SMART PARKING SYSTEM
BASED ON
INTERNET OF THINGS

A PROJECT REPORT

submitted to

SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING

In partial fulfilment of requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

by

ANUMULA SHRAVANI (11506002)

GUDIPATI SAI KRISHNA (11506010)

KATTUBADI SHARUKH BASHA (11506017)

Under the guidance of

Dr. A. RAMA MOHAN REDDY
Professor Department of Computer Science & Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING, TIRUPATI,
2018-19.

SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING, TIRUPATI

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**SMART PARKING SYSTEM BASED ON INTERNET OF THINGS"** genuine and has been carried out under my supervision in the **Department of Computer Science and Engineering**, **Sri Venkateswara University College of Engineering.**

The work is comprehensive, complete and fit for evaluation carried out in partial fulfilment of the requirements for the award of **Bachelor of Technology** in **Computer Science and Engineering** during the academic year 2018-19.

To the best of our knowledge matter embodied in the project has not been submitted to any other University/Institution for the award of any Degree or Diploma.

GUIDE:	**HEAD OF THE DEPT.:**
Dr. A. RAMA MOHAN REDDY Professor, Department of CSE, SVUCE, Tirupati.	Dr. M. HUMERA KHANAM Professor, Department of CSE, SVUCE, Tirupat

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING, TIRUPATI

Declaration

The project entitled "**SMART PARKING SYSTEM BASED ON INTERNET OF THINGS"** is a bona fide work performed by us, for the partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** from **Sri Venkateswara University, Tirupati.**

To the best of our knowledge matter embodied in the project has not been submitted to any other University/Institution for the award of any Degree or Diploma.

(ANUMULA SHRAVANI	**11506002)**
(GUDIPATI SAI KRISHNA	**11506010)**
(KATTUBADI SHARUKH BASHA	**11506017)**

ABSTRACT

People owning vehicles face parking problems in most metropolitan area, especially during peak hours. The difficulty roots from not knowing where the parking spaces are available at the given time, even if this is known; many vehicles may pursue a small number of parking spaces which in turn leads to serious traffic congestion.

In this project, we present an **IoT** based cloud integrate **SMART PARKING SYSTEM**. The proposed Smart Parking system consists of an on-site deployment of an IoT module that is used to monitor and signalize the state of availability of each single parking space. A mobile application is also provided that allows an end user to check the availability of parking space and book a parking slot accordingly. The paper also describes a high-level view of the system architecture. Our proposed system is within low cost and less time to build it.

Towards the end, the paper discusses the working of the system in form of a use case that proves the correctness of the proposed model.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.1 IoT introduction:

The concept of Internet of Things(IoT) started with things with identity communication devices. The devices could be tracked, controlled or monitored using remote computers connected through Internet. IoT extends the use of Internet providing the communication, and thus inter-network of the devices and physical objects, or ‘Things’.

The two prominent words in IoT are “internet” and “things”. Internet means a vast global network of connected servers, computers, tablets and mobiles using the internationally used protocols and connecting systems. Internet enables sending, receiving or communicating of information. Thing in English has number of uses and meanings. Dictionary meaning of ‘Thing’ is a term used to reference to a physical object, an action or idea, situation or activity, in case when we do not wish to be precise.

IoT, in general consists of inter-network of the devices and physical objects, number of objects can gather the data at remote locations and communicate to units managing, acquiring, organizing and analyzing the data in the processes and services. It provides a vision where things (wearable, watch, alarm clock, home devices, surrounding objects with) become smart and behave alive through sensing, computing and communicating by embedded small devices which interact with remote objects or persons through connectivity.

1.2 Cloud Introduction:

The scalable and robust nature of Cloud computing is allowing developers to create and host their applications on it. Cloud acts as a perfect partner for IoT as it acts as a platform where all the sensor data can be stored and accessed from remote locations. These factors gave rise to the amalgamation of both technologies thus leading to the formation of a new technology called Cloud of Things(CoT). In CoT the things(nodes) could be accessed, monitored and controlled from any remote location through the cloud. In simple terms IoT can be explained in form of anS equation stating:

Physical Object + Controller, Sensor and Actuators + Internet= IoT

1.3 .Purpose:

Drivers searching for parking are estimated to be responsible for about 30% of traffic congestion in cities. Historically, cities, businesses, and property developers have tried to match parking supply to growing demand for parking spaces. It has become clear, though, that simply creating more parking spaces is not sufficient to address the problem of congestion.

1.4-Project Details:

New approaches using smart parking systems look to provide a more balanced view of parking that better manages the relationship between supply and demand.

Smart parking can be defined as the use of advanced technologies for the efficient operation, monitoring, and management of parking within an urban mobility strategy. The global market for smart parking systems reached $93.5 million, with the United States representing 46% market share, and offering a strong growth opportunity for companies offering services in the United States and overseas. A number of technologies provide the basis for smart parking solutions, including vehicle sensors, wireless communications, and data analytics. Smart parking is also made viable by innovation in areas such as smartphone apps for customer services, mobile payments, and in-car navigation systems.

1.5-SCOPE:

- The data can be retrived from the server using WiFi module into another module and then onto LCD display or app.
- In future we will continue to add new features and make the system more user friendly.

1.6-OBJECTIVE:

- Save users time in searching for parking slots
- More no. of cars can be parked in minimum space.

2.1 SMART PARKING SYSTEM:

Smart parking systems are very popular in many countries these systems create a nexus of multiple systems together the system help to create better services optimized the space usage, improve the efficiency of parking system and help the traffic in the city move more freely. Currently there are multiple companies who have expressed interest in creating such system smart parking adds to the better customer experience and also create scenarios where visitors spend less time looking for space to park. The smart parking system creates advantages with its cost efficiency and economical space requirements .it serves as better source of revenue generating up to 250% in rent over the span of 10 years. We surveyed multiple owner of such parking space, the benefit shared were seemed to be most welcoming to the new owner of such parking spaces, in our talks with Bryan Peering from the base town center he shared that the technology, aligned with excellent support and inside will be a very good investment it will add to the customer experience where visitors will spend more time in the shops instead of looking space to park. Smart parking will actually help in accurately predicting in sensing spot occupancy in real time. It will simplify the parking experience and will add value for parking stakeholder like drivers. The system will play the major role in creating better urban environment by reducing the emission of CO2 and other pollutants.

2.2 CONCEPTUAL FRAMEWORK:

The rapid growth in the number of vehicle worldwide is intensifying the problem of the scarcity of parking space. Again according to industrial data 30% traffic congestion occur due to vehicle drivers struggling to fine the parking spaces, these in turn are magnifying the necessity of smart parking system. Today's intelligent parking management system is capable of providing extreme level of convenience to driver as well as simplifying and automating the business operation and administrative function of the parking site owners. The higher growth rate in the registration of new cars worldwide the major boom from regional economies such as Asia pacific will open the window of opportunity for parking management system. The global parking management industries are excepted to grow at a compound annual growth rate of 11.4% from 2014 to 2019. the parking management market is estimated to be at 5025.9 million dollars in 2016. the market is excepted to grow in tandem with the growth in vehicle ownership and parking facility development. Need for smooth traffic flow, business benefits to the parking site operators and decreasing hardware and connectivity cost are the key driver for the parking management industries.

2.3 RESEARCH METHODOLOGY:

Using multiple base points and in-depth research to view benefits and the implementation creating a better and faster life for drivers and citizens seen here. Our research using multiple one on one interview with stakeholders created a bond and gave us multiple time bound instigations for innovation and study of new technology. Incidentally, the requirement for this structure is increasing and our research is focused on the same. Multi-dimensional data created this modular infrastructure based sensor system which will go beyond the basic ground sensor. To create a city which is smart the intelligent parking system is an important break through the potential benefits of such an intelligent parking solutions will be:

- 43% less time spent looking for parking.
- 30% less parking related vehicle kilometers travelled.
- 8% less traffic volume when increase in parking availability.
- CO2 gas reduction in emission

Here we propose a system of SMART PARKING using different sensors, Cloud and Android application.

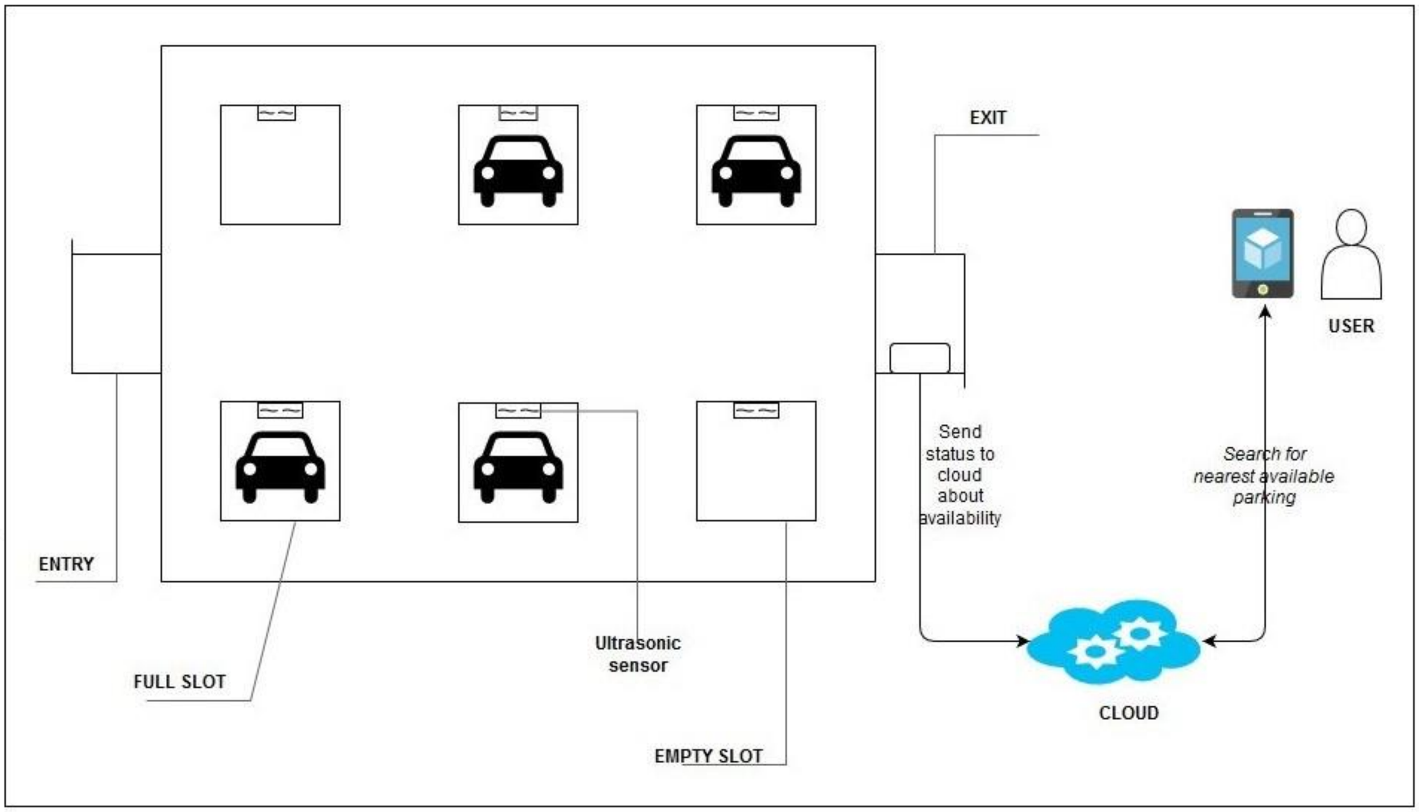

EXIT
USER
Send status to cloud about availability
Search for nearest available parking
ENTRY
FULL SLOT
Ultrasonic sensor
EMPTY SLOT
CLOUD

Fig 3.0.1: Overview of the model

The basic components used in our system are:

Hardware Components:

1. HC-SR04 Ultrasonic
2. Node MCU
3. RTC DS3231

Software Environment:

1. Arduino Integrated development environment (IDE)
2. ThingSpeak Iot platform
3. ESP8266 Library
4. Angular Java script

3.1-HARDWARE COMPONENTS:

3.1.1-HC-SR04 Ultrasonic Sensors:

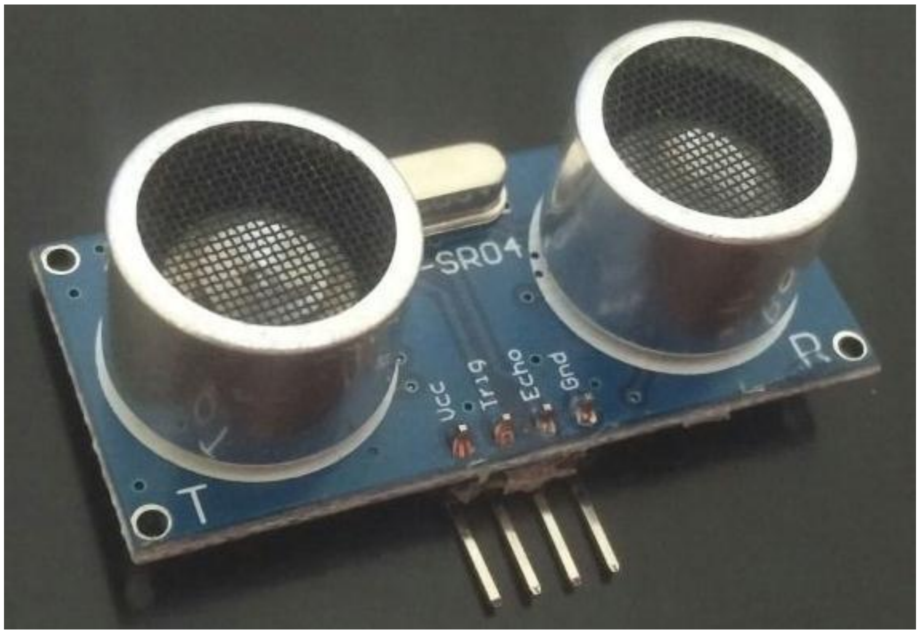
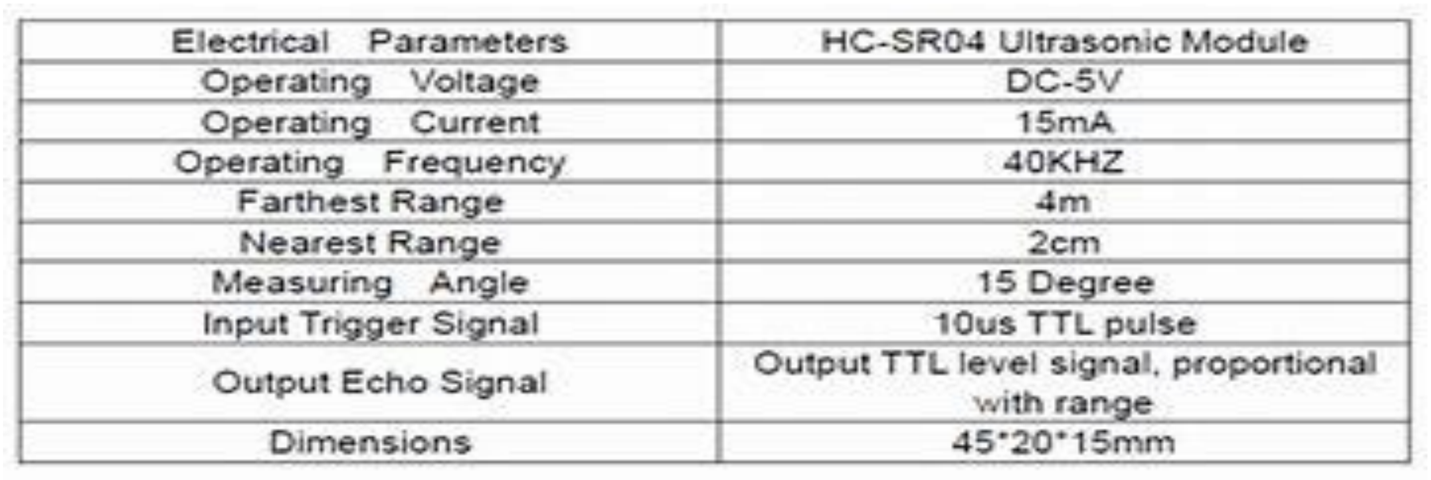


Fig 3.1.1: Ultrasonic sensor

Ultrasonic Range Detection Sensor is used with Arduino in order to calculate distances from objects. It’s an IC (Integrated Circuit) that’s works by sending an ultrasound pulse of about 40KHz. And then it waits and listens for the pulse to echo back, by calculating the time taken in microseconds.

The specifications of HC-SR04 Ultrasonic Sensor are as below :



Electrical Parameters	HC-SR04 Ultrasonic Module
Operating Voltage	DC-5V
Operating Current	15mA
Operating Frequency	40KHZ
Farthest Range	4m
Nearest Range	2cm
Measuring Angle	15 Degree
Input Trigger Signal	10us TTL pulse
Output Echo Signal	Output TTL level signal, proportional with range
Dimensions	45*20*15mm

Table 3.1.1: Electrical Parameters of Ultrasonic sensor

3.1.2-Node MCU:

NodeMCU is a wifi SOC (system on a chip) produced by Espressif Systems. It is based ESP8266 -12E WiFi module. It is an highly integrated chip designed to provide full internet connectivity in a small package. It can be programmed directly through USB port using LUA programming or Arduino IDE. By simple programming we can establish a WiFi connection and define input/output pins according to your needs exactly like arduino, turning into a web server and a lot more.

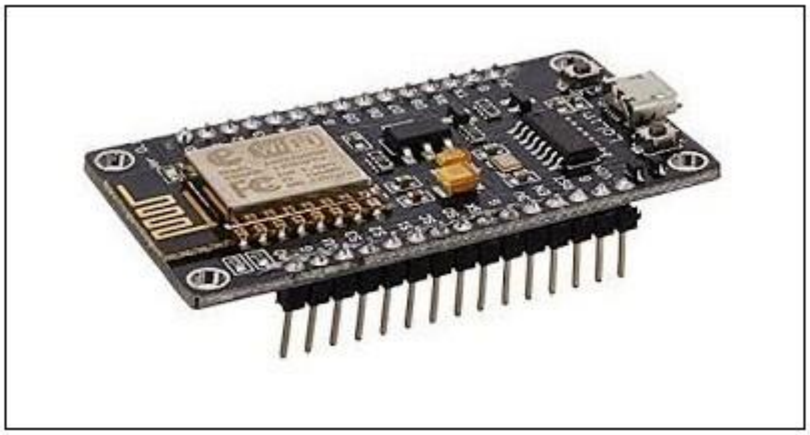


Fig 3.1.2.1: ESP8266

Software Specifications are:

Operating Systems	:	XTOS
CPU	:	ESP8266
Storage	:	128KB
Memory	:	4MB

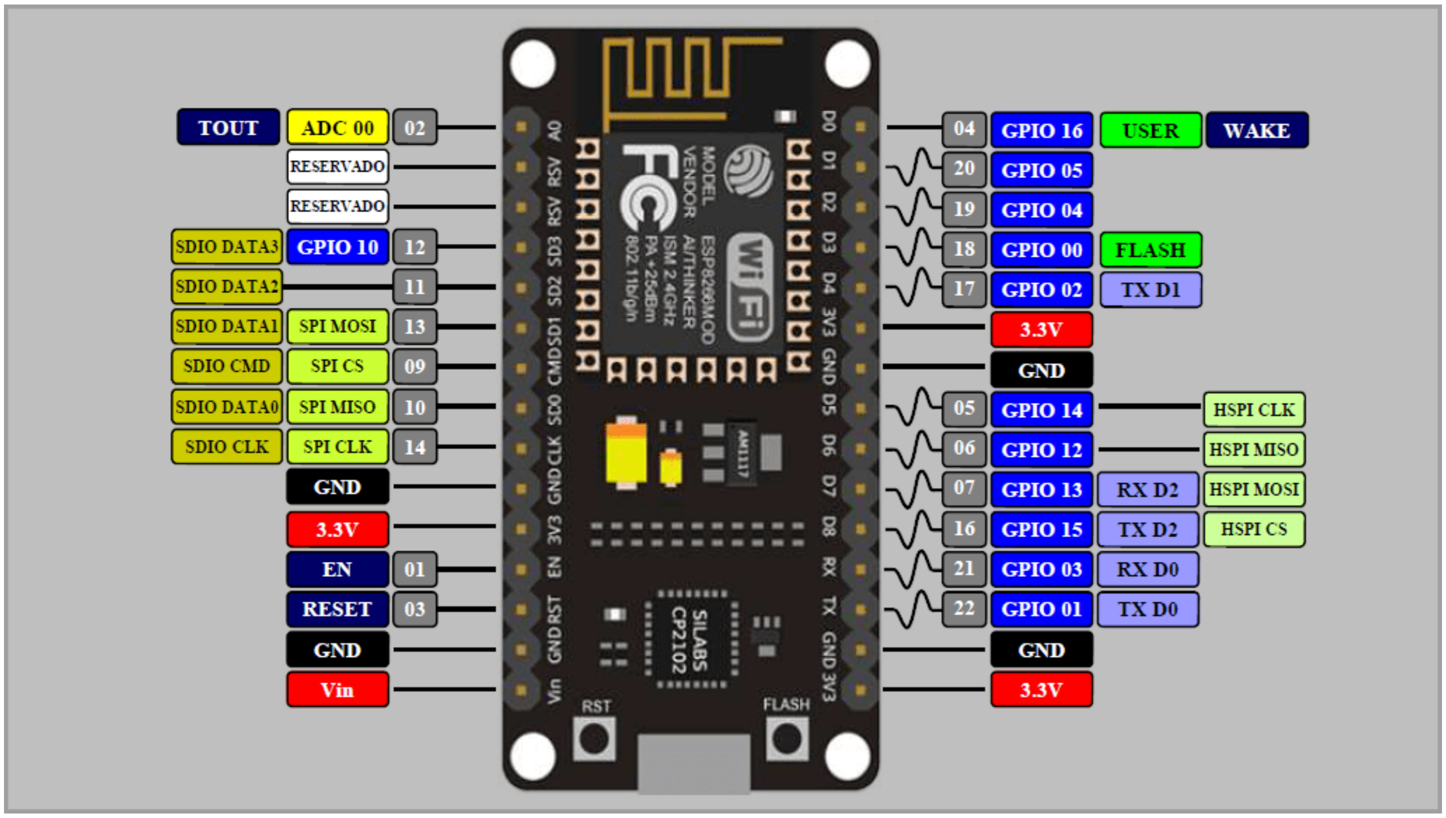
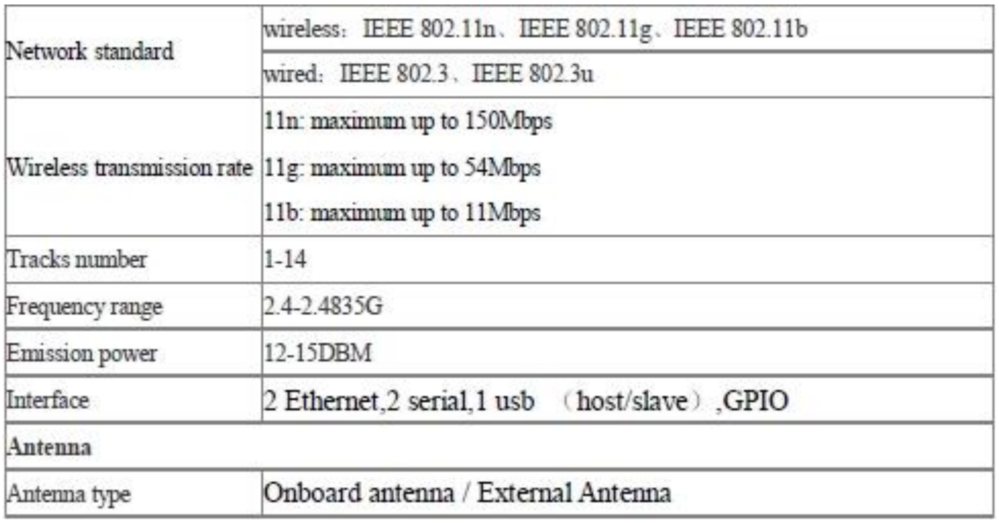


Fig 3.1.2.2: Pins of ESP8266

NodeMCU is the WiFi equivalent of Ethernet module. It combines the features of WiFi access point and station + microcontroller. These features make the NodeMCU extremely powerful tool for WiFi networking. It can be used as access point and/or station, host a web server or connect to internet to fetch or upload data.

Technical Specifications are :



Network standard	wireless: IEEE 802.11n、IEEE 802.11g、IEEE 802.11b
	wired: IEEE 802.3、IEEE 802.3u
Wireless transmission rate	11n: maximum up to 150Mbps 11g: maximum up to 54Mbps 11b: maximum up to 11Mbps
Tracks number	1-14
Frequency range	2.4-2.4835G
Emission power	12-15DBM
Interface	2 Ethernet,2 serial,1 usb （host/slave）,GPIO
Antenna	
Antenna type	Onboard antenna / External Antenna

Table 3.1.2: Technical Specifications of ESP 8266

3.1.3-RTC DS3231:

A real-time clock (RTC) is a computer clock (most often in the form of an integrated circuit) that keeps track of the current time. Real-time clock IC's operate as the clock in various devices. This clock function operates even when the power is turned off. In portable devices, it is important to reduce the current consumption of the real-time clock IC to extend the battery life.

Fig 3.1.3: Real Time Clock

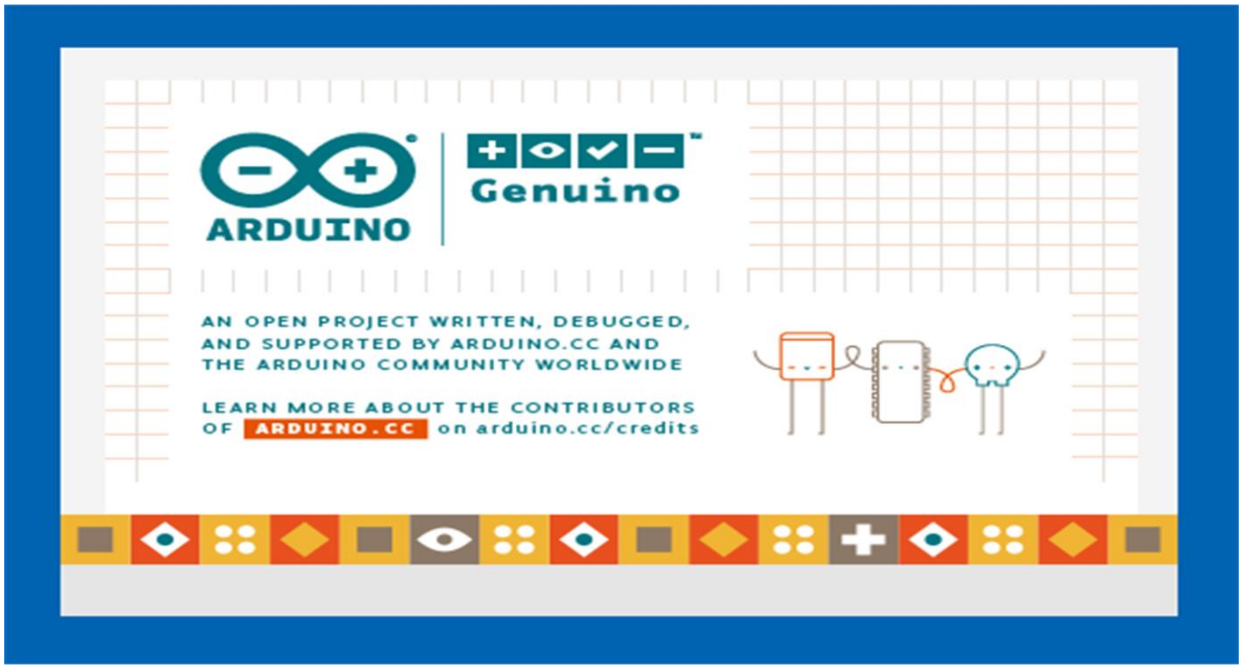
Benefits of using RTC:

- Low power consumption (important when running from alternate power)
- Frees the main system for time-critical tasks
- Sometimes more accurate than other methods

3.2-SOFTWARE ENVIRONMENT:

3.2.1-Arduino IDE:

The Arduino IDE is an open source application that is used to write and upload programs to the Arduino board. This IDE supports C and C++ programming languages with some cut down features. This IDE is available for all operating systems including Windows, Mac and Linux. It has a toolbar which includes Files (new, save), Create new sketches, Edit (copy, paste), Sketch (for compiling) and Tools (to test projects).


ARDUINO
Genuino
AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE
LEARN MORE ABOUT THE CONTRIBUTORS OF ARDUINO.CC on arduino.cc/credits

. **Fig3.2.1: Arduino IDE**

3.2.2-ThingSpeak IoT platform:

ThingSpeak is and open source IoT platform that lets us to collect data from sensors and store data in the cloud. It supports users to visualize the data, analyze it using Matlab. You can either keep your data private or public. First you need to create an account and register with ThingSpeak. Then you should create a channel which can have up to 8 fields. You will get a channel number and two API keys will be generated as write key and read key which are used to read and write the sensors values to the fields in our channel. To use ThingSpeak, we want to include its library file by going to settings in Arduino IDE. Go to manage libraries -> search for ThingSpeak and download it.

3.2.3-ESP8266 Library:

To bring support for the Esp8266 wifi module to the Arduino IDE, we should install the library of esp8266. To install it, follow the instructions as below in the IDE.

- Open the preferences window from the Arduino IDE. Go to File > Preferences

- Enter http://arduino.esp8266.com/stable/package_esp8266com_index.json into the "Additional Board Manager URLs" field as shown in the figure below. Then, click the "OK" button.

- Go to Tools > Board > Boards Manager. Open Boards Manager.

- Select the ESP8266 board menu and install "esp8266".

3.3.1 PROJECT CODE:

```
#include <Wire.h>       //I2C library

#include <RtcDS3231.h>  //RTC library

RtcDS3231<TwoWire> rtcObject(Wire);

#include <SoftwareSerial.h>;

#include <stdlib.h>;

#include <ESP8266WiFi.h>;

 #include <WiFiClient.h>;

#include <ThingSpeak.h>;

const char* ssid = "krishna"; //Your Network SSID

 const char* password = "123456789"; //Your Network Password

WiFiClient client;

unsigned long myChannelNumber = 630305; //Your Channel Number (Without Brackets)

const char * myWriteAPIKey = "FRRLEK2KQZNXTF1Z"; //Your Write API Key


#define trigPin1 2

#define echoPin1 0

#define trigPin2 16
```

```
#define echoPin2 14

#define trigPin3 13

#define echoPin3 15

long duration, distance, UltraSensor1, UltraSensor2,
UltraSensor3,a=0,b=0,c=0,t11,t12,t13,t21,t22,t23,t31,t32,t33;

//we'll use these variable to store and generate data

void setup() {

 WiFi.begin(ssid, password);

  ThingSpeak.begin(client);

  Serial.begin(115200);  //Starts serial connection

  rtcObject.Begin();     //Starts I2C

 RtcDateTime currentTime = RtcDateTime(16, 05, 18, 21, 20, 0); //define date and time object

  rtcObject.SetDateTime(currentTime); //configure the RTC with object



  pinMode(trigPin1, OUTPUT);                    // from where we will transmit the ultrasonic wave

pinMode(echoPin1, INPUT);                      //from where we will read the reflected wave

//pinMode(LED_first_ping, OUTPUT);                // from where we will control the LED

//setup pins second sensor

pinMode(trigPin2, OUTPUT);

pinMode(echoPin2, INPUT);
//pinMode(LED_second_ping, OUTPUT);

//setup pins third sensor

pinMode(trigPin3, OUTPUT);

pinMode(echoPin3, INPUT);

}

void SonarSensor(int trigPinSensor,int echoPinSensor)//it takes the trigPIN and the echoPIN

{

digitalWrite(trigPinSensor, LOW);// put trigpin LOW

delayMicroseconds(2);// wait 2 microseconds


digitalWrite(trigPinSensor, HIGH);// switch trigpin HIGH

delayMicroseconds(10); // wait 10 microseconds

digitalWrite(trigPinSensor, LOW);// turn it LOW again

duration = pulseIn(echoPinSensor, HIGH);

//pulseIn funtion will return the time on how much the  configured pin remain the level HIGH
or LOW; in this case it will return how much time echoPinSensor stay HIGH

distance= (duration/2) / 29.1;

// first we have to divide the duration by two

}
```

```
void loop() {



// Connect or reconnect to WiFi

  if(WiFi.status() != WL_CONNECTED){

    Serial.print("Attempting to connect to SSID: ");

    Serial.println(ssid);

    while(WiFi.status() != WL_CONNECTED)

    {

      WiFi.begin(ssid, password);  // Connect to WPA/WPA2 network. Change this line if using
open or WEP network

          delay(5000);

    }

    Serial.println("\nConnected.");

  }

  RtcDateTime currentTime = rtcObject.GetDateTime();    //get the time from the RTC

 char str[20];   //declare a string as an array of chars

SonarSensor(trigPin1,  echoPin1);                          // look bellow to find the difinition of the
SonarSensor function

UltraSensor1 = distance;     // store the distance in the first variable

if(a==0){

if(UltraSensor1<15)
```

```
{

   t11=(currentTime.Second()+60*currentTime.Minute()+3600*currentTime.Hour());



 a=1;

}

}

if(a==1 && UltraSensor1>=15)

{

  t12=(currentTime.Second()+60*currentTime.Minute()+3600*currentTime.Hour());

  a=0;

  t13=t12-t11;

  Serial.print("Time Period in slot 1: ");

  Serial.print(t13);

  Serial.println(" seconds");

}

SonarSensor(trigPin2,echoPin2);              // call the SonarSensor function again with the second
sensor pins

UltraSensor2 = distance;   // store the new distance in the second variable

if(b==0){

if(UltraSensor2<15){
 t21=(currentTime.Second()+60*currentTime.Minute()+3600*currentTime.Hour());

   b=1;

}

}

if(b==1 && UltraSensor2>=15){

  t22=(currentTime.Second()+60*currentTime.Minute()+3600*currentTime.Hour());

  b=0;

  t23=t22-t21;

  Serial.print("Time Period in slot 2: ");

  Serial.print(t23);

  Serial.println(" seconds");

}

SonarSensor(trigPin3,echoPin3);               // call the SonarSensor function again with the second
sensor pins

UltraSensor3 = distance;   // store the new distance in the second variable

if(c==0){

if(UltraSensor3<15){



t31=(currentTime.Second()+60*currentTime.Minute()+3600*currentTime.Hour());

   c=1;
```

```
}

}

if(c==1 && UltraSensor3>=15){

  t32=(currentTime.Second()+60*currentTime.Minute()+3600*currentTime.Hour());

  c=0;

  t33=t32-t31;

  Serial.print("Time Period in slot 3: ");

  Serial.print(t33);

  Serial.println(" seconds");

}

  ThingSpeak.setField(1, UltraSensor1);

  ThingSpeak.setField(2, UltraSensor2);

  ThingSpeak.setField(3, UltraSensor3);

  ThingSpeak.setField(4, t13);

  ThingSpeak.setField(5, t23);



ThingSpeak.setField(6, t33);

  // write to the ThingSpeak channel

  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

   if(x == 200){
```

```
   Serial.println("Channel update successful.");

 }

 else{

   Serial.println("Problem updating channel. HTTP error code " + String(x));

 }
```

The Proposed System consists of ultrasonic sensor, PIR sensor and a Real Time Clock(RTC) connected with ESP8266 micro-controller which is then connected to ThingSpeak platform and a Mobile application.

4.1-Data flow diagram:

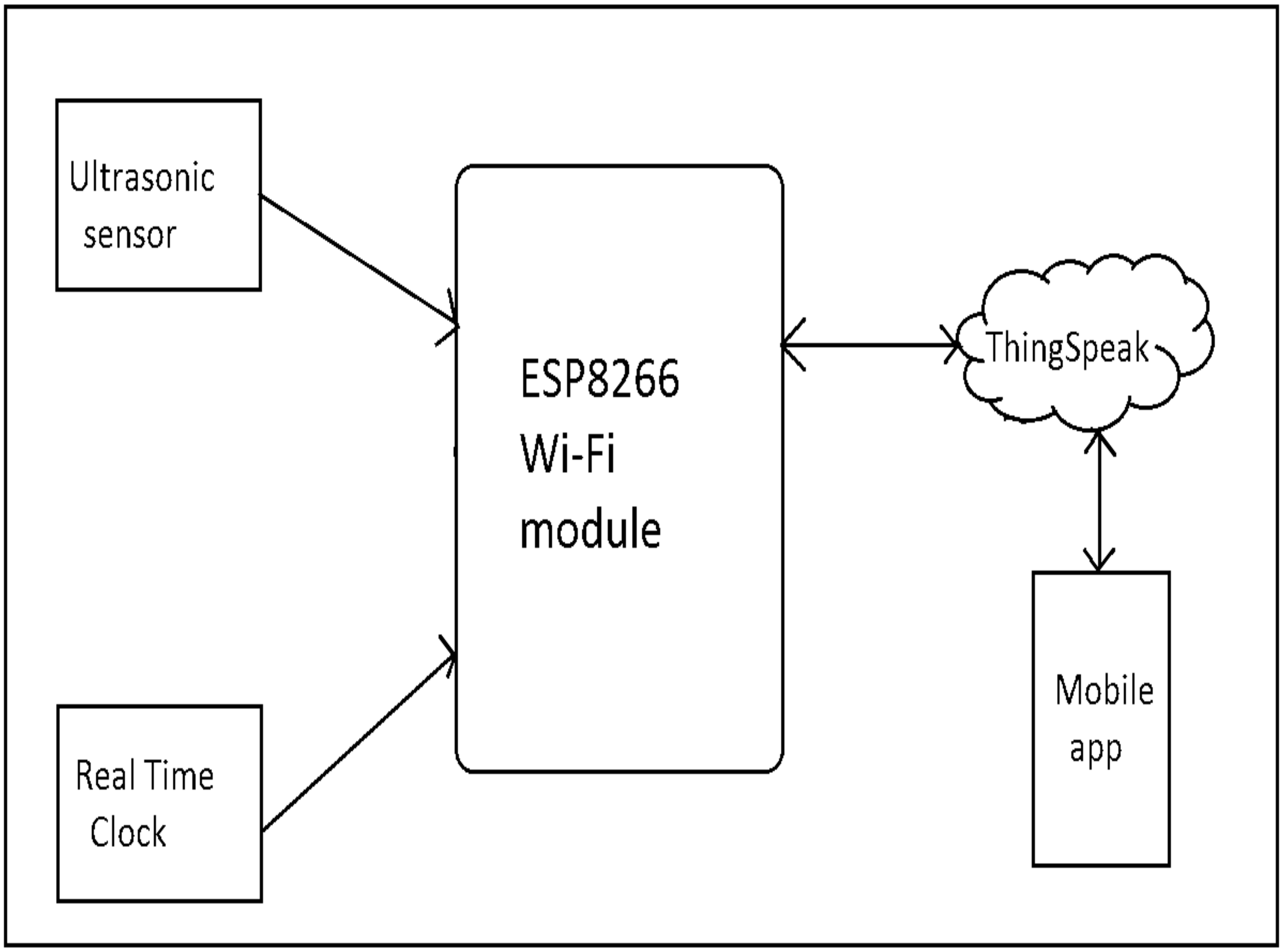


Fig4.1.1: Data flow diagram

The ESP 8266 micro-controller is connected to 3.3V DC power supply. Ultrasonic sensors are used to calculate distance of an object from the sensor. PIR sensor is used to detect the motion of an object. An entry and exit gate are arranged at the parking zone and two PIR sensors are placed at entry and exit gate.

When a vehicle arrives at entry gate, the motion of the object is detected by PIR sensor and gate will open. The vehicle is parked at one of the free parking spaces and the presence of vehicle at particular space can be identified by Ultrasonic sensor by measuring its distance. From the point of time the vehicle is parked, the RTC starts calculating time until vehicle is drawn from the parking space.

The data of the Sensors is fetched by ESP 8266 Wi-Fi module which sends the data to ThingSpeak IoT platform. We can monitor the data using graphs and charts.

4.2-Circuit Diagram:

The Circuit diagram of the proposed system is as follows

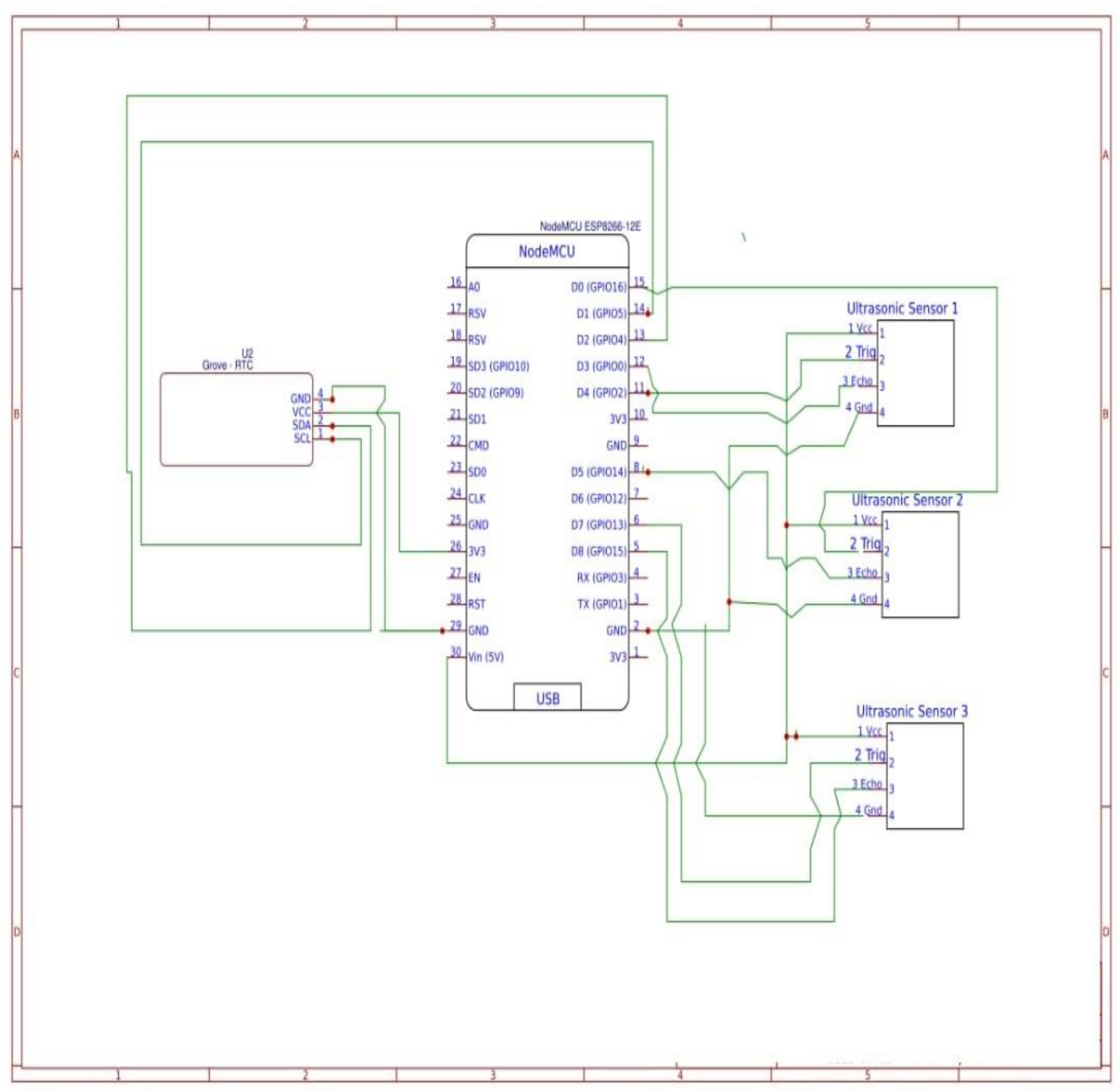


Fig4.2.1: Circuit Diagram

The connections of first Ultrasonic sensor with ESP 8266 are as follows:

Ultrasonic Sensor	**ESP 8266**
Vcc	3.3V
Gnd	Gnd
Echo	D4
Trig	D3

Table4.2.1: Connections of first Ultrasonic sensor with ESP 8266

The connections of second Ultrasonic sensor with ESP 8266 are as follows:

Ultrasonic Sensor	**ESP 8266**
Vcc	3.3V
Gnd	Gnd
Echo	D4
Trig	D3

Table4.2.2: Connections of second Ultrasonic sensor with ESP 8266

The connections of third Ultrasonic sensor with ESP 8266 are as follows:

Ultrasonic Sensor	**ESP 8266**
Vcc	3.3V
Gnd	Gnd
Echo	D4
Trig	D3

Table4.2.3: Connections of third Ultrasonic sensor with ESP 8266

The connections of RTC with ESP 8266 are as follows:

RTC	**ESP 8266**
SDA	D2
SCL	D1
Gnd	Gnd
Vcc	3.3V

Table4.2.4: Connections of RTC with ESP 8266

4.3-Model:

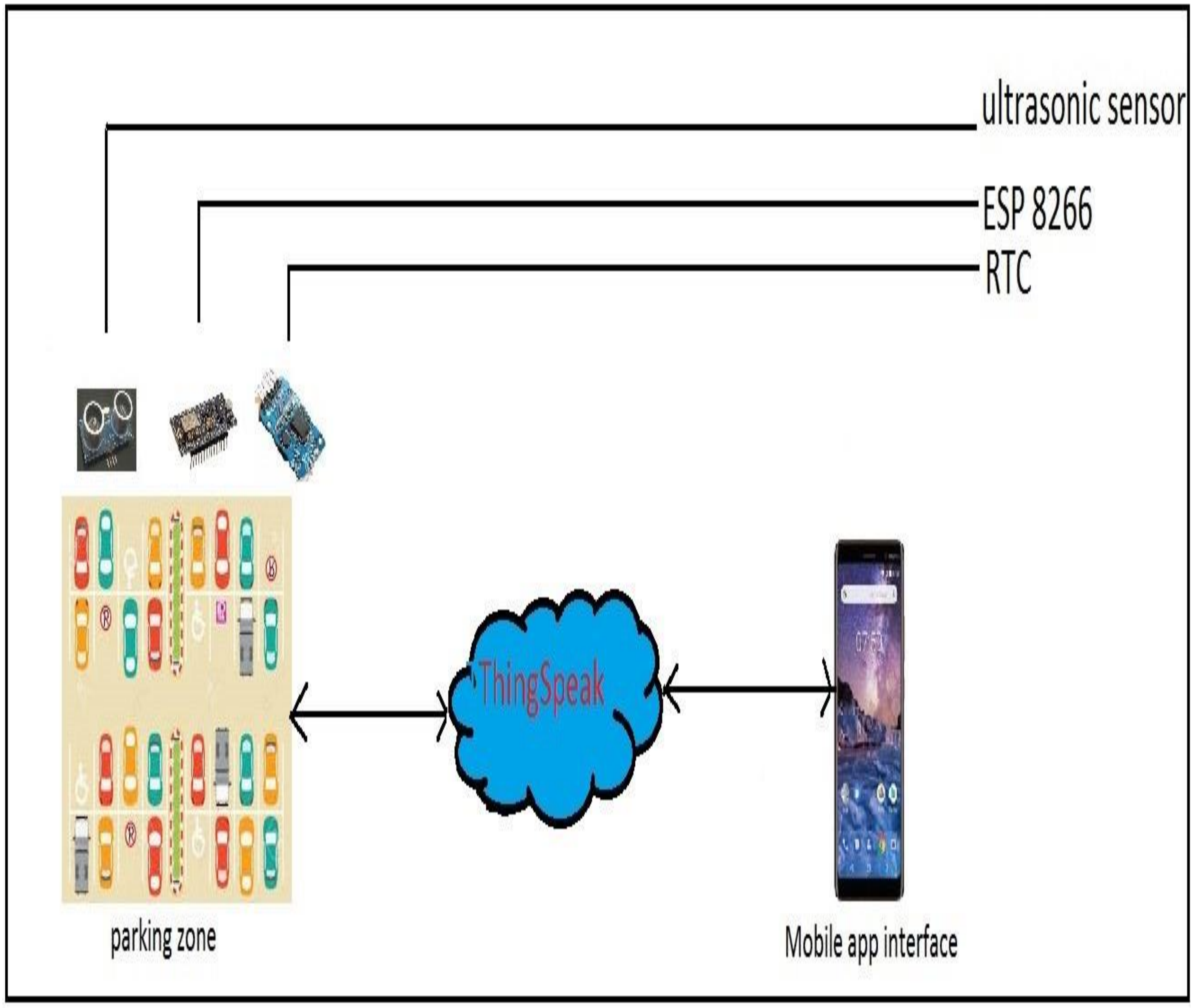


Fig4.3.1: Components of the model

4.4-Process:

- First all the required hardware like Sensors, Wires, Microcontroller and others are brought and required software like ARDUINO IDE are installed.
- Next, an account in ThingSpeak IoT platform is created which is used to monitor the data from the cloud and log in.
- Then a channel is created with some name and then a unique channel ID will be generated which are helpful in reading and writing the data from Sensors.
- Arduino IDR is the open source software using which programs for sensors can be written and uploaded to the board.
- Next, IP address of the ThingSpeak and channel ID, API key are given in the program.
- After all the Programming is done, Data can be observed in ThingSpeak channel.

The Estimated cost for the entire project is –

• NodeMCU	--	380/-
• Node cable	--	50/-
• Bread Board	--	150/-
• SRO4_Ultrasonic sensor	--	240/-
• RTC 3231	--	150/-
• PIR Sensor	--	180/-
• 8*8 LED Matrix	--	300/-
• LED	--	3/-
• Resistor	--	6/-
• Jumper wires – M-M	--	33/-
M-F	--	33/-
F-F	--	33/-

The total cost estimated is One thousand five hundred and fifty eight rupees (1558/-)

Ultrasonic sensor is interfaced with Node MCU as follows and the distance measured can be seen on the serial monitor of Arduino IDE.

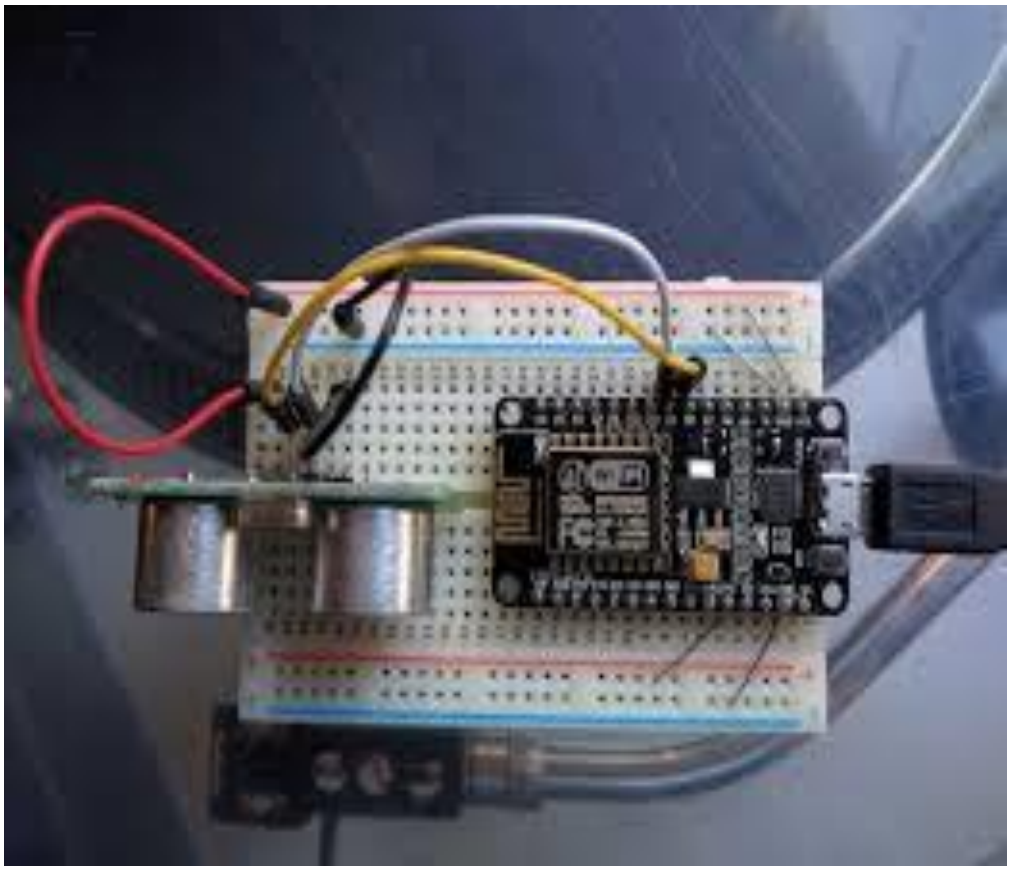
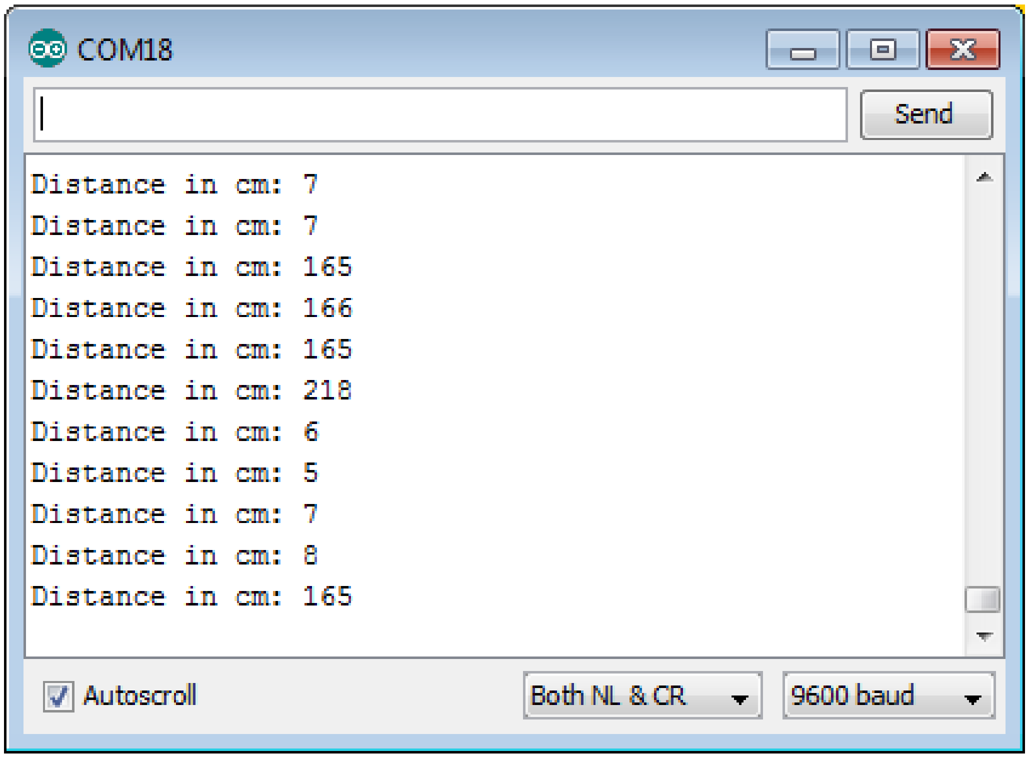


Fig6.1: Interfacing Ultrasonic Sensor with Node MCU

Both Ultrasonic sensor is interfaced with Node MCU and the results are observed on the serial monitor of Arduino IDE.



```
Distance in cm: 7
Distance in cm: 7
Distance in cm: 165
Distance in cm: 166
Distance in cm: 165
Distance in cm: 218
Distance in cm: 6
Distance in cm: 5
Distance in cm: 7
Distance in cm: 8
Distance in cm: 165
```

Autoscroll: ✓ Both NL & CR 9600 baud

Fig6.2: Output in serial Monitor

The prototype of the smart parking system when all slots are empty is shown as

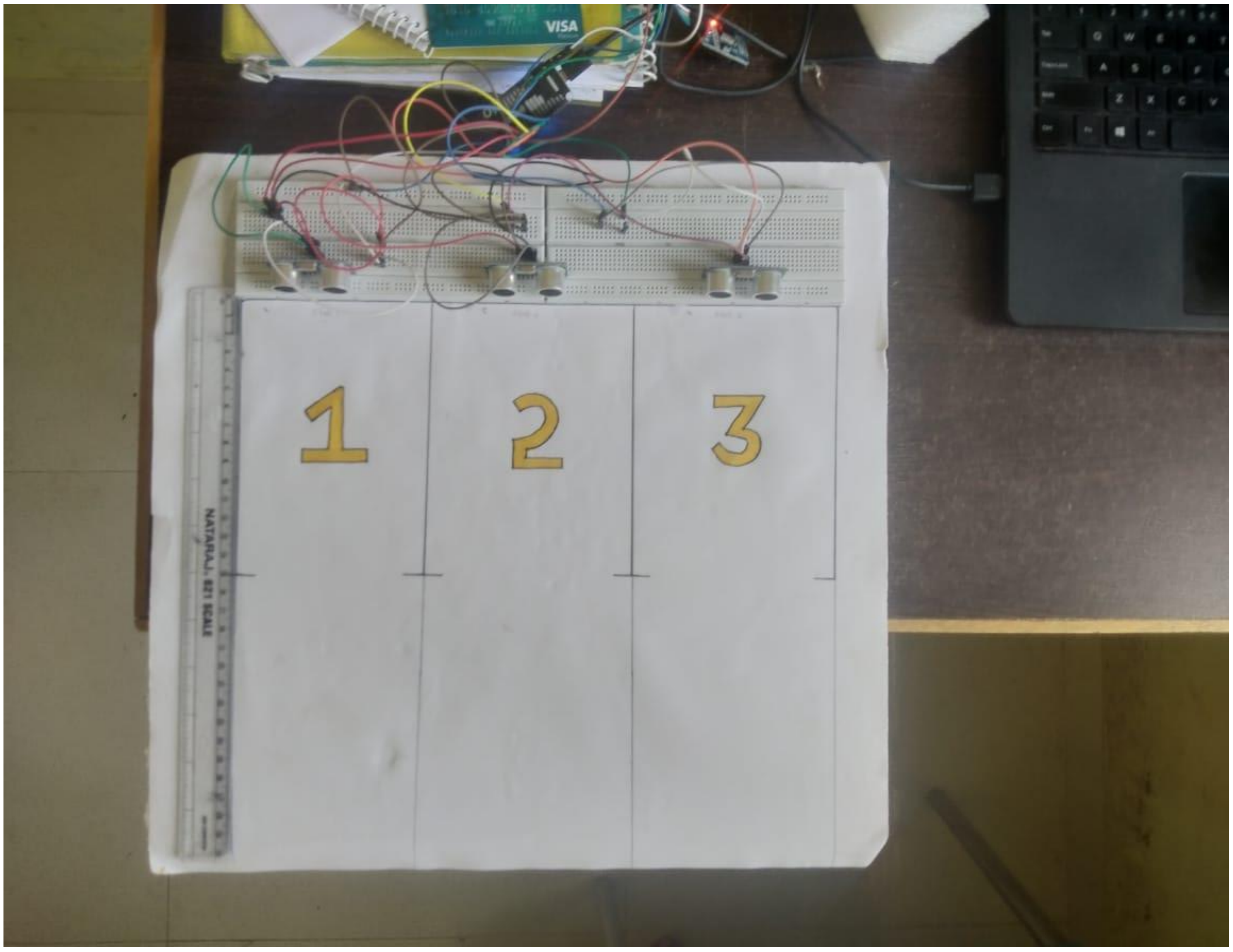
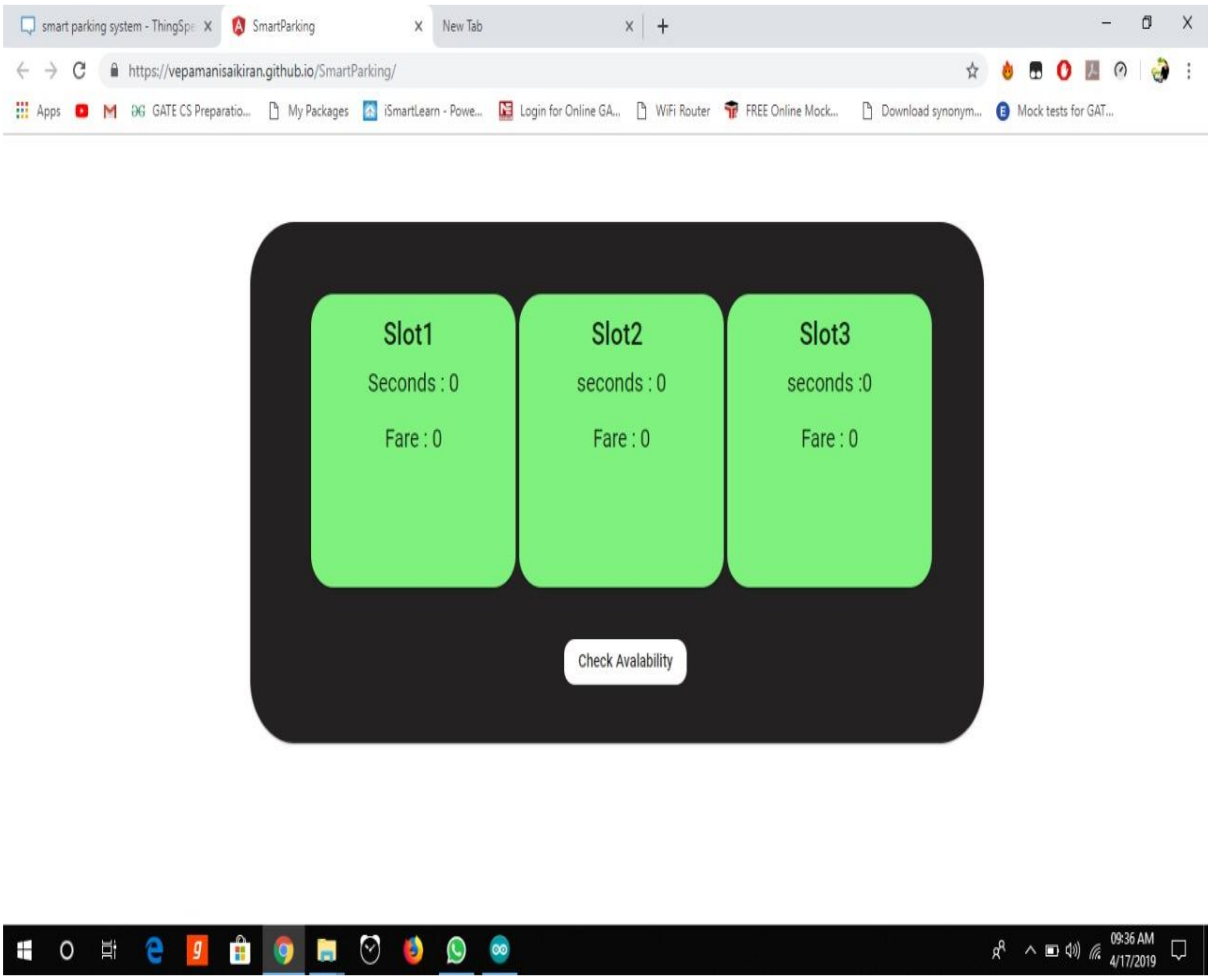


Fig6.3: prototype of the system when all slots are empty

The status of the parking slots when no slot occupied is shown in web app as



Slot1
Seconds : 0
Fare : 0

Slot2
seconds : 0
Fare : 0

Slot3
seconds :0
Fare : 0

Check Avalability

Fig6.4: status of parking slots in web app when all slots are empty

The prototype of the smart parking system when all slots are busy is shown as

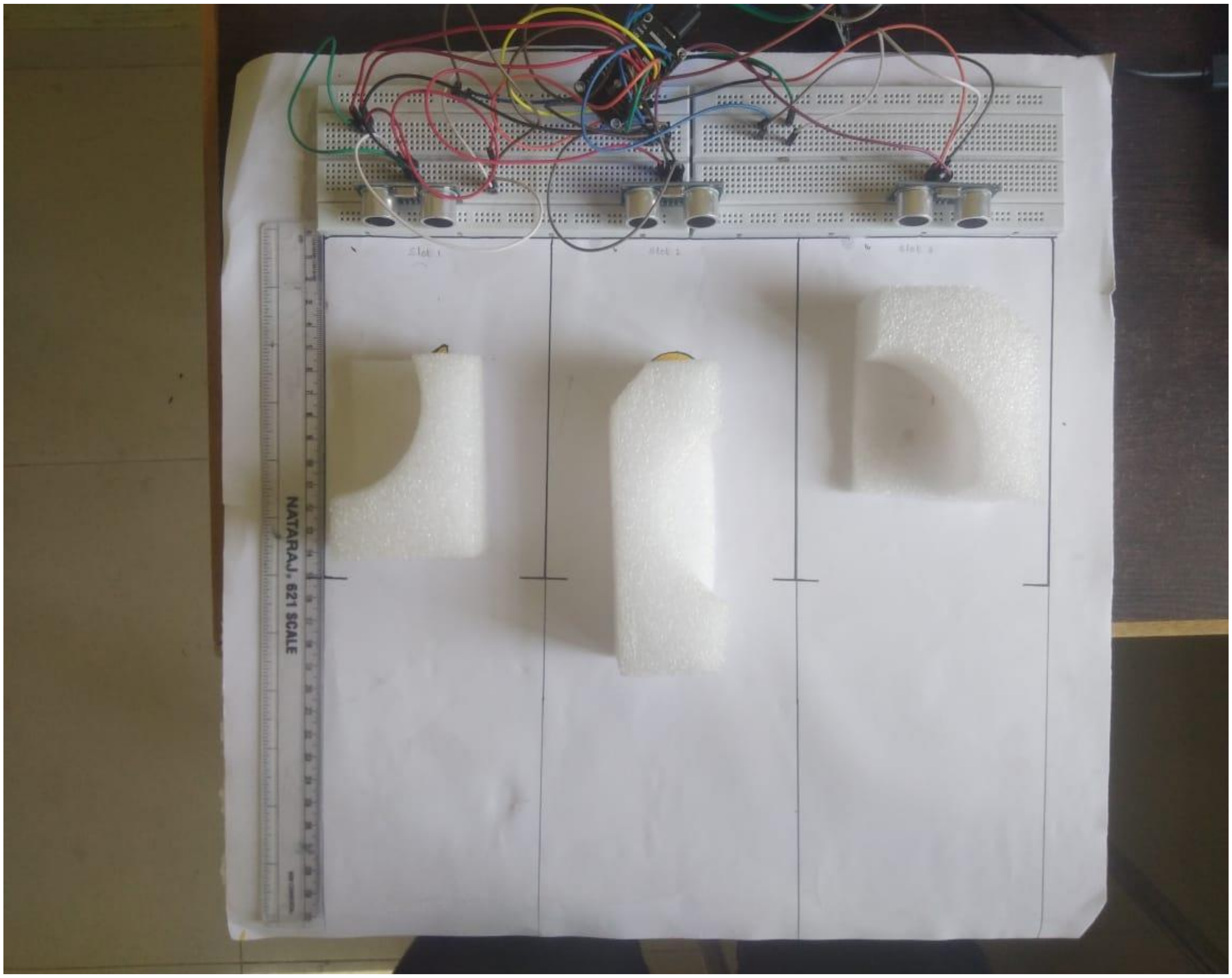
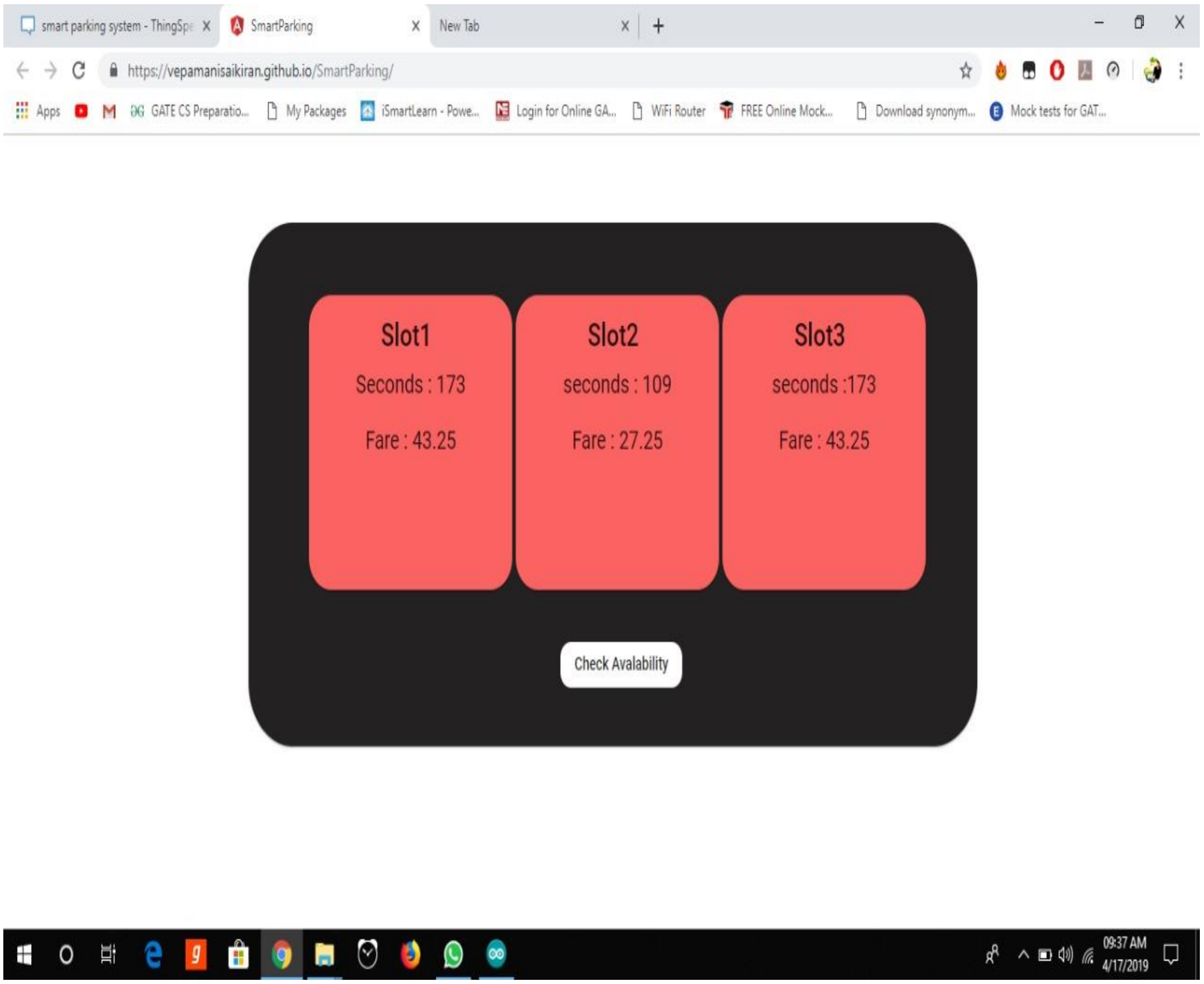


Fig6.5: prototype of the system with objects

The status of the parking slots when all slots are busy is shown in web app as


smart parking system - ThingSpe
SmartParking
New Tab
https://vepamanisaikiran.github.io/SmartParking/
Check Avalability

Slot1
Seconds : 173
Fare : 43.25

Slot2
seconds : 109
Fare : 27.25

Slot3
seconds :173
Fare : 43.25

Fig6.6: status of parking slots in web app when all slots are busy

The data from the sensor1 is observed in ThingSpeak cloud platform in the form of timeline charts, object indicator as follows

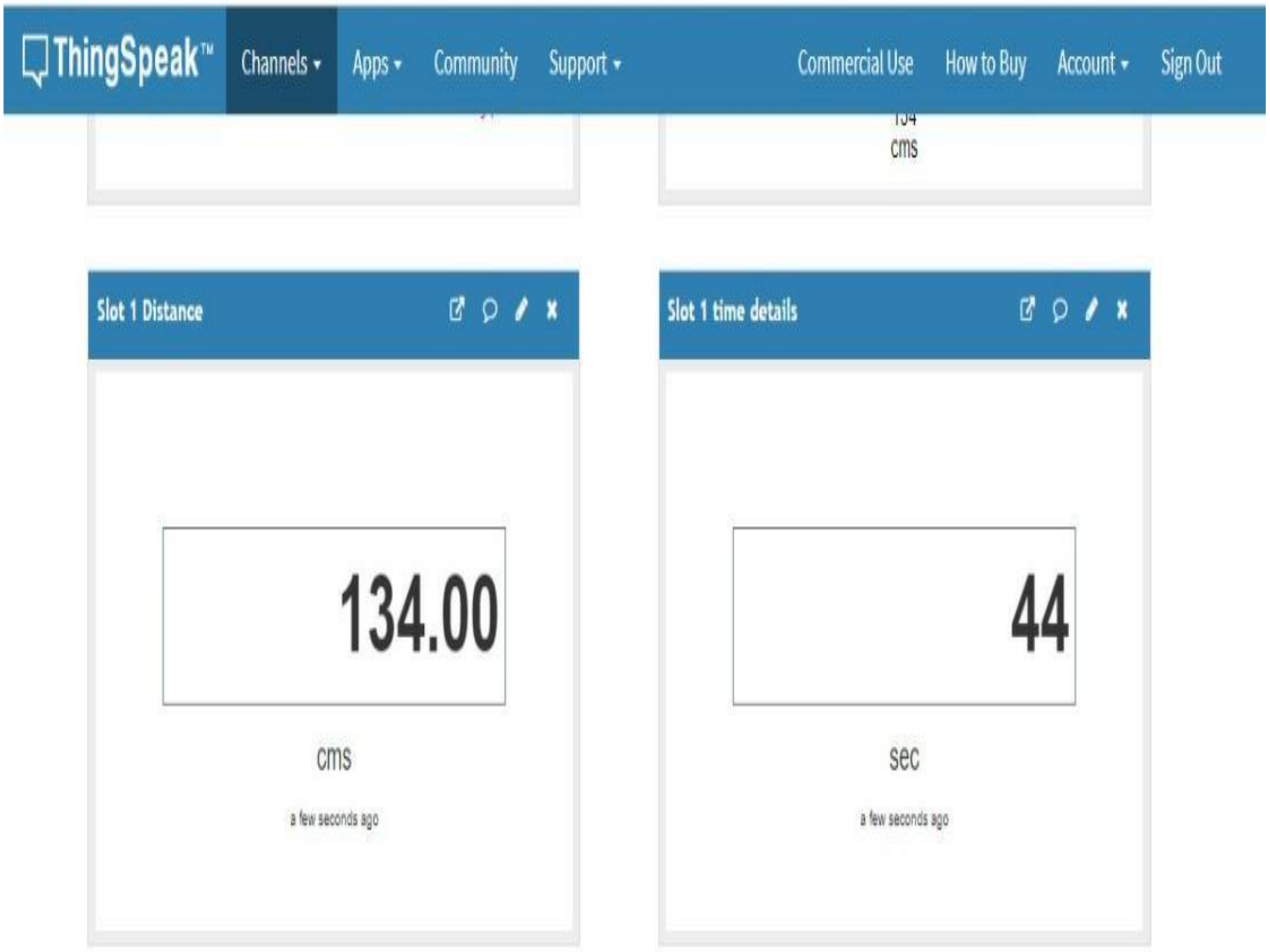


Fig6.7: slot 1 distance and time details in ThingSpeak platform

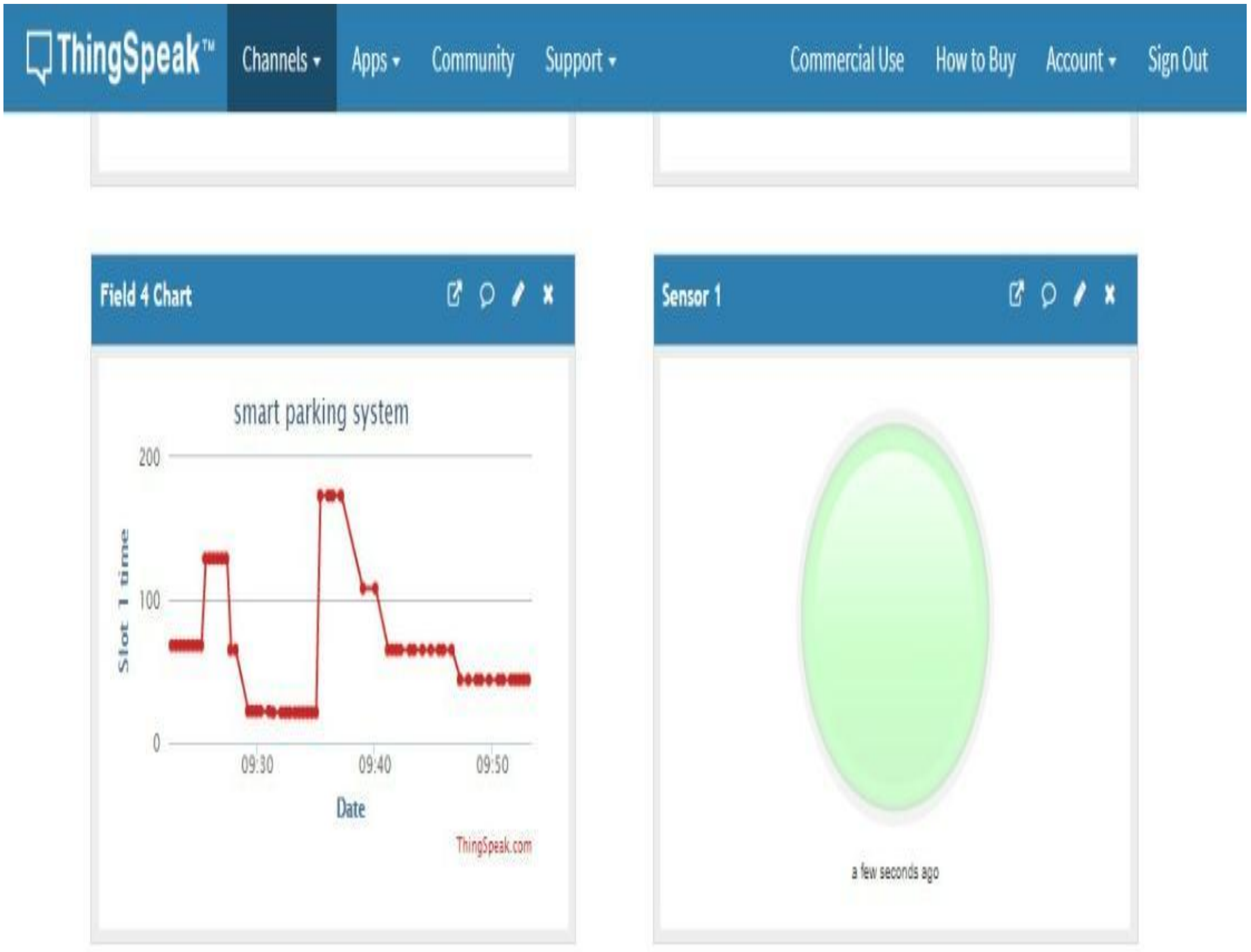


Fig6.8: slot 1 timeline chart and object indication details in ThingSpeak platform

When slot 1 is empty and slots 2and 3 are occupied, the results are shown as

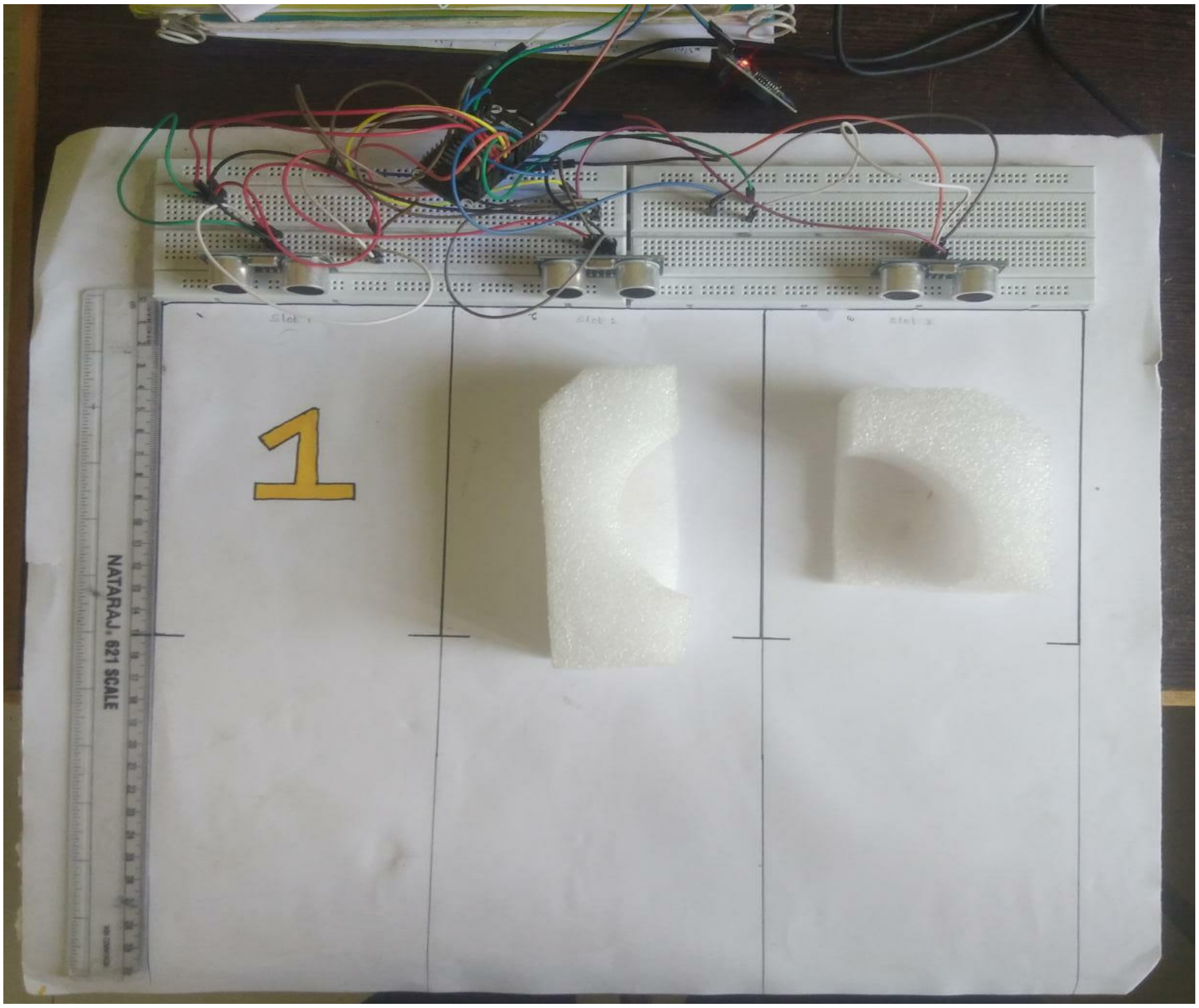
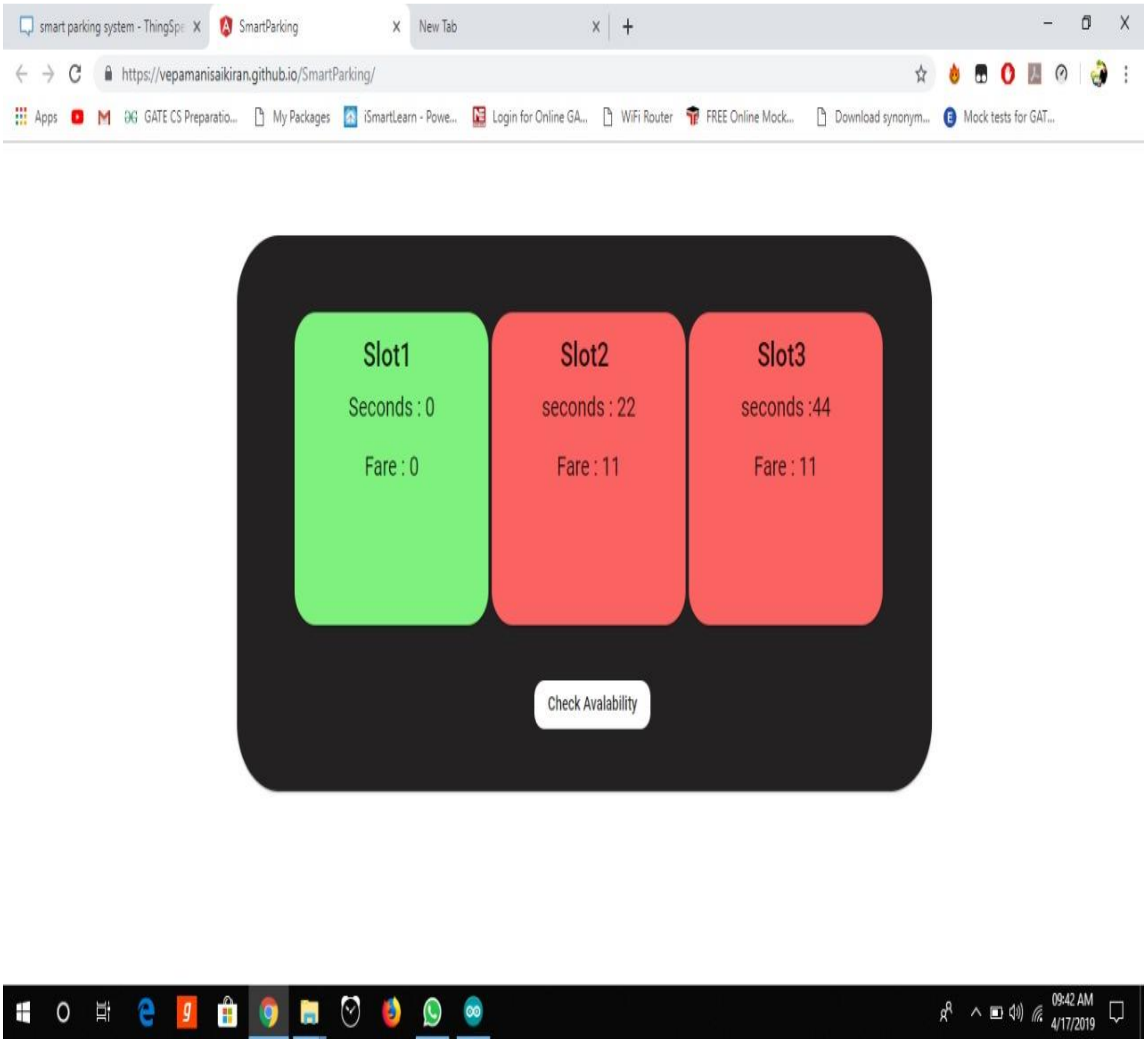


Fig6.16: prototype of the system when only slot 1 is empty



Slot1
Seconds : 0
Fare : 0

Slot2
seconds : 22
Fare : 11

Slot3
seconds :44
Fare : 11

Fig6.17: status of parking slots in web app when only slot 1 is empty

This system helps people who want to park their vehicles in the crowd areas like shopping malls, theaters, offices by using mobile app in their mobile phones. They can reserve a parking slot prior to their journey without worrying about parking. The development of these system reduces the traffic in parking areas with the advanced reservation and they can see the status of the parking zone through the app. This system prevents parking zone authorities from collecting abnormal parking fares and provides a normal charge for the time period the vehicle is parked.

https://www.researchgate.net/publication/303842610_IoT_based_Smart_Parking_System

https://www.sciencedirect.com/science/article/pii/S1877042812043042

https://github.com/arc13-13/Smart-Car-Parking-System/find/master?q

https://ieeexplore.ieee.org/document/7562735

https://www.ukessays.com/essays/information-technology/literature-review-on-car-parking-system-information-technology-essay.php