PROJECT-1 REPORT

ON

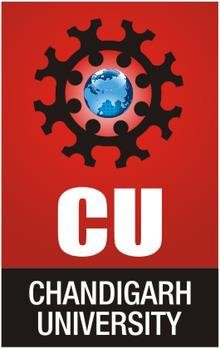
**“TRUSTED DRIVE”**

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

**BACHELOR OF ENGINEERING**

**IN**

(Computer Science and Engineering)



JAN-JUNE, 2022

**SUBMITTED BY:**

PIYUSH RAJ (20BCS2445)

VAISHALI (20BCS2431)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CHANDIGARH UNIVERSITY GHARUAN, MOHALI

**CHANDIGARH UNIVERSITY,GHARUAN,MOHALI**

**BONAFIDE CERTIFICATE**

Certified that this project report “TRUSTED DRIVE” is the bonafide work of “PIYUSH RAJ (20BCS2445) AND VAISHALI (20BCS2431)” who carried out the project work under the supervision of our Supervisor ER. DEEPIKA (E11809) and Co-Supervisor ER. SUGANDHI MIDHA (E8219). This project is submitted for the partial fulfilment of the requirement for the degree **of “Bachelor of Engineering in Computer Science and Engineering”** discipline in “Chandigarh University” During the academic session JAN-JUNE 2022 is a record of bona fide piece of work, carried out by students under proper supervision and guidance in the **“Department of Computer Science and Engineering”**, Chandigarh University.

Project Supervisor Project Co-Supervisor

ER. DEEPIKA (E11809) ER. SUGANDHI MIDHA (E8219)

Submitted for the project viva-voce examination held on 19th May, 2022

INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

We would like to express a deep sense of gratitude and thanks to **Er. DEEPIKA** and **ER. SUGANDHI MIDHA** who assupervisorand co- supervisorprovided wise counciland able guidance. They also provided us unfailing support and constant guidance through all stages of our project work. It was a great privilege for us to study and work under their guidance.

We owe the completion of our project to our project Mentors for their continuous support and guidance.

Students Name

Piyush Raj -20BCS2445

Vaishali -20BCS2431

`

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TOPIC** | **PAGE NO.** |
| 1 | List of figures | 5 |
| 2 | Abstract | 6 |
| 3 | Definition, Acronyms and Abbreviation | 7 |
| 4 | CHAPTER-1 Introduction | 8 to 9 |
| 5 | CHAPTER-2 Literature Survey | 10 to 12 |
| 6 | CHAPTER-3 Design flow / Process | 13 to 18 |
| 7 | CHAPTER-4 Result Analysis and Validation | 19 to 27 |
| 8 | CHAPTER-5 Conclusion and Future Work | 28 |
| 9 | References | 29 |
| 10 | User Manual | 30 to 35 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **FIGURE NAME** | **PAGE NO.** |
| 1 | ER DIAGRAM | 20 |
| 2 | USE CASE DIAGRAM | 21 |

**ABSTRACT**

We live in a country where the count of total vehicles on road is around 295.8 million. Out of the owners and people who are driving these vehicles, very few are aware about techniques required while dealing with the emergencies caused by them. This lag in knowledge leads to a lot of tension, wastages of time, expensive bills, and sometimes also becomes a matter of huge security issue.

Being a student of Computer Science, we have decided to solve this problem by providing users a website and an android application. The name of our project is **“Trusted Drive”.**

Our project is designed for everyone who are having vehicles or using their personal vehicles as daily means of transportation, and still, they are not aware about the methods of dealing with basic emergencies like Tire blowout, Engine overheating, Low fuel milage etc. For all such emergencies we have decided to provide one special feature named as “Emergency Feature” to our users under which, we will be providing a technical team for their vehicles in the least possible time. This project will also increase the employment opportunities for mechanics, as we will develop our own team of mechanics for Emergency situation.

**DEFINITIONS, ACRONYMS AND ABBREVIATION**

1. **HTML TAGS**

HTML tags are like keywords which defines that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content.

1. **HTML ATTRIBUTE**

HTML ATTRIBUTES provide additional information about HTML element. All the elements can have attribute. They are always specified in the start tag and usually come in name/value pair like: name=”value”.

1. **GOOGLE MAP API**

The Google Maps API is one of those clever bits of Google technology that helps you take the power of Google Maps and put it directly on your own site. It lets you add relevant content that is useful to your visitors and customise the look and feel of the map to fit with the style of your site.

1. **MODULES IN NODE.JS**

Module in Node.js is a simple or complex functionality organized in single or multiple JavaScript files which can be reused throughout the Node.js application.

Each module in Node.js has its own context, so it cannot interfere with other modules or pollute global scope.

1. **CODE IDE**

An integrated development environment (IDE) is a software suite that consolidates basic tools required to write and test software. Developers use numerous tools throughout software code creation, building and testing. Development tools often includes text editors, code libraries, compilers, and test platforms, Without an IDE, a developer must select, deploy, integrate and manage all these tools separately. An IDE brings many of those development-related tools together as a single framework, application or service.

**CHAPTER-1 INTRODUCTION**

* 1. **AIM**

Our project is designed for everyone who are having vehicles or using their personal vehicle as daily means of transportation, and still, they are not aware about the methods of dealing with basic emergencies like Tire blowout, Engine overheating, Low fuel mileage etc. With the help of our project, we will be providing a technical team for their vehicle in the least possible time.

* 1. **CONTEMPORARY ISSUES**

We live in a country where the count of total vehicles on road is 295.8 million. Out of the owners and users of these vehicles, very few people are having the knowledge about dealing with the emergencies caused while driving them. This causes a lot of tension, loss of time and also becomes a matter of serious safety issue. So, through our project we want to build a team of mechanics who will be available 24/7 for people facing any mechanical emergences or need a regular service for their vehicles.

* 1. **CLIETS/CUSTOMERS**

1. People who prefer personal vehicles as daily means of transportation.
2. Drivers who are not having enough knowledge about vehicles which they are driving.
3. People who have to drive at late nights and faces some sort of mechanical emergencies.
4. Students who use two-wheeler for going to educational institutes or any other purpose and are not aware that how to deal with vehicles if something goes wrong.
5. People who are still driving old versions of cars and often faces problems related to vehicles.
6. Mechanical engineers and other garage owners who are not able to get enough customers due to some reason but are willing to work. So, we will provide them customers when they will be part of our organization.
   1. **BENEFITS**
7. It will help people by providing them a perfect mechanic for their vehicle based upon its model.
8. It will increase safety for people driving at nights.
9. It will increase employment for mechanical engineers.
10. It will save people time as they will not have to ask every mechanic that if they can fix this or not.
    1. **TASK IDENTIFICATION**

TEAM LEADER -PIYUSH RAJ

**CHAPTER-2 LITERATURE SURVEY**

* 1. **PROBLEM DEFINITION**

Driving daily is not an easy thing to do, before and after having a 9 to 5 shift which sometimes also includes over-time. Many people get trapped in a situation where their vehicle stops working due to one or the other reason. This effects the mental health of the person as dealing with these situations is not as simple as it seems to. So, we have decided to provide a platform where people can get their vehicle repaired in the least possible time.

* 1. **GOAL**

The goal which we are going to fulfil through this project is to help people to deal with emergencies related to their vehicles.

Moreover, our plan is also to increase the employment opportunities for mechanics, as we have decided to develop our own team of mechanics for Emergency situations.

* 1. **OBJECTIVES**

The main objective of our project is to develop a web and android application for people so that they can book appointments with mechanics during emergencies or normal working day in order to get their vehicles repaired or serviced in the least possible time.

* 1. **PROPOSED SOLUTION**

We have decided to provide a web application and with the help of our website and application people can make an account and can book mechanic for the services of their vehicles. We are also having account for mechanics as we are going to appoint mechanics for our application and build a network of them. At the times of emergencies people can directly book a mechanic by just clicking on the “EMERGENCY” button. As soon as someone will click on the emergency button, we will fetch their location and the mechanic who is closest to them will be appointed to them plus the details of the mechanic will also be shared with the user.

* 1. **CONSTRAINTS IDENTIFICATION**

There are four major constraints in project management to consider :

1. TIME

Both the members of our team are giving their best in order to finish this project as soon as possible under the supervision of Er. Sugandhi Midha mam. Team members are very cooperative and ready to help each other in every possible way.

1. COST

No amount of money is required so far for our project. Maybe we will require funding after sometime when our project is ready to be published in public. Also, money will be required while hiring mechanics in the very first phase.

1. QUALITY

The quality of our project is decided by its accuracy to fetch the location of users during the times of emergencies and also by its working while allocating the mechanic during normal booking.

1. RISK

We are trying our level best to build a web and android application which is bug less and easy to use so that users don’t face any sort of trouble while using it. It is little difficult for us to make android application as this is new for us but we will try to make a user-friendly application.

* 1. **LITERATURE REVIEW**

Many of us have been there - a warning light starts to blink and before you know it, your car is rumbling to a halt. Maybe you've got behind the wheel only for the ignition to refuse to turn over. Regardless of the scenario, breakdowns are one of the less enjoyable aspects of car ownership and what's more, they're becoming increasingly common.

As per the latest survey around 25 cars suffers from breakdown every hour. These breakdowns are becoming number one reason for the irritation and tension in our workforces.

So, we have decided to help people of all the age group by ensuring that their vehicles get services on time and if something goes wrong while driving then they get a team of mechanics in the least possible time. For this we have decided to make website and android application.

1. SINGLE ENTITY

As studies are getting very advanced and the literacy rate of people is increasing on a very high scale so the single entity goal is to encourage people to come and make use of their skills in building something useful and productive for the society.

1. LIFE SPAN

Before starting our work on this project, we noted all the parts on which we needed to work and we divided all the work in two parts as are a team of two person in this project. Each one of us needed at least 4 weeks to complete our allocated task. Total time required to complete this project is estimated as 2 months which includes learning also.

1. REQUIRED FUNDS

No amount of money is required so far for our project. Maybe we will require funding after sometime when our project is ready to be published in public. Also, money will be required while hiring mechanics in the very first phase.

1. LIFE CYCLE

We have started our project and we have reached the implementation part of it. Like every other project our project also has a life cycle with different stages like start, growth, maturity, and decay.

1. TEAM SPIRIT

Both the members of our team are giving their best in order to finish this project as soon as possible under the supervision of Er. Sugandhi Midha mam. Team members are very cooperative and ready to help each other in every possible way.

1. RISK AND UNCERTANITY

We are trying our level best to build a web and android application which is bug less and easy to use so that users don’t face any sort of trouble while using it. It is little difficult for us to make android application as this is new for us but we will try to make a user-friendly application.

1. DIRECTION

Project is always performed according to the directions given by the customers with regard to time, quality and quantity, etc. In this case the directions are our motivation is to help and design the best possible solution for people to deal with emergencies caused by vehicles.

1. UNIQUENESS

Our project is built on our own. Each and every part of this project was discussed and developed by us idea and hence it is very unique. Moreover, there are very less platforms which provides services which we have planned to provide through our project.

1. FLEXIBILITY

The project is flexible to new changes required according to the need of the situation as the code of both android and web application is quite easy to merge and understand.

1. SUB-CONTRACTING

Sub-contracting is a subset of every project and without which no project can be completed unless it is a proprietary firm or tiny in nature. The more complexity of a project the more will be the extent of contracting. Every project needs the help of an outsider consultant, engineer, or expert in that field.

**CHAPTER-3 DESIGN FLOW / PROCESS**

* 1. **TECHNOLOGY USED**
     1. **FRONT-END**

Entire work which was related to front-end of our website is done using three languages HTML,CSS and JAVASCRIPT.

Following is the description of languages used and their most commonly used elements, tags, attributes, and other features used by us throughout the making of our website:

1. **HTML**

The HyperText Markup Language is the standard markup languages for documents designed to be displayed in a web browser. 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.

FEATURES USED:

1. HTML heading tag
2. HTML body tag
3. HTML paragraph tag
4. HTML image tag
5. HTML lists
6. HTML Quotations
7. HTML class
8. HTML ID
9. HTML forms
10. HTML input types
11. The href attribute
12. The alt attribute.
13. **CSS**

Cascading Style Sheet is a style sheet language used for describing the presentation of a document written in a markup language such as 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, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification od presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate.

FEATURES USED:

1. CSS Selectors
2. CSS Colours
3. CSS Background
4. CSS Margin
5. CSS Padding
6. CSS Height/ Width
7. CSS Icons
8. CSS forms
9. CSS Rounded Corners
10. CSS Background
11. CSS Buttons
12. RWD Grid View
13. **JAVASCRIPT**

JavaScript is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time complied, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class function. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. Over 97% of website use it client-side for web page behaviour, often incorporating third-party libraries. All major web browser has a dedicated JavaScript engine to execute the code on the user’s device.

FEATURES USED:

1. JS Document object model
2. JS Conditional statement
3. JS Data type
4. JS Function
5. JS Array
6. JS Object
7. JS setTimeOut Event listener
   * 1. **BACK-END**

Entire work which was related to back-end of our website is done using two languages MySQL and NodeJS.

Following are the description of languages used and their few modules with some other features of these two languages which we used in our website:

1. **MONGO DB**

**MongoDB** is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB. Collections contain sets of documents and function which is the equivalent of relational database tables. MongoDB is a database which came into light around the mid-2000s.

1. **NODE JS**

Node.js (Node) is an open-source development platform for execute Java Script code server-side. Node is useful for developing applications that require a persistent connection from the Browser to the server and is often used for real time application such as chat, news feeds and web push notifications.

Node.js is intended to run on a dedicated HTTP server and to employ a single thread with one process at a time. Node.js applications are event based and run asynchronously. Code built on the Node platform does not follow the traditional model of receive, process, send, wait, receive. Instead, Node processes incoming requests in a constant event stack and sends small requests one after the other without waiting for responses.

**MODULES USED OF NODE JS**

1. **BCRYTJS MODULE**

It represents a one-way transfer of data from user to the employee who is handling the data. It was used for the security purpose of the user as it hashes the password.

1. **COOKIE-PARSER**

It is used to update the cookies on our browser.

1. **DOTENV MODULE**

It helps us by securing us before we enter any sensitive data in our code.

1. **EXPRESS JS MODULE**

It is one of the libraries of node.js but once we import it than it is termed as module. This is used to make the back-end coding easy and create the server for websites.

1. **HBS MODULE**

It helps us to enter the data in our files dynamically by creating and using a variable.

1. **JSONWEBTOKEN**

The JSON Web Token (JWT) Authentication module provides a Drupal authentication provider that uses JWTs as the primary factor of authentication.

1. **MONGOOSE**

Mongoose is a JavaScript framework that is commonly used in a Node.js application with a MongoDB database.

Mongoose is an Object Document Mapper (ODM). This means that Mongoose allows you to define objects with a strongly-typed schema that is mapped to a MongoDB document.

1. **VALIDATOR**

This library validates and sanitizes strings only. Passing anything other than a string will result in an error.

* + 1. **GOOGLE MAP API**

The Google Maps API is one of those clever bits of Google technology that helps you take the power of Google Maps and put it directly on your own site. It lets you add relevant content that is useful to your visitors and customise the look and feel of the map to fit with the style of your site.

1. **JAVASCRIPT API**

The Maps JavaScript API lets you customize maps with your own content and imagery for display on web pages and mobile devices. The Maps JavaScript API features four basic map types (roadmap, satellite, hybrid, and terrain) which you can modify using layers and styles, controls and events, and various services and libraries.

1. **PLACE API**

The Places API lets you search for place information using a variety of categories, including establishments, prominent points of interest, and geographic locations. You can search for places either by proximity or a text string. A Place Search returns a list of places along with summary information about each place; additional information is available via a Place Details query.

1. **ROUTE API**

You can draw a route on Google Maps that includes a driving, walking, or cycling route and save it or share it with others. To create a route, open "Your places" in the menu on Google Maps and then choose "Create Map" on the Maps tab.

1. **DIRECTION API**

You can calculate directions (using a variety of methods of transportation) by using the Directions Service object. This object communicates with the Google Maps API Directions Service which receives direction requests and returns an efficient path. Travel time is the primary factor which is optimized, but other factors such as distance, number of turns and many more may be taken into account. You may either handle these directions results yourself or use the Directions Renderer object to render these results.

* + 1. **CODE IDE**

**VISUAL STUDIO CODE**

The Visual Studio integrated development environment is a creative launching pad that you can use to edit, debug, and build code, and then publish an app. An integrated development environment(IDE) is a feature-rich program that can be used for many aspects of software development. Over and above the standard editors and debugger that most IDEs provide, Visual Studio include complies, code completion tools, graphical designers, and many more features to ease the software development process.

\

* + 1. **OTHER SOFTWARE**

1. **POSTMAN**

Postman is one of the most popular software testing tools which is used for API testing. With the help of this tool, developers can easily create, test, share, and document APIs.

This tutorial will help in understanding why Postman is so famous and what makes it unique when compared to other API testing tools. All the examples in this tutorial are tested and can be imported in Postman.

1. **COMPASS**

MongoDB Compass is a powerful GUI for querying, aggregating, and analysing your MongoDB data in a visual environment.

Compass is free to use and source available, and can be run on macOS, Windows, and Linux.

1. **GITHUB**

GitHub is one of the world’s largest community of developers. It’s an intricate platform that foster collaboration and communication between developers. GitHub has a number of useful features that enable development teams to work together on the same project and easily create new versions of software without disrupting the current versions, but it doesn’t stop there.

* 1. **DESIGN SELECTION**

Implementing a design system for every web application starts long before developers open a code editor. The end result is directly related to the quality of collaboration between designers and developers during creation, build-out, and testing. Great design systems aren’t just passed from one team to another, they’re carefully crafted together.

These design systems give teams greater consistency, maintainability, and efficiency for building software. This also rings true for component-based architectures, which allow code to be written in self-contained blocks that connect to form an application.

Though distinct entities, these two pieces are interconnected: two parts in a larger process. We’ll look at how to cross the gap from design system to software components.

This process guide focuses on building component systems destined for a single application, but the principles remain for multi-app usage. The latter case may entail multiple frameworks, require housing the component system as a standalone, and impose additional naming constraints — but that’s a different guide for another day.

* + 1. **Paving the way for a successful design system implementation**

Once the design system is formed, it’s tempting to start building its corresponding components right away. Unless you have unlimited time and budget, jumping code-first is usually a mistake. Following a few steps at the onset can significantly upgrade the implementation process.

**Step 1: Estimate individual builds**

Before the first build, it’s necessary to determine the importance and complexity of components as they relate to the overall project. Gathering this context at the onset will prevent unnecessary rewrites later. Simple per-component estimates work alongside priorities (see the next step) to provide an overview of what’s on your plate.

**Step 2: Prioritize components**

The next step in a good implementation checklist is setting priorities. One best practice is to build the most basic components — such as buttons, colours, headings, icons — before moving to more involved components. Tackling other global elements like navigation and page layouts early can also be a good idea, laying the ground work before filling it in.

**Step 3: Utilize a component-driven development environment**

Once the design system is in place and you’ve finished the previous steps, you now know what you’re building and how you plan to go about it. It’s now time to ask: Where will the development take place? There are a lot of possibilities, but the advent of component-driven development environments has provided an excellent answer to this question.

* + 1. **The real-world advantages of design systems**

How does all this planning, setup, and conversation play out in the real world? For us, developing a robust design system for the Moonlight fantasy soccer web app yielded big benefits during implementation.

Not only did it help the dev team create a consistent and maintainable application, the testing and hand-off points were a success: we saw collaboration improvements and faster-to-production changes. Simply put, both the application and process behind it were better.

**CHAPTER-4 RESULT ANALYSIS AND VALIDATION**

* 1. **FEATURES**

1. **EMERGENCY BOOKING**

If due to one or the other reason the vehicle of the user stops working then they can use our emergency feature which promises to appoint and send the mechanic at their location in the east possible time.

1. **REAL TIME TRACKING OF MECHANIC DURING EMERGENCIES**

When user will make an appointment during the emergency situation using our emergency feature, at that time they will be able to see the live location of the mechanic which will be appointed to them.

1. **EDITABLE PROFILE**

Regular user can make their profile using our application and this profile can be edited by them at any time.

1. **KNOW YOUR MECHANIC BY THE REVIEW OF OTHER**

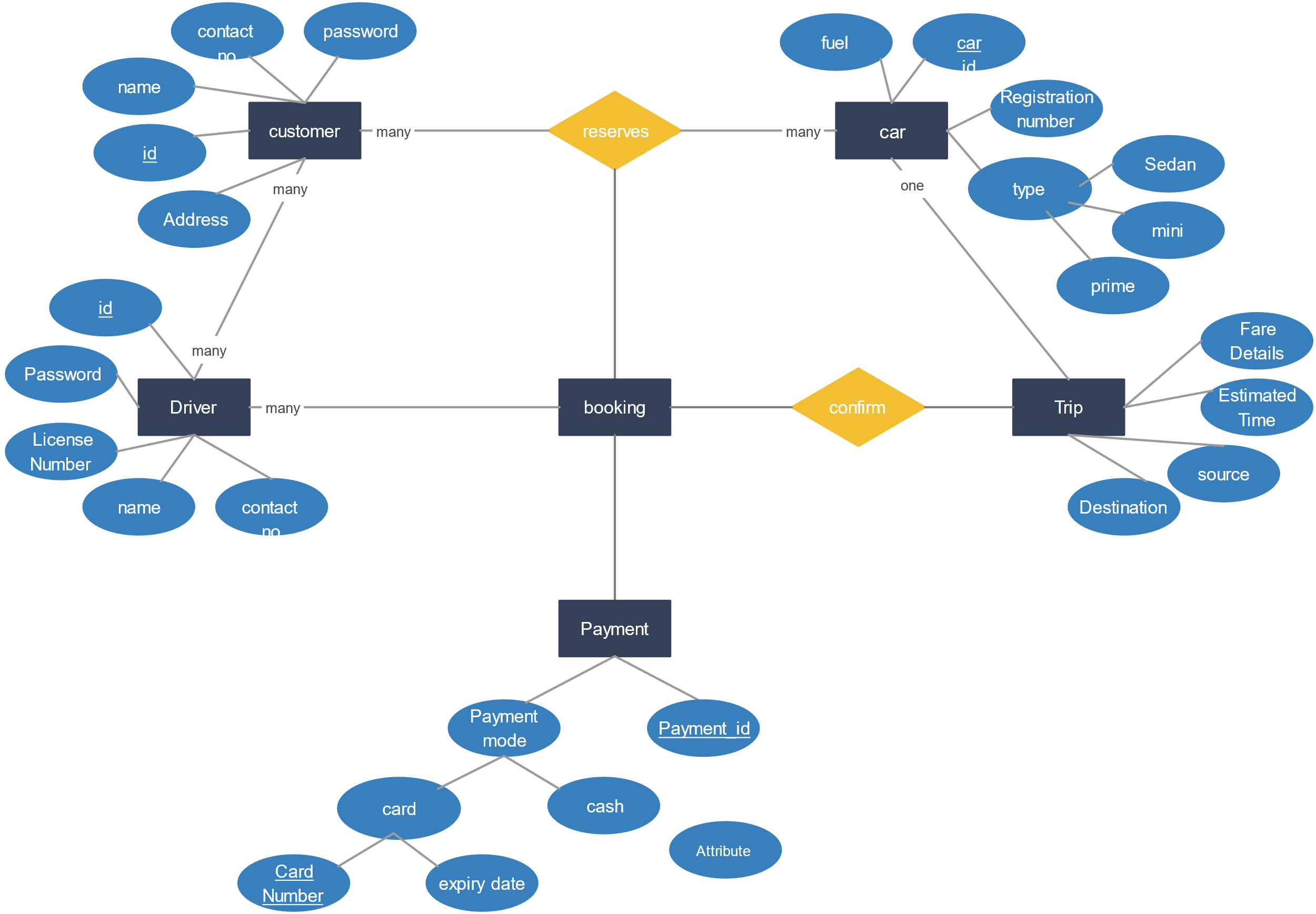
We have a review section for our mechanic which will tell the experiences of the other person with him. In this way our users will get to know about the mechanic which will be appointed to them.

1. **USER FRIENDLY WEB AND ANDROID APPLICATION**

The design and UI of our application is simple to use and is designed in a way that everyone can use it without any difficulty.

* 1. **SOFTWARE RELATED DIAGRAM**
     1. **ER DIAGRAM**

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

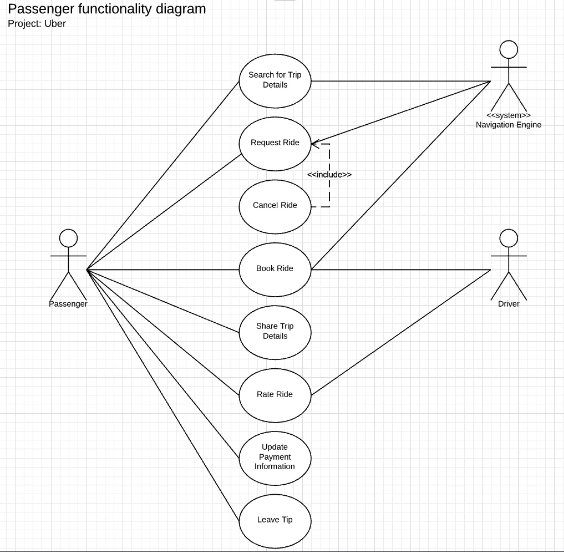


**Figure:1 ER-Diagram**

* + 1. **USE CASE DIAGRAM**

In UML, use-case diagrams model the behaviour of a system and help to capture the requirements of the system.

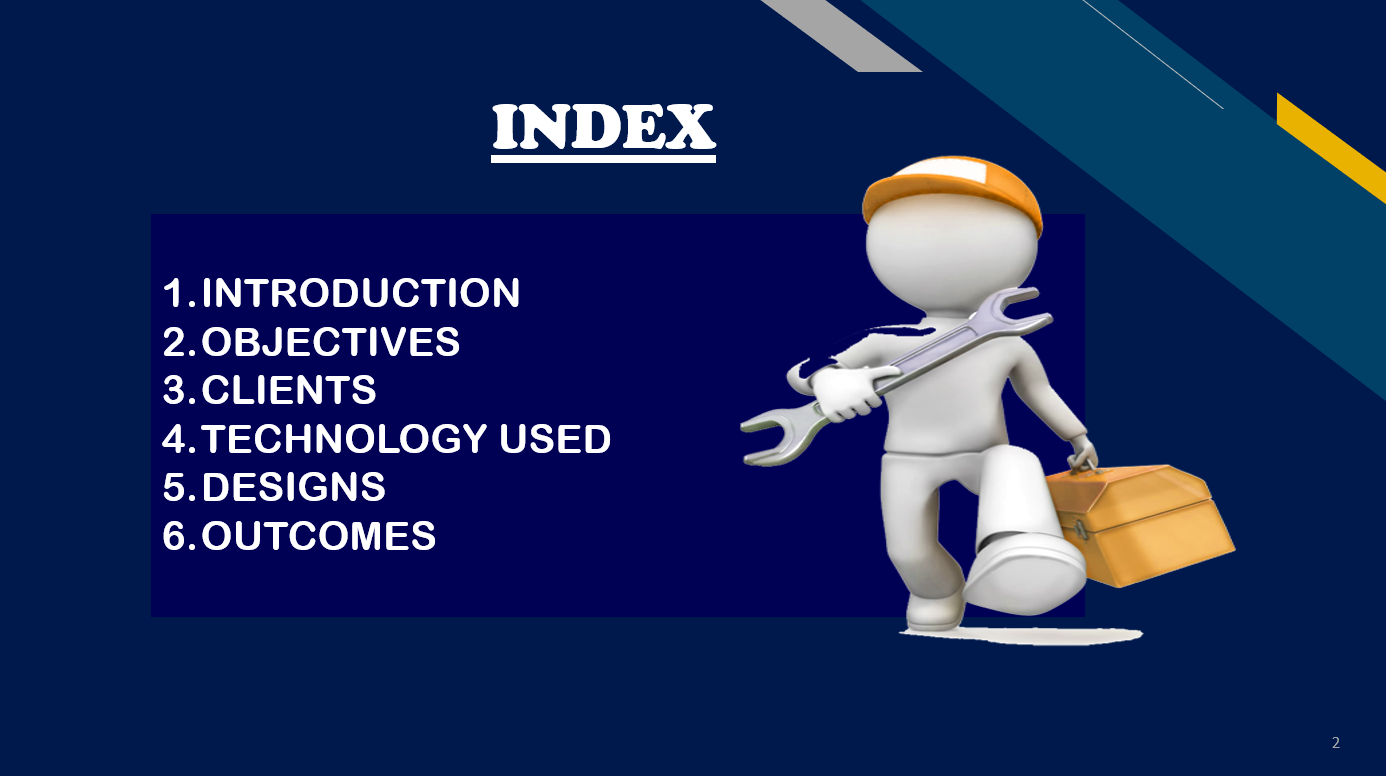
Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

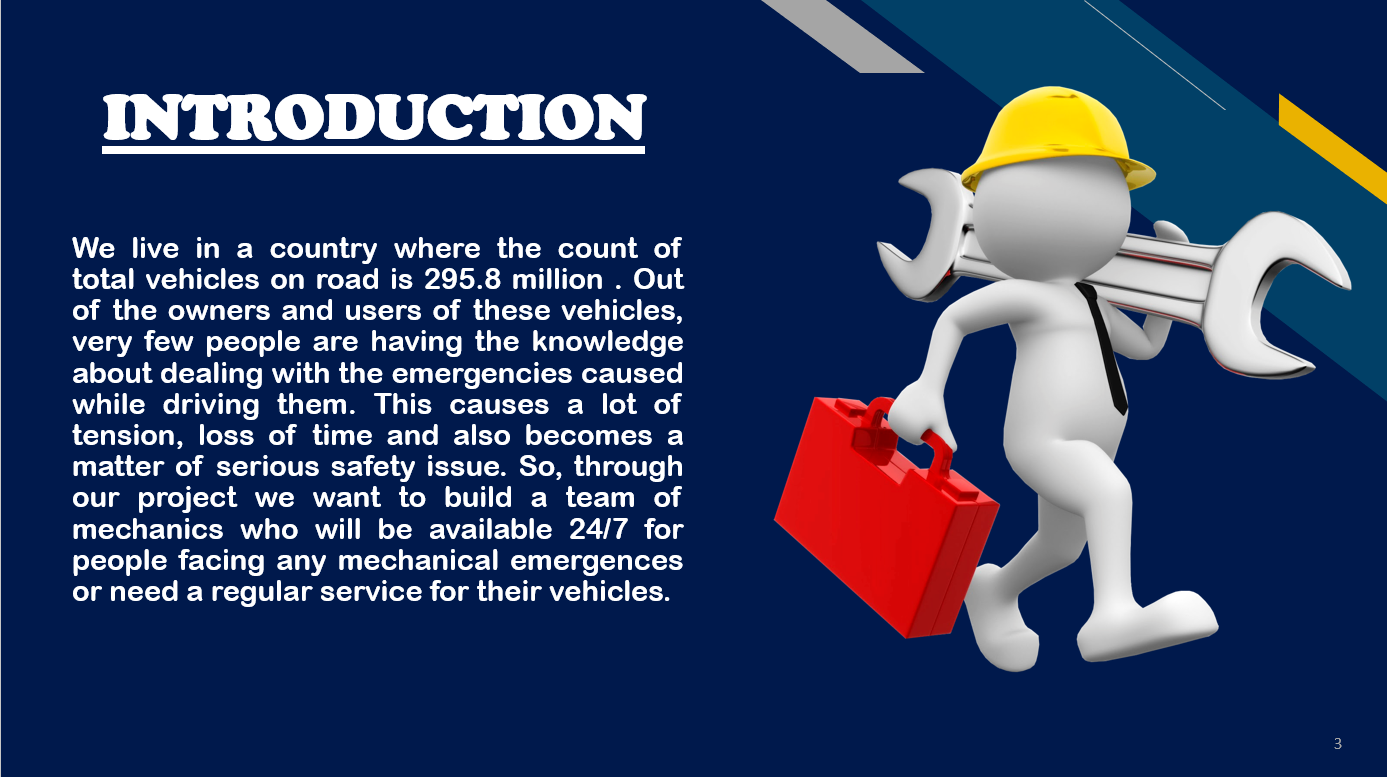


**Figure:2 Use Case Diagram**

* 1. **PROFESSIONAL COMMUNICATION (PRESENTATION)**

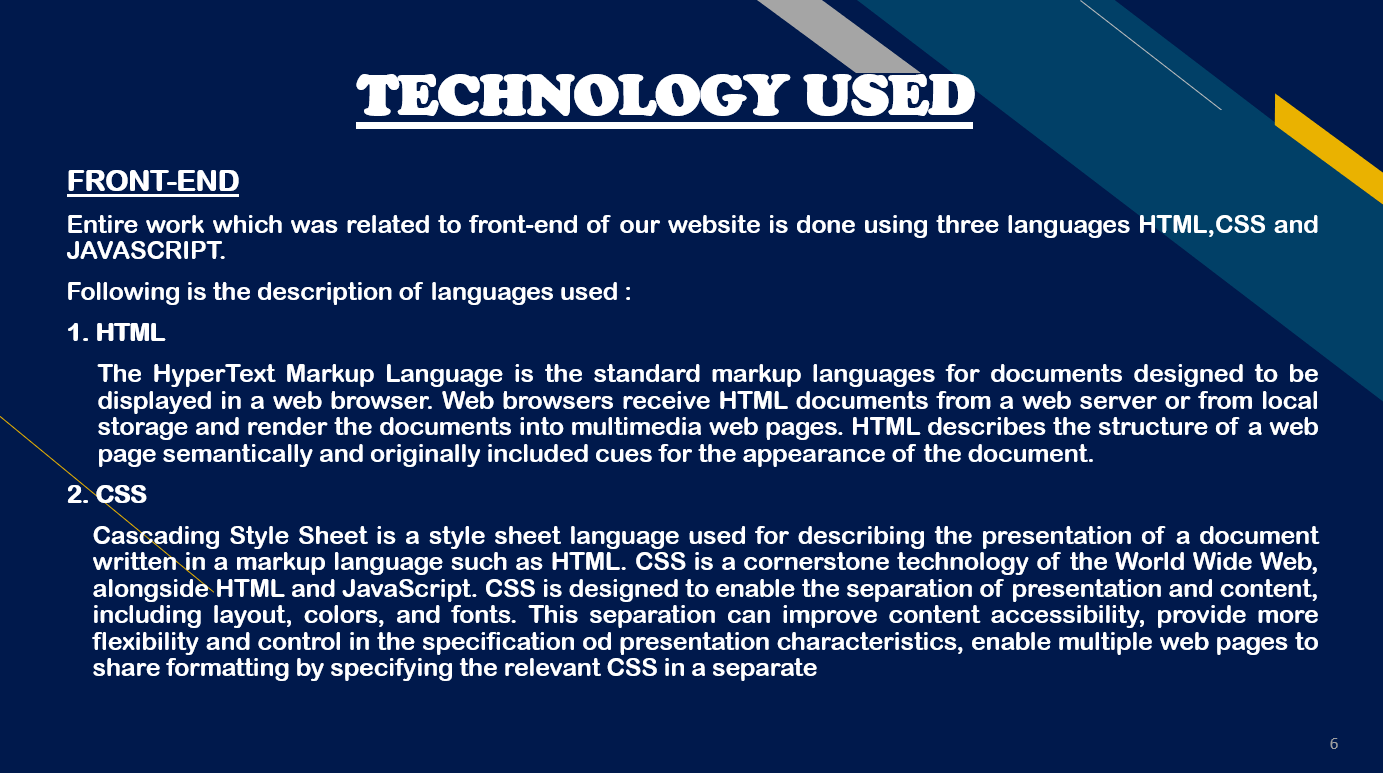
****

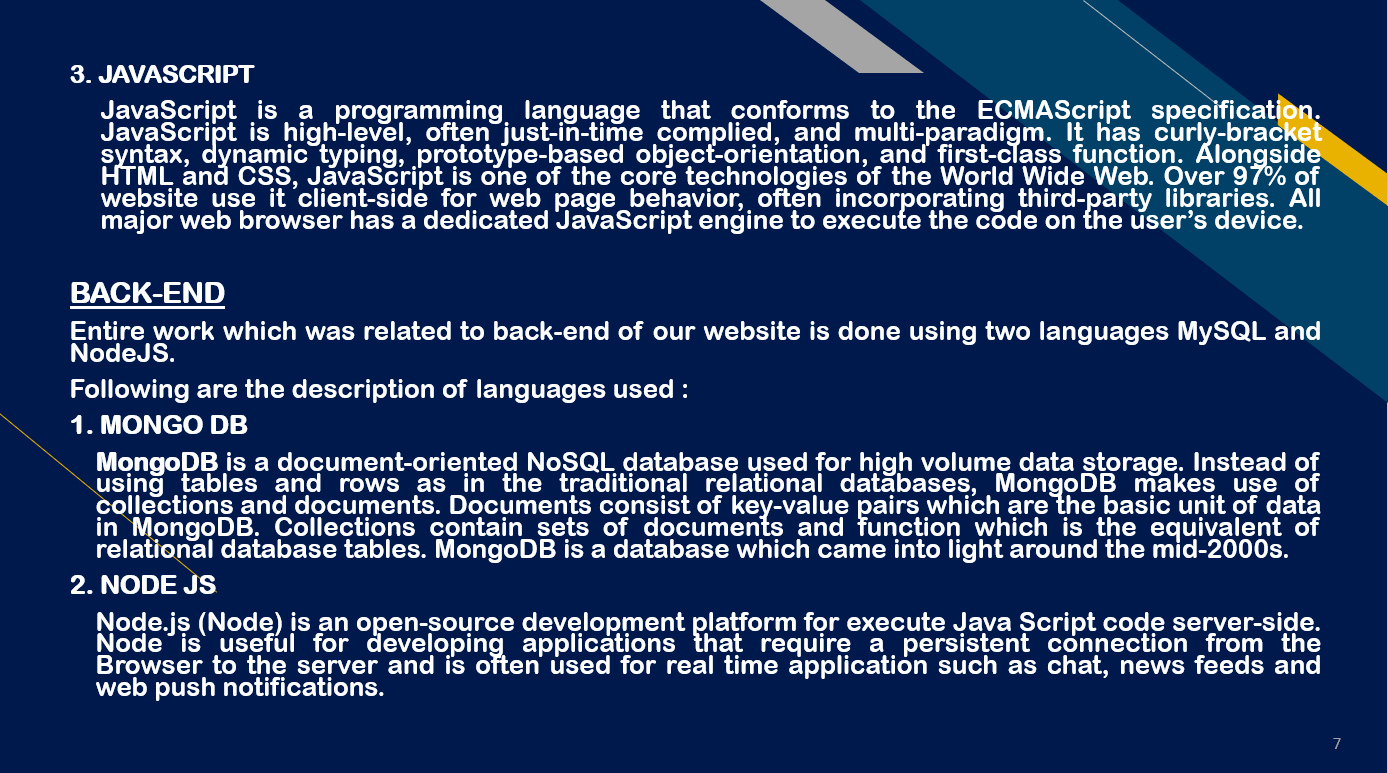
****

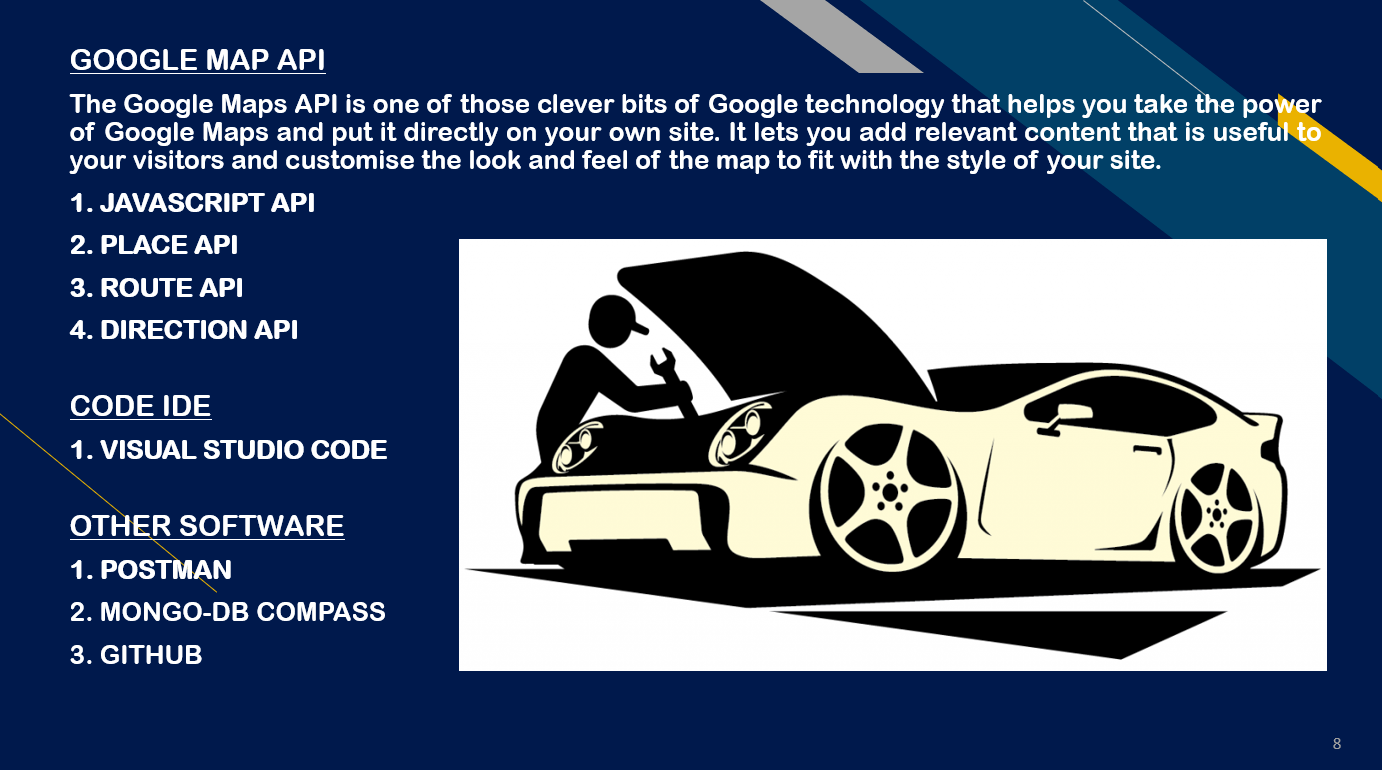
****

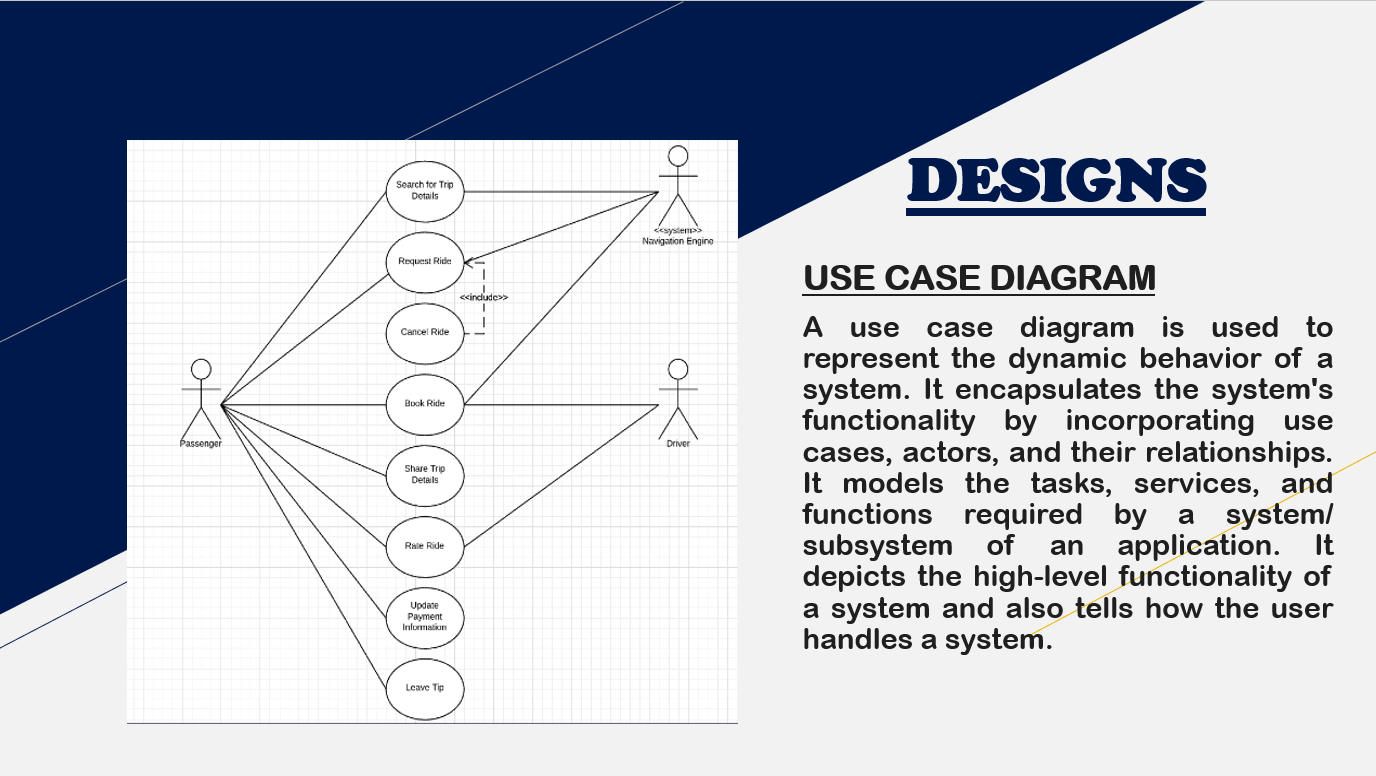
****

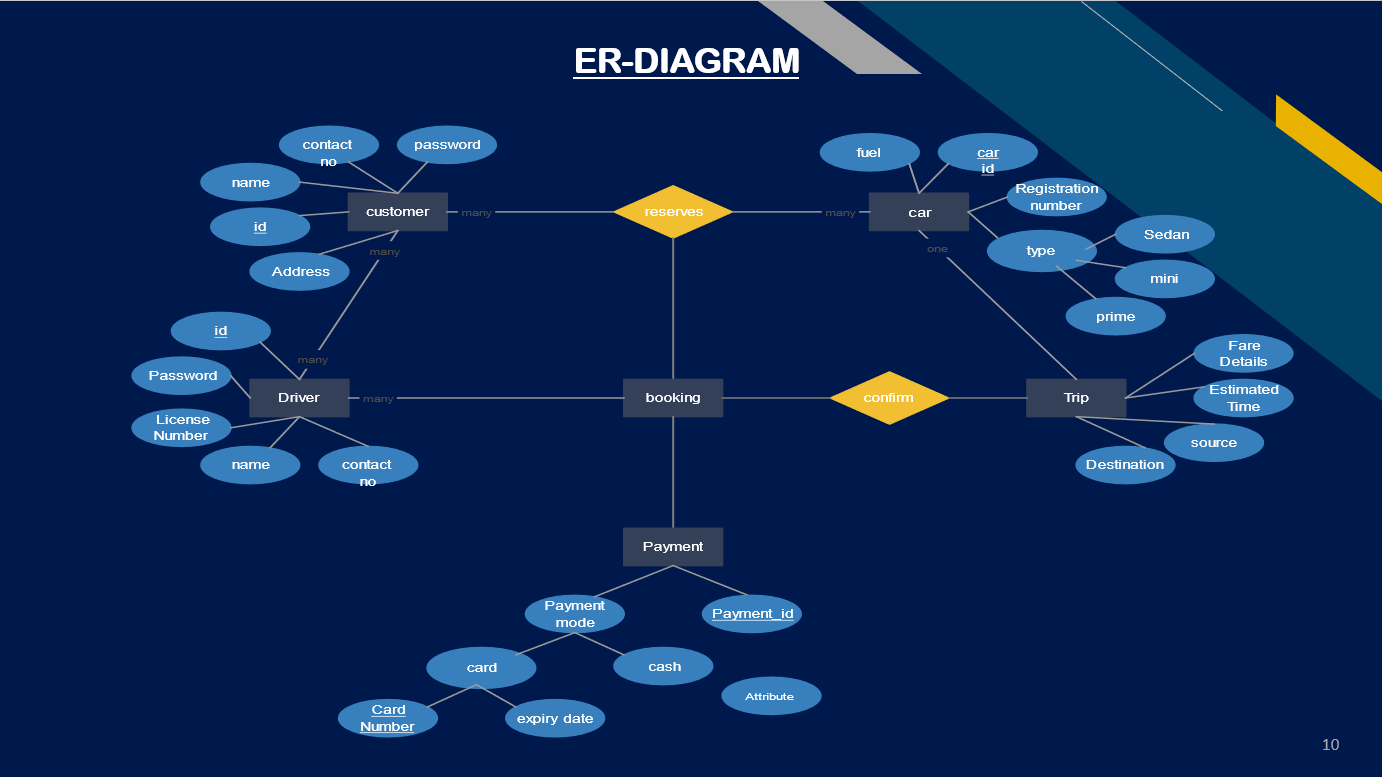
****

****

****

****

****

****

****

****

**CHAPTER- 5 CONCLUSION AND FUTURE WORK**

* 1. **FUTURE WORK**

1. Introducing the driving schools and adding a feature to directly book an appointment to learn driving.

2. Linking the local stores.

3. Adding a feature to buy and sell old vehicles.

4. Feature to find out the best parking space in a particular area.

5. Adding vlogs with simple animation to deal with emergencies on our website and application.

* 1. **OUTCOMES**

At this point of time, we have made a web application to demonstrate the working of our project. Our website is completely working, user friendly and it is responsive as well. Users can make their profile by just giving few information about themselves. Mechanics who are willing to work for and with can also make their profile as we have a separate section for their profile which is named as “Work for Us”. The profile made by them can be edited at any point of time. We are able to fetch the live location of our user through google map API to provide them mechanic at their place. We have used many different modules in the back-end of our website to make it strong and to ensure that the privacy of our users is never compromised.

In this way we can say that, almost all the goals which were decided at the beginning of this project are accomplished. There are many features which we have decided to add in our application in future while launching it. We believe that our website will help this society in every possible way and will reach a great height in future.

* 1. **SUSTAINABLE DEVELOPMENT GOALS**

While talking about Sustainable Development Goals, our project matched goals 8 of this list. Goal 8 is for “Decent Work And Economical Growth” As we have decided to make our own team of mechanics so this will increase the employment opportunities for mechanics and ultimately it will help Indian economy to grow as well.

**REFERENCES**

1. Google Map for Web

<https://developers.google.com/maps/documentation/javascript>

1. Google Map for Android

<https://developers.google.com/maps/documentation/android-sdk>

1. Type of problems related to vehicles

<https://www.bankbazaar.com/insurance/motor-insurance-guide/how-to-handle-driving-emergencies.html>

1. Web-Development

[https://www.w3schools.com/](https://www.w3schools.com/js/default.asp)

1. Android Application

<https://developer.android.com/coursesss>

**USER MANUAL**