SMART PARKING

Phase 5: PROJECT DOCUMENTATION & SUBMISSION

- ★ **To Describe the Projects Objectives, IOT Sensor Setup, Mobile App Development, Raspberry Pi Integration and Code Implementation.**
- ★ **To Include Diagrams, Schematics, and Screenshots Of the IOT Sensors and Mobile App.**
- ★ **To Explain How the Real-Time Parking Availability System Can Benefit Drivers and Alleviate Parking Issues.**

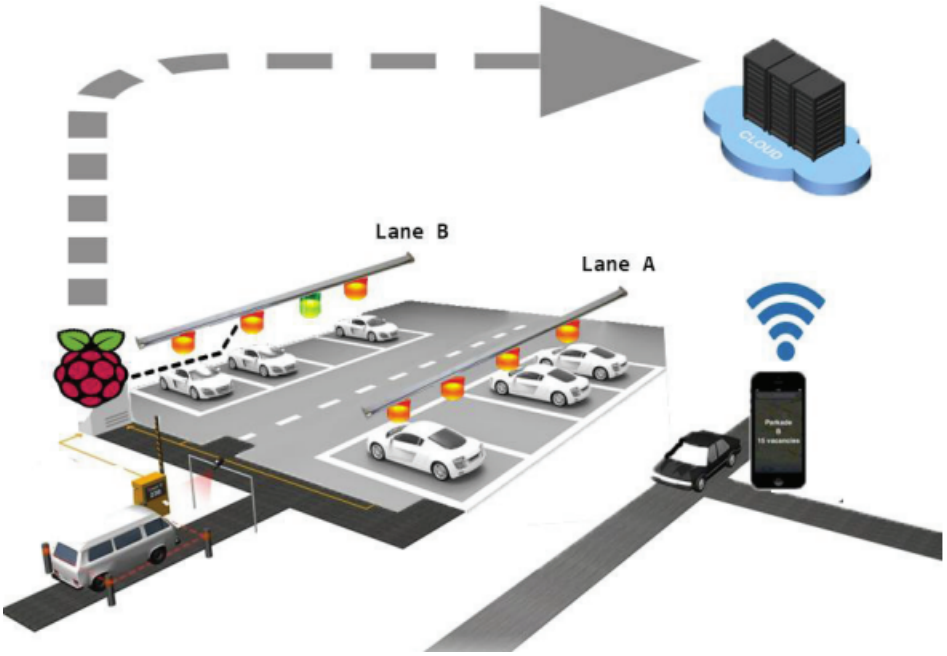
Objective:

To Build a Smart Parking System to Solve The Parking Issues in a Area Using IOT Sensor and Raspberry Pi.

Definition:

- **Smart Parking Refers to an Intelligent Parking Management System that Uses Advanced Technologies, Such as Internet of Things (IoT), Sensors, Data Analytics and Communication Networks.**
- **To Optimise Parking Space Utilisation and Enhance the Overall Parking Experience For Both Operators and Users.**
- **The Primary Goal is to Improve Efficiency, Reduce Congestion, Save Time, and Make Parking In Urban Areas More Convenient and Sustainable.**

Out View Of Working:



Need For Smart Parking:

- **Since The Population Of The World Increases Daily, Vehicle Users Also Increase.**
- **Thus To Park The Vehicle, We Need To Manage The Available Space For Them Using A "Smart Parking" System.**

Advantages:

- **Efficient Space Utilization.**
- **Real-Time Parking Availability Information.**
- **Reduced Traffic Congestion and Emissions.**
- **Cost Savings and Revenue Generation.**
- **Payment Automation and Flexibility.**
- **Integration with Smart City Initiatives.**

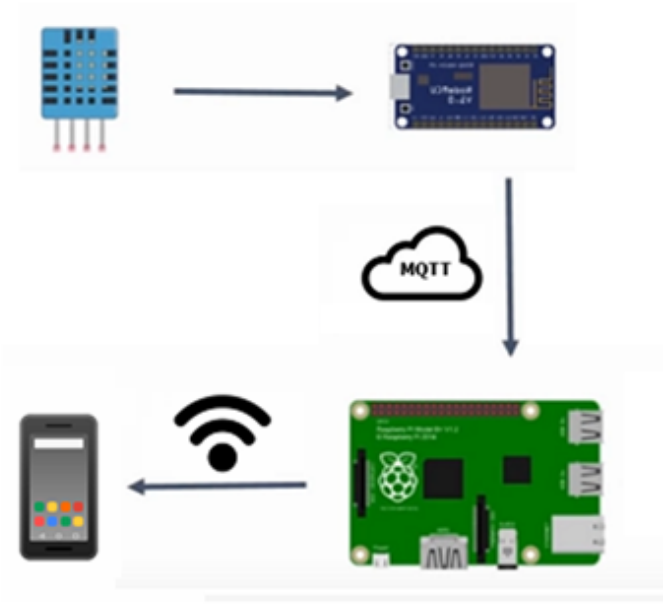
Disadvantages:

- **High Initial Costs**
- **Maintenance**
- **Dependency on Technology**
- **Privacy Concerns**

Applications:

- **Urban Parking Management**
- **Commercial Parking Facilities**
- **Office and Corporate Parking**
- **Transportation Hubs**

Workflow:



Innovations In Smart Parking:

- **Smart Parking Is A Critical Aspect Of Urban Development, and Solving Parking Problems Efficiently Requires Innovative Solutions that Leverage Technology.**

- **Here are Some Design Innovations to Improve Smart Parking.**

Real-Time Parking Availability Apps:

- **Develop a Mobile App That Uses Sensors And Cameras To Provide Real-Time Parking Space Availability in Various Locations.**

- **Utilize Data Analytics To Predict Parking Availability Based On Historical Data And Events In The Area.**

Automated Parking Systems:

- **Implement Automated Parking Systems Where Vehicles Are Automatically Parked in Designated Spots Using Robotic Platforms.**

- **Utilize AI and Computer Vision to Optimize Parking Space Utilization and Minimize Human Intervention.**

Dynamic Pricing and Incentives:

- **Implement a Dynamic Pricing Model For Parking Spots that Incentivizes Off-Peak Parking and Encourages Carpooling.**

- **Use Real-Time Data To Adjust Parking Prices Based On Demand, Time Of Day, and Special Events.**

Integrated Navigation and Parking Guidance:

- **Integrate Navigation Apps With Parking Guidance To Help Drivers Find Available Parking Spots Easily.**

- **Provide Turn-By-Turn Directions To The Nearest Available Parking Based On Real-Time Data.**

Reservation-Based Parking:

- **Allow Users To Reserve Parking Spots In Advance Through a Mobile App or a Web Platform.**
- **Utilize QR Codes or RFID For Seamless Entry and Exit From the Parking Facility.**

Automated Valet Parking:

- **Develop an Automated Valet Parking System Where the Vehicle Self-Parks in Designated Areas Using Advanced Robotics And AI.**
- **This Would Eliminate The Need For Manual Parking, Optimizing Space and Reducing Congestion.**

Parking Space Detection Sensors:

- **Implement IOT-Based Sensors In Parking Spots To Detect The Presence Of A Vehicle And Transmit The Information To a Central Server.**
- **Utilize This Data To Dynamically Update Parking Availability In Real Time.**

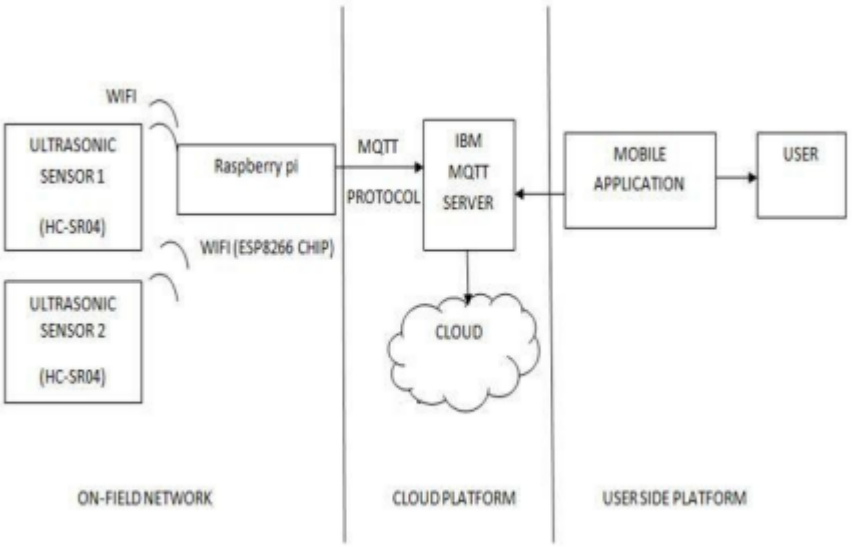
Electric Vehicle (EV) Charging Stations:

- **Integrate EV Charging Stations Within Parking Facilities To Encourage The Adoption Of Electric Vehicles.**
- **Implement Smart Charging Systems That Optimize Charging Based On Demand and Renewable Energy Availability.**

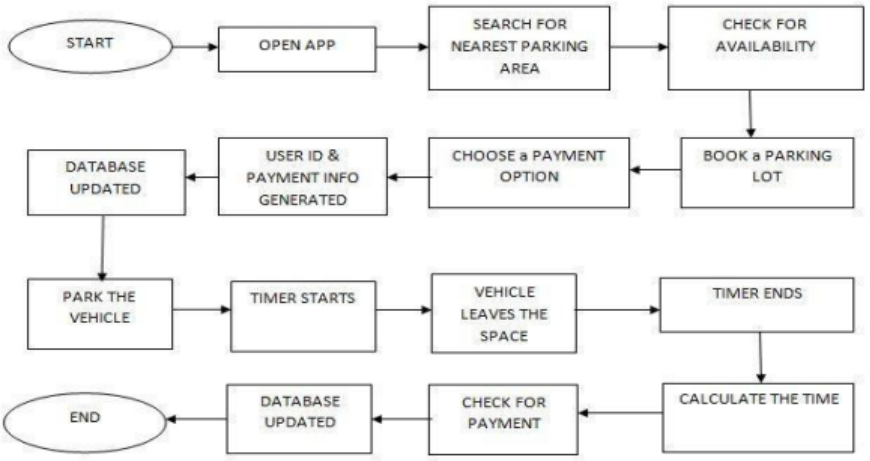
Smart Payment Solutions:

- **Enable Pay-As- You- Go Systems To Reduce Congestion During Exit Times.**
- **Implement Contactless Payment Options Using Mobile Wallets, RFID, or License Plate Recognition For a Seamless and Efficient Payment Process.**

Schematic Diagram:



Overall Working Process:

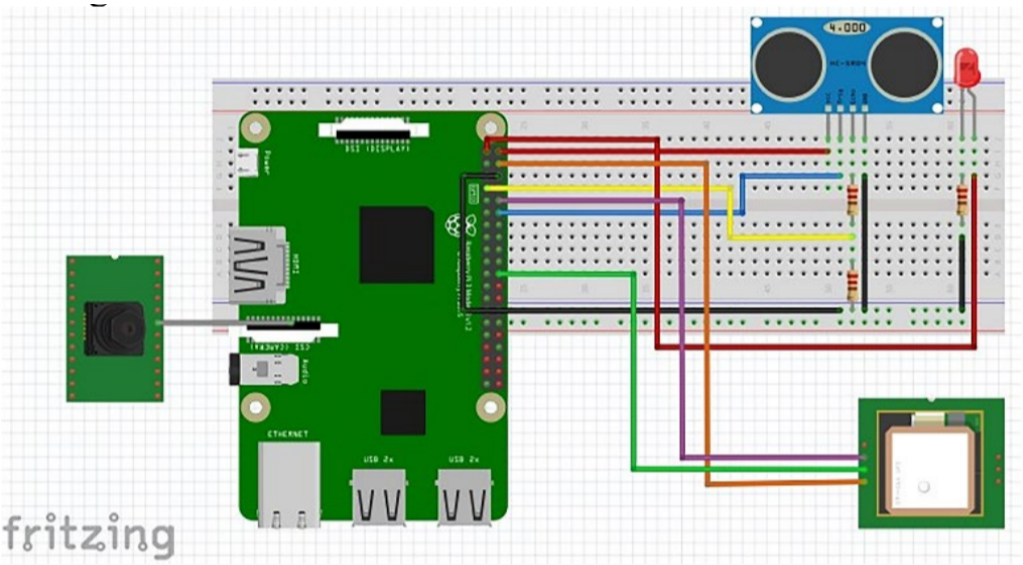


IOT Sensor Setup:

- **In This Section I am Going To Setting Up All The IOT Sensor That are Used with Raspberry Pi.**

REQUIREMENTS:

- **Raspberry Pi 4**
- **HSC-SR04 Ultrasonic Sensors**
- **Pi Camera Module**
- **NEO-6M GPS Sensor**
- **Fritzing**
- **MQTT Server**
- **Firebase**
- **MIT App Inventor**



The Above Pictorial Representation Is The Setting Up All The IOT Sensor With Respect To The Raspberry Pi.

Configuring The Raspberry Pi:

- **In This Section I am Configuring The Raspberry Pi For Using It With The IOT Sensors.**

Python Script:

```
pip install RPi.GPIO
```

Configuring Ultrasonic Sensor:

- **Configuring the Raspberry Pi Pins for The UltraSonic Sensor**
- **Trigger Input Will be given to Specified Pins only**

Python Script:

```
import RPi.GPIO as GPIO
import time

# Set the GPIO mode to BCM
GPIO.setmode(GPIO.BCM)

# Define the GPIO pins for the ultrasonic sensor
TRIG = 23  # GPIO pin for the trigger (TX)
ECHO = 24  # GPIO pin for the echo (RX)

# Set up the GPIO pins
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
```

Configuring NEO-6M GPS Sensor:

- **In This Section I am Configuring The NEO-6M GPS Sensor Using The Python Script.**

Python Script:

```
sudo apt install gpsd gpsd-clients python-gps
```

```
import gpsd
gpsd.connect()
packet = gpsd.get_current()
print(packet.position())
```

Configuring The Pi Camera Module:

- **In This Section I am Configuring The Pi Camera Module Using The Python Script.**

Python Script:

```
pip install picamera
```

```
from picamera import PiCamera
from time import sleep
camera = PiCamera()
camera.capture('image.jpg')
```

Mobile App Development:

- **In This Section I am Developing The Mobile App For Smart Parking**

Developing Mobile App Using Python:

- **The Mobile App Can Be Created Using Python Kivy Framework.**
- **That Retrieve The Stored Data From The MQTT server, shows the available slot for parking.**

Python Script:

```
import kivy
from kivy.app import App
from kivy.uix.label import Label
from kivy.uix.gridlayout import GridLayout
from kivy.uix.textinput import TextInput
from kivy.uix.button import Button
from kivy.uix.widget import Widget
from kivy.properties import ObjectProperty

import paho.mqtt.client as mqtt

class SmartParkingApp(App):
    def build(self):
        layout = GridLayout(cols=2)

        self.lot1 = Label(text="Lot 1: Available")
        layout.add_widget(self.lot1)

        self.lot2 = Label(text="Lot 2: Full")
        layout.add_widget(self.lot2)

        return layout

    def on_start(self):
        self.mqtt_client = mqtt.Client()
        self.mqtt_client.connect("mqtt_broker_ip", 1883)
        self.mqtt_client.subscribe("parking/lot1")
        self.mqtt_client.subscribe("parking/lot2")

        self.mqtt_client.message_callback_add("parking/lot1",
self.lot1_message)
        self.mqtt_client.message_callback_add("parking/lot2",
self.lot2_message)

        self.mqtt_client.loop_start()

    def lot1_message(self, client, userdata, msg):
        data = msg.payload.decode()
        self.lot1.text = "Lot 1: " + data
```

```
    def lot2_message(self, client, userdata, msg):
        data = msg.payload.decode()
        self.lot2.text = "Lot 2: " + data

SmartParkingApp().run()
```

Mobile App Development Framework:

- **The Mobile App Can Be Developed Using Varies Frameworks.**
- **Here, I am Using The Flutter Framework Which is Developed by Google.**
- **It Will Show The Real Time Parking Slot Availability.**
- **The Programming Language Used In Flutter Is Dart.**

Program:

```
import 'package:flutter/material.dart';
import 'package:mqtt_client/mqtt_client.dart';
import 'package:mqtt_client/mqtt_server_client.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   title: 'Parking Availability',
   theme: ThemeData(
    primarySwatch: Colors.blue,
   ),
   home: MyHomePage(),
  );
 }
}

class MyHomePage extends StatefulWidget {
```

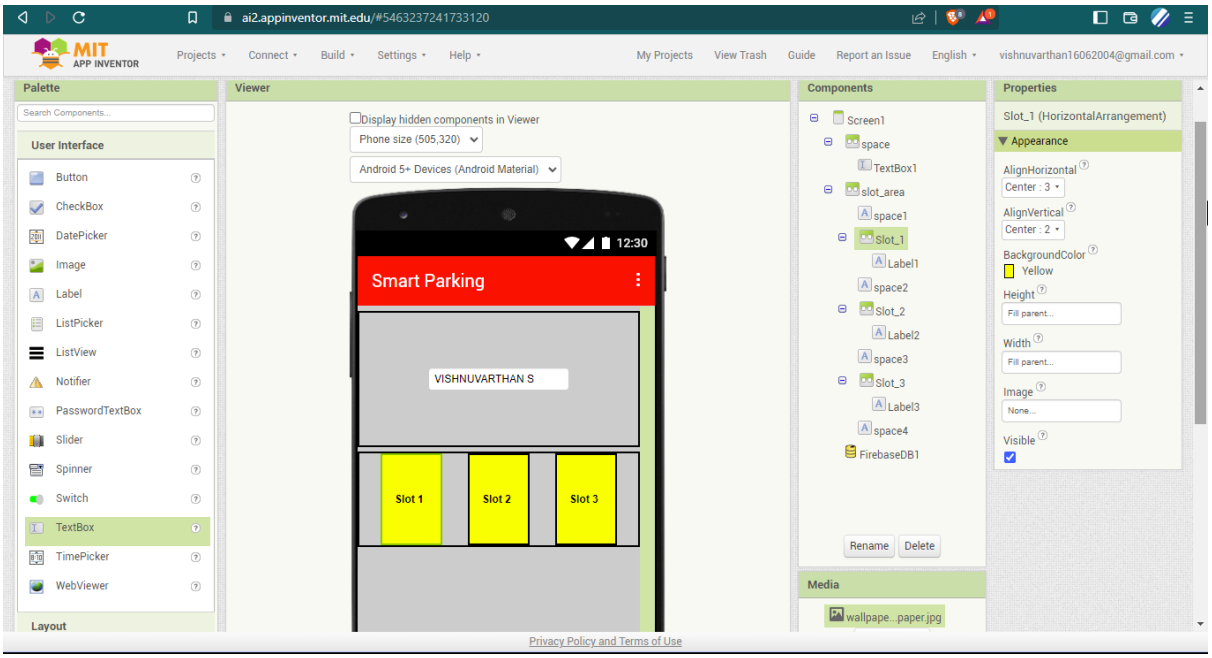
```
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {

  final client = MqttServerClient('localhost', '');

  int availableSpaces = 0;

  @override
  void initState() {
    super.initState();

    client.connect();

    client.subscribe('parking/availability', MqttQos.atLeastOnce);

client.updates.listen((List<MqttReceivedMessage<MqttMessage
>> c) {
      final message = c[0].payload as MqttPublishMessage;
      setState(() {
        availableSpaces = int.parse(message.payload.toString());
      });
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Parking Availability'),
      ),
      body: Center(
        child: Text(
          'Available Parking Spaces: $availableSpaces',
          style: Theme.of(context).textTheme.headline4,
        ),
      ),
    );
  }
}
```

Designing App to Display Parking Availability:

- **In This Section I am Going To Design The App to Receive and Display Parking Space Availability From The Data Received Through the Raspberry Pi.**

- **Here I am using Google's Firebase and MIT App Inventor as Mentioned During The Session.**

- **By using the Firebase Token and Firebase URL the Data Can Be Fetched.**



Raspberry Pi Integration:

- **In This Section I am Integrating The Raspberry Pi With All Parameters**

Collecting Sensor Data:

- **The Ultrasonic Sensor Data is Collected through the Raspberry Pi.**

- **The Collected Data will be processed for various applications.**

Python Script:

```
try:
    while True:
        # Send a short pulse to trigger the ultrasonic sensor
        GPIO.output(TRIG, True)
        time.sleep(0.00001)
        GPIO.output(TRIG, False)

        # Record the start time of the pulse
        while GPIO.input(ECHO) == 0:
            pulse_start = time.time()

        # Record the end time of the pulse
        while GPIO.input(ECHO) == 1:
            pulse_end = time.time()

        # Calculate the duration of the pulse
        pulse_duration = pulse_end - pulse_start

        # Calculate the distance using the speed of sound
(34300 cm/s)
        distance = (pulse_duration * 34300) / 2

        # Print the distance
        print("Distance: {:.2f} cm".format(distance))

        # Sleep for a while to avoid continuous measurements
        time.sleep(1)

except KeyboardInterrupt:
    pass
finally:
    # Clean up the GPIO pins
    GPIO.cleanup()
```

Sending to MQTT Cloud Server:

- **Sending The Collected data from the Ultrasonic Sensor to The MQTT server.**

- **The Received Data Can be Stored in the MQTT Cloud Server.**

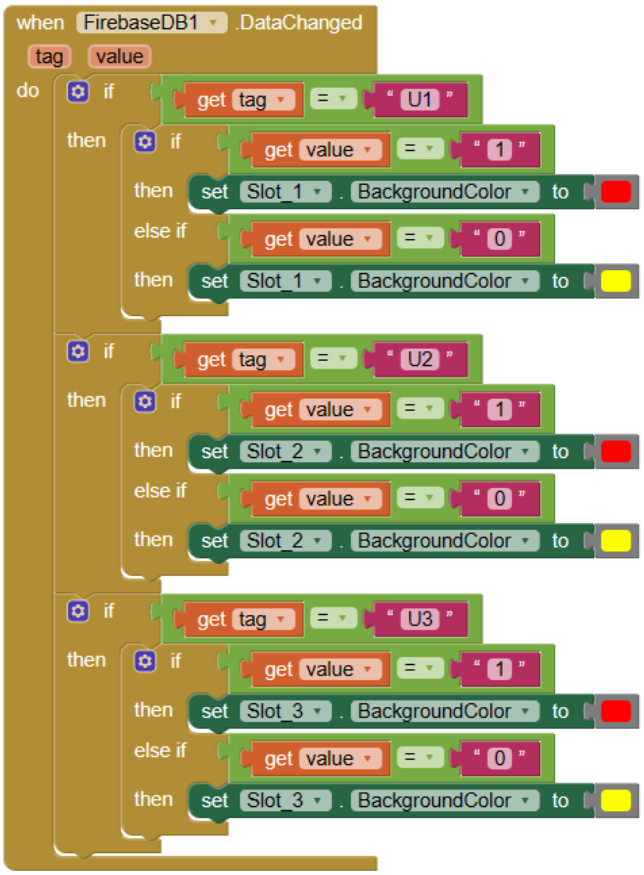
Python Script:

```
import paho.mqtt.client as mqtt

# MQTT settings

mqtt_broker = "My_mqtt_broker_address"
mqtt_port = 1883
mqtt_topic = "Smart_Parking"

# Create an MQTT client

client = mqtt.Client("UltrasonicSensor")

# Connect to the MQTT broker

client.connect(mqtt_broker, mqtt_port, keepalive=60)


# Send the data to the MQTT server

client.publish(mqtt_topic, payload="{:.2f}".format(distance))
```

Code Implementation:

- **For The Code Implementation We Need Many Peripherals Such as Python IDE , Importing Packages and Necessary Library For Raspberry Pi , IOT Sensors**

- **Code Blocks Of MIT App Inventor,**



Benefits Of Smart Parking:

- **In This Section I am Going To Highlight The Benefits Of Smart Parking in Detail.**

Improved User Experience:

- **Smart parking systems provide real-time information about available parking spaces, helping drivers quickly locate and reserve spots.**

Efficient Resource Utilization:

- **Smart parking systems optimize the use of parking spaces, reducing congestion and improving overall parking facility efficiency.**

- **They help prevent overcrowding and ensure a balanced distribution of vehicles throughout the facility, eliminating the need to wait for available spots**

Environmental Benefits:

- **Less time spent circling in search of parking means fewer emissions from idling vehicles, contributing to reduced air pollution and lower greenhouse gas emissions.**

Cost Savings:

- **Drivers save money on fuel by not having to search for parking for extended periods.**

- **Smart parking systems can help drivers avoid parking violations and associated fines by guiding them to legal parking spaces.**

Time Savings:

- **Drivers can quickly find and secure parking spaces, saving them time and reducing frustration.**

- **Smart parking systems often allow for contactless payment and booking, simplifying the parking process.**

Increased Revenue for Operators:

- **Parking facility operators can better manage revenue collection with automated payment systems and real-time monitoring.**

- **Smart parking systems enable operators to implement dynamic pricing strategies based on demand, optimizing revenue.**

Enhanced Safety:

- **By decreasing the time vehicles spend searching for parking, smart parking systems can reduce traffic congestion, making the area safer for pedestrians and other road users.**

- **Some systems incorporate security features, such as surveillance cameras, to improve the safety of parking facilities.**

Data Insights:

- **Smart parking systems generate valuable data that can be used for analytics, such as occupancy patterns, peak usage times, and user preferences. This data can inform better urban planning and resource allocation.**

Reduced Infrastructure Costs:

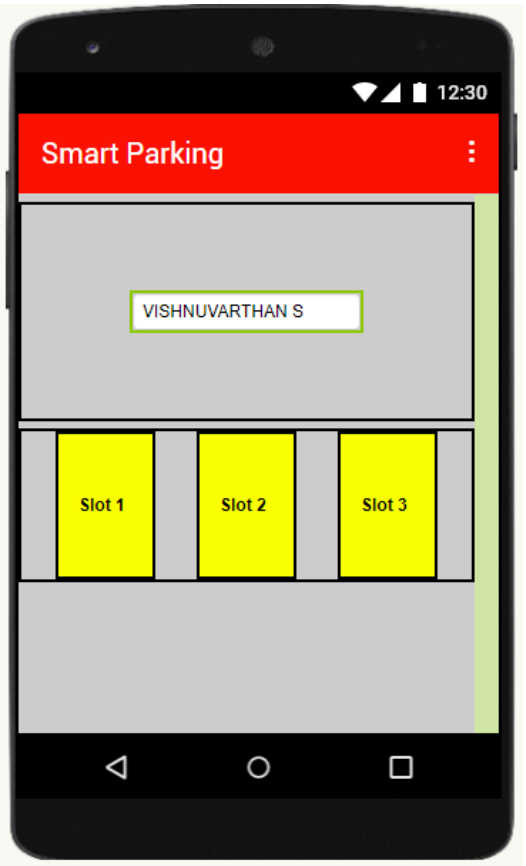
- **Less Need for Expanding Parking Facilities: By optimizing the use of existing spaces, smart parking systems can delay or even eliminate the need for costly expansions of parking facilities.**

City Planning Benefits:

- **Improved parking leads to less congestion on city streets, benefiting overall traffic flow and reducing the risk of accidents.**

- **Reduced Environmental Impact: With fewer vehicles circling in search of parking, there is less traffic-related pollution.**

Output Screen:



Prepared By,

VISHNUVARTHAN S (Team Leader)