Smart Parking Management System with Automatic Billing and IoT Data Storage

G Rishitha[1], Akiti Bhavyasri Reddy[2], K Manoj[3], Kukkala Manoj Kumar[4], Mr. S. Kirankumar[5]

[1,2,3,4]Students, Department of Computer Science and Engineering, Sree Dattha Group of Institutions, Sheriguda, Ibrahimpatnam, Telangana, India

[5]Assistant Professor, Department of Computer Science and Engineering, Sree Dattha Group of Institutions, Sheriguda, Ibrahimpatnam, Telangana, India

To Cite this Article

G Rishitha, Akiti Bhavyasri Reddy, K Manoj, Kukkala Manoj Kumar, Mr. S. Kirankumar, "Smart Parking Management System with Automatic Billing and IoT Data Storage", Journal of Science Engineering Technology and Management Science, Vol. 03, Issue 05, May 2026, pp: 1066-*1075, DOI: http://doi.org/10.64771/jsetms.2026.v03.i05.pp1066-1075*

Submitted: 19-02-2026 Accepted: 25-03-2026 Published: 01-04-2026

Abstract: Growing urbanization has caused more people to use cars, which has made parking more crowded and made manual parked management systems less effective. This project shows how to make a Smart Parking System with Auto Payments and IoT Storage for Data using ESP32. The system is meant to automate car entry, processing payments, parking slot tracking, and storing data storage. An EM18 an RFID tag at an entry gate detects RFID cards that are allowed to enter the system. When the ESP32 sees a card, it checks the balance and automatically takes out the parking fee. The entry gate drive is turned on to let vehicles through if there is enough balance. IR sensors are put at the entrances and exits to see if cars are moving and open the gates on their own. The system keeps track of when vehicles enter and leave and how many slots are filled. It then sends this information to the ThingSpeakIoT cloud service for storage and monitoring in real time. An LCD screen always shows information about available and taken parking spots. This parking system uses IoT and automation to cut down on the need for people to be involved, cut down on wait times, and make parking more efficient overall.

I. INTRODUCTION

Due to the proliferation in the number of vehicles on the road, traffic problems are bound to exist. This is due to the fact that the current transportation infrastructure and car park facility developed are unable to cope with the influx of vehicles on the road. To alleviate the aforementioned problems, the smart parking system has been developed. With the implementation of the smart parking system, patrons can easily locate and secure a vacant parking space at any car park deemed convenient to them. Vehicle ingress and egress are also made more convenient with the implementation of hassle free payment mechanism. With the help of a computerized system we can deliver a good service to citizens who want to park their vehicle into the any organization's premises using Internet of Things (IOT) based on parking management system. In this context, Internet of Things (IOT) uses sensors to connect physical parking space infrastructures with information and communication technologies, where cloud-based smart management services are provided. To implement this concept a mobile based application would be developed. This mobile application will allow an end user to check the availability of parking space and book a particular parking lot accordingly. Each parking lot would be equipped with a control system that enables monitoring of the number of free and occupied parking places and informing users about the parking lot status (open with/without free available parking spaces or closed) Additionally the

application would display parking service payment according to parking time duration. Also it will sense if a vehicle has arrived on the gate for automated gate opening. This allows users to check for available parking space online from anywhere for hassle free parking. Thus, the system solves the parking issue.

II. RELATED WORK

"An IoT Assisted Intelligent Parking System (IPS) for Smart Cities,"

A. M. Said, A. E. Kamal, and H. Afifi.

The transformation of existing infrastructure into smart cities cannot ignore the smart management of the parking systems. The cities with high population density (Metro cities inclusive) specifically face the problem of finding the nearest available parking space. The rising number of vehicles daily makes this problem more severe that questions the safety of vehicles as well. In this study, the issue of finding parking spaces in smart cities is addressed using an IoT-based methodology. The proposed Intelligent Parking System (IPS) consists of an IoT framework that collects real-time data, send it to the cloud, and thereby suggests to the user a suitable place for parking the vehicle at a nearby location. As a component of the framework, a mobile application has been developed that enables users to check the availability of nearby parking spots and subsequently reserve a parking place. This paper also describes different use-cases of a person for finding a parking space and parking it in right place. The proposed system has been implemented using Raspberry Pi, NodeMCU, Radio Frequency Identification (RFID), and Infrared (IR) sensors. The results discussed in the later sections support the usability of this IPS.

"RFID based Smart Parking System,"

S. Shenoy Basty, R. Kiwad, S. Vittal, and M. M. Ullah.

There has been a considerable amount of reduction in transaction costs and decrease in stock shortage with the use of Radio Frequency Identification (RFID) technology in automation. Most of the RFID networks include a wide range of automation technologies. These technologies are RFID readers, RFID writers, RFID barcode scanners, RFID smart sensors and RFID controllers. In this study, a solution has been provided for the problems encountered in parking-lot management systems via RFID technology. RFID readers, RFID labels, computers, barriers and software are used as for the main components of the RFID technology. The software has been handled for the management, controlling, transaction reporting and operation tasks for parking lots located on various parts of the city. Check-ins and check-outs of the parking-lots will be under control with RFID readers, labels and barriers. Personnel costs will be reduced considerably using this technology. It will be possible to see unmanned, secure, automized parking-lots functioning with RFID technology in the future. Check-ins and check-outs will be handled in a fast manner without having to stop the cars so that traffic jam problem will be avoided during these processes. Drivers will not have to stop at the circulation points and parking tickets will be out of usage during check-ins and check-outs. It will be avoided ticket-jamming problems for the ticket processing machines as well. Vehicle owners will not have to make any payments at each check-out thus a faster traffic flow will be possible. Since there won't be any waiting during check-ins and check-outs the formation of emission gas as a result of such waiting will be avoided. An automized income tracking system, a car tracking system for charging and a central parking-car tracking system have been developed and utilized. Instead of cars' parking on streets, a more modern and a fast operating parking-lot system have been developed.

"RFID Based Smart Vehicle Parking System Using IoT,"

P. Rajesh, S. Mahesh, B. Bheemesh, V. Hemanth, and K. Adarsh.

The IoT-based intelligent parking system is designed to enhance parking management by automating realtime space monitoring, vehicle authentication, and billing processes. The system integrates IoT components such as ESP8266, RFID technology, and line sensors to ensure efficient and accurate tracking of parking slot occupancy. RFID authentication enables secure vehicle entry and exit, preventing unauthorized access while ensuring a seamless user experience. Real-time data transmission to a cloud server allows users to check parking availability remotely, reducing congestion and improving space utilization. The automated billing system accurately calculates parking fees based on entry and exit times, eliminating manual errors and enhancing transparency. Wireless communication ensures seamless integration with digital payment systems, further improving convenience. The system is designed to be energy-efficient and scalable, making it suitable for deployment in urban environments and smart city infrastructure. By leveraging IoT and cloud computing, the proposed system offers a cost-

effective and reliable solution to modern parking challenges, optimizing space management while improving security and operational efficiency. The implementation of this system promises to alleviate parking-related issues in urban settings, providing an innovative approach to managing parking resources effectively.

"IoT Based Smart Parking System"

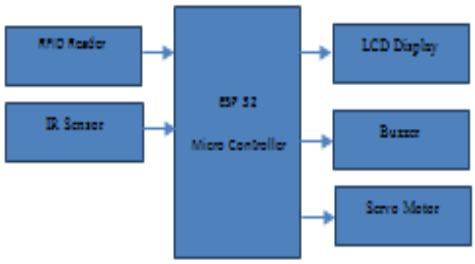
Sharma, R.

This paper presents an IoT-enabled parking system that automates vehicle entry and slot monitoring. The authors propose integrating RFID readers and microcontrollers with IoT platforms for real-time data storage. The study emphasizes reduced waiting time, improved efficiency, and scalability for smart city applications.

III. IMPLEMENTATION

The proposed Smart Parking Management System uses an ESP32 microcontroller integrated with an EM18 RFID reader, IR sensors, gate motors, LCD display, and ThingSpeakIoT platform. At the entry gate, the EM18 reader detects the user's RFID card and the ESP32 automatically verifies and deducts the parking fee. If the balance is sufficient, the gate opens automatically. IR sensors detect vehicle movement at both entry and exit points to control gate operation. Vehicle entry time, exit time, and parking slot details are uploaded to the ThingSpeak cloud for real-time monitoring and data analysis. The LCD displays the number of available and occupied parking slots, helping drivers make quick decisions. This system ensures secure, cashless, and efficient parking management.

Block Diagram



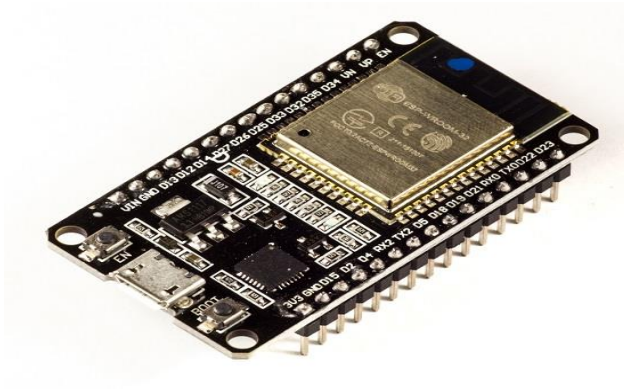
Working

The Smart Parking Management System works by integrating IoT-based automation with secure entry and payment processing. An EM18 RFID reader detects authorized RFID cards at the entry gate, and the ESP32 microcontroller verifies the card balance before automatically deducting the parking fee. If sufficient balance is available, the gate motor is activated to allow vehicle access, while IR sensors at entry and exit points detect vehicle movement to control gate operations. The system records vehicle entry details and slot occupancy, uploading this data to the ThingSpeak IoT cloud platform for real-time monitoring and storage. An LCD display continuously shows available and occupied slots, ensuring transparency. By automating payment, slot monitoring, and data storage, this system reduces human intervention, minimizes waiting time, and improves overall parking efficiency in urban environments.

HARDWARE REQUIREMENTS

ESP-32

ESP32 is the SoC (System on Chip) microcontroller which has gained massive popularity recently. Whether the popularity of ESP32 grew because of the growth of IoT or whether IoT grew because of the introduction of ESP32 is debatable. If you know 10 people who have been part of the firmware development for any IoT device, chances are that 7−8 of them would have worked on ESP32 at some point. So what is the hype all about? Why has ESP32 become so popular so quickly? Let's find out.



Before we delve into the actual reasons for the popularity of ESP32, let's take a look at some of its important specifications. The specs listed below belong to the ESP32 WROOM 32 variant.−

- Integrated Crystal− 40 MHz
- Module Interfaces− UART, SPI, I2C, PWM, ADC, DAC, GPIO, pulse counter, capacitive touch sensor
- Integrated SPI flash− 4 MB
- ROM− 448 KB (for booting and core functions)
- SRAM− 520 KB
- Integrated Connectivity Protocols− WiFi, Bluetooth, BLE
- On−chip sensor− Hall sensor
- Operating temperature range− −40 − 85 degrees Celsius
- Operating Voltage− 3.3V
- Operating Current− 80 mA (average)

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). If you've gone through the previous chapter, you'd have realized that the major operational blocks of any IoT device are sensing, processing, storage, and transmitting. 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.

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. Yep, you heard that right. 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.

With so many features, ESP32 would be costing a fortune, right? That's the best part. ESP32 dev modules cost in the ballpark of ₹ 500. Not only that, the chip dimensions are quite small (25 mm x 18 mm, including the antenna area), allowing its use in devices requiring a very small form factor.

One very big advantage with ESP32, which has aided its quick adoption and massive popularity, is the provision for programming the ESP32 within the Arduino IDE.

Now, I should point out here that Arduino is not the only IDE that helps you compile code for ESP32 and flash it into the microcontroller. There is ESP−IDF which is the official development framework for ESP32, which provides much more flexibility in

terms of configuration options. However, it is hardly as intuitive and user−friendly as the Arduino IDE, and if you are starting out with ESP32, Arduino IDE is ideal to get your hands dirty. Also, with the number of supporting libraries built for ESP32 in Arduino, courtesy of the huge developer community, there's hardly any functionality of ESP32 which can't be realized with the Arduino IDE. ESP-IDF is more suitable for the more advanced and experienced programmers, who need to stretch ESP32 to its limits. If you are one of those, you are looking for the ESP−IDF Getting Started Guide. Others can follow along.

RFID

Radio Frequency Identification (RFID) is a form of wireless communication that incorporates the use of electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic spectrum to uniquely identify an object or person. It uses radio frequency to search, identify, track, and communicate with items and people.

RFID (Radio Frequency Identification) is a technology that uses electromagnetic fields to automatically identify and track tags attached to objects. These tags contain electronically stored information that can be read from several meters away, without requiring direct line-of-sight. RFID is commonly used in inventory management, asset tracking, access control, and supply chain logistics due to its efficiency and accuracy in tracking and managing items.

It is a method that is used to track or identify an object by radio transmission over the web. Data is digitally encoded in an RFID tag which might be read by the reader. This device works as a tag or label during which data is read from tags that are stored in the database through the reader as compared to traditional barcodes and QR codes. It is often read outside the road of sight either passive or active RFID.

RC522 RFID Reader Module

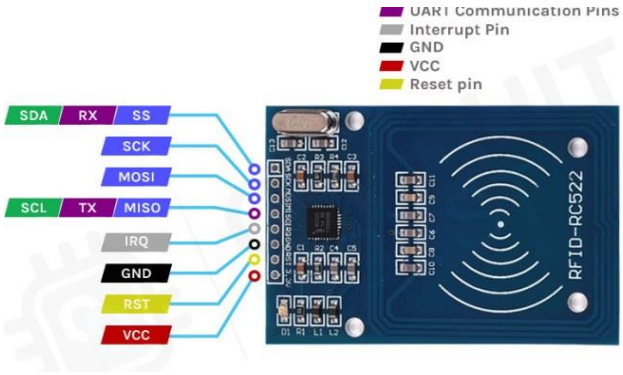
RC522 is a Multi-communication RFID Module for Arduino and Microcontrollers. The RC522 is known as MFRC-522 due to its NFX semiconductor microcontroller. The module allows the developers to interface it with any other SPI, I2C, and UART based microcontrollers. The RC522 module works on 13.56 MHz frequency and it can act as a reader and write for UID/RFID cards. The RFID cards communicate with the module at a short distance with radio frequency due to the mutual induction technique. In most of the security and commercial products, the module is effective because the errors and issues with RFID Tags are detectable by it.

RC522 Pin Configuration Details

In this module, there are only two kinds of pins. So, the first one is power and the second one is the communication pins. Therefore, the device may have its microcontroller chip on itself but it only makes it to works as an RFID. The on-board microcontroller won't make the module a stand-alone device.

RC522 Pinout Diagram

All the pins of MFRC/RC522 RFID card Reader are:



RC522 RFID Reader Features

- RFID RC522 uses mutual induction to activate the cards and 13.56MHz for data transfer.
- The RFID Cards are useable from both sides of the module at max 5cm.
- The only 3.3V is required to activate the device.
- Its auto-sleep mode makes it less power consumption module.
- The module has three kinds of communications (UART, SPI, and I2C). Therefore, it is useable with almost every microcontroller or device in the market.
- The RFID cards and reader (RC522) can transfer data up to 10Mb/s.

RFID cards

An RFID card is a smart card that integrates **radio frequency identification** (RFID) technology. Each RFID card is embedded with an antenna connected to an RFID IC, so it can receive, store, and transmit data via radio waves. The core function of an RFID card is to automatically identify objects and transmit data in a contactless manner.

Fig: RIFD cards

IR SENSOR

The IR sensor or infrared sensor is one kind of electronic component, used to detect specific characteristics in its surroundings through emitting or detecting IR radiation. These sensors can also be used to detect or measure the heat of a target and its motion. In many electronic devices, the IR sensor circuit is a very essential module. This kind of sensor is similar to human's visionary senses to detect obstacles.

IR Sensor

The sensor which simply measures IR radiation instead of emitting is called PIR or passive infrared. Generally in the IR spectrum, the radiation of all the targets radiation and some kind of thermal radiation are not visible to the eyes but can be sensed through IR sensors.

In this sensor, an IR LED is used as an emitter whereas the photodiode is used as a detector. Once an infrared light drops on the photodiode, the output voltage & resistance will be changed in proportion to the received IR light magnitude.

LCD

LCD is a flat display technology, stands for "Liquid Crystal Display," which is generally used in computer monitors, instrument panels, cell phones, digital cameras, TVs, laptops, tablets, and calculators. It is a thin display device that offers support for large resolutions and better picture quality. The older CRT display technology has replaced by LCDs, and new display technologies like OLEDs have started to replace LCDs. An LCD display is most commonly found with Dell laptop computers and is available as an active-matrix, passive-matrix, or dual-scan display. The picture is an example of an LCD computer monitor.

The Liquid Crystal library allows you to control LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

Output of the sketch on a 16x2 LCD

The LCDs have a parallel interface, meaning that the microcontroller has to manipulate several interface pins at once to control the display. The interface consists of the following pins:

- A **register select (RS) pin** that controls where in the LCD's memory you're writing data to. You can select either the data register, which holds what goes on the screen, or an instruction register, which is where the LCD's controller looks for instructions on what to do next.
- A **Read/Write (R/W) pin** that selects reading mode or writing mode

- An **Enable pin** that enables writing to the registers
- 8 **data pins (D0 -D7)**. The states of these pins (high or low) are the bits that you're writing to a register when you write, or the values you're reading when you read.

There's also a display contrast pin (Vo), power supply pins (+5V and GND) and LED Backlight (Bklt+ and BKlt-) pins that you can use to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively.

The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register. The Liquid Crystal Library simplifies this for you so you don't need to know the low-level instructions.

The Hitachi-compatible LCDs can be controlled in two modes: 4-bit or 8-bit. The 4-bit mode requires seven I/O pins from the Arduino, while the 8-bit mode requires 11 pins. For displaying text on the screen, you can do most everything in 4-bit mode, so example shows how to control a 16x2 LCD in 4-bit mode.

SERVO MOTOR

Servo Motor Introduction**:** A servo motor is an electric motor that adjusts its position, speed, or torque in response to controller inputs.
The term servo comes from the Latin word serves, meaning servant or slave. This reflects the historical use of servo motors as auxiliary drives that assist the main drive system However, modern servo motors are capable of providing high performance and precision as main drives in various applications.

A servo motor consists of three main components:

- **A motor:** This can be either a DC motor or an AC motor depending on the power source and the application requirements. The motor provides the mechanical power to rotate or move the output shaft.
- **A sensor:** This can be either a potentiometer, an encoder, a resolver, or another device that measures the position, speed, or torque of the output shaft and sends feedback signals to the controller.
- **A controller:** This can be either an analog or a digital circuit that compares the feedback signals from the sensor with the desired setpoint signals from an external source (such as a computer or a joystick) and generates control signals to adjust the motor's voltage or current accordingly.

The controller employs a closed-loop feedback system, adjusting the motor's movement to closely align with the desired setpoint, maintaining strict accuracy.

The controller can also implement various control algorithms, such as proportional-integral-derivative (PID) control, fuzzy logic control, adaptive control, etc., to optimize the performance of the servo motor.

How Does a Servo Motor Work?
The basic working principle of a servo motor involves the controller receiving two types of input signals:

- A setpoint signal: This is an analog or digital signal that represents the desired position, speed, or torque of the output shaft.
- A feedback signal: This is an analog or digital signal that represents the actual position, speed, or torque of the output shaft measured by the sensor.

The controller compares these two signals and calculates an error signal that represents the difference between them.The error signal is then processed by a control algorithm (such as PID) that generates a control signal that determines how much voltage or current should be applied to the motor.The control signal is sent to a power amplifier (such as an H-bridge) that converts it into an appropriate voltage or current level for driving the motor.The motor then rotates or moves according to the control signal and changes its position, speed, or torque, and sends a new feedback signal to the controller.The process repeats until the error signal becomes zero or negligible, indicating that the output shaft has reached the desired setpoint.

BUZZER

An audio signalling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.

Buzzer Pin Configuration

The **pin configuration of the buzzer** is shown below. It includes two pins namely positive and negative. The positive terminal of this is represented with the '+' symbol or a longer terminal. This terminal is powered through 6Volts whereas the negative terminal is represented with the '-'symbol or short terminal and it is connected to the GND terminal.

IV. RESULT ANALYSIS

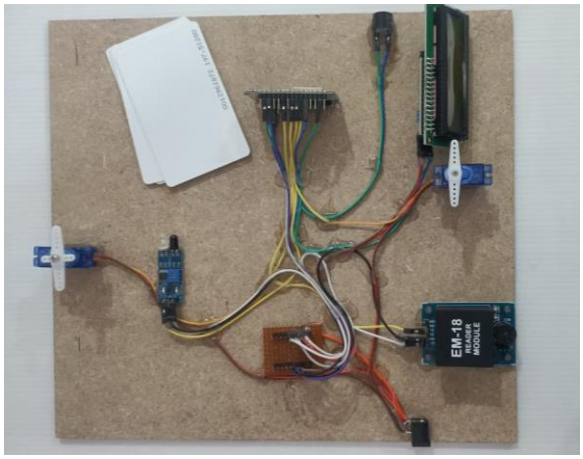


Fig1: project kit

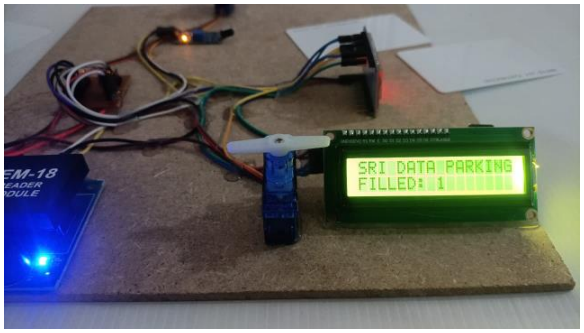


Fig2: 1 slot filled

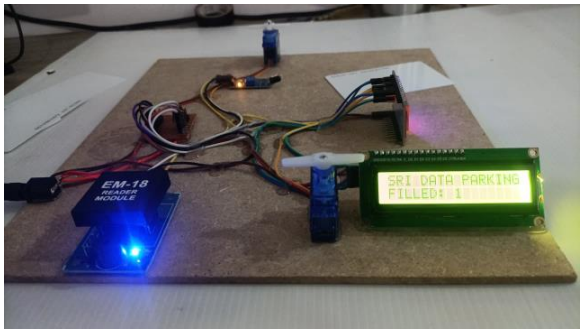


Fig3: 1 vechicle left , so only 1 slot only filled

V. CONCLUSION

The Smart Parking Management System with Auto Payment and IoT Data Storage demonstrates how automation and IoT can revolutionize urban parking management. By integrating RFID-based entry, automatic fee deduction, and cloud-based data storage, the system ensures efficiency, transparency, and convenience. Its scalability and cost-effectiveness make it suitable for deployment in diverse urban environments, contributing to smarter and more sustainable cities.

REFERENCES

[1] A. M. Said, A. E. Kamal, and H. Afifi, "An IoT Assisted Intelligent Parking System (IPS) for Smart Cities," Procedia Computer Science, 2023. Available: https://dl.acm.org/doi/10.1016/j.procs.2023.01.084.

[2] S. Shenoy Basty, R. Kiwad, S. Vittal, and M. M. Ullah, "RFID based Smart Parking System," International Journal of Engineering Research & Technology (IJERT), vol. 11, no. 7, pp. 1-5, July 2022. Available: https://www.ijert.org/research/rfid-based-smart-parking-system-IJERTV11IS070127.pdf

[3] P. Rajesh, S. Mahesh, B. Bheemesh, V. Hemanth, and K. Adarsh, "RFID Based Smart Vehicle Parking System Using IoT," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 9, no. 5, pp. j44-j48, May 2022. Available: https://www.jetir.org/papers/JETIR2205A07.pdf

[4] M. Shaikh, A. Ansari, and H. Phudinawala, "IoT-Based Smart Parking System," International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 10, no. IV, pp. 2450–2456, April 2022. Available: https://www.ijraset.com/research-paper/iot-based-smart-parking-system-usingnode-mcu-esp8266

[5] U. Yahya et al., "RFID-Cloud Integration for Smart Management of Public Car Parking Spaces," arXiv preprint arXiv:2212.14684, December 2022. Available: https://arxiv.org/abs/2212.14684

[6] R. Grodi, D. B. Rawat, and F. Rios-Gutierrez, "Smart Parking Systems: Comprehensive Review Based on Various Aspects," National Center for Biotechnology Information (NCBI), 2021. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC8141779/

[7] N. J. Shah and S. Thakkar, "RFID Based Smart Parking System," International Research Journal of Engineering and Technology (IRJET), vol. 8, no. 5, pp. 542-544, May 2021. [Online]. Available: https://www.irjet.net/archives/V8/i5/IRJET-V8I5115.pdf

[8] M. Assim and A. Al-Omary, "A Survey of IoT-Based Smart Parking Systems in Smart Cities," IET Conference Proceedings, 2021. Available: https://digital-library.theiet.org/doi/10.1049/icp.2021.0911

Poojari, R. Frameworks for Data Management and Lineage in Large-Scale Healthcare Data Systems.

Maturi, S. Y. (2024). Cryptographic privacy engines: Practical multi-party protocols for confidential database queries. Nanotechnology Perceptions, 20(S13), 2770–2785

Gajula, S. (2024). Adaptive zero trust architecture for securing financial microservices. Computer Fraud & Security, 2024(12), 643–655. https://doi.org/10.52710/CFS.845

Akinapalli, S. (2025). Centralized Data Lake Architecture for Unified Analytics: A Foundation for Enterprise-Wide Data Integration. Journal Of Engineering And Computer Sciences, 4(8), 414-422.

Adabala, P. K. (2024). Utilizing predictive analytics to improve efficiency and decision-making in ERP-connected supply chains. International Journal of Intelligent Systems and Applications in Engineering, 12(22s), 2465

Maturi, S. Y. -(2024). Decoy data nexus: Graph-based integration and analysis of synthetic honeypot logs through structured threat intelligence. International Journal of Computational and Experimental Science and Engineering (IJCESEN), 10(4), 4255–4261. https://doi.org/10.22399/ijcesen.5010

Pavan Kumar Adabala. (2026). Best Practices for Enterprise System Integration in Modern Organizations. Journal of Information Systems Engineering and Management, 11(2s), 1137–1146. https://doi.org/10.52783/jisem.v11i2s.14558

Srikanth Kavuri. (2022). Large Language Model (LLM)-Based Automation for Software Test Script Generation. Computer Fraud and Security. https://doi.org/10.52710/cfs.836

Shashank A. (2025). Metadata-driven data integration framework: Automating enterprise data integration through declarative approaches. European Modern Studies Journal, 9(4), 9.

Gajula, S. (2025). Cybersecurity in Supply Chain Management: Role of Identity and Access Management, Zero Trust, and Blockchain. Asian Journal of Computer Science Engineering (AJCSE), 10(2), 1-11.

Harshitha, G. K., Nandigama, C., & Thiripalu, P. (2026). An exploration into identification of opportunities and challenges of establishing and running an enterprise in the area of biofuels. Minnesota Journal of Business Law and Entrepreneurship, 2026(1), 1159–1168.

Susarla, R. S., Boyapati, P. K., & Kandula, S. T. R. (2025, July). Cloud-Based Secure Data Storage in Smart Cities Using Central-Smoothing Hypergraph Neural Networks. In 2025 IEEE 4th World Conference on Applied Intelligence and Computing (AIC) (pp. 279-284). IEEE.

Boyapati, P. K. Building a centralized data operations hub for healthcare enterprise integration. IJSAT-Int. J. Sci. Technol. 16 (2). https://doi.org/10.71097/IJSAT.v16.i2.3219

Garapati, R. S., Aitha, A. R., Yandamuri, U. S., Gottimukkala, V. R. R., Nagubandi, A. R., & Kolla,

S. H. (2026, March). Cloud-Native Orchestration of Multi-Counterparty Derivatives and Collateral in Manufacturing Enterprises via AI-Assisted Financial Audit Engines. In 2026 IEEE International Conference on AI Engineering and Innovations (AIEI) (pp. 1-6). IEEE.

Kumar, Anuj. "Optimization of Process Parameters in Metal Additive Manufacturing for Defect Reduction Using Machine Learning." International Journal of Engineering Research and Science & Technology, Vol. 14, No. 1, 2018, pp. 41-60

Teja, T. V., Bharadwaj, D., Surabhi, K. R., Pokala, H. K., & Kavithamani, B. (2026, April). AI-Driven Quality of Service in Wireless Sensor Network Using Dueling Double Deep Q-Network with Lyapunov Drift-Plus-Penalty Method for Mobile Networks. In 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN) (pp. 1-6). IEEE.

Kumar, Anuj. "Battery Thermal Management Systems for Electric Vehicles Using Phase Change Materials." International Journal of Engineering, Science and Mathematics, Vol. 10, Issue 3, 2021, pp. 202-211

Pokala, H. K. (2026, April). A Secure CI/CD Pipeline using GitHub Actions and Open Policy Agent (OPA) for Kubernetes Applications. In 2026 International Conference on Artificial Intelligence, Systems, and Emerging Technologies (ICAISET) (pp. 1-4). IEEE.