PHASE : 4   SMART PARKING

REAL—TIME SMART PARKING

PYTHON PROGRAMMING :

Creating a full smart parking system program is a complex task that involves various components, such as hardware sensors, databases, and mobile apps. I can provide you with a simple Python example that simulates a basic smart parking system where you can check the availability of parking spaces. This example uses a dictionary to represent parking spots and their availability:

```python
Class SmartParkingSystem:
    Def __init__(self, total_spots):
        Self.total_spots = total_spots
        Self.parking_spots = {spot: False for spot in range(1, total_spots + 1)}

    Def check_availability(self):
        Available_spots = [spot for spot, is_available in self.parking_spots.items() if not is_available]
        Return available_spots

    Def occupy_spot(self, spot):
        If 1 <= spot <= self.total_spots:
            If not self.parking_spots[spot]:
                Self.parking_spots[spot] = True
                Return f"Parking spot {spot} is now occupied."
            Else:
                Return f"Parking spot {spot} is already occupied."
        Else:
            Return "Invalid spot number."
```

~~~
    Def vacate_spot(self, spot):
        If 1 <= spot <= self.total_spots:
            If self.parking_spots[spot]:
                Self.parking_spots[spot] = False
                Return f”Parking spot {spot} is now vacant.”
            Else:
                Return f”Parking spot {spot} is already vacant.”
        Else:
            Return “Invalid spot number.”

# Initialize a smart parking system with 10 spots
Smart_parking = SmartParkingSystem(10)

# Check the availability of parking spots
Print(“Available spots:”, smart_parking.check_availability())

# Occupy spot 3
Print(smart_parking.occupy_spot(3))

# Check availability again
Print(“Available spots:”, smart_parking.check_availability())

# Vacate spot 3
Print(smart_parking.vacate_spot(3))

# Check availability once more
Print(“Availablespots:”) smart_parking.check_availability())
```
~~~

This program represents a simplified parking system in Python. In a real-world scenario, you would integrate this logic with hardware sensors, a database, and possibly a mobile app to create a comprehensive smart parking system