SMART PARKING SYSTEM

REPORT

Submitted by
THIRUMALAI KG (20IT112)

THARUN D (20IT111)

SIVANMANI C (20IT127)

SHYLANDER S (20IT094)

in partial fulfillment for the completion of course Mobile Application Development

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI–15

(A Govt. Aided, Autonomous Institution, Affiliated to Anna University)

JUNE 13

THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI-15

(A Govt. Aided, Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report "**SMART PARKING SYSTEM"** is the bonafide work of "**THIRUMALAI KG (20IT112), THARUN D (20IT112), SIVANMANI C (20IT127), SHYLANDER S (20IT094)** who carried out the project work under my supervision during the Academic Year 2021 -2022.

SIGNATURE

Ms. S. Karthiga, Ms. C.V.Nisha Angeline

ASSISTANT PROFESSOR,

INFORMATION TECHNOLOGY

THIAGARAJAR COLLEGE OF ENGINEERING,

MADURAI-15

Title-SMART PARKING SYSTEM

1. Abstract

The smart parking system is a solution that utilizes the Internet of Things (IoT) and Radio Frequency Identification (RFID) technology to create a more efficient and convenient parking experience. This system enables drivers to find available parking spots easily and reserve them in advance through a mobile application.

The system consists of RFID tags, readers, and a central server that manages the data collected by the readers. RFID tags are attached to vehicles, and readers are installed at entrances and exits of parking lots. The readers detect the RFID tags and communicate with the central server to determine the occupancy status of each parking spot.

The central server processes the data collected by the readers and provides real-time information on parking availability to users via a mobile application. The mobile application allows users to search for available parking spots, reserve them in advance, and make payment.

2. Importance of mobile Application for fishermen:

The mobile application is a crucial component of the smart parking system using IoT and RFID technology. It serves as the interface between the user and the system, allowing users to access real-time information on parking availability, reserve parking spots, and make payments.

Without the mobile application, users would have no way of accessing the system and would be forced to rely on traditional methods of finding parking spots, which can be time-consuming and inefficient. The mobile application provides a convenient and hassle-free way for users to locate available parking spots and reserve them in advance, reducing the time and effort required to find a parking spot.

3. Reading of RFID tags and detection of parking slot:

In our project we use RFID tags for unique identity. These RFID tags are recognized by the readers and the entry and leaving time is sent to the mobile application. Based on the entry and leaving time amount is calculated and shown in the mobile application. Users can pay through the mobile app. Moreover users can select the parking slot through the app. The status of the parking slot is shown on the app using IR sensor. If a vehicle is present in any slot it is indicated as red color in the mobile application. All the entry , leaving time and payment details are stored in firebase.

4. Node MCU

NodeMCU is an open-source firmware and development board that is based on the ESP8266 Wi-Fi module.The NodeMCU board can be used in this project to provide connectivity between the RFID reader, the central server, and the internet.The NodeMCU board can be used as a gateway between the RFID reader and the central server. It can collect the data from the RFID reader and send it to the central server via Wi-Fi. This allows the central server to receive real-time updates on the occupancy status of the parking spots.Furthermore, the NodeMCU board can be used to connect the smart parking system to the internet. By using the Wi-Fi capabilities of the NodeMCU board, the system can communicate with the mobile application, allowing users to access real-time information on parking availability, reserve parking spots, and make payments.

Features of Node MCU:

Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106.
Operating Voltage: 3.3V.
Input Voltage: 7-12V.
Digital I/O Pins (DIO): 16.
Analog Input Pins (ADC): 1. UARTs: 1. SPIs: 1. I2Cs: 1.

Uploading Code to NodeMCU:

1. Connect the NodeMCU with a micro USB cable to your PC.
2. Press and hold flash button and press the reset button once and now you leave the flash button
3. Select the NodeMCU v1.0 as board.
4. Select 115200 as baud rate and select the correct COM port.
5. Press the upload button, it may take 2 min to compile the code and another one minute to upload to NodeMCU.

Code:

https://github.com/TharunDharmaraj/IoT_Connection_Parking_App/blob/master/parking.iso

IOT part:

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <SPI.h>
#include <MFRC522.h>
#include <NTPClient.h>
#include <WiFiUdp.h>


#define FIREBASE_HOST "YOUR_FIREBASE_REALTIME_DB_LINK_EXCEPT_HTTPS://"
#define FIREBASE_AUTH "FIREBASE_AUTH_FROM_YOUR_PACKAGE.JSON"
#define WIFI_SSID "WIFI_NAME"
#define WIFI_PASSWORD "WIFI_PASSWORD"
#define aaa D0

const long utcOffsetInSeconds = 3600;
constexpr uint8_t RST_PIN = D3;
constexpr uint8_t SS_PIN = D4;

WiFiUDP ntpUDP;
String weekDays[7]={"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday"};
String months[12]={"January", "February", "March", "April", "May", "June", "July", "August",
"September", "October", "November", "December"};
NTPClient timeClient(ntpUDP, "pool.ntp.org", utcOffsetInSeconds);


MFRC522 rfid(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;

int count = 0;
bool canCheck = 0;
bool s1;
String tag;
```

```
void setup() {
  Serial.begin(9600);

  pinMode(aaa,INPUT);
  SPI.begin();
  rfid.PCD_Init();

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }

  Serial.println("WiFi connected");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Serial.println("Firebase connected");

  timeClient.begin();
  timeClient.setTimeOffset(0);
}

void loop() {

    if ( ! rfid.PICC_IsNewCardPresent())
      return;

    if (rfid.PICC_ReadCardSerial())
    {
      tag = "";
      for (byte i = 0; i < 4; i++)
      {
        tag += rfid.uid.uidByte[i];
      }

      timeClient.update();
      time_t epochTime = timeClient.getEpochTime();
      String formattedTime = timeClient.getFormattedTime();
      struct tm *ptm = gmtime ((time_t *)&epochTime);
      int monthDay = ptm->tm_mday;
      int currentMonth = ptm->tm_mon+1;
      String currentMonthName = months[currentMonth-1];
      int currentYear = ptm->tm_year+1900;
      String currentDate = String(monthDay) + " " + String(currentMonth) + " " + String(currentYear);
```

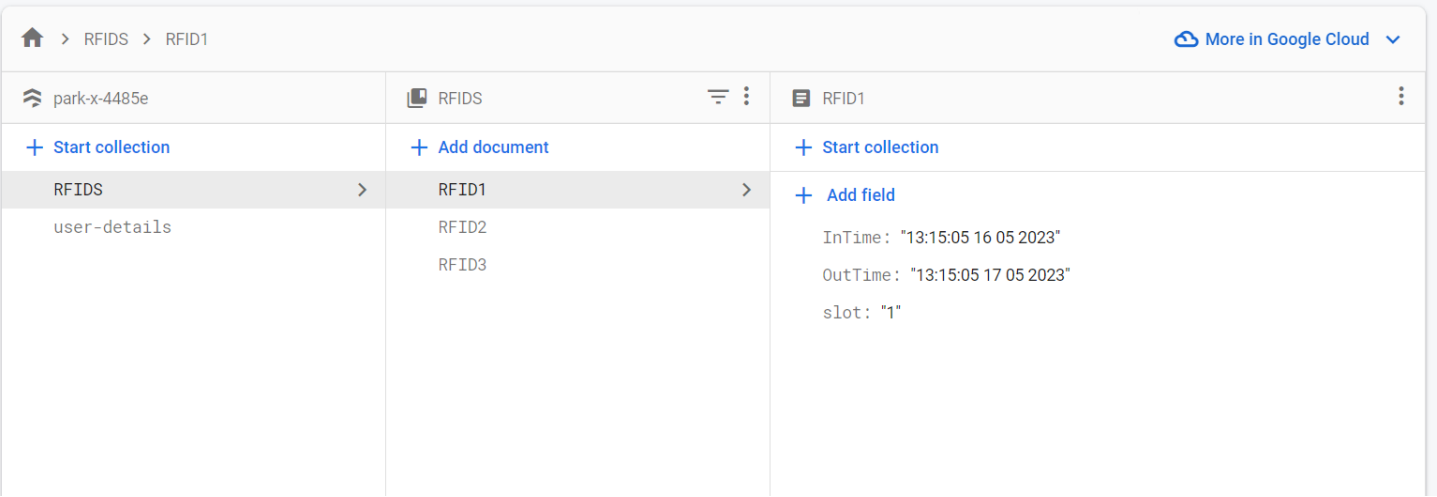
```
      if(count == 0)
      {
         Firebase.setString(tag + "/InTime", formattedTime +" " +  currentDate) ;
         count = 1;
         Serial.println("InTime :  " + formattedTime +" " +  currentDate);
      }
      else if(count == 1)
      {
         //set OutTime
         Firebase.setString(tag + "/OutTime", formattedTime +" " +  currentDate) ;
         Firebase.setString(tag + "/slot", "0");
         canCheck = 0 ;
         count = 0;
         Serial.println("OutTime :  " + formattedTime +" " +  currentDate);
      }

      rfid.PICC_HaltA();
      rfid.PCD_StopCrypto1();
    }

    if(count == 1 && canCheck == 0)
    {
       s1=digitalRead(aaa);
       if(s1 == 0 )
       {
         Firebase.setString(tag + "/slot", "1");
         canCheck == 1;
         Serial.println("Object Present");
       }
    }

    if (Firebase.failed())
    {
       Serial.print("setting /number failed:");
       Serial.println(Firebase.error());
       return;
    }
}
```

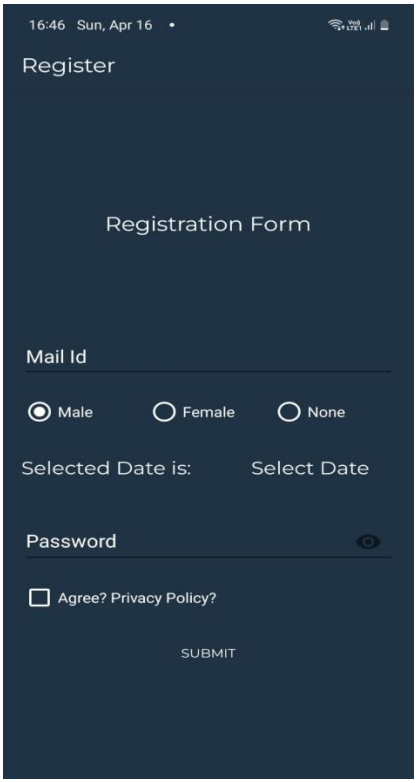
Firebase Database Schema:

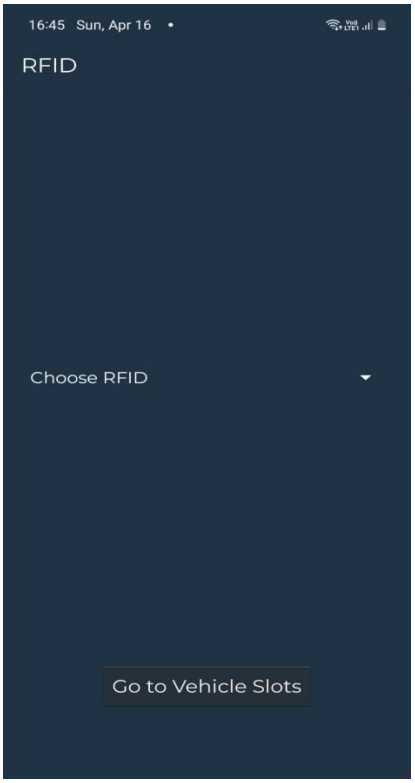


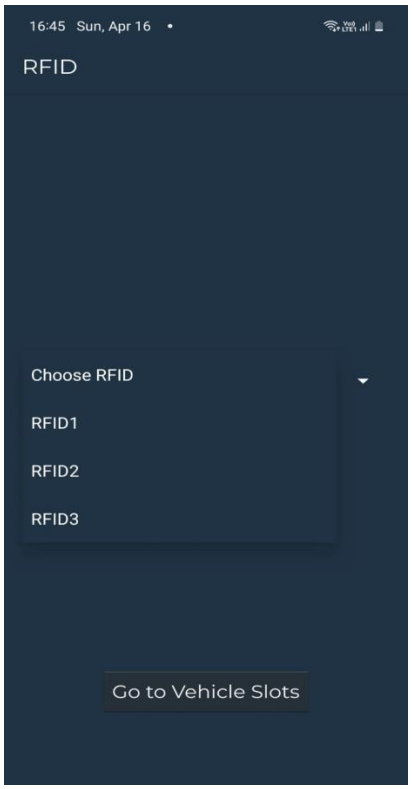
MOBILE APP Part:

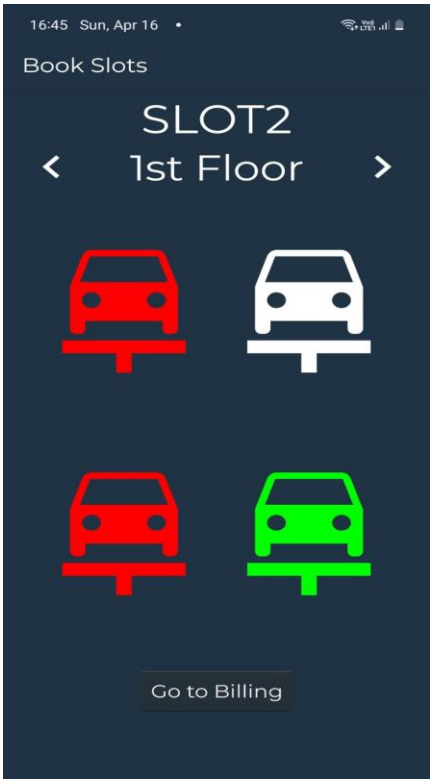
Code: https://github.com/TharunDharmaraj/Parking

Screenshots:

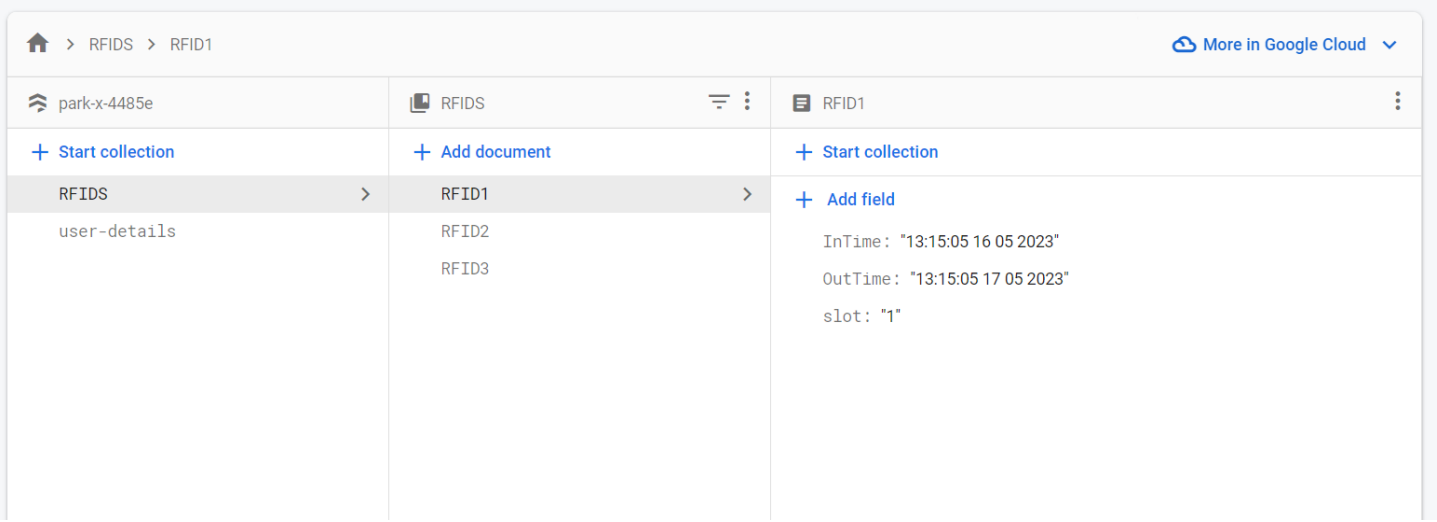


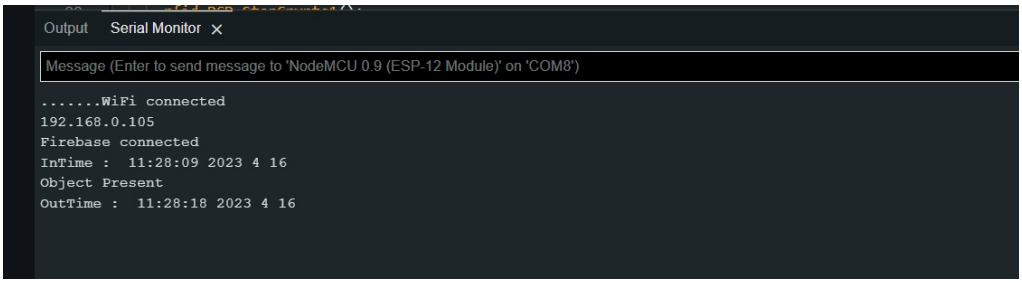


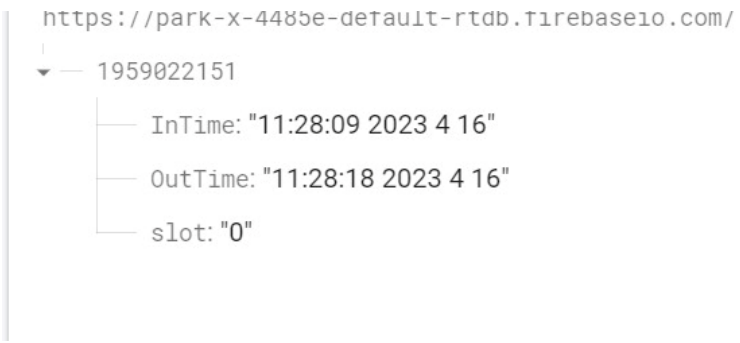


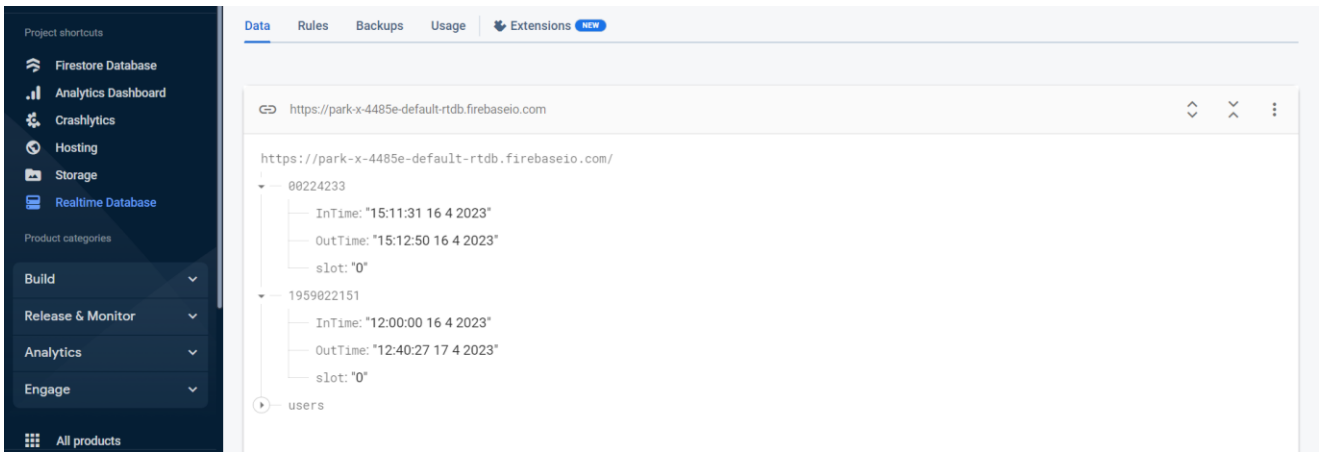


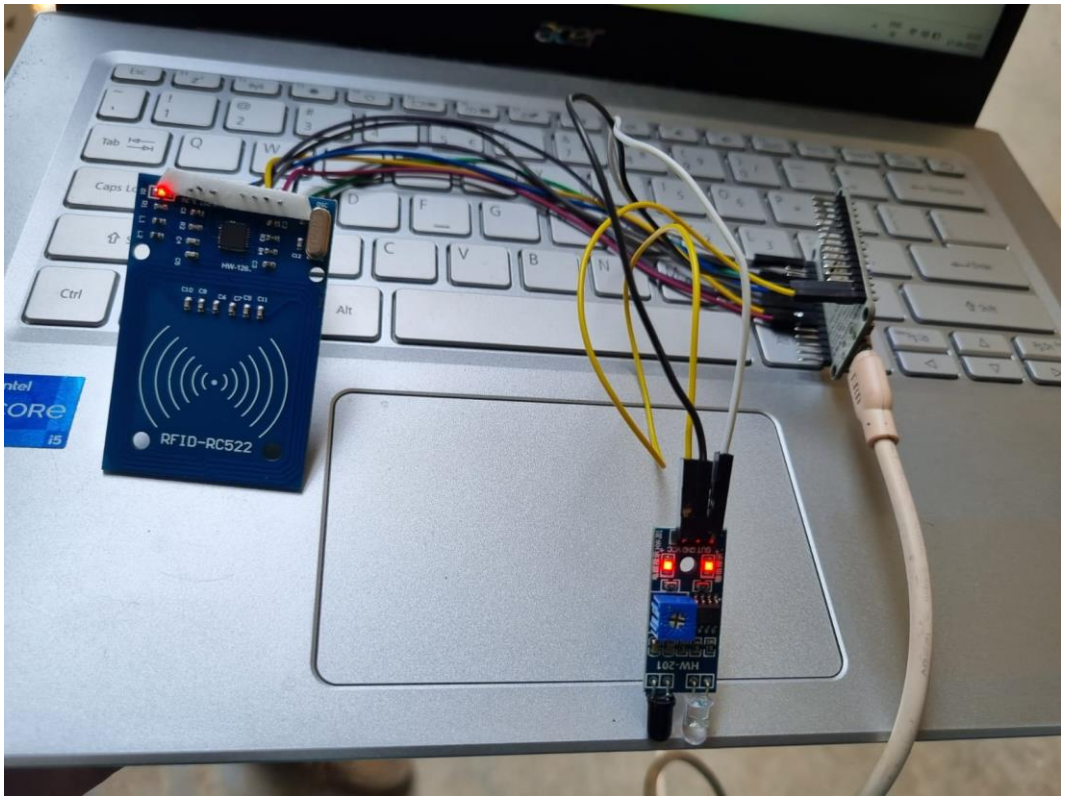
Output Screenshot:

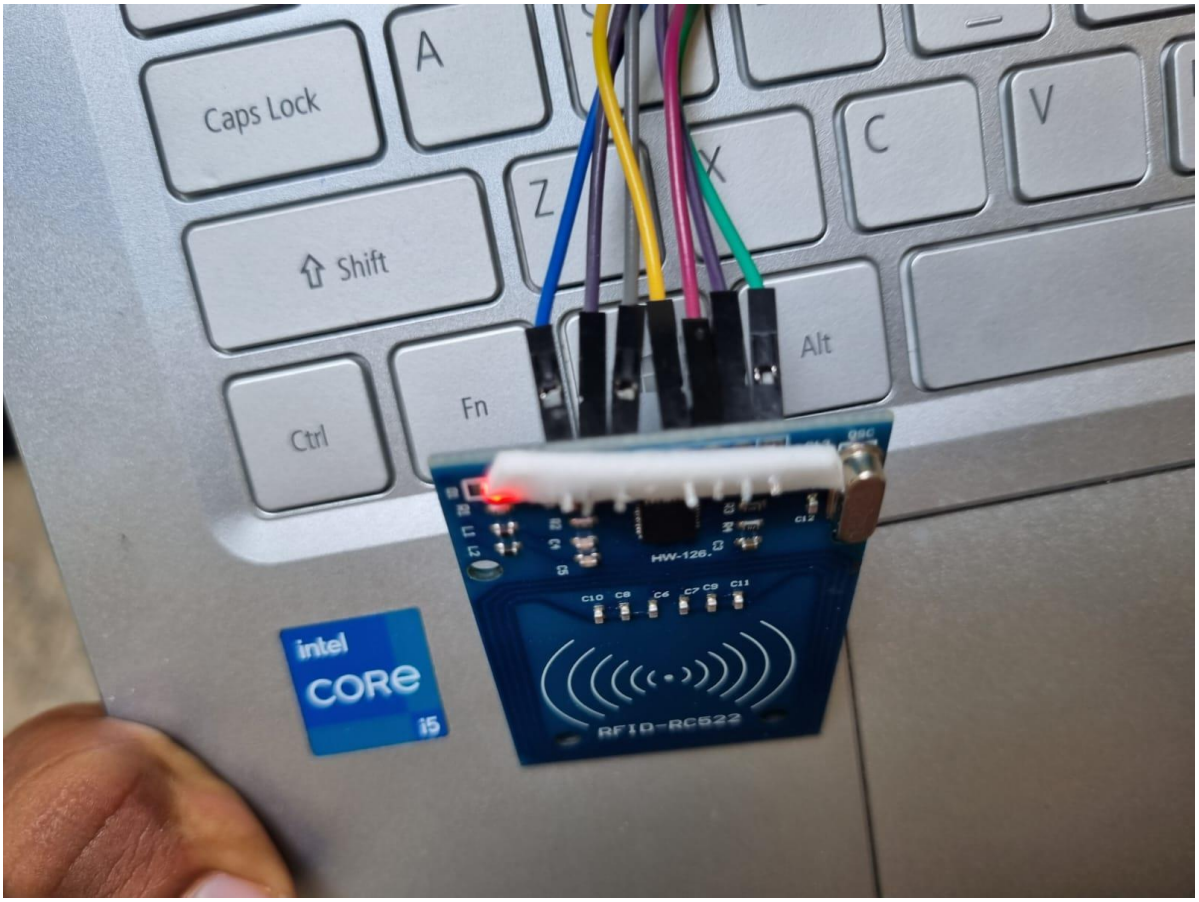


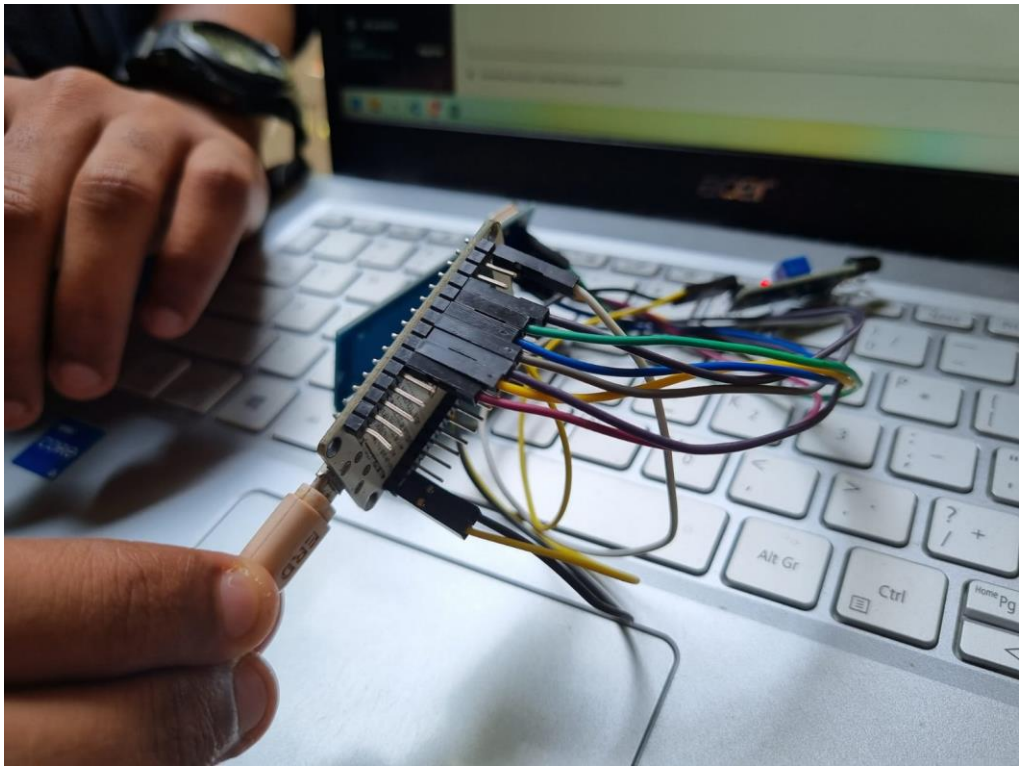


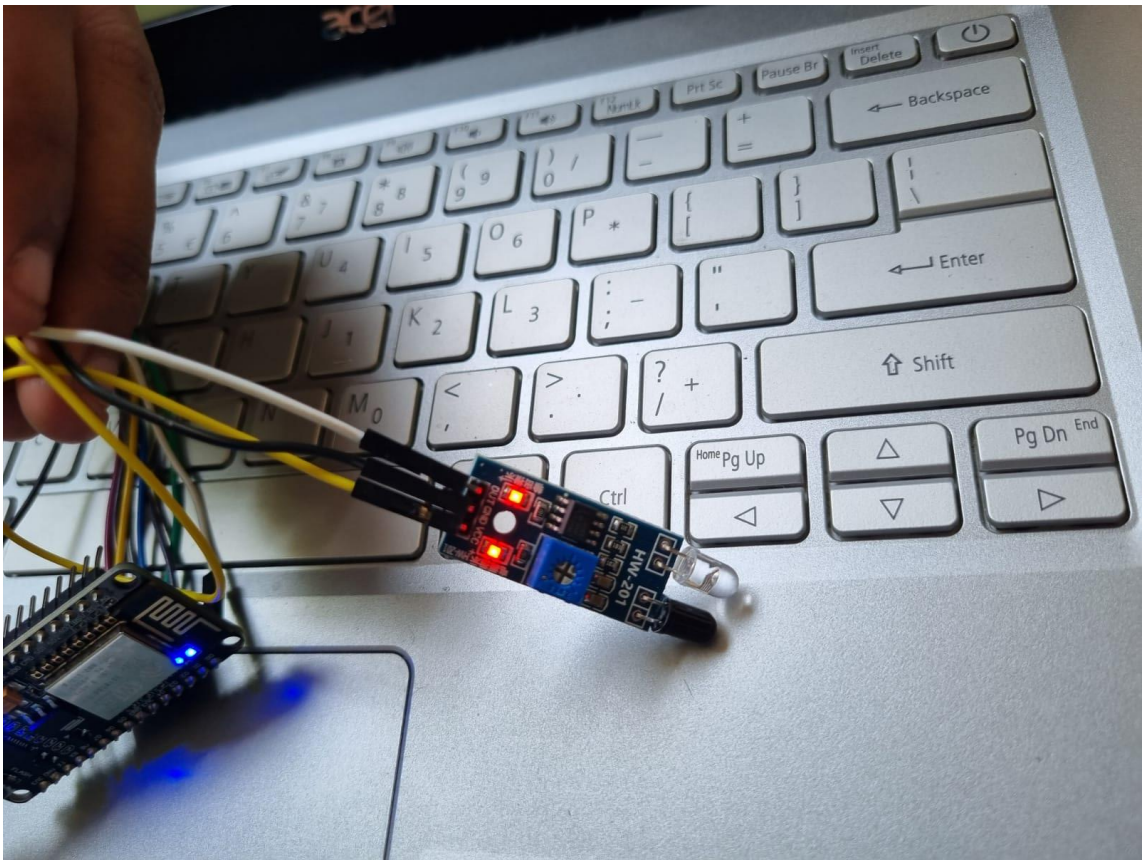


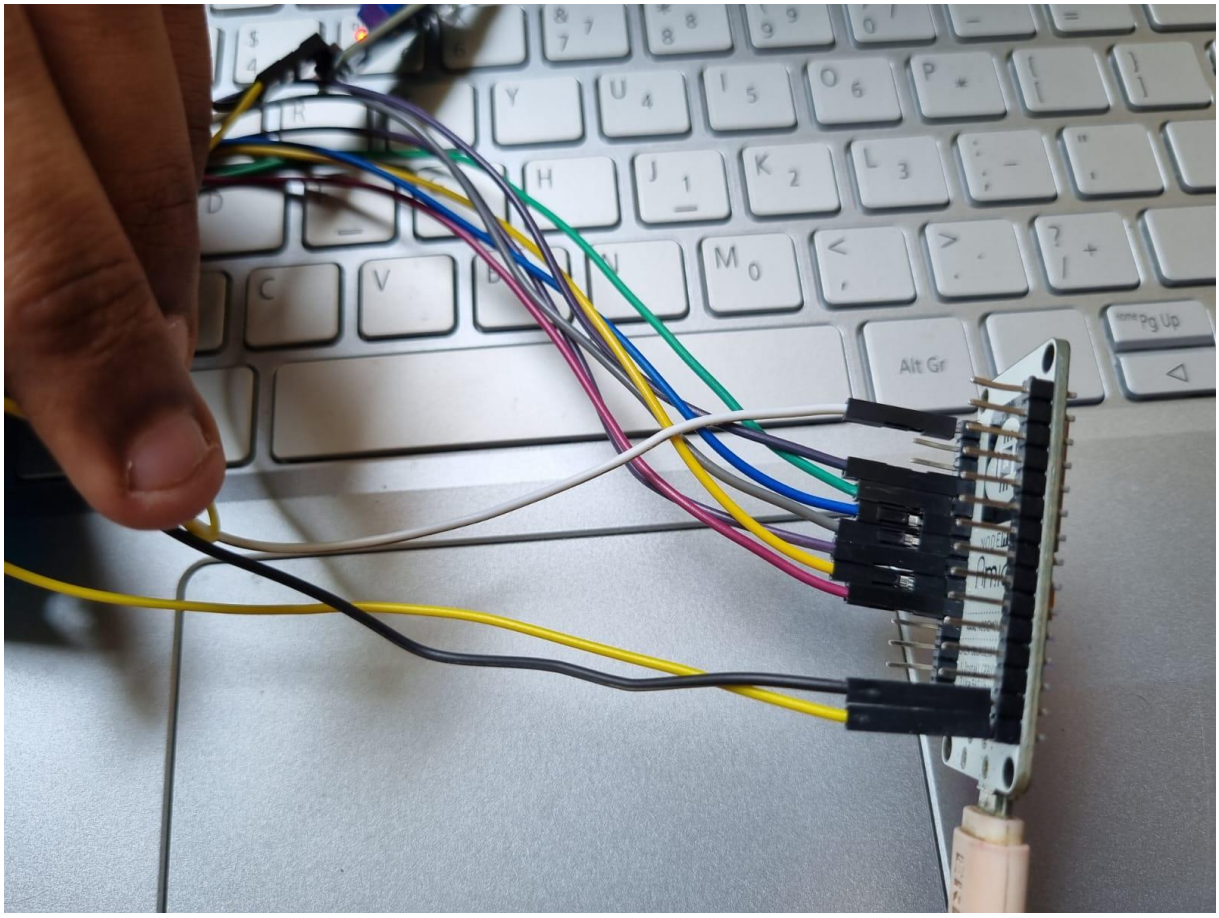



RFID-RC522
HW-201


Caps Lock
Shift
Ctrl
Fn
Alt
RFID-RC522
intel
CORE
i5


Alt Gr
Ctrl


Backspace
Enter
Shift
Ctrl
Pg Up
Pg Dn
HW-201



Conclusion:

In conclusion, the smart parking system using IoT and RFID technology provides a convenient and efficient solution to the problem of finding available parking spots. By utilizing RFID tags, readers, and a central server, the system can detect the occupancy status of parking spots and provide real-time information on parking availability to users through a mobile application. The use of NodeMCU board provides connectivity and allows the system to communicate with the internet and the mobile application. Overall, the system has the potential to revolutionize the parking experience and make it easier for drivers to find and reserve parking spots, reducing the time and effort required to park their vehicles.