Sushan Kunwar

# Automated Car Parking

Metropolia University of Applied Sciences

Bachelor of Engineering

Degree Programme in Electronics

Bachelor’s Thesis

March 12, 2019

| | |
|---|---|
| Author<br>Title<br>Number of Pages<br>Date | Sushan Kunwar<br>Automated Car Parking<br>26 pages + 2 appendices<br>March 12, 2019 |
| Degree | Bachelor of Engineering |
| Degree Programme | Electronics Engineering |
| Professional Major | - |
| Instructors | Janne Mäntykoski, Senior Lecturer, Metropolia UAS |

The objective of this IOT based project was to design and build a prototype of an automated parking system that would provide a solution to the problems encountered in the parking lot management systems of today.

The main components are RFID technology, RFID labels, RFID reader, barriers. All these hardware are interfaced using the Arduino Mega board.The software handles the transaction, controlling, management, controlling, reporting tasks for parking lots. Space availability can be checked remotely by using internet and a web browser. Parking lot check-ins and check-outs are checked and controlled by RFID readers, labels and barriers and cost for parking is deducted considerably using this technology. Drivers will not have to wait for the identification of their vehicles as it will be done automatically by the tags that are provided to them. This will not only save the his/her time and car's fuel but also ensure security as only the registered tags (users) are allowed to enter.

This project has basically solved the problems of efficient management of parking spaces in extremely busy areas by developing a prototype which is user friendly, secure, fast and providing information of parking space remotely. The end result of the project was satisfactory because all the test cases worked as aimed and intended.

P.S.: Remote access (IOT) doesn't work yet.

| | |
|---|---|
| Keywords | IOT, RFID, Arduino, Reader, IR Module, Tags |

## Contents

## Appendices

## List of figures

List of Tables

## List of Abbreviations

| | |
|---|---|
| IOT | **I**nternet **O**f **T**hings |
| RFID | **R**adio **F**requency **I**dentification |
| AIDC | **A**utomatic **I**dentification and **D**ata **C**enter |
| TFT | **T**hin **F**ilm **T**ransistor |
| ADC | **A**nalog to **D**igital **C**onverter |
| DAC | **D**igital to **A**nalog **C**onverter |
| PWM | **P**ulse **W**idth **M**odulation |
| SPI | **S**erial **P**eripheral **I**nterface |
| IC | **I**ntegrated **C**ircuit |
| IR | **I**nfrared **S**ensor |
| UHF | **U**ltra **H**igh **F**requency |
| ORM | **O**bject **R**elational **M**apping. |
| DBMS | **D**atabase **M**anagement **S**ystem. |
| URL | **U**niform **R**esource **L**ocator |
| API | **A**pplication **P**rogramming **I**nterface |
| REST | **RE**presentational **S**tate **T**ransferring |
| SDK | **S**oftware **D**evelopment **k**it |
| PHP | **H**ypertext **P**re-**P**rocessor |
| PC | **P**ersonal **C**omputer |
| QTY | **Q**uantity |
| COAP | **Co**nstrained **A**pplication **P**rotocol |

## 1. Introduction

This "Automated Car Parking" project shows the concept of an automatic car parking system. Automated parking can be defined as using advanced technologies within an urban mobility strategy to operate, monitor and manage parking efficiently. It is hard to find an unoccupied parking space in today's existing manual system in a large parking lot. The automated parking system proposed helps to find the empty slots even before parking. If there are no slots empty, the gate will not be opened by servo motor and the message "Space not available" will be displayed on the monitor in front of the parking gate. This can be done by using multiple hardware and software. On the hardware side, the core of the project is Arduino Mega Board, which is a user friendly device and this device can be easily interfaced with almost any sensors or modules in today’s market. While on the software side, Arduino core is used to run the Arduino board, MySQL is used to keep and track all records in its database, and PHP is used to retrieve all such data. The system's main components are RFID tags, antenna, reader, barriers, IR sensor, servo motor, and the software mentioned above. Software part is equally important not only to run the Arduino core, but also to display all the parking information on the internet that helps to find the parking space situation even before arriving at the parking spot. So, the purpose and aim of the software is to perform various operational tasks, to maintain the records and retrieve those records from the backend. In other words, software is working as an automation tool.

Radio-frequency identification (RFID) is an automatic identification method that remotely retrieves the data stored on RFID tags or transponders. A brief description of the RFID scenario in this project is that the gate will only be opened if the parking area has empty slots and the RFID identity is valid when the passenger vehicle enters the parking lot. Thus, reducing the manual effort to check if the parking lot has empty slots available. A web interface for the parking system can be used to check the parking space availability even before reaching the designated spot. The problem of space in parking lot should be solved programmatically by keeping the count of total number of space available and the total number of vehicles entering the car parking lot. In this way, the time that is being waste to search for a parking space can be avoided. Slow verification problem is also being solved as the manual identification takes more time than the tag identification. Also logs and records can be maintained for a long time period as there is no use of manual registers and the data is maintained in form of files that can be utilized further according to the system’s requirement. This helps to create an efficient parking method.

Innovative areas such as customer service smartphone apps, mobile payments, and within car navigation systems make automated parking viable as well. The concept of smart parking is the ability to access, collect, analyze, disseminate and act on information about parking use. This information is increasingly provided from smart devices in real time, enabling parking managers and drivers to optimize the use of parking capacity. As a result, the proposed system can help society's economic, social and security-based consideration, preserving the fuel and time can be done in the same way. It helps us to carry out the economic analysis and can help find the feasible project to find the better parking system without causing an economic loss. Future work is also taken in account for different technology in order to develop the system which can be more efficient, reliable, secure and inexpensive.

## 1.1 Parking Lot Problems

It is difficult, if not impossible, to quickly find a vacant space in a multilevel parking lot, especially on weekends or public holidays. Approximately 66 percent of visitors can take more than 10 minutes to find spaces during weekends or public holidays. Stadiums, shopping malls, hospitals etc are crowded at peak times, which is a major problem for customers to find vacant slots at these locations. Insufficient parking spaces result in traffic congestion and driver frustration. If a car is parked in such a way that instead of one it occupies two parking slots, then that is called inappropriate parking. Such inappropriate parking may occur if another driver's rights are not taken care of by a driver. Developing an automated smart car parking system will address this. Improper parking often leads to inappropriate space that causes vehicles to damage each other.

## 1.2 Project Objective

The project main objective is to replace the current parking management system in an efficient way. Here are the list of this project objectives:

- To reduce the human effort and parking time.
- To create a cost-effective system of parking.
- Automatic billing system.
- Remote access to the information of availability of parking space.
- Making parking safer, secure and authentic.

## 1.3 Features and Motivation to use Smart parking

To abandon and move towards this proposed parking management system, one must be convinced and satisfied to use this latest technology. Otherwise there is no reason and motivation to switch from old system to the new one. Some of this project features and motivation to use smart parking is mentioned below:

- Optimized parking: Vehicle owner will find the available parking spot, saving time, fuel and effort. The parking area proficiently fills which business and corporate substances can utilize the improved space appropriately.
- Reduced pollution: Parking search burns an unaccountable fuel in a day. Therefore this optimal parking solution will significantly save fuel and reduce driving time, thus reducing the daily emissions of vehicles.
- Reduced traffic: Traffic stream will increase as searching for an open parking spot requires a drive around.
- New Revenue Streams: With this proposed technology, plenty of new revenue streams are possible. Parking spot proprietor, for instance, can permit layered installment choices relying upon the area of the parking spot. So as to energize rehash clients, reward program can likewise be incorporated into existing models.
- Increased security: Parking area working staffs have data in real time which can help to prevent parking violations and any suspicious activities. Lower spot street search traffic can also help in reducing accidents which is caused by distraction from parking searches.
- Integrated Payments: Returning clients can substitute their phone for day to day hand cash payments with account invoicing and application payments. This could likewise permit valuable user feedback and customer loyalty programs.
- Improved User Experience: In a unified action, the entire client experience is integrated in this automated parking solution. Payment by the driver, location query, identification of the spot and notifications for time, all become the part of the entry procedure of the parking spot.
- Real Time Data and Trend Insight: Information can be generated over time by automated parking solution that uncovers correlations between user and lot

trends. These trends can be invaluable to many owners as to how drivers can be adjusted and improved.

- Decreased Management Costs: Such automation which involves less manual activity saves the cost of labor and exhaustion of resources.

### 1.4 Scope

This developed technology can be used in any urban areas where the car parking is most. Some of the heavy traffic places where this project can be installed and used are mentioned here:

- Shopping Malls
- Hospitals
- Airports
- Cinema Hall

## 2. RFID Based Management System

Different methods of developing autonomous or smart parking systems are prevalent. Since the project is based on RFID, the literature used to initiate this project is based on an article published in the Indian Journal of Science and Technology entitled “A Prototype for IOT based Car Parking Management System for Smart Cities” [1]. Obviously one can study that article and may find many differences compared with the way of approaching the matter in this project. For example, the literature cites an idea of using Raspberry pi, dc motor instead of Arduino, servo motor etc. which were used for this project. The article also has not mentioned which language and database has been used. Although some of the hardware and software mentioned in this article are not the same which are used for this project, the main theme of the article and the project remains the same. The article discusses the need for such a project and how today's existing parking system would be made easier. Several people are required in the traditional system to monitor a parking area to evaluate the number of free slots and match it with the capacity of a parking area. If these old fashioned way of parking is replaced by an automatic indicator, the number of individuals employed would be reduced significantly. The article also discusses about the car parking with big issue. Improper parking and damaging other cars while parking the car is just another problem that makes unhappy and frustrate the damaged car park owners. The car parking system communicates to the server with every slot. This automated system guides the client to park the car perfectly and keep all the record in database, so it is possible to find out who has damaged the car. This project's brain is the Arduino board. This Arduino board is controlled by a program which is written on it. The program assessed the number of switches presses where each switch corresponds to a slot and subtracts it from the capacity or the total number of slots present. LCD display is used to display the number of free slots that are empty. The article also has insighted the idea of using a servo motor. The servo motor is basically a motor where it is possible to control and vary the angular velocity. The project was inspired by an RFID-based attendance system that uses an RFID tag together with the reader to input the employee student details to track their attendance. When switching the RFID to the reader, the tagged information is compared to the microcontroller data information which is interfaced with the reader, to identify the user[2]. To display the user's name, an LCD display is interfaced with the microcontroller. In addition, the user's overall attendance is displayed with a status button. Additionally, for the RFID-based application, a different approach is implemented that enhances proper parking management today. Unit consists of user IDs that provide valid parking system authentication.

The idea of using RFID was also taken from a smart library management system (LMS)[3] which is a pilot project built for the universities and schools libraries. A Graphical User Interface (GUI) has been designed to support all the library features and client needs. Various types of shelf antennas have been designed and manufactured with the aim of 100% tag readability for library users in a low-SAR environment. Prototype smart cabinets were assembled, built and tested using various types of near-field shelf antennas to ensure confined coverage near the shelf, thus avoiding unwanted identification of books residing on nearby shelves. In the presence of human phantom models located close to the cabinet, SAR simulations were performed. The shelf antennas have been optimized near the cabinet so it helps in minimizing the electronmagnetic pollution and maximize the readability of tags.

Low frequency RFID works at 125 KHz frequency on the radio wave principle. The RFID tag contains a coil and sends a code of identity to a device for further processing when it is influenced by the magnetic field. For a specific client, the RFID tag is used as an identification. If the client's identity (serial tag number) matches the one already stored in this system, which provides immediate access to the client. There are also many additional features to this secured access system based on RFID. For instance, within the system, a new client can register her/himself. A registered user may also withdraw their system entry. Such features can be accessed by pressing the microcontroller-connected tactile switch. The client is prompted at the beginning to scan his/her identification(RFID Tag). The reader module identifies the tag's serial code and is sent for checking to AT89C51. If the microcontroller matches the ID, the client is granted to have the access of parking space. Despite what might be expected, if the tag isn't distinguished, a message ('Not Valid') is shown on LCD display.

## 3. Design of the System

The design of the automated car parking system is divided into two parts: Hardware (electronic components) and Software. RFID tag, readers, barriers, Arduino mega board, Arduino Ethernet shield, IR Module, Servo Motor, LCD screen are the main electronic components used as hardware parts. On the other side, software like HTML, CSS, JavaScript, MySQL, PHP, Apache are used to have one full functioning project.

### 3.1 Hardware Required

#### 3.1.1 RFID and It’s Working

RFID stands for Radio Frequency Identification. RFID belongs to the Automatic Identification and Data Capture (AIDC) family which is a quick and reliable way of identifying objects. There are two fundamental parts: the RFID Reade, transmitting and receiving the signal, and the Transponder connected to the object. The RFID tag is a device which uses waves of radio to identify and track an objects like person, animal or products. A few meters away, beyond the sight line of the reader, some tags can be read. RFID tags comes in a wide range of sizes and shapes which can be passive or active. Communication is wireless between the RFID reader and tags and generally requires no observable pathway between devices. An RFID reader can read nearly anything aside from conductive materials, for example, water and metal, but even these can be overcome with modifications and positioning.

A product tag is attached to a product in its least complex form, made of a microchip with a small antenna. Electromagnetic waves are exposed by an associated tag reader. The tag of the antenna receives the radio waves and also the tag itself attracts power from the reader generated field, powering the chip, then modulating the reader signal, causation it back wherever it's born-again to digital information. At the low end of the spectrum, the electromagnetic waves are harmless and no more dangerous than a radio car. The basic working of RFID system with antenna, tags and database is shown by figure 1.

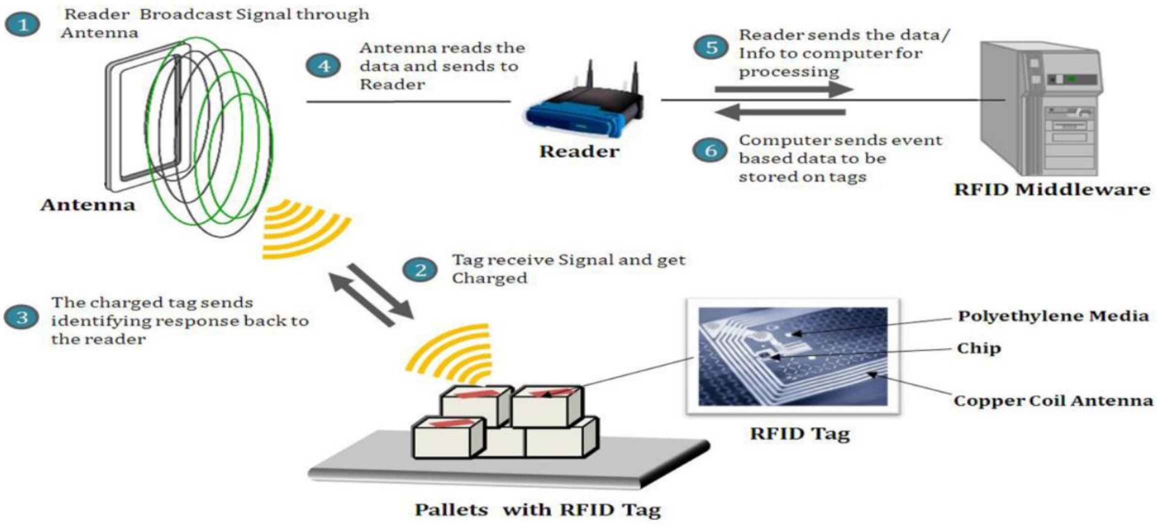

1 Reader Broadcast Signal through Antenna
Antenna
4 Antenna reads the data and sends to Reader
Reader
5 Reader sends the data/ Info to computer for processing
6 Computer sends event based data to be stored on tags
RFID Middleware
2 Tag receive Signal and get Charged
3 The charged tag sends identifying response back to the reader
Polyethylene Media
Chip
Copper Coil Antenna
RFID Tag
Pallets with RFID Tag

*Figure 1: Working of a RFID with database system [4]*

RFID tags are available in a large type of sorts as well as the preceding passive type, multi-frequency, powered, and tag-talks-first. There may also be different types of RFID antennas. In addition, not all RFID systems use low frequency EM waves. Read-only tags and tags are available for read-write tags. Up to 2K tags contain product data and tags contains only one product ID. Tags can also be used for more than product IDs; they can be used in safety devices,environmental monitoring, and mechanisms for product integrity.

i. Types of Tag

Most RFID tags contain at least two parts. One is an integrated circuit where information is stored and processed, a RF signals are modulated and de-modulated. The second part is an antenna where the signal is received and transmitted. Usually two types of RFID tags are found: one is active RFID tags that contain a battery and can therefore autonomously transmit its signal, and the other one is passive tag, a battery free RFID tags which requires an outer source to start the transmission of signal.

- Passive Tag

  Passive tags are commonly littler, lighter, and less expensive than dynamic labels that can be connected to objects in harsh environments are without upkeep, and will keep going for quite a long time. Transponders are activated just within a reader's response range. A low-power radio wave field is emitted by the RFID reader which powers the tag to convey any chip information.

- Active Tag

  Active tags disagree therein they incorporate their own power supply, whereas the tag may be a transmitter instead of a reflector of frequency signals that enables a wider vary of options like read/write and programmable capabilities.

ii. RFID Frequencies

Data carriers between the tags and reader are radio waves. The approach to RFID communication is usually adopted to allocate frequencies depending on the application. The used frequencies cover in the extended range.

These specified bands are:

- Very long wave 9 – 135 kHz
- Short wave 13.56MHz
- UHF 400-1200 MHz
- Microwave 2.45 and 5.8 GHz

Government agencies direct the allocation of frequencies, which require supervision when RFID applications is taken account in different countries. Standardization efforts should avoid these issues. Different applications can work their best at different frequencies; thus, before choosing a specific type of RFID system, it's vital to know the necessities. Security access, quality chase and animal identification are the foremost common uses of low-frequency systems. They often have short reading ranges and low system prices. For applications like railroad car chase and automatic toll assortment, high-frequency systems square measure used. They provide high speed and long ranges of reading. Usually, price depends on the performance which means higher prices for higher performance. The interrogator's power level and therefore the power to reply at intervals the tag can verify the vary of reading that may be achieved in a RFID system. There square measure legislative constraints on power levels, just like the restrictions on carrier frequencies. Environmental conditions also can influence the communication vary, particularly at higher frequencies.

### iii. RFID Module

MF RC522 is a extremely integrated, contactless communication module at 13.56 MHz. For various types of contactless communication methods and protocols at 13.56 MHz. This transmission module uses an impressive modulation and demodulation abstract which is absolutely integrated. Radio-frequency identification (RFID) uses an electromagnetic fields to spot and track the tags connected to the objects. The tags contain the data that has been kept electronically. RFID reader with Tag is shown in figure 4. The RFID reader and tag specifications are listed below:

*Table 1: RFID reader and tag specifications*

| | |
|---|---|
| Operation Current | 13~26mA/DC3.3V |
| Sleep Current | 80µA |
| Peak Current | <30mA |
| Operation Frequency | 13.56MHz Read Range 0~60mm Interface SPI |
| Data Transition Rate | Up to 10Mbits/s |

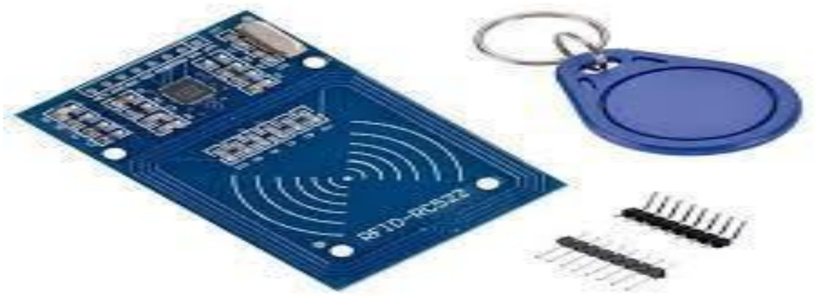


*Figure 2: RFID Reader and Writer with Tag [5]*

### 3.1.2 Arduino Mega Board

Arduino is an open-source, hardware and software based prototyping platform. Arduino boards can perform various automation with other electronic devices such as read light inputs on a sensor, turn on LED, operate with different types of motor, and go online by using network shield. The microcontroller on the board follows the set of instruction which is written on a programming language by which your board can do many awesome things. It's like any project's brain. Arduino can communicate with plenty other electronic devices such as pc, another Arduino, or alternative microcontrollers.

It has 54 digital input/output pins (including 14 as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset key. It has everything you need to support the microcontroller; simply connect it to a USB cable computer or power it to get started with an AC-to-DC adapter or battery. The Mega is compatible with most Arduino shields. Arduino board functions are controlled using Arduino IDE (referred to as uploading software) to send a set of instructions to the microcontroller on board. Arduino core is used for controlling this board which is explained later on software part. Arduino Mega with its pin function are shown in figure 3. The Arduino specifications are listed in table below:

*Table 2: Arduino Specification*

| | |
|---|---|
| Microcontroller | AtMega2560 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | (7-12)V |
| Input Voltage (limit) | (6-20)V |
| Digital I/O Pins: | 54 (of which 14 provide PWM output) |
| Analog I/O Pins: | 16 |
| DC Current per I/O Pin: | 40mA |
| DC Current for 3.3V Pin: | 50mA |
| Flash Memory | 256 KB of which 8 KB used by boot loader |
| SRAM | 8KB |
| EEPROM | 4KB |
| Clock Speed | 16MHz |

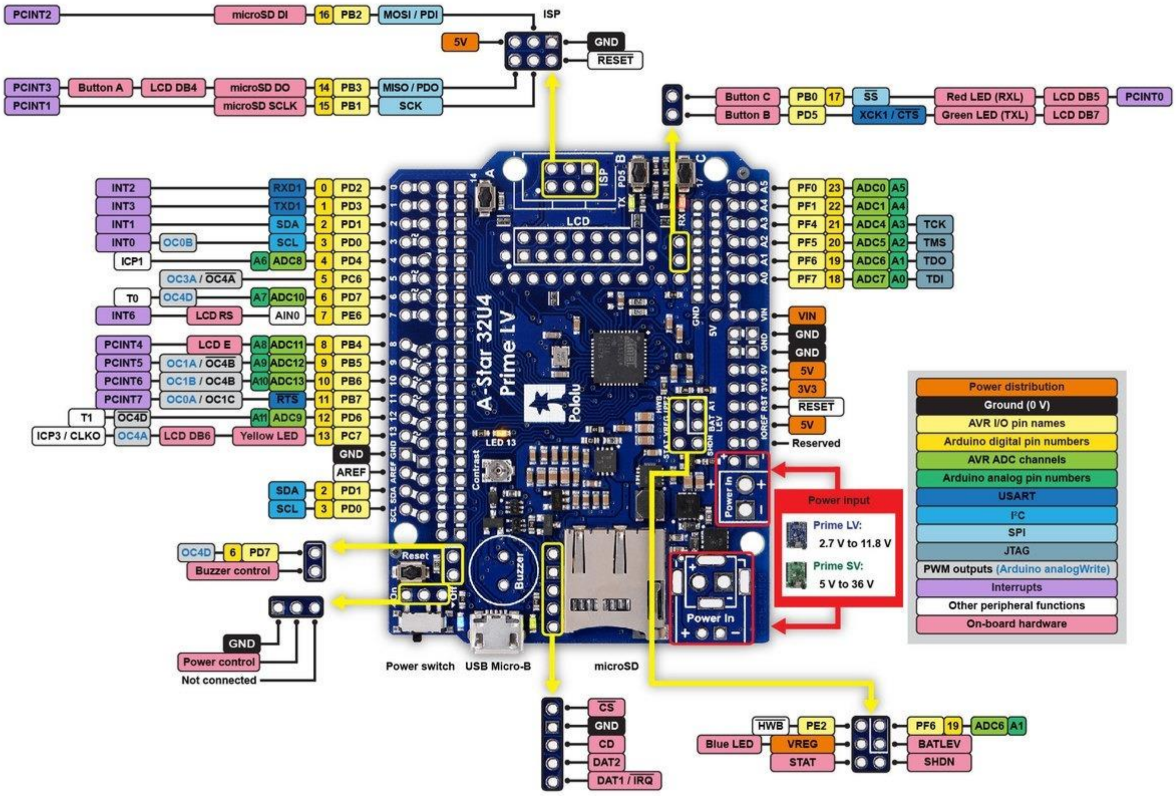


Figure 3: Arduino Mega Board Pinout [6]

### 3.1.3 Arduino Ethernet Shield

The Arduino Ethernet Shield R3 which is assembled with Arduino Mega Board can provide an internet connection to an Arduino board. This shield is based on the ethernet chip datasheet of Wiznet W5100. The Wiznet W5100 provides a TCP and UDP capable network (IP) stack. Four connections can be made simultaneously to this socket. Figure 2 shows the connection of Arduino Ethernet shield to the internet via ethernet cable.

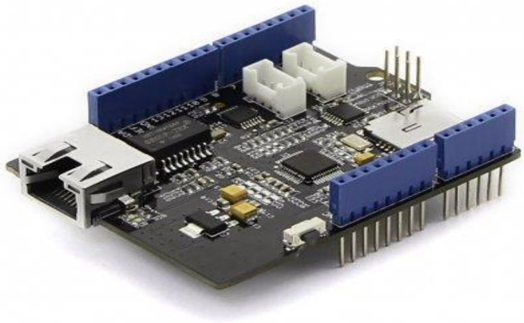


Figure 4: Arduino Ethernet Shield [7]

### 3.1.4 IR Sensor

An infrared sensor is an electronic detector or sensor which assess infrared ray (IR) light divergent from objects in its field view. In this project, it observes the presence of vehicles in the parking space and sends information accordingly. ADC is used here to do the maths which is explained below:

- 1023==5V (occupied)
- 0==0V (not occupied)

Arduino ADC has a 10 bit converter which means 1024 (0-1023) as a decimal number which is then mapped with power supply of (0-5)V. 5V means occupied because IR sensor will sense the vechile in the parking space and 0V means not occupied because IR sensor will not sense anything. IR module is shown in figure 5.

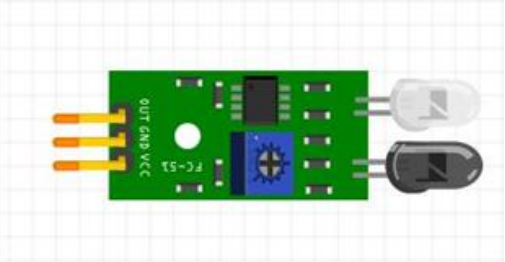


*Figure 5: IR Module* [8]

### 3.1.5 Servo Motor

Servo motor is a mixture of a DC motor, gear set, potentiometer, and unit control circuit. The engine is attached to the control wheel by gears. The resistance of potentionmeter's changes with the rotating motor. Therefore, the control circuit will exactly tune the movement and direction of the motor. Servos are managed from the control wire by forwarding an electrical pulse of fluctuating width, or pulse width modulation (PWM). Type of pulse width are: maximum pulse, a minimum pulse, and a repetition rate.

Servo is used instead of a DC because servo motors are easy to be controlled and are more accurate than standard DC motors. They have three wires, each one for ground, control and power. There is consistent use of capacity to servo engines, with the servo control circuit controlling the attract to drive the engine. Servo engines are intended for progressively explicit assignments where there is a need to precisely characterize position, for example, controlling the boat rudder or moving a mechanical arm or robot leg inside a specific range.

Here in this project, servo motor is required to open the gate for incoming vechiles to park, which means gate can be opened by shifting around 90º-180º.

Figure 6: Servo Motor [9]

### 3.1.6 LCD Screen

LCD (Liquid crystal display) is used in our mobile phones and other various Nano technology. Same as light emitting diode (LED) and gas plasma, LCDs let displays to be much slimmer than the technology which used for cathode ray tube (CRT) which often are very large in size. LCDs expend significantly less power than LED and gas display since they work as opposed to transmitting it on the guideline of blocking light. Connection of LCD screen with microcontroller is done by the pinout shown figure 7.

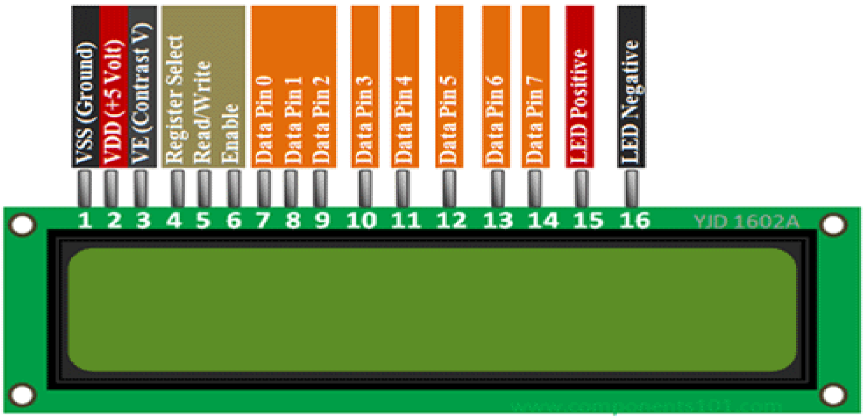


Figure 7: LCD Screen Pinout [10]

## 3.2 Software Required

We discussed the software and modules used during the development of the system in this chapter. A programming language and interface is needed to achieve the desired results in order to devise a logic that will enable the Arduino Mega board to understand its requirements. Several Arduino libraries have been used to make it compatible with hardware and software. To develop a full working web page which can be accessed via internet, necessary languages and software are used.

### 3.2.1 Arduino Core

Any microcontroller viewed around requires a development environment where it can be programmed prior to actual deployment. Fortunately, Arduino maker provides a free downloadable open source SDK. Arduino 1.8.8 is the current version of this IDE. Different libraries are used for the faster development of project. Libraries like Adafruit GFX library for LCD display and Adafruit Font library were used for displaying the text. For many device interface, own libraries are created as the hardware board was new in the market.

### 3.2.2 HTML, CSS and JavaScript

Hyper-text markup language (html), Cascading Style Sheets (CSS) and JavaScript (JS) is required to develop the frontend of the webpage. While html and CSS are declarative languages that tell the browser how some information can be displayed, JavaScript (JS) is a programming language that describes a process. JS is used for website interactions such as opening a pop-up window, validating a login, or getting information to be displayed on the browser from another site. For example if someone wants to know html, CSS and JS in terms of building a house rather than building a web page, then the function of HTML, CSS and JavaScript will be:

i. HTML

An HTML file contains the page's own structure. It is sort of like the building's structure.

ii. CSS

Styling of the page is done through CSS which allows to color, position and many more to change the outlook of the webpage itself. It is like the design of the building.

iii. JavaScript

The dynamic and interactive elements on the page are determined by a JavaScript file. It focuses on the interactive part of the webpage like clicking, hovering or typing within certain elements of the webpage.

When a user request for a webpage, s/he has to go through a certain process which is not visible to the user. The mechanism of loading a webpage using HTML, CSS, JS, databases and PHP is shown in figure 8.

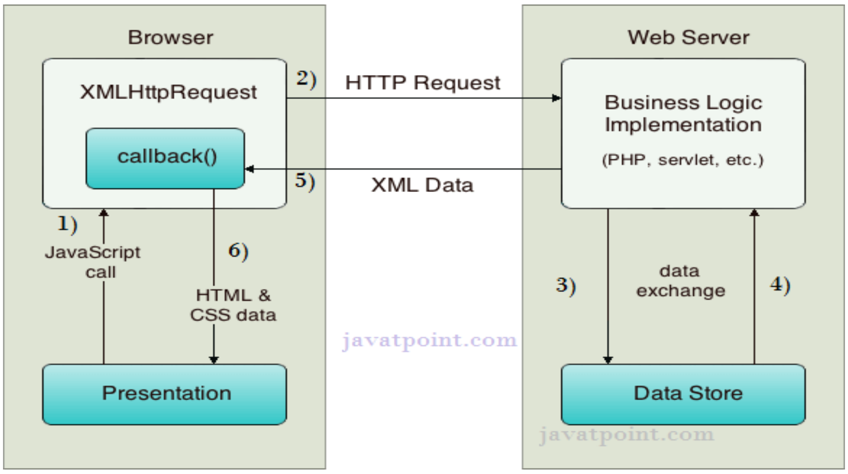


*Figure 8: Working of HTML, CSS, JS with PHP and MYSQL [11]*

### 3.2.3 MySQL Server

MySQL is a Structured Query Language (SQL) based on Relational Database Management system (RDBMS). This runs virtually in all platforms, such as Windows, Linux and UNIX. While MySQL is used in a wide range of different types of applications, it is often found to be used in publishing online and web applications. MySQL is now the RDBMS behind many of the world's top websites and countless web-based applications facing corporate and consumer, including Facebook, Twitter and YouTube. MySQL is a software product and it’s most important character and function is to store and retrieve information as requested by different software applications, regardless of whether those are on a similar PC or those are running on another PC through a system including the web (remote). Many programming languages access this database, but PHP is used for this project.

### 3.2.4 PHP and Laravel Framework

PHP (recursive abbreviation for PHP which is Hypertext Preprocessor) is a broadly utilized universally useful open source scripting language that can be inserted in HTML and is especially suitable for web development. It is programming languages that empowers web developers to make dynamic webpage that associates with databases. PHP is utilized basically to create online programming applications based on web. Laravel is a framework of PHP which is free and open-source. Taylor Otwell is the creator of this framework. Laravel's have many features and some of them are: a modular packaging system with a dedicated dependency manager, number of approaches to access relational databases, utilities that helps in deploying and maintaining applications. Therefore, in this project, the Laravel framework provides clean URI for simple REST calls and binds together with the storage database.

### 3.2.5 Domain Name and Site hosting

To set a website and published it publicly, domain name is required. A domain name refers to the translation of the physical location of the internet to named address and vice versa. To host the content of the website can be referred as website hosting which can be done on a cloud or premise environment. In this project, it is hosted in a premise environment. To set up this environment, one needs (Linux, Apache, MySQL, PHP) simply LAMP or WAMP (Windows, Apache, MySQL, PHP) server environment and all the content of the site is stored on this environment. This environment consists of one operating system and requires services to host, such as web server which is an apache server. Finally this locally set environment is pointed to a domain name which later on can be access publicly. For this project, domain name was required and this is the domain name “http://rfidparkinglatest.000webhostapp.com/” which was hosted from “https://www.000webhost.com/” [12], a free hosting site.

## 4. Mechanism with Flowchart

### 4.1 Block Diagram

To get started with RFID based automatic car parking system, the owner of the vehicle must first register his/her RFID tag with the parking owner/company and receive it. RFID tag which is then placed near the RFID reader, which is installed near the parking lot entrance gate, when the car has to be parked. This system consists of different modules and they are interfaced with Arduino Mega microcontroller which is connected to the internet. All of the operational task will be recorded on a database and is easily accessed from the internet. This allows vehicle owner to check the availability of parking system remotely. The general operational block diagram of this project is shown in figure 9.

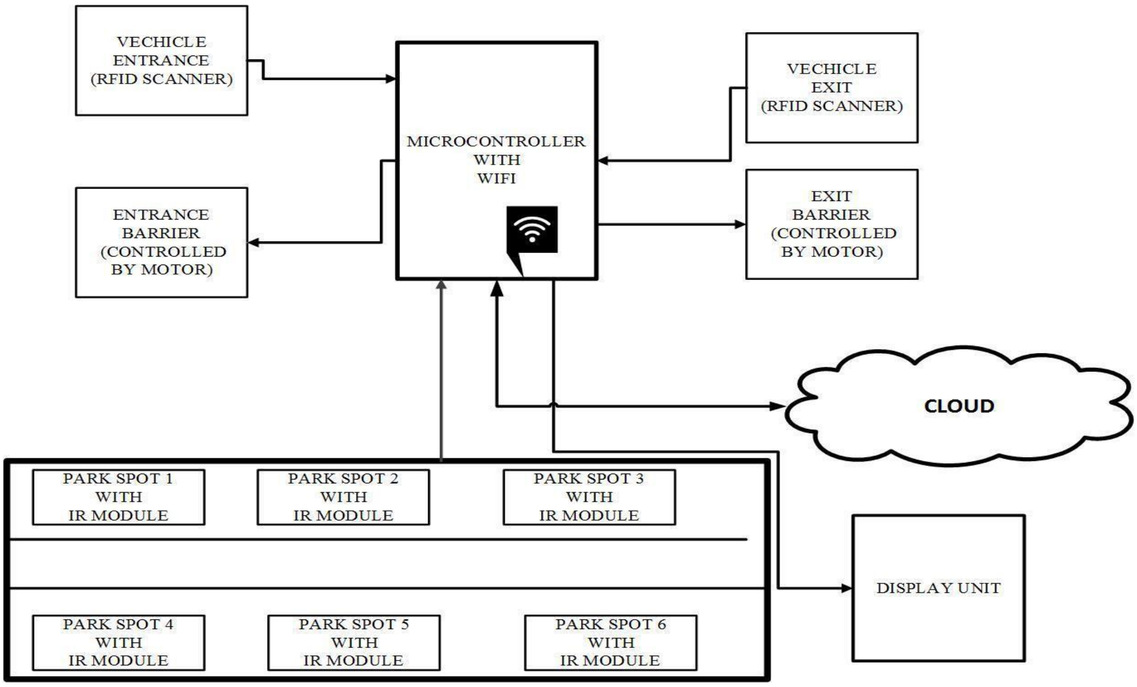

VECHICLE ENTRANCE (RFID SCANNER)
ENTRANCE BARRIER (CONTROLLED BY MOTOR)
MICROCONTROLLER WITH WIFI
VECHICLE EXIT (RFID SCANNER)
EXIT BARRIER (CONTROLLED BY MOTOR)
CLOUD
PARK SPOT 1 WITH IR MODULE
PARK SPOT 2 WITH IR MODULE
PARK SPOT 3 WITH IR MODULE
PARK SPOT 4 WITH IR MODULE
PARK SPOT 5 WITH IR MODULE
PARK SPOT 6 WITH IR MODULE
DISPLAY UNIT

*Figure 9: System Block Diagram*

Fig 9 shows the block diagram of this project. To get started with automatic car parking system on RFID based, the vehicle owner vehicles first must be registered with the parking space owner and get the RFID tag. When then s/he has to park vehicles, then the RFID tag is placed near the RFID reader, which is installed near the entry gate of the parking lot. Now, when RFID tag is read by the reader, the system  goes through a series of process to provide access inside the parking spot. The entrance gate is opened to allow the car inside the parking area once the vehicle has access to the spot. Simultaneously, the parking counts increases one by one and starts the time stamp record. Before leaving the parking spot, the reader reads the RFID tag again and deducts the balance according to the time recorded. Similarly, at the exit gate, the door is opened and the parking counter decreases. The system also provides the facility for each RFID tag to recharge the amount. There is no involvement in manual processing. In addition, the system provides security.

Then a fixed IP address or a domain name is needed, so that it can be assigned it to the reader. The database name “PARKINGDATA” will store the data of vehicles. In the database there are tables which are used to store the data of vehicles. For example user, charge, park are the databases which is shown below in the table. The reports and database is being created by the software itself. The reader is connected to the computer using a cable to provide a communication between the reader and software and there is also a connection for the automation of barrier.

## 4.2 Tables in Databases

MySQL server requires table to store the data. An essential feature is the use of a database in such applications. MySQL stores all the data in tables to provide proper content for a given website. Regardless of the prefix, there are rows and columns in each MySQL database table. The columns specify the data type, whereas the rows themselves contain the actual data. The table’s attributes are shown in table 1 with its table name.

*Table 3: MySQL Database Table*

| **User** | **charge** | **park** |
|---|---|---|
| id | id | id |
| name | user_id | state |
| email | balance | timestamps |
| rfid_tag_no | entry_time | |
| amount | exit_time | |
| password | | |

## 4.3 REST API’s Route

RESTful Web Services are essentially Web Services based on REST Architecture. Everything is a resource in REST architecture. RESTful web services are lightweight, highly scalable and maintainable, and are widely used for web based applications to create APIs. So, here's the API needed to invoke the databases and display the desired result on the webpage.

1. For checking the user validity
   GET http://rfidparkinglatest.000webhostapp.com/isvalid/{rfid_tag}
2. For checking the user balance
   GET http://rfidparkinglatest.000webhostapp.com/balance/{rfid_tag}
3. For editing the used parking available space
   GET http://rfidparkinglatest.000webhostapp.com/state/{used_parkspacs}/edit
4. For creating an entry time for user associated with user rfid_tag
   GET http://rfidparkinglatest.000webhostapp.com/createcharge/{rfid_tag}
5. For sending an exit time for user associated with user rfid_tag
   GET http://rfidparkinglatest.000webhostapp.com/deductcharge/{rfid_tag}

## 4.4 Data flow diagram

Data flow diagrams (DFD) provide a graphical representation of how information moves between processes in a system. Every system needs a DFD before developing system and this helps in presenting the idea of how information flows with in a system. Here, figure 10 shows the graphical representation of how automated car parking data flows within the system.

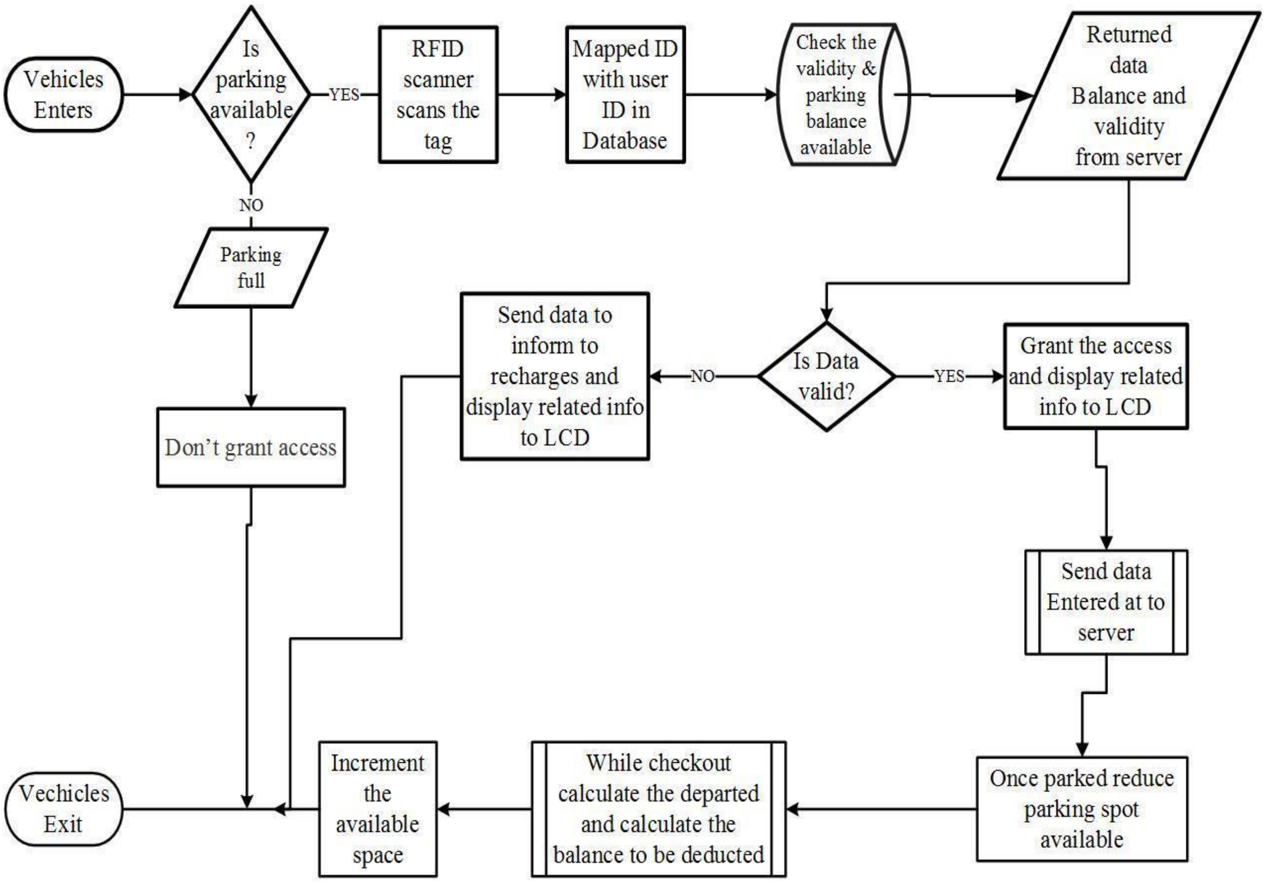

Vehicles Enters
Is parking available ?
YES
NO
RFID scanner scans the tag
Mapped ID with user ID in Database
Check the validity & parking balance available
Returned data Balance and validity from server
Parking full
Don't grant access
Is Data valid?
NO
YES
Send data to inform to recharges and display related info to LCD
Grant the access and display related info to LCD
Send data Entered at to server
Once parked reduce parking spot available
While checkout calculate the departed and calculate the balance to be deducted
Increment the available space
Vechicles Exit

*Figure 10: Data Flow Diagram*

## 4.5 REST API Call Data Flow Diagram

Data flow diagram of how every database is invoked by calling REST API is shown by figure 11.

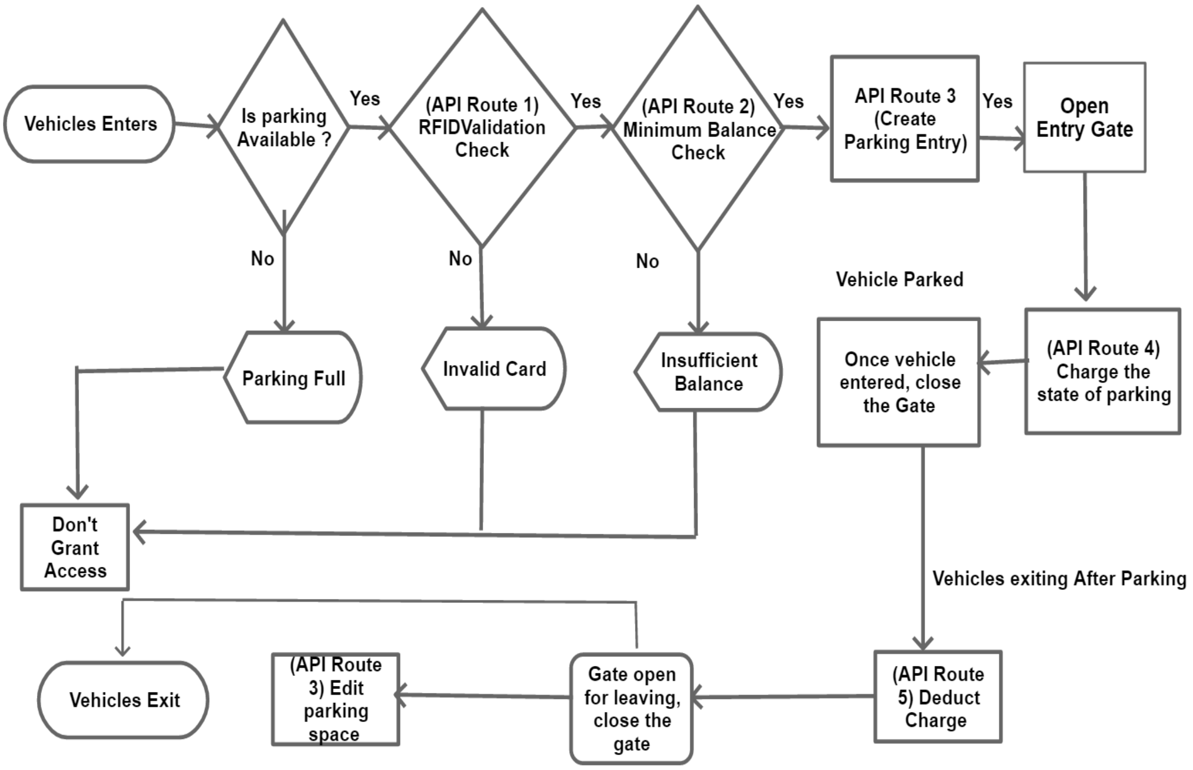

Vehicles Enters
Is parking Available ?
Yes
(API Route 1) RFIDValidation Check
Yes
(API Route 2) Minimum Balance Check
Yes
API Route 3 (Create Parking Entry)
Yes
Open Entry Gate
No
No
No
Parking Full
Invalid Card
Insufficient Balance
Vehicle Parked
Once vehicle entered, close the Gate
(API Route 4) Charge the state of parking
Don't Grant Access
Vehicles exiting After Parking
Vehicles Exit
(API Route 3) Edit parking space
Gate open for leaving, close the gate
(API Route 5) Deduct Charge

*Figure 11: REST API Call Data Flow Diagram*

Information of all vehicles are stored in the database within the respective tags which is created for each vehicle owner. Their tag id's are provided to all the users who has registered into the system, so all the information can be accessed by the system.

As shown in the data flow diagram, when the vehicle checks in, the reader reads the data of the tag. If there is no tag on the vehicle than the barrier will remain close. Now the reader will read all the information of the tag and transfer that information to the software. And all of the respective API's will be called as shown in the figure 11. Now that software compare the information of tag with database and if the id of tag matches, which means the tag is valid and have a limited amount of parking balance (here Nrs 150 which is about 1.5€) then the barrier gate will open and if the id doesn't matches or if there is not sufficient amount of balance then the barrier will not open. Once all of the condition are matched, then the user has access to one of the available parking spot. After the time of parking is finished and when the vehicle goes out (Check out) from the parking lot, the identification information of vehicle is searched in the database. If it is an authorized vehicle and does not have unauthorized access then only the vehicle will be allowed for a checkout otherwise the gate will not open and once authorized, the deduct API will be called and the charge from the user account is deducted accordingly.

# 5. Cost And Time Analysis

## 5.1 Cost Analysis

All of the required components cost are listed below in the table 4.

*Table 4: Cost Analysis*

| S.no. | Components Name | Qty | Rate(€) | Total(€) |
|---|---|---|---|---|
| 1 | Arduino Board | 1 | 25 | 25 |
| 2 | RC 255 RFID reader | 2 | 8 | 16 |
| 3 | LCD Display | 1 | 12 | 12 |
| 4 | IR Module | 4 | 2 | 8 |
| 5 | Servo motors | 2 | 5 | 10 |
| 6 | Arduino Ethernet Shield | 1 | 40 | 40 |
| 7 | Miscellaneous Electronic items | - | - | 20 |
|  | Grand Total |  |  | 131 |

## 5.2 Time Analysis

In order to complete this project, all of the work has been segmented into different parts. The time took to follow each process has been shown in the figure 12.

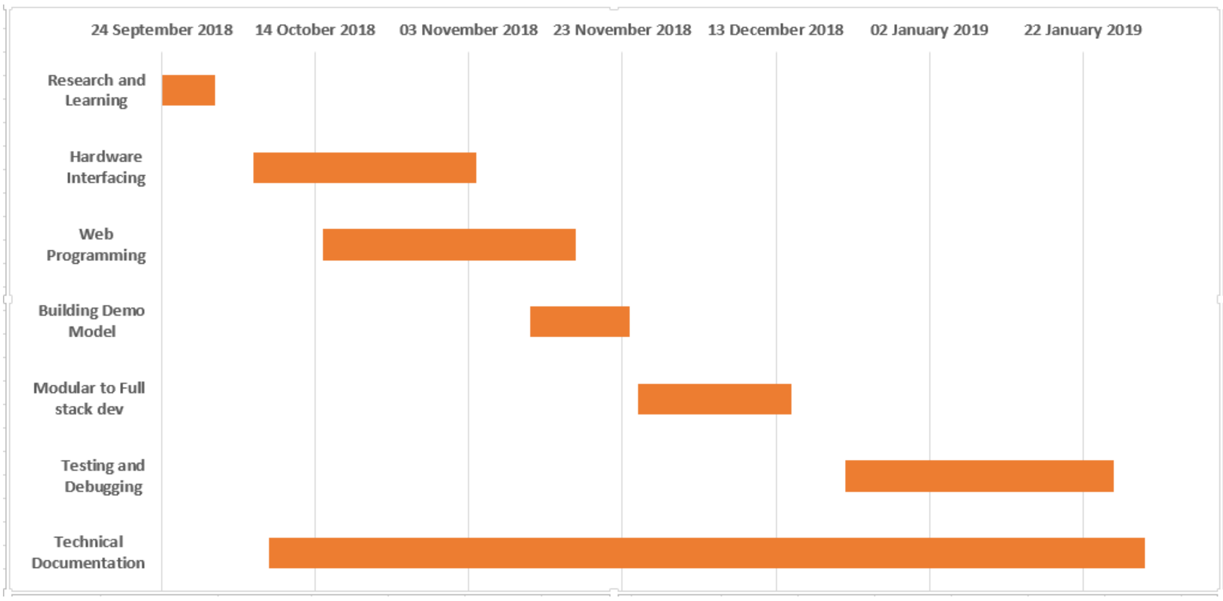


*Figure 12: Project Time Analysis*

## 6. Result

The project platform contains 6 parking space, entry gate, exit gate and LCD display to display the information. 150 Nrs per hour is the cost for parking the car and the same amount should be deducted per hour basis. This information is in the web browser. LCD always displays the availability of free parking space which is 6 in this case. Now to verify how well project went, 4 test cases were performed. Each test case is explained briefly.

### 6.1 Test Case 1: Invalid Tag

First test was done for the unregistered tag. LCD displayed 6 parking space available. In this test case, as car with invalid tag approached towards the entry gate, system did not allow car to enter the parking area and gate remained closed. LCD displayed “Invalid tag” and parking space available is still 6 which means invalid tag has been tested and has performed successfully.

### 6.2 Test Case 2: Valid Tag but low balance

Second test was done for the registered tag but with low balance. LCD displayed 6 parking space available. In this test case, as car approached towards the entry gate, LCD displayed “low balance” and free space for parking is still 6. By this result, it has been confirmed that the low balance tag has worked successfully.

### 6.3 Test Case 3: Valid Tag

Third test was done for the registered tag which has a sufficient balance to park his/her car in this parking lot. LCD displayed 6 parking space available. As car approached towards the entry gate, the gate opened. Car was allowed to go to the parking space. Now, as car entered the parking lot, LCD displayed 5 parking space available as one is occupied by the newly parked car.

Now after sometime the parked car has to leave, LCD still displayed 5 parking space available. As the car approached towards the exit gate, the gate opened. Owner was allowed to leave the parking space and as s/he left, LCD displayed 6 parking space available as one occupied space has been free. Now to check the balance in web browser, browser is reloaded and it shows that Nrs 150 has been deducted from the user balance. By this result, it has been confirmed that the valid tag has worked successfully.

### 6.4 Test Case 4: Remote availabilty

This project aims to find out the situation of parking availabilty remotely from a web browser. Unfortunately the test could not be made because of the IR module, which function was to detect the space availabilty but could not be interfaced with Arduino Mega board. Issue occured because either the IR module or wiring of IR module was faulty.

## 7. Limitation and future enhancement

### 7.1 Limitation

Even though remote availability of free parking space could not be done, this project still provides various automated facilities. Excluding this one feature, there are still various short comings and limitations of this projects. Some of the limitations of this project are:

- The RFID reader used in this project is not powerful. Hence reading of RFID card may be difficult at times.
- Less security feature in the web app.

### 7.2 Future Enhancement

Every project has limitation on its own. This project also has its limitations, as mentioned above. However, different modifications can be done in this project and the project can be enhanced in the future according to need. The web interface can be enhanced with proper security using API authentication. Application now uses REST model for communication which uses more bandwidth. To improve this another protocol COAP which uses thin http protocol can be used to limit the usage of bandwidth. For the vehicle tracking active RFID tag can be used for faster detection. The RFID reader having far field coverage can be used for farther readability of the tag. For the enhanced customer experience, remote booking feature of parking area can be added. Number plate detection is also possible by using image processing which can be even more secure and authentic system.

## 8. Conclusion

This project has basically addressed to solve the problems of efficient management of parking spaces in extremely busy areas. The project model needs to be analyzed while developing a model of life size. The mechanical model has been designed. The project was particularly divided into electronic, Arduino programming and web programming part. In the electronic part, all the required circuits were developed successfully. The programming part which was challenging took long time but it has been successfully integrated with all the hardware that each member was assigned with the arduino board. Later each individual program was combined and a single program was created. Similar process is used to develop the web app. Also, the software alongside the control circuit have been implemented successfully. The project was performed with intensive research and learning from different resources and tools.

The test cases clearly demonstrates the working of the automated car parking. If the customer is registered into the system then it makes their tag valid. Validity of a vehicle was determined by scanning the RFID card. Access to the parking area was provided if the tag was valid. Once validity of the vehicle was confirmed, a billing system is also provided according to the time duration of the vehicles parked in the parking lot. The balance can be recharged by logging into the user account in the website by any top up like E-Sewa [13]. The main advantages of this project over today’s old fashioned parking which is haphazard and improper is, space optimization, less human interference, time saving and cost effectiveness. Also, often owner has to face the damage of vehicle due to improper parking and it is hard to know who did such activities. This project has made this issue easier to resolve as everyone tag is registered in the database with timestamp. This helps in identifying the owner and benefits the victim.

One of the project objective was to provide information remotely about the availability of parking space, the project could not achieve the desired result. However, this unachieved result taught an important lesson that, one must be very careful in selecting the electronic component and getting the precise result out of it.

In this project it has been proved that using RFID tags and reader with a database, a secure and well managed parking lot can be achieved. This project not only provides automized parking but it can also manage records in a better way. By using a centralized database system, easy administration and access is possible. The admin can easily keep a check on the vehicles that are entering and leaving according to the date and timing. By using of this system, personnel cost will cut off and traffic jam problem will be solved by the faster check in and check out.

## 9. References

[1] “Indian Journal of Science and Technology” Chapter 2 [Online]. URL: http://www.indjst.org/index.php/indjst/article/view/92973/69540 [Accessed 22 September 2018].

[2] ”ECE 4231 Final Project,” Cornell University, 2010. Chapter 2.

[3] “International Journal of Computer Science and Information Technologies” Chapter 2 [Online] URL: http://ijcsit.com/docs/Volume%205/vol5issue06/ijcsit2014050610.pdf [Accessed 22 September 2018]

[4] “Figure 1” Chapter 3.1.1 [Online]. URL: http://www.reorientech.com/rfid.aspx [Accessed 13rd Dec 2018].

[5] “Figure 2” Chapter 3.1.1 [Online]. URL: https://www.instructables.com/id/Arduino-RFID-Reader-MFRC522-Turorial/ [Accessed 2nd Jan 2019].

[6] “Figure 3” Chapter 3.1.2 [Online]. URL: https://www.pololu.com/product/3109#lightbox-picture0J6077” [Accessed 23rd Nov 2018].

[7] “Figure 4” Chapter 3.1.3 [Online]. URL: https://raw.githubusercontent.com/SeeedDocument/W5500_Ethernet_Shield_v1.0/master/img/W5500.jpg [Accessed 13rd Dec 2018].

[8] “Figure 5” Chapter 3.1.4 [Online]. URL: https://www.module143.com/m14300164

[9] “Figure 6” Chapter 3.1.5 [Online]. URL: http://www.shopathomes.co/home/248-s3003-servo-motor.html [Accessed 13rd Dec 2018].

[10] “Figure 7” Chapter 3.1.6 [Online]. URL: https://components101.com/16x2-lcd-pinout-datasheet [Accessed 23rd Nov 2018].

[11] “Figure 8” Chapter 3.2.2 [Online]. URL: https://www.oreilly.com/library/view/php-mysql/9781449355517/ch01.html [Accessed 2nd Jan 2019].

[12] “A free hosting website” Chapter 3.2.5 [Online] URL: https://www.000webhost.com/

[13] “e-sewa Electronic Payment” Chapter 8 [Online] URL: https://esewa.com.np/#/home

[14] Alan Forbes, The Joy of PHP Programming: South-Western/Thomson Learning, 2004.

# Appendix

## 1. Code Used in Arduino Board

Comment begins with “//”

**Libraries imported**

```
#include <LiquidCrystal.h>
#include <SPI.h>
#include <Ethernet.h>
#include "RestClient.h"
#include <Servo.h>
#include "RFID.h"
#define LCDCONTRAST 8
#define EN 11
#define RS 12
#define D4 5
#define D5 4
#define D6 3
#define D7 2
#define RFID_2_PIN_SDA  48
#define RFID_2_PIN_RST  22
#define RFID_1_PIN_SDA  53
#define RFID_1_PIN_RST  23
RFID rfid_1(RFID_1_PIN_SDA,RFID_1_PIN_RST);
RFID rfid_2(RFID_2_PIN_SDA,RFID_2_PIN_RST);

int parkedSpace=0;

//For Servo
Servo inservo;
Servo outservo;

//For lcd
LiquidCrystal lcd(RS, EN, D4, D5, D6, D7);
```

**For Ehternet Setup part**

**connecting with the web through internet**

```
RestClient client = RestClient("rfidparkinglatest.000webhostapp.com");
String response;
void setup() {
  //LCD Setup part
  pinMode(LCDCONTRAST,OUTPUT);
  analogWrite(LCDCONTRAST,100);  lcd.begin(20, 4);

//Servo Setup
  Serial.begin(115200);
  inservo.attach(9);
  outservo.attach(10);
  for (int pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
          // in steps of 1 degree
          inservo.write(pos);              // tell servo to go to position in variable 'pos'
          delay(15);                       // waits 15ms for the servo to reach the position
        }
  for (int pos = 180; pos>= 0; pos -= 1) { // goes from 0 degrees to 180 degrees
          // in steps of 1 degree
          outservo.write(pos);              // tell servo to go to position in variable 'pos'
          delay(15);                           // waits 15ms for the servo to reach the position
        }



  lcd.setCursor(0,0);
  lcd.write("RFID SMART PARKING");
 //Ethernet setup
 Serial.println("connect to network");
  client.dhcp();
  // Can still fall back to manual config:
  byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
  //the IP address for the shield:
  byte ip[] = { 192, 168, 0, 11 };
  Ethernet.begin(mac,ip);
  Serial.println("Connected");
  Serial.println("Setup!");
  //Rfid Setup
    rfid_1.init();
    //initialize RFID module
   rfid_2.init();
}
```

**For running this program continuously unless manually stopped “Loop”**

```
  // LCD Part

void loop() {
  lcd.setCursor(0,0);
  lcd.write("RFID SMART PARKING");    //prints availability of Parking in LCD
  lcd.setCursor(0,1);
  lcd.print("Available: ");
  lcd.print(6-parkedSpace);
  lcd.setCursor(0,2);
  lcd.print("Occupied: ");
  lcd.print(parkedSpace);
  state();//Call the function to send the value to web site
  //Read RFID tag if present
  String rfid_1tag = rfid_1.readTag();
  rfid_1.printTag(rfid_1tag);
  Serial.println(rfid_1tag.length());
  if(rfid_1tag.length()>5){
```

**Create Charge**

```
    if(isValid(rfid_1tag)){
      if(checkBalance(rfid_1tag)){
        //send api call for creating charge
        createCharge(rfid_1tag);
        lcd.setCursor(0,4);
        lcd.print("Welcome");
                     parkedSpace++;
```

**Servo Motor Code for Every Condition**

```
        int pos=0;
          for (pos = 180; pos >= 90; pos -= 1) { // goes from 0 degrees to 180 degrees
            // in steps of 1 degree
```

```
      inservo.write(pos);              // tell servo to go to position in variable 'pos'
      delay(50);                    // waits 15ms for the servo to reach the position
    }
    for (pos = 90; pos <= 180; pos += 1) {  // goes from 180 degrees to 0 degrees
      inservo.write(pos);                      // tell servo to go to position in variable 'pos'
      delay(50);                    // waits 15ms for the servo to reach the position
    }
   delay(5000);
   lcd.clear();
   }

  else{                                           //low balance
   lcd.setCursor(0,4);
   lcd.print("Low Balance");
   delay(5000);
   lcd.clear();
   }
  }
 else{                                            //invalid card
   lcd.setCursor(0,4);
   lcd.clear();
   lcd.print("Invalid Card");
   delay(5000);
   lcd.clear();
  }
}
```

**RFID - RC522 RF IC Card Sensor Module #2**

```
String rfid_2tag = rfid_2.readTag();   //Read RFID tag if present
Serial.println(rfid_2tag.length());
if(rfid_2tag.length()>5){



  //Do the Servo work
  for (int pos = 0; pos <= 150; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    outservo.write(pos);              // tell servo to go to position in variable 'pos'
```

```
          delay(15);                             // waits 15ms for the servo to reach the position
        }
        for (int pos = 150; pos >= 0; pos -= 1) {
                                                                 // goes from 180 degrees to 0 degrees
          outservo.write(pos);                                   // tell servo to go to position in variable 'pos'
          delay(15);                                             // waits 15ms for the servo to reach the position
        }

    rfid_2.printTag(rfid_2tag);                                 //print the tag to serial monitor if one was discovered
    deductCharge(rfid_2tag);
                   parkedSpace--;
    }
}
```

### Function to check validity

```
int isValid(String uid){
   String query;
   char querychar[200];
   query="/api/isvalid/"+uid;
   query.toCharArray(querychar,query.length()+1);
   Serial.println(querychar);
   response = "";
   int statusCode = client.get(querychar, &response);
   Serial.print("Status code from server: ");
   Serial.println(statusCode);
   Serial.print("Response body from server: ");
   Serial.println(response);
   delay(1000);
   if(response=="valid"){
    return 1;
  }
  else{
    return 0;
    }
}
```

**Function to check Balance**

```
int checkBalance(String uid){
   String query;
   char querychar[200];
   query="/api/checkamount/"+uid;
   query.toCharArray(querychar,query.length()+1);
   Serial.println(querychar);
   response = "";
   int statusCode = client.get(querychar, &response);
   Serial.print("Status code from server: ");
   Serial.println(statusCode);
   Serial.print("Response body from server: ");
   Serial.println(response);
   delay(1000);
   if(response=="ok"){
    return 1;
  }
  else{
    return 0;
    }
}
```

**Function to create charge**

```
int createCharge(String uid){
   String query;
   char querychar[200];
   query="/api/create/"+uid;
   query.toCharArray(querychar,query.length()+1);
   Serial.println(querychar);
   response = "";
   int statusCode = client.get(querychar, &response);
   Serial.print("Status code from server: ");
   Serial.println(statusCode);
   Serial.print("Response body from server: ");
   Serial.println(response);
   delay(1000);
   if(response=="ok"){
      return 1;
  }
  else{
      return 0;
    }
}
```

**Function to deduct parking charge from the total balance**

```
int deductCharge(String uid){
   String query;
   char querychar[200];
   query="/api/deductcharge/"+uid;
   query.toCharArray(querychar,query.length()+1);
   Serial.println(querychar);
   response = "";
   int statusCode = client.get(querychar, &response);
   Serial.print("Status code from server: ");
   Serial.println(statusCode);
   Serial.print("Response body from server: ");
   Serial.println(response);
   delay(1000);
   if(response=="ok"){
    return 1;
  }
  else{
    return 0;
    }
}
```

**Availability of parking lot**

```
int state(){
  String query=(String)position;
  char querychar[200];
  query="/api/state/"+query;
  query.toCharArray(querychar,query.length()+1);
  Serial.println(querychar);
  response = "";
  int statusCode = client.get(querychar, &response);
  Serial.print("Status code from server: ");
  Serial.println(statusCode);
  Serial.print("Response body from server: ");
  Serial.println(response);
  delay(1000);
  if(response=="ok"){
    return 1;
  }
  else{
    return 0;
    }
}
```

## 2. Project Development and Layouts in Picture

Physical development of the project is shown in figure 13.

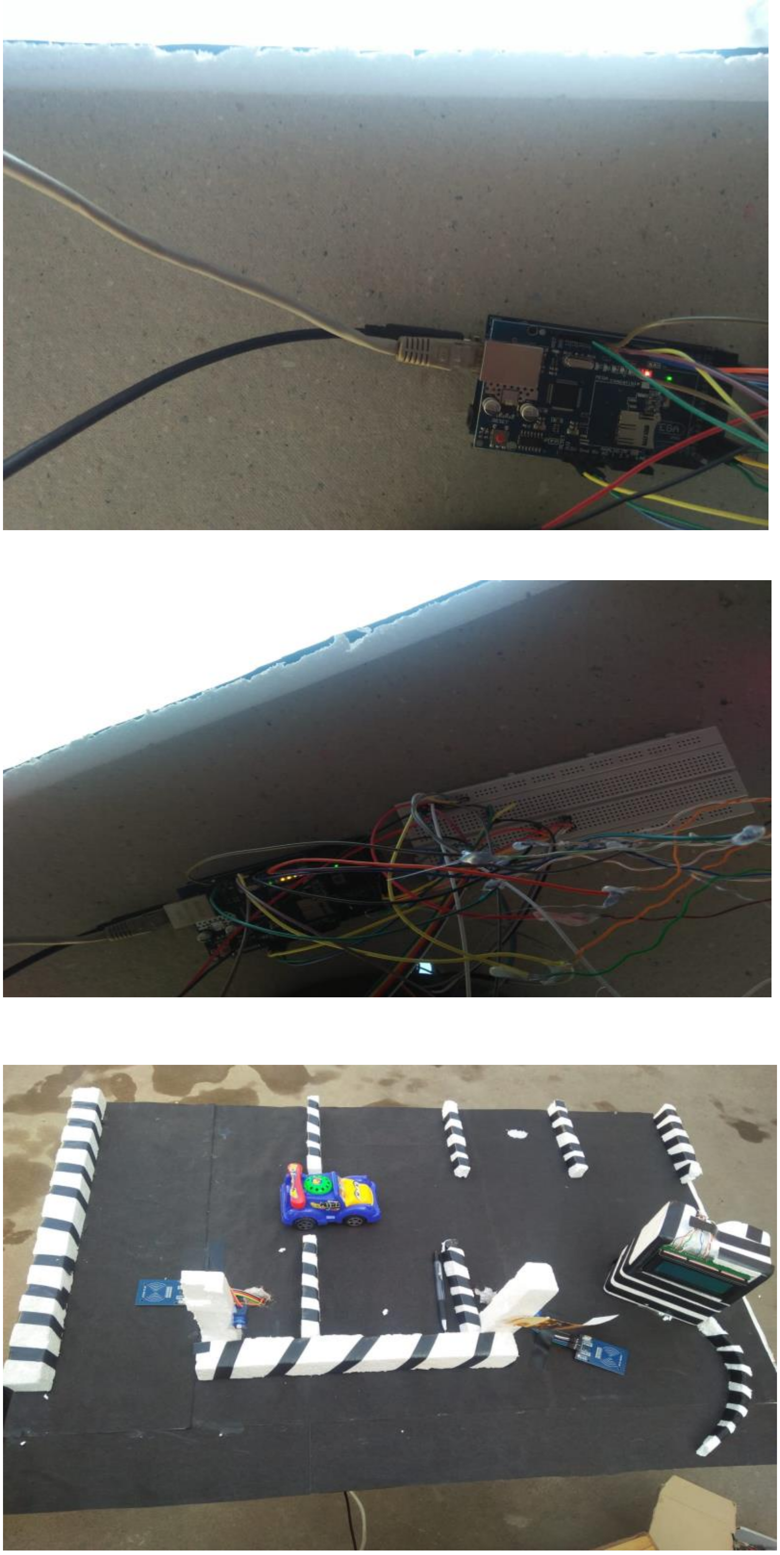
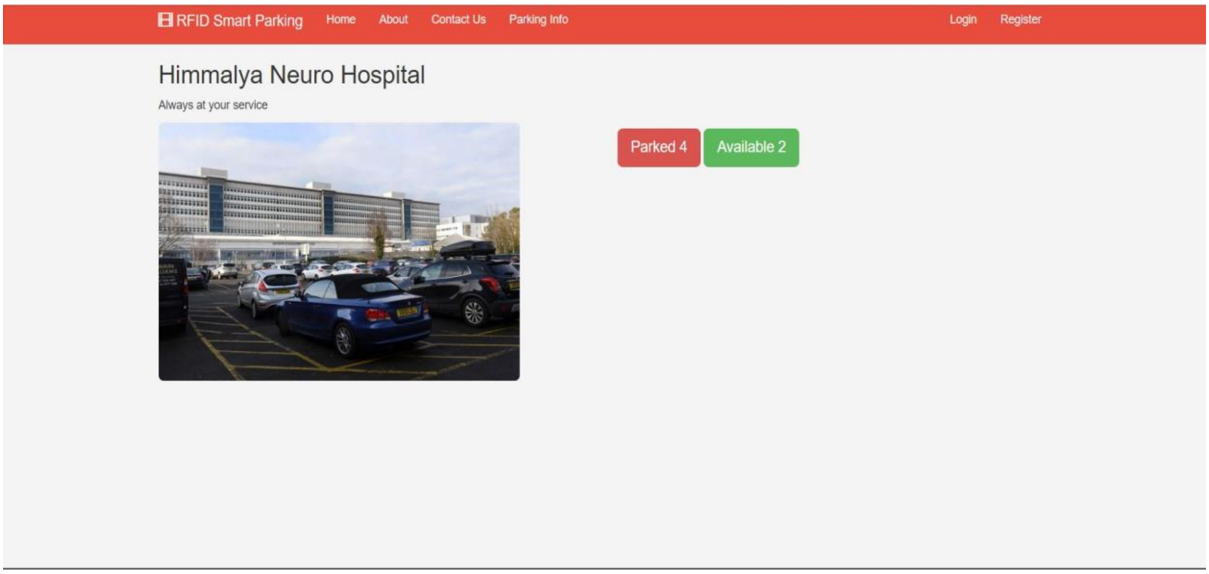


*Figure 13: Project Development Snapshot*

Figure 14 shows the frontend of the webpage.



Himmalya Neuro Hospital

Always at your service

*Fig 14: Frontend of Webpage*