Computer Project Report: T.W.O.P.Y

Prepared by:
P.P.PAVEL
XII-C
19

TABLE OF CONTENTS

OBJECTIVE

- T.W.O.P.Y is a virtual assistant made from Python that automates Desktop experiences along with providing a Hands-Free way to perform tasks.

- This project is ever expanding based on how many features the user wants. T.W.O.P.Y also allows maximum personalization.

- In future Iterations we are working on improving speech recognition with the help of TensorFlow to minimize the errors.

PROBLEM DEFINITION

- Suppose your computer is across the room from where you are seated.
- To do a simple task like a search or a call someone up, you will need to get up and walk all the way to your PC.
- With our Project you just need to speak out very easy commands which will save you the hassle of walking for very minute tasks.

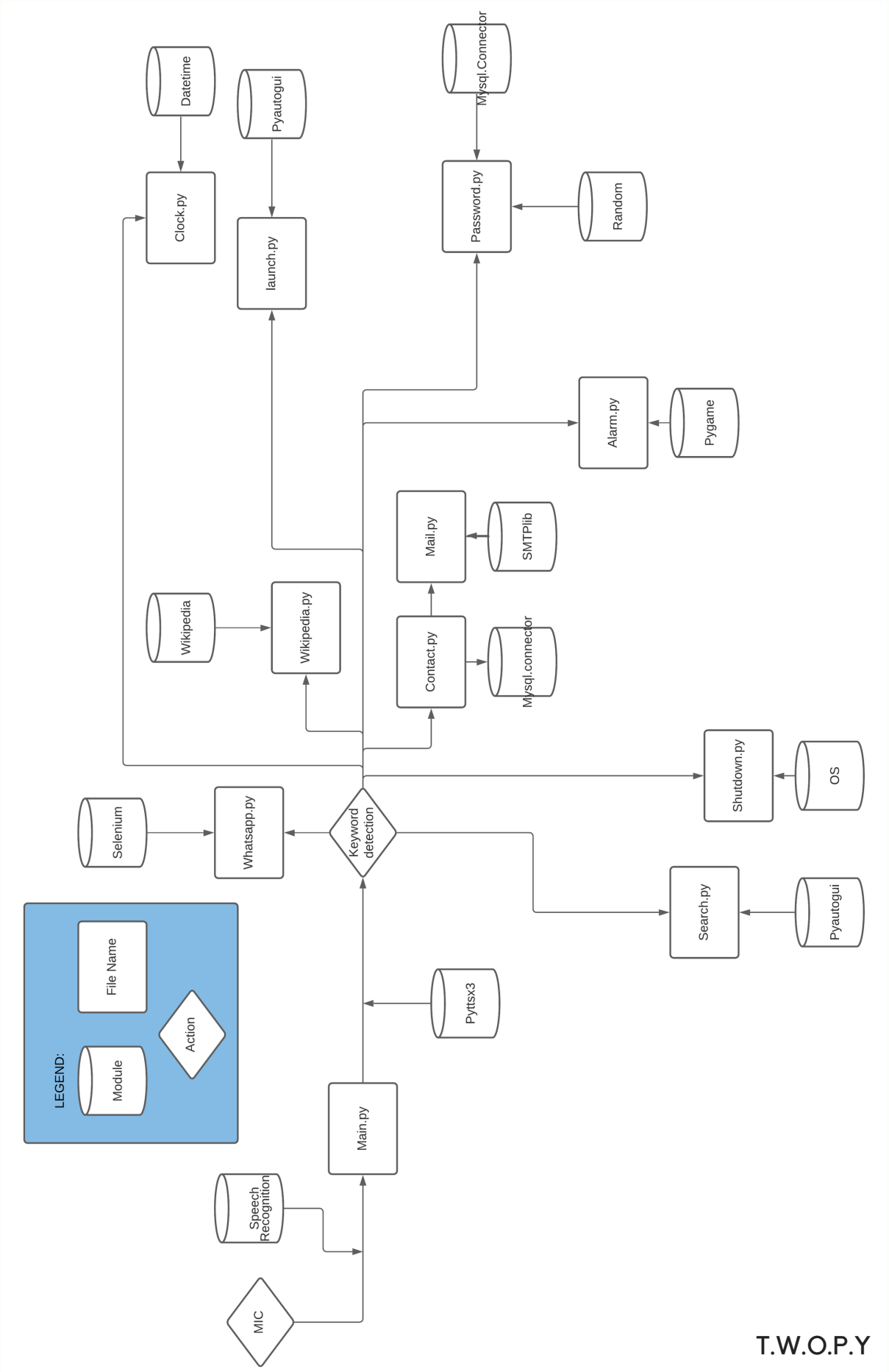
ANALYSIS:

• INPUT:

ALL T.W.O.P.Y INPUTS ARE IN THE FORM OF SPEECH AND ARE TRANSLATED BY THE SPEECH RECOGNITION MODULE INTO TEXT (DIGITAL DATA) AND INTERPRETED BY THE MACHINE.

• OUTPUT:

T.W.O.P.Y HAS SPEECH OUTPUTS. PYTTSX3 IS THE MODULE THAT HELPS US CONVERT TEXT TO SPEECH.


Datetime
Pyautogui
Mysql.Connector
Clock.py
launch.py
Password.py
Random
Alarm.py
Pygame
Mail.py
SMTPlib
Wikipedia
Wikipedia.py
Contact.py
Mysql.connector
Shutdown.py
OS
Selenium
Whatsapp.py
Keyword detection
Search.py
Pyautogui
Pyttsx3
Main.py
Speech Recognition
MIC
LEGEND:
File Name
Module
Action

DATABASES/MODULES USED:

- pyttsx3 : THIS IS A TEXT TO SPEECH CONVERSION LIBRARY.
- speech_recognition : THIS LIBRARY IS USED TO CONVERT TEXT TO SPEECH
- datetime : SINCE DATE IN PYTHON IS NOT A DATA TYPE, WE USE THE datetime MODULE TO WORK WITH DATES AS OBJECTS
- wikipedia : THIS IS USED TO ACCESS AND PARSE DATA FROM WIKIPEDIA. FUNCTIONS INCLUDE GETTING ARTICLE SUMMARIES, LINKS AND IMAGES FROM THE SAME.
- selenium : SELENIUM IS USED FOR BROWSER AUTOMATION AND INTERACTION FROM PYTHON.

- pynput : THIS MODULE IS USED TO CONTROL AND MONITOR INPUT DEVICES, NAMELY KEYBOARD AND MOUSE.

- pyautogui : THIS LETS OUR PYTHON CODE AUTOMATE INTERACTIONS WITH OTHER APPLICATIONS.

- mysql.connector : THIS ENABLES PYTHON PROGRAMS TO ACCESS MySQL DATABASES.

- tkinter : THIS LIBRARY IS A GUI FRAMEWORK THAT IS USED TO BUILD GUI BASED APPLICATIONS IN PYTHON.

- pyperclip : THIS IS A CROSS PLATFORM PYTHON MODULE FOR COPYING AND PASTING TEXT TO THE CLIPBOARD.

- random : THIS GENERATES RANDOM NUMBERS IN PYTHON. NUMBERS GENERATED ARE PSEUDO-RANDOM IN NATURE.

- pyaudio : THIS LIBRARY HELPS IN PLAYBACK AND RECORDING AUDIO WITHIN A PYTHON ENVIRONMENT.

- smtplib : THIS MODULE HELPS IN CREATING A SMPT CLIENT SESSION THAT CAN SEND MAILS TO ANY MACHINE.

- pygame : IT IS A SET OF PYTHON MODULES COMPILED TO WRITE AND DESIGN VIDEO GAMES.

- os : THIS PROVIDES FUNCTIONS TO INTERACT WITH THE OPERATING SYSTEM **THUS THE NAME , os.**

USER-DEFINED MODULES:

- alarm : SETS AN ALARM BASED ON USER INPUT.
- clock : TELLS THE USER THE CURRENT TIME.
- contact : THIS IS OUR VERSION OF A PHONE BOOK(input using tkinter) WHICH STORES NAME, E-MAIL ID, AND PHONE NUMBER IN A SQL TABLE.
- launch : USES pynput AND pyautogui TO AUTOMATE YOUR WEBSEARCHES AND SKYPE CALLS.
- mail : USES THE EMAIL IN THE DATABASE(PHONEBOOK) AND SENDS AN EMAIL TO THE RECIPIENT.

- wiki : USES THE WIKIPEDIA MODULE TO SEARCH FOR ANY ARTICLE ON THE SAME.

- password : GENERATES AND PRINTS A RANDOM PASSWORD. STRENGTH OF THE PASSWORD CAN ALSO BE SET.

- shutdown : USING THE os MODULE THIS AUTOMATES A SHUTDOWN SEQUENCE IF USER WANTS THE SAME.

- srch : USES pynput AND pyautogui TO AUTOMATE SEARCHES IN THE WINDOWS SEARCH BAR.

- whatsapp : USES selenium AND AUTOMATES THE BROWSER AND SENDS WHATSAPP MESSAGES TO ANY CONTACT ON CONTACT LIST AND USER INPUT.

Code

Main.py

```
import pyttsx3
import speech_recognition
import datetime
import wikipedia
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
import clock #User defined
import whatsapp#User defined
import Wiki#User defined
import mail#user defined
import contact#User defined
import srch#User defined
import shutdown#User defined
import alarm#User defined
import launch#User defined
import password#User defined
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')

engine.setProperty('voice', voices[0].id)


def Goverbal(audio):
    engine.say(audio)
    engine.runAndWait()
```

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

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

    else:
        Goverbal("Good Evening!")

    Goverbal("I am TWO PY .")

def Input():

    global query
    r = speech_recognition.Recognizer()
    with speech_recognition.Microphone() as source:
        Goverbal("A moment of silence, please...")
        r.adjust_for_ambient_noise(source)
        Goverbal("Set minimum energy threshold to {}".format(r.energy_threshold))
        Goverbal("Listening...")
        r.pause_threshold = 1
        audio = r.listen(source)

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

    except Exception as e:
        # print(e)
        Goverbal("Sorry i didn't quite catch that. Say it again")
        Input()
```

Main.py(cont.)

```
if __name__ == "__main__":
    Greet()
    while True:
    # if 1:
        query = Input().lower()
        if 'bye' in query:
            break
        l=['wikipedia','whatsapp','the time','contact','alarm','shutdown','password','launch','open']
        if 'contact' in query:
            contact.con()
        if 'mail' in query:
            mail.run()
        if 'wikipedia' in query:
            Wiki.wiki(query)
        if 'whatsapp' in query:
            whatsapp.whatsapp()
        if 'the time' in query:
            clock.time()
        if 'alarm' in query:
            alarm.alarm()

        if 'shutdown' in query:
            shutdown.off()
        if 'password' in query:
            password.key()


        c=0
        for i in l:
            if i in query:
                c=c+1

        print(c)
        if c == 0:
            srch.srch(query)
```

Clock.py(User-Defined)

SPEAKS OUT THE TIME

```
import main
import datetime
def time():
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    main.Goverbal(f" the time is {strTime}")
```

ShutDown.py(User-Defined)

SHUTS DOWN PC

```
import os
import main
def off():

    main.Goverbal("Do you wish to shutdown your computer ? ")
    shutdown = main.Input()

    if shutdown == 'no':
        exit()
    else:
        main.Goverbal('ok bye')
        os.system("shutdown /s /t 1")
```

Whatsapp.py(User-Defined)

OPENS WHATSAPP WEB AND SEND MESSAGE

```
def whatsapp():


    driver = webdriver.Chrome(ChromeDriverManager().install())
    driver.get('https://web.whatsapp.com/')
    main.Goverbal('Enter the name of user or group : ')
    name =main.Input()
    main.Goverbal('Enter your message : ')
    msg = main.Input()


    user = driver.find_element_by_xpath('//span[@title = "{}"]'.format(name))
    user.click()

    msg_box = driver.find_element_by_class_name('_3u328')

    for i in range(0):
        msg_box.send_keys(msg)
        button = driver.find_element_by_class_name('_3M-N-')
        button.click()
```

WIKI.py(User-Defined)

SEARCHES QUERY IN WIKIPEDIA AND READS OUT THE SUMMARY

```
import wikipedia
import main
def wiki(a):
    if a == 'wikipedia':
        main.Goverbal('Please ask your question ....')
        main.Input()

    else:
        main.Goverbal('traversing through the wide collections of wikipedia...')
        a = a.replace("wikipedia", "")
        results = wikipedia.summary(a, sentences=2)
        main.Goverbal("According to Wiki")
        print(results)
        main.Goverbal(results)
```

Contact.py(User-Defined)

SAVES CONTACT DETAILS IN MySQL DATABASE

```
import mysql.connector

import tkinter as tk

root=tk.Tk()


root.geometry("600x400")


name_var=tk.StringVar()
email_var=tk.StringVar()
phone_var=tk.StringVar()
def con():
    def submit():
        a=mysql.connector.connect(host="localhost",port='3306',user="root",passwd="root123",database="pavel")

        global ab
        global emailid
        global phone
        ab=name_var.get()
        emailid=email_var.get()
        phone=phone_var.get()

        cursor=a.cursor()
        b="Insert into contacts (Name,Phone,Email) values "+str((ab,phone,emailid))
        cursor.execute(b)
        a.commit()
        a.close()



        name_var.set("")
        email_var.set("")
        phone_var.set("")


    name_label = tk.Label(root, text = 'Name',
    font=('calibre',10, 'bold'))

    name_entry = tk.Entry(root,
    textvariable = name_var,
    font=('calibre',10,'normal'))


    emailid_label = tk.Label(root,
    text = 'mail-id',
    font = ('calibre',10,'bold'))


    emailid_entry=tk.Entry(root,
    textvariable = email_var,
    font = ('calibre',10,'normal'))

    phone_label = tk.Label(root,
    text = 'phone number',
    font = ('calibre',10,'bold'))


    phone_entry=tk.Entry(root,
    textvariable = phone_var,
    font = ('calibre',10,'normal'))

    sub_btn=tk.Button(root,text = 'Submit',
    command = submit)

    name_label.grid(row=0,column=0)
    name_entry.grid(row=0,column=1)
    emailid_label.grid(row=1,column=0)
    emailid_entry.grid(row=1,column=1)
    phone_label.grid(row=2,column=0)
    phone_entry.grid(row=2,column=1)
    sub_btn.grid(row=3,column=1)
    root.mainloop()
```

Mail.py(User-Defined)

SENDS MAIL YOU WRITE TO SPECIFIED EMAIL

```
import smtplib as s
import tkinter as tk
global root

root=tk.Tk()

root.geometry("600x400")

sub_var=tk.StringVar()
bod_var=tk.StringVar()
mail_var=tk.StringVar()
def run():
    def submit():
        global sa
        global b
        global m


        sa=sub_var.get()
        b=bod_var.get()
        m=mail_var.get()

        sub_var.set("")
        bod_var.set("")
        mail_var.set("")

    sub_label = tk.Label(root, text = 'Subject', font=('calibre',10, 'bold'))

    sub_entry = tk.Entry(root, textvariable = sub_var, font=('calibre',10,'normal'))


    bod_label = tk.Label(root, text = 'Body', font = ('calibre',10,'bold'))


    bod_entry = tk.Entry(root, textvariable = bod_var, font = ('calibre',10,'normal'))

    mail_label = tk.Label(root, text = 'Email-id', font = ('calibre',10,'bold'))


    mail_entry=tk.Entry(root, textvariable = mail_var, font = ('calibre',10,'normal'))


    send_btn=tk.Button(root,text = 'Send', command= submit)

    sub_label.grid(row=0,column=0)
    sub_entry.grid(row=0,column=1)
    bod_label.grid(row=1,column=0)
    bod_entry.grid(row=1,column=1)
    mail_label.grid(row=2,column=0)
    mail_entry.grid(row=2,column=1)
    send_btn.grid(row=3,column=1)
    root.mainloop()



    ob=s.SMTP("smtp.gmail.com",587)
    ob.starttls()
    ob.login("pavelnarayan660@gmail.com","twopy.2020")


    message="Subject:{}\n\n{}".format(sa,b)
```

Srch.py(User-Defined)

SEARCHES THE QUERY IN BING

```
#Opens Windows Search and Searches the query
from pynput.keyboard import Key, Controller
import main
import time
import pyautogui
keyboard = Controller()
def srch(query):
    keyboard.press(Key.cmd)
    keyboard.release(Key.cmd)

    time.sleep(1)


    for char in query:
        keyboard.press(char)
        keyboard.release(char)
        time.sleep(0.11)
    keyboard.press(Key.enter)
    keyboard.release(Key.enter)
    time.sleep(3)
```

ALARM.py(User-Defined)

SETS AN ALARM USING PyGAME MODULE

```
import pygame
import time
import main
def alarm():
    main.Goverbal("Enter the hour")
    hr=int(main.Input())
    main.Goverbal("Enter the minute")
    mn=int(main.Input())
    pygame.mixer.init()
    sound = pygame.mixer.Sound("bullet.mp3")

    n=5

    print("Alarm set for",str(hr),":",str(mn))
    while True:
        if time.localtime().tm_hour==hr and time.localtime().tm_min==mn
            print("Wake up")
            break


    while n>0:
        sound.play()
        time.sleep(2)

    snz=main.Input()
    if snz=='Stop':
        n=3
        time.sleep(100)
        while n>0:
            sound.play()
            time.sleep(2)
    else:
        exit()
        sound()
        alarm()
```

Launch.py(User-Defined)

LAUNCHES SUPPORTED APPS LIKE SKYPE AND CHROME AND AUTOMATES TASKS

```
from pynput.keyboard import Key, Controller
import main
import time
import pyautogui
def launch(query):
    if 'launch' in query:
        query = query.replace("launch", "")
    elif 'open' in query:
        query = query.replace("open", "")
    keyboard = Controller()


    keyboard.press(Key.cmd)
    keyboard.release(Key.cmd)

    time.sleep(1)


    for char in query:
        keyboard.press(char)
        keyboard.release(char)
        time.sleep(0.11)
    keyboard.press(Key.enter)
    keyboard.release(Key.enter)
    time.sleep(3)

    if 'skype' in query:
        main.Goverbal('what would you like to do?')
        options=main.Input()
        if 'video' in options:
            main.Goverbal('who would you like to call?')
            name=main.Input()
            pyautogui.moveTo(201, 152, duration = 0.5)  #158,318
            pyautogui.click(201,152)
            pyautogui.moveTo(185, 101, duration = 0.5)  #158,318
            pyautogui.click(185,101)
            time.sleep(5)
            pyautogui.typewrite(name)
            time.sleep(4)
            pyautogui.moveTo(158, 318, duration = 0.5)  #158,318
            pyautogui.click(158,318)
            pyautogui.moveTo(1144, 80, duration = 0.5)  #158,318
            pyautogui.click(1144,80)
        elif 'voice' in options:
            main.Goverbal('who would you like to call?')
            name=main.Input()
            pyautogui.moveTo(201, 152, duration = 0.5)  #158,318
            pyautogui.click(201,152)
            pyautogui.moveTo(185, 101, duration = 0.5)  #158,318
            pyautogui.click(185,101)
            time.sleep(5)
            pyautogui.typewrite(name)
            time.sleep(4)
            pyautogui.moveTo(158, 318, duration = 0.5)  #158,318
            pyautogui.click(158,318)
            pyautogui.moveTo(1282, 74, duration = 0.5)  #158,318
            pyautogui.click(1282,74)

    if 'chrome' in query:
        main.Goverbal('What would you like to do?')
        option=Input()
        if option == 'search':
            main.Goverbal('enter your search')
            query=main.Input()
            pyautogui.typewrite(query)
            keyboard.press(Key.enter)
            keyboard.release(Key.enter)
        elif option=='history':
             pyautogui.hotkey("ctrlleft", "h")



        elif option == 'private':
            pyautogui.typewrite(["ctrlleft","shift","n"])
```

Password.py(User-Defined)

HELPS GENERATE STRONG PASSWORDS AND SAVES THEM IN MySQL DATABASE

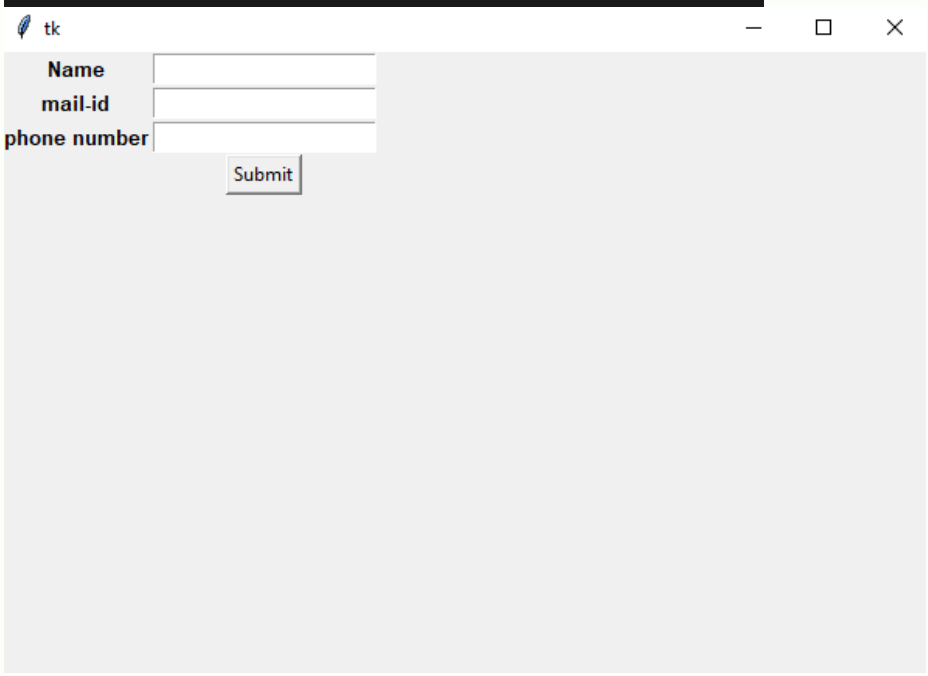
```
from pynput.keyboard import Key, Controller
import main
import time
import pyautogui
def launch(query):
    if 'launch' in query:
        query = query.replace("launch", "")
    elif 'open' in query:
        query = query.replace("open", "")
    keyboard = Controller()


    keyboard.press(Key.cmd)
    keyboard.release(Key.cmd)

    time.sleep(1)


    for char in query:
        keyboard.press(char)
        keyboard.release(char)
        time.sleep(0.11)
    keyboard.press(Key.enter)
    keyboard.release(Key.enter)
    time.sleep(3)

    if 'skype' in query:
        main.Goverbal('what would you like to do?')
        options=main.Input()
        if 'video' in options:
            main.Goverbal('who would you like to call?')
            name=main.Input()
            pyautogui.moveTo(201, 152, duration = 0.5)  #158,318
            pyautogui.click(201,152)
            pyautogui.moveTo(185, 101, duration = 0.5)  #158,318
            pyautogui.click(185,101)
            time.sleep(5)
            pyautogui.typewrite(name)
            time.sleep(4)
            pyautogui.moveTo(158, 318, duration = 0.5)  #158,318
            pyautogui.click(158,318)
            pyautogui.moveTo(1144, 80, duration = 0.5)  #158,318
            pyautogui.click(1144,80)
        elif 'voice' in options:
            main.Goverbal('who would you like to call?')
            name=main.Input()
            pyautogui.moveTo(201, 152, duration = 0.5)  #158,318
            pyautogui.click(201,152)
            pyautogui.moveTo(185, 101, duration = 0.5)  #158,318
            pyautogui.click(185,101)
            time.sleep(5)
            pyautogui.typewrite(name)
            time.sleep(4)
            pyautogui.moveTo(158, 318, duration = 0.5)  #158,318
            pyautogui.click(158,318)
            pyautogui.moveTo(1282, 74, duration = 0.5)  #158,318
            pyautogui.click(1282,74)

    if 'chrome' in query:
        main.Goverbal('What would you like to do?')
        option=Input()
        if option == 'search':
            main.Goverbal('enter your search')
            query=main.Input()
            pyautogui.typewrite(query)
            keyboard.press(Key.enter)
            keyboard.release(Key.enter)
        elif option=='history':
             pyautogui.hotkey("ctrlleft", "h")



        elif option == 'private':
            pyautogui.typewrite(["ctrlleft","shift","n"])
```

Sample Output

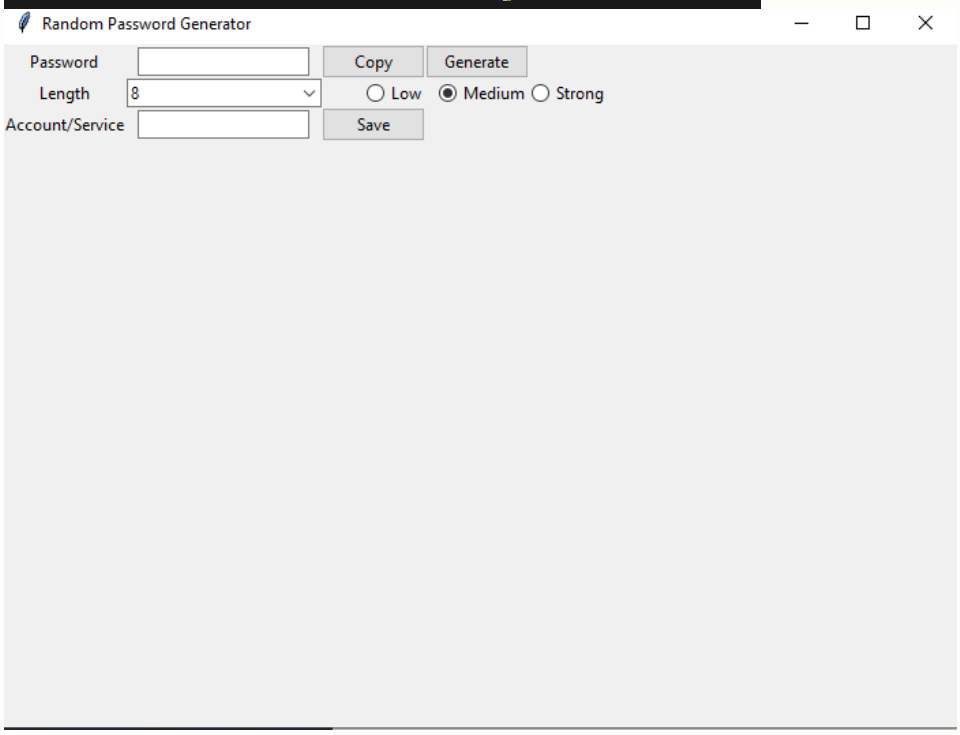
Disclaimer: All Inputs and outputs are in Audio.

```
PS D:\dev\assistant> pipenv run python main.py
pygame 2.0.0 (SDL 2.0.12, python 3.7.9)
Hello from the pygame community. https://www.pygame.org/contribute.html
A moment of silence please...
Set minimum energy threshold to 139.69883646318394
Listening..
Recognising...
User said: Python Wikipedia

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability
 with its notable use of significant whitespace.
1
A moment of silence please...
Set minimum energy threshold to 142.36440322957472
Listening..
Recognising...
User said: set an alarm

A moment of silence please...
Set minimum energy threshold to 131.4735394260647
Listening..
Recognising...
User said: 12

A moment of silence please...
Set minimum energy threshold to 123.14641308577936
Listening..
Recognising...
User said: 45

Alarm set for 12 : 45
```

```
User said: save a contact
```



```
User said: set a password
```




```
User said: what's the time

10:58:44
```

REFERENCES

- Python Package Index: https://pypi.org/
- "Computer Science with Python" by Sumita Arora
- GeeksforGeeks: https://www.geeksforgeeks.org/
- Stack Overflow: https://stackoverflow.com/
- W3Schools:https://www.w3schools.com/
- Unofficial Python Binaries: https://www.lfd.uci.edu/~gohlke/pythonlibs/