

A MINI PROJECT REPORT ON
“CRICKET GROUND BOOKING SYSTEM”

in the partial fulfilment of the requirements

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

APPLICATION DEVELOPMENT USING JAVA LAB
(IIIrd Year - 1st semester)

Submitted by

K. RAJA GOPAL
(18B91A05B3)

Under the guidance of
Smt. K. Durga Bhavani
Assistant professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SRKR ENGINEERING COLLEGE (A)

Chinna Amiram, Bhimavaram, West Godavari Dt.

[2020-2021]



S R K R ENGINEERING COLLEGE (A), BHIMAVARAM
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the mini project entitled “**Cricket Ground Booking System**” is being submitted by **K. RAJA GOPAL** in partial fulfilment of the requirement for the degree of B. Tech IIIrd year 1st semester in the Department of Computer Science and Engineering, SRKR Engineering College(A), Bhimavaram is a record of bonafide work carried out by her under my guidance from July 2020 to March 2021. The candidate worked right under my guidance.

Supervisor
(K. DURGA BHAVANI)

S. No	TABLE OF CONTENTS	Pg. No
	Abstract	
1	CHAPTER 1	
1.1	Introduction	1
2	CHAPTER 2	
2.1	Problem statement	2
3	CHAPTER 3	
3	Software Requirement Analysis	3
3.1	Purpose	3
3.2	Scope	3
3.3	Objective	3
	FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS	
3.4	Functional requirements	4
3.5	Non-Functional requirements	5
4	CHAPTER 4	
4	Database Design	8
4.1	Database design schema	8
4.2	Tables obtained	9
5	CHAPTER 5	
	IMPLEMENTATION	
5.1	Screenshots	11
5.2	Code	16
6	CHAPTER 6	
6	Conclusion	50
7	CHAPTER 7	
7	Reference	51

LIST OF FIGURES

S. No	5.1 Screenshots	Pg. No
5.1.1	Screen shot for home page	11
5.1.2	Screen shot for login	11
5.1.3	Screen shot for register page	12
5.1.4	Screen shot for booking page	12
5.1.5	Screen shot for user dashboard	13
5.1.6	Screen shot for cancellation page	13
5.1.7	Screen shot for admin page	14
5.1.8	Screen shot for admin dashboard	14
5.1.9	Screen shot for enable bookings	15

LIST OF TABLES

S. No	4.1 Database Design Schema	Pg. No
4.1.1	Table created for user info	8
4.1.2	Table created for admin	8
4.1.3	Table created for book	8
4.1.4	Table created for cancel	8
4.1.5	Table created for payments	9

ABSTRACT

“Cricket Ground Booking System” is a web application used to book cricket ground for cricket matches. The project is aimed at exposing the relevance and importance of Ground Booking Systems. It is projected towards enhancing the relationship between customers and ground authorities through the use of this software, and thereby making it convenient for the customers to book the ground as when they require such that they can utilize this software to make bookings. This software has two parts. First is user part and the administrator part. User part is used as a front end and administrator is the back end. Administrator is used by ground authority. It will allow the customers to access database and allow new customers to sign up for online access.

1.1 INTRODUCTION

The Cricket Ground Booking System in PHP is a simple booking system for cricket ground. The project contains admin and the user side. The admin can enable or disable bookings, view the booking details and cancellations and can manage the payments.

The users are the clients who want to search and book the ground suitable for them. A user has to login first to make the booking. If he/she has not registered, they can register first and then login. A user can search for available slots and can book a slot convenient for them and they can cancel the booked slot. This project is a very simple project that saves time and money for the customer.

2.1 PROBLEM STATEMENT

Cricket is being played everywhere in the world. So many cricket matches are being played in the world. Good cricket ground is very essential to conduct tournaments. Searching for a good cricket ground and booking that ground is a complex task and it wastes customer time. To overcome this problem Cricket Ground Booking System has developed.

3.SOFTWARE REQUIREMENT SPECIFICATIONS

3.1 PURPOSE:

The Cricket Ground Booking System is a project that will help the cricket ground authorities to make their business online as they can increase their customers. This project has 2 sides of user's restriction, which are the admin and the client/website visitor. The admin can enable or disable bookings and manage the payments. The client will browse the website and find available slots and they can book a slot which is convenient to them according to their schedule.

The system itself has user-friendly functionalities. The online Ground booking system is easy to modify to add the additional features you wished for.

3.2 SCOPE:

The scope of this project is to book cricket ground.

3.3 OBJECTIVE:

It is designed to make easier for players to book ground online. Searching for a good cricket ground and to check whether it is available or not is a complex task for many players. The system heavily relies on human interaction with the physical space and entity.

FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS:

3.4 Functional Requirements:

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it is important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behavior under specific conditions.

MODULES

ADMINISTRATOR

- **Login Page**
 - This feature will help to secure the data of the system. This is the page where staff will enter their system credentials to access the ground booking system administrator's side.
- **Dashboard page**
 - The page where staff will be redirected after logging into the system.
- **Enable Bookings**
 - The page where admin can enable or disable bookings
- **Payment Page**
 - The page where admin can manage the payments

USER

- **Register Page**
 - The page where user can register for a new account.

- **Login Page**
 - The page where user can login into the website.
- **Book page**
 - The page where user can book new slots.
- **Cancel Page**
 - The page where user can cancel booked slots
- **Dashboard Page**
 - The page where user will be redirected after successful login

3.5 Non-Functional Requirements:

1.Efficiency of use:

The average time it takes to accomplish a user's goals, how many tasks a user can complete without any help, the number of transactions completed without errors, etc.

2.Intuitiveness:

How simple it is to understand the interface, buttons, headings, etc.

3.Low perceived workload:

How many attempts are needed by users to accomplish a particular task.

4.Security:

Security requirements ensure that the software is protected from unauthorized access to the system and its stored data. It considers different levels

of authorization and authentication across different users roles. For instance, data privacy is a security characteristic that describes who can create, see, copy, change, or delete information. Security also includes protection against viruses and malware attacks.

5.Reliability:

Reliability defines how likely it is for the software to work without failure for a given period of time. Reliability decreases because of bugs in the code, hardware failures, or problems with other system components. To measure software reliability, you can count the percentage of operations that are completed correctly or track the average period of time the system runs before failing.

6.Performance:

Performance is a quality attribute that describes the responsiveness of the system to various user interactions with it. Poor performance leads to negative user experience. It also jeopardizes system safety when it's overloaded.

7.Availability:

Availability is gauged by the period of time that the system's functionality and services are available for use with all operations. So, scheduled maintenance periods directly influence this parameter. And it's important to define how the impact of maintenance can be minimized. When writing the availability requirements, the team has to define the most critical components of the system that must be available at all time. You should also prepare user notifications in case the system or one of its parts becomes unavailable.

8.Scalability:

Scalability requirements describe how the system must grow without negative influence on its performance. This means serving more users, processing more data, and doing more transactions. Scalability has both hardware and software implications. For instance, you can increase scalability by adding memory, servers, or disk space. On the other hand, you can compress data, use optimizing algorithms, etc.

□ **SOFTWARE REQUIREMENTS**

1. Web server - Xampp server
2. Operating System - Supports WINDOWS/LINUX

□ **HARDWARE REQUIREMENTS**

1. Processor - Intel core i3 or i5
2. RAM - 3GB OR ABOVE
3. Hard Disk - 256GB OR ABOVE

4. DATABASE DESIGN

4.1 Database Design Schema:

4.1.1 USER_INFO

FIELD	TYPE	DESCRIPTION
name	Varchar	User name
email	Varchar	User email
mobile	Varchar	User contact
password	Varchar	password

4.1.2 ADMIN

FIELD	TYPE	DESCRIPTION
email	Varchar	Admin email
password	Varchar	password
accept	Varchar	Enable or Disable bookings

4.1.3 BOOK

FIELD	TYPE	DESCRIPTION
bid	int	Booking Id
email	Varchar	User email
slot_date	Varchar	Slot_date
status	Text	Booking status
payment	Text	Payment status

4.1.4 CANCEL

FIELD	TYPE	DESCRIPTION
bid	int	Booking Id
email	Varchar	User email
slot_date	Varchar	Slot_date

4.1.5 PAYMENT

FIELD	TYPE	DESCRIPTION
bid	int	Booking Id
email	Varchar	User email
slot_date	Varchar	Slot_date
mode	Varchar	Mode Of Payment

4.2 Tables Obtained.

USERS_INFO:

NAME	EMAIL	MOBILE	PASSWORD
rohit	rohit@gmail.com	7095455865	123
raju	raju@gmail.com	9876543210	raju
bhargav	bhargav@gmail.com	9756432108	123
kali	kali@gmail.com	8976543210	kali

BOOK

BID	EMAIL	SLOT_DATE	STATUS	PAYMENT
27	rohit@gmail.com	2021-02-24	expired	paid
28	raju@gmail.com	2021-02-27	booked	paid
35	kali@gmail.com	2021-03-05	booked	pending

CANCEL

BID	EMAIL	SLOT_DATE
24	rohit@gmail.com	2021-02-25
38	raju@gmail.com	2021-03-01
16	bhargav@gmail.com	2021-02-27

ADMIN

EMAIL	PASSWORD	ACCEPT
admin@gmail.com	admin	enable

PAYMENT

BID	EMAIL	SLOT_DATE	MODE
27	rohit@gmail.com	2021-02-24	cash
28	raju@gmail.com	2021-02-27	UPI

5. IMPLEMENTATION

5.1 SCREENSHOTS

5.1.1 HOME PAGE



Fig5.1.1: Displaying the home page.

5.1.2 LOGIN PAGE

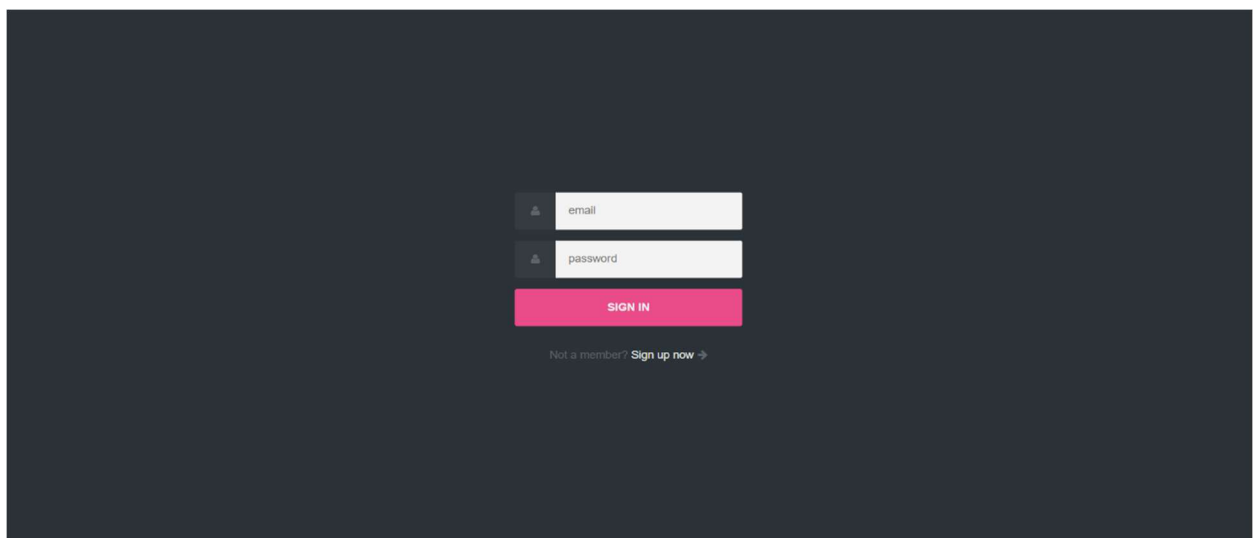
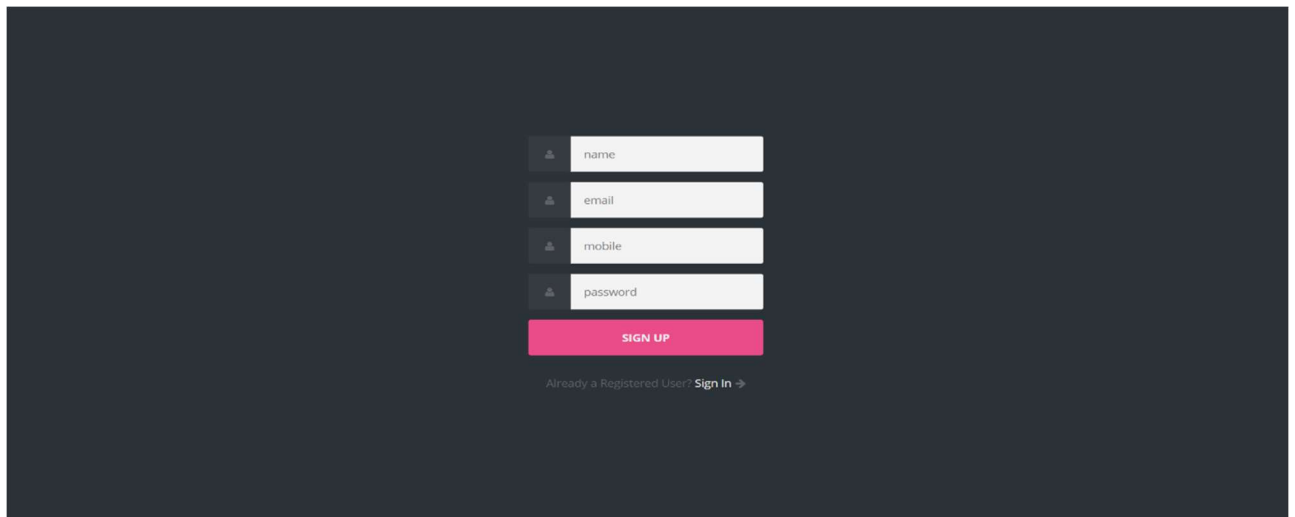


Fig5.1.2 : Displaying the Login page.

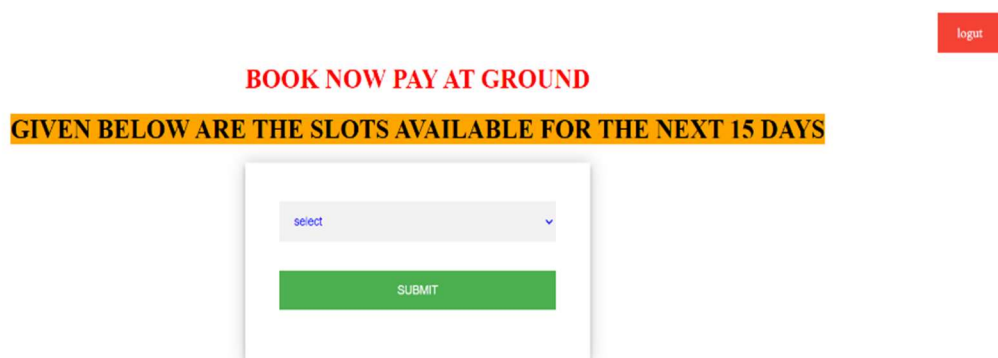
5.1.3 REGISTER PAGE:



A registration form on a dark background. It contains four input fields with labels: 'name', 'email', 'mobile', and 'password'. Each field has a small person icon to its left. Below the fields is a pink 'SIGN UP' button. Underneath the button is a link that says 'Already a Registered User? Sign In →'.

Fig5.1.3 : Displaying the page how to register .

5.1.4 BOOKING PAGE:



A booking page layout. At the top right is a red 'logut' button. Below it is a red header 'BOOK NOW PAY AT GROUND'. Under the header is a yellow banner with the text 'GIVEN BELOW ARE THE SLOTS AVAILABLE FOR THE NEXT 15 DAYS'. Below the banner is a white box containing a dropdown menu with 'select' and a blue arrow, and a green 'SUBMIT' button.

Fig 5.1.4 Display the page where user fill the details to book

5.1.5 USER DASHBOARD:

HI , ROHIT

cancel slot book new slot logout

MY UPCOMING SLOTS

booking id	email	slot_date	status	payment
27	rohit@gmail.com	2021-02-23	expired	

NOTE:BOOKING ID IS MANDATORY FOR PAYMENT AND FOR CANCELLATION

Fig5.1.5 : Displaying the details of user previous booked slots.

5.1.6 CANCELLATION

logout

Enter booking id to cancel

Enter Slot Date To Cancel

SUBMIT

Fig5.1.6 : Displaying the page how to cancel a slot.

5.1.7 ADMIN PAGE:

ADMIN LOGIN

LOGIN

Fig5.1.7 : Displaying the Admin page.

5.1.8 ADMIN DASHBOARD:

enable or disable bookings

payments

logout

booking id	slot_date	status	email	name	mobile	payment
27	2021-02-23	expired	rohit@gmail.com	rohit	7095455865	
36	2021-02-24	expired	rajagopalkureti8@gmail.com	raju	7095455865	pending
37	2021-03-04	booked	rajagopalkureti8@gmail.com	raju	7095455865	pending
39	2021-03-07	booked	puskar438@gmail.com	puskar	8801475200	paid
35	2021-03-09	booked	rajagopalkureti8@gmail.com	raju	7095455865	paid

booking id	slot_date	email	name	mobile
17	2021-02-19	bhargavchunduri@gmail.com	bhargav	8639186083
16	2021-02-20	bhargavchunduri@gmail.com	bhargav	8639186083
22	2021-02-21	ram@gmail.com	sarath	9515268489
18	2021-02-22	sujiithpavan@gmail.com	sujiith	9876543210
24	2021-02-24	rajagopalkureti8@gmail.com	raju	7095455865
34	2021-02-25	rajagopalkureti8@gmail.com	raju	7095455865
32	2021-02-26	rajagopalkureti8@gmail.com	raju	7095455865

Fig5.1.8 : Displaying the booked and cancelled slots .

5.1.9 ENABLE BOOKINGS :

ENABLE OR DISABLE BOOKINGS

select

▼

PASSWORD

admin password

SUBMIT

[dashboard](#) [logout](#)

Fig5.1.9 : Enable or Disable Bookings .

5.2 CODE

5.2 DESCRIPTION ABOUT CODE:

Source code is a human-readable text written in a specific programming language. The goal of the source code is to set exact rules and specifications for the computer that can be translated into the machine's language. As a result, source codes are the foundation of programs and websites. The source code of following project is developed in eclipse IDE. And on further SQL and PHP are mostly used. XAMPP server is thus used as a base sever and further on servlets are included. The total backened contains mysql database, thus PHPMY ADMIN is the database connector over the details acquired from specified page in the web application.

1.INDEX.HTML:

```
<html>

<head>

<title>HomePage</title>

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

<style>

a:link, a:visited {

    background-color: #f44336;color: white;

    padding: 14px 25px;text-align: center;text-decoration: none;display: inline-block;

}

a:hover, a:active {

    background-color: green;
```

```

}

#c{

    text-align:center;

}

#d{

    text-align: center;font:200px;

}

body{

    background-image: url("home.jpg");background-repeat: no-repeat;background-position:
center;background-size: cover;

}

</style>

</head>

<body>

<div id="d">

<h1> CRICKET GROUND BOOKING SYSTEM</h1>

</div><br>

<div id="c">

<br><br><br><br>

<h1>New User</h1>

<a href="signup.html">register</a>

<h1>Login To Book</h1>

<a href="login.html">login </a>

<br><br><br><br><br><br><br><br><br>

<h1 >contact us:9876543210</h1>

<h1 >mail us:admin@bookground.com</h1>

</div>

```

```
</body>
</html>
```

2.SIGNUP.HTML:

```
<!DOCTYPE html>

<html>

<head>

  <meta charset="UTF-8">

  <title>SignUp</title>

  <link rel="stylesheet" href="style.css" media="screen" type="text/css" />

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

</head>

<body>

  <div class="container">

    <div id="login">

      <form action="adduser.php" method="post" onsubmit="return validate()">

        <fieldset class="clearfix">

          <p><span class="fontawesome-user"></span><input type="text" placeholder="name"
name="uname" id="na" required></p>

          <div id="nameErr" style="color:red"></div>

          <p><span class="fontawesome-user"></span><input type="text" placeholder="email"
name="email" id="mail" required></p>

          <div id="mailErr" style="color:red"></div>

          <p><span class="fontawesome-user"></span><input type="text" placeholder="mobile"
name="mobile" id="phnum" required></p>

          <div id="phnumErr" style="color:red"></div>

          <p><span class="fontawesome-user"></span><input type="password"
placeholder="password" name="password" required></p>
```



```

<p><input type="submit" value="Sign Up"></p>

</fieldset> </form>

<p>Already a Registered User? <a href=login.html>Sign In</a><span class="fontawesome-
arrow-right"></span></p>

</div> <!-- end login -->

</div>

<script>

function validate(){

    var name = document.getElementById("na").value;

    var mail = document.getElementById("mail").value;

    var phnum = document.getElementById("phnum").value;

    var ret = true;

    if (name == " || name.length < 4) {

        document.getElementById("nameErr").innerHTML = "Enter Valid Name";

        ret = false;

    } else {

        document.getElementById("nameErr").innerHTML = "";

    }

    var mailformat = /^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-
Z0-9-]+)*$/;

    if (mail == " || !(mail.match(mailformat))) {

        document.getElementById("mailErr").innerHTML = "Enter Valid Mail Address";

        ret = false;

    } else {

        document.getElementById("mailErr").innerHTML = "";

    }

    var phoneno = /^\d{10}$/;

    if (phnum == " || !(phnum.match(phoneno))) {

```

```

        document.getElementById("phnumErr").innerHTML = "Enter Valid Phone Number";

        ret = false;

    } else {

        document.getElementById("phnumErr").innerHTML = "";

    }

return ret;

    }

</script>

</body>

</html>

```

3.ADDUSER.PHP:

```

<html>

<head>

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

<style>

a:link, a:visited {

    background-color: #f44336;color: white; padding: 14px 25px; text-align: center;

    text-decoration: none; display: inline-block;}

a:hover, a:active {

    background-color: green;

}

#c{

    text-align:center;

}

</style>

</head>

```

```

<body>

<?php

$username=$_POST["uname"];

$email=$_POST["email"];

$mobile=$_POST["mobile"];

$password=$_POST["password"];

$con=mysqli_connect("localhost","root","","ground");

if(!$con){

    die('Could not Connect My Sql:' .mysql_error());

}

else{

    $q="SELECT * from user_info where email='$email'";

    $res=mysqli_query($con,$q);

    $row=mysqli_num_rows($res);

    if($row > 0)

    { echo '<div id="c">';

        echo '<h1> user already registered with given mail</h1>';

        echo '<br>';

        echo '<a href="signup.html">signup again</a>';

    echo '</div>';

    }

    else{

        $con=mysqli_connect("localhost","root","","ground");

        $query="INSERT INTOuser_infoVALUES('$username','$email','$mobile','$password')";

        $result=mysqli_query($con,$query);

        if($result == 1)

        {echo '<div id="c">';

```

```

    echo '<h1 align="center">successfully registered</h1>';

    echo '<br>';

    echo '<a href="login.html">login</a>';

    echo '<br>';

    echo '<br>';

    echo '<a href="index.html">home</a>';

    echo '</div>';

    }

    else{

        echo "failed to register";

        }}}

?>

</body>

</html>

```

4.LOGIN.HTML:

```

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Login</title>

<link rel="stylesheet" href="style.css" media="screen" type="text/css" />

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

</head>

<body>

<div class="container">

<div id="login">

```

```

    <form action="logincheck.php" method="post">

    <fieldset class="clearfix">

    <p><span class="fontawesome-user"></span><input type="text" placeholder="email"
    name="email" required></p>

    <p><span class="fontawesome-user"></span><input type="password"
    placeholder="password" name="password" required></p>

    <p><input type="submit" value="Sign In"></p>

    </fieldset></form>

    <p>Not a member? <a href=signup.html>Sign up now</a><span class="fontawesome-
    arrow-right"></span></p>

</div> <!-- end login --></div>

</body>

</html>

```

5.LOGINCHECK.PHP:

```

<?php session_start() ?>

<html>

<head>

<style>

a:link, a:visited {

    background-color: #f44336; color: white;padding: 14px 25px;text-align: center;

    text-decoration: none;display: inline-block;

}

a:hover, a:active {

    background-color: green;

}

#c{

text-align:center;

```

```

}
</style>

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

<body>

<?php

$email=$_POST["email"];

$password=$_POST["password"];

$con=mysqli_connect("localhost","root","","ground");

if(!$con){

    die('Could not Connect My Sql:' .mysql_error());

}

else{

    $q="SELECT * from user_info where email='$email' and password='$password'";

    $res=mysqli_query($con,$q);

    $row=mysqli_num_rows($res);

    if($row ==1)

    {

        $_SESSION["email"]=$email;

        $_SESSION["name"]=$name;

        header("location:dashboard.php");

    }

    else{

        echo '<div id="c">';

        echo '<h1>invalid username or password</h1>';

        echo '<br>';

        echo '<a href="login.html">login again</a>';echo '<br>';

```

```

echo '<h1>go to home page</h1>';
echo '<a href="index.html">home</a>';
echo '</div>';
}}
?>
</body>
</html>

```

6.DASHBOARD.PHP:

```

<html>
<?php
include 'session.php' ?>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="table.css" media="screen" type="text/css" />
</head>
<body >
<nav>
<a href="logout.php">logout</a>
<a href="check_availability.php">book new slot</a>
<a href="cancel.php"> cancel slot</a>
</nav>
<?php
$con=mysqli_connect("localhost","root","","ground");
if(!$con)
{
die('Could not Connect My Sql:' .mysql_error());
}

```

```

else{
$email=$_SESSION["email"];
$q="SELECT name from user_info where email='$email'";
$row=mysqli_query($con,$q);
while($row = mysqli_fetch_assoc($row))
{
echo '<h1 style="float:left">'; echo "HI , ";echo strtoupper($row["name"]);echo '</h1>';}
$query="SELECT * FROM book where email='$email'";
$result=mysqli_query($con,$query);
if (mysqli_num_rows($result) > 0)
{
// output data of each row
echo '<br>';echo '<br>';echo '<br>';echo '<br>';
echo '<h1 > MY UPCOMING SLOTS</h1>'; echo '<br>';
echo '<table id="customers">';
echo '<th> booking id</th>';
echo '<th> email</th>';
echo '<th> slot_date</th>';
echo '<th> status</th>';
echo '<th> payment</th>';
while($row = mysqli_fetch_assoc($result))
{ echo '<tr>';
echo '<td>';echo $row["bid"];echo '</td>';
echo '<td>';echo $row["email"];echo '</td>';
echo '<td>';echo $row["slot_date"];echo '</td>';
echo '<td>';echo $row["status"];echo '</td>';
echo '<td>';echo $row["payment"];echo '</td>';

```



```

        echo '</tr>';
    } echo '</table>';

    }

    else{
echo '<h1 id="f"> you didnt book any slot</h1>';
    } }

?>

<br><br>

<h1><mark>NOTE:BOOKING ID IS MANDATORY FOR PAYMENT AND FOR
CANCELLATION</mark></h1>

</div>

</body>

</html>

```

7.CHECK_AVAILABILITY.PHP:

```

<html>

<?php

include 'session.php' ?>

<head>

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

<link rel="stylesheet" href="style1.css" media="screen" type="text/css" />

<link rel="stylesheet" href="table.css" media="screen" type="text/css" />

</head>

<body>

<a href="logout.php">logout</a>

<a href="cancel.php">cancel slots</a>

<a href="dashboard.php">dashboard</a>

```

```

<?php
$con=mysqli_connect("localhost","root","","ground");
if(!$con)
{
    die('connection failed');
}
else{
    $query="select * from admin where accept='enable'";
    $result=mysqli_query($con,$query);
    if(mysqli_num_rows($result) > 0){
        header("location:book.php");
    }
}
else{
echo '<h1> SORRY WE ARE CURRENTLY NOT ACCEPTING BOOKINGS</h1>';
} }
?>
</body>
</html>

```

8.BOOK.PHP:

```

<html>
<?php
include 'session.php' ?>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="style1.css" media="screen" type="text/css" />
<link rel="stylesheet" href="table.css" media="screen" type="text/css" />

```

```

<style>

mark{

background-color:orange;

}

</style>

</head>

<body>

<a href="logout.php">logut</a>

<div id="c"><br><br>

<h1 >BOOK NOW PAY AT GROUND</h1>

<h1> <mark>GIVEN BELOW ARE THE SLOTS AVAILABLE FOR THE NEXT 15
DAYS</mark> </h1>

<?php

$n=1;

$con=mysqli_connect("localhost","root","","ground");

if(!$con){

    die('connection failed');

}

else{

$temp=array();

$d = date('Y-m-d');

$n=1;

while($n!=15){

    array_push($temp,$d);

$d=date("Y-m-d",strtotime('+ 1days',strtotime($d)));

$n++;

}

```

```

$query="select slot_date from book";
$res=mysqli_query($con,$query);
$temp1=array();
while($row=mysqli_fetch_assoc($res)){
    array_push($temp1,$row["slot_date"]);
}
$result=array_diff($temp,$temp1);
}
?>

<div class="form">

<form action="book.php" method="post">

<select name="date" required>

<option>select</option>

<?php
foreach($result as $x){
    echo "<option value='$x'>$x</option>";
}
?>

</select>

<br><br>

<button>submit</button>

</form>

<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $date=$_POST["date"];
    $email=$_SESSION["email"];
    $con=mysqli_connect("localhost","root","","ground");

```

```

if(!$con)
{
    die('connection failed');
}
else{
    $q="select * from book where slot_date='$date'";
    $r=mysqli_query($con,$q);
    if(mysqli_num_rows($r) > 0)
    { echo '<div id="c">';
        echo '<h1> SORRY SLOT WAS ALREADY BOOKED</h1><br>';
    }
    else{
        $query="INSERT INTO book(email,slot_date,status) VALUES('$email' ,'$date','booked')";
        $res=mysqli_query($con,$query);
        if($res == 1)
        {
            header('location: dashboard.php');
        }
    }
    else{
        echo '<div id="c">';
        echo '<h1> failed to book</h1>';
    }
}
?>

</div>

</body>

</html>

```

9.CANCEL.PHP

```
<html>

<?php include 'session.php' ?>

<head>

<link rel="stylesheet" href="style1.css" media="screen" type="text/css" />

<link rel="stylesheet" href="table.css" media="screen" type="text/css" />

</head>

<body>

<a href="logout.php">logout</a>

<div class="form">

<form action="cancel.php" method="post">

<h3>Enter booking id to cancel</h3> <input type="text" name="bookid" required
><br><br>

<h3>Enter Slot Date To Cancel</h3><input type="text" name="date" required><br><br>

<button>submit</button>

</form>

<?php

if ($_SERVER['REQUEST_METHOD'] == 'POST') {

$email=$_SESSION["email"];

$bid=$_POST["bookid"];

$date=$_POST["date"];

$con=mysqli_connect("localhost","root","","ground");

if(!$con)

{

    die('connection failed');

}
```

```

$query = "select * from book where bid='$bid' and email='$email' and slot_date='$date'";
$res=mysqli_query($con,$query);

if(mysqli_num_rows($res) > 0 )
{
    $q="delete from book where bid=$bid";
    $result=mysqli_query($con,$q);
    if($result == 1){
        header("location:dashboard.php");
    }
    else{
        echo '<h1>failed to cancel</h1>';
        echo '<br>';
    }
}
else{
    echo '<h1> enter a valid booking id and date </h1>';
    echo '<br>';
}
}
?>
</div>
<body>
</html>

```

10.SESSION.PHP

```
<?php
```

```
session_start();

if(!isset($_SESSION['email'])) {

    header('Location: logout.php');

}

?>
```

11.LOGOUT.PHP

```
<?php

session_start();

$_SESSION["email"] = "";

session_destroy();

header("Location: index.html");

?>
```

12.ADMIN.PHP

```
<html>

<?php

session_start()?>

<head>

<title>admin</title>

<link rel="stylesheet" href="style1.css" media="screen" type="text/css" />

</head>

<body>

<div class="login-page">

<h2>ADMIN LOGIN</h2>

<div class="form">
```



```

<form class="login-form" action="admin.php" method="post">
<input type="text" placeholder="email" name="email" required>
<input type="password" placeholder="password" name="password" required>
<button>login</button>
</form>
</div></div>
</form>
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
$email=$_POST["email"];
$password=$_POST["password"];
$con=mysqli_connect("localhost","root","","ground");
if(!$con){
    die('connecton failed');
}
else{
    $query="select * from admin where email='$email' and password='$password'";
    $result=mysqli_query($con,$query);
if(mysqli_num_rows($result) > 0){
    $_SESSION["email"]=$email;
header('location:admin_dashboard.php');
}
else{
    echo '<h1>invalid login credentials</h1>';
}
}}?>
</body>

```

```
</html>
```

13.ADMIN_DASHBOARD.PHP

```
<html>
```

```
<?php include 'session.php' ?>
```

```
<head>
```

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

```
<link rel="stylesheet" href="table.css" media="screen" type="text/css" />
```

```
</head>
```

```
<body>
```

```
<a href='logout.php'>logout</a>
```

```
<a href="payments.php">payments</a>
```

```
<a href="enable_book.php">enable or disable bookings</a>
```

```
<br><br>
```

```
<?php
```

```
$con=mysqli_connect("localhost","root","","ground");
```

```
if(!$con){
```

```
    die('connection failed');
```

```
}
```

```
else{
```

```
    $query="select book.bid,book.slot_date,
book.status,book.email,name,mobile,payment from user_info inner join book on
book.email=user_info.email order by slot_date ";
```

```
    $result=mysqli_query($con,$query);
```

```
    if(mysqli_num_rows($result) == 0)
```

```
    {
```

```
        echo '<h1>NO BOOKINGS YET</h1>';
```

```
}
```

```

else{
    echo '<h1> UPCOMING BOOKED SLOTS</h1>';
    echo '<div id="c">';
echo '<table id="customers">';
    echo '<th> booking id</th>';
echo '<th> slot_date</th>';
    echo '<th> status</th>';
        echo '<th> email</th>';
        echo '<th> name</th>';
        echo '<th> mobile</th>';
        echo '<th> payment</th>';
while($row = mysqli_fetch_assoc($result))
{
echo '<tr>';
echo '<td>';echo $row["bid"];echo '</td>';
    echo '<td>';echo $row["slot_date"];echo '</td>';
    echo '<td>';echo $row["status"];echo '</td>';
    echo '<td>';echo $row["email"];echo '</td>';
    echo '<td>';echo $row["name"];echo '</td>';
    echo '<td>';echo $row["mobile"];echo '</td>';
        echo '<td>';echo $row["payment"];echo '</td>';
    echo '</tr>';
}
echo '</table>';
echo '</div>';
    }
    echo '<h1>CANCELLED SLOTS</h1>';

```

```
$q="select cancel.bid,cancel.slot_date,cancel.email,name,mobile from user_info inner join  
cancel on cancel.email=user_info.email order by slot_date";
```

```
$res=mysqli_query($con,$q);
```

```
if(mysqli_num_rows($res) == 0)
```

```
{
```

```
    echo '<h1>NO BOOKINGS YET</h1>';
```

```
}
```

```
else{
```

```
    echo '<div id="d">';
```

```
echo '<table id="customers">';
```

```
echo '<th> booking id</th>';
```

```
echo '<th> slot_date</th>';
```

```
    echo '<th> email</th>';
```

```
    echo '<th> name</th>';
```

```
    echo '<th> mobile</th>';
```

```
while($row = mysqli_fetch_assoc($res))
```

```
{
```

```
echo '<tr>';
```

```
echo '<td>';echo $row["bid"];echo '</td>';
```

```
    echo '<td>';echo $row["slot_date"];echo '</td>';
```

```
    echo '<td>';echo $row["email"];echo '</td>';
```

```
    echo '<td>';echo $row["name"];echo '</td>';
```

```
    echo '<td>';echo $row["mobile"];echo '</td>';
```

```
    echo '</tr>';
```

```
}
```

```
echo '</table>';
```

```
echo '</div>';
```

```
}}
?>
</body>
</html>
```

14.ENABLE_BOOK.PHP

```
<html>

<?php include 'session.php' ?>

<head>

<link rel="stylesheet" href="style1.css" media="screen" type="text/css" />
<link rel="stylesheet" href="table.css" media="screen" type="text/css" />

</head>

<body>

<a href="logout.php">logout</a>

<a href="admin_dashboard.php">dashboard</a>

<div class="form">

<form action="enable_book.php" method="post">

<h3> ENABLE OR DISABLE BOOKINGS</h3>

<select name="mode" required>

<option>select</option>

<option value="enable">enable</option>

<option value="disable">disable</option>

</select>

PASSWORD <input type="password" name="password" placeholder="admin password"
required>

<button>submit</button>

<?php
```

```

if ($_SERVER['REQUEST_METHOD'] == 'POST') {

    $mode=$_POST["mode"];

    $password=$_POST["password"];

    $email=$_SESSION["email"];

    $con=mysqli_connect("localhost","root","","ground");

    if(!$con)

    {

        die('connection failed');

    }

    else{

$query="select * from admin where email='$email' and password='$password'";

$result=mysqli_query($con,$query);

if(mysqli_num_rows($result) == 1 ){

    $q="update admin set accept='$mode'";

    $res=mysqli_query($con,$q);

    if($res == 1)

    {

        echo '<h1>successful</h1>';

    }

    else {

        echo '<h1> failed</h1>';

    }

}

else{

    echo '<h1>wrong password</h1>';

}

}

?>

```

```
</div>

</body>

</html>
```

15.PAYMENTS.PHP

```
<html>

<?php include 'session.php' ?>

<head>

<link rel="stylesheet" href="style1.css" media="screen" type="text/css" />

<link rel="stylesheet" href="table.css" media="screen" type="text/css" />

</head>

<body>

<a href="logout.php">logout</a>

<a href="dashboard.php">dashboard</a>

<br><br>

<div class="payment">

<h2>PAYMENT</h2>

<div class="form">

    <form class="payment" action="payments.php" method="post">

        BOOKING ID <BR>

        <input type="text" placeholder="booking id" name="bid" required>

        <br><br>

        MODE OF PAYMENT

        <br>

        <select name="mode" required>

            <option>select</option>

            <option value="cash">CASH</option>
```

```

        <option value="card">CARD</option>

        <option value="UPI">UPI</option>

    </select>

    <BR>

    <button>submit</button>

</form>

</div>

</div>

<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $bid=$_POST["bid"];
    $mode=$_POST["mode"];
    $con=mysqli_connect("localhost","root","","ground");
    if(!$con)
    {
        die('connecton failed');
    }
    else{
        $query="select email,slot_date from book where NOT payment='paid' and bid='$bid'
";
        $result=mysqli_query($con,$query);
        if(mysqli_num_rows($result) > 0){
            while($row=mysqli_fetch_assoc($result))
            {
                $b_email=$row["email"];
                $date=$row["slot_date"];
            }
        }
    }
}

```



```

        echo $bid;

        echo $b_email;

        echo $date;

        echo $mode;

        $q="insert into payment values('$bid','$b_email','$date','$mode')";

        $res=mysqli_query($con,$q);

        if($res == 1){

            header("location:admin_dashboard.php");

        }

        else{

            echo '<h1>payment failed</h1>';

        }

    }

    echo '<h1>invalid booking id</h1>';

}

}}?>

</html>

```

16.STYLE.CSS

```

[class*="fontawesome-"]:before {

    font-family: 'FontAwesome', sans-serif;

}

* {

    -moz-box-sizing: border-box;box-sizing: border-box;

}

*:before, *:after {

    -moz-box-sizing: border-box;box-sizing: border-box;

```

```

}
body {
    background: #2c3338;color: #606468;font: 87.5%/1.5em 'Open Sans', sans-serif;margin: 0;
}
a {
    color: #eee;text-decoration: none;
}
a:hover {
    text-decoration: underline;
}
input {
    border: none;font-family: 'Open Sans', Arial, sans-serif;
font-size: 14px;line-height: 1.5em;padding: 0;-webkit-appearance: none;
}
p {
    line-height: 1.5em;
}
.clearfix {
    *zoom: 1;
}
.clearfix:before, .clearfix:after {
    content: ' ';display: table;
}
.clearfix:after {
    clear: both;
}
.container {

```

```

    left: 50%;position: fixed;top: 50%;

    -webkit-transform: translate(-50%, -50%);

-ms-transform: translate(-50%, -50%);transform: translate(-50%, -50%);
}

#login {
    width: 280px;
}

#login form span {
    background-color: #363b41;border-radius: 3px 0px 0px 3px;color: #606468;

    display: block;float: left;height: 50px;line-height: 50px;text-align: center;width: 50px;
}

#login form input {
    height: 50px;
}

#login form input[type="text"], input[type="password"],input[type="tel"] {
    background-color: #f3f3f3;border-radius: 0px 3px 3px 0px;

    color: #1e1c1c;margin-bottom: 1em;padding: 0 16px;width: 230px;
}

#login form input[type="submit"] {
    border-radius: 3px;-moz-border-radius: 3px;-webkit-border-radius: 3px;

    background-color: #ea4c88;color: #eee;

    font-weight: bold;margin-bottom: 2em;text-transform: uppercase;width: 280px;
}

#login form input[type="submit"]:hover {
    background-color: #d44179;
}

#login > p {

```

```

    text-align: center;
}
#login > p span {
    padding-left: 5px;
}

```

17.STYLE1.CSS

```

.login-page {
    width: 360px;padding: 8% 0 0;margin: auto;
}

.form {
    position: relative;z-index: 1;background: #FFFFFF;
    max-width: 360px;margin: 0 auto 100px;padding: 45px;text-align: center;
    box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}

.form input {
    font-family: "Roboto", sans-serif;outline: 0;background: #f2f2f2;width: 100%;
    border: 0;margin: 0 0 15px;padding: 15px;box-sizing: border-box;font-size: 14px;
}

.form select {
    font-family: "Roboto", sans-serif;outline: 0;
    background: #f2f2f2;width: 100%;border: 0;margin: 0 0 15px;
    padding: 15px;box-sizing: border-box;font-size: 14px;color:blue
}

.form button {
    font-family: "Roboto", sans-serif;text-transform: uppercase;
    outline: 0;background: #4CAF50;

```

```

width: 100%;border: 0;padding: 15px;color: #FFFFFF;font-size: 14px;
-webkit-transition: all 0.3 ease;transition: all 0.3 ease;cursor: pointer;
}

.form button:hover,.form button:active,.form button:focus {
    background: #43A047;
}

.form .message {
    margin: 15px 0 0;color: #b3b3b3;font-size: 12px;
}

.form .message a {
    color: #4CAF50;text-decoration: none;
}

.form .register-form {
    display: none;
}

.container {
    position: relative;z-index: 1;max-width: 300px;margin: 0 auto;
}

.container:before, .container:after {
    content: "";display: block;clear: both;
}

.container .info {
    margin: 50px auto;text-align: center;
}

.container .info h1 {
    margin: 0 0 15px;padding: 0;
    font-size: 36px;font-weight: 300;color: #1a1a1a;
}

```

```

}

.container .info span {
    color: #4d4d4d;font-size: 12px;
}

.container .info span a {
    color: #000000;text-decoration: none;
}

.container .info span .fa {
    color: #EF3B3A;
}

h1 {
    text-align:center;color:red;
}

h2 {
    text-align:center ;color:green;
}

```

18.TABLE.CSS

```

a:link, a:visited {
    background-color: #f44336;color: white;padding: 14px 25px;text-align: center;
    text-decoration: none;display: inline-block;float:right;
}

a:hover, a:active {
    background-color: green;
}

.c {
    float:left;

```

```

}
h1{
    text-align:center;
    color:red
}
h2{
    text-align:center;color:red;
}
#customers {
    font-family: Arial, Helvetica, sans-serif;border-collapse: collapse; width: 50%;
    margin-left:auto;margin-right:auto;
}
#customers td, #customers th {
    border: 1px solid #ddd;padding: 8px;
}
#customers tr:nth-child(even){background-color: #f2f2f2;}
#customers tr:hover {background-color: #ddd;}
#customers th {
    padding-top: 12px;padding-bottom: 12px;text-align: left;
background-color: #4CAF50;color: white;
}

```

6. CONCLUSION

This online cricket ground booking system helps cricket players to book the cricket ground in advance. User can simply check whether the ground is available for a particular date or not. User can book or cancel a slot based on their choice. It helps the ground authorities to manage their ground in a systematic way. The authorities can prepare the ground in a better way based on the bookings they receive.

REFERENCES:

1. Html,css code available at <https://www.w3schools.com/html/>
2. js code available at <https://www.w3schools.com/js/DEFAULT.asp>
3. Php code available at <https://www.javatpoint.com/java-tutorial>
- 4.Dbms Queries from Php My Admin <http://localhost/phpmyadmin/sql.php>