Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

IoT Mini Project Report

on

Smart Parking Systems

Submitted by

1. Pallavi Udatewar (PH 08)
2. Rishabh Shinde (PH 09)
3. Ankita Mutta (PH 11)
4. Sumeet Pandhare (PH 16)

T7, Panel No. 8

Under the Guidance of

Mrs. Jyoti Gavhane

At

School of Computer Engineering and Technology

MIT World Peace University, Kothrud, Pune 411 038, India

2020-21, Trimetser-I

MIT-WPU

॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

SCHOOL OF COMPUTER ENGINEERING AND TECHNOLOGY

CERTIFICATE

This is to certify that

Pallavi Udatewar (PH 08

Rishabh Shinde (PH 09)

Ankita Mutta (PH 11)

Sumeet Pandhare (PH 16)

of T. Y. B. Tech. CSE have successfully completed Mini Project on

“Smart Parking Systems”

to my satisfaction and submitted the same during academic year 2020-21, Trimester VII as part of Embedded and Internet of Things Laboratory subject.

Mrs. Jyoti Gavhane
(Mini Project Guide)

Dr. M. V. Bedekar
(Program Head)

Place: School of Computer Engineering and Technology, MIT-WPU, Pune

Date: **21/09/2020**

Table of Contents

Abstract

In today's technological world the concept of smart city has become an area of interest. Concern to parking became impending in an urban area. The parking space problem can be turn into a new opportunity brought by the recent trends to meet the world's connected continuum. In this paper, I am presenting the IoT based Smart Car Parking System. This paper makes easy for the user to find automatically a free space at the low cost and without consuming time and fuel. The whole system is based on Wi-fi network. The android application in mobile is also provided to the user to check the availability of free space for parking and book that slot accordingly.

Introduction

The concept of Internet of thing(IoT) started with the identity communication devices. The devices could be tracked, controlled or monitor using remote computers connected through Internet. The two words in IoT are 'Internet' and 'Things' in which Internet means the vast world's network of connected servers, computers, tablets, mobiles using an internationally used protocols and connecting the systems enabling sending, receiving and communication of information. It provides a vision where things become smart and behave like alive through sensing computing and communication by embedding small devices which has an interaction with the remote objects and persons through connectivity. Physical object + Controller, sensor and Actuators + Internet = Internet Of Things The idea of creating smart city is now possible with an emergence of IoT. Smart cities related to car parking facility and traffic control management has become the major issue. In big cities chasing for an available parking slot is always is not easy for drivers and it tends to become difficult with increasing number of user having their own cars. This situation could be taken as an opportunity in order to make advancement in the efficiency of parking resources which will reduce the searching time, traffic congestion and road accidents. The drivers can be informed in advance about the availability of to their destination. Followed by the development in sensor technology, many modern cities have adopted for deploying different IoT based system around the city for the monitoring. Parking is an expensive process in terms of money, time and efforts made for searching free spot. A recent survey has reflected an increase in number of innovative ideas related to parking system. The smart Parking System proposed by me is worked out using mobile application connected to Wi-Fi. The Wi-Fi network has various advantages like flexibility, inherent intelligence, low cost, rapid deployment and more sensing point, especially in an area where wired communication is not possible. Due to these benefits, largely used in different application such as health monitoring, facility management, environment monitoring, intelligent buildings, disaster relief applications etc. In this work I have developed a Smart Parking System that can monitor an available empty slots economical and reliable and considerably contributing to fuel and time consuming. In this work the most widely used sensors are Infra-Red (IR) sensors. These sensors are used for monitoring the vacancy of parking slot. The reason for using these sensors is they are cheap and use less memory as compared to camera if used as sensors. The sensor will monitor that whether the car is parked or the slot is empty. If the slot is empty, then the sensor would sense it and give the notice to controller and controller will activate the LEDs as per the notification. If the car is parked in any slot then the LED would glow as RED showing that slot is not empty, otherwise when the slot would be empty or available it will glow GREEN. The mobile application used by the user acts like the interface for the end users to interact with the whole system. The Android Application is simply design to see and choose the slot for parking before arrival to the destination. It would be for the user to decide which slot is convenient for exit during departure. Doing this will save time, as well as the fuel and also it would not arise the traffic congestion problem.

Related Work

E-parking: E-parking provides an alternative for patrons to enquire the availability and/or reserve a parking space at their desired parking facility to ensure the availability of vacant car park space when they arrive at the parking facility. The system can be accessed via numerous methods such as SMS or through the internet. Some of the additional benefits of using the E-parking system aside from those collectively gained by smart parking system are that it can be extended easily to incorporate the payment mechanism of smart payment system whereby payments by the patrons are made hassle free using the technologies discussed previously. Customized information can also be provided to the patrons either before or during their trip to the car park.

In a study by Inaba *et al*. (2001), reservations can be made through the utilization of mobile phones or any reservation centers convenient to the patrons. On the other hand, the study by Hodel and Cong (2004) revealed options of using the internet via Wireless Application Protocol (WAP) enabled mobile phones, Personal Digital Assistants (PDAs) and even conventional computer in addition to SMS service for the drivers in accessing the information as well as making reservations. Teodorovic *et al*. (2006) takes the implementation a step further by incorporating fuzzy logic in decision making whereby the parking reservation request can either accepted or rejected. It also facilitates the enforcement of tariff classes to enable the maximization of revenue for car park operators. The system discussed is one of the systems integrating PGIS with E-parking system, where the patrons are able to reserve parking slots after reviewing the status of the car park and its proximity to the patron's current location

Proposed Work

Problems Statement

With increase in the population the number vehicles have increased exponentially as well. This leads to a problem with identifying parking space and due to improper management of parking's it leads to many problem

In a fast-moving world time is an essential element. With increasing vehicles people have to wait in long queues to find parking space. This leads to people getting agitated and frustrated. The lack of management can cause problems to the people looking for parking's as well as those in charge of managing parking areas.

Social Relevance

Smart parking systems are beneficial to society. They help both the drivers as well as the parking management. Smart parking systems help by:

- **Less fuel is wasted**
  Drivers are directed straight to an available parking spot. Therefore they waste fewer kilometers driving around in circles looking for vacant parking space.

- **Save money**
  Obviously, by driving more efficient when in search of parking space, you will save on fuel waste. Resulting in spending less money on petrol.

- **Save time**
  Additionally, by driving fewer kilometers when in search of parking space, you will save valuable time which can be spent on work, fun or hobbies. Resulting in spending less money on petrol.

- **Lowering individual environmental footprint**
  Another benefit of wasting fewer kilometers by searching for a parking spot is that you reduce individual pollution. Fossil fuels, petrol, diesel, and most alternative fuels all produce emissions, especially carbon dioxide (Co2). This pollution will not directly harm human life. However, Co2 is the most significant contributor to greenhouse gases and therefor contributor to climate change.

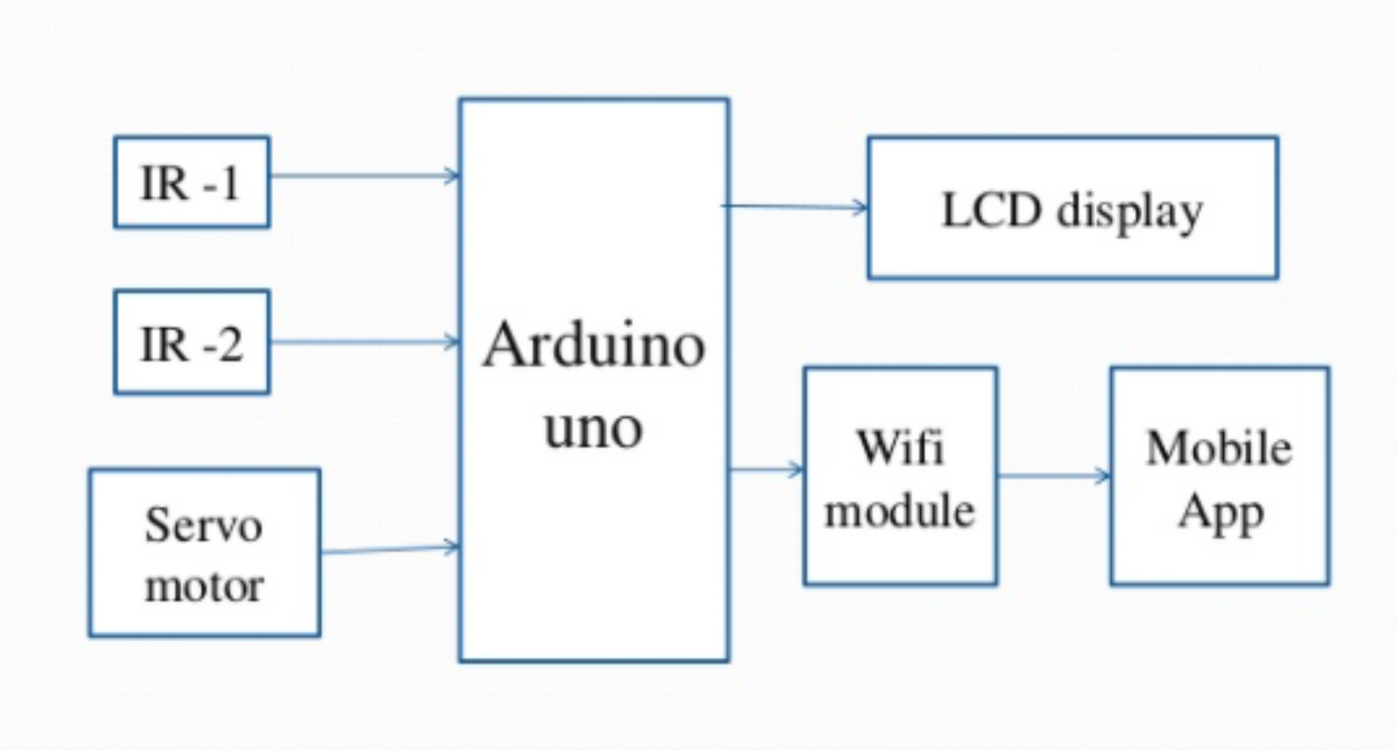
- **Increase in safety.**
  Drivers are less distracted looking around for a spot because they know where they can park their car. They will have their full attention on the road. By having their eyes on the road, accidents will decrease and safety will increase for themselves, other drivers and pedestrians.

- **Smart parking reducing stress while searching for a parking space**
  Driving through the same street over and over again, cars breathing down your neck and no parking spot to be seen. Having uncertainty and pressure to find a parking spot near your destination can be very stressful. With the use of smart parking, you know where the available parking space is located. You can drive straight to an open parking spot, stress-free.

- **Smart Parking takes away the unpredictability of finding a parking spot**
  Not visiting a particular (part of a) city because you do not want the hassle of finding an available parking spot. Knowing you are going to drive around for many minutes and probably are going to find a place far, far away from your destination can be very discouraging. Smart parking will allow you to see where you can park your car, and at what time it is the busiest.

- **Smart parking will reduce search traffic on the streets.**
  Smart parking will make sure there are fewer cars on the streets that drive slowly, circling for ages, looking around for a spot. This will benefit traffic flow and will reduce congestions in neighborhoods with an under capacity in parking space. Therefore there are fewer traffic jams, and drivers will benefit by having less traffic on the streets.

Architecture / Model / Block Diagram



Smart Parking Sensor Block Diagram/ Model

Hardware and Software Requirements:

Hardware Requirements

- **Tinker CAD**

Tinkercad is a free, online 3D modeling program that runs in a web browser, known for its simplicity and ease of use. Since it became available in 2011 it has

become a popular platform for creating models for 3D printing as well as an entry-level introduction to constructive solid geometry in schools.

- **ThingSpeak**

According to its developers, "ThingSpeak is an open – source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. ThingSpeak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates".

- **Ultrasonic Sensor**

Ultrasonic Sensor detects the distance to objects by emitting high-frequency ultrasonic waves. Here we use HC-SR04 ultrasonic sensor.

HC-SR04

HC-SR04 is an ultrasonic sensor mainly used to determine the distance of the target object. It measures accurate distance using a non-contact technology – A technology that involves no physical contact between sensor and object. Transmitter and receiver are two main parts of the sensor where former converts an electrical signal to ultrasonic waves while later converts that ultrasonic signal back to electrical signals.

HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40 Hz

- **ESP8266**

The ESP8266 is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes-style commands. However, at first there was almost no English-language documentation on the chip and the commands it accepted.

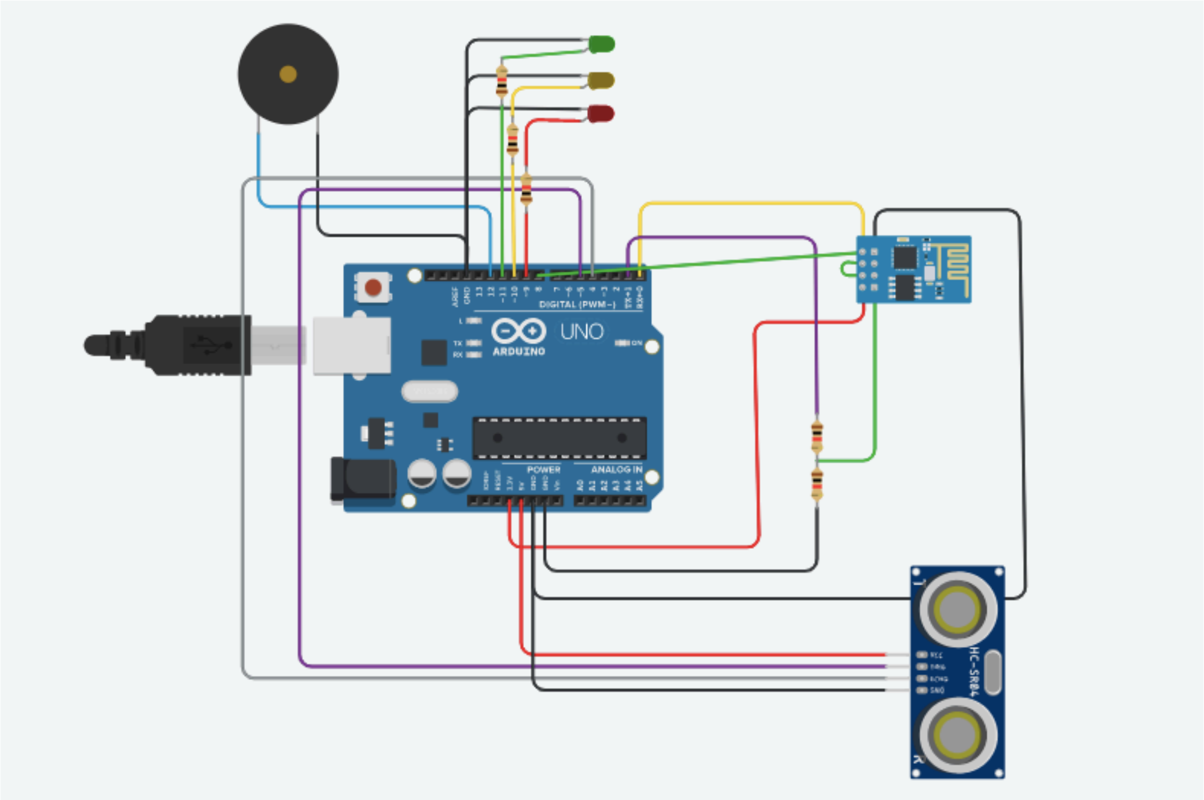
Features

- 16 GPIO pins
- 10-bit ADC
- 32 KiB instruction RAM

Software Requirements:

- Operating System: Windows/ Linux/ Mac OS
- Python 3.7
- Anaconda Navigator

Circuit Diagram



Code for circuit:

```
String ssid     = "Simulator Wifi";  // SSID to connect to
String password = ""; //virtual wifi has no password
String host     = "api.thingspeak.com"; // Open Weather Map API
const int httpPort   = 80;
String url     = "/update?api_key=9DGNFLGB34UGATOQ&field1=";
int len = 0;
//Replace XXXXXXXXXXXXXXXX by your ThingSpeak Channel API Key


void setupESP8266(void) {
  // Start our ESP8266 Serial Communication
  Serial.begin(115200);   // Serial connection over USB to computer
  Serial.println("AT");   // Serial connection on Tx / Rx port to ESP8266
  delay(10);        // Wait a little for the ESP to respond
  if (Serial.find("OK"))
    Serial.println("ESP8266 OK!!!");

  // Connect to Simulator Wifi
  Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
  delay(10);        // Wait a little for the ESP to respond
  if (Serial.find("OK"))
    Serial.println("Connected to WiFi!!!");

  // Open TCP connection to the host:
  //ESP8266 connects to the server as a TCP client.

  Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort);
  delay(50);        // Wait a little for the ESP to respond
  if (Serial.find("OK"))
   Serial.println("ESP8266 Connected to server!!!") ;

}

long readUltrasonicDistance(int triggerPin, int echoPin)

{
pinMode(triggerPin, OUTPUT);
digitalWrite(triggerPin, LOW);
delayMicroseconds(2);
digitalWrite(triggerPin, HIGH);
delayMicroseconds(10);
digitalWrite(triggerPin, LOW);
pinMode(echoPin, INPUT);
return pulseIn(echoPin, HIGH);
```

```
}

/*void anydata(void) {

}*/


void setup() {
  Serial.begin(9600);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
  setupESP8266();

}

void loop() {

 //anydata();
   Serial.println(0.01723 * readUltrasonicDistance(5, 4));
  if (0.01723 * readUltrasonicDistance(5, 4) < 125)
  {
  digitalWrite(9, HIGH);
  digitalWrite(10, LOW);
  digitalWrite(11, LOW);
  noTone(12);
  delayMicroseconds(10);
  len = 1;
  Serial.println(len);
  }
  else
  {
    if (0.01723 * readUltrasonicDistance(5, 4) < 200)
    {
    digitalWrite(9, LOW);
    digitalWrite(10, HIGH);
    digitalWrite(11, LOW);
    tone(12, 523, 500);
    delayMicroseconds(10);
    }
    else
    {
    noTone(12);
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, HIGH);
    len = 0;
    }
  }

  // Construct our HTTP call
  String httpPacket = "GET " + url + String(len) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
  int length = httpPacket.length();
  Serial.println(len);

  // Send our message length
  Serial.print("AT+CIPSEND=");
  Serial.println(length);
  delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;

  // Send our http request
  Serial.print(httpPacket);
  delay(10); // Wait a little for the ESP to respond
  if (Serial.find("SEND OK\r\n"))
    Serial.println("ESP8266 sends data to the server");

  delay(1);
```

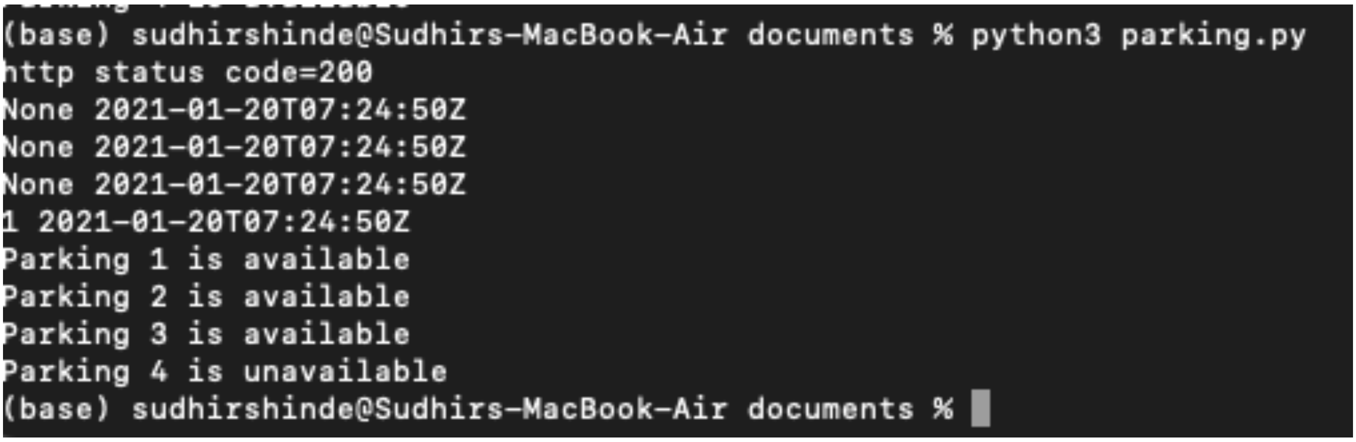
```
}
```

Web Application Code:

```
import urllib,json
import urllib.request
import time
READ_API_KEY = 'X03D257CKZMWZIS0'
CHANNEL_ID = 1151045
def main():
    conn = urllib.request.urlopen("http://api.thingspeak.com/channels/%s/feeds/last.json?api_key=%s" \
                       % (CHANNEL_ID,READ_API_KEY))

    response = conn.read()
    print("http status code=%s" % (conn.getcode()))
    data =json.loads(response)
    print(data['field1'],data['created_at'])
    print(data['field2'],data['created_at'])
    print(data['field3'],data['created_at'])
    print(data['field4'],data['created_at'])

    if(data['field1'] == '1'):
        print("Parking 1 is unavailable")
    else:
        print("Parking 1 is available")

    if(data['field2'] == '1'):
        print("Parking 2 is unavailable")
    else:
        print("Parking 2 is available")

    if(data['field3'] == '1'):
        print("Parking 3 is unavailable")
    else:
        print("Parking 3 is available")
    if(data['field4'] == '1'):
        print("Parking 4 is unavailable")
    else:
        print("Parking 4 is available")
    conn.close()


if _name_ == '_main_':
    main()
```

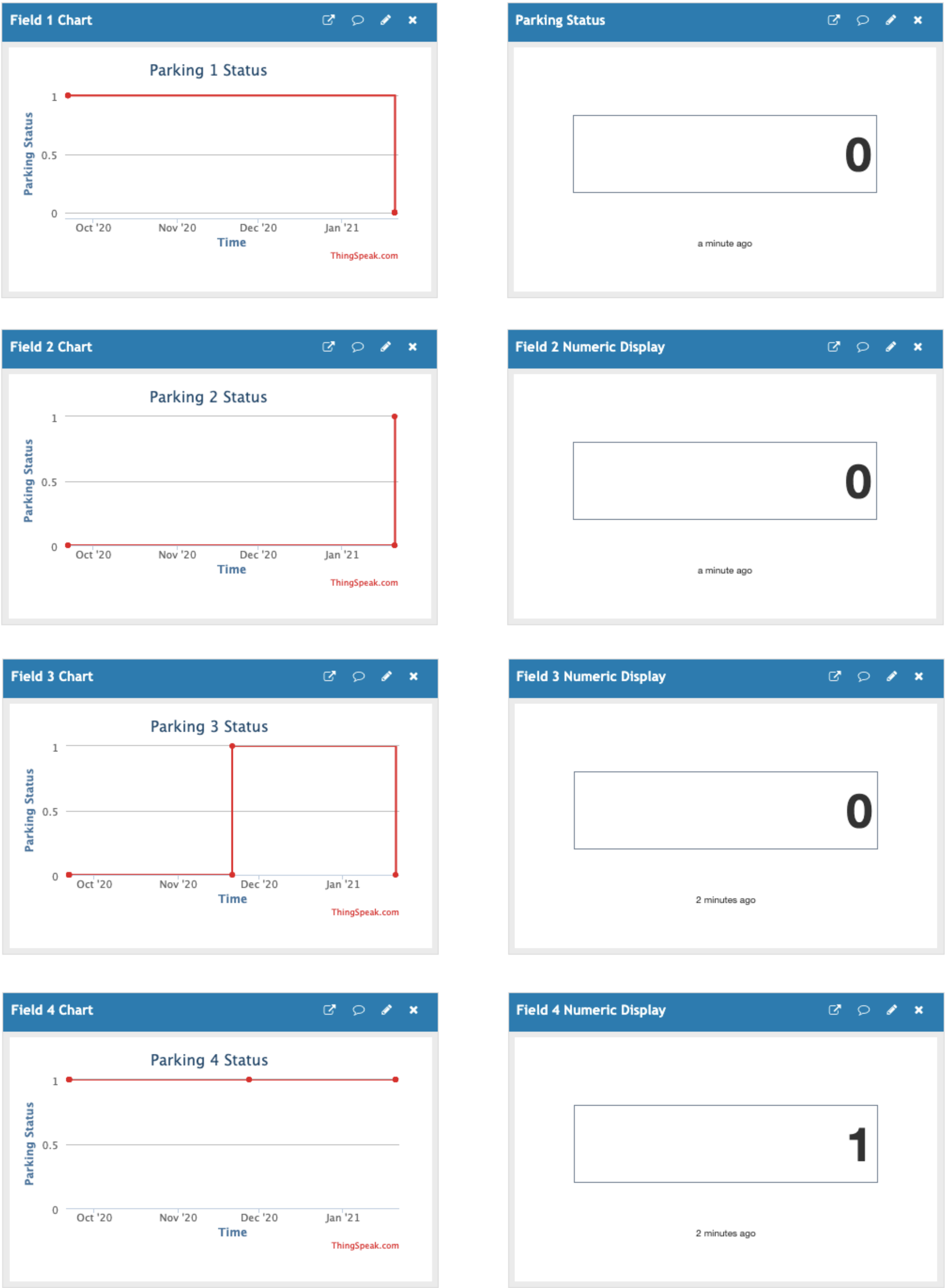
Results Obtained

Web Application:



```
(base) sudhirshinde@Sudhirs-MacBook-Air documents % python3 parking.py
http status code=200
None 2021-01-20T07:24:50Z
None 2021-01-20T07:24:50Z
None 2021-01-20T07:24:50Z
1 2021-01-20T07:24:50Z
Parking 1 is available
Parking 2 is available
Parking 3 is available
Parking 4 is unavailable
(base) sudhirshinde@Sudhirs-MacBook-Air documents %
```

Thingspeak:



Thingspeak provides status of parkings.

Conclusion

implementation of the smart parking system being presented, its efficiency in alleviating the traffic problem that arises especially in the city area where traffic congestion and the insufficient parking spaces are undeniable. It does so by directing patrons and optimizing the use of parking spaces.

With the study on all the sensor technologies used in detecting vehicles, which are one of the most crucial parts of the smart parking system, the pros and cons of each sensor technologies can be analyzed.

References

https://scialert.net/fulltext/?doi=itj.2009.101.113

https://ieeexplore.ieee.org/document/6921729