

Develop the Web Application using Node-Red

Team ID	PNT2022TMID04593
Project Name	Smart waste management system for metropolitan cities

Step 1: In Backend NODE – RED output is fetched to MIT APP INVENTOR

Step2: In MIT APP INVENTOR design the output screen layout using various tools available in the platform

Step 3: Then click on connect then click on AI COMPANION

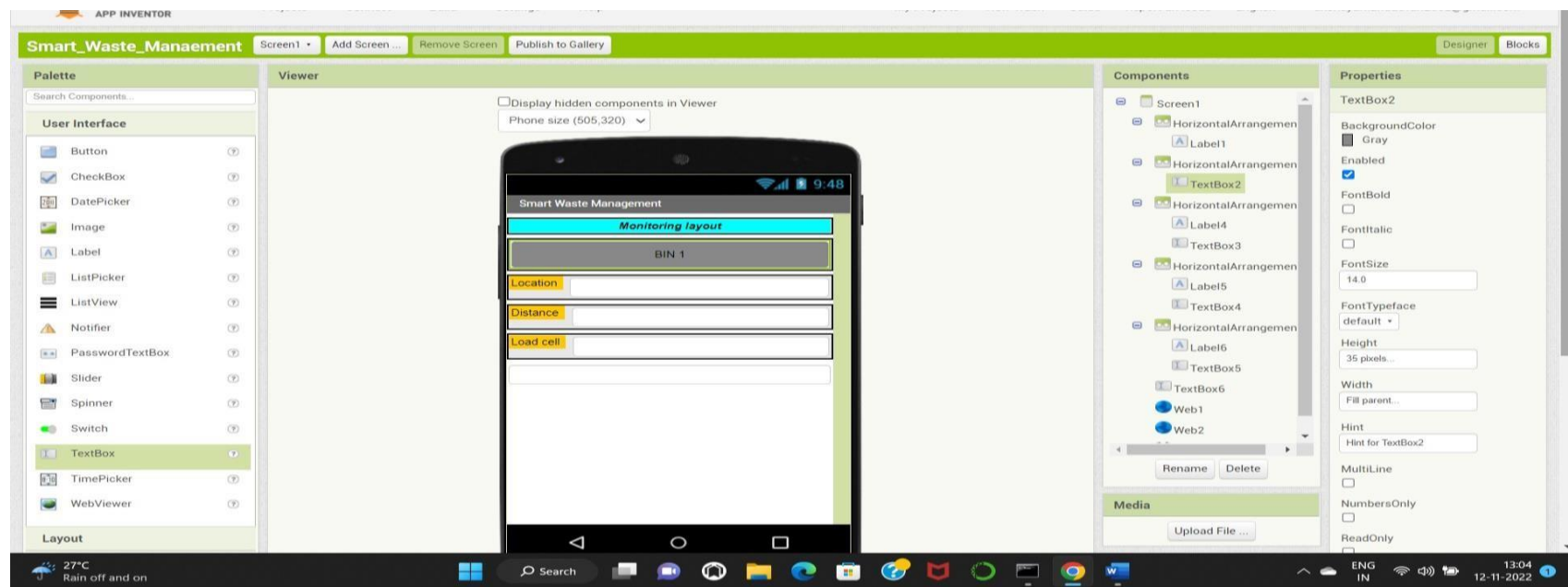
Step 4: Scan the QR code being displayed

Step 5: To scan the code install MIT AI2 COMPANION app in your mobile

Step 6: Now scan the code and the output is shown in mobile.

SCREENSHOTS OF MIT INVENTOR:

MIT WEBPAGE DESIGN WINDOW



BACKEND NODE RED FLOW CONNECTION

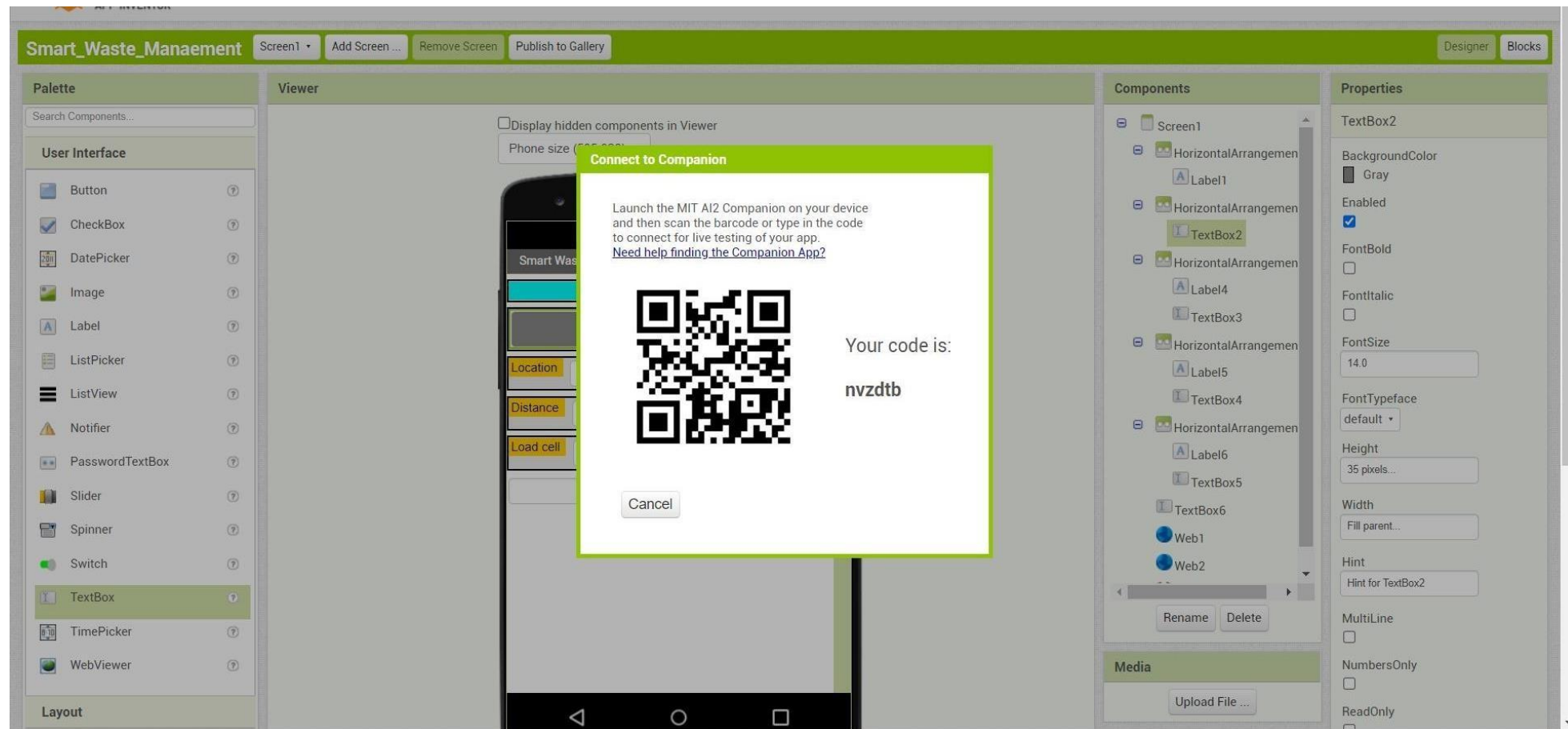
The screenshot displays the App Inventor IDE interface for a project titled "Smart_Waste_Management". The interface is divided into several sections:

- Top Bar:** Includes the project name "Smart_Waste_Management", a "Screen1" dropdown, and buttons for "Add Screen...", "Remove Screen", and "Publish to Gallery". On the right, there are "Designer" and "Blocks" tabs.
- Left Panel (Blocks):** Contains a "Built-in" category with sub-categories like Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, and Procedures. Below this is a "Screen1" category with components like HorizontalArrangemen, Label1, TextBox2, and Label4. At the bottom is a "Media" section with an "Upload File ..." button.
- Center Panel (Viewer):** Shows a Node-RED flow diagram. The flow starts with a "when Clock1.Timer" block, followed by a "do" block containing "set Web1.Url to" and "call Web1.Get". Below this is another "when Web1.GotText" block. This block has a "do" section with three parallel processing steps, each involving a "look up in pairs" block, a "call Web1.JsonTextDecodeWithDictionaries" block, and a "get responseType" block. The "look up in pairs" blocks are configured with keys "dist", "load", and "alert". The "notFound" outputs for these steps are set to "not found".
- Right Panel:** Features a trash can icon and a vertical toolbar with icons for zooming in (+), zooming out (-), and a target icon.

The flow diagram is as follows:

```
graph TD
    Clock1[when Clock1.Timer] --> Do1[do]
    subgraph Do1 [do]
        SetUrl[set Web1.Url to "http://127.0.0.1:1880/sensor"]
        CallGet[call Web1.Get]
    end
    Do1 --> Web1[Web1]
    Web1 -- GotText --> WhenGotText[when Web1.GotText]
    subgraph Do2 [do]
        direction TB
        Step1[look up in pairs key "dist"] --> Call1[call Web1.JsonTextDecodeWithDictionaries] --> Get1[get responseType]
        Step2[look up in pairs key "load"] --> Call2[call Web1.JsonTextDecodeWithDictionaries] --> Get2[get responseType]
        Step3[look up in pairs key "alert"] --> Call3[call Web1.JsonTextDecodeWithDictionaries] --> Get3[get responseType]
    end
    WhenGotText --> Do2
```

OR CODE



OUTPUTS IN MOBILE

2:26 PM | 2.0KB/s | 📶 📶 📶 📶 📶 84

Smart Waste Management

Monitoring layout

BIN 1

Location	Kangeyam
Distance	40
Load cell	10

No need to collect right now

2:25 PM | 2.1KB/s | 📶 📶 📶 📶 📶 84

Smart Waste Management

Monitoring layout

BIN 1

Location	chennai
Distance	12
Load cell	15

NEED BIN CHANGE!!!!

