

1.INTRODUCTION :-

a. Overview:

Heavy traffic caused by vehicles is becoming a major problem and it is growing exponentially. Around 1400 new cars enter the roads of Delhi in India every day. Looking out for a vacant parking space in peak hours is often frustrating for people.

This in turn has made it more difficult for people to park their vehicles in crowded public places like restaurants and malls during peak hours. We burn around one million barrels of oil every day. In a recent survey, it was found that a driver takes nearly 8 minutes to park his vehicle, because he spends more time in searching for a parking lot. This searching leads to 30 to 40% of traffic congestion.

With the implementation of the smart parking system, patrons can easily locate and secure a vacant parking space at any car park deemed convenient to them. Vehicle ingress and egress are also made more convenient with the implementation of hassle free payment mechanism. With vehicle detection sensors aplenty on the market, the choices made may defer due to the different requirements in addition to the its pros and cons.

b.Purpose:

Smart parking is a strategy that smart cities can use to increase parking efficiency and combat numerous issues. Smart parking will reduce fuel use because the driver won't be circling streets looking for a place to park because they'll have an identified space.

Through the wealth of apps available to us on smartphones and other devices, AI shows us the fastest way of parking. This apps can avoid major problems in parking areas like Unclear parking policies, searching of parking area in crowded areas,time consumption..etc,.

2.LITERATURE SURVEY:-

a. Existing problem:

Car parking is a major problem in urban areas in both developed and developing countries. Shortage of parking space, high parking tariffs, and traffic

congestion due to visitors in search for a parking place are only a few examples of everyday parking problems.

Finding a vacant parking space especially in crowded public places. Then it can kill the time which means the user can get effect by time waste.

Poor use of available parking spaces. Because it is difficult to finding empty parking area in crowded public places.then it made the problem of poor use of available parking spaces.

Difficult to entering and leaving, it is cause because of more number of vehicles in parking area.

b.Proposed solution:

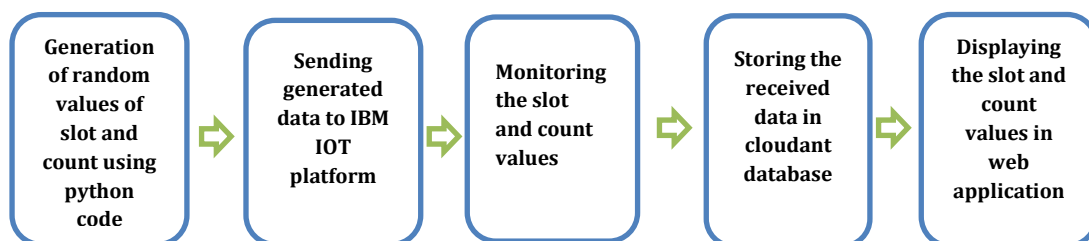
To avoid the above problem, the ***Smart Parking System in cities*** project is implemented. Some wireless sensors are placed in the gateway which can send the data regarding the slot and count values and give a clarity to driver where he should park his vehicle. Through this, the person can easily park the vehicle without searching for the slot to park the vehicle.

By our app we can have solutions for all the problems existing in parking areas. Because of this app we can provide empty parking slod address in advance to the user at the entrance of the parking area. Then it makes users happy by time saving, avoiding of confusions of parking areas, avoid all the problems of user about parking

.

3.THEORETICAL ANALYSIS:-

a. Block Diagram:



b.Hardware/Software designing:

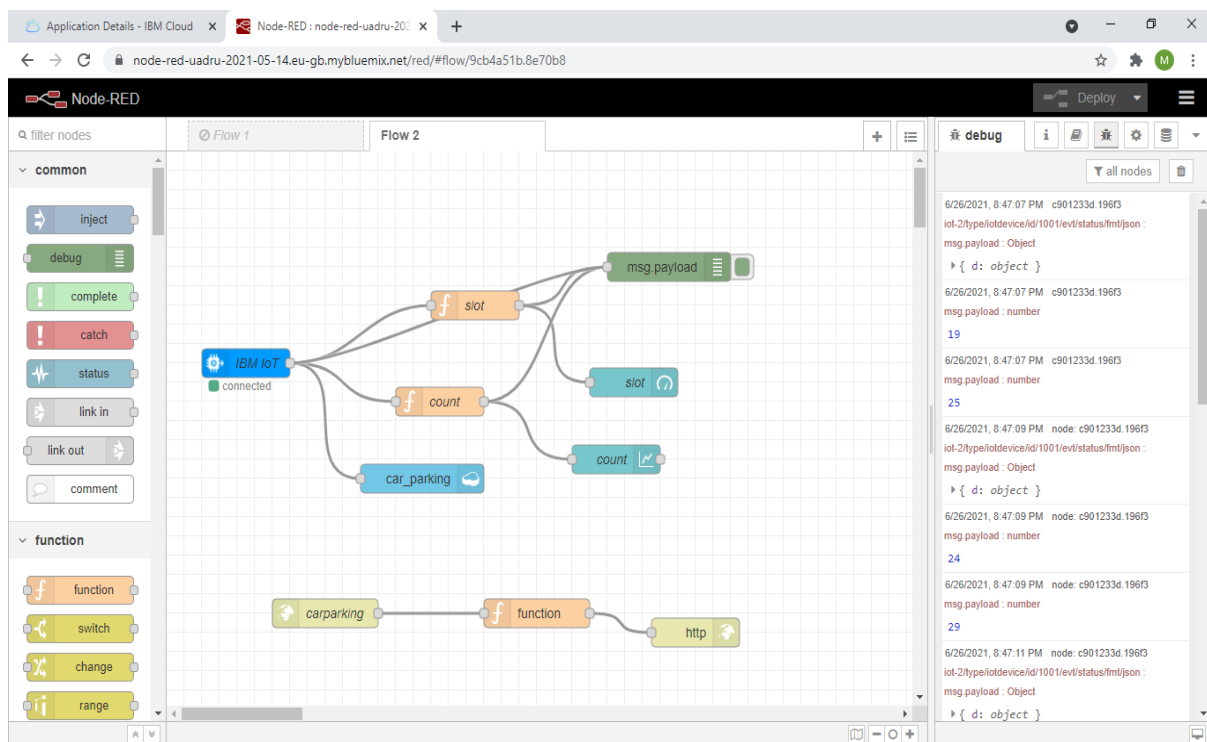
We follow a step-by-step procedure to set up all the interfaces required for our project and develop the code in python to send random sensor data to the cloud. The following software is required:

- Python Idle (with specified packages installed)
- IBM cloud
- Node Red service

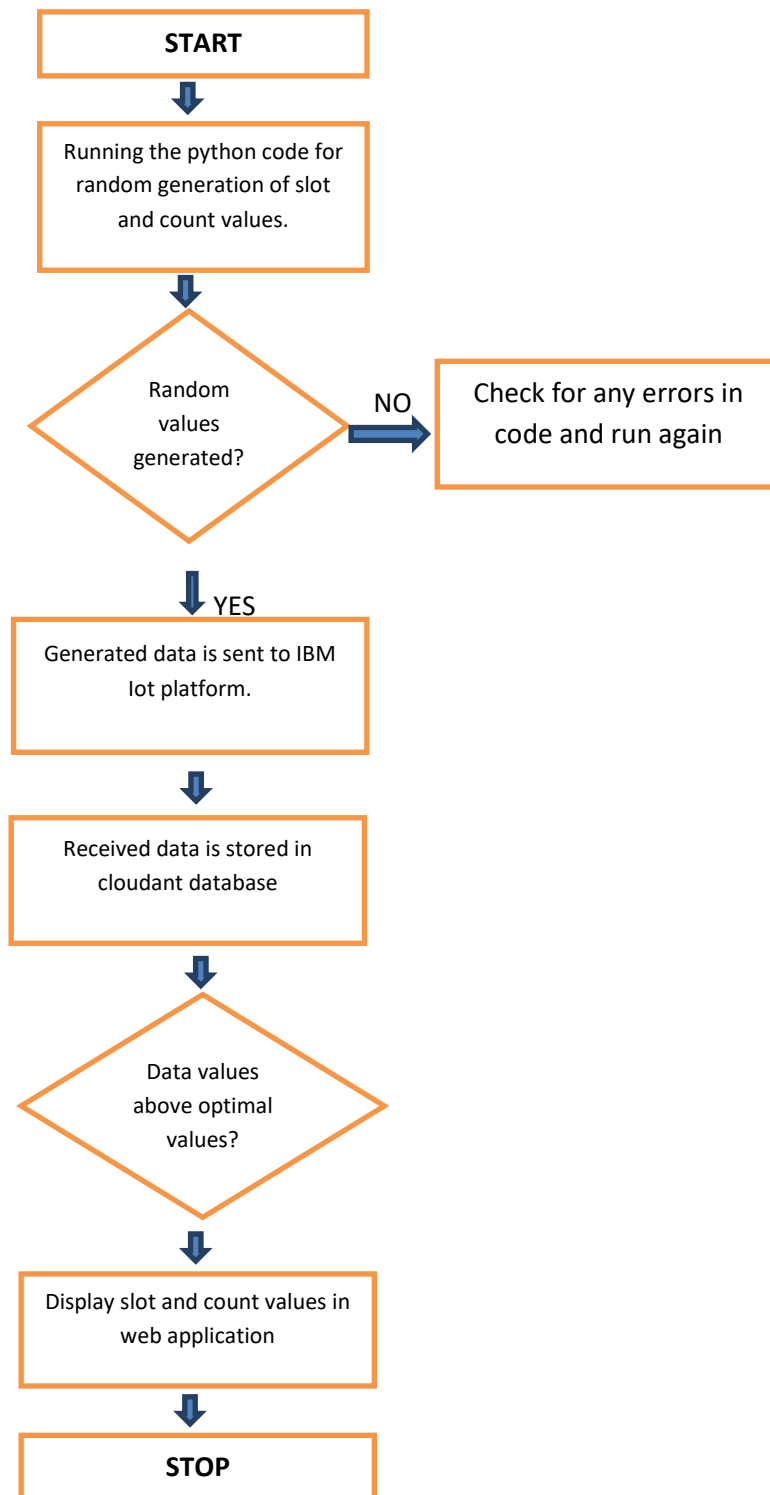
4.Experimental Investigations:-

To successfully allot the slot to vehicles, some sensors are placed at the gateway which automatically allot the slot number and display the count of vehicle. Since hardware is not available, a python code snippet is used to generate random values such as slot and count. This randomly generated data is sent to IBM IoT platform.

Using node red service, a node red flow is created which retrieves the data from IBM IoT platform. This received data is stored in Cloudant database. Since there will be limited slots we have confined ourselves to up to a limit of 30 slots.



5.FLOWCHART:-



6.RESULT:-

The result of this project fulfils our purpose of continuous allocation of slots to vehicles to be parked .This reduces the time, confusion that occur near the place of parking , where it shows the clear cut place to park a vehicle. The following figure is the output of python code, where it randomly allocating the slot and count values.



```
File Edit Shell Debug Options Window Help
Published data Successfully: ('d': {'slot': 30, 'count': 26})
Published data Successfully: ('d': {'slot': 4, 'count': 19})
Published data Successfully: ('d': {'slot': 24, 'count': 4})
Published data Successfully: ('d': {'slot': 20, 'count': 27})
Published data Successfully: ('d': {'slot': 26, 'count': 5})
Published data Successfully: ('d': {'slot': 15, 'count': 30})
Published data Successfully: ('d': {'slot': 19, 'count': 25})
Published data Successfully: ('d': {'slot': 24, 'count': 29})
Published data Successfully: ('d': {'slot': 29, 'count': 0})
Published data Successfully: ('d': {'slot': 29, 'count': 1})
Published data Successfully: ('d': {'slot': 11, 'count': 7})
Published data Successfully: ('d': {'slot': 19, 'count': 29})
Published data Successfully: ('d': {'slot': 8, 'count': 17})
Published data Successfully: ('d': {'slot': 8, 'count': 19})
Published data Successfully: ('d': {'slot': 19, 'count': 25})
Published data Successfully: ('d': {'slot': 2, 'count': 17})
Published data Successfully: ('d': {'slot': 19, 'count': 7})
Published data Successfully: ('d': {'slot': 13, 'count': 1})
Published data Successfully: ('d': {'slot': 6, 'count': 6})
Published data Successfully: ('d': {'slot': 16, 'count': 5})
Published data Successfully: ('d': {'slot': 17, 'count': 19})
Published data Successfully: ('d': {'slot': 25, 'count': 26})
Published data Successfully: ('d': {'slot': 3, 'count': 0})
Published data Successfully: ('d': {'slot': 11, 'count': 24})
Published data Successfully: ('d': {'slot': 3, 'count': 28})
Published data Successfully: ('d': {'slot': 22, 'count': 1})
Published data Successfully: ('d': {'slot': 11, 'count': 21})
Published data Successfully: ('d': {'slot': 2, 'count': 15})
Published data Successfully: ('d': {'slot': 26, 'count': 1})
Published data Successfully: ('d': {'slot': 21, 'count': 27})
Published data Successfully: ('d': {'slot': 26, 'count': 8})
Published data Successfully: ('d': {'slot': 5, 'count': 12})
Published data Successfully: ('d': {'slot': 20, 'count': 27})
Published data Successfully: ('d': {'slot': 27, 'count': 0})
Published data Successfully: ('d': {'slot': 9, 'count': 14})
Published data Successfully: ('d': {'slot': 0, 'count': 10})
Published data Successfully: ('d': {'slot': 13, 'count': 26})
Published data Successfully: ('d': {'slot': 16, 'count': 4})
Published data Successfully: ('d': {'slot': 5, 'count': 13})
Published data Successfully: ('d': {'slot': 14, 'count': 17})
Published data Successfully: ('d': {'slot': 16, 'count': 2})
Published data Successfully: ('d': {'slot': 2, 'count': 12})
Published data Successfully: ('d': {'slot': 8, 'count': 13})
```

Ln: 220 Col: 0

7.ADVANTAGES & DISADVANTAGES:-

Advantages:

- ✓ Limited parking spaces can be utilized efficiently. Guides the drivers to the available vacant parking slots.
- ✓ Manages traffic well inside the malls without any havoc.
- ✓ It helps in managing parking space effectively which results in significant revenue generation.
- ✓ Increased safety.
- ✓ Saving of time and money.
- ✓ Convenient to driver.
- ✓ Less congestion.

Disadvantages:

- ✗ Every user has to be registered in order to use the system.
- ✗ The user needs to park his vehicle in the slot that is allocated to him.
- ✗ The user should be sensible enough to know that he should Vacate the parking space at his specified departure time.
- ✗ The maintenance of sensors is expensive.
- ✗ It is difficult is sensors get damaged.

8.APPLICATIONS:

Airports,
Shopping malls,
Cinema halls,
Super markets,
Hospitals,
Function areas,
Resorts,
Hotels,
Restaurants,
Railway stations,
Bus stands,
Large apartment areas,
Temples,

Banks,
Colleges,
Schools, etc.,

9.CONCLUSION:-

The expression “smart city” has been used for several years by a number of technology companies and serves as a description for the application of compound systems to integrate the operation of urban infrastructure and services such as buildings, transportation, electrical and water distribution, and public safety.

This project focuses on implementation of car parking space detection using of Internet of Things. The system benefits of smart parking go well beyond avoiding time wasting. Developing a smart parking solution within a city reduces the pollution problems.

10.FUTURE SCOPE:-

This project can be extended in future since there is increase in population, vehicles and it also reduces the search of human for parking.

11.BIBILOGRAPHY:-

1.Renuka, R., and S. Dhanalakshmi. "ANDROID BASED SMART PARKING SYSTEM USING SLOT ALLOCATION & RESERVATIONS." (2006).

2.IrisNet: Internet -scale Resource-Intensive Sensor Network Service, <http://www.intel-iris.net>

3.Kumar, Rakesh, Naveen K. Chilamkurti, and Ben Soh. "A comparative study of different sensors for smart car park management." In Intelligent Pervasive Computing, 2007. IPC. The 2007 International Conference on, pp. 499-502. IEEE, 2007.

4.Happy Minds: SMART PARKING <http://www.happiestminds.com>

◇ <https://partheniumprojects.com/parking-system-for-smart-cities-using-ibm-watson/>

◇ <https://inspirationfeed.com/how-the-iot-will-change-our-daily-lives/#Driving>

12.APPENDIX:-

a.Source code:

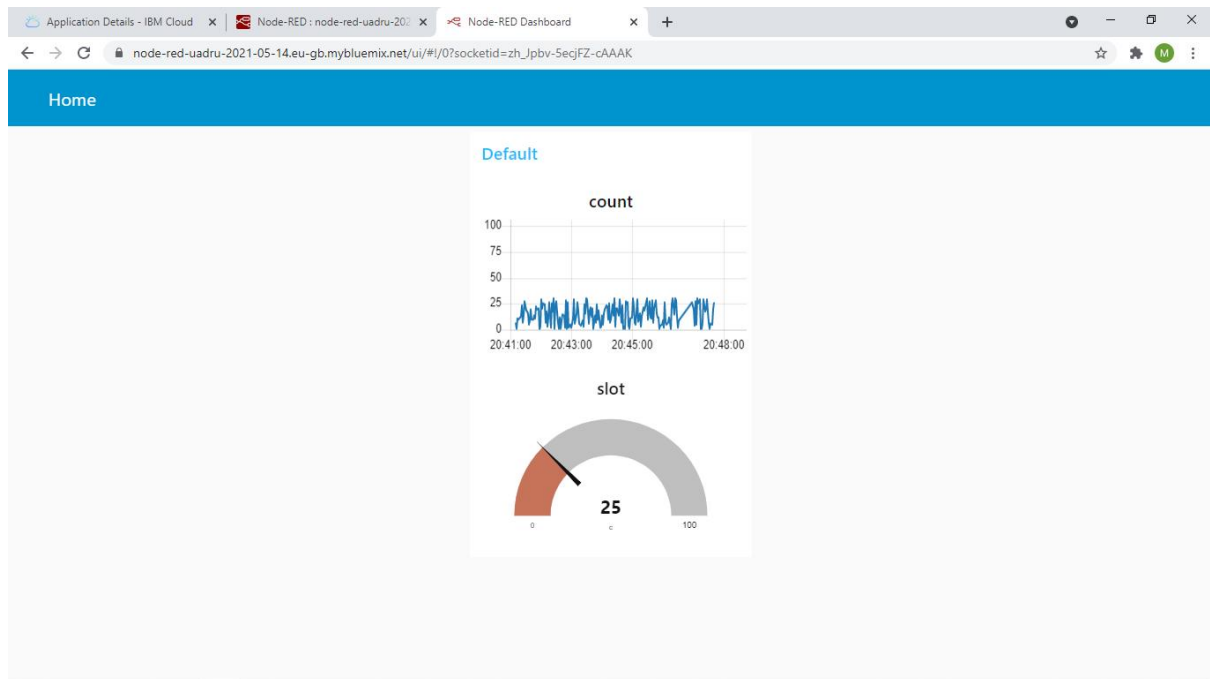
```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "h1ka6y",
        "typeId": "iotdevice",
        "deviceId": "1001"
    },
    "auth": {
        "token": "1234567890"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    slot=random.randint(0,30)
    count=random.randint(0,30)
    #slot=int(input())
    #count=int(input())
    myData={"d":{"slot":slot, 'count':count}}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print("Published data Successfully: ", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```


b.UI OUTPUT:-



MIT APP:

APP APK: <http://ai2.appinventor.mit.edu/b/b25i>

OUTPUT PHOTO:

smartparking		⋮
slot	10	
count	17	