VISTULA UNIVERSITY

Faculty of Computer Engineering

Program of study Computer Science

Manish Sarphare

Student’s number 49256

Implementation of Online Bike Rental System

The bachelor’s thesis
written under the supervision of
Dr. Hab Maciej Janowicz

Warsaw, 2023

Abstract

This thesis recommends the development of a multilingual online application for the rental of homes and rooms to overcome the language barrier. Users will be able to search for and reserve facilities in their preferred language thanks to the way the application is designed. A multi-level engineering strategy will be implemented to construct the desired application. This technique will involve the development of a user interface (UI), an application layer, and a database. Short-term rentals of houses and rooms are gaining ground in popularity all over the world as a direct because of an increase in individuals engaged in international travel and working remotely. Due of the internet, the way we search for, and reserve facilities has transformed, and as a consequence, it is now simpler and easier for travelers to choose the most suitable facility in the optimal location. Yet, language barriers can be a considerable hindrance for tourists who cannot converse in the local tongue of the nation they are visiting. This is particularly true in an international context. Using cutting-edge web development technologies like HTML, CSS, and JavaScript, the front-end user interface will be designed. It would provide a simple entry point for finding and booking facilities in several languages. The main application layer, which is responsible for the program's business logic, will be developed using Java. Customer validation, payment processing, and booking administration are just a few of the features that will be available. A social database management system will be used to develop the back-end database, which will contain information about people, facilities, and reservations. Because the database was designed to accommodate several languages, end users may search for facilities in their preferred language. To ensure its feasibility and use, the application in issue would adopt a client-centric design strategy during its development. The design cycle involves actions such as directing consumer research, creating customer "personas," and overseeing usability testing. The application will be tested by a group of users with varying levels of language proficiency to ensure that it is user-friendly and effective in overcoming language-based barriers. The development of this application will help to improvement in the accessibility and inclusiveness of the short-term rental business by supporting several forms of communication and facilitating travelers’ interactions with other communities. This will assist to enhance the entire sector.

KEYWORDS: Bike Rent, Rental System, Multi-lingual PHP, XAMPP, MYSQ.

Table of Contents

List of Figures

CHAPTER-1 INTRODUCTION

1.1 Project Summary

The goal of this initiative is to be utilized by a bike rental business that specializes in providing consumers with bicycle rentals. Customers may see available bikes, register, view profiles, and reserve bikes using this online system.

1.2 Objective

The rise of technology and the extensive use of the internet, especially in the bike rental industry, have greatly enhanced many business processes and channels of contact with customers. The system for renting electric bikes was created to offer the subsequent services.

Simplify administrative mechanisms that enable the rental company's usage of internet technologies to return on investment by using it to expand their market outside of their immediate location.

1.3 The Current and Proposed System

Current System

Users can reserve a bike through the Bike Rental System service for a set cost. Till far, there hasn't been a simple web interface to guide people through the car rental process. Via their offices, they had to physically rent the car. Managing rental cars was a challenging chore. It was difficult to keep track of all the rental motorcycles.

Suggested Structure

The customer might employ a vehicle thanks to this bike rental system proposal. Logging into the system, the user must check for bike availability. The user specifies the bike type as well as the day and hour of the trip. The Bike Rental System will determine whether the bike is available before renting it to the consumer. Online payments can be made by the user. PHP was used to design the tool. The MySQL database has all the information pertaining to rental bikes. The customer must enter his name, address, and phone number to look for bicycle rentals. The Interface is quite straightforward, and the back-end communication is strong. The key benefit is that the user will have options.

1.4 Possible Analysis

The project's viability and the possibility that the system will be helpful to the organization are both examined in the preliminary study. Precisely, reasonably feasible to insert new elements and bring about fixes to existing structures is the key point of the feasibility study. All procedure would function if time and resources were limitless. A probability study's objectives are to assess the viability of a project using an information system and to present feasible alternatives.

The preliminary investigation's feasibility analysis includes the following elements:

- The feasibility of Technology
- Operational Feasibility
- Economic Feasibility

1.5 Technical

This is the way it refers to whether the currently available software fully supports the application. It analyses the practicality of using a certain software component for production and its advantages and disadvantages. It also looks at any further training that software users may require. The organization's technical capability is then compared to its technical requirements. The solutions plan is viewed as technologically achievable if the organizational sophistication is adequate to meet the design demands. The analyst must determine if the already existing technological resource can be upgraded or supplemented to provide the necessary technical resources.

1.6 Operational

It speaks to whether the thing can really be used. Certain items may function flawlessly during design and execution but may perform poorly in a real-world setting. It also looks at the technical competence of any new human resources that may be needed. The human resource management available for the project determines how well a platform can be utilized after development and installation. It evaluates how well a proposed system resolves problems, seizes chances identified during scope management, and conforms with specifications uncovered even during analysis process of system development. It looks at the company's willingness to support the recommended system.

1.7 Economical

In comparison to the whole cost of funding the project, it relates to the advantages or results we obtain from the product. If the result is substantially like the preceding system, it is not practicable to develop it. Economic analysis is sometimes known as a cost-benefit analysis. It is the approach used the most commonly to assess a new system's efficacy. Economic analysis consists of calculating the expected advantages and incentives from a possible system and evaluating them against the costs. The decision to build and use the system is made if the benefits outweigh costs. A businessman must carefully weigh the benefits and risks before acting.

CHAPTER-2 TOOLS AND TECHNOLOGY

2.1 Overview

Users can communicate with computers via a graphical user interface (GUI) application by utilizing graphical symbols rather than text input. Using HTML5, CSS, and PHP, the suggested BIKE Renting system's GUI will be created.

HTML is the name of the rise dialect used to generate web pages and online applications (HTML). It supports a mechanism to generate coordinated texts through defining structure semantics for text components such headers, sentences, files, networks, references, and additional objects. Their sections are distinguished by identifiers, which also are represented in angle brackets. The building blocks of HTML are used by the BIKE Rental system to create UI components.

CSS a style sheet language, may be exploited to specify how a text composed in a markup language is displayed. The structure of the piece of paper is user-friendly thanks to CSS. Separating appearance from content, including components like fonts, colors, and layout, is the primary goal of CSS. BIKE Hire scheme function utilizes Bootstrap 4.1, a boilerplate made using CSS to hasten the building of GUI designs.

A server-side programming language called PHP (PHP Hypertext Processor) is used to dynamically construct websites. PHP code is adaptable and may be included among HTML code as well as utilized independently, with a wide range of online platforms, web-based editorial systems, and online template systems. An interpreter, which may either be a virtual server component or a Common Gateway Interface (CGI) application, is frequently used to read PHP code. The web server combines the newly created web page with the output of the PHP code that has been understood and accomplished, which could be any type of data, including graphics. Both independent graphical applications and command-line interfaces may be executed using PHP code (CLI). To interface with the MySQL database and show dynamic material on the webpage depending on user queries, the BIKE Renting system employs PHP.

2.2 Requirement for Hardware and Software

2.1.1 Hardware Specification

Processor	Pentium Dual Core or Above
RAM	4GB
Hard Disk	100 GB Hard disk or Above
Monitor	1.5 Inch Color Monitor
Keyboard	102/104 Key
Mouse	Optical Mouse

2.1.2 Software Specification

Operating System	Windows 7/8/10 or Linux
Browser	Chrome or Firefox
Front-end Backend	HTML/CSS/PHP

CHAPTER-3 SOFTWARE REQUIREMENTS & SPECIFICATION

3.1 Functional Specification

Usability: The interface should make use of words and ideas that are derived from the knowledge of those who will use the system the most.

Efficiency: The system must offer simple and quick access without incurring additional costs.

Reliability: The user shouldn't be taken aback by how the system behaves, and speech data may be sent and saved with ease (only way files)

3.2 Unimportant Requirement

It defines structure elements that communicate to how the system fulfils the essential functionalities. As follows:

- Protection: For the stored data, the subsystem must provide a high level of security and integrity by the system; only those who have been given permission by the firm are allowed access.
- Performance and Response Time: The system should execute user input with a great rate of presentation and must be able to respond or offer response promptly, generally in 20 to 25 seconds for simpler tasks and 50 seconds for more difficult tasks. Li, Y. (2019).
- Error management: Mistakes must be drastically reduced, and a proper error message must be provided to aid in the user's recovery.
- User input verification is important. Bai, X. (2014).
- Availability: This network must always be reachable around the clock, every day of the week. Moreover, any severe system issues should be resolved within one to two business days.

CHAPTER-4 ORGANIZATION LAYOUT

System design is the process of deciding on a system's structure, elements, connectors, and information to fulfil predetermined obligations. Structure shape considered the implementation systems approach to the production of products. There are several areas of overlap between the subjects of program management, companies’ architecture, and systems engineering. Jiang Z (2016).

4.1 Module

Bikes:

This is among the most significant modules. This module enables users to hire any bike from any bike vendor. So that no other consumer tries to reserve the same bike, the status of bikes may be updated as soon as they become available or booked. A bike is given information such as its model number, vehicle number, and owner name.

Login:

One can log into the system as the event manager or the customer after registering. When a user registers, they have the choice of enrolling a customer or an employee. This system's interface is dependent on registration. The user will have choices like organizing everything as requested by a customer for an event if they have registered as an employee. B. A. Weitz (2006).

The user interface gives them the ability to view all the event managers that are accessible and choose anybody they want to handle this event.

Admin:

Just one account has access to this module. That is, once an admin account has been created, no one else may register. The admin account has full access to look up information on any specific bike vendor or consumer. To ban any account, compute an employee's income after taking into account their leaves, calculate a payment, update the status of any event, modify a user's account, etc.

User:

Users can fall into one of two categories, and each kind will have a unique interface. Once a user has registered and logged in, they are then able to access the functions that are offered. A user who has registered as a vendor is able to edit information on the free bikes that are available for rental. If the user is a client, he may view all the motorcycles that are available for hire.

About us:

WE ARE THE BIKE RENTAL. The only 100% dedicated bike rental booking website. The first Bike Rental shop was our own bike shop. Ever Since it has been our aim to make bike rental easier for everyone, everywhere. We focus on making bike rentals easier for you. Your rental business has a unique set of challenges.

4.2. Context diagram

A data flow diagram visualizes the movement of data across an information system. It might display both incoming and outgoing data as well as the flow of data being stored. The DFD doesn't mention how data travels. The Flowchart and the system are very different from one another. The flowchart displays the control flow for the program modules. The various layers of data flow in the system are represented by DFDs. DFD lacks any branching or controlling components.

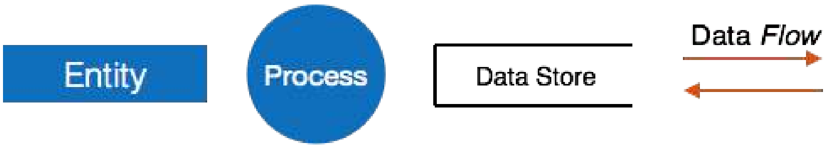
Different DFDs

Both logical and physical data flow diagrams are available. The internal processes and data flow of the system are the main topics of logical DFD. Consider the way information is transferred across several companies in a financial software system.

This specific DFD demonstrates how the data movement is really implemented in the organization. It is more precise and is situated closer to the execution.

DFD Components

Following group of components may be used to describe the source, destination, storage, and movement of data using DFD: -



Entities: Informational data's source and destination are both entities. The entities are represented as rectangles with their names in them.

Procedure - Turns or rectangles with rounded edges show movements and procedures carried out on the data.

Data Storage capacity - There are two types of data storage: an open-sided square with only one side lost, or a rectangle with both of its smaller sides absent.

Data Flow - Data movement is shown by bullets with points.

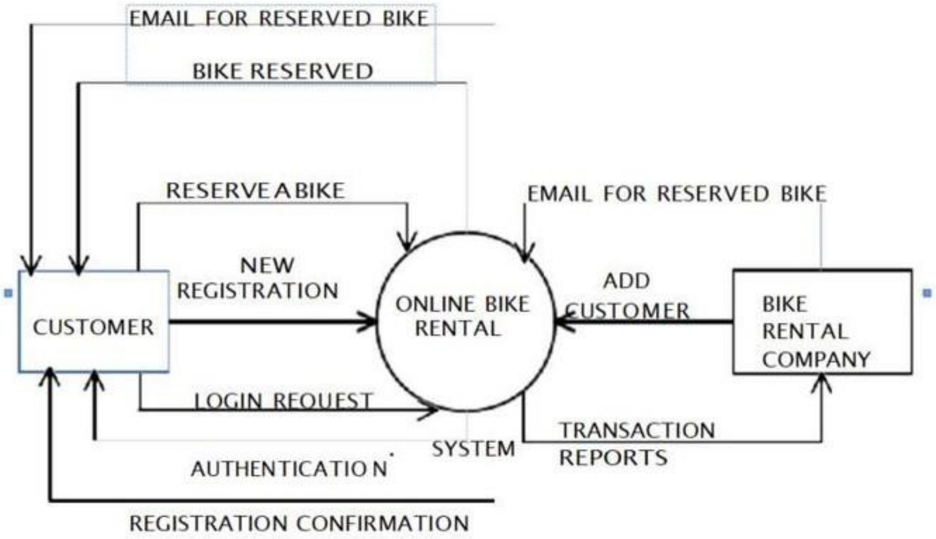
Data movement is displayed moving from the arrow's resource at the base to the arrow's destination at the head.

Levels of DFD

Level 0

- The whole information system is shown as a single image at Level 0 DFD, the greatest level of abstraction, while all the supporting details are hidden. Level 0 DFDs are often referred to as context level DFDs.

Figure 1: Level 0

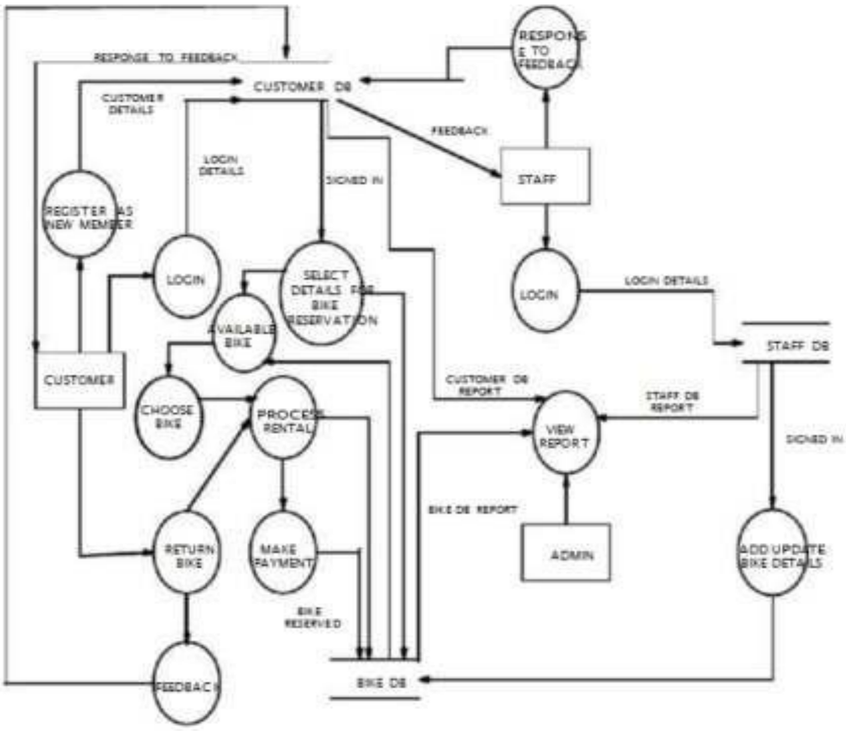


Level 1 - A Level 1 DFD that is more specific is generated from the Level 0 DFD. Level 1 DFD displays the basic system elements and data flow between various modules.

Besides that, Level 1 DFD contains essential processes and data sources.

Change

Figure 2: level 01 DFD



CHAPTER-5 DESIGN

5.1 Class Diagram

Static diagrams involve class diagrams. It exemplifies the application's static view. Class diagrams are helpful for generating software application source code including for visualizing, describing, and supporting a variety of structure parts. It depicts operations and attributes together with the limitations laid on the approach. As class diagrams Flowcharts can couple quickly with purpose-adapted languages, they are frequently used for modelling entity systems. A set of classes, linkages, associations, and boundaries depicted in diagram. Wang Huang too refers to this as a "structural diagram" (2016).

Function of group diagrams

Category diagrams are employed to represent the static view of an application. Class diagrams differ slightly from sequence diagrams in that they may depict any application's flow, not only its sequential flow. The most popular UML diagram among programmers is that one. The objective of the class diagram may well be summed up as the research and development of a software's static view. Explain a system's responsibilities. the place of origin for deployment and component diagrams. Choi, Y. J. (2017).

How Should a Class Graph Be Drawn?

UML diagrams used for generating software structures are class diagrams. Insight into class diagram creation method is essential. It includes a variety of design considerations, however in this instance the diagram will be seen top-down. Shrum, L. J. (2002).

Class diagrams provide a visual representation of both the static view of the system and other aspects connected to applications. The complete system is represented.

In creating a class diagram, keep the following in mind:

The class diagram's name needs to accurately define the system aspect it represents.

It is important to identify each component and how they relate in advance. Each class's responsibilities (attributes and methods) should be clearly defined, and only the essential properties should be shown because adding other properties can confuse the diagram. R. Kumar (2017).

While describing a specific feature of the diagram, take notes. The developer or coder should be able to grasp the drawing after it is finished.

The diagram should be made on plain paper and revised as many times as necessary to make it accurate before creating the final version.

An illustration of an application's order system is shown in the diagram below. It details a specific feature of the whole application. Initially, the two components of the system are named Order and Customer. Because a consumer may place many orders, there is a one-to-many link between them.

Special Order and Regular Order are two concrete classes under the Order class, which is an abstract class with an inheritance connection. All the attributes of the Order class are shared by the two inherited classes. They also do other duties including dispatch () and reception ().

Considering all the considerations, the following class diagram has been created.

5.2 Use Case Diagram

The key to modelling a system is to accurately represent its dynamic behavior. A system behaves dynamically while it is in use or active. Dynamic behavior is more crucial when modelling a system than static actions since static behavior alone is deficient. The use case diagram is one of the five provided in UML to represent the dynamic nature. Soesanto, H. (2019).

The usage sequence diagrams are dynamic because they have previously been constructed, certain internal or external components should be added to encourage interaction.

Actors are the terms used to describe these internal and external agents. Actors, use instances, and their relationships are displayed in use case diagrams. The application's system or subsystem is modelled using the diagram. Each use case illustration represents a certain system feature.

Use Case Illustrations' Goals

Application case diagrams are employed to show a system's dynamic nature. Although the fact that the objectives in the remaining diagrams (activity, sequence, collaboration, and State chart) are comparable, this definition is too broad to effectively explain them. We'll look at a unique feature that distinguishes one of the four diagrams. As a system is examined to gather its functionality, use cases are developed and actors are recognized. Use case models are modelled to show the exterior perspective once the initial work is completed. Use case diagrams are employed for the following reasons, in brief: used to gather a system's requirements. utilized to get a systems outside viewpoint. Establish the external and internal influencing forces on the system. Show the actors' interactions with the conditions.

A Use Case Diagram: How to Draw One

A program's high-level requirements are examined while taking use case diagrams into account. Following requirements analysis, use cases are used to depict the functions of systems.

According to this concept, use cases represent the basic system features that have been gathered and documented. The second component that is important to utilize cases is the actors. Everything that communicates with one another in some manner is an actor. S. Montazeri (2020).

The functionality connections between the actors and the use cases will be represented by use cases. Use case diagrams are made to show the functional requirements of a system. Following the identification of the items, we must follow the following guidelines to produce a useful use case diagram.

The name of a use case is very important. It is important to have a name that makes it obvious what tasks are being performed. Wang, D. (2018).

Name your actors something appropriate. Show dependencies and relationships in the diagram with clarity. Avoid attempting to add every relationship that might exist because the diagram's main objective is to explain the needs. Where required, make notes to better explain any important points.. Consequently, the graph includes one actor, the consumer, and three use scenarios (Order, Special Sequence, and Normal Order). The Special Ordering and Normal Order use cases are added to the Order use case. As a result, they have been together for a long time. Another critical step is to locate the system border, which can be seen in the image. Since the customer is an outsider user of the system.

Can a Use Case Diagram Be Used?

There are five diagrams in UML that depict the dynamic perspective of a system, as we've already discussed. Now, every model has a certain application for which it is intended. These goals are several perspectives of an active system.

We need to employ a variety of diagram kinds to comprehend a system's dynamics. One of these, the use case diagram, has the unique objective of gathering the actors and system needs.

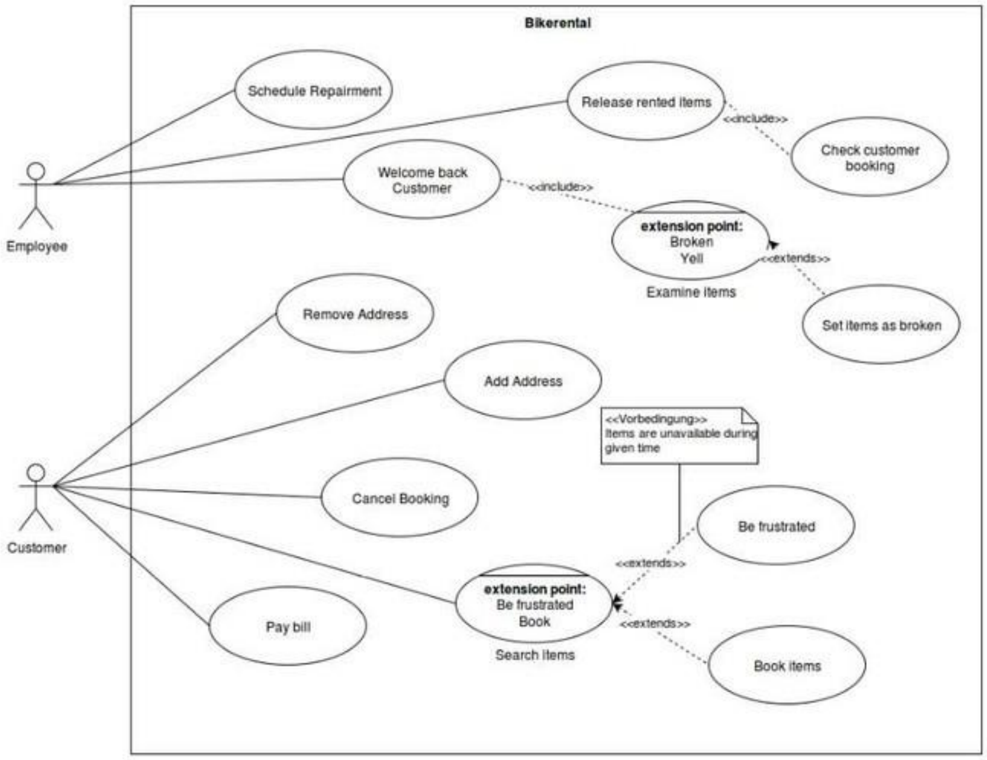
Use case diagrams detail a system's events and their flows. But use case diagrams never specify how they be put into practice. Use case diagrams can be compared to black boxes, where the only things that are known about them are their input, output, and function. Li, H. (2018).

These diagrams are employed at a very advanced design level. To obtain a thorough and useful understanding of the system, this high-level design is continually improved. A well-structured use case also includes descriptions of the prerequisites, outcomes, and exclusions. While creating test cases for the testing, these additional components are used.

Even if a use case is a poor choice for forward and reverse engineering, these techniques are nonetheless applied, but in a somewhat different way. H. Zhang (2019) Reverse engineering operates in a similar manner. To make a use case diagram acceptable for reverse engineering, it is employed in a different way.

Use case diagrams are used in forward engineering to create test cases, and in reverse engineering to produce the need information from the current application. Use case diagrams can be used for high-level design and requirement analysis. Model a system's environment. engineering in reverse. advancing engineering.

Figure 3: Use case Diagram

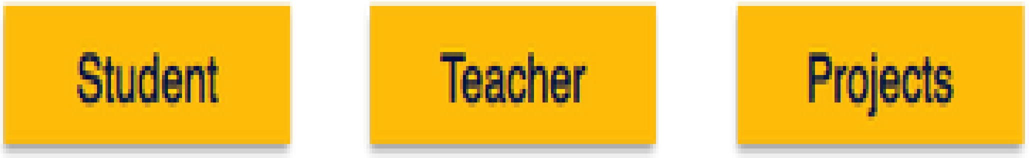


5.3 Diagram of an entity relationship

An ER diagram is used to depict the ER Model. Every object, including objects, characteristics of organizations, relationship sets, and characteristics of relationship sets, may be represented using an ER diagram.

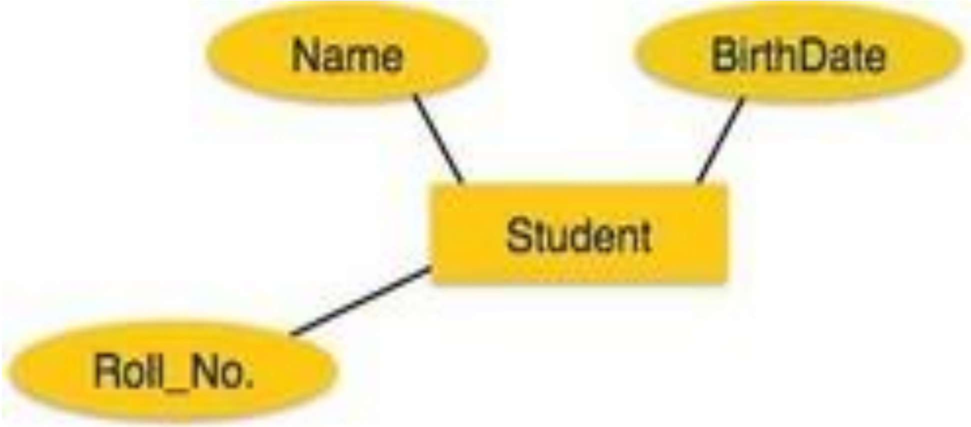
Entity:

Rectangles are used to represent entities. When naming rectangles, consider the entity set they stand for.



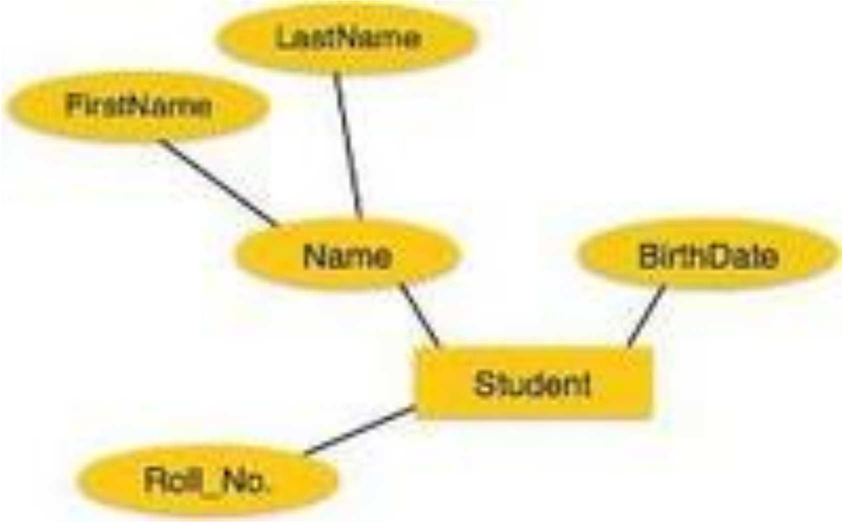
Attributes:

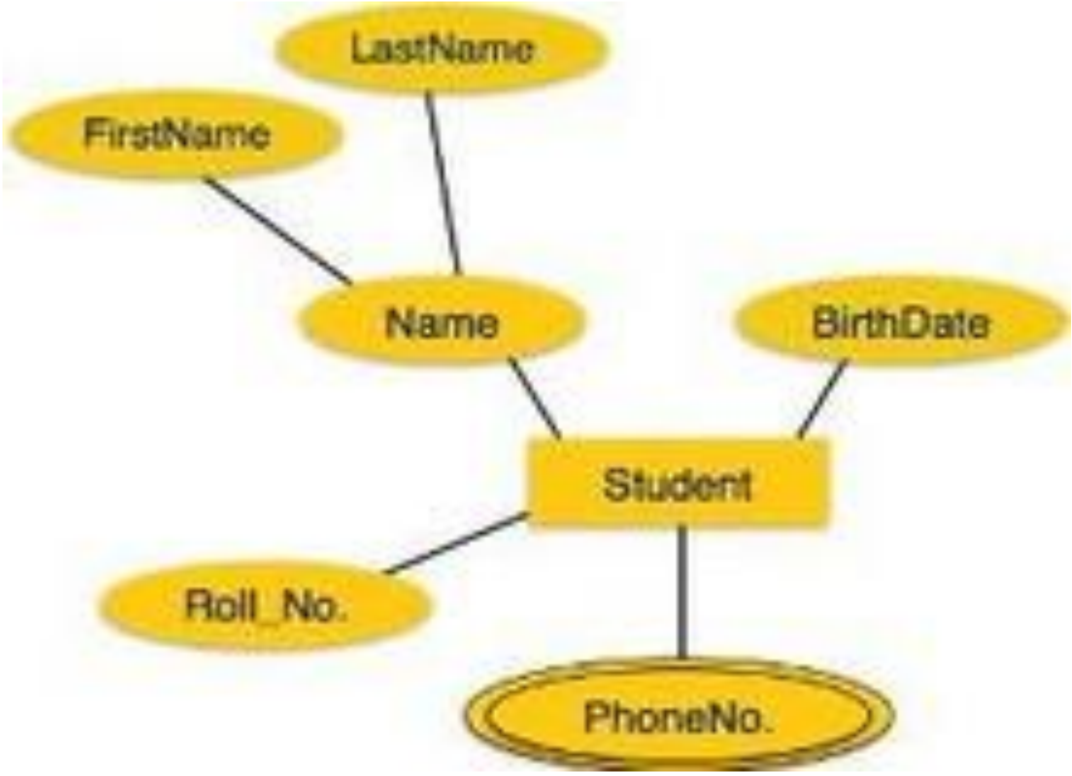
An entity's properties are its characteristics. Attribute symbols employ ellipse shapes. Individual ellipse exemplifies a certain characteristic and is related with its entity (rectangle).



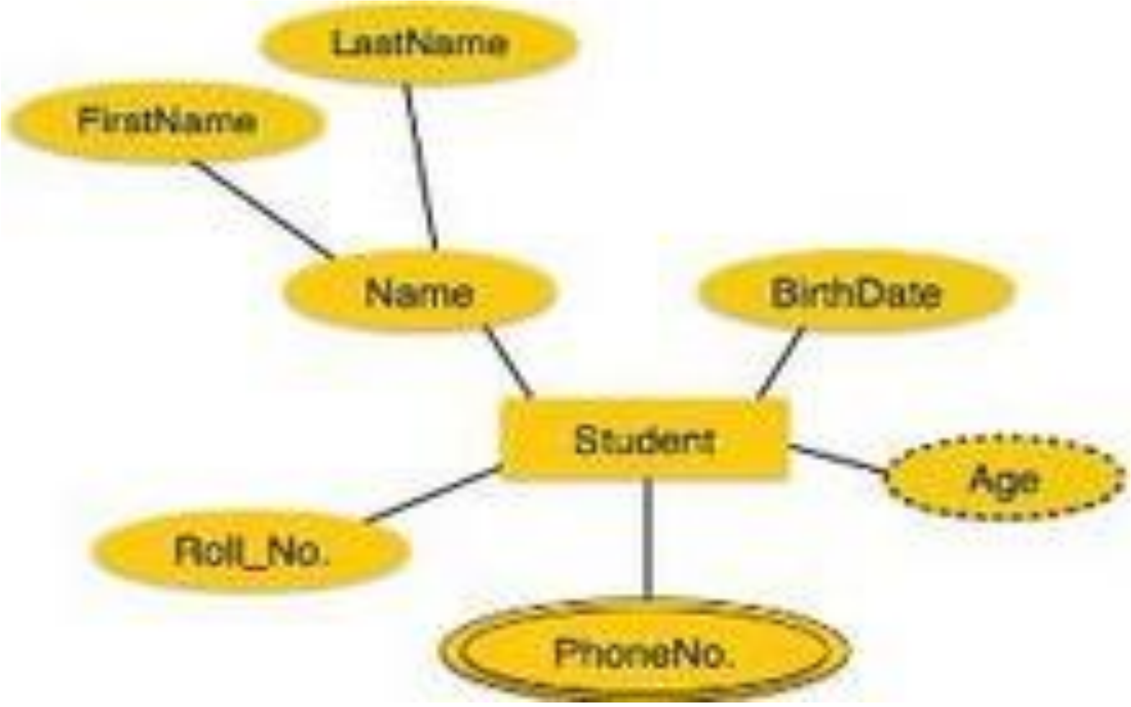
If the characters are compound, they are further in a manner like a tree. The appropriate attribute is subsequently linked to each node. Ellipses are used to join other ellipses to represent composite features.

Double ellipses are used to correspond to multivalued attributes





Dashed ellipses are used to represent derived characteristics.



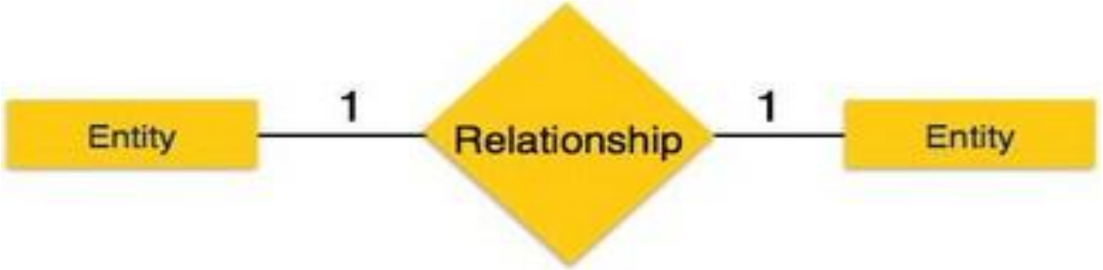
Affiliation

Associations are exhibited using baseball diamond-shaped boxes. Inside the diamond box is a textual reference to the link's name. Each of the entities (rectangles) participated in a association are connected by a line.

Bikedinality and Binary program Relations

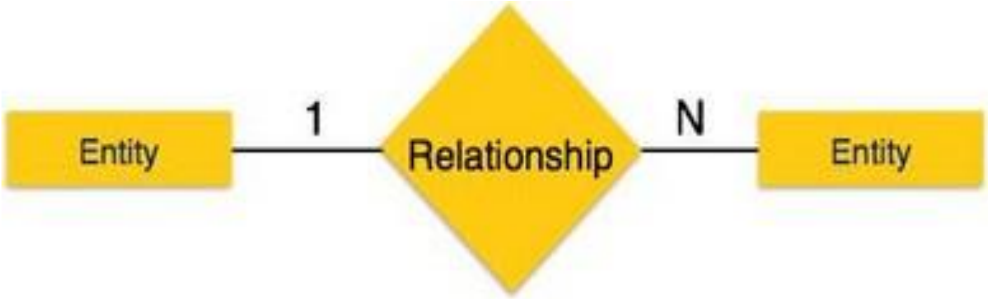
Binary relationships involve two participants and are defined as such. The number of occurrences of an entity from a connection that could be connected to the relation is known as the bickedinality.

One-to-one: When this connection only has one instance of an entity attached to it, it is referred to as a "1:1" relationship. The example that follows shows how each entity should only have one instance attached to the connection. It reveals a close relationship.

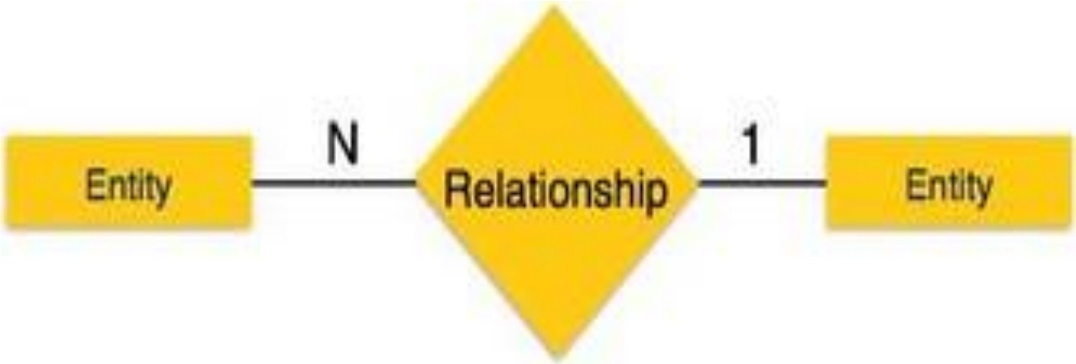


One is too many It is shown as "1: N" when a connection has several instances of the same object. A relationship can link many instances of an entity on the right and just one instance of an entity on the left, as shown in the accompanying figure. One-to-many connections are offered.

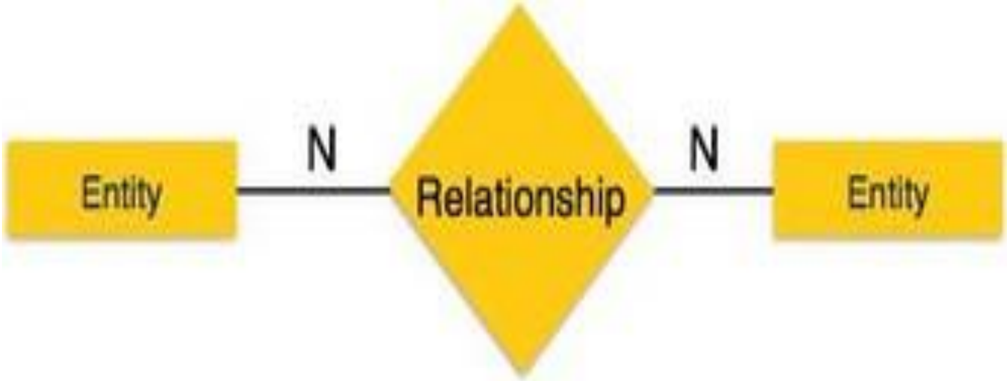
Many-to-one –



When there are many instances of an entity linked to a connection, it is noted as "N:1". The right-hand figure demonstrates that while there may be several instances of an entity on the left, there may only be one instance of that entity on the right that is tied to the connection. It shows the many-to-one relationship.

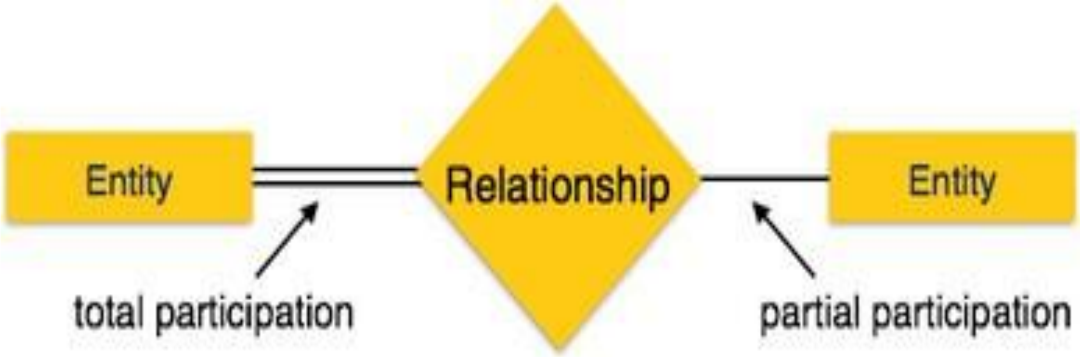


From many to many The accompanying diagram suggests that there may be more than one instance of each thing on the left and right that are connected by the connection. It shows the many-to-many relationship.

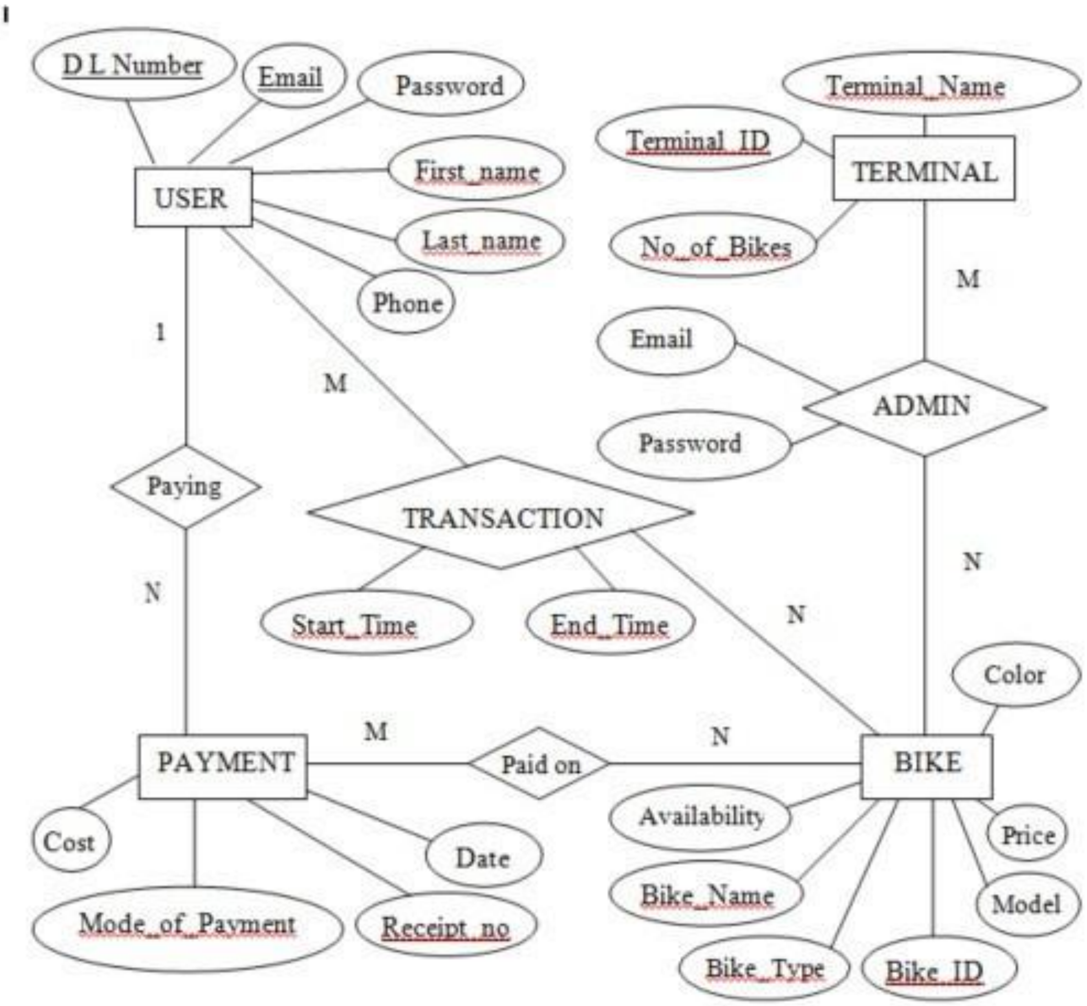


Participation Constraints

Complete Participation: Every party to the connection is involved. Double lines signify the full involvement. Partial involvement Not all parties to the connection are involved. Individual lines are used to denote partial involvement.



5.3.1 ER-Diagram



5.4 Database Design

ADMIN

CREATE TABLE ADMIN (EMAIL VARCHAR2(20), PASSWORD VARCHAR2(15));

USER

CREATE TABLE USER (FNAME VARCHAR2(15), LNAME VARCHAR2(15),

EMAIL VARCHAR2(20) PRIMARY KEY, PASSWORD VARCHAR2(15), PHONE BIGINT (12),

DLNO VARCHAR2(15) UNIQUE);

TERMINAL

```
CREATE TABLE TERMINAL (TERM_ID INT (5) PRIMARY KEY, TERM_NAME
VARCHAR2(15), NO_OF_BIKES INT (2));
```

BIKE

```
CREATE TABLE BIKE (BIKE_ID INT (5) PRIMARY KEY, BIKE_NAME
VARCHAR2(10), MODEL YEAR,

COLOR VARCHAR2(10),    BIKE_TYPE   VARCHAR2(8),        PRICE INT (4),


TERM_IDREFERENCES

TERMINAL(TERM_ID)

ON

DELETE CASCADE,

AVAIL INT (1));
```

TRANSACTION

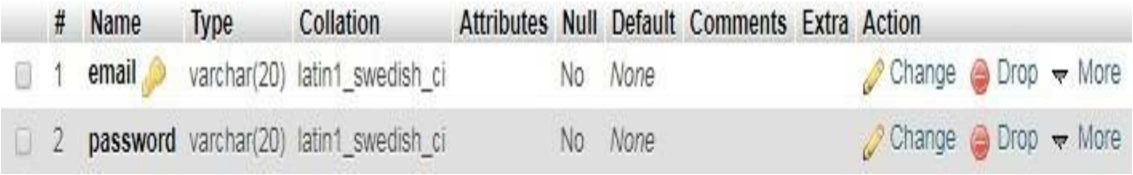
```
CREATE TABLE TRANSACTION (EMAIL REFERENCES USER(EMAIL) ON
DELETE CASCADE,

BIKE_IDREFERENCES BIKE(BIKE_ID) ON DELETE CASCADE,

START_TIME DATETIME,

END_TIME DATETIME);
```

5.5 Details of Tables

ADMIN

DESC ADMIN;

Figure 4: Admin

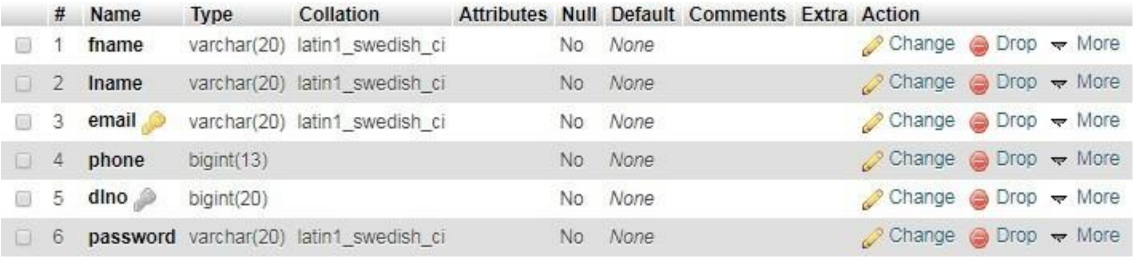


	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	email 🔑	varchar(20)	latin1_swedish_ci		No	*None*			Change Drop More
□	2	password	varchar(20)	latin1_swedish_ci		No	*None*			Change Drop More

USER

DESC USER;

Figure 5: USER

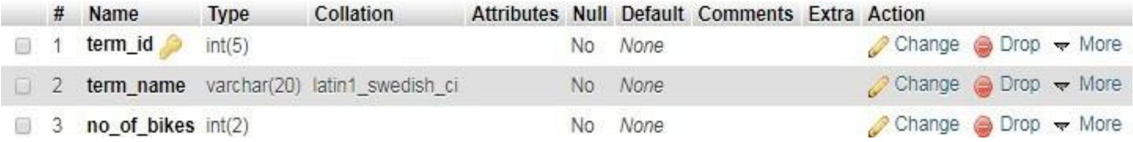


	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	fname	varchar(20)	latin1_swedish_ci		No	*None*			Change Drop More
□	2	lname	varchar(20)	latin1_swedish_ci		No	*None*			Change Drop More
□	3	email 🔑	varchar(20)	latin1_swedish_ci		No	*None*			Change Drop More
□	4	phone	bigint(13)			No	*None*			Change Drop More
□	5	dlno 🔑	bigint(20)			No	*None*			Change Drop More
□	6	password	varchar(20)	latin1_swedish_ci		No	*None*			Change Drop More

TERMINAL

DESC TERMINAL:

Figure 6: TERMINAL



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	term_id 🔑	int(5)			No	*None*			Change Drop More
□	2	term_name	varchar(20)	latin1_swedish_ci		No	*None*			Change Drop More
□	3	no_of_bikes	int(2)			No	*None*			Change Drop More

5.6 SQL Triggers and Stored Procedure

5.6.1 Triggers

Triggers are stored programs, which are automatically executed or fired when some event occur. Triggers are, in fact, written to be executed in response to any of the following events:

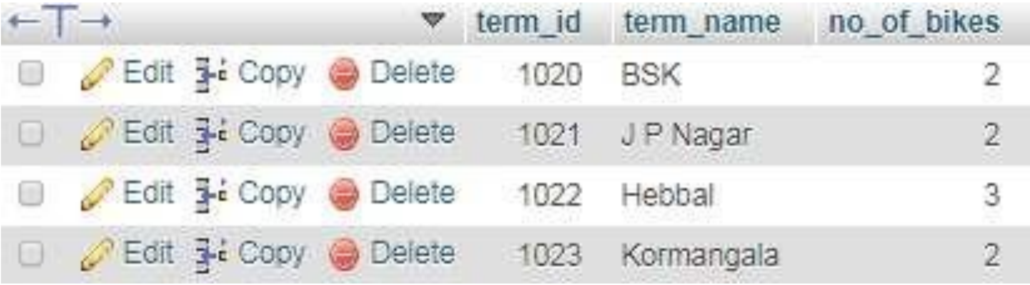
- A Database manipulation (DML) statement (DELETE, INSERT, or UPDATE)
- A Database definition (DDL) statement (CREATE, ALTER, or DROP)
- A Database operation (SERVERERROR,LOGON,LOGOFF,STARTUP,SHUTDOWN)

Triggers can be defined on the table, view, schema, or database with which the event is associated. The trigger used in this application is used to increment the value of no_of_BIKEs in Terminal when BIKE is added to that terminal. Another Trigger is used to decrement the no_of_BIKEs when a BIKE in that terminal is deleted. By knowing the value of no_of _BIKEs, it is easier to get count of BIKEs in particular terminal.

The Trigger is

```
CREATE TRIGGER `BIKEadd` AFTER INSERT ON `BIKE` FOR EACH ROW

UPDATE terminal

set no_of_BIKEs = no_of_BIKEs + 1 WHERE term_id = new.term_id; CREATE
TRIGGER `BIKErem` AFTER DELETE ON `BIKE` FOR EACH ROW

UPDATE terminal

set no_of_BIKEs = no_of_BIKEs - 1 WHERE term_id = old.term_id;
```

Figure 7: Triggers



term_id	term_name	no_of_bikes
1020	BSK	2
1021	J P Nagar	2
1022	Hebbal	3
1023	Kormangala	2

5.6.2 Stored Method

A ready-made SQL script that is usable and can be stored repeatedly is known as a stored procedure. As a result, if a query needs to be performed again, it may be kept as a stored procedure and run by calling it rather than having to create the query each time. Moreover, the saved process can accept arguments. The saved process can therefore respond appropriately based on the situation.

Listed below are some situations when stored processes are helpful:

• If many applications require a database program, Any application software may call it by keeping it on the server. By doing so, effort duplication is eliminated, and software modularity is improved.

• In certain circumstances, running software on the server can lower the cost of communication and data exchange between the server and the client. Du, Q. (2018).

• They can also be used to detect intricate limitations that assertions and triggers are unable to specify. To calculate the cost, the stored method utilised in this application takes two parameters. Two arguments must be given with a call to this stored procedure in order to use it.

Stored Procedure is:

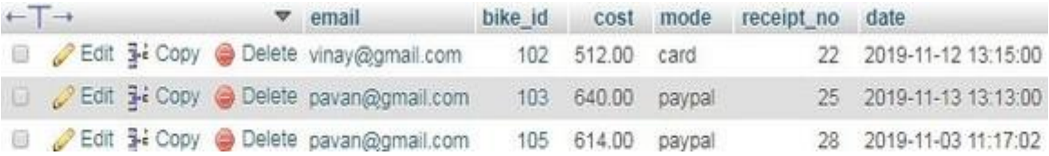
```
DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `spcost` (IN `hour` INT(3), IN
`id` INT (5)) BEGIN
DECLARE price int;

DECLARE BIKEid CURSOR FOR SELECT price FROM BIKE WHERE
BIKE_id=id; OPEN BIKEid;
```

FETCH FROM BIKEid INTO price; UPDATE payment SET cost = (hour * price1)

WHERE BIKE_id =id and date is NULL; CLOSE BIKEid; END$$ DELIMITER;

Show Procedure:

Figure 8: Stored Procedure



←T→			▼	email	bike_id	cost	mode	receipt_no	date
□	Edit	Copy	Delete	vinay@gmail.com	102	512.00	card	22	2019-11-12 13:15:00
□	Edit	Copy	Delete	pavan@gmail.com	103	640.00	paypal	25	2019-11-13 13:13:00
□	Edit	Copy	Delete	pavan@gmail.com	105	614.00	paypal	28	2019-11-03 11:17:02

5.7 Record Connectivity

Adding a few PHP instructions makes it simple to link the front end to the backend end/database (i.e., MySql). The directions below must be included.

```
<?php

//initializing variables

// function OpenCon() {

$servername = "localhost";

$username = "root";

$password = "";

$db ="BIKErental";

//connect to server

$conn = mysqli_connect ($servername , $username , $password,$db) or die("unable to connect to host");

    //return $conn;

     //}

     ?>
```

CHAPTER-6 IMPLEMENTATION

6.1 PSEUDO

CODE Index page Code

```
<?php

session_start();

include('includes/config.php');

error_reporting(0);

?>

<!DOCTYPE HTML>

<html lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width,initial-scale=1">

<meta name="keywords" content="">

<meta name="description" content="">

<title>Bike Rental Portal</title>

<! --Bootstrap -->

<link rel="stylesheet" href="assets/css/bootstrap.min.css" type="text/css"> <link
rel="stylesheet" href="assets/css/style.css" type="text/css">

<link rel="stylesheet" href="assets/css/owl.bikeousel.css" type="text/css"> <link
rel="stylesheet" href="assets/css/owl.transitions.css" type="text/css"> <link
href="assets/css/slick.css" rel="stylesheet">

<link href="assets/css/bootstrap-slider.min.css" rel="stylesheet">

<link href="assets/css/font-awesome.min.css" rel="stylesheet">
```

```
<linkrel="stylesheetid="switcher-css"type="text/css"
href="assets/switcher/css/switcher.css" media="all" />

<link rel="alternate stylesheet" type="text/css" href="assets/switcher/css/red.css"
title="red" media="all" data-default- color="true" />

<linkrel="alternatestylesheet" type="text/css" href="assets/switcher/css/orange.css"
title="orange" media="all" />

<link rel="alternate stylesheet" type="text/css" href="assets/switcher/css/blue.css"
title="blue" media="all" />

<link rel="alternate stylesheet" type="text/css" href="assets/switcher/css/pink.css"
title="pink" media="all" />

<link rel="alternate stylesheet" type="text/css" href="assets/switcher/css/green.css"
title="green" media="all" />

<link rel="alternate stylesheet" type="text/css" href="assets/switcher/css/purple.css"
title="purple" media="all" />

<link rel="apple-touch-icon-precomposed" sizes="144x144"
href="assets/images/favicon-icon/apple-touch-icon-144-precomposed.png">

<link rel="apple-touch-icon-precomposed" sizes="114x114"
href="assets/images/favicon-icon/apple-touch-icon-114-precomposed.html">

<link rel="apple-touch-icon-precomposed" sizes="72x72" href="assets/images/favicon-
icon/apple-touch-icon-72-precomposed.png">

<link rel="apple-touch-icon-precomposed" href="assets/images/favicon-icon/apple-
touch-icon-57-precomposed.png">

<link rel="shortcut icon" href="assets/images/favicon-icon/favicon.png">


<link href="https://fonts.googleapis.com/css?family=Lato:300,400,700,900"
rel="stylesheet">

</head>

<body>
```

```
   <! -- Start Switcher -->

<?php include('includes/colorswitcher.php');?>
<! -- /Switcher -->
<! --Header-->
<?php include('includes/header.php'); ?>
<! -- /Header -->
<! -- Banners -->
<section id="banner" class="banner-section">
<div class="container">
<div class="div_zindex">
<div class="row">
<div class="col-md-5 col-md-push-7">
<div class="banner_content">
<h1>Find the right bike for you. </h1>
<p>We have more than a thousand bikes for you to choose. </p>
<a href="#" class="btn">Read More <span class="angle_arrow"><i class="fa fa-angle-
right" aria-hidden="true"></i></span></a> </div>
</div>
</div>
</div>
</div>
</section>
<!-- /Banners >
<!-- Resent Cat-->
```

```
<section class="section-padding gray-bg">
<div class="container">
<div class="section-header text-center">
<h2>Find the Best <span>BikeForYou</span></h2>
<p>There are numerous different versions of portions from Lorem Ipsum that are available, but the most have been changed in some way, usually by adding humour or randomising phrases that don't appear the slightest bit plausible. You should check a passage from Lorem Ipsum before using it to make sure there isn't whatever humiliating tucked away in the inside of it.</p>
</div>
<div class="row">
<!-- Nav tabs -->
<div class="recent-tab">
<ul class="nav nav-tabs" role="tablist">
<li role="presentation" class="active"><a href="#resentnewbike" role="tab" data-toggle="tab">New Bike</a></li></ul>
</div>
<!-- Recently Listed New Bikes -->
<div class="tab-content">
<div role="tabpanel" class="tab-pane active" id="resentnewbike">
<?php$sql="SELECT tblvehicles.VehiclesTitle,tblbrands.BrandName,tblvehicles.PricePerDay,tblvehicles.FuelType,tblvehicles.ModelYear,tblvehicles.id,tblvehicles.SeatingCapacity,tblvehicles.
VehiclesOverview,tblvehicles.Vimage1 from tblvehicles join tblbrands    on tblbrands.id=tblvehicles.VehiclesBrand";
$query = $dbh -> prepare($sql);
$query->execute();
```

```
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=1;
if($query->rowCount() > 0)
{foreach ($results as $result) {
?>
<div class="col-list-3">
<div class="recent-bike-list">
<div class="bike-info-box"> <a href="vehical-details.php?vhid=<?php echo
htmlentities($result->id);?>"><img src="admin/img/vehicleimages/<?php echo
htmlentities($result->Vimage1);?>" class="img-responsive" alt="image"></a>
<ul>
<li><i class="fa fa-bike" aria-hidden="true"></i><?php echo htmlentities
($result->FuelType);?></li>
<li><i class="fa fa-calendar" aria-hidden="true"></i><?php echo htmlentities($result-
>ModelYear);?> Model</li>
<li><i class="fa fa-user" aria-hidden="true"></i><?php echo htmlentities($result-
>SeatingCapacity);?> seats</li>
</ul>
</div>
<div class="bike-title-m">
<h6><a href="vehical-details.php?vhid=<?php echo htmlentities($result->id);?>">
<?php echo htmlentities($result->BrandName);?> , <?php echo htmlentities($result-
>VehiclesTitle);?></a></h6>
<span class="price">$<?php echo htmlentities($result->PricePerDay);?>/Day</span>
</div>
<div class="inventory_info_m">
<p><?php echo substr($result->VehiclesOverview,0,70);?></p>

</div>

</div>

</div>

<?php }}?>

</div>

</div>

</div>

</section>

<!-- /Resent Cat -->

<!-- Fun Facts-->

<section class="fun-facts-section">

<div class="container div_zindex">

<div class="row">

<div class="col-lg-3 col-xs-6 col-sm-3">

<div class="fun-facts-m">

<div class="cell">

 <h2><I class="fa fa-calendar" aria-hidden="true"></i>40+</h2>

<p>Years in Business</p>

</div>

</div>

</div>
```

```
<div class="col-lg-3 col-xs-6 col-sm-3">

<div class="fun-facts-m">

<div class="cell">

<h2><i class="fa fa-bike" aria-hidden="true"></i>1200+</h2>

<p>New Bikes For Sale</p>

</div>

</div>

</div>

<div class="col-lg-3 col-xs-6 col-sm-3">

<div class="fun-facts-m">

<div class="cell">

<h2><i class="fa fa-bike" aria-hidden="true"></i>1000+</h2>

<p>Used Bikes For Sale</p>

</div>

</div>

</div>

<div class="col-lg-3 col-xs-6 col-sm-3">

<div class="fun-facts-m">

<div class="cell">

<h2><i class="fa fa-user-circle-o" aria-hidden="true"></i>600+</h2>

<p>Satisfied Customers</p>

</div>

</div>

</div>
```

```
</div>

</div>

<!-- Dark Overlay□

<div class="dark-overlay"></div>

</section>

<!-- /Fun Facts-->

<!--Testimonial -->

<section class="section-padding testimonial-section parallex-bg">

<div class="container div_zindex">

<div class="section-header white-text text-center">

<h2>Our Satisfied <span>Customers</span></h2>

</div>

<div class="row">

<div id="testimonial-slider">

<?php

$tid=1;

$sql = "SELECT tbltestimonial.Testimonial,tblusers.FullName from tbltestimonial join tblusers on tbltestimonial.UserEmail=tblusers.EmailId where tbltestimonial.status=:tid";

$query = $dbh -> prepare($sql);

$query->bindParam(':tid',$tid, PDO::PARAM_STR);

$query->execute();

$results=$query->fetchAll(PDO::FETCH_OBJ);

$cnt=1;

if($query->rowCount() > 0)
```

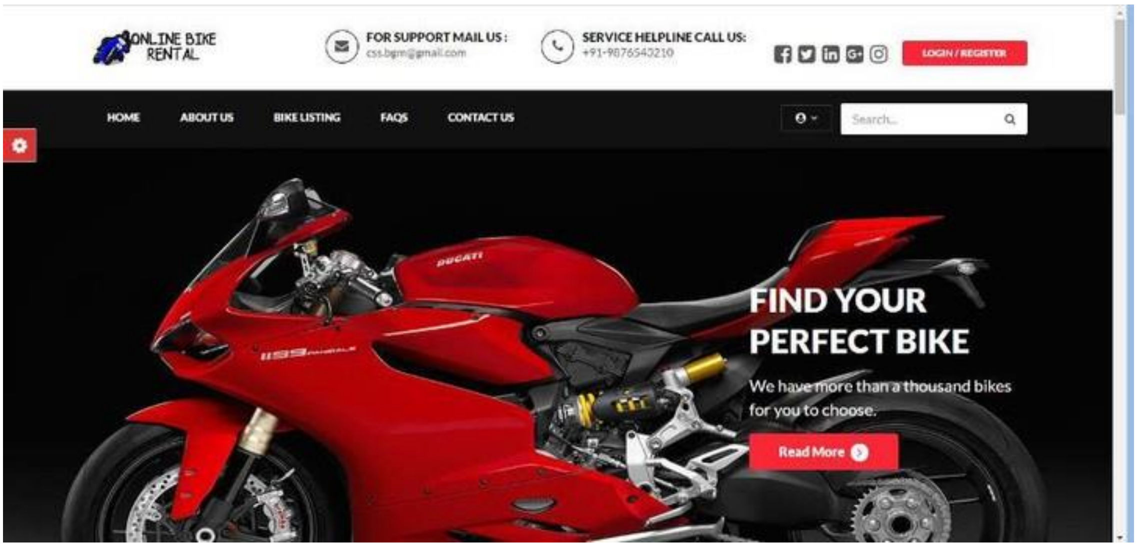
```
{

foreach($results as $result)

{ ?>

<div class="testimonial-m">

<div class="testimonial-img"> <img src="assets/images/cat-profile.png" alt="" />
</div>

<div class="testimonial-content">

<div class="testimonial-heading">

<h5><?php echo htmlentities($result->FullName);?></h5>

<p><?php echo htmlentities($result->Testimonial);?></p>

</div>

</div>

</div>

<?php }} ?>

</div>

</div>

</div>

<!-- Dark Overlay-->

<div class="dark-overlay"></div>

</section>

<!-- /Testimonial-->

<!--Footer -->

<?php include('includes/footer.php');?>

<!-- /Footer-->

<!--Back to top-->
```

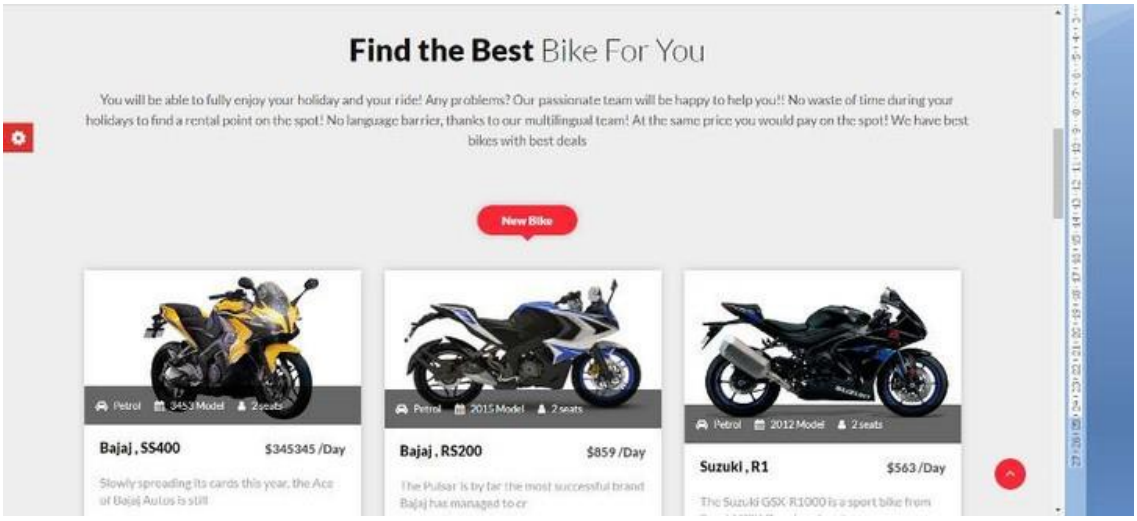
```
<div id="back-top" class="back-top"> <a href="#top"><i class="fa fa-angle-up" aria-hidden="true"></i> </a> </div>

<!--/Back to top-->

<!--Login-Form -->

<?php include('includes/login.php');?>

<!--/Login-Form -->

<!--Register-Form -->

<?php include('includes/registration.php');?>

<!--/Register-Form -->

<!--Forgot-password-Form -->

<?php include('includes/forgotpassword.php');?>

<!--/Forgot-password-Form -->

<!-- Scripts -->

<script src="assets/js/jquery.min.js"></script>

<script src="assets/js/bootstrap.min.js"></script>

<script src="assets/js/interface.js"></script>

<!--Switcher-->

<script src="assets/switcher/js/switcher.js"></script>

<!--bootstrap-slider-JS-->

<script src="assets/js/bootstrap-slider.min.js"></script>

<!--Slider-JS-->

<script src="assets/js/slick.min.js"></script>

<scriptsrc="assets/js/owl.bikeousel.min.js"></script>

</body>
```

```
<!-- Mirrored from themes.webmasterdriver.net/bikeforyou/demo/index.html by
HTTrack Website Copier/3.x [XR&CO'2014], Fri, 16 Jun 2017 07:22:11 GMT -->

</html>
```

6.2 Screenshot

6.2.1 Home page





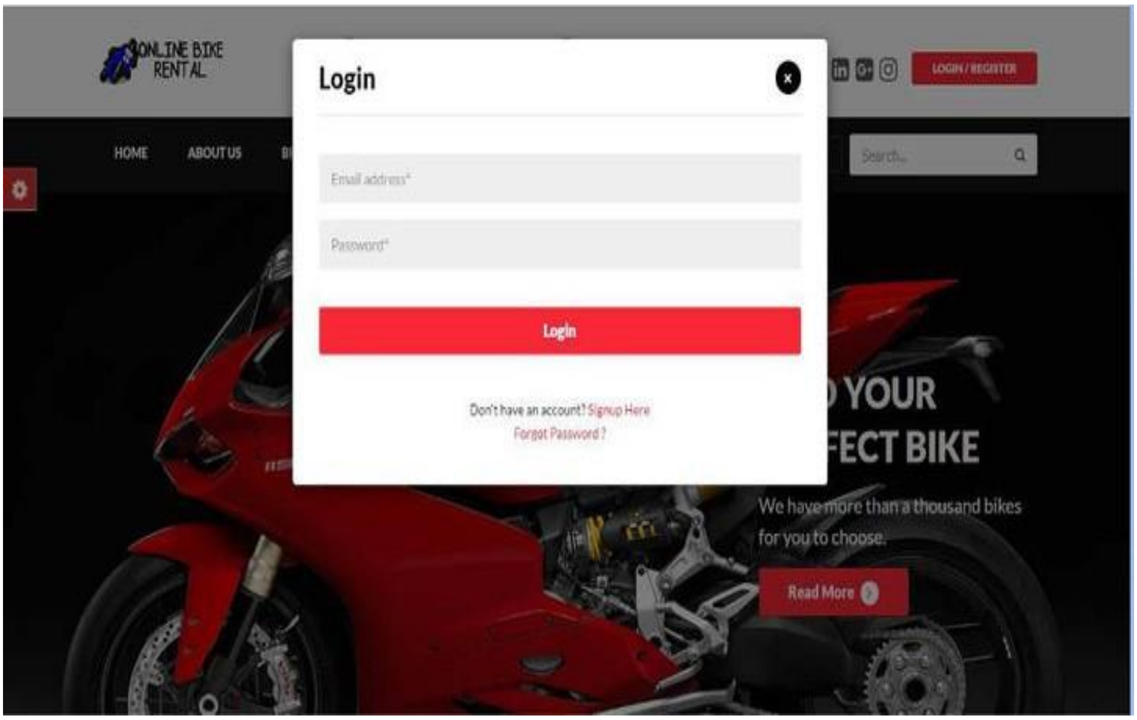
You will be able to fully enjoy your holiday and your ride! Any problems? Our passionate team will be happy to help you!! No waste of time during your holidays to find a rental point on the spot! No language barrier, thanks to our multilingual team! At the same price you would pay on the spot! We have best bikes with best deals

Slowly spreading its cards this year, the Ace of Bajaj Autos is still

The Pulsar is by far the most successful brand Bajaj has managed to cr

The Suzuki GSX-R1000 is a sport bike from

6.2.2 Login



Login

Email address*

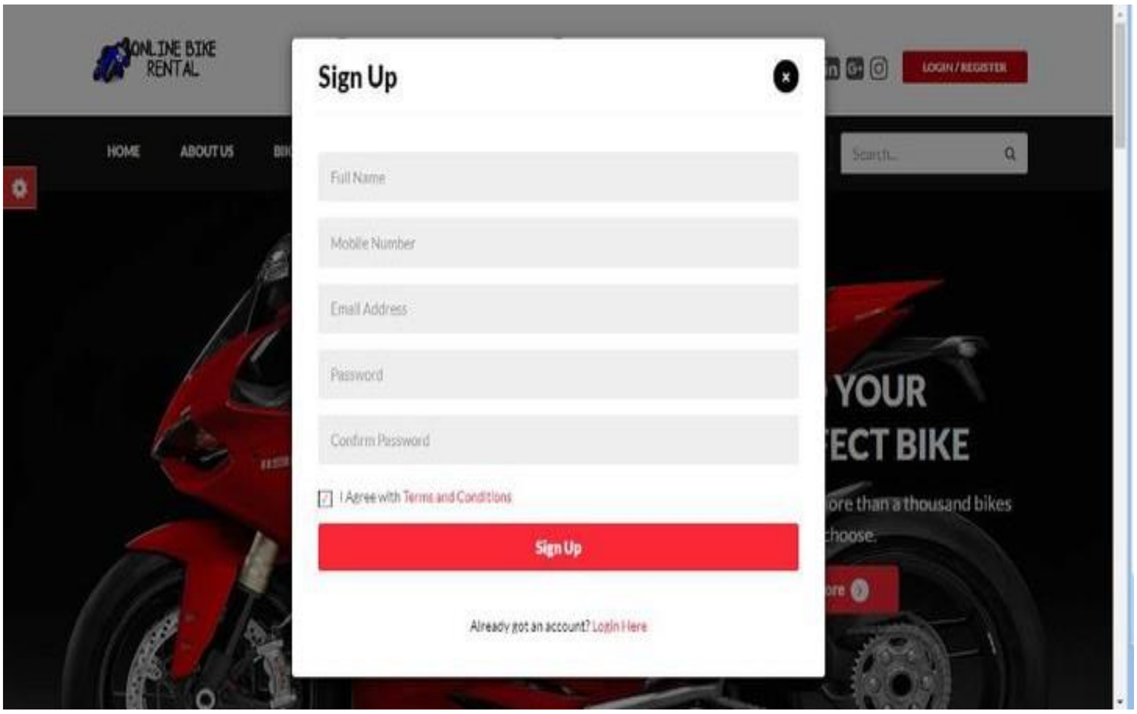
Password*

Login

Don't have an account? Signup Here
Forgot Password ?

We have more than a thousand bikes for you to choose.

6.2.3 Registration



Sign Up

Full Name

Mobile Number

Email Address

Password

Confirm Password

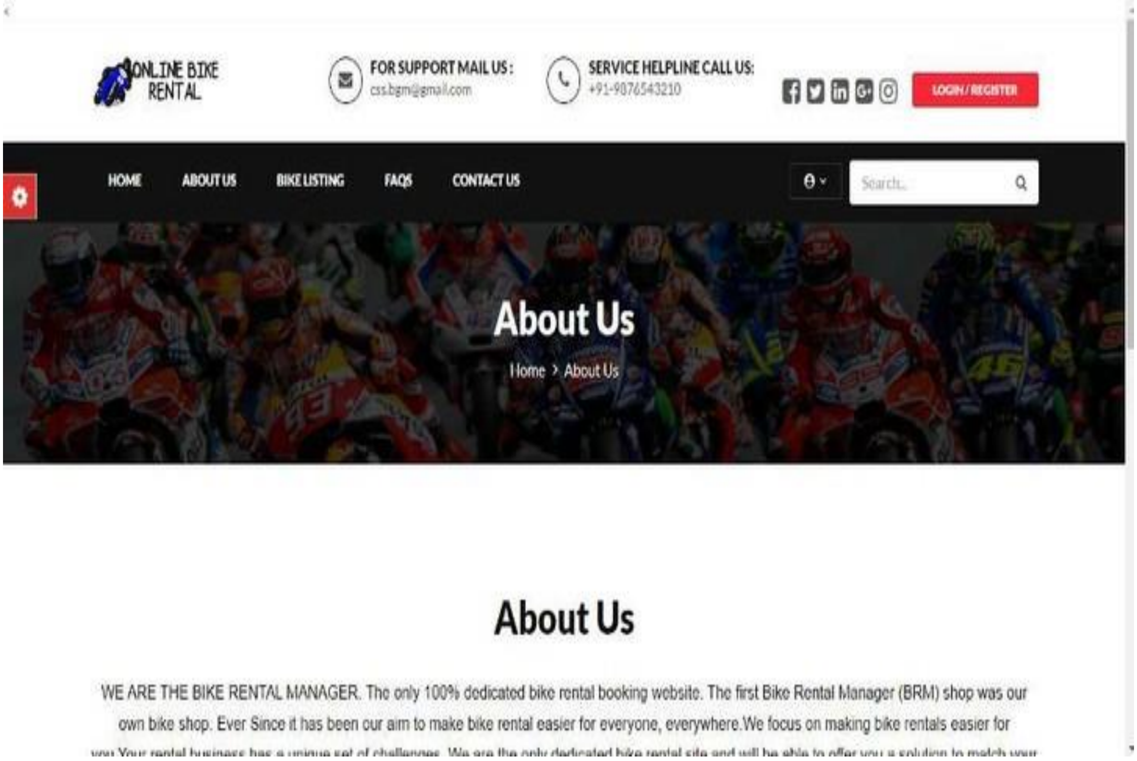
☑ I Agree with Terms and Conditions

Sign Up

Already got an account? Login Here

ore than a thousand bikes choose.

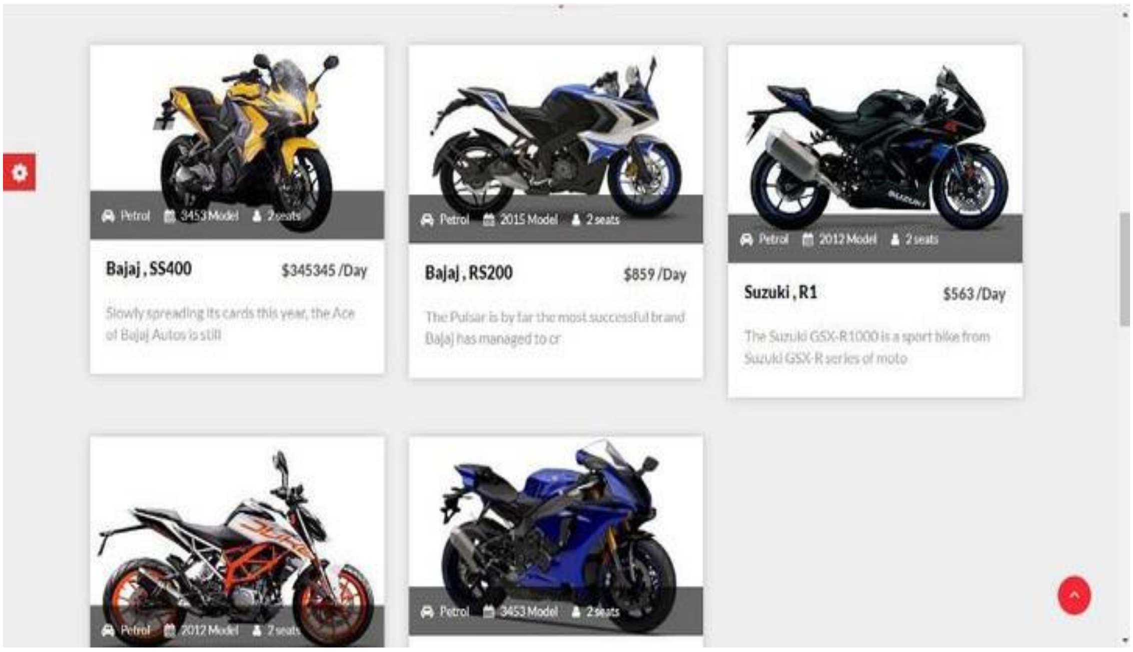
6.2.4 About



About Us

WE ARE THE BIKE RENTAL MANAGER. The only 100% dedicated bike rental booking website. The first Bike Rental Manager (BRM) shop was our own bike shop. Ever Since it has been our aim to make bike rental easier for everyone, everywhere.We focus on making bike rentals easier for you.Your rental business has a unique set of challenges. We are the only dedicated bike rental site and will be able to offer you a solution to match your

6.2.5 Listening Bike



Bajaj , SS400 $345345 /Day

Slowly spreading its cards this year, the Ace of Bajaj Autos is still

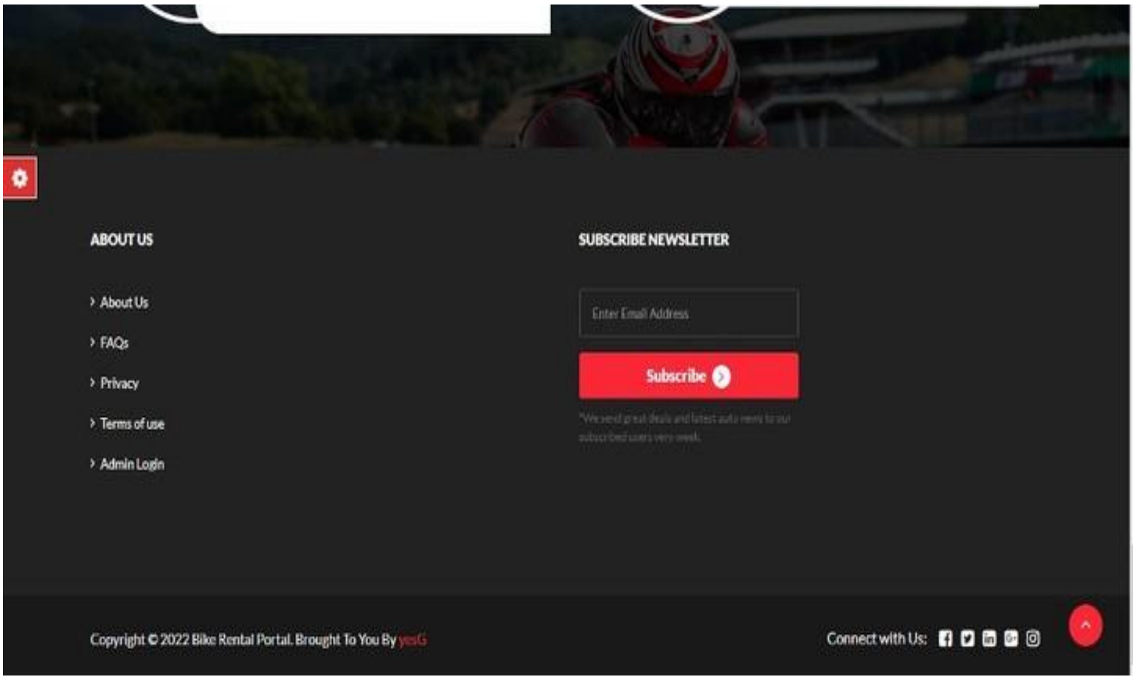
Bajaj , RS200 $859 /Day

The Pulsar is by far the most successful brand Bajaj has managed to cr

Suzuki , R1 $563 /Day

The Suzuki GSX-R1000 is a sport bike from Suzuki GSX-R series of moto

6.2.6 Footer



ABOUT US

- About Us
- FAQs
- Privacy
- Terms of use
- Admin Login

SUBSCRIBE NEWSLETTER

Enter Email Address

*We send great deals and latest auto news to our subscribed users very week.

Copyright © 2022 Bike Rental Portal. Brought To You By yesG

Connect with Us:

CHAPTER-7 TESTING

Software testing is an essential part of ensuring the quality of software since it acts as the last evaluation of description, design, and coding.

Software is tested by being executed with the intention of finding errors. A multitude of test instances are used to run the application under test during testing. The program's output for every testing ground is then examined to see whether it is operating as expected.

Testing Goals:

1. Assessment is the process of operating software in search of errors.

2. An effective test case design will have the probability of identifying a problem that hasn't yet been discovered.

3. A positive test is one that identifies a mistake that hasn’t been found yet.

These objectives point to a major change in perspective . Testing can only show that these are software issues; it cannot show that there are no defects.

Testing methodologies: Unit

Unit testing concentrates verification focuses its attention on the module, the smallest unit of software design to ensure that the system works properly as a whole, this test concentrates on each module independently. Because it guarantees that each component is tested independently, testing phase was selected as the term.

Integrity checking

This is a methodical process for integrating several program modules into a single software framework. This test identifies and validates all faults within the module.

Product testing

To determine if the provided output is accurate or inaccurate, output testing is used.

Validation Examination

when the software for integration testing is complete and meets the specifications. Nonetheless, it has to be verified in accordance with the specification to find any unforeseen future flaws and to increase its dependability.

Strategies for Software Testing

A plan for software testing gives the software developer a direction. Testing is a set of procedures that may be planned of time and executed methodically. For this reason, the software engineering process should specify a template for software testing, or a collection of processes into which we may insert certain test case creation methodologies. Any software testing approach should incorporate the following characteristics:

1. Assessment "outward" towards the integration of the full computer-based system starts at the module level.

2. Several testing methodologies are acceptable at various times.

3. Testing is carried out by the software's creator and a separate test team.

4. Although testing and debugging are separate tasks, every testing method must take debugging into account.

CHAPTER-8 CONCLUSION

A small group of individuals can use the online program known as the "BIKE Rental System." To protect the security of the data and prevent duplicate data, Admin has been granted access permissions and is limited to specific features in this application. A laptop and a system joint are required to access the Application since the Data has been stored electronically. It is software that enables users to hire bicycles in accordance with their needs. This program improves competence and reduces the volume of data entered by people. Ultimately, this program can carry out all duties precisely and perform the function for which it was created.

Any Suggestions are Welcome

Once the final Online car rental, is built, users or clients can book the bike already exists.

in the web application. In future, users or clients can add bikes to the rent section and also we like to add slide bar of trending bikes in web Page. Also, we like to add a Chat Bot which helps user or clients to enquire their doubts related to the application.

Further enhancement, the use of search engine can be customizable using the filter option according to the user or a reader.

CHAPTER-9 BIBLIOGRAPHY

1. Bai, X. (2014). The effect of client made content on wayfarer lead: An observational appraisal on the impacts of e-verbal exchange to inn online plans., 30, 202-212.

2. Choi, Y. (2017). Forerunners and outcomes of confidence in online thing studies: A meta-evaluation., 39, 177-191.

3. Choi, Y. J. (2017). Online outlines as major correspondence: Impacts of valence, language style, and reaction system on consumer decisions., 104-115.

4. Du, Q. (2018). A relative evaluation of major online outline stages: Thoughts for virtual redirection assessment in kindheartedness and the development business. 68, 157-170.

5. Gupta, N. (2018). Effect of virtual entertainment on lodging industry, 6(1), 1-10.

6. Gupta, S. (2017). Respect co-creation and buy point in neighborly trade: The control of client strengthening, social alliance, and thing show., 37(2), 179-189.

7. Jiang, Z. (2016). Analysing the impacts of trust and online outlines on online clients' buy point. Internet Examination, 26(3), 622-637.

8. Kim, Y. K. (2018). Understanding Airbnb as a participatory stage for solace trade. 57(7), 883-898.

9. Kumaidi, A. (2020). The impact of seen help, saw accommodation, and saw charm on the target to utilize online travel reservation: 8(1), 1-9.

10. Kumar, R. (2017). Cross-lingual data recovery, 10(1), 1-133.

11. Lee, C. K. (2017). The impact of site quality on online repurchase point: The places of data quality, redirection, and page trust. 18(3), 196-209.

12. Li, H. (2018). Virtual redirection and its ramifications for genuine picture: An important assessment of Dalian, China., 246-256.

13. Li, Y. (2019). Language dealing with for cross-lingual text assembling: An outline. IEEE Access, 7, 152035-152046

14. Lin, J. C. (2016). Impacts of bilingualism and multilingualism on the mental framework. Surveys in the Neurosciences, 27(7), 739-748.

15. Montazeri, S. (2020). A cognizant outline of eWOM in the development business and warmth industry: A structure of its drivers, advantages, and difficulties. 23(10), 1248-1264.

16. Nasiripour, A. A. (2017). E-business and its effect on the friendliness business: Assessment of likely benefits and inconveniences. 5(2), 1-12.

17. Rule, R. (2015). Actually taking a look at Chinese and German sightseers' dynamic styles: An exploratory review out. Diary of Improvement Examination, 54(1), 63-76.

18. Santosa, P. I. (2017). Staying booking online: Parts affecting clients' choice. Generally speaking Diary of Applied Business and Money related Examination, 15(18), 137-146.

19. Shrum, L. J. (2002). What is information and is it for every circumstance such something to be grateful for? 31(4), 53-64.

20. Sivaramakrishnan, S. (2017). Showing the board. Pearson Direction India.

21. Soesanto, H. (2019). The impact of website page quality and association quality on fulfillment and devotion., 7(2), 23-34.

22. Stepchenkova, S. (2019). Objective picture and endurance: An outline of forming and examination plan. 12, 50-61.

23. Wang, D. (2018). The effect of electronic redirection on the development business: An analysis of observational evaluation., 57(1), 87-99.

24. Wang, N. (2019). Analyzing the control of mental, very close and social confidence in building voyager perseverance towards online the development business stages. 70, 211-223.

25. Weitz, B. A. (2006). When should a retailer spread out an elating store climate? Diary of Showing, 70(1), 107-118.

26. Zhang, H. (2016). An overall assessment of cross-line online business in China, Japan, and South Korea. 28(3), 524-542.

27. Zhang, H., (2019). An exact pack considering the ability to get a handle on individuals on a more significant level on the correspondence skill of neighboring escorts, 40, 27-36.

28. Zhang, Y. (2016). Chinese vacationers' improvement data sources and objective decisions. Diary of Genuine Showing and The board, 5(4), 351-360.