**A SMART APPROACH AND HIGH LEVEL ARCHITECTURE FOR CAR PARKING SYSTEM USING IOT**

A Capstone Phase-II project report submitted

in partial fulfillment of requirement for the award of degree

**BACHELOR OF TECHNOLOGY**

in

**ELECTRONICS & COMMUNICATION ENGINEERING**

by

**G.SUJITH 19K41A04C9**

**P.ARUN KUMAR 19K41A04E3**

**B.RITIKA 19K41A04C5**

**A.RASHMITHA 19K41A04C2**

Under the guidance of

**Mr. B. Girirajan**

Assistant Professor, Department of ECE



**SR ENGINEERING COLLEGE**

Ananthasagar, Warangal.

**

**CERTIFICATE**

This is to certify that this project entitled **“A SMART APPROACH AND HIGH LEVEL ARCHITECTURE FOR CAR PARKING SYSTEM USING IOT”** is the bonafied work carried out by **G. SUJITH**, **P. ARUN KUMAR, B. RITIKA and A. RASHMITHA** as a Capstone Phase-II project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **ELECTRONICS & COMMUNICATION ENGINEERING** during the academic year 2022-2023 under our guidance and Supervision.

**Mr. B. Girirajan Dr. Sandip Bhattacharya**

Assistant Professor Assoc. Prof. & HOD (ECE),

S R Engineering College, S R Engineering College,

Ananthasagar, Warangal. Ananthasagar, Warangal.

**ACKNOWLEDGEMENT**

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Principal, **Dr. V. Mahesh,** for his continuous support and guidance to complete this project in the institute.

We express our thanks to Head of the ECE Department **Dr. Sandip Bhattacharya, Associate Professor,** for his encouragement and support.

We owe an enormous debt of gratitude toour project guide **Mr. B. Girirajan, Assistant Professor, Department of ECE and Head, Collaboratory for Social Innovation** as well as project co-ordinators **Mr.** **S. Srinivas, Asst. Prof., Dr. R. Shashank, Asst. Prof.**, for guiding us from the beginning through the end of the Capstone Phase II project with their intellectual advices and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive, encouraging and ultimately drove us to the right direction.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support

**ABSTRACT**

A simple and easy task such as parking a car is a tedious and time-consuming process due to mismanagement of the parking system. The Current parking system involves huge manpower for management and requires users to search for parking space slot by slot. Such conventional systems utilize more power, along with user’s valuable time. This paper presents an intelligent parking energy management solution for a structured environment such as a multi-storied office parking area. The system proposes the implementation of state of the art Internet of Things (IoT) technology to mold with advanced sensors and controllers to obtain a systematic parking system for users. Unoccupied vehicle parking spaces are indicated using lamps and users are guided to an empty parking space, thus eliminating the need for searching for a parking space. The occupied parking spaces are virtually stored in the cloud to be accessed. The automatically controlled light helps reduce energy usage, lighting up the parking space for the user in the parking space. The entire system being semi-automated leads to reduce manpower involved and improves the aesthetics of the parking area. This paper aims to improve user's time value and convenience in a parking system.

**CONTENTS**

***ACKNOWLEDGEMENT iii***

***ABSTRACT iv***

***LIST OF FIGURES vii***

***LIST OF TABLES viii***

***LIST OF ACRONYMS ix***

**Chapter No. Title Page No.**

**1 INTRODUCTION** 01

1.1 OVERVIEW OF THE PROJECT 01

1.2 EXISTING METHODS 07

1.3 PRESENT WORK 09

1.4 LITERATURE SURVEY 10

**2 Hardware / software tools** 14

2.1 HARDWARE & SOFTWARE COMPONENTS 14

2.2 DESCRIPTION OF HARDWARE & SOFTWARE 14

2.2.1ESP32 MICROCONTROLLER 14

2.2.2 RELAY MODULE 16

2.2.3 SERVO MOTOR 17

2.2.4 IR SENSOR 21

2.2.5 OLED DISPLAY MODULE 22

2.2.6 PUSH BUTTONS 23

2.2.7 USB WIRE 24

2.2.8 ARDUINO IDE 26

2.2.9 MIT APP INVENTOR 28

2.2.10 GOOGLE FIREBASE 29

**3 PROJECT IMPLEMENTATION** 31

3.1 BLOCK DIAGRAM 31

3.2 WORKING OF THE PROJECT 32

**4 SIMULATION RESULTS & ANALYSIS** 40

4.1 RESULTS AND DISCUSSION 40

**5 CONCLUSION & FUTUTRE SCOPE** 43

5.1 CONCLUSION 43

5.2 FUTURE SCOPE 43

**BIBLIOGRAPHY** 44

**PUBLICATION PROOF**  46

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure. No.** | **Figure Name** | **Page. No** |
| 1.1 | High level IoT Architecture | 03 |
| 1.2 | Example of IoT system | 05 |
| 1.3 | Application of IoT | 06 |
| 1.4 | Methodology of proposed system | 09 |
| 2.1 | ESP32 Microcontroller | 15 |
| 2.2 | Relay Pin Diagram | 17 |
| 2.3 | SG90 Servo motor with three wire representation | 19 |
| 2.4 | Input pulse for different angle rotation | 20 |
| 2.5 | IR Sensor | 21 |
| 2.6 | Two-sided image of Oled | 23 |
| 2.7 | Push button | 24 |
| 2.8 | USB cable | 24 |
| 2.9 | Arduino IDE Software | 28 |
| 2.10 | Image of MIT App Inventor page | 29 |
| 2.11 | Activity demonstration for Google firebase | 30 |
| 3.1 | Block diagram of proposed project | 31 |
| 3.2 | Flowchart of the IoT system | 33 |
| 3.3 | Flowchart showing working of entrance and exit of the parking area | 34 |
| 3.4 | Flowchart showing the working of detection and sending of data to the cloud (Firebase) | 36 |
| 3.5 | Flowchart demonstrating the process of application | 39 |
| 4.1 | OLED Display demonstrating the slot availability | 40 |
| 4.2 | Demonstration of login page, registration page, and home page | 41 |
| 4.3 | Demonstration of check slot screen and book your slot screen | 41 |
| 4.4 | Demonstrating of backend data of IR sensors, users credentials, and data of slot booking | 42 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table. No.** | **Table Name** | **Page. No** |
| 1.1 | Power, data rate and ranges of different wireless communication networks | 6 |
| 2.1 | Peripherals of the Esp32 microcontroller | 16 |
| 2.2 | Pin configuration of IR sensor | 22 |
| 2.3 | Pin configuration of OLED | 23 |
| 2.4 | USB Specifications and USB Connections | 26 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF ACRONYMS**

|  |  |
| --- | --- |
| **ACRONYM** | **ABBREVIATION** |
| **DAS** | **D**ata **A**cquiring **S**ystems |
| **ESP** | **E**xpressif **S**ystem **P**rocessor |
| **GSM** | **G**lobal **S**ystem for **M**obile communication |
| **IFTTT** | **I**f **t**his **t**hen **t**hat |
| **IR** | **I**nfrared |
| **LCD** | **L**iquid **C**rystal **D**isplay |
| **MQTT** | **M**essage **Q**ueuing and **T**elemetry **T**ransport |
| **OLED** | **O**rganic **l**ight **e**mitting **d**iodes |
| **RFID** | **R**adio **f**requency **i**dentification |
| **SMS** | **S**hort **m**essage **s**ervice |
| **WHO** | **W**orld **H**ealth **O**rganization |

**CHAPTER 1**

**INTRODUCTION**

**1.1 OVERVIEW OF THE PROJECT**

Imagine a day when a person does no longer have to spend time planning and going in circles to find a parking spot for your car, sounds like a farfetched reality. An average Indian driver spends 20 minutes every day on average to look for a suitable parking spot (as a report by IBM), an incalculable loss to national productivity. Today, finding a parking spot is more tedious than driving a car around in the city. This is an alarming issue, especially in metro cities.

With daily new car registrations of approximately 800 cars in Mumbai and 1500 cars in New Delhi (NCR), the additional daily space required is equivalent to 4 and 7.5 football fields respectively. Especially when per capita open spaces in cities like Mumbai are at 1.28 sq meters as against the 9 sq meter recommended space by WHO. Since there is no new space generation happening for car parking and majority of buildings in the city areas are old, without any provision for parking, the already cramped roads are the natural place to park cars.

Congestion on the roads is caused by the average 40% of cars that are parked there. According to a 2018 study by BCG, it is estimated that just in the four metro cities, congestion costs us 1.47 crores of rupees annually. The estimated loss due to traffic congestion would surpass India's healthcare budget if we simply double this amount to account for the rest of the nation and include fuel inflation. Another difficulty is the constant financial strain it places on our town planners to build flyovers, subways, etc. Parking needs will skyrocket as urbanization and motorization accelerate. It's interesting that there are now technological solutions to meet parking needs, from traditional ramp types to fully automated types. Automated Multilevel Car Parking Systems (AMLCP) has been around for more than 50 years worldwide and for the past 20 years in India. They aid in vertical car parking and are mechanically operated. In comparison to land and construction, it can enable efficiencies like none other at a significantly lower cost. Automated multilevel parking systems not only save a lot of land but are a sustainable choice. Efficiency gains will reach two to twenty times is possible, allowing for the parking of nearly 60 cars in the space previously occupied by three, greatly expanding the available space. An automated car parking system can create parking space for 50 cars in the same area where a conventional parking system can only accommodate 3 cars.

There are 7.6 billion people living in the globe, and 1.2 billion of them have their own cars parked on the street. The need for automobiles increases along with the population. Intensity of traffic is necessary given a city's size, employment, and population growth.

Numerous accidents are caused by clogged up roadways. Only a small percentage of all traffic accidents have predominantly involved people between the ages of 15 and 44, which is the last year for people to make imaginative purchases. Nearly 1% of the 61 persons were classified as "taking the money to cover the cost of supporting their losses,” and the majority of the time, impoverished people from low socioeconomic background die in traffic accidents. The World Bank estimates that the cost of traffic accidents is between one and two percent of the GDP of developing nations, or almost twice the total amount of development aid that these developing nations have ever received.

In the developing countries, there are many parking buildings locatedaround the capital cities. For example, GVK mall in Hyderabad has huge underground parking space. This building has no intelligence and operated manually by cashiers and security guards. By conducting surveys and interviews the users at gvk mall, i.e., visitors, mall members and staff who drive and park at the mall, the team realizes that the main user’s pain point is to search for an available parking spot. That is, it takes more than 15 minutes to drive inside a parking building to realize that there is no parking spot available. In addition, for a vehicle to enter the mall parking, a parking sticker is required, which is manually checked by a cashier who looks for the sticker at the front windshield of a vehicle. Hence, the following problem statement is addressed: The current parking system at the famous malls is not facilitated with information about vacancies of parking lots to the user. The proposed system gives the information about the vacancies in the display/ APP or websites. IoT technology creates a more convenient environment for visitors, vendors of the Mall and other staff. It allows users to make reservation for a parking spot before arriving at a parking building, improve the parking sticker system and minimize the parking manual operations. Since a smart parking system is quite well-known, lessons learned from existing projects can be very well serving as the basis to better design and improve our project.

An essential part of every transportation system is parking. Early human settlements have left a rich legacy on cities and transportation, and popular regions experience heavy traffic congestion. This is due to the excessively long journey times and the high fuel consumption, which has an adverse effect on inhabitant’s socioeconomic well-being and contributes to environmental issues like pollution and a rapid rise in atmospheric carbon dioxide levels. Drivers who don’t care about the people nearby make improper turns just as safely as they stop. Driving too quickly in crowded locations leads to disputes between individuals. These road explosions, whether they’e caused by passing another car in a rage or for pleasure, result in several unnecessary automobile accidents every year. These negative sentiments can be averted if a company makes an effort to assist drivers in finding available parking spaces when they need them.

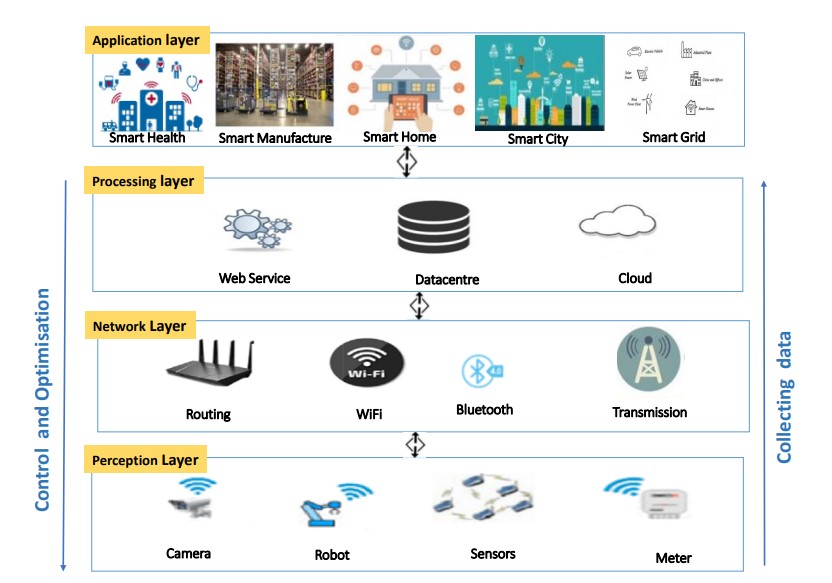


Fig 1.1 High level IoT Architecture

The new technology incorporated into these systems will increase their efficiency in managing parking and assist in preventing traffic jams and all of the negative consequences that come along with them, including time spent looking for and retrieving your car from the parking lot, environmental degradation, and noise and air pollution.

All the problems that people in big cities are currently dealing with can be solved with a parking system that is digitally equipped. With the current system, users can independently navigate every step of the parking process, from knowing the status of the parking lot's occupancy to receiving tickets, parking, and fees. Multiple automated car parking systems can be built in smart cities to take advantage of available space, and parking spaces can be reserved using apps or other technologically assisted systems. A user can receive real-time updates about a suitable parking spot nearby thanks to the digital parking system. It primarily aims to automate the time it takes a driver to physically find a parking space. The user benefits from increased convenience, which eliminates parking issues and traffic congestion. Automated car parking systems aim to significantly improve the ease of vehicular parking systems with a mission to optimize space and mobility in all the cities of India and the rest of the world by leveraging advanced analytics, experience engineering, and newer business models.

Basic security services such as confidentiality, integrity, authentication, availability and non repudiation should be appropriate to guarantee IoT protection. However, IoT devices are constrained by their power and memory. For example, objects and devices communicate together in a complicated way by using different security mechanisms. Therefore, security difficulties have become a concern when employing an appropriate security strategy that considers all those limitations. Even though there have been numerous surveys in this field, these surveys as outlined in the following sections are limited and primarily focused on either specific IoT security aspects, attacks, layer vision or provide a limited evaluation of the implemented security solutions. Our major contributions include the integrity of the industrial IoT systems, which has been highlighted along with the case study discussing in the existing solutions.

The internet of things describes the network of physical objects that are embedded with sensors, software and other technologies for the purpose of connecting and exchanging data with other devices and system over the internet. This term “Internet of Things” is coined by Kevin Ashton in 1999.

The applications of IoT are:

* Home automation
* Medical and healthcare systems
* Smart manufacturing
* Energy resource management
* Agriculture Management System.
* Transport systems etc

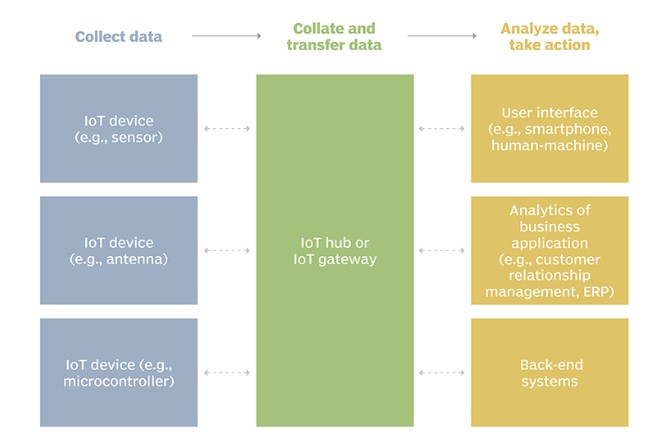


Fig 1.2 Example of IoT system

**Pros and Cons of IoT**

Some of the advantages of IoT include the following:

* Ability to access information from anywhere at any time on any device
* Improved communication between connected electronic devices
* Transferring data packets over a connected network saving time and money
* Automating tasks helping to improve the quality of a business’s services and reducing the need for human intervention.
* The cons of IoT include complexity, compatibility, privacy concerns, lesser employment, and technology addiction.

Some of the disadvantages of IoT include the following:

* As the number of connected devices increases and more information is shared between devices, the potential that a hacker could steal confidential information also increases.
* Enterprises may eventually have to deal with massive numbers maybe even millions of IoT devices, and collecting and managing the data from all those devices.
* If there is a bug in the system, it’s likely that every connected device will become corrupted.
* Since there’s no international standard of compatibility for IoT, it’s difficult for devices from different manufacturers to communicate with each other.



Fig 1.3 Application of IoT

This project focuses on IoT Technology which is a part of wireless communication. In IoT technology Wifi, Infrared communication and Bluetooth are adequately used. Below table represent the power, range and data rate used by different wireless communication networks.

|  |  |  |  |
| --- | --- | --- | --- |
| **Wireless communication networks** | **Power** | **Range** | **Data rate** |
| Wifi 5 and 6 | Moderate | Moderate to long | High |
| Bluetooth | Very low | Short | Low |
| Infrared | Low | Short | High |

Table 1.1 Power, data rate and ranges of different wireless communication networks

When compared among the three networks Wifi has best features so it is used to implement our project to send and receive data from the application or website.

**Benefits of IoT:**

The internet of things offers several benefits to organizations. Some benefits are specific, and some are applicable across multiple industries. Some common benefits of IoT are:

* Reduced costs.
* Improved operational efficiency.
* Data-driven insights for quick decision-making.
* End-to-end, remote monitoring and management of assets/resources.
* Real-time, predictive and prescriptive insights.
* Improve end-customer experience.
* Real-time asset/resource visibility.
* Reduce electricity

**1.2 EXISTING METHODS**

Concerning the “IoT based smart parking system” project, they talked about the issues with conventional parking lots. Additionally, it covers the effects and inconveniences brought on by the inefficiency of conventional parking places. In this project, the team proposed and created a smart parking system that makes use of IOT technology and enables users to locate open parking spaces in a certain location. Additionally, it prevents unnecessary movement through already crowded parking lots. The team members in this project offer a cutting-edge parking system using IoT via Wi-Fi and RFID. The participants propose a smart phone app, IR sensors, RFID, and Arduino as the main elements of an IOT-based solution to the problem. The project’s drawback is that it only considered residential and business structures [1]. They connected an esp32 Nodemcu with a weighbridge load sensor that detects weight and signals the presence of the car to create “An IoT based Smart Outdoor Parking System”. Green LED is used to depict an empty lot, while red LED is used to indicate the presence of a car. To pre-book the spot and to view parking fees, this system also features an online application. Since it is an automated system, nobody else can ask for assistance. However, it has the problem of indicating the presence of a vehicle even if there is no other mass present, such as when a vehicle is absent [7]. It is suggested to develop an automated real-time system for automated vehicle parking called “IoT Based Smart Vehicle Parking System Using RFID”. The internet of things (IOTs) has assisted this approach. IOT often involves the exchange of data or information between two physical devices. A microcontroller called Arduino Uno is utilized in the suggested system. The primary function of Arduino in the suggested system is to operate as a communication platform for interactive items and digital gadgets that can perceive and control real-world machinery. The Arduino Uno board has been used to create the suggested system, and Node MCU has been used to link the parking lot to the internet or the online. Utilizing Nodemcu and IBM Cloud, the “Smart Parking System using MQTT Communication Protocol and IBM Cloud” project concentrated on how to automate the parking process by monitoring parameters like distance and available parking spots. The distance is determined, and the data is transmitted over the MQTT protocol to Node-RED. The user may monitor availability on the Node-RED dashboard from any location. The area is empty if the distance is too large. The owner or person in charge of the parking lot is also informed if the space is completely occupied. IFTTT and Node-RED are combined to do this. Watson is a virtual assistant that offers guidance to customers on a range of issues. It lacks adequate emergency parking instructions [5]. In “Comparing Biometric and Block chain Security Mechanisms in Smart Parking System,” they contrasted the two main security systems used in the system. Block chain technology is a decentralized architecture with full trust and authentication that uses a distributed computer paradigm. While a biometrics system is used to verify a user’s identity using behavioral and physical data collected for both identification and access control. Comparative analysis of these two methods helps to comprehend communication level tiers and provide security to various security setups. Only a few office and residential buildings may utilize this system, and it cannot be used in large malls or public spaces since the data cannot be kept there. It is also expensive and requires a lot of storage space [8]. They concentrated on how IoT may be utilized to tackle issues like traffic congestion and road safety. In this project, the participants use IoT via Wi-Fi to propose a smart parking system. The IoT module in this intelligent parking system aids in tracking the availability of each individual open parking space. The author connected to the internet using an Arduino Uno, which may be integrated over a Wi-Fi module. The data is sent live thanks to this technology. Digital IR sensors are used in this smart parking system to provide information on the status of the parking lot, including whether it is used or unoccupied. The microcontroller receives the data gathered by this sensor. After processing the data, the central database is updated with the

state of the parking lots. In order for the system to cover all parking lots, the IR sensors must be placed where they are needed. A unique id is assigned to each parking lot to help identify it on the network. However, this project's implementation costs are expensive, and it is also challenging [2].

**1.3 PRESENT WORK**

The Proposed Project is developed with ESP32 microcontroller with IR sensors and other actuators. This project is being carried out to save time, make parking simple and lower power costs. The parking area’s data is updated in the firebase cloud, which benefits both users and parking administration. Users may check their available slots using an application and pre-book the session at their preferred time. The suggested solution may be divided into two parts: Software and Hardware

The IR sensor is built into the hardware to check for vehicle availability and is updated to the cloud. A timer and the slots availability are displayed on a display. The bulb is controlled by a circuit in order to consume less power. The entry and exit are maintained using a servo and a button.

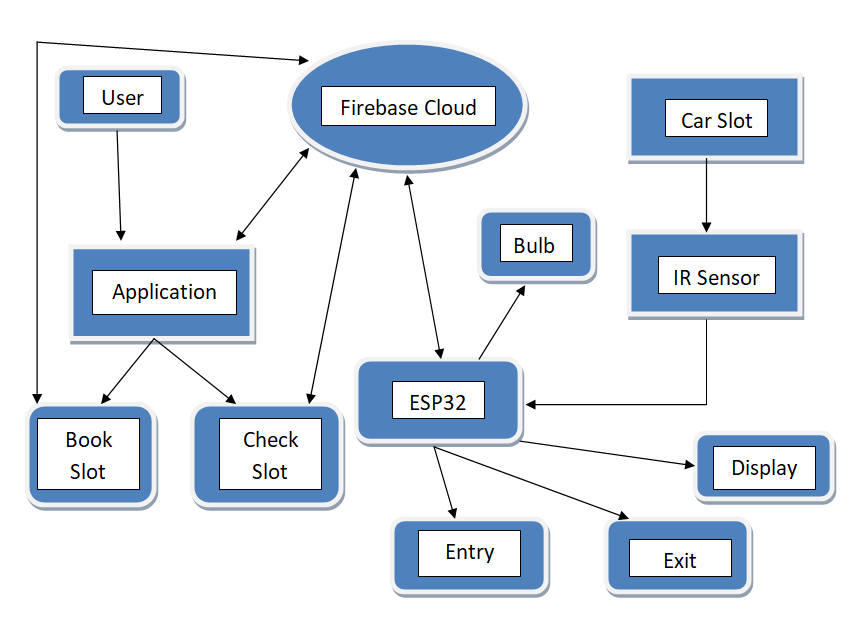


Fig 1.4 Methodology of proposed method

**1.4 LITERATURE SURVEY**

Y. Agarwal *et.al* [1] proposed a project “IoT based smart parking system” they discussed problems faced with traditional parking lots. It also lists the impact and inconvenience caused because of inefficiency in traditional parking spaces. In this project, the team have suggested and designed a Smart Parking System using IOT Technology, which will allow the users to find a vacant parking lot in a given area. It also avoids needless traveling through already filled parking lots. In this project, the team members present a novel parking system with IoT over Wi-Fi and RFID. The members suggest an IOT based solution to the issue using a mobile app, IR sensors, RFID, and Arduino as key components. The disadvantage of this project is that they only focused on residential buildings and official buildings.

S. P. Patro *et.al* [2] proposed a project “IoT based Smart Parking System: Proposed Algorithm and Model”. They focused on problems such as traffic congestion and road safety are being solved by the use of IoT. In this project, the members present a smart parking system with the help of IoT over Wi-Fi. This intelligent parking system consists of an IoT module that helps to track the availability of each single vacant parking space. The author used an Arduino Uno, which can be embedded over the Wi-Fi module to establish a connection to the internet. This technology helps to transfer the data live. In this smart parking system, with the help of digital IR sensors, the system gets the status regarding the parking lot status, whether it is occupied or vacant. This sensor sends the collected data to the microcontroller. Latter the data are processed, and the status of parking lots is updated in the central database. The IR sensors need to be deployed in the appropriate locations so that the system can cover all the parking lots. Each parking lot is identified with a unique id to identify them on the network. But this project implementation cost is high and it is also difficult to implement.

M. P. Thakre *et.al* [3] proposed a project “IoT Based Smart Vehicle Parking System Using RFID” an automatic real-time system for been implemented automated Vehicle parking is proposed. This system has with the help of the internet of things (IOTs). IOT generally exchanges information or data between the two physical devices. Arduino Uno is a microcontroller used in the proposed system. The main use of Arduino in the proposed system is to provide a platform to communicate digital devices and interactive objects that can sense and control physical devices. The proposed system has been implemented with the help of Arduino Uno board for vehicle parking and Node MCU to connect parking area with web or internet. The proposed system incorporated an infrared sensor in each lot for getting information about the vacancy position of the parking lot. The user book-parking lot well in advance, all the necessary information is available on the server. Every user has an exclusive username and password. In case any misuse happened then the system will alert the responsible person. It is very costly and uses large power consumption.

S. Ravishankar *et.al* [4] proposed a solution “Cloud connected smart car park” which is costly to indicate the number of free parking lots in a given parking area. This is implemented using infrared sensors in every bay which are then connected to a Raspberry Pi. The raspberry Pi transfers all the data to a cloud server, which is accessible to users using a mobile application. This system is suitable for the only people who subscribe with their servers. So it has a limitation to only for some users.

C. Ashhwath [5] proposed and implemented a project “Smart Parking System using MQTT Communication Protocol and IBM Cloud” he focused how to automate the parking procedure by monitoring metrics such as distance and available parking spaces, Nodemcu and IBM Cloud are used. The distance is measured, and the information is sent to Node-RED over the MQTT protocol. The Node-RED dashboard allows the user to view availability from any location. If the distance is too great, the space is unoccupied. If the parking area is fully occupied, the owner or person in control of the parking lot is also notified. This is accomplished by combining IFTTT and Node-RED. Watson is a virtual assistant that helps consumers with a variety of questions. It does not have proper guidance in emergency parking.

Anitha.G *et.al* [6] proposed a method “Embedded IoT Car Parking and Billing System” which utilizes equipment like Arduino-Mega, Arduino-UNO, Wi-Fi module, LCD to show the stopping openings accessible and booking affirmation, infrared sensors which are utilized at each press stopping and tells the space accessibility, QR code if relevant, and data set applications. The programming language utilized is C to design Arduino and PHP which are utilized for the online interface while MySQL for an information base-worker. By Smart Parking Application, it's relied upon to help to stop administration clients to ask data and discover void stopping openings through booking inside the application all together that it'll be more proficient as expected and stopping the board will be more controlled. The major problem occurred is that the time of parking is less but the amount is shown more because it calculates the time when a user scan the QR.

GokulKrishna. S *et.al* [7] proposed a project “An IoT based Smart OutdoorParking System” they integrated with esp32 Nodemcu with a weighbridge load sensor that detects the weight and tells about the presence of the vehicle. Red LED is used to represent the presence of vehicle and green led is used to represent the vacant lot. This system also has a web application to pre-book the lot and also for charges for parking. It is an automated system so no other person can require for guidance. But it has a drawback that when other then vehicle if any mass is kept than if there is no vehicle available then also it tells the presence of vehicle.

A. Waheed*et.al* [8] proposed a project “Comparing Biometric and Block chain Security Mechanisms in Smart Parking System” in this they have compared two major mechanisms for providing security to smart parking system. Block chain technology is distributed computing paradigm and decentralized architecture with complete trust and authentication. Whereas, biometrics system is used for authenticating user by their behavioral and physical information obtained for the purposes of both identification and access control. These two systems are compared to help comprehend the various communication levels and security procedures at various security levels. This system is suitable for only some official and residential buildings and cannot be used for huge malls and places and the data cannot be stored so it requires huge storage space and it is also costly.

M. S. Mohammed *et.al* [9] proposed a project “Safety Based Smart Parking Guidance System Using GSM and Zigbee” safety based smart car parking guidance system has been designed and implemented using global system for mobile communications (GSM) technology and Zigbee technology. It assists the drivers to overcome parking problems that usually occur in the urban areas, like time being wasted in looking for an available parking lot and keep on circling until they find a vacant parking spot. This system includes three modules: First, monitoring module; the drivers check on their mobile phones for an available parking spot in real time before arriving using short message service (SMS) without having to go online. Second, guidance module; an approach was suggested to determine the nearest vacant parking spots within the parking lot. This approach depends on the location of the vacant parking spot to the parking lot entrance, exit and pedestrian exit location. Third, safety module; the aim of designing this module is to provide a real time system capable of monitoring sudden events in the parking lot. In case of any emergency, the drivers will be alerted automatically through SMS. It is an automatic system. It consumes more power and in emergency times this may not work.

C. Ajchariyavanich1 *et.al* [10] proposed a project “Park King: An IoT-based Smart Parking System” they present the development and prototyping of Park King – an IoT based cloud-integrated smart parking system for a smart campus. Park King consists of: (i) the IoT module that allows monitoring the availability of each parking spot and controlling a parking flap; and (ii) a web-based application that allows users to reserve a parking space in advance. The system overview, its functional and non-functional requirements, tools and technologies used, prototype development/deployment, together with results from field testing and demonstration, are discussed in this paper. It is expected that this system can serve as a guideline and provide an insight into the development of a smart parking system in a university campus and/or a smart city. It does not focus on lots availability at emergency and it consumes high power as it has lights for every lot.

**CHAPTER 2**

**Hardware / software tools**

**2.1 HARDWARE & SOFTWARE COMPONENTS**

**2.1.1 Hardware components**

1. ESP32 Microcontroller
2. Relay module
3. Servo motor
4. IR sensor
5. OLED Display module
6. Push buttons
7. USB Wire
8. Breadboard

**2.1.2 Software components**

1. Arduino IDE
2. MIT app Inventor
3. Google Firebase

**2.2 DESCRIPTION OF HARDWARE & SOFTWARE COMPONENTS**

**2.2.1 ESP-32 Microcontroller**

ESP32 is the SoC (System on Chip) microcontroller which has gained massive popularity recently. With the above specifications in front of you, it is very easy to decipher the reasons for ESP32's popularity. Consider the requirements an IoT device would have from its microcontroller (μC). Therefore, to begin with, the μC should be able to interface with a variety of sensors. It should support all the common communication protocols required for sensor interface: UART, I2C, SPI. It should have ADC and pulse counting capabilities. ESP32 fulfills all of these requirements. On top of that, it also can interface with capacitive touch sensors. Therefore, most common sensors can interface seamlessly with ESP32.

Secondly, the μC should be able to perform basic processing of the incoming sensor data, sometimes at high speeds, and have sufficient memory to store the data. ESP32 has a max operating frequency of 40 MHz, which is sufficiently high. It has two cores, allowing parallel processing, which is a further add-on. Finally, its 520 KB SRAM is sufficiently large for processing a large array of data onboard. Many popular processes and transforms, like FFT, peak detection, RMS calculation, etc. can be performed onboard ESP32. On the storage front, ESP32 goes a step ahead of the conventional microcontrollers and provides a file system within the flash. Out of the 4 MB of onboard flash, by default, 1.5 MB is reserved as SPIFFS (SPI Flash File System). Think of it as a mini−SD Card that lies within the chip itself. You can not only store data, but also text files, images, HTML and CSS files, and a lot more within SPIFFS. People have displayed beautiful Webpages on WiFi servers created using ESP32, by storing HTML files within SPIFFS.

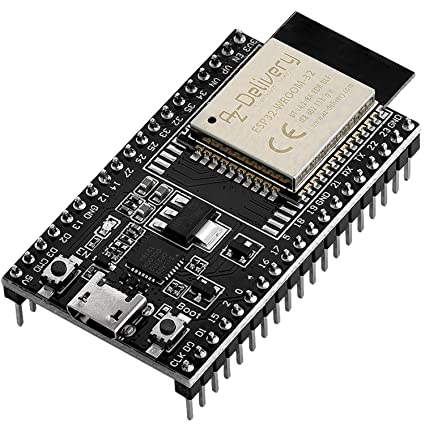


Fig 2.1 ESP32 Microcontroller

Finally, for transmitting data, ESP32 has integrated WiFi and Bluetooth stacks, which have proven to be a game-changer. No need to connect a separate module (like a GSM module or an LTE module) for testing cloud communication. Just have the ESP32 board and a running WiFi, and you can get started. ESP32 allows you to use WiFi in Access Point as well as Station Mode. While it supports TCP/IP, HTTP, MQTT, and other traditional communication protocols, it also supports HTTPS. It has a crypto−core or a crypto-accelerator, a dedicated piece of hardware whose job is to accelerate the encryption process. So you cannot only communicate with your web server, you can do so securely. BLE support is also critical for several applications. Of course, you can interface LTE or GSM or LoRa modules with ESP32. Therefore, on the ‘transmitting data’ front as well, ESP32 exceeds expectations.

|  |  |
| --- | --- |
| 15 ADC channels | 15 channels of 12-bit SAR ADC with selectable ranges of 0-1V, 0-1.4V, 0-2V, or 0-4V |
| 2 UART interfaces | 2 UART interfaces with flow control and IrDA support |
| 25 PWM outputs | 25 PWM pins to control things like motor speed or LED brightness |
| 2 DAC channels | Two 8-bit DACs to generate true analog voltages |
| SPI, I2C and I2S interface | Three SPI and one I2C interfaces for connecting various sensors and peripherals, as well as two I2S interfaces for adding sound to your project |
| 9 Touch Pads | 9 GPIOs with capacitive touch sensing |

Table 2.1 Peripherals of the Esp32 microcontroller

**2.2.2 Relay Module:**

Relay is one kind of [electro-mechanical component](https://www.elprocus.com/electromechanical-relay-working-with-applications/) that functions as a switch. The relay coil is energized by DC so that contact switches can be opened or closed. A single channel 5V relay module generally includes a coil, and two contacts like normally open (NO) and normally closed (NC). This article discusses an overview of the 5V relay module & its working but before going to discuss what is [relay](https://www.elprocus.com/relay-circuit-with-working/) module is, first we have to know what is relay and its pin configuration. A 5v relay is an automatic [switch](https://www.elprocus.com/what-is-a-centrifugal-switch-and-its-working/) that is commonly used in an automatic control circuit and to control a high-current using a low-current signal. The input voltage of the relay signal ranges from 0 to 5V.

The pin configuration of the 5V relay is shown below. This relay includes 5-pins where each pin and its functionality are shown below.

**Pin1 (End 1):** It is used to activate the relay; usually this pin one end is connected to 5Volts whereas another end is connected to the ground.

**Pin2 (End 2):** This pin is used to activate the Relay.

**Pin3 (Common (COM)):** This pin is connected to the main terminal of the Load to make it active.

**Pin4 (Normally Closed (NC)):** This second terminal of the load is connected to either NC/ NO pins. If this pin is connected to the load then it will be ON before the

switch.

**Pin5 (Normally Open (NO)):** If the second terminal of the load is allied to the NO pin, then the load will be turned off before the switch.

The **features of the 5V relay** include the following.

* Normal Voltage is 5V DC
* Normal Current is 70mA
* AC load current Max is 10A at 250VAC or 125V AC
* DC load current Max is 10A at 30V DC or 28V DC
* It includes 5-pins & designed with plastic material
* Operating time is 10msec
* Release time is 5msec
* Maximum switching is 300 operating per minute

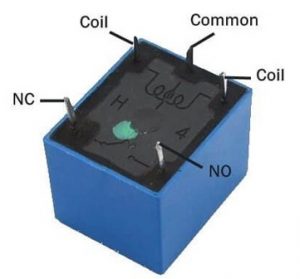


Fig 2.2 Relay Pin Diagram

**2.2.3 Servo motor**

A servo motor is a kind of motor that has extremely precise rotational capabilities. This type of motor often has a control circuit that gives feedback on the motor shaft’s present location. This feedback enables the servo motors to rotate very precisely. A servo motor is used to rotate an item at predetermined angles or distances. It consists of a straightforward motor that drives a servo mechanism. A motor is referred to as a DC servo motor if it is powered by a DC power source, and an AC servo motor if it is driven by an AC power source.

It consists of three parts:

1. Controlled device
2. Output sensor
3. Feedback system

It is a closed-loop system where it uses a positive feedback system to control motion and the final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output signal and reference input signal.

Here reference input signal is compared to the reference output signal and the third signal is produced by the feedback system. And this third signal acts as an input signal to the control the device. This signal is present as long as the feedback signal is generated or there is a difference between the reference input signal and reference output signal. So the main task of servomechanism is to maintain the output of a system at the desired value at presence of noises.

A servo consists of a Motor (DC or AC), a potentiometer, gear assembly, and a controlling circuit. First of all, we use gear assembly to reduce RPM and to increase torque of the motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now the difference between these two signals, one comes from the potentiometer and another comes from other sources, will be processed in a feedback mechanism and output will be provided in terms of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with the potentiometer and as the motor rotates so the potentiometer and it will generate a signal. So as the potentiometer’s angular position changes its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.

Interfacing hobby Servo motors like servo motor with MCU is very easy. **Servos have three wires coming out of them.** Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU.

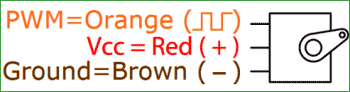
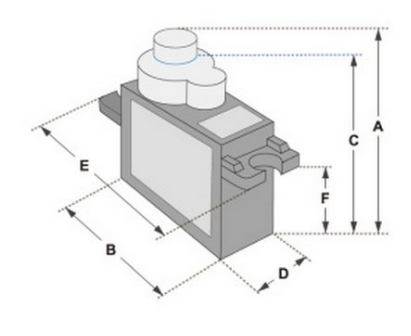


Fig 2.3 SG90 Servo motor with three wire representation

Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degree from either direction from its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°.

Servo motor works on **PWM (Pulse width modulation)** principle means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically servo motor is made up of **DC motor which is controlled by a variable resistor (potentiometer) and some gears.** High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. The potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.

Servo motor can be rotated from 0 to 180 degrees, but it can go up to 210 degrees, depending on the manufacturing. This degree of rotation can be controlled by applying the **Electrical Pulse** of proper width, to its Control pin. Servo checks the pulse in every 20 milliseconds. The pulse of 1 ms (1 millisecond) width can rotate the servo to 0 degrees, 1.5ms can rotate to 90 degrees (neutral position) and 2 ms pulse can rotate it to 180 degree. All servo motors work directly with your +5V supply rails but we have to be careful about the amount of current the motor would consume if you are planning to use more than two servo motors a proper servo shield should be designed.

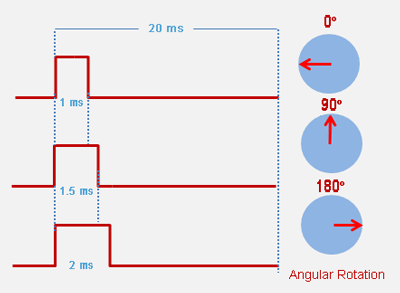


Fig 2.4 Input pulse for different angle rotation

**Specifications:**

* Speed (sec): 0.1
* Torque(kg-cm): 2.5
* Weight(g): 14.7
* Voltage(V): 4.8 – 6

**2.2.4 IR Sensor**

An infrared sensor is an electronic device that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion.IR Sensor or Infrared Sensor has two main parts.

* IR Transmitter
* IR Receiver

The work of IR transmitter or Infrared transmitter is to transmit the infrared waves whereas the work of IR receiver is to receive these infrared waves. IR receiver constantly sends digital data in the form of 0 or 1 to Vout pin of the sensor. If there is an object in front of IR sensor, the transmitted infrared waves from IR transmitter reflect from that object and are received by the IR receiver. IR sensor gives 0 or LOW in this condition. Whereas, if there is no object in front of the IR sensor, the transmitted infrared waves from IR transmitter is not received by the IR receiver. And IR sensor gives 1 or HIGH in this condition.

**IR LED Transmitter:**

IR LED emits light, in the range of Infrared frequency. IR light is invisible to us as its wavelength (700nm – 1mm) is much higher than the visible light range. IR LEDs have light emitting angle of approx. 20-60 degree and range of approx. few centimeters to several feets, it depends upon the type of IR transmitter and the manufacturer. Some transmitters have the range in kilometers. IR LED white or transparent in colour, so it can give out amount of maximum light.

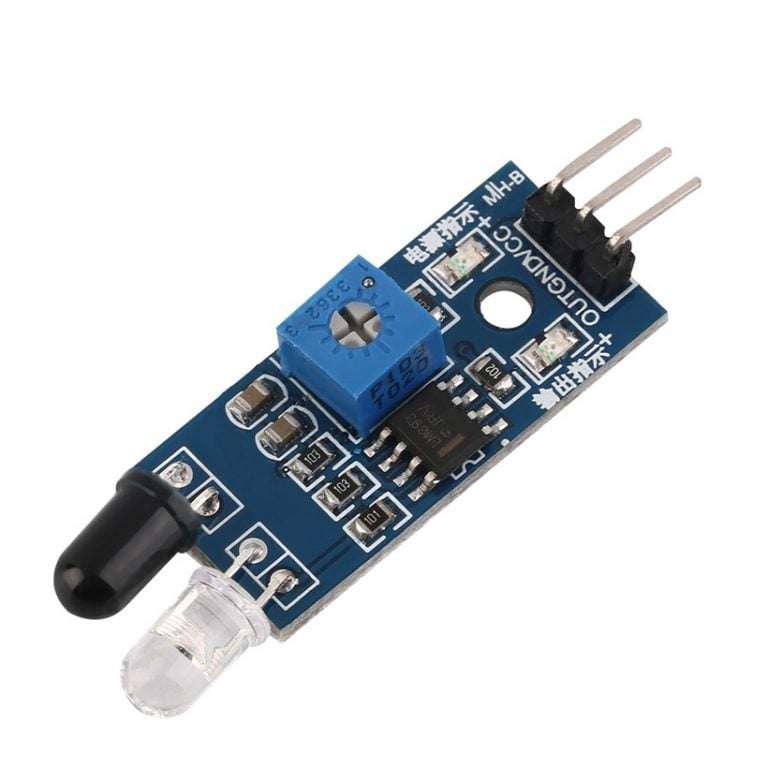


Fig 2.5 IR sensor

**Photodiode Receiver:**

Photodiode acts as the IR receiver as its conducts when light falls on it. Photodiode is a semiconductor which has a P-N junction, operated in Reverse Bias, means it start conducting the current in reverse direction when Light falls on it, and the amount of current flow is proportional to the amount of Light. This property makes it useful for IR detection. Photodiode looks like a LED, with a black color coating on its outer side, Black colour absorbs the highest amount of light.

**Pin Configuration:**

|  |  |
| --- | --- |
| **Pin Name** | **Description** |
| VCC | Power Supply +5v |
| GND | Power Supply Ground |
| OUTPUT | Active High Output |

Table 2.2 Pin configuration of IR sensor

**2.2.5 OLED Display Module**

Organic Light-Emitting Diode (OLED) displays do not require a backlight and are regarded as the ultimate technology for the next generation of flat panel displays. OLED displays are composed of a thin, multi-layered organic film placed between an anode and cathode, which are made up of electric conductive transparent Indium Tin Oxide. The multi-layered organic film includes a Hole Transporting Layer, Emission Layer and Electron Transporting Layer. By applying an appropriate electrical voltage, the holes and electrons are injected into the Emission Layer from the anode and cathode respectively and combine to form exactions, after which electroluminescence occurs.

This 0.96” 128\*64 Blue OLED Module offers 128\*64-pixel resolution. They are featuring much less thickness than LCD Displays with good brightness and produce better and true colors. The connection of this display with Esp32 is made through the I2C (also called as IIC) serial interface. The OLED Display Module produces blue text on black background with very good contrast when supplied with 3.3V-5V Supply. The OLED Display Modules also offers a very wide viewing angle.



Fig 2.6 Two-sided image of Oled

**Pin Description:**

|  |  |  |
| --- | --- | --- |
| **Pin No.** | **Pin Name** | **Description** |
| 1. | Supply Voltage ( Vcc, 5V) | Can be powered by either 3.3V or 5V |
| 2. | Ground (GND) | Pin Ground |
| 3. | Serial Clock(SCL) | Pin SCL of I2C interface |
| 4. | Serial Data(SDA) | Pin SDA of I2C interface |

Table 2.3 Pin configuration of OLED

**Features:**

* Supply voltage: 3.3V-5V
* Pixel: 128\*64
* Display size- 0.96 inch
* Operating temperature range: -40⁰C - +80⁰C
* Use I2C Interface
* Chip No: SSD1306
* Color: Blue
* Drive Duty: 1/64 Duty
* Only need 2 I/O port to control

**2.2.6 Push Button:**

Pushbutton switches are electrical actuators that, when pressed, either close or open the electrical circuits to which they are connected. Push button switches are also known as pushbutton switches. They have the ability to operate a variety of electronic devices.



Fig 2.7 Push button

These switches can be in the shape of a key or a button. They can be permanent or transient. The push button is the most popular type of momentary switch. Push-to-break switches are sometimes used to describe normally closed push button switches,

whereas push-to-make switches are frequently used to describe normally open push button switches.

**2.2.7 USB cable**

The micro B type connector has 5 pins to support USB OTG, allowing smart phones and other mobile devices to read peripherals like a computer might, such as external drives, digital cameras, and other devices. Be aware that a special wiring connection must be used in the cable assembly in order to enable the OTG feature.

The A-Type connector used in USB 2.0 and USB 1.1 applications shares the same design with USB 3.0 A, which also offers a “downstream” connection intended for use only on host controllers and hubs.

Nevertheless, USB 3.0 Type A processes extra pins not present in USB 2.0 Type A. While the USB 3.0 connector is built to support “SuperSpeed”s 5Gbps data transfer, USB 2.0 ports still allow for lower data rates to be transmitted. To help distinguish them from earlier generations, USB 3.0 connectors are frequently blue in color or bear the “SS” logo.



Fig 2.8 USB cable

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **standard** | **Also Known as** | **Max data transfer speed** | **Connector types** | **Max cable length** | **Max charging power** |
| USB 2.0 | Hi-Speed USB | 480 Mbps | USB 2.0 is supported by all USB connectors | 5m | Typically upto 15w |
| USB 3.2 Gen1 | Super Speed USB | 5 Gbps | USB 3 USB- A(Type-A),  USB 3 USB-B, USB-C  USB Micro Series (1m max cable length with micro connector | 3m | Typically upto 15w |
| USB 3.2 Gen 2 | Super Speed 10Gbps | 10Gbps | USB 3 USB-A (Type-A)  USB 3 USB-B  USB-C | 1m | Up to 100W when using a compatible USB-C cable/hub/device. Cables without internal e-Markers are limited to 3A/20v (60W) |
| USB 3.2 Gen 2x2 | SuperSpeed 20Gbps | 20Gbps | USB C | 1m | Up to 100W when using a compatible USB-C cable/hub/device. Cables without internal e-Markers are limited to 3A/20v (60W) |
| Thunderbolt 3 | Thunderbolt | Up to 40Gbps | USB C | 0.8m for passive cable  2m for active cables | Up to 100W when using a compatible USB-C cable/hub/device. Cables without internal e-Markers are limited to 3A/20v (60W) |
| USB 4 / Thunderbolt 4 | TBC | Up to 40Gbps | USB C | TBC | Up to 100W when using a compatible USB-C cable/hub/device. Cables without internal e-Markers are limited to 3A/20v (60W) |

Table 2.4 USB Specifications and USB Connections

**2.2.8 Arduino IDE**

The open-source software known as the Arduino IDE is used to create and upload code to Arduino boards. In addition to a text editor for writing code, a message area, a text terminal, a toolbar with buttons for frequently used operations, and a number of menus, the Arduino Integrated Development Environment, sometimes known as the Arduino Software (IDE), is also available. In order to upload programmes and communicate with them, it connects to the Arduino hardware. For different operating systems, including Windows, Mac OS X, and Linux, the IDE application is appropriate. The programming languages C and C++ are supported. Integrated Development Environment is referred to in this sentence.

Sketching is a common term for writing a programme or piece of code in the Arduino IDE. To upload the sketch created in the Arduino IDE software, we must connect the Genuine and Arduino board with the IDE. The sketch has the ".ino" file extension.

**Manage Library**

The most recent list of all the installed libraries is displayed. This option may also be used to add a new library to the Arduino IDE

**Serial Monitor**

With the attached board on the port, it enables data exchange.

**Toolbar button**

The New, Open, Save, Upload, and Verify icons are all that are shown on the toolbar.

**Verify/Compile**

It will check for the errors in the code while compiling. The memory in the console area is also reported by the IDE.

**Upload**

The Upload button is used to configure the code to the specified board through the port.

**Upload using programmer**

It is employed to replace the board's built-in Bootloader. Using the ‘Upload Using Programmer’ option, we may make use of the Flash memory's whole storage capacity. To put this into action, we must transfer the restored Bootloader to the USB serial port using the Tools->Burn Bootloader option.

**Include library**

Various Arduino libraries are included in Include Library. The libraries are added to our code at the beginning, after the hash symbol (#). Additionally, we may import libraries. The serial data is shown in a plot using the Serial Plotter button. It is already set up in the Arduino IDE.

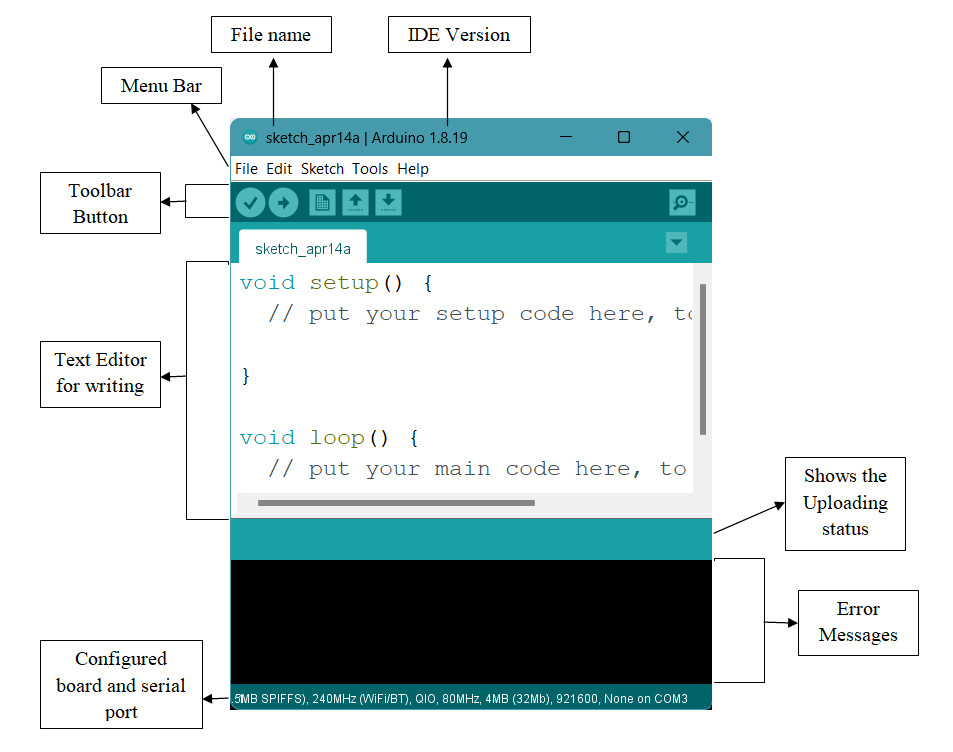


Fig 2.9 Arduino IDE Software

**2.2.9 MIT APP Inventor**

The Massachusetts Institute of Technology (MIT) now manages MIT App Inventor, an integrated development environment (IDE) for web applications that was first made available by Google. It enables those who are new to computer programming to produce application software (apps) for Android and iOS, both of which, as of January 20, 2023, are currently undergoing final beta testing. It is free and open-source software distributed under two licences: an Apache Licence 2.0 for the source code and a Creative Commons Attribution ShareAlike 3.0 Unported licence.

Users can drag and drop visual objects to create applications that can be tested on Android and iOS devices and created to run as Android apps using a graphical user interface (GUI) that is very similar to the programming languages Scratch and StarLogo. It makes use of a mobile companion app that enables immediate live testing and debugging. Google drew on extensive prior research in educational computing as well as internal Google work on online development environments when developing App Inventor.

Constructionist learning theories, which emphasise that programming can be used as a vehicle for engaging compelling ideas through active learning, are the foundation and inspiration for App Inventor and the other projects. As such, it is a part of a larger trend in computers and education that dates back to the 1960s and was started by Seymour Papert and the MIT Logo Group. Mitchel Resnick’s work on Lego Mindstorms and StarLogo has also contributed to this movement. Through its CloudDB component, App Inventor also supports the use of cloud data.

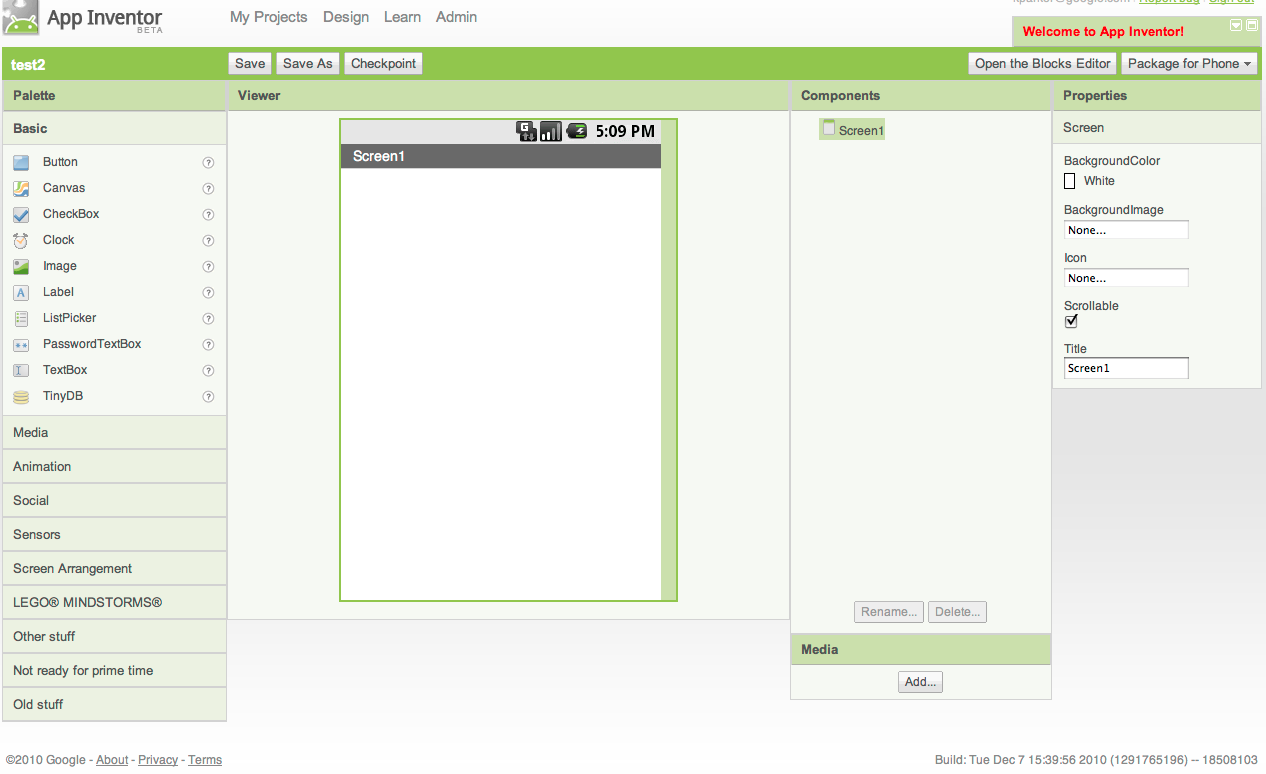


Fig 2.10 Image of MIT App Inventor page

**2.2.10 Google Firebase**

Developers can create iOS, Android, and Web apps using the Google-sponsored application development platform known as Firebase. Tools are available from Firebase for monitoring analytics, reporting and resolving app crashes, as well as developing marketing and product experiments. It uses MQTT protocol.

Among the services provided by Firebase are the following:

**Analytics -** Google Analytics for Firebase provides free, unlimited reporting on up to 500 different events. Analytics provides user behavior data for iOS and Android apps, enabling better performance and app marketing decisions.

**Authentication -** The sign-in and on boarding experiences for users are improved by Firebase Authentication, which makes it simple for developers to create secure authentication systems. With support for email and password accounts, phone auth, Google, Facebook, GitHub, Twitter login, and more, this feature provides a comprehensive identity solution.

**Cloud messaging -** Cloud messaging: Firebase Cloud Messaging (FCM) is a free, cross-platform messaging tool that enables businesses to deliver and receive messages on IOS, Android, and the web with reliability.

**Realtime database -** Data can be stored and synced in real time between users thanks to the Firebase Realtime Database, a NoSQL database hosted in the cloud. When an app is offline, the data is still accessible because it is continuously synced across all clients.



Fig 2.11 Activity demonstrations for Google Firebase

**CHAPTER 3**

**PROJECT IMPLEMENTATION**

**3.1 BLOCK DIAGRAM**

The system consists of ESP32 microcontroller with IR sensors to check vehicle availability at the slots effectively. The data which generated by this system is updated to application using a MQTT protocol. Relay module is used for operating the light present at parking lots. OLED Display is integrated to display the vacant lots present in parking lot which can help at emergency parking. The user can see the availability of lots in the application. At entrance and exit we integrated pushbuttons with servo motors to control the vehicle check-in and check-out. We used the Google firebase to store the data of the sensors, Login and Signup of the application and the booking of slot data. Below is the block diagram of proposed system.

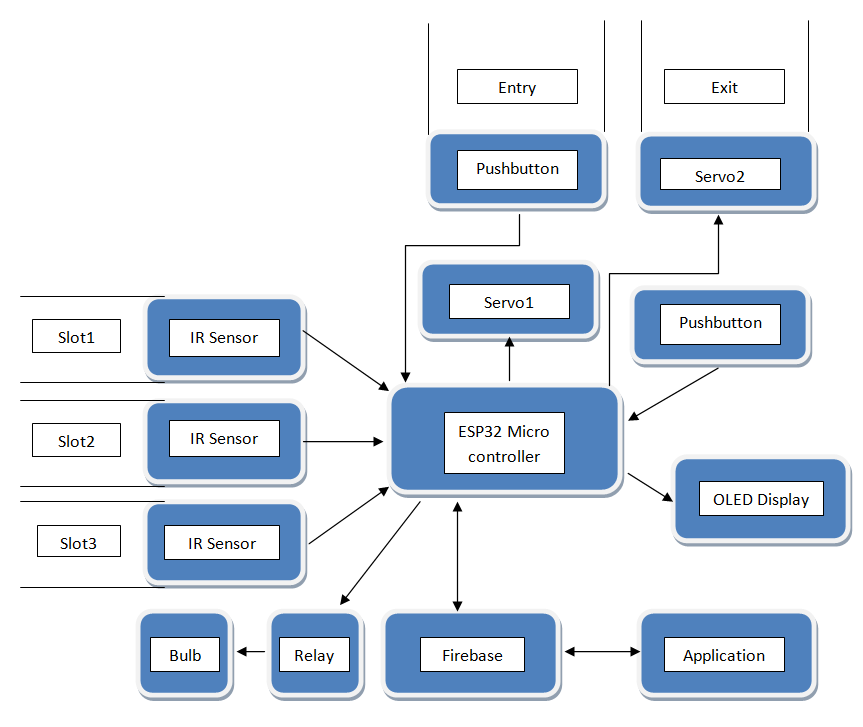
****

Fig 3.1 Block diagram of proposed project

**3.2 WORKING OF THE PROJECT**

The suggested technique uses an ESP32 microcontroller equipped with pushbuttons, an OLED, relay, and infrared sensors. The ESP32 microcontroller manages the entire system. A low-cost, low-power microcontroller called the ESP32 has built-in dual-mode Bluetooth and Wi-Fi. Between the sensors and the cloud, communication is instant. The ESP32 has built-in Wi-Fi and a fast processing speed. A USB (Universal Serial Bus) port is available for powering it. These independent modules are simple to scale, and because they don't need any wiring, scaling can be done while the modules are in use. This system has trouble-free maintenance and is cost-effective. The circuit as a whole receives voltage from the power supply block. We will receive an output of 5V from the voltage regulator as it continuously regulates the voltage. An IR sensor for vehicle detection is located at the parking system's slot. The IR sensor notifies the ESP32 micro-controller of the presence of a vehicle at the slot. The IR sensors are positioned in the parking space slots to determine whether a vehicle is present or not. If a car is there, the information would be sent to the cloud and the application would show that the specific parking space is taken. If no vehicle is present, the application will be updated to reflect that the specific parking space is free. Therefore, the application will display all of this information, including whether or not the parking space is available, in an intuitive manner.

We created a tool that allows users to check available slots and reserve a slot for a specific time. The relay module is used for operating the light present in parking lots. If there is no vehicle in the parking area then the relay helps the bulb to glow else if there is no vehicle available at the parking then the relay helps the light to out. OLED Display is integrated to display the vacant lots present in the parking lot which can help with emergency parking. Push buttons are integrated at the entry and exit of the parking area with servomotor as gates.

The IoT system has vast usage in day-to-day life. IoT technology has become more well-liked and has a wide range of uses. IoT apps function in accordance with how they were created depending on the many application domains. There isn’t a set standard defined architecture of work, nevertheless, that is rigidly followed everywhere. Depending on the particular business job at hand, different architectural layers and levels of complexity are used. The most common and conventional architecture is a four-layer one.The four layers are namely: sensing layer, network layer, data processing layer, and application layer.

**Preception/Sensing Layer**

Any IoT system's initial layer is made up of “things” or endpoint devices that act as a link between the real world and the digital one. The physical layer, which contains sensors and actuators capable of gathering, receiving, and processing data across a network, is referred to as perception. Wireless or wired connections can be used to link sensors and actuators. The components range and locations are not constrained by the design.

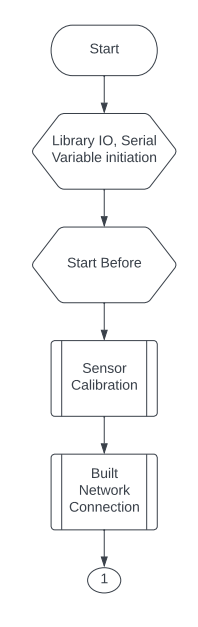


Fig 3.2 Flow chart of the IoT system

**Network Layer**

An overview of the data flow across the programme is given by the network layers. Data Acquiring Systems (DAS) and Internet/Network gateways are present in this tier. Data aggregation and conversion tasks are carried out by a DAS (gathering and aggregating sensor data, converting analogue data to digital data, etc.). Data gathered by the sensor devices must be sent and processed. The network layer performs that function. It enables connections and communication between these gadgets and other servers, smart gadgets, and network gadgets. Additionally, it manages all device data transfers.

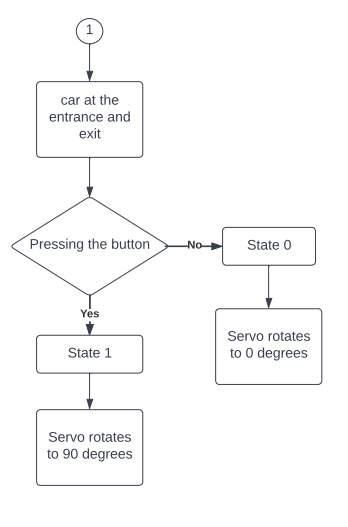


Fig 3.3 Flow chart showing working of entance and exit of the parking area

**Processing Layer**

The processing layer, also known as the middleware layer, is a crucial part of an IoT system design. It often uses several linked computers at once, in the form of cloud computing, to give greater computation, storage, networking, and security performance.

An IoT system architecture's processing layer is in charge of analysing input data to produce fresh insights, valuable forecasts, and timely alerts.

An IoT system often manages massive data volumes produced by several edge devices at numerous points on the network's edges. The “middleware” of the processing layer prepares this data for the application layer in three steps:

The IoT ecosystem's processing layer functions as its brain. Before being transported to the data centre, data is often analysed, pre-processed, and stored here. It is then retrieved by software programmes that handle the data and plan future actions. This is where edge analytics or edge IT comes into play.

**Data Aggregation**

Middleware accurately classifies and allots various data kinds to the proper storage. Unstructured data is generally stored in data lakes since it takes up more storage space and includes things like audio and video feeds and photographs. While organised data, such as measurements, log values, and instrument readings (telemetry data), takes up less space and is kept in data warehouses.

**Data Abstraction**

Aggregating data from various sources and ensuring that it is translated into a format that can be “read” by the software of the application layer are both tasks that are included in data abstraction.

* This level develops data structures and views in the ways that applications require.
* It integrates information from several sources.
* To support client applications, it cleans up, filters, picks, projects, and reformats the data.
* The disparities in data shape, format, semantics, access protocol, and security are reconciled.

**Data Analysis**

Data analysis is the process of finding patterns in massive, apparently random data sets using machine learning (ML) or deep learning algorithms.

The data is delivered to the data centres and servers for final analysis and reporting once it has been appropriately preprocessed, examined, and any gaps have been filled. The management services area includes data centres and cloud services, which often handle data using analytics, device management, and security controls. Data may also be sent to end-user applications like healthcare, retail, environment, emergency, energy, etc. thanks to the cloud.

The data might be transmitted to data centres or cloud-based servers for ultimate processing after analysis. Hardware expenses can be reduced by using the cloud platform, but data security is still a worry. Physical servers and data centres are safer, but they are also more expensive.

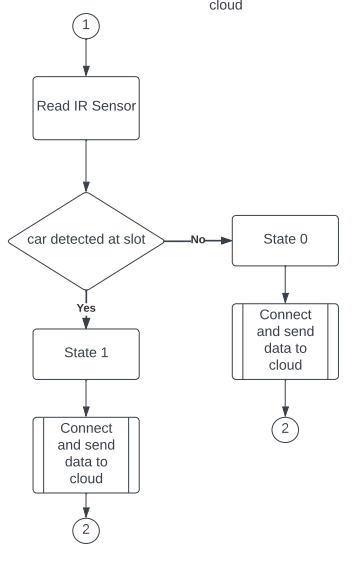


Fig 3.4 Flow chart showing the working of detection and sending of data to the cloud (Firebase)

**Application Layer**

The application layer is the topmost layer that directly connects the end users. This layer includes various software and applications such as mobile apps and portals. The application layer also provides analytics and processing capabilities that allow data to be analyzed and transformed into meaningful insights. In this system, an application is developed to observe the slot availability by the user and also has the facility to pre-book the slot at the required time-lapse.

The Machine-to-machine communication is governed by the standards-based messaging protocol, or collection of rules, known as MQTT. Smart sensors, wearable, and other Internet of Things (IoT) gadgets frequently need to send and receive data via a network with finite resources and bandwidth. These IoT devices utilise MQTT for data transfer because it is simple to set up and effective in exchanging IoT data. MQTT enables communication from devices to the cloud and from the cloud to the device.

The MQTT protocol has the following advantages, which have led to it becoming a standard for IoT data transmission:

**Efficient and portable**

Even modest microcontrollers can utilise MQTT since its implementation on an IoT device uses very little power. A simple MQTT control message, for instance, can just include two data bytes. Additionally, MQTT message headers are compact to maximise network capacity.

**Scalable**

The implementation of MQTT involves a very little amount of code and uses very little electricity. Additionally, the protocol has capabilities that enable connection with a significant number of IoT devices. Consequently, you can use the MQTT protocol to establish connections with a huge number of these devices. The most scalable MQTT broker for Internet of Things applications right now is EMQ X. It supports messaging across more than 100 million clients within a single cluster and can handle millions of MQTT messages per second with sub-millisecond latency.

**Reliable**

Many Internet of Things (IoT) devices communicate via unstable cellular networks with high latency and little capacity. The IoT device can connect to the cloud more quickly because to MQTT's built-in functionalities. Additionally, it specifies three alternative quality-of-service levels - at most once (0), at least once (1), and exactly once (2) to provide dependability for IoT use cases. Reliable aims to quantify the probability that a system performs its intended function correctly throughout its mission time. Making networks safer from online attacks and using intelligent edge-based data storage and computing are two ways to increase IoT dependability. Reliable IoT implementation for smart cities may lower latency, boost performance, and boost energy efficiency.

**Secure**

Through the use of contemporary authentication protocols like OAuth, TLS1.3, Customer Managed Certificates, and others, MQTT makes it simple for developers to encrypt messages and authenticate devices and users.

**MQTT client**

Any machine that uses a MQTT library, from a server to a microcontroller, is referred to as a MQTT client. The client takes on the roles of a publisher and a receiver depending on whether it is sending or receiving messages. Basically, a MQTT client device is any device that uses MQTT to interact across a network.

**MQTT broker**

Any machine that uses a MQTT library, from a server to a microcontroller, is referred to as a MQTT client. The client takes on the roles of a publisher and a receiver depending on whether it is sending or receiving messages. Basically, a MQTT client device is any device that uses MQTT to interact across a network.

* Authorizing and authenticating MQTT clients
* Passing messages to other systems for further analysis
* Handling missed messages and client sessions

**Well-supported**

The MQTT protocol implementation is widely supported by a number of languages, including Python. As a result, programmers may easily apply it with little to no code in any kind of application.

The publish/subscribe mechanism is the foundation of the MQTT protocol. In a conventional network, clients and servers speak directly to one another. Clients ask the server for resources or data, which the server subsequently processes and returns. However, MQTT separates the message sender (publisher) from the message recipient (subscriber) using a publish/subscribe paradigm. Instead, the communication between publishers and subscribers is handled by a third component known as a message broker. All incoming communications from publishers must be filtered by the broker in order to properly distribute them to subscribers.

The publish/subscribe model is implemented by MQTT by creating clients and brokers as shown below.

**MQTT connection**

An MQTT connection is used to start communication between clients and brokers. Clients start a connection by sending the MQTT broker a CONNECT message. By sending back a CONNACK message in response, the broker verifies that a connection has been made. The broker and the MQTT client both need a TCP/IP stack to interact. The only connection clients ever make is with the broker.

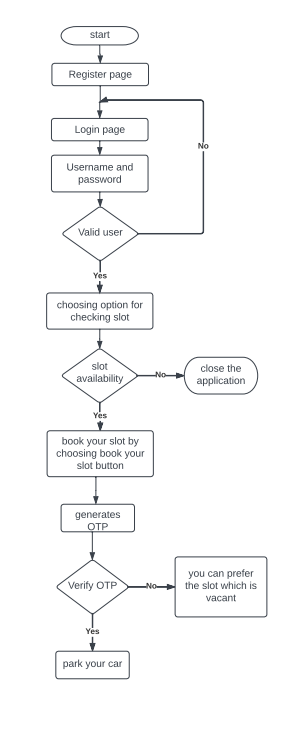


Fig 3.5 Flow chart demonstrating the process of the application

**CHAPTER 4**

**SIMULATION RESULTS & ANALYSIS**

In this section, the experimental analysis and results are given in brief. The result is obtained by an IoT module in which the IR sensor senses the presence of vehicles and the indication to the vehicle owners is shown by the OLED in the module. The IR sensor is designed in such a manner that when the vehicle is detected in the slot, The ESP32 then sends the data to the database to record at which the vehicle is parked, An IR sensor can measure the heat of an object as well as detects the motion. Usually, in the infrared spectrum, all objects radiate some form of thermal radiation. The database contains all the details along with the slot number. Easy to install and no other extra computations are required. The time spent on the parking lot to park or wait for a spot is reduced and can be monitored and further used as data to design efficient and modular parking spaces. Reduction of manpower is a key result as it doesn't require laboring to process and collect fares for parking. It provides an easier way to pay. Therefore, this proposed system helps to avoid minor accidents and traffic.

****

Fig 4.1 OLED Display demonstrating the slot availability

Frontend is a browser that acts as a user interface. The users can check and edit their details. Here the frontend is used for the following features of the application:

**A User Registration** - Once the user registers the details and user in the application, an account will be created.

**B. User login** - The user can log in to the application using their credentials and be guided to the home screen shown in the figure.

**C. Parking space detection** - The user can check the vacancy of parking spots in a particular location. If any vacant spot is available, the user can pre-book the parking spot.

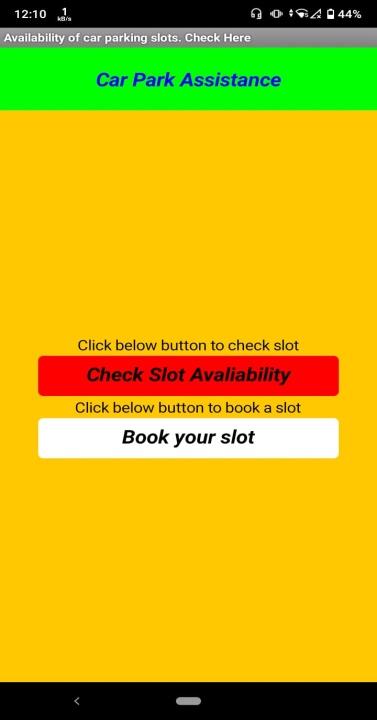
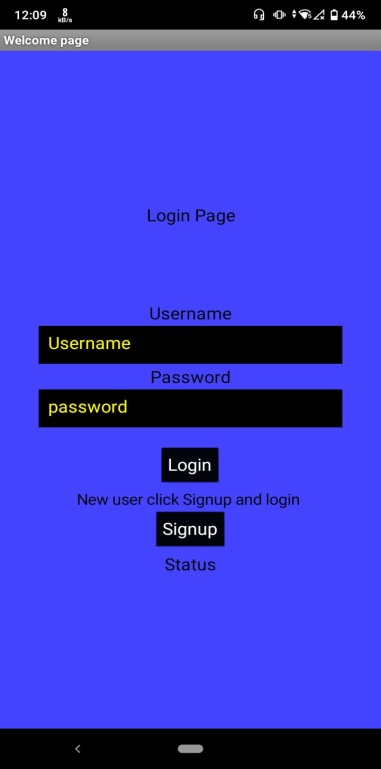


Fig 4.2 Demonstration of login page, registration page, and home page

The backend is a database or a server where all the data is collected, stored, organized, accessed, and managed by the users with the help of application programming. All actions performed during the parking of vehicles are recorded. Data regarding the vacant and occupied parking spots fare for parking vehicles, and details of the user are recorded immediately in the database. Here Firebase is used as the backend.

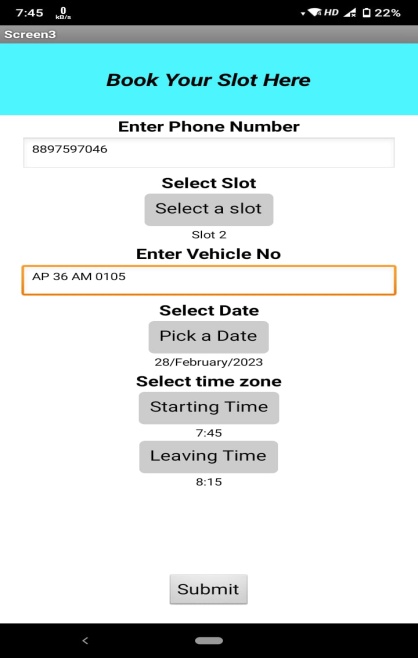
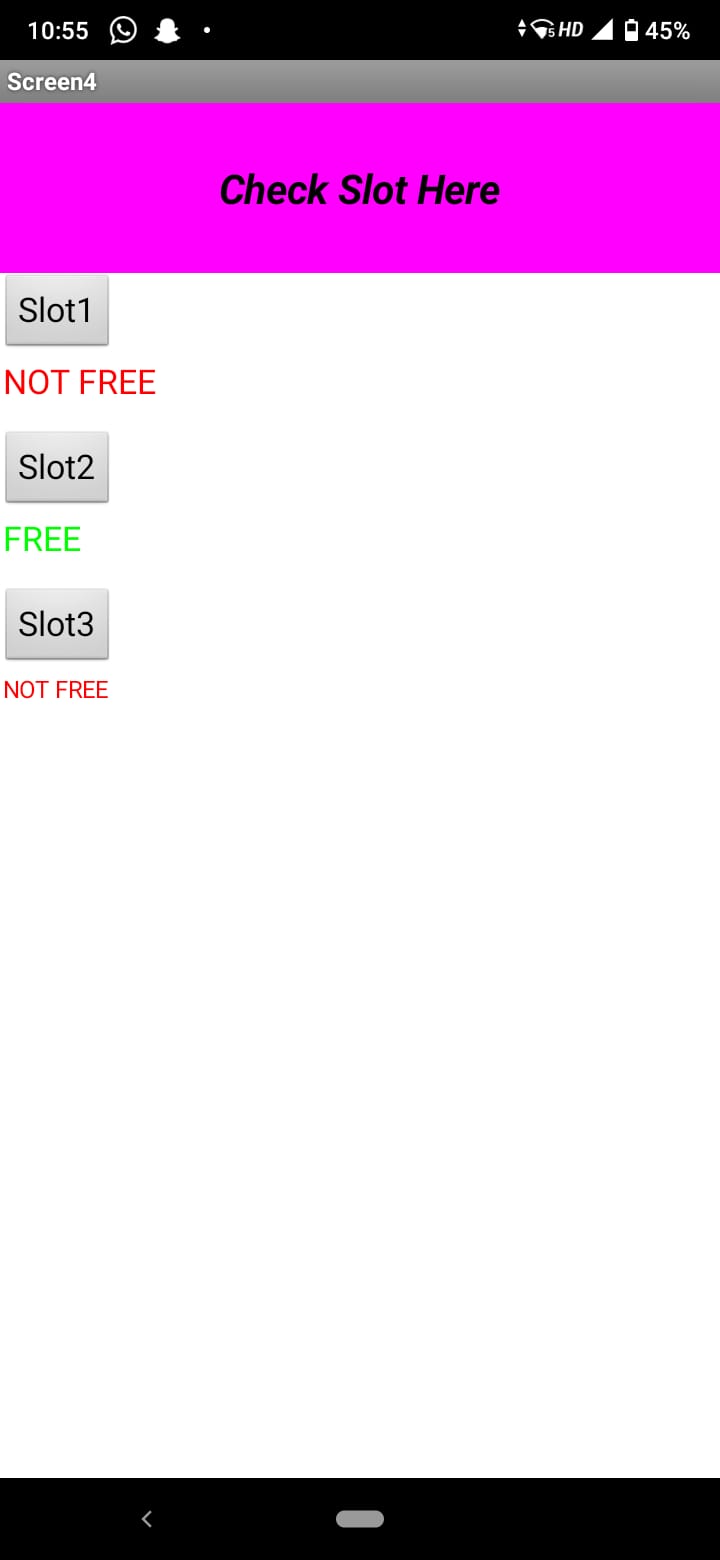


Fig 4.3 Demonstration of check slot screen and book your slot screen

As we can see in the image below, the database contains information from an IR sensor; if the data is 0 or 1, the programme will display a space that is vacant. The information of the person who registered for the application is visible, and the data entered into the booking screen is shown in the left image.

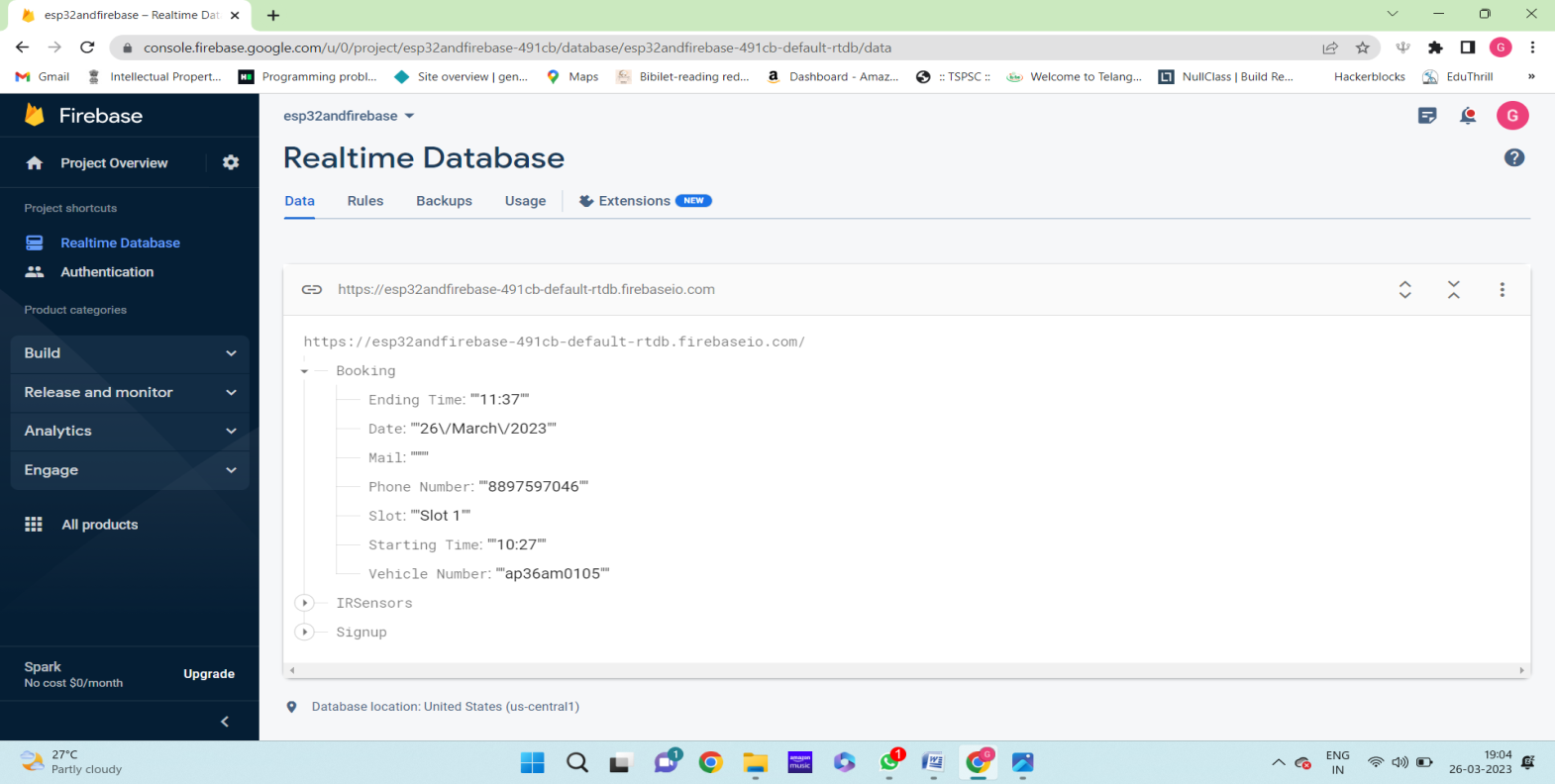


Fig 4.4 Demonstration of backend data of ir sensors, users credentials, and data of slot booking

**CHAPTER 5**

**CONCLUSION & FUTUTRE SCOPE**

**5.1 CONCLUSION**

Though this project finds its application at the multi stored parking lot, where it is challenging to locate vacant parking space, the system can be customized to large buildings, institutions, industries and residential apartments. This technology can save an enormous amount of time, overcome inconvenience, reduce power consumption and pollution by vehicle during searching for parking space. In addition to the above features, Commercial parking spaces can utilize this technology to facilitate smooth check-in and check-out and make parking hassle-free environment.

**5.2 FUTURE SCOPE**

For further development system can be modified by the camera module implementation and can track the vehicle entry and exit system using Rasberry Pi. According to the change in technology new methods and new gadgets are also available so they can be also used to make the project more efficient and satisfacts the user. By making an online payment, we may enhance our project. It is sufficient to use this way of net payment to rent the parking place for the required period of time.

Using AI with IoT can also implement to make the parking system more compatible an efficient. If the user need payment gateway they can include into this project.

**BIBLIOGRAPHY**

[1] Y. Agarwal, P. Ratnani, U. Shah and P. Jain, “IoT based Smart Parking System”, 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 464-470, doi:10.1109/ICIC CS51141.2021.9432196.

[2] S. P. Patro, P. Patel, M. K. Senapaty, N. Padhy and R. D. Sah, “IoT based Smart Parking System: A Proposed Algorithm and Model”, 2020 International Conference on Computer Science Engineering and Applications (ICCSEA), 2020, pp.1-6,doi:10.1109/IC CSEA49143. 20209132923.

[3] M. P. Thakre, P. S. Borse, N. P. Matale and P. Sharma, “IOT Based Smart Vehicle Parking System Using RFID,” 2021 International Conference on Computer Communication and Informatics (ICCCI), 2021, pp. 1-5, doi: 10.1109/ICCCI50826.2021. 9402699.

[4] S. Ravishankar and N. Theetharappan, “Cloud connected smart car park,” 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp. 71-74, doi: 10.1109/I-SMAC.2017.8058269.

[5] M. S. MOHAMMED and S. USTUN ERCAN, “Safety Based Smart Parking Guidance System Using GSM and Zigbee,” 2019 3rd International Symposium on Multidisciplinary Studies and Innovative technologies (ISMSIT), 2019, pp. 1-5, doi: 10. 1109/ISMSIT.2019.8932730.

[6] A. Waheed and P. Venkata Krishna, “Comparing Biometric and Block chain Security Mechanisms in Smart Parking System,” 2020 International Conference on Inventive Computation Technologies (ICICT), 2020, pp. 634- 638, doi:10.1109/ICICT48043.2020. 9112483.

[7] GokulKrishna. S, Harsheetha. J, Akshaya. S, Jeyabharathi, “An IoT based Smart Outdoor Parking System” 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS) | 978-1-6654-0521-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICACCS51430.2021.9441766.

[8] PAPER “Smart Parking System using MQTT Communication Protocol and IBM Cloud” C Ashhwath et al 2021 Journal of Physics: Conference Series 2115 012013.

[9] Anitha.G, Krupasree.P, Kalaivani.K, Nadheera.M, “Embedded IoT Car Parking and Billing System” 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)|978-1-6654-0521-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICACCS51430.2021.9441757.

[10] Park King: An IoT-based Smart Parking System, 5th IEEE International Smart Cities Conference paper (ISC2 2019) by C. Ajchariyavanich1, T. Limpisthira1, N. Chanjarasvichai1, T. Jareonwatanan1, W. Phongphanpanya1, S. Wareechuensuk1, S. Srichareonkul1, S. Tacatanitanont1, C. Ratanamahatana2, N. Prompoon2 and M. Pipattanasomporn.

**PUBLICATION PROOF**