

# **Smart Internz Externships Project Report**

## **Title: Smart Parking System for Smart Cities using IBM Watson**

### **Introduction**

#### **Overview:**

The development and high growth of the Internet of Things (IoT) have improved quality of life and strengthened different areas in society. Many cities worldwide are looking forward to becoming smart. One of the most popular use cases in smart cities is the implementation of smart parking solutions, as they allow people to optimize time, reduce fuel consumption, and carbon dioxide emissions. We have proposed a solution in which the people who has access to internet can use our mobile application/web-UI to know the status of the parking slot availability and additionally have the facility to book the slot they wish and even vacate the slot they have been using. We have made the application as user friendly and interactive experience as possible.

#### **Purpose:**

In recent years, finding a parking spot, particularly in densely populated areas such as malls and offices, has become a daily issue for many because it takes up so much of their valuable time. We proposed an unique smart parking system where people living in the vicinity of crowded cities can easily check for available parking slots using this approach, especially malls can effectively manage the parking problem during rush hours, to overcome difficulties such as waiting in long lines and circling the blocks to find a vacant parking space. This system provides convenience to customers by reducing the amount of time spent to find a vacant parking slot.

Vehicle-related traffic congestion is becoming a big issue that is rapidly worsening. Every day, over 1400 new cars enter Delhi's roads in India. People sometimes find it frustrating to look for a vacant parking space during rush hour. This has made it more difficult for people to park their cars during peak hours in congested public spaces such as restaurants and malls. Every day, we consume almost one million barrels of oil. According to a recent poll, a driver takes approximately 8 minutes to park his car because he spends more time looking for a parking spot. Thirty to forty percent of traffic congestion is caused by this searching.

According to a report, Smart Parking could result in 2,20,000 gallons of fuels saving till 2030 and approx. 3,00,000 gallons of fuel saved by 2050, if implemented successfully. Finding spaces during weekends or public holidays can take more than 10 minutes for about 66% of visitors. Our primary objective remains reducing the two of man's most important factors, cost and time. It also helps malls regulate traffic efficiently in the parking lots. Manpower is reduced considerably by directing the vehicles through our independent system.

## Literature Survey

### Existing Problems:

Paper	Work Done	Inference
<b><i>Smart Parking System for Smart Cities</i></b> <b>By:</b> Supreeth Y S , S G Raghavendra Prasad , Dr. Jitendranath Mungara <b>Year:</b> 2018 <b>DOI:</b> IJERTCONV4IS29047	In order to overcome difficulties such as waiting in long queue and circling around the blocks to find a vacant parking space, in this paper they proposed a unique system wherein people living in the vicinity of crowded streets can rent their unused parking space.	Using this approach, especially malls can productively manage the parking problem effectively during rush hours. This system also provides convenience to customers by reducing the amount of time spent to find a vacant parking slot by allowing them to reserve a vacant slot through pre-booking.
<b><i>Smart Parking Systems</i></b> <b>By:</b> Can Biyik , Zaheer Allam, Gabriele Pieri , Davide Moroni , Muftah O'Fraifer , Eoin O'Connell , Stephan Olariu and Muhammad Khalid <b>Year:</b> 2021 <b>DOI:</b> <a href="https://doi.org/10.3390/smartcities4020032">https://doi.org/10.3390/smartcities4020032</a>	The principal role of this research paper is to analyze smart parking solutions from a technical perspective, underlining the systems and sensors that are available	The project seeks to provide comprehensive insights into the building of smart parking solutions. A holistic survey of the current state of smart parking systems should incorporate the classification of such systems as big vehicular detection technologies.
<b><i>Smart Parking</i></b> <b>By:</b> Jhonattan J. Barriga , Juan Sulca , José Luis León, Alejandro Ulloa, Diego Portero, Roberto Andrade and Sang Guun Yoo <b>Year:</b> 2019 <b>DOI:</b> <a href="https://doi.org/10.3390/app9214569">https://doi.org/10.3390/app9214569</a>	This paper identifies the most used types of every component and highlights usage trends in the established analysis period. It provides a complementary perspective and represents a very useful source of information. The scientific community could use this information to decide regarding the selection of types of components to implement a smart parking solution. For this purpose, herein we review several works related to smart parking solutions deployment.	A semi-cyclic adaptation of the action research methodology combined with a systematic review is used to select papers related to the subject of study. The most relevant papers were reviewed to identify subcategories for each component; these classifications are presented in tables to mark the relevance of each paper accordingly. Trends of usage in terms of sensors, protocols and software solutions are analyzed.
<b><i>Internet of Things (IoT) Technologies for Smart Cities</i></b> <b>By:</b> Supreeth Y S , S G Raghavendra Prasad , Dr. Jitendranath Mungara <b>Year:</b> 2017 <b>DOI:</b> <a href="http://dx.doi.org/10.1049/iet-net.2017.0163">http://dx.doi.org/10.1049/iet-net.2017.0163</a>	As various smart city initiatives and projects have been launched in recent years, they have witnessed not only the expected benefits, but the risks introduced. They describe the current and future trends of smart city and IoT.	They also discuss the interaction between smart cities and IoT and explain some of the drivers behind the evolution and development of IoT and smart city. Finally, they discuss some of the IoT weaknesses and how they can be addressed when used for smart cities.
<b><i>Internet of things oriented elegant parking method for smart cities</i></b> <b>By:</b> Pradeep Bedi, Muruganantham Ponnusamy, P.Ashokkumar, S.Saranya, S.Hariharan <b>Year:</b> 2021 <b>DOI:</b> <a href="https://doi.org/10.1016/j.matpr.2021.01.178">https://doi.org/10.1016/j.matpr.2021.01.178</a>	In this paper, they presented an IoT based cloud coordinated rich parking lot framework using IR sensor and ultrasonic sensor. The proposed parking scaffolds encompass of an IoT module that is exploited to VDT and signalize the circumstance of convenience of each sole parking dot.	An OLED which consents an end patron to verify the openness of parking dot which is controlled by a Wi-Fi module. The paper additionally illustrates an elevated point perspective on the framework design.
<b><i>CIoT-Net: a scalable cognitive IoT based smart city network architecture</i></b> <b>By:</b> Jin-ho Park, Mikail Mohammed Salim, Jeong Hoon Jo, Jose Costa Sapalo Sicato,	Data captured from millions of sensors can be cross implemented across various cognitive computing applications to ensure real-time responses. In this paper, they studied the cognitive internet of things (CIoT) and propose a CIoT-based smart city network (CIoT-Net)	They discuss various technologies such as AI and big data analysis to implement the proposed architecture. Finally, they describe the possible research challenges and opportunities while implementing the proposed architecture.

<p><i>Shailendra Rathore and Jong Hyuk Park</i> <b>Year: 2019</b></p> <p><b>DOI:</b> <a href="https://doi.org/10.1186/s13673-019-0190-9">https://doi.org/10.1186/s13673-019-0190-9</a></p>	<p>architecture which describes how data gathered from smart city applications can be analyzed using cognitive computing and handle the scalability and flexibility problems.</p>	
<p><b><i>A Comprehensive Framework for Analyzing IoT Platforms: A Smart City Industrial Experience</i></b></p> <p><b>By:</b> Mahdi Fahmideh, Jun Yan , Jun Shen , Davoud Mougouei , Yanlong Zhai , and Aakash Ahmad <b>Year: 2021</b></p> <p><b>DOI:</b> <a href="https://doi.org/10.3390/app9214569">https://doi.org/10.3390/app9214569</a></p>	<p>This paper is a tentative response to this critical knowledge gap where we adopted the design science research approach to develop a novel evaluation framework. Their research, on the one hand, stimulates an unbiased competition among IoT platform providers and, on the other hand, establishes a solid foundation for IoT platform consumers to make informed decisions in this multiplicity.</p>	<p>The application of the framework is illustrated in example scenarios. Moreover, lessons learned from applying design science research are shared.</p>
<p><b><i>Cloud based Parking Management System using IoT</i></b></p> <p><b>By:</b> Muhammad Aiman Bin Azhar <b>Year: 2020</b></p> <p><b>DOI:</b> 1170028</p>	<p>The technique of smart parking will be analyzed like in the lecture review such as IoT based smart vehicle presence sensor Spin-V for smart parking system , parking guidance system based on ZigBee and Geomagnetic Sensor Technology, a cloud-based smart-parking system based on Internet-of-Things Technologies using RFID and others.</p>	<p>The sensor will collect the data of the availability of the parking spaces. Next, the mobile application will display the parking status. If the nearest parking spaces is available, the user can reserve the parking space and the data will be save in the cloud and the reservation status will be updated. The user will be notified with the status of the parking.</p>
<p><b><i>Blockchain as a Middleware for Iot Sensing and Authentication Within Smart Cities</i></b></p> <p><b>By:</b> Nada Alasbali, Saaidal Razali Azzuhri , Rosli Salleh , Miss Laiha Mat Kiah <b>Year: 2020</b></p> <p><b>DOI:</b> <a href="https://doi.org/10.21203/rs.3.rs-118074/v1">https://doi.org/10.21203/rs.3.rs-118074/v1</a></p>	<p>The SLR has divided the central concepts and structural solutions illuminated in the blockchain-IoT-related, smart city research conducted in the past four years into its core components: a structural solution of issues of security and authentication in the IoT, a tangible application of smart contract technologies for decentralized smart negotiation and a means of permissions-based performative oversight and administration.</p>	<p>Heterogeneity and proprietary technologies will revise IoT's scope and capabilities and capabilities. However, this intermediary role of the Blockchain has been proven in many experimental and exploratory works and offers a significant advantage over prior structural limitations.</p>
<p><b><i>Real-Time Smart Parking Systems Integration in Distributed ITS for Smart Cities</i></b></p> <p><b>By:</b> Muhammad Alam, Davide Moroni, Gabriele Pieri, Marco Tampucci, Miguel Gomes, José Fonseca, Joaquim Ferreira, and Giuseppe Riccardo Leone <b>Year: 2018</b></p> <p><b>DOI:</b> <a href="https://doi.org/10.1155/2018/1485652">https://doi.org/10.1155/2018/1485652</a></p>	<p>In this paper, they present a new architecture where the intelligence is distributed and the decisions are decentralized. The proposed architecture is scalable since the incremental addition of new peripheral subsystems is supported by the introduction of gateways which requires no reengineering of the communication infrastructure.</p>	<p>The proposed architecture is deployed to tackle the problem of traffic management inefficiency in urban areas, where traffic load is substantially increased, by vehicles moving around unnecessarily, to find a free parking space. This can be significantly reduced through the availability and diffusion of local information regarding vacant parking slots to drivers in a given area.</p>

### Proposed Solution:

Various research papers are considered and are analysed. Each of have its own disadvantage. So, our project will help in rectifying those disadvantages and will prove to be conventional system.

The main disadvantages in most of the works are usage of hardware devices. Some of the journals have mentioned the use of magnetometer for smart parking system. Tough magnetometer is considered to be a feasible equipment, it has its own disadvantage. One of the key disadvantages of magnetometer-based smart car parking sensors is rapidly decreasing battery life with the increasing accuracy requirements. Besides that, the modern electric vehicles often do not have ferromagnetic parts at all. Some of the research paper mentioned the use of radar sensors for smart car parking system. Radars are used widely in lots of fields. The disadvantages of radar are Radar ranging sensors that have a lower frequency and they can be less sensitive to some thin obstructions, e.g. they can be optimized to operate under thin snow cover. These types of sensors are more expensive and may require special high-frequency circuit design knowledge. In radars, signal processing requires much more MCU horsepower (shorter battery life). Since our project is software based, we don't have to deal with problem of hardware equipment. Most of this hardware equipment are more expensive and requires high maintenance. So, our project overcomes this disadvantage of feasibility and maintenance.

Some of the research papers had systems which are more complex and thus very difficult for the users to handle. In our project, we have designed the system which is very much user friendly and way easier to operate.

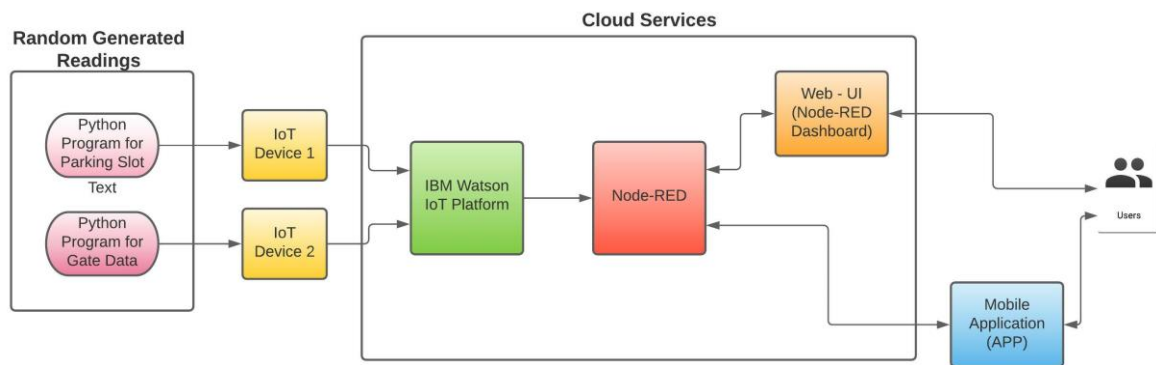
Some systems in these journals have the problem of failure during peak hours. In our project, the parking slots are well planned and the entry and exit of the cars are well monitored to prevent the commotion during peak hours.

In one of the journals, only the drivers who have booked the parking slot has access and other drivers who haven't booked due to busy slots aren't given access to monitor the status. In our project, the drivers who are waiting for a free slot to book can monitor the status and book once it is free.

Hence, by overcoming these disadvantages, the system we have designed proves to be more useful smart car parking system.

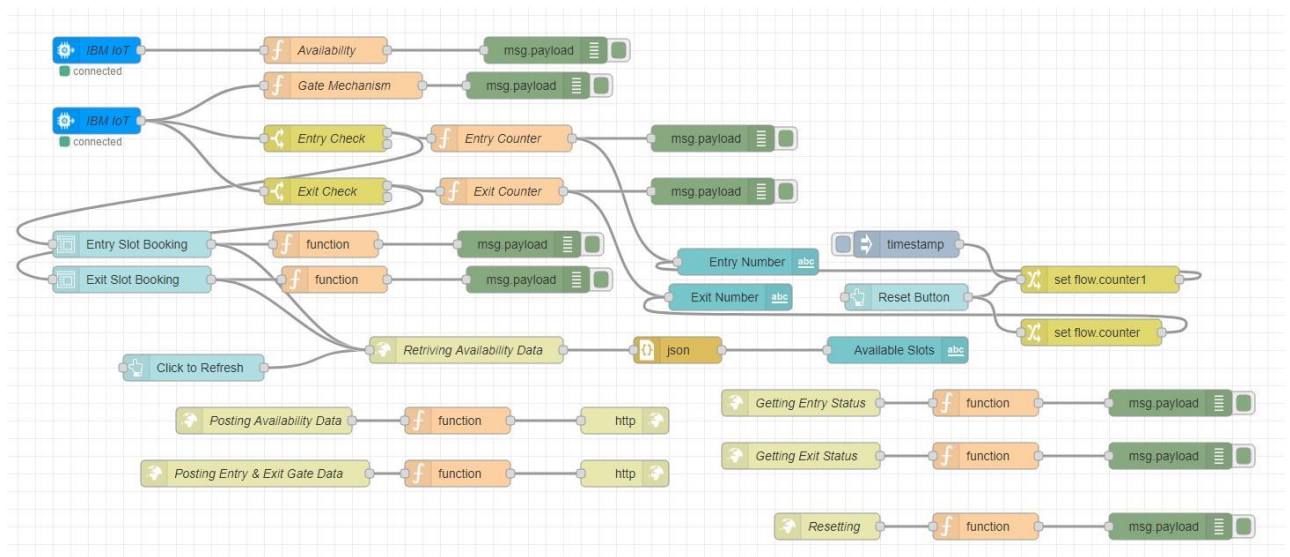
## Theoretical Analysis

### Block Diagram:

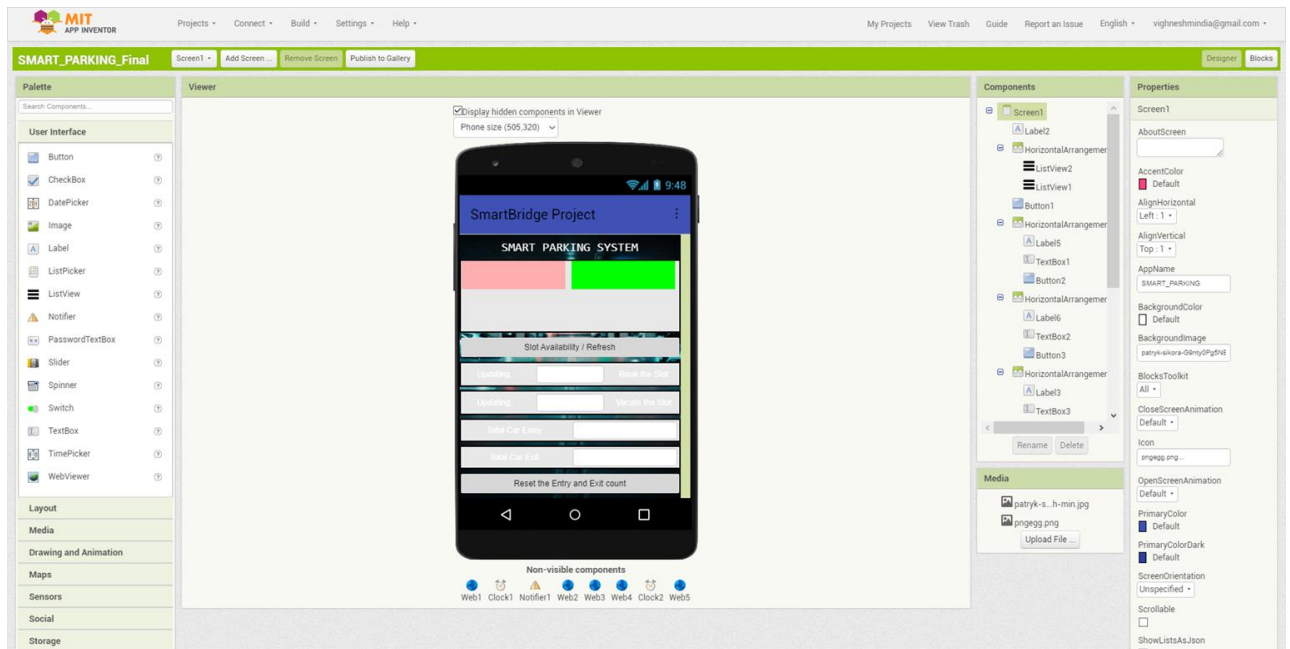


*Fig 1: Overall Block Diagram of the system*

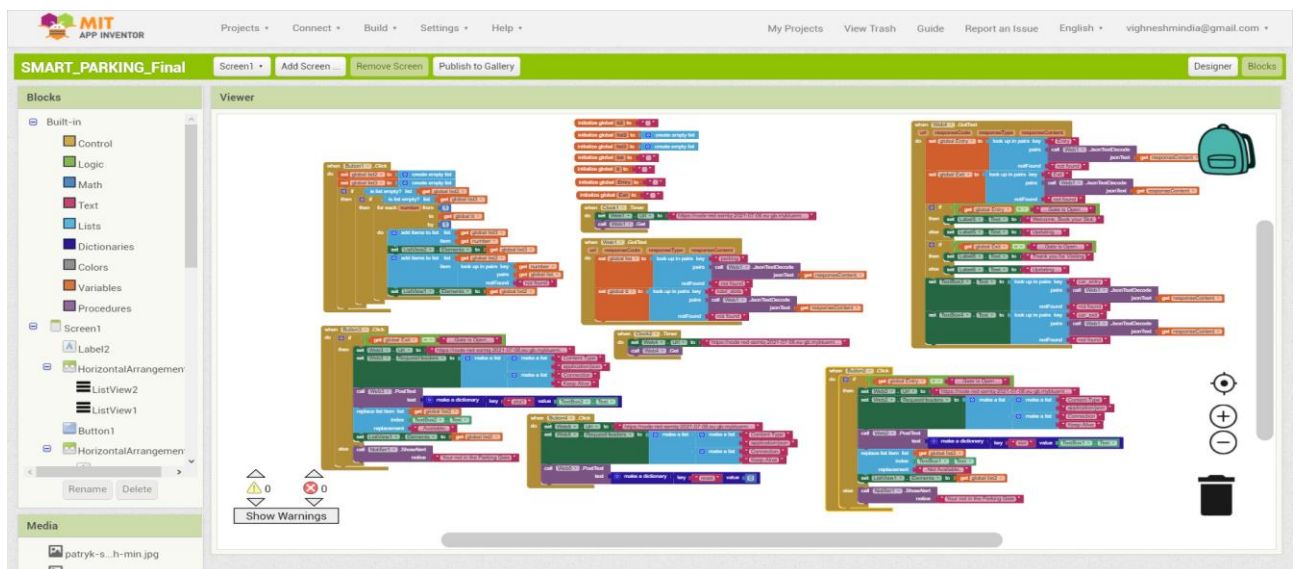
### Software Designing:



*Fig 2: NODE-RED flow*



*Fig 3: MIT APP builder designer*



*Fig 4: MIT APP Blocks*

## Experimental Investigation

The system is made up of four main components: Python code for generating random data, which is needed as an input to the project, and the system itself. Second, IBM Watson offers Node-RED, a cloud service. Almost all calculation and back-end activities are carried out here. Third, the MIT APP is the user's front end. Finally, Web UI refers to a website developed with the help of the user interface in node-RED

The project's explanation is as follows: as shown in the flowchart(fig5), the python code will create random boolean data for the parking slots based on the input provided by the owner regarding how many slots are accessible in the mall/place. The python will generate a dictionary containing information about parking spots as well as the overall number of slots available. Another dictionary containing boolean for each slot is parking slot data (1 for occupied and 0 if it is not occupied). The IBM IoT platform's Devices are used to send this data to

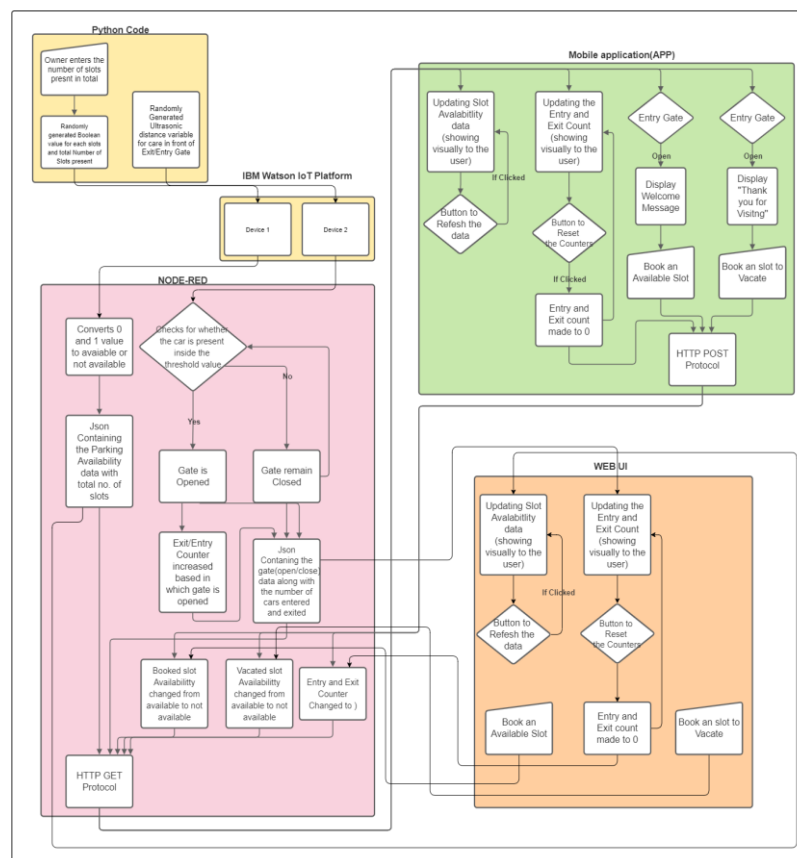
Node-RED. The data is sent in the form of json by these devices. The mechanism in node-red transforms boolean data into a string that the user can comprehend (1 is converted into "Not Available" and 0 is converted into "Available"). The data is then transferred to the Web UI and the MIT App builder, where it is finally presented in the Front-end app and Web UI.

The second python script formulated a series of ultrasonic distance values at random. The distance between the entry/exit gate and the automobile entering or departing the location is measured. After sending the json value to the backend, the distance is verified to see whether it is less than 500cm. The gate will open if the distance is less than 500cm, and it will remain closed if it is more. The system will keep track of how many automobiles arrive and depart the location based on this information. All of these calculated values are translated to json and delivered to the MIT app over HTTP. The same data will be sent to the Web-UI as well.

The user will be able to see the availability data on the Mobile Application, as well as a button to refresh the data. The software will also show the total number of cars arriving and departing the location, as well as an option to reset the count to zero.

The text box (input box) available in the mobile application can be used by the user to reserve a space or even relinquish an occupied slot. The user may only book or vacate a slot if the entry or exit gate is open, otherwise it will indicate that the user must be present at the gate to book/vacate the slot. The software is clever enough to display a message to the user next to the booking and departing input forms. If the user enters the location, a welcome message will be displayed, and if the user exits, a "Thank you for Visiting" message will be displayed; this is only an added function.

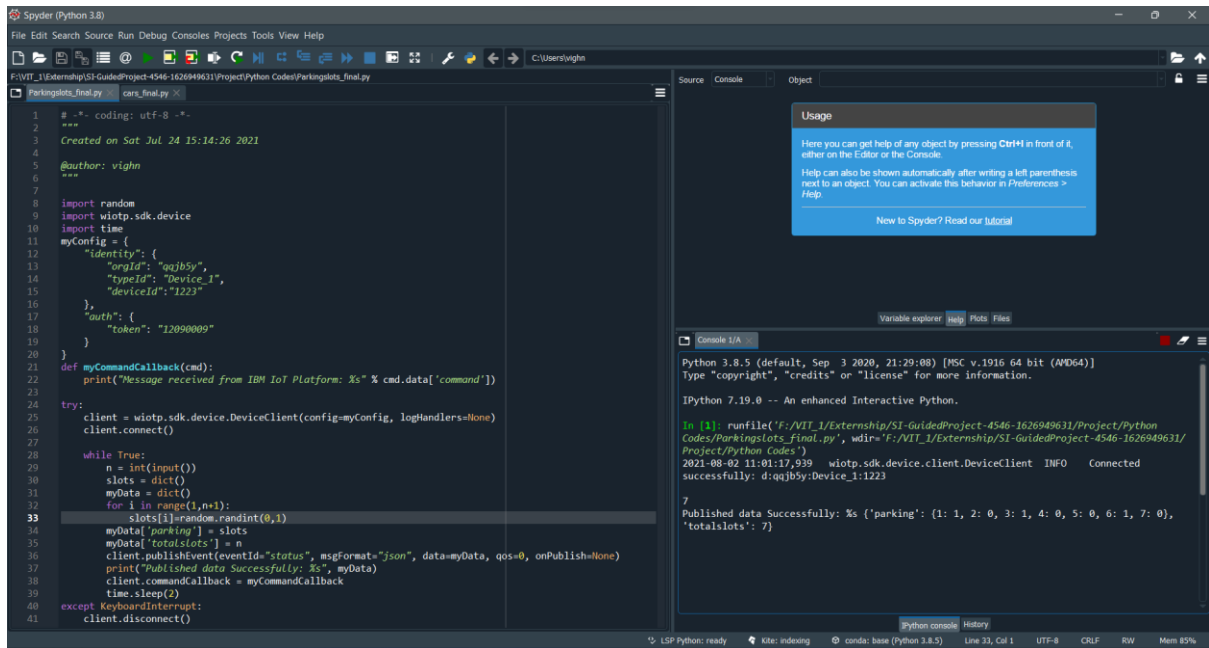
## Flowchart



*Fig 5: Overall flowchart*



## Result



The screenshot shows the Spyder Python IDE interface. The editor displays a Python script named `cars_final.py` with the following content:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sat Jul 24 15:14:26 2021
4
5 @author: vighn
6 """
7
8 import random
9 import wiotp.sdk.device
10 import time
11 myConfig = {
12     "identity": {
13         "orgId": "qqjbsy",
14         "typeId": "Device_1",
15         "deviceId": "1223"
16     },
17     "auth": {
18         "token": "12090009"
19     }
20 }
21 def myCommandCallback(cmd):
22     print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
23
24 try:
25     client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
26     client.connect()
27
28     while True:
29         n = int(input())
30         slots = dict()
31         myData = dict()
32         for i in range(1,n+1):
33             slots[i] = random.randint(0,1)
34         myData['parking'] = slots
35         myData['totalSlots'] = n
36         client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
37         print("Published data Successfully: %s" % myData)
38         client.commandCallback = myCommandCallback
39         time.sleep(2)
40 except KeyboardInterrupt:
41     client.disconnect()
```

The console output shows the following:

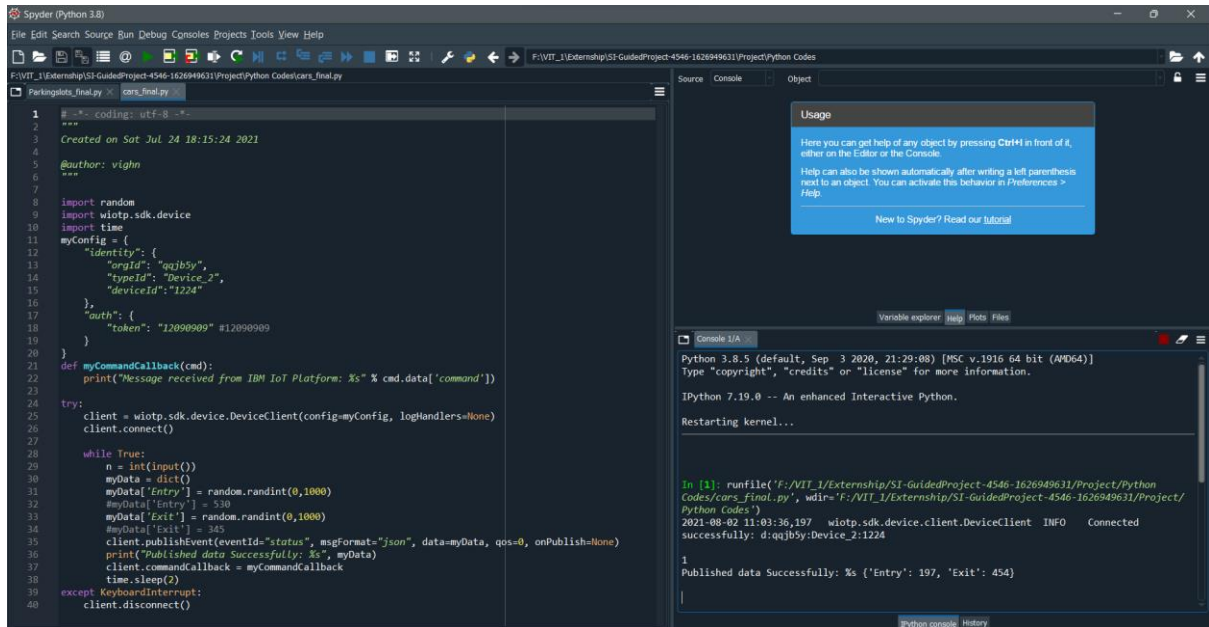
```
Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]: runfile('F:/VIT_1/Externship/SI-GuidedProject-4546-1626949631/Project/Python Codes/Parkingslots_final.py', wdir='F:/VIT_1/Externship/SI-GuidedProject-4546-1626949631/Project/Python Codes')
2021-08-02 11:01:17,939 wiotp.sdk.device.client.DeviceClient INFO Connected
successfully: d:qqjbsy:Device_1:1223

7
Published data Successfully: %s {'parking': {(1: 1, 2: 0, 3: 1, 4: 0, 5: 0, 6: 1, 7: 0), 'totalSlots': 7)}
```

*Fig 6: The Python script generates the random Boolean data for parking slot in json form*



The screenshot shows the Spyder Python IDE interface. The editor displays a Python script named `cars_final.py` with the following content:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sat Jul 24 18:15:24 2021
4
5 @author: vighn
6 """
7
8 import random
9 import wiotp.sdk.device
10 import time
11 myConfig = {
12     "identity": {
13         "orgId": "qqjbsy",
14         "typeId": "Device_2",
15         "deviceId": "1224"
16     },
17     "auth": {
18         "token": "12090009" #12090009
19     }
20 }
21 def myCommandCallback(cmd):
22     print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
23
24 try:
25     client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
26     client.connect()
27
28     while True:
29         n = int(input())
30         myData = dict()
31         myData['Entry'] = random.randint(0,1000)
32         myData['Entry'] = 330
33         myData['Exit'] = random.randint(0,1000)
34         myData['Exit'] = 345
35         client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
36         print("Published data Successfully: %s" % myData)
37         client.commandCallback = myCommandCallback
38         time.sleep(2)
39 except KeyboardInterrupt:
40     client.disconnect()
```

The console output shows the following:

```
Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

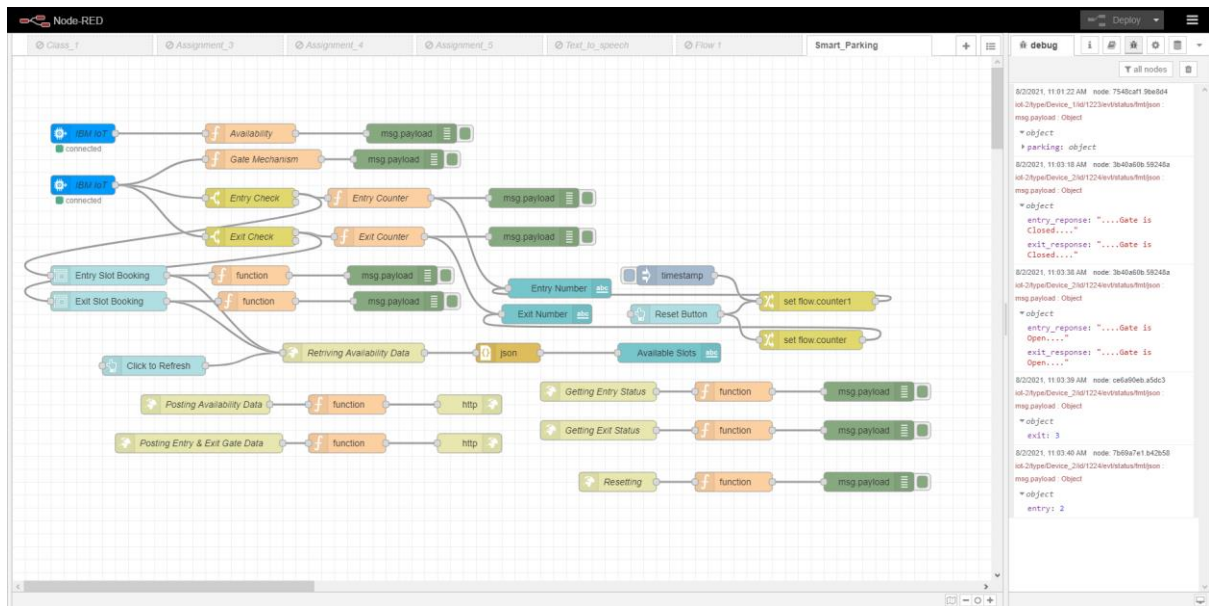
Restarting kernel...

In [1]: runfile('F:/VIT_1/Externship/SI-GuidedProject-4546-1626949631/Project/Python Codes/cars_final.py', wdir='F:/VIT_1/Externship/SI-GuidedProject-4546-1626949631/Project/Python Codes')
2021-08-02 11:03:36,197 wiotp.sdk.device.client.DeviceClient INFO Connected
successfully: d:qqjbsy:Device_2:1224

1
Published data Successfully: %s {'Entry': 197, 'Exit': 454}
```

*Fig 7: The Python script generates the random ultrasonic data for gate entry and exit in json form*





*Fig 8: The data sent by the python is received in the Node-RED (backend) and converted to a user untestable form*

### Smart Parking

Available Slots  
 {"1":"..Not Available..","2":"..Not Available..","3":"..Not Available..","4":"..Not Available..","5":"..Available..","6":"..Not Available..","7":"..Available.."}

CLICK TO REFRESH

Entry Number 2

Exit Number 3

Entry Slot Booking

Book Entry Slot \*

SUBMIT CANCEL

Exit Slot Booking

Book Exit Slot \*

SUBMIT CANCEL

RESET BUTTON

*Fig 9: The Availability data is displayed in the Web-UI*

SmartBridge Project

SMART PARKING SYSTEM

1	..Not Available..
2	..Available..
3	..Not Available..
4	..Available..
5	..Available..
6	..Not Available..
7	..Available..

Slot Availability / Refresh

Welcome, Book your Slot:

Enter the slot num:

Book the Slot

Thank you for Visiting

Enter the slot num:

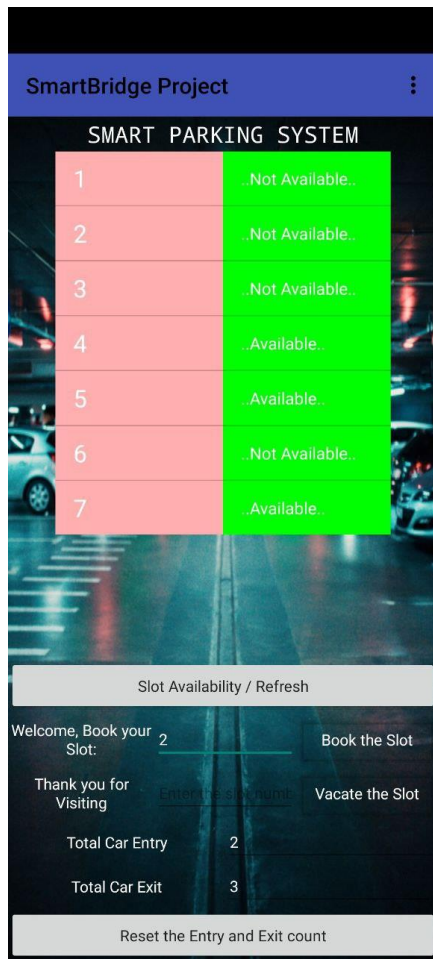
Vacate the Slot

Total Car Entry 2

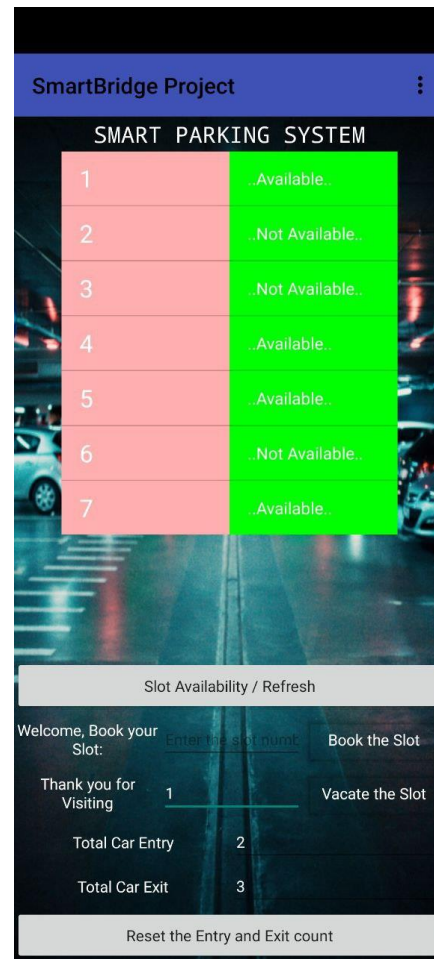
Total Car Exit 3

Reset the Entry and Exit count

*Fig 10: The Availability data is displayed in the Mobile App*



*Fig 11: Case1: The user uses the text box input to book the available slot(2) and we can see that the above table has updated accordingly*



*Fig 12: Case2: The user uses the text box input to vacate the not-available slot(occupied)(1) and we can see that the above table has updated accordingly*

Smart Parking

Available Slots  
{"1":"..Available..","2":"..Not Available..","3":"..Not Available..","4":"..Available..","5":"..Available..","6":"..Not Available..","7":"..Available.."}

CLICK TO REFRESH

Entry Number 2

Exit Number 3

Entry Slot Booking

Book Entry Slot "

SUBMIT CANCEL

Exit Slot Booking

Book Exit Slot "

SUBMIT CANCEL

RESET BUTTON

*Fig 13: Both the updated case 1 and case 2 is also visible in the Web-UI if you click on the "click to refresh button"*

Smart Parking

Available Slots  
{"1":"Available..","2":"Not Available..","3":"Not Available..","4":"Available..","5":"Not Available..","6":"Not Available..","7":"Available.."}  
CLICK TO REFRESH

Entry Number 2

Exit Number 3

Entry Slot Booking  
Book Entry Slot \*  
5

SUBMIT CANCEL

Exit Slot Booking  
Book Exit Slot \*  
6

SUBMIT CANCEL

RESET BUTTON

*Fig 14: Case3: The user uses the text box input to book the available slot(5) and we can see that the above json value has updated accordingly*

Smart Parking

Available Slots  
{"1":"Available..","2":"Not Available..","3":"Not Available..","4":"Available..","5":"Not Available..","6":"Available..","7":"Available.."}  
CLICK TO REFRESH

Entry Number 2

Exit Number 3

Entry Slot Booking  
Book Entry Slot \*  
5

SUBMIT CANCEL

Exit Slot Booking  
Book Exit Slot \*  
6

SUBMIT CANCEL

RESET BUTTON

*Fig 15: Case4: The user uses the text box input to vacate the not-available slot(occupied)(6) and we can see that the above json value has updated accordingly*

SmartBridge Project

SMART PARKING SYSTEM

1	Available..
2	Not Available..
3	Not Available..
4	Available..
5	Not Available..
6	Available..
7	Available..

Slot Availability / Refresh

Welcome, Book your Slot:  Entry Slot Number

Thank you for Visiting  Entry Slot Number

Total Car Entry 2

Total Car Exit 3

Reset the Entry and Exit count

*Fig 16: Both the updated case 1 and case 2 is also visible in the Web-UI if you click on the "click to refresh button"*

SmartBridge Project

SMART PARKING SYSTEM

1	Available..
2	Not Available..
3	Not Available..
4	Available..
5	Not Available..
6	Available..
7	Available..

Slot Availability / Refresh

Welcome, Book your Slot:  Entry Slot Number

Thank you for Visiting  Entry Slot Number

Total Car Entry 0

Total Car Exit 0

Reset the Entry and Exit count

*Fig 17: If the user wishes to rest this count, he can do that by pressing "rest entry and exit count"*

Smart Parking

Available Slots  
 {"1":"Available..","2":"Not Available..","3":"Not Available..","4":"Available..","5":"Not Available..","6":"Available..","7":"Available.."}

CLICK TO REFRESH

Entry Number

Exit Number

Entry Slot Booking  
 Book Entry Slot \*

SUBMIT CANCEL

Exit Slot Booking  
 Book Exit Slot \*  
 6

SUBMIT CANCEL

RESET BUTTON

*Fig 18: If the user wishes to rest this count, he can do that by pressing “rest button” and it will reflect the same in the web application*

## Advantage and Disadvantage

### Advantages:

- There is a greater sense of security due to the fact that patrons do not actually walk to and from their own space.
- It is highly feasible for extremely small sites that are unable to accommodate a conventional ramped parking structure.
- There is high parking efficiency (i.e., sf/space and cf/space).
- There is no need for driving while looking for an available space.
- Emissions are greatly brought down and reduced.
- The patrons wait for their car in a highly controlled environment.
- There are less chances for vehicle vandalism.
- There is a minimal staff requirement if it is used by known parkers.
- It is possible that the retrieval time is lower than the combined driving/parking/walking time in conventional ramped parking structures.
- There is an easier facade integration since there are no ramping floors or openings in exterior walls

### Potential Disadvantages:

- There is a greater construction cost per space (but this may be offset by the chance for lesser land costs per space and the system manufacturers say that the operating and maintenance cost will be lower as compared to a conventional ramped parking structure).
- Use of redundant systems will result in a greater cost.
- It may be a bit confusing for unfamiliar users.
- It is not recommended for high peak hour volume facilities.
- There may be a fear of breakdown (How do I get my car out?).
- There is an uncertain building department review and approval process.
- It requires a maintenance contract with the supplier.

## Applications

There are already many free smart parking applications available online in web and mobile stores of Android or iOS. Previously, reservation of parking space was done by calling the service provider and now with the current usage of internet and smartphones, these services are provided online using mobile and web applications. These applications serve as decision support systems for the driver in occupying a vacant parking space. For instance, if the application shows a particular parking lot of choice to be full, the driver can search for nearby parking lots with available parking spaces or choose another destination. In this way smart parking applications serve as decision support systems in occupying available parking spaces. Smart parking tools improve efficiency based on the following three categories.

1. Guide the driver to parking lot using display boards
2. Reserve and authorize the driver to a parking lot
3. Reserve and guide the driver to a specific parking space using navigational information

The three categories mentioned earlier are the ways in which efficiency of parking is improved using existing smart parking tools. The level of efficiency varies with each category. The first category would improve the parking efficiency, however cruising for the empty parking space would still be involved. The second category reserves a parking space for the driver; however, the driver would still search for the reserved parking space manually which also involves some amount of cruising while searching for the location of reserved parking space. The third category provides improved efficiency compared to the two previous categories as it facilitates in reserving and guiding the driver using navigational information to the parking space

Further the implemented technology in this project can be used for several other purposes and they are as follows:

1. ***Malls and Cinema Theatre Parking System***
2. ***Corporate Offices***
3. ***Smart Logistics***
  - a. Fleet Tracking
  - b. Platooning
  - c. Connected Vehicles
4. ***Smart Retail***
  - a. Supply Chain Control
  - b. Near Field Communication (NFC) Payment
  - c. Layout Optimization
  - d. Smart Product Management
5. ***Smart Factories***
  - a. Enterprise Asset management
  - b. Predictive maintenance
  - c. Industrial process automation/optimization
  - d. Energy Management
6. ***Asset tracking***
  - a. Locate and monitor key assets
  - b. Tracking supply chain
  - c. Optimizing logistics

- d. Maintain inventory levels
- e. Prevent quality issues
- f. Detect theft

## **Conclusion**

Thus we have seen a real time implementation of Smart Parking Systems for Smart Cities using technologies such as IBM Watson, Node-RED, MIT App Inventor and the base coding language for this was python which was used to generate random values. The above proposed system can be further improved and improvised with faster and efficient applications by using RFID Tags under peak hours for making the traffic move quickly and smoothly.

## **Future Scope**

The introduction of autonomous cars (AVs) is likely to have a big impact on the future of the smart parking industry. Self-parking automobiles, dedicated AV parking lots, and robotic parking valets are already being tested in a number of locations across the world.

For example, in Boulder, Colorado, ParkPlus is working on deploying a fully automated parking garage in the Western United States through Boulder's PearlWest mixed-use development. The company's automated parking system uses lasers to scan cars and a robotic valet to park the vehicles. Vehicles are transported by a robotic dolly that lifts and transfers them to storage racks. Using this system, up to 4 times as many cars can be parked in the same amount of space as a traditional garage (since there is no need for extra space in between cars). The automated system is expected to deliver vehicles within 3-5 minutes of a retrieval request.

## Bibliography

- [1] Supreeth Y S , S G Raghavendra Prasad , Dr. Jitendranath Mungara, 2016, Smart Parking System for Smart Cities, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) ICIOT – 2016 (Volume 4 – Issue 29),
- [2] Biyik, C.; Allam, Z.; Pieri, G.; Moroni, D.; O’Fraifer, M.; O’Connell, E.; Olariu, S.; Khalid, M. Smart Parking Systems: Reviewing the Literature, Architecture and Ways Forward. Smart Cities 2021, 4, 623-642.
- [3] Barriga, J.J.; Sulca, J.; León, J.L.; Ulloa, A.; Portero, D.; Andrade, R.; Yoo, S.G. Smart Parking: A Literature Review from the Technological Perspective. Appl. Sci. 2019, 9, 4569.
- [4](PDF) Internet of Things (IoT) Technologies for Smart Cities (researchgate.net)
- [5] Pradeep Bedi, Muruganantham Ponnusamy, P. Ashokkumar, S. Saranya, S. Hariharan, Internet of things oriented elegant parking method for smart cities, Materials Today: Proceedings, 2021, ISSN 2214-7853,
- [6] Park, Jh., Salim, M.M., Jo, J.H. et al. CIoT-Net: a scalable cognitive IoT based smart city network architecture. Hum. Cent. Comput. Inf. Sci. 9, 29 (2019).
- [7]Type of the Paper (Article (arxiv.org))
- [8]The current technique of smart parking will be analyzed like in the lecture | Course Hero
- [9]Blockchain as a Middleware for Iot Sensing and Authentication Within Smart Cities: A Systematic Literature Review - [scite report]
- [10] JOUR, Osaba, Eneko, Alam, Muhammad, Moroni, Davide, Pieri, Gabriele, Tampucci, Marco, Gomes, Miguel, Fonseca, José, Ferreira, Joaquim, Leone, Giuseppe Riccardo, 2018, 2018/10/21, Real-Time Smart Parking Systems Integration in Distributed ITS for Smart Cities, 1485652, 2018
- [11] Paidi, Vijay & Fleyeh, Hasan & Håkansson, Johan & Nyberg, Roger. (2018). Smart parking sensors, technologies and applications for open parking lots: A Review. IET Intelligent Transport Systems. 12. 10.1049/iet-its.2017.0406.

## Team Members

### Member 1:

- **Name:** Vighnesh M
- **Registration No:** 18BEC1223
- Mail ID: [Vighnesh.m2018@vitstudent.ac.in](mailto:Vighnesh.m2018@vitstudent.ac.in)
- GitHub Project Link: <https://github.com/smartinternz02/SI-GuidedProject-4546-1626949631>

### Member 2:

- **Name:** Jivika S
- **Registration No:** 18BLC1099
- Mail ID: [jivika.s2018@vitstudent.ac.in](mailto:jivika.s2018@vitstudent.ac.in)
- GitHub Project Link: <https://github.com/smartinternz02/SI-GuidedProject-4549-1626949854>

### Member 3:

- **Name:** Andrew John
- **Registration No:** 18BEC1278
- Mail ID: [andrewjohn.j2018@vitstudent.ac.in](mailto:andrewjohn.j2018@vitstudent.ac.in)
- GitHub Project Link: <https://github.com/smartinternz02/SI-GuidedProject-4547-1626949646.git>



#### Member 4:

- **Name:** keshanapalli yashwanth chowdary
- **Registration No:** 18BIS0096
- Mail ID: [yashwanth.chowdary2018@vitstudent.ac.in](mailto:yashwanth.chowdary2018@vitstudent.ac.in)
- GitHub Project Link: <https://github.com/smartinternz02/SI-GuidedProject-4548-1626949852>

## Appendix

#### Source code:

##### Python Codes:

##### Parking Slot Availability Generation:

```
"""
Created on Sat Jul 24 15:14:26 2021

@author: vighn
"""

import random
import wiotp.sdk.device
import time
myConfig = {
    "identity": {
        "orgId": "qqjb5y",
        "typeId": "Device_1",
        "deviceId": "1223"
    },
    "auth": {
        "token": "12090009"
    }
}

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

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

```

while True:

    n = int(input())

    slots = dict()

    myData = dict()

    for i in range(1,n+1):

        slots[i]=random.randint(0,1)

    myData['parking'] = slots

    myData['totalslots'] = n

    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)

    print("Published data Successfully: %s", myData)

    client.commandCallback = myCommandCallback

    time.sleep(2)

except KeyboardInterrupt:

    client.disconnect()

```

### Entry & Exit Gate Generation:

```

"""

Created on Sat Jul 24 18:15:24 2021

@author: vighn

"""

import random

import wiotp.sdk.device

import time

myConfig = {

    "identity": {

        "orgId": "qqjb5y",

        "typeId": "Device_2",

        "deviceId": "1224"

    },

    "auth": {

        "token": "12090909" #12090909

    }

}

def myCommandCallback(cmd):

```

```

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

try:

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

    client.connect()

while True:

    n = int(input())

    myData = dict()

    myData['Entry'] = random.randint(0,1000)

    #myData['Entry'] = 530

    myData['Exit'] = random.randint(0,1000)

    #myData['Exit'] = 345

    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)

    print("Published data Successfully: %s", myData)

    client.commandCallback = myCommandCallback

    time.sleep(2)

except KeyboardInterrupt:

    client.disconnect()

```

## UI output Screenshot:

### WEB UI:

Home

### Smart Parking

Available Slots  
{"1": "Available", "2": "Available", "3": "Available", "4": "Available", "5": "Not Available", "6": "Available", "7": "Not Available"}

CLICK TO REFRESH

Entry Number 1

Exit Number 1

Entry Slot Booking

Book Entry Slot \*

SUBMIT CANCEL

Exit Slot Booking

Book Exit Slot \*

SUBMIT CANCEL

RESET BUTTON

## Mobile Application UI:

