VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018

DBMS MINI PROJECT

REPORT ON

"IPL MANAGEMENT SYSTEM"

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

BACHELOR OF ENGINEERING

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Submitted by

D DEEPTHI SWARUPA(1KG21AD004)
K LASYA CHOWDARY(1KG21AD020)
SHANTHAREDDY(1KG21AD044)
SOUJANYA N(1KG21AD047)

Under the Guidance of

Mrs. Madhusmitha mishra
Assistant Professor
Department of AI&DS
K.S.S.E.M

Department of Computer Science & Engineering
K. S. SCHOOL OF ENGINEERING AND MANAGEMENT
#15, Mallasandra, off. Kanakapura Road, Bengaluru – 560109

2023-2024

K. S. SCHOOL OF ENGINEERING AND MANAGEMENT
BENGALURU - 560109

Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that the **DBMS MINI PROJECT** entitled **"IPL MANAGEMENT SYSTEM" presented by Ms.D DEEPTHI SWARUPA,USN:1KG21AD004,Ms.K LASYA CHOWDARY,USN:1KG21ADO20,Mr.SHANTHAREDDY,USN:1KG21ADO44,Ms.SOUJANY A N,USN:1KG21AD047** of **V semester** in partial fulfillment of the award of **Bachelor of Engineering** in **Artificial intelligence and data science** in **Visvesvaraya Technological University**, Belagavi during the academic year **2023-2024**. The **DBMS MINI PROJECT** has been approved as it satisfies the academic requirements in respect of **DBMS Mini Project(21CSL55)** prescribed for the Bachelor of Engineering degree.

Signature of the coordinator

Mrs. K. Padma Priya
Asst Professor, AI & DS

Signature of the Guide

Mrs.Madhusmitha mishra
Asst Professor, AI & DS

Signature of the HOD

Mr. Manjunath T K
Associate Prof. & Head, AI & DS

Signature of the Principal

Dr. K . Rama Narasimha
Principal /Director

Name of the Student:	**USN:**
D Deepthi Swarupa	**1KG21AD004**
K Lasya Chowdary	**1KG21AD020**
Shantharedy	**1KG21AD044**
Soujanya N	**1KG21AD047**

Name of the examiners

1.
2.

Signature with date

I

ACKNOWLEDGEMENT

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

We would like to extend my gratitude to the **MANAGEMENT, KAMMAVARI SANGHAM**, Bengaluru, for providing all the facilities to present the Database Application Mini Project.

We would like to extend my gratitude to **Dr. K. RAMA NARASIMHA,** Principal /Director, K. S. School of Engineering and Management, Bengaluru, for facilitating me to present the Database Application Mini Project.

We thank **Mr. MANJUNATH TK,** Associate Professor and Head, Department of Artificial Intelligence And Data Science, K. S. School of Engineering and Management, Bengaluru, for his encouragement.

We would like to thank our mini project coordinator **Mrs. K. Padma Priya**, Assistant professor and our project guide **Mrs.Madhusmitha mishra**, Assistant Professor, Department of Artificial Intelligence & Data Science, K. S. School of Engineering and Management, Bengaluru, for their constant guidance and inputs.

We would like to thank all the **Teaching** Staff and **Non-Teaching** Staff of the college for their cooperation.

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

Name of the student Signatures

D DEEPTHI SWARUPA
K LASYA CHOWDARY
SHANTHAREDDY
SOUJANYA N

ABSTRACT

IPL Database Management is a cricket scheduling-based application exclusively for the game of cricket. TheApplication features schedules, information about teams, about captains, records of batting and bowling, creating new schedules, can search about players, it displays rank tables for teams and players.

The admin has all authorities to make changes for the database so admin can add players, can add schedules,can add stadiums, and have permission to remove of them from the database. It features searching for playersinvolved in the game and retrieving the players of the match by selecting the match number. Also, they can fetch the schedules with their venue and squad available by the team, players selected for the current match. Admin can also authority to update the rating of the teams and players runs and wickets and other match particulars in this database.

The user's login window also features creating an account, players search for players information, getting future match particulars, rankings, cricket boards, stadiums, schedules, and their venues. Can fetch the schedules with their venue and squad available by the team, players selected for the current match.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 OVERVIEW

IPL Database Management is a user-friendly Application which is based on HTML and CSS which helps members to schedule and manage various Cricket Matches and allows us to manage the records of various players. The application uses HTML and CSS as a front end for interacting with the user and PHP for connection. At the backend we used MySQL for database.

1.2 PROBLEM STATEMENT

The Indian Premier League (IPL) is a cricket tournament where different teams compete. A database system is required to manage the details of players, teams, matches, and scores. The system should allow for player and team management, match scheduling, and score tracking. It should also enable the generation of reports,such as player statistics, team performance, and match summaries. The system must support CRUD operationsfor players, teams, and matches. It should be user-friendly, with an intuitive interface for easy navigation. Thedatabase should be secure, with proper authentication and authorization mechanisms. It should also be scalableto accommodate future growth and changes. Finally, the system should be reliable, with backup and recoverymechanisms in place to prevent data loss.

1.3 DATABASE MANAGEMENT SYSTEM

The backbone of the IPL database Management System lies in its robust DBMS. Expanding on this, the choiceof a specific database system is discussed, taking into consideration scalability, data integrity, and efficiency.The role of the DBMS in organizing and safeguarding vast amounts of member-related information is explored, highlighting its pivotal position in the system's architecture.

1.4 SQL

SQL is used to communicate with a database. According to ANSI (American National Standards Institute), itis the standard language for relational database management systems [4]. SQL statements are used to performtasks such as updating data on a database or retrieving data from a database. Some common relational databasemanagement systems that use SQL are Oracle, Sybase, Microsoft SQL Server, Access, Ingre

1.5 HTML/CSS/PHP/Visual Studio

The visual appeal and functionality of the system are intricately woven together through HTML, CSS, PHP,and Visual Studio. This section elucidates how these technologies collaborate to deliver an intuitive and responsive user interface. The significance of a visually appealing and user-friendly design in enhancing member engagement is underscored.

HTML

Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages suchas JavaScript.

CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written ina markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript Functional Modules

PHP

PHP is a server-side scripting language widely used for web development. Known for its simplicity and versatility, it seamlessly integrates with HTML to create dynamic websites. Supported by frameworks like Laravel, PHP is open source and benefits from fast execution, making it a popular choice for developers.

Visual studio code

Visual studio code is a code editor redefined and optimized for building and debugging modern web and cloudapplications. It is a free of cost user friendly platform.

XAMPP Connection

Seamless integration is key, and XAMPP serves as the linchpin connecting the database and the user interface.Delving into the XAMPP connection, this section discusses how it ensures secure and efficient data transmission, guaranteeing that real-time updates and interactions between members and the system occur seamlessly.

Chapter2

REQUIREMENT 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 iseasy to use and maintain is the requisite.

2.2 SPECIFIC REQUIREMENTS

The specific requirements of the Hospital Management System are stated as follows:

2.3 SOFTWARE REQUIREMENTS

- IDE - Visual Studio Code
- Web Browser – Firefox 50 or later, Google Chrome – 60 or later
- Database support - MySQL5.7
    - MySQL Server 5.7
    - MySQL Workbench
- Operating system – ANYOS (Recommended: windows8 or 10)
- Server deployment - XAMPP server
- Coding languages-PHP, HTML, CSS

2.4 HARDWARE REQUIREMENTS

- Processor – Pentium IV or above
- RAM – 2 GB or more
- Hard disk – 3 GB or more
- Monitor – VGA of 1024x768 screen resolution
- Keyboard and Mouse.

2.5 TECHNOLOGY

- HTML is used for the front-end design. It provides a means to structure text-based information in a document. It allows users to produce web pages that include text, graphics and hyperlinks.
- CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document.
- 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 connects with the front end by generating dynamic HTML, CSS, and JavaScript content based on user requests and data from databases. It handles form submissions, processes user input, interacts with databases to fetch or store data, manages user sessions, and provides API endpoints for communication between the frontend and backend of web applications.
- Overall, PHP serves as the intermediary between the user's browser and the server, enabling dynamic and interactive web experiences.

Chapter 3

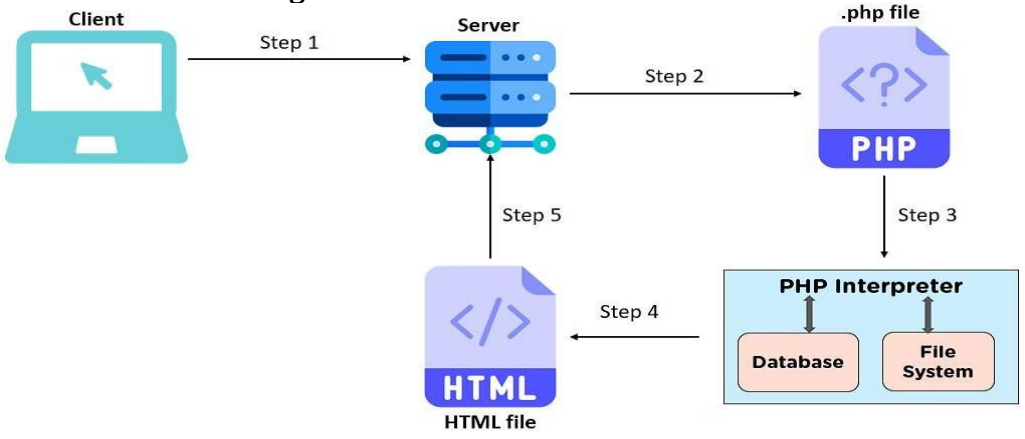
DETAILED DESIGN

3.1 SYSTEM DESIGN

The PHP system design process unfolds as a client's request for a PHP file triggers a sequence of events orchestrated by the server and PHP interpreter. When a user interacts with a web application, their browser sends an HTTP request to the server, targeting a PHP file. Recognizing the PHP extension, the server delegates the file's processing to the PHP interpreter, which meticulously parses the PHP script, executing PHP code and dynamically generating HTML content. This content crafted based on user inputs, database queries, and conditional logic, forms the backbone of dynamic web pages, seamlessly integrating with CSS and JavaScript to enhance interactivity and visual appeal.

The following diagram shows the PHP architecture.

Fig. 3.1: PHP Architecture



Upon completion of PHP script execution and HTML generation, the resulting content, alongwith associated resources, is compiled into an HTTP response by the server and dispatched back to the client. The client's browser then parses and renders the HTML content, transforming abstract markup into tangible visual elements displayed on the user's screen. Modern PHP applications often incorporate additional layers of complexity, such as caching, session management, and security protocols, to optimize performance, reliability, and user security, ensuring seamless and engaging web experiences for users worldwide.

3.2 ENTITY RELATIONSHIP DIAGRAM

An entity–relationship model is usually the result of systematic analysis to define anddescribe what is important to process 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.

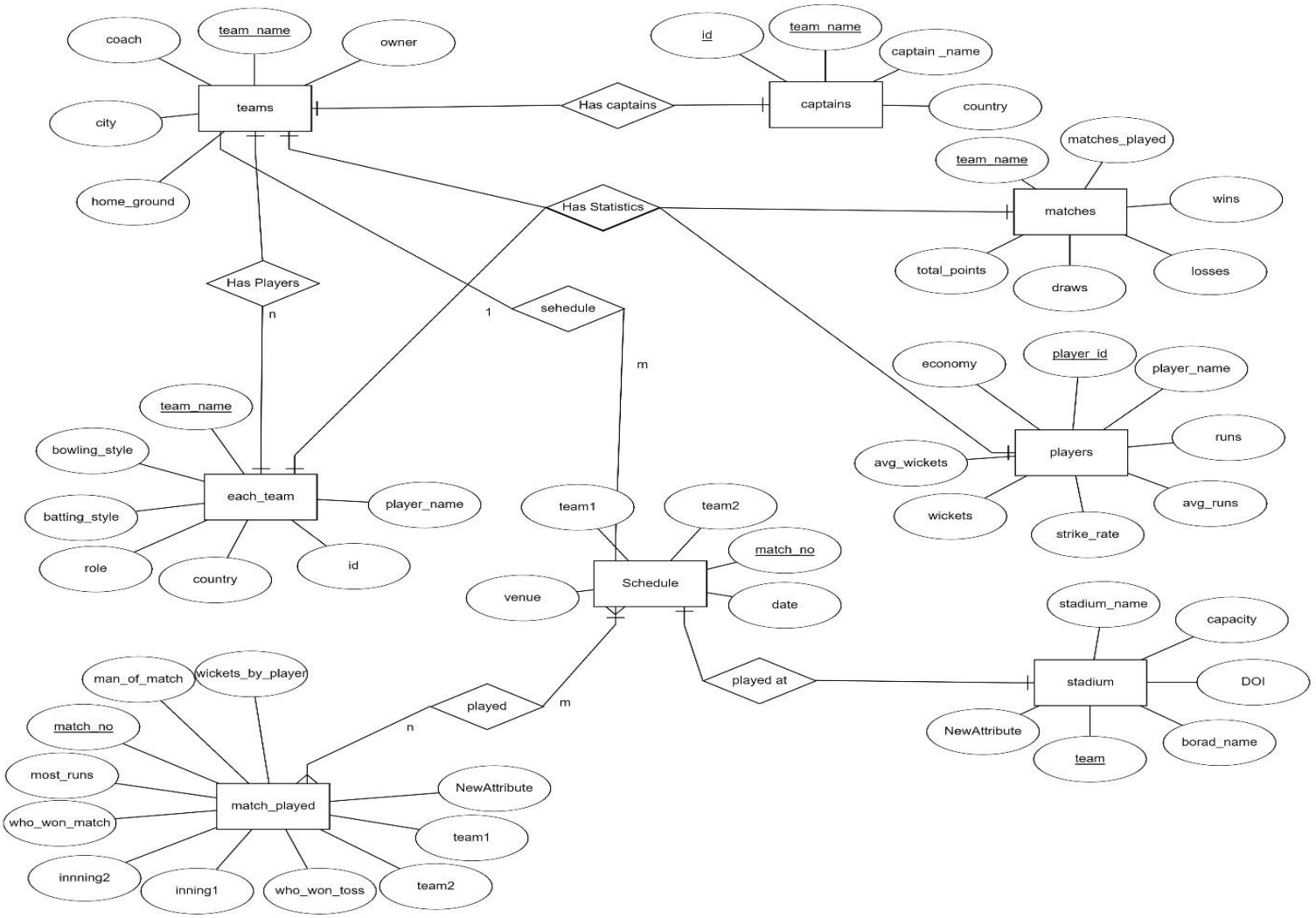


Fig 3.2: ER diagram of IPL Management System

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 inthe database. When a primary key is referenced in another table in the database, it is called aforeign 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.

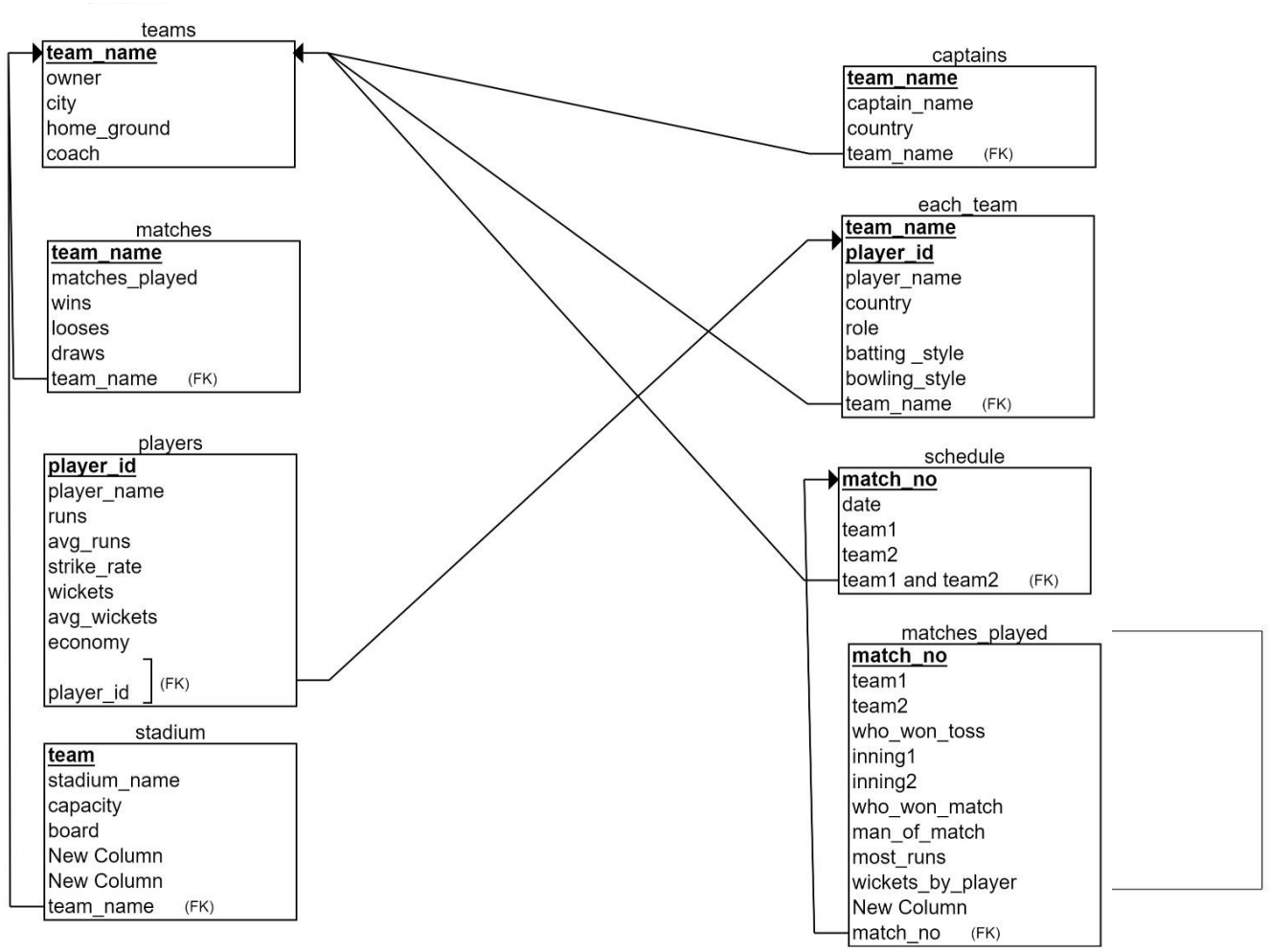


Fig. 3.3: Schema diagram of IPL Management System

3.4 DESCRIPTION OF TABLES

The database consists of four tables:

1. Admin: It stores the admin details.
   - Name: Name of the admin.
   - Password: password set by the admin.

2. User: It stores the user details.
   - Username: Name of the user.
   - Password: password set by the user.

3. Match Schedule: It stores the match schedule details.
   - Match Number: Number of the match.
   - Date: Date of the Match.
   - Team 1: Name of the team 1.
   - Team 2: Name of the team 2.
   - Venue: Venue of the Match.

4. Match Information: It stores the match information.
   - Team Name: Name of the team.
   - Matches: Number of matches played.
   - Wins: No. of matches win.
   - Winning percentage: Percentage of winning matches.
   - Lost: No. of matches lost.
   - Tied: No. of matches tied.
   - Total Points: points gained in the match.

Chapter 4

IMPLEMENTATION

4.1 MODULES AND THEIR ROLES

Index Page

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>IPL Database Management</title>

<style>

body {

font-family: Arial, Helvetica, sans-serif;
background-repeat: no-repeat; background-
size: cover;
margin: 0;

padding: 0;

}

header {

text-align: center;
margin-top: 125px;
}

h2 {

font-size: 50px;

}

button {
```

```
        background-color:
    #4CAF50;color: white;
        padding: 14px
    20px;margin: 8px 0;
    border:  none;
    cursor: pointer;
    width: 10%;
      }

        . button-
    container {
    margin- top:
    25px;display:
    flex;
        justify-content: center;

      }

      button a {

        text-decoration:
    none;color: white;
      }

   </style>

</head>

<body style="background-image: url('ipllogin.jpg');">



   <header>

      <h2><i><b>IPL Database Management</b></i></h2>

   </header>

   <div class="button-container">

      <button><a href="admin.html">ADMIN LOGIN</a></button>

      <button><a href="login.html">USER LOGIN</a></button>
```

```
</div>
</body>

    </html>
```

Admin page

```
    <!DOCTYPE html>

    <html lang="en">

    <head>

        <meta charset="UTF-8">

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

        <title>Admin Login</title>

        <style>

            body {

                font-family: Arial, Helvetica, sans-serif;

                background: url('th.jpeg') no-repeat center center
            fixed;background-size: cover;
            }



            form {

                border: 0px solid #f1f1f1;

            }



            h2 {

                font-size:
            50px; text-align:
            center;
            }
```

```
    input[type=text], input[type=password] {
        width: 20%;
    padding:    12px    20px;
        margin:      8px      0;
        display:   inline-block;
        border: 1px solid #ccc;
    box-sizing: border-box;

}



    button,     button1     {
        width:            10%;
        padding: 14px 20px;
        margin:     8px     0;
        border:           none;
        cursor: pointer;
    text-align: center;

}



button {

    background-color:    #4CAF50;
        color: white;
}



button1 {

    background-color:    #E6BAA2;
        color: white;
}
```

```
   button:hover,   button1:hover
     {opacity: 0.8;
}



   .cancelbtn    {
     width: auto;
   padding:       10px       18px;
     background-color: #b43d35;
}



.imgcontainer {

   text-align:            center;
     margin: 24px 0 12px 0;
}



   img.avatar  {
     width: 0%;
   border-radius: 0%;

}



   .container        {
     padding: 16px;
}


   span.psw    {
     float: right;
```

```
        padding-top: 16px;

        }



        @media screen and (max-width: 300px)
{span.psw {
        display:     block;
float: none;
        }



        .cancelbtn       {
width: 100%;
        }

        }

        </style>

        </head>



        <body style="background-image: url('ipllogin.jpg');">

        <h2>Admin Login</h2>

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

        <div class="imgcontainer"></div>

        <div class="container" style="text-align:center;">

        <label for="uname"><b>Name</b></label>

        <input type="text" placeholder="Enter Name" name="uname" required><br><br>

        <label for="psw"><b>Password</b></label>

        <input type="password" placeholder="Enter password" name="psw" required><br><br>

        <button          type="submit">Login</button><br><button1          style="float:right          ;"><a
href="index.html">Back</a></button1>
```

```
</div>
</form>
    </body>
    </html>
```

Admin php page

```
<?php
    session_start();
    $con = mysqli_connect("localhost", "root","",  "ipl") or die(mysqli_error($con));
    $username = $_POST['uname'];
    $password = $_POST['psw'];

    $check_u = "select * from admin where username ='$username'";
    $res_u = mysqli_query($con,$check_u) or die(mysqli_error($con));
    $check_p = "select * from admin where username = '$username' and password = '$password'";
    $res_p = mysqli_query($con,$check_p) or die(mysqli_error($con));

    if(mysqli_num_rows($res_u)==0)
    {
      echo "<script type='text/javascript'>alert('Incorrect username!!');</script>";header("refresh: 0.01; url=admin.html");
    }
    if(mysqli_num_rows($res_p)==0)
    {
      echo "<script type='text/javascript'>alert('Incorrect password!!');</script>";header("refresh: 0.01; url=admin.html");
```

```
}

else

{

echo "<script type='text/javascript'>alert('Logged in successfully!!');</script>";
header("refresh: 0.01; url=admin1st.html");
}

?>
```

User page

```
<!DOCTYPE html>

<html>

<head>

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

<style>

 body {font-family: Arial, Helvetica, sans-serif;background-repeat: no repeat;
  background-size: 100%;
  }

  form {border: 0px solid
  #f1f1f1;}h2{font-size: 50px;
  text-align:center; }

input[type=text], input[type=password] {
  width: 20%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
```

```
box-sizing: border-box;

}

button {

background-color: #15abb3;
color: white;
padding: 14px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
width: 10%;
}



button:hover {
opacity: 0.8;
}



.cancelbtn {
width: auto;
padding: 10px 18px;
background-color: #dc6d13;
}



.imgcontainer {

text-align: center;
margin: 24px 0 12px 0;
}
```

```
   img.avatar
   { width:
   40%;
   border-radius: 50%;

}



   .container {
   padding: 16px;
}

</style>

</head>

<body background="ipllogin.jpg">

<h2><i>Indian Premier League</h2>

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

  <div class="imgcontainer">

  </div>

  <div class="container" style="text-align:center;">


    <label for="uname"><b>Username</b></label>

    <input type="text" placeholder="Enter Username" name="uname" required><br><br>


    <label for="psw"><b>Password</b></label>

    <input type="password" placeholder="Enter Password" name="psw" required><br><br>


    <button type="submit">Login</button>

<p style="text-align:center;"> <a href="createpage.html" ><b>CREATE AN
```

```
 ACCOUNT</b></a></p><button style="float:right ;" ><a href="index.html">Back</a></button>
</div>

 </form>

</body>

</html>
```

Create new login

```
<!DOCTYPE html>

<html lang="en">

<head>

   <meta charset="UTF-8">

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

   <style>

      body {

         font-family: Arial, Helvetica, sans-serif;
         background-repeat:          no-repeat;
         background-size: 100%;
         margin: 0;

         padding: 0;

         box-sizing: border-box;

      }



      form {

         border: 0px solid #f1f1f1;

      }

      h1 {
```

```
    font-size:    50px;
    text-align: center;
    margin-bottom: 20px;

}



label {

    display: inline-block;
    width: 20%;
    text-align:    right;
    margin-right: 2%;
}



input[type="text"],
input[type="password"],
input[type="email"] {
    width:40%;  padding:
    12px;
    margin-bottom:  8px;
    display: inline-block;
    border: 1px solid #ccc;
    box-sizing: border-box;
}



button {

    background-color: #4CAF50;
    color: white;
    padding: 14px 20px;
```

```
            margin: 8px 0;
            border:  none;
            cursor: pointer;
            width: 10%;
            text-align: center;

        }



        button:hover {
            opacity: 0.8;
        }



        .container {

            text-align: center;
            padding: 16px;
        }

    </style>

</head>

<body style="background: url('createpage.png') no-repeat center center fixed; background-size: cover;">



<h1><b>CREATE AN ACCOUNT</b></h1>

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

    <div class="container">

        <label for="uname"><b>NEW USER:</b></label>

        <input type="text" placeholder="Enter name" name="uname" required><br>

<label for="email"><b>ENTER EMAIL:</b></label>

        <input type="email" placeholder="Enter email" name="email" required><br>
```

```
        <label for="psw"><b>PASSWORD:</b></label>

        <input type="password" placeholder="Enter password" name="psw" required><br>


        <label for="conpsw"><b>CONFIRM PASSWORD:</b></label>

        <input type="password" placeholder="Enter password" name="conpsw" required><br>


        <button type="submit">SUBMIT</button><br>

        <button>Already have an Account<a href="login.html">Login</a></button>
</div>

</form>

</body>

    </html>
```

Home page

```
    <!DOCTYPE html>

    <html>

    <head>
     <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Cricket Home</title>

        <style>
            body {
          font-family: Arial, sans-serif;
   background-color: #f4f4f4; margin:0;

                padding: 0;
                padding:0;
```

```
    text-align: center;

}

header {

    background-color: #333;
    color: white;
    padding: 10px;

}

nav {

    background-color: #4caf50;
    color: white;
    padding: 10px;

}

section             {
    padding: 20px;
}



h2 {

    color: #333;

}

a {

    color: #e0eae1;

    text-decoration: none;
    font-weight: bold;
}

a:hover {

    text-decoration: underline;

}
```

```
        button {

            background-color: #189827;
            color: #fff;
            padding: 10px 20px;
            border: none;
            text-decoration: none;
            cursor: pointer;
        }

    </style>

</head>

<body>

    <header>

        <h1>IPL World</h1>

    </header>

<nav>

        <a href="ipl about.html">About</a> |

        <a href="teams.php">teams</a> |

        <a href="matches.php">Matches</a> |

        <a href="hi.php">Captains</a> |

        <a href="players.php">Players</a>|

        <a href="stadium.php">Stadium</a>|

        <a href="each_team.html"> franchise</a>|

        <a href="schedule.php">schedule </a>|

        <a href="matchplayed.php">Matcheplayed </a>|

        <button><a href="index.html"> logout</a></button>

</nav>

    <section id="about">
```

```
        <h2>About Cricket</h2>
        <p>
    </section>
     <section>
        <h2>IPL News</h2>

        </p>
    </section>
</body>
</html>
```

About page

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>IPL Page</title>

    <style>

        body {

            background-image: url('img/background.jpg'); /* Set your background image path */
            background-size: cover;
            margin: 0;

            font-family: Arial, sans-serif;

        }



        button {

            background-color: #3498db;
            color: #fff;
            padding: 10px 20px;
            border: none;
            text-decoration: none;
            cursor: pointer;
        }



        .header {
            background-color: #2c3e50;
            color: #ecf0f1;
```

```
            padding:     20px;
            text-align: center;
        }



        h1 {

            color: #3498db;

        }



        p {

            text-align: center;
            line-height: 1.6;
        }

    </style>

</head>

<body>



    <button><a href="admin1st.html">Back</a></button>



    <div class="header">

        <h1><u><i>INDIAN PREMIER LEAGUE</i></u></h1>

</div>



    <h1 align="center">ABOUT</h1>

  </body>

  </html>
```

Match page

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Match Information</title>

    <style>

        body {

            font-family: Arial, sans-serif;
            background-color:    #f0f0f0;
            margin: 0;
            padding: 0;

        }



        h2 {

            text-align: center;
            color: #333;
        }
```

```
table {

    width:          80%;
    margin: 20px auto;
    border-collapse: collapse;

}



th, td {

    border: 1px solid #ddd;
    padding: 8px;
    text-align: left;

}



th {

    background-color: #4caf50;
    color: white;
}



form {

    max-width:      400px;
    margin:   20px   auto;
    padding:          20px;
    background-color: #fff;
    border-radius: 5px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}
```

```
    label {

        display:          block;
        margin-bottom: 8px;
    }



    input,   select   {
        width:  100%;
        padding: 8px;
        margin-bottom:    16px;
        box-sizing: border-box;
    }



    input[type="submit"]          {
        background-color: #4caf50;
        color: #fff;
        cursor: pointer;

    }

     .back-btn                  {
   background-color: #333;
   color: #fff;
   padding: 10px;

   text-decoration: none;
   border-radius:     5px;
   display:  inline-block;
   margin-top: 10px;
     }

</style>
```

```
</head>

<body>



<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "ipl";



$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);

}



// Update match information

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

    if (isset($_POST["action"])) {

        if ($_POST["action"] == "update_match" && isset($_POST["team_name"])) {

            $team_name = $_POST["team_name"];

            $matches = $_POST["matches"];

            $wins = $_POST["wins"];

            $lost = $_POST["lost"];

            $tied = $_POST["tied"];

            // Calculate winning percentage and total points based on the provided formula
            $winning_percentage = ($wins / $matches) * 100;
             $total_points = ($wins * 2) + $tied;
```

```
$sql = "UPDATE matches SET matches=$matches, wins=$wins, winning_percentage=$winning_percentage, lost=$lost, tied=$tied, total_points=$total_points WHERE team_name = '$team_name'";


if ($conn->query($sql) === TRUE) {

echo "<p>Match information updated successfully!</p>";

} else {

echo "<p>Error updating match information: " . $conn->error . "</p>";

}

}

}

}


// Display match information for all teams

$sql = "SELECT * FROM matches";

$result = $conn->query($sql);


echo "<h2>Match Information for All Teams</h2>";echo "<table>";
echo "<tr><th>Team Name</th><th>Matches</th><th>Wins</th><th>Winning Percentage (%)</th><th>Lost</th><th>Tied</th><th>Total Points</th><th>Action</th></tr>";


while ($row = $result->fetch_assoc()) {echo "<tr>";
echo "<td>{$row["team_name"]}</td>";
echo "<td>{$row["matches"]}</td>";
echo "<td>{$row["wins"]}</td>";
echo "<td>{$row["winning_percentage"]}</td>";
```

```
    echo "<td>{$row["lost"]}</td>";
    echo "<td>{$row["tied"]}</td>";
    echo "<td>{$row["total_points"]}</td>";
    echo "<td>
            <form method='post' action='{$_SERVER["PHP_SELF"]}'>

                <input type='hidden' name='team_name' value='{$row["team_name"]}'>

                <input type='hidden' name='action' value='update_match'>

                <label for='matches'>Matches:</label>

                <input type='number' name='matches' value='{$row["matches"]}' required>

                <label for='wins'>Wins:</label>

                <input type='number' name='wins' value='{$row["wins"]}' required>

                <label for='lost'>Lost:</label>

                <input type='number' name='lost' value='{$row["lost"]}' required>

                <label for='tied'>Tied:</label>

                <input type='number' name='tied' value='{$row["tied"]}' required>

                <input type='submit' value='Update'>

            </form>

        </td>";
    echo "</tr>";
}



    echo "</table>";

    // Back Button

    echo "<a href='admin1st.html' class='back-btn'>Back to Admin Home</a>";

// Close the connection

    $conn->close();
```

```
?>

</body>

</html>
```

Captains page

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>List of Cricket Captains</title>

<style>
 body {
 font-family:  Arial,  sans-serif;
 background-color:     #f0f0f0;
 margin: 0;
 padding: 0;

 }



 h2 {

 text-align:   center;
 color: #333;
 }



 table {
 width:   80%;   margin:
 20px auto;
```

```
        border-collapse: collapse;
        background-color: #fff;
    }



    th,

    td {

        border: 1px solid #ddd;
        padding: 10px;
        text-align: left;

    }



    th {

        background-color: #4caf50;
        color: white;
    }



  form {

    max-width:      400px;
    margin:   20px   auto;
    padding:          20px;
    background-color: #fff;
    border-radius: 5px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

  }



  label {

</body>

  <?php

  $servername = "localhost";
```

```
$username = "root";

$password = "";

$dbname = "ipl";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);

}



// Update captain information

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

   if (isset($_POST["action"])) {

      if ($_POST["action"] == "update_confirm" && isset($_POST["update_captain_id"])) {

         $captain_id = $_POST["update_captain_id"];

         $new_name = $_POST["new_name"];

         $new_country = $_POST["new_country"];



         $sql = "UPDATE captains SET name='$new_name', country='$new_country' WHERE id =
$captain_id";

 if ($conn->query($sql) === TRUE) {

            echo "<p>Captain updated successfully!</p>";

         } else {

            echo "<p>Error updating captain: " . $conn->error . "</p>";
     display: block;
```

```
  margin-bottom: 8px;

  }



input, select {
  width:
  100%;
  padding:
  8px;
  margin-bottom:  16px;
  box-sizing:  border-
  box;
  }



input[type="submit"]  {
  background-color:
  #4caf50;color: #fff;
  cursor: pointer;

  }



  .back-btn {

  background-color:
  #333;color: #fff;
  padding: 10px;

  text-decoration:
  none;  border-radius:
  5px;  display:  inline-
  block;  margin-top:
  10px;
  }

  </style>

  </head
```

```
        }
    } elseif ($_POST["action"] == "add_captain") {
        $captain_name = $_POST["captain_name"];
        $captain_team = $_POST["captain_team"];
        $captain_country = $_POST["captain_country"];

        $sql = "INSERT INTO captains (name, team, country) VALUES ('$captain_name', '$captain_team',
'$captain_country')";

        if ($conn->query($sql) === TRUE) {
            echo "<p>New captain added successfully!</p>";
        } else {
            echo "<p>Error adding captain: " . $conn->error . "</p>";
        }
    }
  }
}

// Display list of captains
$sql = "SELECT * FROM captains";
$result = $conn->query($sql);

echo "<h2>List of IPL Captains</h2>";
echo "<table>";
echo "<table>";
    echo "<tr><th> Name</th><th>Team</th><th>Country</th></tr>";
while ($row = $result->fetch_assoc()) {
```

```
    echo "<tr>
    <td>{$row["name"]}</td>
    <td>{$row["team"]}</td>
    <td>{$row["country"]}<td>
    </tr>";
}
echo "</table>";

// Update Captain Form
echo "<form method='post' action='{$_SERVER["PHP_SELF"]}'>
        <h2>Update Captain</h2>
        <label for='update_captain_id'>Select Captain to Update:</label>
        <select name='update_captain_id'>";
$result = $conn->query("SELECT id, name FROM captains");
while ($row = $result->fetch_assoc()) {
    echo "<option value='{$row["id"]}'>{$row["name"]}</option>";
}
echo "</select><br>
        <label for='new_name'>New Name:</label>
        <input type='text' name='new_name' required><br>
        <label for='new_country'>New Country:</label>
        <input type='text' name='new_country' required><br>
        <input type='hidden' name='action' value='update_confirm'>
        <input type='submit' value='Update Captain'>
      </form>";
```

```
// Add New Captain Form
echo "<form method='post' action='{$_SERVER["PHP_SELF"]}'>
        <h2>Add New Captain</h2>
        <label for='captain_name'>Name:</label>
        <input type='text' name='captain_name' required><br>
        <label for='captain_team'>Team:</label>
        <input type='text' name='captain_team' required><br>
        <label for='captain_country'>Country:</label>
        <input type='text' name='captain_country' required><br>
        <input type='hidden' name='action' value='add_captain'>
        <input type='submit' value='Add Captain'>
      </form>";

// Back Button
echo "<a href='admin1st.html' class='back-btn'>Back to Admin Home</a>";

// Close the connection
$conn->close();
?>
</body>
</html>
```

SQL Queries

Database: `ipl`

Admin table

```
CREATE TABLE `admin` (

 `username` varchar(50) NOT NULL,

  `password` varchar(8) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
INSERT INTO `admin` (`username`, `password`) VALUES
```

Captains table

```
CREATE TABLE `captains` (

  `id` int(8) NOT NULL,

  `name` varchar(50) NOT NULL,

  `team` varchar(60) NOT NULL,

  `country` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

INSERT INTO `captains` (`id`, `name`, `team`, `country`) VALUES
(20, 'Chiru', 'RCB', 'India'),
(22, 'Hardik Pandya', 'MI', 'India'),

(24, 'MS Dhoni', 'CSK', 'India'),

(28, 'Sanju Samson', 'RR', 'India'),

(29, 'Aiden Markram', 'SRH', 'South Africa'),
(30, 'Shikhar Dhawan', 'KXIP', 'India'),
(31, 'David Warner', 'DC', 'Australia'),

(32, 'Nitish Rana', 'KKR', 'India'),

(38, 'KL Rahul', 'LSG', 'India');
```

Each_team table

```
CREATE TABLE `each_team` (
  `id` int(10) NOT NULL,
  `team_name` varchar(100) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `player_name` varchar(100) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `role` varchar(40) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `country` varchar(100) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `batting_style` varchar(100) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `bowling_style` varchar(100) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Login table

```
CREATE TABLE `login` (
  `username` varchar(50) NOT NULL,
  `password` varchar(8) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```

Matches table

```
CREATE TABLE `matches` (
  `id` int(8) NOT NULL,
  `team_name` varchar(60) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `matches` int(20) NOT NULL,
  `wins` int(10) NOT NULL,
   `winning_percentage` decimal(5,2) NOT NULL,
  `lost` int(10) NOT NULL,
```

```
  `tied` int(10) NOT NULL,

  `total_points` int(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;



-- Triggers `matches`

--

DELIMITER $$

CREATE TRIGGER `calculate_winning_percentage` AFTER INSERT ON `matches` FOR EACH ROW
BEGIN
    DECLARE total_matches INT;
    DECLARE total_wins INT;
    DECLARE winning_percentage DECIMAL(5,2);



    -- Calculate total matches and wins

    SELECT COUNT(*), SUM(wins) INTO total_matches, total_wins
    FROM matches
    WHERE team_name = NEW.team_name;



    -- Calculate winning percentage
    IF total_matches > 0 THEN
        SET winning_percentage = (total_wins / total_matches) * 100;
    ELSE
        SET winning_percentage = 0;
    END IF;


    -- Update the winning percentage in the matches table
    UPDATE matches
```

```
 SET winning_percentage = winning_percentage
WHERE team_name = NEW.team_name;

END

$$
DELIMITER ;
```

Matches played table

```
CREATE TABLE `matches_played` (
  `match_no` int(11) NOT NULL,
  `team1` varchar(255) DEFAULT NULL,
  `team2` varchar(255) DEFAULT NULL,
  `who_won_toss` varchar(255) DEFAULT NULL,
  `inning1` varchar(255) DEFAULT NULL,
  `inning2` varchar(255) DEFAULT NULL,
  `who_won_match` varchar(255) DEFAULT NULL,
  `man_of_match` varchar(255) DEFAULT NULL,
  `most_runs` varchar(255) DEFAULT NULL,
  `wickets_by_player` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Players table

```
CREATE TABLE `players` (
  `id` int(10) NOT NULL,
  `name` varchar(50) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `team` varchar(100) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `type` varchar(100) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
```

```
  `matches` int(10) NOT NULL,
  `runs` int(10) NOT NULL,
  `average_runs` decimal(10,2) NOT NULL,
  `strike_rate` decimal(10,2) NOT NULL,
  `wickets` int(10) NOT NULL,
  `avg_wickets` decimal(10,2) NOT NULL,
  `economy` decimal(10,2) NOT NULL,
  `best_batting` int(20) NOT NULL,
  `best_bowling` varchar(20) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Schedule table

```
CREATE TABLE `schedule` (
  `match_number` int(11) NOT NULL,
  `match_date` date DEFAULT NULL,
  `team1` varchar(255) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `team2` varchar(255) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `venue` varchar(255) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Stadium table

```
CREATE TABLE `stadium` (
  `stadium_ name` varchar (100) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `capacity` int (20) NOT NULL,
  `DOI` date NOT NULL,
  `board_ name` varchar (100) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `team` varchar (50) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL
) ENGINE= Inno DB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Teams table

```
CREATE TABLE `teams` (
  `team_name` varchar(60) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `owner` varchar(200) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `city` varchar(200) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `home_ground` varchar(200) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL,
  `coach` varchar(200) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

CONSTRAINT

```
ADD CONSTRAINT `captains_ibfk_1` FOREIGN KEY (`team`) REFERENCES `teams` (`team_name`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `each_team`
  ADD CONSTRAINT `each_team_ibfk_1` FOREIGN KEY (`team_name`) REFERENCES `teams` (`team_name`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `matches`
  ADD CONSTRAINT `hi` FOREIGN KEY (`team_name`) REFERENCES `teams` (`team_name`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `matches_played`
  ADD CONSTRAINT `matches_played_ibfk_1` FOREIGN KEY (`match_no`) REFERENCES `schedule` (`match_number`);

ALTER TABLE `players`
  ADD CONSTRAINT `players_ibfk_1` FOREIGN KEY (`team`) REFERENCES `teams` (`team_name`) ON UPDATE CASCADE;
```

```
ALTER TABLE `schedule`

  ADD CONSTRAINT `schedule_ibfk_1` FOREIGN KEY (`team1`) REFERENCES `teams` (`team_name`) ON DELETE CASCADE ON UPDATE CASCADE,

  ADD CONSTRAINT `schedule_ibfk_2` FOREIGN KEY (`team2`) REFERENCES `teams` (`team_name`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `stadium`

  ADD CONSTRAINT `fk_each_team` FOREIGN KEY (`team`) REFERENCES `teams` (`team_name`);
COMMIT;
```

Module Import

```
<?php

  $db_server = "localhost";

  $db_user  = "root";

  $db_pass  = "";

  $db_name  = "prmsdb1";

  $conn = mysqli_connect($db_server,$db_user,$db_pass,$db_name);
  if(!$conn){
    die("Connection failed! :".mysqli_connect_error());

  }

?>
```

Triggers

```
DELIMITER $$

CREATE TRIGGER `calculate_winning_percentage` AFTER INSERT ON `matches` FOR EACH ROW BEGIN

    DECLARE total_matches INT;
    DECLARE total_wins INT;
```

```
    DECLARE winning_percentage DECIMAL(5,2);

    -- Calculate total matches and wins

    SELECT COUNT(*), SUM(wins) INTO total_matches, total_wins
FROM matches
    WHERE team_name = NEW.team_name;

    -- Calculate winning percentage IF
total_matches > 0 THEN
    SET winning_percentage = (total_wins / total_matches) * 100;ELSE
    SET winning_percentage = 0;END IF;
    -- Update the winning percentage in the matches table
UPDATE matches
    SET winning_percentage = winning_percentage
WHERE team_name = NEW.team_name;


 END

 $$ DELIMITER ;
```

4.2 RESULT

The resulting system can:

- Authenticate user credentials during login.
- Allow admin to insert, update details of players.
- Allow user to view details of players.

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 the whole software program at once, module testing recommends 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 implementation 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

- Only admin have the authority to insert, update and delete the data.
- In matches _ played we can insert, update the data but we can’t delete the data.
- We don’t have user database to track individuals who have utilized this application.

Chapter 6

SNAPSHOTS

This chapter consists of working screenshots of the project.

6.1 LOGINPAGE

Fig-6.1: Login page

Fig-6.2: Admin login

Fig-6.3: user login page

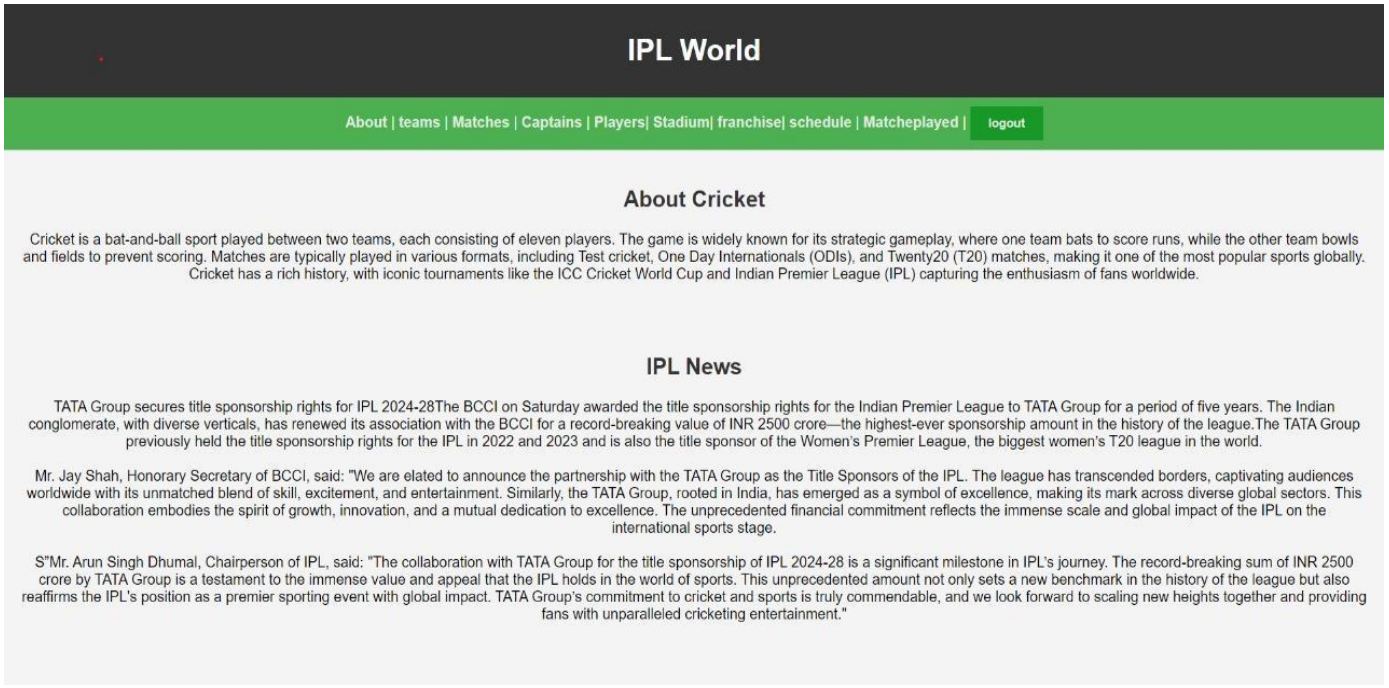


Fig-6.4: Home page

Fig-6.5: About page

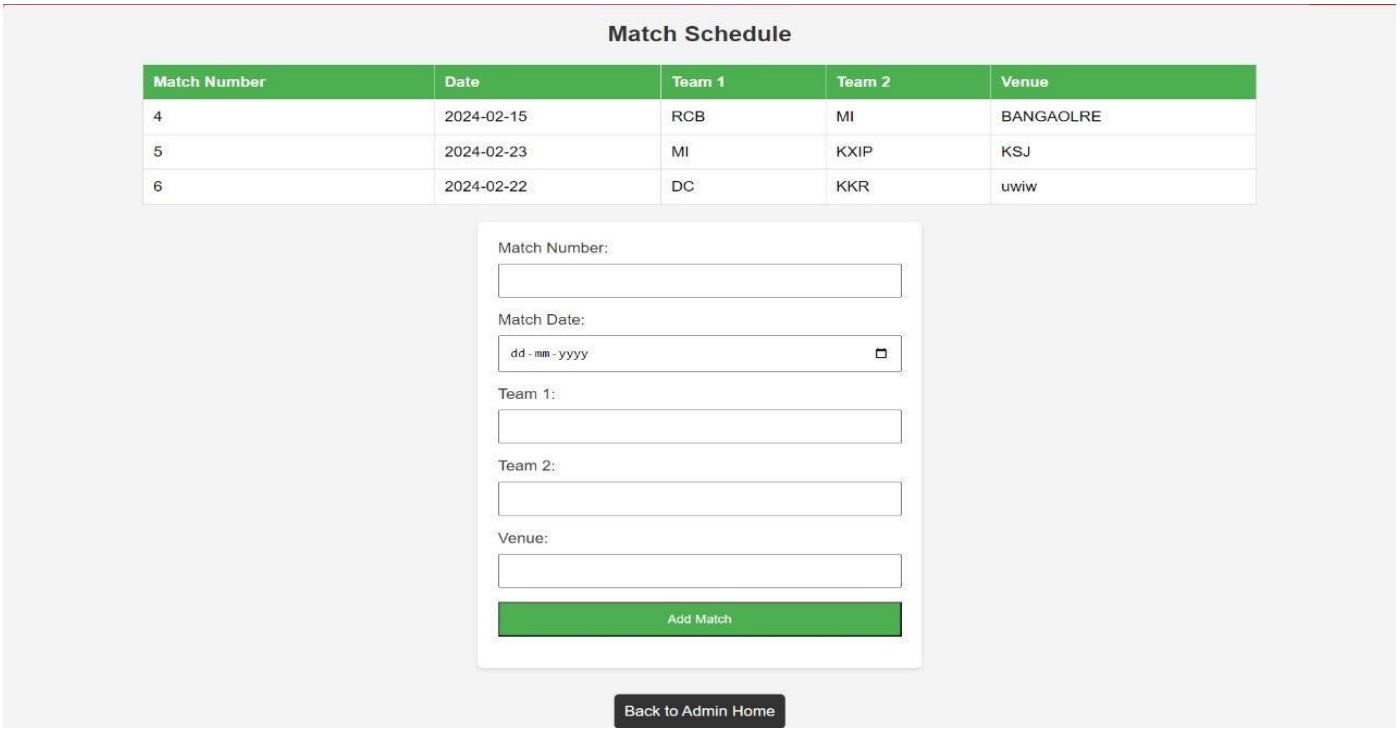


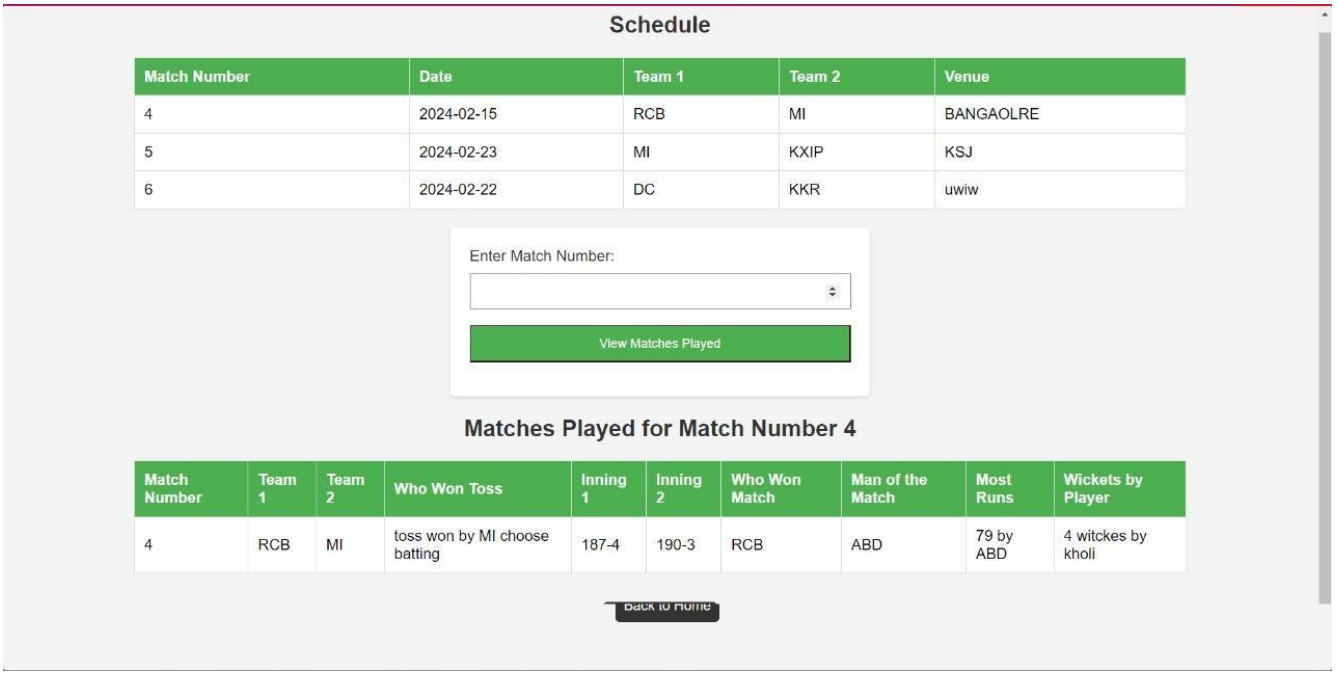
Fig-6.6: Match page

Match Information for All Teams

Team Name	Matches	Wins	Winning Percentage (%)	Lost	Tied	Total Points
RCB	192	91	47.40	95	6	188
MI	214	119	55.61	94	0	239
KKR	214	106	49.53	104	4	216
CSK	186	106	56.99	78	2	214
DC	214	101	47.20	110	3	205
KXIP	186	87	46.77	99	3	177
RR	186	92	49.46	91	5	189
SRH	142	71	50.00	70	1	141

Back to Homepage

Fig-6.7: Matches played page



Schedule

Match Number	Date	Team 1	Team 2	Venue
4	2024-02-15	RCB	MI	BANGAOLRE
5	2024-02-23	MI	KXIP	KSJ
6	2024-02-22	DC	KKR	uwiw

Enter Match Number:

View Matches Played

Matches Played for Match Number 4

Match Number	Team 1	Team 2	Who Won Toss	Inning 1	Inning 2	Who Won Match	Man of the Match	Most Runs	Wickets by Player
4	RCB	MI	toss won by MI choose batting	187-4	190-3	RCB	ABD	79 by ABD	4 witckes by kholi

Back to Home

Fig-6.8: Schedule page

Chapter 7

CONCLUSION

The project, developed using PHP and MySQL, is based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement. The expanded functionality of today's software requires an appropriate approach towards software development. This IPL database management software is designed for peoplewho want to manage various particulars and can be known by recording them in the database. various records and particulars about match increased rapidly. Thereby the numbers of matches and there is going to be increased day-by- day. And hence there is a lot of strain on the person who are watching the IPL to know about future matches and also to see the records done by various players and getting datils in fingertips. Identification of the drawbacks of the existing system leads to the designing of a computerized system that will be compatible with the existing system with the system which is more user friendly and more GUI oriented.

Chapter 8

FUTURE ENHANCEMENTS

The current project is just based on taking the information and storing in respective data tables and representing the information in the different required forms and can search using the attribute. There aresome enhancements which can be implemented further. They are as follows:

- Module that automatically gives information about various cricket boards, stadiums and rankings of various IPL teams and players by selecting or entering the relevant required item.
- Module that gives information about filled and partially filled information on various particulars.
- Can create module such that the user can login and gain information through the window.
- Make this project to the scope for players also.

REFERENCES

[1] Ramez Elmasri and Sham Kant B. Navathe, “Database System Models, Languages,” in, VIIth ed. 2019 Publisher, Pearson.

[2] Raghu Ramakrishnan, “Database Management Systems,” in, IIIth ed. 2014 Publisher, Mc Graw Hills.

[3] A. Abdullah (B)·N. Ahmed Department of Media Technology Engineering, College of Engineering, University of Information Technology and Communications (UOITC), Baghdad, Iraq (2021) Presented at © Springer Nature Singapore Pte Ltd. 2021.

[4] Samuel A. Olajide “DEVELOPMENT OF WEB-BASED EXAMINATION SYSTEM USING OPEN- SOURCE PROGRAMMING MODEL” Presented at the 2017 Turkish Online Journal of Distance Education-TOJDE April 2017 ISSN 1302-6488 Volume: 18

[5] Harshada Satav , Trupti Nanekar, Supriya Pingale, Nupur, “SQL Based programming System ,” Journal of Information Engineering and Applications www.iiste.org ISSN 2224-5782 (print) ISSN 2225-0506 (online)Vol 2, No.2, 2017.