PYTHON SCRIPT

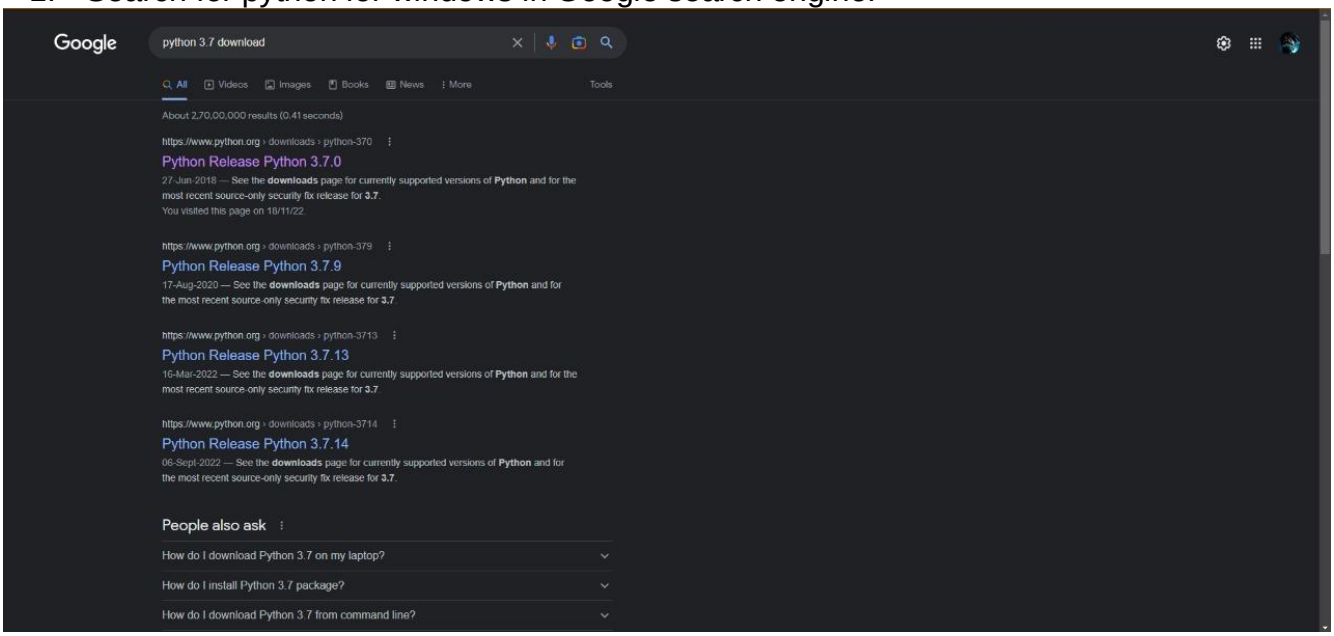
Assignment Date	06 NOVEMBER2022
Team ID	PNT2022TMID32740
Project Name	Gas Leakage Monitoring and Alerting System

AIM:

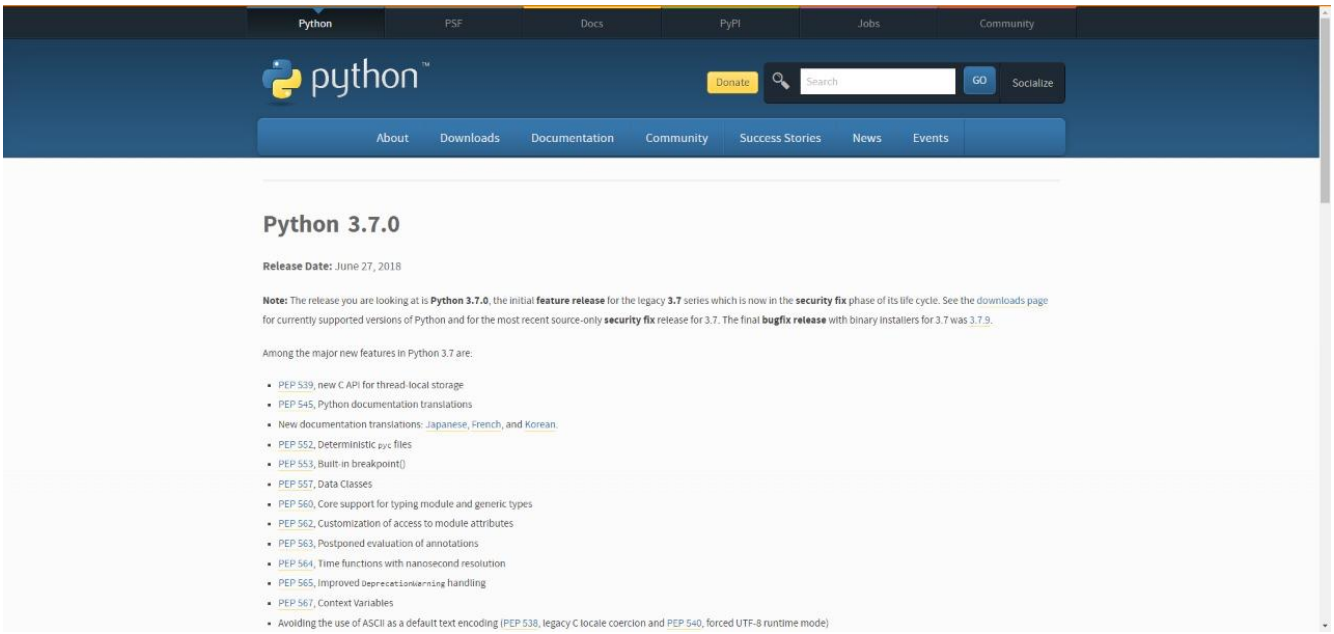
To install python version 3.9.6 and IBM Watson IoT platform packages in python.

STEPS:

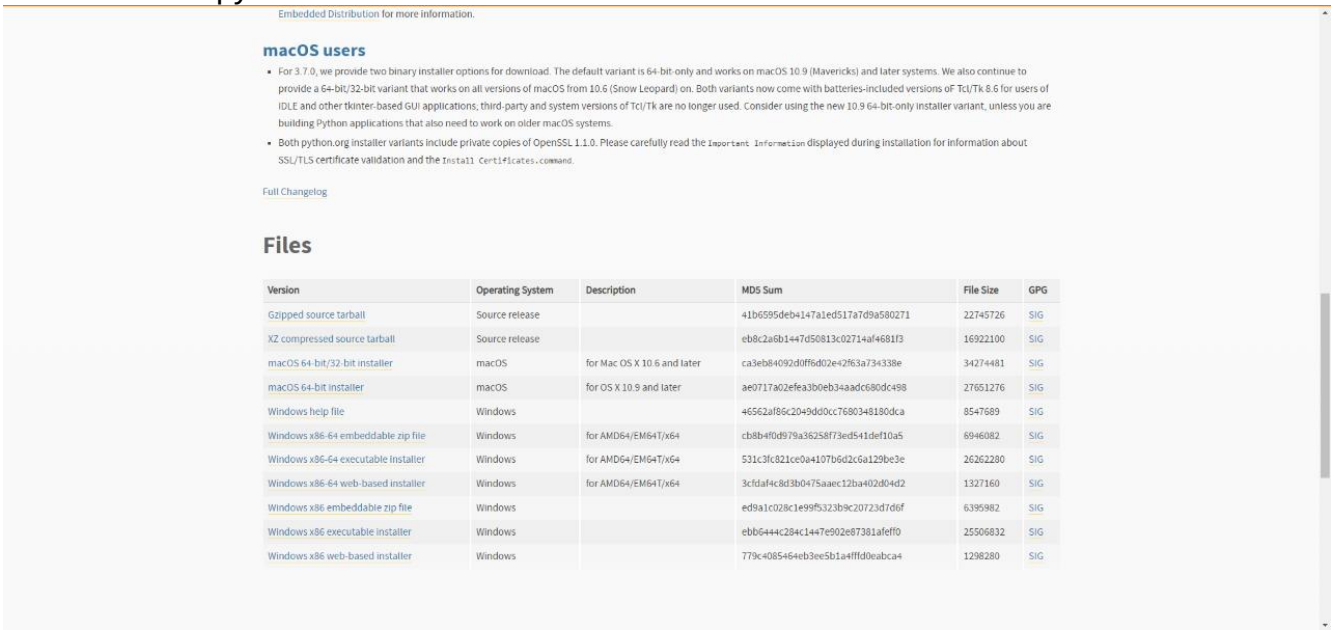
1. Search for python for windows in Google search engine.



2. Click the First link..



3. Click the python version 3.9.6 and download.



Embedded Distribution for more information.

macOS users

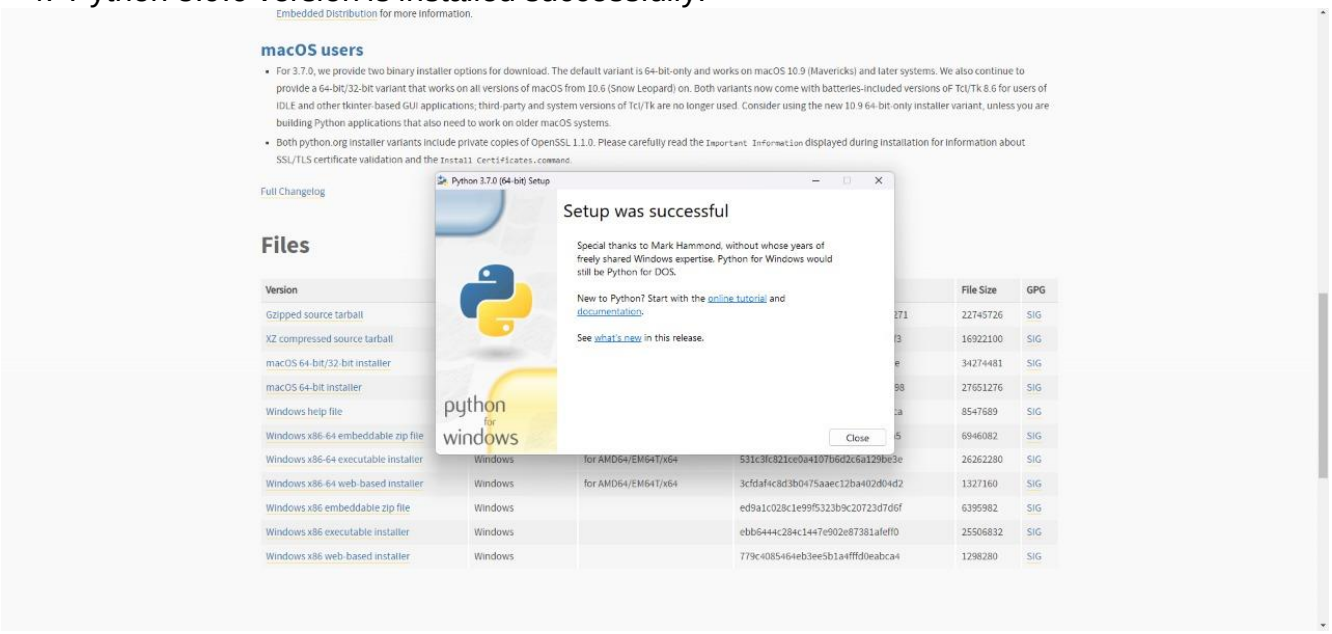
- For 3.7.0, we provide two binary installer options for download. The default variant is 64-bit-only and works on macOS 10.9 (Mavericks) and later systems. We also continue to provide a 64-bit/32-bit variant that works on all versions of macOS from 10.6 (Snow Leopard) on. Both variants now come with batteries-included versions of Tcl/Tk 8.6 for users of IDLE and other tkinter-based GUI applications; third-party and system versions of Tcl/Tk are no longer used. Consider using the new 10.9 64-bit-only installer variant, unless you are building Python applications that also need to work on older macOS systems.
- Both python.org installer variants include private copies of OpenSSL 1.1.0. Please carefully read the `Important Information` displayed during installation for information about SSL/TLS certificate validation and the `Install Certificates.command`.

Full Changelog

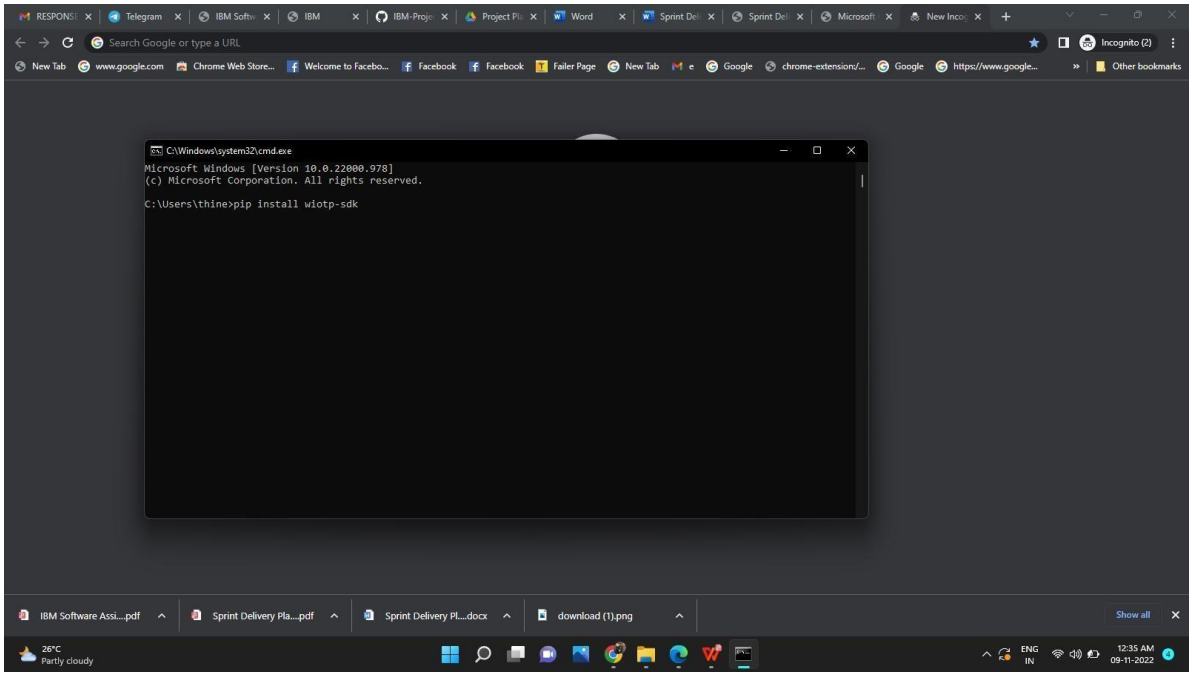
Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		41b6595deb4147a1ed517a7d9a580271	22745726	SIG
XZ compressed source tarball	Source release		eb8c2a6b1447d50813c02714af4681f3	16922100	SIG
macOS 64-bit/32-bit installer	macOS	for Mac OS X 10.6 and later	ca3eb84092d0ff6d02e42f63a734338e	34274481	SIG
macOS 64-bit installer	macOS	for OS X 10.9 and later	ae0717a02efea3b0eb34aadc680dc498	27651276	SIG
Windows help file	Windows		46562af86c2049dd0cc7680348180dca	8547689	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	cb8b4f0d979a36258f73ed541def10a5	6946082	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	531c3fc821ce0a4107b6d2c6a129be3e	26262280	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	3cfdaf4c8d3b047saaec12ba402d04d2	1327160	SIG
Windows x86 embeddable zip file	Windows		ed9a1c028c1e99f5323b9c20723d7d6f	6395982	SIG
Windows x86 executable installer	Windows		ebb6444c284c1447e902e87381afeff0	25506832	SIG
Windows x86 web-based installer	Windows		779c4085464eb3ee5b1a4fffd0eabca4	1298280	SIG

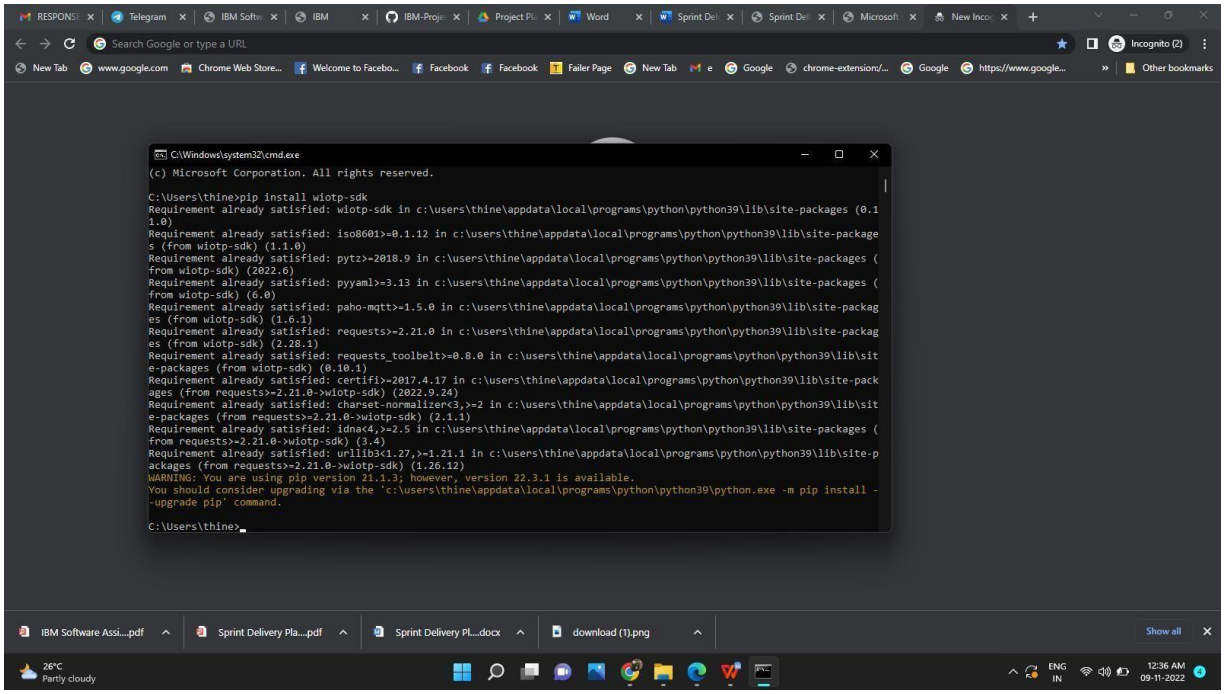
4. Python 3.9.6 version is installed successfully.



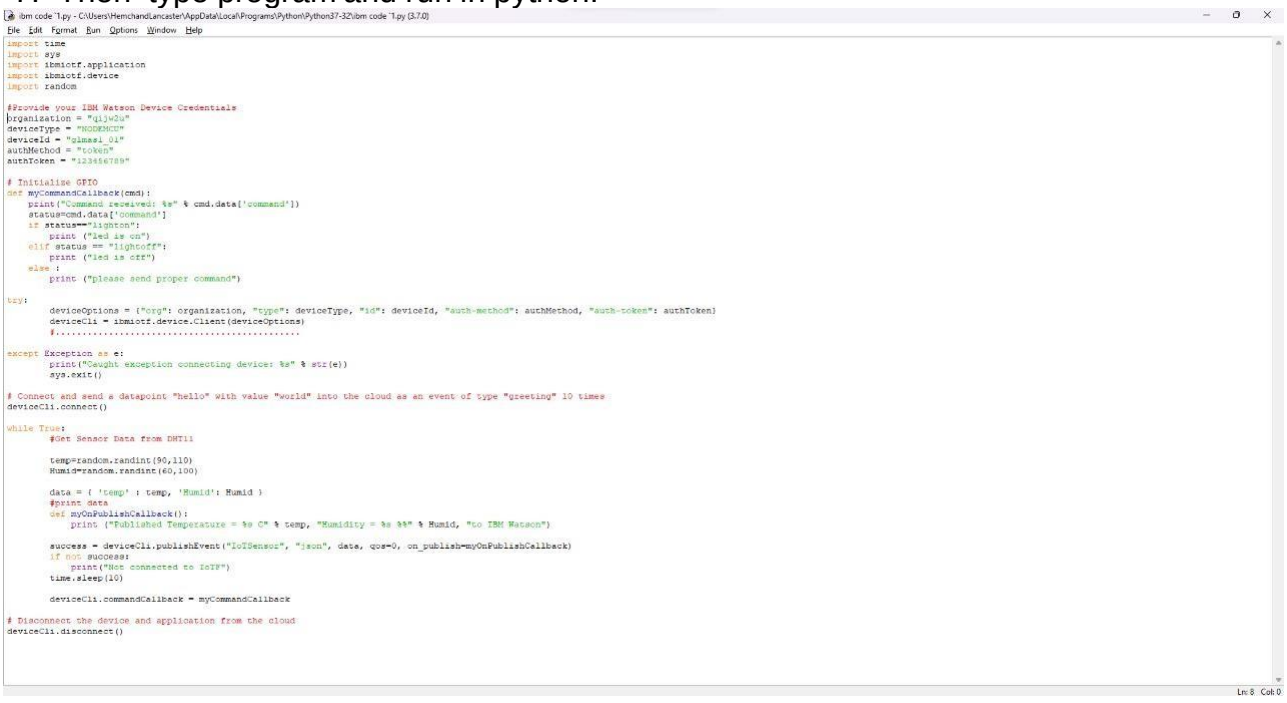
5. Now install the IBM Watson Platform package through command prompt.



6. The package will be installed.



7. Then type program and run in python.



RESULT:

The python version 3.9.6 and IBM Watson IoT platform package are installed successfully.

Python Code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "qijw2u"
deviceType = "NODEMCU"
deviceId = "glmas1_01"
authMethod = "token"
authToken = "123456789"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led is off")
    else :
        print ("please send proper command")

try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
        #.....................................................

except Exception as e:
        print("Caught exception connecting device: %s" % str(e))
        sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type
"greeting" 10 times
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11

    temp=random.randint(90,110)
    Humid=random.randint(60,100)

    data = { 'temp' : temp, 'Humid': Humid }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "to IBM
Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(10)
```

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()