

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JNANA SANGAMA”, BELAGAVI-590018



A Mini Project Report

on

ONLINE BIKE SHOWROOM MANAGEMENT SYSTEM

SUBMITTED IN PARTIAL FULFILLMENT FOR 5TH SEMESTER

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

SUBMITTED BY

SANJANA K M.(1JB18IS093)

Under the Guidance of

Guide

Dr. Rekha B.

Professor

Dept. of ISE, SJBIT

Bengaluru -60



SJBIT

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

SJB INSTITUTE OF TECHNOLOGY

BGS HEALTH AND EDUCATION CITY, KENGERI, BENGALURU-560060

KARNATAKA, INDIA.

2020 - 2021

||Jai Sri Gurudev||
Sri Adichunchanagiri Shikshana Trust ®
SJB INSTITUTE OF TECHNOLOGY
BGS Health & Education City, Kengeri, Bengaluru – 60

Department of Information Science & Engineering



SJBIT
CERTIFICATE

Certified that the Mini project work entitled ONLINE BIKE SHOWROOM MANAGEMENT SYSTEM carried out by **Ms. Sanjana K M** bearing USN **1JB18IS093** is a bonafide student of **SJB Institute of Technology** in partial fulfilment for 5th Semester in **INFORMATION SCIENCE AND ENGINEERING** of the **Visvesvaraya Technological University, Belagavi** during the academic year **2020-21**. It is certified that all corrections/suggestions indicated for Internal Assessment 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 said degree.

Dr. Rekha B
Professor
Dept. of ISE, SJBIT

Dr. MOHAN H S
Professor & Head
Dept. of ISE, SJBIT

1. Internal Examiner _____

2. External Examiner _____



ACKNOWLEDGEMENT

I would like to express my profound thanks to his Divine Soul **Padmabhushan Sri Sri Sri Dr. Balagangadharanatha Maha Swamiji** and his Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji** for providing an opportunity to pursue my academics in this esteemed institution.

I would also like to express my profound thanks to **Revered Sri Sri Dr. Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this mini project in this admired institution.

I express my gratitude to **Dr. Ajai Chandran C K**, Principal, SJB Institute of Technology, for providing an excellent facilities and academic ambience which have helped me in satisfactory completion of mini project work.

I extend my sincere thanks to **Dr. Mohan H S**, Head of the Department, Information Science and Engineering for providing an invaluable support throughout the period of mini project work.

I wish to express heartfelt gratitude to my guide, **Dr. Rekha B**, Head of the Department, Information Science and Engineering for her valuable guidance, suggestions and cheerful encouragement during the entire period of this work.

I express my truthful thanks to, **Dr. Vinod H C**, Assistant Professor Department of Information Science and Engineering for his valuable support.

Finally, I take this opportunity to extend my earnest gratitude and respect to my parents, Teaching & Non-teaching staffs of the department, the library staff and all our friends, who have directly or indirectly supported me during the period of this mini project work.

Regards,
Sanjana K M(1JB18IS093)

ABSTRACT

The entitled project “ONLINE BIKE SHOWROOM DATABASE MANAGEMENT SYSTEM” provides solution for showrooms for organizing data using web application. This application helps administrator to update data in step by step process while selling vehicle to customer. Using this system management team can view customer’s information, vehicle information, payment details, and take orders etc. Showroom web is presently used in every car, bike showrooms for computerizing entire system.

Bike showroom management system provides the complete process of selling a bike to a customer from the dealer’s showroom .Before selling, the bike belongs to the fixed asset of the dealer’s showroom. So the main point of this scenario is posting the bike from a fixed asset to a current asset. While executing the process, the dealer can manually maintain the bike business transaction type, which means the bike can be set as a new or used bike to sell to the customer. The scenario of the entire process goes like this. First the dealer purchases the bike from company. The customer wants to buy bike from dealer’s showroom .The dealer creates a sales order for the end customer. The customer confirms the order. The dealer creates a service order to prepare the bike.

TABLE OF CONTENTS

Title	Page No.
1.INTRODUCTION.....	5
• Introduction	
• Objectives	
• Scope	
2.SYSTEM SPECIFICATION.....	8
• Hardware requirements	
• Software requirements	
3.DESIGN.....	14
4.IMPLEMENTATION AND CODING.....	33
• Source code	
5.SYSTEM TESTING.....	35
6.SNAPSHOTS.....	38
CONCLUSIONS	
AND FUTURE	
ENHANCEMENT.....	39
REFERENCES.....	40

LIST OF FIGURES

Figure no.	Description	Page no.
1.4.2	Components of a database	4
3.1.1	Relations	10
3.2	Schema Diagram	11
3.3	ER Diagram	13
6.1	Index page	25
6.2	Customer Login	25
6.3	Dealer_page1	26
6.4	Payment	26
6.5	Customer page	27
6.6	Dealer login	27

LIST OF TABLES

Table no.	Description	Page no.
4.1.1	adddealer	15
4.1.2	admin	15
4.1.3	Bike	16
4.1.4	Customer	17
4.1.5	Payment	17

Chapter 1

INTRODUCTION

This application is used by three users:

- 1)Admin
- 2)Customer
- 3)Dealer

The main aim is to automate the entire day to day activities of showroom like: customer visiting showroom, registration of new customers, adding the available bikes and finally billing and payment. The customer can login to the login page and can book a bike by providing the necessary details, dealer updates the bike records currently present in a showroom, admin is the superuser and he can add a dealer and give his authorization username and password for the respective dealer. This project is ideal for dealers or customers of any size the bike showroom control panel can be accessed anywhere in any me. Bike showroom management system describes the complete process of selling a bike to a customer from the dealer's showroom. "ONLINE BIKE SHOWROOM MANAGEMENT SYSTEM" has been designed to computerize the following functions that are performed by the system:

- Bike details
- Payment details
- Dealer details
- Order summary
- Add and display new bikes for customers
- Customer details

1.1 OBJECTIVES

- A customer based management system is designed to handle all the primary information required to calculate purchase a bike.

- The intention and deletion is to introduce more user friendliness in the process of adding or deleting bike and adding or deleting dealers.
- The orders placed can be viewed by the top level management and middle level management i.e., admin and dealers.
- The dealer will view the bookings and contacts the customer who is in the same city for further process of delivery of bike.

1.2 SCOPE

PERFORMANCE: Manual handling of the record is time consuming and highly prone to error. Hence to improve the performance, computerized and user friendly system is undertaken.

EFFICIENCY: Basic need is efficiency. So whenever the new user submits his/her details, it has to be updated automatically. This record can be used by others instantly.

CONTROL: The complete control is under the admin who has the password to access and illegal access is not supported so only he has the power to add or delete a middle level management. All controls is under the administrator and has the rights just to see the entry and not to change the records or any entries.

SECURITY: This is the main criteria of the proposed system. Since illegal access may corrupt the database. So security has been given

1.3 ADVANTAGES AND DISADVANTAGES

1.3.1 Advantages

The advantages of the project are:

1. Riskless booking of bikes.
2. Reduction of purchase time.
3. Detailed information about the product.
4. Wide variety and different bikes at one place.

1.3.2 Disadvantages

Disadvantages of the project are:

1. Delay in delivery.
2. Lack of significant Discounts on specific products.
3. Physical Quality check of the product cannot be done.
4. Chances of fraud is more.

1.4 DATABASE

A data is a known fact that can be recorded and that have implicit meaning. A database is a collection of related data and the way it is organized. It represents some aspect of real world, sometimes called the mini world or the universe of discourse(UoD). Changes to the mini world are reflected in the database. A database can be of any size and complexity. A database may be generated and maintained manually or it may be computerized. A computerized database may be created and maintained either by a group of application programs written specifically for the task or by a database management system.

1.4.1 DBMS

A database management system (DBMS) is a computerized system that enables users to create and maintain a database. The DBMS is general-purpose software system that facilitates the process of defining, constructing, manipulating, and sharing databases among various users and applications maintained either by a group of application programs written specifically for the task or by a database management system.

A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible. A DBMS provides concurrency, security, data integrity, consistency, controls redundancy and data independence.

Existing DBMSs provide various functions that allow management of a database and its data which can be classified into four main functional groups:

ONLINE BIKE SHOWROOM MANAGEMENT SYSTEM

- **Data definition** - Creation, modification and removal of definitions that define the organization of the data.
- **Update** - Insertion, modification, and deletion of actual data.
- **Retrieval** - Providing information the form directly usable or for further processing by other applications. The retrieved data may be made available in a form basically the same as it is stored in the database or in a new form obtained by altering or combining existing data from the database.
- **Administration** - Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information that has been corrupted by some event such as an unexpected system failure. Both a database and its DBMS inform to the principles of a particular database model. "Database system" refers collectively to the database model, database management system, and database

1.4.2 COMPONENTS OF A DATABASE

- **Users:** Users may be of any kind, such as database administrators, system developers or database users.
- **Database application:** Database application may be Departmental, Personal, Organizational and /or Internal.
- **DBMS:** Software that allows users to create and manipulate database access.
- **Database:** Collection of logical data as a single unit.

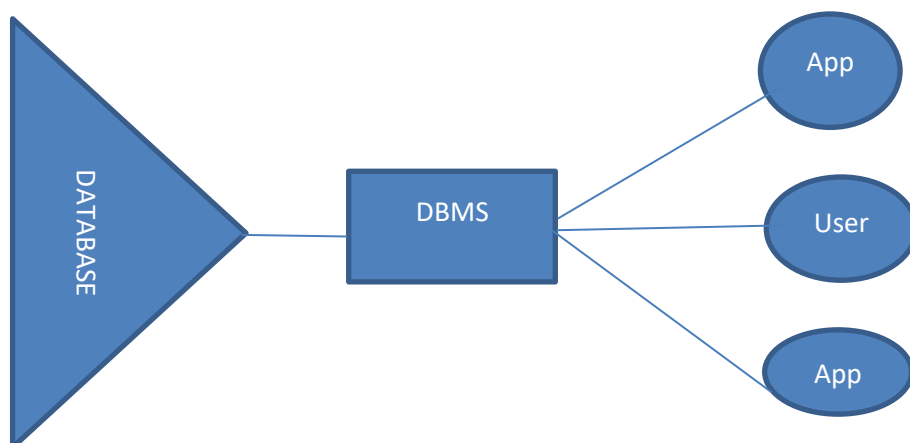


Fig 1.4.2 components of a database

1.4.3 ADVANTAGES OF A DBMS

A Database Management System has many advantages over the traditional file system used in the earlier days, such as:

- **Data independence:** Application programs should be as free or independent as possible from details of data representation and storage. DBMS can supply an abstract view of the data for insulating application code from such facts.
- **Efficient data Access:** DBMS utilizes a mixture of sophisticated concepts and techniques for storing and retrieving data competently and this feature becomes important in cases where the data is stored on external storage devices.
- **Data integrity and security:** If data is accessed through the DBMS, the DBMS can enforce integrity constraints on the data.
- **Data administration:** When several users share the data, integrating the administration of data can offer major improvements. Experienced professionals understand the nature of the data being managed and can be responsible for organizing the data representation to reduce redundancy and make the data to retrieve efficiently.

Chapter 2

SYSTEM SPECIFICATION

2.1 SPECIFIC REQUIREMENTS

A computerized way of handling information about online shopping and customers details is efficient, organized and time saving compared to manual way of doing so. This is done through a database driven web application whose requirements are mentioned in this section.

2.1.1 HARDWARE REQUIREMENTS:

PROCESSOR: Pentium IV or above

MIN RAM: 4 GB

HARD DISK: 1 GB or above

2.1.2 SOFTWARE REQUIREMENTS:

OS: 64-bit windows 10

CODING LANGUAGE:

- **FRONTEND:** HTML, CSS, JAVASCRIPT

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

- **BACKEND:** PHP AND PHPMYADMIN(INNODB)

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994. PHP is a recursive acronym for "PHP: Hypertext Preprocessor". PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases,

session tracking, even build entire e-commerce sites. It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

2.2 HTML

HTML (Hyper Text Markup Language) is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML. It is integrated in PHP. It provides a means to structure text based information in a document. It allows users to produce web pages that include text, graphics and hyperlinks.

2.3 CSS

CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document within a mark-up language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML

2.4 SQL

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for Stream processing in a relational data stream management system (RDSMS). In comparison to older read/write APIs like ISAM or VSAM, SQL offers two main advantages: first, it introduced the concept of accessing many records with one single command; and second, it eliminates the need to specify how to reach a record, e.g. with or without an index. Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

2.5 PHP

PHP (recursive acronym for "PHP: Hypertext Pre-processor"—formerly the acronym for "Personal Home Page") is a server side scripting language designed for web development but also used as a general-purpose programming language. PHP code is interpreted by a web server with a PHP processor module, which generates the resulting webpage: PHP command can be

embedded directly into an HTML source document rather than calling an external file to process data.

2.6 Bootstrap

Bootstrap is a front-end web framework that was created by Twitter for faster creation of device responsive web applications. Bootstrap can also be understood mostly as a collection of CSS classes that are defined in it which can simply be used directly. It makes use of CSS, JavaScript, jQuery etc. in the background to create style, effects and actions for Bootstrap elements. You might know that we use CSS for styling webpage elements and create classes and assign classes to webpage elements to apply style on them. Bootstrap here makes the designing simpler since we only have to include Bootstrap files and mention Bootstrap's predefined class names for our webpage elements and they will be styled automatically through Bootstrap. Through this we get rid of writing our own CSS classes to style web page elements. Most importantly Bootstrap is designed in such a way that it makes your website device responsive. You can also use foundation framework for the same purpose. Bootstrap make you free from writing lot of CSS code and it also saves your time that you spend for designing the webpages.

Chapter 3

DESIGN

3.1 Description of Online Vehicle Showroom Management Database System

- The details of bike, customer, dealer are stored into the respective tables.
- Each entity (Customer, Admin, Dealer, Bike, Payment) contains primary key.
- The entity Bike, Dealer and Payment has binded with foreign key.
- There is one-to-one and one-to-many relationships available between Payment, Dealer, Customers, Bikes.
- All the entities are normalized and reduce duplicity of records.
- Indexing is implemented on each tables of Online Bike showroom Management System tables for fast query execution.

There are many bike showrooms branches in a country for a single owner. Each customer's name, gender, address, contacts are stored. Each customer is provided with a unique customer code.

The customer logs in to his account and books a bike according to his choice. The customer is needed to give details of his email and card details for booking a bike. After booking a bike the dealer can view this details and contact him with by the number or email address provided during the payment and carry out the process of delivering the bike safely to the address given by the customer. The admin is responsible for keeping track of the bikes which are added or deleted by a dealer and order details for the bikes.

3.1.1 Entity, Attributes and Relationships

Entity is a physical object in the real world with an independent existence. Group of distinct or similar objects is called Entity set. Each entity has attributes. Entity type is an object in the real world that can be distinguished from other objects. Each entity has attributes - the particular properties that describe it.

Entities and key attributes:

1. Admin : a_id
2. Dealer: d_id
3. Bike: b_id
4. Customer: c_id
5. Payment: p_id

Relationship

Relationship is an association among two or more entities. Relationship set collects a set of similar relationships into a relationship set. Instance of a relationship set is a set of relationship at a particular point in time.

RELATIONS:

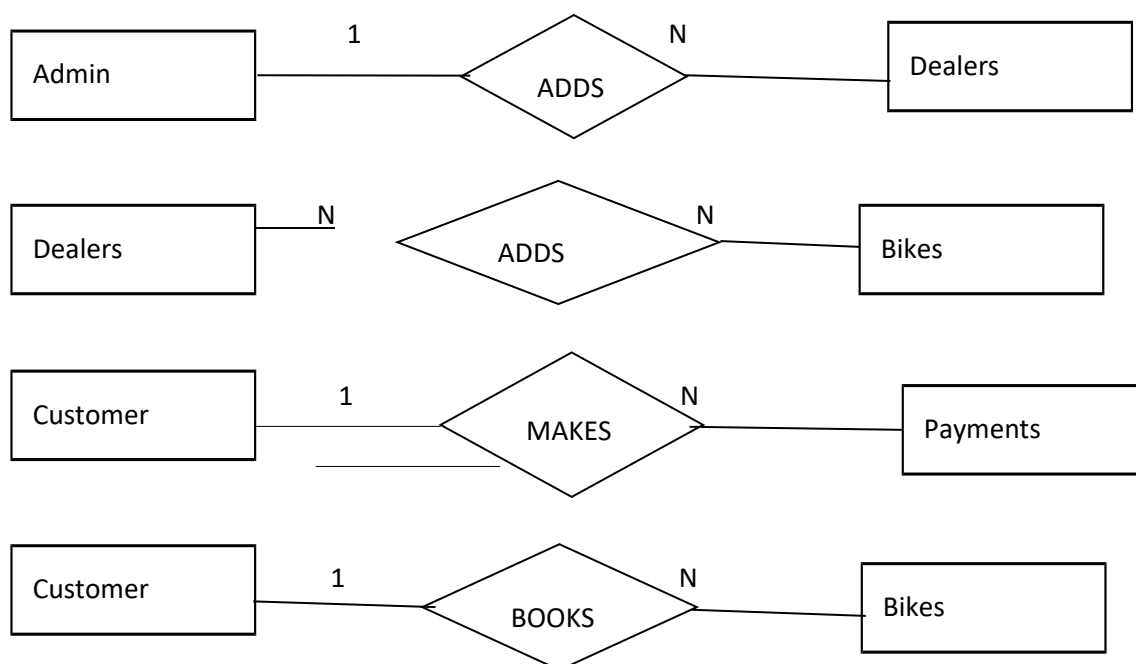


Fig 3.1.1 Relations

ONLINE BIKE SHOWROOM MANAGEMENT SYSTEM

Other attributes:

Customer-c_name,c_gender,c_contactno,c_username,c_password

Dealer-d_name,d_address,d_contactno,d_username,d_password,d_id

Admin-a_id,a_username,a_password,a_contact

Bike-b_image,b_name,b_price,b_description,

Payment- p_name, p_address, p_city, p_state, p_zip, p_nameoncard, p_cardno, p_expmonth, p_expyear, p_cvv, p_price, p_email

3.2.SCHEMA DIAGRAM:

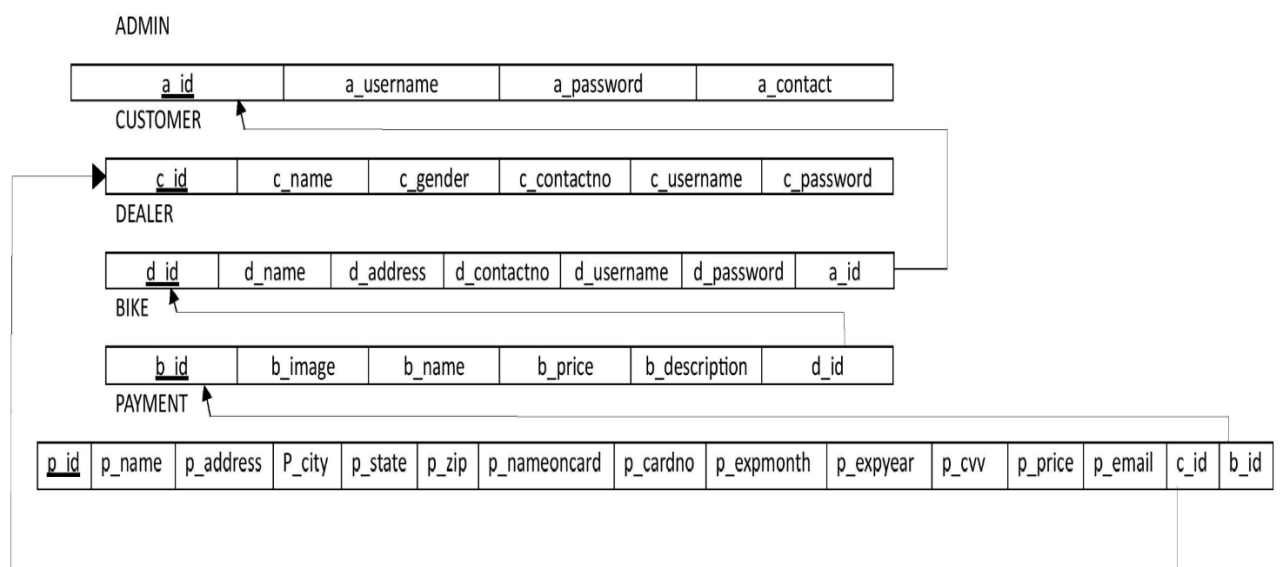


Fig 3.2 Schema diagram

3.3. ER DIAGRAM

An Entity-Relationship model is usually the result of systematic analysis to define and describe what is important to process in area of business. An ER model does not define the business processes; it only presents a business data schema in graphical form. An ER model can also be expressed in a verbal form. Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called “primary keys”. Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagram rather than entity-relationship models. Three main components of ERD are the entities, the relationship between those entities, and the cardinality. Cardinality defines the relation between the entities. The four main cardinal relationships are:

- One-to-One (1:1): For example each customer in a database is associated with one mailing address.
- One-to-many (1:N): For example, a single customer might place an order for multiple products. The customer is associated with multiple entries, but all those entries have a single connection back to the same customer.
- Many-to-One (N:1): For example, many employees will have only one manager above them but one manager can have many employees below him.
- Many-to-Many (M:N): For example, at a company where all call centre agents work with multiple customers. Each agent is associated with multiple, customers, and multiple customers might also be associated with multiple agents. Participation Constraints: There are two types
 - Total participation/Full participation: When all entities are participating in a relationship type
 - Partial participation: When only few entities are participating in a relationship type

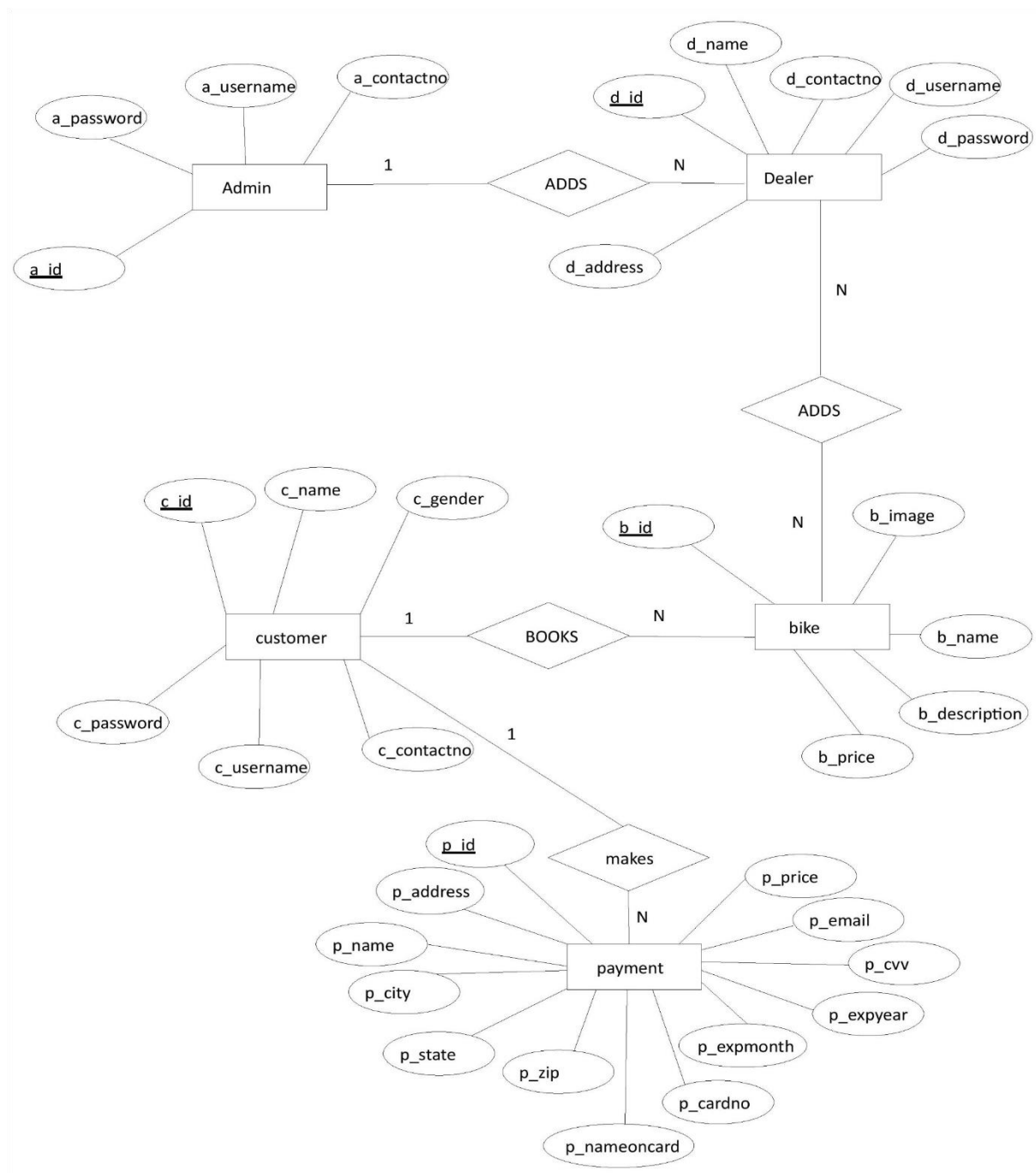


Fig 3.3 ER Diagram

3.4 NORMALISATION

The basic Objectives of normalization are to reduce redundancy, which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of data stored. Relations are normalized so that when relation in the database are to be altered during the lifetime of the database, information is not lost or introduces inconsistencies. The type of alterations normally needed for relation is:

Insertion of new data values to relation. This should be possible without being forced to leave blank fields for some attributes. Deletion of a tuple, namely, a row of a relation. This should be possible without losing vital information unknowingly.

Functional Dependency:

As the concept of dependency is very important, it is essential that it should be understood first and then proceed to the idea of normalization. There is no fool-proof algorithmic method of identifying dependency.

- Properties of normalized relations: Ideals relation after normalization should have the following properties:
- No data values should be duplicated in different rows unnecessarily.
- A value must be specified (and required) for every attribute in a row.
- Each relation should be self-contained. In other words, if a row from a relation is deleted, important information should not be accidentally lost.
- When a row is added to a relation, other relations in the database should not be affected. A value of an attribute in a tuple may be changed independent of other tuples in the relation and other relations.

Consider the PAYMENT table (refer to the schema diagram Fig 3.2). The prime attributes identified are the attributes which is part of candidate key. The non-prime attributes are not part of primary key. There is only one primary key in each table and all the non-key attributes can be derived from the primary key. There is no partial dependency and Hence the table is in 2NF.

Chapter 4

IMPLEMENTATION AND CODING

4.1 SQL

MYSQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is used for the purpose of data definition and data manipulation. Program runs as a server providing multi-user access to a number of databases. MySQL is a multithreaded, multi-user SQL database management system (DBMS). It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents.

Table Creation in SQL

addealer

d_id	d_name	d_address	d_contactno	d_username	d_password	a_id
10	Moni	banglore	9430993213	Moni	moni	0
15	sam	Mumbai	7190443278	Sam	sam	0
20	shanti	chennai	8134509654	Shanti	shanti	0
100	surya	hyderabad	9123459876	Surya	surya	0

Table 4.1.1 addealer

admin

a_id	a_username	a_password	a_contact
1	admin	Admin	123456

Table 4.1.2 admin

ONLINE BIKE SHOWROOM MANAGEMENT SYSTEM

Bike:

b_id	b_image	b_name	b_price	b_description	d_id
1	Bike1	Creg bike	256000	colour-black fuel capacity- 15L	10
1	Bike2	Creg bike	256000	colour-black fuel capacity- 15L	10
10	Bike3	Harley	300000	colour-pitch black 700cc	15
30	Bike4	Yamahar15	182460	colour-black dual channel ABS is max	20
50	Bike5	Kawaski	200000	colour-grey 10000cc	20
80	Bike6	RoyalEnfield	123000	colour-black 10000cc	10
35	Bike7	yamaha	300000	colour-grey 750cc	100
100	Bike8	cregbike	182460	colour-blue 10000cc	10

Table 4.1.3 Bike

ONLINE BIKE SHOWROOM MANAGEMENT SYSTEM

Customer:

c_id	c_name	c_gender	c_contactno	c_username	c_password
0	robin	Male	900123456	robin	robin
0	rachh	female	912345678	rachh	rachh

Table 4.1.4 Customer

Payment:

p_id	p_name	p_address	P_cardno	P_exp date	P_cvv	P_price	P_email	P_cid	B_id
0	sam	Karnataka,Bengaluru	1001	12/23	123	100000	san@gmail.com	10	1
1	Raj	Mysore	2345	10/22	101	200000	raj@yahoo.com	50	10
2	Adi	raichur	4567	9/21	345	182600	Adi4@gmail.com	30	12
3	Moni	Jharkhand	9876	12/22	901	290000	Moni1@yahoo.com	50	34
4	Ritu	Lucknow	1230	8/21	453	300000	Rit1@gmail.com	55	45

Table 4.1.5 payment

4.2 Stored procedures

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again. So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it. You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

```
DELIMITER//
```

```
CREATE PROCEDURE report()
```

```
BEGIN
```

```
SELECT ADMIN.a_username,DEALER.d_name,BIKE.b_name
```

```
FROM DEALER JOIN ADMIN
```

```
ON DEALER.a_id=ADMIN.a_id
```

```
JOIN BIKE ON DEALER.b_id=BIKE.b_id
```

```
ORDER BY BIKE.b_name;
```

```
END//
```

```
DELIMITER;
```


4.3 Triggers

A trigger is a set of actions that are run automatically when a specified change operation (SQL INSERT, UPDATE, or DELETE statement) is performed on a specified table. Triggers are useful for tasks such as enforcing business rules, validating input data, and keeping an audit trail.

```
CREATE TRIGGER discount_price
```

```
Before insert on payment
```

```
for each row
```

```
BEGIN
```

```
SET new.p_price=new.p_price-(new.p_price*0.2);
```

```
END
```

4.4 ASSERTIONS

An assertion is a predicate expressing a condition we wish the database to always satisfy. Domain constraints, functional dependency and referential integrity are special forms of assertion. Where a constraint cannot be expressed in these forms, we use an assertion.

```
Create assertion payment constraint check
```

```
Where not exists(select*from payment)
```

```
Where(p_id>0)
```

```
(select p_id) from payment
```

SOURCE CODE:

adminlogin

```
<?php

    session_start();

    $conn =new mysqli("127.0.0.1","root","","bikeshowroom");

    $usname=$_POST["username"];
    $uspass=$_POST["psw"];

    if ($conn->connect_error)

    {

        die("connection failed:".$conn->connect_error);

    }

    $sql  ="SELECT  a_username,a_password,a_id  FROM  admin  WHERE
a_username='$usname' and a_password='$uspass' ";

    $result      =$conn->query($sql);      if($result-
        >num_rows==1)

    {

        while($row=$result->fetch_assoc()){
            echo"<script>alert('WELCOME ADMIN');

window.location.href='../modules/admin_page1.html';</script>";

            $aid=$row["a_id"];

            $_SESSION["aid"] = $aid;

        }}

        else

        { echo "<script>alert('wrong details'); window.location.href='adminlogin.html';
</script>";
        }

    $conn->close();

?>
```

customerlogin

```
<?php

$conn =new mysqli("127.0.0.1","root","","bikeshowroom");

$username=$_POST["uname"];
$password=$_POST["psw"];

if ($conn->connect_error)

{

    die("connection failed:".$conn->connect_error);

}

    $sql ="SELECT c_username,c_password,c_id FROM customer WHERE
c_username='$username' and c_password='$password' ";

    $result =$conn->query($sql); if($result-
        >num_rows==1)

    {

        while($row=$result->fetch_assoc()){
            echo"<script>alert('WELCOME CUSTOMER');
window.location.href='customer_page1.php';</script>";

            $cid=$row["c_id"];

            session_start();

            $_SESSION["cid"] = $cid;

        }}

        else

        {

            echo "<script>alert('wrong details');
window.location.href='customerlogin.html';</scrip
t>"; }

$conn->close(); ?>
```

Customerregister

```
<?php

$conn =new mysqli("127.0.0.1","root","", "bikeshowroom");

if ($conn->connect_error)

{

    die("connection failed:".$conn->connect_error);

}

$count="";

    $c_name=$_POST["name"];

    $c_gender=$_POST["gender"];

    $c_contactno=$_POST["contact"];

    $c_username=$_POST["username"];

    $c_password=$_POST["psw"];

    $sql ="SELECT * FROM customer WHERE c_username='$c_username' ";

    $result = $conn->query($sql); if($result->num_rows==1)

    {

        echo"<script>alert('user already present please choose another username');

window.location.href='customerregister.html';</script>";

        $count++;

    }

    if($count==0){

        $sql ="insert into

customer(c_name,c_gender,c_contactno,c_username,c_password)

values('$c_name','$c_gender','$c_contactno','$c_username','$c_password')"; if($conn->query($sql)===TRUE)

        {
```

```
        echo "<script>alert('Registration Successfull');  
window.location.href='customerlogin.html';</script>";  
  
    }  
  
    else  
  
    {  
  
        echo "Error in insertion". $sql.$conn-  
        >error; }  
  
    }  
  
$conn->close(); ?>
```

dealerlogin

```
<?php  
session_start();  
  
$conn =new mysqli("127.0.0.1","root","","bikeshowroom");  
  
$usname=$_POST["uname"];  
$uspass=$_POST["psw"];  
  
if ($conn->connect_error)  
  
    {  
  
        die("connection failed:".$conn->connect_error);  
  
    }  
  
    $sql ="SELECT * FROM dealer WHERE d_username='$usname' and  
d_password='$uspass' ";  
  
    $result =$conn->query($sql); if($result-  
        >num_rows==1)  
  
    {  
  
        while($row=$result->fetch_assoc()){  
  
            echo"<script>alert('WELCOME DEALER');  
window.location.href='../modules/dealer_page1.php';</script>";  
  
            $d_id=$row["d_id"];
```

```
echo $d_id;

$_SESSION["did"] = $d_id;

}}

else

{

    echo "<script>alert('wrong
details');
window.location.href='dealerlogin.html';</scrip
t>"; }

$conn-

>close(); ?>
```

showbike

```
<html>

<head>

<style>
.cancelbtn {
width: auto;
padding: 10px 50px;
background-color:
red;
cursor:pointer;

}

button {  color:
white;  padding:
14px 20px;
margin: 8px 0;
border: none;
cursor:pointer;
width: 100%;

    opacity: 1;
```

```
}

</style>
</head>

</html>

<?php
session_start();

if($_SESSION["cid"]==""){
    echo"you must login first";
}

else{

    $conn =new mysqli("127.0.0.1","root","","bikeshowroom");

    if ($conn->connect_error)

    {

        die("connection failed:".$conn-
        >connect_error);  }

        $sql ="SELECT * FROM bike ";

        $result =$conn->query($sql); if($result-
        >num_rows>0)

        { while($row=$result->fetch_array())

        {

            echo '<center>


</center><br>';

            echo" <center> name:" . $row["b_name"]. " <br>". " price:"
.$row["b_price"]. "<br>".
```

```
" description:" . $row["b_description"]. "<br><br>";

    $price=$row[3];

    $bid=$row[0];

    $bnam=$row[2];

    echo "<a href='../payment.php?val=$price&&bid=$bid&&nam=$bnam'><button
class='cancelbtn' type='submit' name='book'>BOOK</button></a><br><br> </center>";

    }

    }

    else

    { echo "no bikes are present";

    }

$conn->close();

}

?>
```

mybookings

```
<?php

session_start();

$cid=$_SESSION["cid"];

$conn =new

mysqli("127.0.0.1","root","","bikeshowroom"); if

($conn->connect_error)

{

    die("connection failed:".$conn->connect_error);

}

$sql ="CALL `mybookings`($cid) ";
```


ONLINE BIKE SHOWROOM MANAGEMENT SYSTEM

```
$result=$conn->query($sql); if($result-
    >num_rows>0)

    {
        echo"<center><u>MY
BOOKINGS</u></center>"; while($row=$result-
>fetch_assoc())

    {

        echo" payment id:" . $row["p_id"]. "<br>";          echo"
Name:" . $row["p_name"]. "<br>";          echo" Price:"
$row["p_price"]. "<br>";          echo" Address:"
$row["p_address"]. "<br>";          echo" bike name:"
$row["b_name"]. "<br>";          echo" bike description:"
$row["b_description"]. "<br><br><br>";

    }

    echo"<center><a href='customer_page1.php'><button class='cancelbtn'
type='submit' name='back'>Back</button></a><br><br> </center>";

    }

    else

    { echo "no bookings are present";

    }

$conn->close(); ?>
```

deletebike

```
<?php

$conn =new mysqli("127.0.0.1","root","","bikeshowroom");

if ($conn->connect_error)

{

    die("connection failed:".$conn->connect_error);
```

```
} else { echo
"conneceted!!!!";

}

$b_id=$_POST["id"];

$sql="delete from bike where b_id='$b_id'"; if($conn-
>query($sql)===TRUE)

{

    echo "<script>alert('Bike removed!!!!');
window.location.href='dealer_page1.php';</script>";

}

else

{ echo "Error in deletion". $sql.$conn->error;
}

$conn-
>close(); ?>
```

adddealer

```
<?php

    session_start();

    $conn =new mysqli("127.0.0.1","root","","bikeshowroom");

    if ($conn->connect_error)

    {

        die("connection failed:".$conn->connect_error);

    }

    $d_id=$_POST["id"];

    $d_name=$_POST["name"];

    $d_address=$_POST["address"];

    $d_contactno=$_POST["contact"];
```

```
$d_username=$_POST["username"];

$d_password=$_POST["psw"];

$a_id=$_SESSION["aid"];

$count="";

$sql="SELECT * FROM dealer WHERE d_username='$d_username' ";
$result=$conn->query($sql);

if($result->num_rows==1)

{

    echo"<script>alert('DEALER already present');

window.location.href='add_dealer.html';</script>";

    $count++;

}

if($count==0){

    $sql="insert into dealer

(d_id,d_name,d_address,d_contactno,d_username,d_password,a_id)

values('$d_id','$d_name','$d_address','$d_contactno','$d_username','$d_password','$a_id'

)"; if($conn->query($sql)===TRUE)

    { echo "<script>alert('Dealer Added Successfully');

window.location.href='admin_page1.html';</script>";

    }

    else

    {

        echo "Error in insertion". $sql.$conn-

>error; }

}

$conn-

>close(); ?>
```

deleteddealer

```
<?php

$conn =new mysqli("127.0.0.1","root","","bikeshowroom");

if ($conn->connect_error)

{

    die("connection failed:".$conn->connect_error);

} else { echo
"conneceted!!!!";

}

    $d_id=$_POST["id"];

    $sql ="delete from dealer where d_id='$d_id'";

    if($conn->query($sql)===TRUE)

    {

        echo "<script>alert('Dealer removed!!!!');
window.location.href='admin_page1.html';</script>";

    }

    else

    { echo "Error in deletion". $sql.$conn->error;
    }

$conn->close();

?>
```

showbookings

```
<?php

$conn =new mysqli("127.0.0.1","root","","bikeshowroom");

if ($conn->connect_error)

{
```

ONLINE BIKE SHOWROOM MANAGEMENT SYSTEM

```
die("connection failed:".$conn->connect_error);

}

$sql = "SELECT p.*,b.b_name,b.b_description,c.c_contactno FROM payment p,
bike b,customer c WHERE p.b_id=b.b_id and p.c_id=c.c_id ";

$result = $conn->query($sql); if($result-
    >num_rows>0)

    { while($row=$result->fetch_assoc())

        {

            echo '<center>
</center><br>';

            echo"<u>PAYMENT DETAILS</u> <br>";
            echo" Payment ID:" . $row["p_id"]. " <br>";
            echo" Price:" . $row["p_price"]. "<br>";
            echo" Name:" . $row["p_name"]. " <br>";
            echo" Address:" . $row["p_address"]. "<br>";
            echo" City:" . $row["p_city"]. "<br>";          echo"
State:" . $row["p_state"]. "<br>";          echo" Zip:"
$row["p_zip"]. "<br>";          echo" Contact no:"
$row["c_contactno"]. "<br>";          echo" Email:"
$row["p_email"]. "<br>";          echo"<u>BIKE
DETAILS</u> <br>";          echo"Bike name:"
$row["b_name"]. "<br>";

            echo" Desctiption:" . $row["b_description"]. "<br><br><br>"

        }

        echo"<center><button    onclick='back()'
name='back'>back</button><script>function
back() { window.history.back();}</script><br><br>
</center>";

    }

    else

        { echo "no bookings are present";

        }

}
```

```
$conn-  
>close(); ?>
```

payment

```
<?php  
session_start();  
  
$conn =new mysqli("127.0.0.1","root","","bikeshowroom");  
  
if ($conn->connect_error)  
{  
    die("connection failed:".$conn->connect_error);  
}  
  
    $p_name=$_POST["name"];  
    $p_email=$_POST["email"];  
    $p_address=$_POST["address"];  
    $p_city=$_POST["city"];  
    $p_state=$_POST["state"];  
    $p_zip=$_POST["zip"];  
    $b_id=$_POST["bid"];  
    $p_nameoncard=$_POST["cardname"];  
    $p_cardno=$_POST["cardnumber"];  
    $p_expmonth=$_POST["expmonth"];  
    $p_expyear=$_POST["expyear"];  
    $p_cvv=$_POST["cvv"];  
    $p_price=$_POST["price"];  
    $customerid=$_SESSION["cid"];
```

```
$sql = "insert into payment
(p_name,p_email,p_address,p_city,p_state,p_zip,p_nameoncard,p_cardno,p_expmonth,p
_ex pyear,p_cvv,p_price,c_id,b_id) values
('$p_name','$p_email','$p_address','$p_city','$p_state','$p_zip','$p_nameoncard',
$p_cardno,$p_expmonth,$p_expyear,$p_cvv,$p_price,$customerid,$b_id)";
if($conn->query($sql)===TRUE)
{
    echo "<script>alert('Order placed successfully!!!!');</script>";
    echo"THANK YOU, OUR DEALER WILL CONTACT YOU
    SOON.....<br><br>"; echo"<center><button
    onclick='back()'>logout</button><script>function back()
    { window.location.replace('../index.html');}</script><br>
    <br><br> </center>";
    session_destroy();
    unset($_SESSION['cid']);
}
else
{
    echo "Error in insertion". $sql.$conn->error; }$conn->close(); ?>
```

Chapter 5

SYSTEM TESTING

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behaviour of a component or a system when it is tested. System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behaviour of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS).

5.1 Unit testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

5.2 Integration testing

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing

Integration testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. See also component integration testing, system integration testing.

- Component integration testing: Testing performed to expose defects in the interfaces and interaction between integrated components
- System integration testing: Testing the integration of systems and packages; testing interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

5.3 User acceptance testing Users or client

This could be either someone who is buying a product (in the case of commercial software) or someone who has had a software custom built through a software service provider or the end user if the software is made available to them ahead of the time and when their feedback is sought out. The team can be comprised of beta testers or the customer should select UAT members internally from every group of the organization so that each and every user role can be tested accordingly.

Chapter 6

SNAPSHOTS

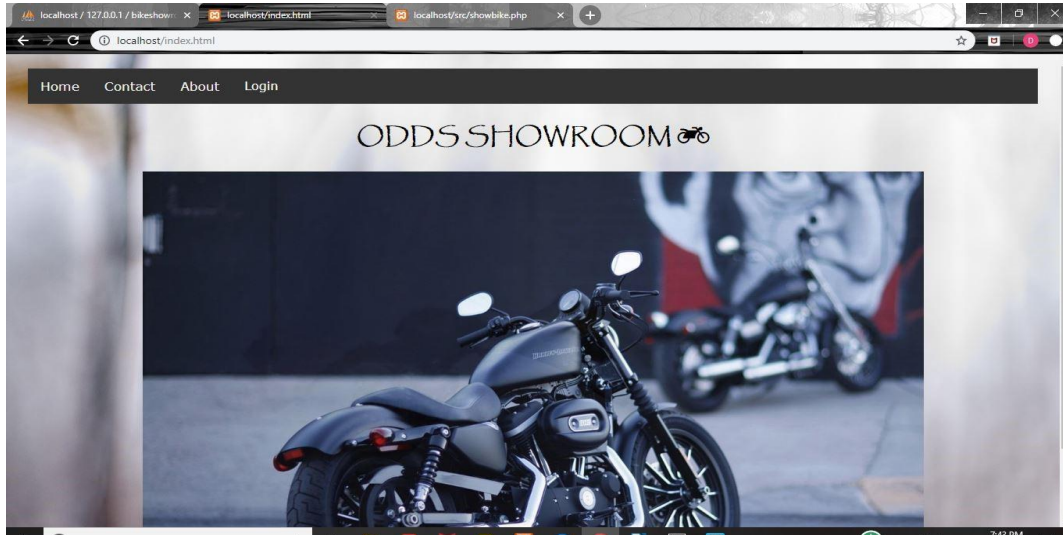


Fig 6.1 index.html

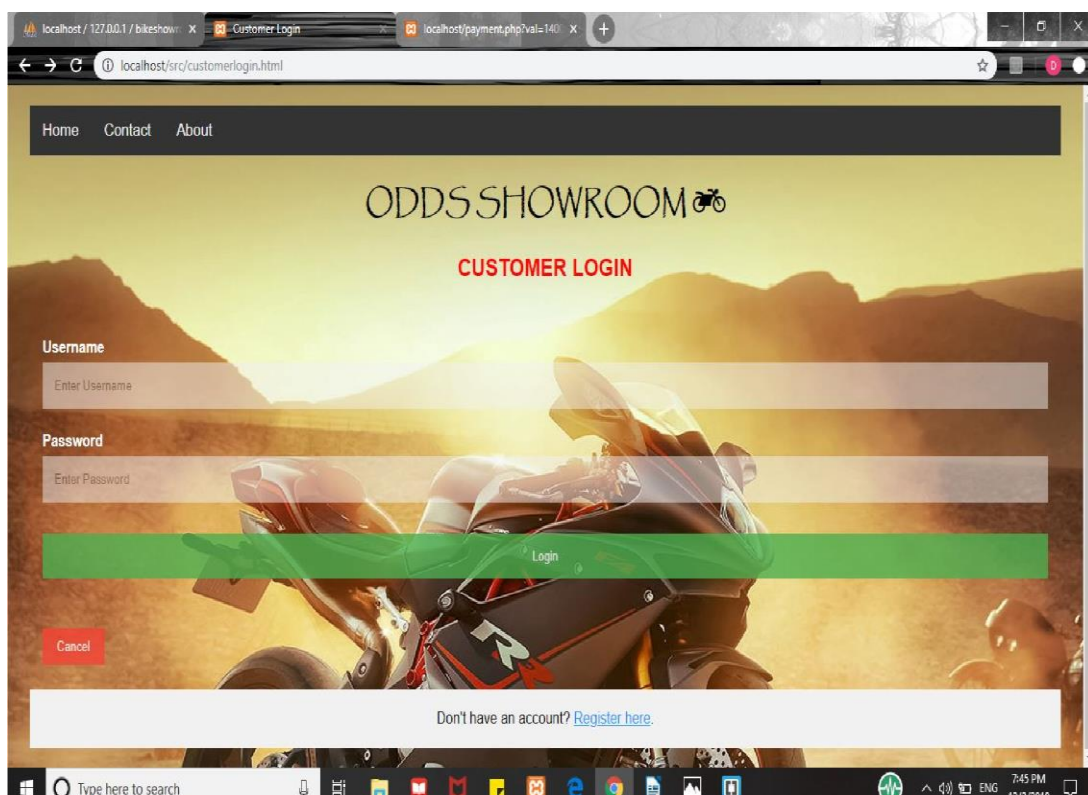


Fig 6.2 customerlogin.html

ONLINE BIKE SHOWROOM MANAGEMENT SYSTEM

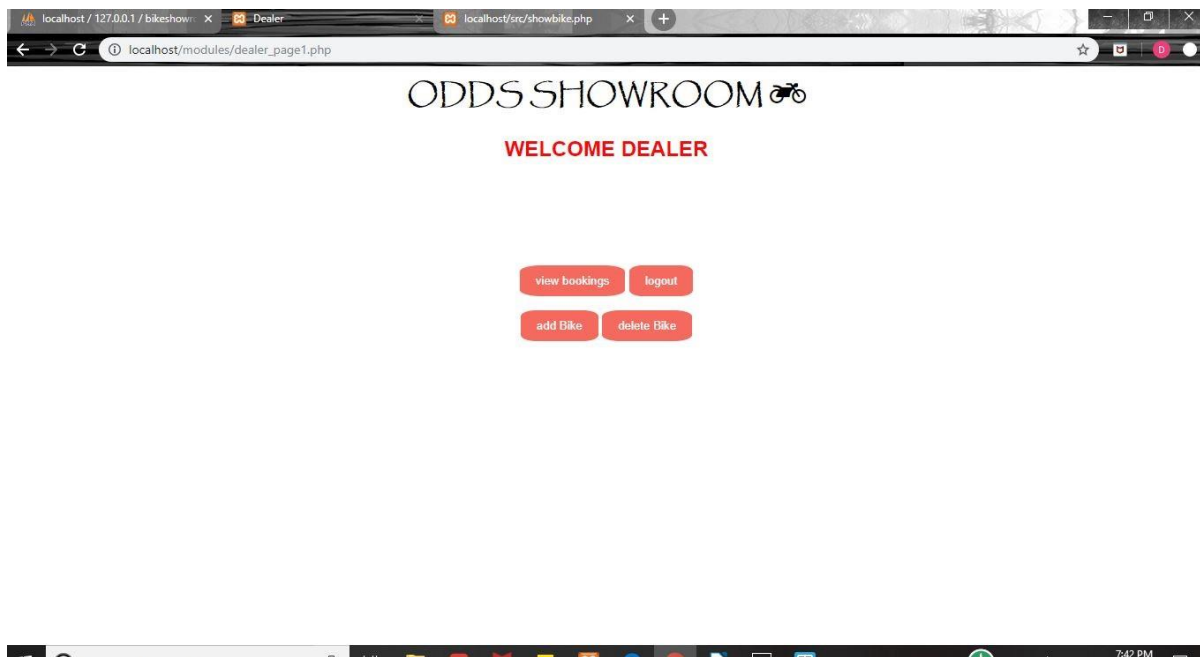


Fig 6.3 dealer_page1.html

A screenshot of a web browser showing the 'payment.php' page. The browser's address bar displays 'localhost/payment.php?val=140000&bid=1&nam=Suzuki%20Gixxer'. The page shows bike details: 'Bike name = Suzuki Gixxer', 'Price = 140000', and 'Bike id = 1'. Below this is a 'Billing Form' with two main sections: 'Billing Address' and 'Payment'. The 'Billing Address' section includes fields for 'Full Name' (Jack The Ripper), 'Email' (jacktheripper@example.com), 'Address' (Bhannerghatta main road, 18th Km), 'City' (Bengaluru), 'State' (Karnataka), and 'Zip' (10001). The 'Payment' section includes 'Accepted Cards' (with logos for Visa, Mastercard, and others), 'Name on Card' (Jack The Ripper), 'Credit card number' (1111-2222-3333-4444), 'Exp Month' (September), 'Exp Year' (2018), and 'CVV' (352).

Fig 6.4 payment.php

ONLINE BIKE SHOWROOM MANAGEMENT SYSTEM

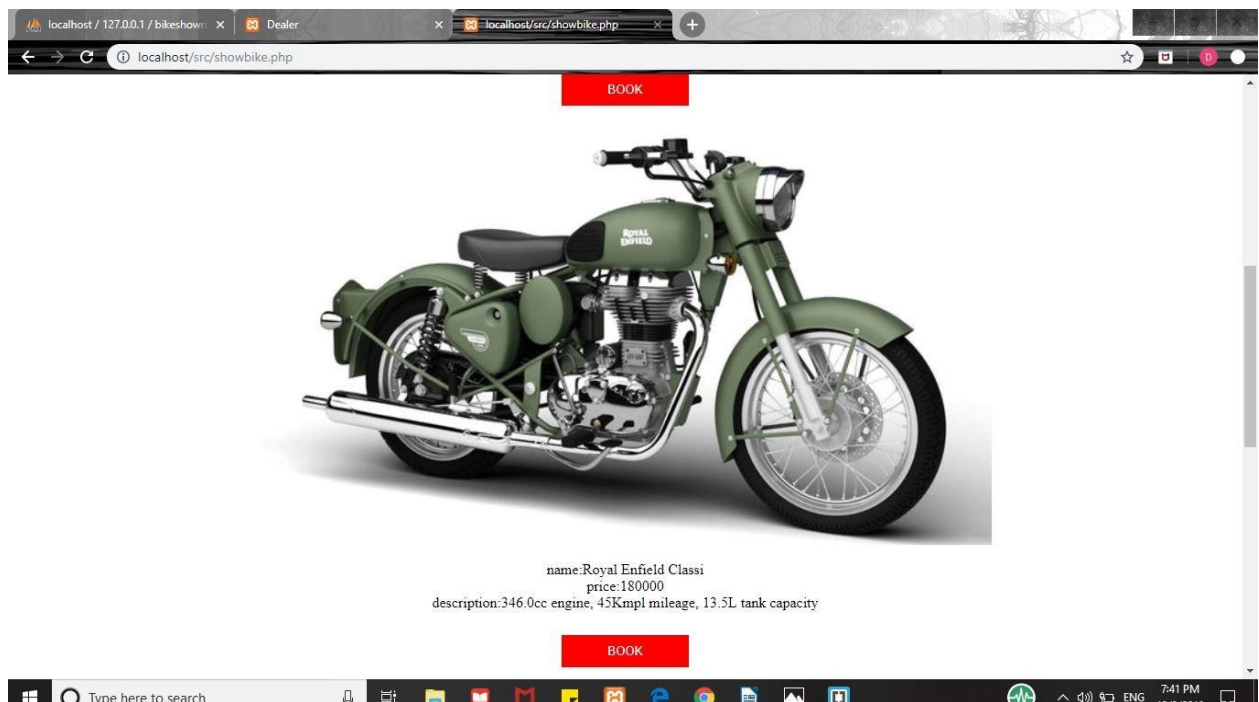


Fig 6.5 Customer page

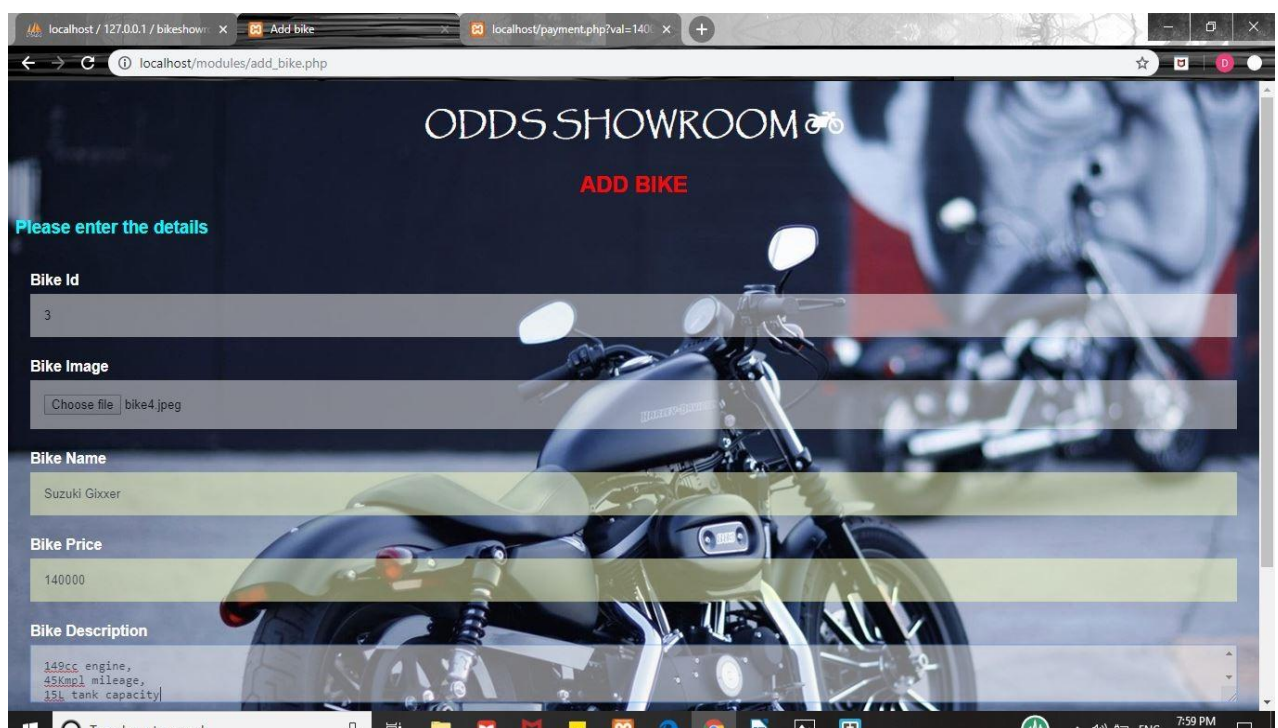


Fig 6.6 Dealer login

CONCLUSION AND FUTURE ENHANCEMENT

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of Phpmyadmin and to some extent Windows Application and SQL Server, but also about all handling procedure related with “VEHICLE SHOWROOM DATABASE SYSTEM”. It also provides knowledge about the latest technology used in developing stand alone application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently. This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity. The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date. Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover there is restriction for him that he cannot change the primary data field. This keeps the validity of the data to longer extent. User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him. From every part of the project the user is provided with the links through framing so that he can go from one option of the project to other as per the requirement. This is bound to be simple and very friendly as per the user is concerned. That is, we can say that the project is user friendly which is one of the primary concerns of any good project. Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database. Further it can be implemented for large size of customers, dealers and with the admin who will be having control over the functions performed, for all the automated purchase of a bike anywhere and anytime even with using a simple smart phone to access the website.

REFERENCES

- [1] Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
- [2] Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGrawHill.
- [3] Silberschatz Korth and Sudharshan, Database System Concepts, 6th Edition, McGrawHill, 2013. [4] Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementation and Management, Cengage Learning 2012.

WEB URL'S:

<https://www.w3schools.com/>

<https://www.tutorialspoint.com/>

<http://stackoverflow.com/>