GLOBAL ACADEMY OF TECHNOLOGY

Approved by AICTE, New Delhi, Recognized by the Govt. of Karnataka
Autonomous Institute affiliated to VTU, Belagavi,
NAAC Accredited with 'A' Grade
Ideal Homes Township, Rajarajeshwari Nagar, Bengaluru-98

Department of Computer Science and Engineering

(Accredited by NBA 2019-2022)

A

Mini Project Report

on

"CAR SERVICE SYSTEM"

Submitted in partial fulfilment of the requirement for the DBMS Laboratory with Mini Project (20CSEL48) of IV semester

Bachelor of Engineering

In

Computer Science and Engineering

Submitted By

SATWIK KADDI	**(1GA20CS126)**
SHRIRAM NARAYANA BHAT	**(1GA20CS134)**

Under the Guidance of

Dr Manjula K
Associate Professor

GLOBAL ACADEMY OF TECHNOLOGY

Department of Computer Science and Engineering

(Accredited by NBA 2019-2022)

Rajarajeshwari Nagar, Bengaluru – 560 098

CERTIFICATE

Certified that the IV Semester Mini Project in Database Management System with Mini Project Entitled **"CAR SERVICE SYSTEM"** carried out by **SATWIK KADDI** , bearing **USN 1GA20CS126** and **SHRIRAM NARAYANA BHAT**, bearing **USN 1GA20CS134** is submitted in partial fulfilment for the award of the **Bachelor of Engineering** in Computer Science and Engineering during the year 2021-2022.The Database Management System with Mini project report has been approved as it satisfies the academic requirements in respect of the mini project work prescribed for the said degree.

Dr. Manjula K
Associate Professor,
Dept of CSE,
GAT, Bengaluru

Dr. Bhagyashri R Hanji
Professor & Head,
Dept of CSE,
GAT, Bengaluru

External Exam

Name of the Examiner

Signature with date

1.________________________ _________________

2.________________________ _________________

GLOBAL ACADEMY OF TECHNOLOGY

Department of Computer Science and Engineering

(Accredited by NBA 2019-2022)

Rajarajeshwari Nagar, Bengaluru – 560 098

DECLARATION

We, **SATWIK KADDI** bearing USN **1GA20CS126**, **SHRIRAM NARAYANA BHAT** bearing USN **1GA20CS134**, students of Fourth Semester B.E, Department of Computer Science and Engineering, Global Academy of Technology, Rajarajeshwari Nagar Bengaluru, declare that the Mini Project entitled "**CAR SERVICE SYSTEM**" has been carried out by us and submitted in partial fulfilment of the course requirements for the award of degree in Bachelor of Engineering in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi during the academic year 2021-2022.

SATWIK KADDI (1GA20CS126) ………………..

SHRIRAM NARAYANA BHAT (1GA20CS134) ………………..

Place: Bengaluru

Date: 12-09-2022

ABSTRACT

This system "Car services" allows providing car for service. If customer got any problem with his car this system helps service company to store details of service and customer. The real power of this project lies not in direct selling of products that is servicing of cars, but in creating of tighter relationships with customers and delivering of a high level of service and support to their cars, which in turn improves organization sales and its goodwill. A service organization is a business entity that takes care of servicing a customer instrument in the after sales domain of a car. As the number of customers and size of operations increases, the organization divides the geographical area into service areas and branch locations, to allow Engineers to be more responsive to the customer-needs.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant encouragement and guidance crowned our efforts with success.

We consider ourselves proud, to be part of **Global Academy of Technology** family, the institution which stood by our way in endeavours.

We express our deep and sincere thanks to our Principal **Dr. N. Ranapratap Reddy** for his support.

We are grateful to **Dr. Bhagyashri R Hanji**, Professor and HOD, Dept of CSE who is source of inspiration and of invaluable help in channelizing my efforts in right direction.

We wish to thank our internal guide **Dr. Manjula K** , Associate Professor, Dept of CSE for guiding and correcting various documents with attention and care. She has taken lot of pain to go through the document and make necessary corrections as and when needed.

We would like to thank the faculty members and supporting staff of the Department of CSE, GAT for providing all the support for completing the Project work.

Finally, we are grateful to our parents and friends for their unconditional support and help during the course of our Project work.

SATWIK KADDI (1GA20CS126)

SHRIRAM NARAYANA BHAT (1GA20CS134)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO SQL

The Structured Query Language (SQL) is the language of databases. All modern relational databases, including Access, FileMaker Pro, Microsoft SQL Server and Oracle use SQL as their basic building block. In fact, it’s often the only way that you can interact with the database itself. All of the graphical user interfaces that provide data entry and manipulation functionality are nothing more than SQL translators. They take the actions you perform graphically and convert them to SQL commands understood by the database.

SQL IS LIKE ENGLISH

At this point, you might be thinking that you’re not a programmer and learning a programming language is certainly not up your alley. Fortunately, at its core, SQL is a simple language. It has a limited number of commands, and those commands are very readable and are almost structured like English sentences.

INTRODUCING DATABASES

To understand SQL, it’s important to have a basic understanding of how databases work. If you’re comfortable with terms like table, relation and query, feel free to flow right ahead! If not, you may wish to read the article Database Fundamentals before moving on.

Let’s look at an example. Suppose you have a simple database designed to keep the inventory for a convenience store. One of the tables in your database might contain the prices of the items on your shelves indexed by unique stock numbers that identify each item.

You’d probably give that table a simple name like “Prices.”

Perhaps you want to remove items from your store that are priced over $25, you would "query" the database for a list of all these items. This is whereSQL comes in.

YOUR FIRST SQL QUERY

Before we get into the SQL statement required to retrieve this information, let's try phrasing our question in plain English.

We want to "select all stock numbers from the prices table where the price is over $25." That's a simple request when expressed in plain English, and it's almost as simple in SQL. Here's the corresponding SQL statement: SELECT StockNumber

FROM Prices

WHERE Price > 5

It's as simple as that! If you read the statement above out loud, you'll find that it's extremely like the English question we posed in the last paragraph.

INTERPRETING SQL STATEMENTS

Now let's try another example. This time, however, we'll do it backwards. First, I'll provide you with the SQL statement and let's see if you can explain it in plain English: SELECT Price

FROM Prices

WHERE StockNumber = 3006

THE RANGE OF SQL STATEMENTS

SQL provides a wide range of statements, of which SELECT is just one. Here are some examples of other common SQL statements:

SQL INSERT and SQL DELETE: Inserts or deletes a record from a table SQL UPDATE: Modifies records in a table

SQL CREATE and SQL DROP: Creates or deletes a table

In addition to these SQL statements, you can use SQL clauses, among them the WHERE clause used in the previous examples. These clauses serve to refine the type of data to act on. In addition to the WHERE clause, here are other commonly-used clauses:

AND or OR

Combine multiple conditions to refine a SQL statement

LIKE: Compares a value to similar values using a wildcard

ORDER BY: Sorts data in ascending or descending order

If you are interested in further exploring SQL, SQL Fundamentals is a multipart tutorial that explores the components and aspects of SQL in more detail.

1.2 INTRODUCTION TO FRONT END SOFTWARE

Frontend: which is markup showed by clients or users browsers, and for doing this we should use HTML (Hyper Text Markup Language), it just shows some elements for users and doesn't run any functions. When you go to a specific URL, your request is sent to your desired server and it'll render for your HTML of the site, in fact, the server runs any server-side functions.

The Front-End used in this project is HTML along with the CSS and JS language.

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages.
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

Advantages of HTML:

1. The first advantage it is widely used.
2. Every browser supports HTML language.
3. Easy to learn and use.
4. It is by default in every window so you don't need to purchase extra software.
5. You can integrate HTML with CSS, JavaScript, PHP etc. The back-end database used in this project is MySQL

Connectivity (front end and Back end):

PHP is an amazing and popular language!

It is powerful enough to be at the core of the biggest blogging system on the web (Word Press)! It is deep enough to run the largest social network (Facebook)! It is also easy enough to be a beginner's first server side language!

- PHP is an acronym for "PHP: Hypertext Pre-processor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use
- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are executed on the server, and the result is returned to the browser as plain HTML
- With PHP you are not limited to output HTML. You can output images, PDF files, and even flash movies. You can also output any text, such as XHTML and XML

1.3 PROJECT REPORT OUTLINE

The report is arranged in the following way:

Chapter 1: Introduction to SQL about its database, sql query, interpreting sql statements, ANDor OR and range if sql statements
Chapter 2: Requirement specification of hardware and software
Chapter 3: Objective of the Project, design of project and developing
Chapter 4: Implementation of ER diagram and it's description
Chapter 5: Front End Design, connecting to database using PHP, Front end code of the Project
Chapter 6: Testing of project by different cases, it's process and testing objectives

Chapter 7: Outcome of the Project

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 SOFTWARE REQUIREMENTS

Operating System : Windows7 or higher

Database : MYSQL

Tools : Xampp Server, Wamp Server

2.2 HARDWARE REQUIREMENTS

Processor : Any Processor above 500 MHz

RAM : 4.00GB

Hard Disk : 512GB

Compact Disk : 700Mb

Input device : Keyboard

Output device : Laptop Display Screen

CHAPTER 3

OBJECTIVE OF THE PROJECT

The main objective of the Project on Vehicle Service System is to provide better information for the users of this system for better results for their maintenance in the product details that is sales, purchases and stock.

Functionality provided by Car Service System are follows:

- Provides easy entry of data into database and store them.
- Project provides signup and login facility.
- It provides easy availability of data when need.
- User can enter brand and all the details of vehicle which is to be entered into database.
- User can enter details of customer into the database.
- User can view all the data entered to the database.
- User can delete and alter the data already entered.
- In signup facility there is a password option it can secure the user account.
- User can logout after the entry of data.

CHAPTER 4

IMPLEMENTATION

4.1 ER DIAGRAM

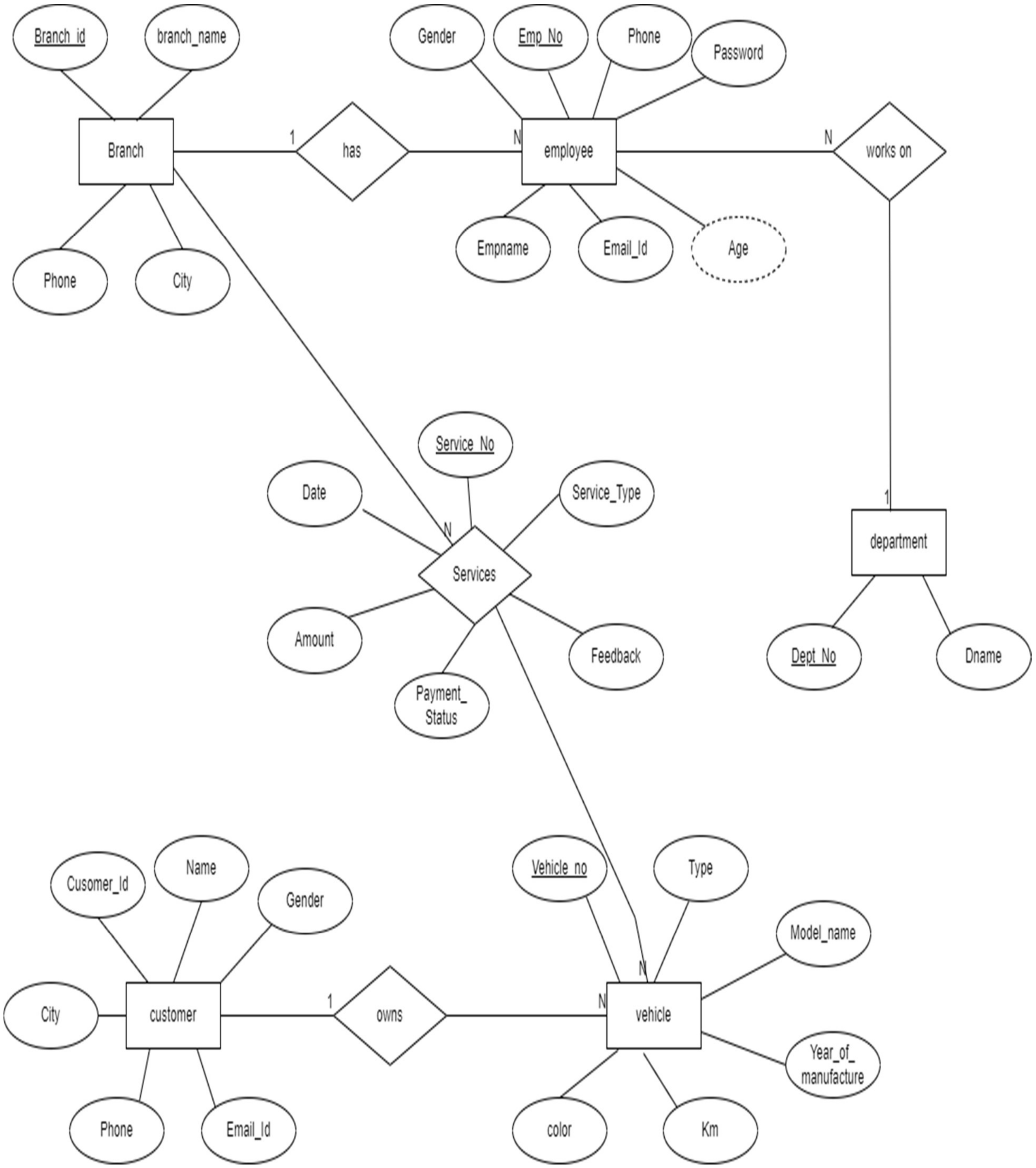


Fig 4.1: ER diagram of CAR SERVICE SYSTEM

ER DIAGRAM:

ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.

It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.

In ER modelling, the database structure is portrayed as a diagram called an entity-relationship diagram.

Entity:

An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

Attribute:

The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.

Key Attribute:

The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.

Composite Attribute:

An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.

Multivalued Attribute:

An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent multivalued attribute.

Relationship:

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

There are four types of relationships:

1. One-to-One Relationship.
2. One-to-many Relationship.
3. Many-to-one Relationship.
4.Many-to-many Relationship.

4.2 MAPPING OF THE ER SCHEMA TO RELATIONS

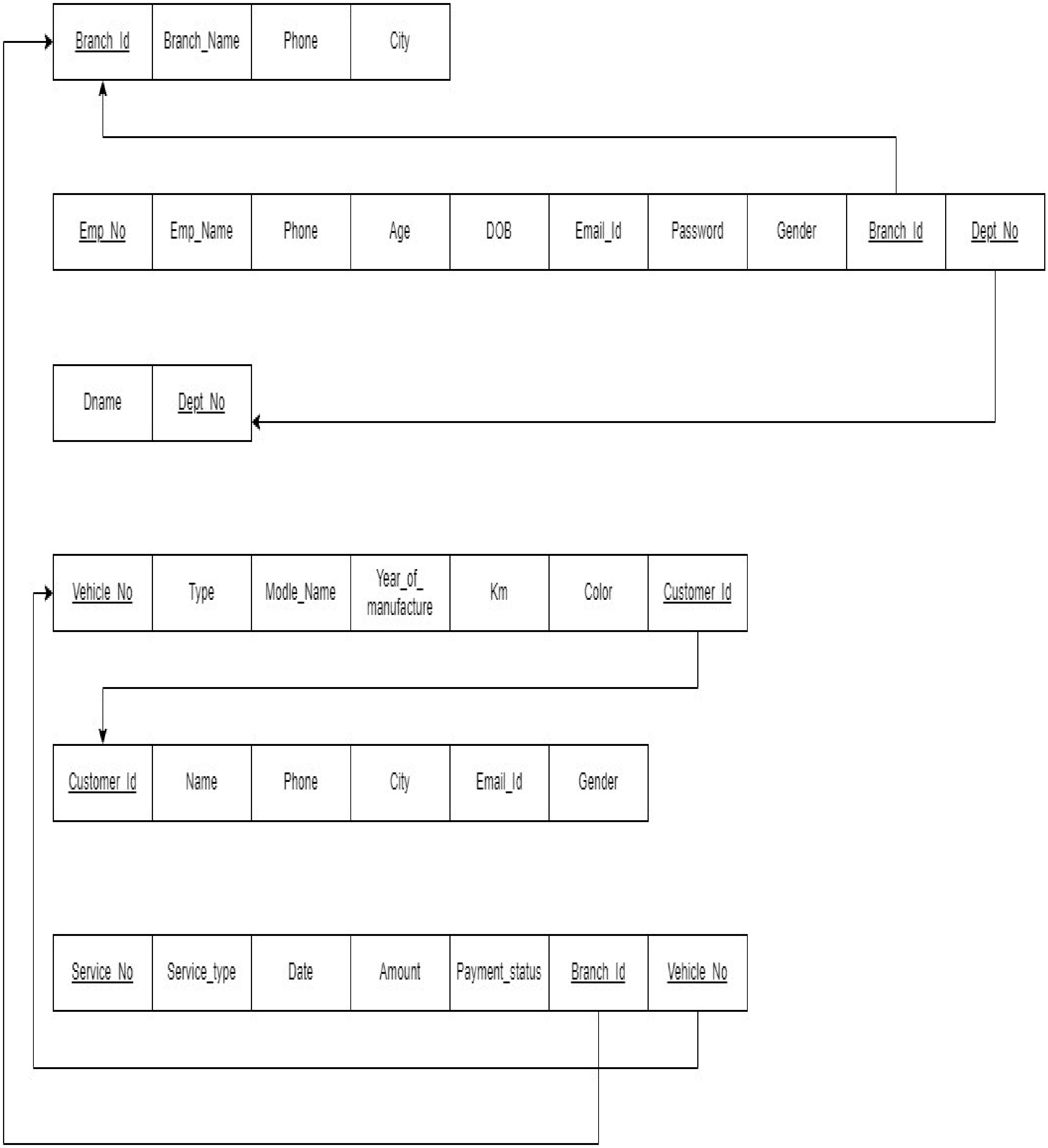


Fig 4.2: Schema Diagram of CAR SERVICE SYSTEM

The design of the database is called a schema. This tells us about the structural view of the database. It gives us an overall description of the database. A database schema defines how the data is organised using the schema diagram. A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database. Schema can be a single table or it can have more than one table which is related. The schema represents the relationship between these tables.

4.3 NORMALIZE THE RELATIONS

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

1. **Update anomalies-**If data items are scattered and are not linked to each other properly, then it could lead to strange situations. For example, when try to update one data item having its copies scattered over several places, a few instances get updated properly while a few others are left with old values. Such instances leave the database in an inconsistent state.

2. **Deletion anomalies-**When tried to delete a record, but parts of it was left undeleted because of unawareness, the data is also saved somewhere else.

3. **Insert anomalies-**When tried to insert data in a record that does not exist at all.

NORMALIZATION FORMS:

1. First Normal Form

First Normal Form is defined in the definition of relations (tables) itself. This rule defines that all the attributes in a relation must have atomic domains. The values in a atomic domain are indivisible units.

2. Second Normal Form

Before learning about the Second Normal Form, need to understand the following

- **Prime attribute -** An attribute, which is a part of the candidate-key, is known as a prime attribute.
- **Non-prime attribute -** An attribute, which is not a part of the prime-key, is said to be a non-prime attribute.

If we follow second normal form, then every non-prime attribute should be fully functionally dependent on prime key attribute. That is, if X -> A holds, then there should not be any proper subset Y of X, for which Y->A also holds true, **partial dependency** is not allowed in Second Normal Form.

3. Third Normal Form

For a relation to be in Third Normal Form, it must be in Second Normal

Form and the following must satisfy

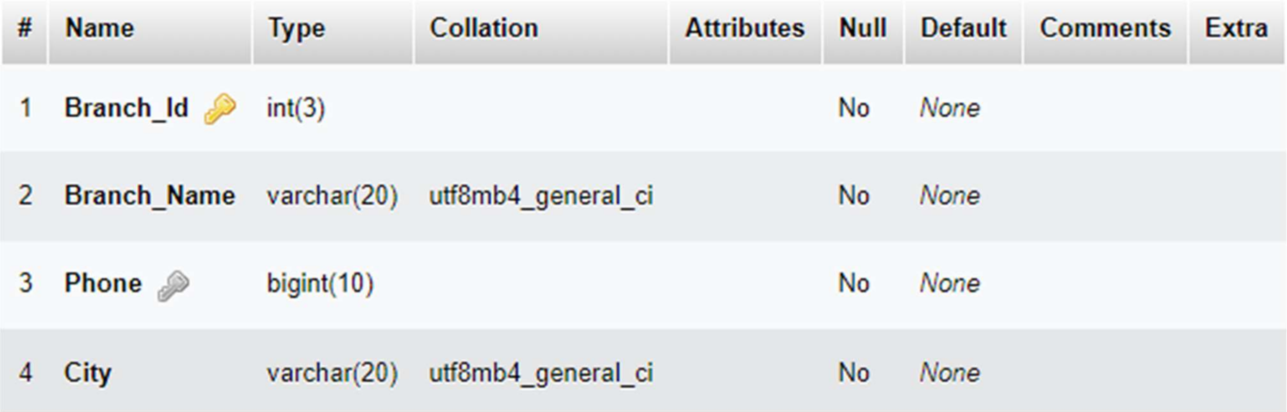
- No non-prime attribute is transitively dependent on prime key attribute.
- For any non-trivial functional dependency, X->A, then either :-

  ★ X is a super key or,

  ★ A is prime attribute, so there not exist a **transitive dependency.**

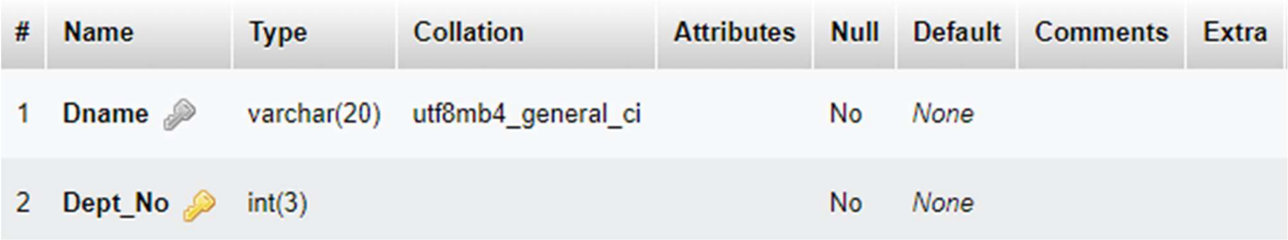
4.4 CREATION OF TABLES

```
CREATE TABLE `branch` (
`Branch_Id` int(3) NOT NULL,
`Branch_Name` varchar(20) NOT NULL,
`Phone` bigint(10) NOT NULL,
`City` varchar(20) NOT NULL
 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



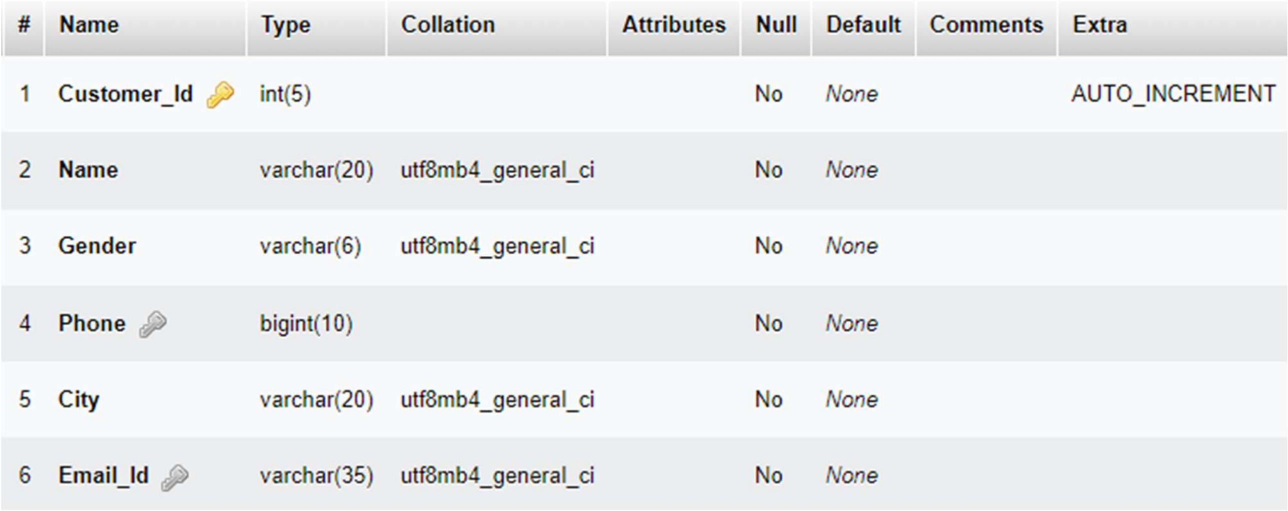
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**Branch_Id** 🔑	int(3)			No	*None*		
2	**Branch_Name**	varchar(20)	utf8mb4_general_ci		No	*None*		
3	**Phone** 🔑	bigint(10)			No	*None*		
4	**City**	varchar(20)	utf8mb4_general_ci		No	*None*		

```
CREATE TABLE `department` (
  `Dname` varchar(20) NOT NULL,
  `Dept_No` int(3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



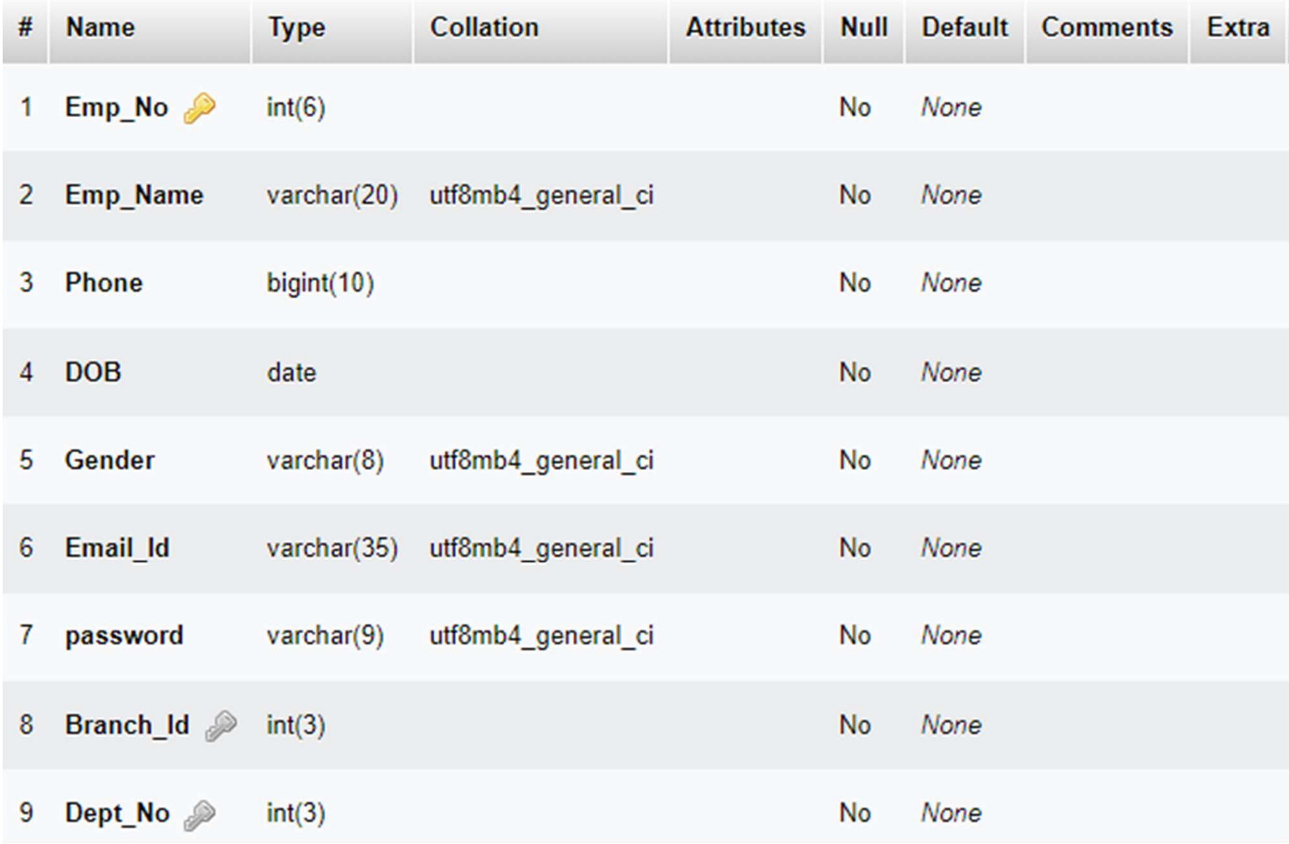
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Dname 🔑	varchar(20)	utf8mb4_general_ci		No	*None*		
2	Dept_No 🔑	int(3)			No	*None*		

```
CREATE TABLE `customer` (
 `Customer_Id` int(5) PRIMARY KEY,
 `Name` varchar(20) NOT NULL,
 `Gender` varchar(6) NOT NULL,
 `Phone` bigint(10) UNIQUE KEY,
 `City` varchar(20) NOT NULL,
 `Email_Id` varchar(35) UNIQUE KEY,
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



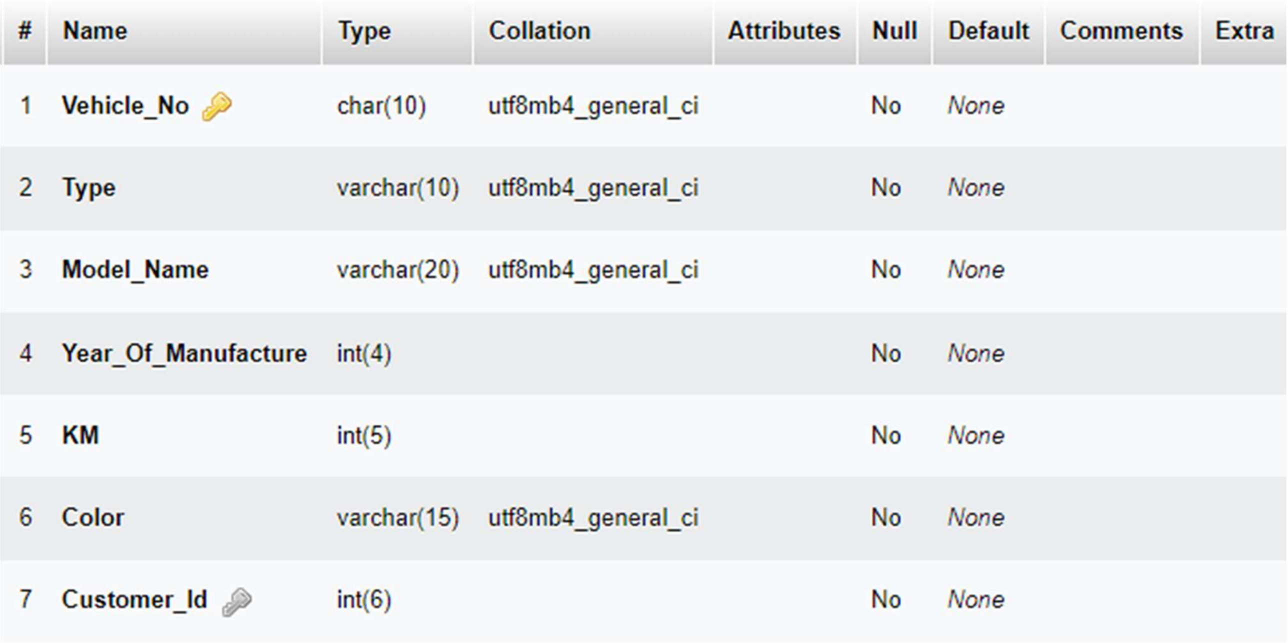
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Customer_Id 🔑	int(5)			No	*None*		AUTO_INCREMENT
2	Name	varchar(20)	utf8mb4_general_ci		No	*None*		
3	Gender	varchar(6)	utf8mb4_general_ci		No	*None*		
4	Phone 🔑	bigint(10)			No	*None*		
5	City	varchar(20)	utf8mb4_general_ci		No	*None*		
6	Email_Id 🔑	varchar(35)	utf8mb4_general_ci		No	*None*		

```
CREATE TABLE `employee` (
 `Emp_No` int(6) PRIMARY KEY,
 `Emp_Name` varchar(20) NOT NULL,
 `Phone` bigint(10) NOT NULL,
 `DOB` date NOT NULL,
 `Gender` varchar(8) NOT NULL,
 `Email_Id` varchar(35) NOT NULL,
 `password` varchar(9) NOT NULL,
 `Branch_Id` int(3) UNIQUE KEY,
 `Dept_No` int(3) UNIQUE KEY,
 FOREIGN KEY (`Branch_Id`) REFERENCES `branch` (`Branch_Id`) ON DELETE CASCADE,
 FOREIGN KEY (`Dept_No`) REFERENCES `department` (`Dept_No`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



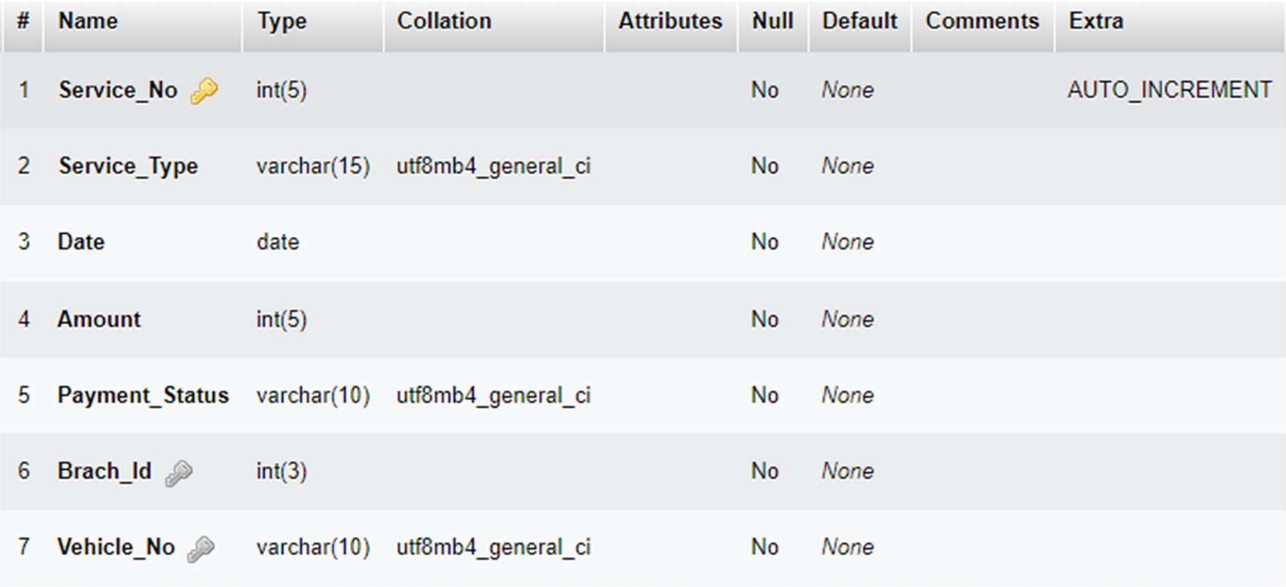
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Emp_No 🔑	int(6)			No	*None*		
2	Emp_Name	varchar(20)	utf8mb4_general_ci		No	*None*		
3	Phone	bigint(10)			No	*None*		
4	DOB	date			No	*None*		
5	Gender	varchar(8)	utf8mb4_general_ci		No	*None*		
6	Email_Id	varchar(35)	utf8mb4_general_ci		No	*None*		
7	password	varchar(9)	utf8mb4_general_ci		No	*None*		
8	Branch_Id 🔑	int(3)			No	*None*		
9	Dept_No 🔑	int(3)			No	*None*		

```
CREATE TABLE `vehicle` (
  `Vehicle_No` char(10) PRIMARY KEY,
  `Type` varchar(10) NOT NULL,
  `Model_Name` varchar(20) NOT NULL,
  `Year_Of_Manufacture` int(4) NOT NULL,
  `KM` int(5) NOT NULL,
  `Color` varchar(15) NOT NULL,
  `Customer_Id` int(6) UNIQUE KEY,
FOREIGN KEY (`Customer_Id`) REFERENCES `customer` (`Customer_Id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**Vehicle_No** 🔑	char(10)	utf8mb4_general_ci		No	*None*		
2	**Type**	varchar(10)	utf8mb4_general_ci		No	*None*		
3	**Model_Name**	varchar(20)	utf8mb4_general_ci		No	*None*		
4	**Year_Of_Manufacture**	int(4)			No	*None*		
5	**KM**	int(5)			No	*None*		
6	**Color**	varchar(15)	utf8mb4_general_ci		No	*None*		
7	**Customer_Id** 🔑	int(6)			No	*None*		

```
CREATE TABLE `services` (
  `Service_No` int(5) PRIMARY KEY AUTO_INCREMENT, AUTO_INCREMENT=1,
  `Service_Type` varchar(15) NOT NULL,
  `Date` date NOT NULL,
  `Amount` int(5) NOT NULL,
  `Payment_Status` varchar(10) NOT NULL,
  `Brach_Id` int(3) UNIQUE KEY,
  `Vehicle_No` varchar(10) UNIQUE KEY,
  FOREIGN KEY (`Brach_Id`) REFERENCES `branch` (`Branch_Id`) ON DELETE CASCADE,
  FOREIGN KEY (`Vehicle_No`) REFERENCES `vehicle` (`Vehicle_No`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Service_No 🔑	int(5)			No	*None*		AUTO_INCREMENT
2	Service_Type	varchar(15)	utf8mb4_general_ci		No	*None*		
3	Date	date			No	*None*		
4	Amount	int(5)			No	*None*		
5	Payment_Status	varchar(10)	utf8mb4_general_ci		No	*None*		
6	Brach_Id 🔑	int(3)			No	*None*		
7	Vehicle_No 🔑	varchar(10)	utf8mb4_general_ci		No	*None*		

INSERT INTO `branch` (`Branch_Id`, `Branch_Name`, `Phone`, `City`) VALUES

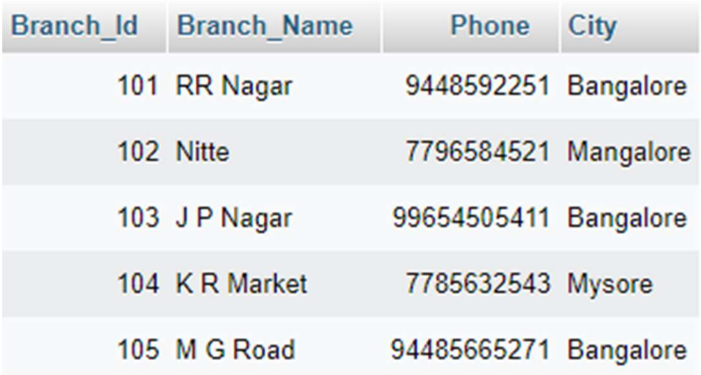
(101, 'RR Nagar', 9448592251, 'Bangalore'),

(102, 'Nitte', 7796584521, 'Mangalore'),

(103, 'J P Nagar', 99654505411, 'Bangalore'),

(104, 'K R Market', 7785632543, 'Mysore'),

(105, 'M G Road', 94485665271, 'Bangalore');

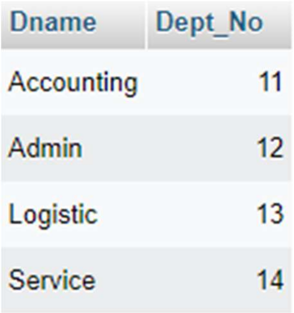


Branch_Id	Branch_Name	Phone	City
101	RR Nagar	9448592251	Bangalore
102	Nitte	7796584521	Mangalore
103	J P Nagar	99654505411	Bangalore
104	K R Market	7785632543	Mysore
105	M G Road	94485665271	Bangalore

INSERT INTO `department` (`Dname`, `Dept_No`) VALUES

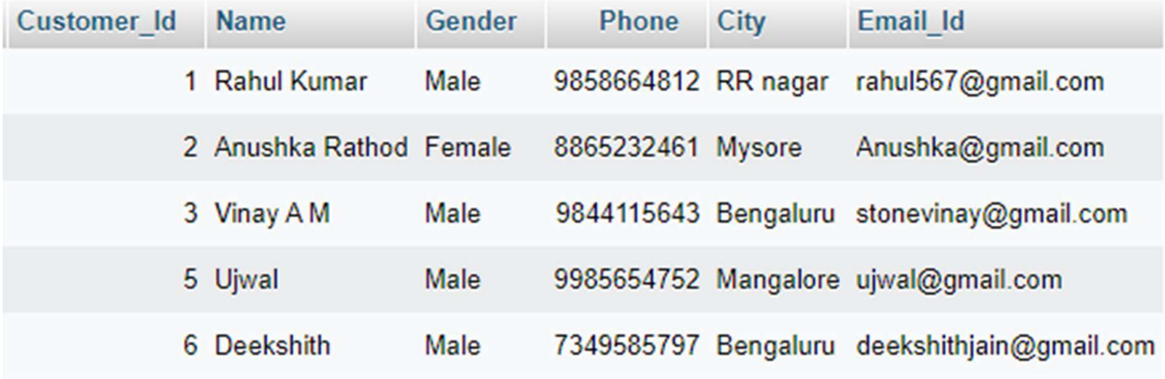
('Accounting', 11),

```
('Admin', 12),

('Logistic', 13),

('Service', 14);
```



Dname	Dept_No
Accounting	11
Admin	12
Logistic	13
Service	14

```
INSERT INTO `customer` (`Customer_Id`, `Name`, `Gender`, `Phone`, `City`, `Email_Id`)
VALUES

(1, 'Rahul Kumar', 'Male', 9858664812, 'RR Nagar', 'rahul567@gmail.com'),

(2, 'Anushka Rathod', 'Female', 8865232461, 'Mysore', 'Anushka@gmail.com'),

(3, 'Vinay A M', 'Male', 9844115643, 'Bengaluru', 'stonevinay@gmail.com'),

(5, 'Ujwal', 'Male', 9985654752, 'Mangalore', 'ujwal@gmail.com'),

(6, 'Deekshith', 'Male', 7349585797, 'Bengaluru', 'deekshithjain@gmail.com');
```



Customer_Id	Name	Gender	Phone	City	Email_Id
1	Rahul Kumar	Male	9858664812	RR nagar	rahul567@gmail.com
2	Anushka Rathod	Female	8865232461	Mysore	Anushka@gmail.com
3	Vinay A M	Male	9844115643	Bengaluru	stonevinay@gmail.com
5	Ujwal	Male	9985654752	Mangalore	ujwal@gmail.com
6	Deekshith	Male	7349585797	Bengaluru	deekshithjain@gmail.com

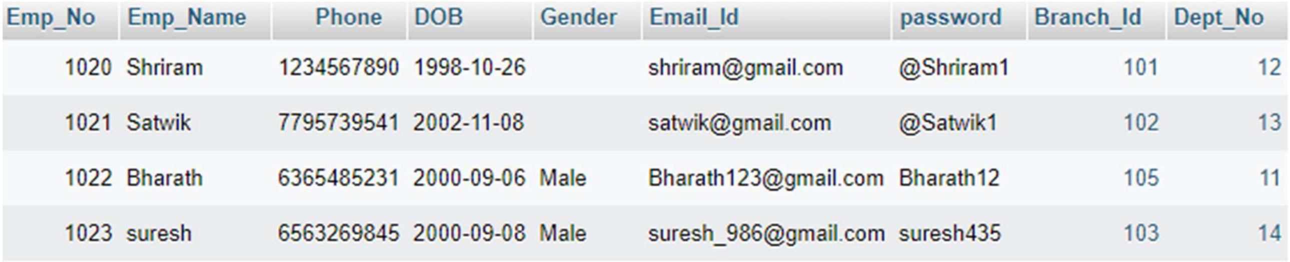
```
INSERT INTO `employee` (`Emp_No`, `Emp_Name`, `Phone`, `DOB`, `Gender`, `Email_Id`,
`password`, `Branch_Id`, `Dept_No`) VALUES
```

(1020, 'Shriram', 1234567890, '1998-10-26', '', 'shriram@gmail.com', '@Shriram1', 101, 12),

(1021, 'Satwik', 7795739541, '2002-11-08', '', 'satwik@gmail.com', '@Satwik1', 102, 13),

(1022, 'Bharath', 6365485231, '2000-09-06', 'Male', 'Bharath123@gmail.com', 'Bharath12', 105, 11),

(1023, 'suresh', 6563269845, '2000-09-08', 'Male', 'suresh_986@gmail.com', 'suresh435', 103, 14);



Emp_No	Emp_Name	Phone	DOB	Gender	Email_Id	password	Branch_Id	Dept_No
1020	Shriram	1234567890	1998-10-26		shriram@gmail.com	@Shriram1	101	12
1021	Satwik	7795739541	2002-11-08		satwik@gmail.com	@Satwik1	102	13
1022	Bharath	6365485231	2000-09-06	Male	Bharath123@gmail.com	Bharath12	105	11
1023	suresh	6563269845	2000-09-08	Male	suresh_986@gmail.com	suresh435	103	14

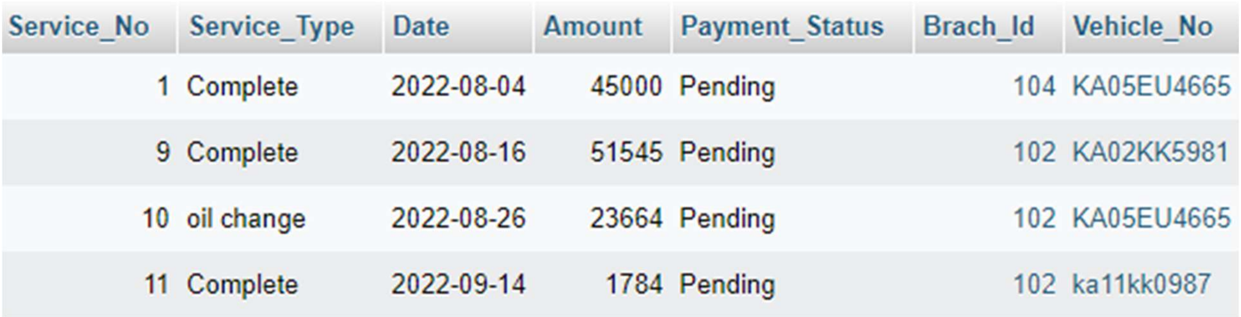
INSERT INTO `services` (`Service_No`, `Service_Type`, `Date`, `Amount`, `Payment_Status`, `Brach_Id`, `Vehicle_No`) VALUES

(1, 'Complete', '2022-08-04', 45000, 'Pending', 104, 'KA05EU4665'),

(9, 'Complete', '2022-08-16', 51545, 'Pending', 102, 'KA02KK5981'),

(10, 'oil change', '2022-08-26', 23664, 'Pending', 102, 'KA05EU4665'),

(11, 'Complete', '2022-09-14', 1784, 'Pending', 102, 'ka11kk0987');

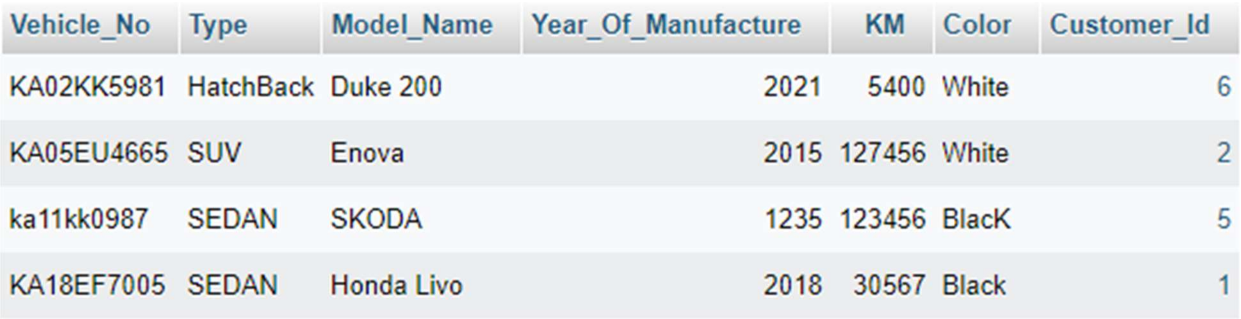


Service_No	Service_Type	Date	Amount	Payment_Status	Brach_Id	Vehicle_No
1	Complete	2022-08-04	45000	Pending	104	KA05EU4665
9	Complete	2022-08-16	51545	Pending	102	KA02KK5981
10	oil change	2022-08-26	23664	Pending	102	KA05EU4665
11	Complete	2022-09-14	1784	Pending	102	ka11kk0987

INSERT INTO `vehicle` (`Vehicle_No`, `Type`, `Model_Name`, `Year_Of_Manufacture`, `KM`, `Color`, `Customer_Id`) VALUES

('KA02KK5981', 'Hatchback', 'Duke 200', 2021, 5400, 'White', 6),

```
('KA05EU4665', 'SUV', 'Enova', 2015, 127456, 'White', 2),

('ka11kk0987', 'SEDAN', 'SKODA', 1235, 123456, 'Black', 5),

('KA18EF7005', 'SEDAN', 'Honda Livo', 2018, 30567, 'Black', 1);
```



Vehicle_No	Type	Model_Name	Year_Of_Manufacture	KM	Color	Customer_Id
KA02KK5981	HatchBack	Duke 200	2021	5400	White	6
KA05EU4665	SUV	Enova	2015	127456	White	2
ka11kk0987	SEDAN	SKODA	1235	123456	BlacK	5
KA18EF7005	SEDAN	Honda Livo	2018	30567	Black	1

4.5 CREATION OF TRIGGERS

```
CREATE TRIGGER `upper_case` BEFORE INSERT ON `customer` FOR EACH ROW

BEGIN

SET NEW.name = UPPER(NEW.name);

END
```

```
CREATE TRIGGER `upper_case` BEFORE INSERT ON `employee` FOR EACH ROW

BEGIN

SET NEW.name = UPPER(NEW.name);

END
```

CHAPTER 5

FRONT END DESIGN

5.1 CONNECTIVITY TO DATABASE

- Most Web Applications :- Retrieve information from a database to alter their on-screen display-Store user data such as orders, tracking, credit card, etc. in a database.

- Permits them to adapt individual users, and provide fresh changing content.

PHP: Built-in Database Access

- PHP provides built in database connectivity for a wide range of databases - MySQL, PostgreSQL, Oracle, Berkeley DB, Informix, MySQL, Lotus Notes, and more - Starting support for a specific database may involve PHP configuration steps.
- Another advantage of using a programming language that has been designed for the creation of web apps.

- Support for each database is described in the PHP manual at :- http://www.php.net/manual/en/

High-Level Process of Using MySQL from PHP

- Create a database connection.

- Select database you wish to use.

- Perform a SQL query.

- Do something processing on query results.

- Close database connection.

CREATING DATABASE CONNECTION

- Use either MySQL_connect or Mysql_pconnect to create database connection

★ MySQL_connect: connection is closed at the end of script (end of page).

★ Mysql_pconnect: creates persistent connection

- connection remains even after the end of the page

- parameters

  ★ Server - hostname of the server.

  ★ Username - username on the database.

  ★ Password - password on the database - New link (MySQL_connect only).

  ★ reuse database connection created by previous call to MySQL_connect - Client Flags.

- MYSQL_CLIENT_SSL :: Use SSL

- MYSQL_CLIENT_COMPRESS :: Compress data sent to MySQL.

SELECTING A DATABASE

- mysql_select_db() - Pass it the database name.
- Related :- mysql_list_dbs()
- List databases available - Mysql_list_tables() ● List database tables available.

PERFORM SQL QUERY

- Create query string - $query = ‘SQL formatted string’ - $query = ‘SELECT*FROM table’.

- Submit query to database for processing - $result = MySQL_query($query); - For UPDATE, DELETE, DROP, etc, returns TRUE or FALSE - For SELECT, SHOW, DESCRIBE or EXPLAIN, $result is an identifier for the results, and does not contain the results themselves.

- $result is called a “resource” in this case.
- A result of FALSE indicates an error.
- If there is an error - mysql_error() returns error string from last MySQL call.

PROCESS RESULTS

- Many functions exist to work with database results.
- mysql_num_rows() - Number of rows in the result set - Usefull for itterating over result set.
- mysql_fetch_array() - Returns a result row as an array.
  - Can be associative or numeric or both (default).
  - $row = mysql_fetch_array($result) :- $row[‘column name’] :: value comes from database row with specified column name.
  - $row[0] :: value comes from first field in result set.

PROCESS RESULTS LOOP

Easy loop for processing results:

```
$result = mysql_query($qstring);
$num_rows = mysql_num_rows($result);  for($i=0;

$i<num_row; $i++)

{

$row = mysql_fetch_array($result)

}
```

CLOSING DATABASE CONNECTION

● mysql_close()

★ closes database connection.

★ Only works for connections opened with mysql_connect().

★ Connections opened with mysql_pconnect() ignore this call.

★ Often not necessary to call this, as connections created by mysql_connect are closed at the end of the script anyway.

5.2 FRONT END CODE

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

Until the 1990s systems design had a crucial and respected role in the data processing industry. In the 1990s standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design.[citation needed] The UML has become the standard language in objectoriented analysis and design.[citation needed] It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations.[citation needed] System design is one of the most important phases of software development process. The purpose of the design is to plan the solution of a problem specified by the requirement documentation. In other words the first step in the solution to the problem is the design of the project.

HTML CODE FOR HOMEPAGE [CAR SERVICE SYSTEM]

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name='viewport' content='width=device-width, initial-scale=1'>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <script src="https://kit.fontawesome.com/a076d05399.js"></script> <!-- sdhgcjs-->
  <title>Admin</title>
</head>
<body>
  <div class="background_">
    <div class="navbar1">
      <?php
      if (!isset($_SESSION['usr_type'])) {
        include("employee.php");
      } else {
        include("admin.php");
      }
      ?>
```

```
</div>
<div class="home" id="home">
  <img src="index.jpg" alt="">
  <div class="content">
    <h1 style="font-weight:bold;">Welcome.<br></br> Have a Good Day, Creww!</h1>
    <h1 style="font-weight:bold;font-size: 35px">Customer Satisfaction is our Motto.</h1>
    <p style="font-weight:bold;font-size: 20px;"> </p>
  </div>
</div>
<div class="contact" id="contact">
  <br>
  <h2>Any Problems? Contact HO.</h2>
  <p></p>
  <br><br>
  <div class="footer">
    <p>
      <i class="fas fa-map-marker-alt"></i> Address <br />
      <?php
      include("connection.php");
      $query1 = "SELECT * FROM branch WHERE Branch_Id = '101' limit 1";
      if (($result1 = mysqli_query($conn, $query1)) && mysqli_num_rows($result1) > 0) {
        while ($row1 = $result1->fetch_row()) {
          $Phone = $row1[2];
          $Address = $row1[3];
          }
        }
        ?><span> <?php echo $Address; ?>
          <br />
          Karnataka, India(IN).
        </span>
      </p>
      <p>
        <i class="fas fa-phone-alt"></i> Mobile No: <br />
        <span> <?php echo $Phone; ?></span>
      </p>
      <p>
        <i class=" far fa-envelope"></i> Mail: <br />
        <span>contact.headoffice@sms.com</span>
      </p>
    </div>
  </div>
 </div>
</body>
</html>
```

PHP CODE FOR HOMEPAGE [VEHICLE SERVICE SYSTEM]

```
<?php

session_start();

if (!isset($_SESSION['Email'])) {

  header("location:login.php");

} else {

  $email = $_SESSION['Email'];

}

?>
```

CSS CODE FOR HOMEPAGE [VEHICLE SERVICE SYSTEM]

```
<style>

    form {

      margin: auto;

      width: 60%;

      overflow: hidden;

      align-items: stretch;

      border-radius: 10px;

    }

   body {
```

```
  font-family: Arial, Helvetica, sans-serif;

  background-size: cover;

  background-repeat: no-repeat;

  background-attachment: fixed;

  background-image: url(./bg1.jpg);

  background-color: whitesmoke;

}

* {

  box-sizing: border-box;

}

input {

  width: 75%S;

}

.container {

  padding: 4px;

  border-radius: 10px;

  margin: 25px;

  background-color: white;
  display: flex;

  flex-flow: column;

}

label {

  font-weight: bolder;

}

input[type=text] {

  padding: 15px;

  margin: 5px 0 22px 0;

  display: block;

  border: none;

  background: #f1f1f1;

  border: 3px solid #ccc;

  width: 30rem;

  margin-top: 6px;

}

input[type=email] {

  padding: 15px;
  margin: 5px 0 22px 0;

  display: block;

  border: none;

  background: #f1f1f1;

  border: 3px solid #ccc;

  width: 30rem;

  margin-top: 6px;

}

input[type=password] {

  padding: 15px;

  margin: 5px 0 22px 0;

  display: block;

  border: none;

  background: #f1f1f1;

  border: 3px solid #ccc;

  width: 30rem;

  margin-top: 6px;

}
```

```
input[type=text]:focus {

  background-color: #ddd;

  outline: none;

  border: 3px solid #555;

}

hr {

  border: 1px solid #f1f1f1;

  margin-bottom: 25px;

}

div {

  padding: 10px 0;

}

hr {

  border: 1px solid #f1f1f1;

  margin-bottom: 25px;

}

.registerbtn {

  background-color: #04AA6D;
  color: white;

  padding: 16px 20px;

  margin: 8px 0;

  border: none;

  cursor: pointer;

  width: 100%;

  opacity: 0.9;

}

.registerbtn:hover {

  opacity: 1;

}

.center {

  display: inline-block;

  text-align: left;

  margin: auto;

}

.background .container .center #branch {

  box-sizing: content-box;
```

```
    height: 5%;

  }

  .background .container .wrong {

    color: red;

    text-align: center;

    font-size: 125%;

  }

</style>
```

CHAPTER 6

TESTING

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TESTING PROCESS

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested.

6.2 TESTING OBJECTIVES

The main objectives of testing process are as follows.

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

● A good test case is one that has high probability of finding undiscovered error.

● A successful test is one that uncovers the undiscovered error.

6.3 TEST CASES

The test cases provided here test the most important features of the project.

Sl No	Test Input	Expected Results	Observed Results	Remarks
1	Insert a Record	New tuple should be inserted	Query OK 1 row affected or inserted	PASS
2	Search a Record	Search from existing Records	Query OK 1 row affected or searched	PASS
3	Update a record	Update from existing records	Query OK 1 row affected or updated	PASS
4	Create Trigger	Trigger Created	Query OK Trigger created	PASS

Table 6.3:Test cases

The test cases are simple and are used to verify incoming data and outgoing data from the function. Various techniques such as cause-effect graphing technique, equivalence partitioning and boundary-value analysis are used to test the functionality of the database.

CHAPTER 7

RESULTS

This section describes the screens of the “CAR SERVICE SYSTEM”. The snapshots are shown below for each module.

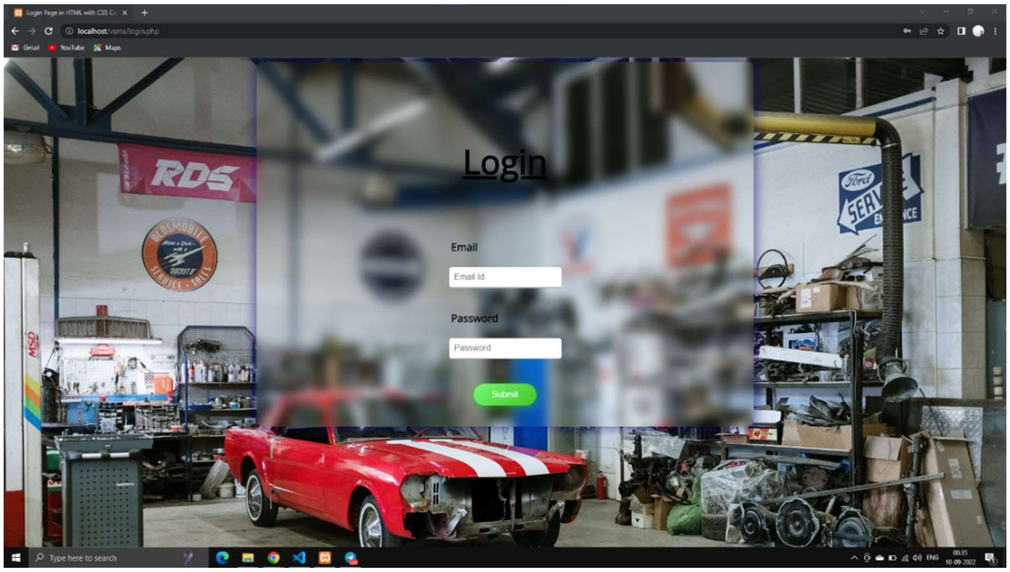
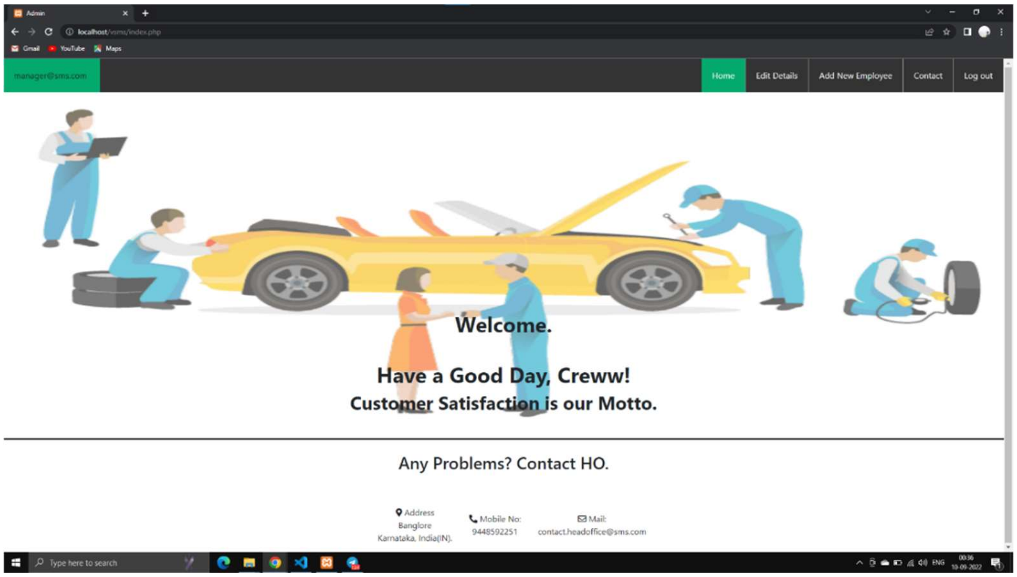


Figure 7.1: LOGIN PAGE
This is the login page, here Employees and Admin can login.



Welcome.

Have a Good Day, Creww!
Customer Satisfaction is our Motto.

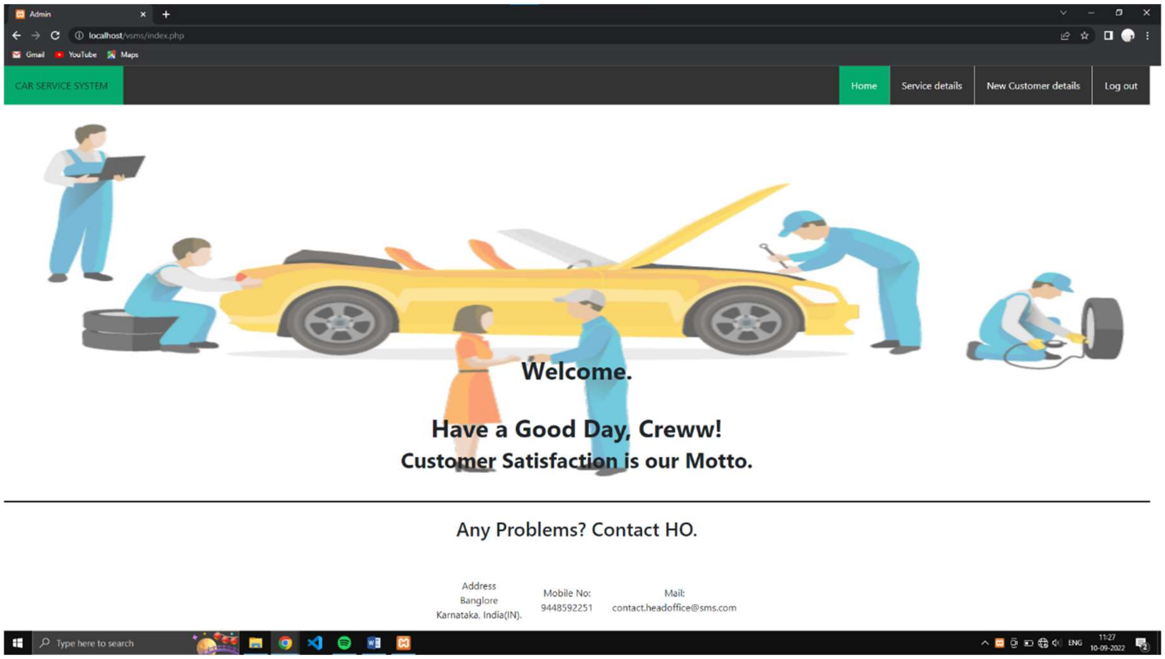
Any Problems? Contact HO.

Address
Banglore
Karnataka, India(IN).

Mobile No:
9448592251

Mail:
contact.headoffice@sms.com

Figure 7.2: Admin Page
This is the Admin page, here admin can add employee details, and edit details in the database.



Welcome.

Have a Good Day, Creww!
Customer Satisfaction is our Motto.

Any Problems? Contact HO.

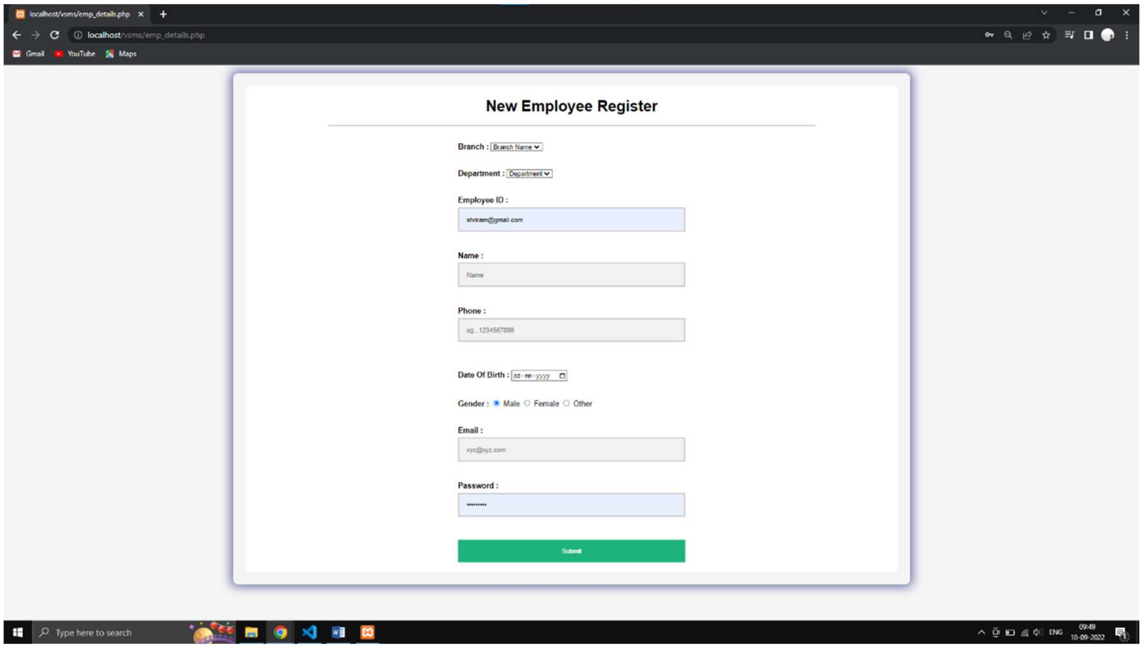
Address
Banglore
Karnataka, India(IN).

Mobile No:
9448592251

Mail:
contact.headoffice@sms.com

Figure 7.3: Employee Page

This is the employee page, here employees can add customer details and service details.


localhost/vsms/emp_details.php
localhost/vsms/emp_details.php
Gmail
YouTube
Maps
Type here to search

New Employee Register

Branch : Branch Name

Department : Department

Employee ID :
shriram@gmail.com

Name :
Name

Phone :
eg., 1234567890

Date Of Birth : dd-mm-yyyy

Gender : ● Male ○ Female ○ Other

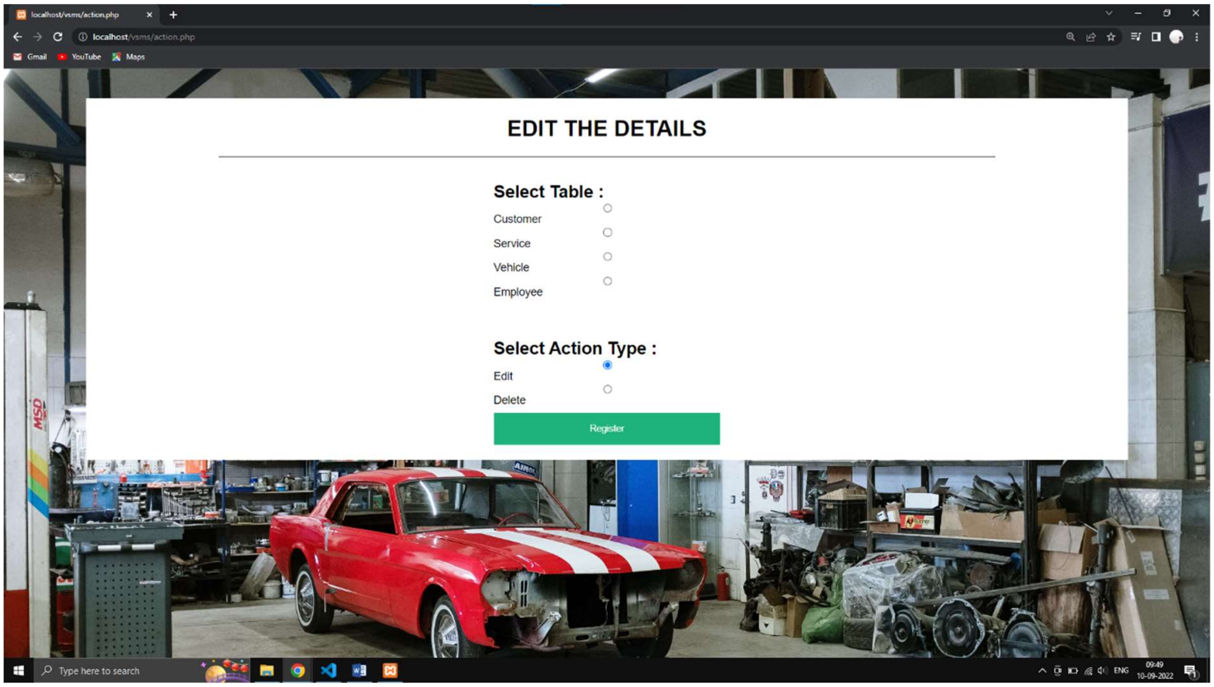
Email :
xyz@xyz.com

Password :
•••••••••

Submit

Figure 7.4: Employee Register Page

This is the Employee Register page, Only Admin has access to this page and enter employees.



EDIT THE DETAILS

Select Table :

Customer ○
Service ○
Vehicle ○
Employee ○

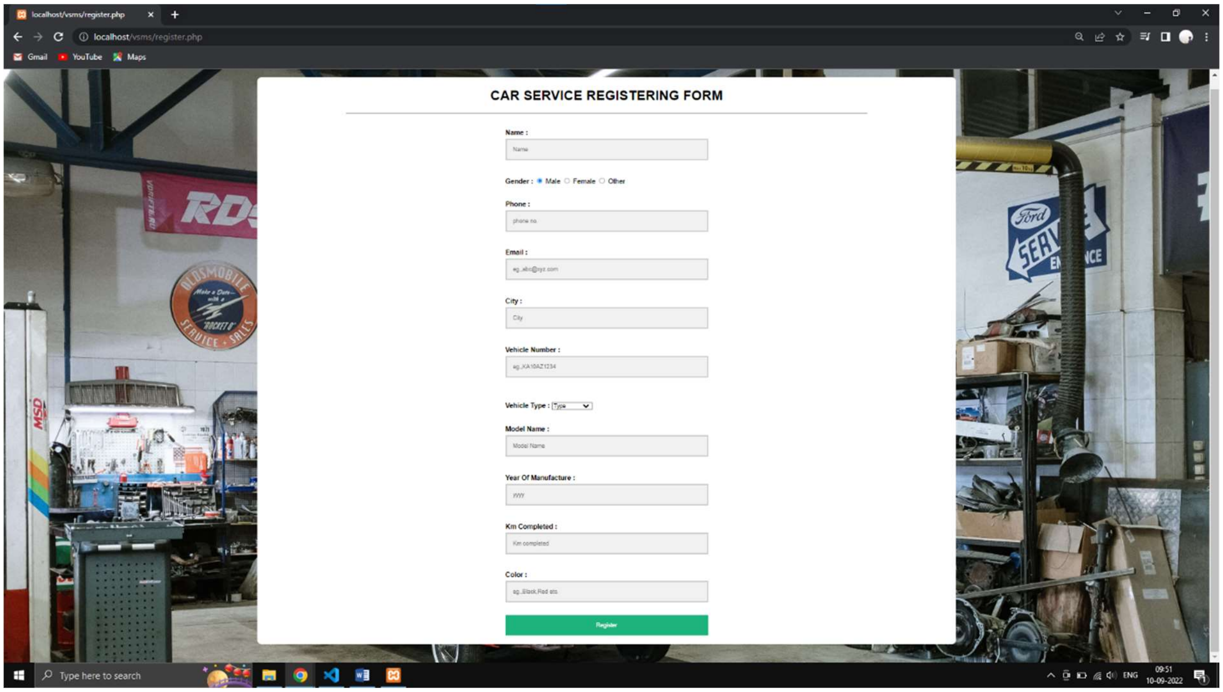
Select Action Type :

Edit ●
Delete ○

Register

Figure 7.5:Edit page

This is the edit page, Only admin can edit all the entries in database.



CAR SERVICE REGISTERING FORM

Name : Name

Gender : ● Male ○ Female ○ Other

Phone : phone no.

Email : eg..abc@xyz.com

City : City

Vehicle Number : eg..KA10AZ1234

Vehicle Type : Type

Model Name : Model Name

Year Of Manufacture : yyyy

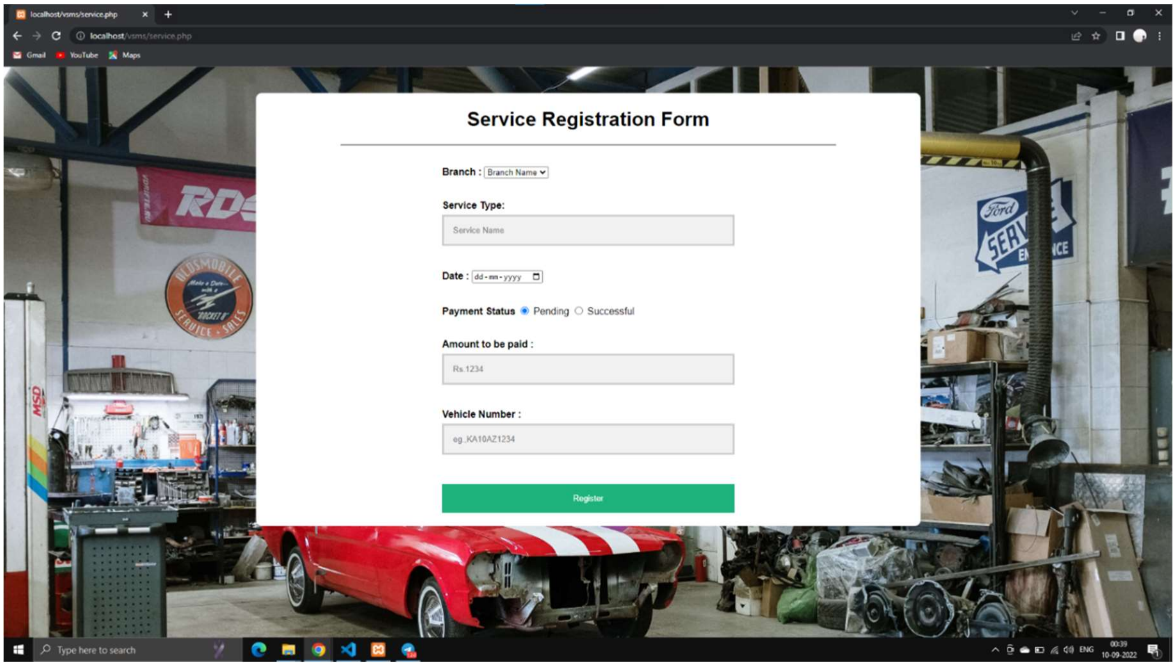
Km Completed : Km completed

Color : eg..Black,Red etc

Register

Figure 7.6: Car Service Customer Page

This is the customer details entry page of cars, Employees can enter the details of customer.



Service Registration Form

Branch : Branch Name

Service Type: Service Name

Date : dd-mm-yyyy

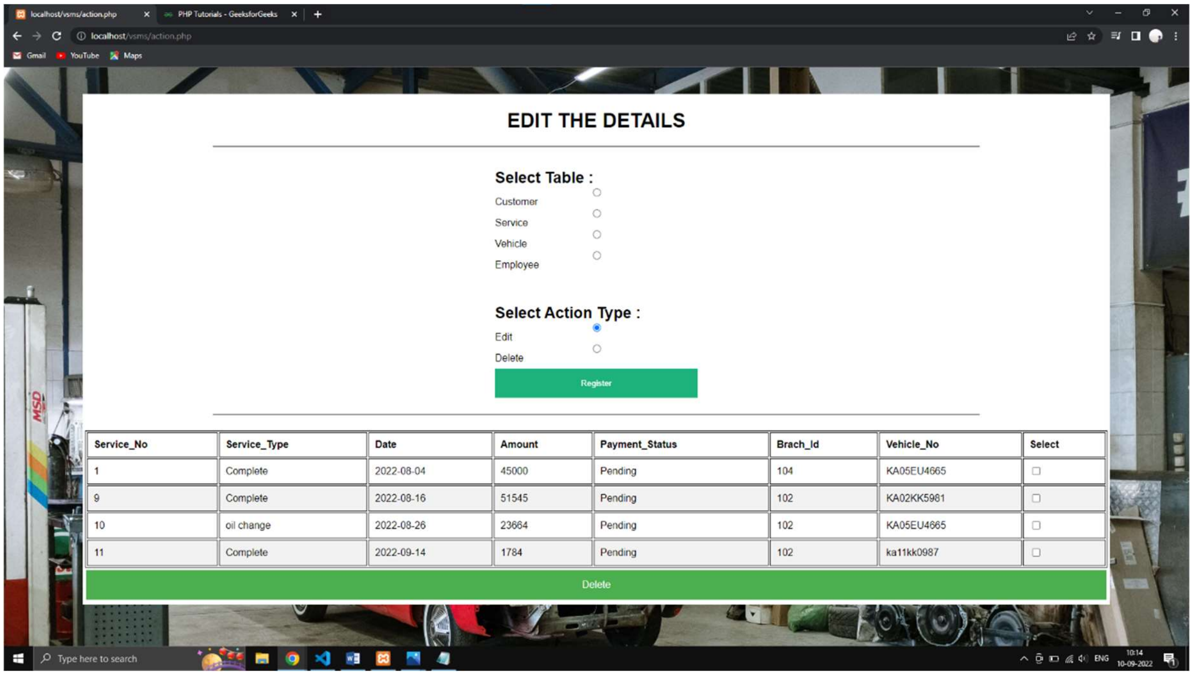
Payment Status ● Pending ○ Successful

Amount to be paid : Rs.1234

Vehicle Number : eg.,KA10AZ1234

Register

Figure 7.7:Service entry page
This is the service entry page, employees can add the service details in this page.


localhost/vsms/action.php
PHP Tutorials - GeeksforGeeks
localhost/vsms/action.php
Gmail
YouTube
Maps
Type here to search
ENG
10:14
10-09-2022

EDIT THE DETAILS

Select Table :
Customer ○
Service ○
Vehicle ○
Employee ○

Select Action Type :
Edit ●
Delete ○

Register

Service_No	Service_Type	Date	Amount	Payment_Status	Brach_Id	Vehicle_No	Select
1	Complete	2022-08-04	45000	Pending	104	KA05EU4665	□
9	Complete	2022-08-16	51545	Pending	102	KA02KK5981	□
10	oil change	2022-08-26	23664	Pending	102	KA05EU4665	□
11	Complete	2022-09-14	1784	Pending	102	ka11kk0987	□

Delete

Figure 7.8:Delete Entries Page
This is the delete page, admin can delete the entries made in the database.

CONCLUSION

With the theoretical inclination of our syllabus it becomes very essential to take the atmost advantage of any opportunity of gaining practical experience that comes along. The building blocks of this Major Project “CAR SERVICE SYSTEM” was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development.

- The planning that goes into implementing a project.
- The importance of proper planning and an organized methodology.
- The key element of team spirit and co-ordination in a successful project.

The project also provides us the opportunity of interacting with our teachers and to gain from their best experience.

REFERENCES

[1] Fundamentals of Database System, Ramez Elmasri and Shamkant B.Navathe, 7th Edition, 2017, Pearson.

[2] Database Management System, Ramakrishna and Gehrke, 3rd Edition, 2014, McGraw Hill.

[3] The Complete Reference PHP by Steven Holzner.

[5] https://www.w3schools.com/css/default.asp

[6] https://www.w3schools.com/html/default.asp

[7] SQL Tutorial (w3schools.com)

[8] https://www.geeksforgeeks.org/html/

[9] W3Schools How TO - Code snippets for HTML, CSS and JavaScript

[10] https://www.geeksforgeeks.org/php-tutorials/

[11] www.freecodecamp.org

[12] Videos in Youtube regarding PHP,HTML and CSS.