Borneo Engineering & Advanced Multidisciplinary International Journal (BEAM)
Volume 1, Issue 2, November 2022, Pages 18-26

AI Virtual Personal Assistant Based on Human Presence

Khairul Huda Yusof[1]*, Vikneshwaran Alagan[1], Gokuela Geshnu[1], Fadhilah Aman[2], Norazliani Md Sapari[3]

[1]Faculty of Information Science & Engineering, Management & Science University, Shah Alam, Selangor, Malaysia
[2]College of Computer, Informatic and Media, Universiti Teknologi MARA Selangor Branch, Puncak Perdana Campus, 40150 Shah Alam, Malaysia
[3]School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, 81300 Johor Bharu, Malaysia

*Corresponding author: khairul_huda@msu.edu.my
Please provide an **official organization email** of the corresponding author

Full Paper

Article history
Received
21 July 2022
Received in revised form
26 October 2022
Accepted
27 October 2022
Published online
1 November 2022

Abstract

This paper proposes an Artificial Intelligence (AI) virtual personal assistant based on human presence. A proposed Virtual Assistant is developed on Raspberry Pi4 using the Speech Recognition method (Speech-to-text and Text-to-Speech). The result shows that a proposed personal assistant allows a user to ask questions, in the same way they would to a human and can even perform simple tasks such as opening apps, reading news, and taking notes with a voice command. Even when no one is around, the artificial intelligence virtual assistant will continue to listen for commands. The system will start gathering data and transmitting it to the cloud when it is designed to listen to commands continuously. So, if someone gets into the system, they may gain access and divulge sensitive information. The suggested system includes a motion sensor that may detect human presence and activate an Artificial Intelligence virtual assistant system before turning off the system after a certain amount of time if no human presence is detected. This prevents the system from being on constantly and increases security. The system can perform several tasks, like playing music, opening websites, sending emails and messages, restarting the computer, and more.

Keywords: - Virtual assistant; Raspberry Pi4; speech recognition; Artificial Intelligence

1. Introduction

Almost all tasks are now digitalized in today's world. We have a Smartphone in our hands, and it's like having the entire world at our fingertips. We don't even use our fingers anymore. We discussed the assignment, and it is completed. There are systems where we can text Mom, saying, "I'll be late today." The text is then sent. A Virtual Assistant's job is to do just that.

Virtual Assistants are computer programmes that assist you with daily tasks such as messaging, reading articles on the internet, playing music, and sending emails. They can accept voice commands. To activate the listener, voice-based intelligent assistants require an invoking phrase or wake word, followed by the order (Abougarair, Aburakhis & Zaroug, 2022).

JARVIS is the wake word for our project, and a motion sensor senses our presence. This system is intended for usage on desktop computers. Personal assistant software helps users be more productive by managing routine tasks and giving information from web sources. JARVIS is simple to operate. The order should be followed by the wake word 'JARVIS', and it's done in a matter of seconds.

Voice searches have surpassed text searches in popularity (Dekate, Kulkarni & Killedar, 2016). According to analysts, web searches performed on mobile devices have recently surpassed those completed on computers, with 48 per cent of customers utilizing virtual assistants for web searches. Virtual assistants are proving to be more intelligent than ever. Allow your smart assistant to manage your email for you. Detect intent, extract critical data, automate processes, and provide customized solutions.

Thus, this paper proposes a project that equips the PC with a Virtual Assistant, which will assist in daily routine tasks such as viewing YouTube, sending WhatsApp messages, reading articles on the internet, and many others, but will also assist in the automation of various operations.

2. Literature Review on Virtual Assistants

Several desktop virtual assistants already exist. This section discusses a few examples of existing virtual assistants on the market and the duties they can perform (O'Shaughnessy, 2003).

2.1 Google Assistant

Google Assistant is the company's virtual assistant. Google Assistant was originally intended to be a personal extension of Google Now, expanding on Google's existing "OK Google" voice commands. After saying the "OK Google" or "Hey Google" wake words, Google Assistant provides voice commands, speech searching, and voice-activated device control, allowing you to execute various tasks. Its purpose is to provide you with conversational interactions.

Google Assistant will:

- Control your smart home and your gadgets
- Access information from your calendars as well as other personal data.
- From meal reservations to directions, weather, and news, you can get it all online.
- Manage your music
- Use your Chromecast or other compatible devices to watch content.
- Set timers and alarms.
- Schedule meetings and send messages
- Open your phone's apps.
- Examine your alerts.
- Translations in real-time
- Play video games

2.2 Amazon's Alexa

Alexa is Amazon's digital virtual assistant, and she's quickly becoming a universal Artificial Intelligence companion in our homes, phones, and even cars. Many people are flocking to Amazon Echo devices due to their popularity, only to wonder, "What can Alexa actually do?"

Alexa can:

- Play radio stations
- Play music from Spotify, Apple Music, and other services.
- Play podcasts
- Set alarms and timers.
- Get personalized news reports
- Pose web-based inquiries.
- Voice control for smart home devices
- Combine smart home gadgets to make them work together with a single command.
- Make phone calls to people you know.
- Home-based intercom between Echo speakers
- Play video games
- Play soothing music or background noise (and toggle it off automatically)
- Allows you to use your voice to operate a TV or AV system.
- Increase the number of levels in your smart security system.
- Looking after an elderly relative

3. Advantages and Disadvantages of Artificial Intelligence

A computer program that is intelligent artificially can think and learn. Anything that involves a program carrying out a task that we would typically believe a human would carry out qualifies as artificial intelligence.

3.1 Advantages

a) Privacy

When it comes to using virtual assistants, privacy is the main issue. They are simple to transfer and store, making it possible for everyone to obtain them. And you particularly don't want it to happen with confidential company information. The user won't need to worry about the proposed virtual assistant since it maintains the need for human presence to make the virtual assistant available to the user, so the security is enhanced and vulnerability is avoided. Other virtual assistants like Siri lacks this feature.

b) Always available

Numerous studies have shown that people only work productively for three to four hours each day on average. People also require breaks and vacation time to balance their personal and professional lives. However, a virtual assistant can operate continuously without rest. They can multitask with accuracy and think far more quickly than humans can. With the aid of Artificial Intelligence algorithms, they can even do difficult, repetitive tasks without difficulty.

c) Everyday applications

Today, the internet and mobile gadgets are necessary for our daily activities. We use several different programmes, such as the search on Google, texting a person via WhatsApp, playing Youtube songs or videos on a Raspberry Pi computer, send an email. Other tools include providing a date and time, giving welcome greetings, and simple games like tossing a coin and rolling a die.

3.2 Disadvantages

a) No inventiveness

The main reason limiting the proposed artificial intelligence's usefulness is that it currently has few cool features. The inability of Artificial Intelligence to learn to think creatively beyond the box is a significant drawback. With pre-fed facts and prior experiences, Artificial Intelligence can learn over time, but it lacks creativity.

b) Limitation of processor

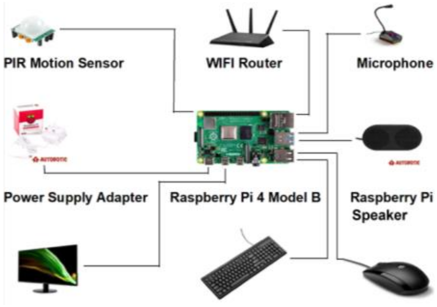
The virtual assistant occasionally needs help understanding what people are saying because it only uses the quad-core Cortex-A72 processor of the Raspberry Pi 4 minicomputer. It would either need help connecting, transcribing your dictation into gibberish, delivering online results you didn't ask for, or doing all three of these things.

4. Methodology of proposed Virtual Assistance

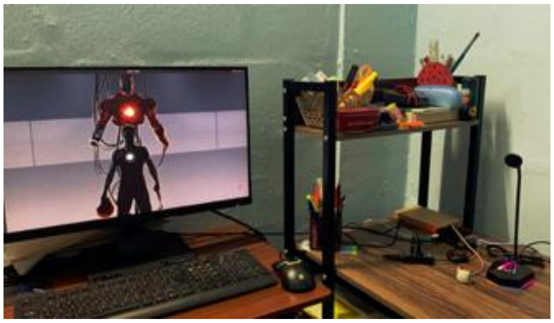
In this section, two major components will be used in this project: hardware and software.

4.1 Hardware Component:

Fig. 1(a) indicates the peripheral used to build the Raspberry Pi computer. The green colour PCB board at the centre of the image is the Raspberry Pi4 Model. It comes with 8 GB built-in RAM, which acts as a computer in this project and is connected to some hardware peripherals to make it a complete computer. Peripherals that are connected, such as a keyboard and mouse for the user to make navigation and input, a power supply adapter to power the Raspberry Pi, monitor and speaker to produce visual and sound output for the user. Other peripherals include a microphone to take voice commands from the user, PIR Sensor to detect human motion or presence and a router connected via Ethernet cable for a stable internet connection. The overall setup of the proposed project is shown in Fig. 1(b).



(a)



(b)

Fig.1. (a) The circuit of the proposed design and (b) the actual setup of the Raspberry Pi4 computer setup

As shown in Fig. 2, the PIR Sensor is required to have three connections, first is the VCC for power input which is connected to 5V pin 2, the second is GND as a ground connection, and it is connected to ground pin number 14 and third which is the output connection. It is connected to pin number 16 GPIO 23. Meanwhile, for the Raspberry Pi4 case, the built-in fan is connected to pin number 4 to 5V voltage power and ground pin number 6.

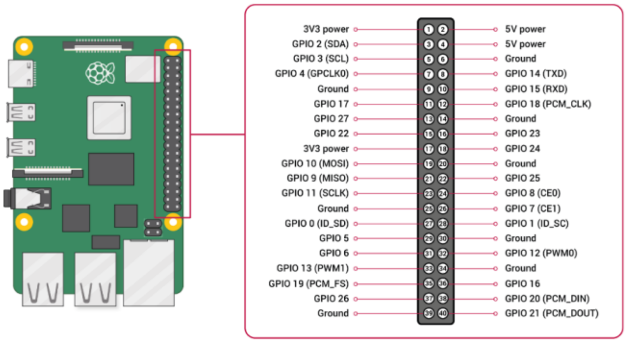
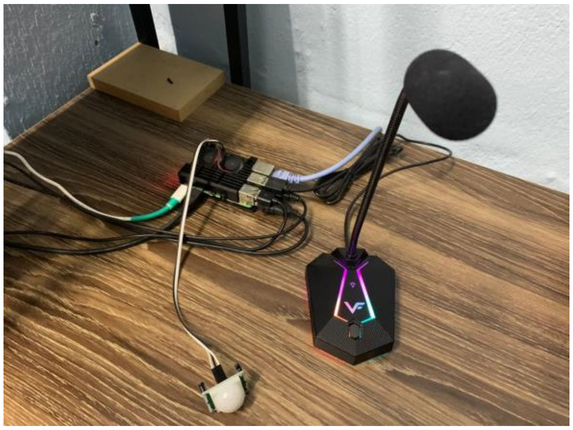
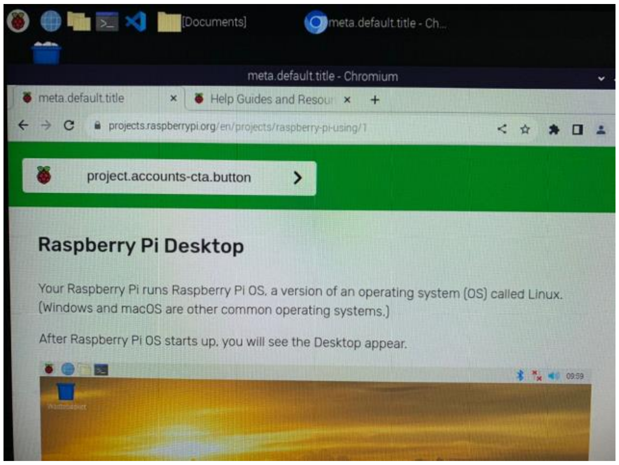


Fig. 2. Shows Raspberry Pi4 GPIO pinout

As shown in Fig. 3(a) and (b), the Raspberry Pi has been installed with the Linux Ubuntu 20.04 operating system and basic computer peripherals. This system has a PIR Motion Sensor installed in it for the activation of an Artificial Intelligence Virtual Assistant when it detects a human presence. The microphone will take the input into the computer, and the Raspberry Pi speaker will deliver the output to the user.



(a)


[Documents]
meta.default.title - Ch...
meta.default.title - Chromium
meta.default.title
Help Guides and Resour
projects.raspberrypi.org/en/projects/raspberry-pi-using/1
project.accounts-cta.button

Raspberry Pi Desktop

Your Raspberry Pi runs Raspberry Pi OS, a version of an operating system (OS) called Linux. (Windows and macOS are other common operating systems.)

After Raspberry Pi OS starts up, you will see the Desktop appear.

(b)

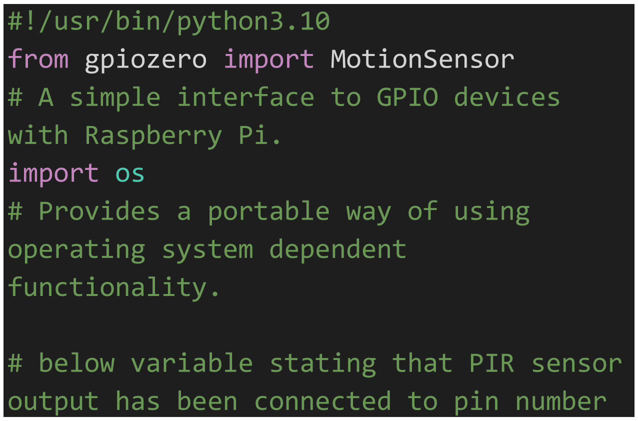
Fig. 3. (a) Top view of Raspberry Pi4 connections and (b) the operating system installed on Raspberry Pi4.

This raspberry Pi device is installed with the Raspberry Pi OS 64-bit version, which Raspberry Pi Foundation develops under the Linux OS family.

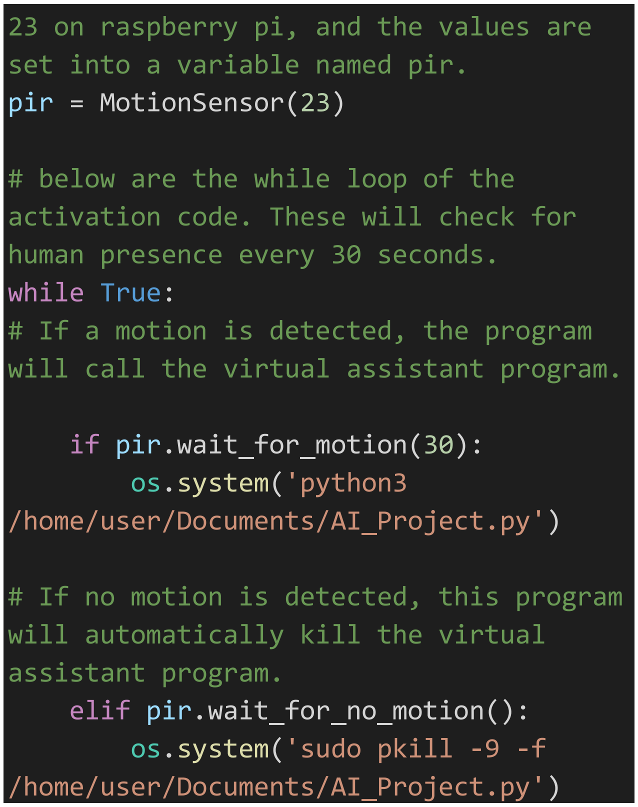
4.2 Software Component:

In this part, the Sensor Activation program will start upon the operating system boot-up. Once it does, it will look for human presence to run the Artificial Intelligence Virtual Assistant program. If it does not detect human presence, it will wait forever for activation. Once the Artificial Intelligence Virtual Assistant program is activated, it will look for the wake-up word, which is the Virtual Assistant's name. If the user does not call by the Virtual Assistant's name, it won't respond, so the name for this Virtual Assistant is JARVIS. Then it goes to the if-else statement query, where the Artificial Intelligence takes commands from the user to the process, executes it, and repeats the loop. If the user is given a quit command, the program will quit and return to sensor activation.

Fig. 4 shows the Motion Sensor code, which acts as an activation code for the Virtual Assistant program. It will keep on checking for human presence, and when there is a human presence, it will call the Virtual Assistant and exit the Virtual Assistant program when there is no one.



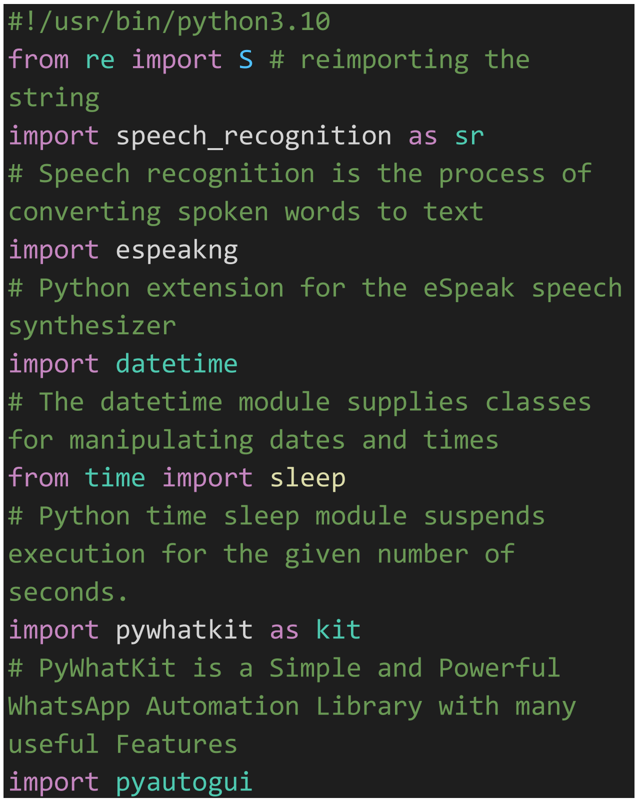
```
#!/usr/bin/python3.10
from gpiozero import MotionSensor
# A simple interface to GPIO devices
with Raspberry Pi.
import os
# Provides a portable way of using
operating system dependent
functionality.

# below variable stating that PIR sensor
output has been connected to pin number
```



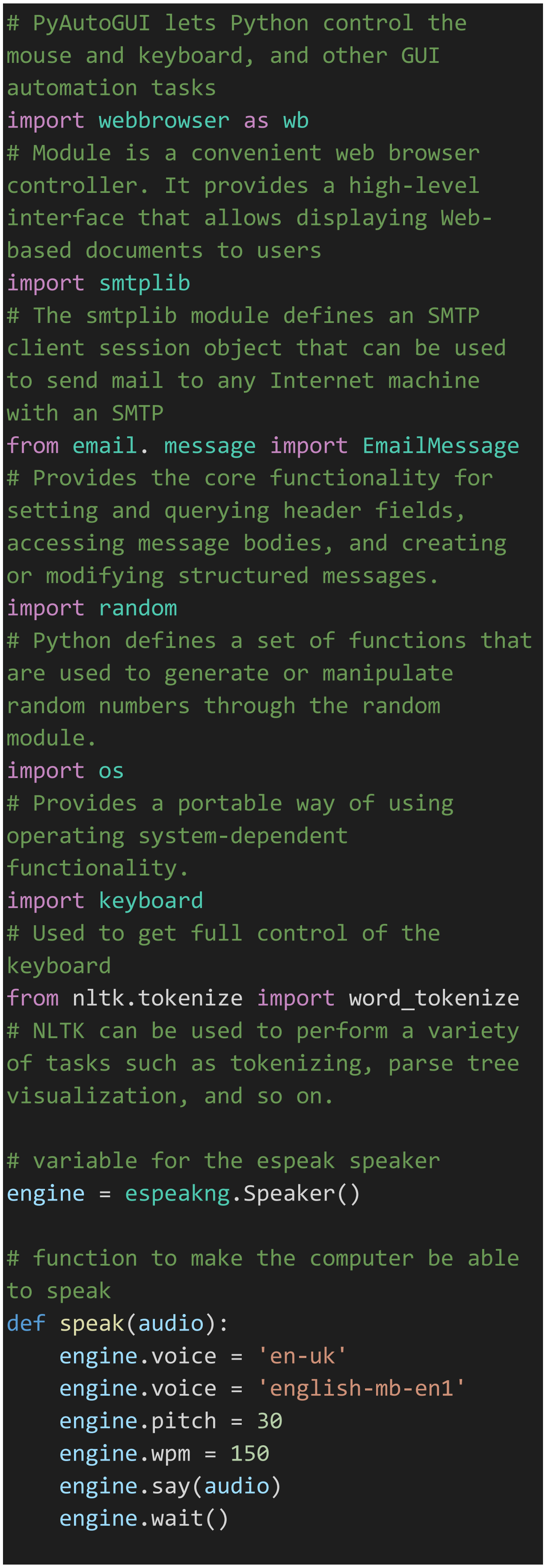
```
23 on raspberry pi, and the values are
set into a variable named pir.
pir = MotionSensor(23)

# below are the while loop of the
activation code. These will check for
human presence every 30 seconds.
while True:
# If a motion is detected, the program
will call the virtual assistant program.

    if pir.wait_for_motion(30):
        os.system('python3
/home/user/Documents/AI_Project.py')

# If no motion is detected, this program
will automatically kill the virtual
assistant program.
    elif pir.wait_for_no_motion():
        os.system('sudo pkill -9 -f
/home/user/Documents/AI_Project.py')
```

Fig. 4: Motion sensor code of the proposed project

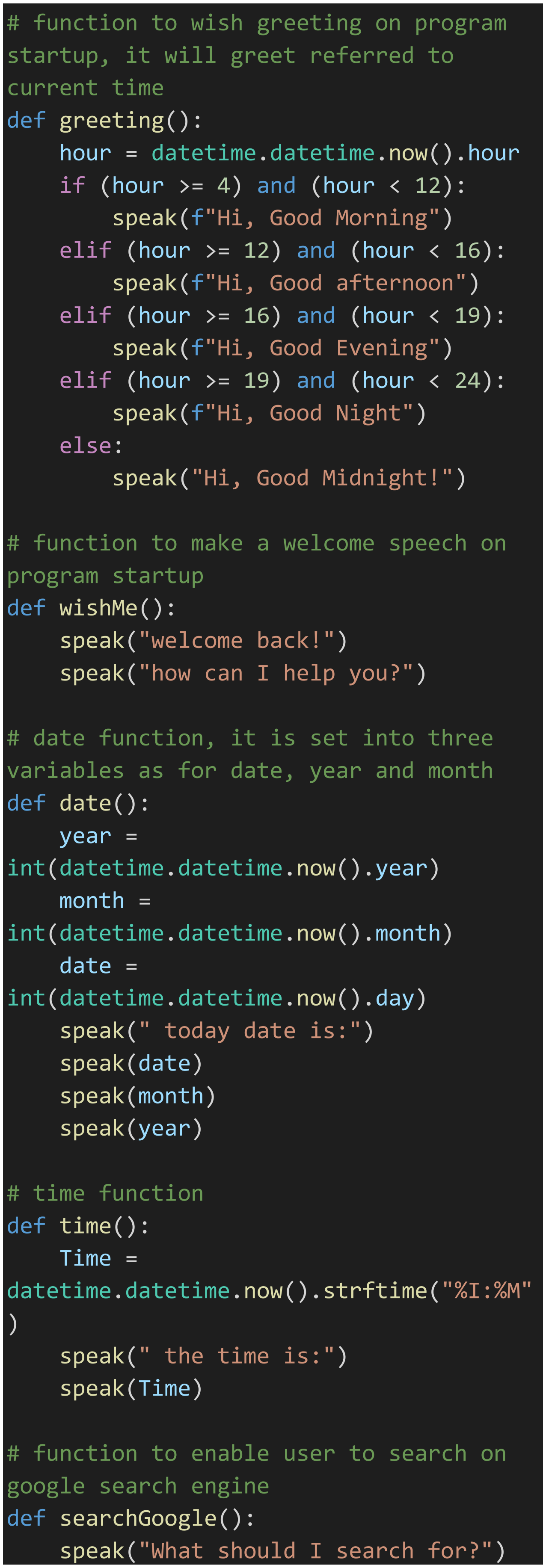
Meanwhile, Fig. 5 below are some of the source codes of Artificial Intelligence Virtual Assistant:



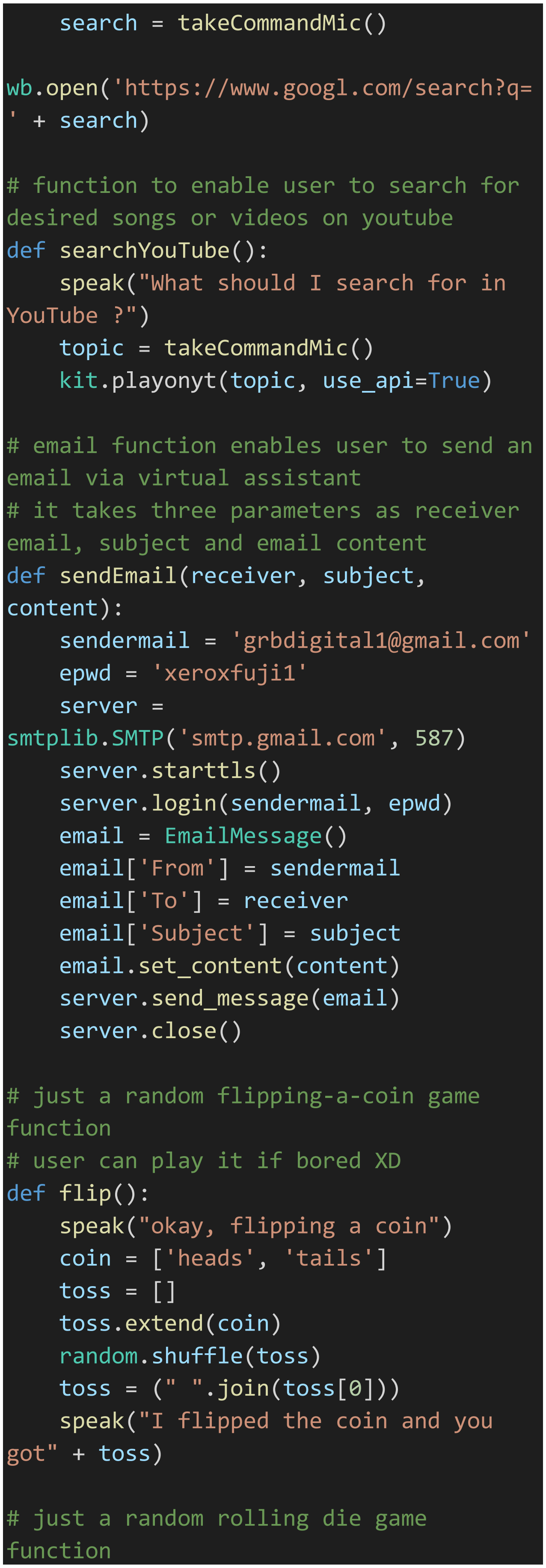
```
#!/usr/bin/python3.10
from re import S # reimporting the
string
import speech_recognition as sr
# Speech recognition is the process of
converting spoken words to text
import espeakng
# Python extension for the eSpeak speech
synthesizer
import datetime
# The datetime module supplies classes
for manipulating dates and times
from time import sleep
# Python time sleep module suspends
execution for the given number of
seconds.
import pywhatkit as kit
# PyWhatKit is a Simple and Powerful
WhatsApp Automation Library with many
useful Features
import pyautogui
```



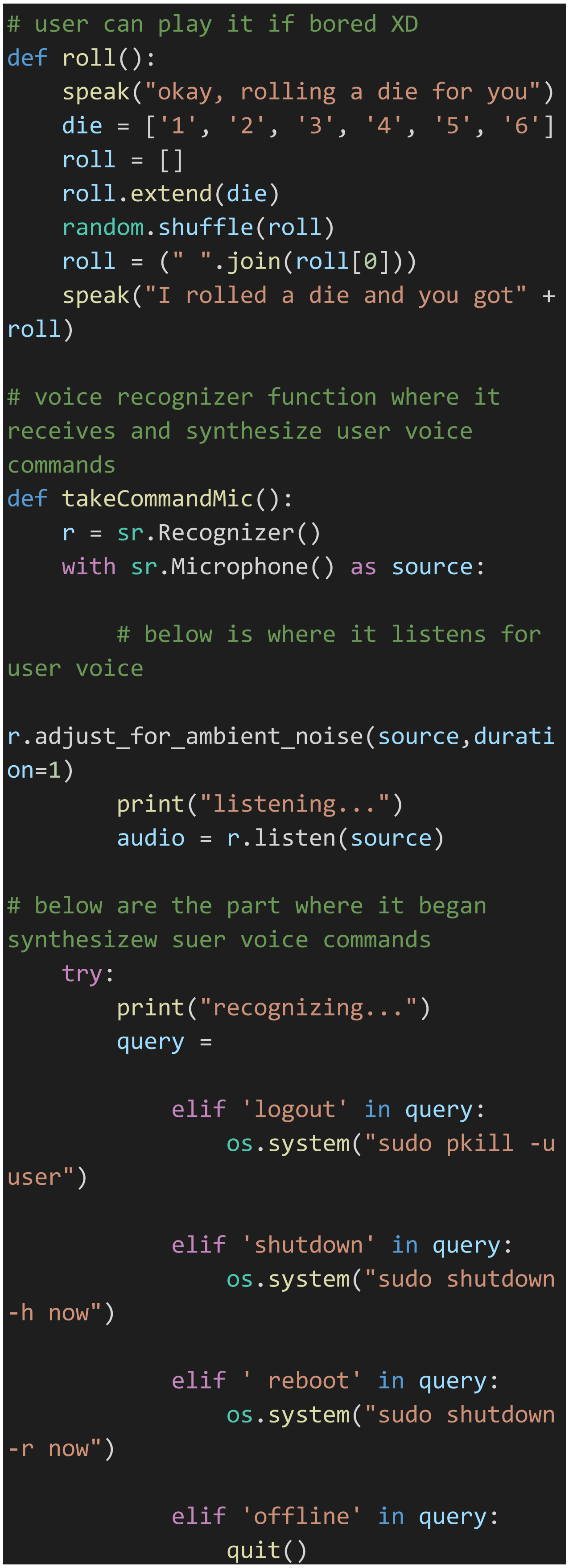
```
# PyAutoGUI lets Python control the
mouse and keyboard, and other GUI
automation tasks
import webbrowser as wb
# Module is a convenient web browser
controller. It provides a high-level
interface that allows displaying Web-
based documents to users
import smtplib
# The smtplib module defines an SMTP
client session object that can be used
to send mail to any Internet machine
with an SMTP
from email. message import EmailMessage
# Provides the core functionality for
setting and querying header fields,
accessing message bodies, and creating
or modifying structured messages.
import random
# Python defines a set of functions that
are used to generate or manipulate
random numbers through the random
module.
import os
# Provides a portable way of using
operating system-dependent
functionality.
import keyboard
# Used to get full control of the
keyboard
from nltk.tokenize import word_tokenize
# NLTK can be used to perform a variety
of tasks such as tokenizing, parse tree
visualization, and so on.

# variable for the espeak speaker
engine = espeakng.Speaker()

# function to make the computer be able
to speak
def speak(audio):
    engine.voice = 'en-uk'
    engine.voice = 'english-mb-en1'
    engine.pitch = 30
    engine.wpm = 150
    engine.say(audio)
    engine.wait()
```



```
# function to wish greeting on program
startup, it will greet referred to
current time
def greeting():
    hour = datetime.datetime.now().hour
    if (hour >= 4) and (hour < 12):
        speak(f"Hi, Good Morning")
    elif (hour >= 12) and (hour < 16):
        speak(f"Hi, Good afternoon")
    elif (hour >= 16) and (hour < 19):
        speak(f"Hi, Good Evening")
    elif (hour >= 19) and (hour < 24):
        speak(f"Hi, Good Night")
    else:
        speak("Hi, Good Midnight!")

# function to make a welcome speech on
program startup
def wishMe():
    speak("welcome back!")
    speak("how can I help you?")

# date function, it is set into three
variables as for date, year and month
def date():
    year =
int(datetime.datetime.now().year)
    month =
int(datetime.datetime.now().month)
    date =
int(datetime.datetime.now().day)
    speak(" today date is:")
    speak(date)
    speak(month)
    speak(year)

# time function
def time():
    Time =
datetime.datetime.now().strftime("%I:%M"
)
    speak(" the time is:")
    speak(Time)

# function to enable user to search on
google search engine
def searchGoogle():
    speak("What should I search for?")
```



```
    search = takeCommandMic()

wb.open('https://www.googl.com/search?q=
' + search)

# function to enable user to search for
desired songs or videos on youtube
def searchYouTube():
    speak("What should I search for in
YouTube ?")
    topic = takeCommandMic()
    kit.playonyt(topic, use_api=True)

# email function enables user to send an
email via virtual assistant
# it takes three parameters as receiver
email, subject and email content
def sendEmail(receiver, subject,
content):
    sendermail = 'grbdigital1@gmail.com'
    epwd = 'xeroxfuji1'
    server =
smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login(sendermail, epwd)
    email = EmailMessage()
    email['From'] = sendermail
    email['To'] = receiver
    email['Subject'] = subject
    email.set_content(content)
    server.send_message(email)
    server.close()

# just a random flipping-a-coin game
function
# user can play it if bored XD
def flip():
    speak("okay, flipping a coin")
    coin = ['heads', 'tails']
    toss = []
    toss.extend(coin)
    random.shuffle(toss)
    toss = (" ".join(toss[0]))
    speak("I flipped the coin and you
got" + toss)

# just a random rolling die game
function
```



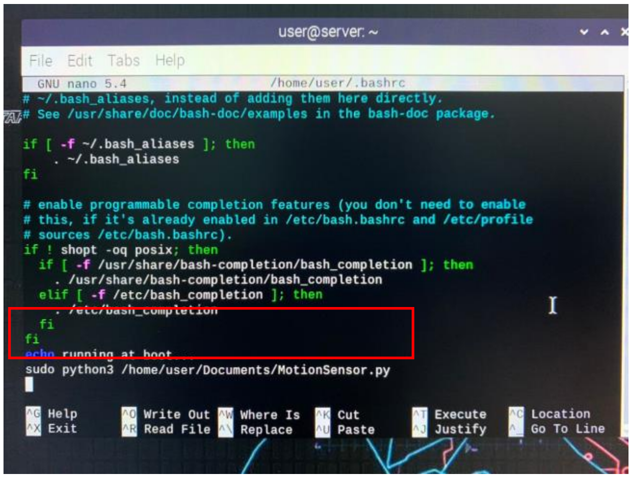
```
# user can play it if bored XD
def roll():
    speak("okay, rolling a die for you")
    die = ['1', '2', '3', '4', '5', '6']
    roll = []
    roll.extend(die)
    random.shuffle(roll)
    roll = (" ".join(roll[0]))
    speak("I rolled a die and you got" +
roll)

# voice recognizer function where it
receives and synthesize user voice
commands
def takeCommandMic():
    r = sr.Recognizer()
    with sr.Microphone() as source:

        # below is where it listens for
user voice

r.adjust_for_ambient_noise(source,durati
on=1)
        print("listening...")
        audio = r.listen(source)

# below are the part where it began
synthesizew suer voice commands
    try:
        print("recognizing...")
        query =

            elif 'logout' in query:
                os.system("sudo pkill -u
user")

            elif 'shutdown' in query:
                os.system("sudo shutdown
-h now")

            elif ' reboot' in query:
                os.system("sudo shutdown
-r now")

            elif 'offline' in query:
                quit()
```

Fig. 5. Source code of Artificial Intelligence Virtual Assistant

5. Artificial Intelligence JARVIS Functions and Result

The program was created using Python 3 programming language with the aid of many Python prebuild libraries, and it will start the program with welcoming greetings. The functions included are telling the current date and time, browsing on google and YouTube, system shutdown, rebooting and logging out, and finally, sending WhatsApp messages and emails via voice command. Throughout this example, the task was given via voice command. First of all, you need to change the.bashrc file to run the Motion Sensor code, which will then activates the Virtual Assistant program on the Raspberry Pi upon reboot.

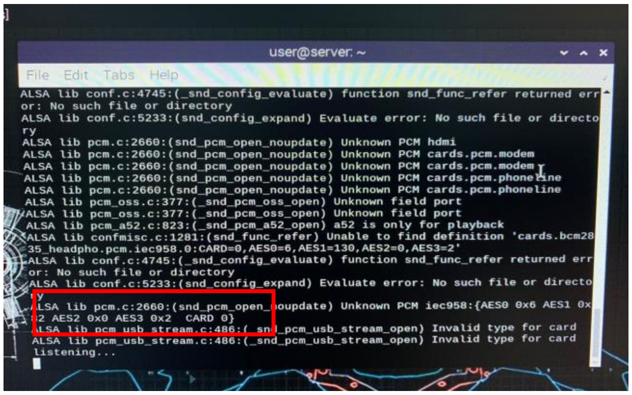
Using the.bashrc technique, your Python application will execute every time a new terminal or SSH connection is launched and when you log in, which happens immediately when you boot up and go straight to the desktop. Place your command at the end of '/home/pi/.bashrc'. The program can be aborted with 'ctrl-c' while it is running! *sudo nano /home/pi/.bashrc* Go to the last line of the script and add: *echo running at boot…* and *sudo python /home/pi/sample.py*. The echo command is used to demonstrate that the commands in the.bashrc file are run upon boot-up and when logging into the bash console, as shown in Fig. 6.



```
  GNU nano 5.4                    /home/user/.bashrc
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
echo running at boot...
sudo python3 /home/user/Documents/MotionSensor.py

^G Help      ^O Write Out ^W Where Is ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace  ^U Paste    ^J Justify  ^_ Go To Line
```

Fig.6. Shows how to add motion sensor code in a bash file

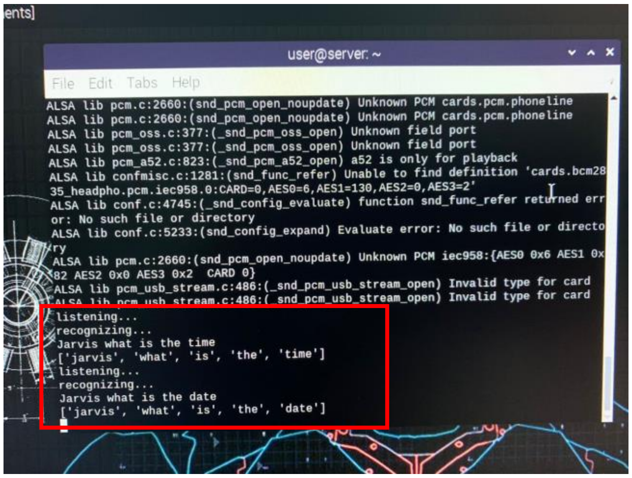
Now reboot the computer, and it should run the Motion Sensor code when it boots up, and it will wake up the Virtual Assistant program when it detects human presence as shown in Fig. 7.



```
ALSA lib conf.c:4745:(_snd_config_evaluate) function snd_func_refer returned err
or: No such file or directory
ALSA lib conf.c:5233:(snd_config_expand) Evaluate error: No such file or directo
ry
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM hdmi
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.modem
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm_oss.c:377:(_snd_pcm_oss_open) Unknown field port
ALSA lib pcm_oss.c:377:(_snd_pcm_oss_open) Unknown field port
ALSA lib pcm_a52.c:823:(_snd_pcm_a52_open) a52 is only for playback
ALSA lib confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.bcm28
35_headpho.pcm.iec958.0:CARD=0,AES0=6,AES1=130,AES2=0,AES3=2'
ALSA lib conf.c:4745:(_snd_config_evaluate) function snd_func_refer returned err
or: No such file or directory
ALSA lib conf.c:5233:(snd_config_expand) Evaluate error: No such file or directo
ry
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM iec958:{AES0 0x6 AES1 0x
82 AES2 0x0 AES3 0x2  CARD 0}
ALSA lib pcm_usb_stream.c:486:(_snd_pcm_usb_stream_open) Invalid type for card
ALSA lib pcm_usb_stream.c:486:(_snd_pcm_usb_stream_open) Invalid type for card
listening...
```

Fig.7. Virtual Assistant is listening for commands from the user

The Virtual Assistant replies to the user via voices command and receives commands from the user via voice. But user spoken preview only was designed to print on the terminal to monitor whether the Virtual Assistant could synthesize user voices correctly or not. All the proposed project functions are shown in Fig. 8-16.



```
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm_oss.c:377:(_snd_pcm_oss_open) Unknown field port
ALSA lib pcm_oss.c:377:(_snd_pcm_oss_open) Unknown field port
ALSA lib pcm_a52.c:823:(_snd_pcm_a52_open) a52 is only for playback
ALSA lib confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.bcm28
35_headpho.pcm.iec958.0:CARD=0,AES0=6,AES1=130,AES2=0,AES3=2'
ALSA lib conf.c:4745:(_snd_config_evaluate) function snd_func_refer returned err
or: No such file or directory
ALSA lib conf.c:5233:(snd_config_expand) Evaluate error: No such file or directo
ry
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM iec958:{AES0 0x6 AES1 0x
82 AES2 0x0 AES3 0x2  CARD 0}
ALSA lib pcm_usb_stream.c:486:(_snd_pcm_usb_stream_open) Invalid type for card
ALSA lib pcm_usb_stream.c:486:(_snd_pcm_usb_stream_open) Invalid type for card
listening...
recognizing...
Jarvis what is the time
['jarvis', 'what', 'is', 'the', 'time']
listening...
recognizing...
Jarvis what is the date
['jarvis', 'what', 'is', 'the', 'date']
```

Fig.8. The Virtual Assistant recognized that the user asked for the current "time and date" and replied to the output to the user via voice output

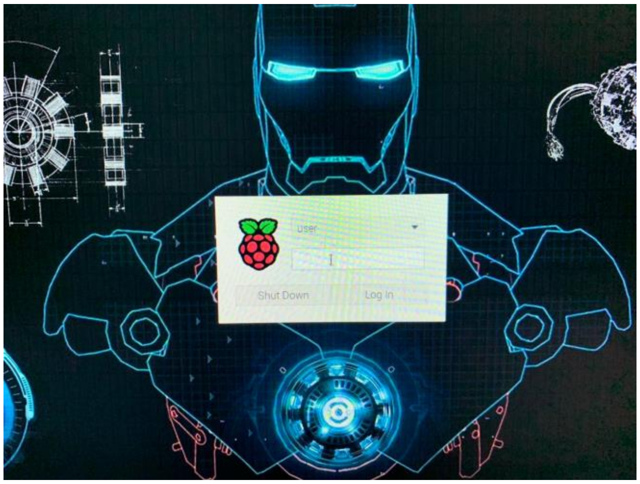
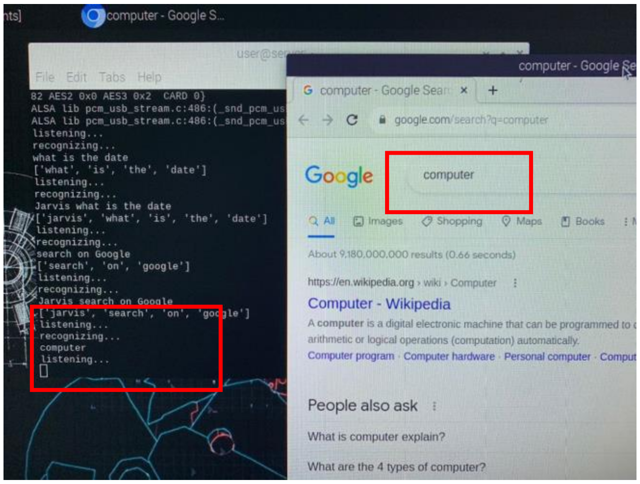


Fig.9. The user requested system "logout", and the Virtual Assistant logged out of the system



```
82 AES2 0x0 AES3 0x2  CARD 0}
ALSA lib pcm_usb_stream.c:486:(_snd_pcm_us
ALSA lib pcm_usb_stream.c:486:(_snd_pcm_us
listening...
recognizing...
what is the date
['what', 'is', 'the', 'date']
listening...
recognizing...
Jarvis what is the date
['jarvis', 'what', 'is', 'the', 'date']
listening...
recognizing...
search on Google
['search', 'on', 'google']
listening...
recognizing...
Jarvis search on Google
['jarvis', 'search', 'on', 'google']
listening...
recognizing...
computer
listening...
```

About 9,180,000,000 results (0.66 seconds)

https://en.wikipedia.org › wiki › Computer

Computer - Wikipedia

A computer is a digital electronic machine that can be programmed to c arithmetic or logical operations (computation) automatically.

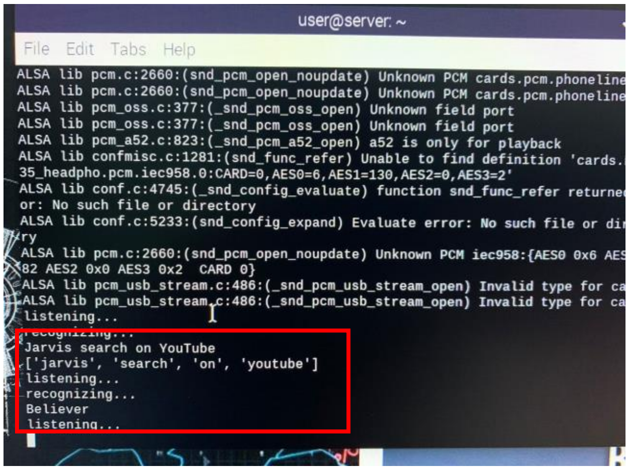
Computer program · Computer hardware · Personal computer · Comput

People also ask

What is computer explain?

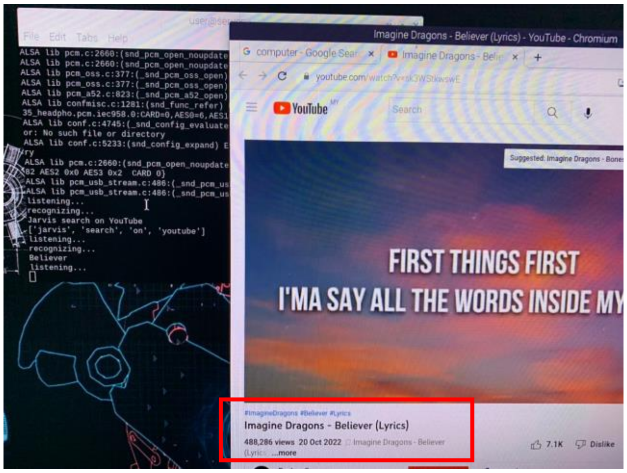
What are the 4 types of computer?

Fig. 10. The user given the command to search for “computer”. The Virtual Assistant searched computers on google as the user requested.



```
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.phoneline
ALSA lib pcm_oss.c:377:(_snd_pcm_oss_open) Unknown field port
ALSA lib pcm_oss.c:377:(_snd_pcm_oss_open) Unknown field port
ALSA lib pcm_a52.c:823:(_snd_pcm_a52_open) a52 is only for playback
ALSA lib confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.
35_headpho.pcm.iec958.0:CARD=0,AES0=6,AES1=130,AES2=0,AES3=2'
ALSA lib conf.c:4745:(_snd_config_evaluate) function snd_func_refer returne
or: No such file or directory
ALSA lib conf.c:5233:(snd_config_expand) Evaluate error: No such file or di
ry
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM iec958:{AES0 0x6 AES
82 AES2 0x0 AES3 0x2  CARD 0}
ALSA lib pcm_usb_stream.c:486:(_snd_pcm_usb_stream_open) Invalid type for ca
ALSA lib pcm_usb_stream.c:486:(_snd_pcm_usb_stream_open) Invalid type for ca
listening...
recognizing...
Jarvis search on YouTube
['jarvis', 'search', 'on', 'youtube']
listening...
recognizing...
Believer
listening...
```

Fig. 11. The user has requested the Virtual Assistant to play a song on YouTube.



```
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate
ALSA lib pcm_oss.c:377:(_snd_pcm_oss_open)
ALSA lib pcm_oss.c:377:(_snd_pcm_oss_open)
ALSA lib pcm_a52.c:823:(_snd_pcm_a52_open)
ALSA lib confmisc.c:1281:(snd_func_refer)
35_headpho.pcm.iec958.0:CARD=0,AES0=6,AES1
ALSA lib conf.c:4745:(_snd_config_evaluate
or: No such file or directory
ALSA lib conf.c:5233:(snd_config_expand) E
ry
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate
82 AES2 0x0 AES3 0x2  CARD 0}
ALSA lib pcm_usb_stream.c:486:(_snd_pcm_us
ALSA lib pcm_usb_stream.c:486:(_snd_pcm_us
listening...
recognizing...
Jarvis search on YouTube
['jarvis', 'search', 'on', 'youtube']
listening...
recognizing...
Believer
listening...
```

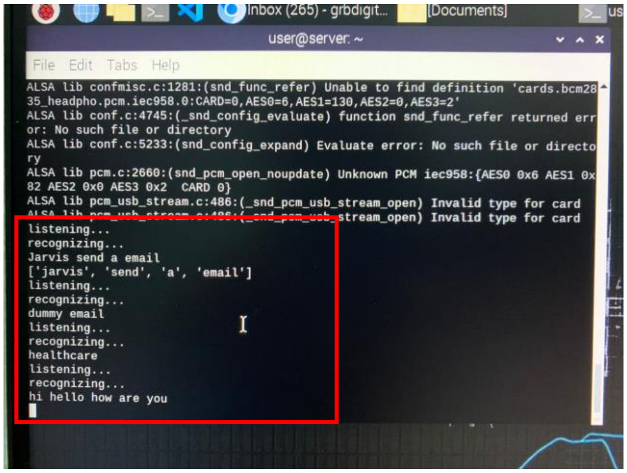
File Edit Tabs Help
Imagine Dragons - Believer (Lyrics) - YouTube - Chromium
computer - Google Sear ×
Imagine Dragons - Beli ×
youtube.com/watch?v=sk3WStkwswE
YouTube MY
Search
Suggested: Imagine Dragons - Bones
FIRST THINGS FIRST
I'MA SAY ALL THE WORDS INSIDE MY
7.1K
Dislike

#ImagineDragons #Believer #Lyrics

Imagine Dragons - Believer (Lyrics)

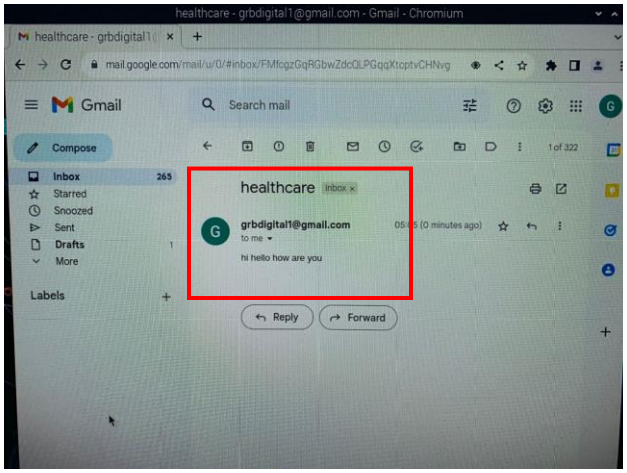
488,286 views 20 Oct 2022 Imagine Dragons - Believer (Lyrics) ...more

Fig. 12. The Virtual Assistant opened YouTube through the web browser and played the requested song



```
ALSA lib confmisc.c:1281:(snd_func_refer) Unable to find definition 'cards.bcm28
35_headpho.pcm.iec958.0:CARD=0,AES0=6,AES1=130,AES2=0,AES3=2'
ALSA lib conf.c:4745:(_snd_config_evaluate) function snd_func_refer returned err
or: No such file or directory
ALSA lib conf.c:5233:(snd_config_expand) Evaluate error: No such file or directo
ry
ALSA lib pcm.c:2660:(snd_pcm_open_noupdate) Unknown PCM iec958:{AES0 0x6 AES1 0x
82 AES2 0x0 AES3 0x2  CARD 0}
ALSA lib pcm_usb_stream.c:486:(_snd_pcm_usb_stream_open) Invalid type for card
ALSA lib pcm_usb_stream.c:486:(_snd_pcm_usb_stream_open) Invalid type for card
listening...
recognizing...
Jarvis send a email
['jarvis', 'send', 'a', 'email']
listening...
recognizing...
dummy email
listening...
recognizing...
healthcare
listening...
recognizing...
hi hello how are you
```

Fig. 13 The user instructed the Virtual Assistant to send an email. The user instructed Virtual Assistant to send it to a "dummy email, " which is already saved in the source code. The Virtual Assistant was asked back to whom it should send, what the subject is and the message body



healthcare - grbdigital1@gmail.com - Gmail - Chromium
healthcare - grbdigital1 ×
mail.google.com/mail/u/0/#inbox/FMfcgzGqRGbwZdcQLPGqqXtcptvCHNvg
Gmail
Search mail
Compose
Inbox 265
Starred
Snoozed
Sent
Drafts 1
More
Labels
1 of 322
Reply
Forward

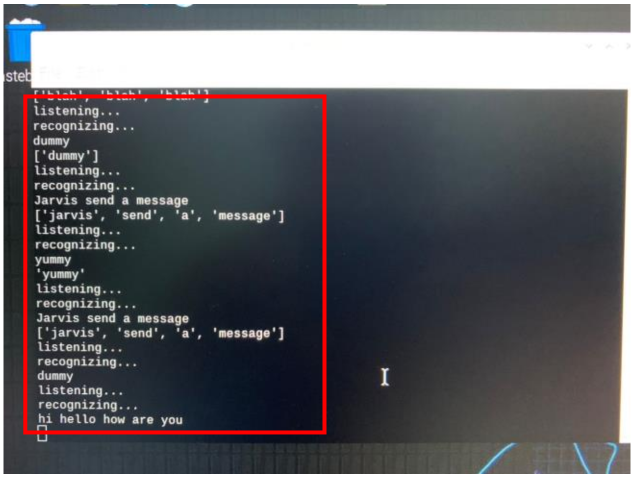
healthcare Inbox ×

grbdigital1@gmail.com 05:05 (0 minutes ago)

to me

hi hello how are you

Fig. 14. The Virtual Assistant has sent the email as per the user's commanded



```
['blah', 'blah', 'blah']
listening...
recognizing...
dummy
['dummy']
listening...
recognizing...
Jarvis send a message
['jarvis', 'send', 'a', 'message']
listening...
recognizing...
yummy
'yummy'
listening...
recognizing...
Jarvis send a message
['jarvis', 'send', 'a', 'message']
listening...
recognizing...
dummy
listening...
recognizing...
hi hello how are you
```

Fig. 15. The Virtual Assistant was given a task to send a message in WhatsApp, to be sent to a "dummy" which is a saved contact in source code, “hi hello how are you is the message”

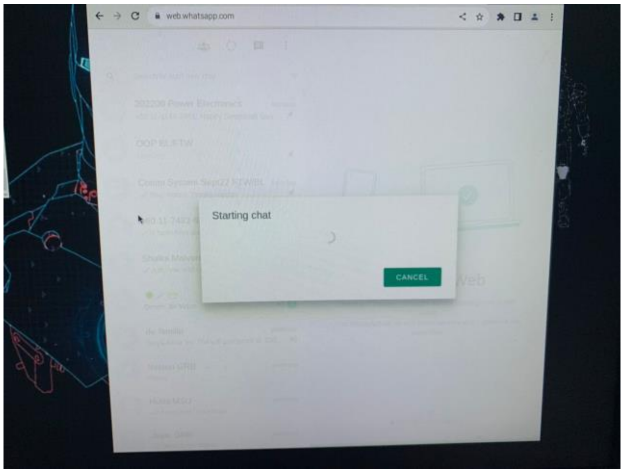


Fig. 16. The Virtual Assistant opens the chat to send the message.

6. Discussion

This article presents comprehensive documentation of the Artificial Intelligence virtual assistant. Using a motion sensor, the virtual assistant in this project is more concerned with protecting the user's privacy and security. Because the virtual assistant system requires human interaction, it must be awake before receiving a command from the user. A feature that other virtual assistant programs or devices, such as Apple's Siri, Yandex's Alice, Samsung's Bixby, Microsoft's Cortana, and Amazon's Alexa, lack.

Although Amazon's Alexa has a capability similar to the project's, which allows it to detect human presence, Alexa is not a true motion detector because most motion detectors employ ultrasonic sound waves, which don't rely on lights to detect movement. Alexa, on the other hand, uses a camera and a computer processor to determine whether someone is present or not. Thus the Echo Show must first detect your presence before Alexa can. The device's built-in camera, microphone, voice recognition, and persistent connectivity have sparked privacy worries because they make it simple to gather audio data from its surroundings.

Alexa was the only virtual assistant that shared the same idea as the proposed project. Still, it turns out that Alexa's implementation needs more privacy because using a camera to trigger Alexa's activation by human presence is not a very effective privacy and security enhancement. This is because anyone could hack into the computer and gain full access, including access to the camera, which could result in the hacker having an eye on your place.

7. Conclusion

In conclusion, several automated services utilizing this virtual assistant with a one-line command are developed. It simplifies most of the user's chores, such as listening to music, sending text messages, reading articles, and other daily routines. Moreover, in this project, a complete server assistant that is intelligent to ease our life has been made.

Acknowledgement

The authors fully acknowledged the Ministry of Higher Education (MOHE) and MSU Shah Alam for the support, which makes this important research viable and effective.

References

Abougarair, A. J., Aburakhis, M. K. I., & Zaroug, M. O. (2022). Design and implementation of smart voice assistant and recognizing academic words. *Int Rob Auto J*, *8*(1), 27-32.

Dekate, A., Kulkarni, C., & Killedar, R. (2016). Study of voice controlled personal assistant device. *International Journal of Computer Trends and Technology (IJCTT)*, *42*(1), 42-46.

O'Shaughnessy, D. (2003). Interacting with computers by voice: automatic speech recognition and synthesis. *Proceedings of the IEEE*, *91*(9), 1272-1305.