VISESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI-590018

A MINI PROJECT REPORT ON

RESORT BOOKING SYSTEM

Mini Project Report submitted in partial fulfilment of the requirement

for the award of degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

R. DEKSHITHA **1KS18CS075**

SWETHA BIJANAPALLI **1KS18CS107**

Under the guidance of

Dr. DAYANANDA R B

Prof, Dept. Of CSE

Mr. KUMAR K

Asst. Prof, Dept. Of CSE

Department of Computer Science & Engineering

K.S.INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

2020-2021

K.S.INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

Department of Computer Science & Engineering

CERTIFICATE

Certified that the Mini Project Work (18CSL58) entitled "RESORT BOOKING SYSTEM" is a bonafide work carried out by

R. Dekshitha 1KS18CS075

Swetha Bijanapalli 1KS18CS107

in the partial fulfilment for V semester B.E., Mini Project Work in the branch of **Computer Science and Engineering** prescribed by **Visvesvaraya Technological University, Belagavi** during the period of September 2020 to January 2021. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated. The Mini Project Work Report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the Bachelor of Engineering degree.

.....	.....
Signature of the Guide [Dr.Dayananda R B]	Signature of the Guide [Kumar.K]
.....	.....
Signature of the HOD [Dr. Rekha B. Venkatapur]	Signature of the Principal & CEO [Dr. K.V.A. Balaji]

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task will be incomplete without the mention of the individuals, we are greatly indebted to, who through guidance and providing facilities have served as a beacon of light and crowned our efforts with success.

We take this opportunity to express our sincere gratitude to our college K.S. Institute of Technology, Bengaluru for providing the environment to work on our project.

We would like to express our gratitude to our MANAGEMENT, K.S. Institute of Technology, Bengaluru, for providing a very good infrastructure and all the kindness forwarded to us in carrying out this project work in college.

We would like to express our gratitude to Dr. K.V.A Balaji, Principal & CEO, K.S. Institute of Technology, Bengaluru, for his valuable guidance.

We like to extend our gratitude to Dr. Rekha.B.Venkatapur, Professor and Head, Department of Computer Science & Engineering, for providing a very good facilities and all the support forwarded to us in carrying out this Project Work Phase-I successfully.

We also like to thank our Project Guides, Dr.DAYANANDA R B, Professor, Mr. Kumar K, Assistant Professor, Department of Computer Science & Engineering for their help and support provided to carry out the Mini Project Work successfully.

We are also thankful to the teaching and non-teaching staff of Computer Science & Engineering, KSIT for helping us in completing the Project Work.

ABSTRACT

This project aims at creating a Resort Management System which can be used by users to reserve different kinds of cottages and other facilities online.

It enables the users to have a glance at all kinds of facilities, activities and different kinds of stay available. Users can get the entire feel of the resort by looking at the site. It enables the users to compare all resorts and plan out their vacation accordingly. Users can book cottages and other facilities in just few minutes without any hassle. The users have to fill their basic details and type of cottage for booking it. The availability of the cottage is checked and details provided is stored. The system generates a confirmation bill once the booking is successful.

The aim of the project is to show the real-world implementation of database management system developed using technologies such as HTML, CSS, JavaScript and PHP. The application is flexible and provides attractive site for the resort.

CONTENTS

Chapter 1

INTRODUCTION

1.1 OVERVIEW

"Resort Booking System" is designed for users to reserve different kinds of cottages and other facilities online. It enables the users to have a glance at all kinds of facilities, activities and different kinds of stay available. Users can book cottages and other facilities in just few minutes without any hassle. The users have to fill their basic details and type of cottage for booking it. The availability of the cottage is checked and details provided is stored. The system generates a confirmation bill once the booking is successful.

1.2 PROBLEM STATEMENT

The main aim of "Resort Booking System" is to make an attractive site and give an overview of the resort and enable customers to book different kinds of cottages and other facilities.

1.3 DATABASE MANAGEMENT SYSTEM

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. The DBMS essentially serves as an interface between the database and end users application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified, and the database schema, which defines the database's logical structure. These three foundational elements help to provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

1.5 HTML / JavaScript

HTML is a markup language used for structuring and presenting content on the web and the fifth and current major version of the HTML standard. HTML5 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. JavaScript often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

1.6 PHP

PHP: Hypertext Preprocessor (or simply PHP) is a server-side scripting language designed for Web development, and also used as a general-purpose programming language. It was originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML code, 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 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 reason behind the popularity of PHP is its several advantages. PHP is most suited for the purpose of web development. The advantages of PHP are discussed briefly below:

- Cross Platform All the PHP based applications can run on various types of platforms. PHP is supported by majority of Operating Systems, some of which includes Solaris, UNIX, Windows and Linux. The mentioned platforms can be used to write codes in PHP and also view web pages or run the PHP based applications.
  PHP easily interfaces with MySQL and Apache both. An effortless integration of PHP can be done with various other technologies like Java and there is no requirement of re-development. Therefore, saving both time and money, giving it an important advantage.

- Easy database connection

A programming language like PHP is widely used on the internet and needs to connect to the database very often. Therefore, having a feature that could help PHP to connect to database easily is mandatory. Several websites such as the e- commerce websites, require good database management system. PHP has a built-in module that helps it in connecting with database easily. Therefore, PHP has a great demand in the field of web development where a data driven website needs to be developed. PHP significantly reduces the time needed in developing the web application that needs an efficient database management system.

- Easy to use

PHP is widely used because it is easy to use. In contrast with other programming languages that are complex, PHP is simple, fluent, clean and organized, hence it is a boon for the new users. PHP has a well-organized syntax which is logical at the same time.

PHP does not require any intensive studying or manual to use it. Command functions of PHP are easily understood as the user can easily figure out from the name of the commands itself what it does. A person who is new to PHP can still code because the syntax is somewhat similar to C.

A person who is new to PHP can still code because the syntax is somewhat similar to C. Hence, if a person who knows C can easily code in PHP. Hence, it is easier to create and optimize the application using PHP.

Speed is the primary need of web development. There are people who face the challenge of slow internet connection and slow data speed. Furthermore, a fast loading website is always preferred by people across the globe. When compared to other programming languages, PHP is found to be the fastest programming language.

In normal circumstances, it takes a lot of time to connect to the database, when you attempt to fetch certain data from the database. It takes a lot of time in connecting to the database, then executing the statement and finally getting the data. PHP performs these set of tasks faster than other scripting languages. PHP is faster in both connecting to the database and in using other important applications.

The high speed of PHP gives it an advantage over other scripting languages and gives it an application in important administrations such as the server administration and mail functionalities.

- Open source

One of the important advantages of PHP is that it is Open Source. Therefore, PHP is readily available and is entirely free. In contrast to other scripting languages used for web development which requires the user to pay for the support files, PHP is open to everyone, anytime and anywhere.

Chapter 2

REQUIREMENTS SPECIFICATION

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

2.1 OVERALL DESCRIPTION

A reliable and scalable database driven web application with security features that is easy to use and maintain is the requisite.

2.2 SPECIFIC REQUIREMENTS

The specific requirements of the Stock Market System are stated as follows:

2.2.1 SOFTWARE REQUIREMENTS

- Web Browser
- Google Chrome 62.0.3202.89 (stable)
- Editor – Microsoft Visual Studio Code v1.29
- XAMPP v7.2.8
  - XAMPP control panel v3.2.2
  - phpMyAdmin v4.8.2
  - PHP v7.2.8
  - Apache v2.4.34

- Operating System – Windows 8 or later
- Database Support – MariaDB v10.1.34
  - Client API library version - mysqlnd 5.0.12-dev – 20150407

2.2.2 HARDWARE REQUIREMENTS

- Processor - Intel Pentium 4 or higher
- RAM – 2 GB (4 GB Recommended)
- HDD – 4 GB
- Monitor – VGA of 1024x768 screen resolution
- Keyboard and Mouse

2.2.3 TECHNOLOGY STACK

- HTML provides a means to structure text based information in a document. It allows users to produce web pages that include text, graphics and hyperlinks.
- JavaScript is a scripting language which supports the development of both client and server applications. It is preferred at client side to write programs that can be executed by a web browser within the context of a web page.
- CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language.
- SQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is issued for the purpose of data definition and data manipulation.
- PHP: Hypertext Preprocessor (or simply PHP) is a server-side scripting language designed for Web development, and also used as a general-purpose programming language.

Chapter 3

3. DETAILED DESIGN

3.1 SYSTEM DESIGN

PHP is written as standard text files with the .php extension. PHP files are often saved within a folder in a web server's public directory (or a web root directory). On most systems this will either be named public or public_html. For example, if a file was saved as index.php in a web root directory, a user could access it by typing http://www.example.org or http://www.example.org/index.php.

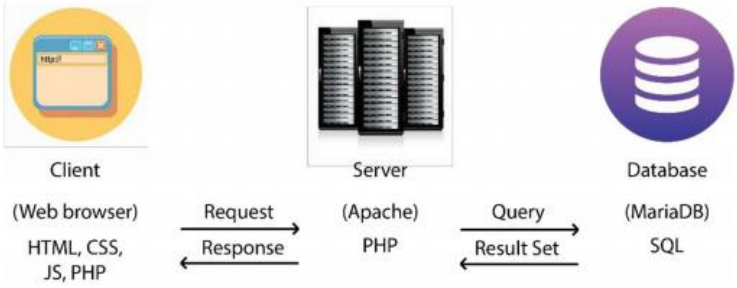


Fig. 3.1 Web architecture

When a user types in http://example.org in a Web client (a browser, for instance), the client issues a GET request to the server (let's assume that we are both using Apache). When Apache gets this request, it looks for a file named index.php (or index.html, remember the directory indexes from earlier?). If a file named index.php is found, Apache essentially says "Hey, this is a PHP file because it has the .php extension. I am going to give this to the PHP interpreter". After Apache decides that is a PHP file, it gives it to the PHP interpreter. When PHP receives the file, it reads through it and executes any PHP code it can find. After it is done with the file, the PHP interpreter gives the output of the code, if any, back to Apache. When Apache gets the output back from PHP, it sends that output back to a browser which renders it to the screen.

3.2 ENTITY RELATIONSHIP DIAGRAM

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business.

An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities. Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three schema approach to software engineering. While useful for organizing data that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ER Diagram is unlikely to be helpful on its own in integrating data into a pre-existing information system.

Cardinality notations define the attributes of the relationship between the entities. Cardinalities can denote that an entity is optional.

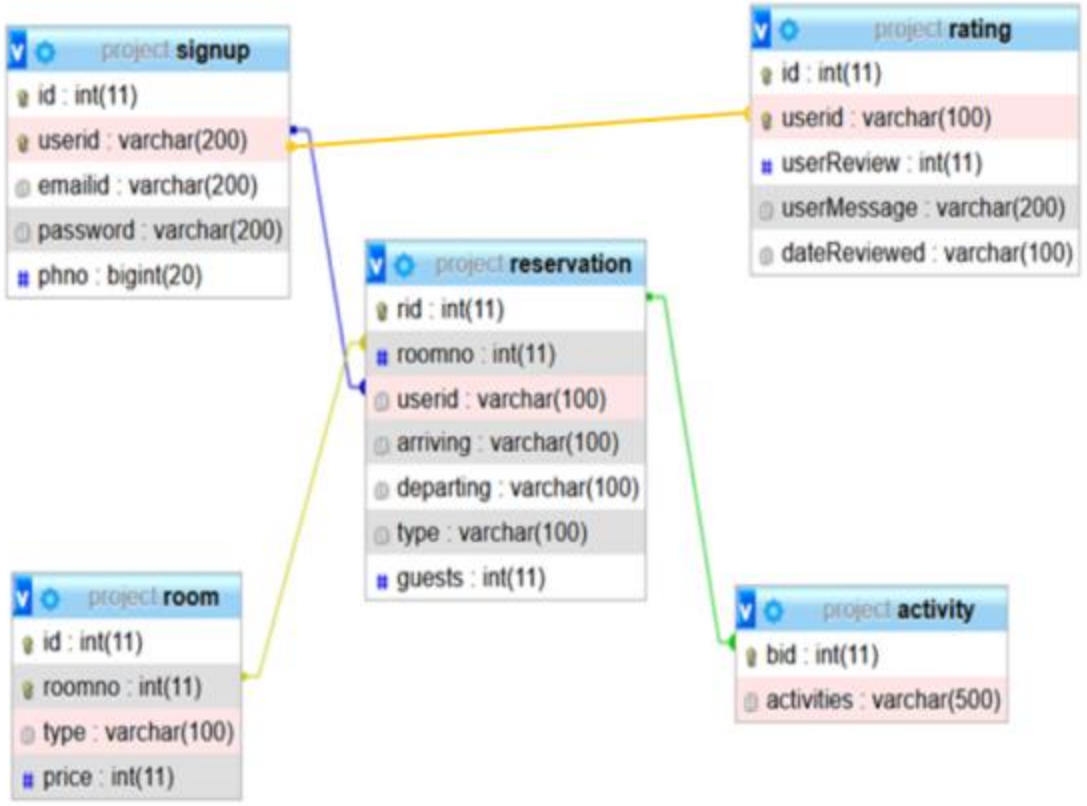


Fig. 3.2.1 Enhanced ER diagram of Resort Booking System

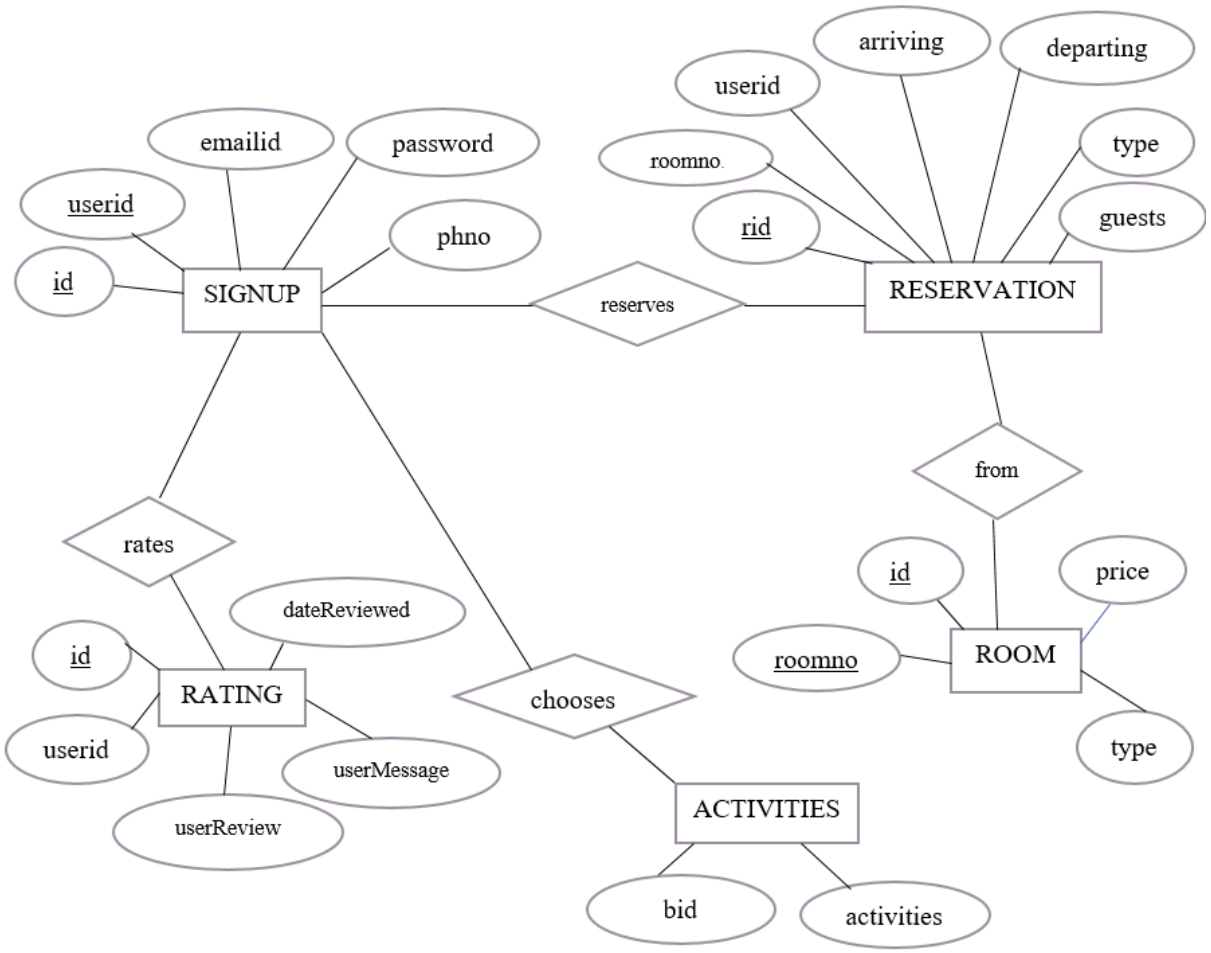


Fig. 3.2.2 ER diagram of Resort Booking System

3.3 RELATIONAL SCHEMA

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute. A schema diagram helps organize values in the database.

The following diagram shows the schema diagram for the database.

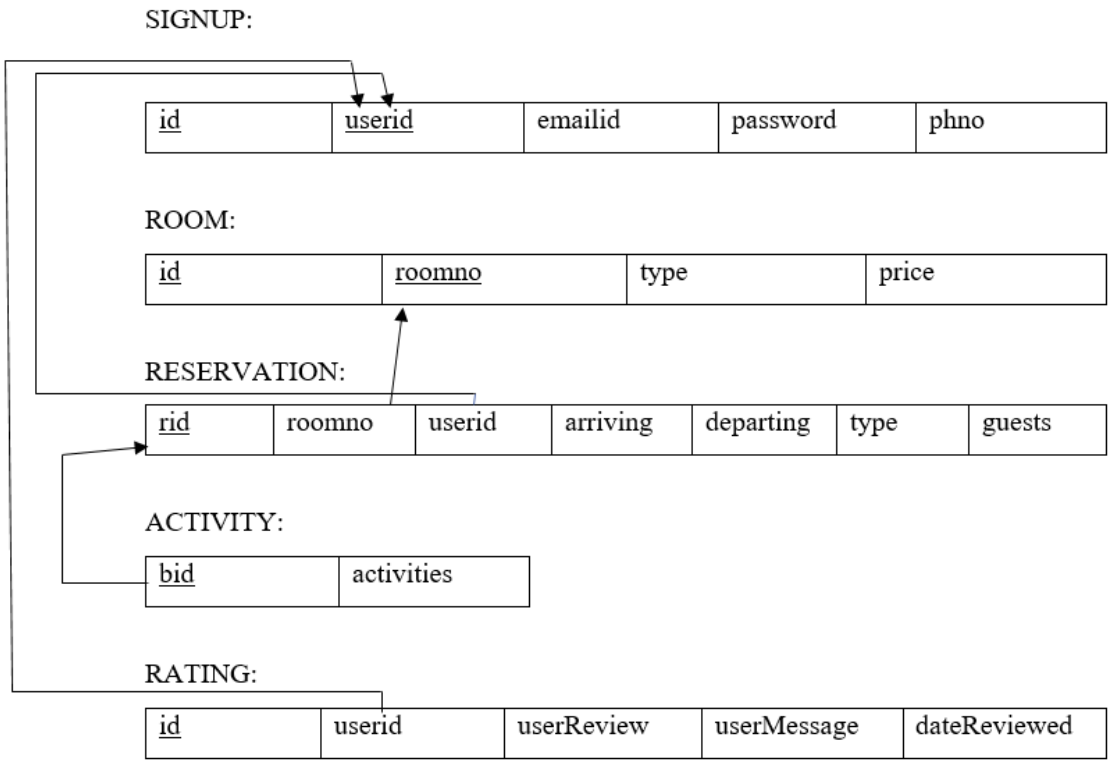


Fig. 3.3 Schema Diagram

3.4 DESCRIPTION OF TABLES

The database consists of five tables:

1. Signup : It stores the user details.
   - id int(11) : Unique user id done by auto increment.
   - userid varchar(200) : Username of the user.
   - Emailid varchar(200) : Email id of the user.
   - Password varchar(200) : Password of the user.
   - Phno bigint(20) : Phone number of the user.

2. Room: It stores room/cottage details.
   - Id int(11) : Unique room id done by auto increment.
   - Roomno int(11) : Room numbers of the present rooms/cottages.
   - Type varchar(100) : Type of room/cottage.
   - Price int(11) : Price of the room/cottage.

3. Reservation: It stores the reservation details.
   - Rid int(11) : Unique reservation id done by auto increment
   - Roonno int(11) : Room number which is reserved.
   - Userid varchar(100) : Username of the user.
   - Arriving varchar(100) : Arriving date.
   - Departing varchar(100) : Departing date.
   - Type : varchar(100) : Type of room/cottage selected.
   - Guests : int(11) : No. of guests.

4. Activity: Stores the selected activities.
   - Bid int(11) : Reservation id.
   - Activities varchar(500) : Activities selected.

5. Rating: Stores the number of stars and rating given by the user.
   - Id int(11) : Rating id done by auto increment.
   - Userid varchar(100): Username of the user.
   - userReview int(11) : No. of star rating.
   - userMessage varchar(200) : Rating message.
   - dateReviewed varchar(100) : Date of review.

Chapter 4

IMPLEMENTATION

4.1 MODULES AND THEIR ROLES

4.1.1 Register: Register for the new user.

```
<?php
session_start();
$con = mysqli_connect("localhost","root","","project") or die("Connect failed: %s\n". $con -> error);
if(isset($_POST['submit'])){
$userid=mysqli_real_escape_string($con,$_POST['userid']);
$emailid=mysqli_real_escape_string($con,$_POST['emailid']);
$password=mysqli_real_escape_string($con,$_POST['password']);
$phno=mysqli_real_escape_string($con,$_POST['phno']);
$s = " select * from signup where userid = '$userid'";
$result=mysqli_query($con, $s);
$num = mysqli_num_rows($result);
if($num == 1){
  echo"<script>alert('username already taken')</script>";
}
else{
  $insert_clint="insert into signup(userid,emailid,password,phno)
  values('$userid','$emailid','$password','$phno')";
  mysqli_query($con, $insert_clint);
  echo"<script>alert('Data upload successful')</script>";
}
mysqli_close($con);
?>
```

4.1.2 Login: Logins the user.

```
<?php
session_start();
$con = mysqli_connect("localhost","root","","project") or die("Connect failed: %s\n". $con -> error);
$userid=mysqli_real_escape_string($con,$_POST['userid']);
$password=mysqli_real_escape_string($con,$_POST['password']);
$s = " select * from signup where userid = '$userid' && password = '$password'";
$result=mysqli_query($con, $s);
$num = mysqli_num_rows($result);
if($num == 1){
    $_SESSION['userid']=$userid;
    header('location:home.php');
}
else{
$sq = " select * from signup where userid = '$userid' ";
$r=mysqli_query($con, $sq);
$n = mysqli_num_rows($r);
if($n >0){
  echo '<body style="background-color:lightblue">';
  echo"<script>alert('Incorrect password')</script>";
  echo "<a href='signup.php'>Try again</a><br>";
}
else{
  echo '<body style="background-color:lightblue">';
  echo"<script>alert('User is not registered, please register')</script>";
  echo "<a href='signup.php'>Register</a><br>";
}
}
mysqli_close($con);
?>
```

4.1.3 Password: Retrieves the password of the authenticated user in case of forgot password.

```
<?php
$con = mysqli_connect("localhost","root","","project") or die("Connect failed: %s\n". $con -> error);
if(isset($_POST['submit'])){
   $userid=mysqli_real_escape_string($con,$_POST['userid']);
   $emailid=mysqli_real_escape_string($con,$_POST['emailid']);
   $phno=mysqli_real_escape_string($con,$_POST['phno']);
   $sq="SELECT * FROM signup where userid='$userid' AND emailid='$emailid' AND phno='$phno'";
   $result = mysqli_query($con, $sq);
  if ($result -> num_rows > 0) {
  while($row = $result-> fetch_assoc()){
     echo '<body style="background-color:lightblue">';
     echo "<h3>Your password is</h3> <br>";
     echo "Username : " . $row["userid"]. "<br>";
     echo "Password : " . $row["password"].  "<br>";
     echo "<a href='signup.php'>Go back to login page.</a><br>";
}
echo"<script>alert('Data retrieved successfully')</script>"
 }
else{echo"<script>alert('INVALID DETAILS')</script>";}
}
?>
```

4.1.4 Reservation: Inputs and stores reservation details and checks room availability.

```
<?php
$con = mysqli_connect("localhost","root","","project") or die("Connect failed: %s\n". $con -> error);
if(isset($_POST['submit'])){
$arriving=mysqli_real_escape_string($con,$_POST['arriving']);
$departing=mysqli_real_escape_string($con,$_POST['departing']);
$type=mysqli_real_escape_string($con,$_POST['type']);
$guests=mysqli_real_escape_string($con,$_POST['guests']);
$userid=$_SESSION['userid'];
if($type=="Superior Luxury Cottage"){
   $n=(rand(100,105));
}
if($type=="Tree House"){
   $n=(rand(400,405));
}
if($type=="Suite Cottage"){
   $n=(rand(300,305));
}
if($type=="Luxury Cottage"){
   $n=(rand(200,205));
}
$s = " select roomno from room where type='$type' AND roomno NOT IN( select roomno from reservation where arriving='$arriving' AND departing='$departing')";
$result=mysqli_query($con, $s);
$num = mysqli_num_rows($result);
if($num > 0){
$i="select roomno from reservation where arriving='$arriving' AND departing='$departing' AND roomno=$n";
$r=mysqli_query($con, $i);
```

```
$nu=mysqli_num_rows($r);
if($nu > 0){
   echo"<script>alert('generated room no is booked, try again sorry')</script>";
}
else{
$insert_clint="insert into reservation(roomno,userid,arriving,departing,type,guests)
values('$n','$userid','$arriving','$departing','$type','$guests')";
$run_costumer = mysqli_query($con, $insert_clint);
if($run_costumer){
   $sq="select * from reservation where arriving='$arriving' AND departing='$departing' AND
roomno=$n";
   $result = mysqli_query($con, $sq);

  if ($result -> num_rows > 0) {
  while($row = $result-> fetch_assoc()){
     echo '<body style="background-color:lightblue">';
     echo "<h1>Booking Details:</h1> <br>";
     echo "Booking id: " . $row["rid"]. "<br>";
     echo "Room No.: " . $row["roomno"].  "<br>";
     echo "Username: " . $row["userid"]. "<br>" ;
     echo "Arriving: " . $row["arriving"]. "<br>" ;
     echo "Departing: " . $row["departing"]. "<br>" ;
     echo "Room type: " . $row["type"]. "<br>" ;
     echo "Guests: " . $row["guests"]. "<br>" ;
     echo "To make your stay with us exciting and eventful, we have curated a wide
variety of activities for you and your loved ones!<br>";
     echo "Please note your booking id for further process. " .$row["rid"]. "<br>";
     echo "<a href='trial.html'>Choose your activities</a><br>";
}
}
echo"<script>alert('Data upload successful')</script>";
```

```
}
else{echo"<script>alert('Data upload not successful !')</script>";
}
}
}
else{echo"<script>alert('Room not available!')</script>";
}
}
mysqli_close($con);
?>
```

4.1.5 Activities: Stores the selected activities and also displays entire booking information.

```
<?php
$con = mysqli_connect("localhost","root","","project") or die("Connect failed: %s\n". $con -> error);
if(isset($_POST['submit'])){
$bid=mysqli_real_escape_string($con,$_POST['bookingid']);
$activities=mysqli_real_escape_string($con,$_POST['activity']);
$i="select rid from reservation where rid=$bid";
$r=mysqli_query($con, $i);
$nu=mysqli_num_rows($r);
if($nu<1){
   echo"<script>alert('Booking id is invalid')</script>";
}
else{
$insert_clint="insert into activity(bid,activities)
values('$bid','$activities')";
$run_costumer = mysqli_query($con, $insert_clint);
if($run_costumer){
```

```
    $sq="select * from reservation,activity where rid=bid AND rid=$bid";
    $result = mysqli_query($con, $sq)
   if ($result -> num_rows > 0) {
   while($row = $result-> fetch_assoc()){
      echo '<body style="background-color:lightblue">';
      echo "<h1>Complete Booking Details:</h1> <br>";
      echo "Booking id: " . $row["rid"]. "<br>";
      echo "Room No.: " . $row["roomno"].  "<br>";
      echo "Username: " . $row["userid"]. "<br>" ;
      echo "Arriving: " . $row["arriving"]. "<br>" ;
      echo "Departing: " . $row["departing"]. "<br>" ;
      echo "Room type: " . $row["type"]. "<br>" ;
      echo "Guests: " . $row["guests"]. "<br>" ;
      echo "activities: " .$row["activities"]. "<br>";
}
echo "<a href='home.php'>Go back to home page.</a><br>";
}
echo"<script>alert('Data upload successful')</script>";
 }
else{echo"<script>alert('Already uploaded!')</script>";
    echo "<a href='home.php'>Go back to home page.</a><br>";
}
}
}
?>
```

4.1.6 : Booking : Retrieves and displays the confirmed booked details.

```
<?php
session_start();
$con = mysqli_connect("localhost","root","","project") or die("Connect failed: %s\n". $con -> error);
 $u=$_SESSION['userid'];
 $sq="SELECT * FROM reservation,activity where rid=bid and userid='$u'";
 $result = mysqli_query($con, $sq);
   if ($result -> num_rows > 0) {
   while($row = $result-> fetch_assoc()){
      echo '<body style="background-color:lightblue">';
      echo "<h1>Complete Booking Details:</h1> <br>";
      echo "Booking id: " . $row["rid"]. "<br>";
      echo "Room No.: " . $row["roomno"].  "<br>";
      echo "Username: " . $row["userid"]. "<br>" ;
      echo "Arriving: " . $row["arriving"]. "<br>" ;
      echo "Departing: " . $row["departing"]. "<br>" ;
      echo "Room type: " . $row["type"]. "<br>" ;
      echo "Guests: " . $row["guests"]. "<br>" ;
      echo "activities: " .$row["activities"]. "<br>";

}
echo"<script>alert('Data retrieved successfully')</script>";
    }
else{echo"<script>alert('No bookings done')</script>";}
?>
```

4.1.7 Rating: Inputs and stores the rating details given by user.

```
<?php
   $POST = filter_var_array($_POST, FILTER_SANITIZE_STRING);
   $POSTI = filter_var_array($_POST, FILTER_SANITIZE_NUMBER_INT);
   $con = mysqli_connect("localhost","root","","project") or die("Connect failed: %s\n". $con
-> error);
        if(isset($_POST['starRate'])){
      $starRate=mysqli_real_escape_string($con,$_POST['starRate']);
      $rateMsg=mysqli_real_escape_string($con,$_POST['rateMsg']);
      $date=mysqli_real_escape_string($con,$_POST['date']);
      $name=mysqli_real_escape_string($con,$_POST['name']);
      $insert_clint="insert into rating(userid,userReview,userMessage,dateReviewed)
      values('$name','$starRate','$rateMsg','$date')";
    $run_costumer = mysqli_query($con, $insert_clint);
      if($run_costumer){
    echo"<script>alert('Data upload successful')</script>";
    }
    else{echo"<script>alert('Data upload not successful !')</script>";}
   }
?>
```

4.2 RESULT

The resulting system is able to:

- Registers if it's a new user.
- Authenticate user credentials during login.
- Salted encryption for security of user passwords.
- Retrieves password of user in case of forgot password.
- Allow users to view the different kinds of rooms/cottages available.
- Allow user to book the rooms/cottages.
- Checks availability and displays booking details if booking is confirmed.
- Allows users to select activities and stores it in their booking confirmation.
- Allows users to give reviews and also displays all reviews.
- Allows user to view their history/previous booking information.

Chapter 5

TESTING

5.1 SOFTWARE TESTING

Testing is the process used to help identify correctness, completeness, security and quality of developed software. This includes executing a program with the intent of finding errors. It is important to distinguish between faults and failures. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors. It can be conducted as soon as executable software (even if partially complete) exists. Most testing occurs after system requirements have been defined and then implemented in testable programs.

5.2 MODULE TESTING AND INTEGRATION

Module testing is a process of testing the individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommend testing the smaller building blocks of the program. It is largely white box oriented. The objective of doing Module testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Module testing allows implementing of parallelism into the testing process by giving the opportunity to test multiple modules simultaneously.

The final integrated system too has been tested for various test cases such as duplicate entries and type mismatch.

5.3 LIMITATIONS

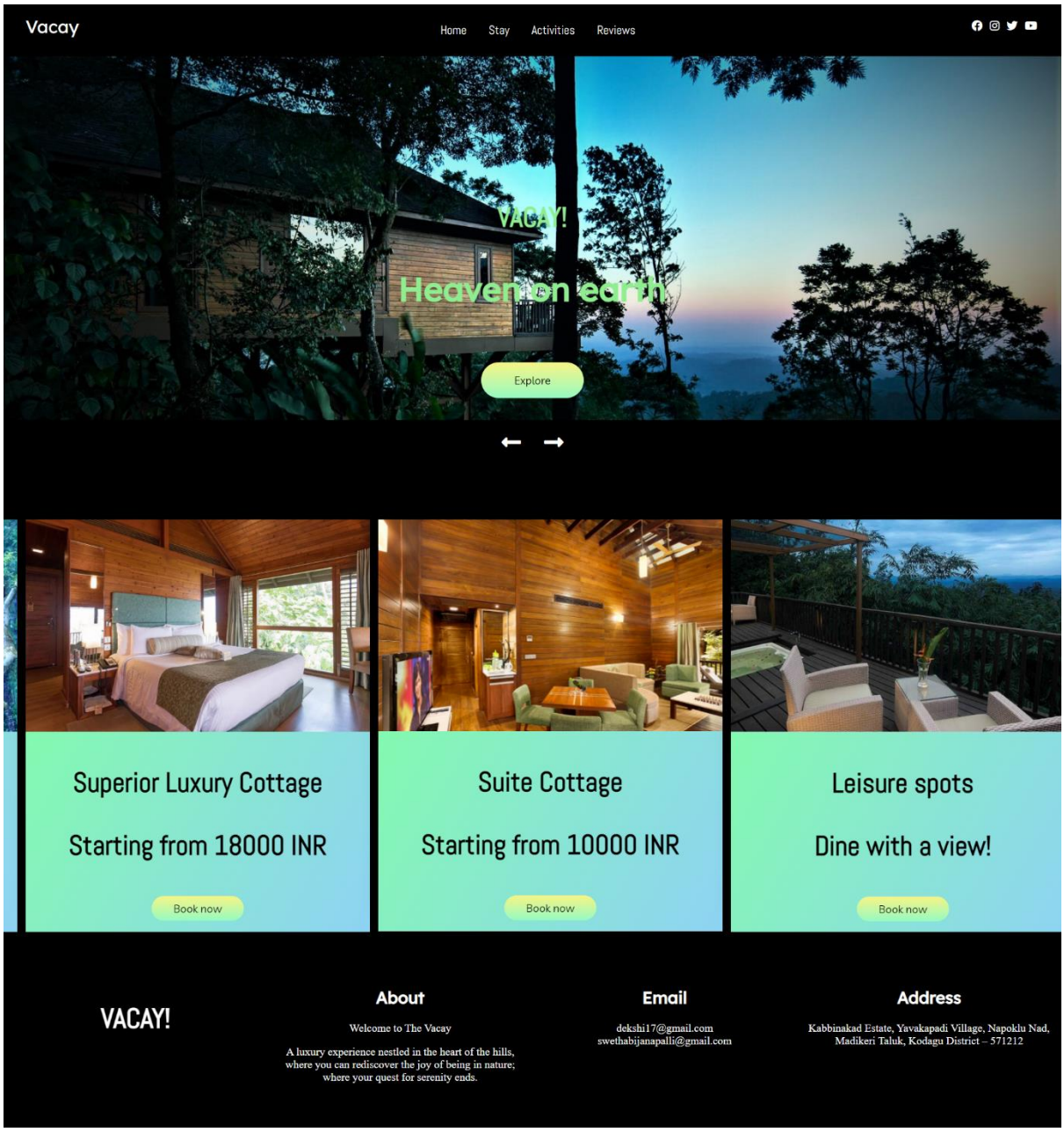
- User's session timing is not recorded.
- Limited number of rooms.
- Final bill including all charges is not computed.
- Better secure interfaces needed for communication with the banks.
- Only restricted to Nifty 50 market currently, needs to be extended to more markets.

Chapter 6

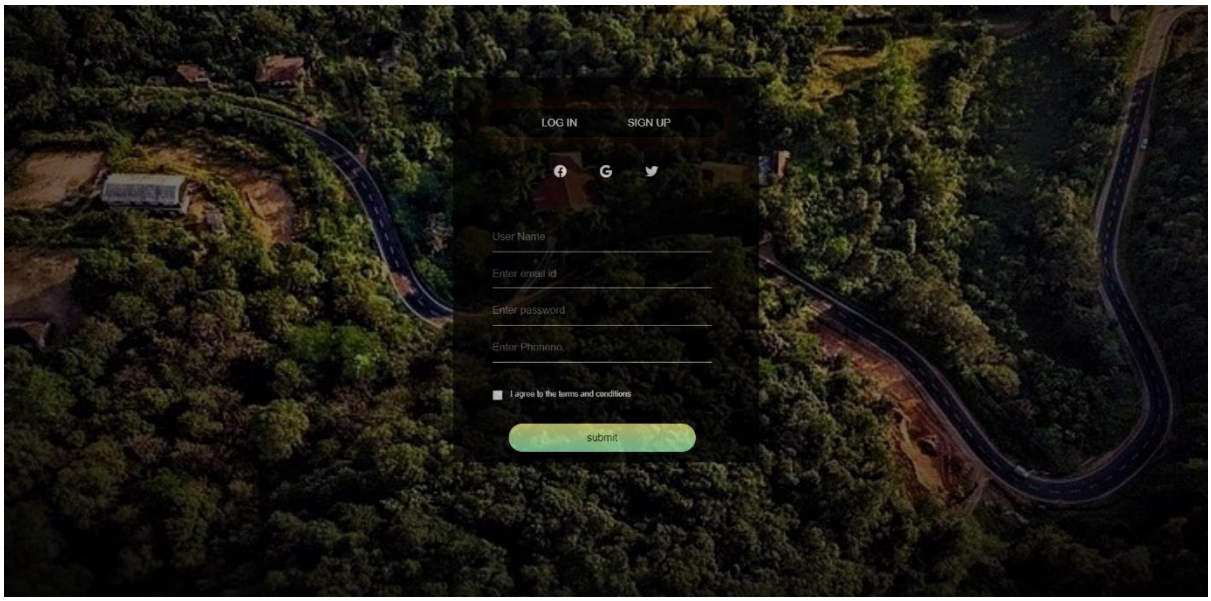
SNAPSHOTS

This chapter consists of working screenshots of the project.

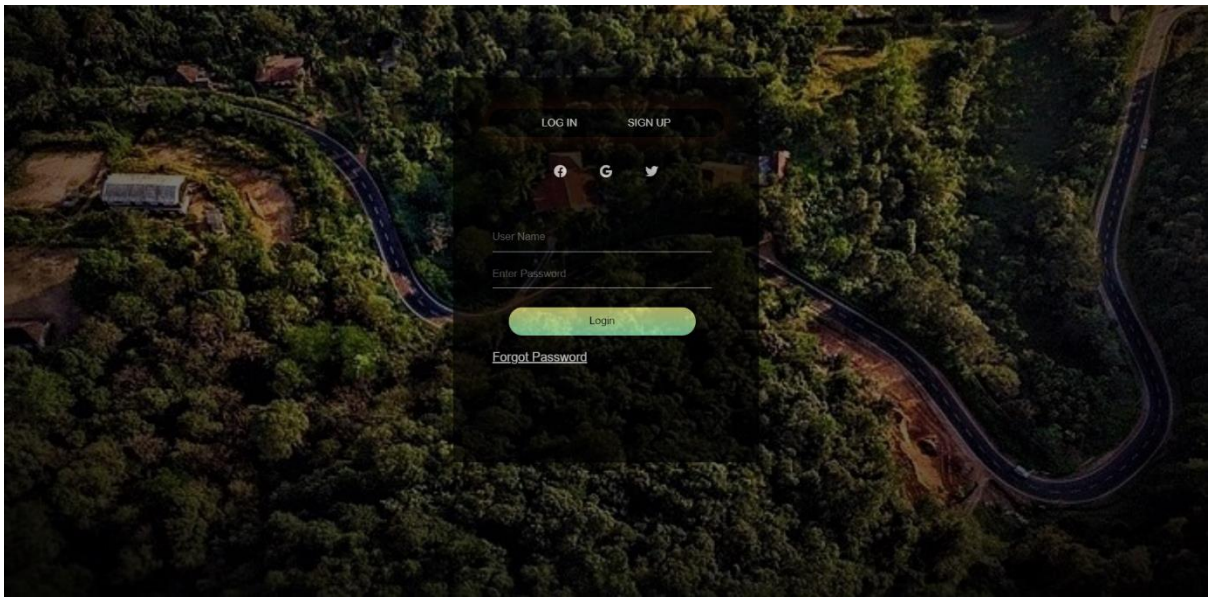
6.1 HOME PAGE



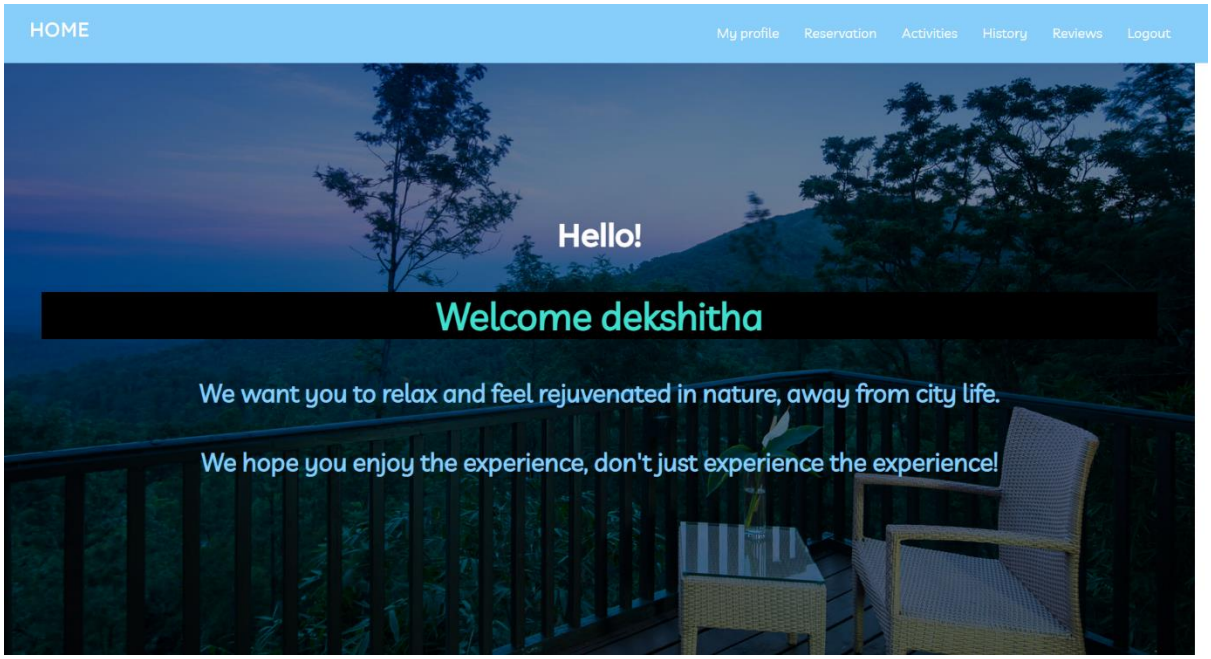
6.2 REGISTRATION PAGE



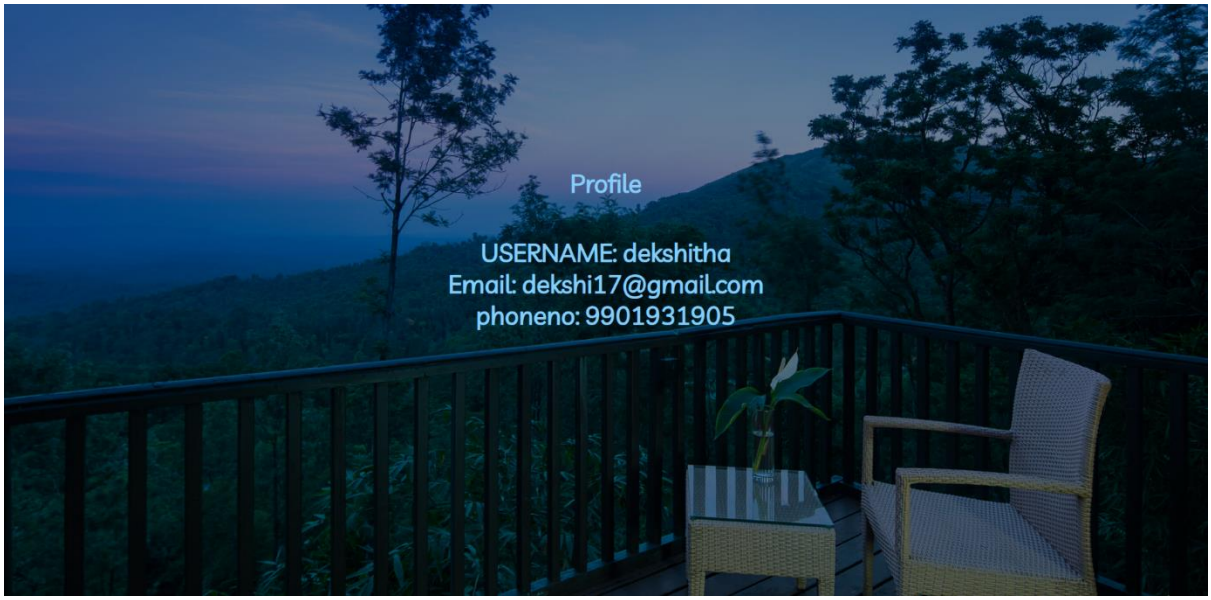
6.3 LOGIN PAGE



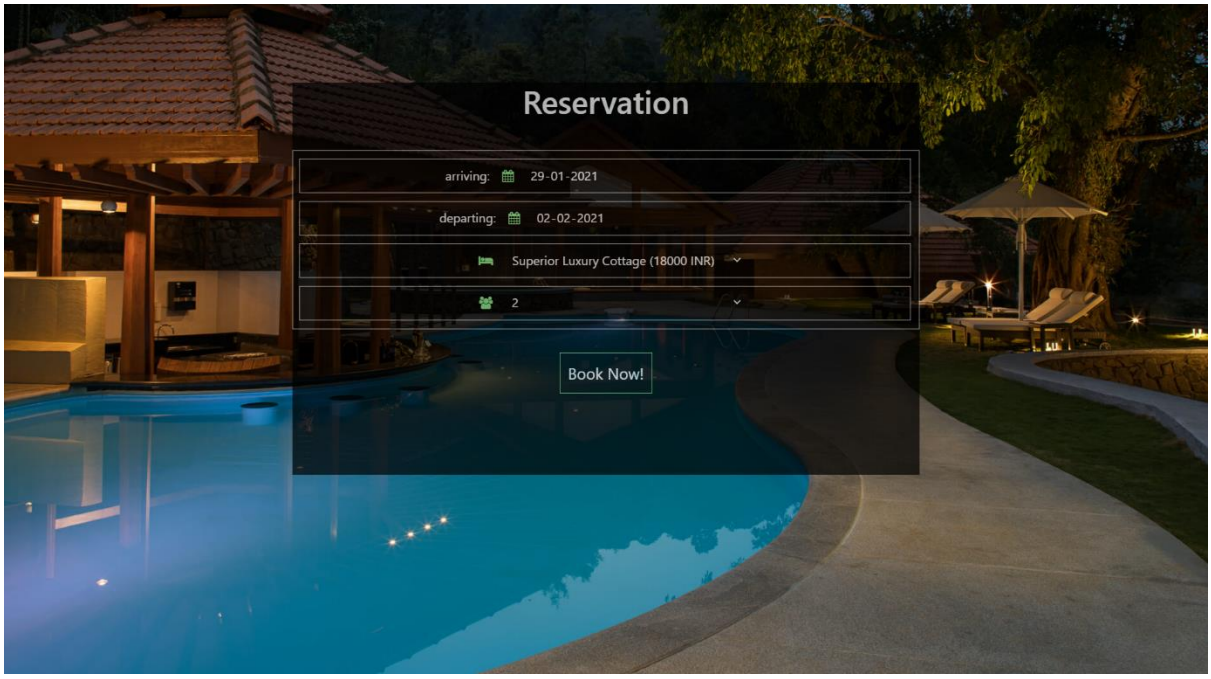
6.4 USER HOME PAGE



6.5 USER PROFILE



6.6 RESERVATION

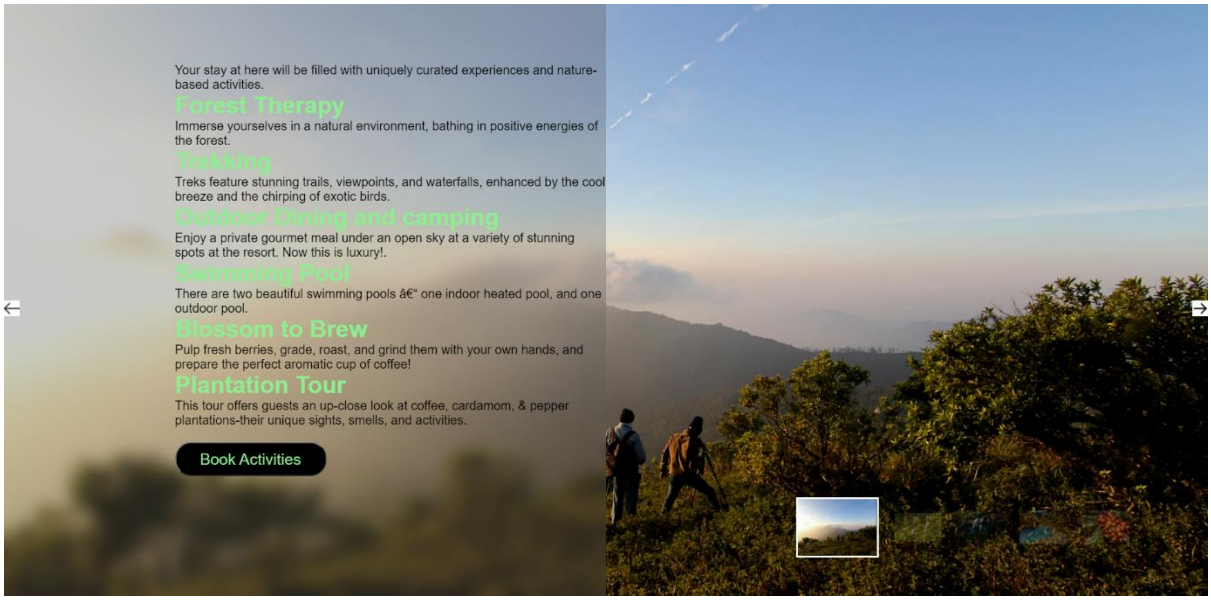


6.7 BOOKING DETAILS

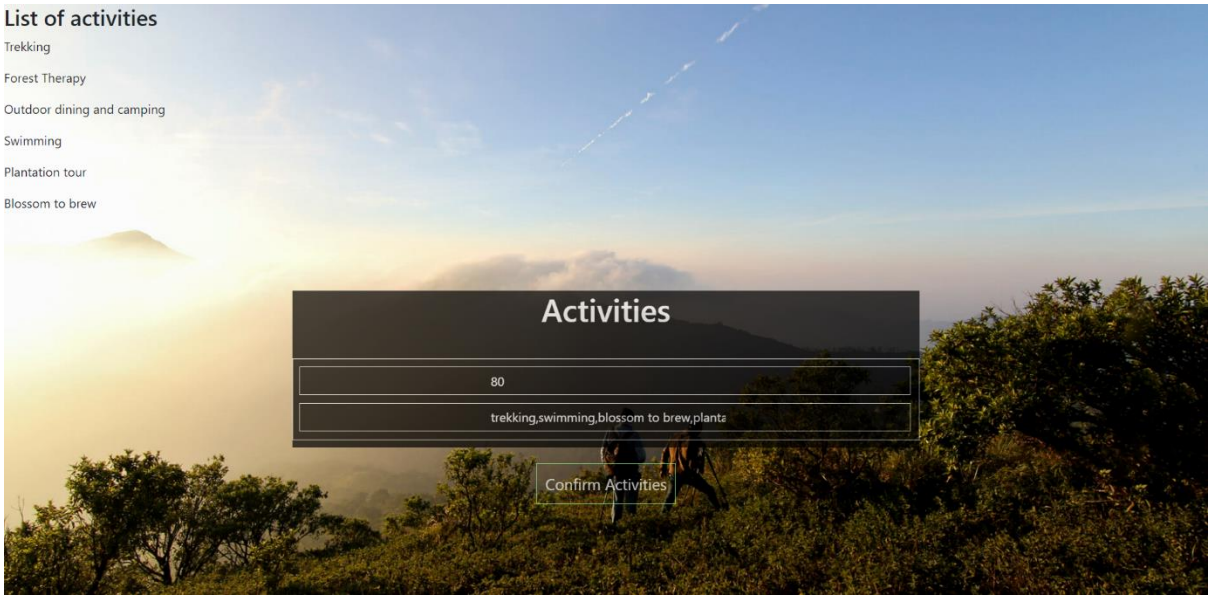
Booking Details:

Booking id: 80
Room No.: 105
Username: dekshitha
Arriving: 2021-01-29
Departing: 2021-02-02
Room type: Superior Luxury Cottage
Guests: 2
To make your stay with us exciting and eventful, we have curated a wide variety of activities for you and your loved ones!
Please note your booking id for further process. 80
Choose your activities

6.7 ACTIVITIES



6.8 BOOKING ACTIVITIES



6.10 HISTORY

Complete Booking Details:

Booking id: 1
Room No.: 301
Username: dekshitha
Arriving: 2021-01-17
Departing: 2021-01-20
Room type: Suite Cottage
Guests: 3
activities: trekking,swimming,blossom to brew,plantation tour,outdoor dining and camping

Complete Booking Details:

Booking id: 61
Room No.: 400
Username: dekshitha
Arriving: 2020-12-27
Departing: 2020-12-29
Room type: Tree House
Guests: 2
activities: trekking,swimming,blossom to brew,plantation tour,outdoor dining and camping

Complete Booking Details:

Booking id: 66
Room No.: 101
Username: dekshitha
Arriving: 2021-01-16
Departing: 2021-01-18
Room type: Superior Luxury Cottage
Guests: 4
activities: trekking,swimming,blossom to brew,plantation tour,outdoor dining and camping

Complete Booking Details:

Booking id: 68
Room No.: 405
Username: dekshitha
Arriving: 2021-01-19
Departing: 2021-01-24
Room type: Tree House
Guests: 2
activities: trekking,swimming,plantation tour

Complete Booking Details:

Booking id: 80
Room No.: 105
Username: dekshitha
Arriving: 2021-01-29
Departing: 2021-02-02
Room type: Superior Luxury Cottage
Guests: 2
activities: trekking,swimming,blossom to brew,plantation tour,outdoor dining and camping

6.11 REVIEWS

CONCLUSION

The Resort Booking System provides attractive interface to view the resort entirely and book cottages among the different kinds rooms/cottages that are available. It also shows brief description of all the activities and facilities available. It enables the users to decide their activities as well. It allows simplified operation and is a time saving platform with the ability to view historical data and confirmation bill. The application has been completed successfully and tested with suitable test cases. It is user friendly and contains suitable options for users. This is developed using HTML5, CSS, JavaScript, Phpmyadmin. The goals achieved by this project are:

- Get a sense of whole resort and the stay.
- Centralized database.
- Easy and quick booking.
- Easier buying, selling of various stocks.
- User friendly environment.
- Efficient management of reservation.
- Ability to view historical data.
- View confirmation details.

FUTURE ENHANCEMENTS

Future upgrades to this project will implement:

- More attractive website with better features.
- Better view of available rooms/cottages.
- Better design for confirmation bill.
- Computation of entire bill.
- Ability to perform online transaction.

REFERENCES

1. Ramakrishnan, R., & Gehrke, J. (2011). Database management systems. Boston: McGraw-Hill.

2. Monson-Haefel, R. (2007). J2EE Web services. Boston, Mass: Addison-Wesley. Silberschatz A., Korth H. F., & Sudarshan S. (2011).

3. Database systems concepts. Estados Unidos: McGraw-Hill Companies, Inc.

4. Hanna P. (2002): JSP 2.0 The Complete Reference, Second Edition McGraw Hill Education.

5. David F. (2011). JavaScript: The Definitive Guide Sixth edition.

6. https://www.w3schools.com

7. https://www.canvasjs.com

8. https://getbootstrap.com/

9. https://fontawesome.com