**IoT BASED SMART PARKING SYSTEM**

*A project*

*submitted in partial fulfillment for the*

*Award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**ELECTRICAL ENGINEERING**

***Submitted By***

|  |  |
| --- | --- |
| Himanshu Raj | GEE/1932524 |
| Anuj Kumar | GEE/1932523 |
| Srishti Bhardwaj | GEE/1840105 |
| Vivek Kumar | GEE/1932514 |
| Anshika Sharma | GEE/1932516 |

***Under the guidance of***

**Dr. Manmohan Singh**

**Associate Professor**

**Department of Electrical & Instrumentation Engg.**

# **Department of Electrical And Instrumentation Engineering**

# **Sant Longowal Institute of Engineering and Technology, Longowal**

May 2022

**Declaration**

I hereby declare that project entitled “**IoT Based Smart Parking System**” submitted by us to the Department of Electrical And Instrumentation Engineering, ***Sant Longowal Institute of Engineering and Technology (SLIET), Longowal, Punjab*** during the degree of **B.E in** **Electrical Engineering** is a record of bonafide work carried out by us under the guidance of Dr. Manmohan Singh.

I further declare that the work reported in this **Major Project** has not been submitted & will not be submitted either in part or in full for the award of any degree in this institute.

|  |  |
| --- | --- |
| Himanshu Raj | GEE/1932524 |
| Anuj Kumar | GEE/1932523 |
| Srishti Bhardwaj | GEE/1840105 |
| Vivek Kumar | GEE/1932514 |
| Anshika Sharma | GEE/1932516 |

**Acknowledgment**

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

It is our greatest pleasure to thank prof. DR. SURITA MAINI [HOD, DEPARTMENT OF EIE (SLIET)] for providing us heart full encouragement support and allowing us to work in such a resourceful lab of this esteemed institute and thereby fulfilling one of our dreams.

We whole heartedly thank my project guide DR. MANMOHAN SINGH for consistent guidance, expert academic and support throughout the project, without his great concepts and inspiration it would have been impossible.

We thank our parents for their emotional and financial support which they provided during this project.

We show gratitude to our Honorable Director PROF.Dr.SHAILENDRA JAIN SIR,for providing all the facilities and support.

We thank all the faculties who directly and indirectly helped us in the completion of this project.

**Certificate**

This is to certify that the project entitled “ IoT Based Smart Parking System”

Submitted by ~

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **NAME OF STUDENTS** | **REG.NO** |
| **1.** | SRISHTI BHARDWAJ | GEE/1840105 |
| **2.** | HIMANSHU RAJ | GEE/1932524 |
| **3.** | ANUJ KUMAR | GEE/1932523 |
| **4.** | ANSHIKA SHARMA | GEE/1932514 |
| **5.** | VIVEK KUMAR | GEE/1932516 |

It is bonafide work carried out by these students under the guidance of DR. MANMOHAN SINGH at SLIET during academic year [2018-2022]. It has been accepted and approved for partial fulfillment of the requirement of SLIET University,Punjab,for the award of the degree Bachelors of Technology (Electrical).This work and project report has not been submitted to any other institute or university for the award of any degree or diploma.

**DR. MANMOHAN SINGH DR. SURITA MAINI**

**[ PROJECT GUIDE] HOD**

**DEPARTMENT OF EIE DEPARTMENT OF EIE**

**PROF.DR.SHAILENDRA JAIN**

**DIRECTOR,SLIET**

**PROF. DATED :**

**[EXTERNAL EXAMINER]**

**Content Table**

|  |
| --- |
| **Chapter 01: Introduction** |
| |  |  | | --- | --- | | 1.1 | Introduction……………………………………………………… 07-08 | | 1.2 | Project approach…………………………………………………. 09-09 | |
| **Chapter 02: Literature Review** |
| |  |  | | --- | --- | | 2.1 | IoT and IoT Cloud Services………………………………………10-10 | | 2.2 | HTTP and MQTT Network protocol……………………………..11-12 | | 2.3 | Web Application and API………………………………………...12-13 | | 2.4 | N0deMcu Esp8266……………………………………………….13-13 | | 2.5 | Ultrasonic Sensor…………………………………………………13-14 | |
| **Chapter 03: IoT device (Sensor Node)** |
| |  |  | | --- | --- | | 3.1 | Description and Block Diagram…………………………………...15-15 | | 3.2 | Circuit diagram…………………………………………………….16-16 | | 3.3 | Working Operation………………………………………………...17-18 | | 3.4 | Physical Model Design…………………………………………….18-20 | |
| **Chapter 04: IoT cloud** |
| |  |  | | --- | --- | | 4.1 | Description and Flow Diagram …………………………………...21-21 | | 4.2 | MQTT Request via ESP8266……………………………………..21-22 | | 4.3 | API Request using JavaScript…………………………………….22-22 | | 4.4 | Web Dashboard…………………………………………………...23-23 | |
| **Chapter 05: Web application** |
| |  |  | | --- | --- | | 5.1 | User Interface……………………………………………………. 24-25 | |
| **Chapter 06: Future scopes** |
| |  |  | | --- | --- | | 6.1 | Scope of Additional Development…………………………………26-28 | | 6.2 | Scope as a Product…………………………………………………29-30 | |
| **Conclusion------------------------------------------------------------------------ 31** |
| **References------------------------------------------------------------------------ 32** |
| **Appendixes----------------------------------------------------------------------- 33-40** |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Sno.** | **Name of Figure** | **Page No.** |
| 1 | Parking Activities | 09 |
| 2 | Block diagram of the General IoT devices | 10 |
| 3 | Block diagram of Sensor node operation | 15 |
| 4 | Circuit Diagram of sensor node | 16 |
| 5 | Flow chart of publishing data to adafruit Io | 17 |
| 6 | Top View of Model | 18 |
| 7 | Left View of Model | 19 |
| 8 | Front View of Model | 19 |
| 9 | Overall Proposed System | 20 |
| 10 | Main Panel Dashboard with console Log | 23 |
| 11 | Feed Graph image of SPP01 | 23 |
| 12 | User Interface images | 24-25 |

**CHAPTER 01: INTRODUCTION**

* 1. ***INTRODUCTION***

***Abstract:*** *Ongoing rapid urbanization has potentially improved the well-being of our societies. Rapid urbanization is also leading to new problems that never existed before one of them is parking issues related to commercial and institutional buildings parking is one of the key components of transportation programs all over the world. With the increasing percentage of vehicle ownership in recent times, parking has become a conflicting and confusing situation for a lot of people. Most often people cannot find suitable or adequate space to park their vehicles and end up encroaching the roads. Particularly in the cities and the big towns, this problem is increasing gradually where the supply-demand ratio makes parking a problem for parking space providers, motorists, or both.*

*The main objective of our project is to tackle the above-mentioned problems, by designing and implementing a system where users can find the perfect parking spot with a single click on a Web application. Our project is “IoT Based Smart Parking System” in this project we place a small IoT device at each parking spot. Every IoT device updates data to the IoT cloud whether this position is occupied or not by detecting the presence of vehicles over it. For the user side, we provide a web application that fetches all data via API call to the IoT cloud and runs an algorithm to provide the best and nearest parking spot to the user. For future extensions of this project, users can book spots in advance. These all features combined provide a smooth motoring experience in cities and reduce traffic congestion which creates due to not finding parking spots*

**T**he recent growth in economy and due to the availability of low-price cars in the market, every average middle-class individual can afford a car, which is good thing, however the consequences of heavy traffic jams, pollution, less availability of roads and spot to drive the motor car. One of the important concerns, which is to be taken in accounting, is that problem of parking those vehicles. Though, if there is space for parking the vehicle but so much time is squandered in finding that exact parking slot resulting in more fuel intake and not also environment friendly.

It will be great deal if in some way we find out that the parking itself can provide the precise vacant position of parking slot then it'll be helpful not limited to the drivers also for the Traffic congestion caused by vehicle is an alarming problem at a global scale and it has been growing exponentially. Car parking problem is a major contributor and has been, still a major problem with increasing vehicle size in the luxurious segment and confined parking spaces in urban cities. Searching for a parking space is a routine (and often frustrating)

activity for many people in cities around the world.

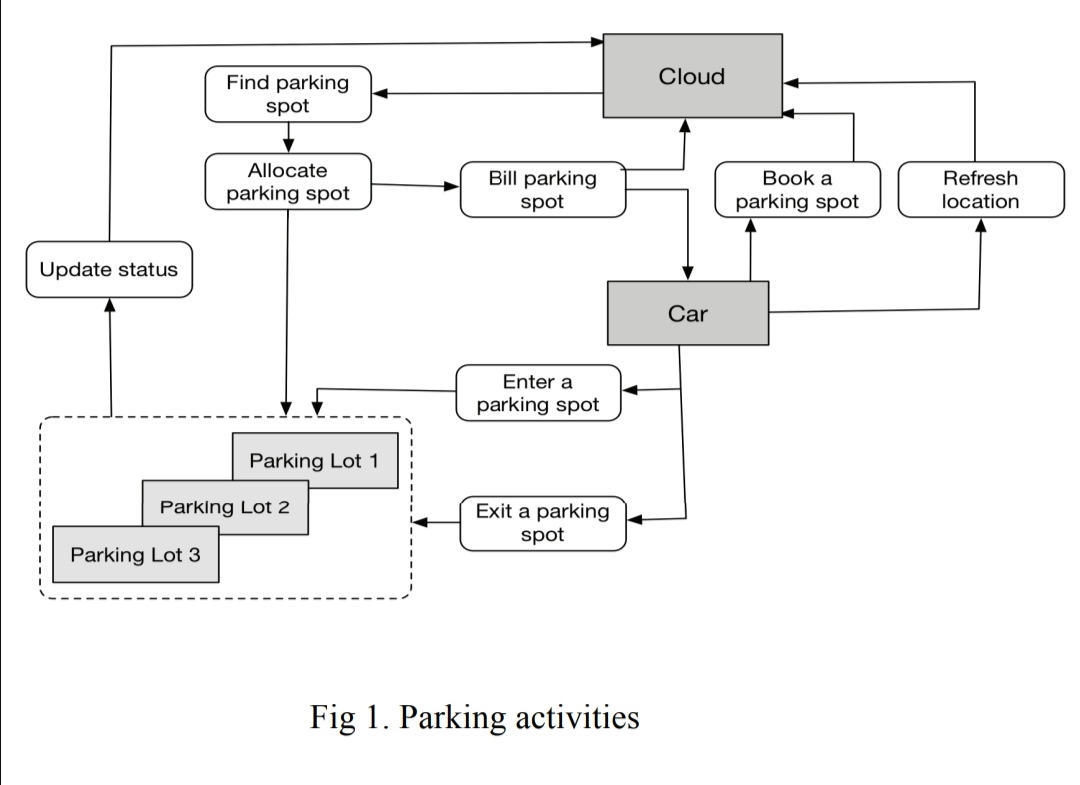
This search burns about one million barrels of the world’s oil every day. As the global population continues to urbanize, without a well-planned, convenience-driven retreat from the car these problems will worsen.

The existing doesn’t have the proper security system and the customers need to park their cars should search and then park but whereas in this project it’s clear that they will notified while they are out of the parking slot the maximum cars that could be able to park and if the parking slot is full they may wait still the car comes out or else they can move to any other parking slots next to them. So, this system makes the user to move from outside of the parking slot which is better than coming inside and searching and going out. This parking system will provide the best security and it will give the best parking experience and a flexible move to the customers park their cars or vehicles in this intelligent parking system lots.

***1.2 PROJECT APPROACH***

Driven by the need to expand parking inventories and the prospects of new business models for parking services, this work-in-progress envisions to extend parking-systems’ cloud architecture to house new opportunities offered by third-parties labelled parking service provides or PSP. PSP is a new business entity serving individual parking owners or intermediaries, which register their parking lot in the cloud directory of available parking spaces. This information is relayed to motorists who request parking spaces, after which the transaction is fulfilled directly between the service provider and seeker.

To enable this vision, a thorough analysis of parking activities shown in *Fig - 1* is investigated to evaluate automation opportunities. Car-mounted or augmented devices are used as communication interfaces to interact with the cloud services and parking infrastructures. The analysis of parking-related needs and problems raise the requirement to adopt a modular research methodology that examines the state of the art for each element of the solution build-up shown in *Fig-2* and discussed further next.



*Fig-1 Parking Activities*

**CHAPTER 02: LITERATURE REVIEW**

***2.1 IoT AND IoT CLOUD SERVICES:***

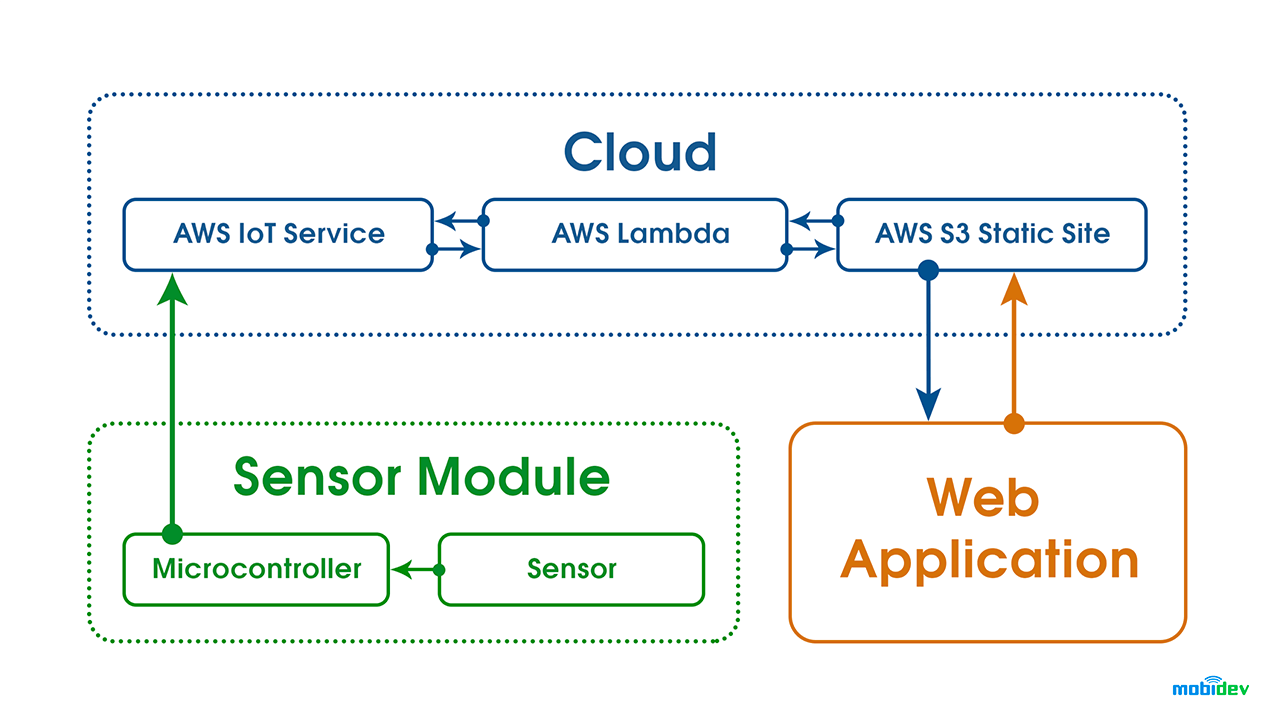
Internet of Things (IOT) uses sensors to connect physical parking space infrastructures with information and communication technologies, where cloud-based smart management services are provided. To implement this concept a mobile based application would be developed.

Parking systems are installed on the outside of buildings or inside of buildings. When a vehicle enters the space, sensors detect its presence and calculate available parking slots. This information is then sent to the driver’s phone via an app.

The smart parking system also has real-time data on occupancy rates, which can be found on the app. This data is collected from each sensor and is updated every five minutes**.**

Smart parking development implies an IoT-based system that sends data about free and occupied parking places via web/mobile application. The IoT-device, including sensors and microcontrollers, is located in each parking place. The user receives a live update about the availability of all parking places and chooses the best one.

In order to investigate technologies behind the smart parking solution, we implemented an internal research project. The main idea was the creation of smart parking using the Internet of Things and ultrasonic sensors, where available parking places could be displayed in a web application.

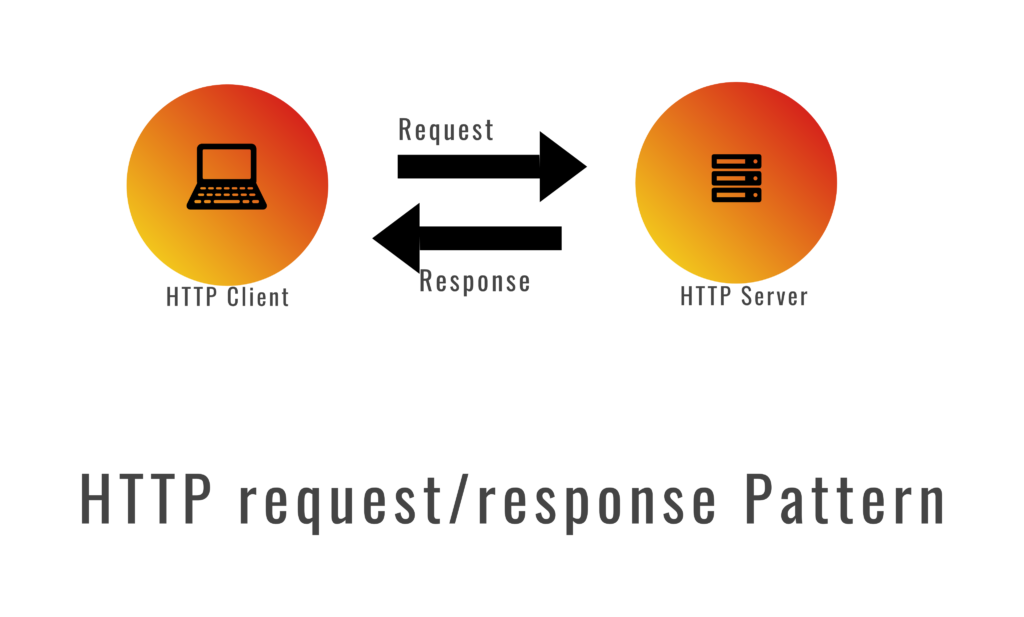


*Fig -2 Block diagram of the General IoT devices*

***2.2 HTTP AND MQTT NETWORK PROTOCOL :***

***Hyper Text Transfer Protocol***

HTTP is a universal web protocol used with HTML to communicate with web browsers. It needs more RAM & Power resources. It is being used vastly for different applications that work in request/response patterns. It works in P2P(peer to peer) technology means the client/server model.



***Message Queuing Telemetry Transport***

MQTT is the lightweight network protocol for resource-constrained devices. A low 2byte header and 256MB message size make the MQTT protocol a perfect use case for IoT/IIOT devices. MQTT Protocol is used in Smart home systems, smart cities, Industrial machines, vehicle tracking, and other applications. MQTT vs HTTP is important to understand to implement IoT applications.

MQTT works on the publish/subscribe pattern, it publishes a Topic with payload to MQTT Broker & then any of the other MQTT Clients can subscribe to the same topic. So, It’s not dependent on other clients if any of the clients stopped working.

Even LWT(Last will testament ) option would publish a message when an MQTT Client disconnected from the network to all the subscribed MQTT Clients.



There are 3 levels of quality of service (QoS) in MQTT:

QoS 0 – at most once – ensures the best possible delivery without acknowledgment.

QoS 1 – at least once – ensures that the message must deliver at least once, but it can deliver more than once. It is the best possible method to be used with most of the applications to reduce network latency and usages.

QoS 2 – only once – ensures that the message reaches a subscriber & receives the acknowledgment until it publishes continuously.

Retained messages are the other functionality of MQTT where MQTT Broker retains the message with retain flag from MQTT Client and keep it retained until it has cleared with the retain flag.

MQTT vs HTTP shows the importance of IoT Applications as just QoS is important for MQTT to publish/subscribe the messages.

This system needs to dynamically assign IP address to each node and users so a connection which is DHCP enabled is used. For a secure and fast connection MQTT protocol is used along with TCP (wired connection).

As HTTP is often too verbose and works on point to point communication method but what actually needed is quality of service delivery, bandwidth efficient, data agnostic and continuous session awareness, which is totally satisfied by MQTT.

This protocol works on pub/sub (publish and subscribe) basis, which allows one to many communication and in this the clients are unknown to each other and every client can be a publisher or a subscriber. It guarantees delivery of service and buffering

as well as retaining of messages. This protocol needs a broker to deal between the clients or between the publisher and subscriber.

***2.3 WEB APPLICATION AND API :***

***Web Application***

A Web application (Web app) is an application program that is stored on a remote server and delivered over the Internet through a browser interface. Web services are Web apps by definition and many, although not all, websites contain Web apps.

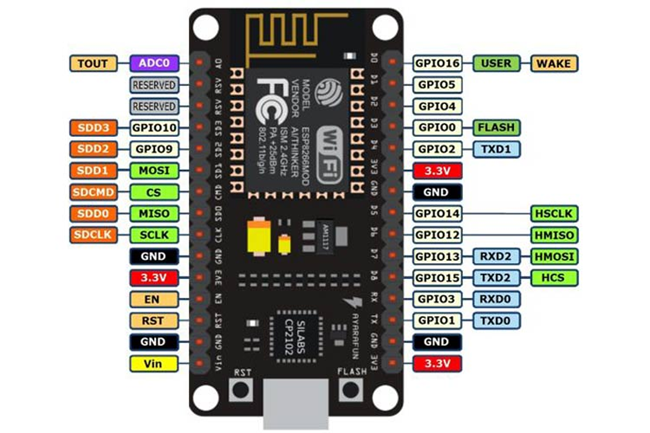
Web applications do not need to be downloaded since they are accessed through a network. Users can access a Web application through a web browser such as Google Chrome, Mozilla Firefox or Safari.

For a web app to operate, it needs a Web server, application server, and a database. Web servers manage the requests that come from a client, while the application server completes the requested task. A database can be used to store any needed information.

***API***

API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. In Context of our project Api is used to communicate between IoT cloud and Web application.

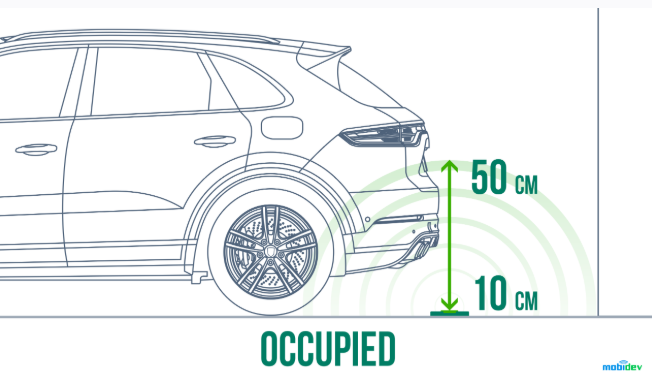
***2.4 NODEMCU ESP 8266:***

It is a development board and open source based firmware mainly designed for IoT based applications. It is having firmware which runs on ESP8266 with WiFi SOC (system on chip),and hardware which is based on ESP-12 module. ESP8266 is the most popular controller to build IoT based applications as it has inbuilt support for Wi-Fi to connect to internet.

***2.5 ULTRASONIC SENSOR:***

A great ultrasonic sensor is a device that measures the length for an object using ultrasonic sound waves. An ultrasonic sensor runs on the transducer to receive and send ultrasonic pulses that relay back again information about an object’s proximity. High-frequency audio waves reveal from boundaries to create unique echo patterns

IoT sensors use an ultrasonic wave to detect the distance to something. Each sensor is embedded in the parking space surface and detects the distance to the undercarriage of a vehicle if the parking space is occupied.



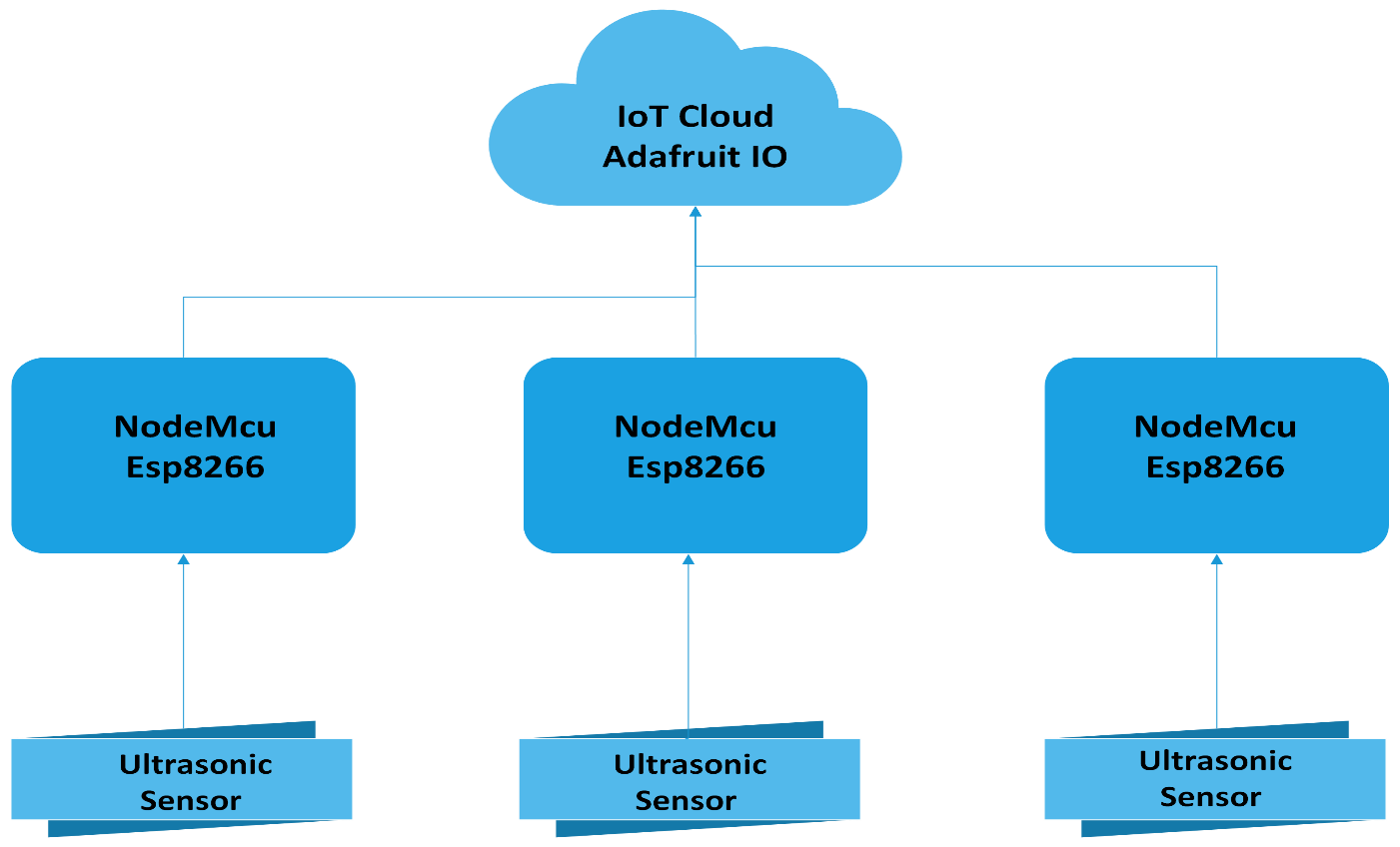
**CHAPTER 03: IOT DEVICES [SENSOR NODE]:**

***3.1 DESCRIPTION AND BLOCK DIAGRAM***

The Smart Parking industry has seen several innovations such Smart Parking Management System, Smart Gate Control, Smart Cameras which can detect types of vehicles, ANPR (Automatic Number Plate Recognition), Smart Payment System, Smart Entry System and many more. Today similar approach will be followed, and a smart parking solution will be built which will use an ultrasonic sensor to detect vehicle presence and upload the same into the IoT cloud and mark the place the as occupied.

So, when the other user search for the parking this space will not appear in the list. Here for data upload main controlled used is the NodeMcu esp8266 to controller for controller of other peripherals connected to it.

For Our case it is only ultrasonic sensor only and for internet connectivity built in Esp8266 is used.

******

*Fig 3– Block diagram of Sensor node operation*

***3.2 CIRCUIT DIAGRAM:***

Below we show a circuit diagram of our system that will be clarifying the system very well.

*Note : Battery and power switch is excluded from the circuit.*

*Fig 4; Circuit diagram of Sensor Node model*

Diagram

Description automatically generated

***3.3 WORKING OPERATION:***

Operation of the Sensor node is quite simple but most important. The main working of the

Sensor node is to upload data to the iot cloud where data is processed for the further used by web application to provide smooth mobility experience.

Here for data upload portion NodeMcu act as brain and handle all process and control peripherals. This board is powered by 9Volt Battery and program structure if explained below using block diagram. Diagram

Description automatically generated

*Fig5 – flow chart of Publishing data to adafruit io*

So, according to the flowchart as we can observe firstly the nodeMCU requests the WiFi for connection. As soon as it connects, it will pick up the API from adafruit servers and for that it will have to ping to adafruit servers .

After the ping got successful, adafruit servers establishes the connection with internet or device, then the sensors will take the input from the environment and publish them on servers for which we can again do remote monitoring in the adafruit UI box.

And if in case data can’t be published then it will observe halt for 5 second and still if data is not published then it will come outside to ping again and if it published then exception got closed and we got the result at adafruit UI.

Also if there is no wifi connection at the starting then it will reconnecting again and again after every 5 sec until it got published.

***3.4 PHYSICAL AND PROPOSED DESIGN:***

In This section discussed our 3D model sensor node iot devices. The major component of our 3D model is a nodemcuesp8266 device and HC-SR04 Ultrasonic sensor and battery pack for Powering our devices. To reducing the designing complexity, we have not shown wire in 3D model, but it will be on circuit diagram.

As we can see that there is the 2 holes on the top side for mounting HC-SR04 Ultrasonic sensor measuring the distance to a wide range of objects. There is no cover on its left or right side as the component is visible. 3D design board has a transparent front cover. As you can see from front view there is rectangular hole in front for programming and debugging purpose.

***Design of Model images from different Perpective***

***Fig6 - Top view***

Graphical user interface

Description automatically generated with low confidence

***Fig7 - Left view***

A picture containing text

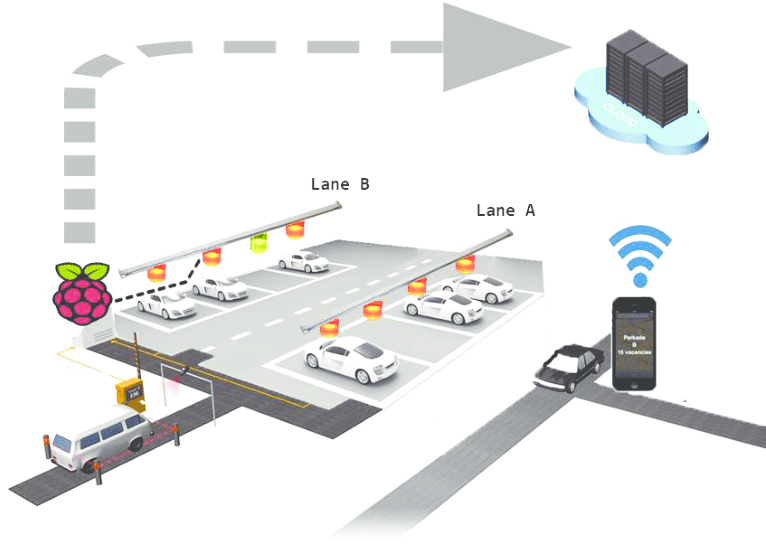
Description automatically generated

***Fig8: Front view***

A picture containing text

Description automatically generated

***Fig9 - Overall proposed System***

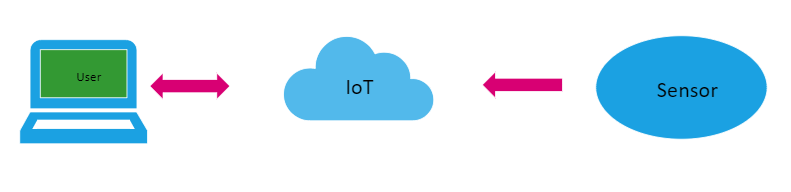


**Chapter 04: IoT cloud**

***4.1 IoT cloud***

An IoT cloud is a massive network that supports IoT devices and applications. This includes the underlying infrastructure, servers and storage, needed for real-time operations and processing. An IoT cloud also includes the services and standards necessary for connecting, managing, and securing different IoT devices and applications.

In context of our Project IoT cloud play critical role in the system acting as the common connecting point and also store data of the current status of each parking spot.



In the first sensor uploads data to the cloud and updates data after each specific time interval when the user login to the website and requests to find space for them based on the user demand of the data of adafruit return with the response and response is parsed into the required user format and displayed to the user

***4.2 MQTT Request via ESP8266***

Adafruit provide a library for making mqtt request from development board like arduino or nodemcu etc MQTT, or message queue telemetry transport, is a protocol for device communication that Adafruit IO supports. Using a MQTT library or client you can publish and subscribe to a feed to send and receive feed data.

To use the MQTT API that Adafruit IO exposes you'll need a MQTT client library. For Python, Ruby, and Arduino you can use Adafruit's IO libraries as they include support for MQTT. For other languages or platforms look for a MQTT library that ideally supports the MQTT 3.1.1 protocol.

For establising Connection Recommended settings are following.

|  |  |
| --- | --- |
| **Host** | io.adafruit.com |
| **Secure (SSL) Port** | 8883 |
| **Insecure Port** | 1883 |
| \*\*MQTT over Websocket | 443 |
| **Username** | Your Adafruit IO Username |
| **Password** | Your Adafruit IO Key |

Approach to publish data to any feeds refer **Appendix 1**.

Api documentation link: https://io.adafruit.com/api/docs/mqtt.html?ruby#adafruit-io-mqtt-api

**4.3 API Request using JavaScript**

Javascript provide native support to the http so in this is case we directly give get request via browser. JavaScript has great modules and methods to make HTTP requests that can be used to send or receive data from a server-side resource. In this article, we are going to look at a few popular ways to make HTTP requests in JavaScript.

Ajax is the traditional way to make an asynchronous HTTP request. Data can be sent using the HTTP POST method and received using the HTTP GET method. Let’s look and make a GET request. I’ll be using JSONPlaceholder, a free online REST API for developers that returns random data in JSON format.

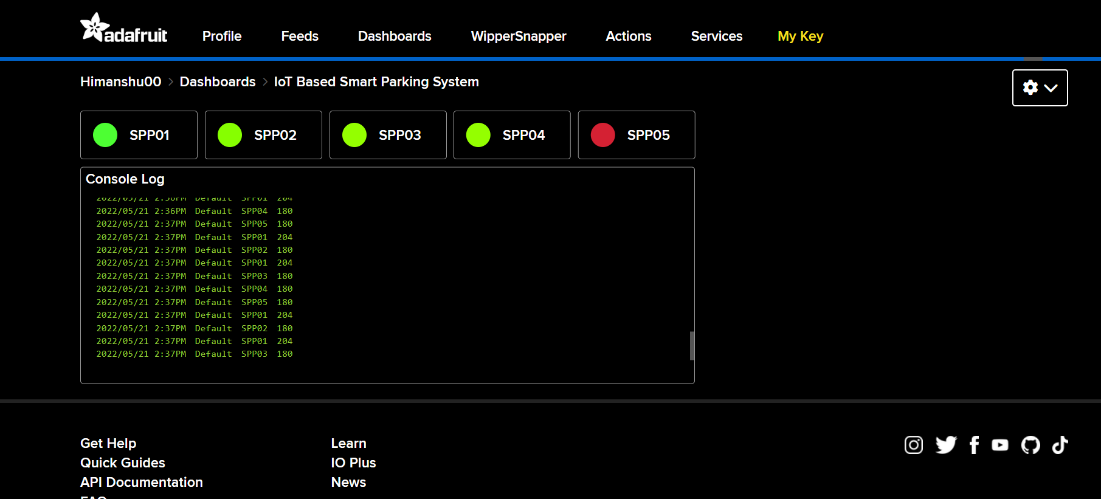
To make an HTTP call in Ajax, you need to initialize a new XMLHttpRequest () method, specify the URL endpoint and HTTP method (in this case GET). Finally, we use the open () method to tie the HTTP method and URL endpoint together and call the send () method to fire off the request.

< Code />

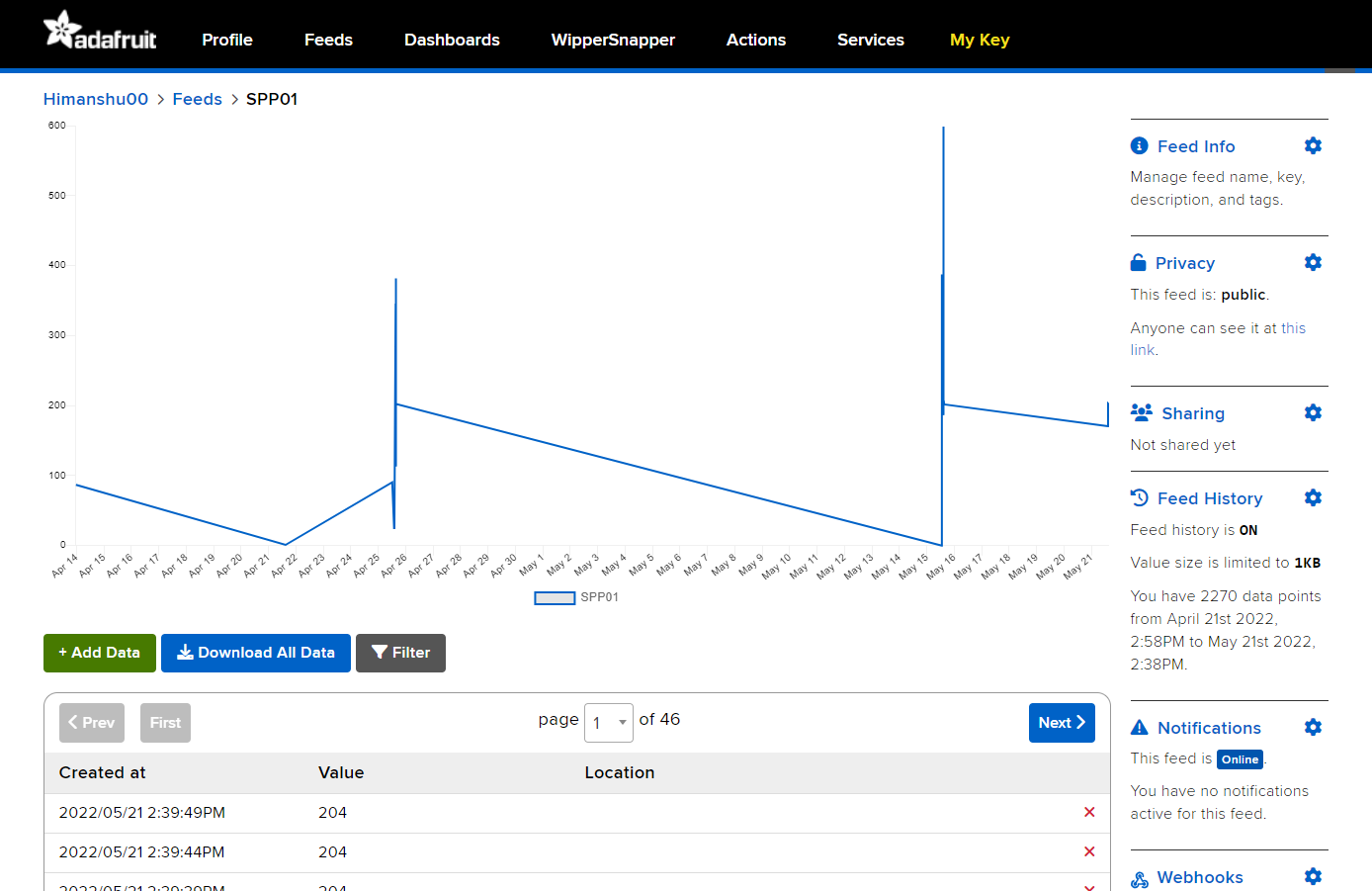
const Http = new XMLHttpRequest();  
const url='https://jsonplaceholder.typicode.com/posts';  
Http.open("GET", url);  
Http.send();  
  
Http.onreadystatechange = (e) => {  
 console.log(Http.responseText)  
}

***4.4 Web Dashboard of the adafruit io***

**Fig10 - Main Panel Dashboard with console Log**



***Fig11 – Feed image of SPP01***



**Chapter 05: Web Application**

Link to Application: <https://himanshuraj-in.github.io/FindParking.github.io/>

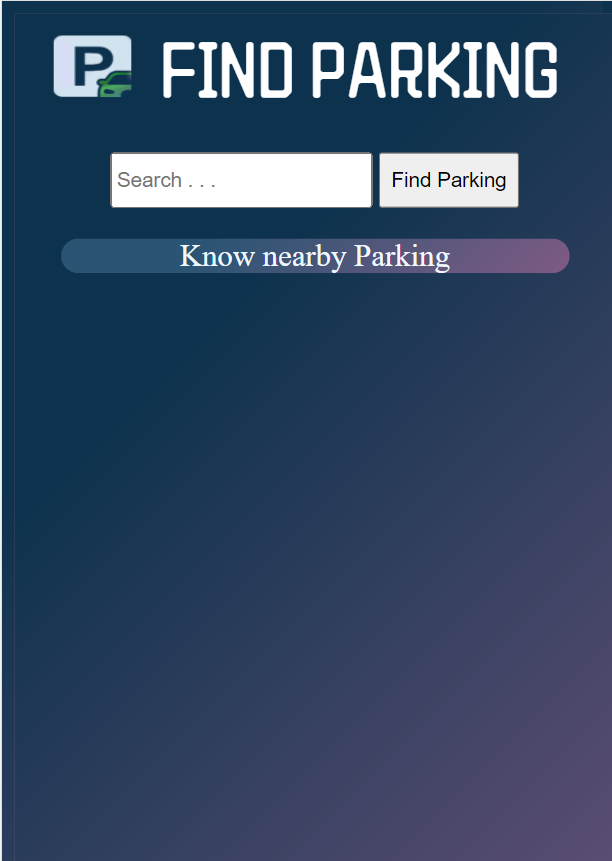
***User Interface***

We have very simple User Interface for finding parking in interested location for finding

Users must enter the interested location either by selecting from drop-down or by entering location name into the Search box Then click on the find parking button After few milliseconds

Search result will start appearing with respective location After few milliseconds search result will start appearing with respective location

*Fig12 – UI Images step Wise*

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generatedA picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

**Chapter 07: Future Scopes**

***7.1 Scopes For additional Development***

**The Future of Smart Parking Systems: Creating Smarter Mobility Networks.**

Vehicular parking lies at the intersection of urban space and mobility management. The first generation, or Parking 1.0, hardly had any service offerings except for a simple space-renting model that was managed and operated manually.

Parking 2.0 offered electronic services such as parking meters that partially automated the fee collection and auditing systems.

The present generation, Parking 3.0, provides basic automation that allows users to independently navigate the complete parking life cycle – from knowing the parking occupancy status to ticketing, parking, and fee settlement. The parking industry has witnessed steady growth in the recent past, with the global market size currently exceeding $2 billion. In fact, market analysts have a very bullish outlook on this segment for the next decade, projecting a 1 compound annual growth rate (CAGR) of 17% , of which about 65% would be from software and services. This forecast is based on the observation that the current pace of urbanization and motorization will lead to an explosion in parking demands. WHITE PAPER WHITE PAPER In the current context, increasing the parking capacity would be a challenge due to the growing scarcity of available urban space, while decreasing the demand will subdue citizens’ mobility experience.

It is here that the premise of Parking 4.0 brings together the best of 2 Industry 4.0 and Business 4.0™. Leveraging advanced analytics, experience engineering, and newer business models, Parking 4.0 will increase the efficiency of parking systems with an urban optimization strategy encompassing both space and mobility dimensions.

By the Parking facility managers will be able to optimize the use of space and resources within their parking lots, efficiently strategize and plan future development. Community leaders will increase the comfort of city residents by implementing IoT parking solutions.

**Scope of additional development considering the following factors**

With the increase of the urban population, the need for living and infrastructure space is higher than ever before. On the other hand, the growing number of car owners — it is estimated as there will be over **two billion cars on the road by 2035** — creates a high demand for parking spaces.

Parking management systems help fulfill drivers’ needs without compromising on living and recreation space.

State of the Market In contrast to traditional PMS that provided only basic services, smart PMS are envisioned to offer a whole range of advanced services with a highly cross-functional management tool. Some features include:

* Parking availability monitoring ranging from general to granular information
* Space optimization n Parking guidance and search time reduction based on operator interest, user preference, and so on
* Parking reservation
* Parking demand management WHITE PAPER
* Parking price and policy optimization (mostly dynamic)
* Parking enforcement such as detection of zones, payment, and overstay violations
* Routine services such as reporting, integrated payment support (through mobile apps featuring pay-by-text, pay-by-voice, pay by-phone, and other such functionalities), and system management in terms of configuration and maintenance.

In order to obtain adequate RoI , the trend in the smart parking market has been to reduce the dependence on dedicated hardware and increase utilization of software and analytics. Hence, the current business models are opex -based and rely more heavily on selling premium software and services on a monthly or annual basis. In the recent past, heavyweights from the automotive, automation, ITS, and telecom industries have shown great interest in the smart parking market as part of a broader smart city and IoT strategy. In fact, many automotive OEMs are vertically integrating their portfolio from being a standalone mobility provider to a mobility ecosystem enabler with interests in car sharing, EV charging, parking, and payment solutions. Strategic alliances across these sectors are noteworthy as multi-disciplinary organizations (dealing in parking, mobility, automotive equipment, and even mobile applications) are collaborating to offer integrated solutions and services in the 6 parking industry. It is a win-win situation for all stakeholders in the value chain. For example, parking operators are able to enhance as well as ease their operations with bundled services and applications, while automotive OEMs get an opportunity to build alliances with the parking and car sharing service providers. This is the new normal, and these are the early signs that the current market is ready to embrace Parking 4.0.

**Parking 4.0: Future in the Market and Opportunities**

The parking market and its business models fall into three broad horizontals.

The first horizontal involves the development of specific sensors and techniques for new or more cost-effective observations.

The second is the deployment horizontal, which focuses on building out to the scale of data gathering, data storage, and data curtain needs of parking systems.

The third horizontal model addresses the need for a WHITE PAPER portfolio of analytical techniques to convert the gathered data into actionable information.

While the first two have been the focus of precursor technologies, Parking 4.0 will generate opportunities in the third segment. The parking industry is expected to grow as a smart and green industry with the commoditization of new technologies 7 (especially, in the IoT /M2M/V2X space). With the arrival of new ecosystem players and participants, it is anticipated that the early positive disruption of this market will (or is beginning to) happen with: n Electric vehicles, which add another dimension to parking management with the need for charging (availability of charging stations, time of the day, charging duration, and pricing are becoming key decision parameters) n Autonomous vehicles, which will fundamentally change the usage of cars and their parking modality with self-parking features and robotic valet parking n Urbanization of parking, which will create a highly reactive and real-time ecosystem connecting those who are interested in lending parking spaces with those who are searching for one There is a growing apprehension that the widely prevalent on demand shared mobility practices (such as ride-hailing and car sharing) will negatively disrupt the parking market with 8 reduced parking demands . However, it must be acknowledged that these practices pose complex trade-offs, with no clear winning strategy for all kinds of mobility requirements. Urbanization is a popular mobility method in the last mile, short distance, or occasional commute segment for its economically rewarding and better travel experience. It will, therefore, continue to co-exist with other commute modalities for serving the broader portfolio of mobility demands. In fact, newer vehicle ownership models are being introduced with innovative offerings that epicenter the vehicle as a commodity, thereby resetting the focus onto privately leased vehicles and parking. Hence, in spite of healthy competition, the future seems to be quite promising for smart parking with Parking 4.0. As the larger parking ecosystem matures into an integrated landscape, many of the routine operations are expected to migrate into in-car systems with parking apps, thereby alleviating the burden of deploying parking systems. This could be augmented with ubiquitous or rapidly growing mobility WHITE PAPER infrastructure (such as the electronic toll collection [ETC] transponders) to achieve the desired scale and density at affordable costs, instead of relying on the massive deployment 9 of custom sensor networks. In addition, monetization engines and brokering services that help connect data consumers with data producers could be enabled using the building blocks of data ownership or incentive mechanism through recognition and attribution, barter, or 10 monetary rewards. As with all new innovation paradigms, these 11 possibilities come with their own research challenges. There have been numerous large-scale systems in the past (such as 12 location-based services) that bear a strong resemblance with the overall smart parking philosophy. Hence, learning from their initial failures and subsequent successes is crucial to better articulate workable technology and business roadmaps.

* 1. ***Scope as a product of smart parking system and its benefits***

1. **Environment**

One goal of Smart Parking is to reduce the time taken and the hassle factor of locating an available parking space. Being able to accurately direct a driver to an available space has many environmental benefits; it reduces CO2 emissions, noise, and other pollutants. Smart Parking can be combined with Smart Environment, measuring air quality and parking space availability.

**A picture containing timeline

Description automatically generated**

1. **Convenience.**

It can be frustrating, especially at peak times, driving around town looking for available spaces. The inability for someone to locate a parking space may result in lost custom or influence them to shop at alternative locations. The ability for a shopper or visitor to quickly identify a space reduces the friction and improves the overall experience. The convenience factor is of particular importance for spaces reserved for disabled drivers, public service or emergency vehicles.

1. **Real-Time Data and Insights**

For a local council, car park operator or business – Smart Parking provides you with rich data-sets that can be used to identify trends, peak-times and other metrics that can be used in forecasting and reporting. With bespoke software, the data and sensors can be integrated into city management systems or MI reports.

1. **Reduced Traffic**

When a driver knows exactly where they need to go; it reduces idling and unnecessary driving – therefore optimises traffic flows in built-up areas.

1. **New Business Models**

Smart Parking creates the possibility of new business models that are only made possible using technology. Reward programs, app-based payments and dynamic parking tariffs are just some examples.

1. **Reduced costs and overhead**

Traditional on-street parking may have required investments in parking meters or parking inspectors. Smart Parking technology can reduce these overheads by automated processes and providing targeted enforcement activity.

1. **Enforcement Effectiveness**

Target enforcement means your staff can be directed towards people who have overstayed to take the necessary action.

1. **Safety**

Decreased searching for spaces can reduce accidents by ensuring drivers maintain their attention rather than browsing for spaces or making rash maneuvers.

1. **Integrated Payments**

Smart Parking systems can include real-time and electronic payment methods via an app or a browser. This makes the parking experience far easier and provides more structured data to income streams (i.e. Categorizing revenue by parking facility, area, road, etc).

1. **Smart City**

Smart Parking will soon become a necessity for any city or town wishing to embrace Smart City technologies or standards such as ISO 37122. The UK Government and others are already working towards Smart Parking standards.

Smart Parking can be utilized in **private parking lots, hospitals, hotels, shopping malls, public parking garages, offices**, etc. to make the parking hassle free and time consuming.

**CONCLUSION**

**“**

Due to advancement in technology, especially the drivers are demanding the parking facilities which are easier and less time consuming. The main benefits of this IoT based smart parking system are time and fuel saving. It can also provides an ecofriendly and sustainable parking management system. It leads to the reduction of hassle in the parking areas and traffic jams. Most importantly it is simple, economic and helps in a greater way for the reduction of carbon in the atmosphere. Thus, air pollution is reduced. With the help of web browser, it becomes easier for the users to access and find the parking slots at any remote location in advance before reaching to the destination which thereby avoids unnecessary travelling of cars in the parking area, which is already filled, so our project is cost effective also. With the help of this project, we indicated the value-added services to both car drivers and parking service providers. It leads to more economic use of parking space to the parking owners which is represented as economic resource in the parking management system.

Therefore, we should implement this project and helps to develop our city with technological advancement.

**”**

**References**

* **Adafruit IO MQTT API -** [**https://io.adafruit.com/api/docs/mqtt.html?ruby#adafruit-io-mqtt-api**](https://io.adafruit.com/api/docs/mqtt.html?ruby#adafruit-io-mqtt-api)
* **Adafruit IO HTTP API –** [**https://io.adafruit.com/api/docs/#adafruit-io-http-api**](https://io.adafruit.com/api/docs/#adafruit-io-http-api)
* **JavaScript Documentation -** [**https://developer.mozilla.org/en-US/docs/Web/JavaScript**](https://developer.mozilla.org/en-US/docs/Web/JavaScript)
* **HTML** [**https://www.w3schools.com/TAgs/default.asp**](https://www.w3schools.com/TAgs/default.asp)

**Appendixes**

***Appendix: 1 Program in NodeMcu Esp8266***

/\*

\* Author : Himanshu Raj

\* Date : 25-07-2022

\* E-mail : himanshuraj9194@gmail.com

\* Contributed to Project : IoT Based Smart Parking System

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Libraries \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ESP8266WiFi.h>

#include "Adafruit\_MQTT.h"

#include "Adafruit\_MQTT\_Client.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WiFi Credentials \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define WLAN\_SSID "JioFiber-ief8V" // WiFi SSID(service set identifier)

#define WLAN\_PASSWORD "Him1932524" // WiFi Password

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Adafruit IO Credentials \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define IO\_SERVER "io.adafruit.com"

#define IO\_SERVERPORT 1883

#define IO\_USERNAME "Himanshu00"

#define IO\_KEY "aio\_JXCV18iXGjdyZYxaDtg70QiZqLLR"

WiFiClient client**;**

Adafruit\_MQTT\_Client mqtt**(&**client**,** IO\_SERVER**,** IO\_SERVERPORT**,** IO\_USERNAME**,** IO\_KEY**);**

Adafruit\_MQTT\_Publish SPS\_Data **=** Adafruit\_MQTT\_Publish**(&**mqtt**,** IO\_USERNAME "/feeds/spp01"**);**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Connect WiFi\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void connectWiFi**()** **{**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Establishing Connection to WiFi \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

WiFi**.**begin**(**WLAN\_SSID**,** WLAN\_PASSWORD**);**

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

**{**

delay**(**500**);**

Serial**.**print**(**"."**);**

**}**

Serial**.**println**(**"WiFi connected"**);**

Serial**.**print**(**"IP address: "**);**

Serial**.**print**(**WiFi**.**localIP**());**

Serial**.**println**(**""**);**

digitalWrite**(**LED\_BUILTIN**,** LOW**);** // Turn On ON board LED (Active Low)

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Connect Adafruit Io Cloud \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void connectAdafruitIo**()** **{**

Serial**.**println**(**"Connecting to Adafruit IO ...... "**);**

int8\_t ret**;**

**while** **((**ret **=** mqtt**.**connect**())** **!=** 0**)**

**{**

**switch** **(**ret**)**

**{**

**case** 1**:** Serial**.**println**(**F**(**"Wrong protocol"**));** **break;**

**case** 2**:** Serial**.**println**(**F**(**"ID rejected"**));** **break;**

**case** 3**:** Serial**.**println**(**F**(**"Server unavailable"**));** **break;**

**case** 4**:** Serial**.**println**(**F**(**"Bad user/pass"**));** **break;**

**case** 5**:** Serial**.**println**(**F**(**"Not authed"**));** **break;**

**case** 6**:** Serial**.**println**(**F**(**"Failed to subscribe"**));** **break;**

**default:** Serial**.**println**(**F**(**"Connection failed"**));** **break;**

**}**

**if** **(**ret **>=** 0**)**

mqtt**.**disconnect**();**

Serial**.**println**(**F**(**"Retrying connection..."**));**

delay**(**10000**);**

**}**

Serial**.**println**(**F**(**"Adafruit IO Connected !"**));**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Ping Adafruit Io Cloud \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void pingAdafruitIo**()** **{**

Serial**.**println**(**"pinging Adafruit Io cloud ... "**);**

**if** **(!** mqtt**.**ping**(**3**))** **{**

**if** **(!** mqtt**.**connected**())**

Serial**.**println**(**"Reconnecting Adafruit Io cloud .... "**);**

connectAdafruitIo**();**

**}**

**}**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Publish Data to Adafruit Io Cloud Feed \*\*\*\*\*\*\*\*\*\*\*/

void publishDataToAdafruitIo**(**int value**)** **{**

**if** **(!** SPS\_Data**.**publish**(**value**))** **{**

Serial**.**println**(**"Failed !!!!!!!!!!!"**);**

**}**

**else** **{**

Serial**.**println**(**"Published !!!!!!!!"**);**

**}**

**}**

const int trigPin **=** 12**;**

const int echoPin **=** 14**;**

#define SOUND\_VELOCITY 0.034

#define CM\_TO\_INCH 0.393701

long duration**;**

float distanceCm**;**

float distanceInch**;**

int readSensorData**(){**

digitalWrite**(**trigPin**,** LOW**);** // Clears the trigPin

delayMicroseconds**(**2**);** // Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite**(**trigPin**,** HIGH**);**

delayMicroseconds**(**10**);**

digitalWrite**(**trigPin**,** LOW**);** // Reads the echoPin, returns the sound wave travel time in microseconds

duration **=** pulseIn**(**echoPin**,** HIGH**);**

distanceCm **=** duration **\*** SOUND\_VELOCITY**/**2**;** // Calculate the distance

Serial**.**print**(**"Distance (cm): "**);**

Serial**.**println**(**distanceCm**);**

**return** distanceCm**;**

**}**

void setup**()** **{**

Serial**.**begin**(**115200**);**

pinMode**(**LED\_BUILTIN**,** OUTPUT**);**

pinMode**(**trigPin**,** OUTPUT**);**

pinMode**(**echoPin**,** INPUT**);**

connectWiFi**();**

connectAdafruitIo**();**

**}**

void loop**(){**

pingAdafruitIo**();**

int value **=** readSensorData**();**

publishDataToAdafruitIo**(**value**);**

delay**(**5000**);**

**}**

***Appendix 2: Codes of the Web application***

***Index.html***

<!DOCTYPE html>

<html lang=**"en"**>

<head>

<meta charset=**"UTF-8"**>

<meta http-equiv=**"X-UA-Compatible"** content=**"IE=edge"**>

<meta name=**"viewport"** content=**"width=device-width, initial-scale=1.0"**>

<title>**Find Parking**</title>

<link href=**"https://fonts.googleapis.com/css2?family=ZCOOL+QingKe+HuangYou&display=swap"** rel=**"stylesheet"**>

<link rel=**"stylesheet"** href=**"index.css"**>

</head>

<body>

<div id=**"headerBox"**>

<img src=**"img/logo.png"** alt=**"logo"** style=**"width:50px;height:40px;padding:0px 10px 0px 0px;"**>

<span>**FIND PARKING**</span>

</div>

<div id=**"content"**>

<div id=**"search"**>

<input list=**"parkinglots"** id=**"location"** placeholder=**"Search . . . "** style=**"height: 30px;"**>

<datalist id=**"parkinglots"**>

<option value=**"ABCParking"**></option>

<option value=**"XYZParking"**>

<option value=**"DEFParkig"**>

</datalist>

<button type=**"button"** class=**"button"** id=**"findparkingbtn"** onclick=**"test()"**>**Find Parking**</button>

</div>

<div id=**"searchResultTiltle"** class=**"SearchResultTitle"**>**Know nearby Parking**</div>

<div id=**"resultbox"**>

<div id=**"searchResult"**></div>

</div>

</div>

<footer>

</footer>

<script src=**"index.js"**></script>

</body>

**</html>**

***Index.css***

html**,**body**{**

**height: 100%;**

**width: 100%;**

**background-color: #7f5a83;**

**background-image: linear-gradient(315deg, #7f5a83 0%, #0d324d 74%);**

**}**

**#headerBox{**

**text-align: center;**

**position: relative;**

**padding: 10px 10px 0px 0px;**

**font-size: 50px;**

**margin: 0;**

/\* height: 50px; \*/

**font-family:** 'ZCOOL QingKe HuangYou'**, cursive;**

**color: rgb(255, 255, 255);**

**}**

**#content{**

**text-align: center;**

**position: relative;**

**margin: 10px;**

**}**

**.**button**{**

**height: 36px;**

**transition-duration: 0.4s;**

**}**

**.**button**:hover** **{**

**background-color: #4CAF50;**

**color: white;**

**}**

**#search{**

**align-items: center;**

**padding: 20px 20px 20px 20px;**

**}**

**.**SearchResultTitle**{**

**margin-right: 20px;**

**margin-left: 20px;**

**margin-bottom: 20px;**

**color: aliceblue;**

**height: 20%;**

**text-align: center;**

**font-size: 20px;**

**font:bold;**

**border-radius: 20px;**

**background-color: #7f5a83;**

**background-image: linear-gradient(315deg, #7f5a83 0%, #295473 74%);**

**}**

**#resultbox{**

**margin-right: 20px;**

**margin-left: 20px;**

**color: rgb(255, 255, 255);**

**border-radius: 20px;**

**background-color: #7f5a83;**

**background-image: linear-gradient(315deg, #7f5a83 0%, #275779 74%);**

**}**

**#searchResult{**

**align-items: center;**

**}**

**.**BookButton**{**

**height: 25px;**

**color: rgb(9, 135, 49);**

**transition-duration: 0.4s;**

**border-radius: 25px;**

**}**

**.**BookButton**:hover{**

**background-color: #4CAF50;**

**color: rgb(248, 244, 244);**

**}**

footer**{**

**position: relative;**

**padding: 0;**

**margin: 0;**

**height: 20px;**

**font-family:** 'ZCOOL QingKe HuangYou'**, cursive;**

**color: rgb(255, 255, 255);**

/\* background-color: #1E1E1E; \*/

**font-size: 10px;**

**}**

**#div1{**

**animation-duration: 2s;**

**animation-name: slidein;**

**animation-iteration-count: 1;**

**color: aliceblue;**

**height: 20%;**

**text-align: center;**

**}**

**#div2{**

**animation-duration: 1.5s;**

**animation-name: slidein;**

**animation-iteration-count: 1;**

**color: aliceblue;**

**height: 20%;**

**text-align: center;**

**}**

**#div3{**

**animation-duration: 2s;**

**animation-name: slidein;**

**animation-iteration-count: 1;**

**color: aliceblue;**

**height: 20%;**

**text-align: center;**

**}**

**#div4{**

**animation-duration: 2.5s;**

**animation-name: slidein;**

**animation-iteration-count: 1;**

**color: aliceblue;**

**height: 20%;**

**text-align: center;**

**}**

**#div5{**

**animation-duration: 3s;**

**animation-name: slidein;**

**animation-iteration-count: 1;**

**color: aliceblue;**

**height: 20%;**

**text-align: center;**

**}**

**@**keyframes slidein **{**

from **{**

**margin-left: 100%;**

**width: 300%;**

**}**

to **{**

**margin-left: 0%;**

**width: 100%;**

**}**

**}**

***Script.js***

**function** test**(){**

**const** value **=** **document.**querySelector**(**'#location'**).**value**;**

**if(**value **==** "ABCParking"**){**

**document.**getElementById**(**"searchResultTiltle"**).**innerHTML **=** '<div class="SearchResultTitle">Spots Available in '**+**value**+**'</div>'

**let** str **=** responseGenerator**(**"Himanshu00"**);**

**document.**getElementById**(**"searchResult"**).**innerHTML **=** str

**}else{**

**document.**getElementById**(**"searchResultTiltle"**).**innerHTML **=** '<div class="SearchResultTitle">Spots Available in '**+**value**+**'</div>'

**document.**getElementById**(**"searchResult"**).**innerHTML **=** "Opps Not Found !! <br> Try other nearby Location"

**}**

**}**

**function** httpGet**(**theUrl**)**

**{**

**var** xmlHttp **=** **new** XMLHttpRequest**();**

xmlHttp**.open(** "GET"**,** theUrl**,** **false** **);** // false for synchronous request

xmlHttp**.**send**(** **null** **);**

**return** xmlHttp**.**responseText**;**

**}**

**function** checkAvailability**(**GetURL**){**

**const** response **=** httpGet**(**GetURL**)**

**let** temp **=** response**.**replaceAll**(**","**,**""**)**

// console.log(temp)

**if** **(parseInt(**response**)>**50**)** **{**

**return** **true**

**}**

**return** **false**

**}**

**function** responseGenerator**(**username**){**

**let** rStr **=** ""

**for** **(let** index **=** 1**;** index **<** 6**;** index**++)** **{**

**let** URL **=** "https://io.adafruit.com/api/v2/"**+**username**+**"/feeds/spp0"**+**index**+**"/data/retain"

// console.log(checkAvailability(URL))

buttonStr **=** '<button type= "button" class ="BookButton"id="BookButton'**+**index**+**'" onclick="BookSpot()">Book</button>'

**if(**checkAvailability**(**URL**)** **==** **true){**

rStr **=** rStr**+**"<div id=div"**+**index**+**"> SPP0"**+**index**+**" STATUS : Available "**+**buttonStr**+**"</div>"

**}**

**}**

// console.log(rStr)

**return** rStr

**}**

**function** BookSpot**(){**

**alert(**"Hi Your booking Confirmed. Please aquire spot in next 20 Minutes."**)**

console**.**log**(**"booked"**)**

**}**