

SPRINT - 4

Date	13 NOV 2022
Team ID	PNT2022TMID53567
Project Name	Smart Waste Management System for Metropolitan Cities

1, Simulate python code in Python IDE software to transmit data to IBM Watson IOT platform

Python code:

smartbin.py:

```
#Project: Smart Waste Management System for Metropolitan cities  
#Team ID: PNT2022TMID53567
```

```
#Installing necessary libraries
```

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

```
import time
```

```
import random
```

```
import requests
```

```
import math
```

```
#Configuration details for connecting python script to IBM Watson IOT Platform
```

```
myConfig = {
```

```
"identity": {
```

```
"orgId": "mldk59",
```

```
"typeId": "pythoncode",
```

```
"deviceId": "252525"
```

```
},
```

```
"auth": {
```

```
"token": "QZqODYo6U*Q6b+IpuC"
```

```
} }
```

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

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

```
    m=cmd.data['command']
```

#Connecting the client to ibm watson iot platform

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

#Generate Random values for latitude, longitude in a circular distribution from the current location and

#alert the garbage collector to go to the particular location where the bin level and bin weight exceeds the threshold

while True:

```
    res = requests.get('https://ipinfo.io/')
    data = res.json()
    loc = data['loc'].split(',')
    theta = random.uniform(0,2*math.pi)
    area = (0.05**2)*math.pi
    radius = math.sqrt(random.uniform(0,area/math.pi))
    latitude,longitude = [float(loc[0])+radius*math.cos(theta), float(loc[1])+radius*
    math.sin(theta)]

    binlevel=random.randint(10,100)
    binweight = random.randint(50,1500)

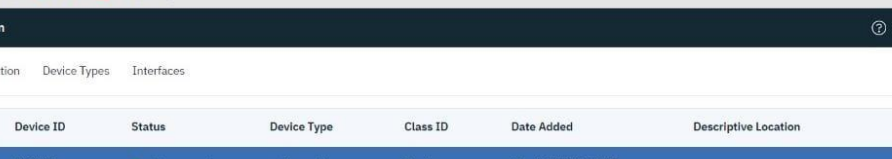
    if binweight>=1000 and binlevel>80:
        myData={'latitude':latitude, 'longitude':longitude,'binlevel':binlevel,
                'binweight':binweight}
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=
0, onPublish=None)
        ##print("Published data Successfully: %s", myData)
        print("BIN IS FULL..TIME TO EMPTY IT!!!!\n",myData)
        client.commandCallback = myCommandCallback
        time.sleep(2)
        #break

    else :
        print("BIN IS IN NORMAL LEVEL...")
        time.sleep(2)
```

Python IDE output:

2. Data is transferred to IBM Watson IoT platform.

IBM Platform output:

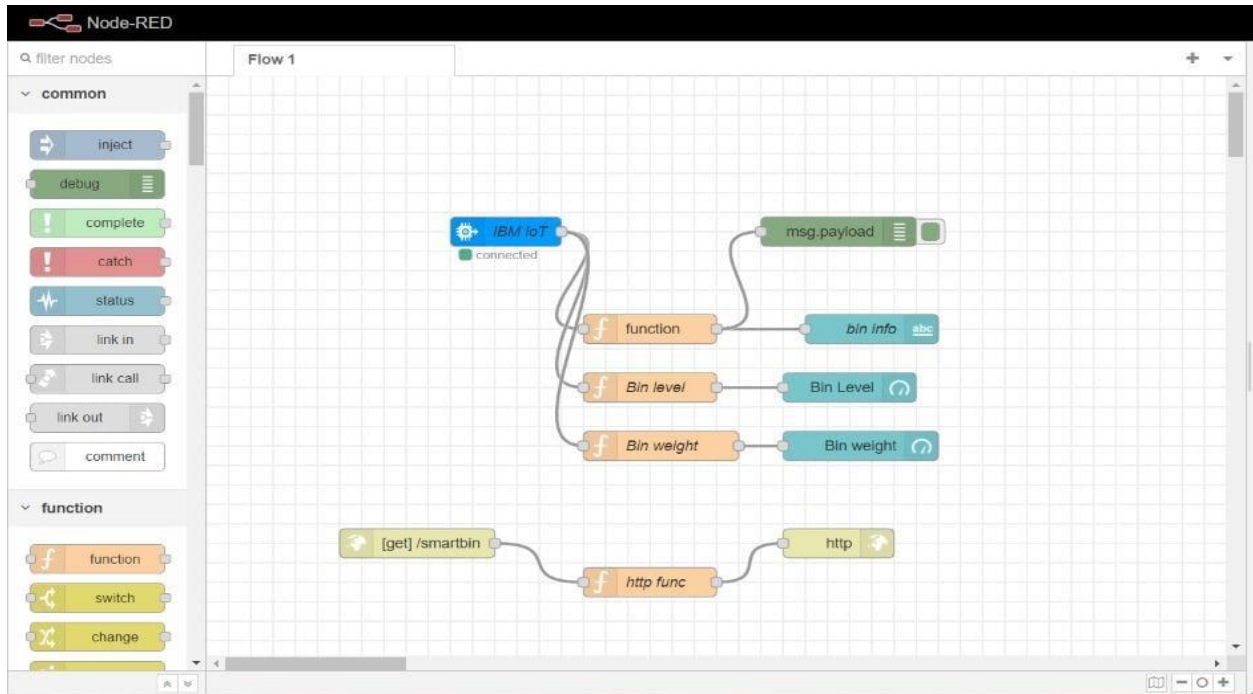


The screenshot shows the IBM Watson IoT Platform interface. At the top, there's a navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is also present. Below this, a table lists devices. The selected device is '252525', which is 'Connected' and has a 'pythoncode' device type. Below the device table, there's a section for 'Recent Events'. It contains a text block stating: 'The recent events listed show the live stream of data that is coming and going from this device.' Below this text is a table of recent events.

Event	Value	Format	Last Received
status	{"latitude":93.38291147072071,"longitude":85....	json	a few seconds ago
status	{"latitude":123.34633147794314,"longitude":7...	json	a few seconds ago
status	{"latitude":120.69034946242466,"longitude":9...	json	2 minutes ago
status	{"latitude":82.92484862339958,"longitude":93....	json	2 minutes ago
status	{"latitude":44.682952261624024,"longitude":9...	json	2 minutes ago

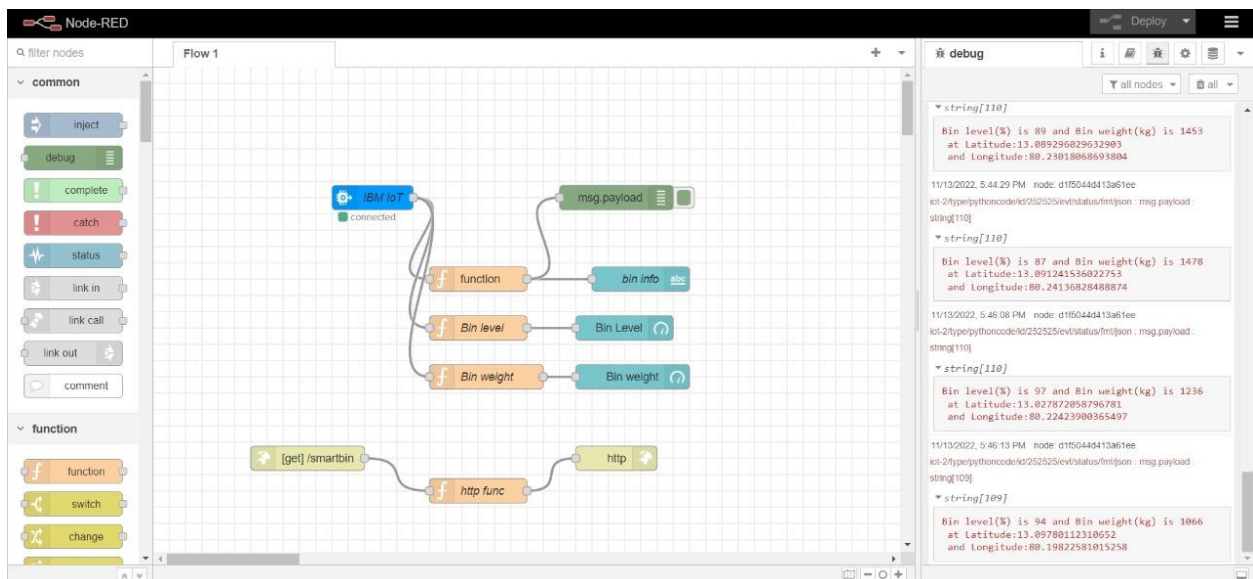
3. Data transfer from IBM Watson IOT platform and Python IDE to Node RED.

Node-RED:

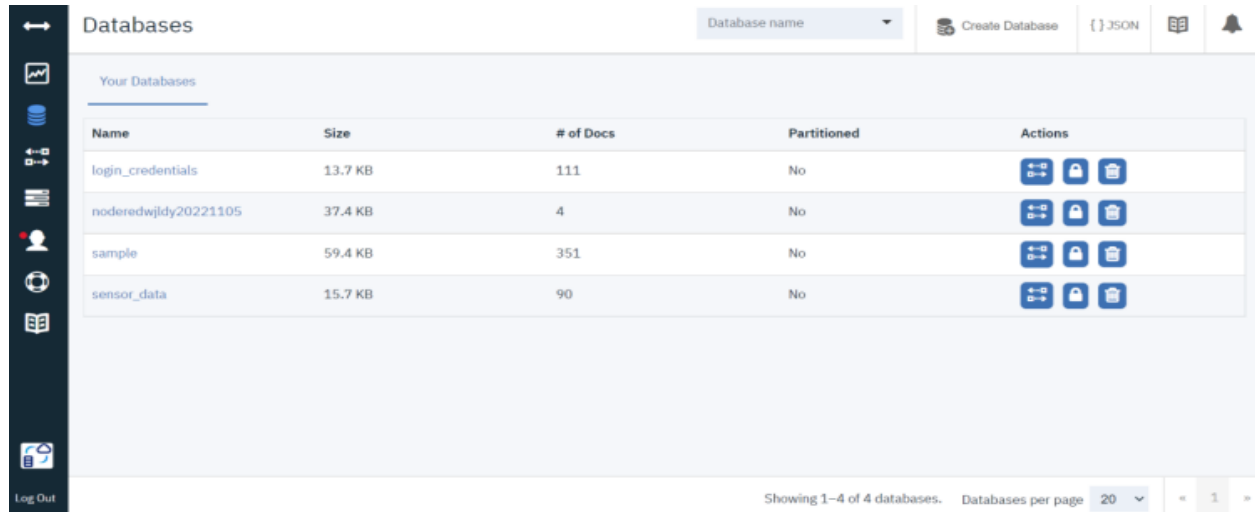


4. Node-RED Connection setup for data transmission from IBM Watson IoT platform to Node-RED dashboard and viewing in Web UI .

Node-RED:



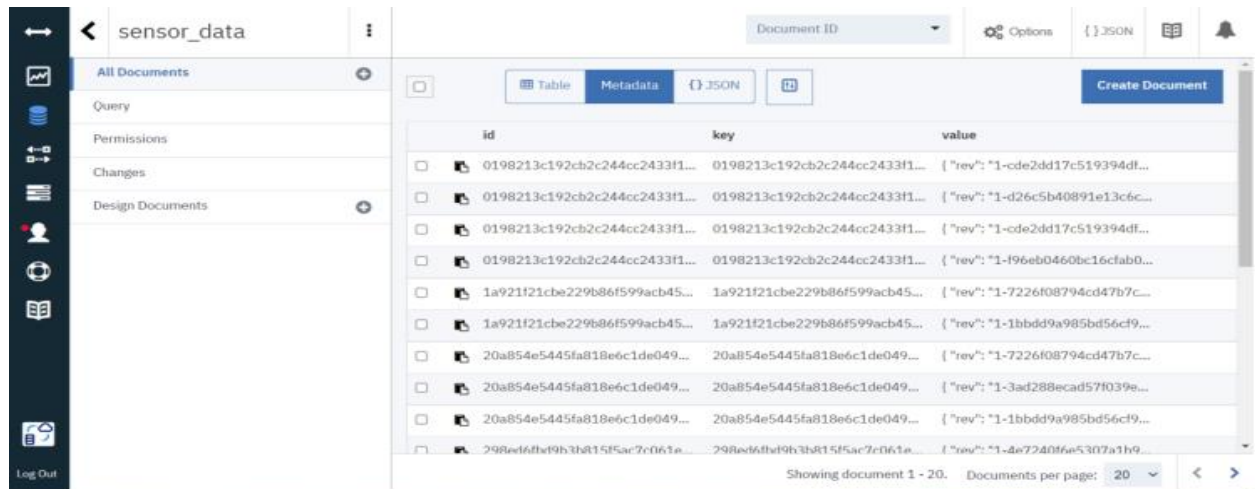
5. Storing database in IBM Cloudant DB



The screenshot shows the IBM Cloudant 'Databases' page. At the top, there's a search bar for 'Database name' and buttons for 'Create Database', 'JSON', and a notification bell. Below this is a section titled 'Your Databases' containing a table with the following data:

Name	Size	# of Docs	Partitioned	Actions
login_credentials	13.7 KB	111	No	[Icons for edit, lock, delete]
noderedwjldy20221105	37.4 KB	4	No	[Icons for edit, lock, delete]
sample	59.4 KB	351	No	[Icons for edit, lock, delete]
sensor_data	15.7 KB	90	No	[Icons for edit, lock, delete]

At the bottom, it says 'Showing 1-4 of 4 databases. Databases per page: 20'.



The screenshot shows the 'sensor_data' database document view. On the left is a sidebar with 'All Documents', 'Query', 'Permissions', 'Changes', and 'Design Documents'. The main area has tabs for 'Table', 'Metadata', 'JSON', and 'Table' (selected). A 'Create Document' button is in the top right. The table displays a list of documents with columns 'id', 'key', and 'value'. The first document is expanded, showing its JSON content:

```
{
  "_id": "0198213c192cb2c244cc2433f1802b91",
  "_rev": "1-cde2dd17c519394df...",
  "topic": "iot-2/type/SWMSMC/id/ibmproject/evt/data/fmt/json",
  "payload": {
    "Warning!!": "244.971e1f"
  },
  "deviceId": "ibmproject",
  "deviceType": "SWMSMC",
  "eventType": "data",
  "format": "json"
}
```

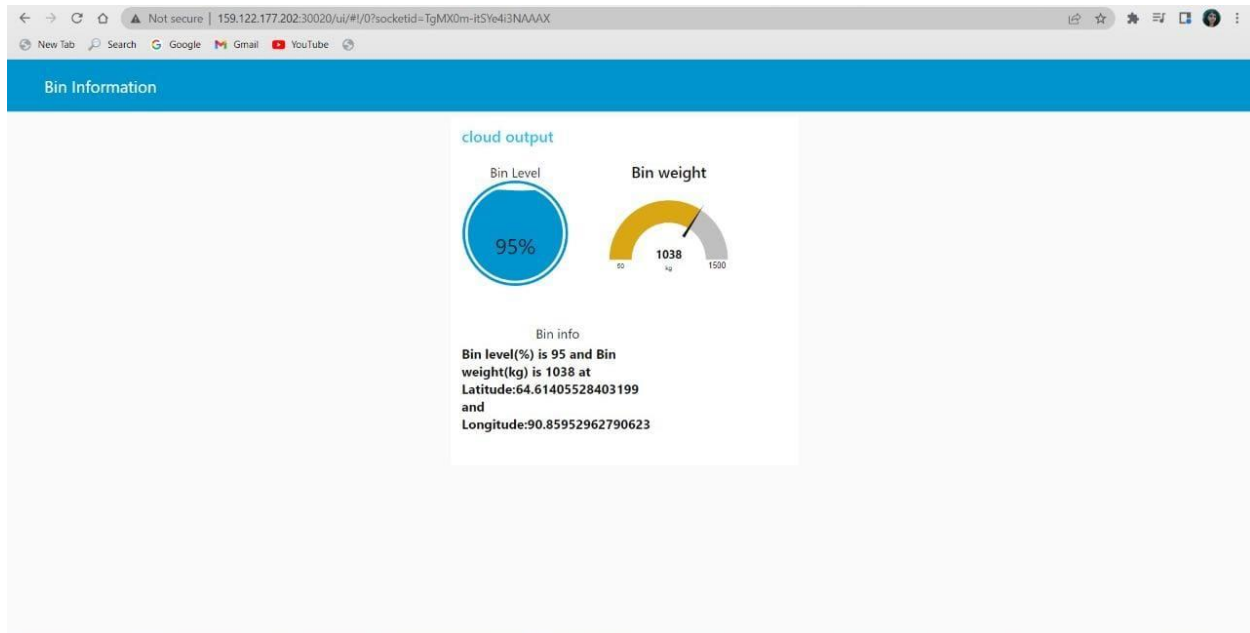
At the bottom, it says 'Showing document 1 - 20. Documents per page: 20'.

6. Data is stored in JSON format



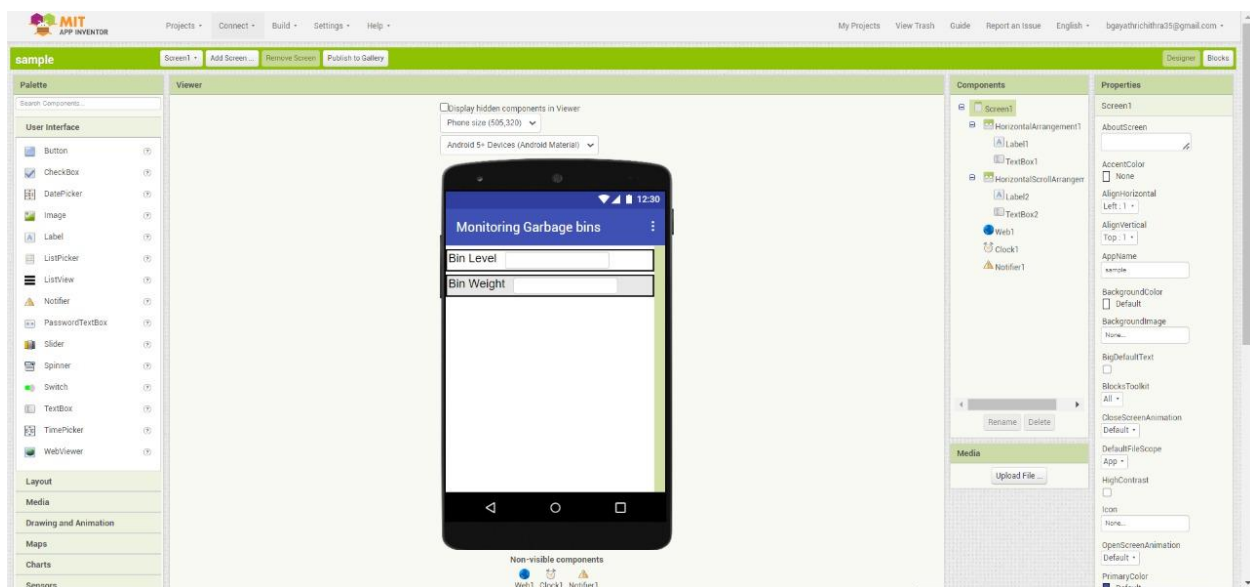
The screenshot shows the document editor for the 'sensor_data' database. At the top, there's a breadcrumb 'sensor_data > 0198213c192cb2c244cc2433f1802b91' and buttons for 'JSON', a notification bell, 'Save Changes', 'Cancel', 'Upload Attachment', 'Clone Document', and 'Delete'. The main area shows the JSON content of the document, which is the same as the one shown in the previous screenshot.

Web UI:

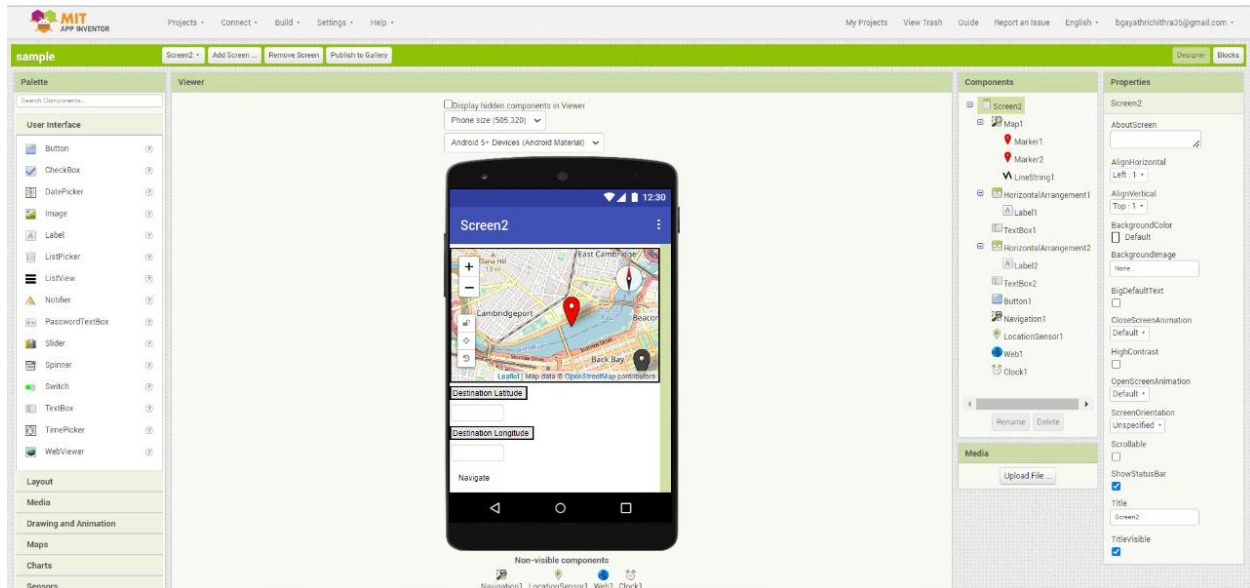


7. App is created using MIT App inventor

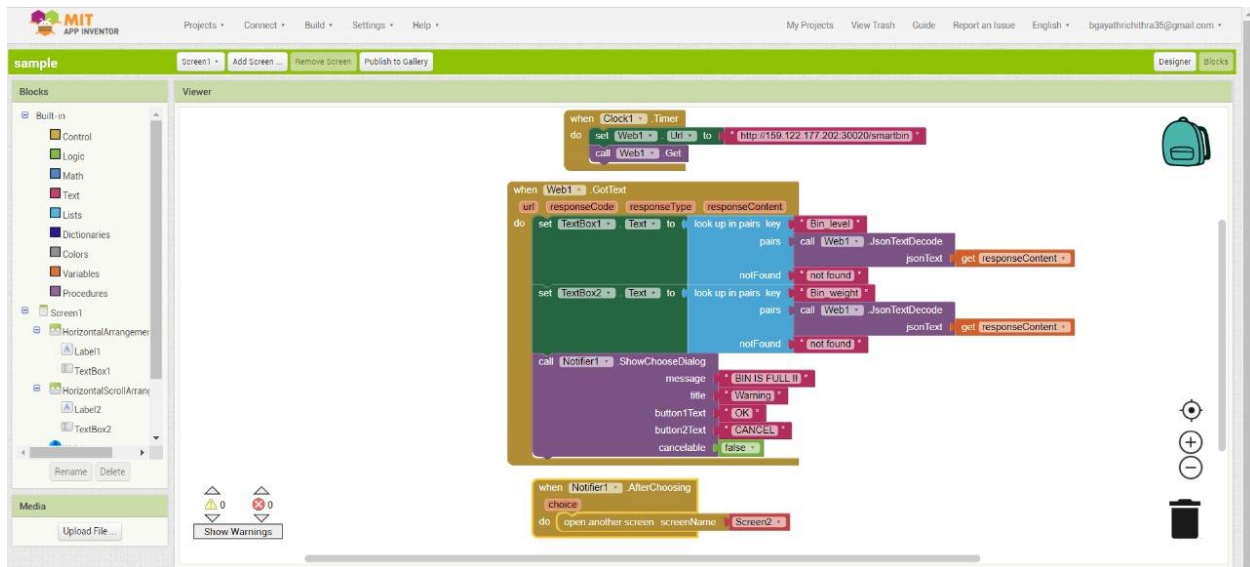
Screen 1:



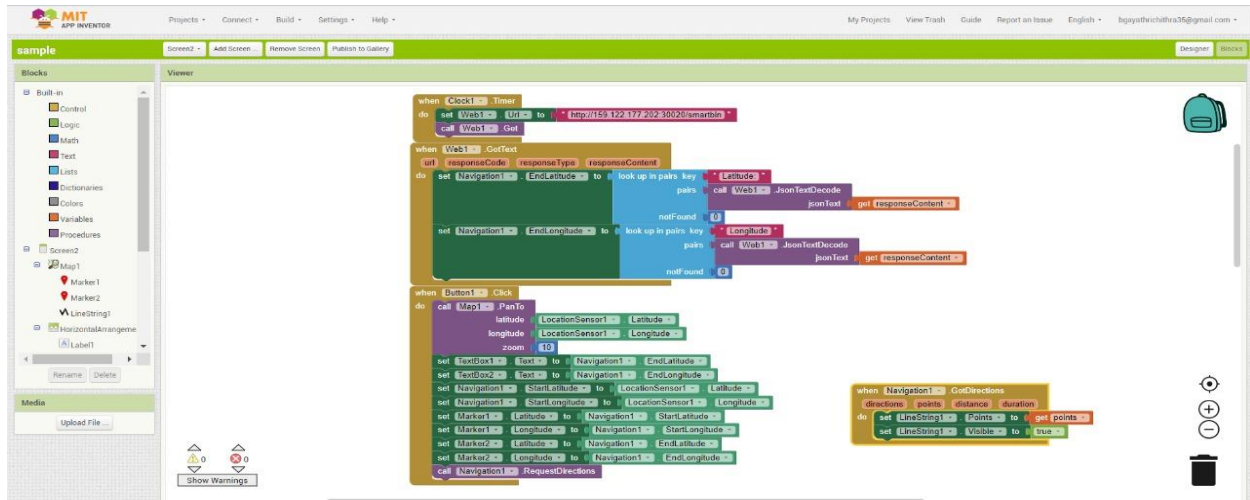
Screen 2:



Screen 1 blocks:



Screen 2 blocks:



8. Install MIT AI2 Companion in phone and scan the QR code showed in AI connect

