***SMART PARKING SYSTEM***

**Name: Soumya Singh Reg.No.: 21BPS1612**

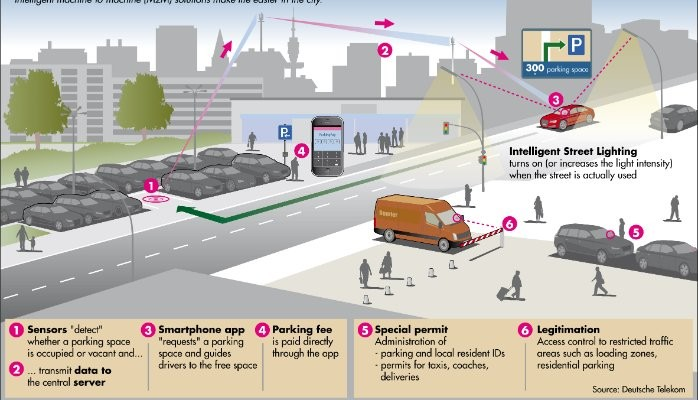
**Aim:** Demonstrate the Smart Parking System.

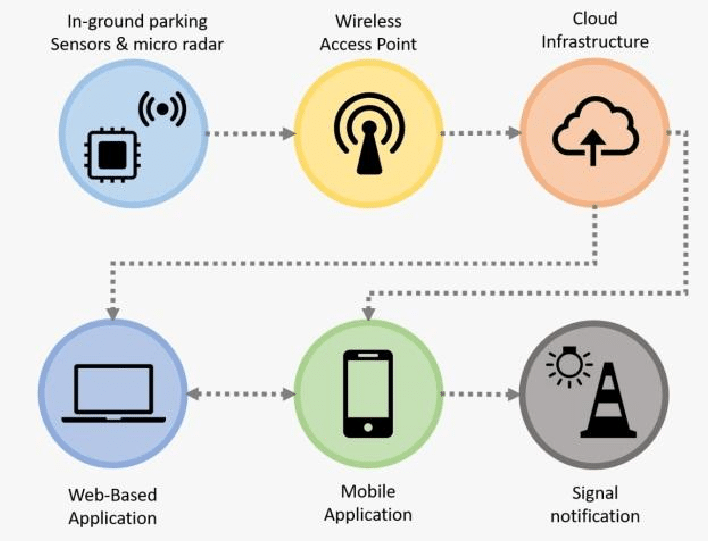
**Theory:**

Many large cities face the dire problem of providing available parking spaces to their citizens at the peak hours of a day. As a result, citizens spend a massive amount of time searching for the perfect parking space or waiting in line to get one. This, in turn, creates traffic congestion. Considering the problems, many researchers have suggested different SPS approaches and technologies to mitigate this problem.

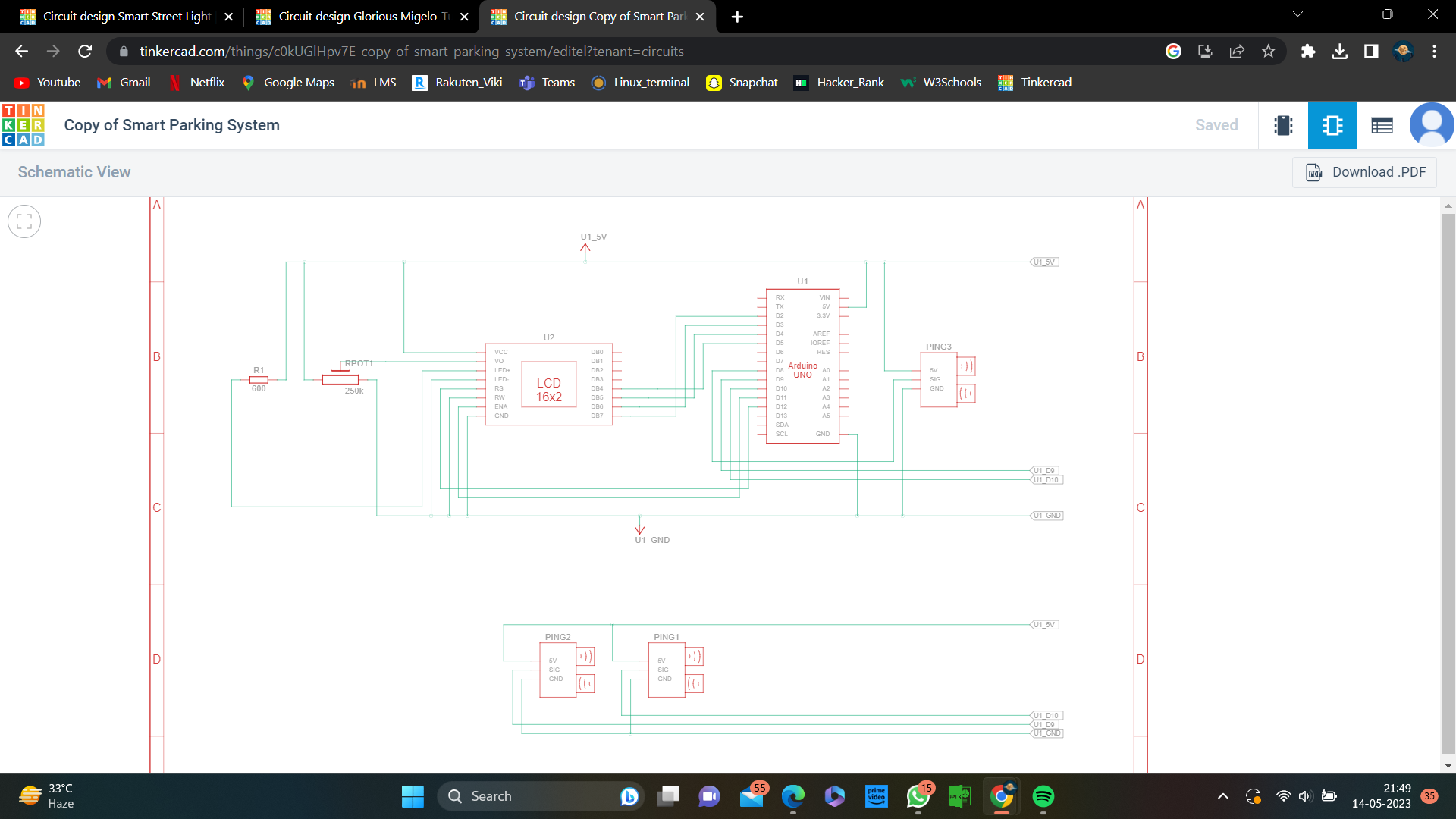
The authors suggested the adaptation of a cloud-based platform as a Service (PaaS) to develop an Internet of Things (IoT) based SPS. PaaS can be divided into two parts: a back-end dashboard platform and a front-end data platform. The back-end data platform provides data storage, management, and processing facility. On the other hand, the front-end dashboard platform deals with reporting and visualization of data.

A Multi-Agent System (MAS) based SPS has been designed. The system uses agent networks that coordinate between the driver and the SPS. It utilizes a negotiation algorithm that provides a negotiable parking fee depending upon different criteria. Moreover, the system offers vehicle guidance for the shortest path to the parking space and parking reservation facility.

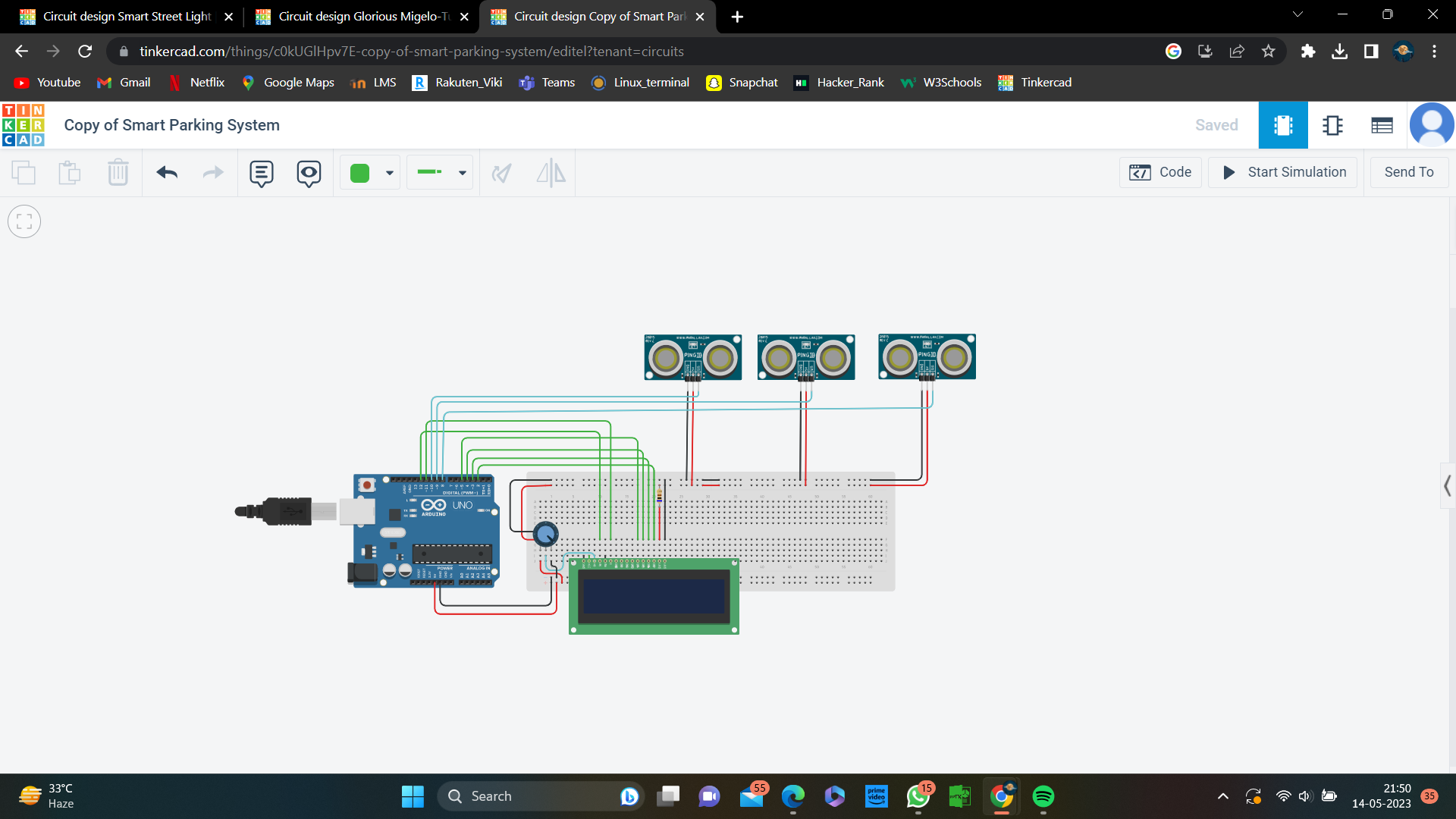




**Circuit:**



**Tinkercad Representation:**

****

**Components Used:**

| **Name** | **Quantity** | **Component** |
| --- | --- | --- |
| U1 | 1 | Arduino Uno R3 |
| U2 | 1 | LCD 16 X 2 |
| Rpot1 | 1 | 250 kΩ Potentiometer |
| R1 | 1 | 600 Ω Resistor |
| PING1  PING2  PING3 | 3 | Ultrasonic Distance Sensor |

**Code:**

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

#define t1 10

#define t2 9

#define t3 8

int distanceThreshold = 100;

void setup() {

lcd.begin(16,2);

lcd.setCursor(0,0);

Serial.begin (9600);

}

long readDistance(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 loop()

{

float d1 = 0.01723 \* readDistance(t1, t1);

float d2 = 0.01723 \* readDistance(t2, t2);

float d3 = 0.01723 \* readDistance(t3, t3);

Serial.println("d1 = " + String(d1) + "cm");

Serial.println("d2 = " + String(d2) + "cm");

Serial.println("d3 = " + String(d3) + "cm");

if (d1>100 & d2>100 & d3>100){

lcd.setCursor(0,0);

lcd.print("3 Slots Free");

lcd.setCursor(0,1);

lcd.print("Slot 1 2 3 Free");

delay(500);

}

else if((d1>100 & d2>100)|(d2>100 & d3>100)|(d3>100 & d1>100))

{

lcd.setCursor(0,0);

lcd.print("2 Slots Free");

lcd.setCursor(0,1);

if(d1>100 & d2>100)

lcd.print("Slot 1 & 2 Free");

else if(d1>100 & d3>100)

lcd.print("Slot 1 & 3 Free");

else

lcd.print("Slot 2 & 3 Free");

delay(500);

}

else if(d1<100 & d2<100 & d3<100)

{

lcd.setCursor(0,0);

lcd.print("No Slot Free");

lcd.setCursor(0,1);

lcd.print("Parking Full");

delay(500);

}

else if((d1<100 & d2<100)|(d2<100 & d3<100)|(d3<100 & d1<100))

{

lcd.setCursor(0,0);

lcd.print("1 Slot Free");

lcd.setCursor(0,1);

if(d1>100)

lcd.print("Slot 1 is Free");

else if (d2>100)

lcd.print("Slot 2 is Free");

else

lcd.print("Slot 3 is Free");

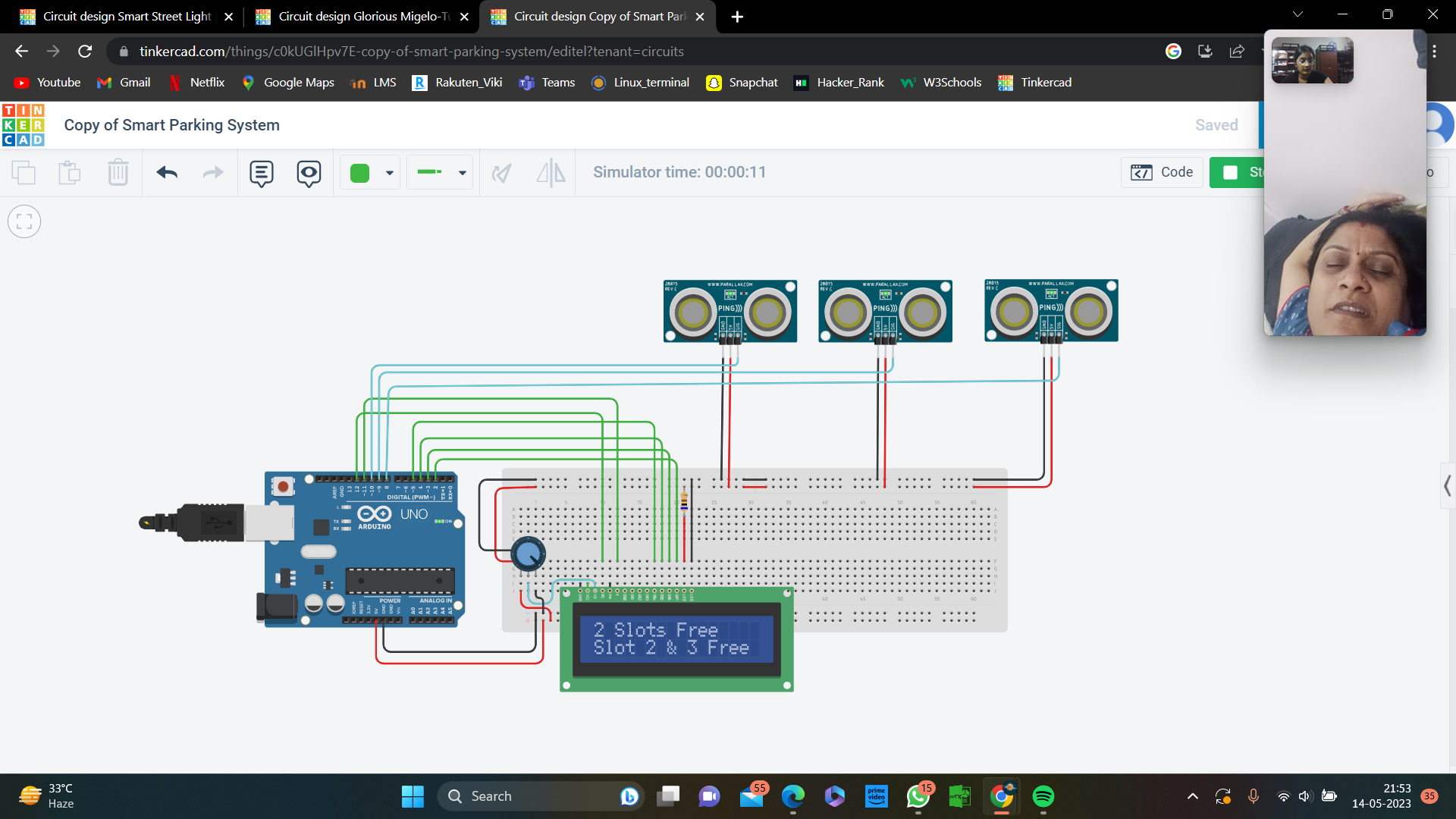
delay(500);

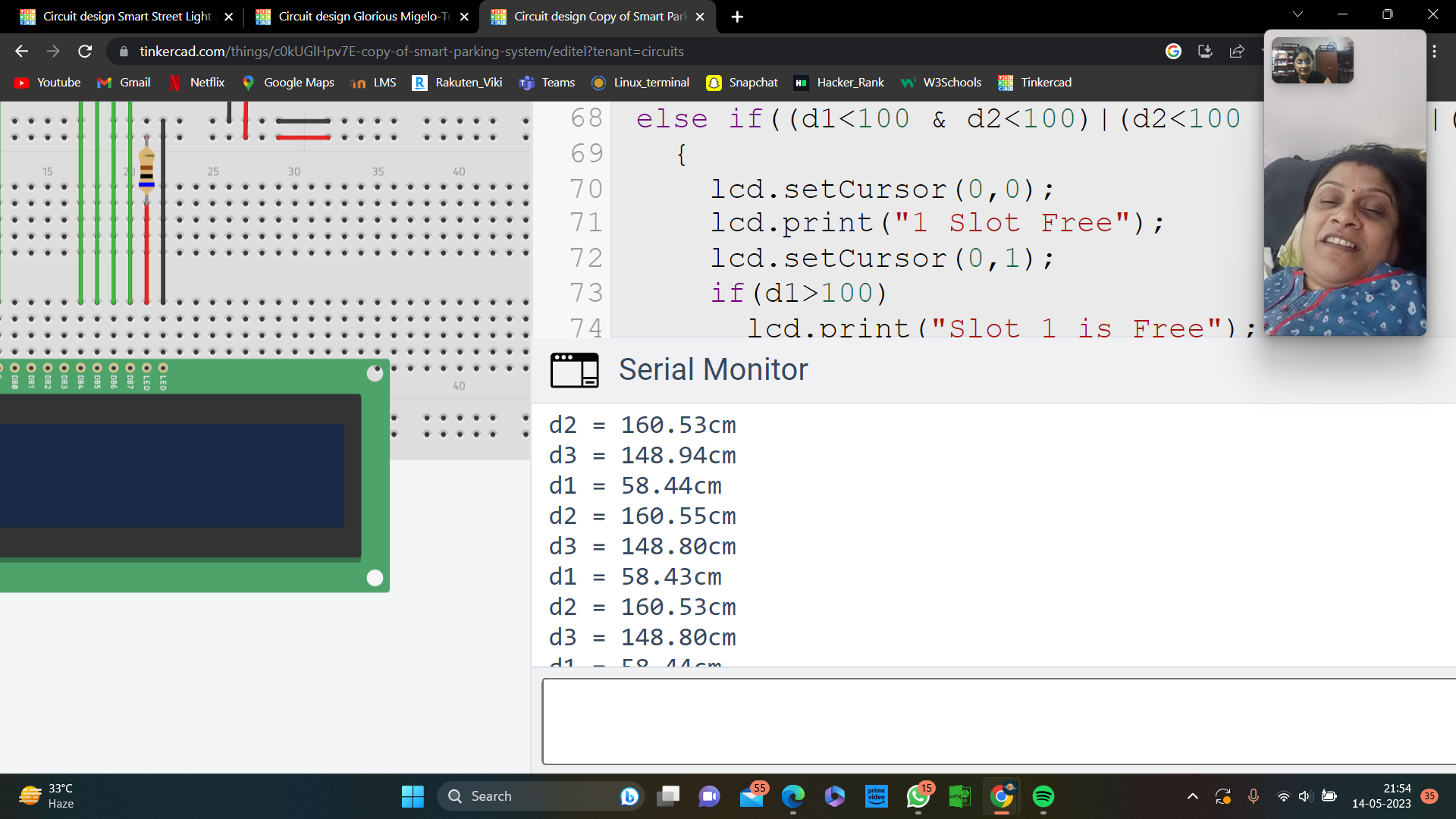
}

delay(100);

}

**Simulation Output:**

****

****

**Conclusion:-**

Due to the rapid increase in urban population and unplanned urbanization, there is a decrease in the number of urban parking spaces and an increment in traffic congestion. As a result, smart parking has become the subject of interest for both researchers and urban planners.