### **Project 10: Automation Script**

#### **Objective:**

Create an automation script that performs a daily task, such as sending an email reminder, downloading files, or interacting with APIs (e.g., fetching daily weather or stock data).

### **Instructions**

#### **Step 1: Set Up the Environment**

1. Create a new Python file called automation\_script.py.

Install the required libraries:  
Copy code  
pip install requests smtplib schedule

#### **Step 2: Import Necessary Libraries**

Import the required libraries at the top of your script:  
python  
Copy code  
import requests

import smtplib

import schedule

import time

#### **Step 3: Create a Function to Perform the Task**

Define a function called fetch\_weather() to get the weather data from an API (e.g., OpenWeatherMap):  
python  
Copy code  
def fetch\_weather():

api\_key = "your\_api\_key"

city = "Sydney"

url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api\_key}"

response = requests.get(url)

if response.status\_code == 200:

weather\_data = response.json()

temp = weather\_data["main"]["temp"] - 273.15 # Convert from Kelvin to Celsius

print(f"Current temperature in {city} is {temp:.2f}°C")

else:

print("Failed to fetch weather data.")

#### **Step 4: Schedule the Task**

Use the schedule library to run the function every day at a specific time:  
python  
Copy code  
schedule.every().day.at("09:00").do(fetch\_weather)

#### **Step 5: Create a Function to Send an Email**

Define a function to send an email using smtplib:  
python  
Copy code  
def send\_email(subject, body):

from\_email = "your\_email@gmail.com"

to\_email = "recipient\_email@gmail.com"

password = "your\_password"

with smtplib.SMTP("smtp.gmail.com", 587) as server:

server.starttls()

server.login(from\_email, password)

message = f"Subject: {subject}\n\n{body}"

server.sendmail(from\_email, to\_email, message)

Modify fetch\_weather() to send an email with the weather report:  
python  
Copy code  
def fetch\_weather():

api\_key = "your\_api\_key"

city = "Sydney"

url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api\_key}"

response = requests.get(url)

if response.status\_code == 200:

weather\_data = response.json()

temp = weather\_data["main"]["temp"] - 273.15 # Convert from Kelvin to Celsius

weather\_report = f"Current temperature in {city} is {temp:.2f}°C"

send\_email("Daily Weather Report", weather\_report)

print(weather\_report)

else:

print("Failed to fetch weather data.")

#### **Step 6: Run the Scheduler**

Add the following code to keep the script running:  
python  
Copy code  
while True:

schedule.run\_pending()

time.sleep(1)

#### **Step 7: Test the Automation**

1. Run the script and verify that it fetches the weather data and sends an email at the scheduled time.