

**SUSHGANGA POLYTECHNIC, WANI**

**PROJECT REPORT**

**ON**

**JARVIS**

**SUBMITTED BY**

**MUDDASSIR SHEIKH, ADIB HAQUE, MIHIR JHA,  
PRATHAMESH URKUDE, PRANJALI GIRATKAR**

**GUIDED BY**

**PROF. DIKSHA HIWARE**

**DEPARTMENT OF COMPUTER TECHNOLOGY**

**DIPLOMA IN ENGINEERING**

**SUSHGANGA POLYTECHNIC, WANI**

**YEAR 2023-24**

**SUSHGANGA POLYTECHNIC, WANI**

**PROJECT REPORT**

**ON**



**SUBMITTED BY**

**MUDDASSIR SHEIKH, MIHIR JHA, ADIB SIDDIQUI,  
PRATHAMESH URKUDE, PRANJALI GIRATKAR**

**GUIDED BY**

**PROF. DIKSHA HIWARE**

**DEPARTMENT OF COMPUTER TECHNOLOGY**

**DIPLOMA IN ENGINEERING**

**SUSHGANGA POLYTECHNIC, WANI**

**YEAR 2023-24**

**DEPARTMENT OF COMPUTER TECHNOLOGY**  
**SUSHGANGA POLYTECHNIC, WANI**

2023-24

**Certificate**



**THIS IS TO CERTIFY THAT PROJECT ON:**  
**JUST A RATHER VERY INTELLIGENT**  
**SYSTEM [J.A.R.V.I.S]**

**SUBMITTED BY**

**MUDDASSIR SHEIKH, ADIB HAQUE, MIHIR JHA,**  
**PRATHAMESH URKUDE, PRANJALI GIRATKAR**

**The students of final year DIPLOMA IN COMPUTER TECHNOLOGY of institute has successfully completed the project work and have submitted satisfactory report as guided by Prof. DIKSHA HIWARE on topic as per the syllabus prescribed by MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION, MUMBAI during academic year 2023-24.**

**prof. Shamali Kadu**

**H.O.D**

**C.M. dept**

**Prof Diksha Hiware**

**guide**

**External**

**Prof. Pushpa Rani**

**principal**

**S.G.P. Wani**

**SUSHGANGA POLYTECHNIC, WANI**

## **SUBMISSION**

**WE,**

**Students of final year of course computer technology humbly submit that we have completed from time to time project works or described in this report by our own skills and study between period 2023-24 as per instructions and guidance of our guide.**

**And that, following students were associated with us in this work however quantum of our contribution has been approved by our lecturer. We conclude that we have not copied the report or its any appreciable part from other literature in contravention of academic ethics.**

**Date: \_\_/\_\_/2024**

**Signature of students with respective names:**

# **Acknowledgement**

- ❖ We find pleasure to introduce this project in front of you. We would like to place on record our indebtedness to various individual whose co-operation made it possible to complete this work.
- ❖ We pay thanks to Prof. Shamali Kadu, Head of the Department of Computer technology, Sushganga polytechnic, Wani; who took interest in our project and provided us all the required components to make the project and also for encouraging spirit among us.
- ❖ We pay special thanks to our guide, prof. Diksha Hiware who took interest in project and encouraged us.
- ❖ We are very grateful to our parents by whom we got moral support and precious encouragement.

**---STUDENTS:**

- 1. Muddassir Sheikh**
- 2. Mihir Kumar Jha**
- 3. Adib Siddiqui**
- 4. Prathamesh Urkude**
- 5. Pranjali Giratkar**

# **CONTENTS**

## Information

- 1.Objective
- 2.Introduction of preferred IDE
- 3.Physical technical and software support
- 4.Stages of development
- 5.Algorithm
- 6.Part I: Main program
- 7.Part II: Hotword detection
- 8.Part III: Virtual mouse
- 9.Part IV: Face recognition
10. Part V: Lock Wakeup
11. Part VI: final version
12. Outputs
13. Further developments
14. Bibliography

## **INFORMATION**

In the year 2008, **Marvel Studios** launched its first movie **IRON-MAN** which as a kid fascinated as well as imperated us about the existence of such technology JARVIS which was a very smart Artificial Intelligence personal virtual assistant of our protagonist Tony Stark. In many ways this influenced the minds of enthusiasts.

Similarly in todays, world we see many personal digital assistants based on artificial intelligence like Alexa by Amazon Inc. and siri by Apple Inc. Being an emerging computer engineer we should be aware of workings of such applications and tools.

For this we developed a basic personal assistant which can do some basic as well as some advanced things. We named it as JARVIS to sound familiar to the movie's technology.

# **CHAPTER: 1**

## **Objective**



# OBJECTIVES

As we know we are making a virtual assistant so we will need at least of some basic capabilities of systems commercially available. Some requirements are:

1. Program should greet and introduce to us about itself.
2. Program should take our query by listening, not by keyboard input.
3. After taking command as query it should be converted into text.
4. It should tell time.
5. It should open apps like google, youtube.
6. It should access Wikipedia
7. It should do facial recognition of owner
8. It should wake up on calling its name
9. It should access our pre developed virtual mouse

# **CHAPTER: 2**

## **Introduction to preferred IDE**

# **PYCHARM**

PyCharm is an integrated development environment (IDE) Used for programming in python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems and supports web development with Django. PyCharm is developed by Czech company JetBrains.

It is cross-platform, working on Microsoft Windows, macOS and Linux. PyCharm has a professional Edition, released under a proprietary License and a Community Edition released under the Apache License. We are using the community edition in our project as it has vibrant color schemes and brilliant font types. It also provides in-built system for installing external libraries and supports multiple interpreters.

# **CHAPTER: 3**

**Physical, technical and libraries  
required**

## **Libraries required:**

- **Cv2**
- **Datetime**
- **Face\_recognition**
- **Mediapipe**
- **os**
- **Ptttsx3**
- **PyAutoGui**
- **Speech\_recognition**
- **Subprocess**
- **Sys**
- **Webbrowser**
- **Wikipedia**

## **Physical support:**

- **Microphone**
- **Webcam**
- **Speaker**

## **Technical support:**

- **Python interpreter sync done**
- **All required permissions should be given like microphone access, camera access, internet access, etc.**

# **CHAPTER: 4**

## **Stages of development**

## **WE DEVELOPED THIS PROGRAM IN FIVE VERSIONS TILL NOW:**

### **v 1.0.0: Only main program:**

**In our first ever version of JARVIS, we only gave some basic functionalities to it like only wishing, taking queries and opening google, youtube or searching Wikipedia.**

### **v 1.1.0 Hotword detection:**

**In our mark 2 version, we gave a vocal stimulator where if a code “wakeup JARVIS” to activate our assistant.**

### **v 1.2.0 Addition of Virtual mouse:**

**We have added a virtual mouse where the webcam detects positions of our index finger and thumb in air and it acts as mouse when glided through air.**



### **v 1.3.0 Face recognition:**

**At this point of development, we added a face recognition program which can recognize only the face of owner i.e. Muddassir Sheikh.**

### **v 1.4.0 Lock Wakeup:**

**Here, we made slight advancement in hotword detection where in hearing wakeup call, firstly it will authenticate the person who made the call, if it is the owner so the further process will be done.**

# **CHAPTER: 5**

## **Algorithm**

# **ALGORITHM:**

**Since we are having three different files here, we are having three sets of algorithm:**

## **Main File:**

1. Our program will wish us and ask for our query.
2. We will speak our query where, program will convert speech to text.
3. If our query is recognized, further proceedings will be done else, it will ask to re- ask query.
4. Procedure of query will be done and after it will be processed for output.
5. If the query is open google; it will open [www.google.com](http://www.google.com)
6. Elif; the query is open youtube, [www.youtube.com](http://www.youtube.com) will be opened.

7. Elif, the query of asking the time, time will be fetched with datetime module and shown.
8. Elif; Wikipedia occurs in the statement, the corresponding statement will be fetched from Wikipedia.com.
9. Elif; the virtual mouse is to be used, it will be accessed from pre-developed file.

## **Virtual mouse:**

1. Due to cv2 [openCV] module, it is easy to recognize the person's hand where index finger and thumb with the help of webcam.
2. Mediapipe module will create drawing tools like circle discs on the fingers and the thumb. It makes it easier to track.
3. PyAutoGui module helps to track the trails of waves made by hand as well as clicks can be made by touching tips of thumb and finger.

## **Hotword detection:**

1. Microphone will always be active and ready to take inputs.
2. On hearing “wake up JARVIS” it will be triggered to verify and authenticate the face of person sitting in front of webcam.
3. Comparison of the live face with pre entered image of master user.
4. If authenticated successfully, main file will be opened.
5. Else nothing will happen.

# **CHAPTER: 6**

## **Part I:**

### **Main program**

# MAIN PROGRAM:

Our main program consists of the main abilities like taking commands, wishing the user, showing outputs, converting text to speech or vice versa and it is also able to perform tasks like opening applications.

## Code: File name: JARVIS.py

```
import subprocess
import pyttsx3
import speech_recognition as sr
import datetime
import wikipedia
import webbrowser
import sys

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

def speak(audio):
    engine.say(audio)
    engine.runAndWait()
def exit_program():
    sys.exit(0)

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 JARVIS created by students of Computer branch, final year Sooshganga polytechnic. How may I help you?")
    print("I am JARVIS created by students of Computer branch, final year Sushganga polytechnic. How may I help you?")

def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
```

## J.A.R.V.I.S [ JUST A RATHER VERY INTELLEAGENT SYSTEM ]

```
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 in wikipedia...")
            query = query.replace("Wikipedia", "")
            results = wikipedia.summary(query, sentences=2)
            speak("According to Wikipedia")
            print(results)
            speak(results)
            exit_program()

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

        elif 'how are you jarvis' in query:
            speak("I am fine sir. Thank you.")

        elif 'open google' in query:
            speak("opening google")
            webbrowser.open("google chrome.com")
            exit_program()

        elif 'jarvis sleep' in query:
            print("exiting program...")
            speak("exiting program")
            subprocess.run(["python", "hotword detection+face recognition.py"])
            exit_program()

        elif 'tell me the time' in query:
            strTime = datetime.datetime.now().strftime("%H:%M:%S")
            speak(f"Sir, the time is {strTime}")
            print(f"sir, the time is {strTime}")
            speak("sir, I do anything else for you?")

        elif 'virtual mouse' in query:
            subprocess.run(["python", "virtual mouse mark 1.py"])
```



# **CHAPTER: 7**

## **Part II:**

### **Hotword detection**

# **HOTWORD DETECTION:**

This segment of our program is used to wake up our main program verbally, if our main program is closed, it will be triggered only by our words:

“wake up JARVIS”

## **Code: File name: Hotword detection.py**

```
import speech_recognition as sr
import subprocess
import sys

def exitprg():
    sys.exit(0)
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:
        return "None"
    return query

while True:
    wake_up = takeCommand()

    if 'wake up Jarvis' in wake_up:
        subprocess.run(["python", "jarvis.py"])
        exitprg()
    else:
        print("nothing....")
```

**CHAPTER: 8**

**Part III:**

**Virtual Mouse**

# **VIRTUAL MOUSE:**

This is the segment where we are enabling the technique of air mouse. Here, our index finger and thumb will act as a mouse. All the movements which are made by them are captured by the webcam and are replicated by cursor. When the distance between vertical dimensions i.e. y-axis of the tips of index finger and thumb is lesser than 20 pts; a click is made.

## **Code: File name: Virtual mouse mark 1**

**.py**

```
import cv2
import mediapipe as mp
import pyautogui
cap = cv2.VideoCapture(0)
hand_detector = mp.solutions.hands.Hands()
drawing_utils = mp.solutions.drawing_utils
screen_width, screen_height = pyautogui.size()
index_y = 0
while True:
    _, frame = cap.read()
    frame = cv2.flip(frame, 1)
    frame_height, frame_width, _ = frame.shape
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    output = hand_detector.process(rgb_frame)
    hands = output.multi_hand_landmarks
    if hands:
        for hand in hands:
            drawing_utils.draw_landmarks(frame, hand)
            landmarks = hand.landmark
            for id, landmark in enumerate(landmarks):
                x = int(landmark.x*frame_width)
                y = int(landmark.y*frame_height)
                print(x, y)
```

## J.A.R.V.I.S [ JUST A RATHER VERY INTELLEAGENT SYSTEM ]

```
if id == 8:
    cv2.circle(img=frame, center=(x,y), radius=10, color=(0, 255,
255))

    index_x = screen_width/frame_width*x
    index_y = screen_height/frame_height*y
    pyautogui.moveTo(index_x, index_y)
if id == 4:
    cv2.circle(img=frame, center=(x,y), radius=10, color=(0, 255,
255))

    thumb_x = screen_width/frame_width*x
    thumb_y = screen_height/frame_height*y
    print('outside',abs(index_y - thumb_y))
    if abs(index_y - thumb_y) < 20:
        pyautogui.click()
        pyautogui.sleep(1)
        print('click')

cv2.imshow('Virtual mouse', frame)
cv2.waitKey(1)
```

# **CHAPTER: 9**

## **Part IV:**

### **Face recognition**

# **FACE RECOGNITION:**

This particular part defines about a machine learning technique called as face recognition. Our system already has the face data of the owner stored in it. Here the data of Muddassir Sheikh is stored. Now starting upon the action comparison of real-time face will be done and if authentication done and stands true, it will print “Authenticated successfully...”

## **Code: File name: Face recognition.py**

```
import cv2
import face_recognition

video = cv2.VideoCapture(0)

face = face_recognition.load_image_file("md.jpg")
faceencoding = face_recognition.face_encodings(face)[0]

face_encodings_list = [faceencoding]

face_encodings = []
s = True
face_coordinates = []

while True:
    _, frame = video.read()
    resized_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
    resized_frame_rgb = resized_frame[:, :, :-1]

    if s:
        face_coordinates = face_recognition.face_locations(resized_frame_rgb)
        face_encodings = face_recognition.face_encodings(resized_frame_rgb,
        face_coordinates)
        for faces in face_encodings:
            matches = face_recognition.compare_faces(face_encodings_list, faces)
            if matches[0] == True:
                print("Authenticated successfully...")
                video.release()
                cv2.destroyAllWindows()
```

# **CHAPTER: 10**

## **Part V:**

### **Lock Wakeup**



## **LOCK WAKEUP:**

Here, we combined the abilities of the two segments, Hotword detection and face recognition. It makes a system where, on triggering the system firstly the user will get authenticated and if the authorized user is there; then only the system will be activated. Authorization of the user will be taken place with webcam placed on the device and where the user will be compared with the pre-analyzed image of an authorized user.

# Code: File name: hotword detection+face recognition.py

```
import speech_recognition as sr
import subprocess
import sys
import cv2
import face_recognition

def exitprg():
    sys.exit(0)
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:
        return "None"
    return query

while True:
    wake_up = takeCommand()

    if 'wake up Jarvis' in wake_up:
        video = cv2.VideoCapture(0)

        face = face_recognition.load_image_file("md.jpg")
        faceencoding = face_recognition.face_encodings(face)[0]

        face_encodings_list = [faceencoding]

        face_encodings = []
        s = True
        face_coordinates = []

        while True:
            _, frame = video.read()
            resized_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
            resized_frame_rgb = resized_frame[:, :, :-1]

            if s:
                face_coordinates =
                face_recognition.face_locations(resized_frame_rgb)
                face_encodings = face_recognition.face_encodings(resized_frame_rgb,
                face_coordinates)
                for faces in face_encodings:
                    matches = face_recognition.compare_faces(face_encodings_list,
                    faces)

                    if matches[0] == True:
                        print("Authenticated successfully...")
                        video.release()
                        cv2.destroyAllWindows()
                        subprocess.run(["python", "jarvis.py"])
                        exitprg()
                    else:
                        print("nothing....")
                        cv2.imshow('face scan', frame)

                if cv2.waitKey(1) & 0xFF == ord('q'):
                    break
            video.release()
            cv2.destroyAllWindows()
```

# **CHAPTER: 11**

## **Part VI:**

### **Final version**

## **FINAL VERSION:**

We connected every file described above and we are using them according to the stages of requirement. At last, we got only three files namely:

1. Hotword detection+ face recognition.py
2. JARVIS.py
3. Virtual mouse mark 1.py

And one file for image comparison i.e. “md.jpg”

# Code 1:

```

import speech_recognition as sr
import subprocess
import sys
import cv2
import face_recognition

def exitprg():
    sys.exit(0)
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:
        return "None"
    return query

while True:
    wake_up = takeCommand()

    if 'wake up Jarvis' in wake_up:
        video = cv2.VideoCapture(0)

        face = face_recognition.load_image_file("md.jpg")
        faceencoding = face_recognition.face_encodings(face)[0]

        face_encodings_list = [faceencoding]

        face_encodings = []
        s = True
        face_coordinates = []

        while True:
            _, frame = video.read()
            resized_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
            resized_frame_rgb = resized_frame[:, :, :-1]

            if s:
                face_coordinates =
face_recognition.face_locations(resized_frame_rgb)
                face_encodings = face_recognition.face_encodings(resized_frame_rgb,
face_coordinates)
                for faces in face_encodings:
                    matches = face_recognition.compare_faces(face_encodings_list,
faces)

                    if matches[0] == True:
                        print("Authenticated successfully...")
                        video.release()
                        cv2.destroyAllWindows()
                        subprocess.run(["python", "jarvis.py"])
                        exitprg()
                    else:
                        print("nothing....")
            cv2.imshow('face scan', frame)

            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
        video.release()
        cv2.destroyAllWindows()

```

## Code 2:

```

import subprocess
import pyttsx3
import speech_recognition as sr
import datetime
import wikipedia
import webbrowser
import sys

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

def speak(audio):
    engine.say(audio)
    engine.runAndWait()
def exit_program():
    sys.exit(0)

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 JARVIS created by students of Computer branch, final year Sooshganga polytechnic. How may I help you?")
    print("I am JARVIS created by students of Computer branch, final year Sushganga polytechnic. 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

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

        query = takeCommand().lower()

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

```

## J.A.R.V.I.S [ JUST A RATHER VERY INTELEAGENT SYSTEM ]

```
elif 'open youtube' in query:
    speak("opening youtube")
    webbrowser.open("youtube.com")
    exit_program()

elif 'how are you jarvis' in query:
    speak("I am fine sir. Thank you.")

elif 'open google' in query:
    speak("opening google")
    webbrowser.open("google chrome.com")
    exit_program()

elif 'jarvis sleep' in query:
    print("exiting program...")
    speak("exiting program")
    subprocess.run(["python", "hotword detection+face recognition.py"])
    exit_program()

elif 'tell me the time' in query:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    speak(f"Sir, the time is {strTime}")
    print(f"sir, the time is {strTime}")
    speak("sir, I do anything else for you?")

elif 'virtual mouse' in query:
    subprocess.run(["python", "virtual mouse mark 1.py"])
```

# Code 3:

```

import cv2
import mediapipe as mp
import pyautogui
cap = cv2.VideoCapture(0)
hand_detector = mp.solutions.hands.Hands()
drawing_utils = mp.solutions.drawing_utils
screen_width, screen_height = pyautogui.size()
index_y = 0
while True:
    _, frame = cap.read()
    frame = cv2.flip(frame, 1)
    frame_height, frame_width, _ = frame.shape
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    output = hand_detector.process(rgb_frame)
    hands = output.multi_hand_landmarks
    if hands:
        for hand in hands:
            drawing_utils.draw_landmarks(frame, hand)
            landmarks = hand.landmark
            for id, landmark in enumerate(landmarks):
                x = int(landmark.x*frame_width)
                y = int(landmark.y*frame_height)
                print(x, y)
                if id == 8:
                    cv2.circle(img=frame, center=(x,y), radius=10, color=(0, 255,
255))

                    index_x = screen_width/frame_width*x
                    index_y = screen_height/frame_height*y
                    pyautogui.moveTo(index_x, index_y)
                if id == 4:
                    cv2.circle(img=frame, center=(x,y), radius=10, color=(0, 255,
255))

                    thumb_x = screen_width/frame_width*x
                    thumb_y = screen_height/frame_height*y
                    print('outside',abs(index_y - thumb_y))
                    if abs(index_y - thumb_y) < 20:
                        pyautogui.click()
                        pyautogui.sleep(1)
                        print('click')

            cv2.imshow('Virtual mouse', frame)
            cv2.waitKey(1)

```



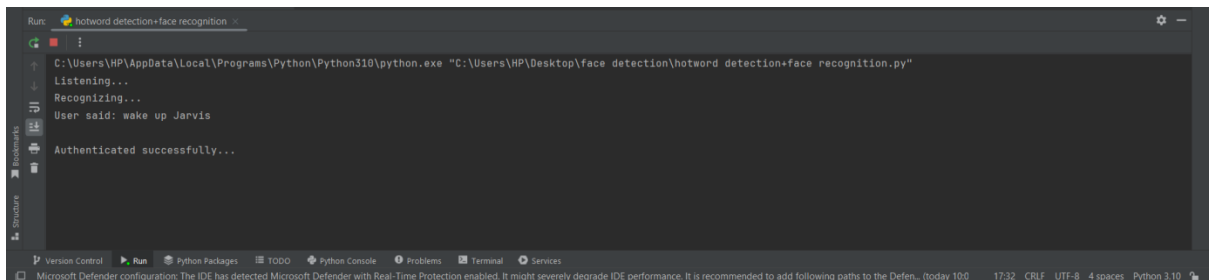
# **CHAPTER: 12**

## **OUTPUTS**

# OUTPUTS:

**The terminal view of the results of each query are shown below:**

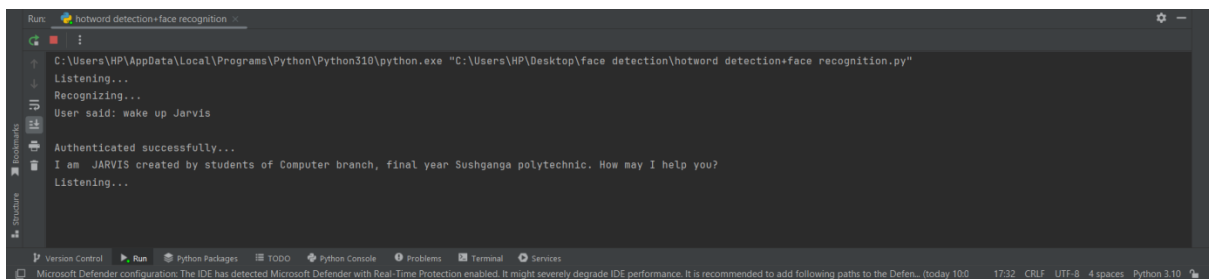
## **1. Waking up:**



```
Run: hotword detection+face recognition
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\HP\Desktop\face detection\hotword detection+face recognition.py"
Listening...
Recognizing...
User said: wake up Jarvis

Authenticated successfully...
```

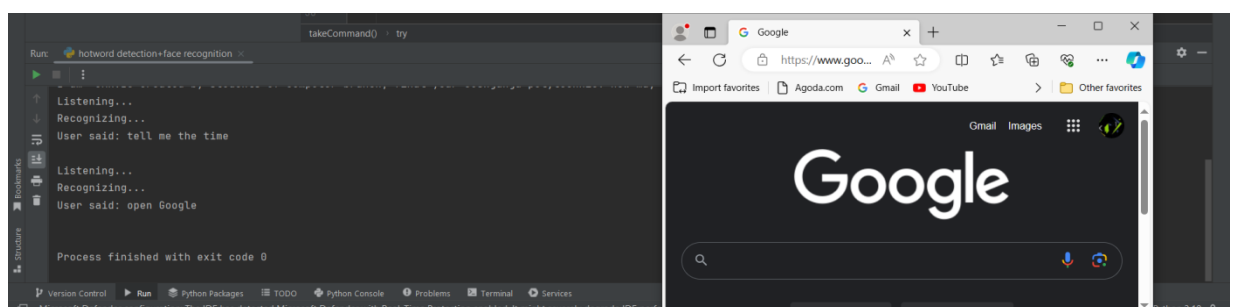
## **2. Main program has awaken**



```
Run: hotword detection+face recognition
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\HP\Desktop\face detection\hotword detection+face recognition.py"
Listening...
Recognizing...
User said: wake up Jarvis

Authenticated successfully...
I am JARVIS created by students of Computer branch, final year Sushganga polytechnic. How may I help you?
Listening...
```

## **3. Opening of google**

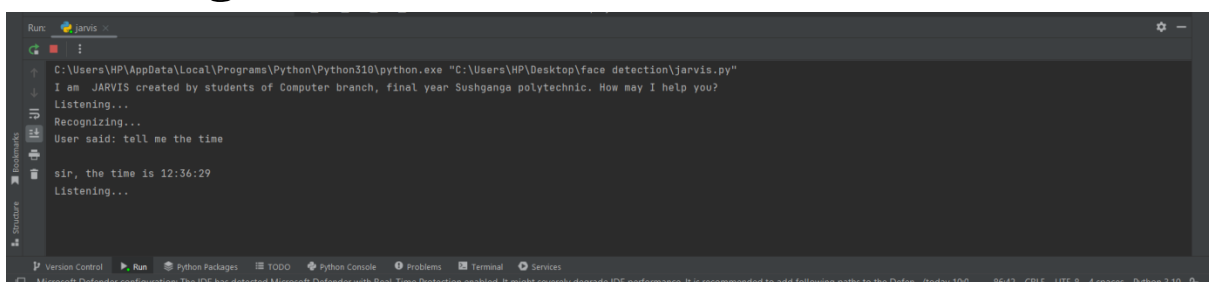


```
Run: hotword detection+face recognition
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\HP\Desktop\face detection\hotword detection+face recognition.py"
Listening...
Recognizing...
User said: tell me the time

Listening...
Recognizing...
User said: open Google

Process finished with exit code 0
```

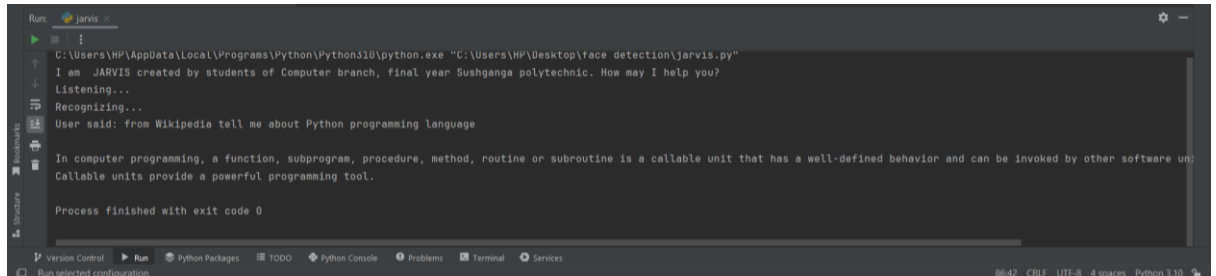
## **4. Showing the time**



```
Run: jarvis
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\HP\Desktop\face detection\jarvis.py"
I am JARVIS created by students of Computer branch, final year Sushganga polytechnic. How may I help you?
Listening...
Recognizing...
User said: tell me the time

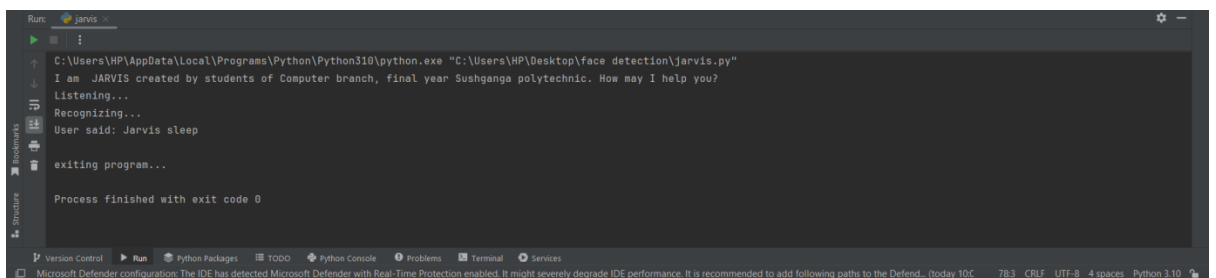
sir, the time is 12:36:29
Listening...
```

## 5. Searching from Wikipedia



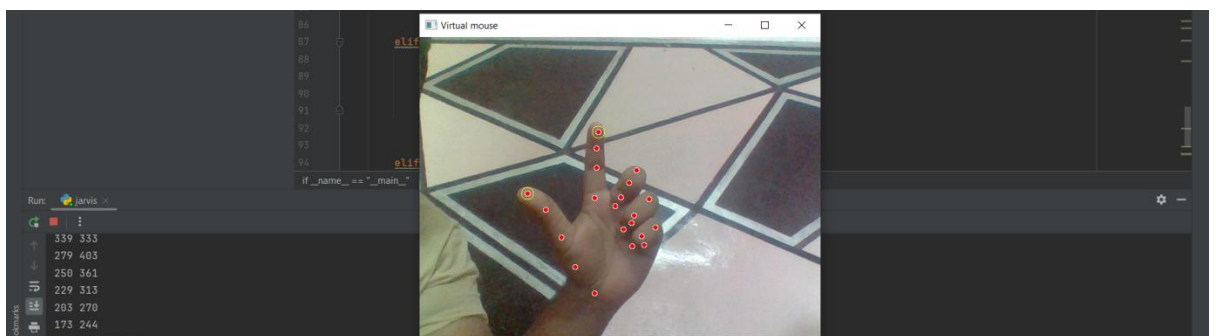
```
Run: jarvis x
1
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\HP\Desktop\face detection\jarvis.py"
I am JARVIS created by students of Computer branch, final year Sushganga polytechnic. How may I help you?
Listening...
Recognizing...
User said: from Wikipedia tell me about Python programming language
In computer programming, a function, subprogram, procedure, method, routine or subroutine is a callable unit that has a well-defined behavior and can be invoked by other software units. Callable units provide a powerful programming tool.
Process finished with exit code 0
```

## 6. Asking program to sleep



```
Run: jarvis x
1
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\HP\Desktop\face detection\jarvis.py"
I am JARVIS created by students of Computer branch, final year Sushganga polytechnic. How may I help you?
Listening...
Recognizing...
User said: Jarvis sleep
exiting program...
Process finished with exit code 0
```

## 7. Opening Virtual mouse



# **CHAPTER: 13**

## **FURTHER DEVELOPMENTS**

## **FURTHER DEVELOPMENTS:**

**We are wishing to perform some further developments to enhance the efficiency, versatility, functionality and ability of our project. Some of our future plans are:**

**1. Adding GUI:**

We want to add a sort of user interface to our project for providing seamless experience.

**2. Voice recognition:**

Just like face recognition, we will try to add a feature called voice recognition where the project will be smartly able to differentiate in between the voice of owner and that of other people.

**3. Run without a large computer:**

Installation on miniature computers like arduino nano, raspberry pi pico, arduino uno, etc. which will act as an independent system.

**4. Multiple machine learning abilities:**

We will try to add some machine learning abilities like turning on/off to electrical appliances and electronic devices.

**5. Multiple languages support:**

Our program is currently available in English language only, we will try to make it support to other languages like Hindi, Marathi, etc.

# **CHAPTER: 14**

## **Bibliography**

## **BIBLIOGRAPHY:**

**We took references, resources, help and research from some external sources which are:**

- [www.google.com](http://www.google.com)
- [www.youtube.com](http://www.youtube.com)
- [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
- [www.codewithharry.com](http://www.codewithharry.com)
- <https://github.com>