

**SIGNS WITH**  
**SMART**  
**CONNECTIVIT**  
**Y FOR BETTER**  
**ROAD SAFETY**  
**SOURCE CODE:**

**TEAM ID:**  
**PNT2022TMID12702**

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

#Provide your IBM Watson Device Credentials
organization = "8dxkha"
deviceType = "madhu"
deviceId = "madhu"
authMethod = "token"
authToken = "yah&46&uqf!k4Rq!n+" #

Initialize GPIO
```

```

temp=random.randint(20,50)
humid=random.randint(20,50)
lat =random.uniform(10.781377,10.78643) lon =
random.uniform(79.781377,79.78643)

```

```

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command']) print(cmd)

```

```

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 and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting"
10 times
deviceCli.connect()

```

```

while True:
    #Get Sensor Data from DHT11

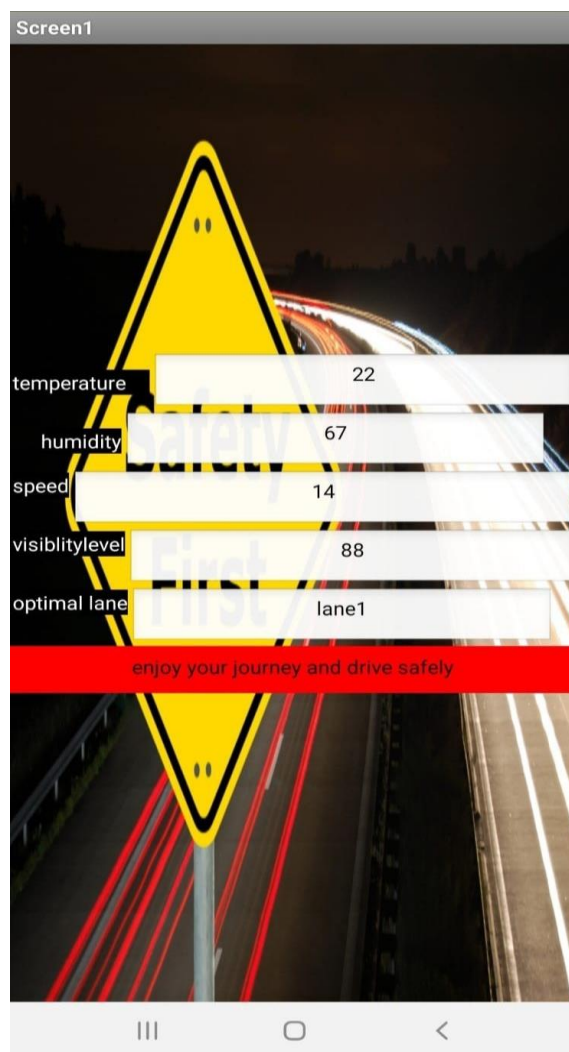
    data = {"d":{ 'temp' : temp,"lat":lat,"lon":lon}}
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %" % humid, "to IBM
Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)

```

```
if not success:  
    print("Not connected to IoTF") time.sleep(1)  
  
deviceCli.commandCallback = myCommandCallback  
  
# Disconnect the device and application from the cloud  
deviceCli.disconnect()
```

## MIT APP INVENTOR SNAPSHOTS: (FINAL OUTPUT)



**TO SHOW DIFFERENCE IN SPEED AND OTHER PARAMETERS**

