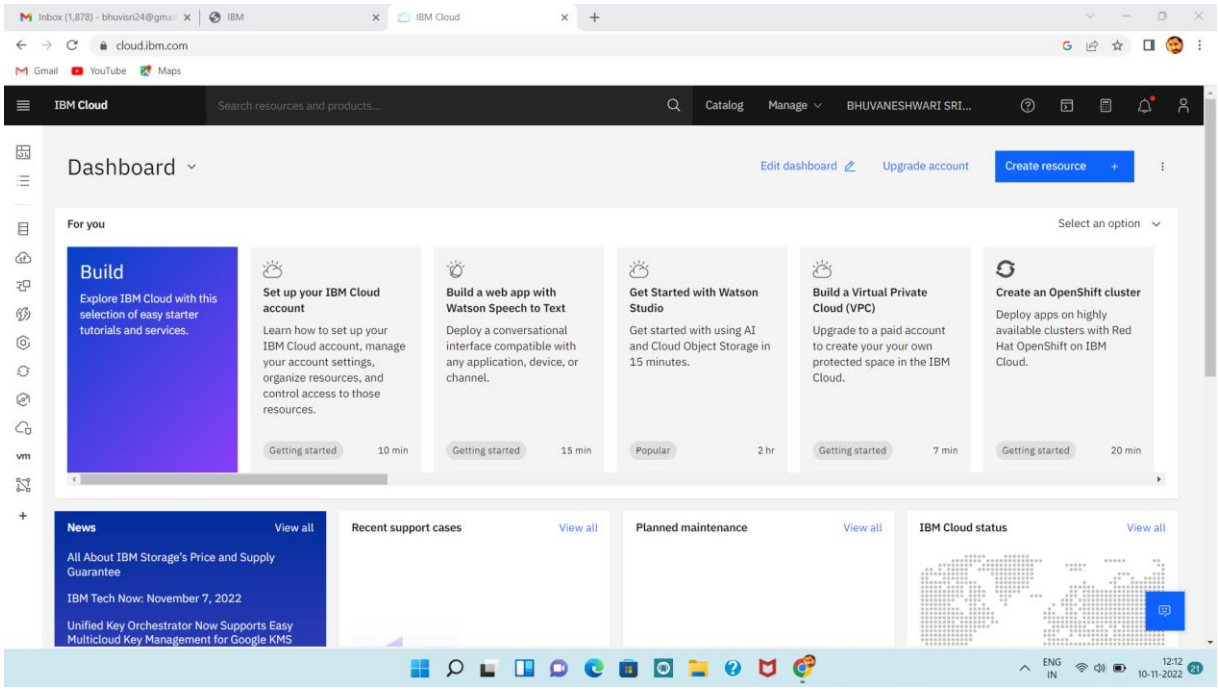
REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

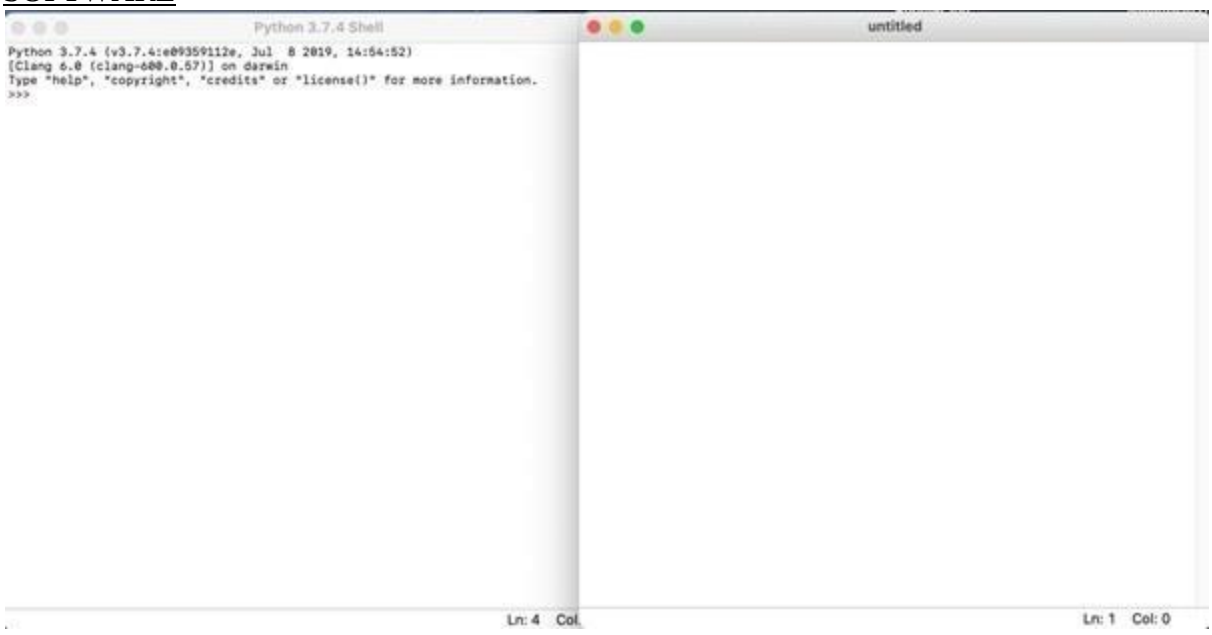
TEAM ID	PNT2022TMID23838
PROJECT NAME	REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM
LEADER NAME	BHUVANESHWARI SRIDHAR S
TEAM MEMBERS NAME	BAVANI P DIVEDHA V KAVIYARASI P

PREREQUISITES:

IBM CLOUD SERVICES

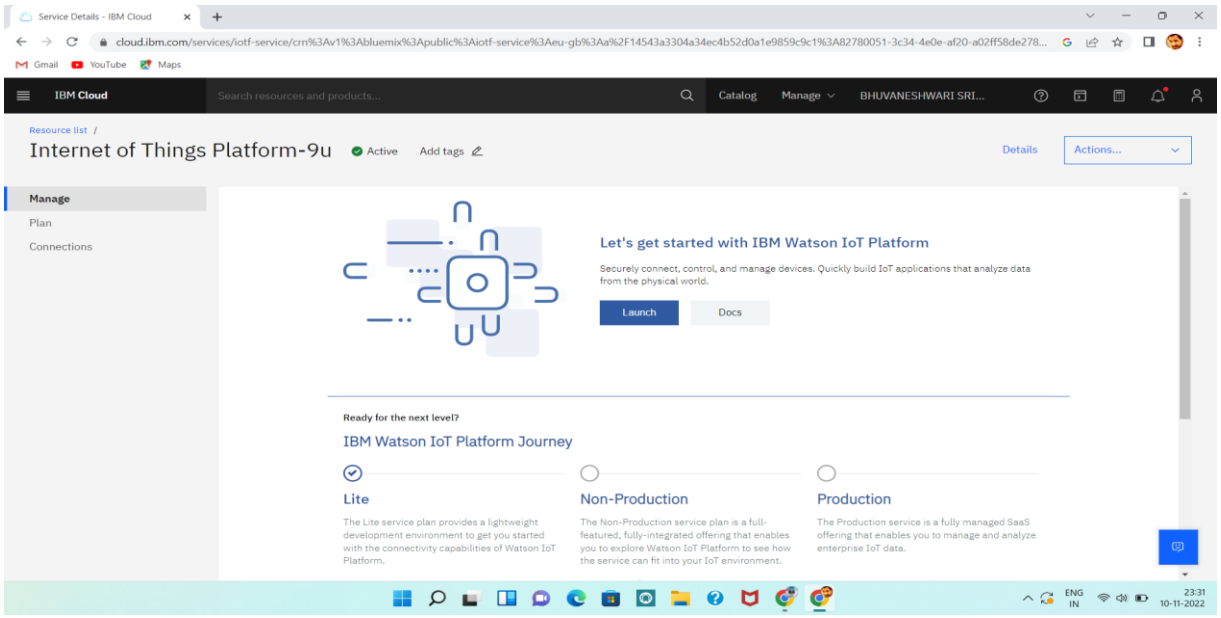


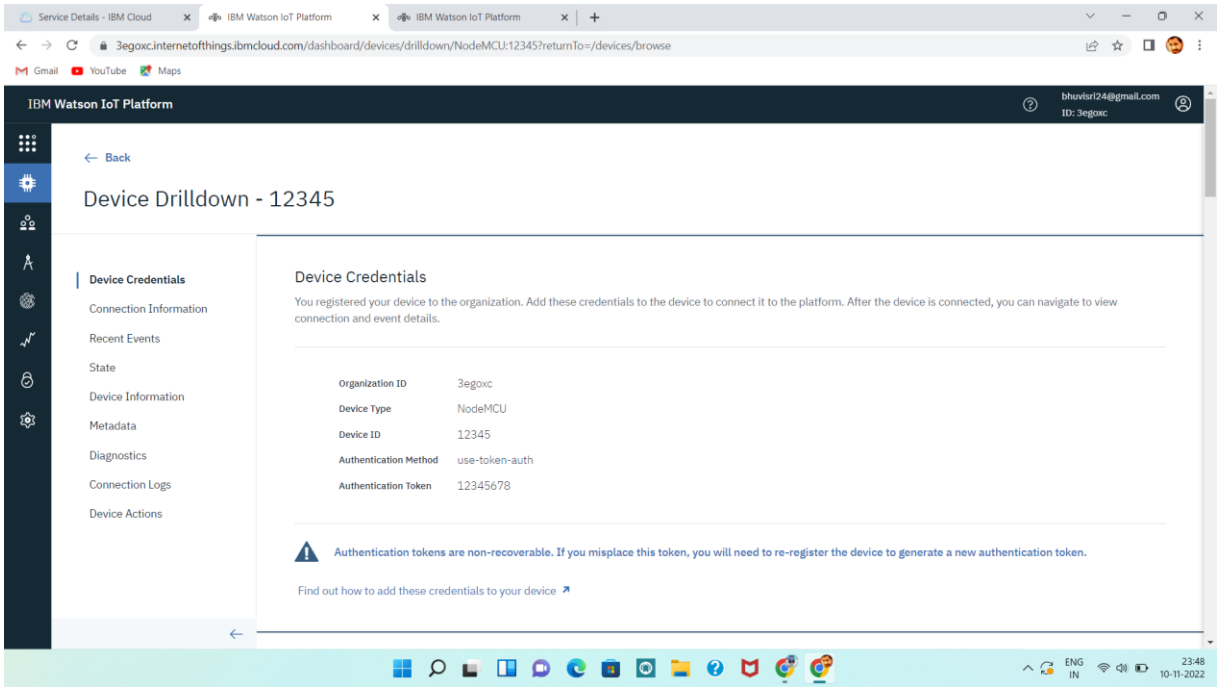
SOFTWARE



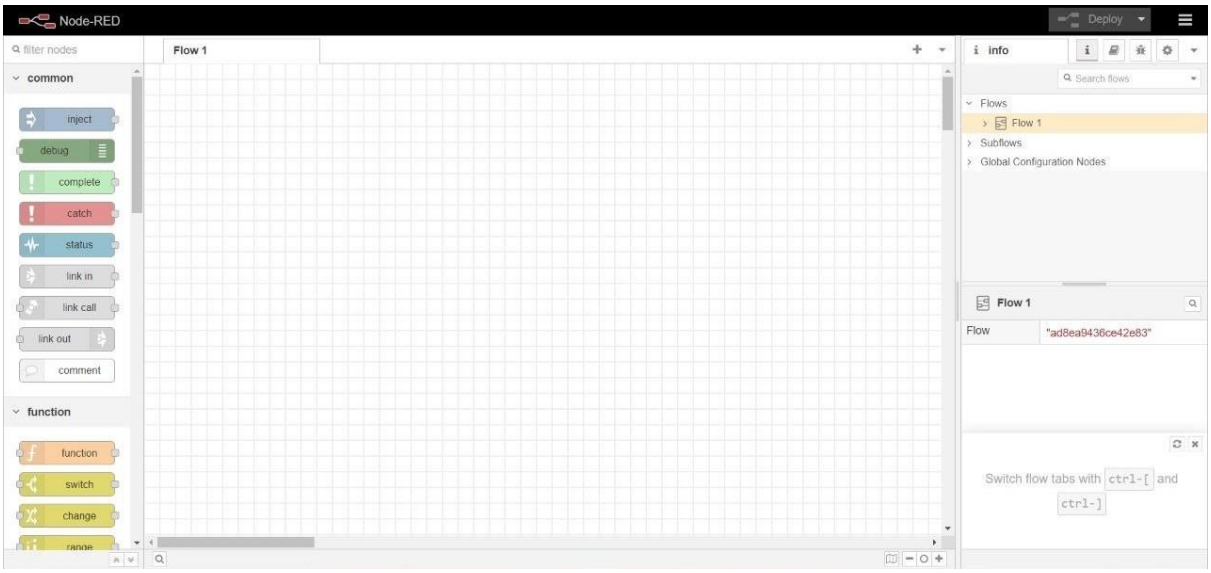
CREATE AN CONFIGURE IBM CLOUD SERVIVES:

CREATE IBM WATSON IOT PLATFORM AND DEVICE


Service Details - IBM Cloud
cloud.ibm.com/services/iotf-service/crn%3Av1%3Abluemix%3Apublic%3Aiotf-service%3Aeu-gb%3Aa%2F14543a3304a34ec4b52d0a1e9859c9c1%3A82780051-3c34-4e0e-af20-a02ff58de278...
Gmail
YouTube
Maps
IBM Cloud
Search resources and products...
Catalog
Manage
BHUVANESHWARI SRI...
Resource list /
Internet of Things Platform-9u
Active
Add tags
Details
Actions...
Manage
Plan
Connections
Let's get started with IBM Watson IoT Platform
Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.
Launch
Docs
Ready for the next level?
IBM Watson IoT Platform Journey
Lite
The Lite service plan provides a lightweight development environment to get you started with the connectivity capabilities of Watson IoT Platform.
Non-Production
The Non-Production service plan is a full-featured, fully-integrated offering that enables you to explore Watson IoT Platform to see how the service can fit into your IoT environment.
Production
The Production service is a fully managed SaaS offering that enables you to manage and analyze enterprise IoT data.
ENG
IN
23:31
10-11-2022


Service Details - IBM Cloud
IBM Watson IoT Platform
IBM Watson IoT Platform
3egoxc.internetofthings.ibmcloud.com/dashboard/devices/drilldown/NodeMCU:12345?returnTo=/devices/browse
Gmail
YouTube
Maps
IBM Watson IoT Platform
bhuvisrl24@gmail.com
ID: 3egoxc
Back
Device Drilldown - 12345
Device Credentials
Connection Information
Recent Events
State
Device Information
Metadata
Diagnostics
Connection Logs
Device Actions
Device Credentials
You registered your device to the organization. Add these credentials to the device to connect it to the platform. After the device is connected, you can navigate to view connection and event details.
Organization ID 3egoxc
Device Type NodeMCU
Device ID 12345
Authentication Method use-token-auth
Authentication Token 12345678
Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the device to generate a new authentication token.
Find out how to add these credentials to your device
ENG
IN
23:48
10-11-2022

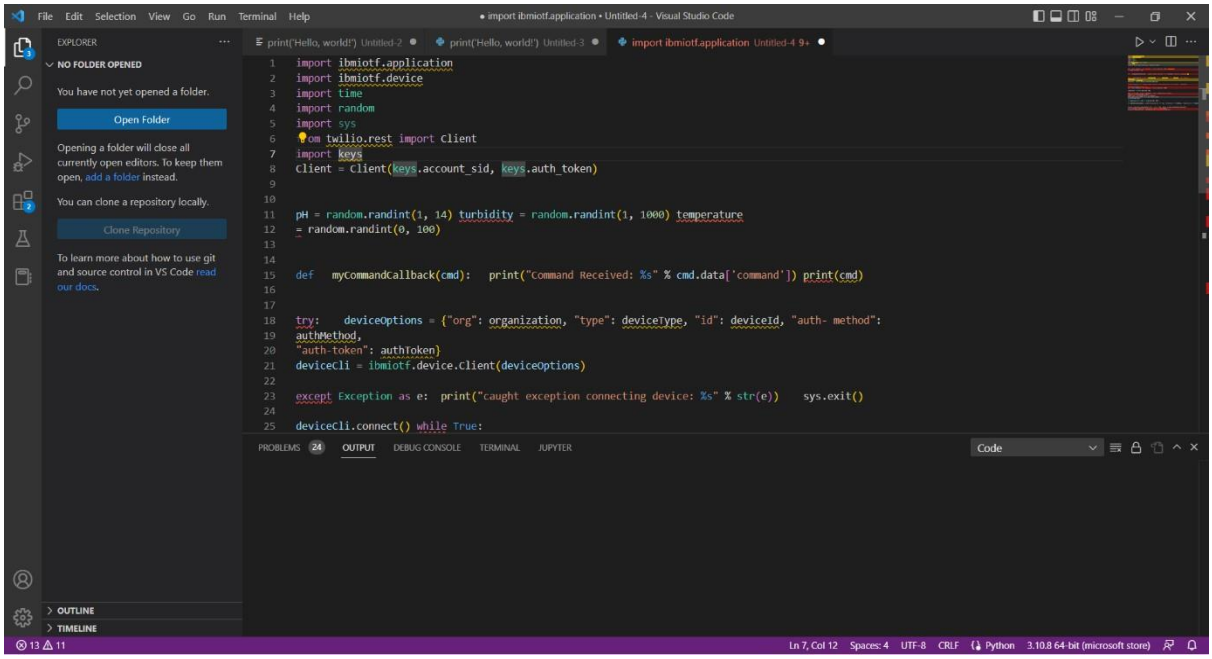
CREATE NODE-RED SERVICE

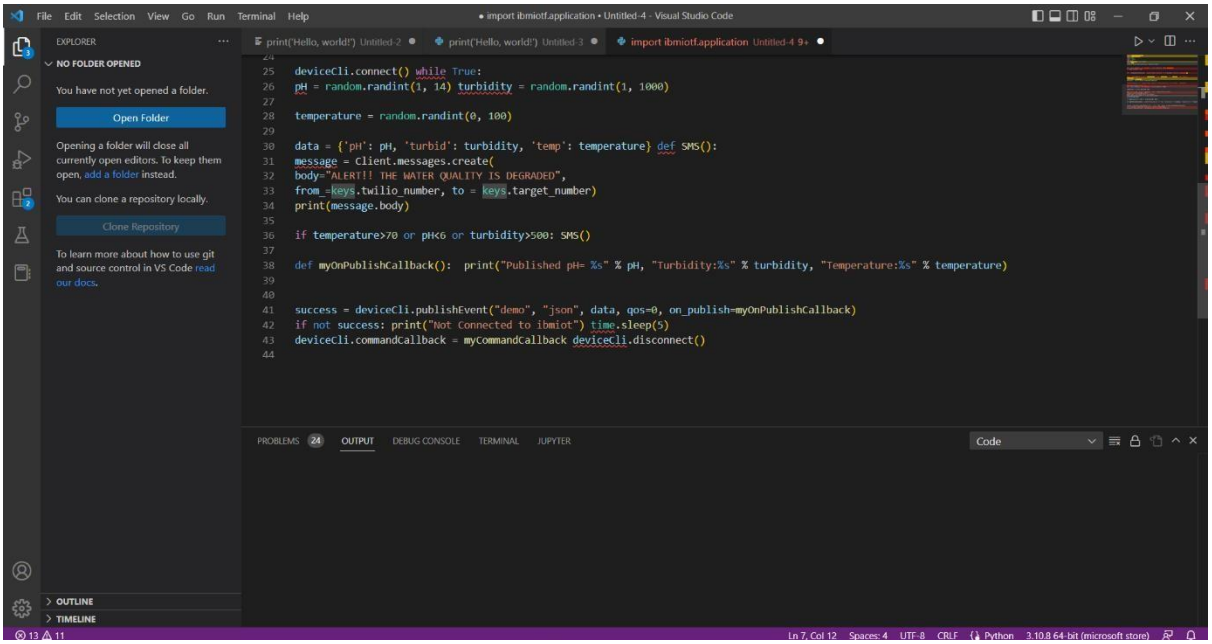


DEVELOP THE PYTHON SCRIPT

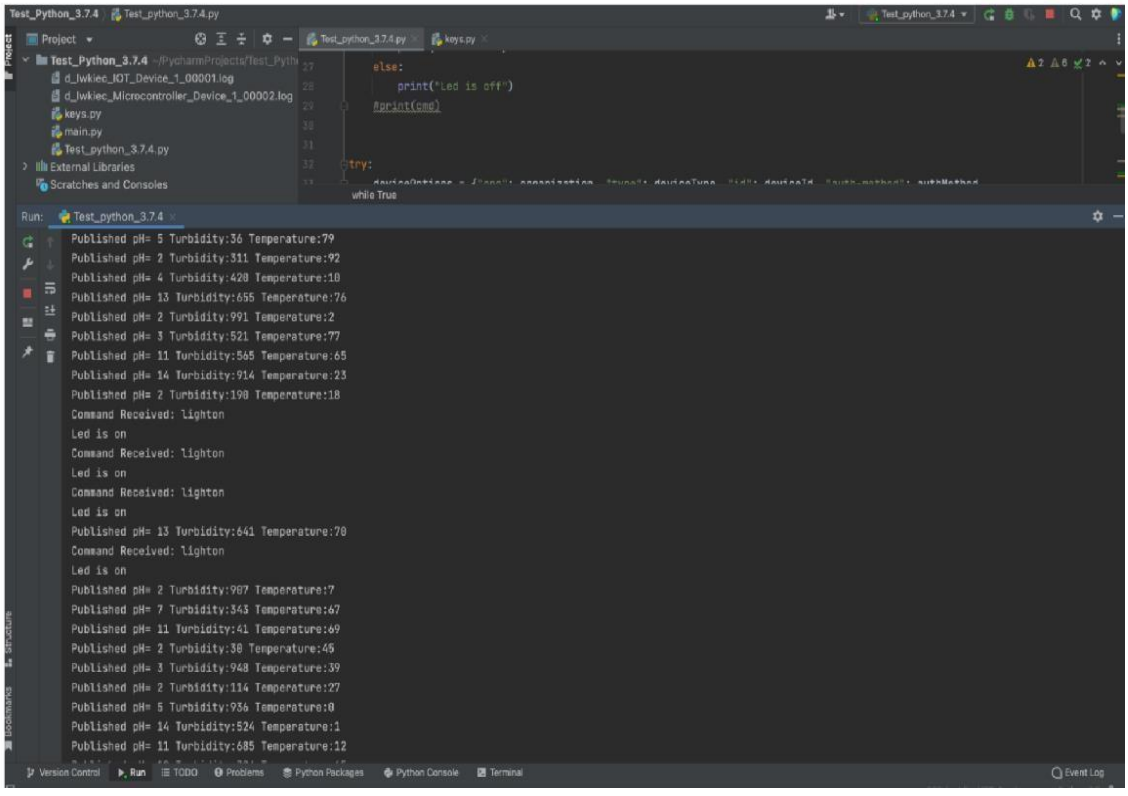
DEVELOP A PYTHON SCRIPT

CODING



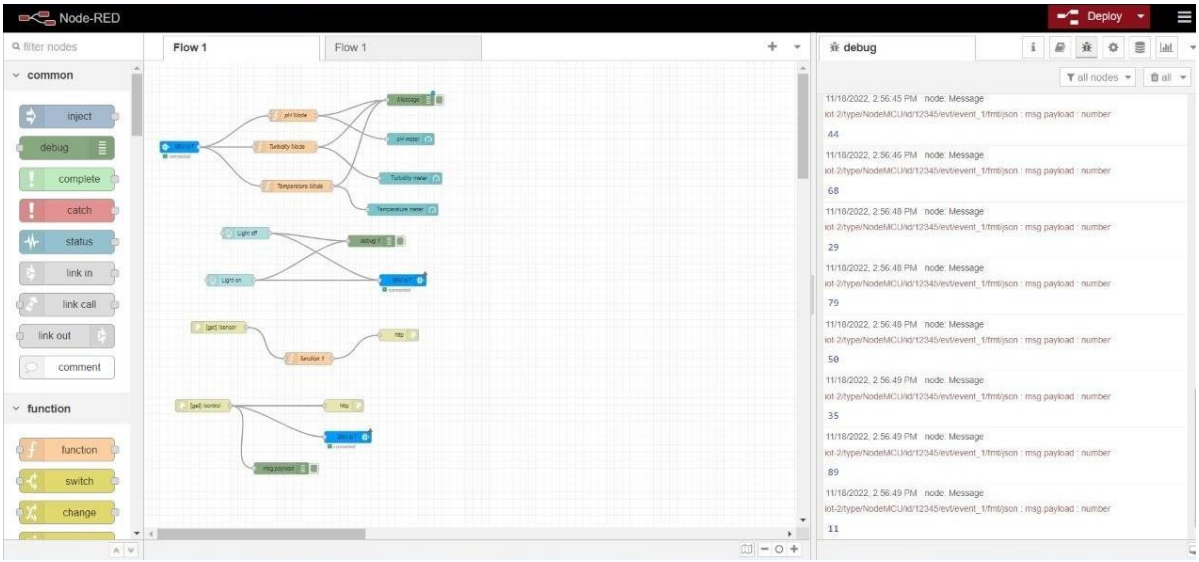

File Edit Selection View Go Run Terminal Help
• import ibmiotf.application • Untitled-4 - Visual Studio Code
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
You can clone a repository locally.
Clone Repository
To learn more about how to use git and source control in VS Code read our docs.
print('Hello, world!') Untitled-2
print('Hello, world!') Untitled-3
import ibmiotf.application Untitled-4 9+
deviceCli.connect() while True:
pH = random.randint(1, 14) turbidity = random.randint(1, 1000)
temperature = random.randint(0, 100)
data = {'pH': pH, 'turbid': turbidity, 'temp': temperature} def SMS():
message = Client.messages.create(
body="ALERT!! THE WATER QUALITY IS DEGRADED",
from_=keys.twilio_number, to = keys.target_number)
print(message.body)
if temperature>70 or pH<6 or turbidity>500: SMS()
def myOnPublishCallback(): print("Published pH= %s" % pH, "Turbidity:%s" % turbidity, "Temperature:%s" % temperature)
success = deviceCli.publishEvent("demo", "json", data, qos=0, on_publish=myOnPublishCallback)
if not success: print("Not Connected to ibmiot") time.sleep(5)
deviceCli.commandcallback = myCommandCallback deviceCli.disconnect()
PROBLEMS 24
OUTPUT
DEBUG CONSOLE
TERMINAL
JUPYTER
Code
OUTLINE
TIMELINE
Ln 7, Col 12 Spaces: 4 UTF-8 CRLF Python 3.10.8 64-bit (microsoft store)

OUTPUT

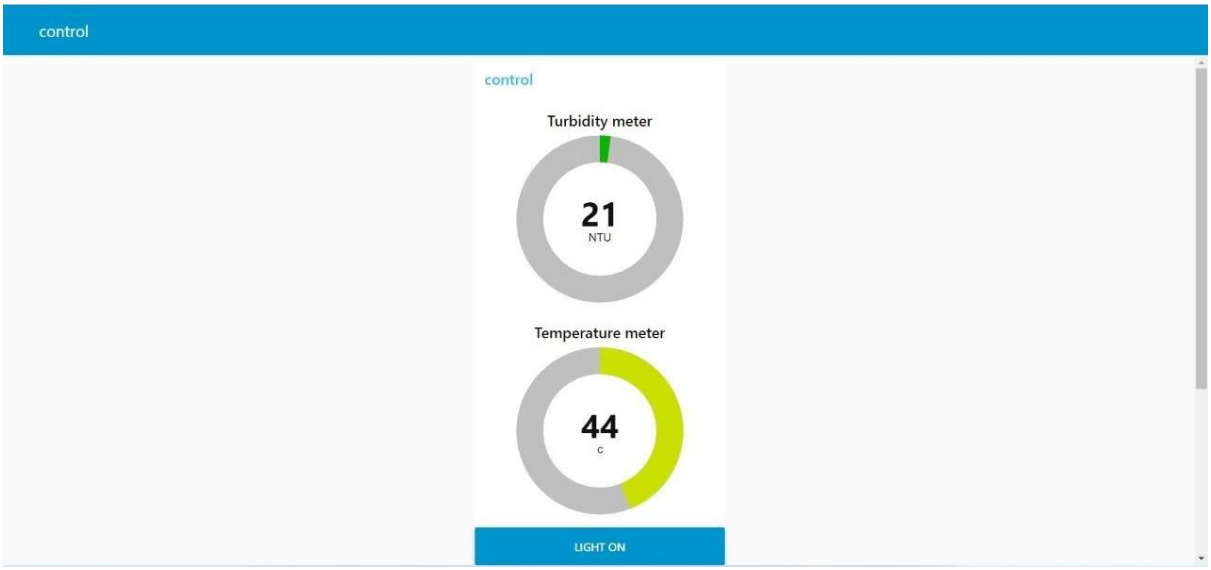

Test_Python_3.7.4
Test_python_3.7.4.py
Project
Test_Python_3.7.4
d_lwkiec_IOT_Device_1_00001.log
d_lwkiec_Microcontroller_Device_1_00002.log
keys.py
main.py
Test_python_3.7.4.py
External Libraries
Scratches and Consoles
else:
print("Led is off")
#print(cmd)
try:
while True
Run:
Test_python_3.7.4
Published pH= 5 Turbidity:36 Temperature:79
Published pH= 2 Turbidity:311 Temperature:92
Published pH= 4 Turbidity:428 Temperature:10
Published pH= 13 Turbidity:655 Temperature:76
Published pH= 2 Turbidity:991 Temperature:2
Published pH= 3 Turbidity:521 Temperature:77
Published pH= 11 Turbidity:565 Temperature:65
Published pH= 14 Turbidity:914 Temperature:23
Published pH= 2 Turbidity:190 Temperature:18
Command Received: lighton
Led is on
Command Received: lighton
Led is on
Command Received: lighton
Led is on
Published pH= 13 Turbidity:641 Temperature:70
Command Received: lighton
Led is on
Published pH= 2 Turbidity:907 Temperature:7
Published pH= 7 Turbidity:343 Temperature:67
Published pH= 11 Turbidity:41 Temperature:69
Published pH= 2 Turbidity:30 Temperature:45
Published pH= 3 Turbidity:948 Temperature:39
Published pH= 2 Turbidity:114 Temperature:27
Published pH= 5 Turbidity:936 Temperature:0
Published pH= 14 Turbidity:524 Temperature:1
Published pH= 11 Turbidity:685 Temperature:12
Version Control
Run
TODO
Problems
Python Packages
Python Console
Terminal
Event Log

DEVELOP A WEB APPLICATION USING NODE-RED SERVICES

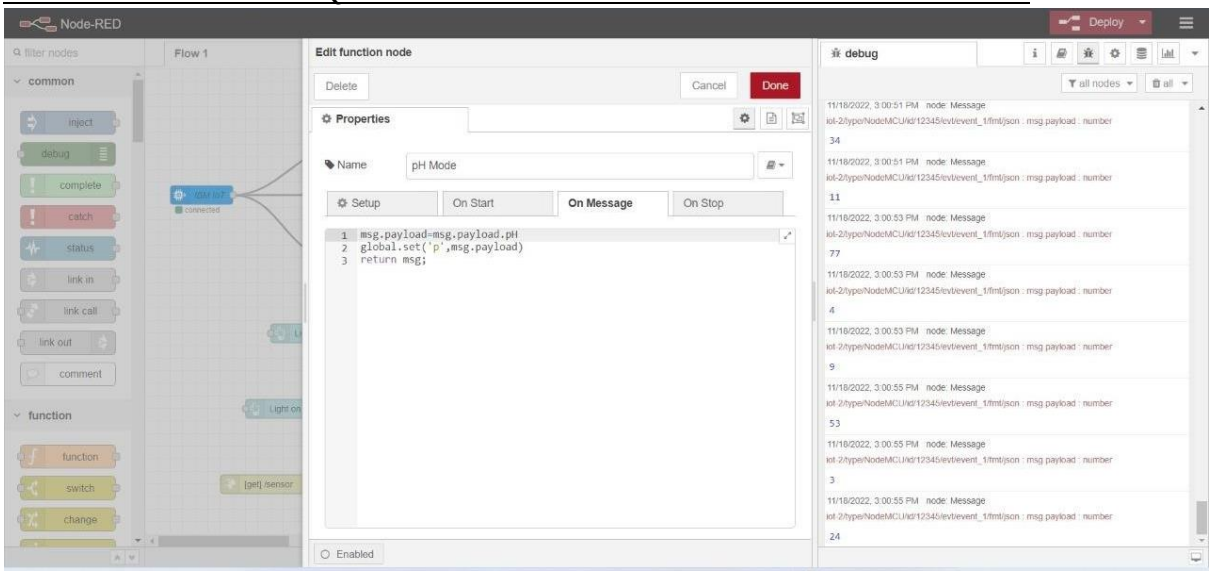
DEVELOP THE WEB APPLICATION USING NODE-RED



USING DASHBOARD NODES FOR CREATING UI(WEB APP)

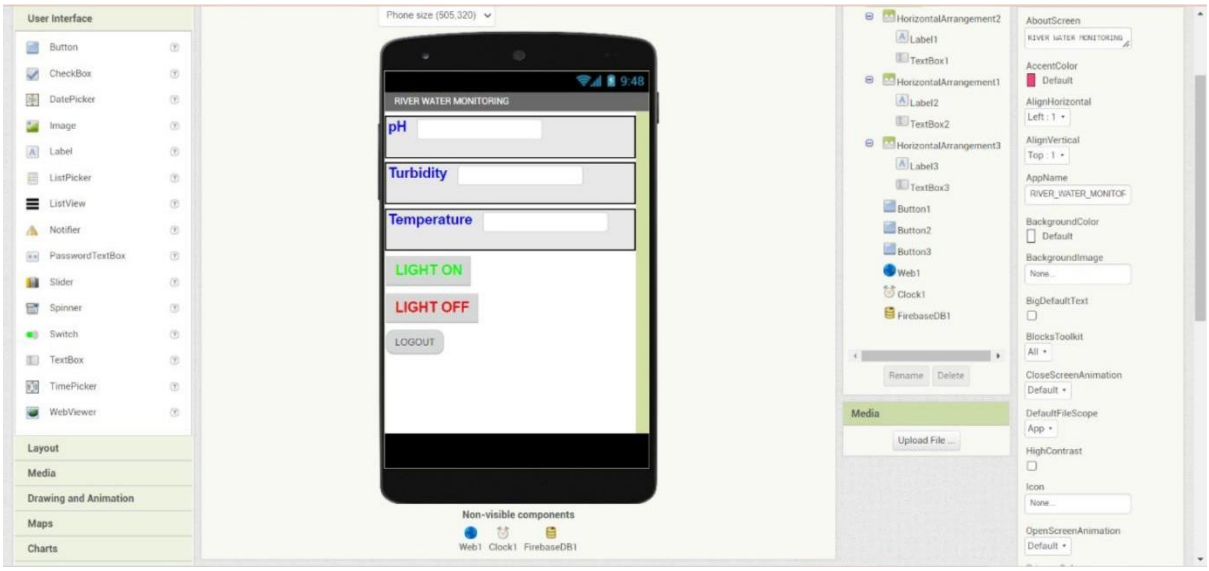


CREATE AN HTTP REQUESTS TO COMMUNICATE WITH MOBILE APP



BUILDING MOBILE APP

DESIGN YOUR UI TO DISPLAY THE WATER,TURBIDITY,PH VALUES



CONFIGURE THE APPLICATION TO RECEIVE THE DATA FROM CLOUD