

## PROJECT DEVELOPMENT PHASEPROJECT

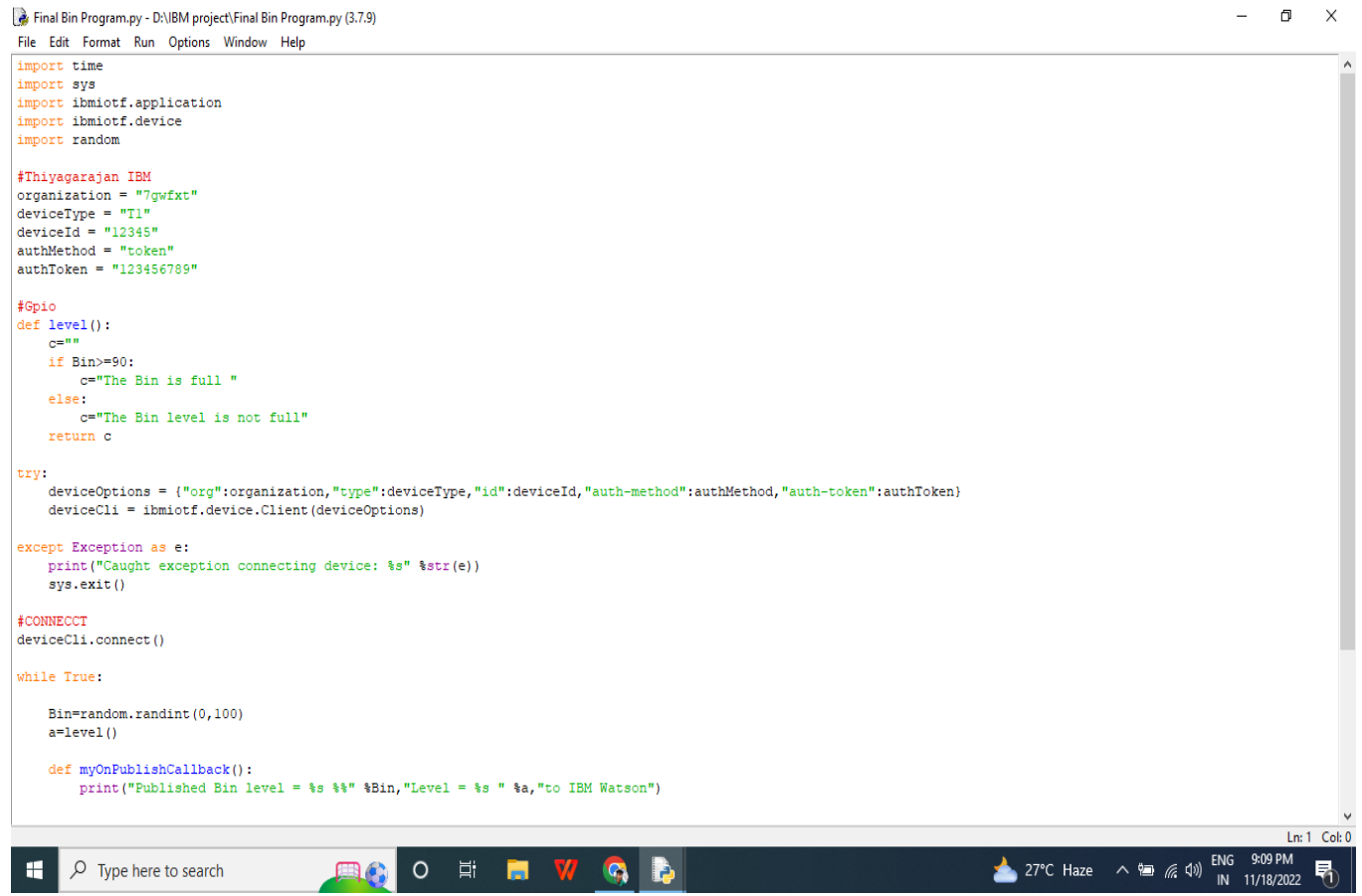
### DEVELOPMENT DELIVERY OF SPRINT 4

Date	19 Nov 2022
Team ID	PNT2022TMID38568
Project Name	Smart Waste Management System For Metropolitan Cities
Marks	

## SPRINT DESCRIPTION

In this Sprint we are about to describe about the Application we have developed and the Final Testing of the Python Code.

## PYTHON CODE



```
Final Bin Program.py - D:\IBM project\Final Bin Program.py (3.7.9)
File Edit Format Run Options Window Help

import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Thiyagarajan IBM
organization = "7gwfxr"
deviceType = "T1"
deviceId = "12345"
authMethod = "token"
authToken = "123456789"

#Gpio
def level():
    c=""
    if Bin>=90:
        c="The Bin is full "
    else:
        c="The Bin level is not full"
    return c

try:
    deviceOptions = {"org":organization,"type":deviceType,"id":deviceId,"auth-method":authMethod,"auth-token":authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:
    print("Caught exception connecting device: %s" %str(e))
    sys.exit()

#CONNECT
deviceCli.connect()

while True:

    Bin=random.randint(0,100)
    a=level()

    def myOnPublishCallback():
        print("Published Bin level = %s %%" %Bin,"Level = %s " %a,"to IBM Watson")
```

```
Final Bin Program.py - D:\IBM project\Final Bin Program.py (3.7.9)
File Edit Format Run Options Window Help

#Gpio
def level():
    c=""
    if Bin>=90:
        c="The Bin is full "
    else:
        c="The Bin level is not full"
    return c

try:
    deviceOptions = {"org":organization,"type":deviceType,"id":deviceId,"auth-method":authMethod,"auth-token":authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:
    print("Caught exception connecting device: %s" %str(e))
    sys.exit()

#CONNECT
deviceCli.connect()

while True:

    Bin=random.randint(0,100)
    a=level()

    def myOnPublishCallback():
        print("Published Bin level = %s" %Bin,"Level = %s" %a,"to IBM Watson")

    data={'bin':Bin,'level':level()}

    success = deviceCli.publishEvent("IoTSensor","json",data,qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
        time.sleep(10)

#Disconnect

deviceCli.disconnect()
```

Ln: 1 Col: 0

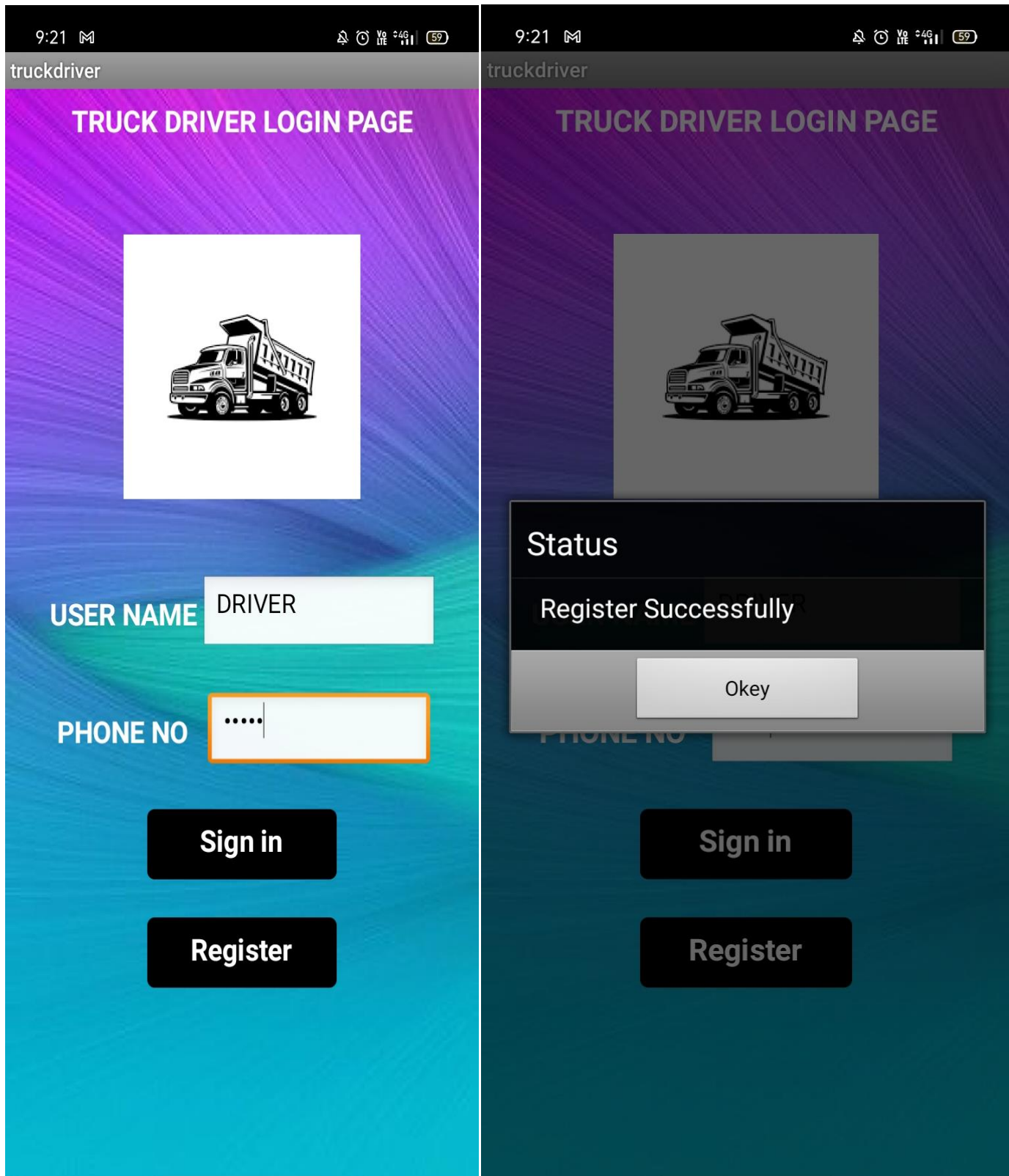
Type here to search 27°C Haze 9:09 PM 11/18/2022

# PYTHON EXECUTED OUTPUT

```
Python 3.7.9 Shell
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\IBM project\Final Bin Program.py =====
2022-11-18 21:06:19,479 ibmiotf.device.Client INFO Connected successfully: d:7gwfxTl:12345
Published Bin level = 40 % Level = The Bin level is not full to IBM Watson
Published Bin level = 63 % Level = The Bin level is not full to IBM Watson
Published Bin level = 37 % Level = The Bin level is not full to IBM Watson
Published Bin level = 6 % Level = The Bin level is not full to IBM Watson
Published Bin level = 70 % Level = The Bin level is not full to IBM Watson
Published Bin level = 23 % Level = The Bin level is not full to IBM Watson
Published Bin level = 0 % Level = The Bin level is not full to IBM Watson
Published Bin level = 61 % Level = The Bin level is not full to IBM Watson
Published Bin level = 85 % Level = The Bin level is not full to IBM Watson
Published Bin level = 64 % Level = The Bin level is not full to IBM Watson
Published Bin level = 73 % Level = The Bin level is not full to IBM Watson
Published Bin level = 90 % Level = The Bin is full to IBM Watson
Published Bin level = 63 % Level = The Bin level is not full to IBM Watson
Ln: 14 Col: 0
```

# APPLICATION SCREENS

HERE WE DISPLAYED SCREEN 1 OF OUR CREATED APPLICATION



HERE WE DISPLAYED SCREEN 2 OF OUR CREATED APPLICATION

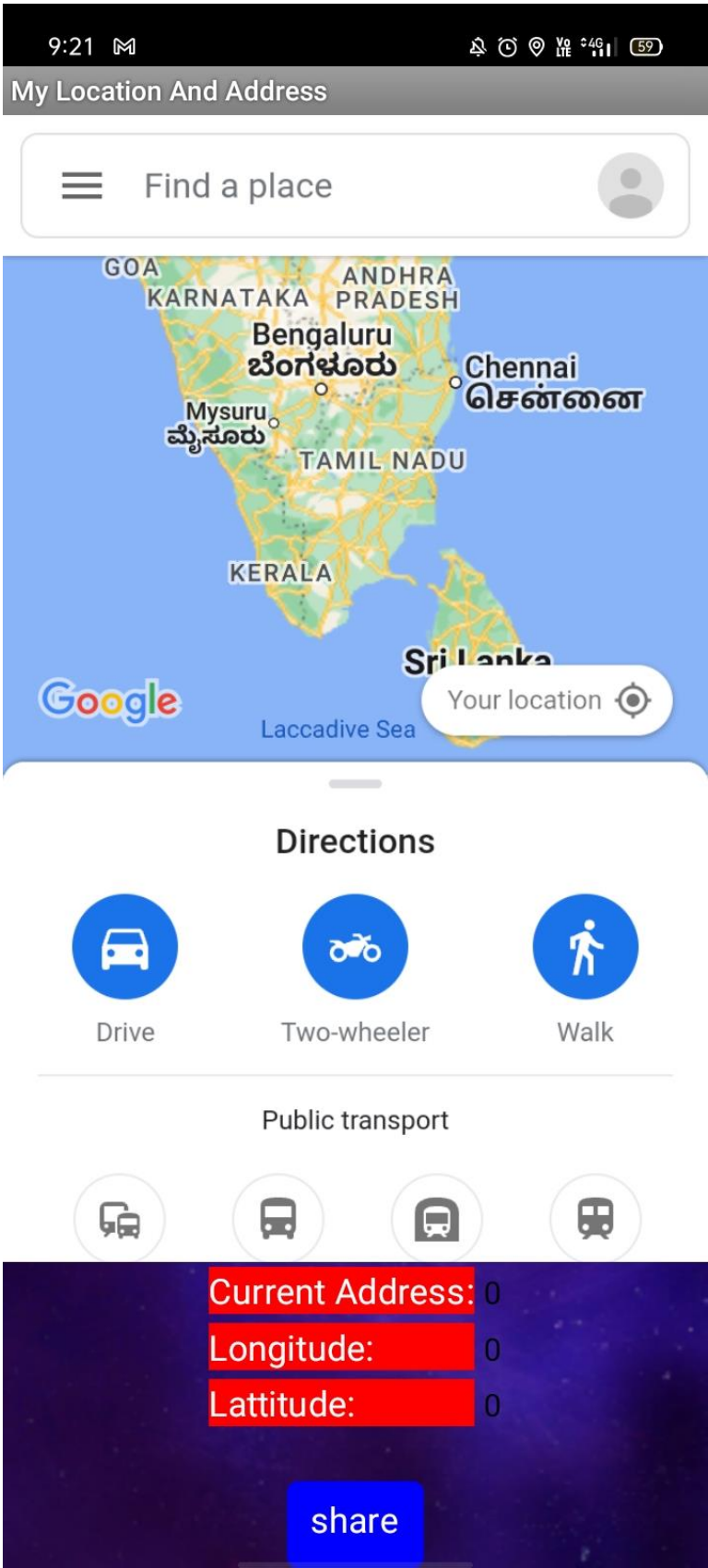


HERE WE DISPLAYED SCREEN 3 OF OUR CREATED APPLICATION



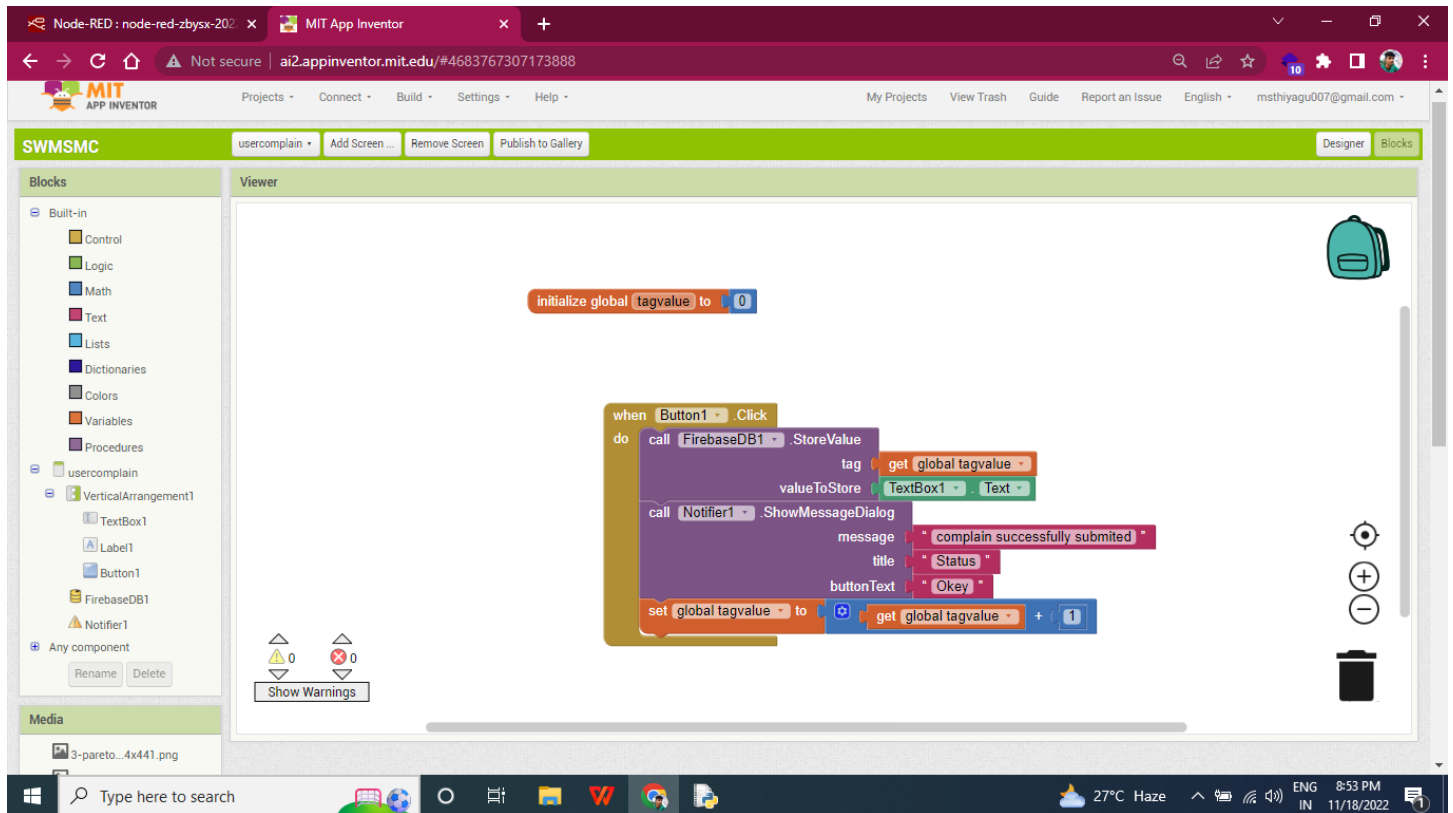


HERE WE DISPLAYED SCREEN 4 OF OUR CREATED APPLICATION



# BLOCKS FUNCTIONS USED IN OUR APPLICATION

## SCREEN 1





## SCREEN 2

The screenshot displays the MIT App Inventor web interface for a project named "SWMSMC". The interface is divided into several sections:

- Top Navigation Bar:** Includes "Projects", "Connect", "Build", "Settings", and "Help" menus. The user's email "msthiyagu007@gmail.com" is visible.
- Project Bar:** Shows the project name "SWMSMC" and buttons for "Add Screen...", "Remove Screen", and "Publish to Gallery".
- Blocks Panel (Left):** Lists built-in blocks categorized by type: Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, and Procedures. Specific components like "WebView1", "Label3", "Address", "Longitude", "Latitude", and "Label1" are listed under "truckdrivers".
- Viewer Area (Center):** Displays the block diagram for the "truckdrivers" component. The logic includes:
  - Initialization:** "when truckdrivers.Initialize" triggers a "do" block containing "call WebView1.GoToUrl" with a URL constructed from "https://www.google.com/maps/@", "get global LAT", and "19z".
  - Location Handling:** "when LocationSensor1.LocationChanged" triggers a "do" block that sets "Address.Text" to "LocationSensor1.CurrentAddress", "Longitude.Text" to "get longitude", "Latitude.Text" to "get latitude", and updates "global LONG" and "global LAT".
  - Message Sharing:** "when Button1.Click" triggers a "do" block that calls "Sharing1.ShareMessage" with a message constructed from "https://www.google.com/maps/@", "get global LAT", and "19z".
- Media Panel (Bottom Left):** Shows a file named "3-pareto...4x441.png".
- Bottom Status Bar:** Displays the system clock as 8:56 PM on 11/18/2022, along with weather information (27°C Haze) and network status.

## SCREEN 3

The screenshot displays the MIT App Inventor web interface in a browser window. The address bar shows the URL `ai2.appinventor.mit.edu/#4683767307173888`. The interface is divided into several sections:

- Top Bar:** Includes the MIT App Inventor logo, navigation links (Projects, Connect, Build, Settings, Help), and user information (My Projects, View Trash, Guide, Report an Issue, English, msthiyagu007@gmail.com).
- Project Bar:** Shows the project name "SWMSMC" and buttons for "usercomplain", "Add Screen...", "Remove Screen", and "Publish to Gallery".
- Left Panel (Blocks):** Contains a "Built-in" category with various block types (Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, Procedures) and a "usercomplain" category with components like "VerticalArrangement1", "TextBox1", "Label1", "Button1", "FirebaseDB1", and "Notifier1".
- Center Canvas (Viewer):** Displays a Scratch-like block editor with the following code:
  - `initialize global tagvalue to 0`
  - `when Button1.Click` event trigger:
    - `do` loop containing:
      - `call FirebaseDB1.StoreValue` with `tag` set to `get global tagvalue` and `valueToStore` set to `TextBox1.Text`.
      - `call Notifier1.ShowDialog` with `message` set to `"complain successfully submitted"`, `title` set to `"Status"`, and `buttonText` set to `"Okay"`.
      - `set global tagvalue to` `get global tagvalue` + `1`.

The bottom of the screen shows a Windows taskbar with the search bar, task icons, and system tray information (27°C Haze, 8:53 PM, 11/18/2022).

## SCREEN 4

The screenshot displays the MIT App Inventor web interface. The browser address bar shows the URL `ai2.appinventor.mit.edu/#4683767307173888`. The project name is **SWMSMC**. The interface is divided into three main sections: **Blocks**, **Viewer**, and **Media**.

**Blocks Section:**

- Built-in:** Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, Procedures.
- truckdrivers:** WebView1, TableArrangement1, Label3, Address, Longitude, Latitude, Label1.

**Viewer Section:**

The code blocks are as follows:

- when truckdrivers Initialize**
  - do call **WebView1 GoToUri** url join `https://www.google.com/maps/@`, `get global LAT`, `get global LONG`, `19z`
- initialize global LAT to**
- initialize global LONG to**
- when LocationSensor1 LocationChanged**
  - do **set Address Text to LocationSensor1 CurrentAddress**
  - set Longitude Text to get longitude**
  - set Latitude Text to get latitude**
  - set global LONG to get longitude**
  - set global LAT to get latitude**
- when Button1 Click**
  - do call **Sharing1 ShareMessage** message join `https://www.google.com/maps/@`, `get global LAT`, `get global LONG`, `19z`

**Media Section:**

- 3-pareto...4x441.png

The bottom status bar shows the system clock as 8:56 PM on 11/18/2022, with weather information (27°C Haze) and language settings (ENG IN).