

A MINI-PROJECT REPORT

ON

“Cloud Based E-Showroom Website using Firebase”

BY

Arnav Sankhe (BE - B848)

Aman Sarawgi (BE - B849)

Amit Kumar Sharma (BE - B855)

Under the supervision of

Prof. Bhushan Patil


MANJARA CHARITABLE TRUST
RAJIV GANDHI INSTITUTE OF TECHNOLOGY

Juhu-Versova Link Road Versova, Andheri(W), Mumbai-53

Department of Computer Engineering

University of Mumbai

April- 2020



CERTIFICATE

Department of Computer Engineering

This is to certify that

1. Arnav Sankhe (BE - B848)
2. Aman Sarawgi (BE - B849)
3. Amit Kumar Sharma (BE - B855)

Have satisfactorily completed this project entitled

“Cloud Based E-Showroom Website using Firebase”

Towards the partial fulfilment of the

**BACHELOR OF ENGINEERING
IN
(COMPUTER ENGINEERING)**

as laid by the University of Mumbai.

Supervisor

Prof Bhushan Patil

H.O.D.

Dr.Satish Y. Ket

Principal

Dr. Sanjay Bokade



CERTIFICATE

This is to certify that the mini project report entitled "**Cloud Based E>Showroom Website using Firebase**" is a bona-fide work of Arnav Sankhe, Aman Sarawgi and Amit Sharma during the academic year 2019-20 towards the fulfillment of the requirement for Final Year in **Computer Engineering**.

Prof. Bhushan Patil

Dr. Satish Ket

Head Of Department

Declaration

We wish to state that the work embodied in this project titled “**Cloud Based E-Showroom Website using Firebase**” forms our own contribution to the work carried out under the guidance of Prof Bhushan Patil at the Rajiv Gandhi Institute of Technology.

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Students Signatures)

Arnav Sankhe (BE-B848)

Aman Sarawgi (BE-B849)

Amit Kumar Sharma (BE-B855)

Abstract

In today's world, Cloud technology is a trending buzzword among business environments. It enables a user to store and share data like applications, files, and more to remote locations. To buy, sell or rent a car, the customer has to go to the showroom and invest a lot of time, even the showroom isn't available all the time. In today's digitally advanced world, this is very inconvenient. In our project we have implemented a cloud based E-Showroom site using Firebase which is a website where a customer can buy, sell or rent a car conveniently, efficiently and quickly. As it is a website, it is available at all times and can be accessed from any part of the world as long as the user is registered and can be very useful for the customer.

Contents

List of Figures		1
1	Introduction	
	1.1 Overview.....	2
	1.2 Objectives.....	2
.		
2	Framework	3
3	Details of Hardware & Software	6
	3.1 Hardware Requirement.....	
	3.2 Software Requirement.....	
4	Design Details	7
	4.1 System Architecture.....	
5	Implementation	10
5	Results	13
6	Conclusion	18

LIST OF FIGURES

Figure No.	Name	Page no.
1	System workflow	7
2	Firebase Initialisation	10
3	Firebase login Authentication	10
4	Firebase Image Loading Script	11
5	CSS code for webpage	11
6	Selling code	12
7	Renting car code	12
8	Login page for Customer	13
9	Main landing page/ dashboard.	13
10	Various Brands option	14
11	Car models selection for a particular brand	14
12	Cars model Images and Specification	15
13	User selling car	15
14	Renting a car	16
15	Firebase User Authentication	16
16	Image data storage	17

CHAPTER 1

Introduction

1.1 Overview

In today's world, Cloud technology is a trending buzzword among business environments . It enables a user to store and share data like applications, files, and more to remote locations. Today's shoppers and owners have higher expectations for your brand. They want a seamless journey through every stage of the customer life cycle.

Among numerous services provided by cloud computing, cloud storage service, such as Apple's iCloud, Microsoft's Azure and Amazon's S3, can offer a more flexible and easy way to share data over the Internet, which provides various benefits for our society.

In our project we would be making a cloud based E-showroom website using Firebase for a car dealership that wants to sell their products online and keep up with the competition and raise the bars for the competitors to enter. In this website, user is able to buy a car, sell his old car and rent any cars that are currently available. This data is stored on cloud using Firebase. The log in credentials and the personal data of the user is protected by the Firebase security system.

1.2 Objectives

The main motive or inspiration behind creating the project is to ease the work of the user. The main objectives are:-

- To ease the work of the customer.
- To make an easy way for customer to buy a car.
- To make an easy way for customer to sell a car.
- To make an easy way for customer to rent a car.
- To make the whole system cloud based.
- Access of showroom available 24/7.
- Provide secure interface for user to shop online.

CHAPTER 2

Framework

- **HTML :-**

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as `` and `<input/>` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

- **CSS :-**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices.

- **Javascript :-**

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behavior, and all major web browsers have a dedicated JavaScript engine to execute it. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). However, the language itself does not include any input/output (I/O), such as networking, storage, or graphics facilities, as the host environment (usually a web browser) provides those APIs.

- **Firebase :-**

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of March 2020, the Firebase platform has 19 products, which are used by more than 1.5 million apps.

Firebase frees developers to focus crafting fantastic user experiences. You don't need to manage servers. You don't need to write APIs. Firebase is your server, your API and your datastore, all written so generically that you can modify it to suit most needs. Yeah, you'll occasionally need to use other bits of the Google Cloud for your advanced applications. Firebase can't be everything to everybody. But it gets pretty close.

- **Node.js :-**

Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

- **PHP :-**

PHP is a popular general-purpose scripting language that is especially suited to web development. It was originally created by Rasmus Lerdorf in 1994; the PHP reference implementation is now produced by The PHP Group. PHP originally stood for *Personal Home Page*, but it now stands for the recursive initialism *PHP: Hypertext Preprocessor*. PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of a HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control. Arbitrary PHP code can also be interpreted and executed via command line interface (CLI). The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

CHAPTER 3

Detail of Hardware and Software

3.1 Hardware requirements

- Duo core+
- RAM -1GB+
- Hardisk-50GB+
- Input Devices-Mouse, keyboard etc.
- Active Internet Connectivity.

3.2 Software requirements

- Windows 7+
- Node.js for backend connectivity
- Internet Browser

CHAPTER 4

Design Details

4.1 System Architecture:

The Basic Flow of the system is as follows.

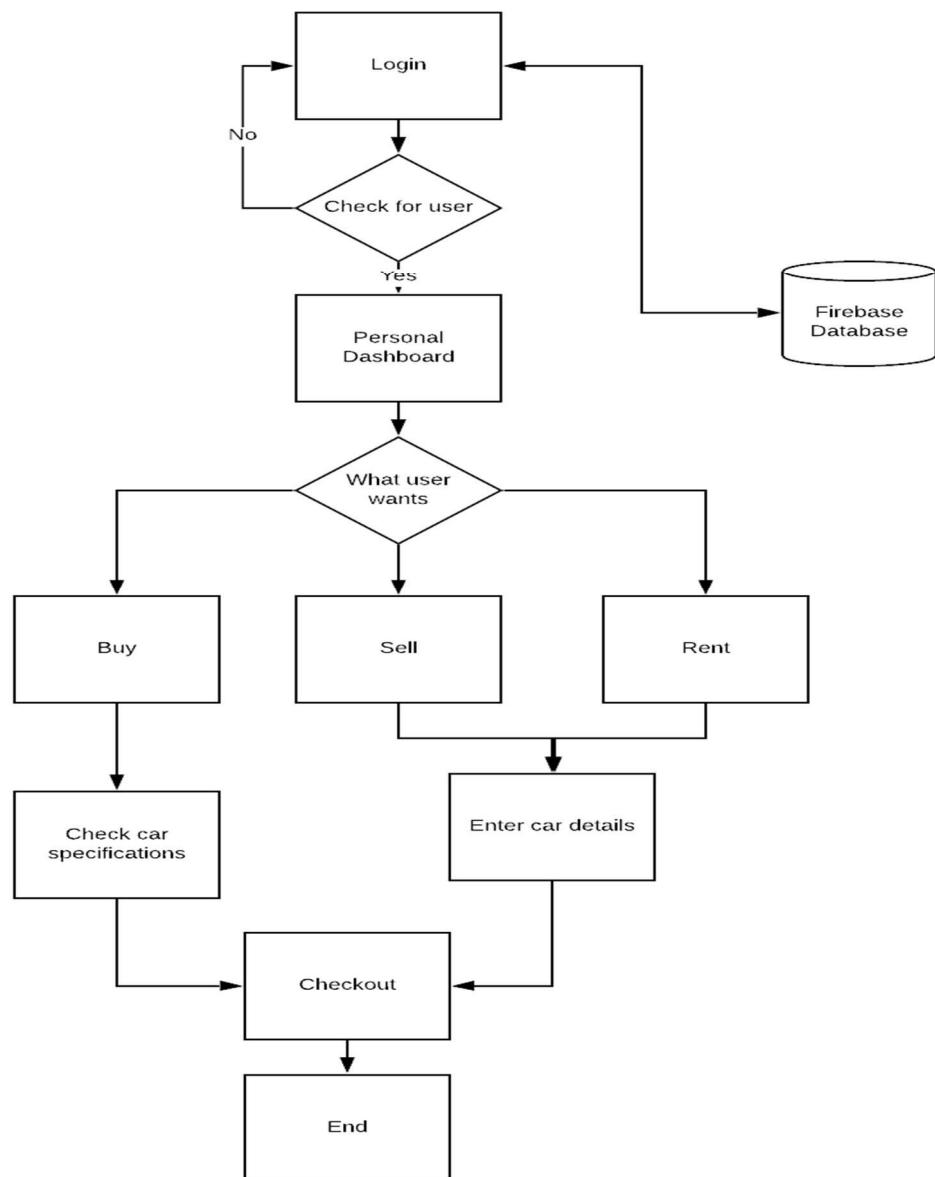


Fig 1 : System workflow

The user is first on the landing page where the user has to provide his login information for accessing the features of the website. If the user is not registered with the website then he can provide his information such as Name, Email, Password(Confidential) before logging in. After login the process is successful the user is redirected to the dashboard of the website. In the dashboard the user can access the services provided by website: Buy, Sell, Rent, Contact for information.

If the user wants to buy any car then they can select the car brand before finding out which car models are available to buy. The models of the cars have detailed specifications of the cars such as Year, Class, Mileage, Engine Size, Horsepower, etc. They can check the details and images of the cars before the customer makes up the mind whether he will buy the car or not. In this website the customer can also become seller if they want to sell their own cars to the dealership.

The user must have to provide information about the car he wants to sell. The information of the car provided must be authentic so they can assess with reasonable price. If the customer wants to get a car temporarily then he can also rent a car by selecting the available models of the cars. After the customer has selected a car from "buy" & "rent" option then they can checkout by paying the correct amount of money they have been asked. All users data and the information of the cars are stored in Firebase cloud platform which provides quick and efficient method for cloud computing.

Key Capabilities:

Realtime	Instead of typical HTTP requests, the Firebase Realtime Database uses data synchronization—every time data changes, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code.
Availability	Available 24/7
Accessible from Client Devices	The Firebase Realtime Database can be accessed directly from a mobile device or web browser; there's no need for an application server. Security and data validation are available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or written.

Scale across multiple devices	With Firebase Realtime Database on the Blaze pricing plan, you can support your app's data needs at scale by splitting your data across multiple database instances in the same Firebase project. Streamline authentication with Firebase Authentication on your project and authenticate users across your database instances.
--------------------------------------	---

Chapter 5

Implementation

```
// Your web app's Firebase configuration
var firebaseConfig = {
  apiKey: "AIzaSyA2ckAeTAG8ch1nbJMWWHk0l3-p8IB7N1M",
  authDomain: "cars-91a26.firebaseio.com",
  databaseURL: "https://cars-91a26.firebaseio.com",
  projectId: "cars-91a26",
  storageBucket: "cars-91a26.appspot.com",
  messagingSenderId: "713005989129",
  appId: "1:713005989129:web:024d128c0213b56697e7d8",
  measurementId: "G-7B11QDBQDL"
};

// Initialize Firebase
firebase.initializeApp(firebaseConfig);

var storage = firebase.storage();

// Create a storage reference from our storage service
var storageRef = storage.ref();
```

Fig 2: Firebase Initialisation

```
const auth = firebase.auth();

function signUp() {
  var email = document.getElementById("email");
  var password = document.getElementById("password");

  const promise = auth.createUserWithEmailAndPassword(email.value, password.value);
  promise.catch(e => alert(e.message));
  alert("SignedUp");
}

function signIn() {
  var email = document.getElementById("email");
  var password = document.getElementById("password");

  const promise = auth.signInWithEmailAndPassword(email.value, password.value);
  promise.catch(e => alert(e.message));
  alert("Welcome: " + email.value);
  window.location = "dashboard.php"
}
```

Fig 3: Firebase login Authentication

```

<script src="https://www.gstatic.com/firebasejs/7.13.2.firebaseio-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.13.2/firebase-auth.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.13.2/firebase.js"></script>
<script src="form.js"></script>

<script>
  storageRef.child('/car1.jpg').getDownloadURL().then(function(url) {
    // `url` is the download URL for 'images/stars.jpg'

    // This can be downloaded directly:
    var xhr = new XMLHttpRequest();
    xhr.responseType = 'blob';
    xhr.onload = function(event) {
      var blob = xhr.response;
    };
    xhr.open('GET', url);
    xhr.send();

    // Or inserted into an <img> element:
    var img = document.getElementById('car1');
    img.src = url;
  }).catch(function(error) {
    // Handle any errors
  });
</script>

```

Fig 4: Firebase Image Loading Script

```

a {
  background-color: transparent;
}

.generalTitle {
  color: #bb8847;
  margin: 1.5em auto;
  font-weight: 300;
  padding-left: 80px;
}

.carinfo{
  position: inherit;
  background-color:white;
  height: 120px;
  width:100%;
  border:solid 3px #bb8847;
  border-radius: 20px;
  margin-top: 10px
}

body {
  background: black;
  font-family: "Lato", sans-serif;
  margin: 10px;
}

.sidenav {
  height: 100%;
  width: 0;
  position: fixed;
  z-index: 1;
  top: 0;
  left: 0;
}

```

Fig 5: CSS for webpage

```

<div class="container">
    <center></center>
    <div class="bottom-left" align="left" ><h1>SELL YOUR CAR</h1>
        WE LOVE CARS. NEW AND OLD. BIG AND SMALL. CONVERTIBLE AND SUV. <br>
        LET US TAKE CARE OF THE SALES PROCESS. </div>
</div>
<center>
<form method=POST>
<table cellpadding = "5">

<tr>    <td><input type="text" size="30px" placeholder="YOUR NAME*" name="name" echo $row['F_NAME']; required="required"></td></tr>
<tr>    <td><input type="text" size="30px" placeholder="YOUR EMAIL*" name="email" required="required"></td></tr>
<tr>    <td><input type="text" size="30px" placeholder="YOUR PHONE*" name="phone1" required="required"></td></tr>
<tr>    <td> <select name="brand">
        <option>CAR BRAND*</option>
        <option value="bmw">BMW
        <option value="tesla">TESLA
            </select>
        </td></tr>
    <tr><td><input type="text" placeholder="CAR MODEL*" name="cmodel" required="required"></td></tr>
    <tr> <td> <input type="text" size="30px" placeholder="YEAR*" name="year" required="required"></td></tr>
    <tr><td> <input type="text" size="30px" placeholder="MILAGE (KM)*" name="cmileage" required="required"></td></tr>
    <tr> <td> <input type="text" size="30px" placeholder="EXTERIOR CONDITION* : BEST/GOOD" name="excon" required="required"><td>BEST/GOOD</td></tr>
    <tr> <td> <input type="text" size="30px" placeholder="INTERIOR CONDITION* : BEST/GOOD" name="incon" required="required"><td>BEST/GOOD</td></tr>
    <tr> <td> <input type="text" size="30px" placeholder="ACCIDENT HISTORY* : YES/NO" name="ahistory" required="required"><td>YES/NO</td></tr>
    <tr> <td> <input type="text" size="30px" placeholder="MODIFIED/GCC SPECS*: YES/NO" name="modified" required="required"><td>YES/NO</td></tr>
    <tr>
        <td><center><input class="btn" type="submit" name="SELL"></center>
    </td></tr>
</tr>
```

Fig 6: Selling car code

```

<h1 style="font-size: 20px; text-align:center; color:#bb8847; ">RENT A CAR</h1>
<form method="POST">

    <table cellpadding = "15">

        <tr>
            <td><input type="text" size="30px" placeholder="NAME*" required="required" name="name" maxlength="30"/></td>
        </tr>

        <tr>
            <td><input type="text" size="30px" placeholder="PHONE*" required="required" name="phone1" maxlength="10"/></td>
        </tr>

        <tr>
            <td><input type="text" size="30px" placeholder="EMAIL*" required="required" name="email" maxlength="30"/></td>
        </tr>
        <tr>
            <td><select name="brand" style="width: 340px; ">
                <option>CAR BRAND*</option>
                <option value="tesla">TESLA
                <option value="astonmartin">ASTON MARTIN
                <option value="bmw">BMW
                <option value="jeep">JEEP
                <option value="infinity">INFINITY
            </select></td>
        <tr>
            <td><input class="btn" type="submit" name="SHOW AVAILABILITY"></td>
```

Fig 7: Renting car code

Chapter 5

Result:

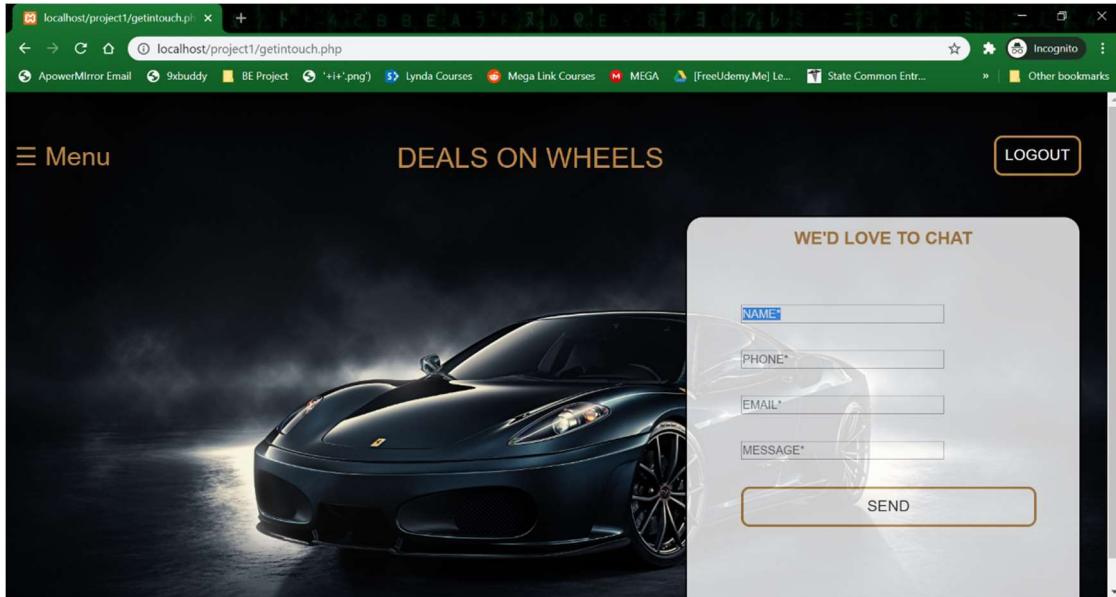


Fig 2 : Login page for Customer

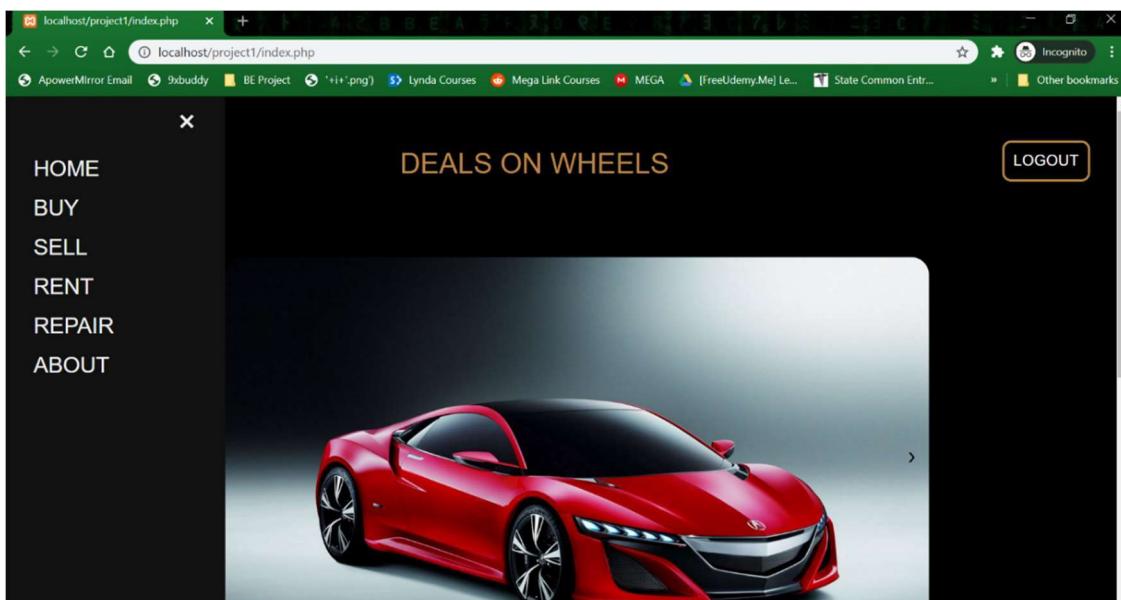


Fig 3 : Main landing page/ dashboard.

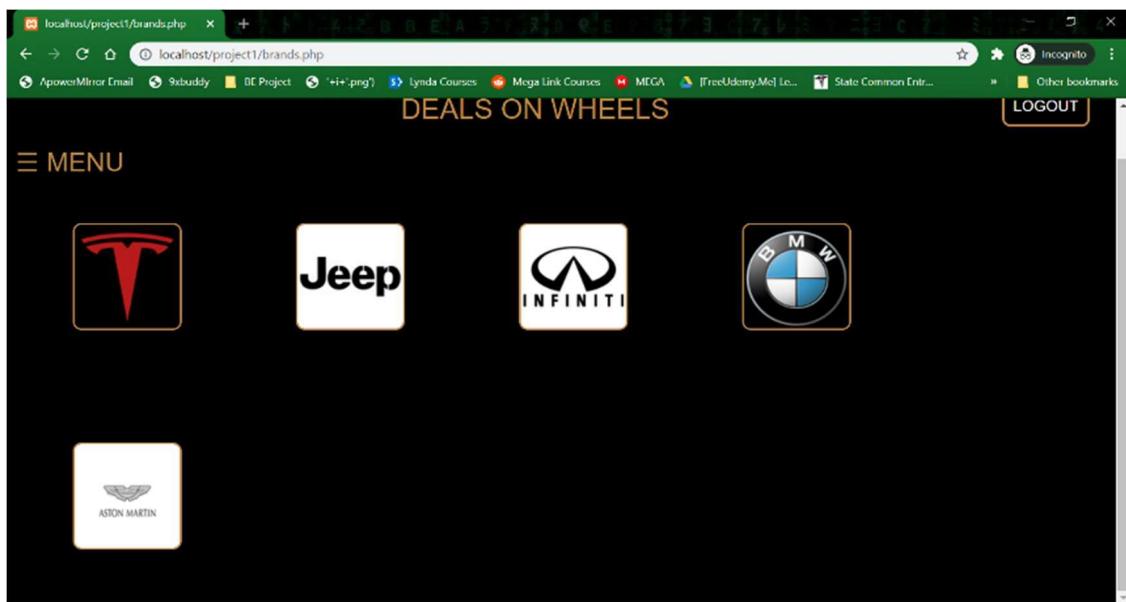


Fig 4: Various Brands Options

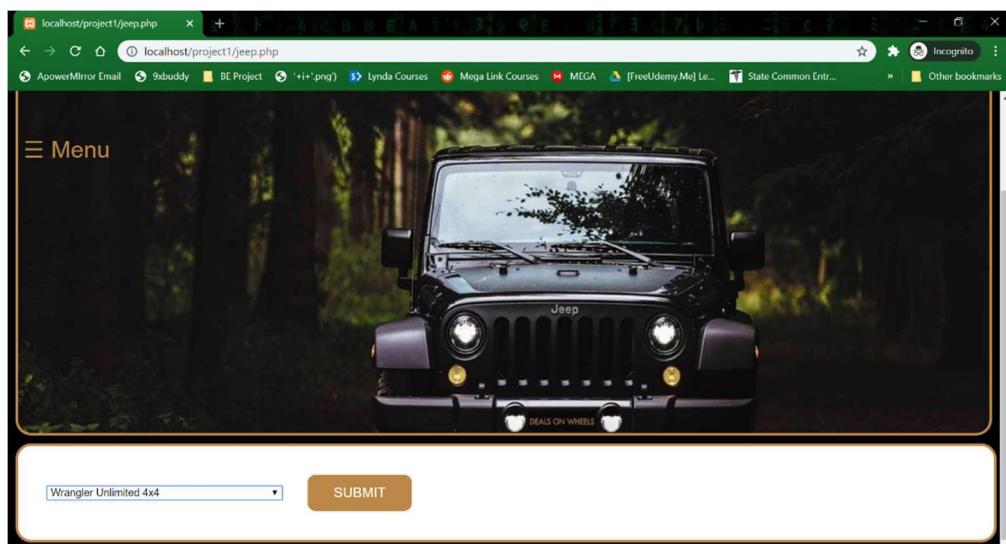


Fig 5: Car models selection for a particular brand

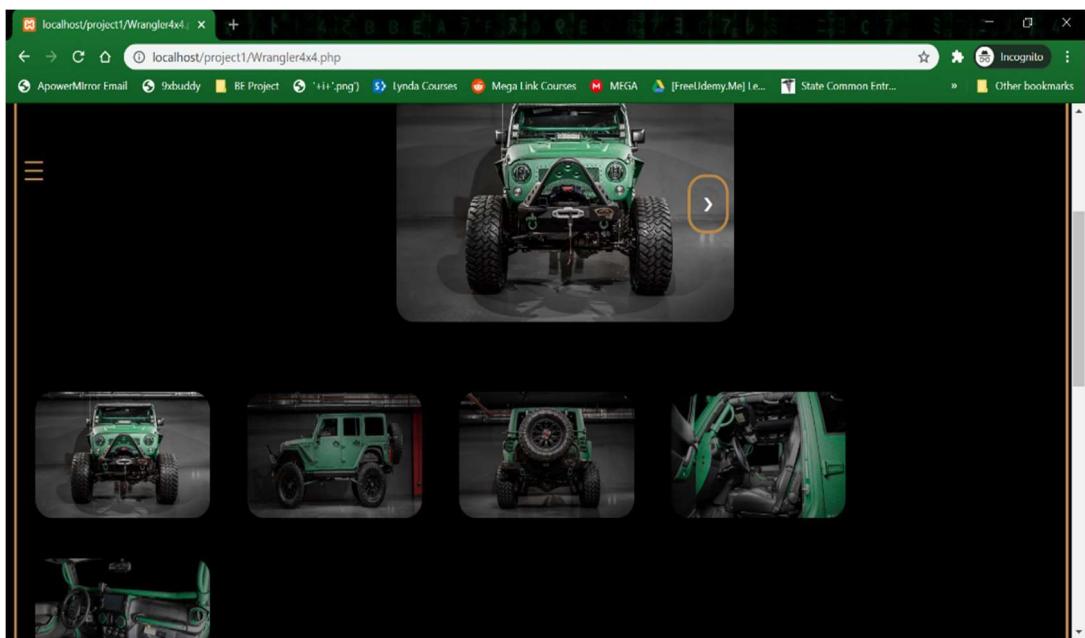


Fig 6: Cars model Images and Specification.

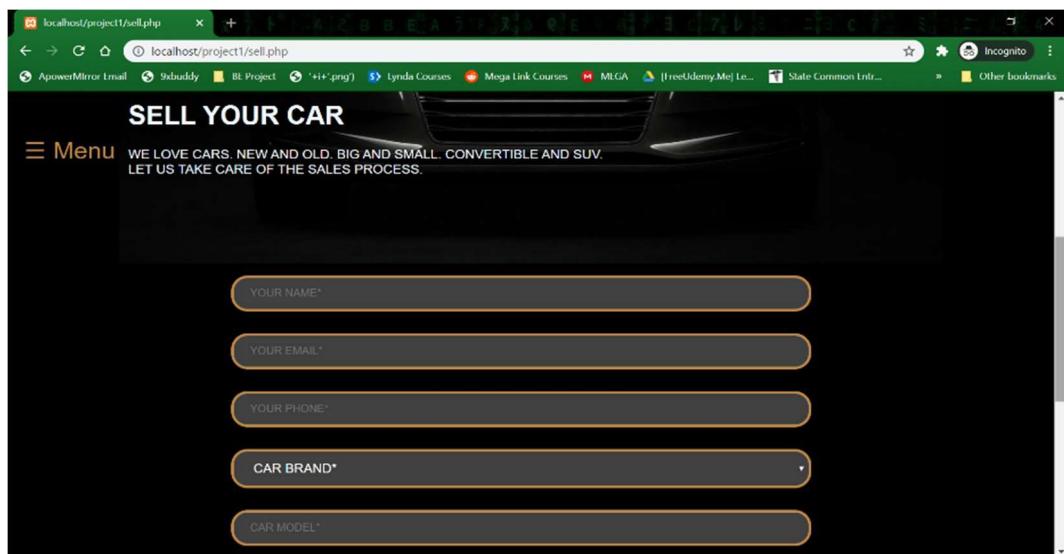


Fig 7: User selling car.

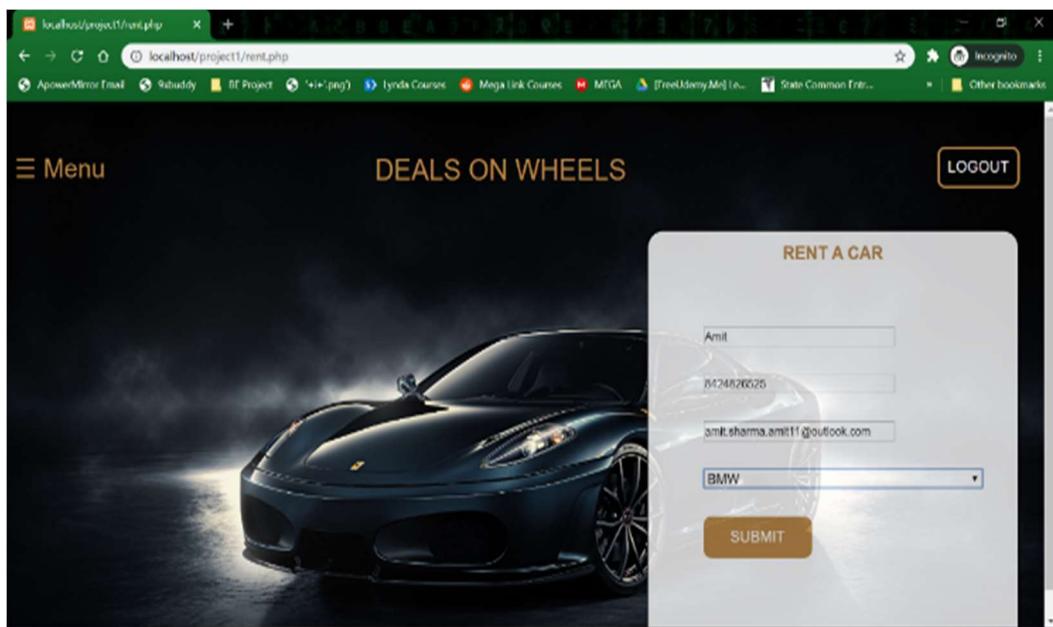


Fig 8: Renting a car.

A screenshot of the Firebase Authentication console. The sidebar shows "Project Overview" and "Develop" sections with "Authentication" selected. The main area is titled "Authentication" and shows a table of users. The columns are "Identifier", "Providers", "Created", "Signed In", and "User UID". The data rows are:

Identifier	Providers	Created	Signed In	User UID
amit@123.com	[Email]	Apr 5, 2020		L4Dyd9BYgJhQdgrTQQfQQfMS1D...
amitsharma@gmail.com	[Email]	Apr 5, 2020		71kX1ZFJ9GaAcEzJE1bXgWp5Rwc2...
aman@gmail.com	[Email]	Apr 5, 2020		iDrD9hicy2Y9VJd5BYx7TwYkLQB3...
arnav@gmail.com	[Email]	Apr 5, 2020		HJ9vDr1JYBfd234bAzirWyuqSSq2...

With a search bar at the top and pagination controls at the bottom.

Fig 9: Firebase User Authentication

The screenshot shows the Firebase Storage interface for a project named 'Cars'. The left sidebar includes links for Project Overview, Authentication, Database, Storage, Hosting, Functions, and ML Kit. The Storage section shows a list of uploaded files:

Name	Size	Type	Last modified
2.jpg	124.81 KB	image/jpeg	Apr 5, 2020
5.jpg	140.35 KB	image/jpeg	Apr 5, 2020
6.jpg	191.43 KB	image/jpeg	Apr 5, 2020
7.jpg	296.72 KB	image/jpeg	Apr 5, 2020
as1.jpg	307.3 KB	image/jpeg	Apr 5, 2020
as11.jpg	120.18 KB	image/jpeg	Apr 5, 2020
as12.jpg	102.98 KB	image/jpeg	Apr 5, 2020

Buttons for 'Upload file' and 'Extensions' are visible at the top right.

Fig 10: Image data storage

Chapter 6

Conclusion

In conclusion, cloud computing is recent technological development that has the potential to have a great impact on the world. It has many benefits that it provides to its users and businesses. For example, some of the benefits that it provides to businesses, is that it reduces operating cost by spending less on maintenance and software upgrades and focuses more on the businesses itself. But there are other challenges the cloud computing must overcome. People are very skeptical about whether their data is secure and private. We provide a safe and secure platform for customers to buy, sell and rent a car and carry out the transactions. The data of the customer is kept safe and confidential and thus a website is very quick and convenient for the customer. Cloud computing is the future of technology and once the standards and regulations are set worldwide, it will revolutionize the future.