

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“Jnana Sangama”, Belagavi-590018



DBMS LABORATORY WITH MINI PROJECT
(17CSL58)
REPORT ON
“RESORT MANAGEMENT SYSTEM”

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING
ON
COMPUTER SCIENCE AND ENGINEERING

Submitted by

PARTH P SHAH
SHASHANK SHET K

1KS17CS052
1KS17CS075

Under The Guidance Of

Mr. CHILLARA ANIL KUMAR
Assistant Professor

Ms. RANJITHA KN
Assistant Professor

Department of Computer Science & Engineering
K.S.I.T.



KSIT
K S INSTITUTE OF TECHNOLOGY

Department of Computer Science & Engineering
K.S. INSTITUTE OF TECHNOLOGY
#14, Raghuvanahalli, Kanakapura Road, Bengaluru- 560109
2019-2020

**K.S. INSTITUTE OF TECHNOLOGY
BENGALURU -560109**
Department of Computer Science & Engineering



CERTIFICATE

This is to certify that the **DBMS LABORATORY WITH MINI PROJECT** entitled “**RESORT MANAGEMENT SYSTEM**” presented by **Mr. PARTH P SHAH, USN: 1KS17CS052** of **V semester** in partial fulfillment of the award of **Bachelor of Engineering in Computer Science & Engineering** in **Visvesvaraya Technological University, Belagavi** during the academic year **2019-2020**. The **DBMS MINI PROJECT** has been approved as it satisfies the academic requirements in respect of **DBMS LABORATORY WITH MINI PROJECT (17CSL58)** prescribed for the Bachelor of Engineering degree.

Mr. Chillara Anil Kumar
Assistant Professor, CSE
K.S.I.T., Bengaluru

Mrs. Ranjitha KN
Assistant Professor, CSE
K.S.I.T., Bengaluru

Dr. Rekha .B .Venkatapur
Prof. & Head, CSE
K.S.I.T., Bengaluru

Dr.T.V.Govindaraju
Principal / Director
K.S.I.T., Bengaluru

Signature of Student: _____

Signature of Internal Examiner:

Signature of External Examiner:

ACKNOWLEDGEMENT

The successful presentation of the **DBMS LABORATORY WITH MINI PROJECT (17CSL58)** would be incomplete without the mention of the people who made it possible and whose constant guidance crowned my effort with success.

I would like to extend my gratitude to the **MANAGEMENT, KAMMAVARI SANGHAM**, Bengaluru, for providing all the facilities to present the DBMS LABORATORY WITH MINI PROJECT.

I would like to extend my gratitude to **Dr. T.V. GOVINDARAJU**, the Principal/ Director, K.S. Institute of Technology, Bengaluru, for facilitating me to present the DBMS LABORATORY WITH MINI PROJECT.

I thank **Dr. REKHA B VENKATAPUR**, Professor and Head, Department of Computer Science and Engineering, K.S. Institute of Technology, Bengaluru, for her encouragement.

I would like to thank our Project Guide's, **MR. CHILLARA ANIL KUMAR**, Assistant Professor, and **Ms. RANJITHA K N**, Assistant Professor, Department of Computer Science and Engineering, K.S. Institute of Technology, Bengaluru, for their constant guidance and inputs.

I would like to thank all the **Teaching** Staff and **Non-Teaching** Staff of the college for their co-operation.

Finally, I extend my heart-felt gratitude to my **family** for their encouragement and support without which I would not have come so far. Moreover, I thank all my **friends** for their invaluable support and cooperation.

PARTH P SHAH

ABSTRACT

The system aims at the maintenance and management of the different accommodations of rooms in the resort that are available. It mainly takes care of the resort management at the core area of the database.

The system provides the information regarding the different rooms that are available and their status specific to availability.

The database also manages the atomic information regarding the different units that are available under one resort. Each unit is well furnished and is well designed for the basic flexibility of the tourists who are expected to stay.

This resort management system provides proper management of data and transactions in a centralized and organized manner and also provides a user friendly interface with which the user can interact easily with the just little or elementary knowledge of operating computers.

This project accomplished the task of building a system that ensures accurate record maintenance which was done through proper identification of customers and the proper designation of user functions . A resort management information system is required to assist management of data in the hospitality industry and also to make the entire resort management process easier.

CONTENTS

PARTICULARS	PAGE NO
ACKNOWLEDGEMENT	I
ABSTRACT	II
1. Chapter 1: Introduction	7
1.1. Introduction	7
1.1.1. Purpose of the project	7
1.1.2. Scope	7
1.2. Connection	8
1.2.1. MySQL	8
1.2.2. Apache	8
1.2.3. XAMPP	8
2. Chapter 2: Requirement Specifications	9
2.1. Software Requirements Specifications	9
2.2. Operating Environment	9
2.2.1. Hardware Requirements	9
2.2.2. Software Requirements	9
2.3. Functional Requirements	10
2.4. Non functional requirements	10
2.5. Advantages of Airline Reservation System	11
2.6. Summary	11
3. Chapter 3: Design	12
3.1. System Design	12
3.2. ER diagram	13
3.3. Relational Schema	14
3.4. Description of tables	15

4. Chapter 4: Implementation	18
4.1. Implementation	18
4.2. Programming Language Selection	18
4.3. Snapshot	19
4.4. Procedures	23
4.5. Triggers	23
5. Chapter 5: Conclusion	24
6.1. Conclusion	24
6.2. Limitations of Project	24
6.3. Future Enhancement (if any)	24
6.4. References	24

CHAPTER 1

INTRODUCTION

1.1. Introduction

Considering the volumes of data that needs to be tracked and accessed, it would be very difficult to manage the accuracy and quality of data manually and deliver them accordingly. It would be almost impossible to get the details required in case of manual maintenance of data. The following is an innovative solution that helps in managing room booking orders as they can be assigned subject to their availability. The Resort Reservation System simplifies the manual work and allows smooth administration of these operations.

1.1.1. Purpose

This project is aimed to reduce the manual work involved in data maintenance in the Room Booking and automates the Resort Management System. This project is developed mainly to simplify the manual work and allows smooth administration of the operations of resorts. The purpose of the project is to computerize the administrative operations of a Typical resort and to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of Users, Flights and Booking information, Fare details, etc. Traditionally, it was done manually. The main function of the system is to enter and book Flights and retrieve these details as and when required, and also to manipulate these details meaningfully.

1.1.2. Scope

The project provides a very simple application which simplifies the manual work done by the operation team of Resort Management System. This application saves the data of users in the database. Allows users to search for rooms and request booking. Our project allows admins to view the requested rooms and assign rooms to them respectively in the database and to see the statistics.

1.2. Connection

1.2.1. MY-SQL (BACKEND)

MySQL in July 2013, it was the world's second most widely used RDBMS, and the most widely used open-source client server model RDBMS. It is named after cofounder Michael Widenius's. The SQL abbreviation stands for Structured Query Language. The MySQL development project

has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements.

1.2.2. PHP

The **PHP Hypertext Preprocessor (PHP)** is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server. PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them. PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user. You add, delete, modify elements within your database through PHP. Access cookies variables and set cookies. Using PHP, you can restrict users to access some pages of your website. It can encrypt data.

1.2.3. XAMPP

XAMPP is a small and light Apache distribution containing the most common web development technologies in a single package. Its contents, small size, and portability make it the ideal tool for students developing and testing applications in PHP and MySQL. XAMPP is available as a free download in two specific packages: full and lite. While the full package download provides a wide array of development tools, XAMPP Lite contains the necessary technologies that meet the Ontario Skills Competition standard.

1.2.4. Apache

The Apache HTTP Server is web server software notable for playing a key role in the initial growth of the World Wide Web. In 2009 it became the first web server software to surpass the 100 million web site milestone. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. Since April 1996 Apache has been the most popular HTTP server software in use.

CHAPTER 2

REQUIREMENTS SPECIFICATIONS

2.1. Software Requirements Specifications

The Resort Reservation System operates with client-server architecture, and as such, must have minimum hardware and software to run the server/browser along with all its dependencies.

The system is used by Admin who can assign rooms on request or manipulates data using a computer with a HTML compatible browser. The system is also used by the customers of the Resort who can book rooms based on their preference using a computer at his/her home, with a HTML/php compatible browser.

The server software runs in a dedicated centralized server hosting center for the Resort database. The scripts and http server run on the server, and require a Php interpreter, along with the dependencies for the scripts, as well as the MySQL server.

2.2. Operating Environment

The production ready software is meant to run on a variety of verified hardware and software. As such, many of the required dependencies are available cross platform, both for the front end as well as the backend. Some of the verified software and hardware are specified below, along with software and hardware that are supposed to be compatible.

2.2.1. Hardware Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.

- Processor - Intel 486/Pentium processor or better
- Processor Speed - 500 MHz or above
- Hard Disk - 20GB(approx)
- RAM - 64MB or above
- Storage Space - Approx. 2MB

2.2.2 Software Requirements

- Technology Implemented : Apache Server, MySQL Server
- Language Used : PHP
- Database : My SQL
- User Interface Design : HTML, CSS
- Web Browser : Google Chrome

2.3. Functional Requirements

The functional requirements of this project are:

The project allows login of only Admin:

The Admin has a set of functionalities described as:

- > He can login into the system.
- > He can log out from his current account.
- > He can assign rooms to the customers based on room requests in the database.
- > He can view all the records of all the bookings that have been made.

The Customer has a set of functionalities described as:

- > He can book a room of his choice.
- > He can search his bookings and view it.
- > He can send a message using the 'Contact us' page.

2.4. Non-functional requirements

The non-functional requirements of these projects are:

- **Security:**

The website does not allow access to any functionality by directly jumping to any particular link to that function's page. Additionally, anything that is needed to be done can only be done by first logging in. There are multiple accounts, with login by Admin only allowing Admin functionalities and Customer allowing only Customer functionalities.

- **Data Integrity:**

The project does not allow entry of data in case data is invalid. This is very important as if invalid data is added, then it can cause large problems, such as getting tickets for flights which are invalid, creating aircrafts for invalid or incorrect destinations, or inserting invalid source/destination, each of which can have cascading effects.

- **Automatic data processing:**

A lot of information is processed by the project instead of relying on the user to add perfect information and perform numerous functions each time. Examples include deleting old dates and flights, retrieving prices, validating inserted information in aircrafts etc. This is an important task as it can be performed much more efficiently and quickly by the system than by a human.

2.5. Advantages of Resort Management System

The advantages of the Resort Reservation System are:

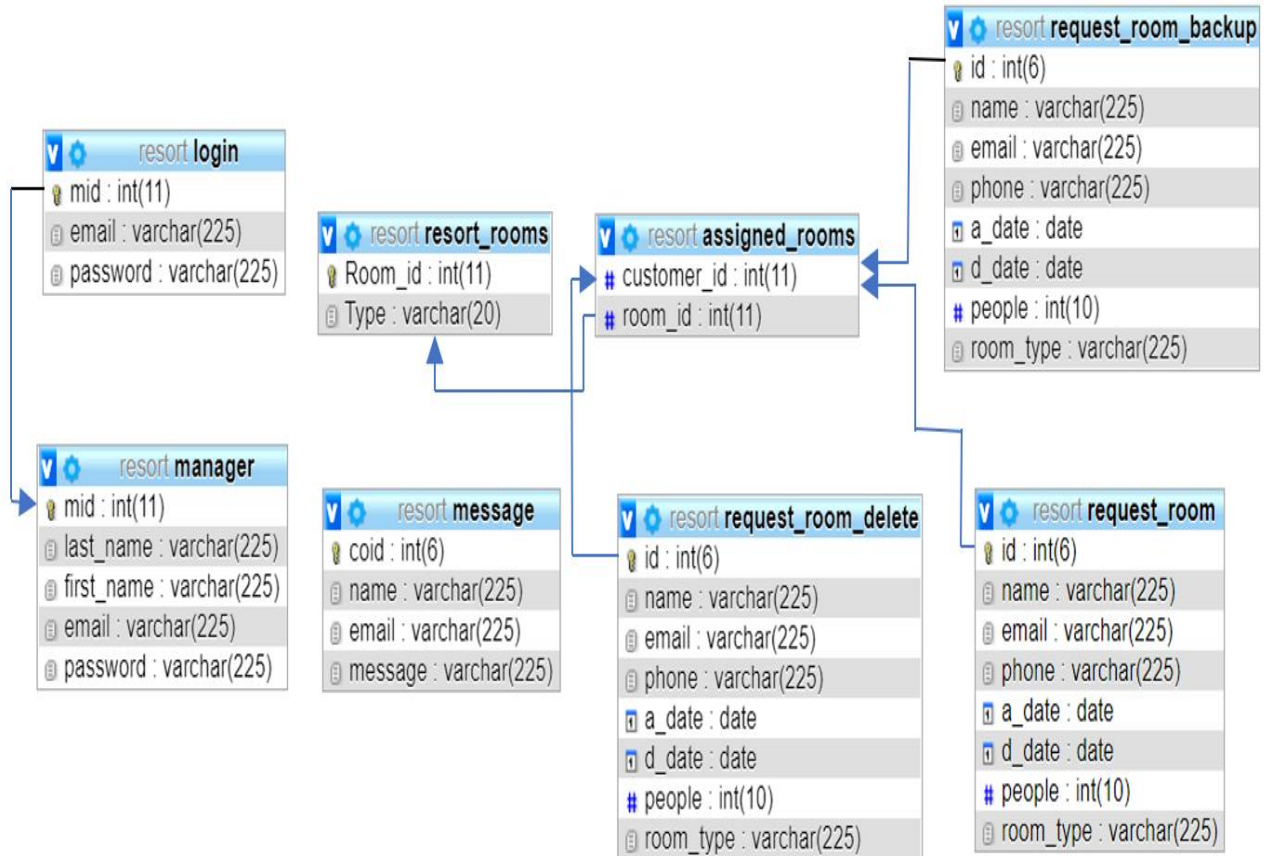
- It is based on a Client-Server System, meaning multiple Admins and customers can be supported.
- It operates on a minimalistic User Interface so that any user of the system can do what they need to do with almost no training and extreme ease.
- Erroneous data is not entered into the system and rendered invalid

2.6. Summary

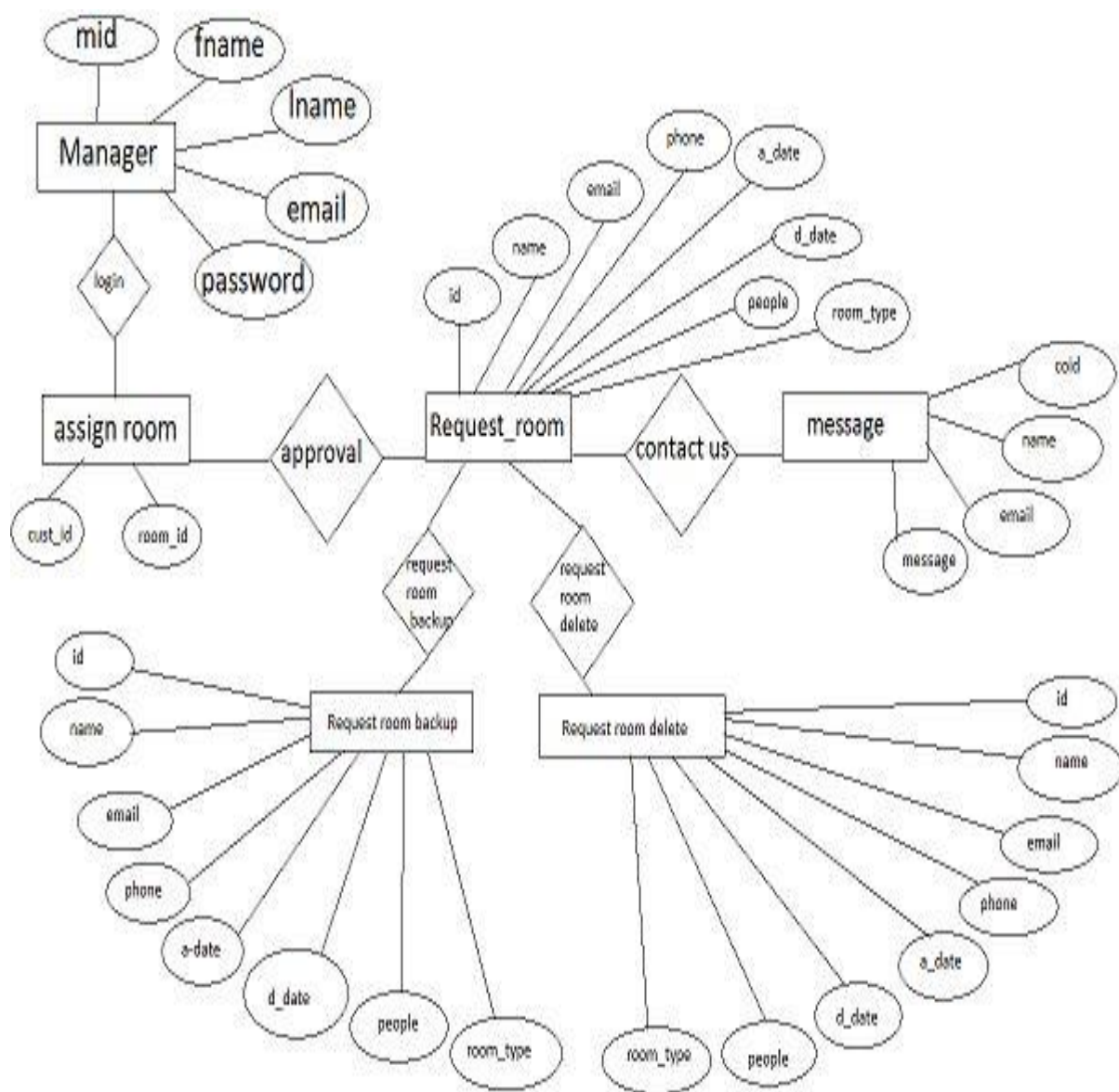
We have developed a highly comprehensive and easy to use system for any resort. It is easy to implement and requires no training to use. It provides options for Admins and customers. It is error-proof and does large amount of work in the background. Thus, the system aids to simplify the processes used by cashier and manager as well as reduce operational costs, the primary concerns of all businesses.

CHAPTER 3 DESIGN

3.1. System Design



3.2. ER Diagram



3.3. Relational Schema

ASSIGNED_ROOMS

<u>CUST_ID</u>	<u>ROOM_ID</u>
----------------	----------------

LOGIN

<u>MID</u>	EMAIL	PASSWORD
------------	-------	----------

MANAGER

<u>MID</u>	LAST_NAME	FIRST_NAME	EMAIL	PASSWORD
------------	-----------	------------	-------	----------

MESSAGE

<u>COID</u>	NAME	EMAIL	MESSAGE
-------------	------	-------	---------

REQUEST_ROOM

<u>ID</u>	NAME	EMAIL	PHONE	A_DATE	D_DATE	PEOPLE	ROOM_TYPE
-----------	------	-------	-------	--------	--------	--------	-----------

REQUEST_ROOM_BACKUP

<u>ID</u>	NAME	EMAIL	PHONE	A_DATE	D_DATE	PEOPLE	ROOM_TYPE
-----------	------	-------	-------	--------	--------	--------	-----------

REQUEST_ROOM_DELETE



<u>ID</u>	NAME	EMAIL	PHONE	A_DATE	D_DATE	PEOPLE	ROOM_TYPE
-----------	------	-------	-------	--------	--------	--------	-----------

RESORT_ROOMS


<u>ROOM_ID</u>	TYPE
----------------	------

3.4. Description of Tables


1. Assigned_rooms table :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	customer_id 	int(11)			No	None		
2	room_id 	int(11)			No	None		


2. Login Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	mid 	int(11)			No	None		AUTO_INCREMENT
2	email	varchar(225)	latin1_swedish_ci		No	None		
3	password	varchar(225)	latin1_swedish_ci		No	None		

3. Manager Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	mid 	int(11)			No	None		AUTO_INCREMENT
2	last_name	varchar(225)	latin1_swedish_ci		No	None		
3	first_name	varchar(225)	latin1_swedish_ci		No	None		
4	email	varchar(225)	latin1_swedish_ci		No	None		
5	password	varchar(225)	latin1_swedish_ci		No	None		

4. Message table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	coid 	int(6)			No	None		AUTO_INCREMENT
2	name	varchar(225)	latin1_swedish_ci		No	None		
3	email	varchar(225)	latin1_swedish_ci		No	None		
4	message	varchar(225)	latin1_swedish_ci		No	None		


5. Request_room Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(6)			No	None		AUTO_INCREMENT
2	name	varchar(225)	latin1_swedish_ci		No	None		
3	email	varchar(225)	latin1_swedish_ci		No	None		
4	phone	varchar(225)	latin1_swedish_ci		No	None		
5	a_date	date			No	None		
6	d_date	date			No	None		
7	people	int(10)			No	None		
8	room_type	varchar(225)	latin1_swedish_ci		No	None		

6. Request_room_backup Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(6)			No	None		AUTO_INCREMENT
2	name	varchar(225)	latin1_swedish_ci		No	None		
3	email	varchar(225)	latin1_swedish_ci		No	None		
4	phone	varchar(225)	latin1_swedish_ci		No	None		
5	a_date	date			No	None		
6	d_date	date			No	None		
7	people	int(10)			No	None		
8	room_type	varchar(225)	latin1_swedish_ci		No	None		

7. Request_room_delete Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(6)			No	None		AUTO_INCREMENT
2	name	varchar(225)	latin1_swedish_ci		No	None		
3	email	varchar(225)	latin1_swedish_ci		No	None		
4	phone	varchar(225)	latin1_swedish_ci		No	None		
5	a_date	date			No	None		
6	d_date	date			No	None		
7	people	int(10)			No	None		
8	room_type	varchar(225)	latin1_swedish_ci		No	None		

8. Resort_rooms table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	Room_id 	int(11)			No	None		
2	Type	varchar(20)	latin1_swedish_ci		Yes	NULL		

CHAPTER 4 IMPLEMENTATION

4.1. Implementation

The project is implemented in HTML, Php and CSS for Front-End (website) and MySQL for Back-End (database). It has three-tier architecture with Front-End forming Application Layer and Back-End forming Middle Layer and Database. Input validation is done in Php. Php communicates with the local server. The server responds with HTML code that is then displayed on the Front-End. The server communicates with MySQL using mysqlconnector.

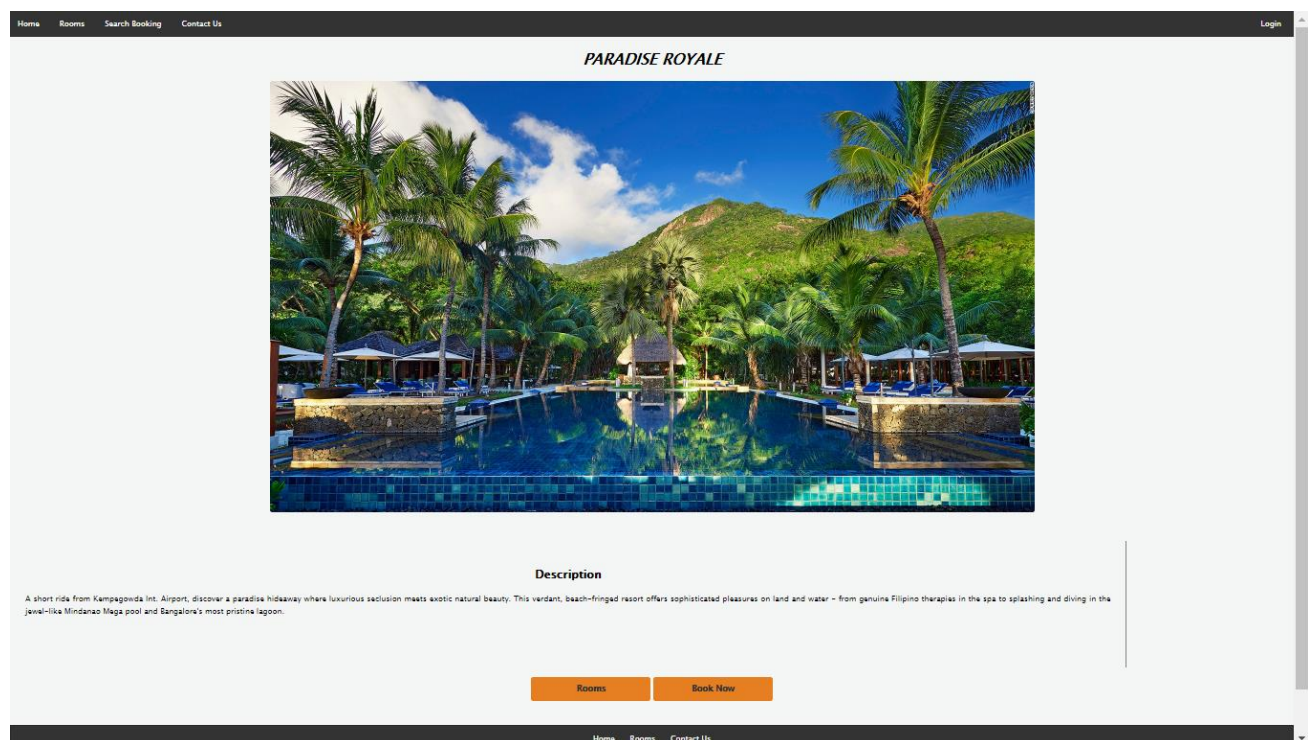
4.2. Programming Language Selection

Php was used to communicate with MySQL because PHP is the most popular scripting language for web development. It is free, open source and server-side (the code is executed on the server). MySQL is a Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). It is also free and open source. The combination of PHP and MySQL gives unmet options to create just about any kind of website - from small contact form to large corporate portal.

HTML and CSS is used for the front-end as it versatile and easy to use.

4.3. Snapshot

i. Home page



ii. Admin login page:

[Home](#) [Rooms](#) [Search Booking](#) [Contact Us](#) [Login](#)

Login

[SIGN IN](#)

[Register](#)

[Only for Staff](#)

iii. Admin registration page:

[Home](#) [Rooms](#) [Search Booking](#) [Contact Us](#) [Login](#)

Registration

[Submit](#)

[Back to Login](#)

[Only for Staff](#)

iv. Room booking page:

Booking & Contact

Details

Arrive :

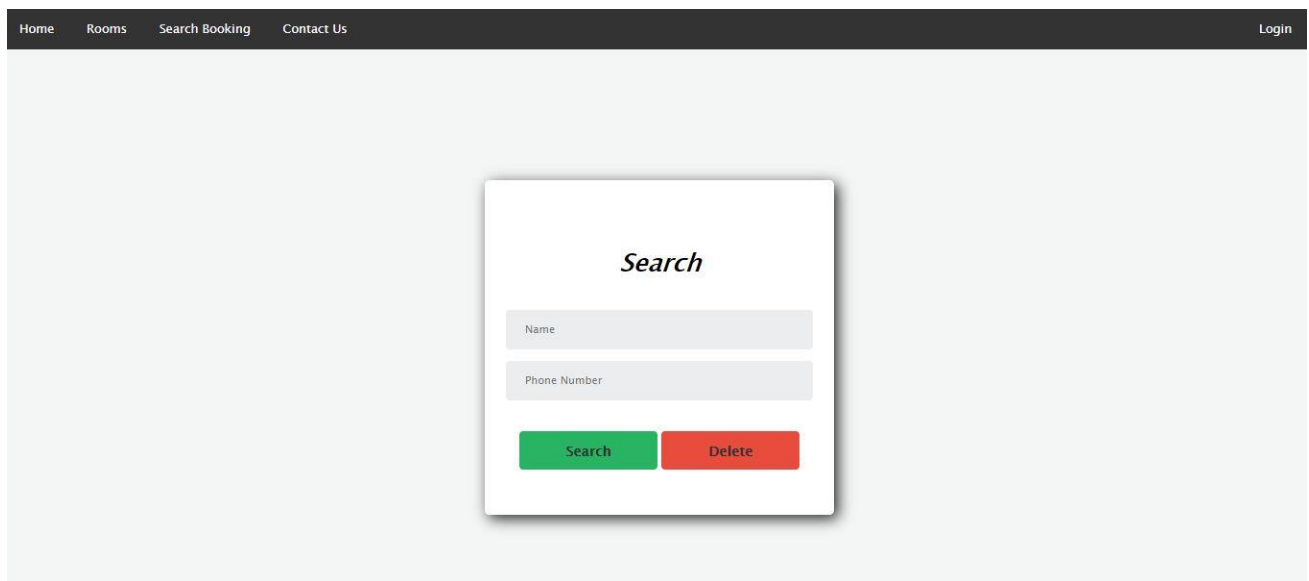
Depart :

People :

Rooms :

SUBMIT

v. Search booking page:



The image shows a web page header with a dark grey navigation bar containing the links 'Home', 'Rooms', 'Search Booking', and 'Contact Us'. A 'Login' link is positioned on the right side of the header. The main content area has a light grey background and features a white search form centered on the page. The form is titled 'Search' in a bold, italicized font. It includes two input fields: 'Name' and 'Phone Number'. Below these fields are two buttons: a green 'Search' button and a red 'Delete' button.

Home Rooms Search Booking Contact Us Login

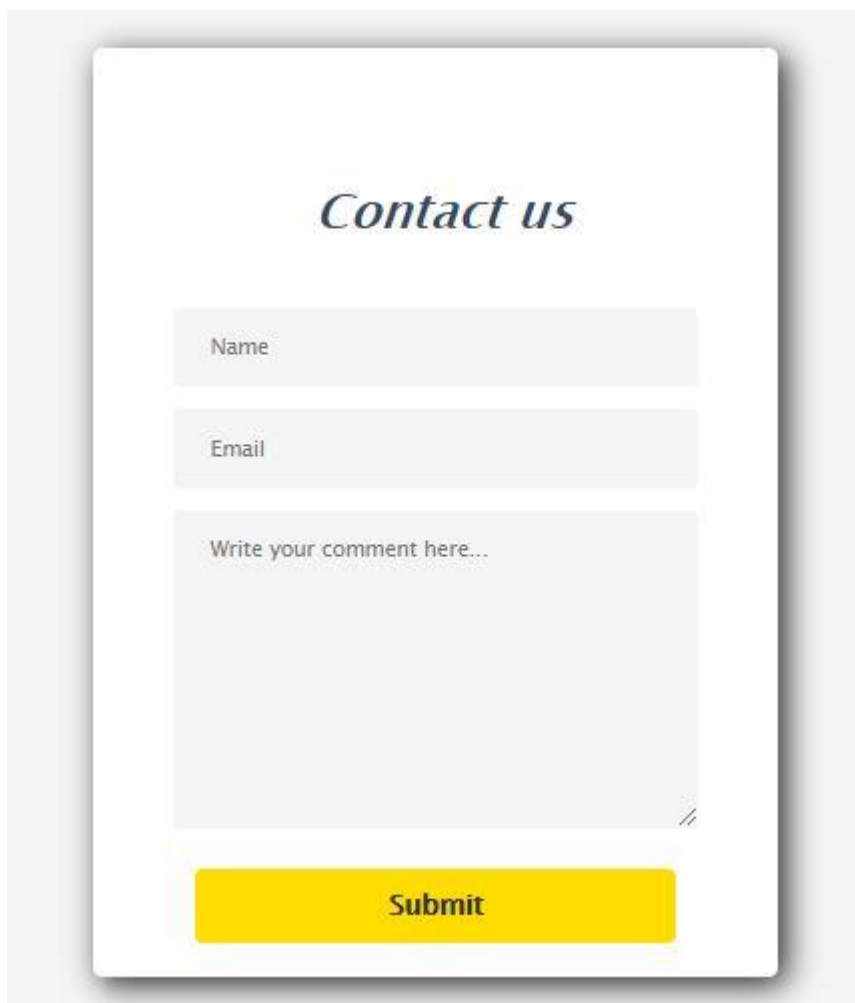
Search

Name

Phone Number

Search Delete

vi. Contact us page



The image shows a contact form centered on a light grey background. The form is titled 'Contact us' in a bold, italicized font. It contains three input fields: 'Name', 'Email', and a larger text area for a comment with the placeholder text 'Write your comment here...'. A yellow 'Submit' button is located at the bottom of the form.

Contact us

Name

Email

Write your comment here...

Submit

vii. Rooms status page

<i>Rooms</i>			
<i>Rooms empty</i>		<i>Rooms Filled</i>	
Room ID	Room Type	Customer ID	Room ID
603	Deluxe	1	701
605	Deluxe	3	702
103	Single	7	101
105	Single	8	102
106	Single	9	201
107	Single	10	602
108	Single	10	604
109	Single	13	202

viii. Assign room page:

Assign Room

Customer ID

Room ID

Assign Room

4.4. Procedures

PL/SQL is a block-structured language that enables developers to combine the power of SQL with procedural statements. A stored procedure in PL/SQL is nothing but a series of declarative SQL statements which can be stored in the database catalogue. A procedure can be thought of as a function or a method. They can be invoked through triggers, other procedures, or applications on Java, PHP etc.

```
--
-- Dumping data for table `manager`
--

INSERT INTO `manager` (`mid`, `last_name`, `first_name`, `email`, `password`) VALUES
(1, 'sharlock', 'iam', 'sherlock@sa.com', '202cb962ac59075b964b07152d234b70'),
(2, 'admin', 'admin', 'admin@r.com', '202cb962ac59075b964b07152d234b70'),
(6, 'taslim', '', 'taslim@gmail.com', '250cf8b51c773f3f8dc8b4be867a9a02'),
(5, 'ratul', '', 'ratul@gmail.com', '202cb962ac59075b964b07152d234b70'),
(7, 'all', 'admin', 'a', '202cb962ac59075b964b07152d234b70');
```

This procedure retrieves the PNRNumber of a passenger who booked a ticket and the PNRNumber is used as an input for the view tickets page which enables the user to print his or her ticket.

4.5. Triggers

A trigger is a special kind of a stored procedure that executes in response to certain action on the table like insertion, deletion or updating of data. It is a database object which is bound to a table and is executed automatically. You can't explicitly invoke triggers. The only way to do this is by performing the required action on the table that they are assigned to.

This trigger automatically updates the number of tickets booked attribute (noticketsbooked) and the capacity in the Flights table once the user books a ticket or cancels a ticket.

```
--
-- Triggers `request_room`
--

DROP TRIGGER IF EXISTS `request_delete`;
DELIMITER $$
CREATE TRIGGER `request_delete` BEFORE DELETE ON `request_room` FOR EACH ROW BEGIN
INSERT INTO request_room_delete
VALUES (old.id,old.name,old.email,old.phone,old.a_date,old.d_date,old.people,old.room_type);
END
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `request_insert`;
DELIMITER $$
CREATE TRIGGER `request_insert` AFTER INSERT ON `request_room` FOR EACH ROW BEGIN
INSERT INTO request_room_backup VALUES (NEW.id,NEW.name,NEW.email,NEW.phone,NEW.a_date,NEW.d_date,NEW.people,NEW.room_type);
END
$$
DELIMITER ;
```

CHAPTER 5 CONCLUSION

5.1. Conclusion

The Resort Management System is a great improvement over the manual system which uses lots of manual work and paper. The computerization of the system speeds up the process. This system was thoroughly checked and tested with dummy data and found to be very reliable.

Thus, we have implemented a fully comprehensive and minimalistic efficient system for use by Admins and customers without any additional training.

5.2. Limitations of the Project

A user once books his/her room cannot make any changes to his/her details like name, phone number etc. moreover there is no feature to edit the customer details. Payment verification steps needs to be added.

5.3. Future Enhancements

The Resort Management System can be enhanced by including more functionality like payment functionality, sending confirmation acknowledgement , maintain Billing etc.

We can further add an improvised booking system which can be far more efficient and reliable.

REFERENCES

- <https://youtu.be/qjwc8ScTHnY>
- <https://stackoverflow.com/questions/tagged/php+mysql>
- <https://www.youtube.com/playlist?list=PLillGF-Rfqbp2IB6ZS4BBBcYPagAjpjn>
- https://www.w3schools.com/php/php_mysql_connect.asp
- <https://www.sitepoint.com/stored-procedures-mysql-php/>
- <http://www.cse.chalmers.se/edu/year/2016/course/TDA357/HT2016/tutorials/4-triggers/index.html>
- <https://www.c-sharpcorner.com/UploadFile/c63ec5/create-mysql-trigger-in-php/>
- <https://youtu.be/L5RpqspNAuc>

•