SMART PARKING

PROBLEM STATEMENT**:**

In recent research in metropolitan cities the parking management problem can be Viewed from various angles such as high vehicle density on roads. This results in Annoying issues for the drivers to park their vehicles as it is very difficult to find a Parking slot.The drivers usually waste time and effort in finding parking space and end up parking Their vehicles finding a space on the street which further leads to space congestion. In Worst case, people fail to find any parking space especially during peak hours and Festive season.

OBJECTIVES:

Optimized parking – Users find the best spot available, saving time, resources and Effort. The parking lot fills up efficiently and space can be utilized properly by Commercial and corporate entities.

Reduced traffic – Traffic flow increases as fewer cars are required to drive around in Search of an open parking space.

Reduced pollution – Searching for parking burns around one million barrels of oil a Day.

Increased Safety – Parking lot employees and security guards contain real-time lot Data that can help prevent parking violations and suspicious activity. License plate Recognition cameras can gather pertinent footage. Also, decreased spot-searching Traffic on the streets can reduce accidents caused by the distraction of searching for Parking.

Decreased Management Costs – More automation and less manual activity saves on Labor cost and resource exhaustion.

Enhanced User Experience – A smart parking solution will integrate the entire user Experience into a unified action. Driver’s payment, spot identification, location search And time notifications all seamlessly become part of the destination arrival process.

METHODOLOGY:

In this project we are using NodeMCU, IR sensors, and servo motors. One IR sensor is Used at entry and exit gate to detect the car while two IR sensors are used to detect the Parking slot availability. Servo motors are used to open and close he gets accordingly to the sensor value. NodeMCU is an open source IoT platform .It includes firmware Which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware, which Is based on the ESP-12 module. The term “NodeMCU” by default refers to the Firmware rather than the dev kits. The firmware uses the Lua scripting language. The ESP8266 is a low-cost Wi-Fi enabled microchip with full TCP/IP stack and Microcontroller capability. NodeMCU includes CPU core, faster Wi-Fi, more GPIOs, And supports Bluetooth 4.2, and low power Bluetooth. The ESP8266 is a low-cost WiFi

enabled microchip with full TCP/IP stack and microcontroller capability. NodeMCU Includes CPU core, faster Wi-Fi, more GPIOs, and supports Bluetooth 4.2, and low Power Bluetooth. As soon as the IR sensors get the presence of a car in front of the Entrance, it will send signal to the NodeMCU to check if there is an empty slot inside The parking lot. When NodeMCU acknowledges that there is an empty slot or more Then it will send a signal to the dc servo motor which will open the main entrance. On The other hand if an NodeMCU encounters no empty slots at the time of a car trying to Make an entrance, the gate will just not open. In addition, there will be a website linked. With the NodeMCU board to show the number of parking.

The idea behind our methodology is very simple , usually users spend most of their Time in looking for an empty slot where they can park their vehicle which increases fuel Consumption and time wastage. We came-up with a new method where we provide the User an empty slot number where he can park his vehicle without wasting his time for Finding one. Similarly we try to display the start time and end time so that the user can Know for what amount of time he has parked his vehicle.

COMPONENTS AND TECHNOLOGIES:

Sensors: Deploy various types of sensors (ultrasonic, magnetic, infrared, or cameras) in each parking space to detect vehicle occupancy.

Microcontrollers/IoT Devices: Connect sensors to microcontrollers or IoT devices (e.g., Raspberry Pi, Arduino, or dedicated IoT modules) to collect data and control sensors.

Connectivity: Establish a reliable network connection (Wi-Fi, LoRa, or cellular) to transmit sensor data to a central server.

Central Server: Set up a cloud-based or on-premises server to receive and process data from all sensors.

Data Processing: Implement data processing algorithms to analyze sensor data and determine parking space availability.

User Interface: Develop a mobile app or web portal for users to check parking space availability, reserve spots, and make payments.

Payment Gateway: Integrate a payment gateway for users to pay for parking reservations.

Parking Management System: Implement a backend system to manage parking reservations, allocate spots, and monitor the overall system.

Database: Use a database to store information about parking spots, reservations, and user accounts.

Notifications: Send notifications to users regarding parking availability, reservations, and payment confirmations.

PROGRAM (PYTHON – RASPBERRY PI):

```
Import RPi.GPIO as GPIO
Import time
# Configure ultrasonic sensor pins
TRIG_PIN = 18
ECHO_PIN = 24

# Set GPIO mode and warnings
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
# Initialize sensor pins
GPIO.setup(TRIG_PIN, GPIO.OUT)
GPIO.setup(ECHO_PIN, GPIO.IN)
Def measure_distance():
    # Trigger ultrasonic sensor
    GPIO.output(TRIG_PIN, True)
    Time.sleep(0.00001)
    GPIO.output(TRIG_PIN, False)
    # Measure echo time
    While GPIO.input(ECHO_PIN) == 0:
        Pulse_start = time.time()
    While GPIO.input(ECHO_PIN) == 1:
        Pulse_end = time.time()
    Pulse_duration = pulse_end – pulse_start
    # Calculate distance (in cm)
    Distance = pulse_duration * 17150
    Return distance
Try:
```

```
    While True:
        Distance = measure_distance()
        If distance < 10:  # Adjust this threshold based on your parking spot dimensions
            Print("Parking spot occupied")
        Else:
            Print("Parking spot vacant")
        Time.sleep(2)  # Adjust the frequency of sensor readings as needed
Except KeyboardInterrupt:
    Print("Exiting…")
    GPIO.cleanup()
```

CONCLUSION:

The concept of Smart Cities has always been a dream for humanity. Since the past Couple of years ago large advancements have been made in making smart cities a Reality. The growth of Internet of Things and Cloud technologies have given rise to new Possibilities in terms of smart cities. Smart parking facilities and traffic management Systems have always been at the core of constructing smart cities. In this project, we Address the issue of parking and present an IoT based Cloud integrated smart parking System.

The system that we propose provides real time information regarding Availability of parking slots in a parking area. Users from remote locations could book A parking slot for them by the use of our mobile application. The efforts made in this project are intended to improve the parking facilities of a city And thereby aiming to enhance the quality of life of its people.