CHAPTER 1

INTRODUCTION

1.1 Overview of Database Management Systems

A Data base Management System (DBMS) is a general-purpose software system that allows creation, definition and manipulation of a database, allowing users to store, process and analyze data easily. A Data base Management System (DBMS) provides us with an interface tool, to perform various operations like creating data base, storing data in it, updating data, creating tables in the database and a lot more. Modern Database Management Systems (DBMS) also provide protection and added security features to the databases. In addition, it also maintains data consistency in case of multiple users. Some examples of the most commonly used Database Management Systems are My SQL, ORACLE DB, IBM DB2, Amazon Simple DB, etc.

1.1.1 Characteristics of a Database Management System

- Reduced redundancy of data stored throughout the database with the help of concepts like normalization which divides the data in the database to reduce repeated data.
- Data consistency is maintained throughout the database with data being constantly added, updated and deleted.
- Multiuser support and concurrent access, allows multiple users to work on the database at the same time.
- Simple query language can be uses to easily fetch, insert, update and delete data from the database.
- Security is built in which restricts unauthorized access to the database. Different users have their own associated permissions.

1.1.2 Advantages of a Database Management System

- Provides data abstraction and segregation of application program from the data.
- Reduced redundancy of data ensures maximum cost efficiency for the storage of data

- Reduced development time building applications that use database.

1.2 Problem statement

Vehicle Management System is developed and customized for commercial fleet owners and organizations. It really reduces your vehicles search time. Eliminate costly unscheduled maintenance. Keep accurate records for any type of vehicle. Help you plan annual vehicle budgets faster, easier and more accurately Keep track of Party Ledgers Payment Receipt Keep track of customers and other employee ledgers. Will be able to create new users and change the passwords. He can add or update the employee information in the company. He can add and view the information of customers and categories.

1.3 Objectives

Depending on the user demands. It is every company's need to see the welfare of its employees by providing easy user interface for the employees. Users can check their vehicle information after ordering the vehicle. Special feature of this management is admin can manage employees and can also add customers and place orders for them simultaneously. External links like Facebook, twitter and Instagram are provided for extra interaction with the management.

1.4 Dataset Description

Given below are the entities along with its attributes and relations present in the database of this application that are used to retrieve information from the database as per requirement of user.

- An entity called CUSTOMER is created with the c_id being the Main Key Attribute also known as the Primary key. The other Attributes belonging to the Entity is password.
- The Entity MANUFACTURER is created with the Manufacturer_id being the Primary key. The other attributes are Manufacturer_Name, Manufacturer_logo.
- The Entity MODEL consists of the Model id acting as a key attribute, Manufacture_name (referenced from the entity MANUFACTURER), Model_name.
- The Entity USERS consists of the User id acting as the primary key and the U_email,f_Name,l_name,U_bday,u_position,u_type,u_pass,u_mobile,u_gender,u_gender,u_address,u_question,u_ans.

- The Entity VEHICLE consists of v_id as the primary key. It also involves other attributes such as Manufacturer_name (referenced from entity MANUFACTURER),model_name (referenced from entity MODEL),Category,b_price,s_price,mileage,add_date,sold_date,status,registration_year,insurance_id,gear,doors,seat,tank,image,e_no,c_no,u_id,v_color.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 Software and Hardware Requirements

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application.

2.1.1 Software Requirements

2.1.1.1 Front End

- HTML5/CSS
- Bootstrap Boilerplate
- JavaScript
- Google Chrome (Web Browser)

2.1.1.2 Back End

- XAMPP Server (v3.2.2) for Apache Server (local host)
- My SQL (v8.0.12) for Database Management System
- PHP (v7.2.10) for Server-Side Scripting
- Sublime (Source Code Editor)
- Windows 7 and above

2.1.2 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware.

- CPU: Intel or AMD processor
- Cores: Dual-Core (Quad-Core recommended)
- RAM: minimum 1GB (>1GBrecommended)

- Graphics: Intel Integrated Graphics or AMD Equivalent
- Secondary Storage:25G

CHAPTER 3

SYSTEM DESIGN

3.1 Entity Relationship Diagram

An Entity Relationship Diagram describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between instances of those entity types. In software engineering, an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure which can be implemented in a database, typically a relational database

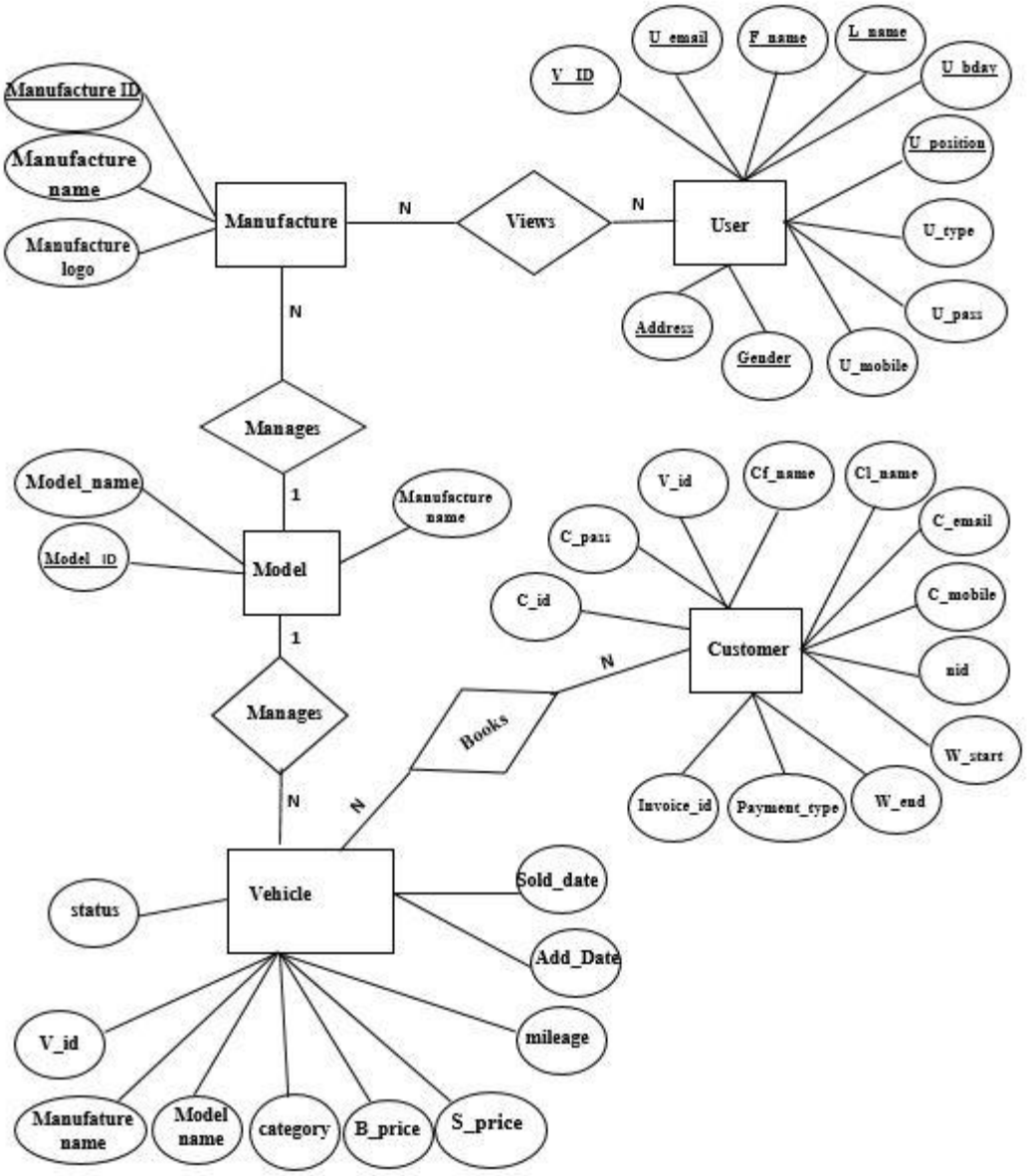


Figure 3.1 Entity Relationship Diagram

3.2 Schema Database Relationship Diagram

A database schema is the skeleton structure that represents the logical view of the entire database. It formulates all the constraints that are to be applied on the data. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams.

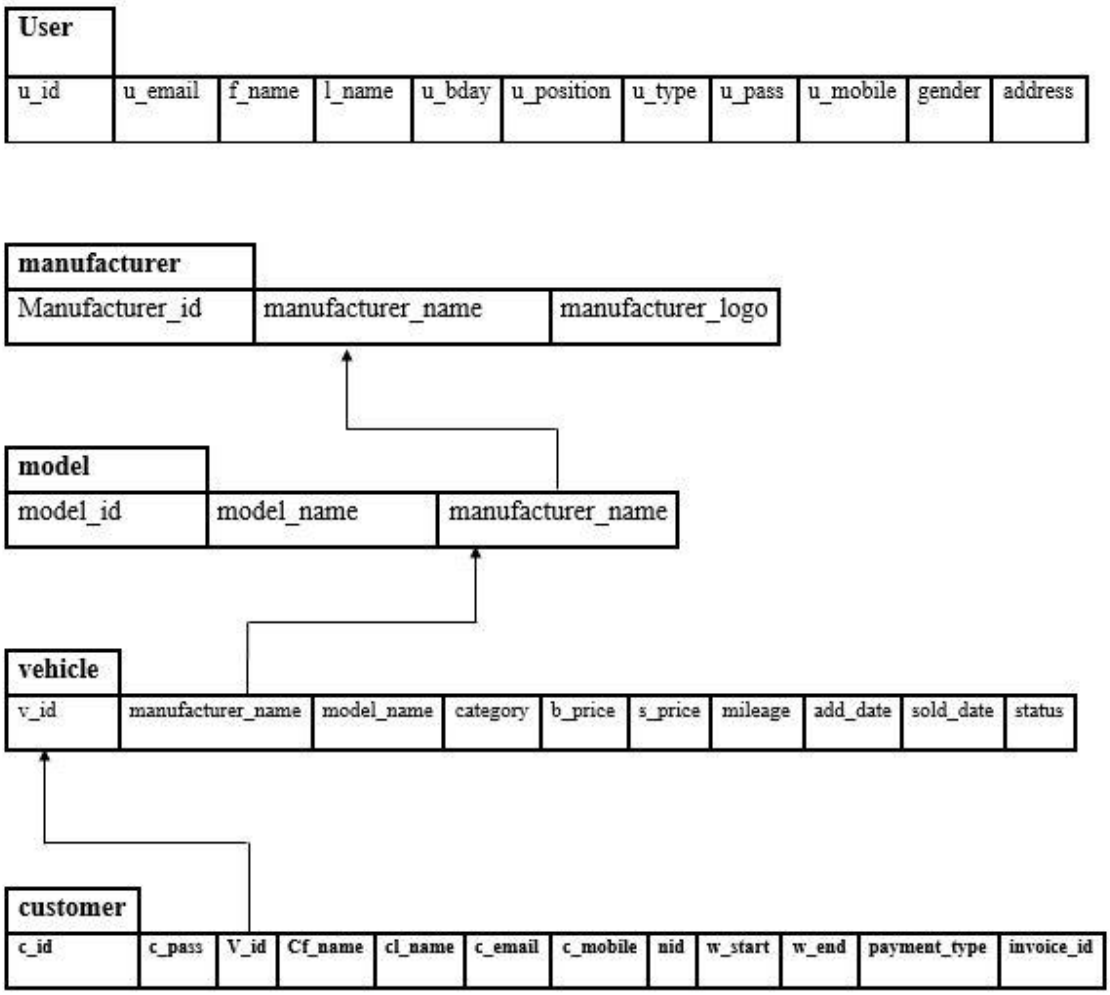


Figure 3.2 Schema Database Relationship Diagram

3.3 Overview of GUI

The **graphical user interface** (**GUI**), is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.

The actions in a GUI are usually performed through direct manipulation of the graphical elements. Beyond computers, GUI's are used in many handheld mobile devices such as MP3 players, portable media players, gaming devices, smartphones and smaller household, office and industrial controls. The term GUI tends not to be applied to other

lower-display resolution types of interfaces, such as video games, or not including flat screens able to describe generic information, in the tradition of the computer science research at the Xerox Palo Alto Research Centre.

The following buttons have been used in my project:

1. Menu Bar

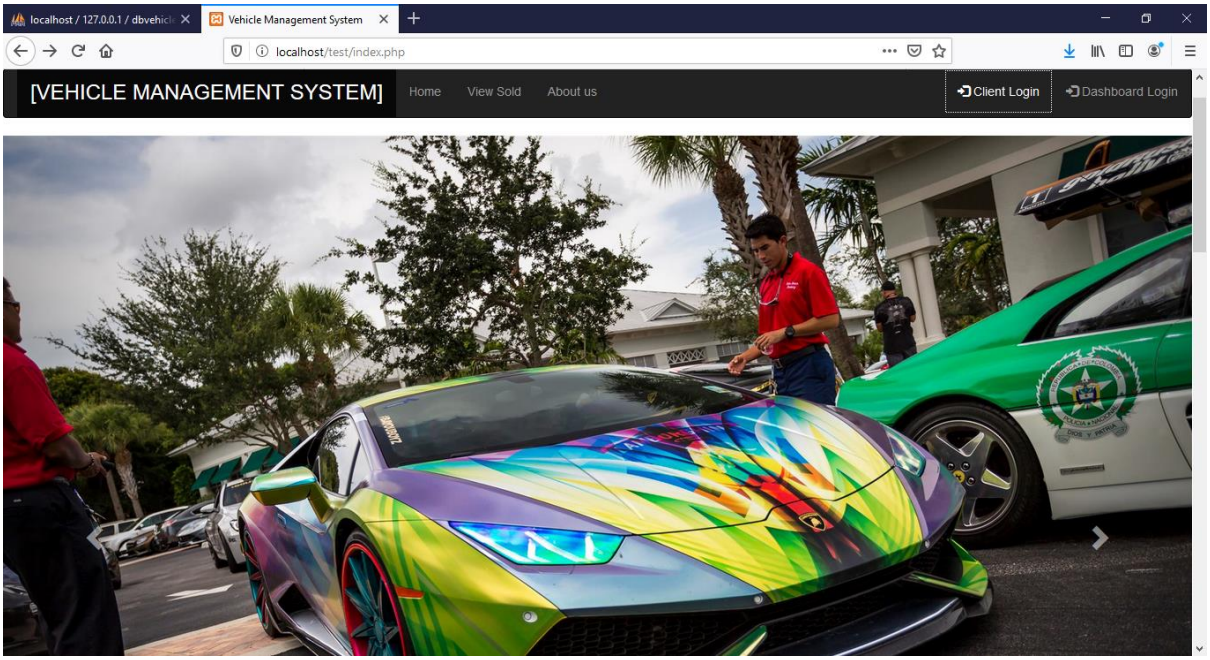


Fig.3.3: Menu Bar

2. Submit Button

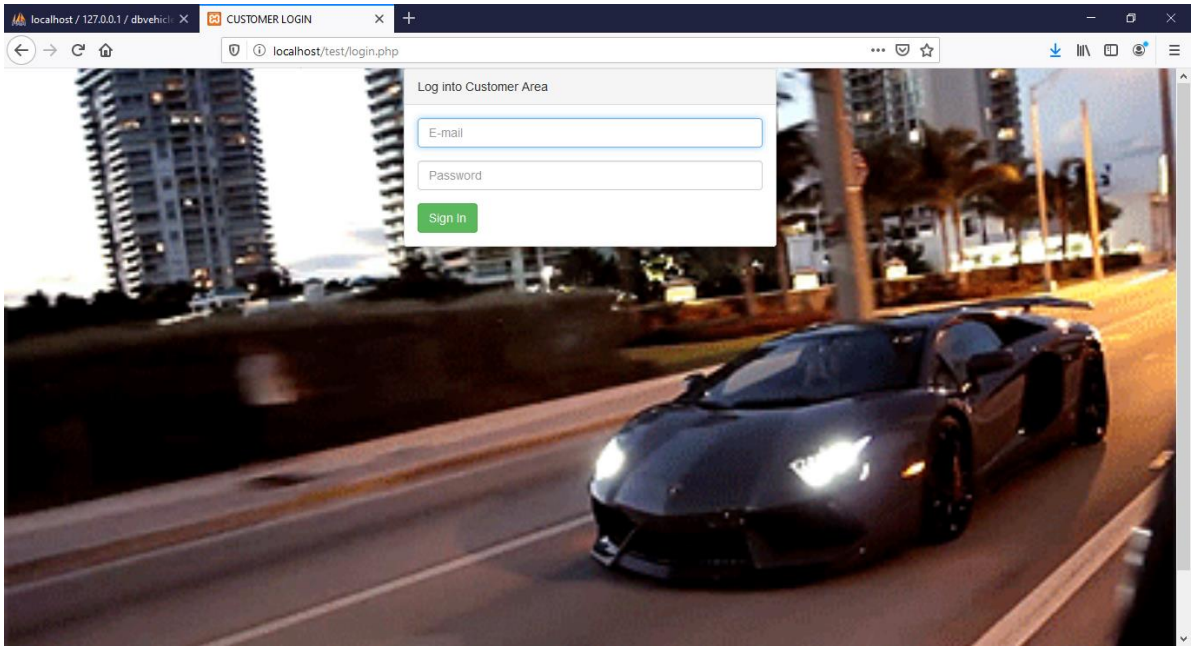


Fig.3.4: Submit Button

3.4 Normalization

Normalization is a process of analysing the given relation schema based on their functional dependencies and primary key to achieve desirable properties of minimizing redundancy and minimising insert, delete, update anomaly. The normalization process takes a relation schema through a series of tests to certify whether it satisfies a certain normal form. The normal form of a relation refers to the highest normal form condition that it meets, and hence the degree to which it has been normalized.

There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

Database Normalization Examples:

Assume a model database, in which different purposes are there. Without any normalization all the information stored in one table is as shown below.

Id	Name	type	Phone no	Address
1	Rahul	User, Employee	9591104193	Chintamani
2	Gishnu	User, Employee	9697779898	Bannerghatta
3	Nikhil	User, Employee	9845967891	Isro layout

Table 3.1

3.4.1 1NF (First Normal Form):

1NF Rules:

- Each table cell should contain a single value.
- Each record needs to be unique.

It is seen that in the above table the column course branch is an multi-valued attribute. So in order to be in 1NF form we have to split the table like below.

The below table in 1NF:

Id	Name	type	Phone no	Address
1	Rahul	User	9591104193	Chintamani
2	Rahul	Employee	9591104193	Chintamani
3	Gishnu	User	9697779898	Bannerughatta
4	Gishnu	Employee	9697779898	Bannerughatta
5	Nikhil	Employee	9845967891	Isro layout

Table 3.2

3.4.2 2NF (Second Normal Form):

2NF Rules:

- Rule 1- Be in 1NF.
- Rule 2- Single Column Primary Key.

It is clear that we cannot move forward to make our simple database in 2nd Normalization form unless we partition the table above:

Employee id	Name	Phone no	Address
1	Rahul	9591104193	Chintamani
2	Gishnu	9697779898	Bannerughatta
3	Nikhil	9845967891	Isro layout

Table 3.3

Employee id	type
1	User
1	Employee
2	User
2	Employee
3	Employee

Table 3.4

3.4.3 3NF (Third Normal Form):

3NF Rules:

- Rule 1- Be in 2NF.
- Rule 2- Has no transitive functional dependencies.

A transitive functional dependency is when changing a non-key column, might cause any of the other non-key columns to change.

To move 2NF table into 3NF, we again need to divide the table.

Employee id	Name	Manufacturername	Phoneno	Address
1	Rahul	BMW	9591104193	Chintamani
2	Gishnu	FORD	9591104193	Bannerughatta
3	Rahul	BMW	9697779898	Chintamani
4	Gishnu	FORD	9697779898	Bannerughatta
5	Gishnu	FORD	9845967891	Bannerughatta

Table 3.5

Manufacturername	type
BMW	User
FORD	Employee

Table 3.6

CHAPTER 4

IMPLEMENTATION

4.1 Table Creation

Customer:

CREATE TABLE customer (

c_id int(11) NOT NULL,

v_id int(11) NOT NULL,

cf_name varchar(100) NOT NULL,

cl_name varchar(100) NOT NULL,

c_email varchar(100) NOT NULL,

c_mobile varchar(100) NOT NULL,

nid varchar(100) DEFAULT NULL,

w_start date NOT NULL,

w_end date NOT NULL,

payment_type varchar(100) NOT NULL,

invoice_id varchar(100) DEFAULT NULL,

c_address varchar(400) DEFAULT NULL,

c_pass varchar(30) NOT NULL,

extra varchar(300) DEFAULT NULL);

Manufacturer:

```
CREATE TABLE manufacturer (
manufacturer_id int(11) NOT NULL,
manufacturer_name varchar(100) NOT NULL,
manufacturer_logo varchar(300) DEFAULT NULL);
```

Model:

```
CREATE TABLE model (
model_id int(11) NOT NULL,
model_name varchar(100) NOT NULL,
manufacturer_name varchar(100) NOT NULL);
```

Users:

```
CREATE TABLE users (
u_id int(11) NOT NULL,
u_email varchar(100) NOT NULL,
f_name varchar(100) NOT NULL,
l_name varchar(100) NOT NULL,
u_bday date NOT NULL,
u_position varchar(100) NOT NULL,
u_type varchar(100) NOT NULL,
u_pass varchar(100) NOT NULL,
u_mobile varchar(100) NOT NULL,
u_gender varchar(30) NOT NULL,
u_address varchar(100) NOT NULL,
s_question varchar(100) DEFAULT NULL,
s_ans varchar(100) DEFAULT NULL);
```

Vehicle:

```
CREATE TABLE vehicle (
 v_id int(11) NOT NULL,
 manufacturer_name varchar(100) NOT NULL,
model_name varchar(100) NOT NULL,
category varchar(100) NOT NULL,
 b_price double NOT NULL,
 s_price double DEFAULT NULL,
 mileage double NOT NULL,
  add_date date NOT NULL,
  sold_date date DEFAULT NULL,
  status varchar(40) NOT NULL,
  registration_year int(11) NOT NULL,
  insurance_id int(11) DEFAULT NULL,
  gear varchar(100) NOT NULL,
  doors int(11) NOT NULL,
  seats int(11) NOT NULL,
  tank float NOT NULL,
  image varchar(400) DEFAULT NULL,
  e_no varchar(40) NOT NULL,
```

c_no varchar(40) NOT NULL,

u_id int(11) DEFAULT NULL,

v_color varchar(20) DEFAULT NULL);

4.2 Description of Tables

Customer: desc customer;

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	c_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	v_id	int(11)			No	None			Change Drop More
☐	3	cf_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	4	cl_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	5	c_email	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	6	c_mobile	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	7	nid	varchar(100)	latin1_swedish_ci		Yes	None			Change Drop More
☐	8	w_start	date			No	None			Change Drop More
☐	9	w_end	date			No	None			Change Drop More
☐	10	payment_type	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	11	invoice_id	varchar(100)	latin1_swedish_ci		Yes	None			Change Drop More
☐	12	c_address	varchar(400)	latin1_swedish_ci		Yes	None			Change Drop More
☐	13	c_pass	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
☐	14	extra	varchar(300)	latin1_swedish_ci		Yes	None			Change Drop More

Fig.4.1: Customer details

Manufacturer: desc manufacturer;

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	manufacturer_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	manufacturer_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	3	manufacturer_logo	varchar(300)	latin1_swedish_ci		Yes	None			Change Drop More

Fig.4.2: Manufacturer details

Model: desc model;

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	model_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	model_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	manufacturer_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More

Fig.4.3: Model details

Users: desc users;

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	u_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	u_email	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	f_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
4	l_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
5	u_bday	date			No	None			Change Drop More
6	u_position	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
7	u_type	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
8	u_pass	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
9	u_mobile	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
10	u_gender	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
11	u_address	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
12	s_question	varchar(100)	latin1_swedish_ci		Yes	None			Change Drop More
13	s_ans	varchar(100)	latin1_swedish_ci		Yes	None			Change Drop More

Fig.4.4: Users

Vehicle: desc vehicle;

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	v_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	manufacturer_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	3	model_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	4	category	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	5	b_price	double			No	None			Change Drop More
☐	6	s_price	double			Yes	None			Change Drop More
☐	7	mileage	double			No	None			Change Drop More
☐	8	add_date	date			No	None			Change Drop More
☐	9	sold_date	date			Yes	None			Change Drop More
☐	10	status	varchar(40)	latin1_swedish_ci		No	None			Change Drop More
☐	11	registration_year	int(11)			No	None			Change Drop More
☐	12	insurance_id	int(11)			Yes	None			Change Drop More
☐	13	gear	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	14	doors	int(11)			No	None			Change Drop More
☐	15	seats	int(11)			No	None			Change Drop More
☐	16	tank	float			No	None			Change Drop More
☐	17	image	varchar(400)	latin1_swedish_ci		Yes	None			Change Drop More
☐	18	e_no	varchar(40)	latin1_swedish_ci		No	None			Change Drop More
☐	19	c_no	varchar(40)	latin1_swedish_ci		No	None			Change Drop More
☐	20	u_id	int(11)			Yes	None			Change Drop More

Fig.4.5: Vehicle

4.3 Populated Tables

Customer:

Select * from customer;

	c_id	v_id	cf_name	cl_name	c_email	c_mobile	nid	w_start	w_end	payment_type	invoice_id	c_address	c_pass	extra
Delete	3	110	shmee	150	shmee@gmail.com	9874563214	12121	2019-11-15	2021-02-15	Cash	#IE9S110S	mount road,chennai-90	mycar	racing body kit
Delete	4	114	rahul	gn	gn@gmail.com	321564978	9	2019-11-16	2023-01-02	Cash	#IE9S114S	chinthamani	11111111	good ownder
Delete	5	115	nikhil	gowdru	nikhilgp@gmail.com	9874565478	999	2019-11-16	2020-02-01	Cash	#IE9S115S	isro lyout	nikhilgp	vinyls added
Delete	6	112	gishnu	govind	gishnu@gmail.com	9874563214	100	2019-11-16	2020-01-18	Cheque	#IE9S112S	bannerughatta	11111111	exhaust

Fig.4.6: customer

Manufacturer:

Select * from manufacturer;

	manufacturer_id	manufacturer_name	manufacturer_logo
Edit Copy Delete	30	BMW	NULL
Edit Copy Delete	33	Newww	NULL
Edit Copy Delete	35	oasdad	NULL
Edit Copy Delete	36	porsche	NULL
Edit Copy Delete	37	lamborghini	NULL
Edit Copy Delete	38	audi	NULL
Edit Copy Delete	39	tesla	NULL
Edit Copy Delete	40	ford	NULL

Fig.4.7: manufacturer

Model:

Select * from model;

	model_id	model_name	manufacturer_name
Edit Copy Delete	27	JXER	BMW
Edit Copy Delete	29	Laxus	BMW
Edit Copy Delete	30	911 turbo	porsche
Edit Copy Delete	31	huracan	lamborghini
Edit Copy Delete	32	r8	audi
Edit Copy Delete	33	roadster	tesla
Edit Copy Delete	34	mustang	ford

Fig.4.8: model

Users:

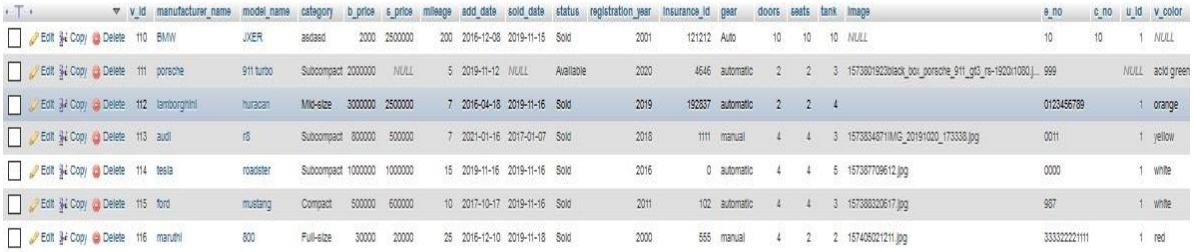
Select * from users;

u_id	u_email	f_name	l_name	u_bday	u_position	u_type	u_pass	u_mobile	u_gender	u_address	s_question	s_ans
1	admin@admin.com	admin	admin	2016-04-14	Manager	Admin	admin	1	Male	test	test?	test
9	abcd@abc.com	ffff	asdasd	2015-11-30	iwskwk	Employee	12345678	00202	Male	kkasd	NULL	NULL
14	nayansurya46@gmail.com	nayan	surya	2020-02-01	customer	Employee	11111111	09564512547	Male	east end a main road	What is your 1st Phone No?	987456321
15	kerala@gmail.com	aaa	bbb	2014-01-01	customer	Employee	zxcvbnm	3256987451	Female	100,33rd cross,st.pattric church	Who is your 1st teacher?	mother
16	motor@gmail.com	shahid	khan	2024-02-01	customer	Employee	motormind	4567891235	Male	mg road bengaluru	What is your 1st Phone No?	457896435
17	ram@yahoo.com	ram	sam	2017-09-30	customer	Employee	11111111	2355789451	Male	ayodhya	What is your 1st Phone No?	987456213
18	bnmit@hotmail.com	bnm	it	2025-03-29	customer	Employee	00000000	2354789456	Female	banashankari,bangalore	Who is your 1st teacher?	mother
19	aa@gmail.com	a	b	2012-10-30	customer	Employee	123123123	7896545654	Female	bangalore	Who is your 1st teacher?	mother

Fig.4.9: Users

Vehicle:

Select * from vehicle;



	v_id	manufacturer_name	model_name	category	b_price	s_price	mileage	add_date	sold_date	status	registration_year	insurance_id	gear	doors	seats	tank	image	e_no	c_no	u_id	v_color
Edit Copy Delete	110	BMW	JXER	asdasd	2000	2600000	200	2016-12-08	2019-11-15	Sold	2001	121212	Auto	10	10	10	NULL	10	10	1	NULL
Edit Copy Delete	111	porsche	911 turbo	Subcompact	2000000	NULL	5	2019-11-12	NULL	Available	2020	4646	automatic	2	2	3	1573801923Iblack_boi_porsche_911_gt3_rs-1920x1080.j...	999		NULL	acid green
Edit Copy Delete	112	lamborghini	huracan	Mid-size	3000000	2500000	7	2016-04-18	2019-11-16	Sold	2019	192837	automatic	2	2	4		0123456789		1	orange
Edit Copy Delete	113	audi	r8	Subcompact	800000	500000	7	2021-01-16	2017-01-07	Sold	2018	1111	manual	4	4	3	1573834571IMG_20191020_173338.jpg	0011		1	yellow
Edit Copy Delete	114	tesla	roadster	Subcompact	1000000	1000000	15	2019-11-16	2019-11-16	Sold	2016	0	automatic	4	4	5	1573877099612.jpg	0000		1	white
Edit Copy Delete	115	ford	mustang	Compact	500000	600000	10	2017-10-17	2019-11-16	Sold	2011	102	automatic	4	4	3	1573883320617.jpg	997		1	white
Edit Copy Delete	116	maruthi	800	Full-size	30000	20000	25	2016-12-10	2019-11-18	Sold	2000	555	manual	4	2	2	1574082021211.jpg	33332222111		1	red

Fig.4.10: Vehicle

4.4 SQL triggers & stored procedures

4.4.1 Triggers

Triggers are stored programs, which are automatically executed or fired when some events occur.

- Triggers are, in fact, written to be executed in response to any of the following events: A database manipulation (DML) statement (DELETE, INSERT, or UPDATE)
- A database definition (DDL) statement (CREATE, ALTER, or DROP).
- A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN). Triggers can be defined on the table, view, schema, or database with which the event is associated. The trigger used in this application is used to record the time and date of User Creation by the admin. By knowing the date and time we can have a backup for the date of registration.

The triggers are:

```
CREATE TABLE custdata (
 c_id int(5) NOT NULL,
 cf_name varchar(20) NOT NULL,
 c_address varchar(25) NOT NULL);
```

c_id	cf_name	c_address
6	gishnu	bannerughatta
7	swat	hyderabad

Fig.4.12: trigger

4.4.2 Stored procedure:

A stored procedure is a prepared SQL code that can be saved and can be reused over and over again. So, if a query has to be written over and over again, instead of having to write that query each time, it can be saved as a stored procedure and can be executed just by calling the procedure.

In addition, parameters can also be passed to the stored procedure. So depending on the need, the stored procedure can act accordingly.

Stored procedures are useful in the following circumstances:

- if a database program is needed by several applications, it can be stored at the server and invoked by any of the application programs. This reduces duplication of effort and improves software modularity.
- Executing a program at the server can reduce data transfer and communication cost between the client and server in certain situations.
- These procedures can enhance the modelling power provided by views by allowing, more complex types of derived data to be made available to the database users via the stored procedures. Additionally, they can be used to check for complex constraints that are beyond the specification power of assertions and triggers.

  The stored procedures used in this application are:

  CREATE DEFINER=`root`@`localhost` PROCEDURE `getdata` (IN `id` INT(5)) NO SQL

  select u_email,u_position,u_type

  from users

  where u_id like id$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `model_pro` () NO SQL

SELECT * from model$$Results :

employeeid	machineid	salesdate	amount
101	101	2019-11-01	1000
105	101	2019-11-03	2050
104	101	2019-11-05	1000

Fig.4.14: Stored procedure

4.5 THE DATABASE CONNECTIVITY

The front end can easily be connected to the back end/database (i.e., Mysql)

by adding a few instructions in PHP. The following instructions are to be added.

```
<?php

$db_host        = 'localhost';
$db_user        = 'root';
$db_pass        = ' ';
$db_database  = 'model';

$db = new PDO('mysql:host='.$db_host.';dbname='.$db_database, $db_user, $db_pass);
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

?>
```

CHAPTER 5

RESULTS

1. Homepage

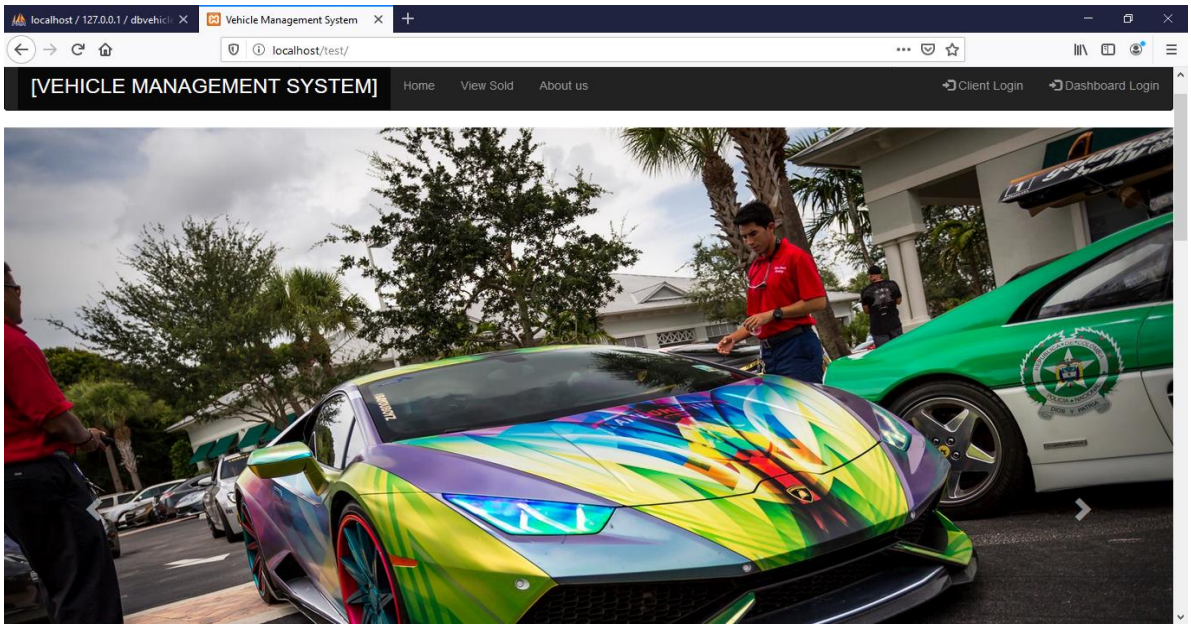


Fig.5.1: Homepage

2. Admin login

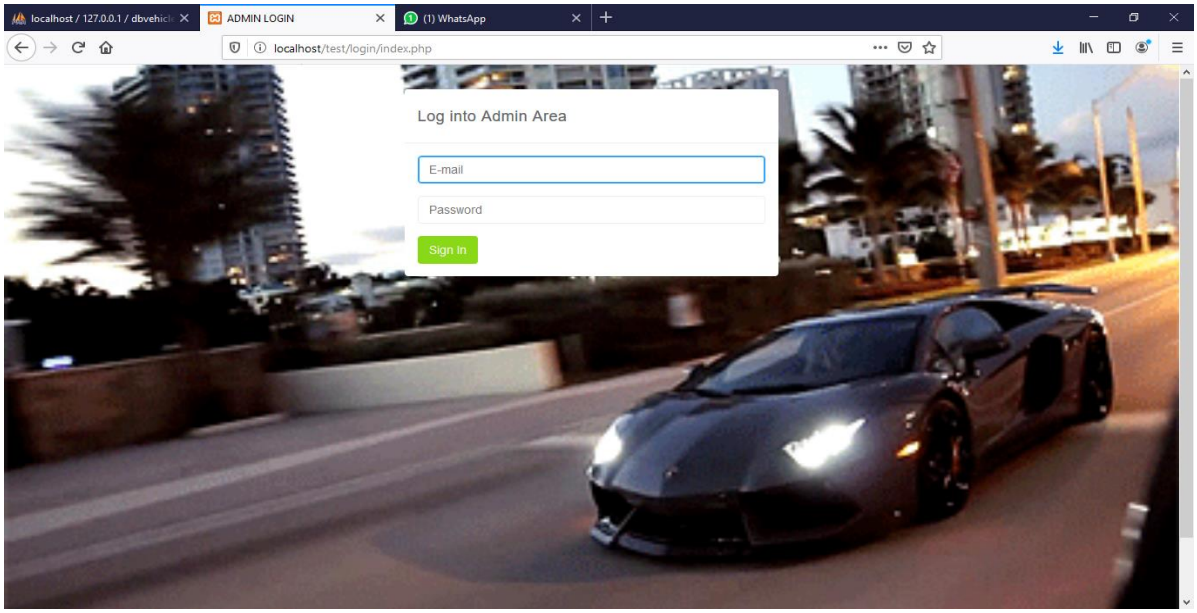


Fig.5.2: Admin Login

2. Client Login

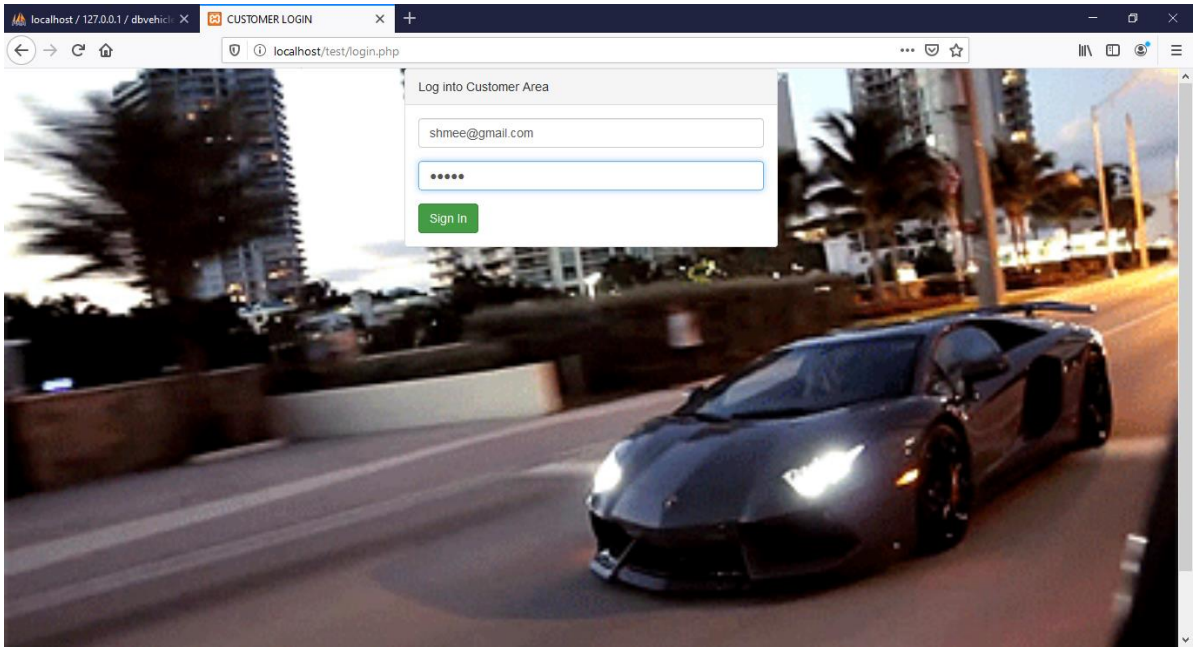


Fig.5.3: Client Login

3. Add employee

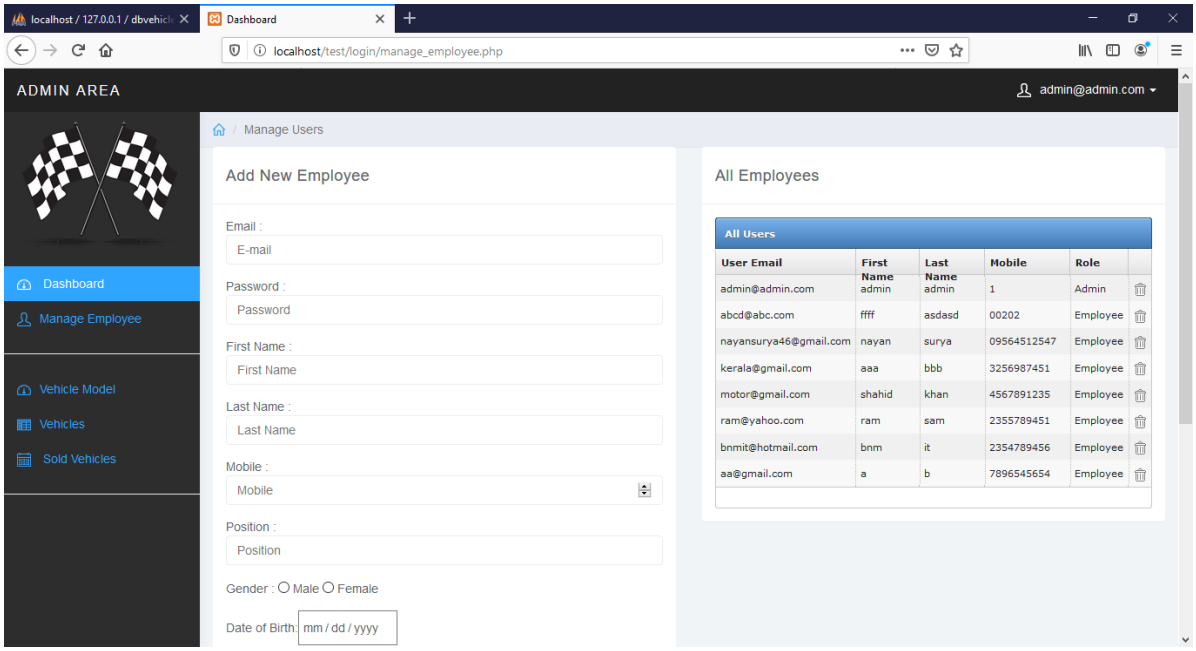


Fig.5.4: Add employee

4. Admin Dashboard

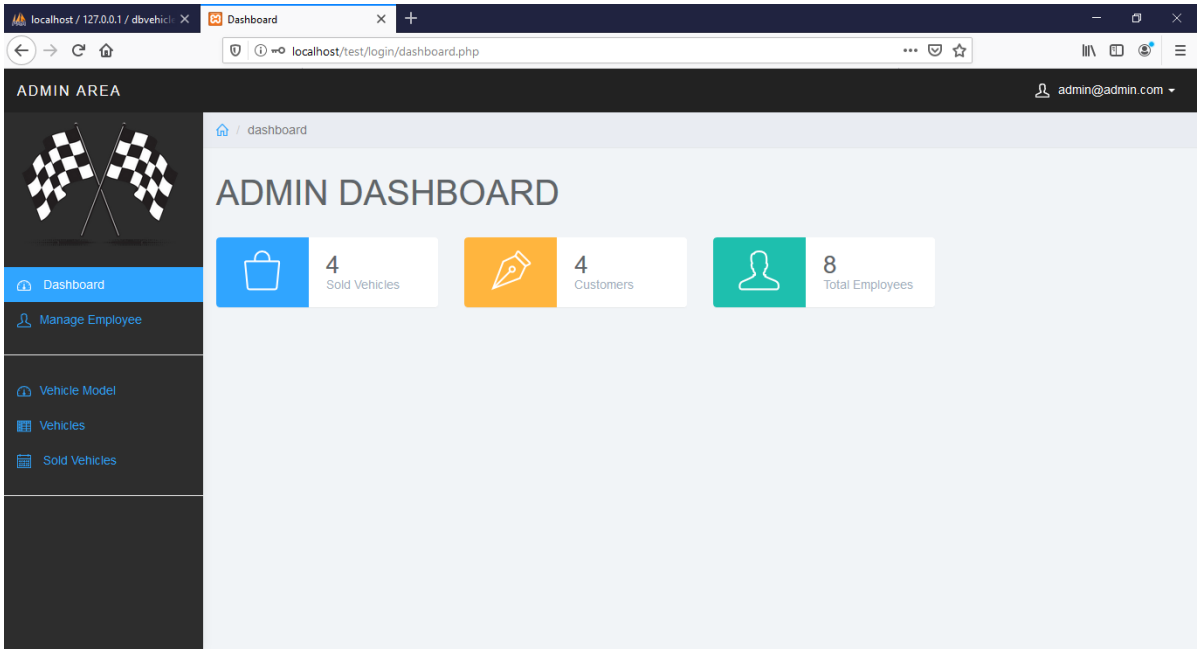


Fig.5.5: Admin Dashboard

5. Latest models

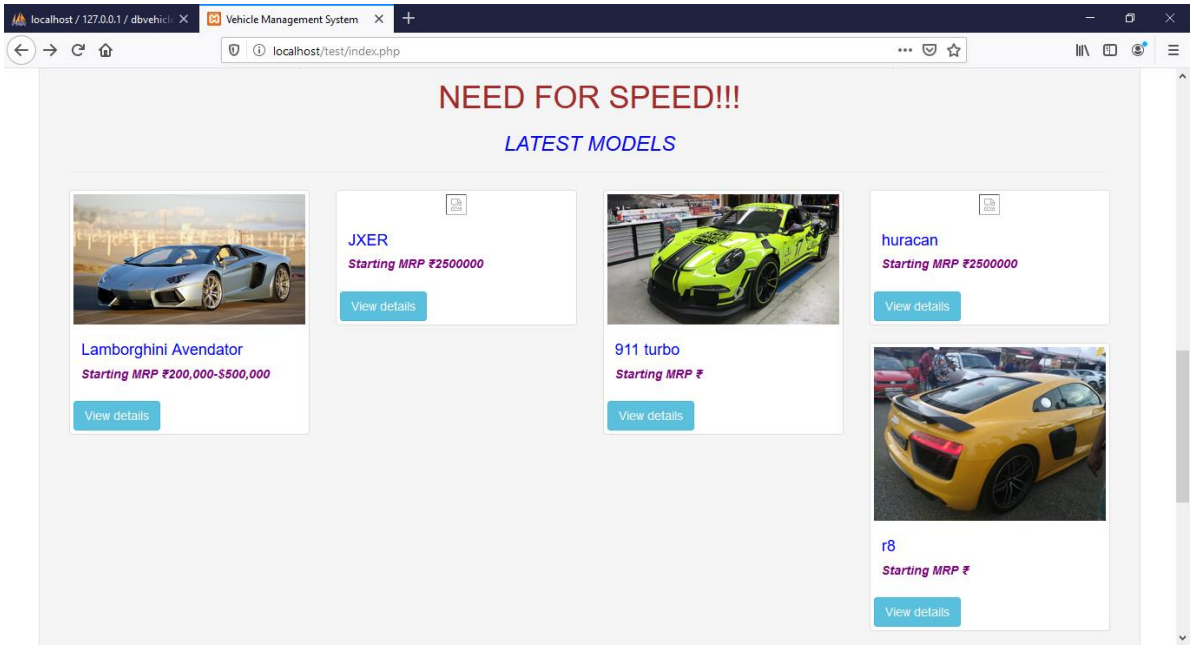


Fig.5.6: Latest models

6. Client vehicle information

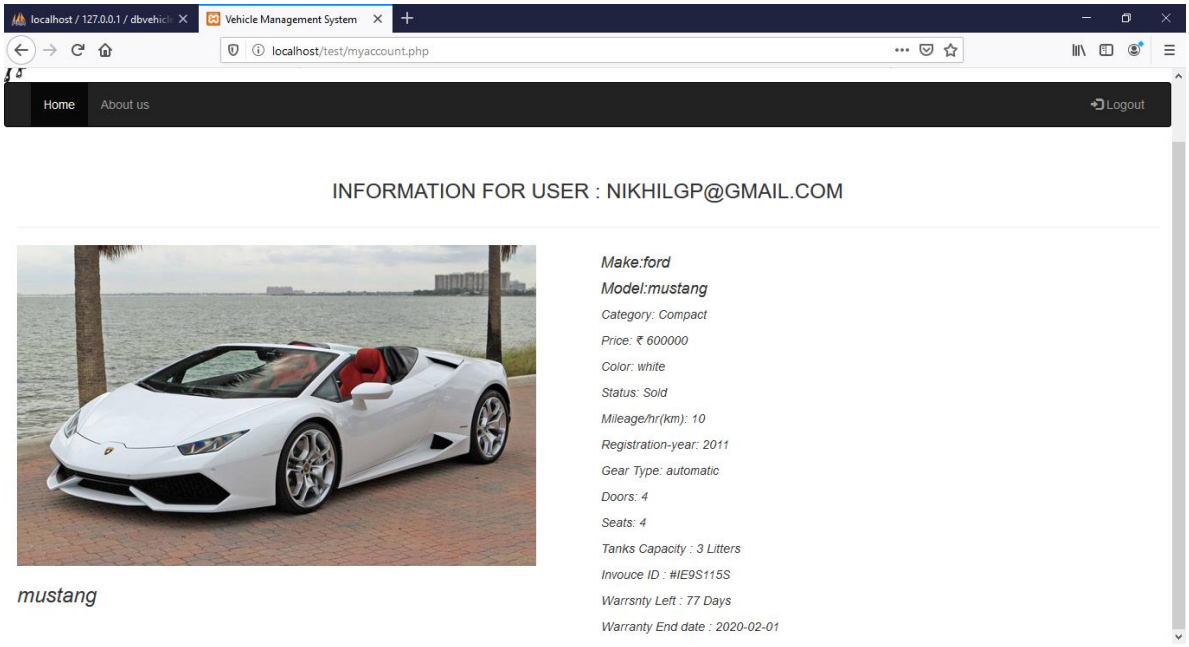


Fig.5.7: Client vehicle information

7. Add vehicle manufacturer

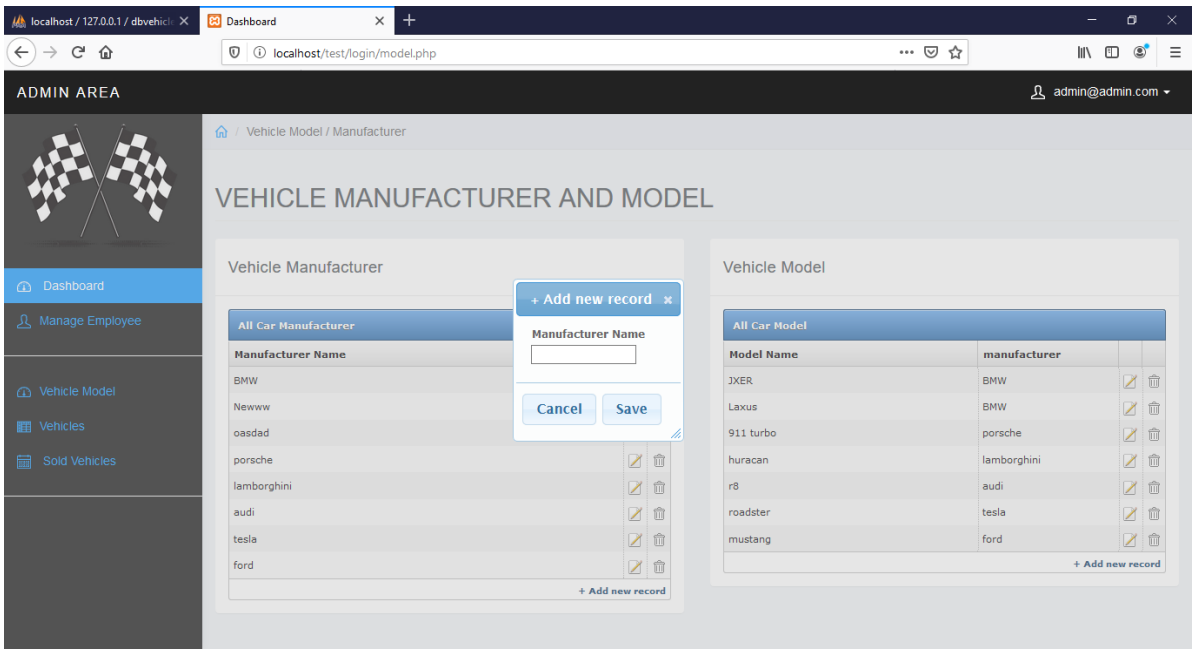


Fig.5.8: Add vehicle manufacturer

8. Add vehicle model

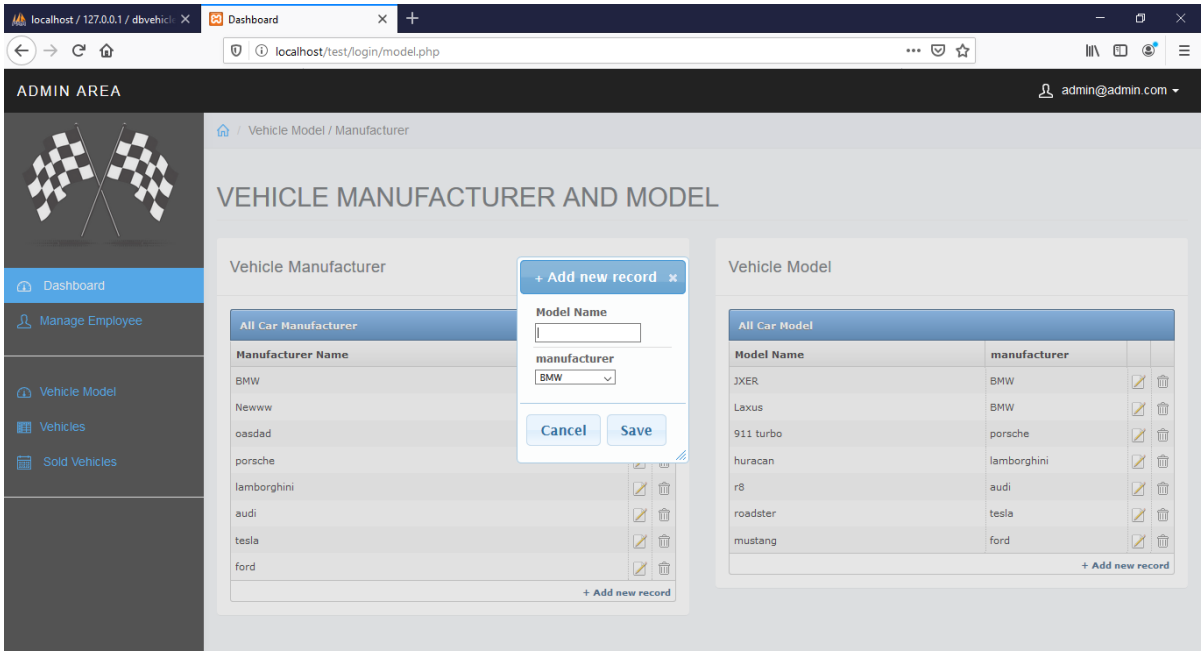


Fig.5.9: Add vehicle model

9. Add new vehicle

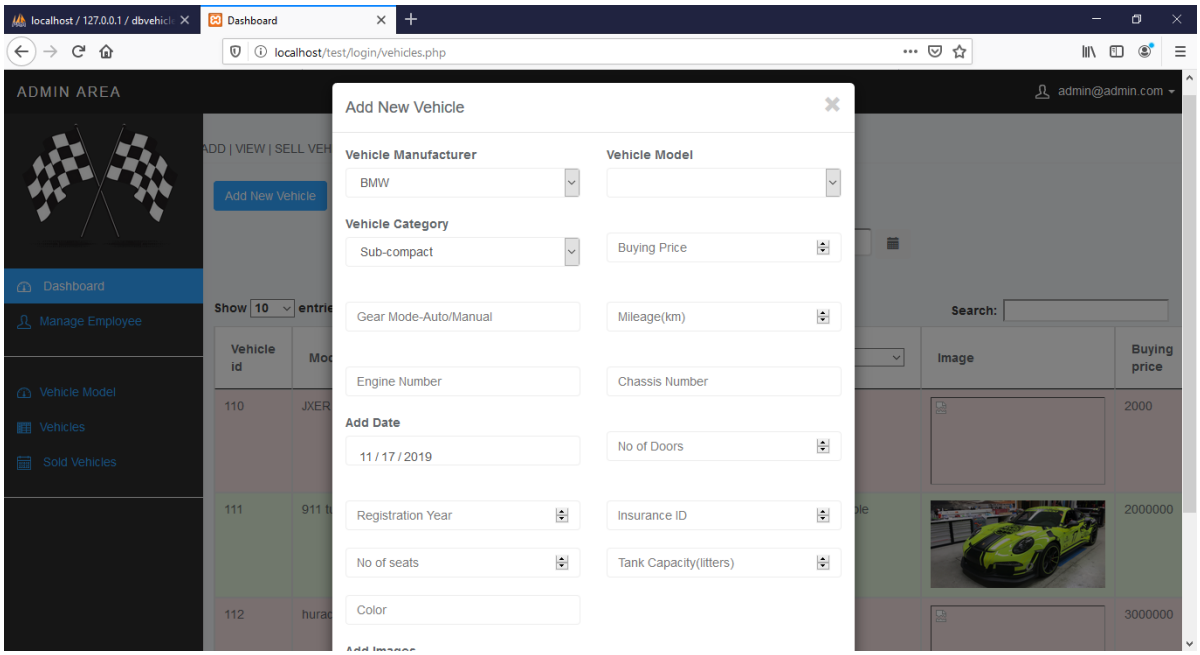


Fig.5.10: Add new vehicle

10. Add/View vehicle

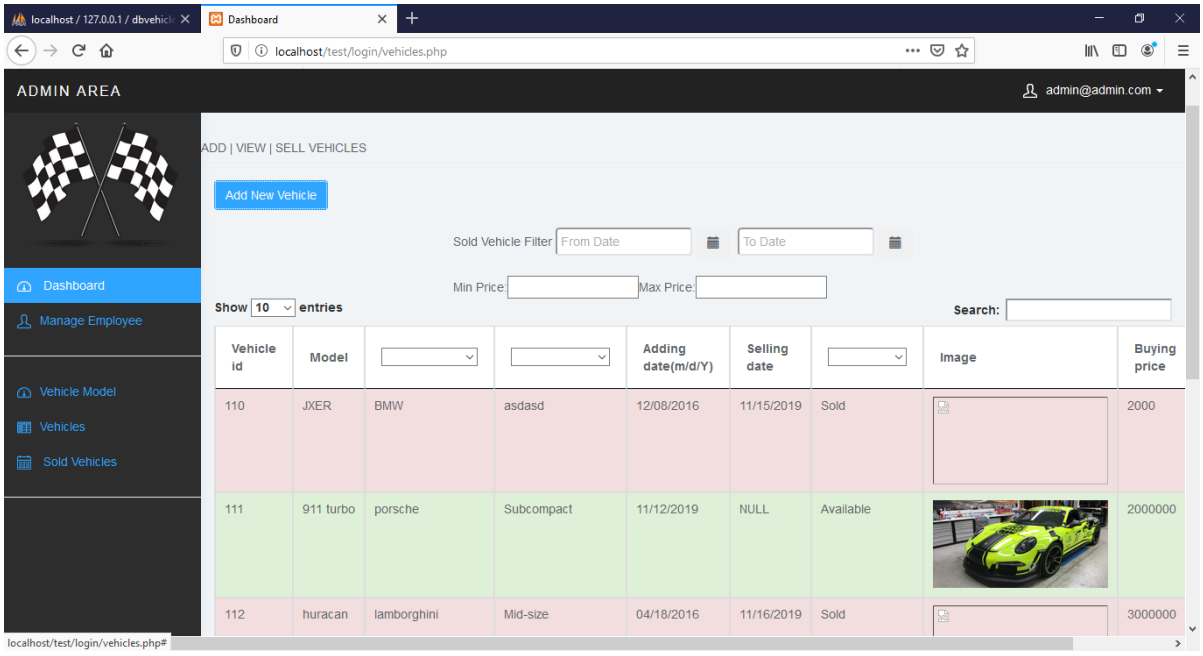


Fig.5.11: Added/View vehicle

11. View/Delete customers

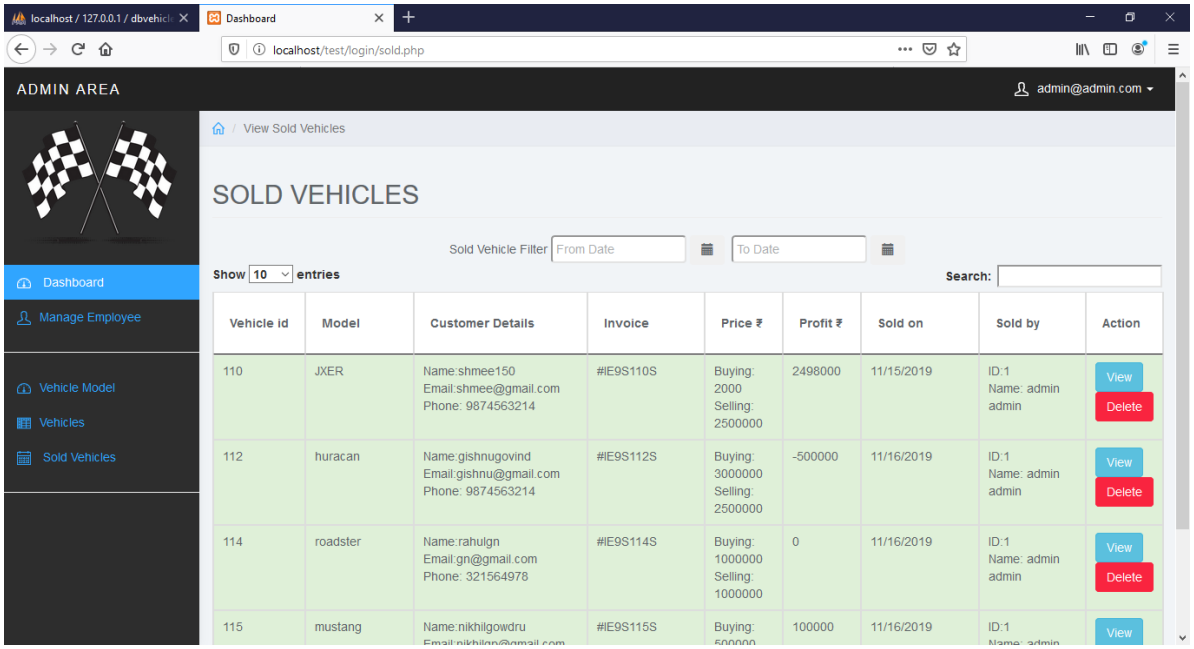


Fig.5.12: View/Delete customers

12.View Sold vehicles

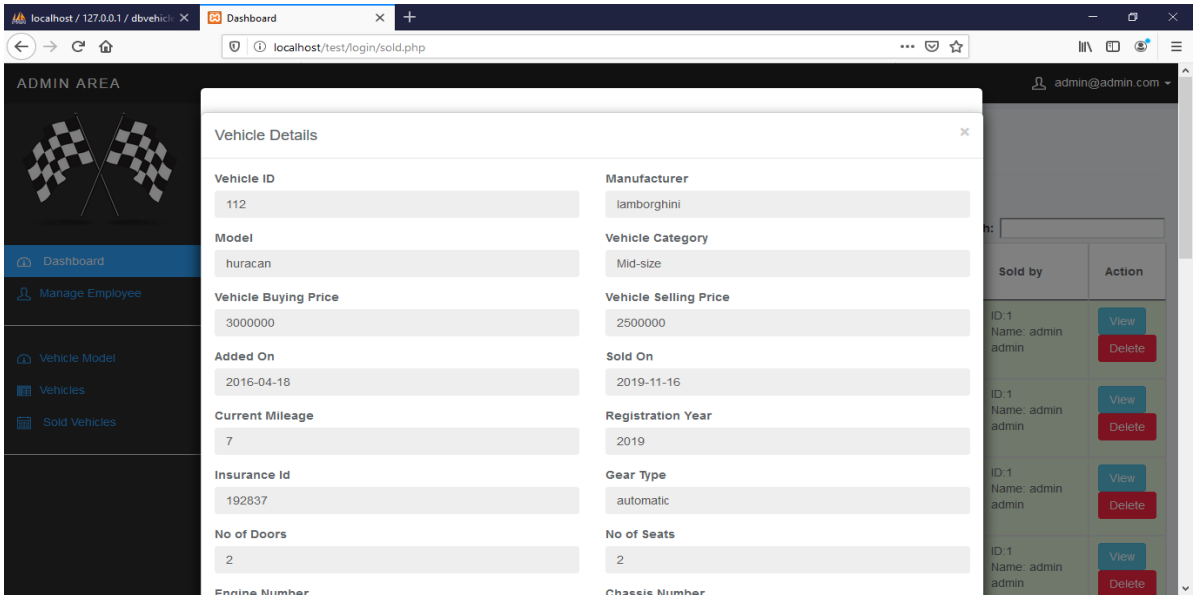


Fig.5.13: View Sold vehicles

CONCLUSION

Vehicle Management System deals about Adding, Selling and Viewing vehicles that you want to purchase by signing in with the help of admin and employees. Search can be performed on performed on various category of cars, based on brands and corresponding specifications is provided. Basic specifications like transmissions, mileage, no. of seats, paints color etc. are covered and provides a complete picture of vehicle to the users. In addition to this additional information if any, is also provided. Thus, the user is provided with an overall picture of the vehicle in a single click

6.1 Further enhancements

The project uses simple Schema which is easy to implement and also enhance the application as per future requirements. The tables are normalized in appropriate manner, so there will not be any ambiguity as the data increases. Hence, this application can be enhanced to meet the growing demands of the vehicles.