SOURCE CODE

main.py

```
import speech_recognition as sr
import pyttsx3
import datetime
import webbrowser
import pywhatkit as kit
import os
# Initialize the speech engine
engine = pyttsx3.init()
# Function to make the assistant speak
def speak(text):
engine.say(text)
engine.runAndWait()
# Function to listen to the user's command
def listen():
recognizer = sr.Recognizer()
with sr.Microphone() as source:
print("Listening...")
audio = recognizer.listen(source)
try:
command = recognizer.recognize_google(audio)
print("You said:", command)
return command.lower()
speak("Sorry, I could not understand that.")
return None
exceptsr.RequestError:
speak("Sorry, there was an issue with the speech service.")
return None
# Function to get the current time
def tell_time():
time = datetime.datetime.now().strftime("%H:%M:%S")
speak(f"The current time is {time}")
# Function to search on the web
def search_web(query):
speak(f"Searching for {query}")
webbrowser.open(f"https://www.google.com/search?q={query}")
# Function to play a YouTube video
def play_video(query):
speak(f"Playing {query} on YouTube")
kit.playonyt(query)
# Function to open applications
def open_application(app_name):
if 'chrome' in app_name:
```

```
speak("Opening Google Chrome")
os.system("start chrome")
elif 'notepad' in app_name:
speak("Opening Notepad")
os.system("start notepad")
elif 'calculator' in app_name:
speak("Opening Calculat
speak("Sorry, I could not understand that.")
return None
import cv2
import mediapipe as mp
import pyautogui
cam = cv2.VideoCapture(0)
face_mesh = mp.solutions.face_mesh.FaceMesh(refine_landmarks=True)
screen_w, screen_h = pyautogui.size()
while True:
_, frame = cam.read()
frame = cv2.flip(frame, 1)
cv2.circle(frame, (x, y), 3, (0, 255, 0))
if id == 1:
screen_x = screen_w * landmark.x
screen_y = screen_h * landmark.y
pyautogui.moveTo(screen_x, screen_y)
left = [landmarks [145], landmarks [159]]
for landmark in left:
x = int(landmark.x * frame_w)
y = int(landmark.y * frame_h)
cv2.circle(frame, (x, y), 3, (0, 255, 255))
if (left [0].y - left[1].y) < 0.004:
pyautogui.click()
pyautogui.sleep(1)
cv2.imshow('Eye Controlled Mouse', frame)
cv2.waitKe
```