
CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

The concept of **Bike Rental Database** is vast by itself. The bike rental data set is a classic data analytics problem. It is set in the context of bike sharing system where users can rent a bike at Location A and return it at Location B. The user's movement is monitored for sensing mobility or events where users are using the bikes in a conducive environment. The whole objective of this project is to provide the user a streamlined experience to access and fetch the details of their respective bookings. This system has only one admin account and cannot have more than one admin account.

1.2 OBJECTIVES

The aim of the project, **Bike rental database** is to meet the following objectives:

- To provide a great experience to the user.
- To provide a platform to collect and optimize the data-based relevancy.
- To help the administrators easily access, edit, add and delete information.
- To ensure safety of the information in the database.
- To ultimately provide an interactive and user-friendly platform for the users to have a good experience with least number of glitches and delays.

1.3 SCOPE

The scope of this project is to provide a platform for users and rental system owner an effective and efficient communication on rent age of bike and also to provide an error free or glitch free experience for the users.

CHAPTER 2

LITERATURE STUDY

2.1 TRADITIONAL DATABASE

In the early days of computing, data management and storage were a very new concept for organizations. The traditional approach to data handling offered a lot of the convenience of the manual approach to business processes (e.g., handwritten invoices & account statements, etc.) as well as the benefits of storing data electronically.

The traditional approach usually consisted of custom-built data processes and computer information systems tailored for a specific business function. An accounting department would have their own information system tailored to their needs, where the sales department would have an entirely separate system for their needs.

Initially, these separate systems were very simple to set up as they mostly mirrored the business process that departments had been doing for years but allowed them to do things faster with less work. However, once the systems were in use for so long, they became very difficult for individual departments to manage and rely on their data because there was no reliable system in place to enforce data standards or management.

Separate information systems for each business function also led to conflicts of interest within the company. Departments felt a great deal of ownership for the data that they collected, processed, and managed which caused many issues among company-wide collaboration and data sharing. This separation of data also led to unnecessary redundancy and a high rate of unreliable and inconsistent data

2.2 PROS AND CONS OF TRADITIONAL DATABASE

PROS:

- Simple
- Low initial investment

CONS:

- Separated ownership
- Unmanaged redundancy
- Data inconsistency
- Lack of data sharing
- Expensive on the long run

2.3 DOWNFALL OF TRADITIONAL MANAGEMENT SYSTEM

Conceived in a relatively centralized era when software was deployed in static environments, legacy database architectures fail to support an increasingly mobile world where applications are accessed anytime, anywhere. Today software users want consistent improvements in usability and expect SaaS vendors to deliver new features and functionalities needed to achieve their business objectives.

However, legacy database technologies fall short in serving the needs of today's distributed and cloud environments for the following reasons:

- Inadequate Latency issues
- Insufficient provisions during peak demands
- Lack- of high availability at all times
- Increasing operational costs
- Inability to meet the demands of global markets
- The failover capabilities

For all of these reasons, traditional databases are unable to deliver results in a rapidly growing environment where the workload is geographically distributed across heterogeneous data centers. Upgrading to a more distributed data model is costly and complicated and your DBAs can't just sit back and give up on this situation. Hence, due to these various reasons, the downfall of the traditional system was inevitable.

2.4 INTRODUCTION TO THE DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) refers to the technology for creating and managing databases. Basically, a DBMS is a software tool to organize (create, retrieve, update and manage) data in a database.

The main aim of a DBMS is to supply a way to store and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft ACCESS, or EXCEL to store data in the form of database. A datum is a unit of Data Meaningful data combines to form information. Hence, information is interpreted data-data provided with semantics. MS ACCESS is one of the most common database management software.

Database systems are meant to handle large collections of information. Management of data involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crash or attempts at unauthorized access.

2.5 INACTIVE AREAS FOR THE USE OF DBMS

- Airlines: Reservations, Schedules etc.
- Telecom: Calls made, Customer details, Network usage etc.
- Universities: Registration, Results, Grades, etc.
- Sales: Products, Purchases, Customers etc.
- Banking: All transactions, Customer details etc.

2.6 ADVANTAGES OF DBMS

- **DATA INDEPENDENCE:** Application programs should be as free or independent as possible from details of data representation and storage.
- **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 in external devices
- **DATA INTEGRITY AND SECURITY:** If data is accessed through DBMS, the DBMS can enforce integrity constraint on the data.
- **DATA ADMINISTRATION:** When several users share the data, integrating the administration of the data can offer major improvements. Experienced professionals understand the nature of data being managed and can be responsible for organizing the data representation to reduce redundancy and make the data to retrieve efficiently.

2.7 COMPONENTS OF DBMS

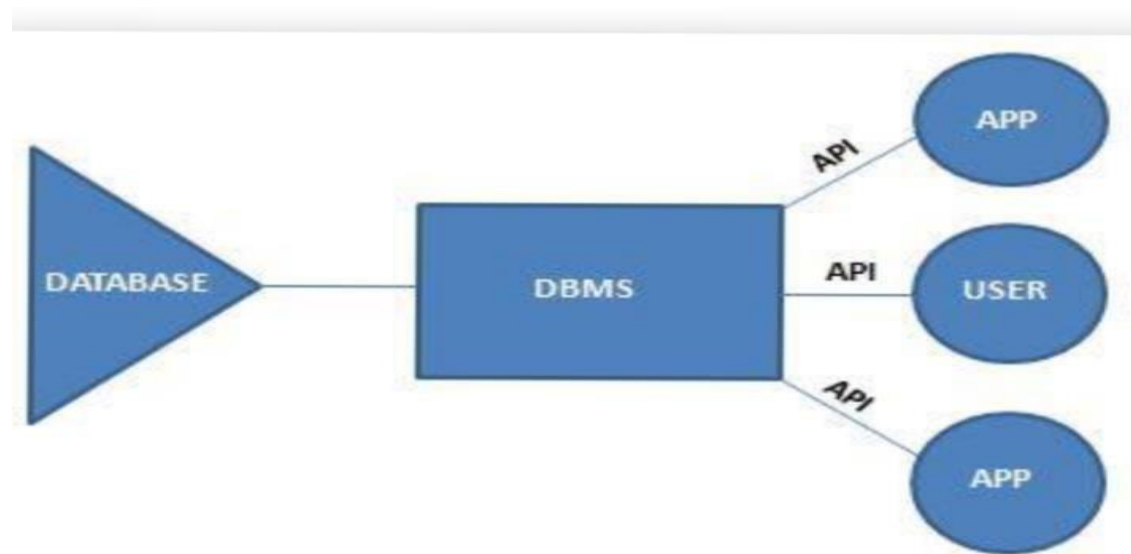


Figure 2.1: Components of DBMS

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

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS:

- Processor: Intel Core 2 Duo or above
- RAM: 2GB or above
- Hard Disk: 2GB or above

3.2 TECHNOLOGIES USED:

- Front End: CSS, PHP, HTML, Bootstrap
- Connection/Controller: PHP
- Back-End Database: MySQL

3.3 SOFTWARE:

- Server: XAMPP 7 server
- Operating System: windows 10
- Database Support: MySQL 5

CHAPTER 4

SYSTEM DESIGN

4.1 ENTITIES AND ATTRIBUTES:

An **entity** set is a collection of similar types of entities. An entity set may contain entities with attribute sharing values. For example, a student's set may contain all the students of a school; likewise, a teacher set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

An **attribute** is defined as a quality or characteristics of a person, place, or thing. Real life individuals and fictional characters possess various **attributes**. For example, someone might be labelled beautiful, charming, funny, or intelligent.

USERS:

- **id(primary key)**
- fname
- lname
- Username
- Password
- Avatar
- Last_login
- Type
- Data_added
- Data_updated

SYSTEM_INFO:

- **id(primary key)**
- metafeild
- meta_value

CLIENTS:

- **id(primary key)**
- fname
- lname
- gender
- contact
- Email
- Password

- Address
- Date_created

BRAND_LIST:

- **id(primary key)**
- status
- Date_created

CATEGORIES:

- **id(primary key)**
- category
- description
- Status
- Date_created

BIKE_LIST:

- **id(primary key)**
- Brand_id
- Category_id

RENT_LIST:

- **id(primary key)**
- Client_id
- Bike_id
- Date_start
- Date_end
- Rent_days
- Amount
- Status
- Date_created

4.2 ENTITY-RELATIONSHIP DIAGRAM:

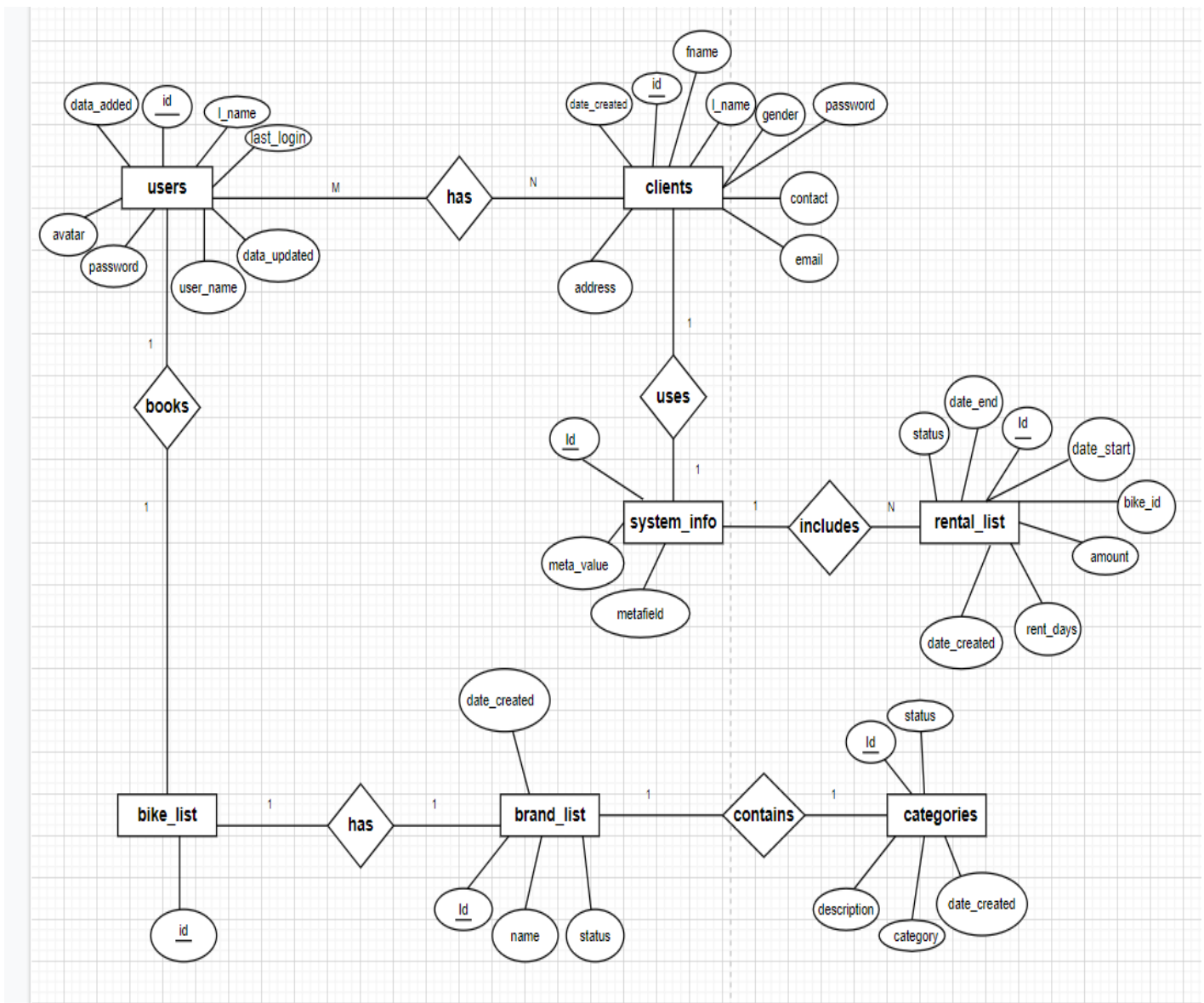


Figure 4.1: E R Diagram of Bike Rental Database

The ER (Entity Relationship) Diagram represents the model of Bike Rental Database Management System Entity. The entity-relationship diagram of Bike Rental Database Management System shows all the visual instrument of database tables and the relations between Booking, Bike, Payment etc. It used structure data and to define the relationships between structured data groups of Bike Rental Database Management System functionalities. The main entities of the Bike Rental Database Management System are USER, CLIENTS, SYSTEM_INFO, RENTAL_LIST, BIKE_LIST, BRAND_LIST etc.

4.3 RELATIONAL SCHEMA DIAGRAM:

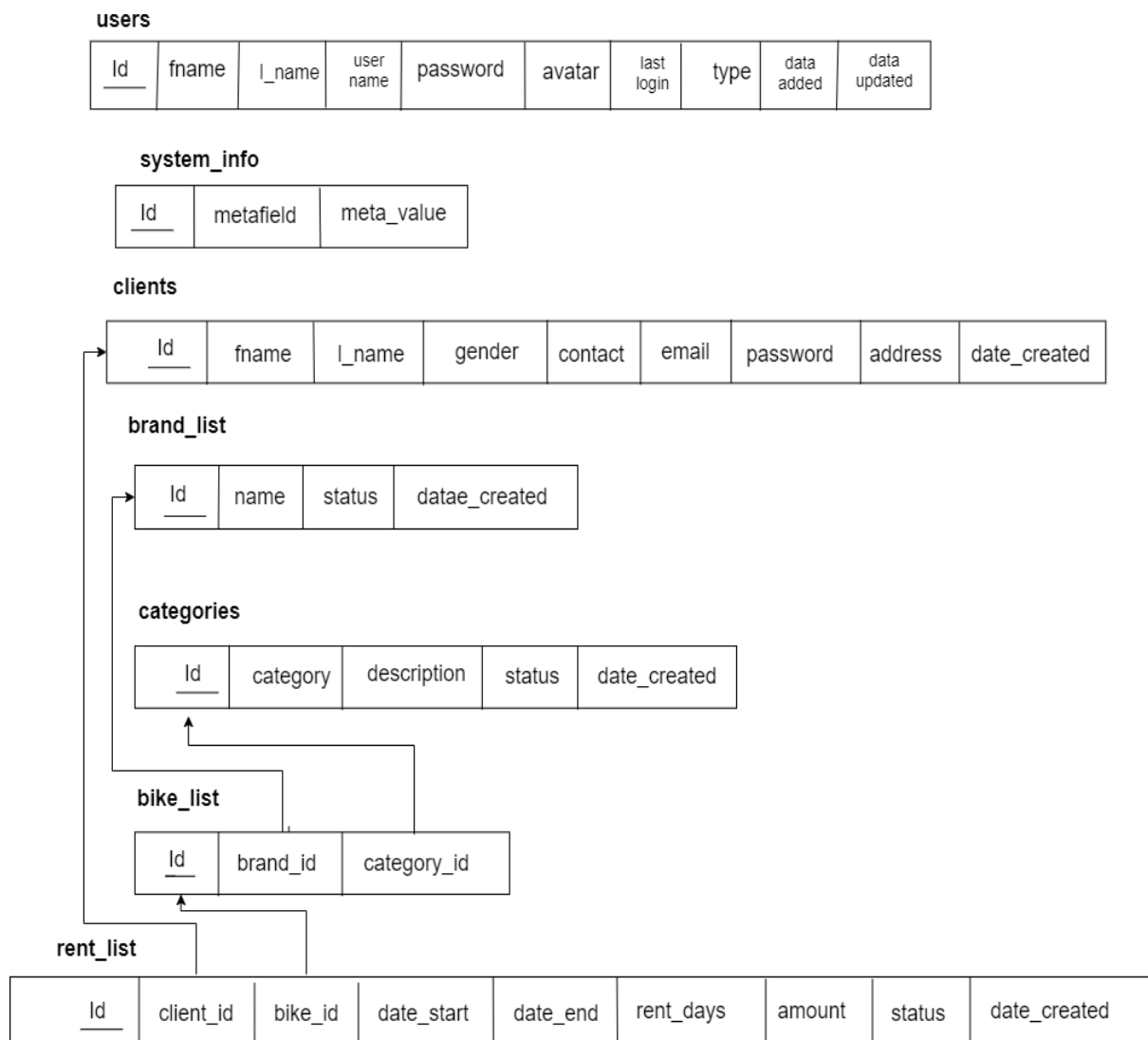


Figure 4.2: Relational schema of bike rental database

The project Online bike rental system is a platform that allows owners to post their bikes available for rental. The web application will serve as a middle man between the owner and customers.

The relational schema will provide you the list of tables and field/columns for every table in the design of database structure/schema of bike rental system.

This database design has 7 tables with their respective fields and columns as well as their relationships among each other.

4.4 NORMALIZATION:

Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relations may cause insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

4.4.1 FIRST NORMAL FORM:

If a relation contains composite or multi-valued attribute, it violates first normal form or a relation is in first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if every attribute in that relation is single valued attribute.

4.4.2 SECOND NORMAL FORM:

To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency. A relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

4.4.3 THIRD NORMAL FORM:

A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

4.4.4 FOURTH NORMAL FORM:

A relation schema R is in 4NF w.r.t a set of dependencies F if, for every non, trivial multivalued dependency $X \twoheadrightarrow Y$ in F^+ , X is a super key for R.

CHAPTER 5

IMPLEMENTATION

5.1 HTML:

HTML 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 standard.

It was published in October 2014 by the World Wide Web Consortium (W3C) to improve the language with support for the latest multimedia, while keeping it both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc. HTML is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.

HTML includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML is also a candidate for cross-platform mobile applications, because it includes features designed with low-powered devices in mind.

Many new syntactic features are included. To natively include and handle multimedia and graphical content, the new <video>, <audio> and <canvas> elements were added, and support for scalable vector graphics (SVG) content and MathML for mathematical formulas. To enrich the semantic content of documents, new page structure elements such as <main>, <section>, <article>, <header>, <footer>, <aside>, <nav> and <Figure >, are added. New attributes are introduced, some elements and attributes have been removed, and others such as <a>, <cite> and <menu> have been changed, redefined or standardized.

The APIs and Document Object Model (DOM) are now fundamental parts of the HTML specification and HTML also better defines the processing for, any invalid documents.

5.2 BOOTSTRAP:

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front- end web development. It contains CSS and JavaScript based design templates for typography,forms buttons, navigation and other interface components.



Figure 5.1: Cover picture of Bootstrap

Bootstrap is the third-most-starred project on GitHub, with more than 1,35,000 stars, behind only freeCodeCamp and marginally behind Vue.js framework. According to Alexa Rank, Bootstrap getbootstrap.com is in the top 2000 in US while vuejs.org is in top 7000 in US.

5.3 CASCADING STYLE SHEETS (CSS):

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of theWorld Wide Web, alongside HTML and JavaScript.



Figure 5.2: Cover picture of CSS

CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristic, enable multiple web pages to share formatting by specifying the relevant CSS file, and reduce complexity and repetition in the structural content.

5.4 PHP:

PHP is a servlet-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

PHP is a scripting language that helps people make web pages more interactive by allowing them to do more things.

A website programmed with PHP can have pages that are password protected. A website with no programming cannot do this without other complex things. Standard PHP file extensions are:php, php3or, HTML, but a web server can be set up to use any extension

5.5 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 [SAM or VSAM, SQL offers two main advantages

At 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.

SQL was one of the first commercial languages for Edgar F Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large SharedDataBanks".⁹¹ Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SQL became a standard of the American National Standards Institute (ANSI) in 1986 and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different data base systems without adjustments

5.6 CODE SNIPPETS

5.6.1 CREATE TABLE COMMANDS

USER TABLE

```
---CREATE TABLE `users` (  
  
    `id` int(50) NOT NULL,  
  
    `firstname` varchar(250) NOT NULL,  
  
    `lastname` varchar(250) NOT NULL,  
  
    `username` text NOT NULL,  
  
    `password` text NOT NULL,  
  
    `avatar` text DEFAULT NULL,  
  
    `last_login` datetime DEFAULT NULL,  
  
    `type` tinyint(1) NOT NULL DEFAULT 0,  
  
    `date_added` datetime NOT NULL DEFAULT current_timestamp(),  
  
    `date_updated` datetime DEFAULT NULL ON UPDATE current_timestamp()  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

SYSTEM_INFO TABLE

```
CREATE TABLE `system_info` (  
    `id` int(30) NOT NULL,  
    `meta_field` text NOT NULL,  
    `meta_value` text NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CLIENTS TABLE

```
CREATE TABLE `clients` (  
    `id` int(30) NOT NULL,  
    `firstname` varchar(250) NOT NULL,  
    `lastname` varchar(250) NOT NULL,  
    `gender` varchar(20) NOT NULL,  
    `contact` varchar(15) NOT NULL,  
    `email` varchar(250) NOT NULL,  
    `password` text NOT NULL,  
    `address` text NOT NULL,  
    `date_created` datetime NOT NULL DEFAULT current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

BRAND_LIST TABLE

```
CREATE TABLE `brand_list` (  
    `id` int(30) NOT NULL,  
    `name` text NOT NULL,  
    `status` tinyint(1) NOT NULL DEFAULT 1,  
    `date_created` datetime NOT NULL DEFAULT current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

CATEGORIES TABLE

```
CREATE TABLE `categories` (  
    `id` int(30) NOT NULL,  
    `category` varchar(250) NOT NULL,  
    `description` text DEFAULT NULL,
```

```
`status` tinyint(1) NOT NULL DEFAULT 1,  
`date_created` datetime NOT NULL DEFAULT current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

BIKE_LIST TABLE

```
CREATE TABLE `bike_list` (  
  `id` int(30) NOT NULL,  
  `brand_id` int(30) NOT NULL,  
  `category_id` int(30) NOT NULL,  
  `bike_model` text NOT NULL,  
  `description` text NOT NULL,  
  
  `quantity` tinyint(3) NOT NULL DEFAULT 0,  
  `daily_rate` float NOT NULL DEFAULT 0,  
  `status` tinyint(1) NOT NULL DEFAULT 1,  
  `date_created` datetime NOT NULL DEFAULT current_timestamp(),  
  `date_updated` datetime DEFAULT NULL ON UPDATE current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

RENT_LIST TABLE

```
CREATE TABLE `rent_list` (  
  `id` int(30) NOT NULL,  
  `client_id` int(30) NOT NULL,  
  `bike_id` int(30) NOT NULL,  
  `date_start` date NOT NULL,  
  `date_end` date NOT NULL,  
  
  `rent_days` int(11) NOT NULL DEFAULT 0,  
  `amount` float NOT NULL DEFAULT 0,  
  `status` tinyint(1) NOT NULL DEFAULT 0 COMMENT  
'0=Pending,1=Confirmed,2=Cancelled,3=Picked -up, 4 =Returned',  
  `date_created` datetime NOT NULL DEFAULT current_timestamp(),  
  `date_updated` datetime DEFAULT NULL ON UPDATE current_timestamp()
```

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

5.6.2 SIGN IN / LOG IN

5.6.2.1 HTML

```
<!doctype html>
<head>
<meta charset="utf-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>Login - Bike rental Management System</title>
<meta name="viewport" content="width=device-width, initial-scale=1"> <link
rel="shortcut icon" type="image/png" href="assets/images/icon/favicon.ico"> <link
rel="stylesheet" href="assets/css/bootstrap.min.css">
<link rel="stylesheet" href="assets/css/font-awesome.min.css">

<link rel="stylesheet" href="assets/css/themify-icons.css">
<link rel="stylesheet" href="assets/css/metisMenu.css">
<link rel="stylesheet" href="assets/css/owl.carousel.min.css">
<link rel="stylesheet" href="assets/css/slicknav.min.css">
<!-- amchart css -->
<link rel="stylesheet"
<!-- others css -->
<link rel="stylesheet" href="assets/css/typography.css">
<link rel="stylesheet" href="assets/css/default-css.css">
<link rel="stylesheet" href="assets/css/styles.css"> <link
rel="stylesheet" href="assets/css/responsive.css"> <!--
modernizr css -->
<script src="assets/js/vendor/modernizr-2.8.3.min.js"></script>
</head>

<body>
<div id="preloader">

<div class="loader"></div>

</div>
<div class="login-area">
<div class="container">
<div class="login-box pt--100">
```

5.6.2.2 CSS

```
html,
.login-area {
background: #F3F8FB;

}
.login-box {
display: -webkit-box;
```

```
display: -ms-flexbox;
display: flex;
min-height: 100vh;
}
.login-box form {
margin: auto;
width: 450px;
max-width: 100%;
background: #fff;
border-radius: 3px;
}
.login-form-head {
text-align: center;
background: #8655FC;
padding: 50px;
}
.login-form-head h4 {
letter-spacing: 0;
text-transform: uppercase;
font-weight: 600;
margin-bottom: 7px;
color: #fff;
}
.login-form-head p {
color: #fff;
font-size: 14px;
line-height: 22px;
}
.login-form-body {
padding: 50px;
}
.login-other a i {
margin-left: 5px;
}
.login-other a.fb-login {
background: #8655FC;
color: #fff;
}
.login-other a.fb-login:hover {
box-shadow: 0 5px 15px rgba(44, 113, 218, 0.38);
}
.login-other a.google-login {
background: #fb5757;
color: #fff;
}
.login-other a.google-login:hover {
box-shadow: 0 5px 15px rgba(251, 87, 87, 0.38);
}
.login-bg {
background: url(../images/bg/singin-bg.jpg) center/cover no-repeat;
position: relative;
```

```
z-index: 1;}
.login-bg:before {
content: "";
position: absolute;
left: 0;
top: 0;
z-index: -1;
height: 100%;
width: 100%;
background: #272727;
opacity: 0.7;
}
```

5.6.2.3 PHP

```
<?php session_start(); error_reporting(0);

include('includes/dbconnection.php');
if(isset($_POST['login']))
{
$adminuser=$_POST['username'];
$password=md5($_POST['password']);
$query=mysqli_query($con,"select ID from tbladmin where UserName='$adminuser'
&& Password='$password' ");
$ret=mysqli_fetch_array($query);
if($ret>0){
$_SESSION['zmsaid']=$ret['ID'];
header('location:dashboard.php');
}
else{
echo '<script>alert("Invalid Detail.")</script>';
}
}
?>
```

5.6.3 Triggers

A **database trigger** is procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database.

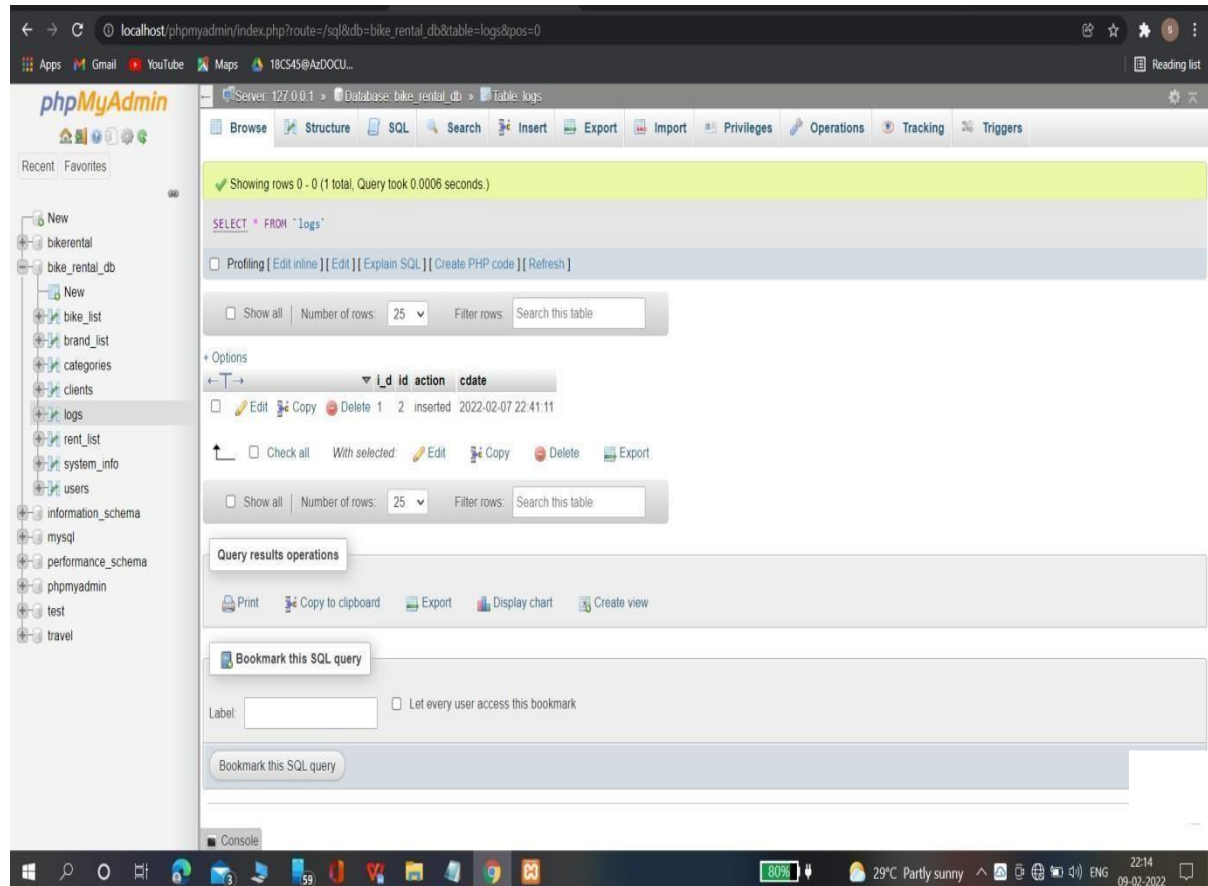
5.6.3.1 Code Snippet

```
CREATE TRIGGER `insertlogs`
AFTER INSERT ON `users`
FOR EACH ROW
INSERT INTO logs VALUES (null, NEW.id,'inserted', NOW ())
```

5.6.3.2 Snapshot of trigger

In this database, the admin can verify any modifications done to the database from the log which is created by the trigger operation.

Table 5.1: Trigger inserted as log table



CHAPTER 6

SNAPSHOTS AND DISCUSSIONS

6.1 Home Page

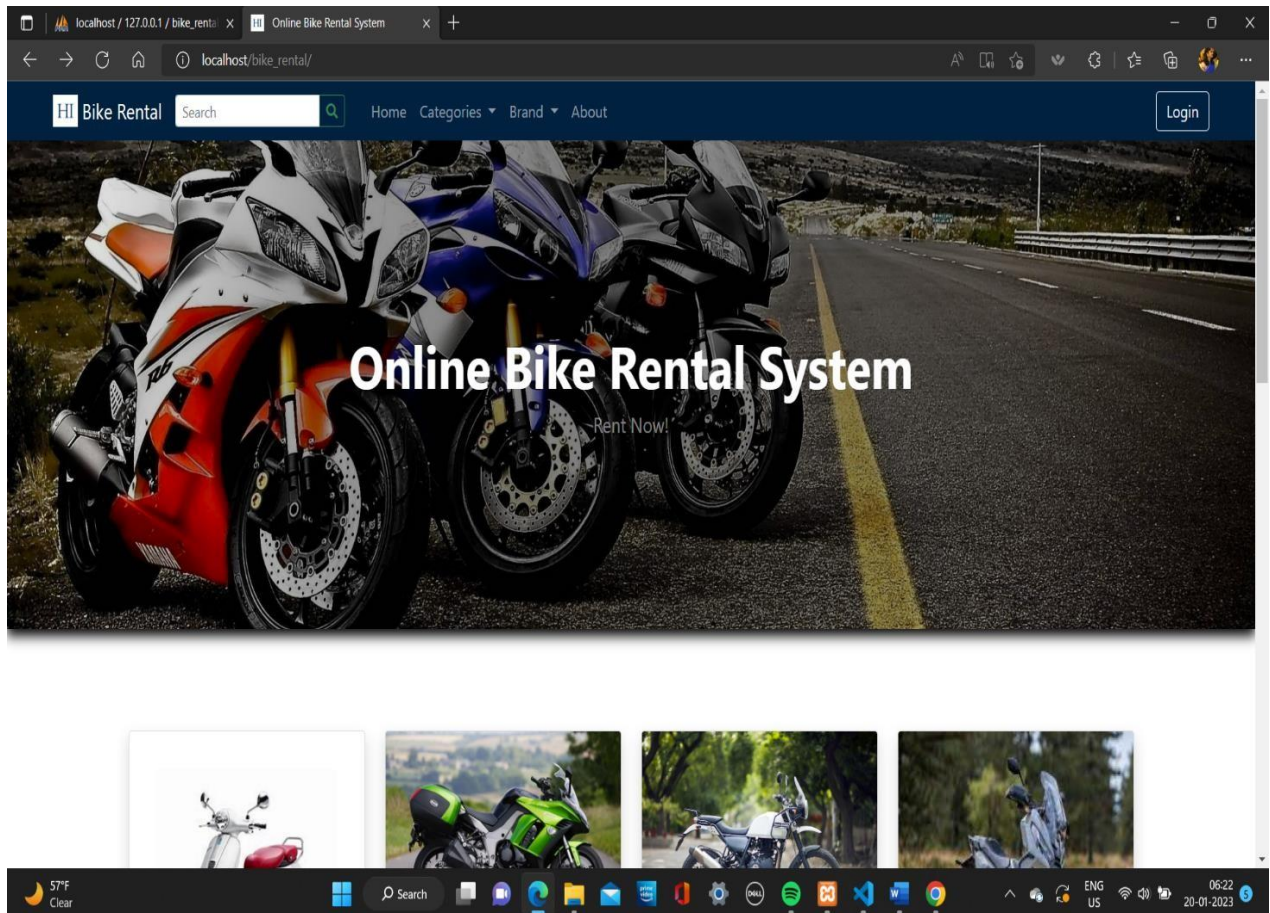


Figure 6.1: The home page

Figure 6.1 is a Home page that servers as the starting point of Bike Rental System website. It is the default web page that loads when you visit a website that only contain a domain addre

6.2 Login Page

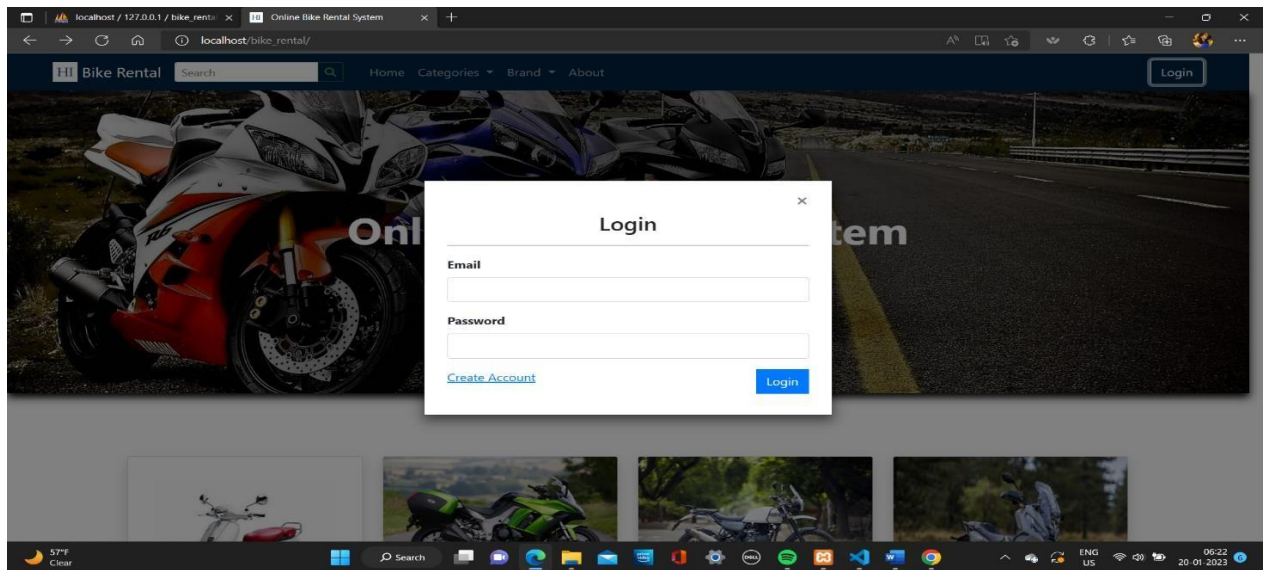


Figure 6.2: Login page

Figure 6.2 is a Login page or an entry page to a website that requires user identification and authentication, regularly entering username and password combination logins may provide access to an entire site or part of a website... some websites use cookies to track users during their logged in sessions.

6.3 Create new account

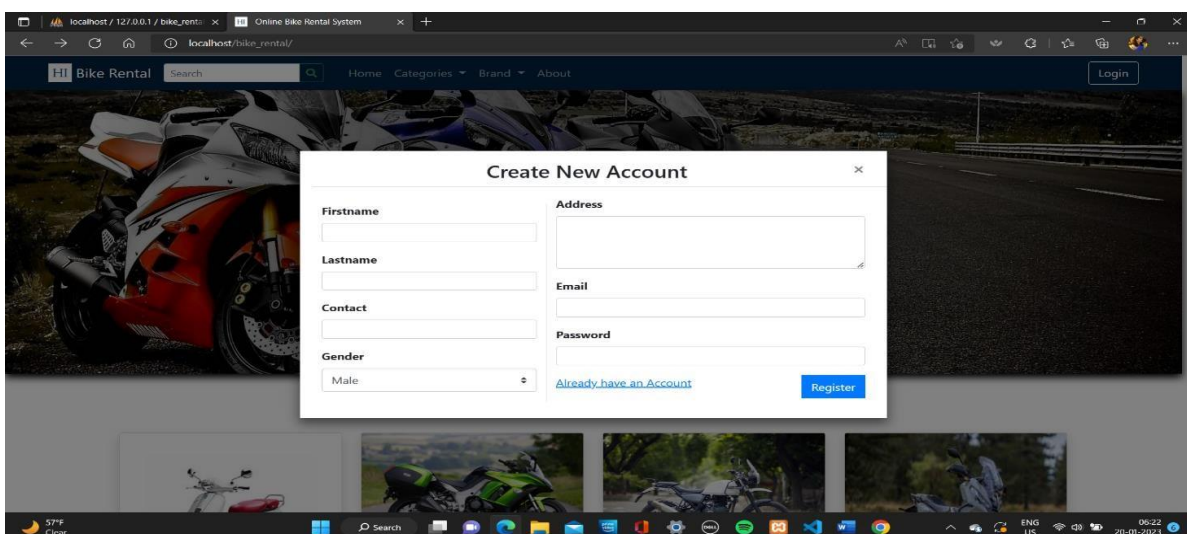


Figure 6.3: Create new account

Figure 6.3 is used to create new account which is useful to book a bike.

6.4 Bike categories

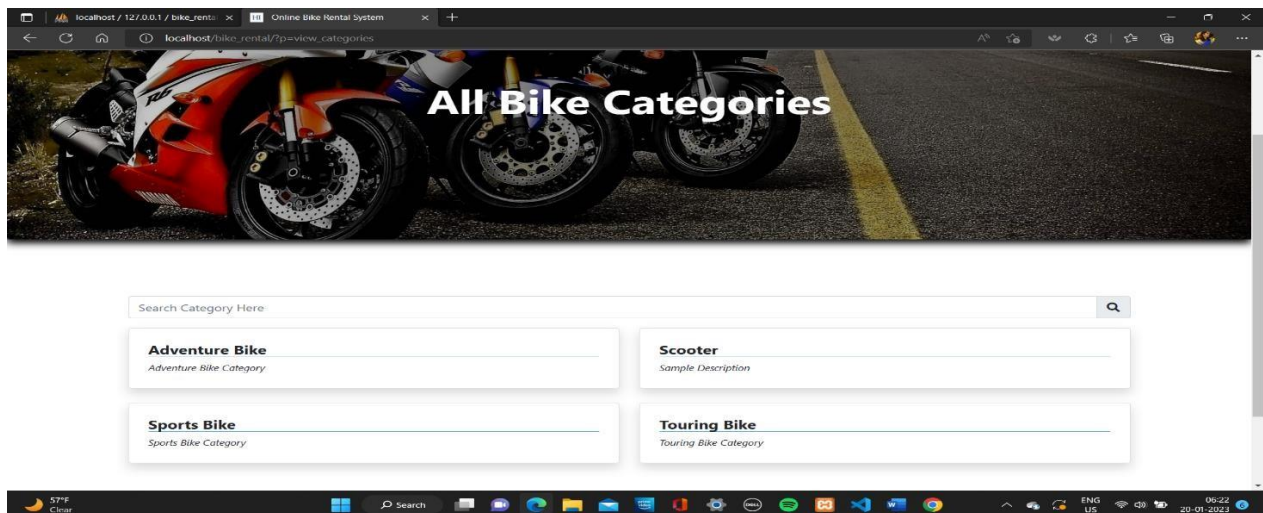


Figure 6.4: Bike categories

Figure 6.4 will show the bikes which available in the website

6.5 Bike type

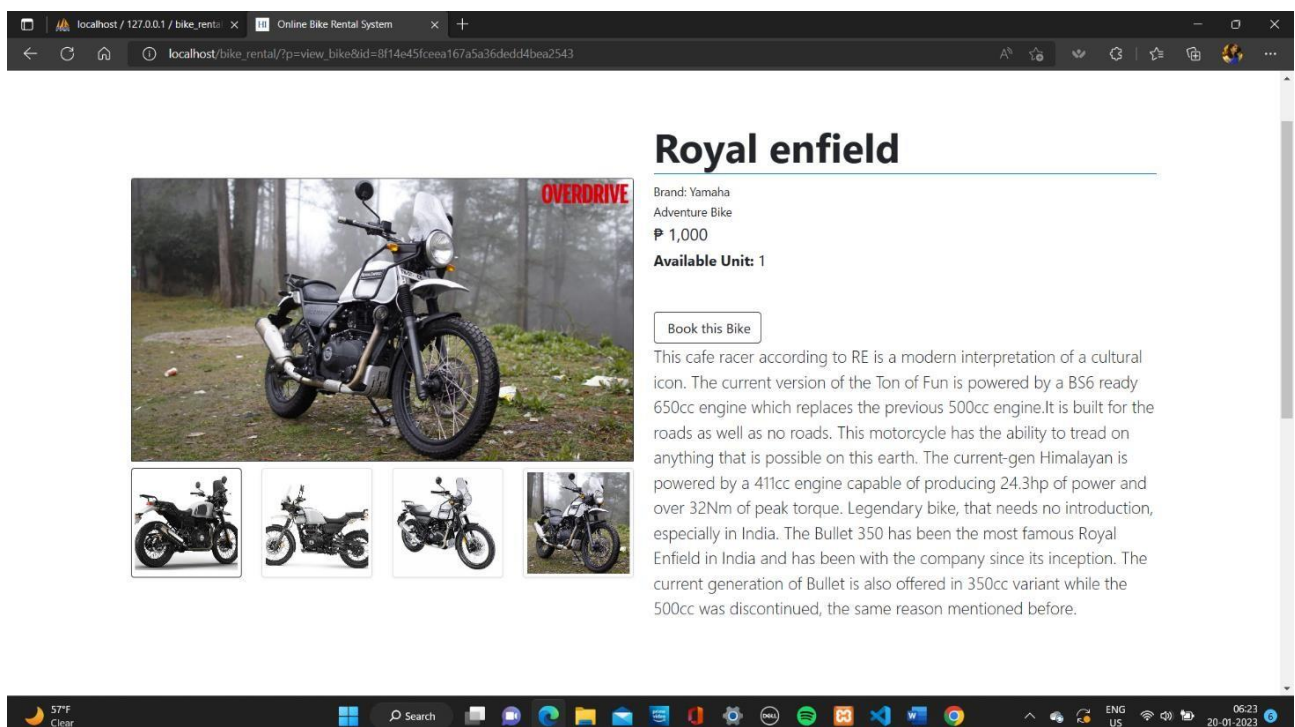


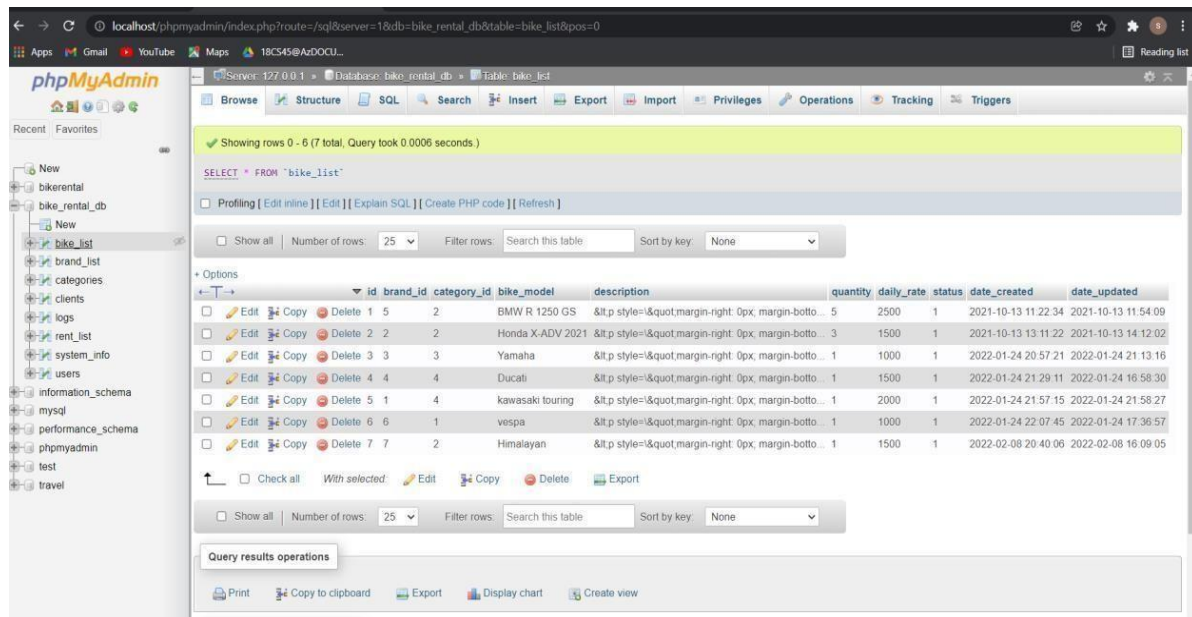
Figure 6.5: Bike type

Figure 6.5 will show the features of bike and cost per day

6.6 Bike list

Table 6.1 contains list of bikes available and their specifications, rate per day, category ID

Table 6.1: Bike list



Showing rows 0 - 6 (7 total, Query took 0.0006 seconds)

SELECT * FROM 'bike_list'

Options: Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

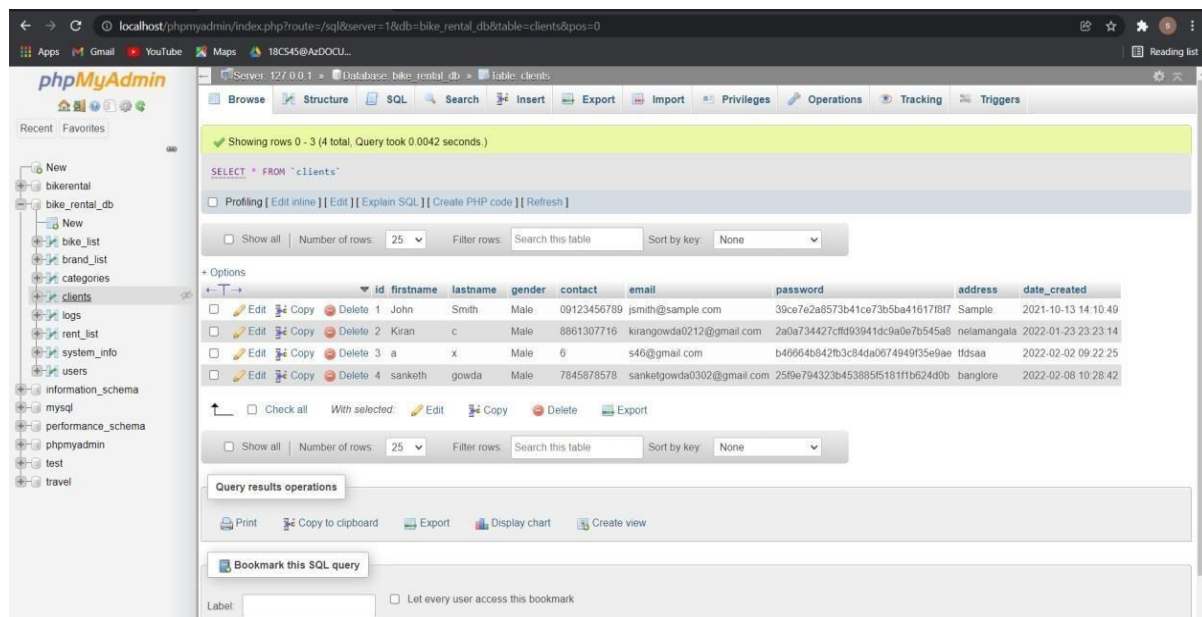
	id	brand_id	category_id	bike_model	description	quantity	daily_rate	status	date_created	date_updated
<input type="checkbox"/>	1	5	2	BMW R 1250 GS		5	2500	1	2021-10-13 11:22:34	2021-10-13 11:54:09
<input type="checkbox"/>	2	2	2	Honda X-ADV 2021		3	1500	1	2021-10-13 13:11:22	2021-10-13 14:12:02
<input type="checkbox"/>	3	3	3	Yamaha		1	1000	1	2022-01-24 20:57:21	2022-01-24 21:13:16
<input type="checkbox"/>	4	4	4	Ducati		1	1500	1	2022-01-24 21:29:11	2022-01-24 16:58:30
<input type="checkbox"/>	5	1	4	kawasaki touring		1	2000	1	2022-01-24 21:57:15	2022-01-24 21:58:27
<input type="checkbox"/>	6	6	1	vespa		1	1000	1	2022-01-24 22:07:45	2022-01-24 17:36:57
<input type="checkbox"/>	7	7	2	Himalayan		1	1500	1	2022-02-08 20:40:06	2022-02-08 16:09:05

Query results operations: Print | Copy to clipboard | Export | Display chart | Create view

6.7 Clients

Table 6.2 shows the users who can log in to the website and book the bikes through email and password.

Table 6.2: Clients



Showing rows 0 - 3 (4 total, Query took 0.0042 seconds)

SELECT * FROM 'clients'

Options: Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	firstname	lastname	gender	contact	email	password	address	date_created
<input type="checkbox"/>	1	John	Smith	Male	09123456789	jsmith@sample.com	39ce7e2a8573b41ce73b5ba416178f7	Sample	2021-10-13 14:10:49
<input type="checkbox"/>	2	Kiran	c	Male	8861307716	kirangowda0212@gmail.com	2a0a734427cfd93941dc9a0e7b545a8	nelamangala	2022-01-23 23:23:14
<input type="checkbox"/>	3	a	x	Male	6	s46@gmail.com	b46664b8427b3c84da0674949f35e9ae	tfdsaa	2022-02-02 09:22:25
<input type="checkbox"/>	4	sanketh	gowda	Male	7845878578	sanketgowda0302@gmail.com	25f9e794323b453885f518111b624d0b	bangalore	2022-02-08 10:28:42

Query results operations: Print | Copy to clipboard | Export | Display chart | Create view

Bookmark this SQL query

Label: ☐ Let every user access this bookmark

6.8 Rent list

Table 6.3 contains rent days of all bikes, amount for that particular bike, date when it was taken for rent and date of return

Table 6.3: Rent list

Showing rows 0 - 2 (3 total, Query took 0.0004 seconds)

```
SELECT * FROM `rent_list`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	client_id	bike_id	date_start	date_end	rent_days	amount	status	date_created	date_updated
<input type="checkbox"/>	7	2	4	2023-01-19	2023-01-28	10	25000	1	2023-01-19 15:48:19	2023-01-19 15:48:09
<input type="checkbox"/>	8	2	6	2023-01-21	2023-01-29	9	7200	1	2023-01-19 22:22:14	2023-01-19 22:22:58
<input type="checkbox"/>	9	3	5	2023-01-26	2023-01-29	4	8400	2	2023-01-19 22:27:15	2023-01-19 22:28:21

6.9 User

Table 6.4 shows the users, their first name, last name, password and username

Table 6.4: User details

Showing rows 0 - 2 (3 total, Query took 0.0005 seconds)

```
SELECT * FROM `users`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	firstname	lastname	username	password	avatar	last_login	type	date_added	date_updated
<input type="checkbox"/>	1	Administrator	Admin	admin	0192023a7bbd73250516f069d1f8b500	uploads/1624240500_avatar.png	NULL	1	2021-01-20 14:02:37	2021-06-21 09:...
<input type="checkbox"/>	4	John	Smith	jsmith	1254737c076cd867dc53d60a0364f38e	NULL	NULL	0	2021-06-19 08:36:09	2021-06-19 10:...
<input type="checkbox"/>	5	Claire	Blake	cblake	4744ddea876b11dcb1d169fadf494418	NULL	NULL	0	2021-06-19 10:01:51	2021-06-19 12:...

Report of user bookings on a particular day

My Bookings

Manage Account

Show 1 entries

Search:

#	Date Booked	Rent Schedule	Client	Status	Action
1	2023-01-19 22:22	Pick up: 2023-01-21 Return: 2023-01-29	keerthana MS	Confirmed	Action
2	2023-01-19 15:48	Pick up: 2023-01-19 Return: 2023-01-28	keerthana MS	Confirmed	Action

Showing 1 to 2 of 2 entries

Previous 1 Next

Copyright © Hithashree & Inchara
Developed By: [Hithashree&Inchara](#)

OneDrive
Screenshot saved
The screenshot was added to your OneDrive.

57°F Clear
Search
ENG US
06:24
20-01-2023

Figure 6.6: Details of the user who booked a bike

Figure 6.6 displays the required information of users who had booked a bike on that particular date. We can search for any number of users who had booked a bike on those dates.

CONCLUSION

With the theoretical knowledge of this subject, we created "BIKE RENTAL DATABASE" & gained a practical experience about how to create and develop a project which is a necessary factor for all the students. It becomes very necessary to take the utmost advantage of any opportunity of gaining practical knowledge. The construction of this mini project was one of those opportunities and thereby would like to thank all our teachers for guiding us through this project.

The whole objective of this project is to provide a streamlined experience for all the users to access information in a easy and efficient manner. The management of various data related to a bike through a admin is achieved. We tried to some extent in order to reach our motive, it can be still increased by doing with future development and iterations to this project.

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, McGraw Hill
- [3] Silberschatz Korth and Sudharshan, Database System Concepts, 6th Edition, Mc-GrawHill, 2013.
- [4] Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementation and Management, Cengage Learning 2012
- [5] Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version, 3rd / 4th Edition, Pearson Education, 2011
- [6] Edward Angel: Interactive Computer Graphics- A Top Down approach with OpenGL, 5th edition. Pearson Education, 2008
- [7] W3 Schools (HTML, CSS and PHP reference) – <https://www.w3schools.com/>
- [8] The Bootstrap Website (for Design) – <https://getbootstrap.com/>
- [9] PHP Official Documentation – <https://php.net/docs.php/>
- [10] Java Script learning – <https://www.learn-js.org/>
- [11] Wikipedia – <https://www.wikipedia.org/>
- [12] Official Websites - Bounce bike rental-<https://www.bounceshare.com/>
Vogo bike rentals : <https://www.vogo.in/>