**IOT(internet of things)**

**project on smart parking system for smart cities using iot**

**by-- team 16**

**1.T. Rajasree** **(17481A04I7)**

**2.T. Gagana Sai** **(17481A04I8)**

**3.T. Kavya** **(17481A04M2)**

**TABLE OF CONTENTS:**

**1.INRODUCTION**

1.1 Overview

1.2 Purpose

**2.LITERATURE SURVEY**

2.1 Existing problem

2.2 Proposed solution

**3.THEORETICAL ANALYSIS**

3.1 Block diagram

3.2 Hardware/Software designing

**4.EXPERIMENTAL INVESTIGATIONS**

**5.FLOW CHART**

**6.RESULT**

**7.ADVANTAGES & DISADVANTAGES**

**8.APPLICATIONS**

**9.CONCLUSION**

**10.FUTURE SCOPE**

**11.BIBLOGRPHY**

**APPENDIX**

A. Source code4R

**1.INRODUCTION:**

**1.1 Overview**

Nowadays the congestion of traffic levels increases with the increasing development of the population rapidly. Concerning the amount of population, the utilization of personal vehicles also increased. Due to more use of cars, traffic congestion occurred on the road. Most of the people choose personal vehicles than public transportation. It is very difficult and time-consuming to find parking space in Metropolitan areas, commercial areas, especially during the rush hours.

It is often costly almost all-over big cities around the world to find proper and secure parking space. The project we are proposed is a smart parking system that delivers information to people finding a parking space online.

**1.2 Purpose**

This is used to reduce the traffic jam that occurs in urban areas which are caused by vehicles searching for parking.

**2.LITERATURE SURVEY:**

**2.1 Existing Problem**

**1.shortage of parking space**

**2.high parking traffic**

**2.2 Proposed Solution**

The proposed system consists of some phases:

1.Development of Android app

2.Free space identification

3.Authenticating user vehicle

4.Classify parking slot

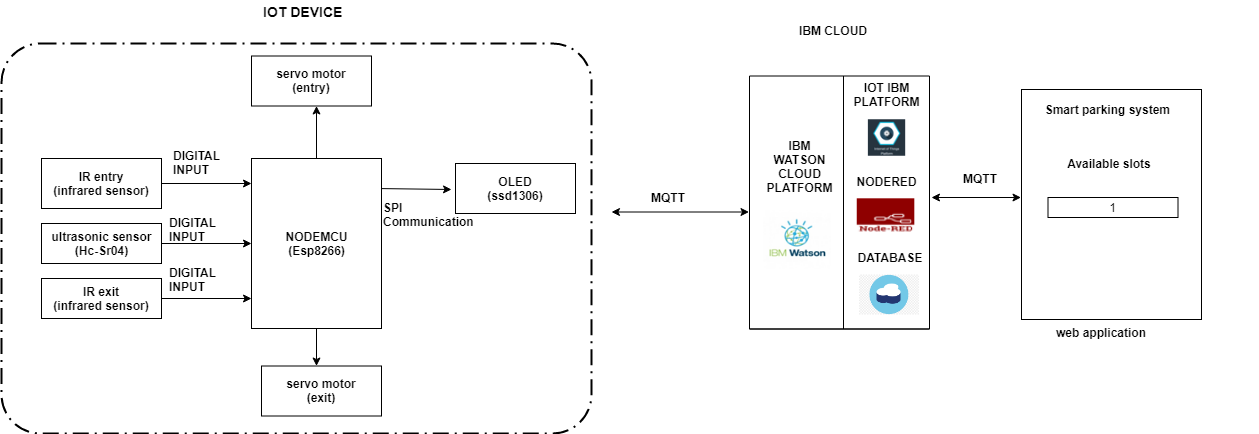
5.Navigating to parking slot

6.Visualization in server for owner to analyze

* Provides optimum use of all available spaces
* Congestion can be decreased by diverting traffic to available spaces
* Decrease fuel consumption to search space
* Decreases time to search space
* Provides real time parking space tracking
* Provides efficient

**3.THEORITICAL ANALYSIS :**

**3.1 Block Diagram**



**3.2 Hardware/Software designing :**

**Software Designing:**

The software used for smart parking system is:

a. Python

b. IOT Open Hardware platforms

c. IOT Application Development

d. IOT Cloud Platform

e. IOT Communication Technologies

f. IOT Communication Protocols

**4.EXPERIMENTAL INVESTIGATION :**

*The Internet of things (IoT) is a system of interrelated computing devices, mechanical and digital machines provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. The definition of the Internet of things has evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems.*

*The IoT can assist in the integration of communications, control, and information processing across various transportation systems. Application of the IoT extends to all aspects of transportation systems (i.e. the vehicle, the infrastructure, and the driver or user). Dynamic interaction between these components of a transport system enables inter- and intra-vehicular communication, smart traffic control, smart parking, electronic toll collection systems, vehicle control, safety, and road assistance. For example, an IoT platform can continuously monitor the location and vacancies of spaces in parking.*

*This can only be possible with the IoT and its seamless connectivity among devices. Sensors such as GPS, Humidity, and Temperature send data to the IoT platform and then the data is ANALYSED and then sent to the users. This way, users can track the real-time status of vehicles and can make appropriate decisions. The smart parking system is an IOT based device which is capable of automatic sensing of vehicles. ALSO, the data of sensors will be displayed in graphical form on the IBM cloud page.*

**5.FLOWCHART:**

DRIVER KEEPS LOOKING FOR THE AVAILABLE SLOT

SLOT AVAILABLE

DRIVER OPEN THE APP

APP

APP

DRIVER PARKS IN THE AVAILABLE SLOT

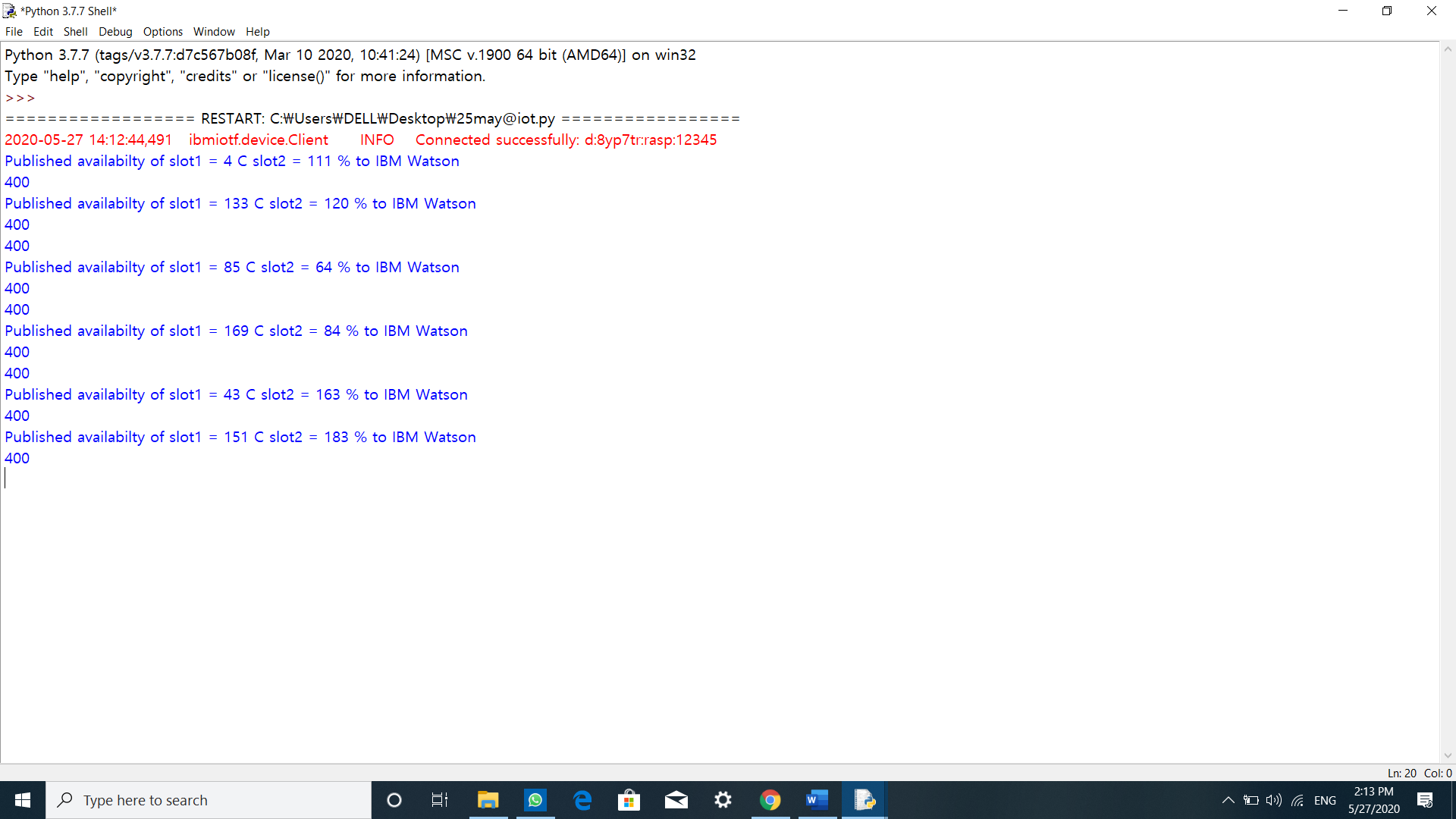
CAR ALREADY PARKED?

SEND MESSAGE TO DRIVER THAT SLOT IS NOT AVAILABLE

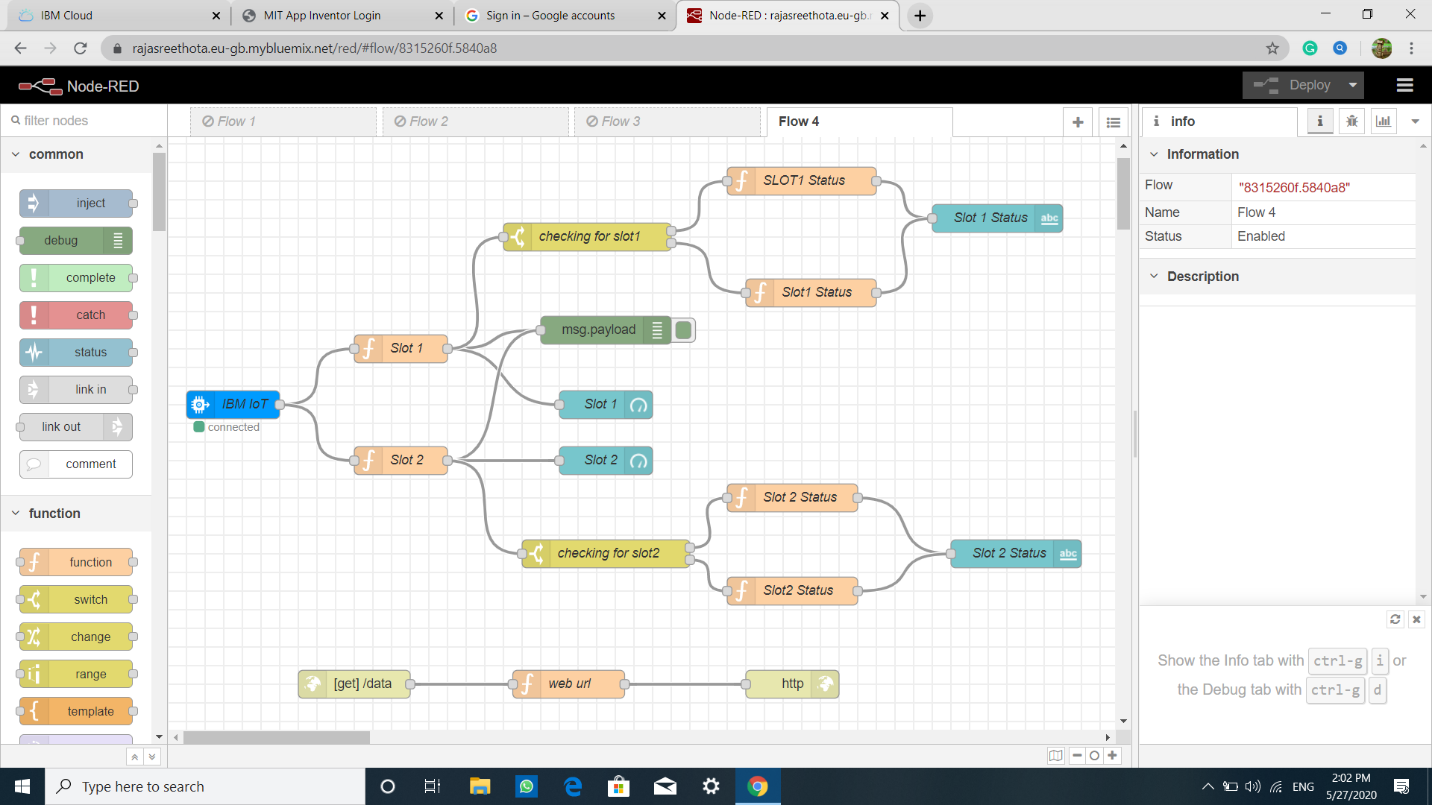
SEND MESSAGE TO DRIVER THAT SLOT IS AVAILABLE

**6.RESULT :**

**Python code:**

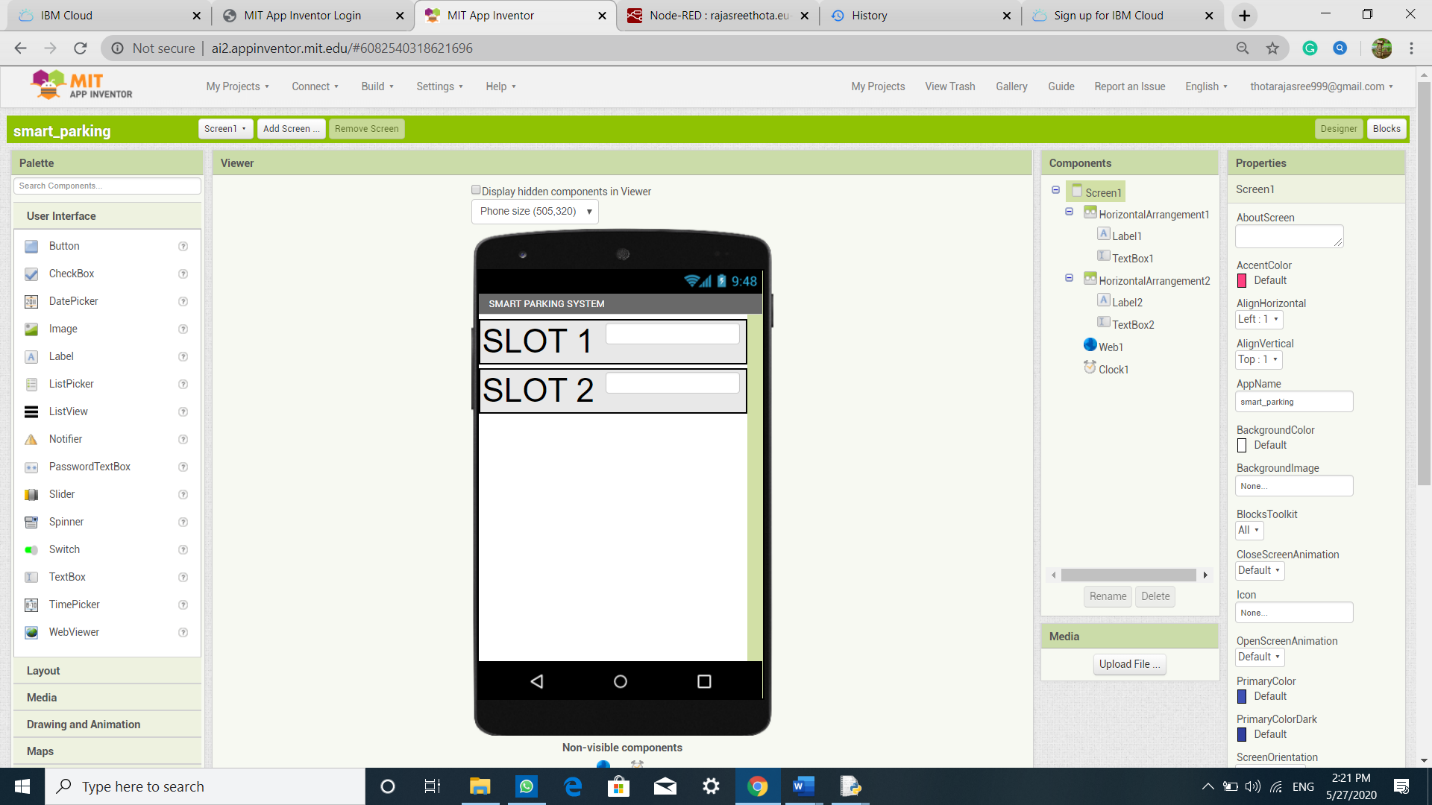
****

**Node-red:**

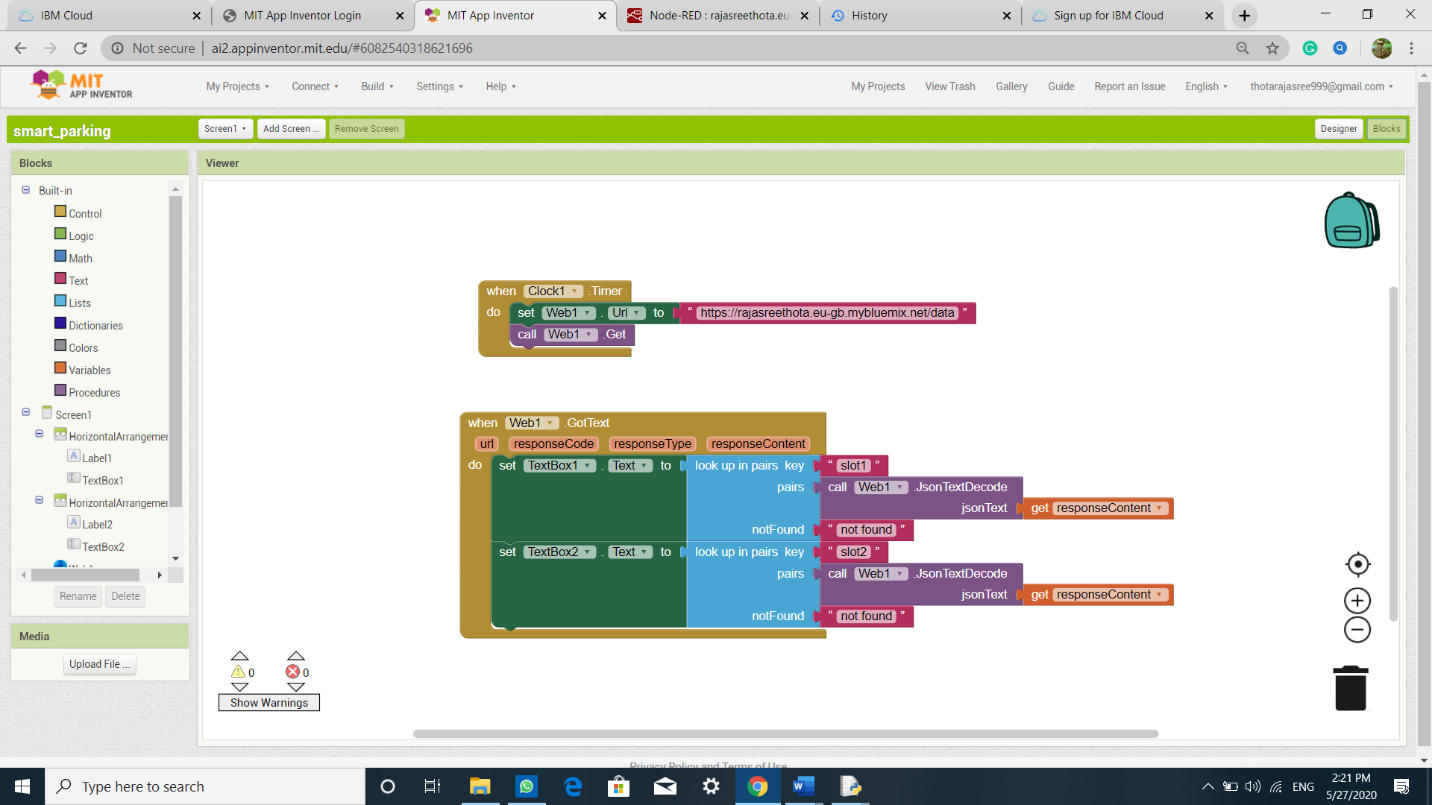
****

[https://rajasreethota.eu-gb.mybluemix.net/ui/#](https://rajasreethota.eu-gb.mybluemix.net/ui/)

**Mit app:**

****

**Design:**

****

**7.ADVANTAGES AND DISADVANTGES :**

**Advantages :**

* Ease of maintenance
* Shorter waiting time at parking place
* Reduced pollution
* Reduced traffic
* Carbon emission is reduced
* Very faster communication
* No wastage of time
* Required less manpower
* Efficiency

**Disadvantages:**

* In places where there is no provision of GSM networks, it is difficult for communication.

**8. APPLICATIONS:**

The smart car parking systems can be implemented in

* Shopping malls
* Restaurants
* Theatres

**9.CONCLUSION:**

* The systems benefit of smart parking go well beyond avoiding time wasting.
* Developing a smart parking solution with in a city solves the pollution problems.
* The growth of Internet of Things and Cloud technologies have give rise to new possibilities in terms of smart cities. Smart parking facilities and traffic management systems have always been at the core of constructing smart cities.
* In this, we address the issue of parking and present an IOT based Cloud integrated smart parking system. The system that we propose provides real time information regarding availability of parking slots in a parking area.
* Users from remote locations could book a parking slot for them by the use of our mobile application. The efforts made are indented to improve the parking facilities of a city and thereby aiming to enhance the quality of life of its people.

**10.FUTURE SCOPE:**

The future of the smart parking system is expected to be significantly by the arrival of automated vehicles (AVs). Several cities around the world are already beginning to trial self-parking vehicles, specialized AV parking lots, and robotic parking vehicles.

**11.BIBILOGRAPHY :**

<https://cloud.ibm.com/registration>

<https://cloud.ibm.com/catalog/services/iot>

<http://Ai2.appinventor.mit.edu>

<https://flows.nodered.org/node/node-red-dashboard>

<https://developer.ibm.com/recipes/tutorials/ui-dashboard-for-iot-device-data-using-node-red/>

<https://appinventor.mit.edu/>

**APPENDIX:**

A.Source Code

**import time**

**import sys**

**import ibmiotf.application**

**import ibmiotf.device**

**import random**

**import requests**

**#Provide your IBM Watson Device Credentials**

**organization = "8yp7tr"**

**deviceType ="rasp"**

**deviceId = "12345"**

**authMethod = "token"**

**authToken = "123456789"**

**def myCommandCallback(cmd):**

**print("Command received: %s" % cmd.data)#Commands**

**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:**

**slot1=random.randint(0, 200)**

**#print(slot1)**

**slot2 =random.randint(0, 200)**

**#Send availabilty of slot1 & slot2 area of parking to IBM Watson**

**data = { 'slot1' : slot1, 'slot2': slot2}**

**#print (data)**

**def myOnPublishCallback():**

**print ("Published availabilty of slot1 = %s C" % slot1, "slot2 = %s %%" % slot2, "to IBM Watson")**

**success = deviceCli.publishEvent("Parking", "json", data, qos=0, on\_publish=myOnPublishCallback)**

**if(slot1>=51):**

**r = requests.get('https://www.fast2sms.com/dev/bulk?authorization=WDxqL3Yf3gWocizB51pWW8kTMmsL37oekGBRPqIoiZr6WJiLcHkao6BO4jM8&sender\_id=FSTSMS&message=SLOT1 IS FULL.....PLEASE WAIT FOR SOMETIME UNTIL THE SLOT GETS EMPTY...&language=english&route=p&numbers=7995317541')**

**print(r.status\_code)**

**if(slot2>=51):**

**r = requests.get('https://www.fast2sms.com/dev/bulk?authorization=WDxqL3Yf3gWocizB51pWW8kTMmsL37oekGBRPqIoiZr6WJiLcHkao6BO4jM8&sender\_id=FSTSMS&message=SLOT2 IS FULL.....PLEASE WAIT FOR SOMETIME UNTIL THE SLOT GETS EMPTY...&language=english&route=p&numbers=799531754')**

**print(r.status\_code)**

**if not success:**

**print("Not connected to IoTF")**

**time.sleep(2)**

**deviceCli.commandCallback = myCommandCallback**

**# Disconnect the device and application from the cloud**

**deviceCli.disconnect()**