**SMART PARKING USING INTERNET OF THING (IOT)**

**USING PYTHON SCRIPT**

**INTRODUCTION**:

\*The proposed IoT-based Smart Parking System leverages a network of sensors and communication devices to monitor and manage parking spaces in real-time. These sensors are strategically deployed in parking areas, and they continuously collect data related to parking space occupancy and availability. This data is then transmitted to a central cloud-based platform for analysis and dissemination.

**IDEAS** :

• A sensor that can detect the presence of the vehicle.

• A micro control that can help you processing the data.

• A cloud platform will restore the data.

• A mobile application enables you to control the smart parking process.

**SENSORS USED IN SMART PARKING SYSTEM**:

Raspberry Pi

 The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python.

Ultrasonic Sensors

The integration of ultrasonic sensors, which measure the distance between objects using ultrasonic waves, allows for precise parking. However, such devices have one drawback — the sensor might get blocked with dirt.

Electromagnetic Field Sensors

An electromagnetic field sensor detects and measures changes in the magnetic field. This way, it reacts to the approach of metal objects.

Infrared Sensors

Infrared (IR) sensors emit an infrared signal and catch the reflection of this signal from the environment. Integrating such a device enables measuring the temperature

**Python script on Raspberry Pi:**

import machine

import time

import urequests

import network

# WiFi credentials

WIFI\_SSID = "Wokwi-GUEST"

WIFI\_PASSWORD = ""

# ThingSpeak API settings

THINGSPEAK\_API\_KEY = "Y9M5L1HHUIBOYQXX"

THINGSPEAK\_CHANNEL\_ID = 2320595

# Define the GPIO pins for the sensor

TRIG\_PIN = machine.Pin(2, machine.Pin.OUT)

ECHO\_PIN = machine.Pin(3, machine.Pin.IN)

# Set the occupancy threshold in centimeters

OCCUPANCY\_THRESHOLD = 30.0 # Adjust this value as needed

# Connect to WiFi

wifi = network.WLAN(network.STA\_IF)

if not wifi.isconnected():

print("Connecting to WiFi...")

wifi.active(True)

wifi.connect(WIFI\_SSID, WIFI\_PASSWORD)

while not wifi.isconnected():

pass

print("Connected to WiFi")

def send\_to\_thingspeak(occupancy):

url = "https://api.thingspeak.com/update?api\_key={}&field1={}".format(THINGSPEAK\_API\_KEY, occupancy)

response = urequests.get(url)

response.close()

try:

while True:

# Trigger the ultrasonic sensor

TRIG\_PIN.on()

time.sleep\_us(10)

TRIG\_PIN.off()

pulse\_start = time.ticks\_us()

pulse\_end = time.ticks\_us()

# Wait for the echo signal

while ECHO\_PIN.value() == 0:

pulse\_start = time.ticks\_us()

while ECHO\_PIN.value() == 1:

pulse\_end = time.ticks\_us()

# Calculate the distance

pulse\_duration = time.ticks\_diff(pulse\_end, pulse\_start)

distance = (pulse\_duration / 1000000) \* 17150 # Speed of sound in cm/s

# Check for occupancy

if distance < OCCUPANCY\_THRESHOLD:

srinaath = "Occupied"

print(srinaath)

send\_to\_thingspeak(1) # Send 1 to indicate occupancy to ThingSpeak

else:

srinaath1 = "Not Occupied"

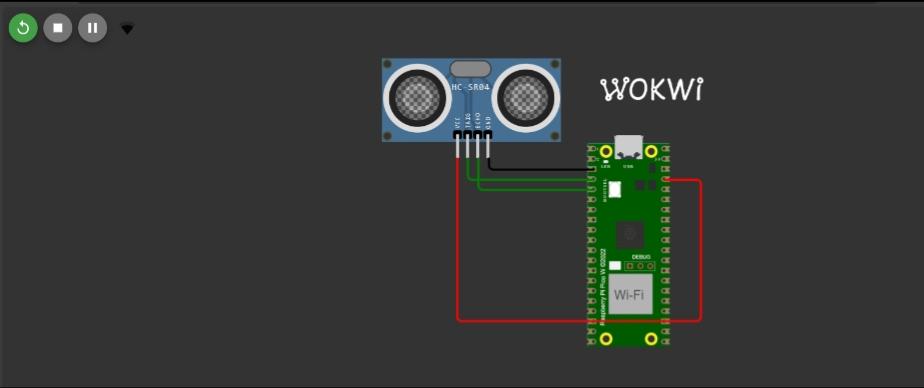
print(srinaath1)

send\_to\_thingspeak(0) # Send 0 to indicate no occupancy to ThingSpeak

time.sleep(5) # Delay for 15 seconds (ThingSpeak allows up to 15-second updates)

except KeyboardInterrupt:

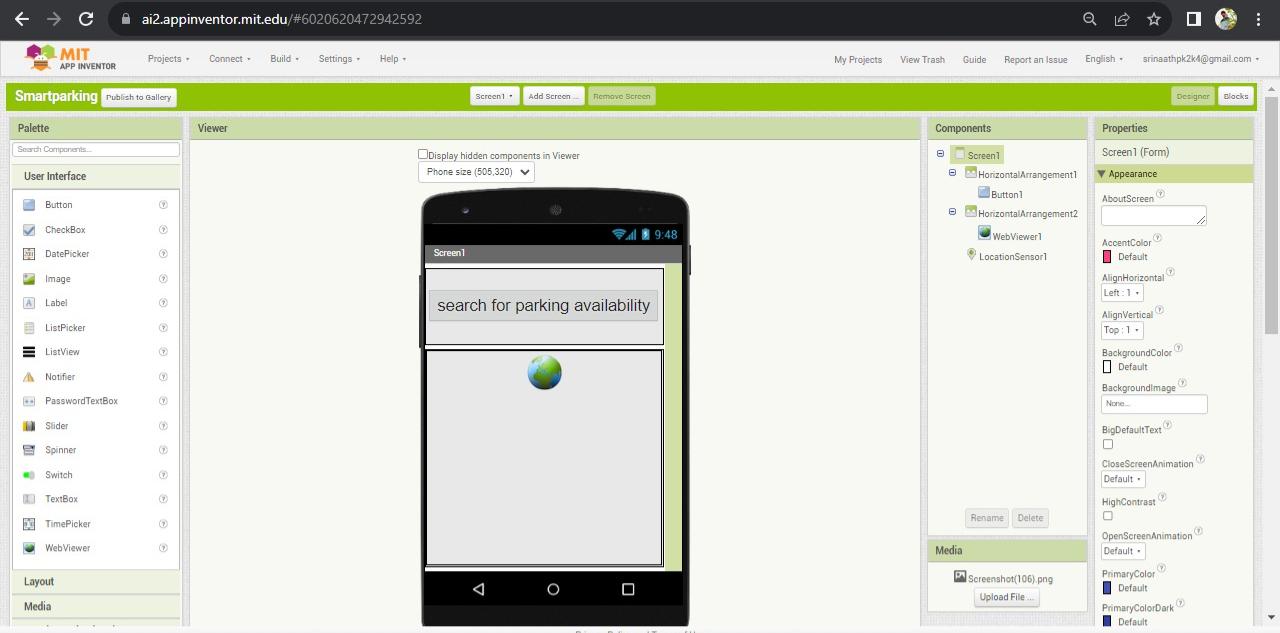
pass

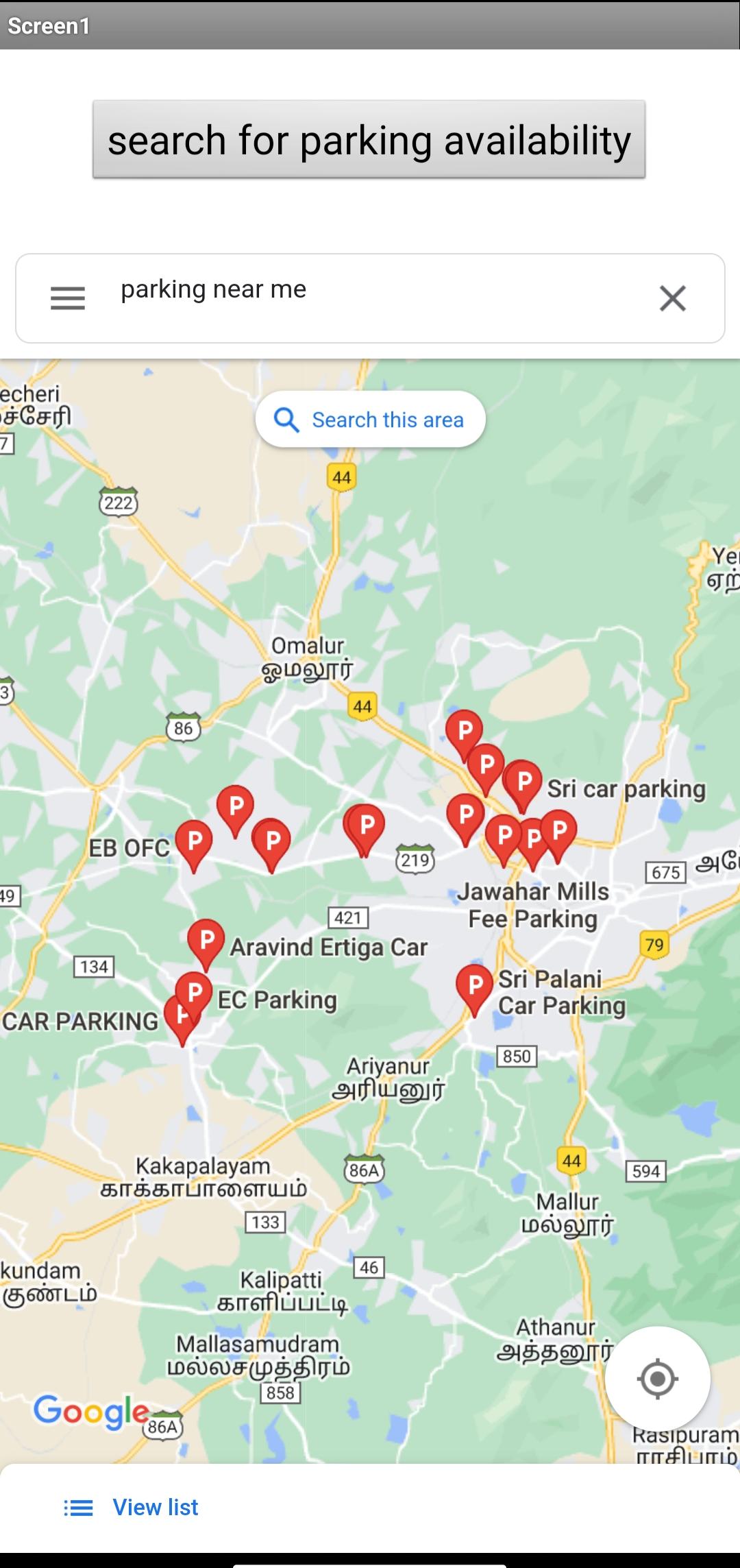
**Simulation circuit** :

 **In ThingSpeak**:

**Mobile apk link:**

<https://ai2.appinventor.mit.edu/b/4hdb8>

**Mobile app design using MIT:**

**Mobile app apk output:**

**GitHub link:**

https://github.com/srinaathpk/smartparking\_phase1.git

**APPLICATION OF SMART PARKING SYSTEM USING IOT:**

An IoT-based smart parking system is a decent solution for businesses and consumers,

Providing real-time data on parking space availability, pricing, payments, and more. It can positively impact the environment and traffic.

Moreover, IoT solutions ensure efficient parking reservation and management.