JARVIS Normal Source Code :

```
from __future__ import with_statement
import pyttsx3
import speech_recognition as sr
import datetime
import wikipedia
import webbrowser
import os
import random
import cv2
import pywhatkit as kit
import sys
import pyautogui
import time
import operator
import requests
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)
engine.setProperty('rate', 150)
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("Ready To Comply. What can I do for 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
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 "channel analytics" in query:
webbrowser.open("https://studio.youtube.com/channel/UCxeYbp9rU_HuIwVcuHvK0pw/analytics/tab-overview/period-default")
elif 'search on youtube' in query:
query = query.replace("search on youtube", "")
webbrowser.open(f"www.youtube.com/results?search_query={query}")
elif 'open youtube' in query:
speak("what you will like to watch ?")
qrry = takeCommand().lower()
kit.playonyt(f"{qrry}")
elif 'close chrome' in query:
os.system("taskkill /f /im chrome.exe")
elif 'close youtube' in query:
os.system("taskkill /f /im msedge.exe")
elif 'open google' in query:
speak("what should I search ?")
qry = takeCommand().lower()
webbrowser.open(f"{qry}")
results = wikipedia.summary(qry, sentences=2)
speak(results)
elif 'close google' in query:
os.system("taskkill /f /im msedge.exe")
elif 'play music' in query:
music_dir = 'E:\Musics'
songs = os.listdir(music_dir)
os.startfile(os.path.join(music_dir, random.choice(songs)))
```

```
elif 'play iron man movie' in query:

npath = "E:\ironman.mkv"

os.startfile(npath)

elif 'close movie' in query:

os.system("taskkill /f /im vlc.exe")

elif 'close music' in query:

os.system("taskkill /f /im vlc.exe")

elif 'the time' in query:

strTime = datetime.datetime.now().strftime("%H:%M:%S")

speak(f"Sir, the time is {strTime}")

elif "shut down the system" in query:

os.system("shutdown /s /t 5")

elif "restart the system" in query:

os.system("shutdown /r /t 5")

elif "Lock the system" in query:

os.system("rundll32.exe powrprof.dll,SetSuspendState 0,1,0")

#elif "open notepad" in query:

#npath = "C:\WINDOWS\system32\\notepad.exe"

#os.startfile(npath)

elif "close notepad" in query:

os.system("taskkill /f /im notepad.exe")

elif "open command prompt" in query:

os.system("start cmd")

elif "close command prompt" in query:

os.system("taskkill /f /im cmd.exe")

elif "open camera" in query:

cap = cv2.VideoCapture(0)

while True:
```

```
ret, img = cap.read()
cv2.imshow('webcam', img)
k = cv2.waitKey(50)
if k==27:
break;
cap.release()
cv2.destroyAllWndows()
elif "go to sleep" in query:
speak(' alright then, I am switching off')
sys.exit()
elif "take screenshot" in query:
speak('tell me a name for the file')
name = takeCommand().lower()
time.sleep(3)
img = pyautogui.screenshot()
img.save(f"{name}.png")
speak("screenshot saved")
elif "calculate" in query:
r = sr.Recognizer()
with sr.Microphone() as source:
speak("ready")
print("Listning...")
r.adjust_for_ambient_noise(source)
audio = r.listen(source)
my_string=r.recognize_google(audio)
print(my_string)
def get_operator_fn(op):
return {
```

```
'+' : operator.add,
'-' : operator.sub,
'x' : operator.mul,
'divided' : operator.__truediv__,
}[op]
def eval_bianary_expr(op1,oper, op2):
op1,op2 = int(op1), int(op2)
return get_operator_fn(oper)(op1, op2)
speak("your result is")
speak(eval_bianary_expr(*(my_string.split())))
elif "what is my ip address" in query:
speak("Checking")
try:
ipAdd = requests.get('https://api.ipify.org').text
print(ipAdd)
speak("your ip adress is")
speak(ipAdd)
except Exception as e:
speak("network is weak, please try again some time later")
elif "volume up" in query:
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
pyautogui.press("volumeup")
elif "volume down" in query:
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
pyautogui.press("volumedown")
elif "mute" in query:
pyautogui.press("volumemute")
elif "refresh" in query:
pyautogui.moveTo(1551,551, 2)
pyautogui.click(x=1551, y=551, clicks=1, interval=0, button='right')
pyautogui.moveTo(1620,667, 1)
pyautogui.click(x=1620, y=667, clicks=1, interval=0, button='left')
elif "scroll down" in query:
pyautogui.scroll(1000)
elif "drag visual studio to the right" in query:
pyautogui.moveTo(46, 31, 2)
pyautogui.dragRel(1857, 31, 2)
elif "rectangular spiral" in query:
pyautogui.hotkey('win')
time.sleep(1)
pyautogui.write('paint')
time.sleep(1)
pyautogui.press('enter')
pyautogui.moveTo(100, 193, 1)
pyautogui.rightClick
pyautogui.click()
distance = 300
while distance > 0:
pyautogui.dragRel(distance, 0, 0.1, button="left")
distance = distance - 10
pyautogui.dragRel(0, distance, 0.1, button="left")
pyautogui.dragRel(-distance, 0, 0.1, button="left")
distance = distance - 10
pyautogui.dragRel(0, -distance, 0.1, button="left")
elif "close paint" in query:
os.system("taskkill /f /im mspaint.exe")
elif "who are you" in query:
print('My Name Is Six')
```

```
speak('My Name Is Six')
print('I can Do Everything that my creator programmed me to do')
speak('I can Do Everything that my creator programmed me to do')
elif "who created you" in query:
print('I Do not Know His Name, I created with Python Language, in Visual Studio Code.')
speak('I Do not Know His Name, I created with Python Language, in Visual Studio Code.')
elif "open notepad and write my channel name" in query:
pyautogui.hotkey('win')
time.sleep(1)
pyautogui.write('notepad')
time.sleep(1)
pyautogui.press('enter')
time.sleep(1)
pyautogui.write("How To Manual", interval = 0.1)
elif "subscribe" in query:
print("Everyone Who are watching This, Please Subscribe Our Channel How To Manual for Interesting tutorials and information, Thanks For Watching")
speak("Everyone Who are watching This, Please Subscribe Our Channel How To Manual for Interesting tutorials and information, Thanks For Watching")
elif 'type' in query: #10
query = query.replace("type", "")
pyautogui.write(f"{query}")
```

Chrome Automation Source Code :

```
import pyttsx3
import speech_recognition as sr
import pyautogui
import time
import os
engine = pyttsx3.init('sapi5')
```

```
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)
engine.setProperty('rate', 150)
def speak(audio):
engine.say(audio)
engine.runAndWait()
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
if __name__ == "__main__":
while True:
query = takeCommand().lower()
if 'open chrome' in query:
os.startfile('C:\Program Files\Google\Chrome\Application\chrome.exe')
elif 'maximize this window' in query:
pyautogui.hotkey('alt', 'space')
time.sleep(1)
```

```
pyautogui.press('x')

elif 'google search' in query:

query = query.replace("google search", "")

pyautogui.hotkey('alt', 'd')

pyautogui.write(f"{query}", 0.1)

pyautogui.press('enter')

elif 'youtube search' in query:

query = query.replace("youtube search", "")

pyautogui.hotkey('alt', 'd')

time.sleep(1)

pyautogui.press('tab')

pyautogui.press('tab')

pyautogui.press('tab')

pyautogui.press('tab')

time.sleep(1)

pyautogui.write(f"{query}", 0.1)

pyautogui.press('enter')

elif 'open new window' in query:

pyautogui.hotkey('ctrl', 'n')

elif 'open incognito window' in query:

pyautogui.hotkey('ctrl', 'shift', 'n')

elif 'minimise this window' in query:

pyautogui.hotkey('alt', 'space')

time.sleep(1)

pyautogui.press('n')

elif 'open history' in query:

pyautogui.hotkey('ctrl', 'h')

elif 'open downloads' in query:
```

```
pyautogui.hotkey('ctrl', 'j')
elif 'previous tab' in query:
pyautogui.hotkey('ctrl', 'shift', 'tab')
elif 'next tab' in query:
pyautogui.hotkey('ctrl', 'tab')
elif 'close tab' in query:
pyautogui.hotkey('ctrl', 'w')
elif 'close window' in query:
pyautogui.hotkey('ctrl', 'shift', 'w')
elif 'clear browsing history' in query:
pyautogui.hotkey('ctrl', 'shift', 'delete')
elif 'close chrome' in query:
os.system("taskkill /f /im chrome.exe")
```

Image Recognition source code :

```
import pyttsx3
import speech_recognition as sr
import pyautogui
import time
import os
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)
engine.setProperty('rate', 150)
def speak(audio):
engine.say(audio)
engine.runAndWait()
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
if __name__ == "__main__":
while True:
query = takeCommand().lower()
if 'open chrome' in query:
img = pyautogui.locateCenterOnScreen('Screenshot1.png') #[take a screenshot of chrome and crop it, then save the image in jarvis folder]
pyautogui.doubleClick(img)
time.sleep(1)
pyautogui.hotkey('alt', 'space')
time.sleep(1)
pyautogui.press('x')
time.sleep(1)
img1 = pyautogui.locateCenterOnScreen('screenshot2.png') #[take screenshot where you want to make the click]
pyautogui.click(img1)
time.sleep(2)
img2 = pyautogui.locateCenterOnScreen('screenshot3.png')
```

```
pyautogui.click(img2)

time.sleep(1)

pyautogui.typewrite('How To Manual',0.1)

pyautogui.press('enter')

time.sleep(1)

pyautogui.press('esc')

img3 = pyautogui.locateCenterOnScreen('screenshot4.png')

pyautogui.click(img3)

elif 'close chrome' in query:

os.system("taskkill /f /im chrome.exe")
```

If You Face Any Problem tell me in the comment section.