VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi: 590 018

A Mini Project report on

" Vehicle Management System"

Submitted in partial fulfillment of the requirement for the award of Degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

By

Saksham dani(1AY17CS083) **Umang shrivastava(1AY17CS110)**

Under the guidance of

Prof. Geetha N

Prof.Ancy thomas

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ACHARYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)

2019-2020

ACHARYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)
Soladevanahalli, Bangalore – 560090

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Mini Project entitled **"VEHICLE MANAGEMENT DATABASE "** is a bonafide work carried out by **Umang Shrivastava(1AY17CS110)** and **Saksham Dani(1AY17CS089)** in partial fulfillment for the award of degree of **Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University**, **Belagavi**, during the year **2019-2020.** It is certified that all corrections/ suggestions indicated for internal assessments have been incorporated in the Report deposited in the departmental library. The Mini Project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the **Bachelor of Engineering Degree**.

Signature of Guides **Signature of H.O.D**

Name of the examiners **Signature with date**

1.

2

ACKNOWLEDGEMENT

We express our gratitude to our institution and management for providing us with good infrastructure, laboratory, facilities and inspiring staff whose gratitude was of immense help in completion of this seminar successfully.

We express our sincere gratitude to our principal, **Dr. Prakash M R** for providing required environment and valuable support for developing this mini project.

Our sincere thanks to **Dr. Prashanth C M,** Head of the Department, Computer Science and Engineering, Acharya Institute of Technology for his valuable support and for rendering us resources for this mini project work.

We express our gratitude to **Prof. Ancy Thomas and Prof. Geetha N** Assistant Professors, Dept. Computer Science and Engineering, Acharya Institute of Technology who guided we with their valuable suggestions in completing this miniproject at every stage.

Our gratitude thanks rendered to many people who helped us in all possible ways.

SAKSHAM DANI (1AY17CS083)

UMANG SHRIVASTAVA(1AY17CS110)

ABSTRACT

- This project is a retail database system for an vehicle showroom. This software will help salespersons in managing the various types of records pertaining to his/her customer. This product will help the user to work in a highly effective and efficient environment.

- Vehicle management refers to the process which helps the customers to obtain the desired resources from the showroom. Vehicle Management System allows the customers to have a good experience and leave the store with a smile. It helps them to shop without any difficulty. It saves time of the customers since they will be able to buy the desired car or bike. One can design this retail application database project for the effective maintenance of the retail application. At the completion of a sale, a receipt is created for the customer and sales information is collected for the generation of reports at a later time.

CONTENTS

Chapter 1

INTRODUCTION

Our project is based on Data base management system. In this mini project I made a database for managing the vehicle sales system. In my project a showroom employee can enter the data or can keep the information of all the transaction and the vehicles available. Previously these work was done by writing them into a book and make a note for all the transactions and maintaining a logbook for the vehicle available but as we move towards the advancement a need of these type of software is much higher because everybody need the speed and want their work to be done on click so this is the smart solution to create a database , where a user can note its all the important transaction and can be store it for the longer time interval. In this data base vehicle database keeper can add productlines with text descriptions ,vehicles with their price , its customers by entering their names, which product to deliver to which customer and product type all these types of information a user can note in this system so whenever it is required he can open his system and can look the information regarding its product supply.

So, A database system provide, A suitable way to ease and fulfill not only the business requirements but also to provide easy of data handling in many field..

1.2 ABOUT THE PROJECT

In my mini project there are three interrelated classes with which I can synchronize the whole data these are written below.

Home page **-:** After entering into the system home page will appear.

Dealer Page **-:**Information about dealer and addition of new dealer.

Sale page **-:**Here one can keep track of sales and add new sales.

Showroom Page**-:**Here all the showroom are there.

Tax Details-: Where one see all the tax details and add them.

1.3 Development Tools

Programming Language and Tools used in making the project-:

I make my module in PHP and in CSS

Language Used: CSS, HTML, PHP, MySQL

Tools Used:

1. Code editor tool-: WE used brackets as my code editor software. Because it an extent and customizable UI and it also stands up to the high demand of a full time developer and content creator. and this software WE used for making Front end.

2.Table making tool-: WE used PhpMyAdmin for creating data base tables. It is the most compatible software for MYSQL because table is made by using MYSql in PhpMyAdmin and the information about the data is called in the table by using the approach call by Query.if we want a specific info about the related table content then query will help in that by calling it

3.Xampp server-: It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. It connect Mysql and php through Apache server.

Web-Browser: Mozilla Firefox/Google Chrome/Opera / Safari /IE11/Microsoft Edge.

Chapter 2

REQUIREMENT

2.1 Software Requirements :

Number	Description
1	PHP 5 :- Used for creating the front end of our project
2	MySQL :- Basically whole backend in done in mysql
3	XAMPP SERVER :- Helped in connection of mysql and php via providing a platform
4	PHPMYADMIN :- It's a free web application which in providing convenient GUI.

2.2 Hardware Requirements:

Number	Description
1	WINDOWS 7 and above
2	520 MB RAM or higher

Chapter 3

DESIGN

3.1 E-R Diagram

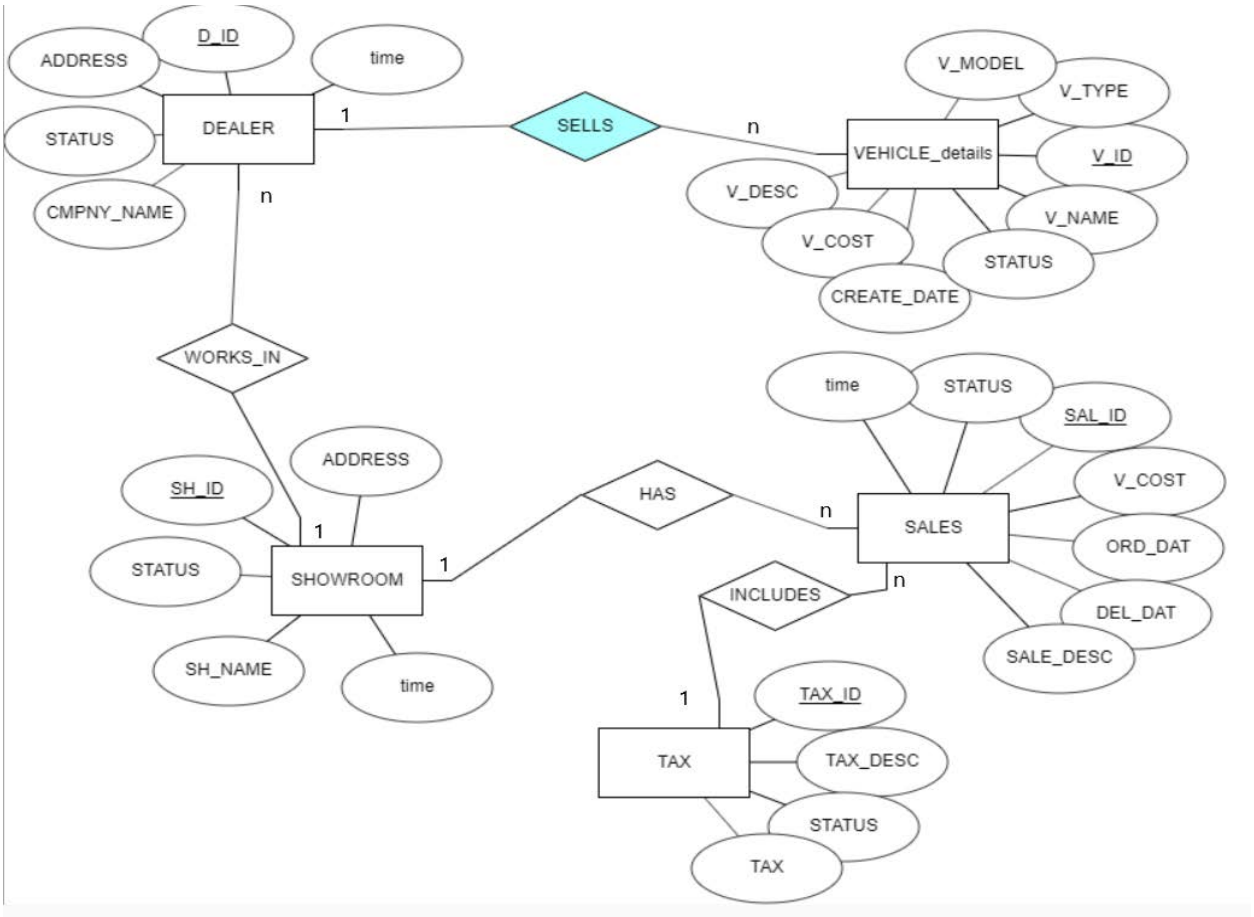


Fig 3.1 E-R Diagram for vehicle management database

3.2 Schema Diagram

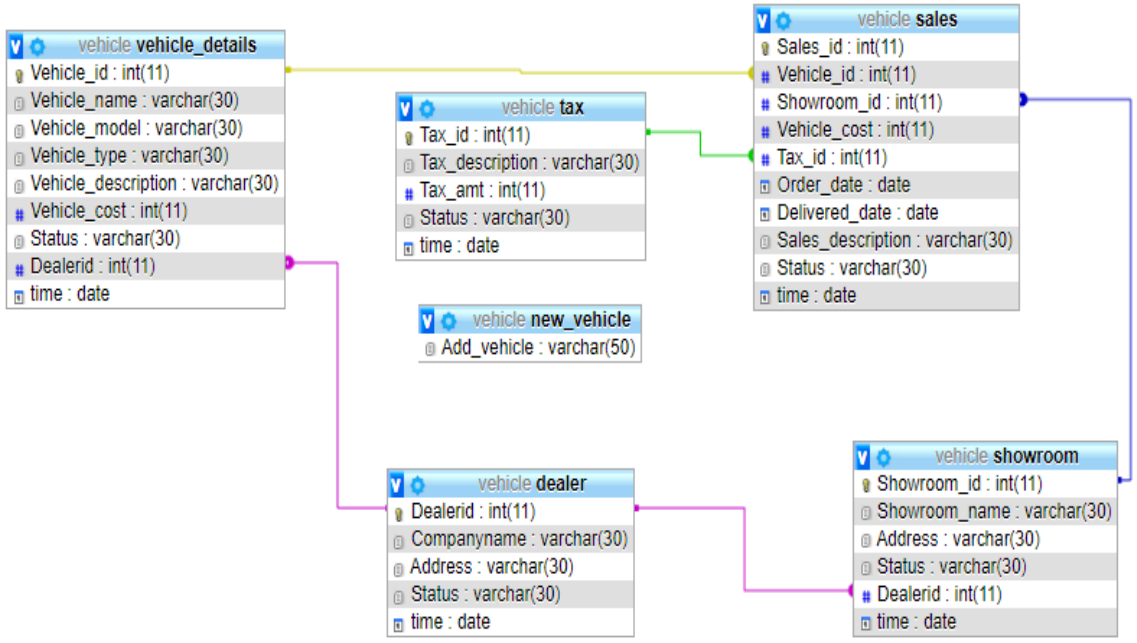


Fig 3.2 Schema Diagram for vehicle management database

Made from xampp

Chapter 4

IMPLEMENTATION

4.1 MYSQL

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

Primary key- the field that is unique for all the record occurrences.

Foreign key-the field used to set relation between tables.

MySQL is multithreaded, multi user SQL database management System (DBMS). The basic program run as server providing multiuser access to a number of databases. MySQL is a database. The data in a MySQL is stored in a Database objects called tables. A table is a collection of related data entries and it consists of columns and rows. The databases are useful when storing information categorically.

4.2 TABLES

- Dealer
- Showroom
- Tax
- Vehicle_details
- Sales

TABLE DEALER

- Create table dealer(dealerid int,cmpny_name varchar(30),address varchar(30)status varchar(30),time date,constraint dealer_pk1 primary key(dealerid) on delete set null);

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Dealerid	int(11)			No	None			Change Drop More
2	Companyname	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
3	Address	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
4	Status	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
5	time	date			Yes	NULL			Change Drop More

Fig4.21

TABLE SHOWROOM

- Create table showroom(showroom_id int,dealerid int,showroom_name varchar(30) ,address varchar(30),status varchar(30),constraint sh_pk1 primary key (showroom_id),constraint sh_fk1 foreign key(dealerid) references dealer(dealerid) ondelete set null);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Showroom_id	int(11)			No	None			Change Drop More
2	Showroom_name	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
3	Address	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
4	Status	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
5	Dealerid	int(11)			No	None			Change Drop More
6	time	date			Yes	NULL			Change Drop More

Fig4.22

TABLE TAX

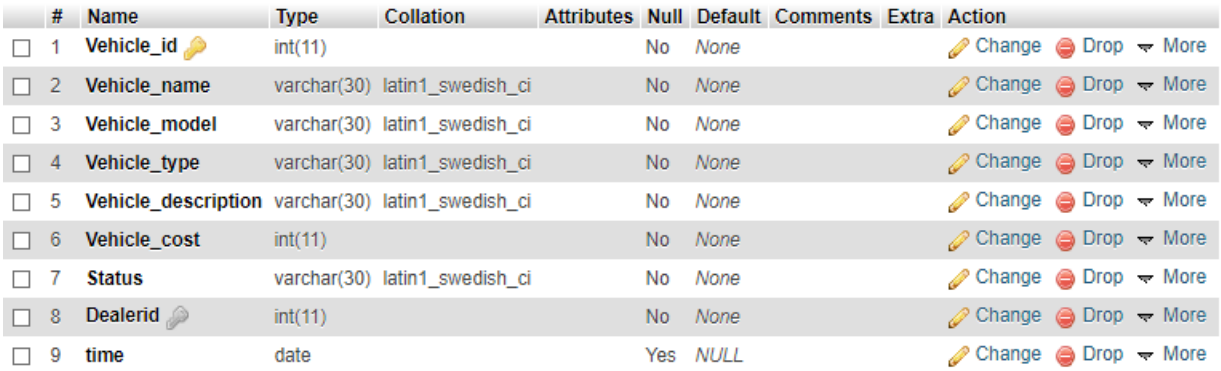
- Create table tax(tax_id int,tax_description varchar(30),tax_amt int not null,status varchar(30),constraint tax_pk1 primary key(tax_id));

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Tax_id	int(11)			No	None			Change Drop More
2	Tax_description	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
3	Tax_amt	int(11)			No	None			Change Drop More
4	Status	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
5	time	date			Yes	NULL			Change Drop More

Fig4.23

TABLE VEHICLE_DETAIL

- Create table vehicle_details(vehicle_id int,vehicle_name varchar(30),vehicle_model varchar(30),vehicle_type varchar(30),vehicle_description varchar(30),vehicle_cost int not null,time date,status,dealerid int varchar(30),constraint v_pk1 primary key(vehicle_id),constraint v_fk1 foreign key(dealerid) references dealer(dealerid) ondelete set null);

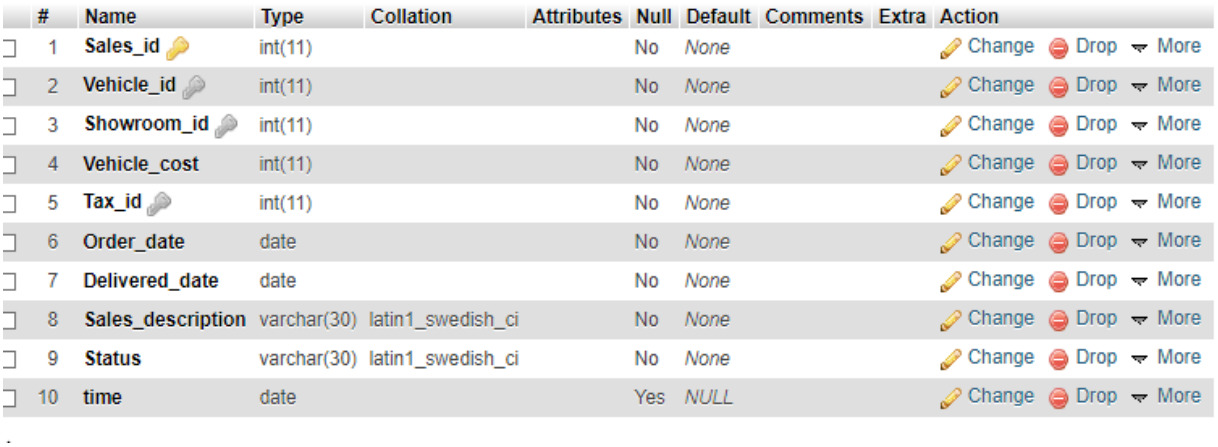


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Vehicle_id	int(11)			No	None			Change Drop More
2	Vehicle_name	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
3	Vehicle_model	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
4	Vehicle_type	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
5	Vehicle_description	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
6	Vehicle_cost	int(11)			No	None			Change Drop More
7	Status	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
8	Dealerid	int(11)			No	None			Change Drop More
9	time	date			Yes	NULL			Change Drop More

Fig4.24

TABLE SALES

- Create table sales(sales_id int,vehicle_id int,showroom_id int,vehicle_cost int,tax_id int,order_date date,deliver_date date,sales_description varchar(30), status varchar(30),time date,constraint sal_pk1 primary key(sales_id),constraint sal_fk1 foreign key(showroom_id) references showroom(sh_id)on delete set null);



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Sales_id	int(11)			No	None			Change Drop More
2	Vehicle_id	int(11)			No	None			Change Drop More
3	Showroom_id	int(11)			No	None			Change Drop More
4	Vehicle_cost	int(11)			No	None			Change Drop More
5	Tax_id	int(11)			No	None			Change Drop More
6	Order_date	date			No	None			Change Drop More
7	Delivered_date	date			No	None			Change Drop More
8	Sales_description	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
9	Status	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
10	time	date			Yes	NULL			Change Drop More

Fig4.25

4.3 QUERIES

As you might have observed from the simple program in the previous section, MySQL uses mainly uses six commands in which SELECT is used to retrieve rows selected from one or more tables. FROM refers to the table from which we need to select the attributes. WHERE clause, if given, indicates condition or conditions that rows must satisfy to be selected. where_ condition is expression that evaluates to true for each row to be selected. This statement selects all rows if there is no where clause. GROUP BY clause used to Group the values of the attributes provided that values must be same. HAVING clause is applied nearly last, just before items are sent to the client, with no optimization. If the HAVING clause refers to a column that is ambiguous, warning occurs. ORDER BY clause is used for the purpose of sorting the values of the attributes in a result. If you use GROUPBY ,output rows are sorted according to GROUP BY columns as if you had an ORDER BY for the same columns.

- SELECT VEHICLE_NAME FROM VEHICLE_DETAILS WHERE V_TYPE='BIKE';
- SELECT D.DEALERID FROM DEALER D, SALES S ,SHOWROOM SH WHERE S.SHOWROOM_ID=SH.SHOWROOM_ID AND SH.DEALERID=D.DEALERID AND ORDER_DATE='2016-01-16';
- SELECT VEHICLE_NAME,MAX(VEHICLE_COST) FROM VEHICLE_DETAILS GROUP BY VEHICLE_TYPE;
- SELECT T.TAX_DESCRIPTION,V.VEHICLE_NAME,T.STATUS FROM TAX T, DEALER D, SALES S, VEHICLE_DETAILS V WHERE S.VEHICLE_ID =V.VEHICLE_ID AND S.TAX_ID=T.TAX_ID AND V.VEHICLE_ID=11;
- SELECT S.SHOWROOM_NAME,D.CMPNY_NAME FROM DEALER D,SHOWROOM S WHERE S.DEALERID=D.DEALERID AND D.DEALERID=104;
- SELECT VEHICLE_NAME,VEHICLE_TYPE,VEHICLE_COST FROM VEHICLE_DETAILS WHERE STATUS="AVAILABLE"
- SELECT SHOWROOM_NAME,DEALERID FROM SHOWROOM S,DEALER D WHERE S.FEALERID=D.DEALERID AND D.STATUS="SELLING"
- SELECT VEHICLE_NAME,VEHICLE_TYPE FROM VEHICLE_DETAILS V,SHOWROOM S WHERE V.DEALERID=S.DEALERID AND S.ADDRESS ='BANGLORE'

- SELECT VEHICLE_NAME FROM VEHICLE_DETAILS V,SALES S,DEALER D WHERE V.VEHICLE_ID=S.VEHICLE_ID AND D.DEALERID=V.DEALERID AND DEALERID=105;
- SELECT V.VEHICLE_NAME,V.VEHICLE_ID, FROM VEHICLE_DETAILS V, SALES S WHERE S.VEHICLE_ID=V.VEHICLE_ID AND ORD_DAT<'2017-01-01';

4.4 Triggers

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database. For example, when anew record (representing a new worker) is added to the employees table, new records should also be created in the tables of the taxes, vacations and salaries. Triggers can also be used to log historical data.

CREATE TRIGGER `new` AFTER INSERT ON `new_vehicle`

FOR EACH ROW

VALUES("new vehicle inserted")

end if;

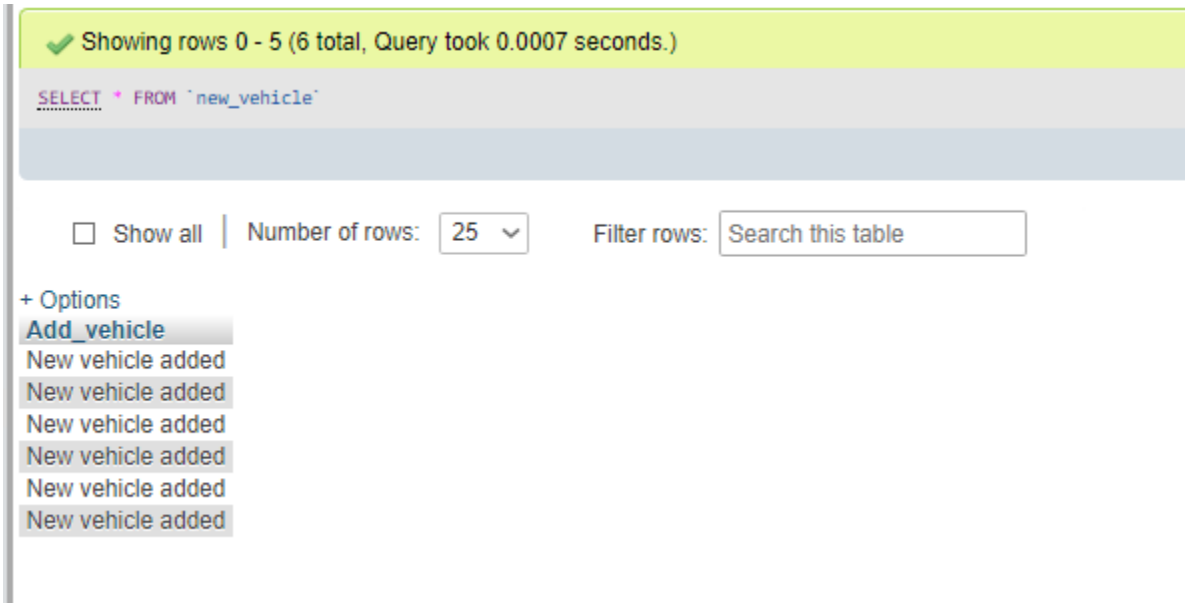


Fig4.26 Trigger

4.5 Stored procedure

A stored procedure is a subroutine available to applications that access a relational database management system (RDBMS). Such procedures are stored in the database data dictionary. Uses for stored procedures include data-validation (integrated into the database) or access-control mechanisms. Furthermore, stored procedures can consolidate and centralize logic that was originally implemented in applications. To save time and memory, extensive or complex processing that requires execution of several SQL statements can be saved into stored procedures, and all applications call the procedures. One can use nested stored procedures by executing one stored procedure from within another

CREATE PROCEDURE ADD_VEHICLE (IN iid INT)

NOT DETERMINISTIC NO SQL SECURITY DEFINER SELECT VEHICLE_MODEL FROM VEHICLE ;

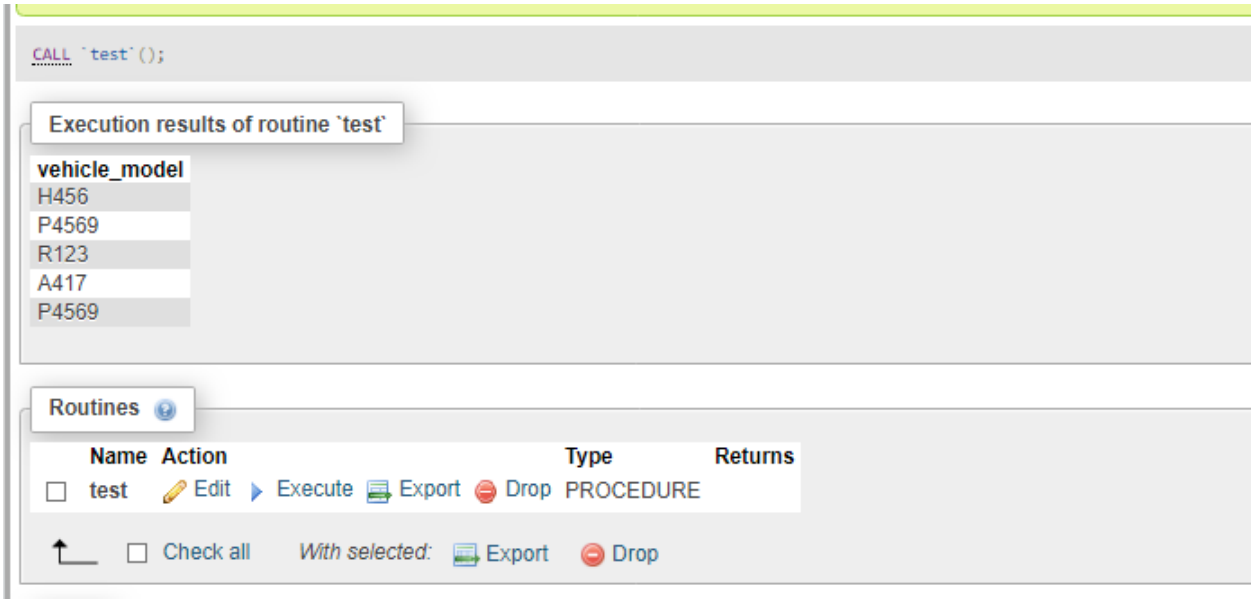


Fig4.27 Stored procedure

Query1: SELECT ALL VEHICLE NAME WHERE VEHICLE TYPE IS BIKE

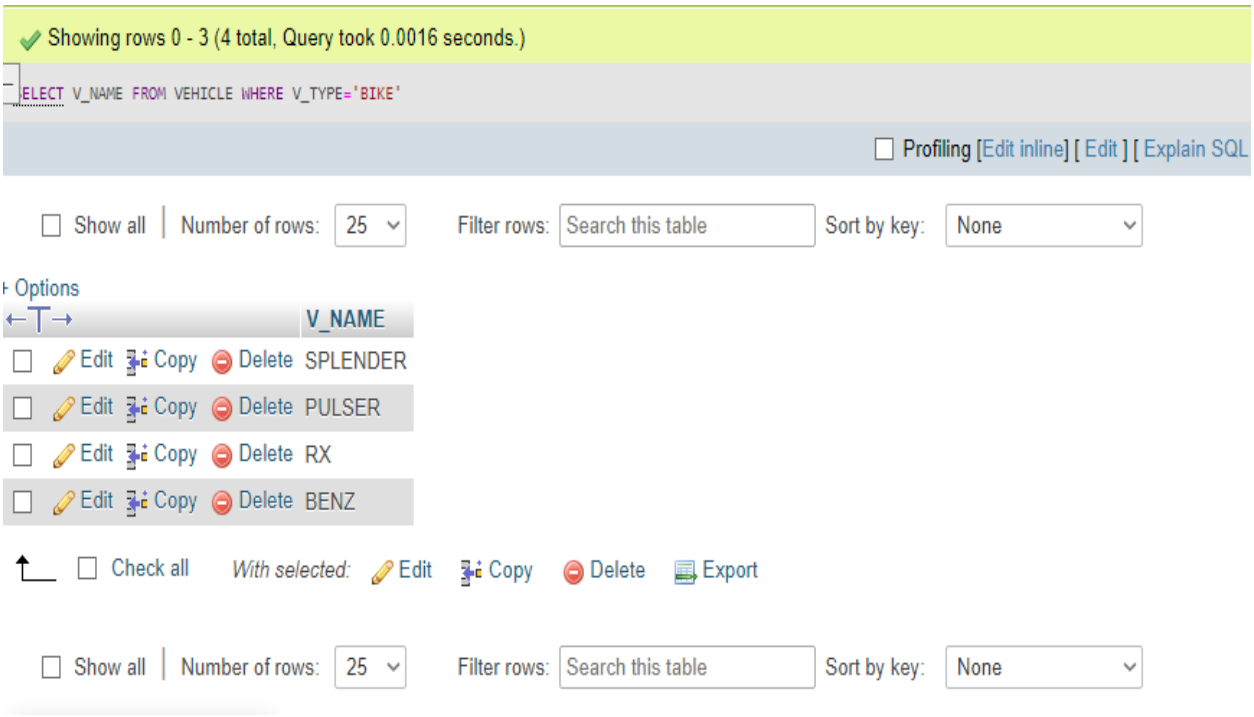


Fig4.28

Query2: LIST ALL dealerS WHO sold VEHICLE ON DATE 16-01-2016

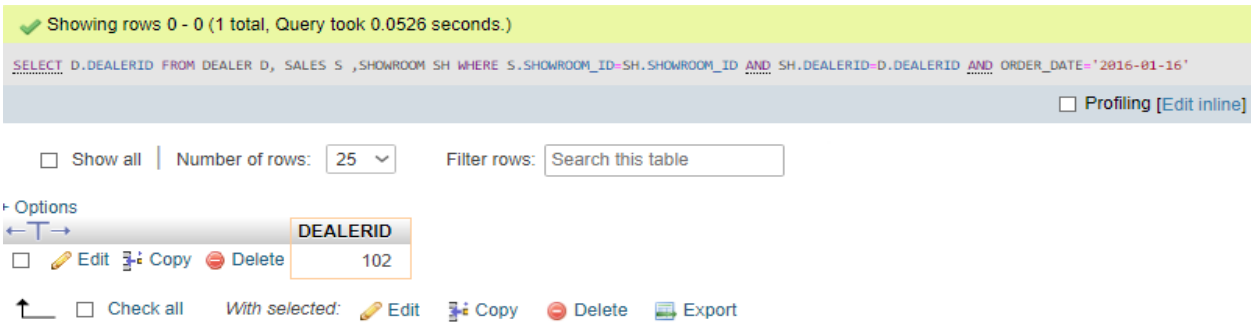


Fig4.29

Query3: LIST VEHICLE_NAME WHICH HAS HIGH COST AMONG ALL VEHICLES;

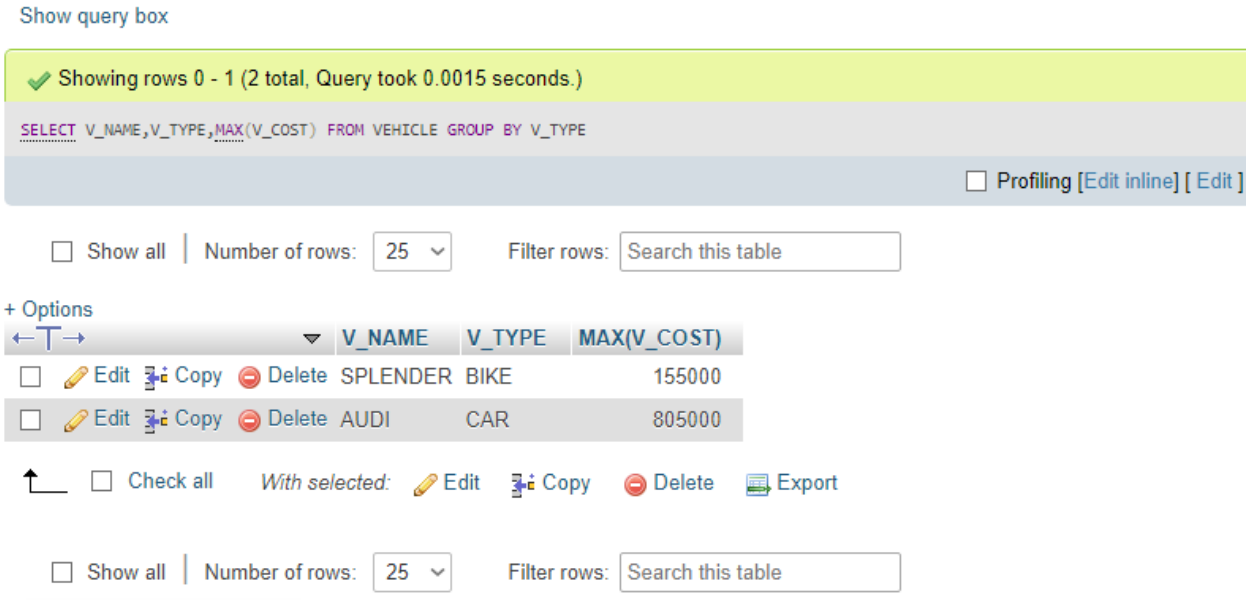


Fig 4.30

Query4: DISPLAY TAX DEATAILS OF VEHICLE WHERE VEHICLE ID IS 11

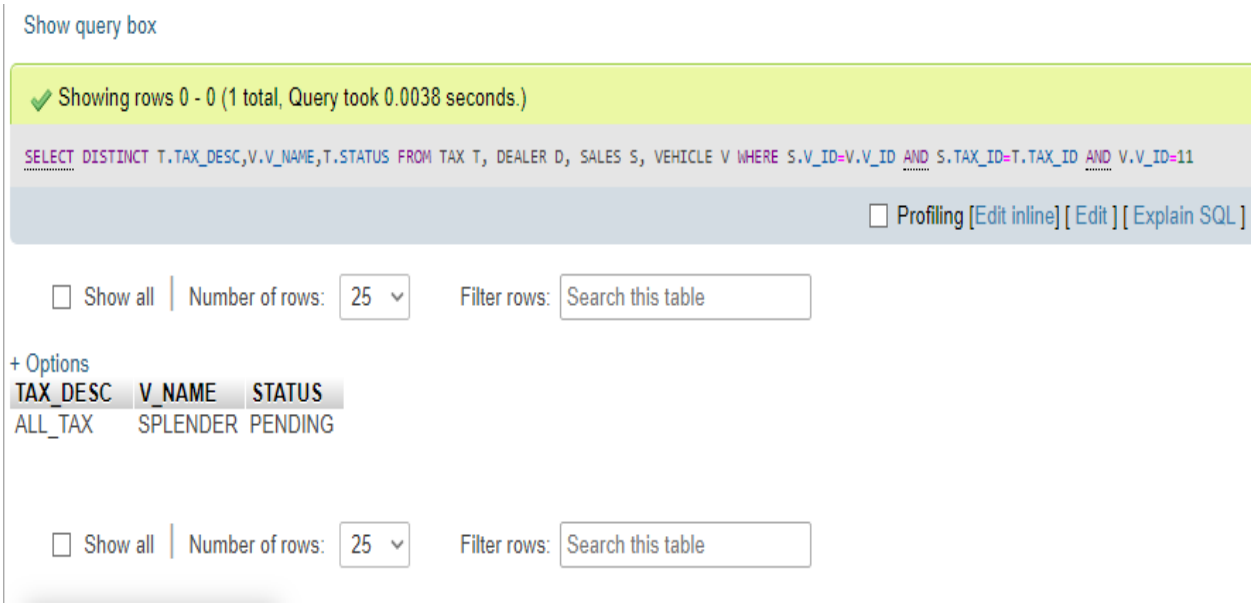


Fig 4.31

Query5: LIST ALL SHOWROOM NAME THAT COMES UNDER DEALER WITH ID=104

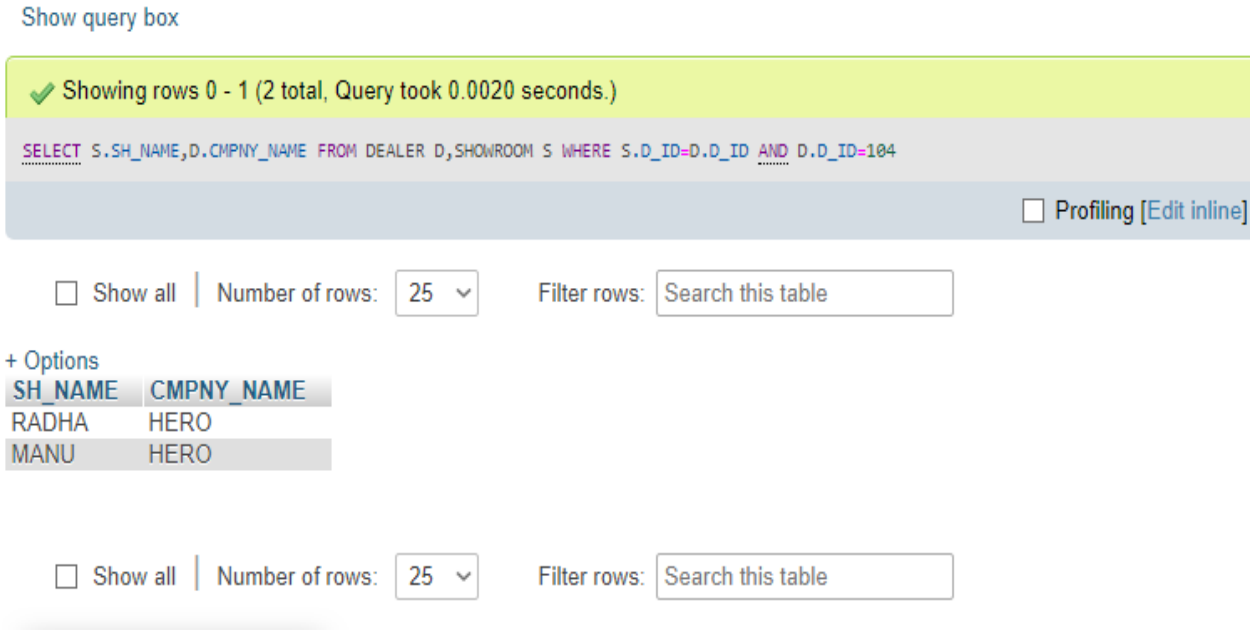


Fig 4.32

Query6: LIST ALL THE VEHICLES WHICH ARE AVAILABLE

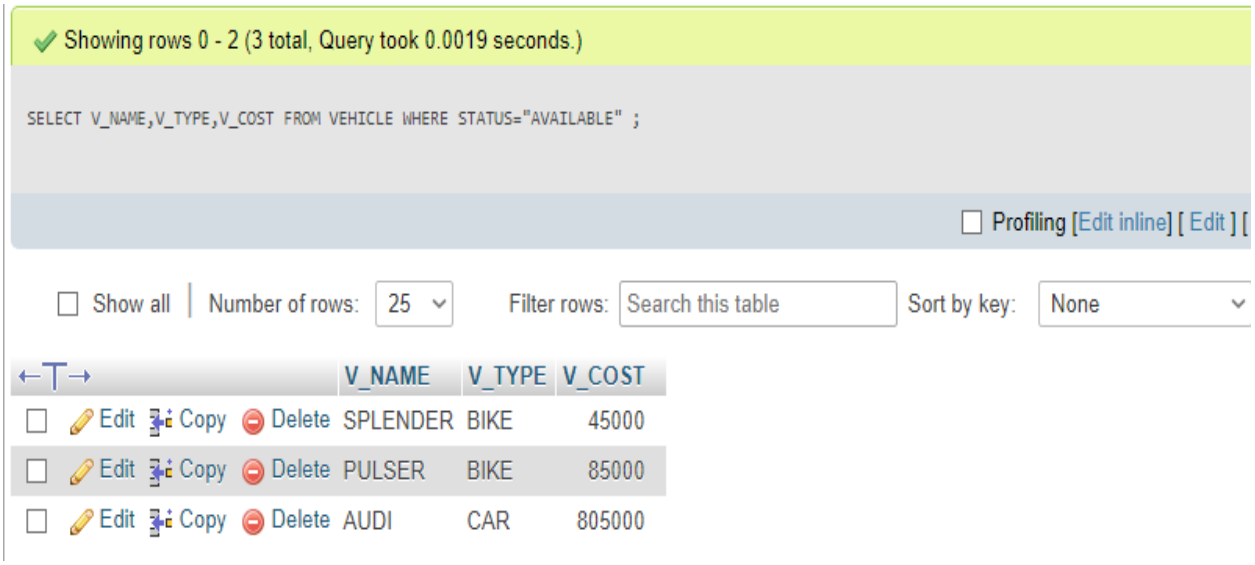


Fig 4.33

Query7:LIST ALL THE DEALERS WHICH ARE CURRENTLY SELLING

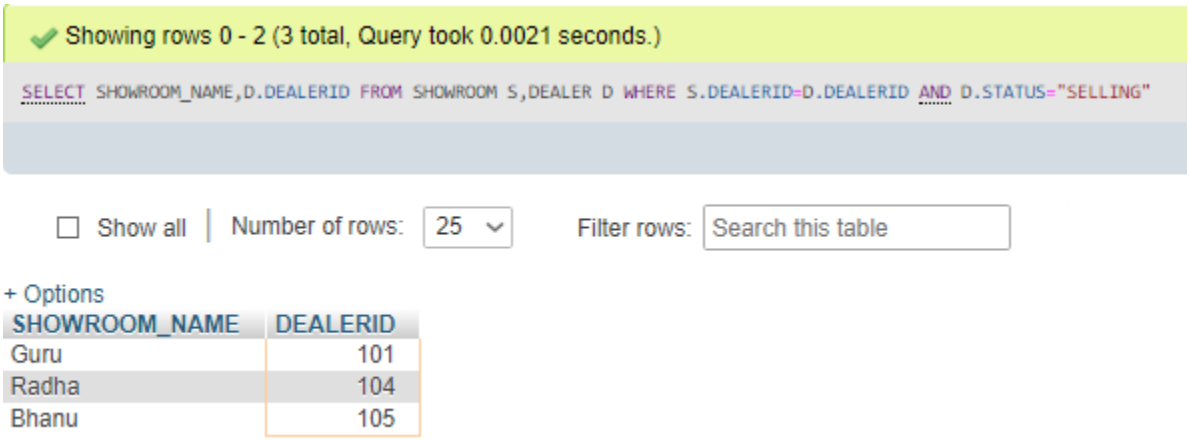


Fig 4.34

Query8:LIST ALL THE VEHICLES PRESENT AT SHOWROOM IN BANGALORE

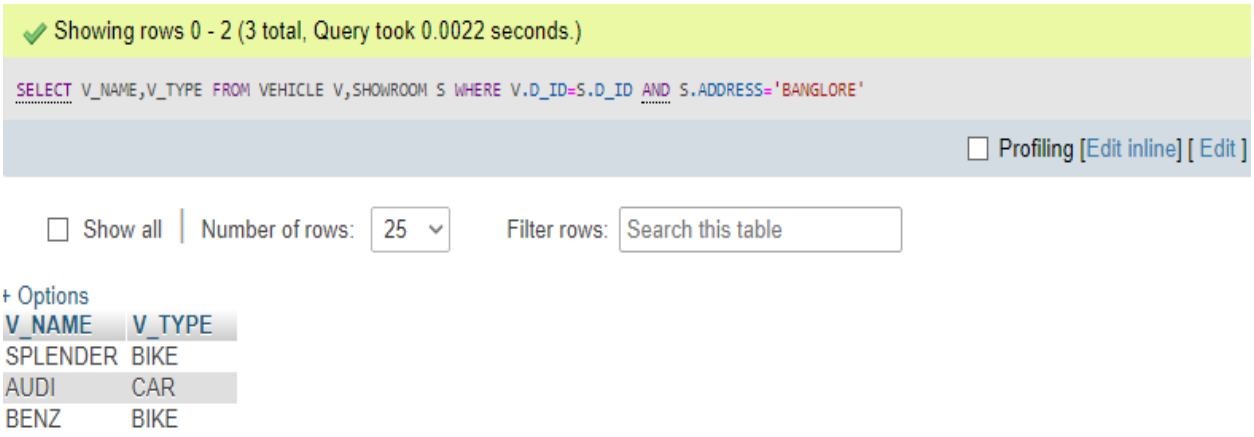


Fig 4.35

Query9: LIST THE VEHICLE SOLD BY DEALER ID=105

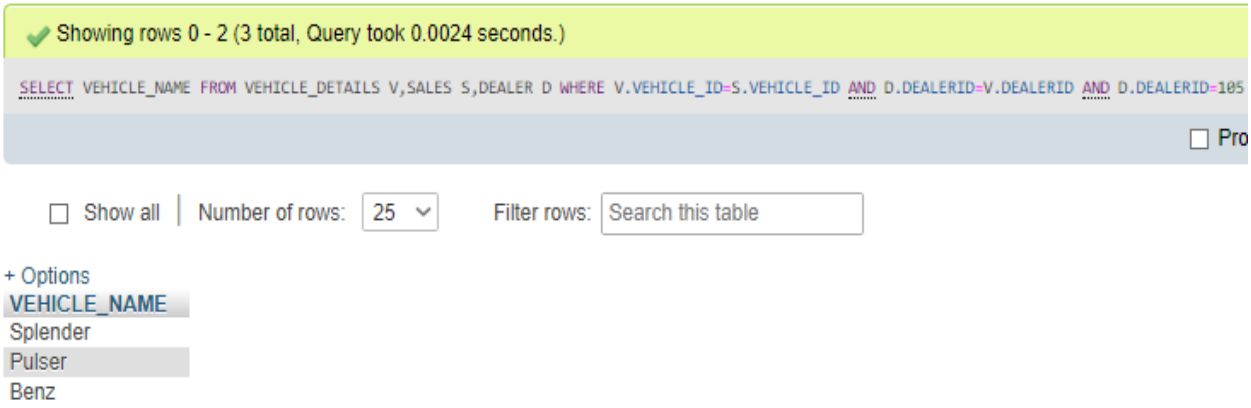


Fig 4.36

Query10: DISPLAY THE VEHICLE ORDERED BEFORE 2017

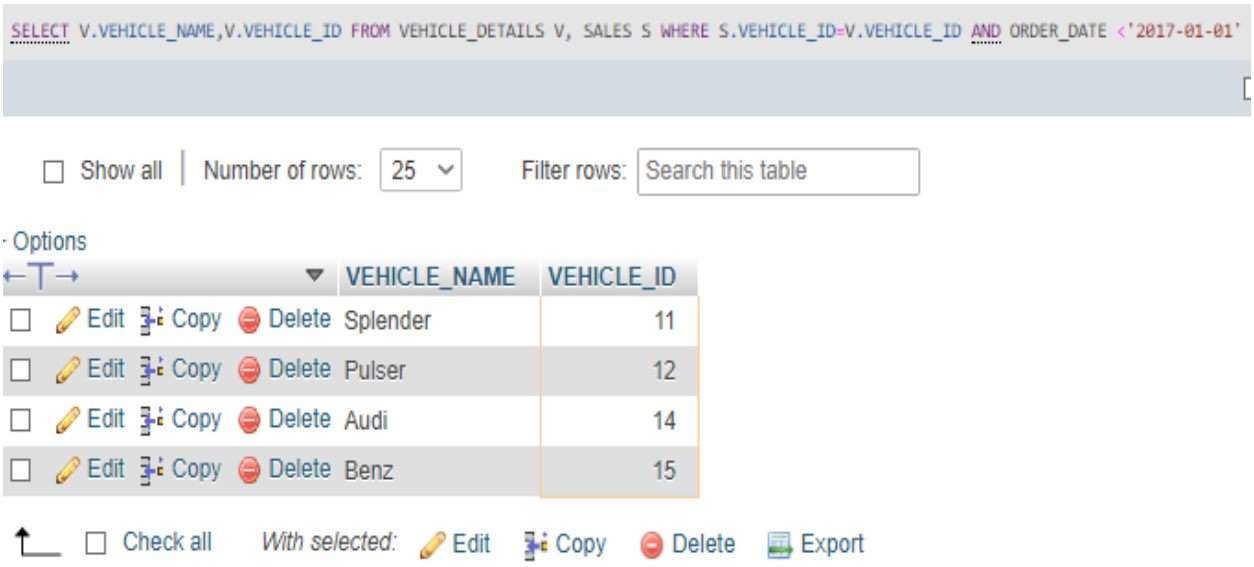


Fig 4.37

Chapter 5

RESULTS AND SNAPSHOTS

HOME PAGE:

In Home page there are 5 tabs namly Dealer ,Sales ,Showroom,Tax,Vehicle_details.After clicking on each tab it will redirect to the particular page where insertion can be done.

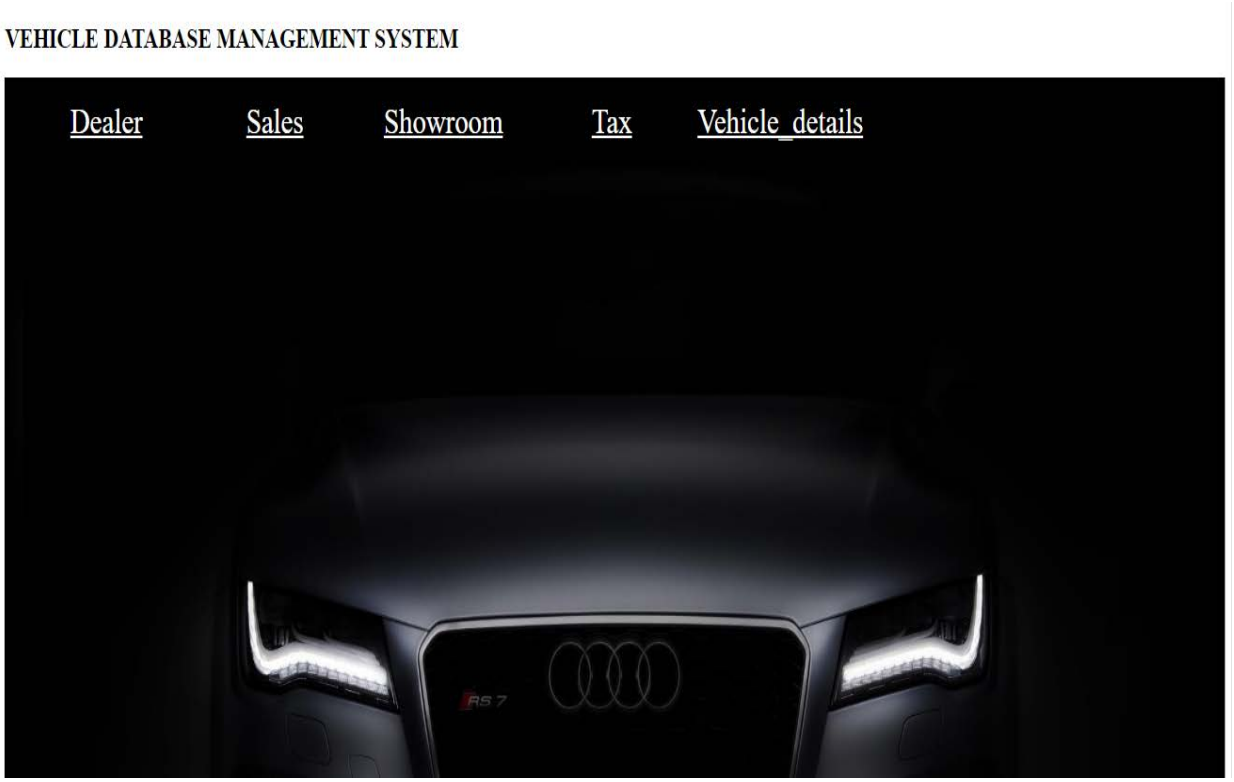


Fig 5.1 page showing our homepage

Dealer :

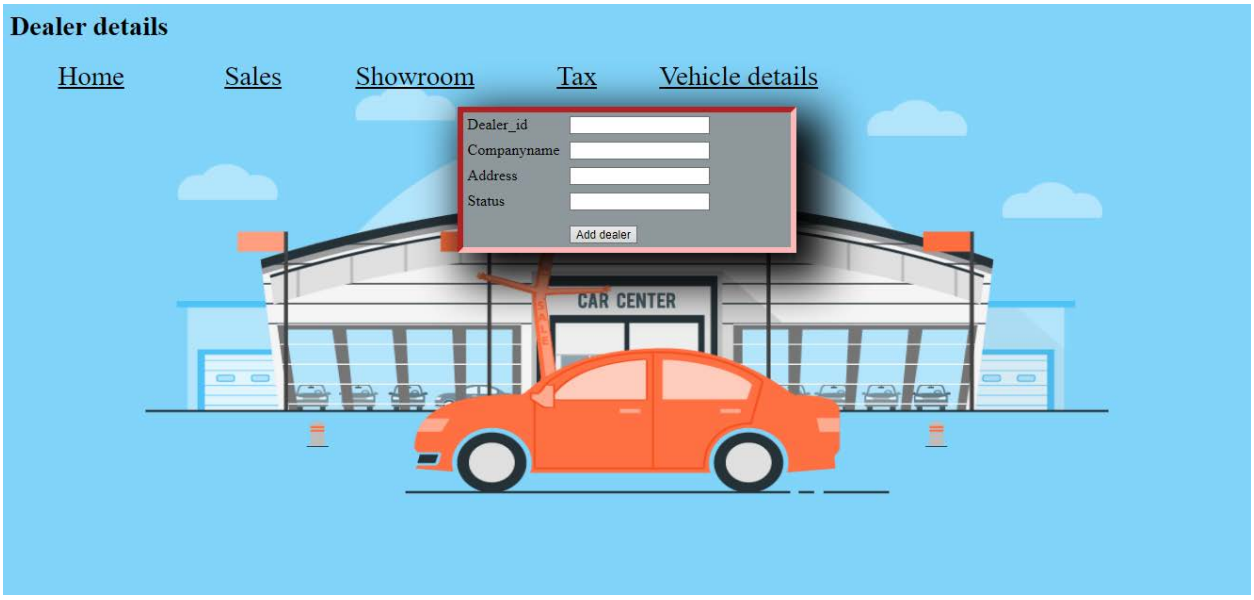


Fig 5.2 Dealer's page insertion can be done

+ Options

	Dealerid	Companyname	Address	Status	time
Edit Copy Delete	101	Bajaj	Bangalore	Selling	2019-11-29
Edit Copy Delete	102	Honda	Bellary	Not selling	2019-11-29
Edit Copy Delete	103	Yamaha	Mysore	Not selling	2019-11-29
Edit Copy Delete	104	Hero	Gadag	Selling	2019-11-29
Edit Copy Delete	105	Bullet	Bangalore	Selling	2019-11-29
Edit Copy Delete	423	xlx	pune	pending	2019-11-30

Fig 5.3 Backend insertion

Sales:

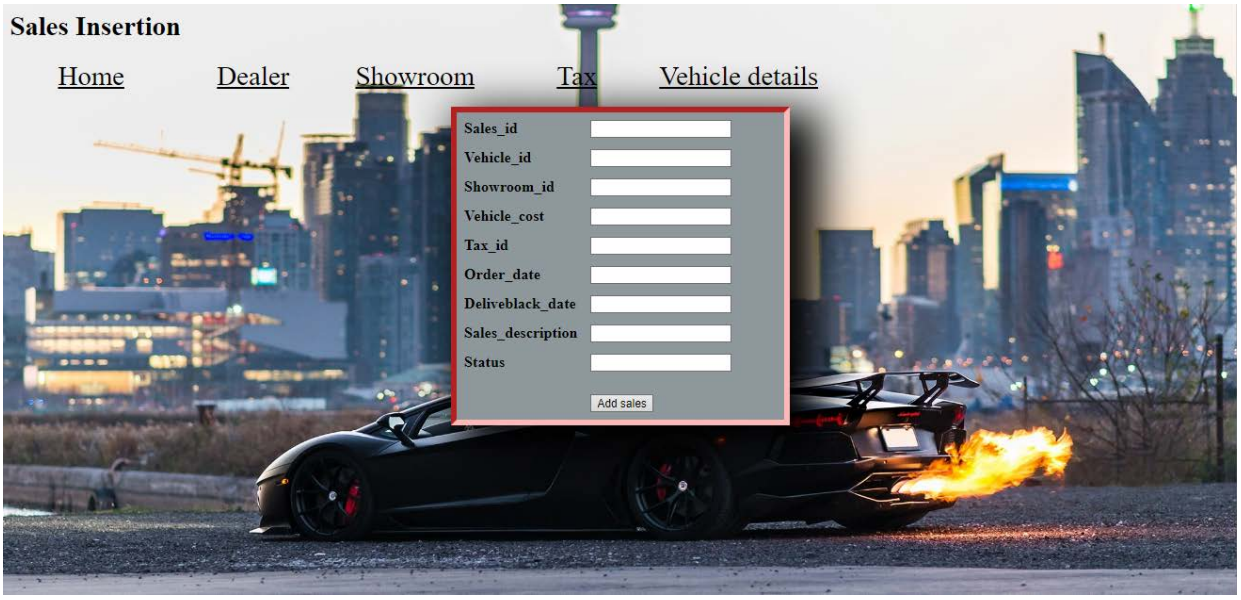


Fig 5.4 Sales page insertion can be done

+ Options

	Sales_id	Vehicle_id	Showroom_id	Vehicle_cost	Tax_id	Order_date	Delivered_date	Sales_description	Status	time
Edit Copy Delete	110	11	111	45000	1000	2015-01-26	2016-01-06	High	Super	2019-11-29
Edit Copy Delete	120	12	112	85000	2000	2016-01-16	2018-01-20	High	Medium	2019-11-29
Edit Copy Delete	130	13	113	70000	3000	2019-01-06	2019-06-20	High	Average	2019-11-29
Edit Copy Delete	140	14	114	805000	4000	2016-01-14	2017-01-16	High	Super	2019-11-29
Edit Copy Delete	150	15	115	1555000	5000	2015-01-01	2016-03-26	High	Medium	2019-11-29

Check all With selected: Edit Copy Delete Export

Fig 5.5 Backend insertion

Showroom:

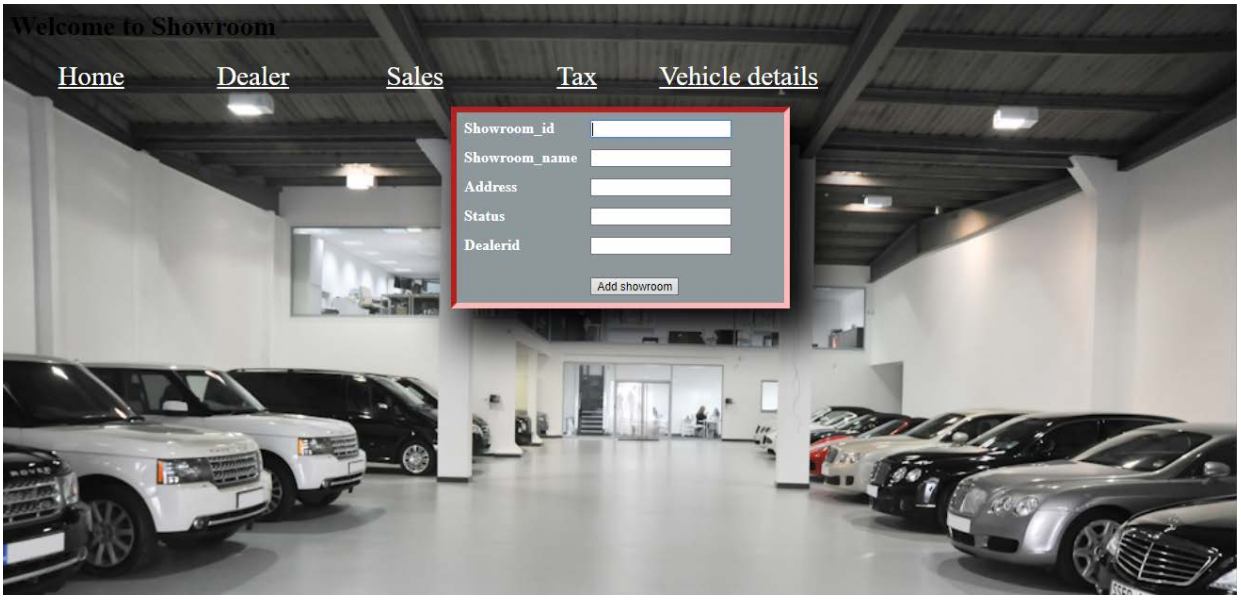


Fig 5.6 Showroom page insertion can be done

+ Options

	Showroom_id	Showroom_name	Address	Status	Dealerid	time
Edit Copy Delete	111	Guru	Bangalore	full	101	2019-11-29
Edit Copy Delete	112	Gayatri	Mysore	Avg	102	2019-11-29
Edit Copy Delete	113	Lakshmi	Tumkur	full	103	2019-11-29
Edit Copy Delete	114	Radha	Bangalore	full	104	2019-11-29
Edit Copy Delete	115	Bhanu	Bangalore	full	105	2019-11-29

Fig 5.7 Backend insertion

Tax:

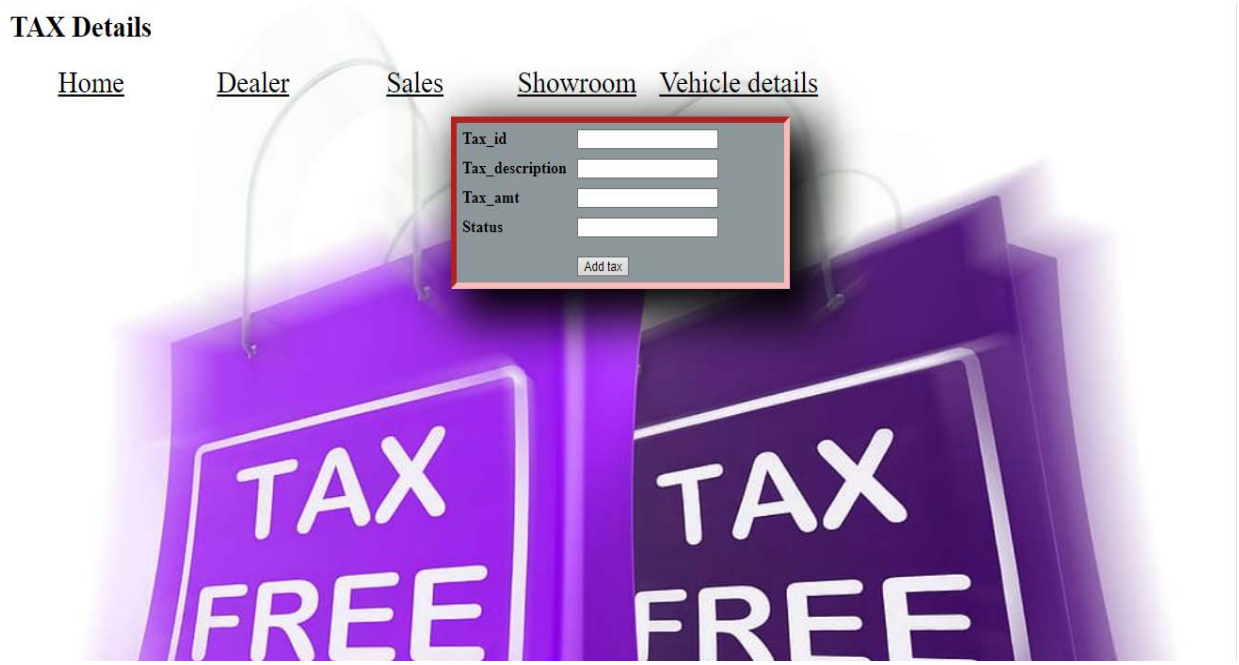


Fig 5.8 Tax page insertion can be done

+ Options

	Tax_id	Tax_description	Tax_amt	Status	time
Edit Copy Delete	1000	All tax	542	Pending	2019-11-29
Edit Copy Delete	2000	Road tax	5000	Pending	2019-11-29
Edit Copy Delete	3000	al tax	5000	Pending	2019-11-29
Edit Copy Delete	4000	Insurance tax	533	Clear	2019-11-29
Edit Copy Delete	5000	All tax	5412	Pending	2019-11-29

Fig 5.9 Backend insertion

Vehicle_Details:

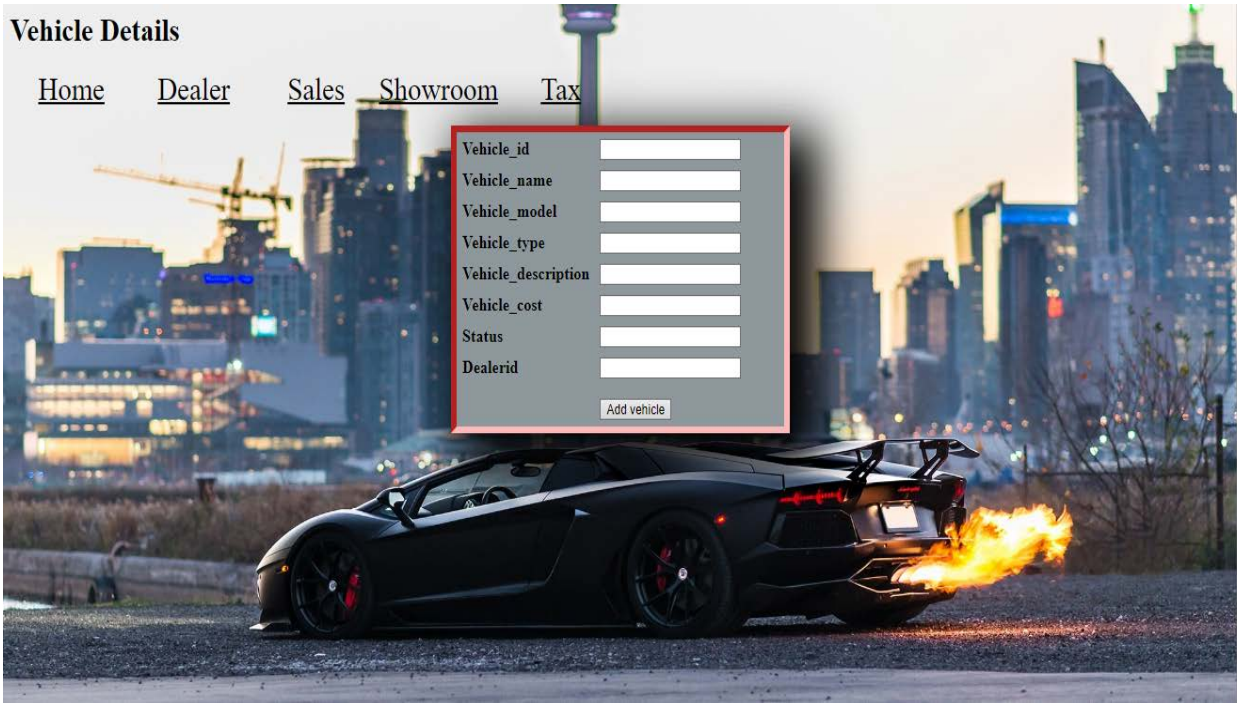


Fig 5.10 Vehicle_details table insertion can be done

Options

	Vehicle_id	Vehicle_name	Vehicle_model	Vehicle_type	Vehicle_description	Vehicle_cost	Status	Dealerid	time
Edit Copy Delete	11	Splender	H456	Bike	Good	45000	Sold out	105	2019-11-29
Edit Copy Delete	12	Pulser	P4569	Bike	Good	85000	Available	105	2019-11-29
Edit Copy Delete	13	RX	R123	Bike	Good	70000	Sold out	103	2019-11-29
Edit Copy Delete	14	Audi	A417	Car	Good	805000	Available	104	2019-11-29
Edit Copy Delete	15	Benz	P4569	Bike	Good	1555000	Sold out	105	2019-11-29

Check all With selected: Edit Copy Delete Export

Fig 5.11 Backend insertion

CHAPTER 6

CONCLUSION AND FUTURE WORK

Analyzing the cost benefit analysis of the current system and the present vehicle database management system, we conclude that present system is the best. Our software is going to serve the long-term needs of showroom. Our software will also reduce the burden of work of data entry operator and makes the whole system effective, efficient and fast. Now by introducing this system selling, buying of vehicles from showroom becomes easier. As a result more DEALER will be attracted towards the “Vehicle Management System”.

The following conclusions can be deduced from the development of the project.

- Automation of the entire system improves the efficiency
- We can provide the communication between Customer and Employee.
- The System has adequate scope for modification in future if it is necessary.
- It stores the available vehicle and their details to be sold.
- It also stores the customer details and show the detail description of vehicle which ever the type of vehicle selected by the customer.
- This is the best way for customer as well as employee to interact with each other without much effort.

Future scope:

An admin can store the data it has all the record for its incoming and outgoing products also it hold the detail information of the customer with their name, id, address,order details. Our software can be used for future reference. Or in future.The project “Vehicle Database Management System” has been tried to develop a robust and fault free system. Several user friendly coding have been adopted in the software development, still enough flexibility has been provided for further enhancements and modifications. As we has mention earlier, the designed forms are typically reflections of the developer, so we strongly believe that the enhancement that has to be done with the design changes, coding changes. there is always some scope of technical modifications in the project that may lead to find code redundancy & storage space minimization.

BIBILOGRAPHY

Web references

- http://www.mysql.com

  For getting handson MySQL and JSON
- http://www.xamppserver.com

  To get MySQL and Apache servers
- http://www.w3schools.com

  To understand the basics of dbms and MySQL
- http://www.stackoverflow.com

  To overcome prolems faced and encounter those

Book references

- **PHP: The Complete Reference** by Steven Holzner
- **PHP & MySQL Web Development** by Luke Welling & Laura Thompson
- **Database Management Systems** 3rd Edition by Raghu Ramakrishnan