SMARTINTERNZ-EXTERNSHIP PROGRAM

Internet Of Things

SMART PARKING SYSTEM

PROJECT REPORT

Submitted by:

ROHITAA R(20BEC0634), VIT VELLORE

JEFF EASOW (20BCT0028), VIT VELLORE

ADNAN SAINUDHEEN VADAKKAN (20BEC0748), VIT VELLORE

ADDANKI SREEKAR (20BCY10127), VIT BHOPAL

1. INTRODUCTION

1.1 Overview

The Internet of Things (IoT) is a network of physical objects, including devices, cars, buildings, and other things, that are connected to the internet and equipped with electronics, software, ultrasonic sensors, and network connectivity. The Internet of Things (IoT) makes it possible for objects to be sensed and controlled remotely across existing network infrastructure, opening opportunities for a more direct integration of the physical world into computer-based systems and producing improvements in efficiency, accuracy, and economic benefit. When IoT is enhanced with sensors and actuators, the technology becomes an example of the more general class of cyber-physical systems, which also includes technologies like smart grids, smart homes, and smart cities. Through its embedded computing system, each object can be uniquely identified, but it can also communicate with other things using the current Internet infrastructure. The goal of smart parking is to achieve faster, easier, and denser parking of automobiles during the bulk of the time they are stationary while utilising the least number of resources (such as fuel, time, and space). Because of heavy traffic and available parking, it is now quite difficult to locate a place to park a car in the city. Smart parking systems are being used to locate parking for vehicles in this era of technology. This helps you save time and fuel. On the Android app, we will additionally display the parking space's current location. To create an Android app, we'll use the MIT App Inventor tool. The Android app uses the real-time database to display the parking space's current location in real time. Wokwi turns your source code into a binary firmware and runs the binary firmware as a real microcontroller would, one instruction at a time.

1.2 Purpose

Drivers are directed directly to available parking spaces in metropolitan areas with integrated smart parking solutions. This removes the need to travel farther distances in search of parking spaces. Drivers may be able to save time and money by using smart parking solutions. Smart parking also lowers carbon emissions from automobiles by reducing traffic and the mobility of the vehicles looking for space. The environmental impact of each driver increases when they move from one parking spot to another. The discharge of individual environmental footprint, especially the release of carbon dioxide, is ultimately decreased when smart parking solutions are integrated into metropolitan settings. Most individuals stay away from the crowded areas of the city because they don't want to be caught in the parking woes that make drivers anxious and stressed. The goal of smart parking solutions is to relieve drivers' tension when they are parking. In the end, this improves traffic flow and attempts to reduce as little as possible the amount of search traffic on the streets.

2.LITERATURE SURVEY

2.1 Existing problem

Today's drivers and transportation systems face significant issues related to specific parking systems, for which smart city engineers and planners must be prepared. According to several recent studies, new smart parking systems are required in nearly every major city in the world, particularly in the next ten years, to address several issues, including fuel consumption and pollution emission as well as to improve timesaving and lessen frustration when looking for a parking space. Because of this, any proposed solution must at a minimum meet the following criteria in order to be considered smart in regard to the parking process:

- Be able to detect vehicle occupancy quickly and reliably.
- Users should receive information regarding parking options.
- Simplify the parking process and increase value for parking users, including drivers.
- Enabling the use of data for informed decision-making, including historical analytic reports and real-time status apps.
- Be able to give the user all the information they require regarding the status of any alterations to the parking space that may take place in real time.

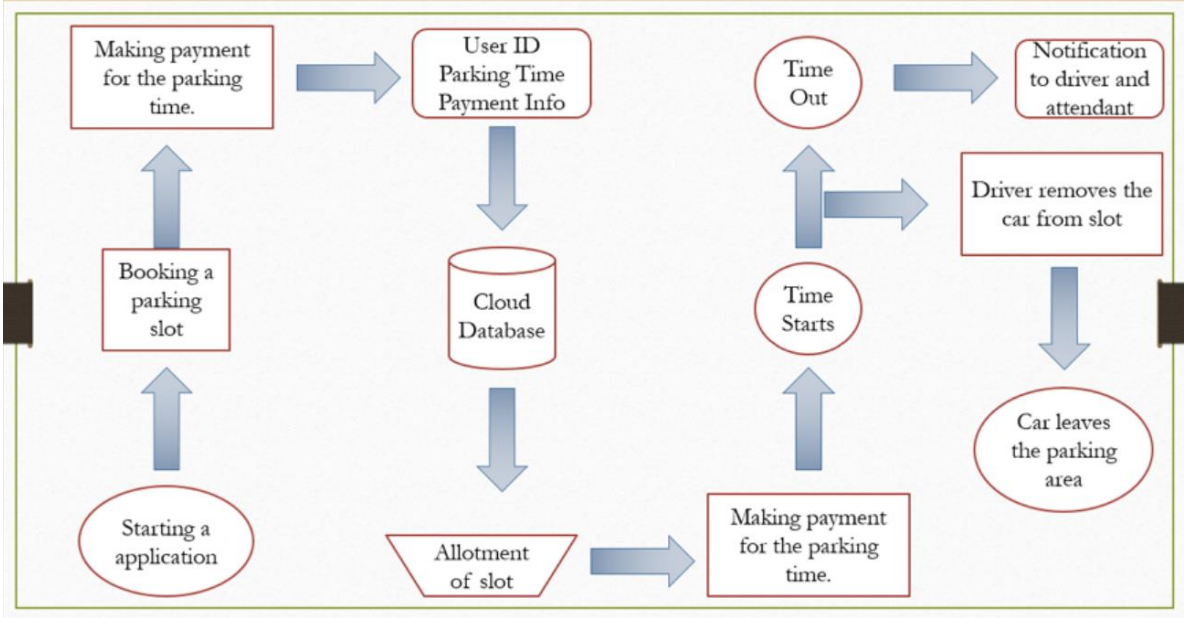
To guarantee that the system will function effectively, these issues must be dealt with right away. In the past ten years, several research on conventional smart parking systems have found that they are neither cost-effective nor meet drivers' needs. Recent studies in large cities have shown that there are various perspectives from which to approach the issue of parking management. roads with a high vehicle density. This makes it incredibly challenging for cars to find parking spaces, which is an inconvenient problem. Typically, drivers waste time and energy looking for parking spaces and end up finding a space for their vehicles to park on the streets. In the worst situation, customers are unable to find a parking space, particularly during busy times and holiday seasons.

2.2 Proposed solution

The IoT module that is deployed on-site as part of the planned smart parking system is used to track and signalise the availability of each individual parking space. Additionally, a smartphone application is offered, enabling users to check the availability of parking spaces and reserve a spot accordingly. There is an Internet of Things device with sensors and microcontrollers in every parking space. The customer receives up-to-the-minute information on the availability of every parking spot, giving them the choice to pick the best one. In metropolitan regions where parking is frequently difficult, this approach alone starts a domino effect of advantages, from less traffic congestion to reduced fuel efficiency.

3.THEORITICAL ANALYSIS

3.1 Block diagram



3.2 Hardware / Software designing

3.2.1 Wokwi

i. Breadboard

Temporary circuits are constructed using a breadboard, often known as a plug block. Designers may quickly remove and change components thanks to its usefulness. It is helpful for someone who wants to construct a circuit to show how it works before reusing the parts in another circuit.

ii. ESP-32

The ESP32 family of system on a chip microcontroller features integrated Wi-Fi and dual-mode Bluetooth and is inexpensive and low power.

iii. Servo motor

Servo motors, or "servos" as they are sometimes referred to, are electronic gadgets and rotary or linear actuators that precisely rotate and push elements of a machine. Servos are mostly utilised for linear or angular position, as well as for a set speed and acceleration.

iv. Ultrasonic sensor

An ultrasonic sensor is a device that uses ultrasonic sound waves to calculate a distance to an item. An ultrasonic sensor transmits and receives ultrasonic pulses from a transducer to determine the proximity of an item.

v. Led

A semiconductor wafer or diode, a wire bond, a reflecting cavity, a lead frame with two leads, an anode, and a cathode are all components of an

LED light. The semiconductor wafer or diode that is covered with impurities is the essential component of an LED light.

3.2.2 IBM Cloud

The IBM Cloud platform combines platform as a service (PaaS) with infrastructure as a service (IaaS) to provide an integrated experience. The platform scales and supports both small development teams and organizations, and large enterprise businesses.

3.2.3 MIT App Inventor

MIT App Inventor is an intuitive, visual programming environment that allows everyone even children to build fully functional apps for smartphones and tablets. Those new to MIT App Inventor can have a simple first app up and running in less than 30 minutes.

4.EXPERIMENTAL INVESTIGATIONS

In this section, we discuss the system's implementation and operation in a real-world setting. The following flow chart can also be built to show the entire process of reserving a parking space, parking a vehicle in that space, and leaving the parking area. The "Smart Parking" technology will be available to consumers in their mobile devices as we increase the testing in a real-world setting.

The steps that a driver needs to follow to park its car using our parking system.

Step 1: Install the smart parking application on your mobile device.

Step 2: With the help of the mobile app look for a parking area for the vacant space available.

Step 3: Select a particular parking area.

Step 4: Select the amount of time (in hours) for which you would like to park your car for.

Step 5: After booking OTP is generated.

Step 6: At the time of entry, you need to enter the OTP through keypad.

Step 7: Gate will be opened only if OTP is valid.

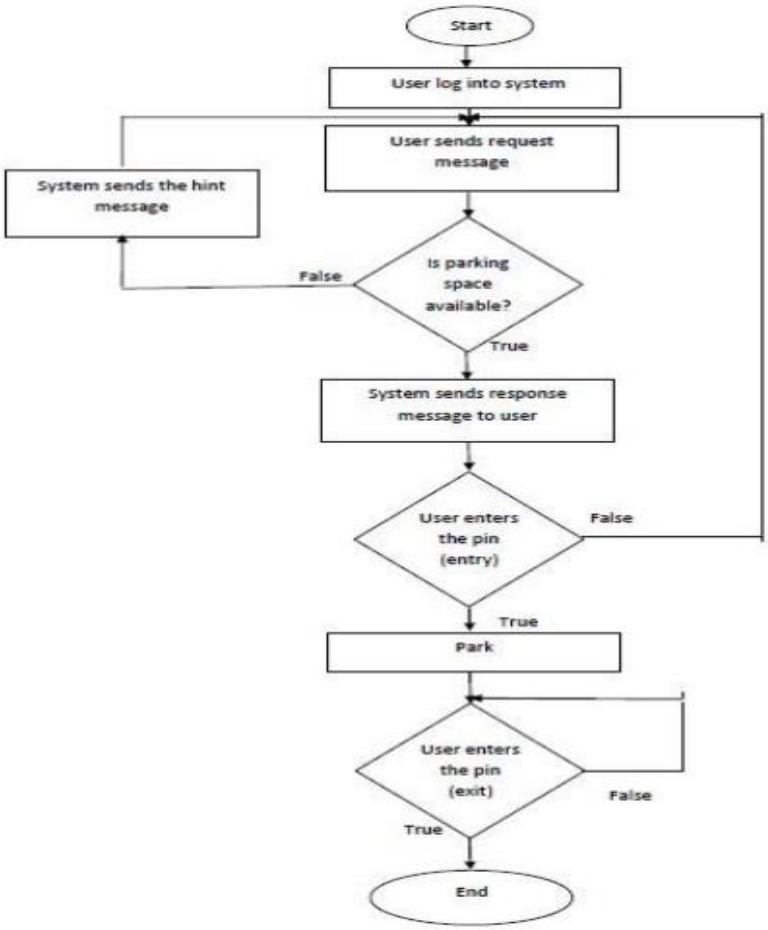
Step 8: Park your car.

Step 9: At the time of exit again enter OTP.

Step 10: You will get the total payable amount (along with fine if any) on your android app.

Step 11: Pay the parking charges.

5.FLOWCHART



6.RESULT

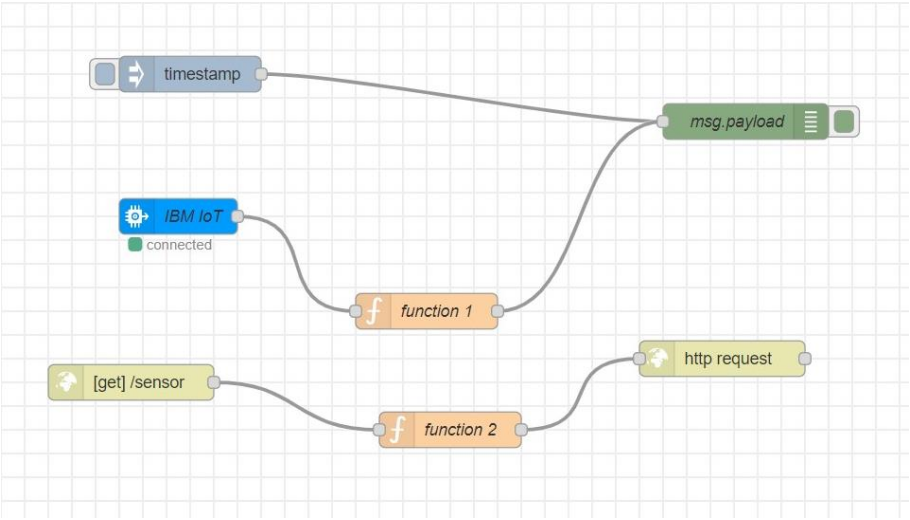


Fig.1 Node red flow

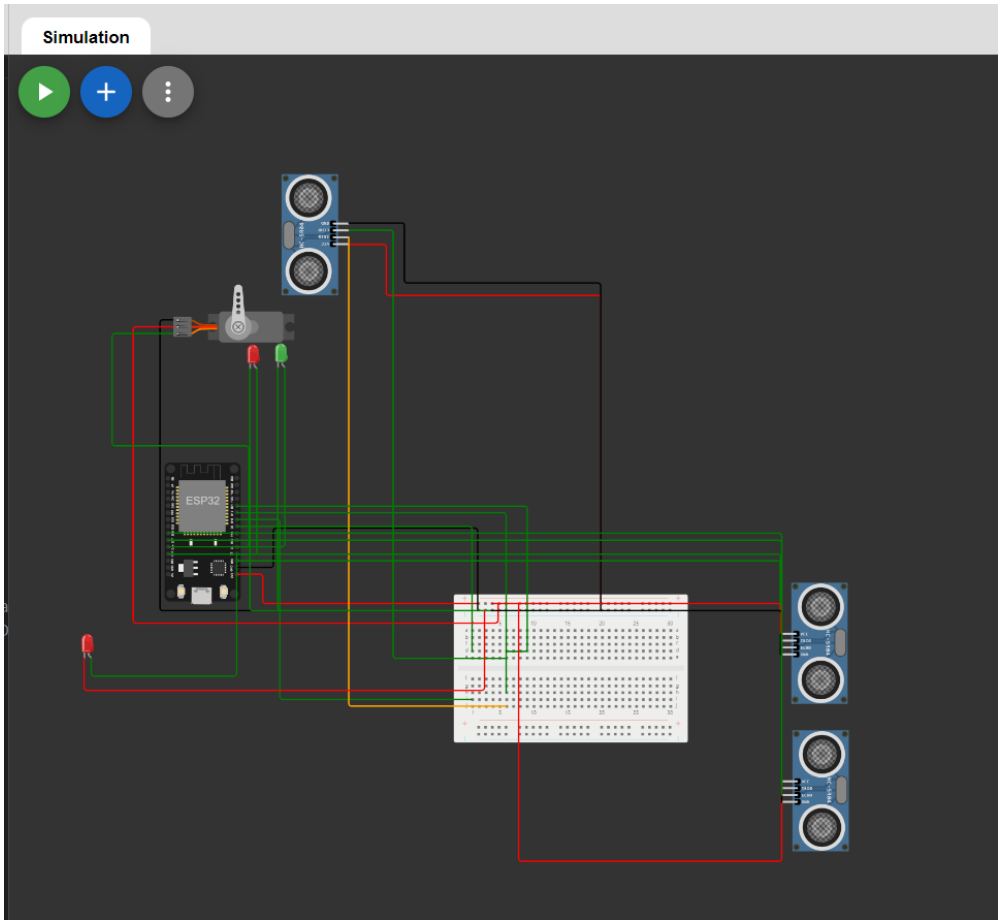


Fig.2 wowki circuit

As shown in Fig.1 a node red connection has been created. Then a circuit is developed in wowki using the required components for the smart parking system as shown in Fig.2. This is the crucial stage as it gives the base to the system.

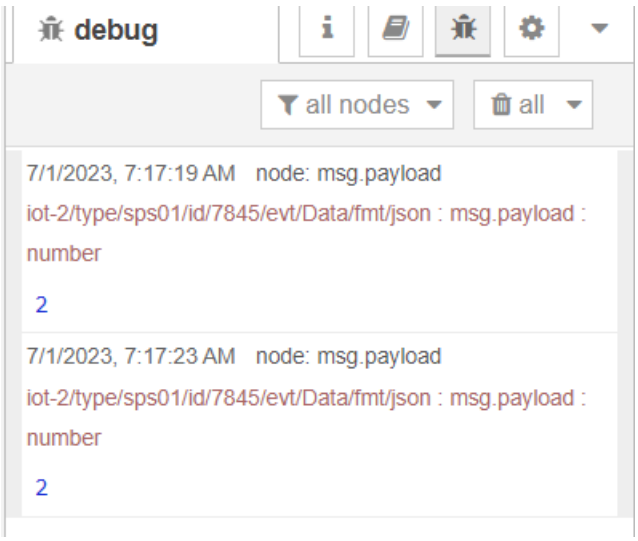


Fig.3 Node red receiving wowki data.

The wokwi page will send the data regarding the available parking slot to the node red as displayed in Fig.3. The device is being connected to IBM cloud as shown in Fig.4

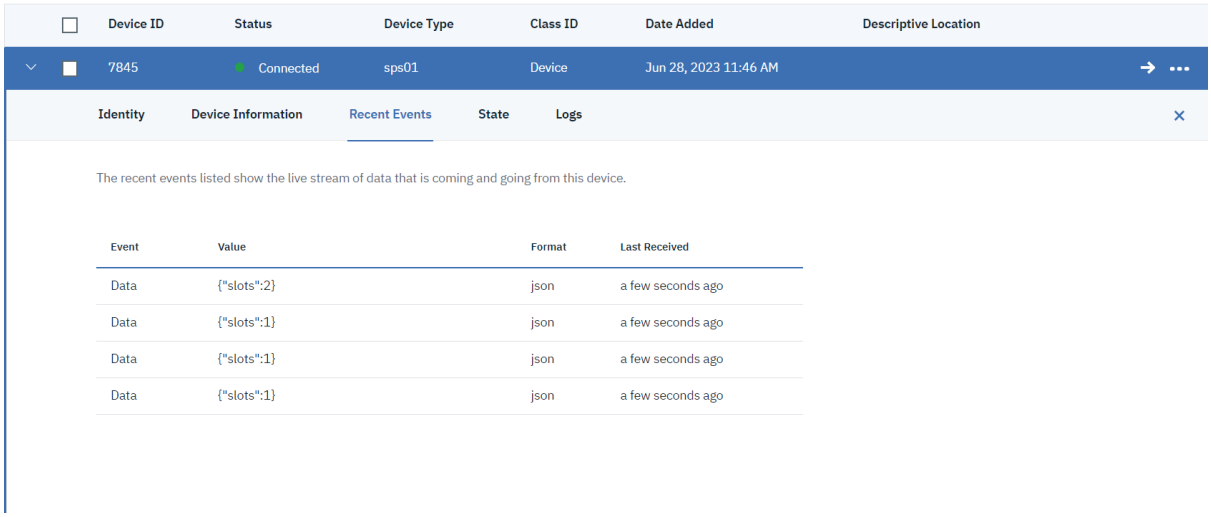


Fig.4 IBM cloud device

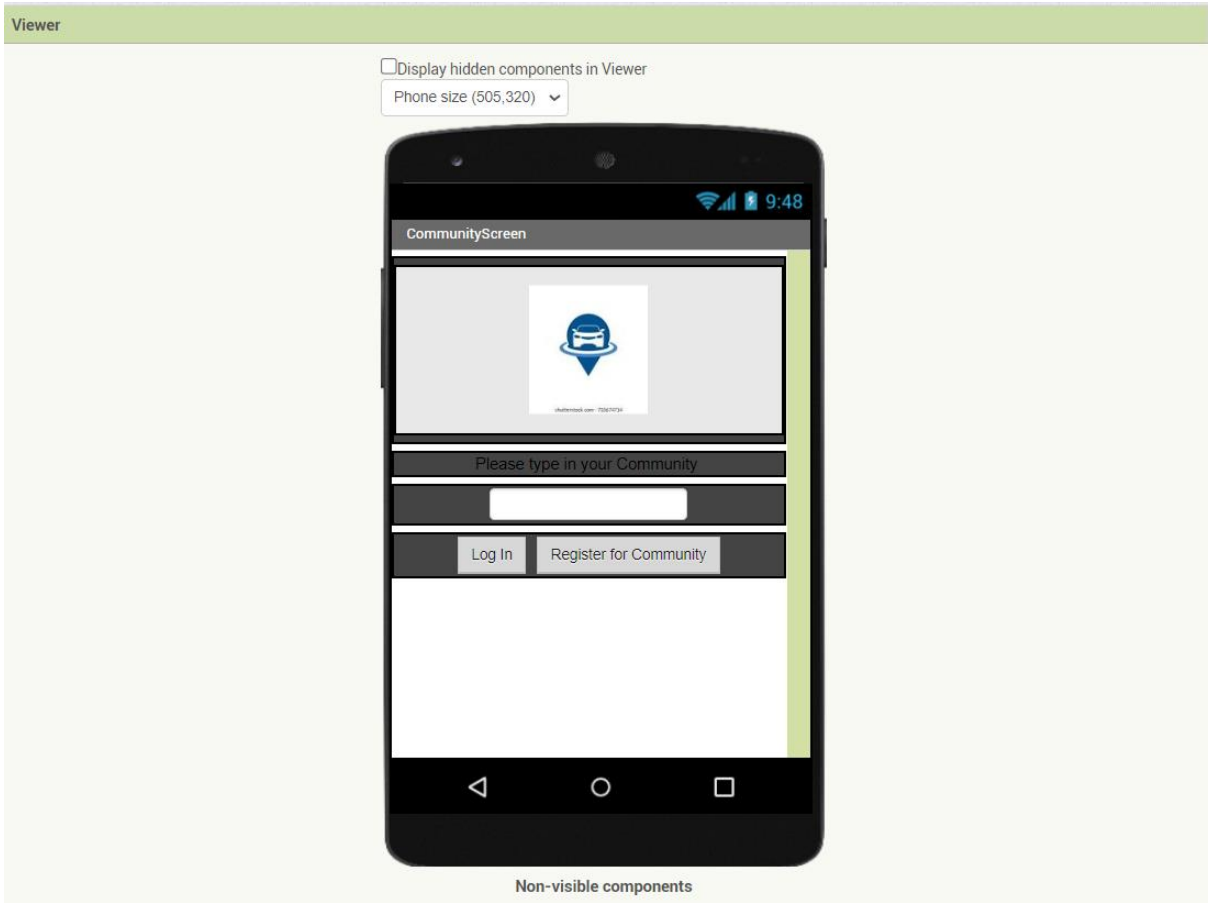


Fig.5 MIT APP Login

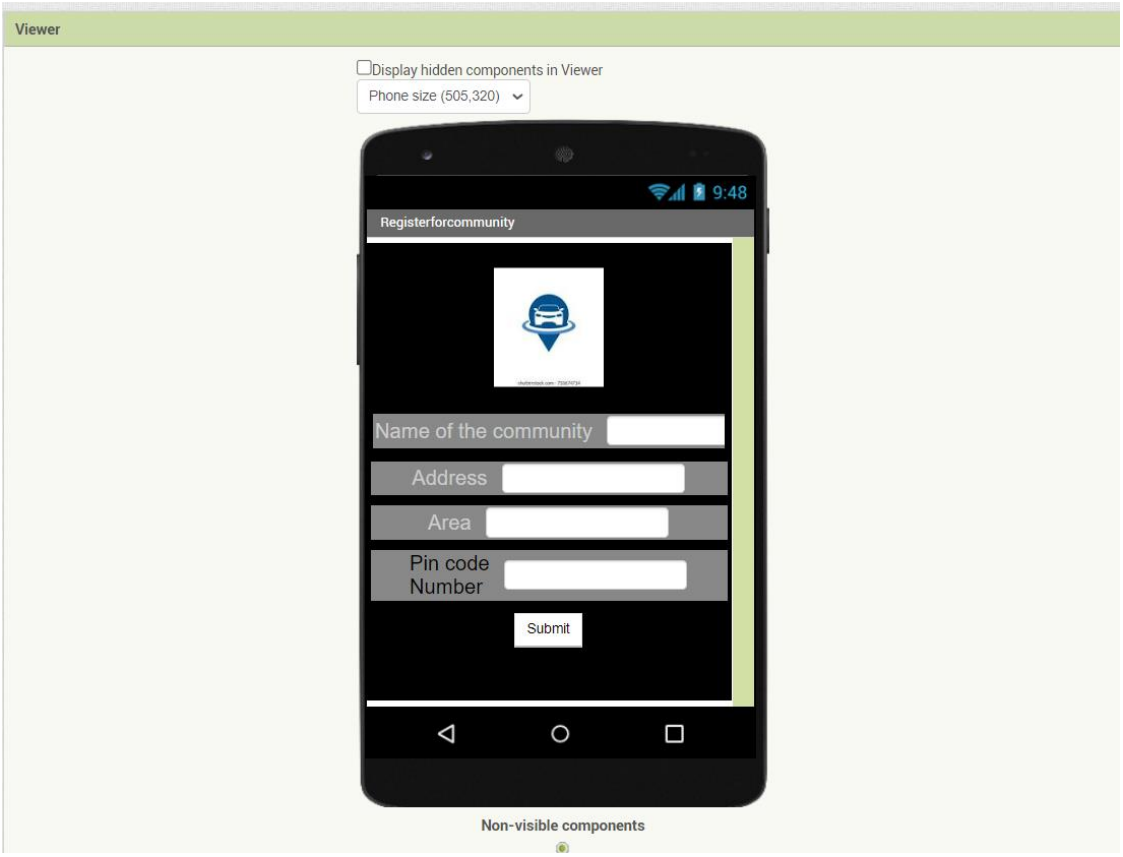


Fig.6 Locating Free parking slots in MIT APP

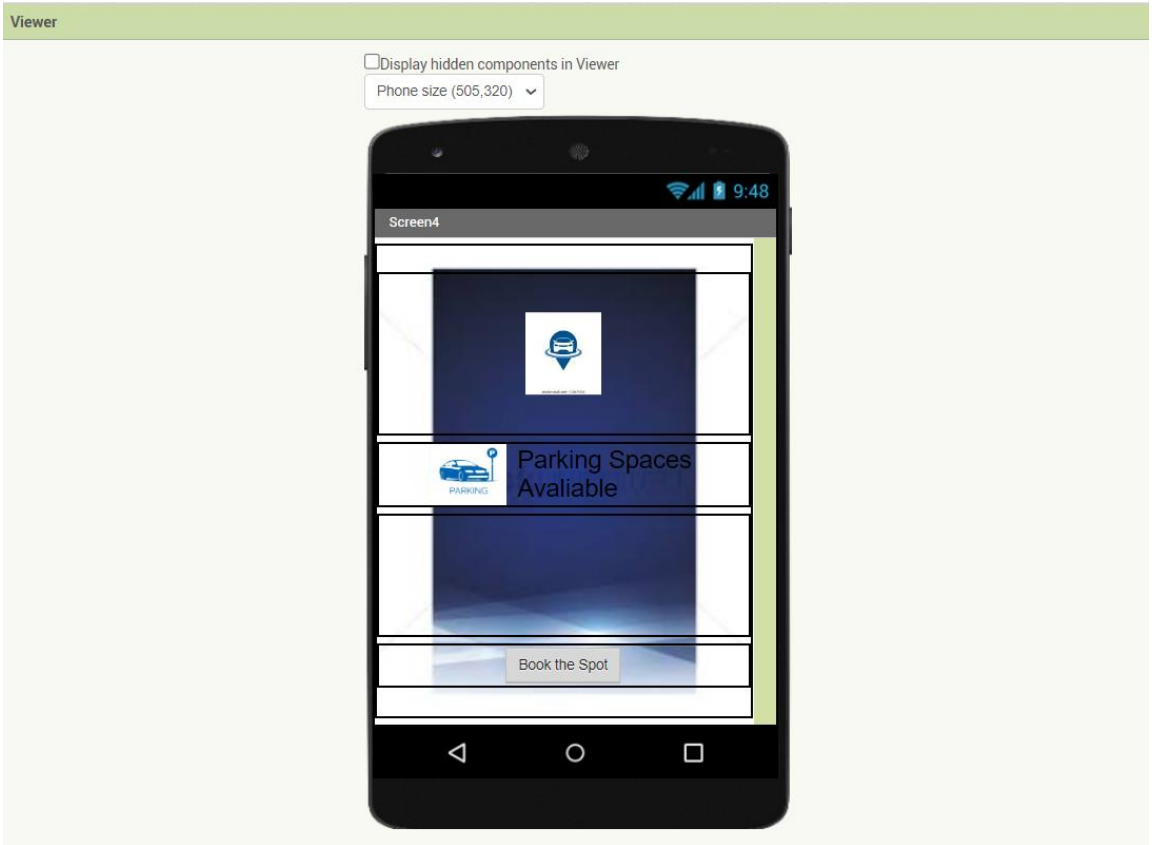


Fig.7 MIT APP showing free parking slots.

As shown in Fig.5,6 & 7 The user can access the MIT APP inventor in their mobile phones to locate free parking slots available in the specified destination. By checking the app, the user can decide whether to go to the place for parking or somewhere else.

7.ADVANTAGES & DISADVANTAGES

7.1 Advantages

1. Superior Technology
2. Better parking experience
3. Increased Protection
4. Reduced traffic and pollution
5. Easy implementation and management
6. Cost-effective
7. Uses integrated software and applications

7.2 Disadvantages

1. Expensive Construction & Installation
2. Requires Regular Maintenance
3. System Breakdown
4.Uncertainty in the building structure
5.Operation

8.APPLICATIONS

Advanced and intelligent parking solutions not only reduce the traffic on the roads but also prevent drivers from getting stuck in the areas due to the non-availability of parking.

9.CONCLUSION

Automation is a positive step towards a successful future in the transportation industry. The common issue raised by this design has a practical remedy provided by this design. When the threshold distance was calibrated and the obstruction was recognised, the smart auto parking system that was created, made, and tested produced accurate results. Based on the presence or absence of a car in the parking area, the LEDs instantly switched between the two states.

The design is adaptable and may be changed depending on the available space. It can even be installed in a small, cramped area. A popular information board that displays the number of parking spaces available is displayed based on the number of Green LEDs that are found.

It may be inferred that an automatic smart automobile parking system can be made, which would save pointless driving, wasteful use of fuel, and time, as well as make parking a lot easier, with the proper connection of a few basic electrical components.

10. FUTURE SCOPE

Utilising an Android application, the Smart Parking system based on Slot Booking is deployed. The slot allocation approach allows us to reserve our own, cheapest parking space. It is effective at resolving parking issues and provides automatic billing in addition to reducing traffic congestion. This project might be expanded into a fully automated system employing a layered parking technique. It is also possible to design safety features like tracking the vehicle's number, driver face recognition to prevent theft, and automatic payment. The "Smart Parking" technology will be available to consumers in their mobile devices as we increase the testing in a real-world setting.

11. BIBILOGRAPHY

[1]Robin Grodi, Danda B. Rawat, Fernando Rios-Gutierrez: Smart parking-parking occupancy monitoring and visualization system for smart cities.

[2] Abhirup Khanna, Rishi Anand: IoT based smart parking system.2016 International conference on Internet of Things and application(IOTA).

[3] DharminiKanteti, D V S Srikar and T K Ramesh: smart parking system for commercial stretch in cities.

[4] Georgios Tsaramirsis, IoannisKaramitsos, CharalamposApostolotpoulos: smart parking-an IoT application for smart cities.

[5] Rosario Salpietro, Luca Bedogni, Marco Di Felice, LucianoBononi: Park here! A smart parking based on smart phones' embedded sensors and short range communication technologies.

[6] Thanh Nam Pham1, Ming-Fong Tsai1, Duc Binh Nguyen1, Chyi-Ren Dow1, And Der-Jiunn Deng2 "A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies".

[7] El Mouatezbillah Karbab, Djamel Djenouri, Sahar Boulkaboul, Antoine Bagula, CERIST Research Center, Algiers, Algeria University of the Western Cape, Cape town, South Africa, Car Park Management with Networked Wireless Sensors and Active Research Publications.

[8] M. M. Rashid, A. Musa, M. Ataur Rahman, and N. Farahana, A. Farhana, "Automatic Parking Management System and Parking Fee Collection Based on Number Plate Recognition.", International Journal of Machine Learning and Computing.

[9] Hilal Al-Kharusi, Ibrahim Al-Bahadly, “Intelligent Parking Management System Based on Image Processing”, World Journal of Engineering and Technology.

[10] X. Zhao, K. Zhao, and F. Hai, ``An algorithm of parking planning for smart parking system,'' in Proc. 11th World Congr. Intell. Control Autom. (WCICA).

[11] L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and R. Vergallo,``Integration of RFID and WSN technologies in a smart parking system,''in Proc. 22nd Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM).

APPENDIX

```
1  //SMARTINTERNZ PROJECT
//SMART PARKING SYSTEM
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt

//ESP32 SERVO LIBRARY
#include <ESP32Servo.h>

#define LED 2

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "3v6pvz"//IBM ORGANITION ID
#define DEVICE_TYPE "sps01"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "7845"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;

//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id

//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential
```

```
Servo s;

const int totalSlots = 2;

//VARIABLES
int lr = 12;
int lg = 14;

int tg = 19;
int eg = 21;

int availableSlots = totalSlots;

void setup() {
  Serial.begin(115200);
  s.attach(4);

  pinMode(lr,OUTPUT);
  pinMode(lg,OUTPUT);
  pinMode(tg,OUTPUT);
  pinMode(eg, INPUT);
  pinMode(15,OUTPUT);
  pinMode(2, INPUT);
  pinMode(27, OUTPUT);
  pinMode(26, INPUT);

  wificonnect();
  mqttconnect();
}

void loop() {

  digitalWrite(tg,LOW);
  digitalWrite(tg,HIGH);
  delayMicroseconds(10);
  digitalWrite(tg,LOW);


  float dur = pulseIn(eg,HIGH);
  float dis = (dur*0.0343)/2;

  if(dis < 200 ){
    s.write(90);
    digitalWrite(lr,LOW);
    digitalWrite(lg,HIGH);
    delay(5000);
```

```
}

else{
  s.write(0);
  digitalWrite(lg,LOW);
  digitalWrite(lr,HIGH);

}


digitalWrite(15,LOW);
digitalWrite(15,HIGH);
delayMicroseconds(10);
digitalWrite(15,LOW);

float dur1 = pulseIn(2,HIGH);
float dis1 = (dur1*0.0343)/2;

int slot1, slot2;

int totalSlots;

if(dis1 < 200){
  slot1 = 0;
}

else{
  slot1 = 1;
}

digitalWrite(27,LOW);
digitalWrite(27,HIGH);
delayMicroseconds(10);
digitalWrite(27,LOW);

float dur2 = pulseIn(26,HIGH);
float dis2 = (dur2*0.0343)/2;

if(dis2 < 200){
  slot2 = 0;
}

else{
  slot2 = 1;
}

totalSlots = slot1 + slot2;
```

```
  PublishData(totalSlots);
  delay(4000);
  if (!client.loop()) {
    mqttconnect();
  }


}

/*.....retrieving to Cloud.....*/

void PublishData(int slots) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"slots\":";
  payload += slots;
  payload += "}";


  Serial.print("Sending payload: ");
  Serial.println(payload);


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print
publish ok in Serial monitor or else it will print publish failed
  } else {
    Serial.println("Publish failed");
  }

}

void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
```

```
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="lighton")
  {
Serial.println(data3);
digitalWrite(LED,HIGH);
  }
  else if(data3=="{\"command\":\"lighton\"}")
  {
Serial.println(data3);
digitalWrite(LED,HIGH);
  }
  else
```

```
 {
Serial.println(data3);
digitalWrite(LED,LOW);
 }
data3="";
}
```