

## **CHAPTER 1**

### **INTRODUCTION**

## **1.1 PROJECT OVERVIEW**

The real power of this project is not just managing spare parts, but in the formation of stronger relationships with customers and delivering of a high level of service and support, which in turn improves organization sales and its goodwill. The Auto Trader Project is a software application which avoids more manual hours that need to waste in record keeping and generating reports. This website keeps the data in a centralized way which is available to all the users at the same time. It manages historical data in database.

New thing that included in this project is that the customers how come to service center also have access to the web where he/she can view all the service history of their vehicle. And also, the customer can book online for servicing and select the kilometer at which the car is going to service which will provide the bill estimate by evaluating car wear based on kilometer it has ran.

## **1.2 PROJECT SPECIFICATION**

The proposed system is a website in which user can buy and sell vehicles online. Also that the customers did not need to search vehicles everywhere instead simply need to the web were he/she can view all the vehicles and their details

The system includes 2 modules. They are:

1. Registration
2. Login
3. Searching a Vehicles
4. Selling
  - Add,View, edit and delete Vehicles to Sell
  - View solded Vehicles
5. Buying
  - View buyed Vehicle
  - Buy a Vehicle
  - Add a Vehicle to wish List

## **CHAPTER 2**

### **SYSTEM STUDY**

---

## 2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

## **2.2 EXISTING SYSTEM**

Existing system is not a fully automated system. Customer can register and they can book for service. Each customer can create their own profile. The proposed system rectify the drawbacks of the present system.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. Using the new system customers can view all information about his car, bill etc.

## **2.3 DRAWBACKS OF EXISTING SYSTEM**

- No proper online management of system
- Human effort is needed.
- It is difficult to maintain important information in books.
- More manual hours need to generate required reports.

## **2.4 PROPOSED SYSTEM**

The proposed system is defined to meet all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for growth of service center; on such consideration the system is proposed. In our proposed system there is admin who can view all the customers. It allows customers to make their service booking and do their transactions by using online payment method. Users of this proposed system are admin and customer. The software application which avoids more manual hours that need to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to all the users simultaneously. It is very easy to manage historical data in database. No specific training is required for the distributors to use this application. They can easily use the tool that decreases manual hours spending for normal things and hence increases the performance. It is very easy to record the information of online sales and purchases in the databases.

## **2.5 ADVANTAGES OF PROPOSED SYSTEM**

The system is very simple in design and to implement. The system requires very low system resources, and the system will work in almost all configurations. It has got following features:

### **➤ Better security: -**

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

### **➤ Ensure data accuracy: -**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

### **➤ Better service: -**

The product will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

## **CHAPTER 3**

# **REQUIREMENT ANALYSIS**

### 3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

#### 3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, DREAMS was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.



### 3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

### 3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

---

## 3.2 SYSTEM SPECIFICATION

### 3.2.1 Hardware Specification

Processor	- Intel core i3
RAM	- 4 GB
Hard disk	- 1 TB

### 3.2.2 Software Specification

Front End	- HTML, CSS
Backend	- MYSQL
Client on PC	- Windows 7 and above.
Technologies used	- JS, HTML5, AJAX, J Query, PHP, CSS

## 3.3 SOFTWARE DESCRIPTION

### 3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP:HypertextPreprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

### 3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992,

“SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

## **CHAPTER 4**

### **SYSTEM DESIGN**

## 4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

## 4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has

become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- Statechart diagram
- Deployment diagram
- Component diagram

#### **4.2.1 USE CASE DIAGRAM**

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

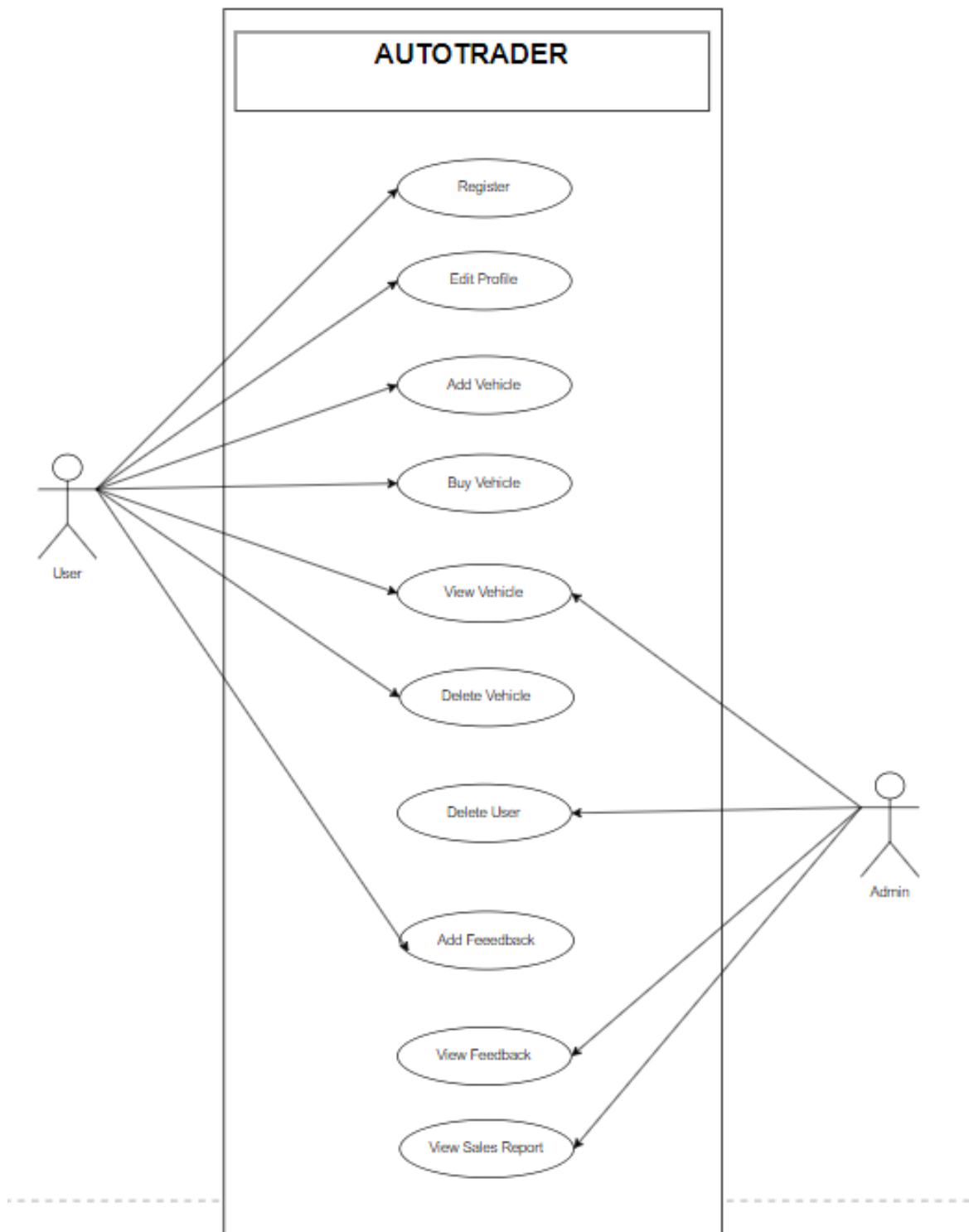
- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.

- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.





## 4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

### Sequence Diagram Notations –

- i. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.
- ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram
- iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

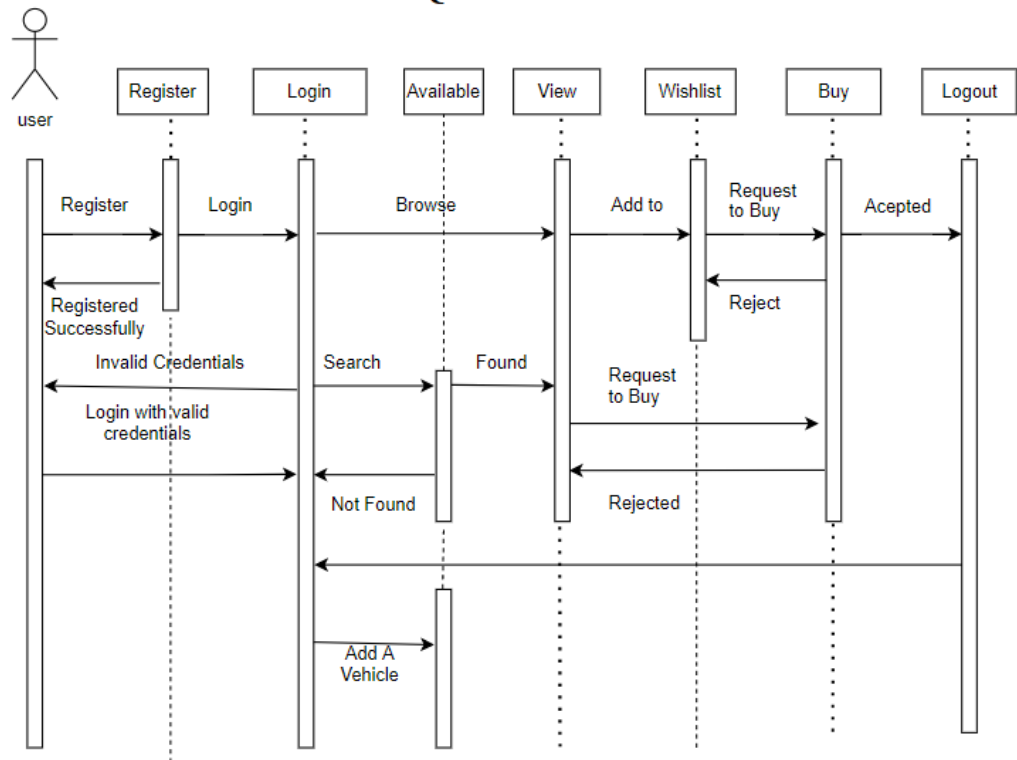
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message

- Lost Message

**iv. Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

**Uses of sequence diagrams –**

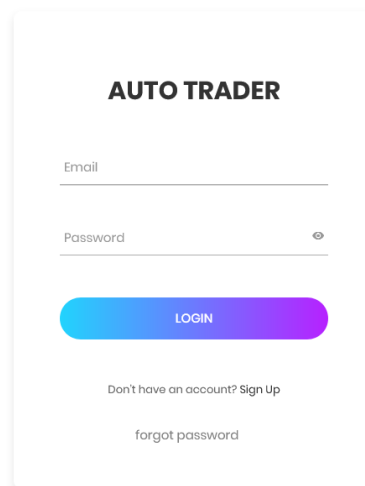
- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

**SEQUENCE DIAGRAM**

## 4.5 USER INTERFACE DESIGN

### 4.5.1-INPUT DESIGN

#### User Login



A login form titled "AUTO TRADER". It features two input fields: "Email" and "Password". The "Password" field has a toggle icon (an eye) to the right. Below the fields is a blue gradient button labeled "LOGIN". At the bottom, there are two links: "Don't have an account? Sign Up" and "forgot password".

**AUTO TRADER**

Email

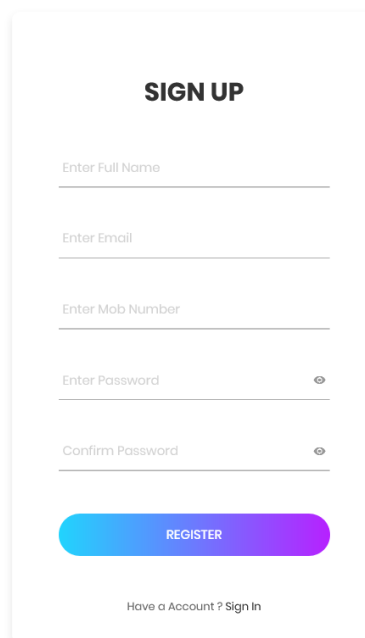
Password

LOGIN

Don't have an account? Sign Up

forgot password

#### User Registration



A registration form titled "SIGN UP". It features five input fields: "Enter Full Name", "Enter Email", "Enter Mob Number", "Enter Password", and "Confirm Password". The "Enter Password" and "Confirm Password" fields have toggle icons (an eye) to the right. Below the fields is a blue gradient button labeled "REGISTER". At the bottom, there is a link: "Have a Account ? Sign In".

**SIGN UP**

Enter Full Name

Enter Email

Enter Mob Number

Enter Password

Confirm Password

REGISTER

Have a Account ? Sign In

## Sell a Vehicle page

### ENTER VEHICLE DETAILS

BRAND

BRAND

DATE

dd / mm / yyyy

FUEL

-SELECT-

TRANSMISSION

Automatic

Manuel

KM DRIVEN

NO OF OWNERS

AD TITLE

DESCRIPTION

SET A PRICE

STATE

Select State

DISTRICT

Select District

CITY

IMAGE OF VEHICLE

Browse...

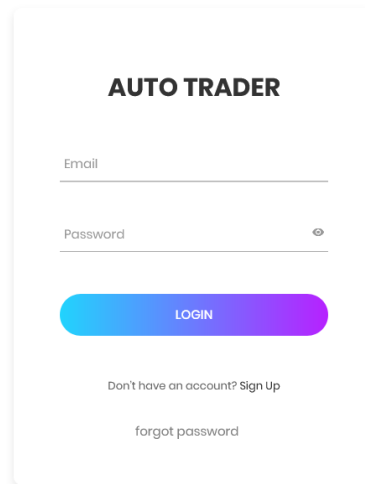
No file selected.

Mobile Number

Register

## 4.5.2 OUTPUT DESIGN

### User Login



A login form titled "AUTO TRADER". It features two input fields: "Email" and "Password". The "Password" field has a toggle icon (an eye) to the right. Below the fields is a blue-to-purple gradient button labeled "LOGIN". At the bottom, there are two links: "Don't have an account? Sign Up" and "forgot password".

**AUTO TRADER**

Email

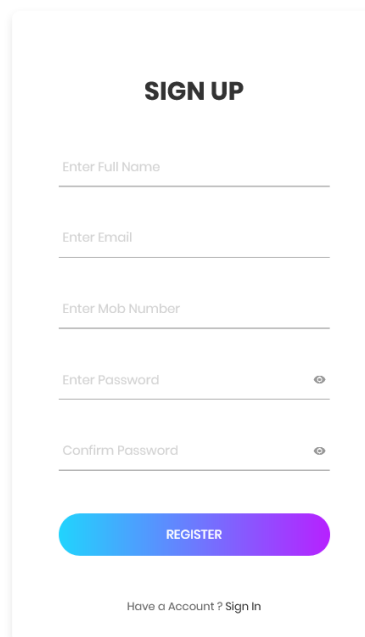
Password

LOGIN

Don't have an account? Sign Up

forgot password

### User Registration



A registration form titled "SIGN UP". It features five input fields: "Enter Full Name", "Enter Email", "Enter Mob Number", "Enter Password", and "Confirm Password". The "Enter Password" and "Confirm Password" fields have toggle icons (an eye) to the right. Below the fields is a blue-to-purple gradient button labeled "REGISTER". At the bottom, there is a link: "Have a Account ? Sign In".

**SIGN UP**

Enter Full Name

Enter Email

Enter Mob Number

Enter Password

Confirm Password

REGISTER

Have a Account ? Sign In

## Sell a Vehicle page

### ENTER VEHICLE DETAILS

BRAND

BRAND

DATE

dd/mm/yyyy

FUEL

-SELECT-

TRANSMISSION

Automatic

Manuel

KM DRIVEN

NO OF OWNERS

AD TITLE

DESCRIPTION

SET A PRICE

STATE

Select State

DISTRICT

Select District

CITY

IMAGE OF VEHICLE

Browse...

No file selected.

Mobile Number

Register



## 4.6. DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

### 4.6.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a table represents a set of related values.

#### Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of  $n$  elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain  $D$  is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

## Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

### 4.6.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

**First Normal Form**

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

**Second Normal Form**

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

**Third Normal Form**

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

**TABLE DESIGN****vehreg**Primary key **vid**Foreign key **id** references id (usereg)

SI No	Name	Type	Description
1.	vid	Int(10)	Vehicle Id
2.	id	Int(10)	User ID
3.	brand	Varchar(20)	Brand
4.	year	Varchar(20)	year
5.	fuel	Varchar(20)	fuel
6.	transmission	Varchar(20)	transmission
7.	kmdriven	Varchar(20)	Km Driven
8.	noofowner	Varchar(20)	No of owner
9.	adtitle	Varchar(30)	Ad Title
10.	description	Varchar(50)	Description
11.	price	Varchar(20)	Price
12.	state	Varchar(20)	State
13.	district	Varchar(25)	District
14.	city	Varchar(20)	City
15.	image	Varchar(100)	Image of Vehicle
16.	contact	Varchar(25)	Contact Number
17.	username	Varchar(25)	User Owned

**feedback**Primary key **fid**

SI No	Name	Data Type	Description
1.	fid	Int(20)	Feedback Id
2.	Name	Varchar(20)	Name of User
3.	Email	Varchar(30)	Email ID
4.	Message	Varchar(200)	Feedback Message
5.	Username	Varchar(20)	Username

**wishlist**

Primary key **wid**

Foreign key **vid** references vid (vehreg)

SI No	Name	Type	Description
1.	wid	Int(20)	Wish List Id
2.	vid	Int(10)	Vehicle Id
3.	brand	Varchar(20)	Brand
4.	year	Varchar(20)	year
5.	kmdriven	Varchar(20)	Km Diven
6.	adtitle	Varchar(30)	Ad Title
7.	description	Varchar(50)	Description
8.	city	Varchar(20)	City
9.	district	Varchar(25)	District
10.	state	Varchar(20)	State
11.	price	Varchar(20)	Price
12.	image	Varchar(100)	Image of Vehicle
13.	username	Varchar(50)	Username

**buyedveh**

Primary key : **bid**

SI No	Name	Data Type	Description
1.	bid	Int(20)	Vehicle Id
2.	brand	Varchar(20)	Brand
3.	year	Varchar(20)	year
4.	fuel	Varchar(20)	fuel
5.	transmission	Varchar(20)	transmission
6.	adtitle	Varchar(30)	Ad Title
7.	price	Varchar(20)	Price
8.	contact	Varchar(20)	Contact number
9.	userbuyed	Varchar(25)	userbuyed
10.	image	Varchar(100)	Image of Vehicle
11.	Dateselled	Date	Date of vehicle Sold

**request**Primary key **rid**Foreign key **vid** references vid (vehreg)

SI No	Name	Data Type	Description
1.	rid	Int(20)	Request Id
2.	vid	Int(10)	Vehicle Id
3.	AdTitle	Varchar(50)	Ad Title
4.	brand	Varchar(20)	Brand
5.	year	Varchar(20)	year
6.	kmdriven	Varchar(20)	Km Diven
7.	description	Varchar(50)	Description
8.	price	Varchar(20)	Price
9.	image	Varchar(100)	Image of Vehicle
10.	vehreq	Varchar(50)	Vehicle Requested
11.	vehowner	Varchar(50)	Vehicle Owner

**selledveh**Primary key **sid**

SI No	Name	Data Type	Description
1.	sid	Int(20)	Vehicle Id
2.	brand	Varchar(20)	Brand
3.	year	Varchar(20)	year
4.	fuel	Varchar(20)	fuel
5.	transmission	Varchar(20)	transmission
6.	kmdriven	Varchar(20)	Km Diven
7.	adtitle	Varchar(30)	Ad Title
8.	price	Varchar(20)	Price
9.	contact	Varchar(20)	Contact number
10.	userselled	Varchar(25)	userbused
11.	image	Varchar(100)	Image of Vehicle
12.	Dateselled	Date	Date of vehicle Sold

**userreg**Primary key : **id**

SI No	Name	Type	Description
1.	<b>id</b>	Int(20)	User id
2.	image	Varchar(500)	User Image
3.	name	Varchar(50)	Name
4.	username	Varchar(50)	Username
5.	password	Varchar(100)	Password
6.	cpassword	Varchar(100)	Confirm Password
7.	address	Varchar(100)	Address
8.	locality	Varchar(100)	Local Area
9.	city	Varchar(50)	City
10.	district	Varchar(75)	District
11.	state	Varchar(50)	State
12.	gender	Varchar(30)	Gender
13.	mob	Varchar(20)	Mobile
14.	dob	Date	Date of Birth
15.	usertype	Varchar(30)	User Type
16.	status	Int(11)	User Active or Not
17.	Sellername	Int(30)	Seller Name
18.	Sellermob	Int(50)	Seller Mobile Number

## **CHAPTER 5**

### **SYSTEM TESTING**



## 5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

## 5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

### 5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

### **5.2.2 Integration Testing**

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

### **5.2.3 Validation Testing or System Testing**

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

---

#### **5.2.4 Output Testing or User Acceptance Testing**

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

## **CHAPTER 6**

# **IMPLEMENTATION**

## 6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after thorough testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- ❑ Careful planning.
- ❑ Investigation of system and constraints.
- ❑ Design of methods to achieve the changeover.

## 6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to

ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

### **6.2.1 User Training**

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

### **6.2.2 Training on the Application Software**

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

### **6.2.3 System Maintenance**

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

## **CHAPTER 7**

### **CONCLUSION AND FUTURE SCOPE**



## **7.1 CONCLUSION**

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for customer to book service online and view all information.

## **CHAPTER 8**

## **BIBLIOGRAPHY**

---

**REFERENCES:**

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- PankajJalote, “*Software engineering: a precise approach*”, 2006.
- James lee and Brent ware Addison, “Open source web development with LAMP”, 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

**WEBSITES:**

- [www.w3schools.com](http://www.w3schools.com)
- [www.jquery.com](http://www.jquery.com)
- <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>
- [www.agilemodeling.com/artifacts/useCaseDiagram.html](http://www.agilemodeling.com/artifacts/useCaseDiagram.html)

## **CHAPTER 9**

## **APPENDIX**

## 9.1 Sample Code

### Logi.html

```
<title>Login</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" type="image/png" href="styles/images/icons/favicon.ico" />
<link rel="stylesheet" type="text/css" href="styles/vendor/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="styles/fonts/font-awesome-4.7.0/css/font-
awesome.min.css">
<link rel="stylesheet" type="text/css" href="styles/fonts/iconic/css/material-design-iconic-
font.min.css">
<link rel="stylesheet" type="text/css" href="styles/vendor/animate/animate.css">

<link rel="stylesheet" type="text/css" href="styles/vendor/css-hamburgers/hamburgers.min.css">
<link rel="stylesheet" type="text/css" href="styles/vendor/animasition/css/animasition.min.css">

<link rel="stylesheet" type="text/css" href="styles/vendor/select2/select2.min.css">
<link rel="stylesheet" type="text/css" href="styles/vendor/daterangepicker/daterangepicker.css">
<link rel="stylesheet" type="text/css" href="styles/css/util.css">
<link rel="stylesheet" type="text/css" href="styles/css/main.css">

<style>
  .bg {
    /* The image used */
    /*background-image: url("./styles/images/banner4.jpg");

    /* Full height */
    height: 100%;

    /* Center and scale the image nicely */
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
  }
  .trans {
    padding: 20px 90px 20px;
    font-size: 15px;
    background-position: center;
    cursor: pointer;
    color: #808080;
    background-color: transparent;
  }
</style>
<script type="text/javascript" language="javascript">
  function val() {

    var email = document.login.username.value;
    var pwd = document.login.password.value;

    if (email == "") {
      alert("Please Enter a email Id");
```

```

        document.login.username.focus();
        return false;
    }

    var regemail = /@gmail\.com$/;
    if (regemail.test(email) === false) {
        alert("Please enter a valid email address ");
        document.login.username.focus();
        return false;
    }

    var regpwd = /(?!.*\d)(?!.*[a-z])(?!.*[A-Z]).{6,}/;
    if (pwd == "") {
        alert("Please Enter a Password");
        document.login.password.focus();
        return false;
    }
}
</script>
<body>
    <div class="bg">
        <form method="post" name="login" action="login.php">
            <div class="limiter">
                <div class="container-login100">
                    <div class="wrap-login100">
                        <form class="login100-form validate-form">
                            <span class="login100-form-title p-b-48">AUTO TRADER</span>
                            <div class="wrap-input100 validate-input" data-
                                validate="Valid email is: a@gmail.com">
                                <input class="input100" type="email"
                                    name="username" required>
                                <span class="focus-input100" data-
                                    placeholder="Email"></span>
                                </div>
                                <div class="wrap-input100 validate-input" data-
                                    validate="Enter password">
                                    <span class="btn-show-pass">
                                        <i class="zmdi zmdi-eye"></i>
                                    </span>
                                    <input class="input100" type="password"
                                        name="password">
                                    <span class="focus-input100" data-
                                        placeholder="Password"></span>
                                </div>
                                <div class="container-login100-form-btn">
                                    <div class="wrap-login100-form-btn">
                                        <div class="login100-form-
                                            bgbtn"></div>
                                        <button onclick="return val()"
                                            name="login" class="login100-form-btn">
                                            Login
                                        </button>

```

```

                                </div>
                            </div>

                            <div class="text-center p-t-50">

                                <span class="txt1">
                                    Don't have an account?
                                </span>

                                <a class="txt2" href="./user/user_reg.php">
                                    Sign Up
                                </a>
                            </div>

                            <div class="container-login100-form-btn">
                                <div class="wrap-login100-form-btn">

                                    <button class="trans"
name="fpassword">
                                        forgot password
                                    </button>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <div id="dropDownSelect1"></div>
    </form>
</div>

<!--
=====
=====-->
<script src="styles/vendor/jquery/jquery-3.2.1.min.js"></script>
<!--
=====
=====-->
<script src="styles/vendor/ansition/js/ansition.min.js"></script>
<!--
=====
=====-->
<script src="styles/vendor/bootstrap/js/popper.js"></script>
<script src="styles/vendor/bootstrap/js/bootstrap.min.js"></script>
<!--
=====
=====-->
<script src="styles/vendor/select2/select2.min.js"></script>
<!--
=====
=====-->
<script src="styles/vendor/daterangepicker/moment.min.js"></script>
<script src="styles/vendor/daterangepicker/daterangepicker.js"></script>
<!--
=====
=====

```

```

=====-->
<script src="styles/vendor/countdowntime/countdowntime.js"></script>
<!--
=====
=====-->
<!--<script src="styles/js/main.js"></script>-->

</body>

```

### **Veh\_filter.php**

```

<?php
session_start();
if (isset($_SESSION['username'])) {
    $username = $_SESSION['username'];
    include 'include/connect.php';

    $iid = $_SESSION["id"];
    $s1 = "SELECT * FROM userreg where id='$iid'";
    $res1 = mysqli_query($conn, $s1);
    while ($r1 = mysqli_fetch_array($res1)) {
        $img = $r1["image"];
    }
}

?>
<!DOCTYPE html>
<html>
<script type="text/javascript">
    function val() {
        var brand = document.filter.brand.value;
        var sdate = document.filter.sdate.value;
        var edate = document.filter.edate.value;
        var sp = document.filter.sprice.value;
        var ep = document.filter.eprice.value;
        var km = document.filter.km.value;
        var fuel = document.filter.fuel.value;
        var trans = document.filter.trans.value;

        /* if (brand == "") {
            alert("Please Select a Brand");
            document.filter.brand.focus();
            return false;
        } */

        if (sdate > edate) {
            alert('Start Date must be less than End Date');
            return false;
        }

        if (sp > ep) {
            alert('Starting Price must be less than Ending Price');
            return false;
        }
        /* if (km == "") {

```



```

        alert("Please Enter Kilometer");
        document.filter.km.focus();
        return false;
    }
    if (fuel == "") {
        alert("Please Select a Fuel Type");
        document.filter.fuel.focus();
        return false;
    }
    if (trans == "") {
        alert("Please Select a Transmission");
        document.filter.trans.focus();
        return false;
    }
}*/
}
</script>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="Start your development with a Dashboard for Bootstrap
4.">
    <meta name="author" content="Creative Tim">
    <title>User Home</title>
    <!-- Favicon -->
    <link rel="icon" href="assets/img/brand/a.jpg" type="image/jpg">
    <link rel="stylesheet" href="vendors/feather/feather.css">
    <link rel="stylesheet" href="vendors/ti-icons/css/themify-icons.css">
    <link rel="stylesheet" href="vendors/css/vendor.bundle.base.css">
    <!-- Fonts -->
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700">
    <!-- Icons -->

    <link rel="stylesheet" href="assets/vendor/nucleo/css/nucleo.css" type="text/css">
    <link rel="stylesheet" href="assets/vendor/@fortawesome/fontawesome-free/css/all.min.css"
type="text/css">
    <!-- Argon CSS -->
    <link rel="stylesheet" href="assets/css/argon.css?v=1.2.0" type="text/css">

    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500&display=swap"
rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="css/home.css">
    <link rel="stylesheet" type="text/css" href="css/style123.css">

</head>

<body>
    <!-- Sidenav -->
    <nav class="sidenav navbar navbar-vertical fixed-left navbar-expand-xs navbar-light bg-
white" id="sidenav-main">
        <div class="scrollbar-inner">
            <!-- Brand -->

```

```

<div class="sidenav-header align-items-center">
  <a class="navbar-brand" href="user_home.php">
    
  </a>
</div>
<div class="navbar-inner">
  <!-- Collapse -->
  <div class="collapse navbar-collapse" id="sidenav-collapse-main">
    <!-- Nav items -->
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="user_home.php">
          <i class="ni ni-tv-2 text-primary"></i>
          <span class="nav-link-text">Home</span>
        </a>
      </li>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="profile.php?id=<?php echo $r1['id']; ?>">
          <i class="ni ni-single-02 text-yellow"></i>
          <span class="nav-link-text">Profile</span>
        </a>
      </li>
      <li class="nav-item">
      <li class="nav-item">
        <a class="nav-link" href="veh_filter.php">
          <i class="ni ni-bullet-list-67 text-violet"></i>
          <span class="nav-link-text">Filter</span>
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="vehicle_reg_for_sell.php?id=<?php echo $r1['id'];
?>">
          <i class="ni ni-shop text-red"></i>
          <span class="nav-link-text">Sell a vehicle</span>
        </a>
      </li>
      <?php
    }
    ?>
    <li class="nav-item">
      <a class="nav-link" data-toggle="collapse" href="#ui-basic" aria-
expanded="false" aria-controls="ui-basic">
        <i class="ni ni-folder-17 text-default"></i>
        <span class="menu-title">Vehicle</span>
        <i class="menu-arrow"></i>
      </a>
      <div class="collapse" id="ui-basic">
        <ul class="nav flex-column sub-menu">

```

```

        <li class="nav-item"> <a class="nav-link"
href="view_vehicles_need_to_sell.php"><i class="ni ni-pin-3 text-primary"></i>View Vehicles
Needed To Sell</a></li>
        <li class="nav-item"> <a class="nav-link" href="view_solded_veh.php"><i
class="ni ni-bullet-list-67 text-default"></i>View Solded Vehicles</a></li>
        <li class="nav-item"> <a class="nav-link" href="view_buyes_veh.php"><i
class="ni ni-bullet-list-67 text-teal"></i>View Bought Vehicles</a></li>
    </ul>
</div>
</li>
<li class="nav-item">
    <a class="nav-link" href="request_view.php">
        <i class="ni ni-active-40 text-pink"></i>
        <span class="nav-link-text">Requests</span>
    </a>
</li>
<?php
$sqlc = "SELECT COUNT(*) FROM `request` WHERE vehowner =
'$_username';";
$resc = mysqli_query($conn, $sqlc) or die(mysqli_error($conn));
while ($rowc = mysqli_fetch_array($resc)) {
    $count = $rowc[0];
?>
    <li class="nav-item">
        <a class="nav-link" href="notification.php">
            <i class="ni ni-notification-70 text-green"></i>
            <span class="nav-link-text">Notifications
            <?php
                if ($count != 0) {
                    ?>
                    <label style="color:#000000;font-weight:bold"><?php echo $count;
?></label>
                <?php
                }
            }
            ?>
        </span>
    </a>
</li>
<li class="nav-item">
    <a class="nav-link" href="wishlist.php">
        <i class="ni ni-favourite-28 text-red"></i>
        <span class="nav-link-text">Wishlist</span>
    </a>
</li>
<li class="nav-item">
    <a class="nav-link" href="feedback.php">
        <i class="ni ni-email-83 text-cyan"></i>
        <span class="nav-link-text">Review</span>
    </a>
</li>
</ul>
<!-- Divider -->
<hr class="my-3">

```

```

        <!-- Heading -->

        <!-- Navigation -->

    </div>
</div>
</div>
</nav>
<!-- Main content -->
<div class="main-content" id="panel">
    <!-- Topnav -->
    <nav class="navbar navbar-top navbar-expand navbar-dark bg-primary border-bottom">
        <div class="container-fluid">
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <!-- Search form -->
                <?php include 'search.php' ?>
                <!-- Navbar links -->
                <ul class="navbar-nav align-items-center ml-md-right">
                    <li class="nav-item d-xl-none">
                        <!-- Sidenav toggler -->
                        <div class="pr-3 sidenav-toggler sidenav-toggler-dark" data-action="sidenav-
pin" data-target="#sidenav-main">

                            </div>
                        </li>
                    </ul>
                    <ul class="navbar-nav align-items-center ml-auto ml-md-left">
                        <li class="nav-item dropdown">
                            <a class="nav-link pr-0" href="#" role="button" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
                                <div class="media align-items-center">
                                    <span class="avatar avatar-sm rounded-circle">
                                        
                                    </span>
                                    <div class="media-body ml-2 d-none d-lg-block">
                                        <span class="mb-0 text-sm font-weight-bold">
                                            <h1 style="color:white"><?php echo strtok($_SESSION["name"], '
'); ?></h1>
                                        </span>
                                    </div>
                                </div>
                            </a>
                            <div class="dropdown-menu dropdown-menu-right">
                                <div class="dropdown-header noti-title">
                                    <h6 class="text-overflow m-0">Welcome!</h6>
                                </div>
                                <div class="dropdown-divider"></div>
                                <a href="logout.php" class="dropdown-item">
                                    <i class="ni ni-user-run"></i>
                                    <span>Logout</span>
                                </a>
                            </div>
                        </li>
                    </ul>

```

```

        </div>
    </div>
</nav>
<!-- Header -->
<!-- Header -->
<div class="header bg-primary pb-1">
    <div class="container-fluid">
        <div class="header-body">
            <div class="row align-items-center py-4">
                <div class="col-lg-6 col-7">
                    <h6 class="h2 text-white d-inline-block mb-0"></h6>
                    <nav aria-label="breadcrumb" class="d-none d-md-inline-block ml-md-4">

                        </nav>
                    </div>
                </div>
            </div>
        </div>
    </div>
<!-- Page content -->
<div class="container-fluid mt--6">
    <div class="row">
        <div class="col">
            <div class="card">
                <!-- Card header -->
                <div class="card-body">
                    <div class="pl-lg-4">
                        <div class="row">
                            <div class="col-md-12">
                                <div class="form-group">

                                    <label class="form-control-label" for="input-city">BRAND</label>
                                    <form action="veh_filter_sql.php" name="filter" method="post">
                                        <?php include './include/veh_brand.php' ?>
                                        <label class="form-control-label" for="input-city"><B>Vehicle
Between</B> </label>

                                        </br>
                                        <label class="form-control-label" for="input-city">Start
DATE</label>

                                        <input type="date" class="form-control" name="sdate" />
                                        <label class="form-control-label" for="input-city">End
DATE</label>

                                        <input type="date" class="form-control" name="edate" />
                                        <label class="form-control-label" for="input-city"> KM Driven
Under</label>

                                        <input type="number" class="form-control" min="1"
max="100000" name="km" />

                                        <label class="form-control-label" for="input-city"><B> Price
Between</B></label>

                                        <br>
                                        <label class="form-control-label" for="input-city"> Starting
Price</label>

                                        <input type="number" class="form-control" min="1000"
name="sprice" />

                                        <label class="form-control-label" for="input-city">Ending

```

```

Price</label>
name="eprice" />
<input type="number" class="form-control" min="1000"

<label class="form-control-label" for="input-city"> Fuel</label>
<select class="form-control" name="fuel" id="make">
  <option value="">--SELECT--</option>
  <option value="PETROL">PETROL</option>
  <option value="DIESEL">DIESEL</option>
  <option value="ELECTRIC">ELECTRIC</option>
  <option value="CNG & Hybrid">CNG & Hybrid</option>
</select>
<label class="form-control-label" for="input-city">

Transmission</label>

<select class="form-control" name="trans" id="make">
  <option value="">--SELECT--</option>
  <option value="Automatic">Automatic</option>
  <option value="Manuel">Manual</option>
</select>
</div>
</div>
<div class="row">
  <div class="col-lg">
    <div style="padding-left:12px;" class="form">
      <input class="btn btn-primary" onclick="return val()"
type="submit" name="btnbrand" value="Submit" />
    </div>
  </div>
  <hr class="my-4" />
  <!-- Light table -->
<?php

}
?>

</div>
</div>
</div>
</div>
</div>
</div>
<!-- Card footer -->
<div class="card-footer py-4">
  <nav aria-label="...">
    <ul class="pagination justify-content-end mb-0">
      <li class="page-item disabled">
        <a class="page-link" href="#" tabindex="-1">
          <i class="fas fa-angle-left"></i>
          <span class="sr-only">Previous</span>
        </a>
      </li>
    </ul>
  </nav>
</div>
</div>
</div>

```

```

    </div>
    <!-- Dark table -->
    <div class="row">
        <div class="col">
            <div class="card bg-default shadow">
            </div>
        </div>
    </div>
    <!-- Footer -->
    <?php include './include/footer.php'; ?>
</div>
</div>
<!-- Argon Scripts -->
<!-- Core -->
<script src="assets/vendor/jquery/dist/jquery.min.js"></script>
<script src="assets/vendor/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
<script src="assets/vendor/js-cookie/js.cookie.js"></script>
<script src="assets/vendor/jquery.scrollbar/jquery.scrollbar.min.js"></script>
<script src="assets/vendor/jquery-scroll-lock/dist/jquery-scrollLock.min.js"></script>
<!-- Argon JS -->
<script src="assets/js/argon.js?v=1.2.0"></script>
</body>

</html>

```

### Admin\_home.php

```

<?php
session_start();
if (isset($_SESSION['username'])) {
    $username = $_SESSION['username'];
    include './user/include/connect.php';
    $iid = $_SESSION["id"];
    $s1 = "SELECT * FROM userreg where id='$iid'";
    $res1 = mysqli_query($conn, $s1);
    while ($r1 = mysqli_fetch_array($res1)) {
    ?>

    <!DOCTYPE html>
    <html lang="en">

    <?php include './include/header.php' ?>
    <!-- partial -->
    <div class="main-panel">
        <div class="content-wrapper">
            <div class="row">
                <div class="col-md-12 grid-margin">
                    <div class="row">
                        <div class="col-12 col-xl-8 mb-4 mb-xl-0">
                            <h3 class="font-weight-bold">Welcome <?php echo $r1['name'] ?></h3>
                            <h6 class="font-weight-normal mb-0">All systems are running smoothly! <span
class="text-primary"></span></h6>
                        </div>

```

```

        </div>
    </div>
</div>
<div class="row">
    <div class="col-md-6 grid-margin stretch-card">
        <div class="card tale-bg">
            <div class="card-people mt-auto">
                
                <div class="weather-info">
                    <div class="d-flex">
                        <div>
                            <h2 class="mb-0 font-weight-normal"><i class="icon-sun mr-2"></i>31<sup>C</sup></h2>
                        </div>
                        <div class="ml-2">
                            <h4 class="location font-weight-normal">Bangalore</h4>
                            <h6 class="font-weight-normal">India</h6>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>

<?php
}

$sql1 = "SELECT COUNT(*) FROM userreg WHERE usertype='user'";
$result1 = mysqli_query($conn, $sql1);
while ($row1 = mysqli_fetch_array($result1)) {
    $count1 = $row1[0];
}
?>
<div class="col-md-6 grid-margin transparent">
    <div class="row">
        <div class="col-md-6 mb-4 stretch-card transparent">
            <div class="card card-tale">
                <div class="card-body">
                    <a style="color: white;" href="user_details.php">
                        <p class="mb-4">Number of Users</p>
                        <p class="fs-30 mb-2"><?php echo $count1; ?></p>
                    </a>
                </div>
            </div>
        </div>
    </div>
</div>
<?php
$sql2 = "SELECT COUNT(*) FROM vehreg";
$result2 = mysqli_query($conn, $sql2);
while ($row2 = mysqli_fetch_array($result2)) {
    $count2 = $row2[0];
}
?>
<div class="col-md-6 mb-4 stretch-card transparent">

```



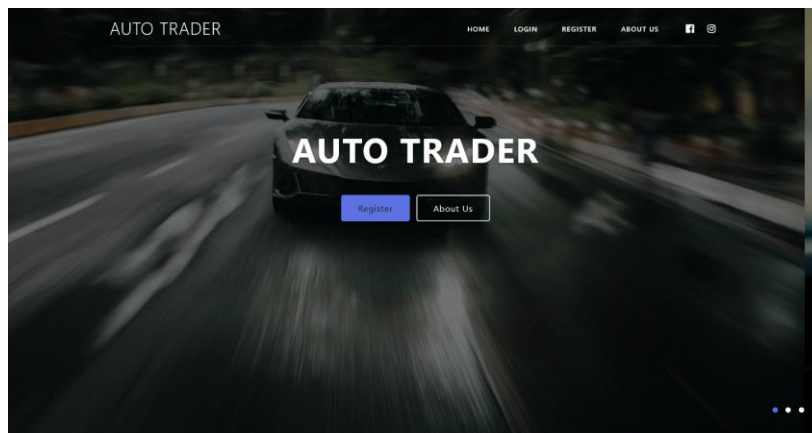
```

<div class="card card-dark-blue">
  <div class="card-body">
    <a style="color: white;" href="#">
      <p class="mb-4">Total Vehicle</p>
      <p class="fs-30 mb-2"><?php echo $count2; ?></p>
    </a>
  </div>
</div>
</div>
<?php
$sql3 = "SELECT COUNT(*) FROM soldedveh;";
$result3 = mysqli_query($conn, $sql3);
while ($row3 = mysqli_fetch_array($result3)) {
  $count3 = $row3[0];
}
?>
<div class="row">
  <div class="col-md-6 mb-4 mb-lg-0 stretch-card transparent">
    <div class="card card-light-blue">
      <div class="card-body">
        <a style="color: white;" href="selled_veh.php">
          <p class="mb-4">SOLD</p>
          <p class="fs-30 mb-2"><?php echo $count3; ?></p>
        </a>
      </div>
    </div>
  </div>
  <?php
  $sql4 = "SELECT COUNT(*) FROM buyedveh;";
  $result4 = mysqli_query($conn, $sql4);
  while ($row4 = mysqli_fetch_array($result4)) {
    $count4 = $row4[0];
  }
  ?>
  <div class="col-md-6 stretch-card transparent">
    <div class="card card-light-danger">
      <div class="card-body">
        <a style="color: white;" href="buyed_veh.php">
          <p class="mb-4">BOUGHT</p>
          <p class="fs-30 mb-2"><?php echo $count4; ?></p>
        </a>
      </div>
    </div>
  </div>
</div>
</div>
</div>
</div>
</div>

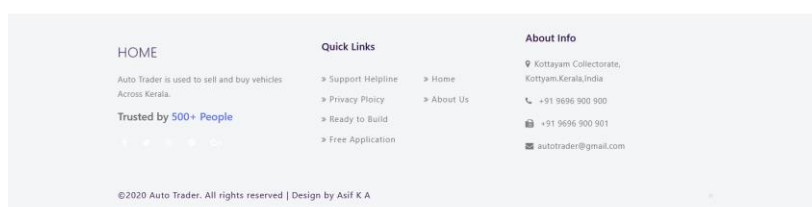
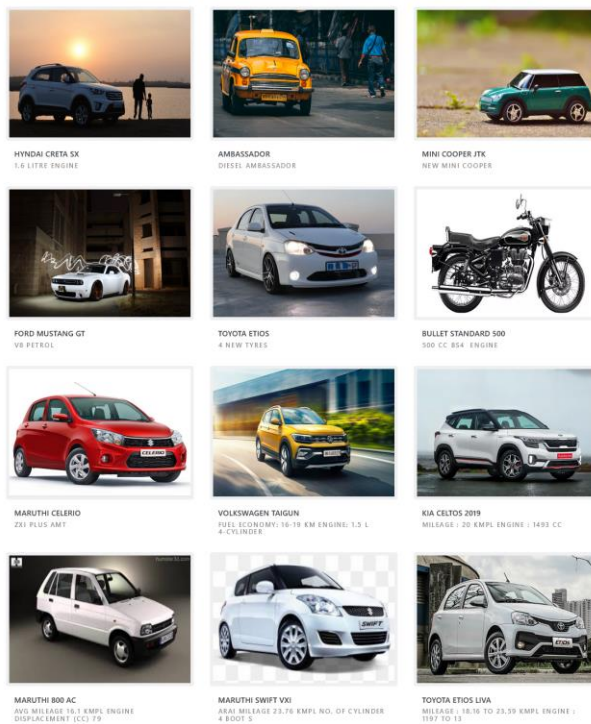
```

## 9.2 Screen Shots

### Home page



#### VEHICLE'S



## About Us page



### Contact Us

Whether you're an individual looking to sell your car, an organisation interested in talking about offloading a fleet of vehicles or would just like to find out more about what we do and how we could work together, get in touch using the details below.

#### Address

📍 Kottayam Collectorate, Kottayam,Kerala,India

#### Mobile

☎ +91 9696 900 900



#### HOME

Auto Trader is used to sell and buy vehicles Across Kerala.

Trusted by **500+ People**



#### Quick Links

- » Support Helpline
- » Privacy Policy
- » Ready to Build
- » Free Application
- » Home
- » About Us

#### About Info

📍 Kottayam Collectorate, Kottayam,Kerala,India


☎ +91 9696 900 900

☎ +91 9696 900 901

✉ autotrader@gmail.com

©2020 Auto Trader. All rights reserved | Design by Asif K A

## User Home page



Home

Profile

Filter

Sell a vehicle


Vehicle

Requests

Notifications

Wishlist


Review

**Asif**


WELCOME!

Logout


### VEHICLE'S




**Mini Cooper JTK**  
K M Driven :6000  
Price : 80000




**Ford Mustang GT**  
K M Driven :5000  
Price : 8000000




**Toyota Etios**  
K M Driven :50000  
Price : 5000060




**Bullet Standard 500**  
K M Driven :35000  
Price : 180000




**Volkswagen Taigun**  
K M Driven :4999  
Price : 1800000



**Kia Celtos 2019**  
K M Driven :36000  
Price : 900000



**Maruthi 800 Ac**  
K M Driven :892021  
Price : 250000




**Toyota Etios Liva**  
K M Driven :374998  
Price : 500000

© 2022 Auto Trader

Vehicle's About Us Blog Licensed

## User Profile page



Home

Profile

Filter

Sell a vehicle

Vehicle

Requests


Notifications

Wishlist

Review

Hello Asif K A

This is your profile page. You can see the your details here's



Asif K A

Address : Kizhakekara

DOB : 1981-12-14

Mob : 9495170060

UPDATE PROFILE PIC

USER INFORMATION

Full name

Asif K A

Gender

Male

Address

Kizhakekara

Locality

Kuruvamoozhy

City

Erumely

District

Kottayam

State

Kerala

Date of Birth

14 / 12 / 1981

Mobile Number

9495170060

UPDATE

© 2022 Auto Trader

Vehicle's

About Us

Blog

Licensed

## Vehicle Filter page

Automobile

Search.....

Asif

Home

Profile

Filter

Sell a vehicle

Vehicle

Requests

Notifications

Wishlist

Review

BRAND

BRAND

Vehicle Between

Start DATE

dd / mm / yyyy

End DATE

dd / mm / yyyy

KM Driven Under

Price Between

Starting Price

Ending Price

Fuel

--SELECT--

Transmission

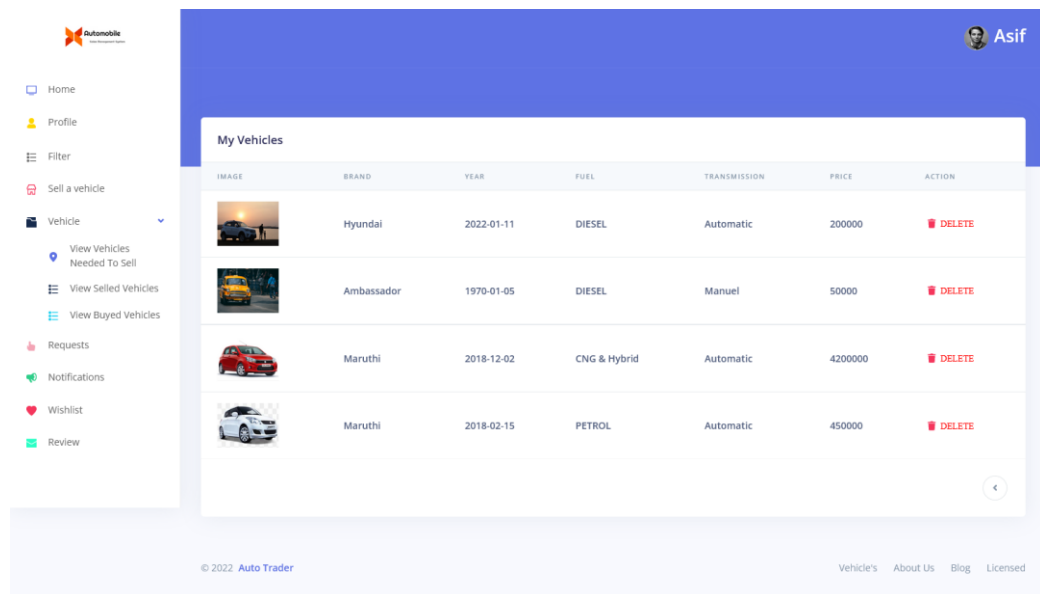
--SELECT--

Submit

© 2022 Auto Trader

Vehicles About Us Blog Licensed

## View vehicle Needed To Sell Page



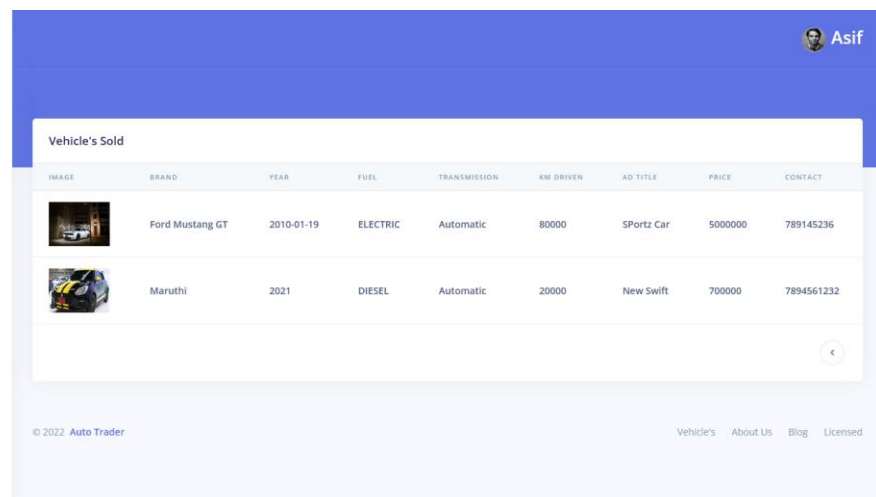
**My Vehicles**

IMAGE	BRAND	YEAR	FUEL	TRANSMISSION	PRICE	ACTION
	Hyundai	2022-01-11	DIESEL	Automatic	200000	DELETE
	Ambassador	1970-01-05	DIESEL	Manuel	50000	DELETE
	Maruthi	2018-12-02	CNG & Hybrid	Automatic	4200000	DELETE
	Maruthi	2018-02-15	PETROL	Automatic	450000	DELETE

© 2022 Auto Trader

Vehicle's About Us Blog Licensed

## View Solded vehicle page



**Vehicle's Sold**

IMAGE	BRAND	YEAR	FUEL	TRANSMISSION	KM DRIVEN	AD TITLE	PRICE	CONTACT
	Ford Mustang GT	2010-01-19	ELECTRIC	Automatic	80000	SPortz Car	5000000	789145236
	Maruthi	2021	DIESEL	Automatic	20000	New Swift	700000	7894561232

© 2022 Auto Trader

Vehicle's About Us Blog Licensed

## View Bought page

Automobile

Home

Profile

Filter

Sell a vehicle

Vehicle

View Vehicles Needed To Sell

View Sold Vehicles

View Bought Vehicles

Requests

Notifications

Wishlist

Review

Vehicles Bought

Record Not Found...

© 2022 Auto Trader

Vehicle's About Us Blog Licensed

## View Request page

Automobile

Home

Profile

Filter

Sell a vehicle

Vehicle

Requests

Notifications 1

Wishlist

Review

Request's

IMAGE	BRAND	YEAR	KM DRIVEN	AD TITLE	DESCRIPTION	PRICE	ACTION
	Mini Cooper	2022-01-04	6000	Mini Cooper JTK	New Mini Cooper	80000	REMOVE

© 2022 Auto Trader

Vehicle's About Us Blog Licensed

## View Notification page

Automobile

Home

Profile

Filter

Sell a vehicle

Vehicle

Requests

Notifications 1

Wishlist

Review

Notification's

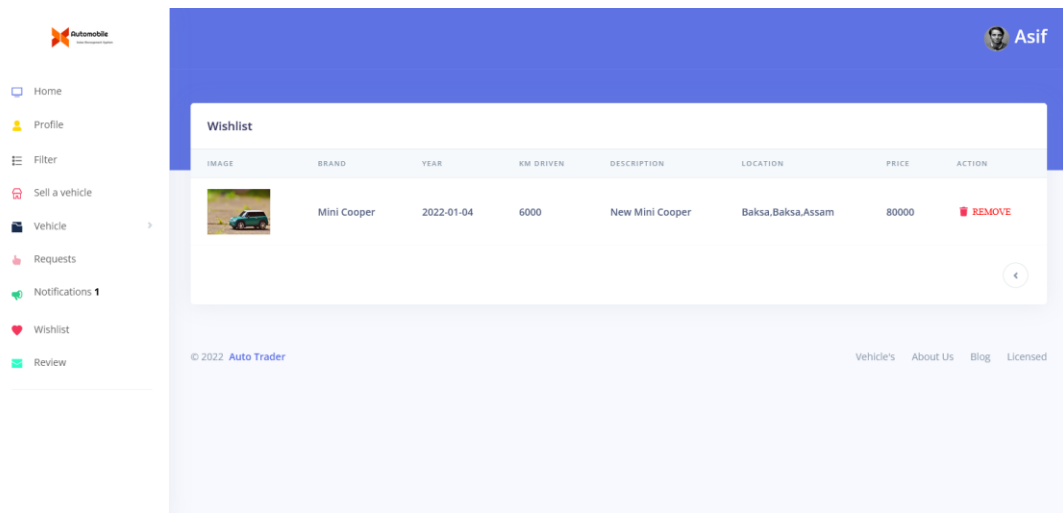
IMAGE	BRAND	YEAR	AD TITLE	DESCRIPTION	PRICE	USER REQUESTED	ACTION
	Hyundai	2022-01-11	Hyndal Creta SX	1.6 Litre engine	200000	ashik@gmail.com	Accept Reject

© 2022 Auto Trader

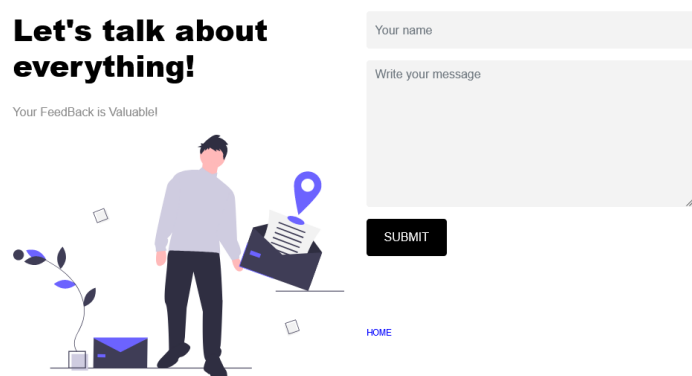
Vehicle's About Us Blog Licensed



## View Wishlist page



## Feedback page



## Admin Home page

... Search now admin

Dashboard

Users >

Vehicle >

Sale's Reports

Feedback

Welcome admin  
All systems are running smoothly!

31°C Bangalore India

Number of Users  
4

Total Vehicle  
12

SOLD  
2

BOUGHT  
2

Reviews

Logout

Copyright © 2021. Premium Auto Trade. All rights reserved.

## Admin View User Details page

... Search now admin

Dashboard

Users >

Users Details

Vehicle >

Sale's Reports

Feedback

Welcome Admin  
All systems are running smoothly!

User's Details

Name	Username	Address	Date of Birth	Gender	Mobile Number	Action
Asif K A	asifka9116@gmail.com	Kizhakekara,Kuruvamoozhy,Kottayam,Kerala	1981-12-14	Male	9495170060	REMOVE
ram	ram@gmail.com	KK,Koovapally,Kottayam,Kerala	0000-00-00	Male	8211147963	REMOVE
John Connor	johnconnor@gmail.com	24 Street,North Avenue,Brooklyn,New York	1994-01-10	Male	+98 7852 125456	REMOVE
Ashik K A	ashik@gmail.com	KK,Koratty,Kottayam,kerala	1986-02-04	Male	9846284006	REMOVE

Copyright © 2021. Premium Auto Trade. All rights reserved.

## Admin View Vehicles To Sell Details page

... Search now admin

Dashboard Users > Vehicle

- Vehicle's On Sale
- Selled Vehicle's
- Buoyed Vehicle's

Sale's Reports Feedback

### Welcome Admin

All systems are running smoothly!

#### Vehicle's On Sale

Image	Title	Brand	Year	Fuel	Transmission	KM Driven	Sale's Price	Username
	Hyndai Creta SX	Hyundai	2022-01-11	DIESEL	Automatic	5000	200000	asifka9116@gmail.com
	Ambassador	Ambassador	1970-01-05	DIESEL	Manuel	1000056	50000	asifka9116@gmail.com
	Mini Cooper JTK	Mini Cooper	2022-01-04	ELECTRIC	Automatic	6000	80000	ram@gmail.com
	Ford Mustang GT	Ford	2022-01-04	DIESEL	Automatic	5000	8000000	ram@gmail.com
	Toyota Etios	Toyota	2018-12-05	DIESEL	Automatic	50000	5000060	ram@gmail.com
	Bullet Standard 500	Royal Enfield	2018-01-15	PETROL	Manuel	35000	180000	ram@gmail.com
	Maruthi Celerio	Maruthi	2018-12-02	CNG & Hybrid	Automatic	420000	4200000	asifka9116@gmail.com
	Volkswagen Taigun	Volkswagen	2022-02-07	PETROL	Automatic	4999	1800000	ashik@gmail.com
	Kia Celta 2019	Kia	2019-02-11	DIESEL	Automatic	36000	900000	ashik@gmail.com
	Maruthi 800 Ac	Maruthi	1999-01-13	PETROL	Manuel	892021	250000	ashik@gmail.com
	Maruthi Swift VXi	Maruthi	2018-02-15	PETROL	Automatic	230000	450000	asifka9116@gmail.com
	Toyota Etios Liva	Toyota	2018-02-14	PETROL	Automatic	374998	500000	ashik@gmail.com

Copyright © 2021. Premium Auto Trade. All rights reserved.

## Admin View Selled Vehicle's page

... Search now admin

Dashboard Users > Vehicle

- Vehicle's On Sale
- Selled Vehicle's
- Buoyed Vehicle's

Sale's Reports Feedback

### Welcome Admin

All systems are running smoothly!

#### Selled Vehicle's

Image	Title	Brand	Year	Fuel	Transmission	KM Driven	Selled Price	Seller's Name
	Ford Mustang GT	SPortz Car	2010-01-19	ELECTRIC	Automatic	80000	5000000	asifka9116@gmail.com
	Maruthi	New Swift	2021	DIESEL	Automatic	20000	700000	asifka9116@gmail.com

Copyright © 2021. Premium Auto Trade. All rights reserved.

## Admin View Buyed Vehicle's page

... Search now admin

Dashboard Users Vehicle Sale's Reports Feedback

**Welcome Admin**  
All systems are running smoothly!

**Buyed Vehicle's**

Image	Title	Brand	Year	Fuel	Transmission	Bought Price	Buyer's Name
	SPortz Car	Ford Mustang GT	2010-01-19	ELECTRIC	Automatic	5000000	ram@gmail.com
	New Swift	Maruthi	2021	DIESEL	Automatic	700000	ram@gmail.com

Copyright © 2021. Premium Auto Trade. All rights reserved.

## Admin View Sale's Report page

... Search now admin

Dashboard Users Vehicle Sale's Reports Feedback

**Welcome Admin**  
All systems are running smoothly!

**Sale's Reports**

Start Date  
dd / mm / yyyy

End Date  
dd / mm / yyyy

SUBMIT

Copyright © 2021. Premium Auto Trade. All rights reserved.

## Admin View Feedback page

... Search now admin

Dashboard Users Vehicle Sale's Reports Feedback

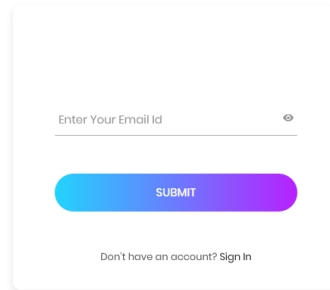
**Welcome Admin**  
All systems are running smoothly!


**User's Feedback**

Name	Message	Username
Ram K P	Very Good Website Vehicle can be bought Affordable cost	ram@gmail.com
Ashik K A	Very good website for Selling and Buying Second hand Vehicle, Genuie Service, Good customer Service Good customer Service	ashik@gmail.com
John Connor	Good customer Service	johnconnor@gmail.com

Copyright © 2021. Premium Auto Trade. All rights reserved.

## User forgot Password page

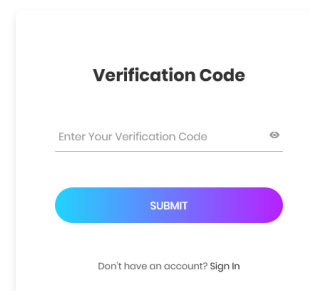


Enter Your Email Id 


**SUBMIT**

Don't have an account? [Sign In](#)

## OTP Verification page



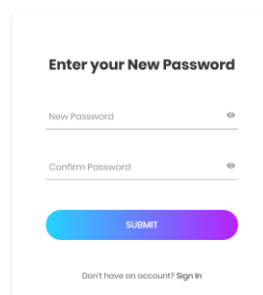
**Verification Code**

Enter Your Verification Code 


**SUBMIT**


Don't have an account? [Sign In](#)

## New Password Set page



**Enter your New Password**

New Password 

Confirm Password 

**SUBMIT**

Don't have an account? [Sign In](#)