TOPIC: SMART PARKING USING IOT

PHASE\_3 – DEVELOPMENT PART

Project Description:

Define the specific requirements for your smart parking system, such as the number of parking spots, location, and the features you want to implement (e.g., real-time monitoring, mobile app integration, payment processing).

Hardware Selection: Choose IoT hardware components, including sensors (e.g., ultrasonic sensors to detect vehicle presence), microcontrollers, and communication modules (e.g., Wi-Fi, LoRa, or cellular).

Sensor Deployment: Install sensors in parking spots to detect vehicle presence and relay this data to a central system.

Connectivity: Set up the necessary connectivity infrastructure to ensure data from the sensors can be transmitted to a central server. This might involve Wi-Fi or cellular networks.

Data Processing: Design a central server or cloud-based system to collect, process, and analyze the data from the sensors. This system can make sense of the data and provide real-time updates on parking spot availability.

User Interface: Develop a user-friendly interface, such as a mobile app or a website, to display real-time parking availability information to users.

Payment Integration: If you want to implement payment processing, integrate payment gateways within your system to allow users to pay for parking through the app or website.

Security: Implement security measures to protect data, including encryption and access control, to ensure the privacy and integrity of the system.

Testing: Thoroughly test the system to ensure it works as expected and can handle different scenarios.

Scaling: Plan for future scalability by ensuring your system can accommodate additional parking spots or locations.

Maintenance and Updates: Regularly maintain and update the system to address issues, enhance functionality, and stay current with technology advancements.

User Education: Provide information to users on how to use the system effectively, including how to access real-time parking availability information and make payments.

Monitoring and Optimization: Continuously monitor the system's performance and gather feedback from users to make improvements and optimize its operation.

Compliance: Ensure your system complies with any local regulations and standards, especially in relation to data privacy and accessibility.

**Python Program:**

import random

import time

# Simulate a parking spot with a sensor

class ParkingSpot:

def init(self):

self.occupied = False

def occupy(self):

self.occupied = True

def vacate(self):

self.occupied = False

# Function to monitor and update parking spot status

def monitor\_parking\_spot(parking\_spot):

while True:

# Simulate vehicle arrivals and departures

if random.random() < 0.5:

parking\_spot.occupy()

else:

parking\_spot.vacate()

# Print the status of the parking spot

status = "Occupied" if parking\_spot.occupied else "Vacant"

print(f"Parking Spot is {status}")

time.sleep(2)

if name == "main":

parking\_spot = ParkingSpot()

print("Smart Parking System Simulation")

try:

monitor\_parking\_spot(parking\_spot)

except KeyboardInterrupt:

print("Simulation terminated.")