# 6115-mahendra institute of engineering and technology

# Smart parking system

# TEAM ID :572

**TEAM: PROJ\_223289\_TEAM\_1**

# Team members:

1. DEVAPRAKASH R
2. LOGESHWARAN G
3. MANIKANDAN K.A
4. ARUNKUMAR P
5. KAMALNATH A
6. GUNASEKARAN M

**FACULTY MENTOR NAME:**

SANTHANARAJ M

**DECLARATION:**

We, the students of Computer Science and Engineering,

MAHENDRA INSITUTE OF ENGINEERING AND TECHNOLOGY,

TAMIL NADU

that the work entitled " SMART PARKING SYSTEM " has been successfully completedunder the guidance

of Asst Prof. Ms. santhana raja M, Computer Science and EngineeringDepartment, mahendra institute of

engineering and techonology , namakkal. This dissertation work is submitted in partialfulfillment of the

requirements for the award of Degree of Bachelor of Engineering in ComputerScience and Engineering

during the academic year 2021-2025.

**ABSTRACTION**

* The project entitled SMART PARKING SYSTEM using Iot , the major motivation of this project is toreduce the traffic congestion in roads, multi-storeyed buildings and malls due to unavailability of parking spaces .
* The project displays the nearest empty slot if present with respect to user location. Ourproject aims to make efficient use of parking spaces. We track vacant slots in the parking space andassign that to the user.
* Smart parking system as described above can lead to an error-free , reliable,secure and fast management system.
* In recent times the concept of smart cities have gained greatpopularity. Thanks to the evolution of the Internet of things the idea of smart city now seems to beachievable.
* Consistent efforts are being made in the field of IoT in order to maximize the productivity and reliability of urban infrastructure.accordingly.
* The paper also describes a high-level view of the system architecture. Towards the end, the paper discusses the working of the systemin form of a use case that proves the correctness of proposed model.

**INTRODUCTION**

* The project entitled smart parking system is to manage all the parking facilities to an user.
* The recent growth in economy and due to the availability of low price cars in themarket, an every average middle-class individual can afford a car, which is good thing,
* however the consequences of heavy traffic jams, pollution, less availability of roadsand spot to drive the motor car.
* One of the important concerns, which is to be taken in accounting, is the problem of parking those vehicles .
* Though, if there is space forparking the vehicle but so much time is squandered in finding exact
* parking slotresulting in more fuel intake and not also environment friendly. It will be a great dealif in some way.
* we find out that the parking itself can provide the precise vacant position of a parking slot then it'll be helpful not limited to the drivers also for the environment
* Initially when the user is about to enter the location the LCD displays the number ofempty and filled spots and when the user is with its vehicle near to the parking detect
* sensor ,he/she would be thrown with a notification on their mobile app of the parking
* slot number ,where they should park there vehicle.

**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.



## Objective

* Smart Parking involves the use of low cost sensors, real-time data and applications thatallow users to monitor available and unavailable parking spots.
* The goal is to automateand decrease time spent manually searching for the optimal parking floor, spot andeven lot.



* Some solutions will encompass a complete suite of services such as onlinepayments, parking time notifications and even car searching functionalities for verylarge lots.
* A parking solution can greatly benefit both the user and the lot owner.

**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. An optimal parking solution will significantly decrease driving time, thus lowering the amount of daily vehicle emissions and ultimately reducing the global environmental footprint



**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.

**ADVANTAGES**

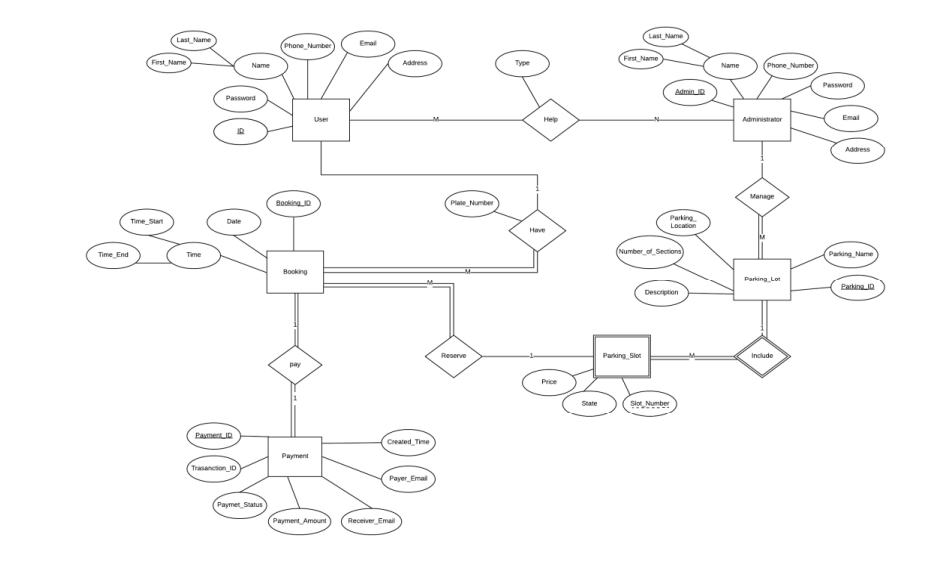
1. Reduced traffic congestion
2. Reduced emissions
3. Improved driver experience
4. Increased revenue for parking managers
5. Improved security

**Scope of the project**

* At present some countries have portals which users can gain information about parking areas via the internet.
* This system can give users the information about parking space, but it won’t be able to give which parking slot is vacant and occupied. Hence, such a system cannot smartly handle the issue.
* Car lifts along with an automated robotic system, which automatically takes the car to a particular parking spot as soon as the car enters on a platform.
* This system cannot be installed by medium scale shopping malls, movie theatres as it can cost them a huge amount.
* At many public places, the system only shows the availability but it cannot show the exact slot and path to the slot available.
* Hence, there is the need to smartly find the path to the vacant spot.

**Developing a Smart Parking Management System Using the Internet of Things**:

* Searching for parking wastes significant amounts of time and effort and leads to substantial financial costs. This is particularly the case for people who are always pressured to be on time. Smart cities employ all kinds of modern technologies to manage and enhance resources effectively. Urban parking facilities are one of the essential assets that must be managed. We developed a smart parking management system (SPMS) as a modern solution to manage parking and save users time, effort and cost. In the context of today’s modern life, it has become necessary to improve search methods for available parking and minimize the congestion that occurs at the parking entrance. Searching or booking available parking online earlier is a better substitute than searching at a parking lot where there is a possibility of not being able to find parking. Our smart parking management system was developed to: • Manage parking and solve problems efficiently using technology • Apply technical solutions to improve the smart cities concept The proposed system uses a variety of technologies that help manage parking. It provides essential services for users, including searching for parking, reservations and payment. It is extended to cover more advanced services such as receiving notifications, statistics and monitoring the parking state. The system is connected to sensors to detect occupancy and an automatic number plate recognition (ANPR) camera to control access. The remainder of the paper is organized as follows.



**An IoT-based E-Parking System for Smart Cities**

The huge proliferation in the number of vehicles on the road along with mismanagement of the available parking space has created parking related problems as well as increased the traffic congestion in urban areas. Thus, it is required to develop an automated smart parking management system that would not only help a driver to locate a suitable parking space for his/her vehicle, but also it would reduce fuel consumption as well as air pollution. It has been found that a drivers search for a suitable parking facility takes almost 15 minutes which increases the fuel consumption by the vehicle, traffic congestion and air pollution. A significant amount of research works exist in the area of design and development of smart parking system. Various features of smart parking system are listed below

• Inquiry on availability of parking space and reservation of parking lot

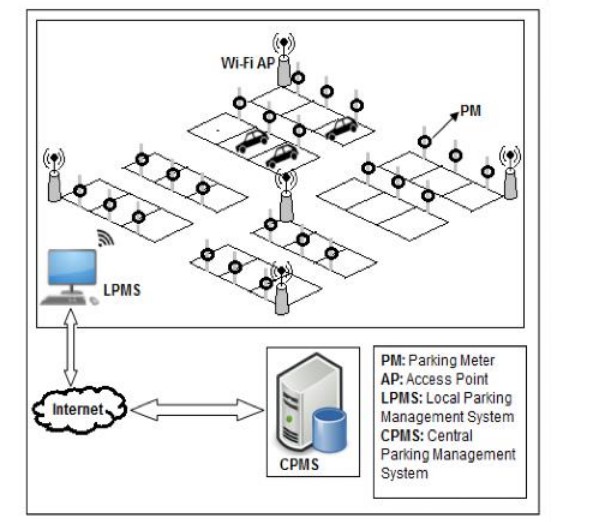
• Real-time parking navigation and route guidance

• Vehicle occupancy detection and management of parking lots . Most of the smart parking systems (SPS) proposed in literature over the past few years provides solution to the design of parking availability information system, parking reservation system, occupancy detection and management of parking lot, real-time navigation within the parking facility etc. However, very few works have paid attention to the real time detection of improper parking and automatic collection of parking charges. Thus, this paper presents an internet-of thing (IoT) based E-parking system that employs an integrated component called parking meter (PM) to address the following issues.

• Real-time detection of improper parking

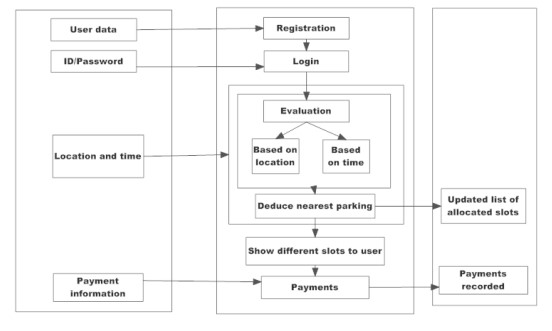
• Estimation of each vehicles duration of parking lot usage

• Automatic collection of parking charges The E-parking system proposed in this paper also provides city-wide smart parking management solution via providing parking facility availability information and parking lot reservation system and it is named as parking meter (PM) based E-parking (PM-EP).



**Smart Parking based System for smarter cities**

India is getting motorized i.e. the rate of private vehicles is more as compared to public transports. As the rate of people owning their vehicles increases ,the need of parking slots to park vehicles also increases. But currently the scenario is that there are not sufficient parking slots available or there is also possibility that people are not now aware about the legal parking slots available in their locality. This situation leads to the unnecessary crowding of vehicles on the road and also results in inconveniency of people walking on the road. To overcome above problems, We are proposing the solution in the form of a multilingual android application which will be helpful for the people to find their parking slots digitally. By digitally we mean that this particular system will assign the parking slot based on the current location of the user and the parking slot which the user wants according to his/her ease. Ease in terms of finding the exact slot. The payments can be done digitally or through vending machines. The end user can register and login with his/her account which will help the system to find the location and displaying the nearest parking area and nearest parking slot ,whether it is available or not. If not then it will direct user to the next nearest slot and so on. The existing system comprises of both traditional and application based approach for parking. If we talk about the traditional approach it utilizes manual method of parking i.e user has to find the spot for parking by traveling to far distances and paying extra money. An application based approach consist of the applications which provides the parking slots for the particular locality for example .The application named ‘Parking Panda ’ provides the parking slots to the areas like stadium, sports leagues

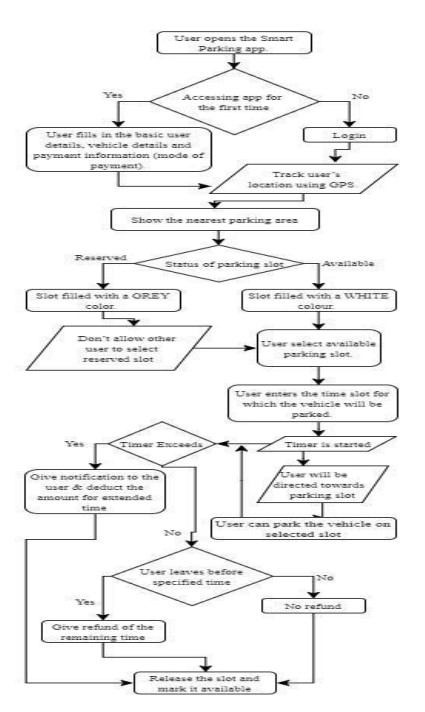


**SMART PARKING SYSTEM TO REDUCE TRAFFIC CONGESTION**

Transportation is the key-success for any of the country. Now a day, many people have options to use their own vehicle for travelling. This will surely increase the demand in trading but one of the problems created by road traffic is "parking". To park all these vehicles in the major metro cities is quite tedious and difficult task and it became problematic to park vehicles. Lot of research and development is being done all over the world to implement better and smarter parking management mechanisms. The current smart parking systems or Wireless Sensors Network Parking requires the combination of wireless sensor networks module, Embedded web-server, Central Web Server. Sensor networks make use of Infrared (IR) Sensor nodes to check the parking slot state and send this information to embedded web-server. It thereby displays the information on a LED screen with which the user can check for empty vehicle slots. These systems not guide the users to reach to the parking lot. If the slot is not available

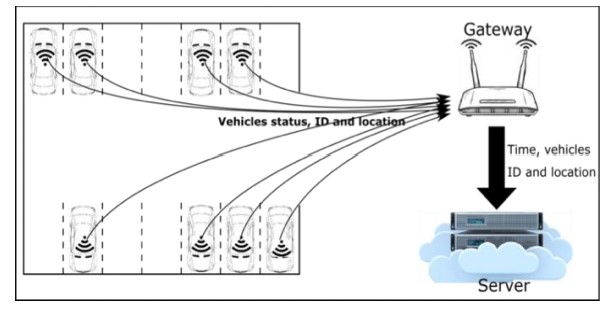
**SMART PARKING SYSTEM TO REDUCE TRAFFIC CONGESTION**

Transportation is the key-success for any of the country. Now a day, many people have options to use their own vehicle for travelling. This will surely increase the demand in trading but one of the problems created by road traffic is "parking". To park all these vehicles in the major metro cities is quite tedious and difficult task and it became problematic to park vehicles. Lot of research and development is being done all over the world to implement better and smarter parking management mechanisms. The current smart parking systems or Wireless Sensors Network Parking requires the combination of wireless sensor networks module, Embedded web-server, Central Web Server. Sensor networks make use of Infrared (IR) Sensor nodes to check the parking slot state and send this information to embedded web-server. It thereby displays the information on a LED screen with which the user can check for empty vehicle slots. These systems not guide the users to reach to the parking lot. If the slot is not available parking. Then in search of parking space they have to again roam with their vehicle to find available parking.



**Flow chart of Smart Parking System**

An IoT-Based Intelligent System for Real-Time Parking Monitoring and Automatic Billing today, the parking industry is being transformed by new technologies that are allowing cities to reduce rates of congestion significantly. Sensor networks that sense vehicle occupancy are providing the basic intelligence behind smart parking systems. Thanks to the Smart Parking technology, it is now possible to know in real-time the location of free parking spaces and to help drivers to get to their ultimate destination. A variety type of vehicle detectors has been used in parking information acquisition. These vehicle detectors mainly include the inductive loop , acoustic sensor , infrared sensor , or ultrasonic sensor . System using video camera sensor technologies have been proposed to collect the information in vehicle parking field. However, a video camera sensor is vulnerable to bad weather and night time operation. Furthermore, it is expensive, and can generate a large amount of data that can be difficult to transmit in a wireless network. The magneto-resistive based detection systems combined with a wireless area network are the most popular technique due to their high accuracy. Yet, this type of sensor is facing different issues, i.e. it can be bedeviled by electromagnetic interference, which affects the accuracy , the reading from sensor needs to be collected constantly which will result in wearing out the battery . To extend the battery lifetime and increase the vehicle detection accuracy, a parking sensor system has been proposed. While power management technique has been implemented to optimize energy consumption, high occupancy monitoring accuracy is achieved using two-fold sensing approach. It is a sequence of darkness and Signal Strength Indicator (RSSI) measurement based techniques. The wireless sensors are still intrusive, they are embedded in the pavement, or taped to the surface of each individual parking lot. Existing sensors, such as ground based parking sensors costs up to $200 per parking lot . As consequence, smart-parking technology using wireless sensors for outdoor parking is costly due to the large number of sensors units required to cover the entire parking lot . Although, parking occupancy monitoring systems have made a significant progress, smart parking payment is rarely studied in smart parking research . Yet, there are companies working on the patents of parking systems for payments. A first approach consists in using a camera or an RFID transceiver for vehicle detection and identification . A limitation of this solution lies in that the system is complex and its implementation is expensive when a detection device is installed on each parking lot. Furthermore, when only RFID transceiver is used for vehicle detection and identification, the system can be bedeviled by electromagnetic interference, which affects the accuracy. Moreover, this system is designed to detect a vehicle when entering a parking and seek payment, whereas information on vacant parking lots is not provided. A technique for monitoring vehicle parking using one camera to record the entrance of a vehicle and a second camera to record the vehicle leaving the parking has been proposed . Moreover, in a system and method for obtaining and displaying information on vacant parking space is described. When a user occupies a parking space designated with an individual ID, he enters this ID into a parking meter or via a smart phone mobile app., and pays the parking fees. The database processes the received data and changes the status of the parking space with its ID from unpaid to paid. These data are used as information on the occupation of a parking space. In this paper, we propose a smart sensor system allowing outdoor parking monitoring and payment without requiring any user/driver interaction. It will be deployed without having to install new components on each parking lot. The proposed sensor has benefits in terms of detection and payment reliability, and reduced expense by reducing the system complexity and installation, and extending batteries lifetime through the reduction of the system power consumption



**Proposed system architecture; wireless occupancy sensor; wireless gateway; data storage and processing unit.**

SYSTEM REQUIREMENTS SPECIFICATION

Functional Requirement Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs/or conditions. These may include calculations, data manipulation and processing and other specific functionality. In these systems following are the functional requirements

• The application should not display in-appropriate message for valid conditions.

• The application must not stop working when kept running for even a long time.

• The application should process information for any kind of input case.

• The application should generate the output for a given input test case .

**Non-Functional Requirement**

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours.

Given below are the non-functional requirements:

• Product requirements

• Organizational requirements

• Basic operational requirements

**Hardware Specifications**

• ENODE MCU (ESP8266)

• JUMPER WIRES

• INFRARED SENSORS

• 16\*2 LED DISPLAY

• DC MOTOR

**Software Specification**

• ARDUINO IDE

**INITIALIATION:**

* Functional Requirement Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs/or conditions. These may include calculations, data manipulation and processing and other specific functionality. In these systems following are the functional requirements

• The application should not display in-appropriate message for valid conditions. • The application must not stop working when kept running for even a long time

• The application should process information for any kind of input case.

• The application should generate the output for a given input test case .

**Non-Functional Requirement**

* Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. Given below are the

**Non-functional requirements**:

• Product requirements

• Organizational requirements

• Basic operational requirements

**Hardware Specifications**

• ENODE MCU (ESP8266)

• JUMPER WIRES

• INFRARED SENSORS

• 16\*2 LED DISPLAY

• DC MOTOR

**Software Specification**

• ARDUINO IDE

**EXECUTION:**

**SYSTEM ANALYSIS AND DESIGN**

**Node Mcu**

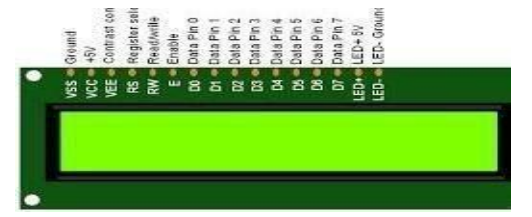
* The Node MCU as shown in has assimilated TCP/IP protocol that can give any microcontroller entrance to the Wi-Fi network that supports 2.4 GHz Wi-Fi (802.11 Wi-Fi standards). Node MCU is capable of either connecting to an existing wireless connection or hosting an application over http protocol. Each Node MCU module comes pre-programmed with an AT command set firmware which means one can simply link this up to your Raspberry Pi device and get about like Wi-Fi shield. The reason why we use node mcu is that it is more cost-efficient with respect to Arduino uno , in Arduino we have to use ethernet shield which provides us secure ethernet connectivity whereas all these features are provided by node mcu and it also comes with a updated feature of wi-fi , where you can power or connect your system by WiFi .



**Node MCU Module**

**16\*2 LCD Display**

* An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in DIY’s and circuits. The 16×2 translates o a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5×7 pixel matrix. The 16\*2 display is used to display the number of vacant and spilled spot . It also gets updated on the display LCD when a vehicle parks or unparks the vehicle .

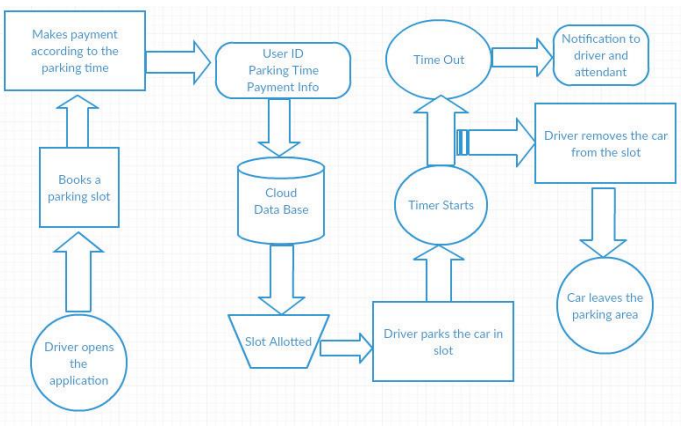
****

IR sensor An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. Infrared radiation was accidentally discovered by an astronomer named William Herchel in 1800. While measuring the temperature of each color of light (separated by a prism), he noticed that the temperature just beyond the red light was highest. IR is invisible to the human eye, as its wavelength is longer than that of visible light (though it is still on the same electromagnetic spectrum). Anything that emits heat (everything that has a temperature above around five degrees Kelvin) gives off infrared radiation. We are using three IR detect sensor in our project , one IR detect sensor is used to sense the vehicle near the parking sensor and other two IR detect sensor is used to send data to the node mcu which is the brain of our system whether a vehicle is parked in that slot or is unparked .

****

* System Architecture The below diagram shows the pin diagram of our model. It consists of one node mcu , one dc motor , one 16\*2 LCD display and three IR sensors .The node mcu is the brain of our system which powers all the other devices .The 16\*2 LCD display is powered by node mcu by connecting jumper wires from the display to node mcu . The DC motor is also powered by node mcu with connecting its pins to node mcu. The IR sensor consists of three pins, where two pins refer to the power supply and ground and the other pins refer to the pin which is going to be connected in the Node mcu. On successfully connecting all the components in the given figure now we have to connect the blynk app. While using the blynk app we have to specify the widgets used in our android app and the pin number to which they are connected to node mcu in the actual model so that the mobile app will react exactly to the inputs provided in the model .

**IMPLEMENTION**

****

* Below are the steps that a driver needs to follow in order to park its car using our parking system.

Step 1: Install the smart parking application on your mobile device.

Step 2: On the 16\*2 display the number of vacant and filled spots are displayed so that the user can see the status of parking zone.

Step 3: Once the user logs into the app he would see the parking architecture with the cars filled at which position and positions which are empty .

Step 4: When the user is near to the parking IR detect sensor , he would receive a notification on his app on which slot he can park his vehicle if there is a empty slot.

Step 5: If there is no empty slot the user will be displayed with an appropriate message on the mobile application .

Step 6: On availability of parking area and user parking into the respective slot he/she would receive a message which states the start time of the parking and the slot in which he/she has parked.

Step 7: On successfully un-parking your vehicle from the parking slot the user will receive a message which states the start time and end time of his parking time and an amount which he needs to pay for the parking duration.

**Design of the System**

The picture shows the miniature model of the Automated Car Parking Lot.



**Experimental Setup**

* This model has the capacity of containing two cars. There are two sensors at the entrance to detect the presence of a car before going inside or outside of the parking lot. The other two sensors are plotted inside the parking lot to detect the car individually for each parking slot. A DC Servo motor has been used at the entrance to open and close the gate according to the signals sent by the sensors through Arduino.

* The projection on the screen corresponds to the system model parking slots. This is a real time display regarding the status of the parking lot. As this is a web-based representation, anyone will be able to get the status of the parking lot by visiting the website on the URL through their cell phones, laptops, desktops and other internet supporting devices. The model of the parking lot has two parking slots. Thus, we can park a maximum number of two cars through the system.

* We have used two IR sensors which when vehicle parked will show appropriate message to the user and when all the parking slots the dc motor would not open gate for the vehicle to be parked. Displaying of appropriate message for any action which takes place in the parking zone is done effectively and efficiently

**Network Time protocol :-**

* The Network Time Protocol is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks. We have used NTP for fetching time from the NTP server so that we can show the start time and end time for the user when he parks or unparks his vehicle making information real-time.

**Blynk app:-**

* Blynk app is a hardware-agnostic IoT platform with white label mobile apps, private cloud ,device management, data analytics and machine learning .On using the blynk app we tried to pop notification to every possible event that is occurring in the parking zone .
* Used a serial algorithm to display the slot number to the user who is going to park his vehicle .For example we display the empty slot number in a serial manner which gets filled , if the slot 1 is filled and when an another vehicle turns up we display slot 2 and further like these for all other vehicles , and if any vehicle leaves the slot number then we display the earlies slot number , not making the user to travel long if an initial spot is vacant .

**Controling:**

* Embedded systems are growing rapidly with the ability to provide high-speed information exchange between devices in communication systems. It has been one of the most common subjects of study since the 1960s. In the past ten years, the world has witnessed a technological revolution by access to Embedded systems and the Internet of Things (IoT) in many fields. This has reflected great development in our daily life qualities, services, and gives high efficiency.
* Modern embedded systems are often based on microcontrollers. That is the reason why the embedded system is dedicated to specific tasks. Besides, design engineers can optimize embedded systems to reduce the size and cost of the product and increase the reliability and performance.
* Recently, a lot of studies and researches on smart cities have been monitored with the expansion of the concept of the IoT, the widespread of what is known as the cloud, how to benefit from it in storing and sharing the data accurately and high quality, and creating the databases. Which made life easier and more civilized.
* The IoT applications have a great impact on communication system everywhere. it can remotely operate and analyze data, as well as communicate with many around sensors devices and managed complex systems, via controlling them through mobile systems applications such as IOS, and Android.
* One of the most important examples of the Internet of Things applications, smart parking, allow drivers to park their vehicle easily and quickly. It reduces traffic congestion and thus reduces environmental pollution resulting from car exhaust.

* The smart parking based on authorized access is proposed in this work, which allows entering parking via ID card working with the Radio Frequency Identification (RFID) device placed at the gate of parking.
* In this work, all sensors are connected to the microcontroller using Arduino Mega 2560 to contains enough input/output pins. To send data, the number and location of empty places to the cloud, a Message Queuing Telemetry Transport protocol (MQTT) through the WiFi of the Nodemcu esp8266, have been used.
* The mobile application has been implemented on the Android system to receive the data from the JavaScript Object Notation database (JSON), which stored in the cloud, which display all information on a screen to allow users park their vehicle easier and faster. This paper is organized as follows, introduction in section one, related work in section two, description of the equipment that was used in this work in section three, algorithm and operating system in section four, conclusion, future work, and finally acknowledgments.

**Description of equipment:**

In this section the equipment that is used in this work includes are considered:

* Arduino Mega 2560.
* It's an open-source physical measure description based on a direct microcontroller board that includes an ATmega328 microcontroller and an improved environment for writing software for the board. Arduino is used to connecting and communicating between various devices.
* Arduino Mega 2560 is one of the most famous boards for microcontrollers ATmega 2560, with many input/outputs pins (54 digital pins and 16 analog pins, using 16 MHz with local crystal) to implement large projects that contain sensors, power supply, switches as input while outputs that control the various devices such as lighting, sound …etc. as shown in Figure1. Also any projects that rely on the Arduino can be accomplished by the computer through different programs.
* Nodemcu esp8266
* There are a lot of devices that are used with the embedded systems and the Internet of Things (IoT) with various features, NodeMCU is a low-cost open-source IoT platform of the microcontrollers with builtin a Wi-Fi 802.11 support b/g/n.
* As shown in Figure 1. This module comes with a built-in USB connector and a rich assortment of pin-outs. With a micro USB cable, it can use to publish or respond in the database, which uses the MQTT protocol, MQTT is more accurate and requires less time to complete the transmission.
* From the above, Nodemcu using in many applications, IoT devices, low power battery-operated applications, network projects. Projects requiring multiple I/O interfaces with Wi-Fi and Bluetooth functionalities.

**CLOSING:**

* IoT has many different applications, but one of the most exciting is its use in smart parking. IoT-based parking systems are able to better track the availability of parking spots on a given lot, making it easier to find an available parking spot.
* It is important to note that not all IoT-based parking systems are the same. For example, some use QR codes to identify available parking spots, while others use sensors to detect when a car leaves a parking spot. The benefits of an IoT-based smart parking system are that it is more creative, efficient, and convenient for both drivers and owners of the parking lot.

**INITIALIATION:**

* + Functional Requirement Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs/or conditions. These may include calculations, data manipulation and processing and other specific functionality. In these systems following are the functional requirements
* The application should not display in-appropriate message for valid conditions. • The application must not stop working when kept running for even a long time
* The application should process information for any kind of input case.
* The application should generate the output for a given input test case .

**Non-Functional Requirement**

* Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. Given below are the

**Non-functional requirements**:

* Product requirements
* Organizational requirements
* Basic operational requirements

**Hardware Specifications**

* + ENODE MCU (ESP8266)
* JUMPER WIRES
  + INFRARED SENSORS
  + 16\*2 LED DISPLAY
* DC MOTOR

**Software Specification**

• ARDUINO IDE

**EXECUTION:**

**SYSTEM ANALYSIS AND DESIGN**

**NODEMCU**

**Node Mcu program for smart parking system**

* Include the necessary libraries #include <ESP8266WiFi.h> #include <WiFiClient.h> #include <ESP8266HTTPClient.h> #include <Wire.h>

#include <LiquidCrystal\_I2C.h>

* Define the pins for the ultrasonic sensor and servo motors #define TRIG D1

#define ECHO D2

#define SERVO\_ENTRY D4 #define SERVO\_EXIT D5

* Define the Wi-Fi credentials

#define SSID "YOUR\_SSID"

#define PASSWORD "YOUR\_PASSWORD"

// Define the API URL for sending the parking status

#define API\_URL "https://YOUR\_API\_URL.com/parking/status"

* Create an instance of the LiquidCrystal\_I2C library LiquidCrystal\_I2C lcd(0x27, 16, 2);
* Create an instance of the Servo library

Servo servoEntry;

Servo servoExit;

// Define the parking status variables

int parkingStatus[3]; // 0 = empty, 1 = occupied

int previousParkingStatus[3]; // Used to detect changes in the parking status

void setup() {

* Initialize the serial port Serial.begin(115200);
* Connect to the Wi-Fi network WiFi.begin(SSID, PASSWORD);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.println("Connecting to Wi-Fi...");

}

Serial.println("Connected to the Wi-Fi network");

* Initialize the ultrasonic sensor pinMode(TRIG, OUTPUT); pinMode(ECHO, INPUT);
* Initialize the servo motors servoEntry.attach(SERVO\_ENTRY); servoExit.attach(SERVO\_EXIT);
* Initialize the LCD display lcd.init(); lcd.backlight(); lcd.print("Smart Parking System");

}

void loop() {

* Read the ultrasonic sensor to detect the presence of a car float distance = ping();
* Update the parking status

for (int i = 0; i < 3; i++) {

if (distance < 100) { // Car is present

parkingStatus[i] = 1;

} else { parkingStatus[i] = 0;

}

}

* Detect changes in the parking status for (int i = 0; i < 3; i++) {

if (parkingStatus[i] != previousParkingStatus[i]) {

* + Send the parking status to the API HTTPClient http; http.begin(API\_URL); http.addHeader("Content-Type", "application/json");

string jsonBody = "{";

Create the JSON body

jsonBody += "\"slot1\": " + String(parkingStatus[0]) + ",";

jsonBody += "\"slot2\": " + String(parkingStatus[1]) + ","; jsonBody += "\"slot3\": " + String(parkingStatus[2]) + "}"; jsonBody += "}";

http.POST(jsonBody);

// Get the response

int statusCode = http.GET();

if (statusCode == 200) {

Serial.println("Parking status updated successfully"); } else {

Serial.println("Failed to update the parking status");

}

* Update the previous parking status previousParkingStatus[i] = parkingStatus[i];

}

}

* Display the parking status on the LCD lcd.clear();

lcd.print("Slot 1:"); lcd.print(parkingStatus[0]); lcd.print(" Slot 2:"); lcd.print(parkingStatus[1]); lcd.print(" Slot 3:"); lcd.println(parkingStatus[2]);

* Delay for 1 second

delay(1000);

}

// Ping the ultrasonic sensor to measure the distance to an object

**The program explaination**

The NodeMCU code you provided for a smart parking system can be explained as follows

**Set up function:**

* initialize the serial port for debugging
* Connects to the Wi-Fi network using the SSID and password provided
* Initializes the ultrasonic sensor by setting the TRIG pin as output and the ECHO pin as input
* Initializes the servo motors by attaching them to the SERVO\_ENTRY and SERVO\_EXIT pins
* Initializes the LCD display by calling the init() and backlight() functions
* Prints the message "Smart Parking System" to the LCD display

**Loop function:**

* Reads the ultrasonic sensor to detect the presence of a car. If the distance to an object is less than 100cm, the parking slot is considered occupied
* Updates the parking status array based on the ultrasonic sensor readings
* Detects changes in the parking status by comparing the current parking status array to the previous parking status array
* If any changes are detected, sends the updated parking status to the API using the HTTP POST method
* Displays the parking status on the LCD display
* Delays for 1 second before repeating the loop

**Ping function:**

* Sends a pulse to the ultrasonic sensor and measures the time it takes for the pulse to return
* Calculates the distance to the object based on the speed of sound and the time it took for the pulse to return

The API URL and API credentials can be customized to suit your needs. For example, you could use a cloud-based database to store the parking status data, or you could use a self-hosted API.

Once the code is uploaded to the NodeMCU, it will start monitoring the parking status and sending updates

to the API. You can then use the parking status data to develop other smart parking features, such as a mobile app that allows users to find available parking spaces.

**IR sensor programing**

**IR sensor programing**

import RPi.GPIO as GPIO

import time

* Define the GPIO pins for the IR sensor PIR\_PIN = 18
* Define the parking status variables PARKING\_STATUS = []

def setup():

* Set up the PIR sensor pin as an input GPIO.setmode(GPIO.BCM) GPIO.setup(PIR\_PIN, GPIO.IN)
* Initialize the parking status variables for i in range(3):

PARKING\_STATUS.append(False)

def loop():

# Read the PIR sensor pin

pir\_value = GPIO.input(PIR\_PIN)

# If the PIR sensor detects a car, set the corresponding parking status to True

if pir\_value:

PARKING\_STATUS[0] = True

else:

PARKING\_STATUS[0] = False

# Send the parking status to the API

send\_parking\_status\_to\_api(PARKING\_STATUS)

* Delay for 1 second time.sleep(1)

def send\_parking\_status\_to\_api(parking\_status):

* TODO: Implement this function to send the parking status to the API pass

setup()

while True:

loop()

**Program Explaination**

The code you provided is a Python code for a smart parking system using an IR sensor. It can be explained as follows:

**Imports:**

* import RPi.GPIO as GPIO: This imports the RPi.GPIO library, which is used to control the GPIO pins on the Raspberry Pi.
* import time: This imports the time library, which is used to delay the execution of the program.

**Constants:**

* PIR\_PIN: This defines the GPIO pin number that the IR sensor is connected to.

**Variables:**

* PARKING\_STATUS: This is a list of parking status variables, one for each parking space. Each element in the list is a Boolean value, indicating whether the corresponding parking space is occupied (True) or empty (False).

**Functions:**

* setup(): This function sets up the PIR sensor pin as an input and initializes the parking status variables.
* loop(): This function is the main loop of the program. It reads the PIR sensor pin and updates the corresponding parking status variable. It then sends the parking status to the API and delays for 1 second.
* send\_parking\_status\_to\_api(): This function is responsible for sending the parking status to the API. It is a TODO function, which means that you need to implement it to suit your needs.

**Main code:**

**Python**

if \_\_name\_\_ == '\_\_main\_\_':

setup()

while True:

loop()

This code starts by calling the setup() function to set up the PIR sensor pin and initialize the parking status variables. Then, it enters a while loop and repeatedly calls the loop() function. The loop() function reads the PIR sensor pin, updates the corresponding parking status variable, sends the parking status to the API, and delays for 1 second.

**Example documentation:**

This is a Python code for a smart parking system using an IR sensor. It works by reading the

PIR sensor pin and updating the corresponding parking status variable. The parking status is

then sent to an API.

Functions:

* setup(): Sets up the PIR sensor pin as an input and initializes the parking status variables.
* loop(): Reads the PIR sensor pin, updates the corresponding parking status variable, sends the parking status to the API, and delays for 1 second.
* send\_parking\_status\_to\_api(): Sends the parking status to the API. Usage:

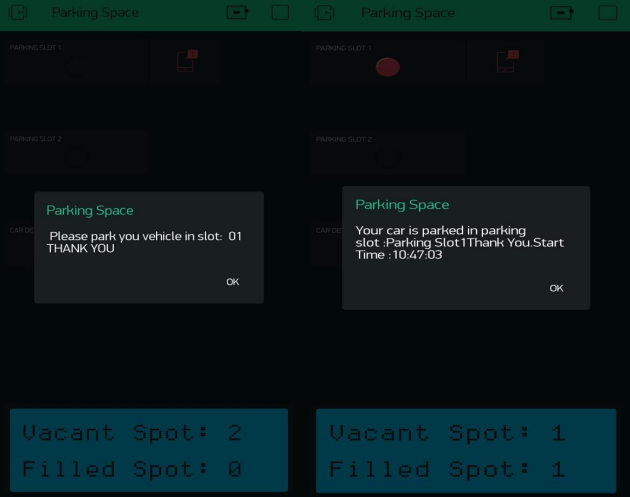
Import the RPi.GPIO and time libraries.

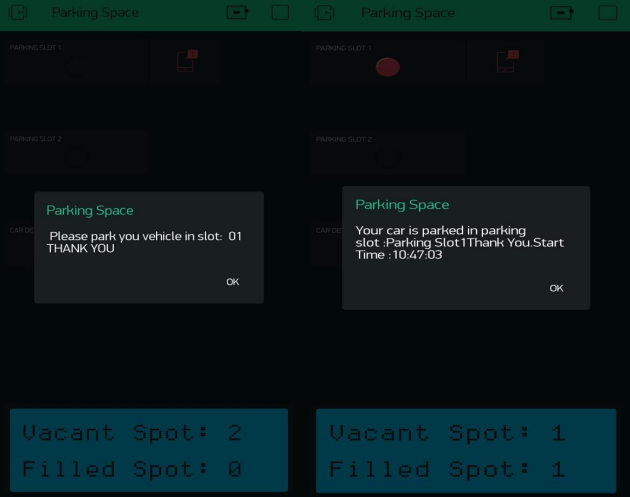
1. Define the PIR\_PIN constant.
2. Create a list of parking status variables, one for each parking space.
3. Implement the send\_parking\_status\_to\_api() function.
4. Call the setup() function.
5. Import the RPi.GPIO and time libraries.
6. Define the PIR\_PIN constant.
7. Create a list of parking status variables, one for each parking space.
8. Implement the send\_parking\_status\_to\_api() function.
9. Call the setup() function.

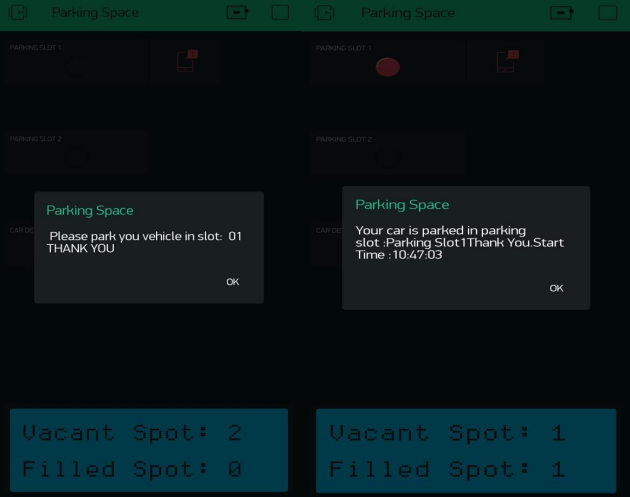
Output:



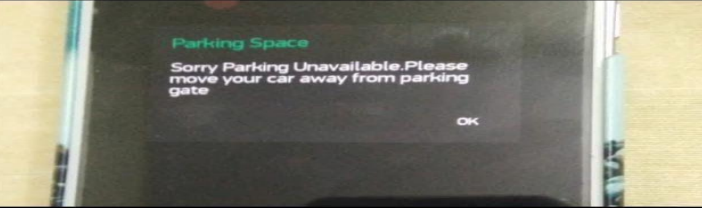












Conclusion

project has done this by using low-cost sensors, real-time data, and applications to allow users to monitor available and unavailable parking spots. This information is then used to guide drivers to available parking spots and to help them pay for parking more conveniently.

The smart parking project has a number of benefits for users, including:

* Reduced time spent searching for parking
* Improved parking efficiency
* Reduced traffic congestion
* Increased convenience
* Potential to save money

The project is also beneficial for businesses and cities. It can help to reduce traffic congestion, improve the efficiency of parking operations, and make cities more attractive to residents and visitors.

THANK YOU