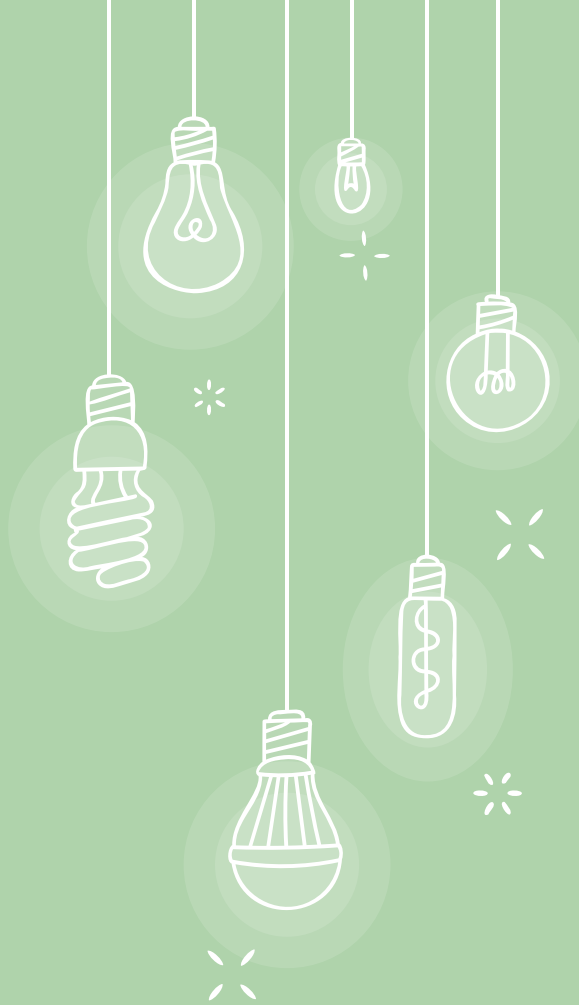
Conclusion

- Our voice assistant named Jarvo uses specific voice commands and returns some of important information or performs specificed actions as required by the user by using speech recognition, language processing and voice synthesis.



REFERENCES

- YouTube.com
- Google.com
- Github.com



Social Media Post