

# PROJECT DEVELOPMENT PHASE

## DELIVERY OF - SPRINT 3

TEAM ID	PNT2022TMID06962
PROJECT NAME	Smart Waste managementsystem for metropolitan cities

## NODE RED FUNCTION

The screenshot displays the IBM Cloud 'Resource list' interface. The browser address bar shows 'https://cloud.ibm.com/resources'. The page header includes the IBM Cloud logo, a search bar, and navigation links for 'Catalog', 'Manage', and 'IYAPPAN V's Account'. A 'Create resource' button is visible in the top right corner.

The main content area is titled 'Resource list' and contains a table with the following columns: Name, Group, Location, Product, Status, and Tags. The table lists several resource categories, each with a count in parentheses: Compute (1), Containers (0), Networking (0), Storage (0), AI / Machine Learning (0), Analytics (0), Blockchain (0), Databases (2+), Developer tools (5+), Logging and monitoring (0), and Migration (0).

Under the 'Compute (1)' category, a single resource is listed: 'Node RED DECVR 2022-11-12'. The details for this resource are as follows:

Name	Group	Location	Product	Status	Tags
Node RED DECVR 2022-11-12	DGGI / IYAPPAN V	London	Node.js	Started	—

The bottom of the screenshot shows the Windows taskbar with the search bar, taskbar icons, and system tray information including the date and time (18-11-2022, 06:08).

IBM Cloud

Node RED DECVR 2022-11-12 Running [Visit App URL](#) [Add tags](#) [Details](#) [Actions...](#)

Getting started

**Overview**

Runtime

Connections

Logs

API Management

Autoscaling

**Instances**

Health

100%

1/1 instance(s) are running

MB memory per instance

0 2048 256

**Runtime**

Node.js

256

Total MB allocation

1.75 GB still available

Free Used

**Runtime cost**

Current and estimated cost excludes connected services.

\$ 0.00 \$ 0.00

Estimated total for billing period

**Connections (1)**

[node-red-wgttm-2022--cloudant-1668220095820-72328](#)

Node-RED on IBM Cloud

# Node-RED

Flow-based programming for the Internet of Things

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

More information about Node-RED, including documentation, can be found at [nodered.org](https://nodered.org).

[Go to your Node-RED flow editor](#)

[Learn how to customise Node-RED](#)

Customising your instance of Node-RED

# NODE RED IN RANDOM FUNCTION

This screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow with an 'IBM IoT' node connected to three function nodes labeled 'temp', 'hum', and 'random'. The 'Edit ibmiot in node' panel is open on the right, showing the following configuration:

- Authentication:** API Key
- API Key:** API key
- Input Type:** Device Event
- Device Type:** All or +
- Device Id:** All or device id e.g. ab12cd231a21
- Event:** All or +
- Format:** All or json
- QoS:** 0
- Name:** IBM IoT
- Service:** registered

A yellow tooltip at the bottom of the panel reads: "Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to". The 'Enabled' checkbox is checked.

This screenshot shows the same Node-RED interface, but the 'Edit function node' panel is open for the 'temp' function node. The configuration is as follows:

- Name:** temp
- Setup:** (selected tab)
- On Start:** (empty)
- On Message:** (empty)
- On Stop:** (empty)

The code editor shows the following JavaScript code:

```
1 msg.payload = msg.payload.temp
2 return msg;
```

The 'Enabled' checkbox is checked.

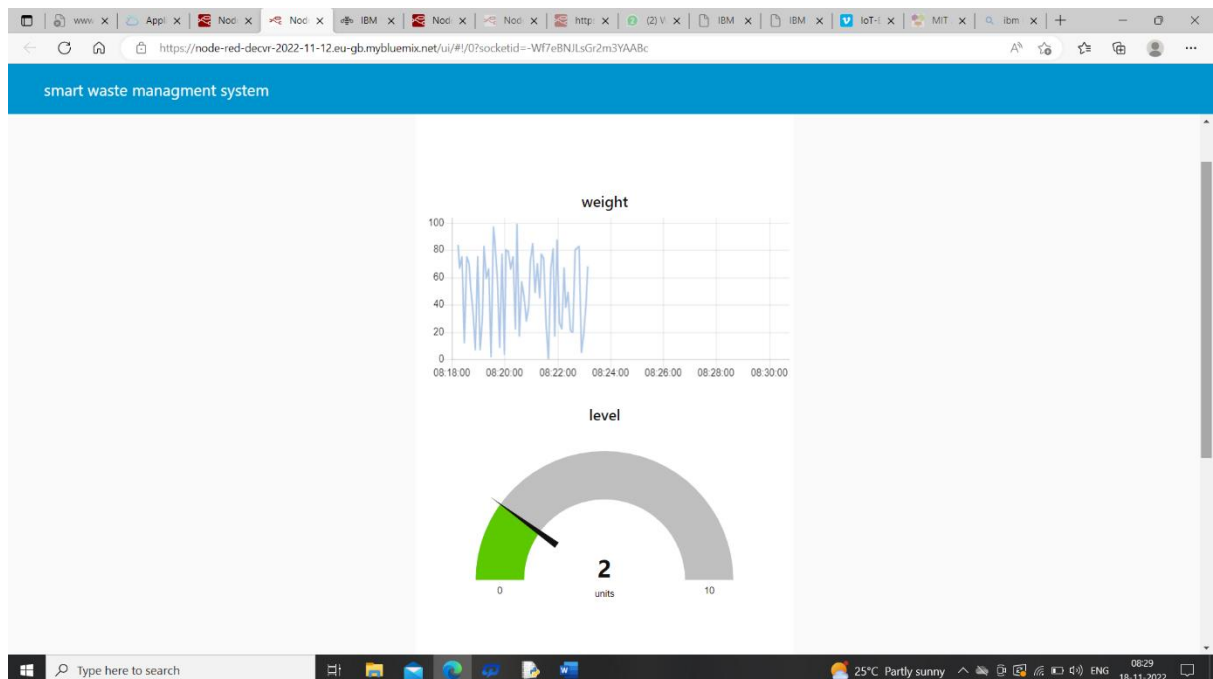
Node-RED interface showing a flow diagram and a debug console.

**Flow Diagram:**

- Input:** IBM IoT (connected)
- Processing:** The flow splits into two parallel paths. The top path contains a **weight** node, and the bottom path contains a **level** node. Both paths then merge into a **msg.payload** node.
- Output:** The merged flow connects to two output nodes: **weight** and **level**.

**Debug Console:**

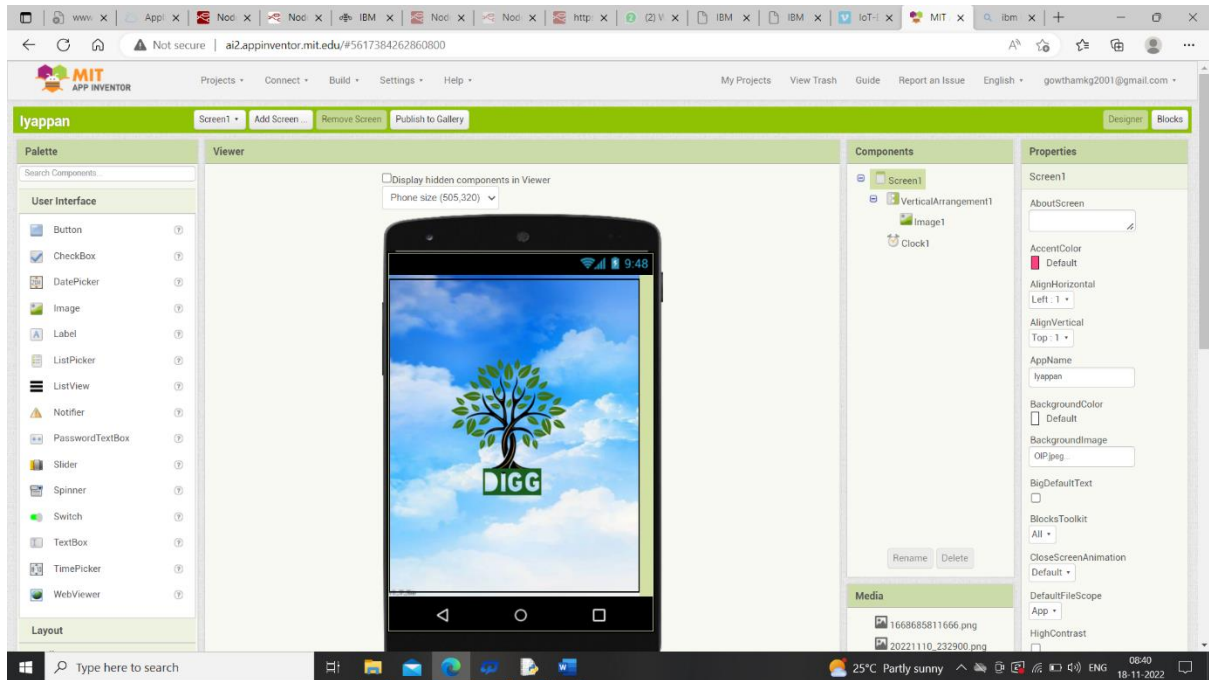
```
undefined
18/11/2022, 8:27:56 am node: f1a2be5d840c2b7a
iot-2?type=DGGId%1234%evl%yappan%mt%json :
msg.payload : Object
  { randomNumber: 35, temp: 91, hum: 38 }
18/11/2022, 8:27:56 am node: f1a2be5d840c2b7a
iot-2?type=DGGId%1234%evl%yappan%mt%json :
msg.payload : undefined
undefined
18/11/2022, 8:27:56 am node: f1a2be5d840c2b7a
iot-2?type=DGGId%1234%evl%yappan%mt%json :
msg.payload : undefined
undefined
18/11/2022, 8:28:02 am node: f1a2be5d840c2b7a
iot-2?type=DGGId%1234%evl%yappan%mt%json :
msg.payload : Object
  { randomNumber: 93, temp: 103, hum: 40 }
18/11/2022, 8:28:02 am node: f1a2be5d840c2b7a
iot-2?type=DGGId%1234%evl%yappan%mt%json :
msg.payload : undefined
undefined
18/11/2022, 8:28:02 am node: f1a2be5d840c2b7a
iot-2?type=DGGId%1234%evl%yappan%mt%json :
msg.payload : undefined
undefined
```



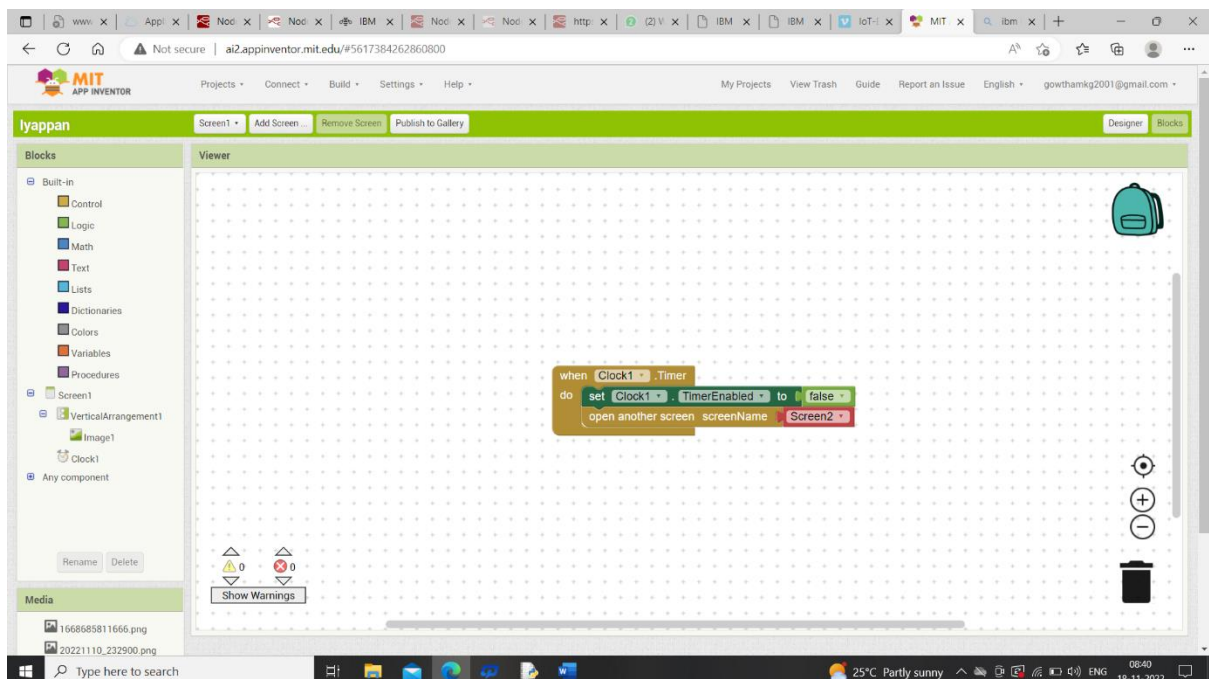
# MIT APP INVENTOR

## SCREEN 1:

## Design:

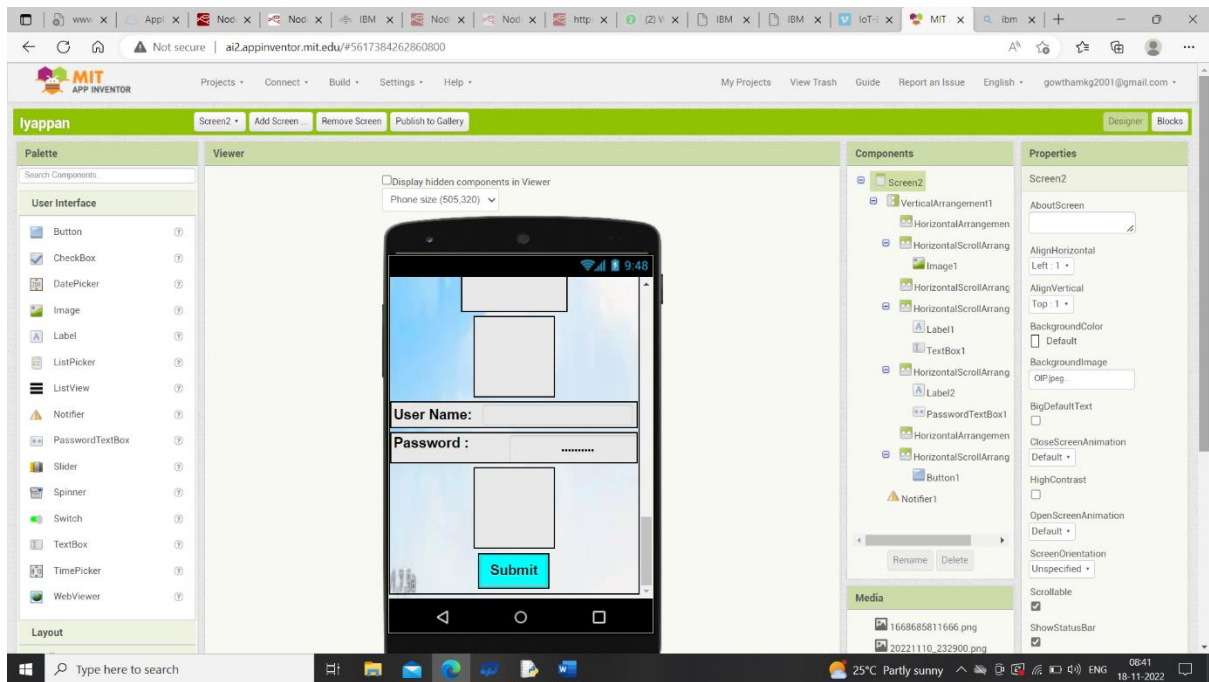


## Blocks:

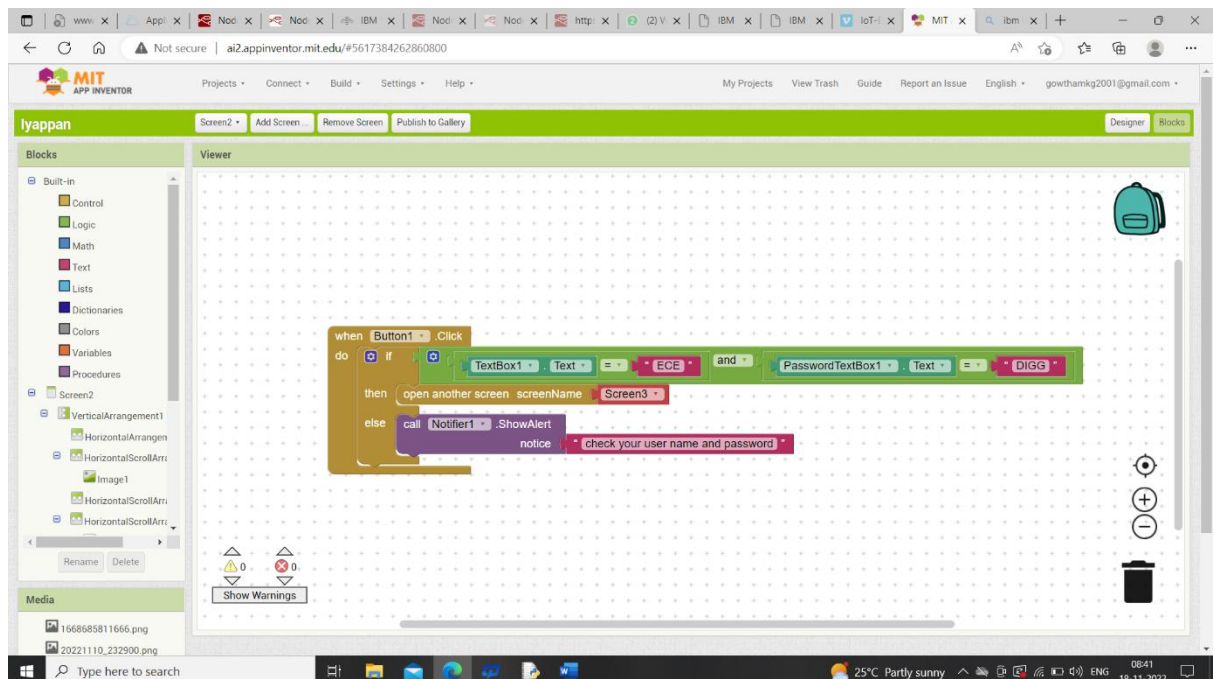


# SCREEN 2

## Design



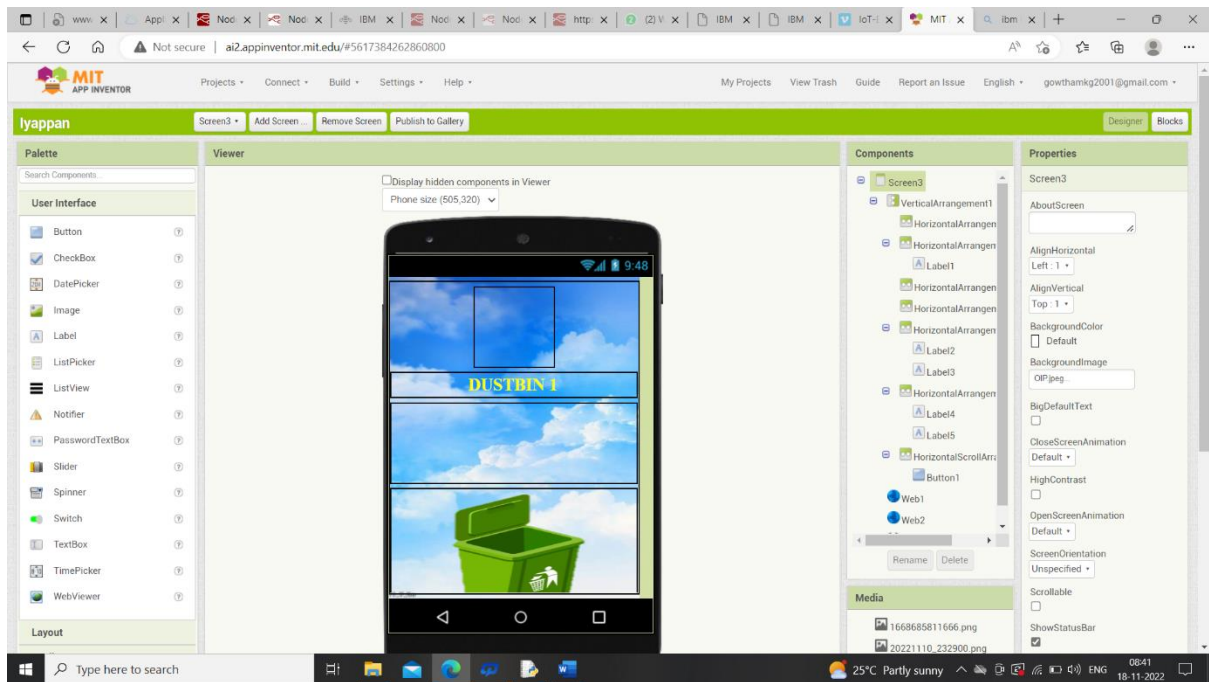
## Blocks:



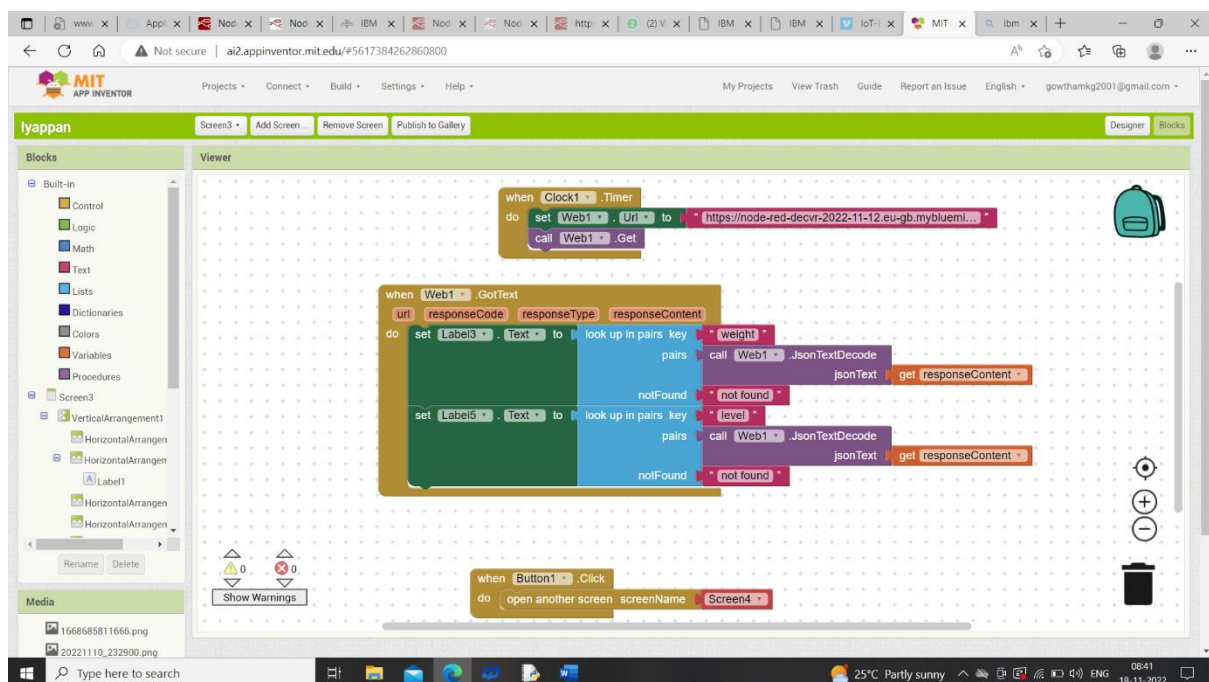


# SCREEN 3

## Design:

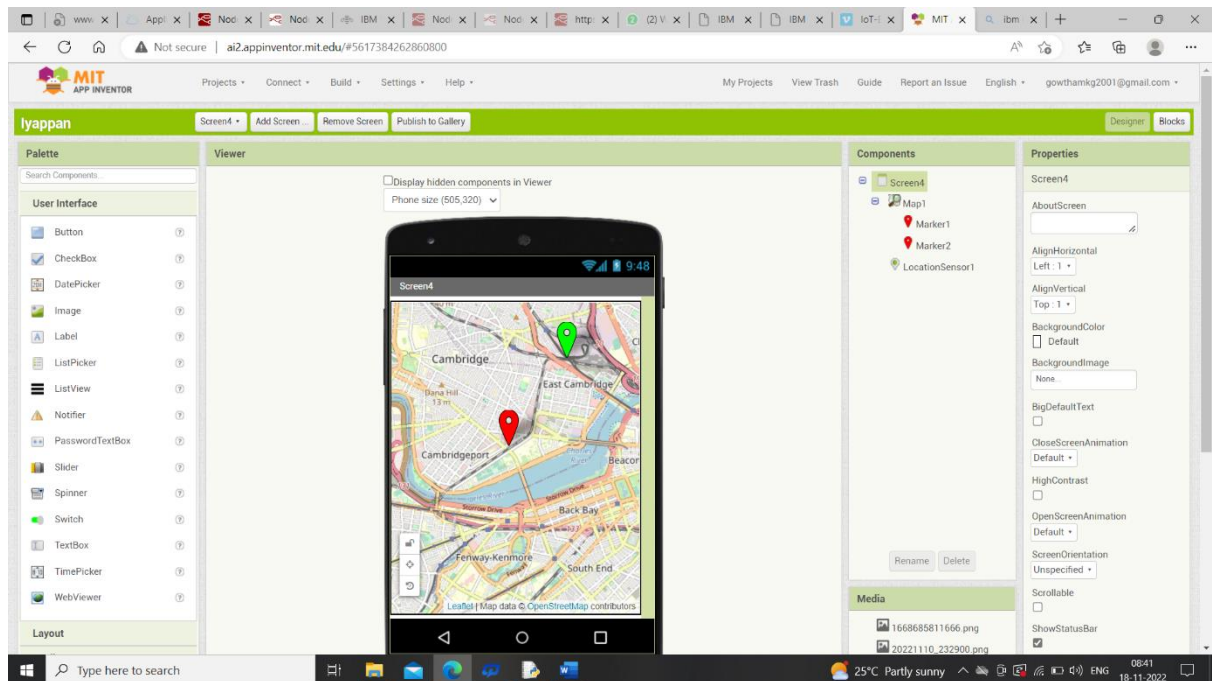


## Blocks:

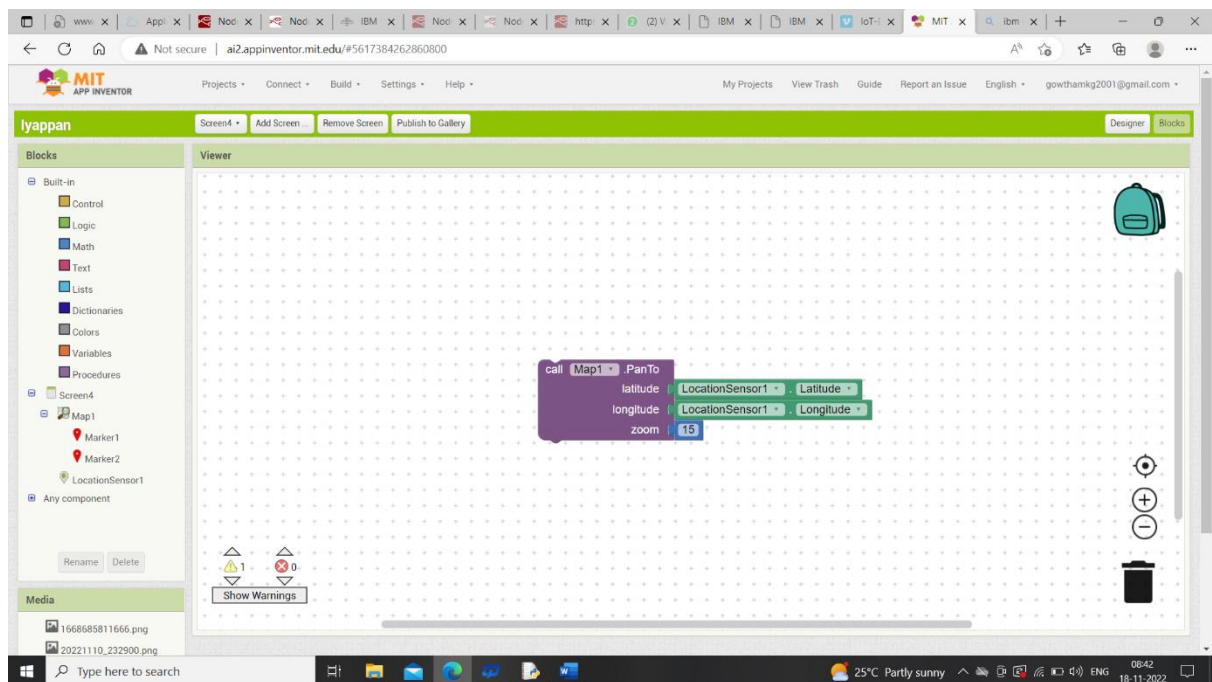


# SCREEN 4

## Design:

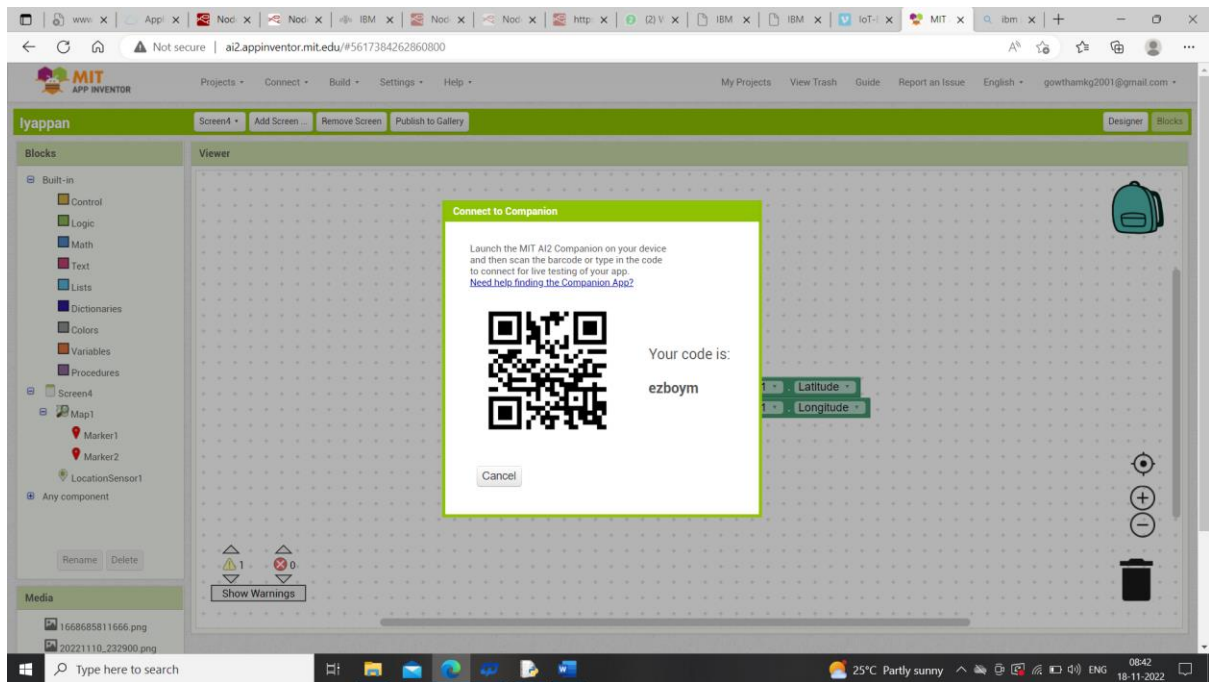


## Blocks:





# APP COMPANAION



# QR CODE

