**Smart parking system in iot**

**phase 3**

Internet of Things (IoT) leverages sensors and technology to provide real-time information about parking availability and help manage parking spaces more efficiently. Here's how it works:

1 . Sensors: IoT-enabled sensors are installed in parking spaces or garages. These sensors can be in-ground sensors, ultrasonic sensors, or cameras that detect the presence of vehicles.

2 . Data Collection: The sensors collect data on parking space occupancy, such as whether a space is vacant or occupied. This data is then transmitted to a central server or cloud platform.

3 . Data Processing: The collected data is processed and analyzed to determine parking space availability. Algorithms can predict and report parking availability in real-time.

4. User Access: Users can access this information through various means, such as mobile apps, websites, or digital signage. They can see which parking spaces are available, reserve spots in advance, or receive notifications when a spot becomes available.

5.. Payment Integration: Some systems also include payment processing, allowing users to pay for parking through the app or via automated payment gates.

6. Traffic Management: Smart parking systems can help reduce traffic congestion by guiding drivers to available parking spaces, preventing unnecessary circling in search of parking.

7. Maintenance and Alerts: The system can also provide alerts for maintenance needs, such as sensor malfunctions or empty payment kiosks, ensuring smooth operations.

8. ata Analytics: Data collected from these systems can be used for further analysis to optimize parking management, such as identifying peak usage times or underutilized spaces.

IoT-based smart parking systems not only enhance the user experience but also improve the overall efficiency of parking facilities, reduce environmental impact, and contribute to smart city initiatives.

A smart parking system involved in the Internet of Things (IoT) uses connected sensors and technology to manage parking spaces more efficiently. Here's how it works:

* Sensor Deployment: IoT sensors are installed in parking spaces, either on the ground or overhead, to monitor the availability of each spot.
* Data Collection: These sensors collect data on the occupancy status of parking spaces, which includes information on whether a spot is empty or occupied.
* Data Transmission: The sensor data is transmitted to a central server or cloud platform through IoT connectivity, such as Wi-Fi, cellular networks, or LoRaWAN.
* Mobile App or Website: Users can access real-time parking availability information through a mobile app or a website. This helps them find and reserve available parking spots.
* Payment and Booking: Users can make reservations and payments through the app or website, making the parking process more convenient.
* Data a nalytics: The system can analyze historical parking data to optimize parking space allocation and pricing.
* Alerts and Notifications: Users can receive alerts when their parking time is about to expire, or if there are any issues with their reservation.
* Integration with Navigation: Some systems integrate with navigation apps to provide users with directions to the available parking spots.

Overall, an IoT-based smart parking system improves the efficiency of parking management, reduces congestion, and enhances the user experience by providing real-time information and convenient payment options

Integrating AI (Artificial Intelligence) into a smart parking system based on IoT can further enhance its capabilities. Here's how AI can be applied to smart parking systems:

* Predictive Analytics: AI can analyze historical parking data and predict parking space availability based on patterns, events, and special occasions. This helps users plan their parking in advance.
* Dynamic Pricing: AI algorithms can adjust parking prices in real-time based on demand, time of day, or other factors. This helps optimize parking space utilization and revenue.
* Image Recognition: AI can be used for license plate recognition and vehicle identification, ensuring secure access control and assisting in enforcing parking regulations.
* Anomaly Detection: AI can identify unusual or unauthorized parking behavior, such as vehicles parked in no-parking zones, and alert authorities or security personnel.
* Traffic Flow Optimization: AI can provide real-time traffic information to guide users to available parking spaces and reduce congestion around parking facilities.
* Personalized Recommendations: AI can offer personalized parking suggestions to users based on their past preferences, location, and historical data.
* Customer Support Chatbots: AI-powered chatbots can assist users with parking-related inquiries and provide customer support through the app or website.
* Energy Efficiency: AI can help optimize the use of IoT sensors to reduce energy consumption and maintenance costs.

By integrating AI into IoT-based smart parking systems, you can create a more intelligent and responsive parking solution that adapts to changing conditions, enhances user experience, and maximizes efficiency.

Program for smart parking in iot

Copy code

import time

import RPi.GPIO as GPIO

import requests

# Define GPIO pins for sensors

sensor\_pins = [2, 3, 4] # Replace with actual GPIO pin numbers

# Define the server URL for updating parking status

server\_url = "http://your\_server\_address/parkingsystem"

# Initialize GPIO

GPIO.setmode(GPIO.BCM)

for pin in sensor\_pins:

GPIO.setup(pin, GPIO.IN)

def update\_parking\_status(parking\_spot, status):

payload = {"spot\_id": parking\_spot, "status": status}

try:

response = requests.post(server\_url, json=payload)

if response.status\_code == 200:

print(f"Parking spot {parking\_spot} is {status}")

else:

print(f"Failed to update status for spot {parking\_spot}")

except Exception as e:

print(f"Error updating parking status: {e}")

try:

while True:

for spot, pin in enumerate(sensor\_pins):

status = GPIO.input(pin)

update\_parking\_status(spot, "occupied" if status == 0 else "vacant")

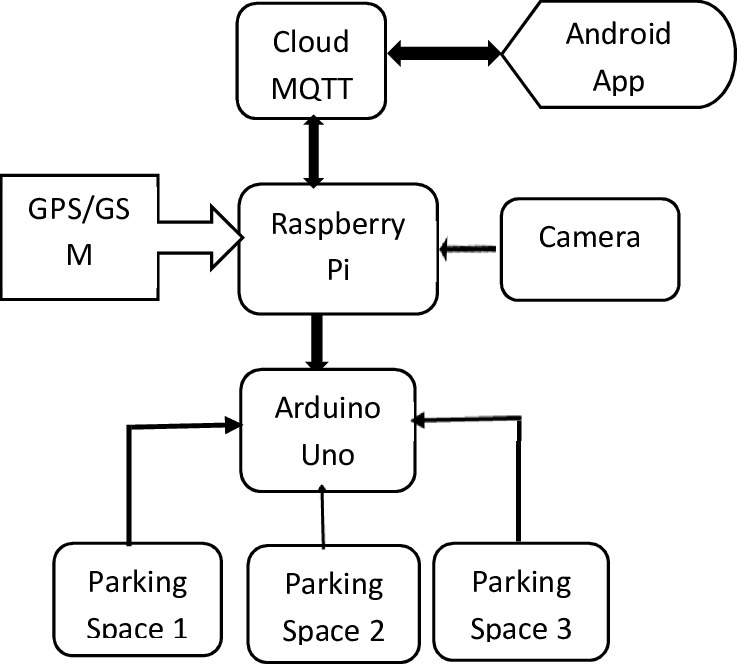
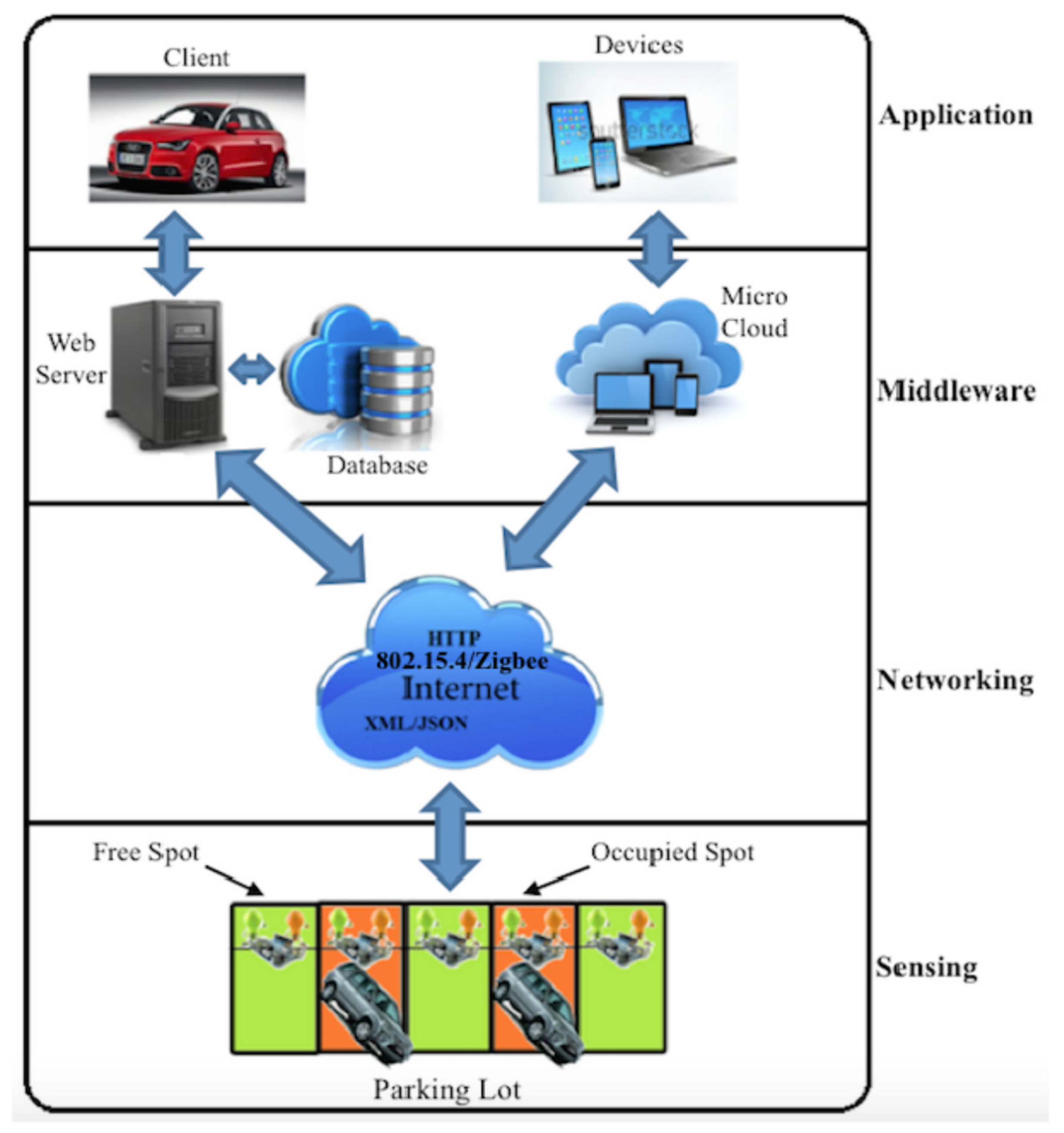
time.sleep(1) # Update status every 1 second

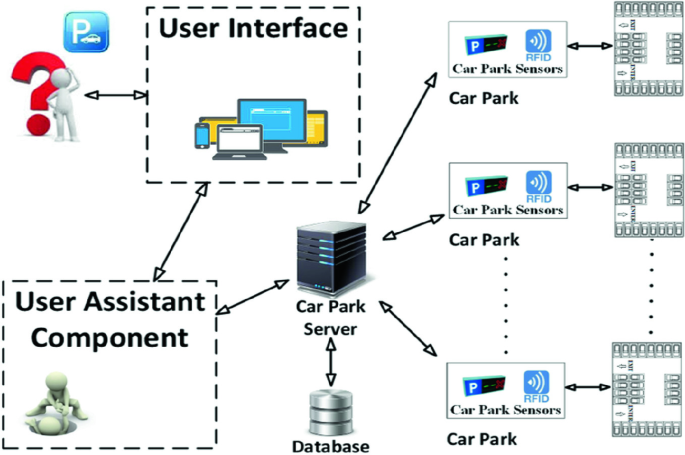
except KeyboardInterrupt:

GPIO.cleanup()

In this simplified example, we're using Raspberry Pi with GPIO pins to connect the parking sensors. The program continuously monitors the status of parking spots (occupied or vacant) and updates a server (represented by server\_url) with this information.

In a real-world implementation, you'd have to consider more factors like database integration, security, user interfaces, and scaling for multiple parking lots. Additionally, you'd need sensors to detect vehicle presence, a central server to manage the data, and a user interface for displaying parking availability to users.

FLOW



***Project was carried out by***

* ***V .Dhivyasri ((T.L)***
* ***R. Dhanasekar***
* ***P. Doss***