

PROFESSIONAL TRAINING REPORT
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Technology Degree in Information Technology

By
Manish Singh (Reg. No – 40120054)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

April 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Manish Singh (40120054)** who carried out the project entitled "AUTOMOBILE SELLING APPLICATION" under my supervision from Jan 2023 to April 2023.

Internal Guide

Dr.R. Jeberson Retna Raj, M.Tech., Ph.D

Head of the Department

Dr. R. Subhashini, M.E., Ph.D

Submitted for Viva Examination held on **28/04/23**

Internal Examiner

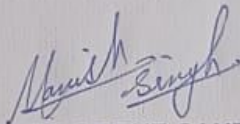
External Examiner

DECLARATION

I, **MANISH SINGH** (Reg. No – 40120054), hereby declare that the project report entitled " **AUTOMOBILE SELLING APPLICATION**" was done by me under the guidance of Dr. Jeberson Retna Raj is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Information Technology.

DATE: 28/04/23

PLACE: Chennai


SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D**, Dean, School of Computing, **Dr.R.Subhashini, M.E., Ph.D.**, Heads of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide, **Dr.R.Jeberson Retna Raj** for his valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

Car selling website is where the customer can buy their dream cars and development of an interactive car sale system which lets a customer to find a car and its details is the main objective of this project. Both the user and the administrators can access enter the details of every car. This website is user friendly and easy to use for any customer, it has effective popular vehicles and featured vehicles sliders to get the precise and suitable car for the user. A rich customer reviews facility is always there from where customers can read the existing clients reviews and this will work as a trust factor for a new user. The website displays information of the contact person on the homepage so that user can directly contact them or meet them which help in decreasing the communication gap. This website also provides a facility or services including parts repair, regular service, insurance, battery change, etc. The social media plugin will help the company to know their popularity over the social media.

CONTENTS

Chapter No.	Table of contents	Page Number
	Abstract	I
	List of Figures	II
	List of Abbreviations	III
1	Introduction	1
1.1	Introduction to Automobile Selling Application	1
1.2	Introduction to Web Development	1
1.3	Front-End Development	1
1.4	Back-End Development	3
1.5	RWD	3
1.6	Types of Web Applications	6
2	Aim and Scope of Present Investigation	7
2.1	Aim	7
2.2	Scope	7
2.3	Important Features	10
3	Methods and Algorithms used	13
3.1	Technologies Used	13
3.2	HTML	13
3.3	CSS	18
3.4	Javascript	23

3.5	Swiper JS	24
3.6	PHP	27
3.7	XAMPP	29
4		30
	Discussions, Results and Performance Analysis	
4.1	Requirements	30
4.2	Results and Performance Analysis	30
5		31
	Summary	
5.1	Summary and Conclusion	31
	APPENDIX	32
	REFERENCES	36

LIST OF FIGURES

Figure No	Figure Title	Page Number
1.5.1	Responsive Web Development	5
1.5.2	RWD Example	5
2.2.1	Nissan India website	8
2.2.2	Shop@Home	8
3.2.1	Viewport	13
3.2.2	Favicon	14
3.2.5	LOGIN Form	16

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
HTML	Hyper Text Markup Language.
CSS	Cascading Style Sheets.
JS	JavaScript.
RWD	Responsive Website Development.
SPA	Single Page Application.
CMS	Content Management System.
GIF	Graphical Interchange Format.
PHP	Hypertext Preprocessor.
XML	Extensible Markup Language.
API	Application Programming Interface.
ASP	Active Server Pages.
AI	Artificial Intelligence.
XAMPP	X-operating system, Apache, Mysql, Php, Perl
RAM	Random Access Memory.
DOM	Document Object Manipulation.

CHAPTER 1:

INTRODUCTION

1.1 Introduction to Automobile Selling Application.

The main purpose behind developing an automobile selling application is to facilitate the buyers to have a wide range of options of cars to buy under a single roof. Buyers have a access to the details regarding the cars those are available so If buyer's are after a stress-free car buying experience, super savings and a wide selection of quality used cars, visit the website and select your dream car for affordable price.

1.2 Introduction to Web Development.

Web programming, also known as web development, is the creation of dynamic web applications. Examples of web applications are social networking sites like Facebook or e-commerce sites like Amazon.

There are two broad divisions of web development – front-end development (also called client-side development) and back-end development (also called server-side development).

Hardware, computer programs, and websites comprise many components: code, databases, user interfaces (UI). Those that interact with users are in the front end; those that operate the technology are situated in the back end, also called the data-access layer. The back end provides the resources to support the front end.

1.3 Front-End Development.

Front-end development refers to constructing what a user sees when they load a web application – the content, design and how you interact with it. This is done

with three codes – HTML, CSS and JavaScript. It facilitates building components that interact with users. Examples are the user interface, buttons, user-entered data, websites, and user experience (UX) features. The front end aims at meeting user requirements and delivering a positive user experience.

1.3.1 Introduction to HTML.

HTML, short for Hyper Text Markup Language, is a special code for ‘marking up’ text in order to turn it into a web page. Every web page on the net is written in HTML, and it will form the backbone of any web application.

1.3.2 Introduction to CSS.

CSS, short for Cascading Style Sheets, is a code for setting style rules for the appearance of web pages. CSS handles the cosmetic side of the web.

1.3.3 Introduction to Javascript.

Javascript is a scripting language that’s widely used to add functionality and interactivity to web pages.

Front-end frameworks accord you ready-made code and components, such as prewritten standard functions packaged as libraries, with no need to build common functionality and components from scratch. Some of them are...

- Angular
- React
- jQuery
- Vanilla.js
- Vue.js

1.4 Back-End Development.

Back-end development controls what goes on behind the scenes of a web application. A back-end often uses a database to generate the front-end. Examples are components for data storage, infrastructure, integration with external systems, and code written in one or more programming languages. Users cannot access the back end.

Back-end scripts are written in many different coding languages and frameworks, such as...

- PHP
- Ruby on rails
- ASP.NET
- Perl
- Java
- Node.js
- Python

1.5 Responsive Web Development

RWD, also known as responsive design, is a web-design approach that emphasizes user experience. The objective is to ensure consistent rendering of webpages on various devices, screens and windows of different shapes and sizes, etc. Such an approach has become important given the rise of mobile devices, with most web traffic coming from mobile-device users. Real-world implementations of RWD abound, which most users, who focus on only the convenience and appeal of websites, rarely notice.

Following are several techniques with which to make websites responsive to various conditions:

- **Flexible grids:** Grids are a standard design tool for building websites. Responsive websites require a flexible grid that can load in different ways, depending on the screen or window size.
- **Breakpoints:** Similar to flexible grids, breakpoints are dots on a page that identify cut-off points so that information can move on the screen. Most websites have numerous breakpoints, but a minimum should be three to correspond to the three most common device types.
- **Prioritization:** Effective designs usually place the most important images and information higher up on the webpage, ensuring that visitors see them first. This visual hierarchy is especially crucial for responsive websites because smaller screens display less of a page at a given time. Prioritizing the important elements keeps mobile-device users engaged.
- **Flexible images:** Images are often harder to fit into different screen sizes than plain text. Web designers use various techniques to ensure that web pages display images appropriately on different screens. For example, the display might crop parts of an image for a better fit. In the case of multiple image versions for a site, select the version to render based on the device type or screen size.
- **Responsive media queries:** These are coding commands that set the maximum and minimum dimensions for media assets and that specify their orientation. As a result, all media assets on a webpage load at the appropriate size.
- **Mouse and touchscreen-friendly elements:** Desktop users are adept with their mouse device, but some actions are less intuitive on mobile devices. Responsive websites must accommodate smartphones by ensuring that every action is easily accessible through a touchscreen. Links must be large and obvious on small screens, and scrolling must be easy across device types. Be sure to test the user experience to ensure that all the elements work well on different devices.

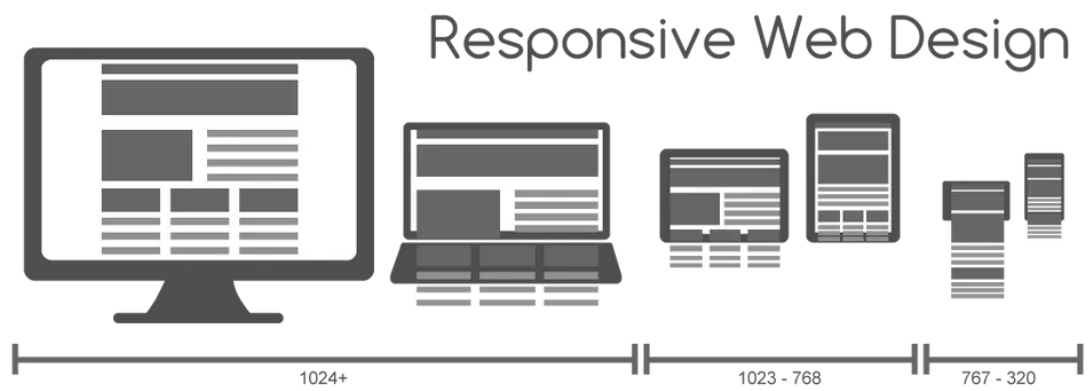


Fig.1.5.1: Responsive Web development according to different screen sizes.

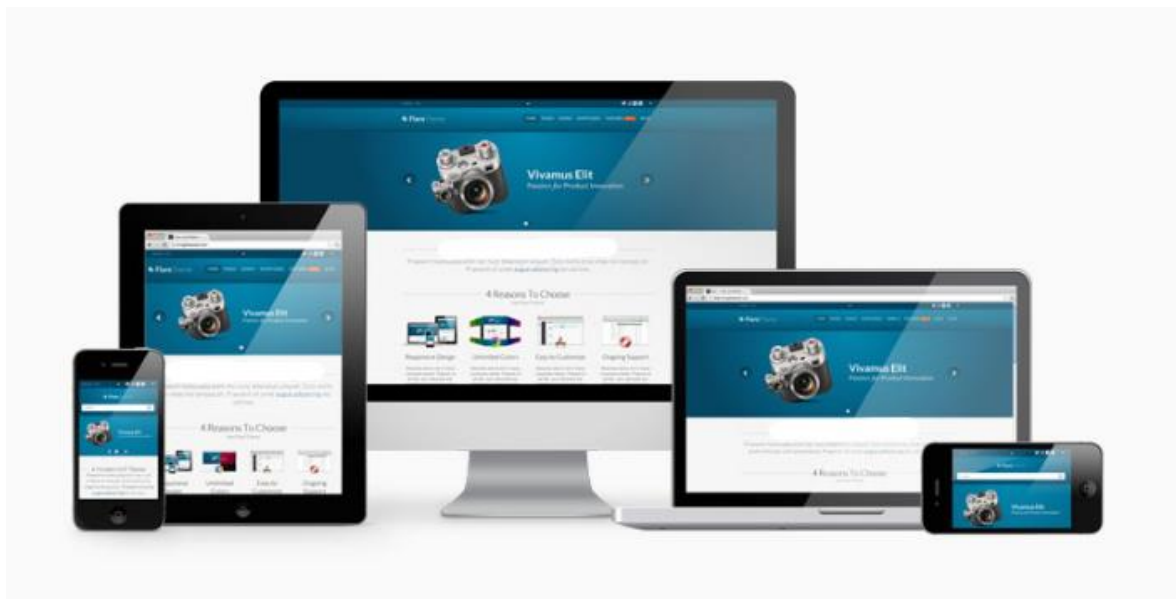


Fig.1.5.2: Example of a responsive website in different devices with different screen sizes.

1.6 Types of Web Applications.

1.6.1 Static Web Applications.

Based on HTML and CSS, static web applications contain no dynamic elements and are primarily for displaying content and data only, allowing no interactions between users and servers. Hence, these applications, the common ones being portfolio sites and official company sites, are relatively easy and simple to build, modify, and manage. Note that despite being static, these applications can include videos, GIFs, and animated banners.

1.6.2 Dynamic Web Applications.

Dynamic web applications offer interactions between the server and user, i.e., the user makes a request, which the server accepts and then generates content in real time. These applications, which often hold databases or forums and which constantly update or modify the content, usually perform updates through a content management system (CMS). You can build these applications with various web languages, but PHP and ASP are the best for structuring content.

1.6.3 Single Page Applications.

Single-page applications (SPAs) display updated content by loading a single document with a JavaScript API, e.g., XMLHttpRequest and Fetch. Consequently, since users need not load new webpages from the server, higher performance and a dynamic user experience follow. However, because SPAs are more complex than other web applications, more effort is required to monitor the performance, maintain the state, and implement navigation capabilities.

CHAPTER 2:

AIM AND SCOPE OF THE PRESENT INVESTIGATION

2.1 Aim.

Car dealer websites can make or break the dealership, and it all comes down to how feature-rich the website is. As with anything else in life, quality takes precedence over quantity here. The primary reason behind people using online car dealers before visiting an actual showroom is because of the ease of use of such a system. If a car dealer website only complicates things, then that will betray the primary objective of having the website.

2.2 Scope.

The dealerships will continue to exist and do business like they always did. What the online model provides is a point of sale but deliveries will be handled at the dealer end. You will still walk into a showroom, get a box of sweets, flowers and there will be people around to clap and celebrate as you drive out of the showroom in your new car. Plus customers still need a touch point where they can get into a car, feel it, smell it and build their first impressions. This isn't going to happen online and the dealership still remains your sole point of contact with the car you want to purchase.

Nissan India recently announced a digital business model where customers can purchase their preferred Nissans online. Considering how retail buying has penetrated the web this sounds like a fantastic idea. Several other websites exist that conduct online automotive transactions. In fact, Nissan will use some of these online portals where they will place payment gateways to transact business rather than directly through their website, though the latter will also come up shortly. There is nothing new in this and several manufacturers already employ various

transactional websites to help them generate leads. It is just the first time that a manufacturer has come out with a direct online sales point. In effect, you pay the entire amount for your Nissan online.

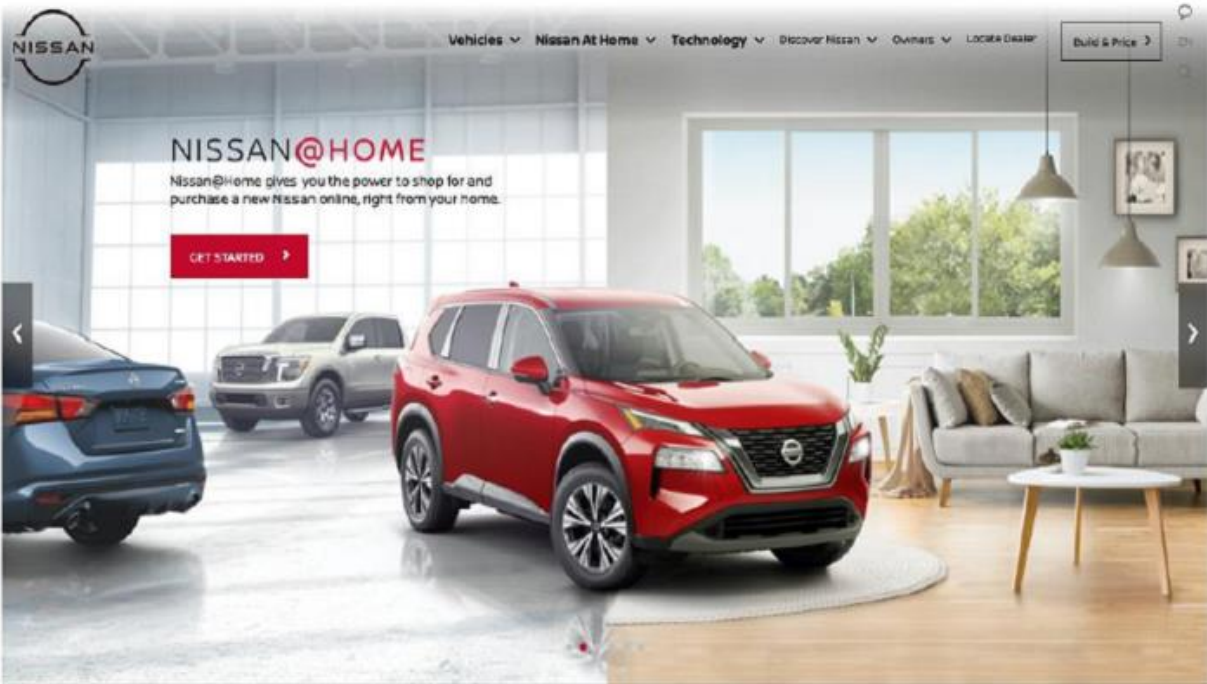


Fig. 2.2.1 Nissan India Website.

Fig. 2.2.2 Shop@home (A nissan initiative).

As of now Nissan claims that the dealerships will still earn their margins. Post purchase your nearest dealer or you can even choose which dealer you would like to work with to process your paperwork and that counts as a sale and earnings for that dealer.

2.2.2 Disadvantages.

The disadvantage I see is that you won't be able to negotiate prices like you would in a dealership. This means other than the specified discounts offered at festivals or at certain times you obviously won't be able to haggle with your computer. Having said that, Nissan claims they are offering some discount schemes and adding freebies for the online buyer as well, so not to worry there is still some scope of getting better prices. There are however further bigger unseen challenges.

Firstly, can an e-commerce venture of this sort also empower a manufacturer to control opinion? Most of the transaction portals also provide editorial opinion. Can the content then be manipulated to bend opinion towards certain brands? It's a heavy thought because most online buyers find it easy to make a purchase if the review or a rating of that product is alongside the 'buy' button. While that sort of maneuvering can apply almost anywhere, it takes on a darker tone where the seller and reviewer are one and the same entity.

The second problem I see with models like these is that most commodities are purchased online through a credit card. Now most purchases online aren't very expensive, your highest spend may not have gone beyond a lakh of rupees. In several cases your credit card supports that expense, principally because you can afford to repay those amounts. But in a nation obsessed with EMIs, where almost every car is purchased on a loan, will a credit company's high interest rates encourage online purchasing? Or will you head back to the bank to avail a loan to pay off the financial institution? It's definitely a work in progress.

The whole initiative though boils down to convenience because isn't that what Nissan is selling at the end of the day? In this regard, Nissan is trying to build a

network in the distant future that will not just accept your money but also ensure your car is delivered home to you. It is currently a work in progress but Nissan is ironing out the creases and putting systems in place that would make the operation as seamless as that at Flipkart or BookMyShow. This initiative may not entirely be foolproof at the moment and I worry there may be glitches but this initiative I sense will find lots of acceptance with a younger lot, an older generation will look at it with a lot of scepticism. Nissan is undoubtedly aiming at a wired age where all the information and opinion is available online. Yet somehow I can't help but feel Indians are not ready to let go of their dealerships just yet.

A lot of customers, especially those in India, prefer checking a product physically before buying it. Despite businesses adding all the information on the website, customers take a long time to build trust in a brand they are buying from. Even now, a huge portion of India's population enjoys going to an offline store and talking to staff while shopping. They used to consider shopping as an experience and make a whole day of it. While eCommerce allows for a simple and easy browsing experience, it also does not allow customers to have complete assurance on the brand/product. They do not get a chance to speak with a brand representative while shopping.

However, with the help of AI, online shopping is on its way to becoming the fun, interactive experience Indians would crave on the weekends with family. Social eCommerce, Government marketplace trading, augmented virtual shopping, are some of the few big changes eCommerce will see in time.

2.3 Most important features a car dealer website should have :-

2.3.1 Advanced Search:

Choosing a car is one of the most complicated ventures a person can embark upon. There are so many options and variables that a simple search option just won't cut it. Your website should have a more elaborate search option where the customers are presented with multiple options like choosing a price range and

filtering content based on a specific feature, colour, engine capacity, number of doors, luggage space, etc. This helps the customers to narrow down the search results to exactly what they want which can help them in making the right choice with minimal ease. If only a simple search option is available, then the customer will be forced to sift through hundreds or even thousands of results with most of them being useless with regards to what they are looking for. They would probably opt to close the website instead. By providing dynamic and targeted results, you can convert prospective customers into faithful long-term associations.

2.3.2 Full Vehicle Details

The best thing about cars today is that there are so many choices within a single price range. Most customers will be able to narrow their options down to a few vehicles that all have the same significant features. They will only have minute differences like a different sound system or an additional driver-aid element. One of these features will decide the customer's final choice. That is why it is paramount that your website displays the full vehicle details. An incomplete or partial description will irritate customers, and they will find it difficult to trust your dealership with their car-buying decisions.

2.3.3 High quality car photos.

Looks are often what ultimately decides if a car will be sold or not. If two vehicles with a very similar feature-set are presented to a customer, then they will choose the car that looks better to them. Your website has to make this easier for them, and that is where high-quality photos come in. Always endeavour to upload multiple images of the highest quality for each car. This will allow your customers to get the best visual impression of the vehicle in question from the comfort of their homes. If, on the other hand, the pictures on your website are blurry, skewed or does not show the car correctly then the customer will be left with more questions than answers and this doubt will drive them away from your website and your business. Just as a good salesperson will exhibit a product in the best manner possible, your site has to do the same, and high-quality photos are crucial to achieving that.

2.3.4 Easy contact with dealer.

Say your website manages to do all the hard work from attracting a prospective buyer to getting them interested in buying a car from your car dealership but then providing them with no easy way to get in touch with the dealership would render all that hard work equivalent to a big pile of nothing. Provide detailed contact information including all the phone numbers, email addresses and physical addresses of the dealership. Adding a form for the customers to fill out is also a great idea as you can get a clearer idea of what the customer is looking for. Just make sure that you follow up at the earliest.

2.3.5 Mobile Friendly.

As per the latest statistics, between 51 and 52 percent of all users who are online do so from mobile devices. This means that more than half your prospective customers will be viewing your website on a smaller screen at lower resolutions compared to a computer or a laptop. Your website and all its features should be easily accessible from devices of all screen configurations. Websites that aren't properly optimized for a smaller screen can contain overlapping elements and features that aren't visible or usable on a smaller screen. This is the worst experience you could provide to a mobile user. Thankfully, you do not have to pull your hair out trying to make your website mobile-friendly for you.

CHAPTER 3

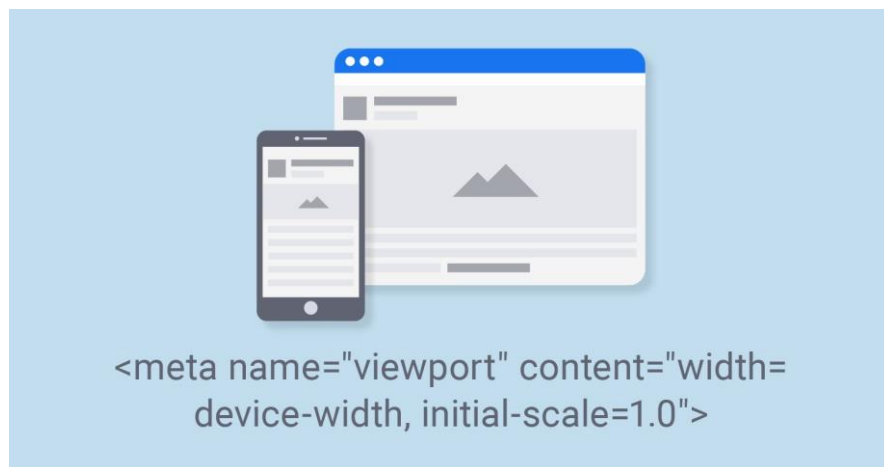
METHODS AND ALGORITHMS USED :

3.1. Technologies Used

- Html
- Css
- Vanilla JS
- Swiper JS
- Atom Editor
 - version 1.16.0
- Google Chrome
 - latest version
- Chrome Developer Tools
- PHP
- XAMPP
- MySQL
- Git and Github

3.2 HTML

3.2.1 Viewport



- The viewport is the user's visible area of a web page.

Fig 3.2.1 Viewport

- This gives the browser instructions on how to control the page's dimensions and scaling.

- Viewport in our program -

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

3.2.2 Favicon

- A favicon is a small image, so it should be a simple image with high contrast.
- A favicon image is displayed to the left of the page title in the browser tab.
- Favicon in our program -

```
<link rel="shortcut icon" type="image/png" href="image/favicon.png">
```

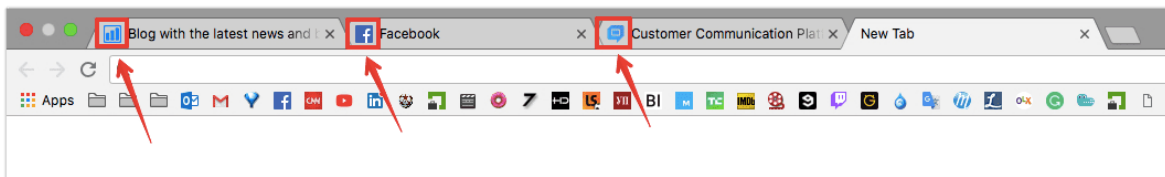


Fig 3.2.2 Favicon Example

3.2.3 Navigation

- The `<nav>` tag defines a set of navigation links.
- `<nav>` in our program -

```
<nav class="navbar">
```

```
  <a href="#home">home</a>
```

```
  <a href="#vehicles">vehicles</a>
```

```
  <a href="#services">services</a>
```

```
  <a href="#featured">featured</a>
```

```
  <a href="#reviews">reviews</a>
```

```
  <a href="#contact">contact</a>
```

```
</nav>
```

3.2.4 Button

- The <button> tag defines a clickable button.
- Inside a <button> element you can put text (and tags like <i>, , ,
, , etc.). That is not possible with a button created with the <input> element!
- Button in our program -

```
<div id="login-btn">
    <button class="btn">login</button>
    <i class="far fa-user"></i>
</div>
```

3.2.5 HTML Form

- An HTML form is used to collect user input. The user input is most often sent to a server for processing.
- The <form> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.
- The HTML <input> element is the most used form element and it has various attributes that defines the type of input data.
- Form in our program -

```
<div class="login-form-container">
    <span id="close-login-form" class="fas fa-times"></span>
    <form method="post" action="connect.php">
        <h3>user login</h3>
        <input type="email" placeholder="email" class="box" name="email">
        <input type="password" placeholder="password" class="box"
name="passwd">
        <p> forget your password <a href="#">click here</a> </p>
        <input type="submit" value="login" class="btn" name="submit">
        <p>or login with</p>
        <div class="buttons">
            <a href="#" class="btn"> google </a>
```

```

        <a href="#" class="btn"> facebook </a>
    </div>
    <p> don't have an account <a href="#">create one</a> </p>
</form>
</div>

```

Fig 3.2.5 HTML Form

```

<section class="newsletter">
    <h3>subscribe for latest updates</h3>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatum,
suscipit.</p>
    <form action="">
        <input type="email" placeholder="enter your email">
        <input type="submit" value="subscribe">
    </form>
</section>

```

```

<form action="">
    <h3>get in touch</h3>
    <input type="text" placeholder="your name" class="box">
    <input type="email" placeholder="your email" class="box">
    <input type="tel" placeholder="subject" class="box">
    <textarea placeholder="your message" class="box" cols="30"
rows="10"></textarea>
    <input type="submit" value="send message" class="btn">
</form>

```

3.2.6 iFrames

- An HTML iframe is used to display a web page within a web page.
- An inline frame is used to embed another document within the current HTML document.
- The HTML <iframe> tag specifies an inline frame.
- Iframe in our program -

```

<iframe class="map"
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d30153.78825
2261566!2d72.82321484621745!3d19.141690214227783!2m3!1f0!2f0!3f0!3m2!1i
1024!2i768!4f13.1!3m3!1m2!1s0x3be7b63aceef0c69%3A0x2aa80cf2287dfa3b!2s
Jogeshwari%20West%2C%20Mumbai%2C%20Maharashtra%20400047!5e0!3m2
!1sen!2sin!4v1632137920043!5m2!1sen!2sin" allowfullscreen=""
loading="lazy"></iframe>

```

3.2.7 SwiperJs HTML import

```

<link rel="stylesheet" href="https://unpkg.com/swiper@7/swiper-bundle.min.css" />

```

// this link tag should be used in head section of HTML file

```

<script src="https://unpkg.com/swiper@7/swiper-bundle.min.js"></script>

```

// this link tag should be used in bottom most section of HTML file

3.3 CSS

3.3.1 CSS Selectors

- CSS selectors are used to "find" (or select) the HTML elements you want to style.
- We can divide CSS selectors into five categories:
- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

3.3.2 CSS :root selector

- The <root> selector matches the document's root element.
- :root selector in our program is being used to define colour theme of the website -

```
:root{  
  --yellow:#f9d806;  
  --light-yellow:#ffee80;  
  --black:#130f40;  
  --light-color:#666;  
  --box-shadow:0 .5rem 1rem rgba(0,0,0,.1);  
  --border:.1rem solid rgba(0,0,0,.1);  
}
```

3.3.3 Universal Selector (*)

- The universal selector (*) selects all HTML elements on the page.
- * in our program -

```
* {  
  font-family: 'Poppins', sans-serif;  
  margin:0; padding:0;  
  box-sizing: border-box;  
  outline: none; border:none;  
  text-decoration: none;  
  text-transform: capitalize;  
  transition: .2s linear;  
}
```

3.3.4 Class Selector (.)

- The class selector selects HTML elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the class name.
- . in our program -

```
.btn{  
  display: inline-block;  
  margin-top: 1rem;  
  padding:.8rem 3rem;  
  background:var(--light-yellow);  
  color:var(--black);  
  cursor: pointer;  
  font-size: 1.7rem;  
  border-radius: .5rem;  
  font-weight: 500;  
  text-align: center;  
}
```

```
.heading span::before{
  content: "";
  position: absolute;
  bottom:1rem; left:0;
  height: 100%;
  width: 100%;
  background: var(--light-yellow);
  z-index: -1;
  clip-path: polygon(0 90%, 100% 80%, 100% 100%, 0% 100%);
}
```

3.3.5 CSS Element Selector

- The element selector selects HTML elements based on the element name.
- Element Selector in our program -

```
html {
  font-size: 62.5%;
  overflow-x: hidden;
  scroll-padding-top: 7rem;
  scroll-behavior: smooth;
}
```

```
section {
  padding:2rem 9%;
}
```

3.3.6 CSS Id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id selector is used to select one unique element!

- To select an element with a specific id, write a hash (#) character, followed by the id of the element.
- ID selector in our program -

```
#login-btn i{
    display: none;
    font-size: 2.5rem;
    color:var(--light-color);
}
```

```
#menu-btn{
    font-size: 2.5rem;
    color:var(--light-color);
    display: none;
}
```

3.3.7 CSS Media Queries

- Using media queries are a popular technique for delivering a tailored style sheet to desktops, laptops, tablets, and mobile phones (such as iPhone and Android phones).
- Media queries can be used to check many things, such as:
 - width and height of the viewport
 - width and height of the device
 - orientation (is the tablet/phone in landscape or portrait mode?)
 - resolution
- Media queries in our program -

```
@media (max-width:991px){
    html{
        font-size: 55%;
    }
    .header{
        padding:2rem;
    }
}
```

```

.header.active{
  padding:2rem;
}
section{
  padding:2rem;
}
}

```

```

@media (max-width:768px){
  #menu-btn{
    display: block;
  }
  #menu-btn.fa-times{
    transform:rotate(180deg);
  }
  #login-btn .btn{
    display: none;
  }
  #login-btn i{
    display: block;
  }
  .header .navbar{
    position: absolute;
    top:99%; left:0; right:0;
    background: #fff;
    border-top: var(--border);
    clip-path: polygon(0 0, 100% 0, 100% 0, 0 0);
  }
  .header .navbar.active{
    clip-path: polygon(0 0, 100% 0, 100% 100%, 0% 100%);
  }
  .header .navbar a{
    margin:2rem;
  }
}

```

```
display: block;
font-size: 2rem;
}
}
```

```
@media (max-width:450px){
  html{
    font-size: 50%;
  }
  .heading{
    font-size: 3rem;
  }
}
```

3.4 Javascript

3.4.1 HTML DOM

- When an HTML document is loaded into a web browser, it becomes a document object.
- The document object is the root node of the HTML document.
- The document object is a property of the window object.
- The document object is accessed with:
window.document or just document.

3.4.1.1 HTML DOM `querySelector()`

- The `querySelector()` method returns the first element that matches a CSS selector.
- To return all matches (not only the first), use the `querySelectorAll()` instead.
- `querySelector()` in our program -

```
let menu = document.querySelector('#menu-btn');
let navbar = document.querySelector('.navbar');
```

```
document.querySelector('#login-btn').onclick = () =>{  
  document.querySelector('.login-form-container').classList.toggle('active');  
}
```

```
document.querySelector('#close-login-form').onclick = () =>{  
  document.querySelector('.login-form-container').classList.remove('active');  
}
```

3.5 Swiper Js

- Swiper is a powerful and modular javascript library to implement responsive, accessible, flexible, touch-enabled carousels/sliders on your mobile websites and apps.
- Features :-
 - Responsive design
 - Small and fast
 - Fully accessible
 - Supports any html elements, not only images
 - Supports Vertical/Horizontal animations and swipes
 - Cool transition effects: Fade, Flip, 3D, Cube, 3D Coverflow, Parallax
 - Hash navigation
 - Grid layout
 - Image lazy load
 - Virtual Slides for better performance
 - Keyboard interactions
 - RTL layout
 - Works on iOS, Android and latest Desktop browsers
 - Available for Angular, React, Vue, and Svelte.
- Swiper JS in our program -

```

var swiper = new Swiper(".vehicles-slider", {
  grabCursor: true,
  centeredSlides: true,
  spaceBetween: 20,
  loop:true,
  autoplay: {
    delay: 9500,
    disableOnInteraction: false,
  },
  pagination: {
    el: ".swiper-pagination",
    clickable:true,
  },
  breakpoints: {
    0: {
      slidesPerView: 1,
    },
    768: {
      slidesPerView: 2,
    },
    1024: {
      slidesPerView: 3,
    },
  },
});

```

```

var swiper = new Swiper(".featured-slider", {
  grabCursor: true,
  centeredSlides: true,
  spaceBetween: 20,
  loop:true,
  autoplay: {
    delay: 9500,
    disableOnInteraction: false,

```

```

    },
    pagination: {
      el: ".swiper-pagination",
      clickable: true,
    },
    breakpoints: {
      0: {
        slidesPerView: 1,
      },
      768: {
        slidesPerView: 2,
      },
      1024: {
        slidesPerView: 3,
      },
    },
  });

```

```

var swiper = new Swiper(".review-slider", {
  grabCursor: true,
  centeredSlides: true,
  spaceBetween: 20,
  loop: true,
  autoplay: {
    delay: 9500,
    disableOnInteraction: false,
  },
  pagination: {
    el: ".swiper-pagination",
    clickable: true,
  },
  breakpoints: {
    0: {
      slidesPerView: 1,

```

```

    },
    768: {
        slidesPerView: 2,
    },
    1024: {
        slidesPerView: 3,
    },
    },
    });

```

3.6 PHP

- PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.
 - PHP files can contain text, HTML, CSS, JavaScript, and PHP code
 - PHP code is executed on the server, and the result is returned to the browser as plain HTML
 - PHP files have extension “.php”
- What Can PHP Do?
 - PHP can generate dynamic page content
 - PHP can create, open, read, write, delete, and close files on the server
 - PHP can collect form data
 - PHP can send and receive cookies
 - PHP can add, delete, modify data in your database
 - PHP can be used to control user-access
 - PHP can encrypt data
- PHP in our program -

```

<?php
$server = 'localhost';
$user = 'root';
$pass = '';
$db = 'carswebsite';
$conn = new mysqli($server,$user,$pass,$db);

```

```

if ($conn->connect_error){
    die("connection failed: " . $conn->connect_error);
}
/*$sql = "CREATE TABLE logindetails(email varchar(30) NOT NULL,passwd
varchar(30) NOT NULL)";
if ($conn->query($sql) == TRUE) {
    echo "Table LoginDetails Created";
}
else {
    echo "Error Creating Tabel: " . $conn->error;
}*/
$Fname = $_POST["name"];
$email = $_POST["email"];
$sub = $_POST["subject"];
$msg = $_POST["msg"];
$sql = "INSERT INTO git(Fname,email,sub,msg)
values('$Fname','$email','$sub','$msg')";
if($conn->query($sql) == TRUE){
    echo "query submitted";
}
else{
    echo "ERROR: " . $conn->error;
}
$conn->close();
?>

```

```

<?php
$server = 'localhost';
$user = 'root';
$pass = "";
$db = 'carswebsite';
$conn = new mysqli($server,$user,$pass,$db);
if ($conn->connect_error){
    die("connection failed: " . $conn->connect_error);
}

```

```

}
/*$sql = "CREATE TABLE logindetails(email varchar(30) NOT NULL,passwd
varchar(30) NOT NULL)";
if ($conn->query($sql) == TRUE) {
    echo "Table LoginDetails Created";
}
else {
    echo "Error Creating Tabel: " . $conn->error;
}*/
$email = $_POST["email"];
$password = $_POST["password"];
$sql = "INSERT INTO LoginDetails(email,password) values('$email','$password')";
if($conn->query($sql) == TRUE){
    echo "NEW RECORD ADDED";
}
else{
    echo "ERROR: " . $conn->error;
}
$conn->close();
?>

```

3.7 XAMPP

- XAMPP is the most popular PHP development environment.
- XAMPP is a completely free, easy to install Apache distribution containing.
- MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.
- XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language.

•

CHAPTER 4 & CHAPTER 5

DISCUSSION, RESULTS AND PERFORMANCE ANALYSIS

4.1 Requirements :

Minimum Hardware Requirements:

1. CPU: 2 cores
2. RAM: 2 GB
3. Storage: 1 GB available space
4. Processor: 1.0 GHz Intel Pentium Processor.
5. 1 mbps uninterrupted Internet Connection

Minimum Software Requirements:

1. Operating System : Windows-xp/vista/7/8/10/Linux/Mac
2. **NET 4.5 Framework required**
3. VS CODE
4. Microsoft Dev tools for html and css
5. Chrome Developer Tools
6. Latest version of any browser.

4.2 RESULTS AND PERFORMANCE ANALYSIS:

- Title with favicon
- Responsive header / navbar with toggle navbar menu and login form effect using vanilla javascript.
- Responsive home / hero section with parallax hover effect using vanilla javascript
- Responsive vehicles touch mobile slider section using swiper.js
- Responsive services section using css grid.
- Responsive featured car section touch mobile slider using swiper.js
- Responsive newsletter.
- Responsive review / testimonial touch mobile slider using swiper.js
- Responsive contact section using css flexbox.
- Responsive footer section using css grid.
- Database for login page.
- Database for Get in touch feature.

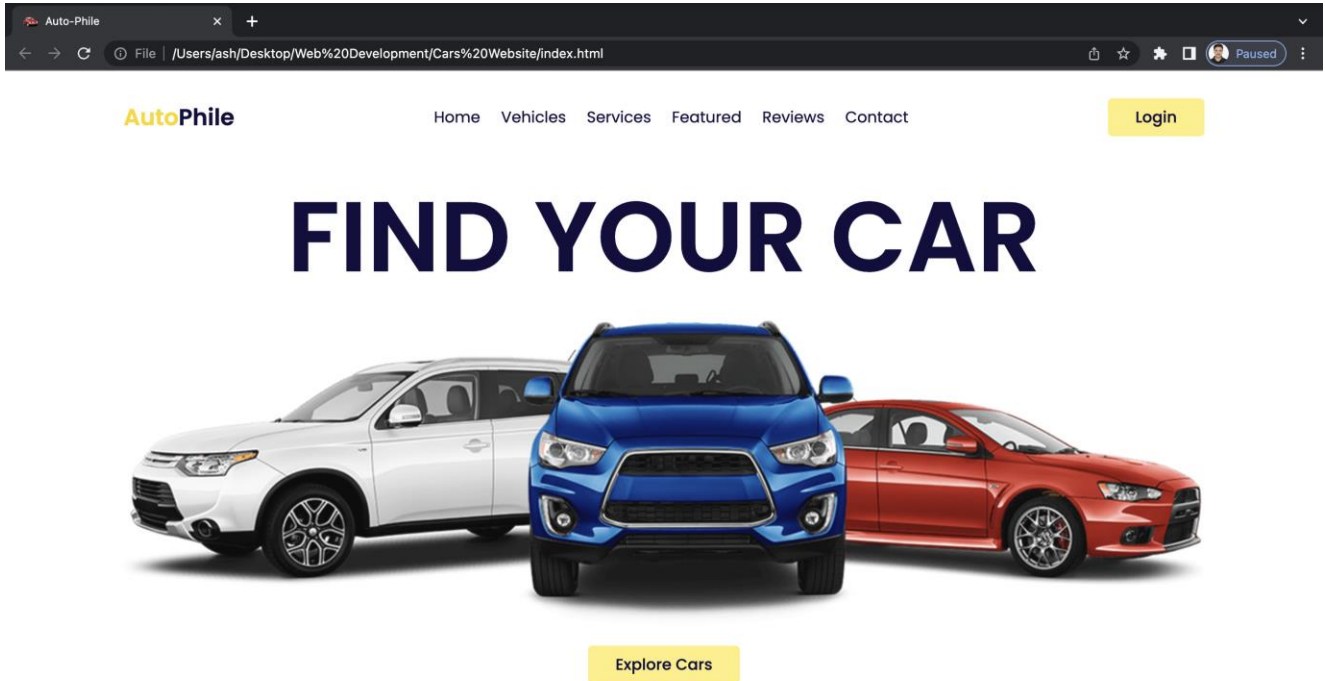
5.1 SUMMARY AND CONCLUSION:

Our project is only a humble venture to satisfy the needs to manage car selling dealership. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the project. The objective of software planning is to provide a frame work that enables the user to make reasonable estimates made within a limited time frame at the beginning of the use and should be updated regularly as the project progresses.

- A description of the background and context of the project and its relation to work already done in the area.
- Made statement of the aims and objectives of the project.
- The description of Purpose, Scope, and applicability.
- We define the problem on which we are working in the project.
- We describe the requirement Specifications of the system and the actions that can be done on these things.
- We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
- We included features and operations in detail, including screen layouts.
- We designed user interface and security issues related to system.
- Finally the system is implemented and tested according to test cases.

APPENDIX

A. Screenshots of Application.



Our Services



Car Selling

Lorem, Ipsum Dolor Sit Amet Consectetur
Adipiscing Elit. Corporis, Nisi.

[Read More](#)



Parts Repair

Lorem, Ipsum Dolor Sit Amet Consectetur
Adipiscing Elit. Corporis, Nisi.

[Read More](#)



Car Insurance

Lorem, Ipsum Dolor Sit Amet Consectetur
Adipiscing Elit. Corporis, Nisi.

[Read More](#)



Battery Replacement

Lorem, Ipsum Dolor Sit Amet Consectetur
Adipiscing Elit. Corporis, Nisi.

[Read More](#)



Oil Change

Lorem, Ipsum Dolor Sit Amet Consectetur
Adipiscing Elit. Corporis, Nisi.

[Read More](#)



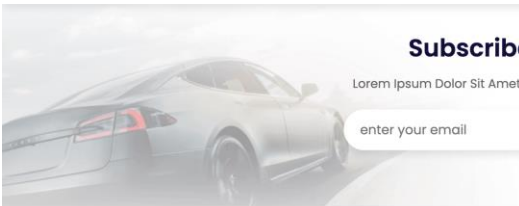
24/7 Support

Lorem, Ipsum Dolor Sit Amet Consectetur
Adipiscing Elit. Corporis, Nisi.

[Read More](#)

New • 2021 • Automatic • Petrol • 183mph

[Check Out](#)



Subscribe For Latest Updates

Lorem Ipsum Dolor Sit Amet Consectetur Adipiscing Elit. Voluptatum, Suscipit.

[Subscribe](#)

Client's Review



Lorem Ipsum Dolor Sit Amet Consectetur,
Adipiscing Elit. Ipsam Incidunt Quod
Praesentium Iusto Id Autem Possimus
Assumenda At Ut Saepe.

John Deo
★★★★★



Lorem Ipsum Dolor Sit Amet Consectetur,
Adipiscing Elit. Ipsam Incidunt Quod
Praesentium Iusto Id Autem Possimus
Assumenda At Ut Saepe.

John Deo
★★★★★

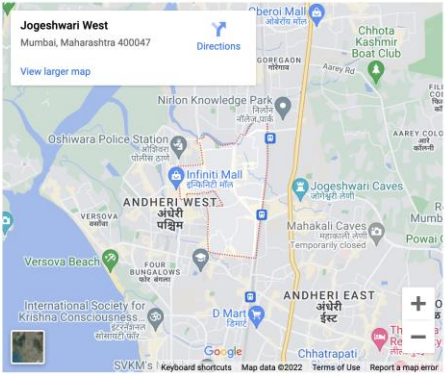


Lorem Ipsum Dolor Sit Amet Consectetur,
Adipiscing Elit. Ipsam Incidunt Quod
Praesentium Iusto Id Autem Possimus
Assumenda At Ut Saepe.

John Deo
★★★★★



Contact Us



Jogeshwari West
Mumbai, Maharashtra 400047

View larger map

Get In Touch

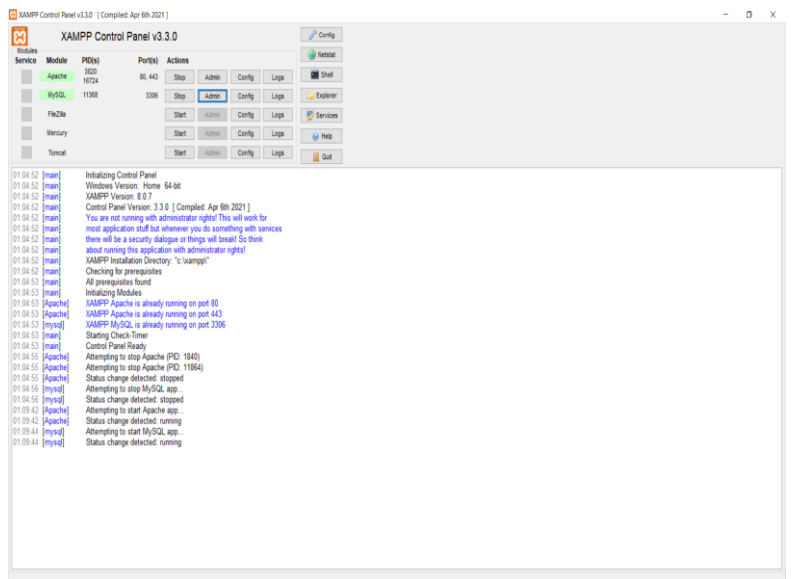
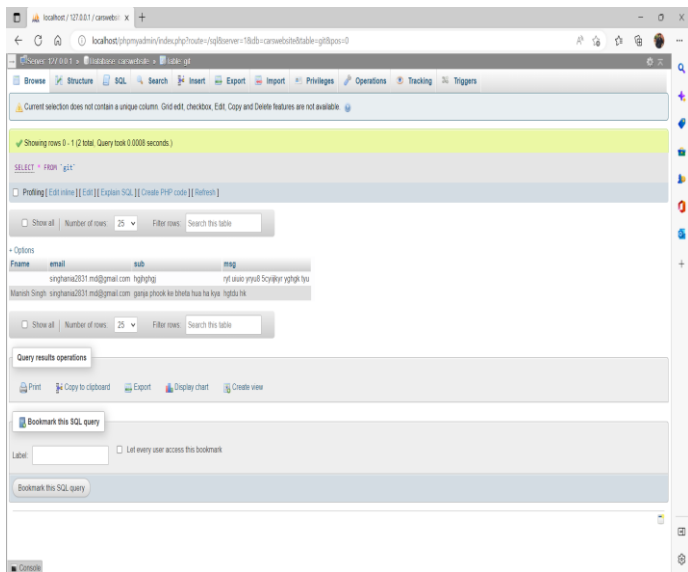
your name

your email

subject

your message

Send Message



REFERENCES:

- [FreeCodeCamp.org](https://www.freecodecamp.org/) - Legacy Responsive Web Design Curriculum
- [w3schools.com](https://www.w3schools.com/)
- Udemy - The Complete 2022 Web Development Bootcamp by Angela Yu
- MDN Web Docs by Mozilla - <https://developer.mozilla.org/en-US/>
- Nissan India Website - <https://www.nissan.in>