**NAAN MUDHALVAN**

**INTERNET OF THINGS- IBM**

**PROJECT – SMART PARKING USING IOT**

**PHASE 5**

**TEAM MEMBERS**

**1.E HEMALATHA 2021116008**

**2.V M LAKSHMI PRIYA 2021116021**

**3. C RASHNI PREETHI 2021116031**

**4.S SHYBI 2021116039**

**5.VAIBHAVADARINI 2021116049**

**5TH SEMESTER – 3RD YEAR**

**B.E.BIOMEDICAL ENGINEERING**

**COLLEGE OF ENGINEERING GUINDY**

**ANNA UNIVERSITY**

**SMART PARKING USING IOT**

**PHASE 1**

**DEFINING THE OBJECTIVE AND PROBLEM SOLUTION**

**PHASE 2**

**INNOVATION**

**PHASE 3**

**WOWKI SIMULATION USING SENSOR**

**PHASE 4**

**HTML WEB PAGE DEVELOPMENT**

**PHASE 5**

**DOCUMENTATION**

**SMART PARKING**

**PHASE-1**

**PROJECT DEFINITION:**

A "Smart Parking" project using the Internet of Things (IoT) can be defined as the implementation of a technology-driven system that aims to optimize the management and utilization of parking spaces in urban or crowded areas. The primary goal is to enhance the overall parking experience for both drivers and parking lot operators while reducing congestion, fuel consumption, and environmental impact.

**PROJECT OBJECTIVES:**

**1.Real-time Parking Space Monitoring**:

Implement sensors (e.g., ultrasonic, infrared, or cameras) to monitor parking space occupancy in real-time. Provide accurate, up-to-date information on available parking spaces to drivers and parking operators.

**2.Optimized Space Utilization**:

Utilize data analytics and algorithms to optimize parking space allocation and reduce wasted space. Maximize revenue for parking operators by efficiently using available parking spots.

**3.Reduced Traffic Congestion:**

Reduce traffic congestion and air pollution by guiding drivers directly to available parking spaces. Minimize the time spent circling for parking, which can lead to lower fuel consumption and emissions.

**4.Enhanced User Experience**:

Develop user-friendly mobile apps or web interfaces that allow drivers to easily find, reserve, and pay for parking spaces. real timeProvide navigation and wayfinding assistance to guide users to their chosen parking spots.

**5.Data Analytics and Insights:**

Collect and analyze data on parking usage, trends, and patterns. Use insights to make data-driven decisions, improve operations, and plan for future expansion.

**PROJECT DESIGNING:**

**1.Real-time Parking Space Monitoring:**

Objective: To monitor the availability of parking spaces in real-time and provide accurate information to drivers.

Key Components:

Sensor Network: Deploy sensors (e.g., ultrasonic or infrared) in each parking space to detect occupancy.

Data Processing: Collect and process sensor data to determine parking space availability.

Real-time Updates: Ensure the system updates parking availability information in real-time.

**2.Mobile App Integration:**

Objective: To create a mobile application that seamlessly integrates with the IoT-based parking system.

Key Features:

Parking Space Reservation: Allow users to reserve parking spaces in advance.

Navigation and Directions: Provide users with directions to the nearest available parking space.

Payment Integration: Enable users to pay for parking through the app.

Push Notifications: Send notifications about parking availability and reservations.

**3.Efficient Parking Guidance:**

Objective: To guide drivers efficiently to available parking spaces within the parking facility.

Key Components:

Wayfinding System: Implement digital signage, LED indicators, or mobile app navigation to guide drivers to vacant parking spots.

Algorithms: Utilize algorithms to optimize parking space allocation based on occupancy patterns.

Data Analytics: Analyze historical parking data to predict peak usage times and plan for traffic flow.

By defining these specific objectives, you can create a clear roadmap for your IoT-based parking project, ensuring that it addresses the key challenges and provides valuable features to both parking operators and users.

**PHASE-2**

**Innovation proposal: IoT-Based Smart Parking**

**Introduction**

The rapid urbanization and increasing vehicle ownership have led to a pressing issue: finding parking spaces efficiently in congested urban areas. IoT-based Smart Parking systems offer a viable solution by leveraging technology to optimize parking space utilization, reduce traffic congestion, and enhance the overall urban mobility experience. This proposal outlines an innovative IoT-based Smart Parking solution that addresses these challenges and provides significant benefits to both cities and motorists.

**Objectives**

The primary objectives of this IoT-based Smart Parking innovation proposal are as follows:

* Improve urban mobility and reduce traffic congestion.
* Optimize the utilization of parking spaces, reducing the time spent searching for parking.
* Enhance the convenience of parking for motorists.
* Enable cities to better manage and monetize their parking assets.
* Reduce the environmental impact of vehicles searching for parking.

**Key Components of the Proposal**

**3.1. IoT Sensors and Infrastructure**

Install IoT sensors in parking spaces to detect occupancy and transmit data to a central server. The IoT infrastructure should include:

Ultrasonic sensors or cameras to detect vehicle presence.

Low-power, long-range communication protocols (e.g., LoRaWAN or NB-IoT) for efficient data transmission.

Edge computing devices for real-time data processing.

A centralized cloud-based server for data storage and analysis.

**3.2. Mobile Application**

Develop a user-friendly mobile application for motorists to access real-time parking information and features such as:

Find available parking spaces in their vicinity.

Reserve parking spots in advance.

Pay for parking digitally.

Receive navigation assistance to the selected parking spot.

Set up notifications for parking availability.

**3.3. Data Analytics and Predictive Algorithms**

Implement data analytics and predictive algorithms to:

Analyze historical parking data to identify trends and patterns.

Predict future parking availability based on historical and real-time data.

Optimize parking space allocation and pricing dynamically.

**3.4. Smart Payment Solutions**

Integrate secure and convenient payment methods within the mobile application, including:

Contactless payments via mobile wallets or credit cards.

Payment discounts for off-peak hours or sharing rides to parking spots.

Subscription-based payment models for frequent users.

**3.5. Integration with Urban Infrastructure**

Collaborate with local authorities to integrate the IoT-based Smart Parking system with existing urban infrastructure:

Traffic management systems to reduce congestion by guiding vehicles to available parking spots.

Public transportation systems to promote multi-modal transportation.

Environmental monitoring systems to measure the reduction in emissions from reduced parking search times.

**Conclusion**

This IoT-based Smart Parking innovation proposal presents an opportunity to address the growing challenges of urban mobility and parking congestion. By leveraging IoT technology, data analytics, and user-friendly interfaces, cities can improve the parking experience for motorists while optimizing the utilization of parking spaces and reducing traffic congestion. This innovation aligns with the goals of creating smart, sustainable, and efficient urban environments for the future.

**PHASE 3**

**SENSOR DESIGN SIMULATION USING WOKWI**

**INTRODUCTION:**

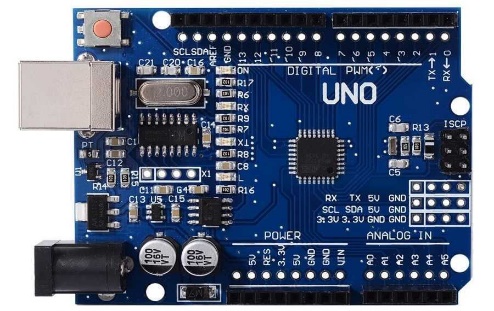
In phase 3, we have interfaced the ultrasonic sensor(HC-SR04) with the Arduino Uno. The simulation of the circuit was done using WOKWI simulator. Our project primarily focuses on techniques of smart parking and detection of distances between two cars and if there is enough distance available to park your cars. The output is displayed using LED display.

**REQUIREMENTS:**

Creating a complete smart parking system in an online simulation platform like Wokwi requires multiple components, including sensors, a microcontroller (such as an Arduino), and a way to visualize and control the system.

* Arduino Uno:

[**Arduino UNO**](https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html) is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output.



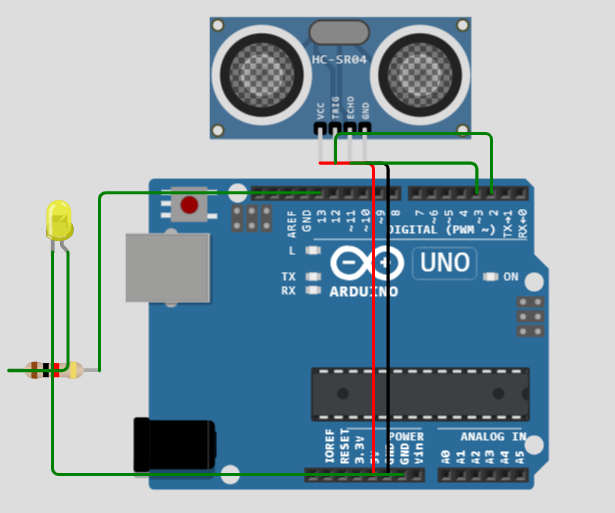
* Ultrasonic sensor (HC-SR04):

The HC-SR04 sensor is a distance sensor.This economical sensor provides 2cm to 400cm of non-contact measurement functionality with a ranging accuracy that can reach up to 3mm. Each HC-SR04 module includes an ultrasonic transmitter, a receiver and a control circuit.

There are only four pins that you need to worry about on the HC-SR04: VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground).

* LEDs (to represent parking spots)
* Breadboard and wires
* Resistor : 1kΩ

**CIRCUIT DESIGN:**



**PROGRAM:**

#define TRIGGER\_PIN 2

#define ECHO\_PIN 3

#define LED\_PIN 13

void setup() {

  pinMode(TRIGGER\_PIN, OUTPUT);

  pinMode(ECHO\_PIN, INPUT);

  pinMode(LED\_PIN, OUTPUT);

**Serial**.begin(9600);

}

void loop() {

  digitalWrite(TRIGGER\_PIN, LOW);

  delayMicroseconds(2);

  digitalWrite(TRIGGER\_PIN, HIGH);

  delayMicroseconds(10);

  digitalWrite(TRIGGER\_PIN, LOW);

  long duration = pulseIn(ECHO\_PIN, HIGH);

  int distance = duration / 58.2;

**Serial**.print("Distance: ");

**Serial**.print(distance);

**Serial**.println(" cm");

  if (distance < 10) {

    digitalWrite(LED\_PIN, HIGH);

**Serial**.println("Parking spot occupied");

  } else {

    digitalWrite(LED\_PIN, LOW);

**Serial**.println("Parking spot available");

  }

  delay(1000);

}

**EXPLANATION:**

In this code, we use an ultrasonic sensor to detect the presence of a vehicle in each parking spot. The Arduino controls LEDs that represent the status of each parking spot (occupied or empty). The status is displayed in the serial monitor.

**1. Variable and Constant Definitions:**

- `TRIGGER\_PIN` and `ECHO\_PIN` are defined to specify the Arduino pins connected to the ultrasonic sensor's trigger and echo pins, respectively.

- `numParkingSpots` is set to 4, indicating that there are four parking spots to monitor.

- An array `isOccupied` is defined to keep track of the status of each parking spot. Initially, all spots are assumed to be empty (`false`).

**2. Setup Function:**

- `Serial.begin(9600)` initializes the serial communication for debugging, allowing you to view the status of parking spots on the serial monitor.

- `pinMode(TRIGGER\_PIN, OUTPUT)` and `pinMode(ECHO\_PIN, INPUT)` set the trigger pin as an output and the echo pin as an input for the ultrasonic sensor.

- A `for` loop is used to set pins 2, 3, 4, and 5 as OUTPUT. These pins are connected to LEDs that represent the parking spot status.

**3. Loop Function:**

- The `loop` function is where the continuous monitoring of parking spots takes place.

- A `for` loop iterates through each parking spot (from 0 to 3).

- An ultrasonic sensor is used to measure the distance to an object in front of it. The steps include triggering a pulse, measuring the time it takes for the pulse to bounce back (echo), and converting it into a distance in centimeters.

- If the measured distance is less than 10 centimeters (you can adjust this threshold), it's considered that a vehicle is occupying the spot, and `isOccupied[spot]` is set to `true`. Otherwise, the spot is considered empty (`false`).

- LEDs connected to pins 2, 3, 4, and 5 are updated to indicate the status of each parking spot (lit if occupied, off if empty).

**4. Status Printing:**

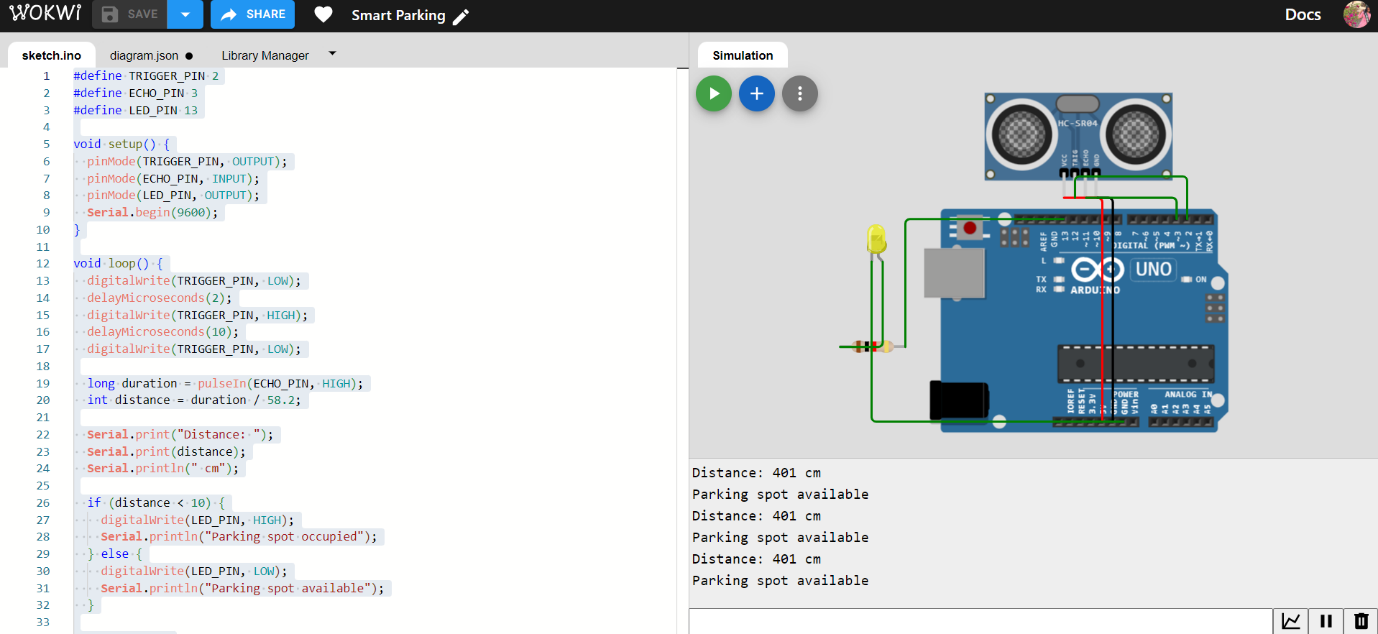
- After checking the status of all parking spots, a second `for` loop is used to print the status of each spot on the serial monitor. It will display the spot number and whether it is "Occupied" or "Empty."

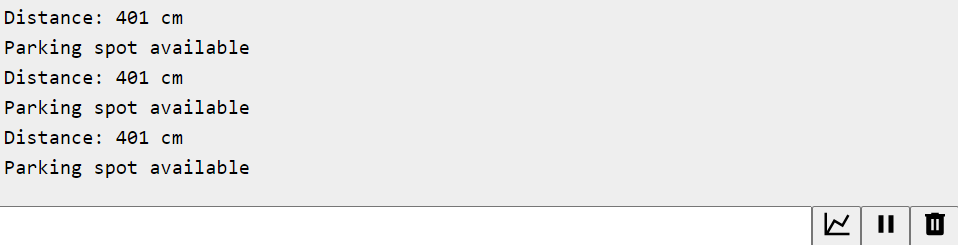
**5. Delay:**

- A delay of 1000 milliseconds (1 second) is added to prevent constant readings and updates. This provides a more manageable output on the serial monitor and avoids rapid changes in the LED status.

This code continuously monitors the distance in front of the ultrasonic sensor for each parking spot and updates the status of the parking spots (occupied or empty) on the LEDs and in the serial monitor. This is a basic example of a smart parking system.

**OUTPUT:**

****

****

**PHASE-4 WEB DEVELOPMENT**

In this phase, we have developed a embedded HTML page of our Wowki simulation, which would show whether the parking lot is available or not with a distance.

**HTML CODE:**

html

<!DOCTYPE html>

<html>

<head>

<title>Wokwi App Embed</title>

</head>

<body>

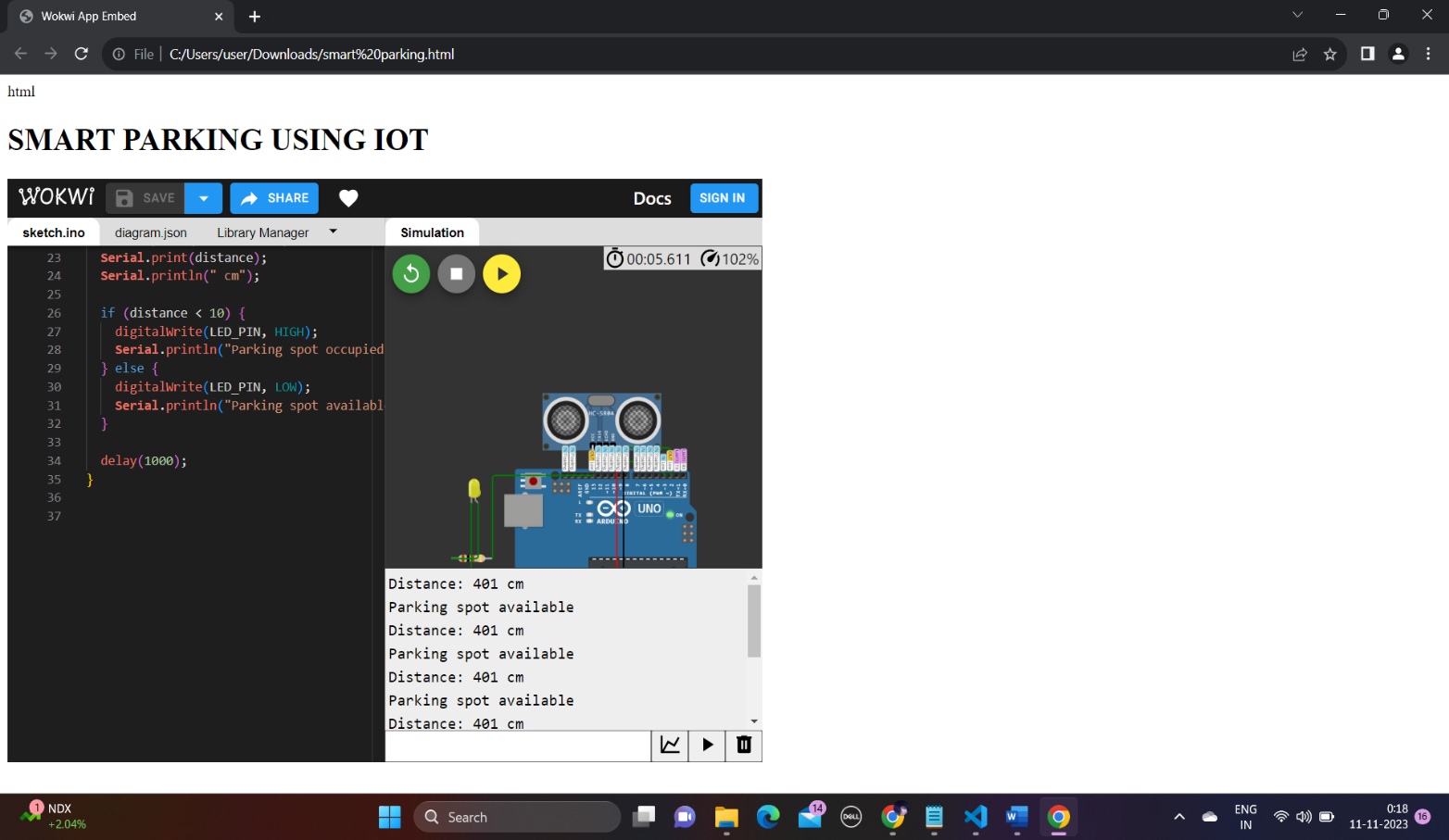
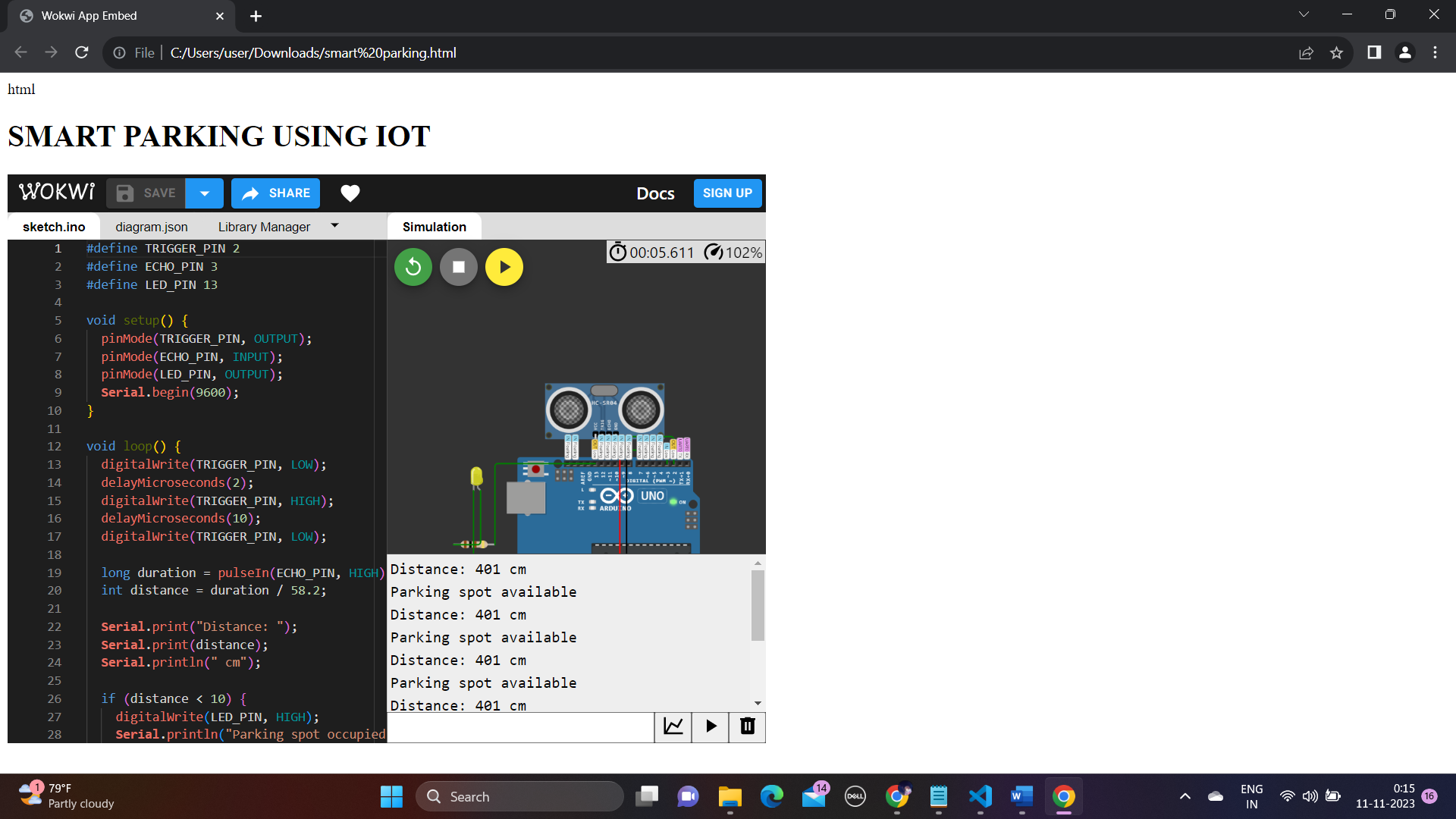
<h1>SMART PARKING USING IOT</h1>

<!-- Replace the URL in the src attribute with the URL of your Wokwi app -->

<iframe src="https://wokwi.com/projects/380023077302077441" width="800" height="600" frameborder="0"></iframe>

</body>

</html>

**WEBPAGE:**

