ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 8, August 2020

IoT Based Parking System for Smart Cities

Jayalakshmi A[1], Lavanya C[2], Mohana Mala D.L[3], Pooja B.C[4], Thyagaraj. R[5]

UG Student, Dept. of Telecommunication Engineering, Bangalore Institute of Technology, Bangalore, India[1]
UG Student, Dept. of Telecommunication Engineering, Bangalore Institute of Technology, Bangalore, India[2]
UG Student, Dept. of Telecommunication Engineering, Bangalore Institute of Technology, Bangalore, India[3]
UG Student, Dept. of Telecommunication Engineering, Bangalore Institute of Technology, Bangalore, India[4]
Asst. Professor, Dept. of Telecommunication Engineering, Bangalore Institute of Technology, Bangalore, India[5]

Abstract: With growing popularity of smart Cities, there is always a demand for smart solutions for every province. The IoT has enabled the possibility of smart Cities with its internet control feature. There are multiple domains in a smart City and smart parking is one of the popular domain in the smart City. In this paper, an IoT based Smart Parking System consists of an on-site deployment of an IoT module that is used to monitor and signalize the state of availability of each single parking space is presented. A Mobile Application is also provided that allows an end user to check the availability of parking space and book a parking slot accordingly. Towards the end, the paper discusses the working of the system in the form of a use case that proves the correctness of the proposed Model.

Keywords: Smart Parking, IoT, Mobile Application, Firebase.

I. INTRODUCTION

During recent years with the increase in population in larger Cities the number of vehicles have also increased. This leads to an issue for the Drivers to park their Cars in parking lots. It needs an extra time, efforts and fuel consumption to find a parking space and they end up parking their Cars on the streets which later leads to Road blocks during peak hours [1]. Another issue linked to this problem is air pollution which delays the time to find the parking space resulting in more fuel consumption and release of pollutant gases.

To resolve these issues many advanced countries are adopting IoT based parking system and thus the Driver's safety is reinforced [2]. Smart Parking Systems helps in providing the information about the vacant space by using sensors deployed in the parking site. Smart Parking Systems are capable of providing necessary information making the parking space easily accessible to Drivers [3]. It also helps the Drivers to find a space for parking in a specific area which reduces the overall time used to search for it. This helps in reduction of pollutant gases.

II. PROBLEM IDENTIFICATION

As a part of problem analysis Orion Mall parking lot, Bangalore was observed. The parking system used is a traditional one. The vehicles entering the mall had to stop at the entry of the parking space, the person at the counter would issue a parking ticket which contained the time at which the vehicle entered the parking lot [4]. The person would then be permitted inside the parking lot and we could see the parking lot was almost filled and many people were searching around to find an empty parking slot. We could see them wasting their time just finding a parking slot leading to wastage of fuel as well [5]. When the person wished to exit, there would also be a commotion of remembering where they had parked previously [6].

Once again at the exit the person had to stop the vehicle give the slip and pay the amount and then leave the parking slot. The significant conclusion we drew from the visit was wastage of both fuel and time and the disarray of finding a parking slot [7].

Fig 1. Parking lot

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 8, August 2020

III. PROPOSED DESIGN

All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified.

A. Methodology

The main components of the block diagram is Ultrasonic sensors, ESP 8266. Ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object. High frequency sound waves reflect from boundaries to produce distinct echo patterns. ESP8266 Development Board is programmable via Arduino IDE. This is an ESP8266 based Wi-Fi enabled microprocessor unit on an Arduino-UNO footprint. The ESP8266 will be used as the main controller to control all the peripherals attached to it. ESP8266 is the most popular controller to build IoT based applications as it has inbuilt support for Wi-Fi to connect to internet.

Fig 1. shows the block diagram of the proposed system. The main idea of the project that is methodology beyond the block diagram is

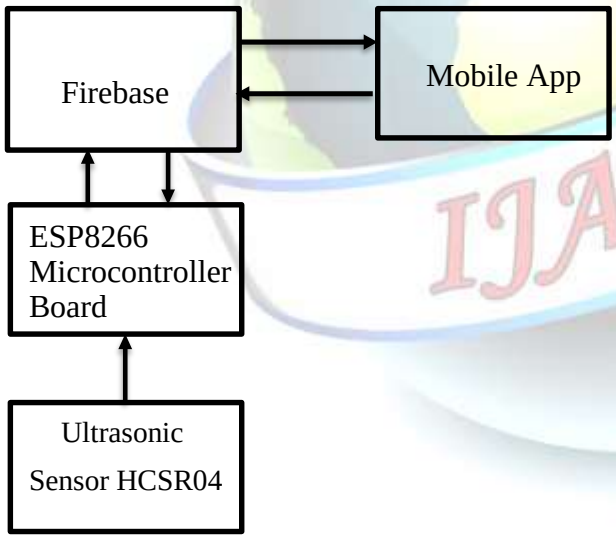

Fig 2. Block Diagram

- Firstly, the ultrasonic sensors are used to detect if the parking slot is available or occupied and sends the data to ESP8266.
- The ESP8266 will control the complete process and also sends the parking availability information to firebase.
- This data is sent to the Firebase for looking up the availability of space for vehicle parking. Here we are using firebase as database system as it is easy for developing app in multiple framework.
- Next a Mobile Application is developed to view and manage number of vehicles that can be parked at any given time based on the availability of parking slots.

B. Hardware Requirements

1. Ultrasonic sensor (HC-SR04)
2. Wi-Fi module (ESP 8266)

1) ULTRASONIC SENSOR(HC-SR04):

Ultrasonic sensor is an instrument that measures the distance to an Object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object proximity. Ultrasound is an acoustic wave with a very high frequency, beyond human hearing. Since the audible frequency range is said to be between 20Hz and 20 kHz, ultrasound generally means acoustic waves above 20 kHz. Bats, with their echolocation (biological ultrasonic radar), can hear sounds up to 200 kHz, way beyond the capabilities of the human ear.

Fig 3. HC-SR04 Ultrasonic Sensor

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 8, August 2020

2) ESP8266 MICROCONTROLLER:

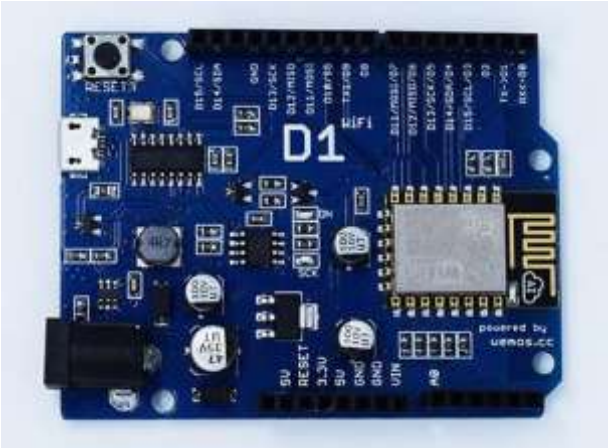

Fig 4. ESP8266 Development Board

This small module allows microcontrollers to connect to a Wi-Fi network and make simple connections ESP8266 Arduino core comes with libraries to communicate over Wi-Fi using TCP and UDP, set up HTTP, mDNS, SSDP, and DNS servers, do OTA updates, use a file system in flash memory, and work with SD Cards, servos, SPI and I2C peripherals. Besides adding Wi-Fi capability, the main claim to fame for the ESP8266 processor over the AVR processor of the standard Arduino is that it has a larger 4MB of Flash memory and runs at clock speeds of 80MHz and can sometimes optionally be overclocked to 160MHz and has a very fast processing speed. The Digital I/O except for D0 all support PWM and interrupts. In addition they can be configured to have pull-up or pull-down resistors. On the down-side, it has only 1 analog input which is probably the most significant limitation. That can always be overcome by using an external Analog.

C. Software Requirements

1. Arduino IDE
2. Firebase
3. HTML
4. CSS
5. JavaScript
6. Node.js

1) ARDUINO IDE:

Fig 5. Arduino IDE Screen

Arduino IDE where IDE stands for Integrated Development Environment-An official software introduced by Arduino.cc, that is mainly used for writing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go. It is easily available for operating systems like MAC, Windows and Linux runs on the Java platform that comes with inbuilt functions and commands that play a vital for debugging, editing and compiling the code in the environment. A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code. The main code also known as a sketch created on the IDE platform will ultimately generate a HEX file which is then transferred and uploaded in the controller on the board. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for the compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages

2) FIREBASE:

Firebase is a Mobile and web application development platform created by Firebase, Inc. Now it belongs to Google. It is used to develop web, Android, and iOS applications. Firebase consists of several integrated toolS

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 8, August 2020

for analysis, development, and maintenance of Mobile Applications. Each of these tools can be used separately.

Fig 6. Overview of Firebase

- *Firebase Hosting*

Firebase Hosting is built for the modern web developer. Websites and apps are more powerful than ever with the rise of front-end JavaScript frameworks like Angular and static generator tools like Jekyll. Whether you are deploying a simple app landing page or a complex Progressive Web App (PWA), Hosting gives you the infrastructure, features, and tooling tailored to deploying and managing websites and apps.

Firebase Hosting has lightweight hosting configuration options for you to build sophisticated PWAs. You can easily rewrite URLs for client-side routing or set up custom headers.

- *Firebase Authentication*

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

- *Firebase Realtime Database*

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in real time to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data. We can use any Firebase Realtime Database URL as a REST endpoint. All we need to do is append. json to the end of the URL and send a request from HTTPS client

Fig 7. Web Technology Structure

3) NODE.JS:

Node.js is an open source server environment. Node.js is a server-side platform.Node.js is free.Node.js runs on various platforms (Windows, Linux, UNIX, Mac OS X, etc.).

Node.js uses JavaScript on the server. A common task for a web server can be to open a file on the server and return the content to the client.

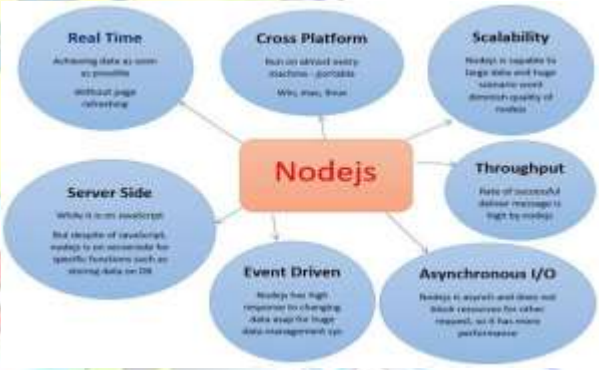

Fig 8. Node.js Features

IV. IMPLEMENTATION

Steps involved in the project:

1. Set up firebase platform
2. Preparing & Programming Arduino
3. Connecting the Sensors to Arduino
4. Developing Firebase
5. Developing Android Application

A. Firebase Console

Setting up Firebase Console:

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 8, August 2020

1. If you have Gmail id then you don't need to Sign Up for firebase, if you don't have Gmail id then Sign Up for one and then you can go to next step.
2. Open your browser and go to "firebase.google.com".
3. At the right top corner go to "Go to Console".
4. Click on "Add project".

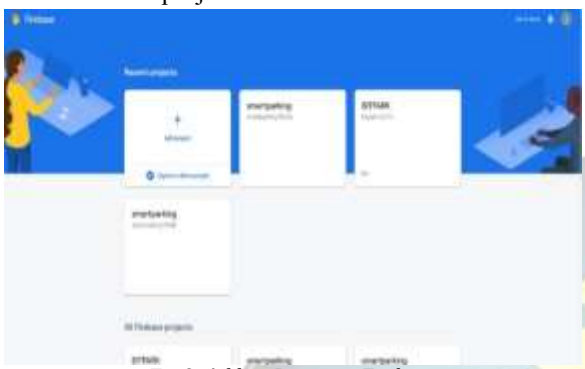

Fig 9. Adding Project to Firebase

5. Input your Project Name.
6. Accept the terms and condition, Create project and click on "Continue".

Now we have successfully created project. **Look for the Host Name and Authorization Key** also known as Secret Key. For this, follow steps given below:

7. Go to Settings Icon (Gear Icon) and click on "Project Settings".
8. Now click on "Service Accounts".

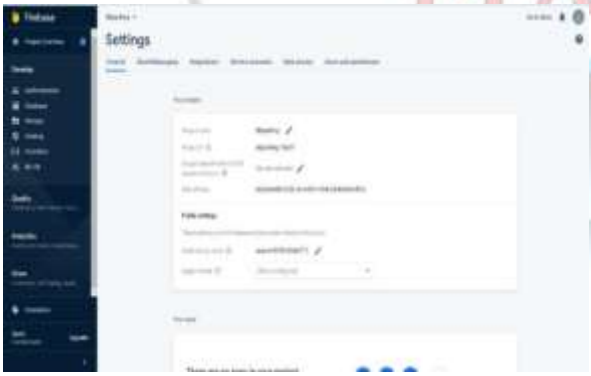

Fig 10. Service account in Firebase

9. You can see two options "Firebase admin SDK" and "Database Secrets".
10. Click on "Database Secrets".
11. Scroll on your project name and "Show" option appears at right side of your project.
12. Click on "Show" and now you can see secret key created for your project.
13. Copy the secret key and save it to notepad. This is your "FIREBASE_AUTH" string which we have written in Arduino program above.

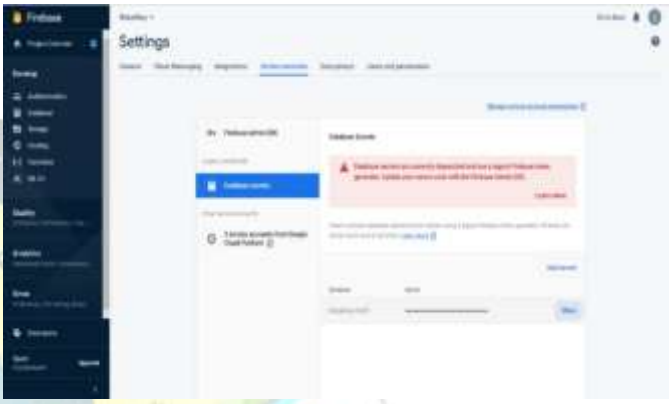

Fig 11. Firebase Authentication Key

14. Now go to "Database" on left control bar and click on it.
15. Scroll down and click on "Create Database".
16. Choose "Start in test mode" and click on "Enable".
17. Now your database is created and you will have to come to this section again to control LED.
18. Now just above the database you can see https://bitparking-16d37.firebaseio.com

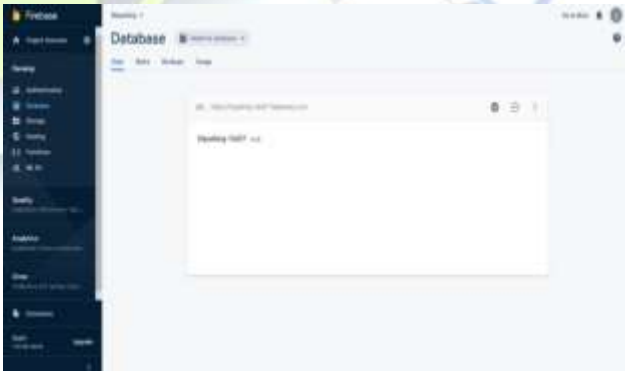

Fig 12. Project Created In Database

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 8, August 2020

19. Just copy https://bitparking-16d37.firebaseio.com without any slash and https and save it again to notepad just you had saved for secret key.
20. This is your "FIREBASE_HOST" string which we have written in Arduino program.

B. Programming The ESP8266 Using Arduino IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Fig 13. Arduino IDE

1) DOWNLOAD ESP8266-ARDUINO CORE:

With the Arduino IDE installed and operating on your computer, use your browser to download the ESP8266 support software from GitHub. We can download these files and manually copy them into the hardware folder of your Arduino IDE installation, but there is an easier and safer way. You should use the Arduino IDE Boards Manager. Search for the ESP8266 module in the Boards Manager. There are a few megabytes of file data to download and install, so be patient.

2) TEST THE ESP8266 SUPPORT:

A while later, when the utility completes the download and installation of the software, confirm that the ESP8266 Support is there in the Arduino IDE menu, click on Tools, Board, and scroll down to see the new ESP8266 section. There should be numerous boards there. The one we are interested mostly about, is the generic ESP8266 Dev Module, Like in the figure below.

Click on "ESP8266 Dev Module" to select it, and make it the active target in the IDE.

Copy the hardware definitions URL to the URLs field in the Preferences window. Click Ok to dismiss the Preferences window. Next, open the Board Manager utility by clicking Tools, Board and Boards Manager in the IDE menu.

Fig 14. Adding ESP Board Link to Arduino

Bring up the Boards Manager utility. Search for "ESP8266" in the text box. A single result should appear. Click on "Install" to do just that. In the Figure below, the ESP8266 support is already installed in my Arduino IDE, so the "Install" button is inactive.

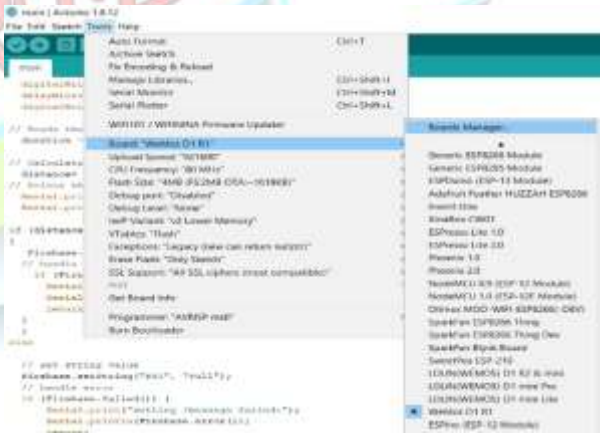

Fig 15. Appending WeMos D1 R1 into IDE

The ESP8266 support software provides support for numerous ESP8266 boards. Now that we have selected the WeMos D1 R1 confirm the selection in the IDE window. The status line should contain the text "WeMos D1 R1

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 8, August 2020

Module," as well as its various parameters, like its frequency and flash memory size. ESP8266 is the Arduino IDE programming target. Let's do one last test: Load one of the example ESP8266 sketches.

In the IDE menu, select File, Examples, ESP8266, AnalogOut, ledcWrite_RGB.

This concludes the installation and verification of the Arduino IDE support for the ESP8266 board where we can see the storage used maximum bytes stored locally and globally.

Fig 16. Program Window in Arduino IDE

C. *Interfacing ESP 8266 With HCSR04*

Fig 17. ESP8266 with HCSR04

ESP8266 works on 5 volts and its pins are also 5v TTL compatible. Whereas ultrasonic sensor Hcsr04 also works on 5 volts. Trigger and echo pins of hcsr04 ultrasonic sensor is directly connected to GPIO-2 and GPIO-0 or D9 and D8 pins of ESP8266. Code is written in Arduino ide. Esp8266 support package for Arduino ide. Let's move towards the ultrasonic distance calculation. Ultrasonic sensor did not calculate the distance by itself. Rather it only grabs the time taken by waves to leave transmitter and bounce back to receiver. Time is converted in to distance using the speed of sound in air formula. According to universal speed of sound in air for

Time=Distance/Speed

Where Speed=speed of sound in air. Which is 340 m/s.

D. *Deploying Firebase*

For the deployment of firebase, we used the node.js command prompt where first setup the file, by initialization of the command using '**Firebase init**' as shown in Fig 18. Firebase provides many features. Now we will use firebase hosting, so use <space> to select firebase CLI features.

Fig 18. Firebase Initialization

Once the initialization then set the project by giving the name as mentioned in the firebase to set off the project as shown in Fig 19.

Fig 19. Project Setup

Once the project is setup then add the web-app programs to the directory to be deployed which will build process for the assets, and once done the initialization will be completed as shown in Fig 20.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 8, August 2020

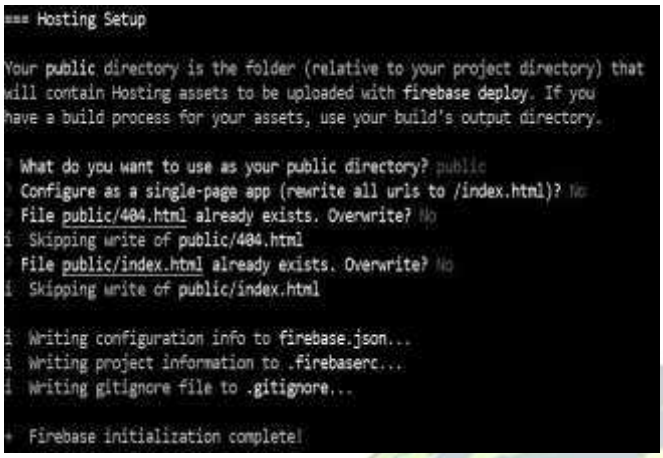

Fig 20. Hosting Setup

Finally the project is deployed using the command firebase deploy as shown in Fig 21.

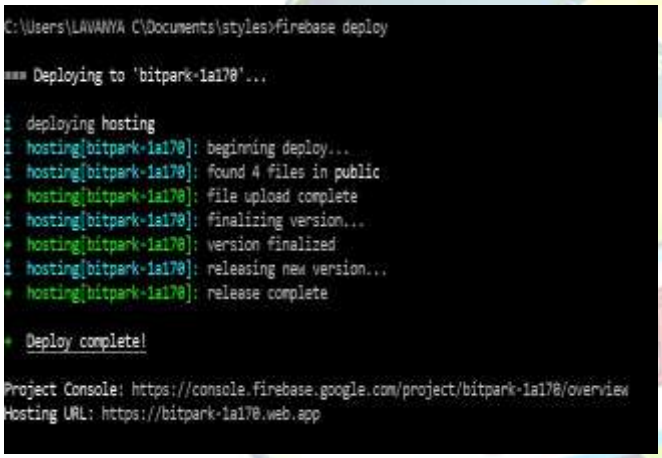

Fig 21. Firebase Deploy

It is very easy and fast to implement. It only takes 10 minutes for the deployment process, our project can be accessed online. Once done with deployment the project should be updated to the web-app where the hosting URL: http://bitpark1a170.web.app is displayed which is updated for creating the application.

E. Creating Mobile Application

For creating an android application the MIT App Inventor which , it uses a graphical user interface (GUI) very similar to the programming languages Scratch (programming language) and the Star Logo, which allows users to drag and drop visual objects to create an application that can run on Mobile devices.

Once clicked on the create app a Mobile is displayed by selecting a web browser a URL is displayed where the project URL is pasted in it.

Fig 22. Virtual App in MIT App Inventor

Next click on the build and select an App (provide QR code) to obtain a barcode as shown Fig 23. which is scanned from the Mobile to create our android application.

Fig 23. Barcode for Mobile App

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 8, August 2020

By scanning a QR code we get a package in the Mobile once it is installed the web application is created as we see the screen shot of the Mobile where the app **BIT_Parking_System** created is shown in Fig 24.

Fig 24. Mobile App Created Screen

F. Working of Proposed Model

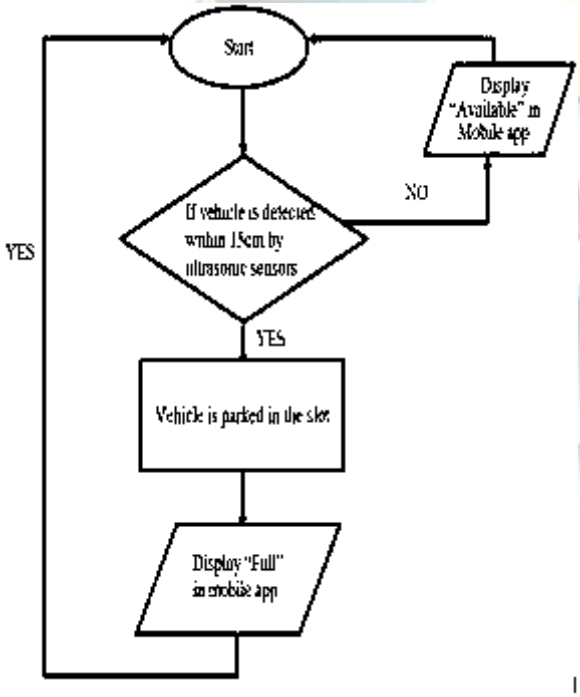

Fig 25. Flowchart of Proposed Model

V. RESULTS

The proposed Mobile Application displays the availability of a parking slot in a real time basis. The people who to use the app can view the number of parking slots available. ESP8266 Wi-Fi module helps in uploading the data to the cloud on a real time basis, hence the information displayed in the app is authentic.

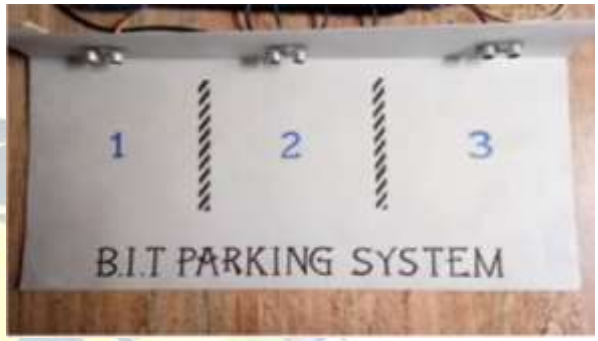

Fig 26. Parking system with sensors

Fig 27. Parking system with parked vehicle

In the figure shown in the Fig 27. The ultrasonic sensors are fixed in a perpendicular angle to ground. When ESP8266 is powered up (which is fixed internally) and Wi-Fi connection is established the ultrasonic sensors starts to sense the presence of a vehicle within 15cm from it and if a vehicle is found the same is updated in the Mobile app.

As shown in Fig 28. We can see that as the Cars are parked in the respective slots the Mobile app screen is also updated displaying the availability information in a real time basis.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 8, August 2020

Fig 28. Mobile App screen

VI. CONCLUSION

This documentation was about the Smart Parking System. This stemmed from the need for the better time management and productively along with the inspiration of new, developing technologies now available [8]. The Smart Parking idea was created to give instant access to information in a convenient and time-saving environment, the parking lot. The proposed project provides real time information of a Car parking lot and is able to coordinate with the Mobile Application.

This project is currently an Android-Platform Mobile Application. In the future this can be rebuilt in a multi-platform structure so that Windows and iPhone users can also use this App. We are currently using basic security protocols for the Application and the Cloud Database. But when the application is considered for real life implementation, more versatile encrypted security system can be considered.

REFERENCES

[1]. 2016 International Conference on Internet of Things and Applications (IOTA) Maharashtra Institute of Technology, Pune, India 22 Jan - 24 Jan, 2016.

[2]. International Journal of Scientific and Research Publications, Volume 5, Issue 12, December 2015 629 ISSN 2250-3153 www.ijsrp.org : Automatic Smart Parking System Using Internet of Things.

[3]. 2018 5th International Conference on Business and Industrial Research (ICBIR), Bangkok, Thailand: Smart Parking Using IoT Technology.

[4]. 2017 7th International Conference on Cloud Computing, Data Science & Engineering – Confluence: IoT Based Vehicle Parking Manager.

[5]. International Journal of Applied Engineering Research ISSN 09743-4562.

[6]. 2016 IEEE International Conference on Consumer Electronics (ICCE): Smart Parking System for Internet of Things.

[7]. www.rip.publications.com: Cloud Based Smart Parking System using IoT Technology.

[8]. 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON) GLA University, Mathura, Oct 26-28, 2017*: PSPS: An IoT Based Predictive Smart Parking System Volume 13, Number 8 (2018) pp. 5759-5765 © Research India Publications. http://www.ripublication.com 5759.*

BIOGRAPHY

JAYALAKSHMI A. is a final year student of Telecommunication Engineering in Bangalore Institute of Technology, Bangalore.

LAVANYA C. is a final year student of Telecommunication Engineering in Bangalore Institute of Technology, Bangalore.

MOHANA MALA D.L is a final year student of Telecommunication Engineering in Bangalore Institute of Technology, Bangalore

POOJA B.C. is a final year student of Telecommunication Engineering in Bangalore Institute of Technology, Bangalore.

Mr. THYAGARAJ R. is working as Assistant Professor in Telecommunication Department in Bangalore Institute of Technology, Bangalore.