

# Visvesvaraya Technological University

BELGAUM, KARNATAKA - 590 014.



A MINI-PROJECT REPORT  
ON  
**“CELESTIAL RECORDS”**

Submitted By  
**ANKIT GAURAV [4PM17CS009]**  
**BUDEESH B K [4PM17CS021]**

*Submitted in partial fulfillment of the requirement for the award of degree of*

**BACHELOR OF ENGINEERING**  
**IN**  
**COMPUTER SCIENCE AND ENGINEERING**

Under the Guidance  
of  
**Mr. Raghavendra K**  
Asst. Prof., Dept. of CS & E.  
PESITM, Shivamogga



**PES Institute of Technology and Management**  
**Department of Computer Science & Engineering**  
NOV-2019

# PES Institute of Technology & Management

NH-206, Sagar Road, Shivamogga-577 204

(Affiliated to Visvesvaraya Technological University, Belgaum)

## Department of Computer Science and Engineering



### CERTIFICATE

Certified that the Mini-project work entitled “**CELESTIAL RECORDS**” carried out jointly by **ANKIT GAURAV** USN **4PM17CS009** and **BUDEESH B K** USN **4PM17CS021** bonafide students of **PES INSTITUTE OF TECHNOLOGY & MANAGEMENT** in partial fulfillment for the award of Bachelor of Engineering in **COMPUTER SCIENCE & ENGINEERING** of the Visvesvaraya Technological University, Belgaum during the year **2019**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

#### GUIDE

**Mr. Raghavendra K**  
Assistant Professor,  
Dept. of CS&E.  
PESITM, Shivamogga.

#### Coordinator

**Mr. Sunil Kumar H R**  
Assistant Professor,  
Dept. of CS&E.  
PESITM, Shivamogga.

#### HOD

**Dr. Jagadeesha S N**  
Professor and Head,  
Dept. of CS&E  
PESITM, Shivamogga.

#### Name of the Examiners

1.....

2.....

#### Signature with date

## **Abstract**

Celestial record database system is a Database application which is helpful for people who want to know about the details on planet ,meteorite ,stars,comets etc. user can search the object by specifying the name of objects . This mini-Project is implemented using HTML,CSS and PHP.

Operations supported by the application are insert, delete, update and retrieve. Admin has rights to insert, update and delete the values of table. An user just view the contents of the table and he can search for the object by giving the object name.

# Acknowledgement

We take this opportunity to express our deep sense of gratitude to our Project guide **Mr.Raghavendra K.**, Asst. Prof., Department of CS&E, PESITM, for his keen interest and invaluable help throughout the completion of the project.

We would also like to express our sincere gratitude to **Mr.Sunil kumar H R.**,Asst. Prof., Dept. of CS&E, PESITM., for the kind support and guidance as project co-ordinator.

We are very much indebted and thankful to **Dr. Jagadeesha S N.**,Prof. and Head, Dept. of CS&E, PESITM, for his valuable guidance, encouragement and support.

We are highly grateful to **Dr. Chaitanya Kumar M V**, Principal, PESITM, for permitting us to carry out this project work in the institution.

Finally, we would like to thank all the teaching and non-teaching staff of Dept. of CS&E for their kind co-operation during the course of the work. The support provided by the College, the IT Department and Departmental library is gratefully acknowledged.

# Contents

Abstract	i
Acknowledgment	ii
Contents	iii
List of Figures	vi
Chapters	
1 Introduction	1
1.1 HTML	1
1.2 PHP	6
1.3 Database	7
1.4 Mysql	7
1.5 WAMP	8
1.6 Notepad++	8
1.7 Web Browser	8
2 System Analysis and Design	10
2.1 Requirement analysis	10
2.2 ER-Diagram	11
2.3 Relational Schema	12
2.4 Functional Requirements	13
2.5 Non-functional requirements	13
2.6 Use Case Diagram	14
3 System Implementation	15
3.1 Database design	15
3.2 Database connectivity	17
3.3 Implementation of database operation	18
4 Results and Discussion	25
Conclusion	29
References	30

## **List of Figures**

2.1	ER Diagram for celestial database	11
2.2	Relational Schema for celestial database	12
2.3	Use Case diagram	14
4.1	Login page of project	25
4.2	Welcome page of project	26
4.3	Update and View page	26
4.4	View and search page	27
4.5	Insert page	27
4.6	Search table page	28
4.7	Update page	28

## **CHAPTER 1**

# **INTRODUCTION**

CELESTIAL RECORDS is a Database application which is helpful for users of the database or website to surf data. This mini-Project is implemented using HTML and PHP. The project also showcases embedded multimedia and Cascaded Style Sheet.

Operations supported by the application are insert, delete, update and retrieve and search. Admin has rights to insert, update and delete the detail related object in space. A Registered user can view record related to object such as stars, planet or other celestial objects etc. He can also search on the basis of planet name.

In the following sections, a brief introduction about the tools, languages and the databases used to develop the project are discussed.

### **1.1 HTML**

HTML, which stands for Hyper Text Mark-Up Language, is the language for describing structured documents as well as the language used to create web pages in the Internet. The language is based on an existing, international formatting standard SGML, Standard Generalized Mark-Up Language, which is used for text processing.

HTML documents are nothing but web pages which contains HTML tags and plain text. The purpose of a web browser is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

#### **History**

HTML, which stands for Hyper Text Mark-Up Language, is the language for describing structured documents as well as the language used to create web pages in the Internet. The language is based on an existing, international formatting standard SGML, Standard Generalized Mark-Up Language, which is used for text processing. HTML is a simplified version of SGML.

## Tools in order to use HTML

Tools help us in process of creating HTML document. Some are as follows

- **TEXT EDITOR:** To create the HTML code we require a text editor or a word processor. Such as, Notepad, WordPad. We are using notepad++ in developing this project.
- **WEB BROWSER:** The code created by an editor should be executed. This operation can be performed with help of a web browser. Such as Internet Explorer, Netscape navigator, Mozilla Firefox etc.
- **GRAPHICS SOFTWARE:** To include picture we require a graphic software like Adobe Photoshop.
- **Web server:** To make the document is to be available on the internet then, we will have to host it on a web server.

### **1.1.1 Significant Language Features**

HTML files are written in ACSII text, so the user can use any text editor to create his/her web page, though a browser of one sort or another is necessary to view the web page. HTML is case insensitive with its language commands. The characters within the document, however, are case sensitive. The language consists of various "tags" which are known as elements. These allow the browser to understand (and put into the desired/specified format) the layout, background, headings, titles, lists, text and/or graphics on the page. The elements are classified according to their function in the HTML document. There are head elements and body elements. The head elements identify properties of the entire document, while body elements actually mark text as content and show a change in the appearance in one way or another. Most elements have a beginning and an ending which encompass the text the user wishes to mark with the tag. All HTML documents must begin with the element and end with the element. Some of the other elements which may be used are tags to create lists--both ordered lists as well as unordered lists. The user may also create larger or smaller, bolder, italicized, or underlined text. Attributes may be used along with the elements. These perform functions such as placement of text, indication of the source files of images, and identification of links to the document or part of the document.



### 1.1.2 HTML Code

Copy and paste the following HTML code into your newly open text file. Which just displays hello world..

```
<html>
<header><title>This is title</title></header>
<body>
  This is sample text...
  <!-- We use this syntax to write comments -->
  <!-- Page content and rest of the tags here.... -->
  <!-- This is the actual area that gets shown in the browser →
Hello world
</body>
</html>
```

### 1.1.3 HTML TAGS

HTML tags are keywords surrounded by angle brackets like <html>. These are in pair format such that every first tag in pair is start tag where as second tag is end tag. These start and end tags are also called as opening tags and closing tags respectively.

#### Tags Used In Project

The HTML tags are the basis, in order to do this Project. By using some of the important and basically taught tags are used in this Project. Here are some of the tags used in making the Project called AUTOMOBILE MANAGEMENT SYSTEM.

#### HTML Attributes

Attributes provide additional information about HTML elements.

- HTML elements can have **attributes**
- Attributes provide **additional information** about an element

- Attributes are always specified in **the start tag**
- Attributes come in name/value pairs like: **name="value"**

Some basic text formatting HTML tags are listed:

Tag	Description
<html>	Defines an HTML document
<body>	Defines the document's body
<h1> to <h6>	Defines header 1 to header 6
<p>	Defines a paragraph
 	Inserts a single line break
<b>	Defines bold text
<!-->	Defines a comment
<small>	Defines small text

Some of the HTML tags used to create a table are listed:

In an HTML file we can create tables with the Table tags, which in turn will render the browser to display the table in the web page.

Tag	Description
<table>	Defines a table
<th>	Defines a table header
<tr>	Defines a table row
<td>	Defines a table cell
<tbody>	Defines a table body
<tfoot>	Defines a table footer

## A Simple Form

A form in a web page allows the users to input various data online. In an HTML document; forms can be created with the Form tags. In the following table, some basic Form tags are listed:

Tag	Description
-----	-------------

<form>	Defines a form for user input
<input>	Defines an input field
<textarea>	Defines a text-area
<label>	Defines a label to a control
<fieldset>	Defines a fieldset
<legend>	Defines a caption for a fieldset
<select>	Defines a selectable list
<optgroup>	Defines an option group
<option>	Defines an option in the drop box
<button>	Defines a push button

## Image Tags

In an HTML document we can insert and display images by using the image tags.

In the following table, some basic Image tags are listed:

Tag	Description
<IMG SRC attributes>	Defines an image

The “src” attribute is used to display an image on a web page. “src” stands for “source”, and its value is the url of the image to be displayed on the page. The url indicates the location where the image is stored. Attributes may be height, width, align so on.

## Background colour

Using bgcolor attribute this can be done. This is body tag attribute. Six digit hexadecimal code represent the colours.

Syntax: <body text=”text\_color” bgcolor = ”background\_color”>

## **Anchor tag**

Anchor tag is used to link two or more different web pages.

Ex: `<a href="next page to be opened">click here</a>` where href stands for hyper link reference.

## **Areas of Application**

HTML only has one area of application at this time and that is the development of web pages. However, not all browsers support all the tags in all versions of HTML. Because of this, it is wise not to design your web page for a specific browser, because what may look fantastic on your browser has no guarantee of looking great on someone else's browser.

## **1.2 PHP**

PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere.

PHP originally stood for Personal Home Page, but it now stands for the recursive backronym PHP. Hypertext Pre-processor. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks.

## **PHP developer**

PHP developers develop programs, applications, and web sites using the dynamic scripting language PHP. PHP is known for web development and business applications. Depending on job function, PHP developers may be classified as software developers or web developers.

## **Tags Description**

`<?php` to open PHP section

`?>` to close PHP sections

`ECHO` prints the lines

## 1.3 DATABASE

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images. **Database** software systems are programmed in SQL, and examples include Microsoft SQL Server, MySQL, Oracle SAP HANA and FoxPro.

A DBMS system is also required to protect the integrity of data and provide its security. A database management system (**DBMS**) is system software for creating and managing databases. The **DBMS** provides users and programmers with a systematic way to create, retrieve, update and manage data.

## 1.4 MYSQL

MySql is a powerful database. It's very good and free of charge. Many developers in the world selected mysql and php for developing their website.

The MySQL database has become the world's most popular open source database because of its consistent fast performance, high reliability and ease of use. It's used in more than 6 million installations ranging from large corporations to specialized embedded applications on every continent in the world. (Yes, even Antarctica!)

Not only is MySQL the world's most popular open source database, it's also become the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python.) MySQL runs on more than 20 platforms including Linux, Windows, OS/X, HP-UX, AIX, Netware, giving you the kind of flexibility that puts you in control.

Whether you're new to database technology or an experienced developer or DBA, MySQL offers a comprehensive range of certified software, support, training and consulting to make you successful.

## **1.5 WAMP**

The acronym WAMP refers to a set of free (open source) applications, combined with Microsoft Windows, which are commonly used in Web server environments. The WAMP stack provides developers with the four key elements of a Web server: an operating system, database, Web server and Web scripting software. The combined usage of these programs is called a server stack. In this stack, Microsoft Windows is the operating system (OS), Apache is the Web server, MySQL handles the database components, while PHP, Python, or PERL represents the dynamic scripting languages.

## **1.6 Notepad++**

Notepad++ is a text editor and source code editor for use with Microsoft Windows. Unlike Microsoft Notepad, the built-in Windows text editor, it supports tabbed editing, which allows working with multiple open files in a single window. The project's name comes from the C increment operator.

Notepad++ is distributed as free software. At first the project was hosted on SourceForge.net, from where it has been downloaded over 28 million times.

## **1.7 Web Browser**

Google Chrome is a free web browser from Google which we are using here. With its clean design and advanced features, Chrome has quickly become one of the most popular web browsers worldwide. In this lesson, we'll talk about the features of Google Chrome, how

to download and install Chrome to your computer, and how to sign in to Chrome using a Google account.

## **CHAPTER 2**

### **System Analysis and Design**

In this chapter, a complete description of the project development is discussed. The requirements of the project identified are showcased. The database design is done Using High-Level Conceptual Data Models

#### **2.1 Requirement Analysis**

Following requirements were identified during the requirement collection and analysis.

1. Record related to extraterrestrial objects is organized for lookup and analysis.
2. Admin will have unique id.
3. Each user will have unique id and password.
4. User can look up into various table such as celestial table, scientist table etc.
5. As there are many extraterrestrial objects so we have given them unique id.
6. The database will keep track of user login time.
7. New data can be only updated by admin only.

#### **Feasibility Study**

The feasibility study carried out showed that the requirements that were to be included could be provided by the use of RDBMS software such as MySQL which is available as an open source and for the front end HTML pages with processing capability provided by the Scripting language such as PHP and Javascript.



## 2.2 ER-Diagram

Following is the conceptual representation of the requirements identified as an ER-Diagram

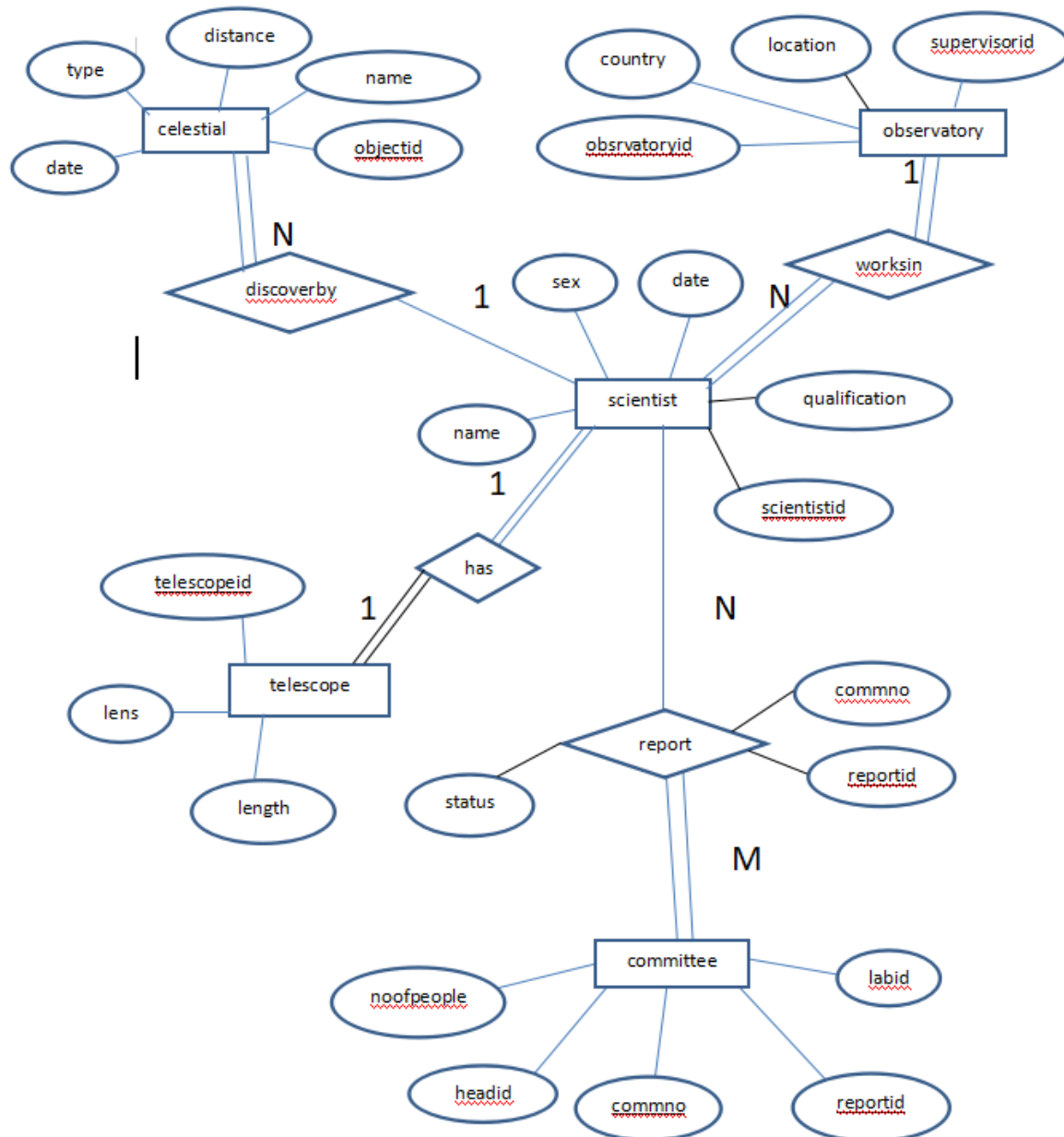
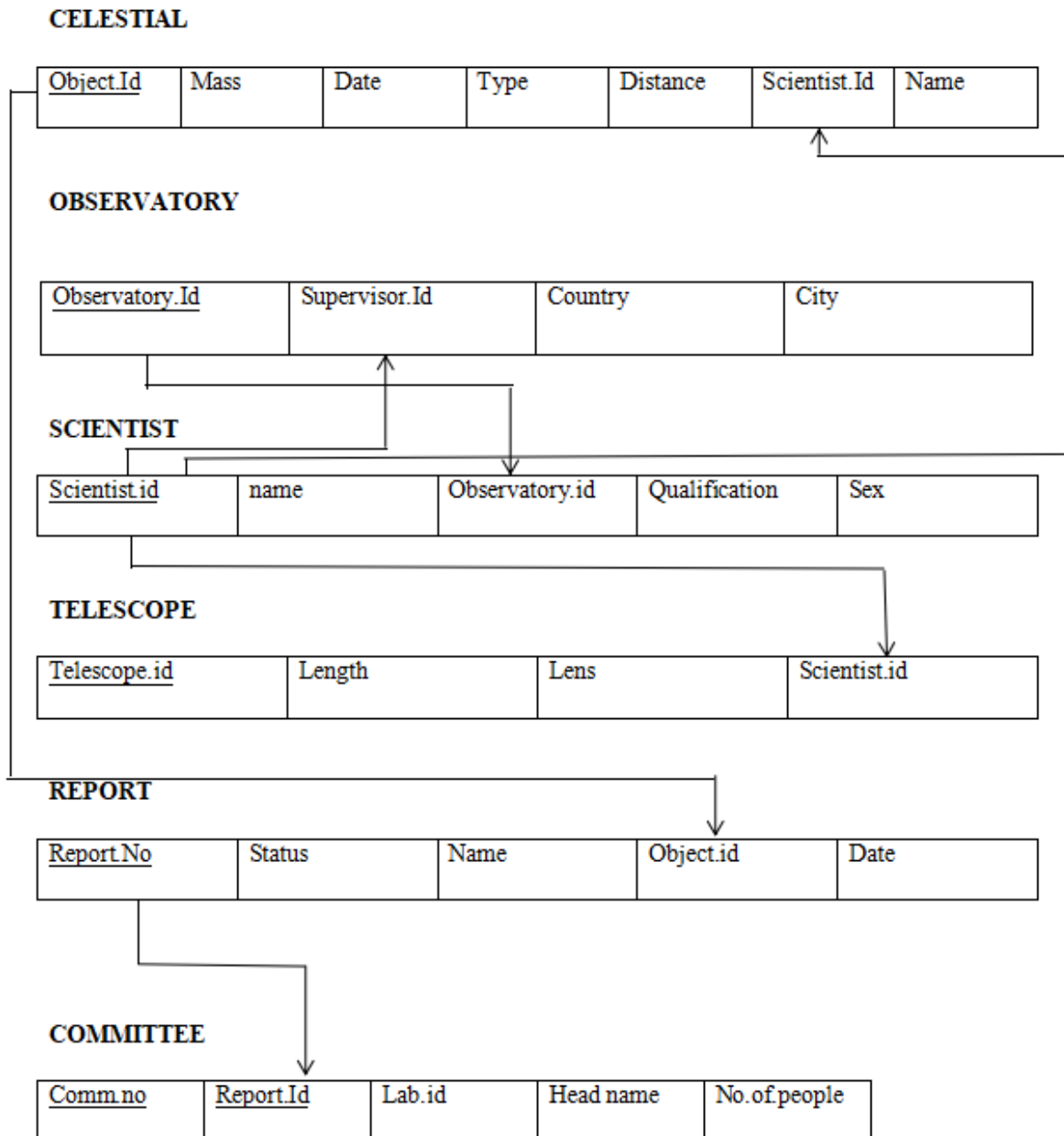


Figure 2.1 ER Diagram for Celestial records

## 2.3 Relational Schema

The relational schema diagram has been derived from the ER-Diagram in Figure 2.1 using the ER-Relational mapping algorithm



**Figure 2.2 Relational Schema Diagram Showing the Primary key and Foreign key relationships**

## 2.4 Functional Requirements

Functional requirements of a software project interpret the function of a part. It defines its functions, input and output. The typical functional requirements include:

Application contains 2 modules:

- Admin module
- User module

Admin module

- Admin can delete information of planet, stars, comets etc.
- Admin can view information. of planet, stars, comets etc.
- Admin can insert information. of planet, stars, comets etc.
- Admin can update information. of planet, stars, comets etc.

User module

- User can able to view the details of planet, stars, comets etc.
- User can Search details of planet, stars, comets etc.

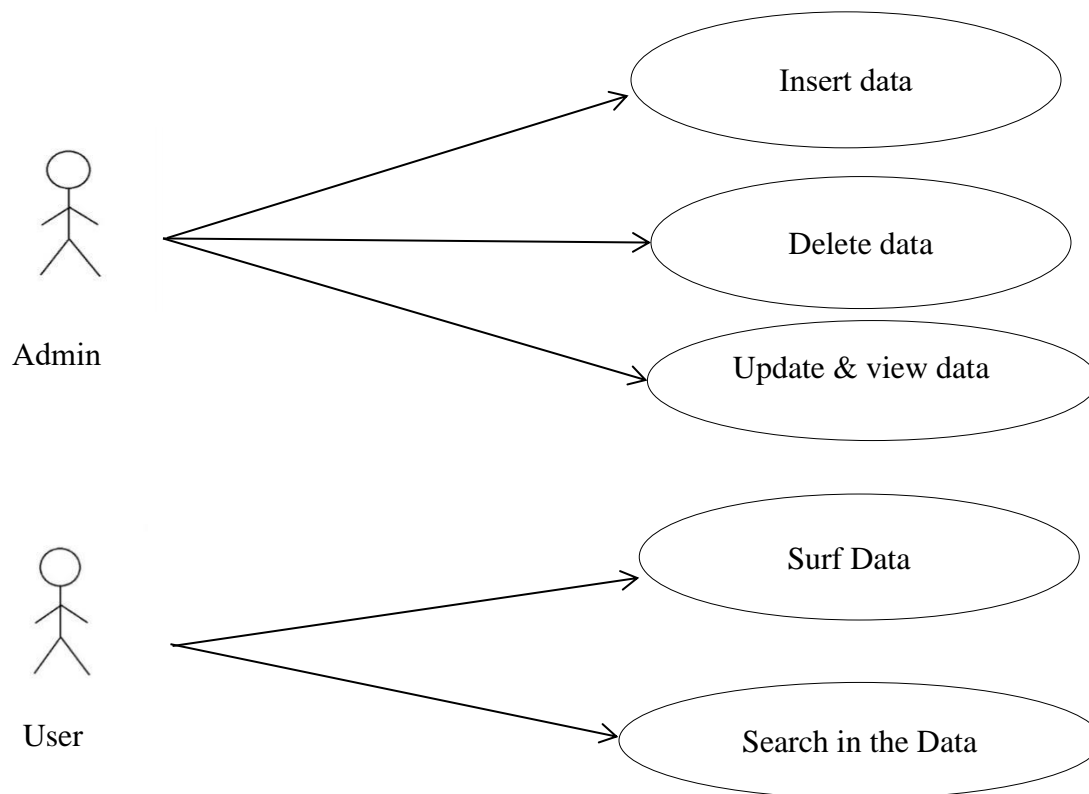
## 2.5 Non- Functional Requirements

A non-functional requirement specifies the canon of the articular process not the particular judgment of the system and particular behavior of the process. Non-functional requirements define how the system works.

- This is made so that user can find all relevant data at one go and need not to search all web.
- This web application work efficiently as it works on all logical paths and independently .
- .This application is available during all the time user can use it whenever it want .
- This application reduces all the complexity during searching and surfing.
- To run this application efficiently web browser is important.
- The is important for student doing project on planets,moon etc.

## 2.6 Use Case Diagram

The use case diagrams usually refer to behavioral diagrams helps people to understand the interaction between user and system. Use case diagram identify different users of the system. It is used to define some set of actions, which is called as use cases. Actors are the result of some valuable use cases. Use case figures are also called as unified modeling language.



**Figure 2.4 Use Case Diagram**

## CHAPTER 3

### SYSTEM IMPLEMENTATION

This chapter gives brief introduction about project implementation first the database design, database connectivity is discussed and is followed by the implementation of database operation.

#### 3.1 Database Design

##### COMMITTEE

```
CREATE TABLE `committee` (  
  `commno` varchar(10) NOT NULL,  
  `reportid` varchar(10) NOT NULL,  
  `labid` varchar(10) NOT NULL,  
  `headname` text NOT NULL,  
  `noofpeople` int(11) NOT NULL,  
  PRIMARY KEY (`commno`,`reportid`),  
  KEY `2fkk` (`reportid`),  
  CONSTRAINT `2fkk` FOREIGN KEY (`reportid`) REFERENCES `report` (`reportno`) ON  
  DELETE CASCADE ON UPDATE CASCADE  
)
```

##### OBSERVATORY

```
CREATE TABLE `observatory` (  
  `observatoryid` varchar(10) NOT NULL,  
  `supervisor` varchar(20) NOT NULL,  
  `country` varchar(20) NOT NULL,  
  `city` varchar(20) NOT NULL,  
  PRIMARY KEY (`observatoryid`),
```

```
KEY `superfk` (`supervisor`),  
CONSTRAINT `superfk` FOREIGN KEY (`supervisor`) REFERENCES `scientist` (`id`) ON  
DELETE CASCADE ON UPDATE CASCADE  
)
```

## **REPORT**

```
CREATE TABLE `report` (  
  `reportno` varchar(10) NOT NULL,  
  `status` varchar(10) NOT NULL,  
  `name` varchar(20) NOT NULL,  
  `objectid` varchar(10) NOT NULL,  
  `date` date NOT NULL,  
  PRIMARY KEY (`reportno`),  
  KEY `celfk` (`objectid`),  
  CONSTRAINT `celfk` FOREIGN KEY (`objectid`) REFERENCES `celestial` (`objectid`) ON  
  DELETE CASCADE ON UPDATE CASCADE  
)
```

## **SCIENTIST**

```
CREATE TABLE `scientist` (  
  `id` varchar(10) NOT NULL,  
  `name` varchar(20) NOT NULL,  
  `observatoryid` varchar(10) NOT NULL,  
  `qualification` varchar(20) NOT NULL,  
  `sex` varchar(2) NOT NULL,  
  PRIMARY KEY (`id`),
```

```
KEY `obsfk` (`observatoryid`),  
CONSTRAINT `obsfk` FOREIGN KEY (`observatoryid`) REFERENCES `observatory`  
(`observatoryid`) ON DELETE CASCADE ON UPDATE CASCADE  
)
```

## TELESCOPE

```
CREATE TABLE `telescope` (  
  `telid` varchar(10) NOT NULL,  
  `length` int(11) NOT NULL,  
  `lens` int(11) NOT NULL,  
  `scientistid` varchar(10) NOT NULL,  
  PRIMARY KEY (`telid`),  
  KEY `sciifk` (`scientistid`),  
  CONSTRAINT `sciifk` FOREIGN KEY (`scientistid`) REFERENCES `scientist` (`id`) ON  
  DELETE CASCADE ON UPDATE CASCADE)
```

## 3.2 Database Connectivity

```
<?php  
$database="celestial";  
$servername="localhost";  
$username="root";  
$password="";  
$connect = mysqli_connect($servername,$username,"",$database);  
?>
```

### 3.3 Implementation of Database Operations

#### INSERT

```
<?php
$connect = mysqli_connect("localhost", "root","", "celestial");
// php code to search data in mysql database and set it in input text

if(isset($_POST['Insert']))
{
    // id to search
    $name = $_POST['name'];
    $objectid = $_POST['objectid'];
    $mass = $_POST['mass'];
    $date = $_POST['date'];
    $type = $_POST['type'];
    $scientistid = $_POST['scientistid'];

    $insert ="insert into celestial (name,objectid,mass,date,type,scientistid) values
(' $name',' $objectid',' $mass',' $date',' $type',' $scientistid)";

    $sel="SELECT name, objectid, mass,date,type,scientistid FROM celestial";
    $result2=mysqli_query($connect,$insert);
    $result1=mysqli_query($connect,$sel);
    if($result2)
    {
        echo 'Data Inserted';
    }else{
        echo 'Data Not Inserted';
    }
    if($result1->num_rows > 0)
```



```
{

while($row = $result1->fetch_assoc()) {

echo "<tr><td>" . $row["name"]. "</td><td>" . $row["objectid"] . "</td><td>"
. $row["mass"]. "</td><td>" . $row["date"] . "</td><td>" . $row["type"] . "</td><td>" .
$row["scientistid"] . "</td></tr>";
}} else {
    echo "0 results";
}

mysqli_close($connect);

}

?>
```

## VIEW

```
<?php

$conn = mysqli_connect("localhost", "root", "", "celestial");

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT name,objectid,mass,date,type,scientistid FROM celestial";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "<tr><td>" . $row["name"]. "</td><td>" . $row["objectid"] . "</td><td>"
```

```
. $row["mass"]. "</td><td>" . $row["date"] . "</td><td>" . $row["type"] . "</td><td>" .  
$row["scientistid"] . "</td></tr>";  
}  
echo "</table>";  
} else { echo "0 results"; }  
$conn->close();  
?>
```

## DELETE

```
<?php  
$connect = mysqli_connect("localhost", "root", "", "celestial");  
// php code to search data in mysql database and set it in input text  
  
if(isset($_POST['Delete']))  
{  
  
    $objectid = $_POST['objectid'];  
    $planetname = $_POST['planetname'];  
    $mass = $_POST['mass'];  
    $scientistid = $_POST['scientistid'];  
    $up = "delete from `celestial`  
    where `objectid` = '$objectid'";  
    $sel="SELECT objectid, name, mass,date,type,scientistid FROM celestial";  
    $result=mysqli_query($connect,$up);  
  
    $result1=mysqli_query($connect,$sel);  
    if($result)  
    {  
        echo 'Data Updated';  
    }  
}
```

```
}else{
    echo 'Data Not Updated';
}
if($result1->num_rows > 0)
{
    // output data of each row
    while($row = $result1->fetch_assoc()) {
echo "<tr><td>" . $row["objectid"] . "</td><td>" . $row["name"] . "</td><td>" . $row["mass"] .
"</td><td>" . $row["date"] . "</td><td>" . $row["type"] . "</td><td>" . $row["scientistid"] .
"</td></tr>";
    } } else {
        echo "0 results";
    }
    mysqli_close($connect);
}
```

## UPDATE

```
<?php
$connect = mysqli_connect("localhost", "root", "", "celestial");
// php code to search data in mysql database and set it in input text

if(isset($_POST['update']))
{
    // id to search

$name = $_POST['name'];
$mass = $_POST['mass'];
$date = $_POST['date'];
$type = $_POST['type'];
```

```
$scientistid = $_POST['scientistid'];

// connect to mysql

// mysql search query
$up = "UPDATE `celestial` SET `name` = '$name', `mass` = '$mass', `date` = '$date', `type` =
'$type', `scientistid` = '$scientistid' where `objectid` = '$objectid'";
$sel="SELECT objectid,name, mass, date,type,scientistid FROM celestial";
$result=mysqli_query($connect,$up);

$result1=mysqli_query($connect,$sel);
if($result)
{
    echo 'Data Updated';
}else{
    echo 'Data Not Updated';
}

if($result1->num_rows > 0)
{
    // output data of each row
    while($row = $result1->fetch_assoc()) {
echo "<tr><td>" . $row["objectid"] . "</td><td>" . $row["name"] . "</td><td>" . $row["mass"] .
"</td><td>" . $row["date"] . "</td><td>" . $row["type"] . "</td><td>" . $row["scientistid"] .
"</td></tr>" ;
    } } else {
    echo "0 results";
}

mysqli_close($connect);
```

```
// in the first time inputs are empty
```

```
}
```

```
?>
```

## INSERT

```
<?php
```

```
$connect = mysqli_connect("localhost", "root", "", "celestial");
```

```
// php code to search data in mysql database and set it in input text
```

```
if(isset($_POST['Insert']))
```

```
{
```

```
    // id to search
```

```
    $name = $_POST['name'];
```

```
    $objectid = $_POST['objectid'];
```

```
    $mass = $_POST['mass'];
```

```
    $date = $_POST['date'];
```

```
    $type = $_POST['type'];
```

```
    $scientistid = $_POST['scientistid'];
```

```
    // connect to mysql
```

```
    // mysql search query
```

```
    $insert = "insert into celestial (name,objectid,mass,date,type,scientistid) values  
('".$name."','".$objectid."','".$mass."','".$date."','".$type."','".$scientistid")";
```

```
    $sel = "SELECT name, objectid, mass,date,type,scientistid FROM celestial";
```

```
    $result2 = mysqli_query($connect,$insert);
```

```
    $result1 = mysqli_query($connect,$sel);
```

```
if($result2)
{
    echo 'Data Inserted';
}else{
    echo 'Data Not Inserted';
}
if($result1->num_rows > 0)
{
    while($row = $result1->fetch_assoc()) {

echo "<tr><td>" . $row["name"]. "</td><td>" . $row["objectid"] . "</td><td>"
. $row["mass"]. "</td><td>" . $row["date"] . "</td><td>" . $row["type"] . "</td><td>" .
$row["scientistid"] . "</td></tr>";
    } } else {
        echo "0 results";
    }
    mysqli_close($connect);
}
?>
```

### 3.4 Trigger

Following trigger have been implemented in the project.

#### TRIGGER

```
CREATE TRIGGER `usersignup`
AFTER INSERT ON `users`
FOR EACH ROW INSERT INTO usersignups (firstname,`email`,time) VALUES
(NEW.firstname,NEW.email,CURRENT_TIMESTAMP)
```

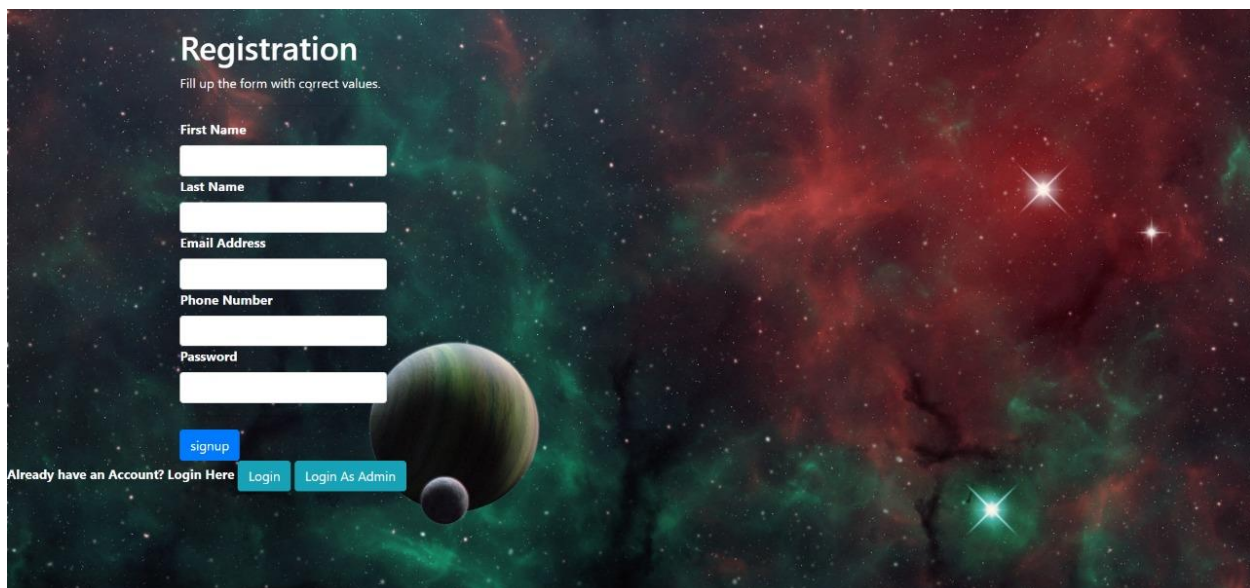
#### TRIGGER

```
CREATE TRIGGER `afterlogin`  
AFTER INSERT ON `login`  
FOR EACH ROW INSERT INTO userlog (`email`,time)  
VALUES (NEW.email,CURRENT_TIMESTAMP)
```

## CHAPTER 4

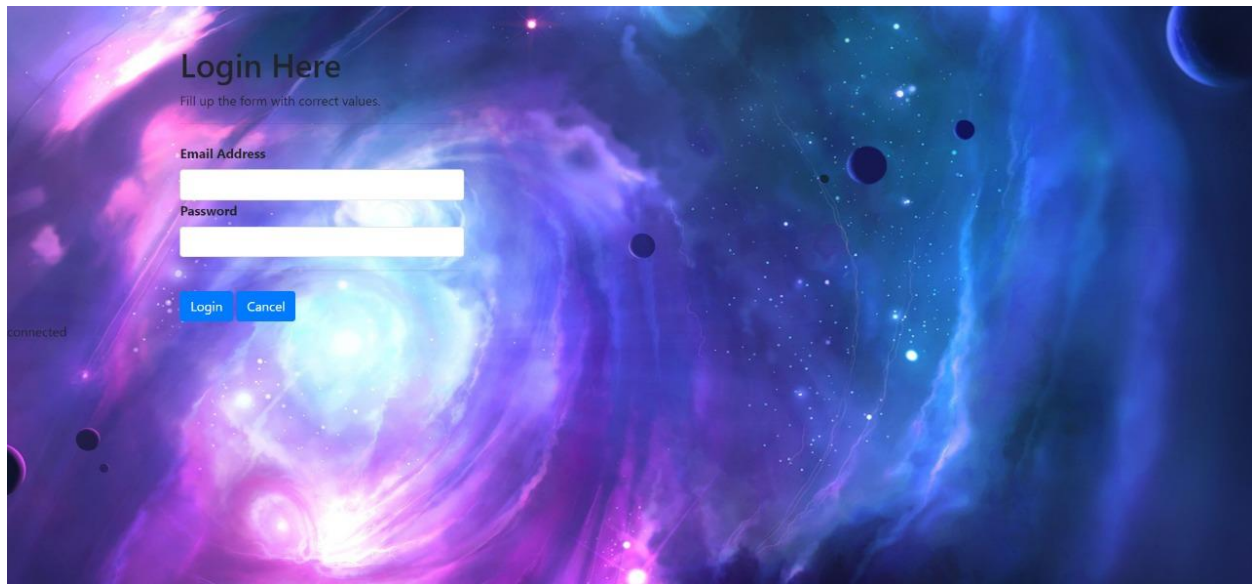
### Results and Discussion

In this chapter the results of the project are discussed. The snapshot of the project showing various functionalities like insert, delete, update and retrieval are showcased.



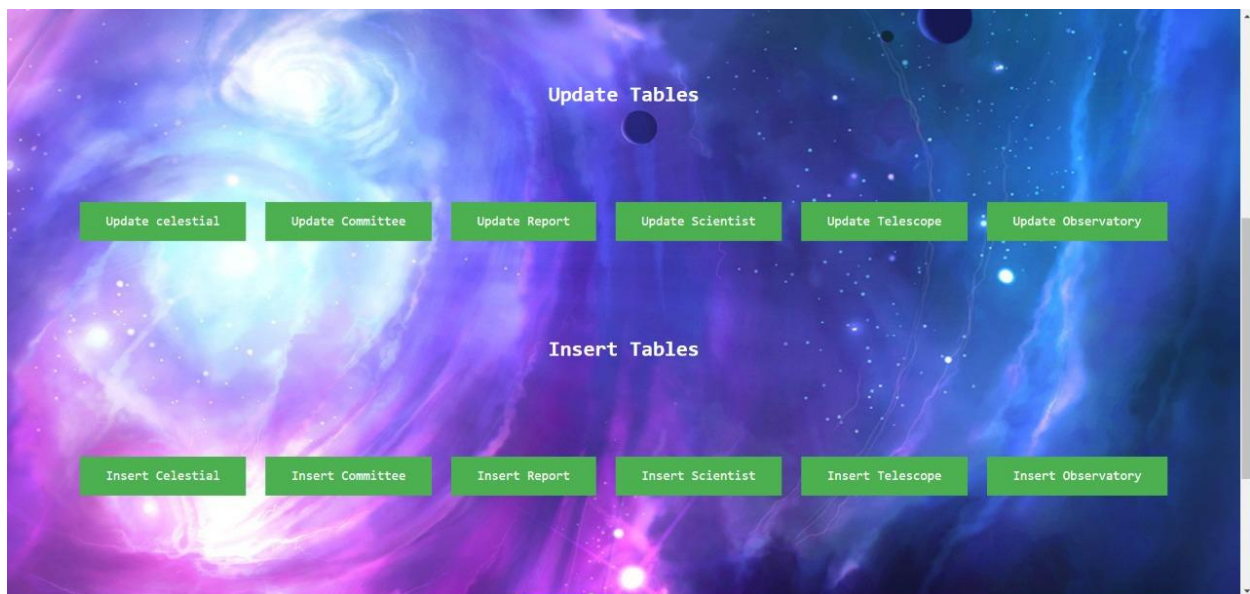
**Fig. 4.1 Welcome page of project**

Fig 4.1 shows welcome page of the project . Here user can register him\her by entering firstname,lastname,email,phonenumber and password. Here we also have two buttons login and Login as Admin.log in redirect us to login page for user and Login as Admin redirect us to login page for admin.



**Fig. 4.2 Login page**

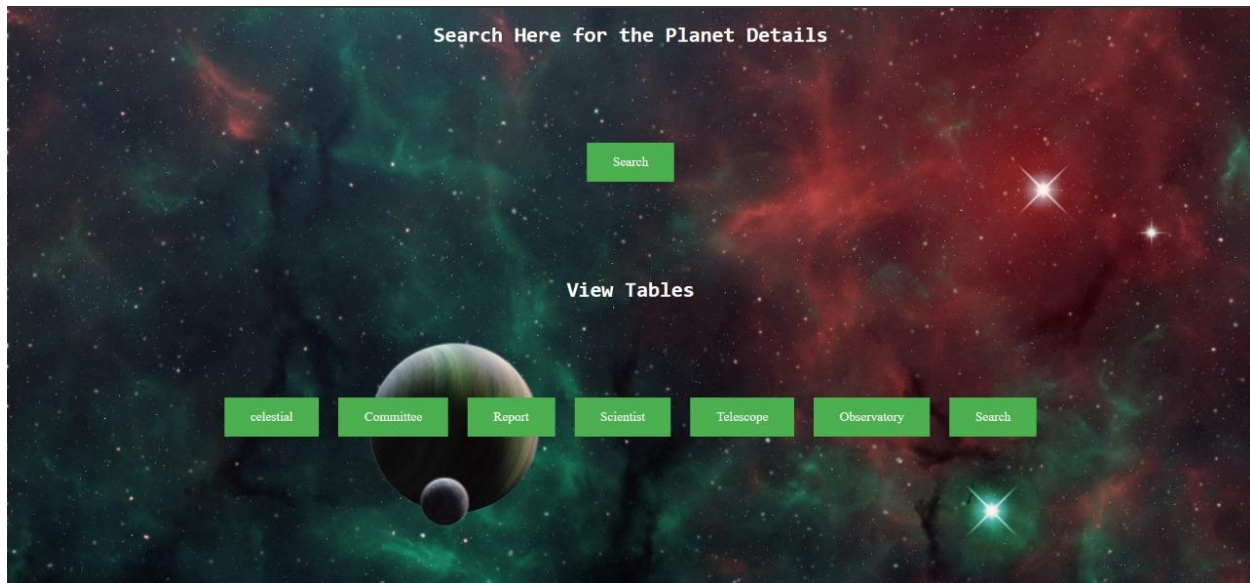
Login page takes two parameter email and password . After user enter details he/she can click Login if data matches then it goes to fig 4.4 view and search page if not it goes back to welcome page



**Fig. 4.3 Update and Insert page**



Update and insert page can only be accessed by admin update redirects us to update page  
Where as insert redirects us to insert page



**Fig. 4.4 View and Search Page**

View and Search Page can be used to view and search data. View button shows us the whole table. search button search.php

Name	Objectid	Mass	Date	Type	Scientistid
------	----------	------	------	------	-------------

**Fig. 4.5 Insert page**

Insert page can only accessed by admin and is used to insert new data into table.

**Celestial Info**

Name

Name	Objectid	Mass	Date	Type	Scientistid
------	----------	------	------	------	-------------

**Fig. 4.6 Search page**

Search page allows user to search planets, comet, stars etc . It is needed so that user does not need to find names manually.

**Committee**

Commno..

Reportid

Labid

Headname

Noofpeople

Commno	Reportid	Labid	Headname	Noofpeople
--------	----------	-------	----------	------------

**Fig. 4.7 update page**

update page allows admin to update existing value we can used drop down to select a row as it make interaction easier.

## **CHAPTER 5**

### **CONCLUSION**

Celestial record database management helps us in centralizing the data used for managing the record for scientific and knowledge use. The theoretical process involved in database design has been practically implemented. The project provides a user-friendly interface for the users to interact with the database. All database operations including insertion, deletion, updation and Retrievals are supported along with support for trigger and stored procedure.

## **REFERENCES**

[1] Database systems Models, Languages, Design and Application Programming, RamezElmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.

[2] [w3schools.com](http://w3schools.com)

[3][tutorialpoint.com](http://tutorialpoint.com)