```
from __future__ import print_function
import datetime
import pickle
import os.path
from googleapiclient.discovery import build
from google_auth_oauthlib.flow import InstalledAppFlow
from google.auth.transport.requests import Request
import os
import pyttsx3
import speech_recognition as sr
import pytz
import subprocess

SCOPES = ["https://www.googleapis.com/auth/calendar.readonly"]
MONTHS = [
    "january",
    "february",
    "march",
    "april",
    "may",
    "june",
    "july",
    "august",
    "september",
    "october",
    "november",
    "december",
]
DAYS = ["monday", "tuesday", "wednesday", "thursday", "friday",
"saturday", "sunday"]
DAY_EXTENTIONS = ["rd", "th", "st", "nd"]


def speak(text):
    engine = pyttsx3.init()
    engine.say(text)
    engine.runAndWait()


def get_audio():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        audio = r.listen(source)
        said = ""

        try:
            said = r.recognize_google(audio)
            print(said)
        except Exception as e:
            print("Exception: " + str(e))

    return said.lower()


def authenticate_google():
    """Shows basic usage of the Google Calendar API.
    Prints the start and name of the next 10 events on the user's
calendar.
```

```
    """
    creds = None
    if os.path.exists("token.pickle"):
        with open("token.pickle", "rb") as token:
            creds = pickle.load(token)

    if not creds or not creds.valid:
        if creds and creds.expired and creds.refresh_token:
            creds.refresh(Request())
        else:
            flow =
InstalledAppFlow.from_client_secrets_file("credentials.json", SCOPES)
            creds = flow.run_local_server(port=0)

        with open("token.pickle", "wb") as token:
            pickle.dump(creds, token)

    service = build("calendar", "v3", credentials=creds)

    return service


def get_events(day, service):
    # Call the Calendar API
    date = datetime.datetime.combine(day, datetime.datetime.min.time())
    end_date = datetime.datetime.combine(day,
datetime.datetime.max.time())
    utc = pytz.UTC
    date = date.astimezone(utc)
    end_date = end_date.astimezone(utc)

    events_result = (
        service.events()
        .list(
            calendarId="primary",
            timeMin=date.isoformat(),
            timeMax=end_date.isoformat(),
            singleEvents=True,
            orderBy="startTime",
        )
        .execute()
    )
    events = events_result.get("items", [])

    if not events:
        speak("No upcoming events found.")
    else:
        speak(f"You have {len(events)} events on this day.")

        for event in events:
            start = event["start"].get("dateTime",
event["start"].get("date"))
            print(start, event["summary"])
            start_time = str(start.split("T")[1].split("-")[0])
            if int(start_time.split(":")[0]) < 12:
                start_time = start_time + "am"
            else:
                start_time = (
```

```
                    str(int(start_time.split(":")[0]) - 12) +
start_time.split(":")[1]
                )
                start_time = start_time + "pm"

            speak(event["summary"] + " at " + start_time)


def get_date(text):
    text = text.lower()
    today = datetime.date.today()

    if text.count("today") > 0:
        return today

    day = -1
    day_of_week = -1
    month = -1
    year = today.year

    for word in text.split():
        if word in MONTHS:
            month = MONTHS.index(word) + 1
        elif word in DAYS:
            day_of_week = DAYS.index(word)
        elif word.isdigit():
            day = int(word)
        else:
            for ext in DAY_EXTENTIONS:
                found = word.find(ext)
                if found > 0:
                    try:
                        day = int(word[:found])
                    except:
                        pass

    if (
        month < today.month and month != -1
    ):  # if the month mentioned is before the current month set the year
to the next
        year = year + 1

    if month == -1 and day != -1:  # if we didn't find a month, but we
have a day
        if day < today.day:
            month = today.month + 1
        else:
            month = today.month

    # if we only found a day of the week
    if month == -1 and day == -1 and day_of_week != -1:
        current_day_of_week = today.weekday()
        dif = day_of_week - current_day_of_week

        if dif < 0:
            dif += 7
            if text.count("next") >= 1:
                dif += 7
```

```
        return today + datetime.timedelta(dif)

    if day != -1:  # FIXED FROM VIDEO
        return datetime.date(month=month, day=day, year=year)


def note(text):
    date = datetime.datetime.now()
    file_name = str(date).replace(":", "-") + "-note.txt"
    with open(file_name, "w") as f:
        f.write(text)

    subprocess.Popen(["notepad.exe", file_name])


WAKE = "hey tim"
SERVICE = authenticate_google()
print("Start")

while True:
    print("Listening")
    text = get_audio()

    if text.count(WAKE) > 0:
        speak("I am ready")
        text = get_audio()

        CALENDAR_STRS = ["what do i have", "do i have plans", "am i
busy"]
        for phrase in CALENDAR_STRS:
            if phrase in text:
                date = get_date(text)
                if date:
                    get_events(date, SERVICE)
                else:
                    speak("I don't understand")

        NOTE_STRS = ["make a note", "write this down", "remember this"]
        for phrase in NOTE_STRS:
            if phrase in text:
                speak("What would you like me to write down?")
                note_text = get_audio()
                note(note_text)
                speak("I've made a note of that.")
```

Output & voice feedback :

```
Start
Listening
hey tim
I am ready
Listening
what do i have today?
you (printed on console)
Getting the upcoming 2 days
2025-07-24T09:00:00-04:00 Doctor’s Appointment
```

You have 2 events on this day.
Doctor’s Appointment at 9am
Project Review at 2pm
Listening
hey tim
I am ready
Listening
make a note
you
What would you like me to write down?
you
Remember to call mom
I've made a note of that.