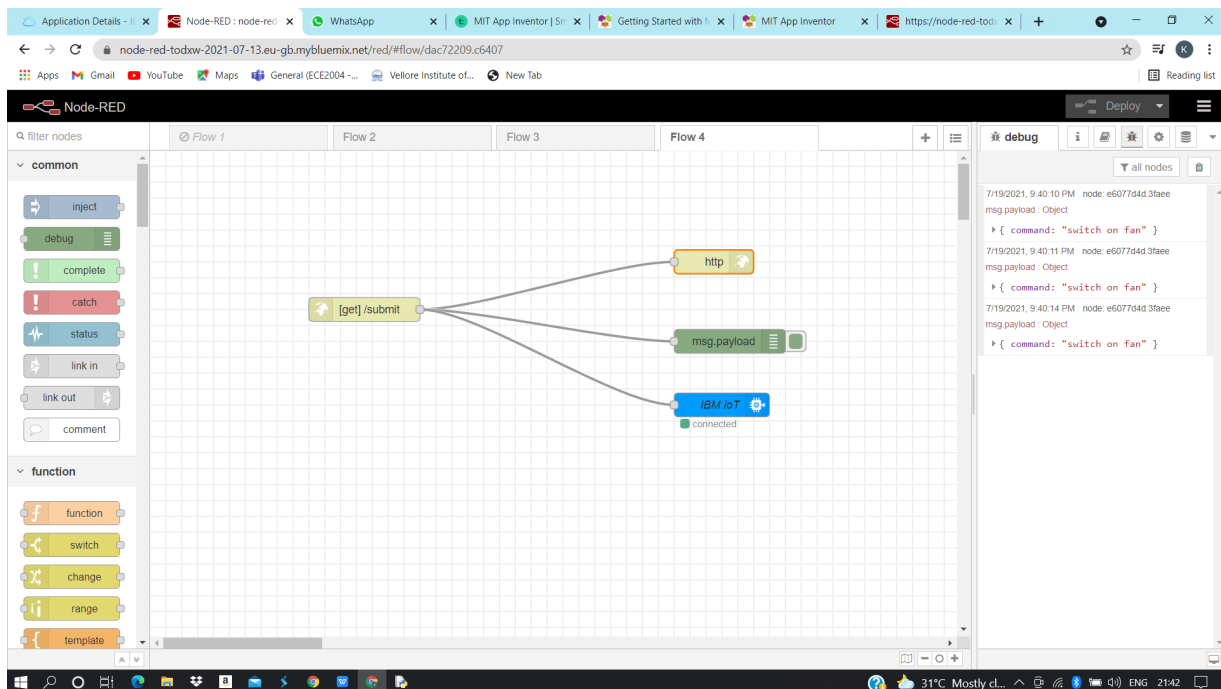


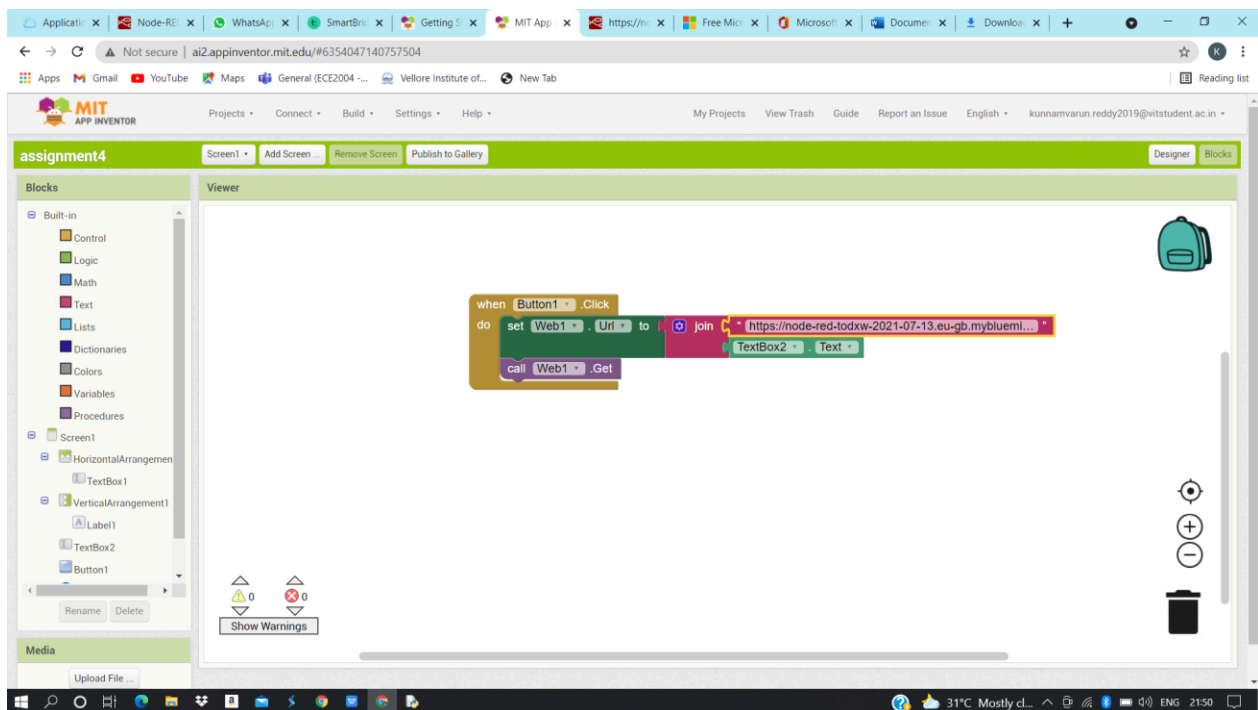
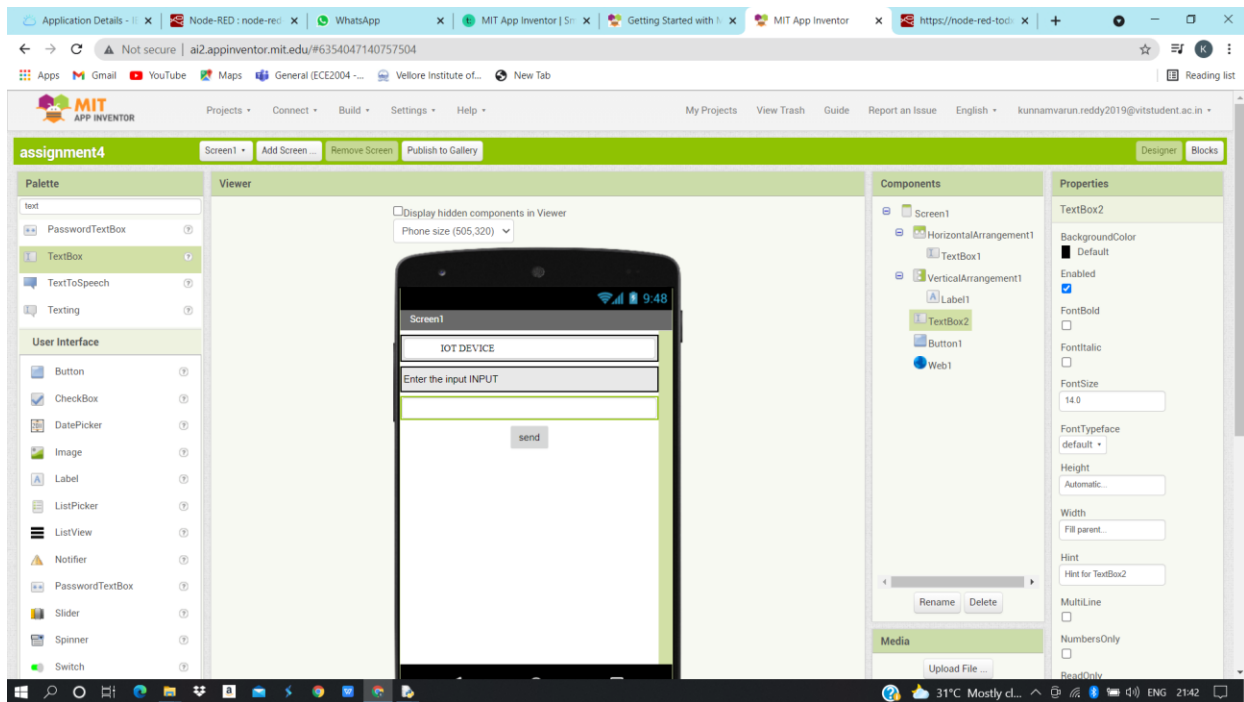
ASSIGNMENT-4

NAME : KUNNAM VARUN REDDY

KUNNAMVARUN.REDDY2019@VITSTUDENT.AC.IN

REG NO : 19BEC0279





PYTHON CODE :

```
import wiotp.sdk.device
```

```
import time
```

```
import random
```

```
myConfig = {  
    "identity": {  
        "orgId": "xmf5vy",  
        "typeId": "V1",  
        "deviceId": "9399996166"  
    },  
    "auth": {  
        "token": "9399996166"  
    }  
}
```

```
def myCommandCallback(cmd):
```

```
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
```

```
    print()
```

```
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
```

```
client.connect()
```

while True:

 client.commandCallback = myCommandCallback

 time.sleep(2)

client.disconnect()



The screenshot shows a code editor window titled "assignment4.py - C:/Users/NITHIN/AppData/Local/Programs/Python/Python39/assignment4.py (3.9.6)". The code is a Python script that interacts with the IBM IoT Platform. It imports the necessary modules and defines a configuration dictionary. A function named myCommandCallback is defined to handle incoming commands. The script then creates a DeviceClient, connects to the IoT Platform, and enters a while loop that sets the command callback, sleeps for 2 seconds, and disconnects.

```
assignment4.py - C:/Users/NITHIN/AppData/Local/Programs/Python/Python39/assignment4.py (3.9.6)
File Edit Format Run Options Window Help
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "xmfs5vy",
        "typeId": "v1",
        "deviceId": "9399996166"
    },
    "auth": {
        "token": "9399996166"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    print()

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    client.commandCallback = myCommandCallback
    time.sleep(2)
    client.disconnect()
```

Ln: 11 Col: 28

AFTER EXECUTION :

9:41 0.19 KB/s 97%

Screen1

IOT DEVICE

Enter the input INPUT

switch on fan

send

```
*IDLE Shell 3.9.6*
File Edit Shell Debug Options Window Help
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/NITHIN/AppData/Local/Programs/Python/Python39/assignment4.py =====
2021-07-19 21:38:21,450  wiotp.sdk.device.client.DeviceClient  INFO  Connected successfully: d:xmf5vy:Vl:9399996166
Message received from IBM IoT Platform: switch on fan
Message received from IBM IoT Platform: switch on fan
Message received from IBM IoT Platform: switch on fan
===== RESTART: C:/Users/NITHIN/AppData/Local/Programs/Python/Python39/assignment4.py =====
2021-07-19 21:40:20,911  wiotp.sdk.device.client.DeviceClient  INFO  Connected successfully: d:xmf5vy:Vl:9399996166
|
```

Application Details - x Node-RED : node-red x WhatsApp x MIT App Inventor | S x Getting Started with x MIT App Inventor x https://node-red-tod x +

node-red-todxw-2021-07-13.eu-gb.mybluemix.net/red/#flow/dac72209.c6407

Apps Gmail YouTube Maps General (ECE2004 +... Vellore Institute of... New Tab Reading list

Node-RED

filter nodes

Flow 1 Flow 2 Flow 3 Flow 4

common

- inject
- debug
- complete
- catch
- status
- link in
- link out
- comment

function

- function
- switch
- change
- range
- template

[get] /submit

http

msg.payload

IBM IoT

connected

debug

all nodes

```
7/19/2021, 9:40:10 PM node: e6077d4d.3faee  
msg.payload : Object  
  > { command: "switch on fan" }  
7/19/2021, 9:40:11 PM node: e6077d4d.3faee  
msg.payload : Object  
  > { command: "switch on fan" }  
7/19/2021, 9:40:14 PM node: e6077d4d.3faee  
msg.payload : Object  
  > { command: "switch on fan" }
```

31°C Mostly cl... ENG 21:42