PHASE 3.smart parking

Connect Sensors:

Connect your sensors to the Raspberry Pi. Make sure to follow the specific wiring instructions for each sensor.

Install Required Libraries:

Install any necessary Python libraries for your sensors. This might include libraries for I2C/SPI communication, GPIO access, etc.

Read Sensor Data:

Write Python scripts to read data from your sensors. This may involve using GPIO pins or I2C/SPI interfaces depending on the type of sensors you're using.

Process Data (optional):

Process the sensor data if needed (e.g., convert it to a specific unit or format).

Set Up Communication with Cloud/Server:

Choose a method to send data to the cloud or server. This could be through HTTP requests, MQTT, or other protocols.

Create Cloud/Mobile App Server:

Set up a server or use a cloud service (like AWS, Google Cloud, or Azure) to receive and store the sensor data.

Send Data to the Cloud/Server:

Write Python code to send the sensor data to your server/cloud service.

Handle Authentication (if required):

If your server/cloud service requires authentication, make sure to include the necessary credentials.

Handle Error Cases:

Implement error handling to deal with network issues or server unavailability.

Below, I'll provide an example Python script to read data from a DHT22 temperature and humidity sensor and send it to a hypothetical cloud server using HTTP requests.

python

Copy code

```
import requests
import Adafruit_DHT

# Define sensor type and pin
sensor = Adafruit_DHT.DHT22
pin = 4

# Function to read sensor data
def read_sensor_data():
    humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
    return {'temperature': temperature, 'humidity': humidity}

# Function to send data to the server
```

```
def send_data_to_server(data):
    url = 'http://your_server_url_here/data_endpoint'
    headers = {'Content-Type': 'application/json'}
    response = requests.post(url, json=data, headers=headers)
    return response

if __name__ == "__main__":
    sensor_data = read_sensor_data()

    if sensor_data['temperature'] is not None and sensor_data['humidity'] is not None:
        response = send_data_to_server(sensor_data)

        if response.status_code == 200:
            print("Data sent successfully!")
        else:
            print(f"Error sending data. Status code: {response.status_code}")
    else:
        print("Failed to read data from sensor”)
```