**SMART PARKING PYTHON SCRIPT**

Developing an IoT-based Python script for smart parking involves several steps.

1. **Define Requirements:**

\*\*Hardware:\*\* Identify the sensors and devices needed for parking detection (e.g., ultrasonic sensors, cameras).

\*\*Platform:\*\* Choose an IoT platform or hardware (e.g., Raspberry Pi, Arduino) to run your script.

1. **Set Up Hardware:**

Connect sensors and devices to your chosen platform. Ensure you follow the hardware specifications and pin configurations.

1. **Install Required Libraries:**

Install any necessary Python libraries for your hardware. For example, if you're using Raspberry Pi, you might need to install the `RPi.GPIO` library.

```bash

pip install RPi.GPIO

```

1. **Create Python Script:**

Write a Python script that reads sensor data and communicates with the IoT platform.

```python

import RPi.GPIO as GPIO import time import requests

SENSOR\_PIN = 17

LED\_PIN = 18

GPIO.setmode(GPIO.BCM)

GPIO.setup(SENSOR\_PIN, GPIO.IN)

GPIO.setup(LED\_PIN, GPIO.OUT)

API\_ENDPOINT = "https://your-api-endpoint.com/update\_parking\_status"

def update\_parking\_status(status):

data = {'status': status} requests.post(url=API\_ENDPOINT, data=data)

try: while True:

if GPIO.input(SENSOR\_PIN) == GPIO.HIGH: print("Parking occupied")

GPIO.output(LED\_PIN, GPIO.HIGH) update\_parking\_status("occupied")

else:

print("Parking vacant")

GPIO.output(LED\_PIN, GPIO.LOW) update\_parking\_status("vacant") time.sleep(5)

except KeyboardInterrupt: GPIO.cleanup()

```

1. **API Integration:**

Implement the API integration part in your script. Replace

`"https://your-api-endpoint.com/update\_parking\_status"` with your actual API endpoint.

1. **Test:**

Run the script on your IoT device and test it with the parking sensors. Make sure the script can accurately detect parking status.

1. **Deployment:**

Deploy your IoT device with the script in the parking area.

1. **Monitoring:**

Implement monitoring mechanisms or logging to track the performance of your smart parking system.

1. **Security:**

Consider security measures for your IoT device and communication. Use secure connections (HTTPS) and implement authentication if necessary.

1. **Documentation:**

Document your script, hardware setup, and any configurations for future reference or collaboration.