Chapter 1

INTRODUCTION

The web-application is on Soccer Database Management system that stores the details of different matches, teams associated with different countries, different players along with the details of different stadiums worldwide. When the user wants to check any details regarding team, match or about the players, all he needs to do is click a button to find it out. With the ever-increasing need for easy data manipulation and storage, comes the need to create a system that enables the user to perform these tasks with minimal effort and margin of error.

The application was made using HTML & CSS for the front end and storing data in a MySQL backend using WAMP. Notepad++ was used to facilitate creation of the front end. The use of PHP to connect the front end to the backend helps simplifying the connection.

The application begins with a search screen. The search screen has the option to enter detail they want to see or search for on the database and then click on search button to see the results. The navigation bar has different options that the user can look for.

Chapter 2

SYSTEM REQUIREMENTS

2.1 Hardware Requirements

Windows

- Operating System : Windows 7 or later
- Processor : Intel Pentium 4 or later
- Memory : 2 GB minimum, 4 GB recommended
- Screen resolution : 1280*1024 or larger
- Application Window Size : 1024*680 or larger
- Internet Connection : Not required

Mac

- Operating System : Mac OS X 10.9.x or later
- Processor : Intel
- Memory : 2 GB minimum, 4 GB recommended
- Screen resolution : 1280*1024 or larger
- Application Window Size : 1024*680 or larger
- Internet Connection : Not required

2.2 Software Requirements

- Client : Operating System(any)
- Web Server : Apache Web Server
- Browser : Any, Google Chrome recommended
- Database : MYSQL
- Language : HTML, CSS, PHP

2.3 Software and Executables memory size

Fig 2.1 Memory consumption by Notepad++ (64-bit) editor

Fig 2.2 Memory consumption by Chrome

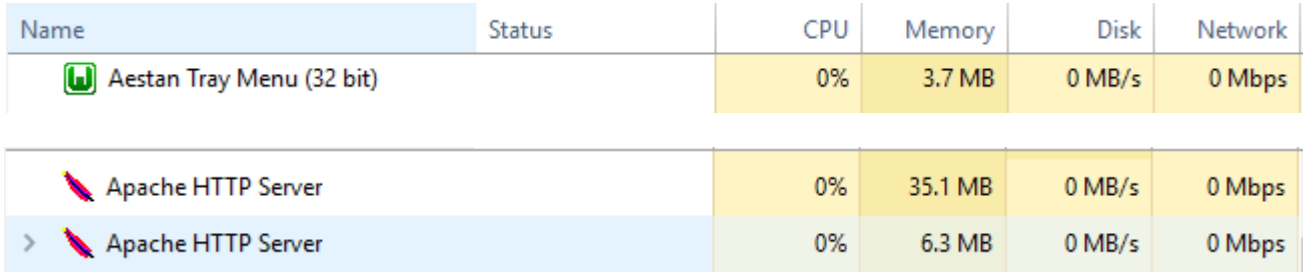


Fig 2.3 Memory consumption by local server (Wampserver 64-bit)

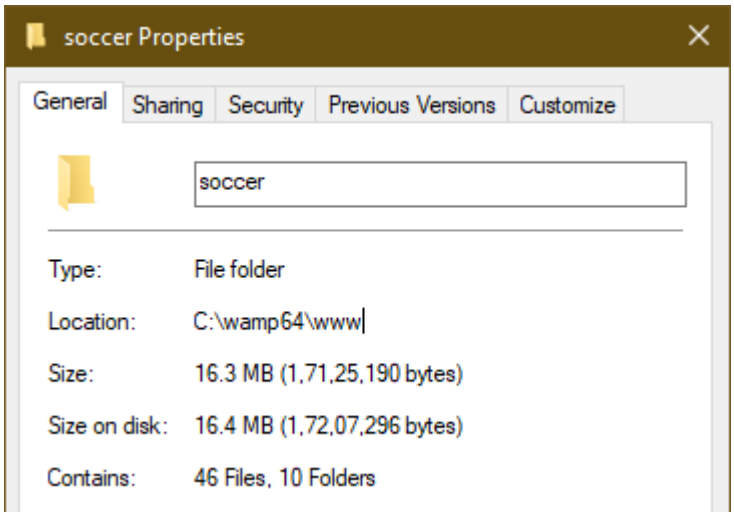


Fig 2.4 Project files' size on disk

Chapter 3

PROBLEM DESCRIPTION

Soccer Database Management System consists of **six tables**. The project contains **three stored procedures** for deleting and **three triggers** to keep a track of Insertion, Deletion and Update in various tables in the database.

The database uses the following tables for maintaining the details:

- Team
- Matches
- Player
- Goals
- Stats
- Coach

This project is a simple prototype of managing larger numbers of players across different nations with different stats and attributes. It helps to access players. It also helps in monitoring player stats.

3.3 Table Structures

3.3.1 Team details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	TEAMNAME	varchar(40)	latin1_swedish_ci		No	*None*			Change Drop More
2	ABB	varchar(6)	latin1_swedish_ci		Yes	*NULL*			Change Drop More
3	WEBSITE	varchar(60)	latin1_swedish_ci		Yes	*NULL*			Change Drop More
4	CITY	varchar(40)	latin1_swedish_ci		Yes	*NULL*			Change Drop More
5	POINTS	int(11)			Yes	*NULL*			Change Drop More

Fig 3.1 Team table structure

The Team table consists of five columns. TeamName is the primary key and it has references from other tables as well.

3.3.2 Match details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	MATCHID	int(11)			No	None			Change Drop More
2	RESULT	varchar(20)	latin1_swedish_ci		Yes	NULL			Change Drop More
3	HOMETEAM	varchar(40)	latin1_swedish_ci		Yes	NULL			Change Drop More
4	AWAYTEAM	varchar(40)	latin1_swedish_ci		Yes	NULL			Change Drop More
5	HOSTEDBY	varchar(40)	latin1_swedish_ci		Yes	NULL			Change Drop More
6	TEMP_DEG	int(11)			Yes	NULL			Change Drop More
7	PRECIP_PERCENT	int(11)			Yes	NULL			Change Drop More
8	HUMID_PERCENT	int(11)			Yes	NULL			Change Drop More

Fig 3.2 Match table structure

The Match table consists of eight columns. MatchID is the primary key of the table. HomeTeam, AwayTeam and HostedBy are the foreign keys in the table. HostedBy refers the Stadium table. HomeTeam and AwayTeam refers the Team table.

3.3.3 Stadium details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	CITY	varchar(40)	latin1_swedish_ci		Yes	NULL			Change Drop More
2	STDNAME	varchar(40)	latin1_swedish_ci		No	None			Change Drop More
3	CAPACITY	int(11)			Yes	NULL			Change Drop More
4	POSTAL	int(11)			Yes	NULL			Change Drop More

Fig 3.3 Stadium table structure

The Stadium table consists of four columns. STDName is the primary key of the table and has reference from Match table.

3.3.4 Coach details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	COACHID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	FNAME	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
3	LNAME	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
4	PHONENO	bigint(40)			No	None			Change Drop More
5	TYPE	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
6	COACHES	varchar(40)	latin1_swedish_ci		No	None			Change Drop More

Fig 3.4 Coach table structure

The Coach table has six columns. CoachID is the primary key of the table and Coaches is the foreign key that refers to the TeamName of Team table.

3.3.5 Player details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	PLAYERID	int(11)			No	*None*			Change Drop More
3	FNAME	varchar(40)	latin1_swedish_ci		Yes	*NULL*			Change Drop More
4	LNAME	varchar(40)	latin1_swedish_ci		Yes	*NULL*			Change Drop More
5	POSITION	varchar(20)	latin1_swedish_ci		Yes	*NULL*			Change Drop More
6	WEIGHT	int(11)			Yes	*NULL*			Change Drop More
7	HEIGHT	int(2)			Yes	*NULL*			Change Drop More
8	NATIONALITY	varchar(40)	latin1_swedish_ci		Yes	*NULL*			Change Drop More
9	KITINTEGER	int(11)			Yes	*NULL*			Change Drop More
10	TEAMNAME	varchar(40)	latin1_swedish_ci		Yes	*NULL*			Change Drop More

Fig 3.5 Player table structure

The Player table consists of ten columns. This table has the most number of attributes in the database. The column PlayerID is the primary key of the table and has references from Goal table and Stats table. The ID column has unique values and is auto incremented every time a row is added. TeamName is the foreign key of this table and refers to the Team table.

3.3.6 Goals' details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	GOALID	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	TIME	varchar(20)	latin1_swedish_ci		Yes	*NULL*			Change Drop More
3	PLAYERID	int(11)			Yes	*NULL*			Change Drop More

Fig 3.6 Goal table structure

The Goal table consists of three columns. GoalID is the primary key of the table. PlayerID is the foreign key of the table and refers the Player table. This table records the time at which a player has hit a goal in the Time column.

3.3.7 Player's Stats

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	PLAYERID	int(11)			No	*None*			Change Drop More
2	REDCARDS	int(11)			No	0			Change Drop More
3	YELLOWCARDS	int(11)			No	0			Change Drop More
4	GOALS	int(11)			No	0			Change Drop More
5	ASSISTS	int(11)			No	0			Change Drop More
6	CLEANSHEETS	int(11)			Yes	*NULL*			Change Drop More

Fig 3.7 Stats table structure

The Stats table has six columns, it records the number of Red Cards, Yellow Cards, Goals, Assists, and Clean Sheets of every player individually as the PlayerID is the Primary key as well as the foreign key of the table.

3.3.8 Insert Log

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	ID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	TABLENAME	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
☐	3	ACTION	varchar(40)	latin1_swedish_ci		No	None			Change Drop More
☐	4	TIME	datetime			No	None			Change Drop More

Fig 3.8 Insert log (trigger) structure

The Insert logs' table consist of four columns. ID column is the unique primary key of the table and is set to auto increment. The TableName column store the table at which the insert operation has been performed at. The Action column stores a phrase for successful insertion. Moreover, the Time column records the time at which insertion took place through the triggers stored in the database.

3.3.9 Update Log

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	ID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	TABLENAME	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
☐	3	ACTION	varchar(40)	latin1_swedish_ci		No	None			Change Drop More
☐	4	TIME	datetime			No	None			Change Drop More

Fig 3.9 Update log (trigger) structure

The Delete logs' table consist of four columns. ID column is the unique primary key of the table and is set to auto increment. The TableName column store the table at which the update operation has been performed at. The Action column stores a phrase for successful update. In addition, the Time column records the time at which update took place through the triggers stored in the database.

3.3.10 Delete Log

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	ID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	TABLENAME	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
☐	3	ACTION	varchar(40)	latin1_swedish_ci		No	None			Change Drop More
☐	4	TIME	datetime			No	None			Change Drop More

Fig 3.10 Delete log (trigger) structure

The Delete logs' table consist of four columns. ID column is the unique primary key of the table and is set to auto increment. The TableName column store the table at which the delete operation has been performed at. The Action column stores a phrase for successful deletion. In addition, the Time column records the time at which deletion took place through the triggers stored in the database.

3.3.11 Stored Procedures

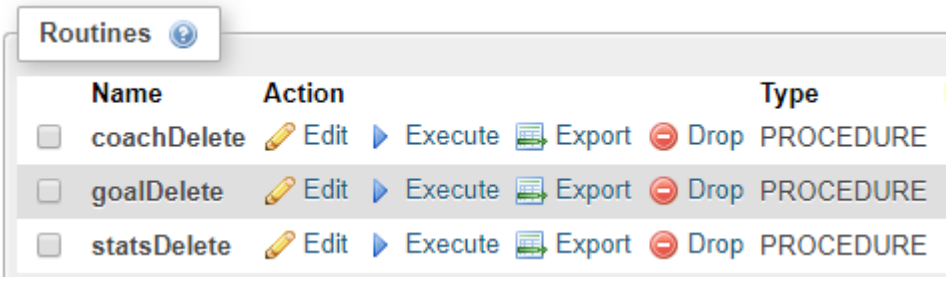


Fig 3.10 Stored procedure structure

There are three stored procedures present in the delete web page of the mini-project. These stored procedures are used to delete records from the respective tables.

Chapter 4

SYSTEM DESIGN

4.1 Schema Diagram

A database Schema is a skeleton structure that represents the logical view of the entire database.

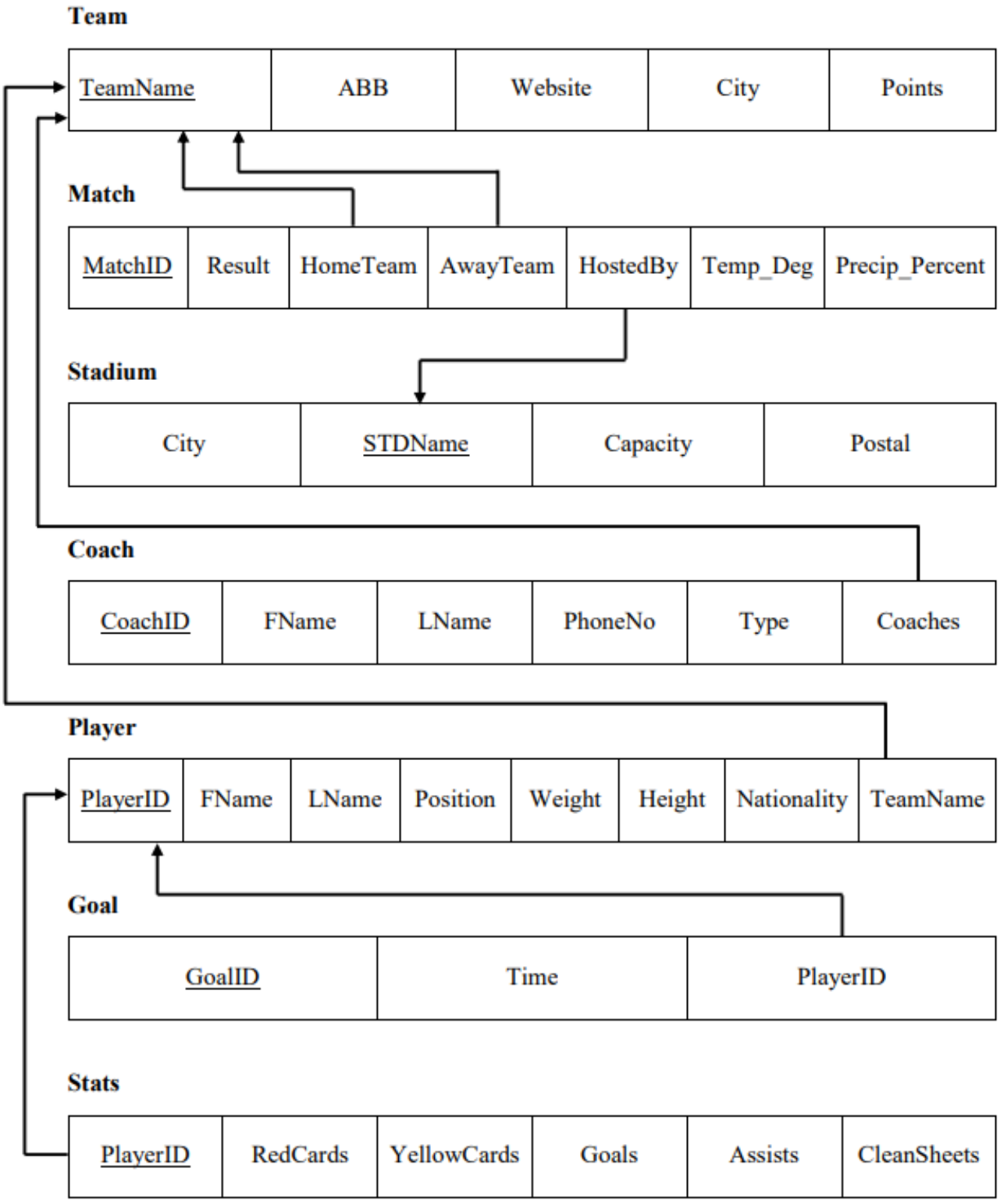


Fig 4.1 Schema diagram for Soccer database

4.2 ER Diagram

An Entity Relationship Diagram is a data modelling technique that graphically illustrates an information systems entity and the relationships between those entities.

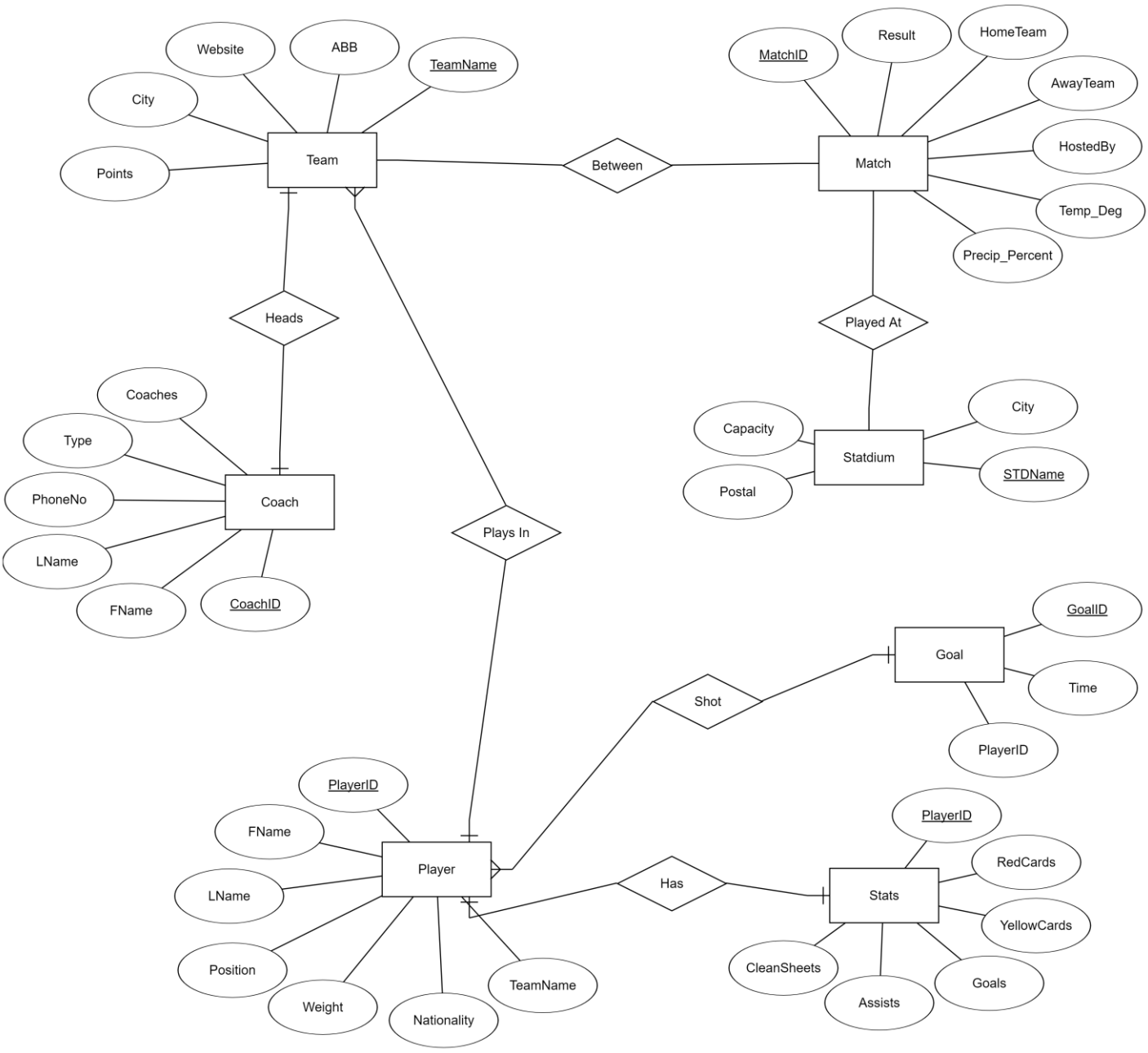


Fig 4.1 ER Diagram for Soccer Database

Chapter 5

IMPLEMENTATION

5.1 SQL Commands

The given below are the SQL queries used to create the database structure of the Soccer Database Management System.

```
CREATE TABLE TEAM (
TEAMNAME VARCHAR(40) PRIMARY KEY,
ABB VARCHAR(6),
WEBSITE VARCHAR(60),
CITY VARCHAR(40),
POINTS INTEGER );

CREATE TABLE PLAYER (
PLAYERID INTEGER PRIMARY KEY,
FNAME VARCHAR(40),
MNAME VARCHAR(40),
LNAME VARCHAR(40),
POSITION VARCHAR(20),
WEIGHT INTEGER,
HEIGHT INTEGER(2),
NATIONALITY VARCHAR(40),
KITINTEGER INTEGER,
TEAMNAME VARCHAR(40) );

CREATE TABLE MATCH (
MATCHID INTEGER PRIMARY KEY,
RESULT VARCHAR(20),
HOMETEAM VARCHAR(40),
AWAYTEAM VARCHAR(40),
HOSTEDBY VARCHAR(40),
```

```
TEMP_DEG INTEGER,
PRECIP_PERCENT INTEGER,
HUMID_PERCENT INTEGER );

CREATE TABLE GOAL (
GOALID INTEGER PRIMARY KEY,
TIME VARCHAR(20),
PLAYERID INTEGER );

CREATE TABLE STADIUM (
CITY VARCHAR(40),
STDNAME VARCHAR(40) PRIMARY KEY,
CAPACITY INTEGER,
POSTAL INTEGER );

CREATE TABLE COACH (
COACHID INTEGER PRIMARY KEY,
FNAME VARCHAR(20),
MNAME VARCHAR(20),
LNAME VARCHAR(20),
PHONENO INTEGER,
TYPE VARCHAR(20),
COACHES VARCHAR(40) );

CREATE TABLE STATS(
PLAYERID INTEGER PRIMARY KEY,
REDCARDS INTEGER,
YELLOWCARDS INTEGER,
GOALS INTEGER,
ASSISTS INTEGER,
CLEANSHEETS INTEGER );
```

ALTER TABLE **PLAYER** ADD CONSTRAINT PFK2 FOREIGN KEY (TEAMNAME) REFERENCES TEAM(TEAMNAME);

ALTER TABLE **MATCH** ADD CONSTRAINT MFK1 FOREIGN KEY (HOMETEAM) REFERENCES TEAM(TEAMNAME);

ALTER TABLE **MATCH** ADD CONSTRAINT MFK2 FOREIGN KEY (AWAYTEAM) REFERENCES TEAM(TEAMNAME);

ALTER TABLE **MATCH** ADD CONSTRAINT MFK3 FOREIGN KEY (HOSTEDBY) REFERENCES STADIUM(STDNAME);

ALTER TABLE **GOAL** ADD CONSTRAINT GFK FOREIGN KEY (PLAYERID) REFERENCES PLAYER(PLAYERID);

ALTER TABLE **COACH** ADD CONSTRAINT CHFK2 FOREIGN KEY (COACHES) REFERENCES TEAM(TEAMNAME);

ALTER TABLE **STATS** ADD CONSTRAINT STFK FOREIGN KEY (PLAYERID) REFERENCES PLAYER(PLAYERID);

5.2 Source Code

5.2.1 index.html

```
<html>

  <head>

     <title>Soccer Management System</title>

   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

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

     <link rel="icon" href="images/icon.jpg">

   <meta http-equiv="X-UA-Compatible" content="IE=edge" />

     <link type="text/css" rel="stylesheet" href="css/navbar.css">

   <meta name="author" content="colorlib.com">

   <link href="https://fonts.googleapis.com/css?family=Poppins" rel="stylesheet" />

   <link href="css/main.css" rel="stylesheet" />


         </head>


  <body>
```

```
<nav id="navbar" class="navigation-bar">
  <div class="nav-wrapper">
     <!-- Navbar Logo -->
    <div class="logo">
     <!-- Logo Placeholder for Inlustration -->
     <a href="#home"><i class="fas fa-chess-knight"></i>Soccer DBMS </a>
    </div>

    <!-- Navbar Links -->
    <ul id="menu">
     <li><a href="index.html">Search</a></li><!--
          --><li><a href="php/matches.php">Matches</a></li><!--
          --><li><a href="php/operations.php">Operations</a></li><!--
            --><li><a href="php/logs.php">Logs</a></li><!--
              --><li><a href="php/database.php">Database</a></li><!--
   --><li><a href="about.html">About</a></li>
    </ul>
  </div>
     </nav>

<!-- Menu Icon -->
<div class="menuIcon">
  <span class="icon icon-bars"></span>
  <span class="icon icon-bars overlay"></span>
</div>
<!-- <div class="overlay-menu">
  <ul id="menu">
     <li><a href="#home">Home</a></li>
```

```
    <li><a href="#services">Services</a></li>
    <li><a href="#about">About</a></li>
    <li><a href="#contact">Contact</a></li>
   </ul>
</div> -->


   <div class="s003">
    <form action="php/submit.php" method="post">
     <div class="inner-form">
      <div class="input-field first-wrap">
       <div class="input-select">
        <select data-trigger="" name="choices-single-defaul">
         <option placeholder="">Category</option>
                                <option>Team Name</option>
                                <option>Player ID</option>
         <option>Player Name</option>
         <option>Nationality</option>
         <option>Coach Name</option>
        </select>
       </div>
      </div>
      <div class="input-field second-wrap">
       <input id="search" type="text" placeholder="Search..." name="searchtext"/>
      </div>
      <div class="input-field third-wrap">
       <button class="btn-search" type="submit" onClick="php/submit.php">
        <svg class="svg-inline--fa fa-search fa-w-16" aria-hidden="true" data-prefix="fas" data-
icon="search" role="img" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 512 512">
```

```
<path fill="currentColor" d="M505 442.7L405.3 343c-4.5-4.5-10.6-7-17-7H372c27.6-35.3 44-79.7 44-128C416 93.1 322.9 0 208 0S0 93.1 0 208s93.1 208 208 208c48.3 0 92.7-16.4 128-44v16.3c0 6.4 2.5 12.5 7 17l99.7 99.7c9.4 9.4 24.6 9.4 33.9 0l28.3-28.3c9.4-9.4 9.4-24.6.1-34zM208 336c-70.7 0-128-57.2-128-128 0-70.7 57.2-128 128-128 70.7 0 128 57.2 128 128 0 70.7-57.2 128-128 128z"></path>
        </svg>
      </button>
    </div>
  </div>
 </form>
</div>
<script src="js/extention/choices.js"></script>
<script>
 const choices = new Choices('[data-trigger]',
 {
  searchEnabled: false,
  itemSelectText: '',
 });


</script>
</body><!-- This templates was made by Colorlib (https://colorlib.com) -->
</html>
```

5.2.2 submit.php

This page displays the record of the table according to the user needs. This acts upon the input given by the page index.html

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "football_league";
```

```
$conn = new mysqli($servername, $username, $password, $database);


// Check connection

if ($conn->connect_error) {

die("Connection failed: " . $conn->connect_error);

}


$inputchoice = $_POST["choices-single-defaul"];

$inputtext=$_POST["searchtext"];

?>


<?php

if($inputchoice=="Player Name" || $inputchoice=="Player ID"){


if($inputchoice=="Player Name"){

$sql = "SELECT * FROM football_league.player p, football_league.stats s WHERE ( fname=\"$inputtext\" OR lname=\"$inputtext\" ) AND p.playerid=s.playerid ";

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

} else {

$sql = "SELECT * FROM football_league.player p, football_league.stats s WHERE p.playerid=\"$inputtext\" and p.playerid=s.playerid ";

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

}

?>

<div class="s003">

<table>

<th>Player ID</th>

<th>First Name</th>

<th>Last Name</th>
```

```
<th>Position</th>

<th>Weight</th>

<th>Height</th>

<th>Naitonality</th>

<th>Number</th>

<th>Team</th>



<?php

echo "<tbody>";

if ($result->num_rows > 0) {

// output data of each row

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

echo "<tr>";

echo "<td> " . $row["PLAYERID"] ."</td><td>". $row["FNAME"] . "</td><td>" . $row["LNAME"]. "</td><td>". $row["POSITION"]. "</td><td>". $row["WEIGHT"]. "</td><td>". $row["HEIGHT"]. "</td><td>". $row["NATIONALITY"]."</td><td>".$row["KITINTEGER"]. "</td><td>" . $row["TEAMNAME"]. "</td><br>";

echo "</tr>";

}

}

echo "</tbody>";

?>

</table>


<?php

}

else if($inputchoice=="Team Name"){

$sql = "SELECT * FROM football_league.team WHERE TEAMNAME=\"$inputtext\"";

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

```
?>

<table>

<th>Team Name</th>

<th>Abbreviation</th>

<th>Website</th>

<th>City</th>

<th>Points</th>


<?php

echo "<tbody>";

if ($result->num_rows > 0) {


// output data of each row

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

echo "<tr>";

echo "<td> " . $row["TEAMNAME"] ."</td><td>". $row["ABB"] . "</td><td>" . $row["WEBSITE"]. "</td><td>". $row["CITY"]. "</td><td>". $row["POINTS"]. "</td><br>";

echo "</tr>";

}

}

echo "</tbody>";

?>

</table>



<?php

}

else if($inputchoice=="Coach Name"){
```

```
$sql = "SELECT * FROM football_league.coach WHERE FNAME=\"$inputtext\" OR LNAME=\"$inputtext\"";

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

?>

<table>

<th>Coach ID</th>

<th>First Name</th>

<th>Last Name</th>

<th>Phone No.</th>

<th>Type</th>

<th>Team</th>


<?php

echo "<tbody>";

if ($result->num_rows > 0) {


// output data of each row

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

echo "<tr>";

echo "<td> " . $row["COACHID"] ."</td><td>  ". $row["FNAME"] . "</td><td>  " . $row["LNAME"]. "</td><td>  ". $row["PHONENO"]. "</td><td>  ". $row["TYPE"]. "</td><td>". $row["COACHES"]. "</td><br>";

echo "</tr>";

}

}

echo "</tbody>";

?>

</table>

<?php
```

```
}else if($inputchoice=="Nationality"){

$sql="SELECT * FROM player WHERE nationality=\"$inputtext\"";

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

?>

<table>

<th>First Name</th>

<th>Last Name</th>

<th>Position</th>

<th>Nationality</th>

<th>Team</th>


<?php

echo "<tbody>";

if ($result->num_rows > 0) {

//echo "<tbody>";


// output data of each row

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

echo "<tr>";

echo "<td> " . $row["FNAME"] ."</td><td>". $row["LNAME"] . "</td><td>" . $row["POSITION"]. "</td><td>". $row["NATIONALITY"]. "</td><td>". $row["TEAMNAME"]. "</td><br>";

echo "</tr>";

}

//echo "</tbody>";

}

echo "<tbody>";

?>

</table>

<?php
```

```
}
else{
echo "wrong choice";
}
$conn->close();
?>
</tbody>
```

5.2.3 matches.php

This page takes input in the match table to record the details of a match.

```
<html>
  <head>
     <title>Soccer Management System</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
     <link rel="icon" href="../images/01.jpg">
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
     <link type="text/css" rel="stylesheet" href="../css/navbar.css">
    <meta name="author" content="colorlib.com">
    <link href="https://fonts.googleapis.com/css?family=Poppins" rel="stylesheet" />
    <link href="../css/main.css" rel="stylesheet" />
<nav id="navbar" class="navigation-bar">
  <div class="nav-wrapper">
     <!-- Navbar Logo -->
    <div class="logo">
     <!-- Logo Placeholder for Inlustration -->
     <a href="../index.html"><i class="fas fa-chess-knight"></i>Soccer DBMS</a>
    </div>

    <!-- Navbar Links -->
```

```
    <ul id="menu">
      <li><a href="../index.html">Search</a></li><!--
          --><li><a href="matches.php">Matches</a></li><!--
          --><li><a href="operations.php">Operations</a></li><!--
            --><li><a href="logs.php">Logs</a></li><!--
   --><li><a href="database.php">Database</a></li><!--
            --><li><a href="about.html">About</a></li>
    </ul>
  </div>
</nav>
<style>
table {
    border-left:1px solid #ddd;
          border-right:1px solid #ddd;
          border-top:1px solid #ddd;
          background:white;
          margin:3%;
}
td,th {
          padding:15px;
text-align:center;
border-left:1px solid #ddd;
          border-bottom:1px solid #ddd;
          }
tr:nth-child(even) {
  background-color: #dddddd;
}
br {display:none;}
```

```
</style>
        </head>
  <body>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "football_league";

$conn = new mysqli($servername, $username, $password, $database);

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

$fmatchid = $_POST["matchid"];
echo $fmatchid;
$fresult=$_POST["result"];
$fhometeam = $_POST["hometeam"];
$fawayteam = $_POST["awayteam"];
$fhost = $_POST["host"];
$ftemp = $_POST["temp"];
$fprecip = $_POST["precip"];
$fhumid = $_POST["humid"];
```

```
$sql2="Insert into football_league.matches(matchid, result, hometeam, awayteam, hostedby, temp_deg, precip_percent, humid_percent) values(\"$fmatchid\", \"$fresult\", \"$fhometeam\", \"$fawayteam\", \"$fhost\", \"$ftemp\", \"$fprecip\", \"$fhumid\")";

$result2 = $conn->query($sql2);

        if ($result2 === TRUE) {

   echo "New record created successfully";

} else {

   echo "Error: " . $sql2 . "<br>" . $conn->error;

}

$sql = "SELECT * FROM football_league.matches";

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

        ?>

<div class="s003">

<table>

<th>Match ID</th>

<th>Result</th>

<th>Home Team</th>

<th>Away Team</th>

<th>Hosted By</th>

<th>Temperature (Celsius)</th>

<th>Precipitation (%)</th>

<th>Humidity (%)</th>


        <?php

        $count=1;

        echo "<tbody>";

        if ($result->num_rows > 0) {

                // output data of each row

                while($row = $result->fetch_assoc()) {
```

```
                    echo "<tr>";

                    echo "<td> " . $row["MATCHID"] ."</td><td>". $row["RESULT"] . "</td><td>" .
$row["HOMETEAM"]. "</td><td>". $row["AWAYTEAM"]. "</td><td>". $row["HOSTEDBY"].
"</td><td>". $row["TEMP_DEG"]. "</td><td>".
$row["PRECIP_PERCENT"]."</td><td>".$row["HUMID_PERCENT"]. "</td><br>";

                    echo "</tr>";

                    $count++;

                }

        }

        echo "</tbody>";

?>


<tr>

<form class="form-inline" action="match.php" method="post">

        <td style="border:3px solid #ddd;"><input type="number" id="matchid" value=<?php echo $count ?>
name="matchid" style="width:100%;"></td>

        <td style="border:3px solid #ddd;"><input type="text" id="result" placeholder="Result"
name="result" style="width:100%;"></td>

        <td style="border:3px solid #ddd;"><input type="text" id="hometeam" placeholder="Home Team"
name="hometeam" style="width:100%;"></td>

    <td style="border:3px solid #ddd;"><input type="text" id="awayteam" placeholder="Away Team"
name="awayteam" style="width:100%;"></td>

    <td style="border:3px solid #ddd;"><input type="text" id="host" placeholder="Hosted By" name="host"
style="width:100%;"></td>

    <td style="border:3px solid #ddd;"><input type="number" id="temp" placeholder="Temperature"
name="temp" style="width:100%;"></td>

    <td style="border:3px solid #ddd;"><input type="number" id="precip" placeholder="Precipitation"
name="precip" style="width:100%;"></td>

    <td style="border:3px solid #ddd;"><input type="number" id="humid" placeholder="Humidity"
name="humid" style="width:100%;"></td>


<button type="submit" name="submit" value="submit" onClick="match.php"></button>

</form>
```

```
</tr>
</table>
</div>
 </body>
</html>
```

5.2.4 operation.php

This pages guides the user to select an operation and respective table to perform it.

```
<html>
  <head>
     <title>Soccer Management System</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
     <link rel="icon" href="../images/icon.jpg">
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
     <link type="text/css" rel="stylesheet" href="../css/navbar.css">
    <meta name="author" content="colorlib.com">
    <link href="https://fonts.googleapis.com/css?family=Poppins" rel="stylesheet" />
    <link href="../css/main.css" rel="stylesheet" />
<nav id="navbar" class="navigation-bar">
  <div class="nav-wrapper">
     <!-- Navbar Logo -->
    <div class="logo">
     <!-- Logo Placeholder for Inlustration -->
     <a href="../index.html"><i class="fas fa-chess-knight"></i>Soccer DBMS </a>
    </div>

    <!-- Navbar Links -->
    <ul id="menu">
     <li><a href="../index.html">Search</a></li><!--
```

```
        --><li><a href="matches.php">Matches</a></li><!--
        --><li><a href="operations.php">Operations</a></li><!--
          --><li><a href="logs.php">Logs</a></li><!--
            --><li><a href="database.php">Database</a></li><!--
  --><li><a href="about.html">About</a></li>
   </ul>
  </div>
</nav>

<style>
table, th, td {
   border: 1px solid black;
}

.button {
  display: inline-block;
  border-radius: 4px;
  background-color: #f4511e;
  border: none;
  color: #FFFFFF;
  text-align: center;
  font-size: 28px;
  padding: 20px;
  width: 200px;
  transition: all 0.5s;
  cursor: pointer;
  margin: 50px;
}
div {
```

```
        margin:2px;
}
.button span {
  cursor: pointer;
  display: inline-block;
  position: relative;
  transition: 0.5s;
}

.button span:after {
  content: '\00bb';
  position: absolute;
  opacity: 0;
  top: 0;
  right: -20px;
  transition: 0.5s;
}

.button:hover span {
  padding-right: 25px;
}

.button:hover span:after {
  opacity: 1;
  right: 0;
}

br {display:none;}
table {margin:80px;}
```

```
</style>
</head>
  <body>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "football_league";

$conn = new mysqli($servername, $username, $password, $database);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully<br>";
?>
<div>
<form method="post" action="operation/insert.php">
<div class="s003">
<div class="input-field first-wrap" >
          <div class="input-select">
            <select data-trigger="" name="choices-single-defaul" style="padding:20px; font-size: 160%;">
              <option placeholder="">Table</option>
                                  <option >Player</option>
              <option>Goal</option>
              <option>Stats</option>
              <option>Coach</option>
```

```
          </select>
        </div>
      </div>

<button class="button" style="vertical-align:middle" onclick="insert.php"><span>Insert</span></button>

 </div>
</form>
<form method="post" action="operation/update.php">
<div class="s003"  >
<div class="input-field first-wrap" >
        <div class="input-select">
          <select data-trigger="" name="choices-single-defaul" style="padding:20px; font-size: 160%;">
            <option placeholder="">Table</option>
                              <option>Player</option>
            <option>Stats</option>
            <option>Coach</option>
          </select>

        </div>
      </div>

<button class="button" style="vertical-align:middle"
onclick="operation/update.php"><span>Update</span></button>
 </div>
</form>
        <form method="post" action="operation/delete.php">
<div class="s003"  >
<div class="input-field first-wrap" >
```

```
        <div class="input-select">
          <select data-trigger="" name="choices-single-defaul" style="padding:20px; font-size: 160%;">
            <option placeholder="">Table</option>
            <option>Goal</option>
            <option>Stats</option>
            <option>Coach</option>
          </select>
        </div>
      </div>
<button class="button" style="vertical-align:middle" onclick="delete.php"><span>Delete</span></button>


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

5.2.5 Code snippet for insertion

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "football_league";

$conn = new mysqli($servername, $username, $password, $database);

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

```
echo "Connected successfully<br>";


$iplayerid=$_POST["playerid"];

echo $iplayerid;

$ifname=$_POST["fname"];

$ilname = $_POST["lname"];

$ipostion = $_POST["position"];

$iweight = $_POST["weight"];

$iheight = $_POST["height"];

$inationality = $_POST["nationality"];

$ikitinteger = $_POST["number"];

$iteam = $_POST["team"];


        $sql_p="Insert into football_league.player(playerid, fname, lname, position, weight, height, nationality, kitinteger, teamname) values(\"$iplayerid\",\"$ifname\", \"$ilname\", \"$ipostion\", \"$iweight\", \"$iheight\", \"$inationality\", \"$ikitinteger\", \"$iteam\")";

$result_p = $conn->query($sql_p);

        if ($result_p === TRUE) {

    echo "New record created successfully";

        header("Location:../../../html/success.html");

} else {

    echo "Error: " . $sql_p . "<br>" . $conn->error;

        header("Location:../../../html/unsuccessful.html");

}

?>
```

5.2.6 Code snippet for Update

```
<?php

$servername = "localhost";

$username = "root";
```

```
$password = "";

$database = "football_league";

$conn = new mysqli($servername, $username, $password, $database);


// Check connection

if ($conn->connect_error) {

die("Connection failed: " . $conn->connect_error);

}

echo "Connected successfully<br>";


$iplayerid=$_POST["playerid"];

$ifname=$_POST["fname"];

$ilname = $_POST["lname"];

$ipostion = $_POST["position"];

$iweight = $_POST["weight"];

$iheight = $_POST["height"];

$inationality = $_POST["nationality"];

$ikitinteger = $_POST["number"];

$iteam = $_POST["team"];


$sql_p="Update football_league.player set fname=\"$ifname\", lname=\"$ilname\", position=\"$ipostion\", weight=\"$iweight\", height=\"$iheight\", nationality=\"$inationality\", kitinteger=\"$ikitinteger\", teamname=\"$iteam\" where playerid = \"$iplayerid\"";

$result_p = $conn->query($sql_p);

if ($result_p === TRUE) {

echo "New record created successfully";

header("Location:../../../html/success_update.html");

} else {

echo "Error: " . $sql_p . "<br>" . $conn->error;

header("Location:../../../html/unsuccessful.html");

}
```

?>

5.2.6 Stored procedure for deletion

The Stored procedures are executed using PHP and MySQL in the application file.

Use: Use of stored procedures allows modular programming, reduces network traffic, faster execution, and can be used as security mechanism. Given below is a code snippet of implementation of all the 7 stored procedure present inside the web application.

```
<?php

$servername = "localhost";

$username = "root";

$password = "";

$database = "football_league";


//Create Connection

$conn = new mysqli($servername, $username, $password, $database);


// Check connection

if ($conn->connect_error) {

die("Connection failed: " . $conn->connect_error);

}


$input_id=$_POST["coachid"];


#Stored Procedure

$dpforcoach= mysqli_query($conn, "CREATE DEFINER=`root`@`localhost` PROCEDURE `coachDelete`(IN `c_id` INT(11)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER DELETE FROM coach WHERE coachid=c_id");


$call = "CALL coachDelete('$input_id')";


$result_C = mysqli_query($conn, $call);
```

```
if ($result_C === TRUE) {

echo "Record deleted successfully";

header("Location:../../../html/success_delete.html");

} else {

echo "Error: " . $call . "<br>" . $conn->error;

header("Location:../../../html/unsuccessful.html");

}

?>
```

Chapter 6

SCREENSHOTS

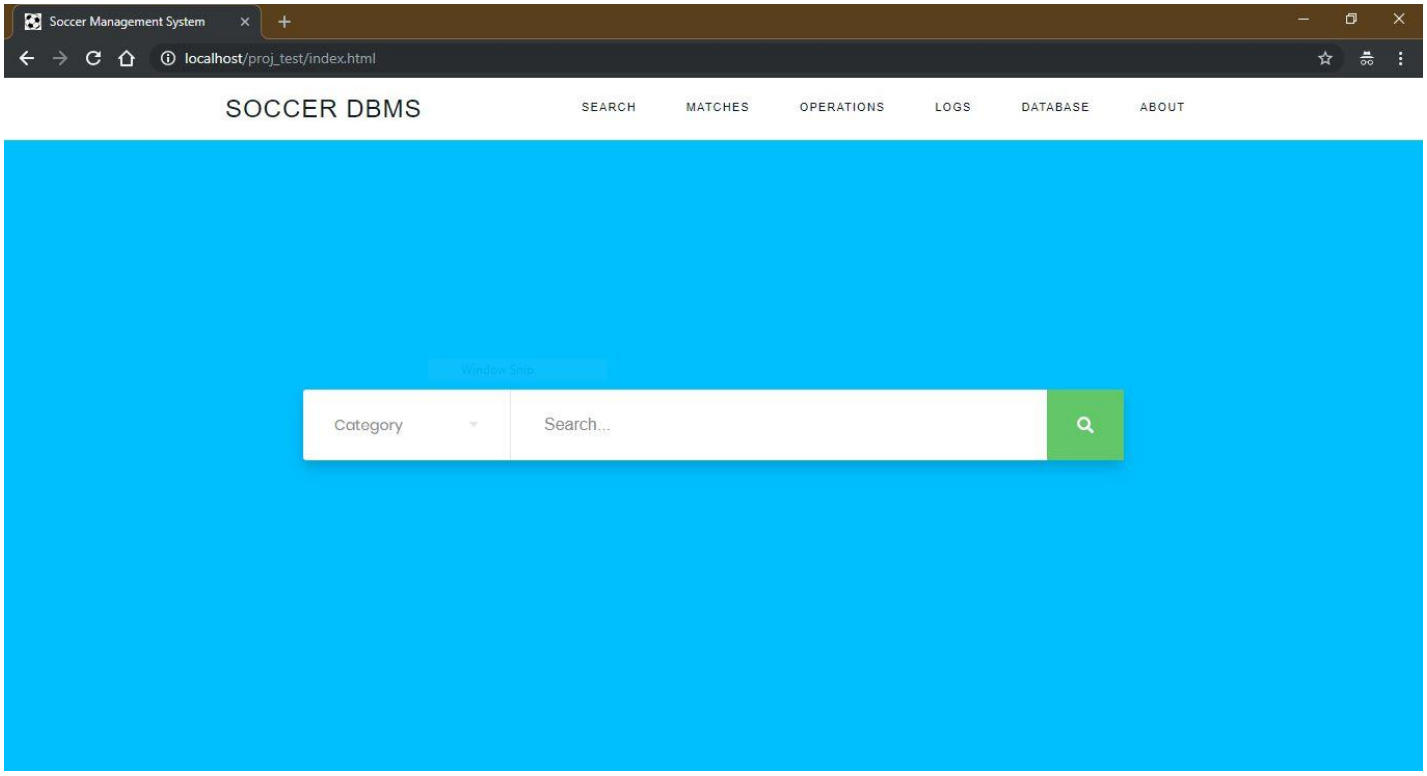
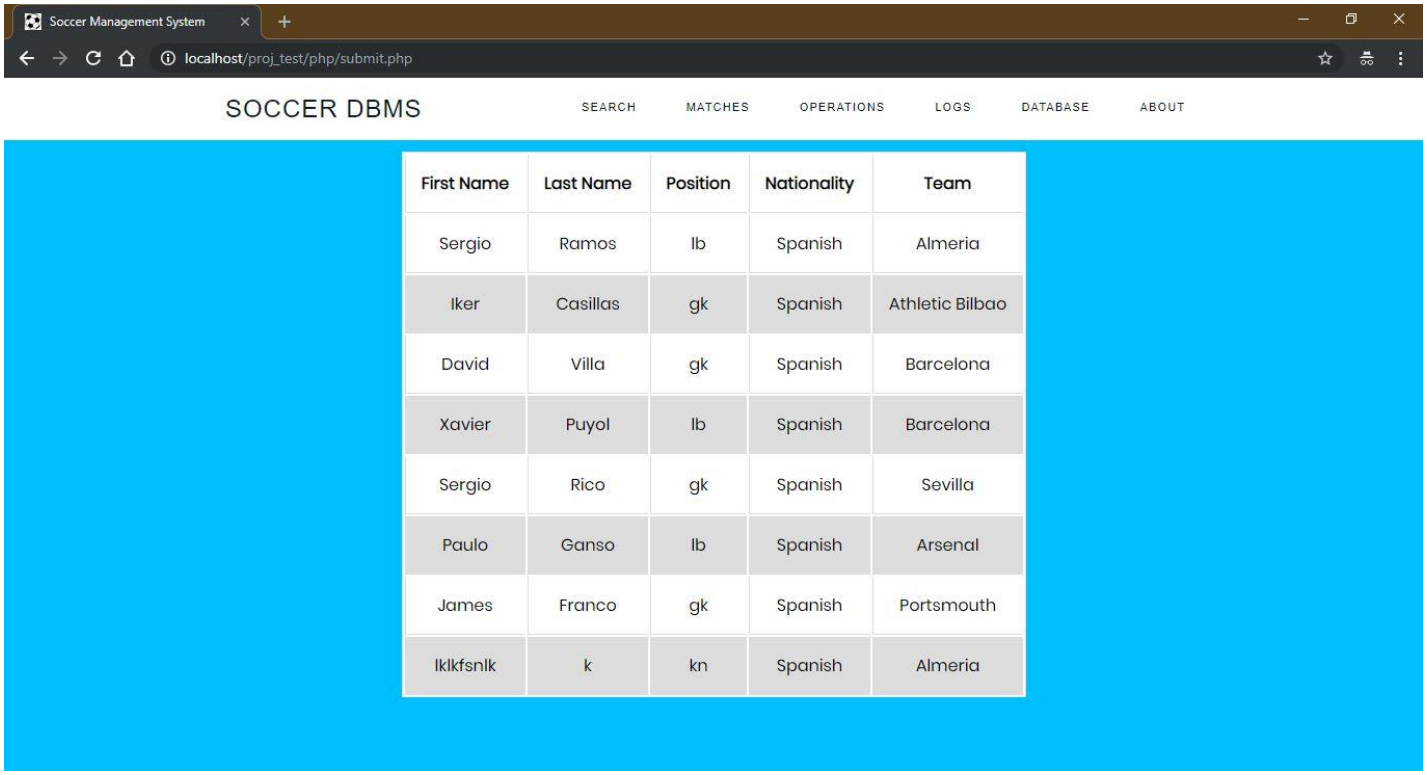


Fig 6.1 Search Input for traversing through the database



First Name	Last Name	Position	Nationality	Team
Sergio	Ramos	lb	Spanish	Almeria
Iker	Casillas	gk	Spanish	Athletic Bilbao
David	Villa	gk	Spanish	Barcelona
Xavier	Puyol	lb	Spanish	Barcelona
Sergio	Rico	gk	Spanish	Sevilla
Paulo	Ganso	lb	Spanish	Arsenal
James	Franco	gk	Spanish	Portsmouth
lklkfsnlk	k	kn	Spanish	Almeria

Fig 6.2 Result of the search operation

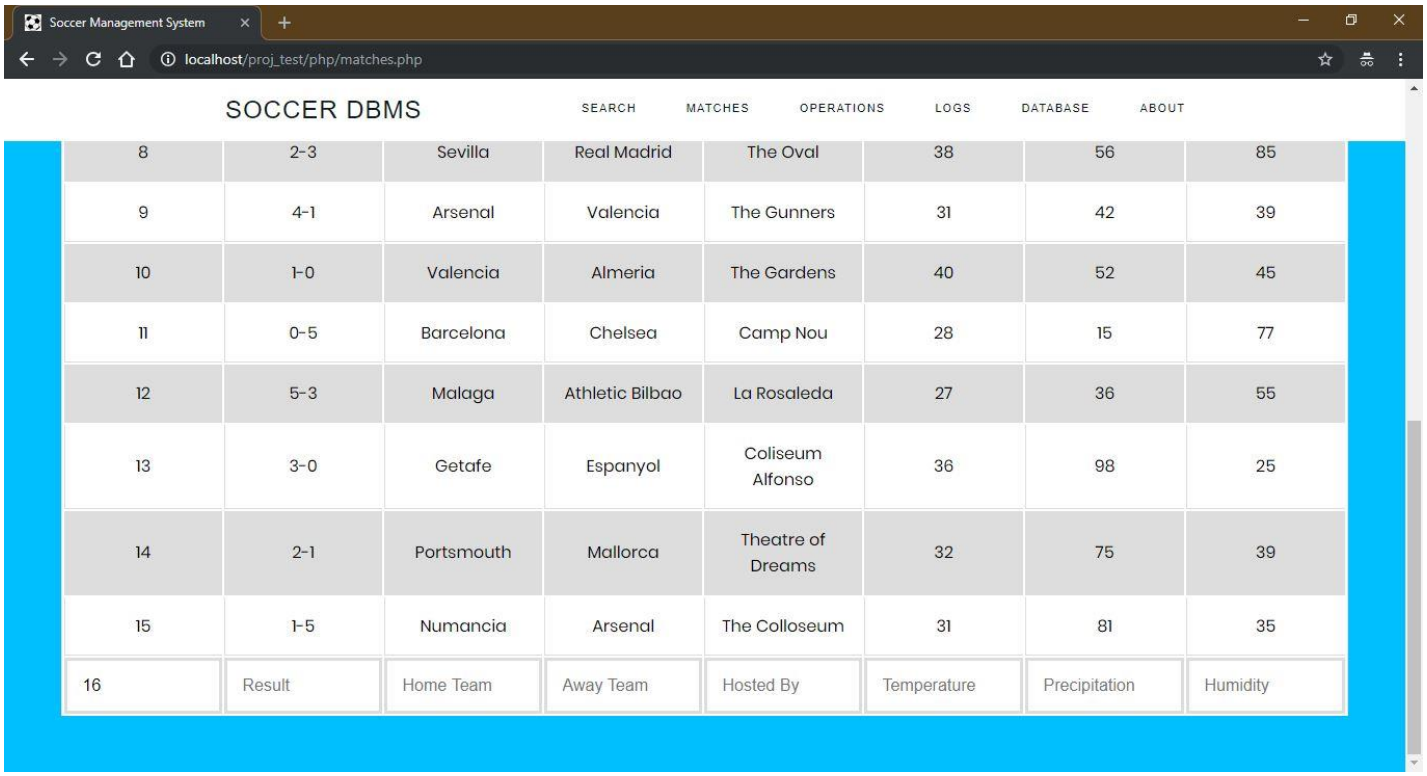


Fig 6.3 Match page to see the past matches and enter the new match

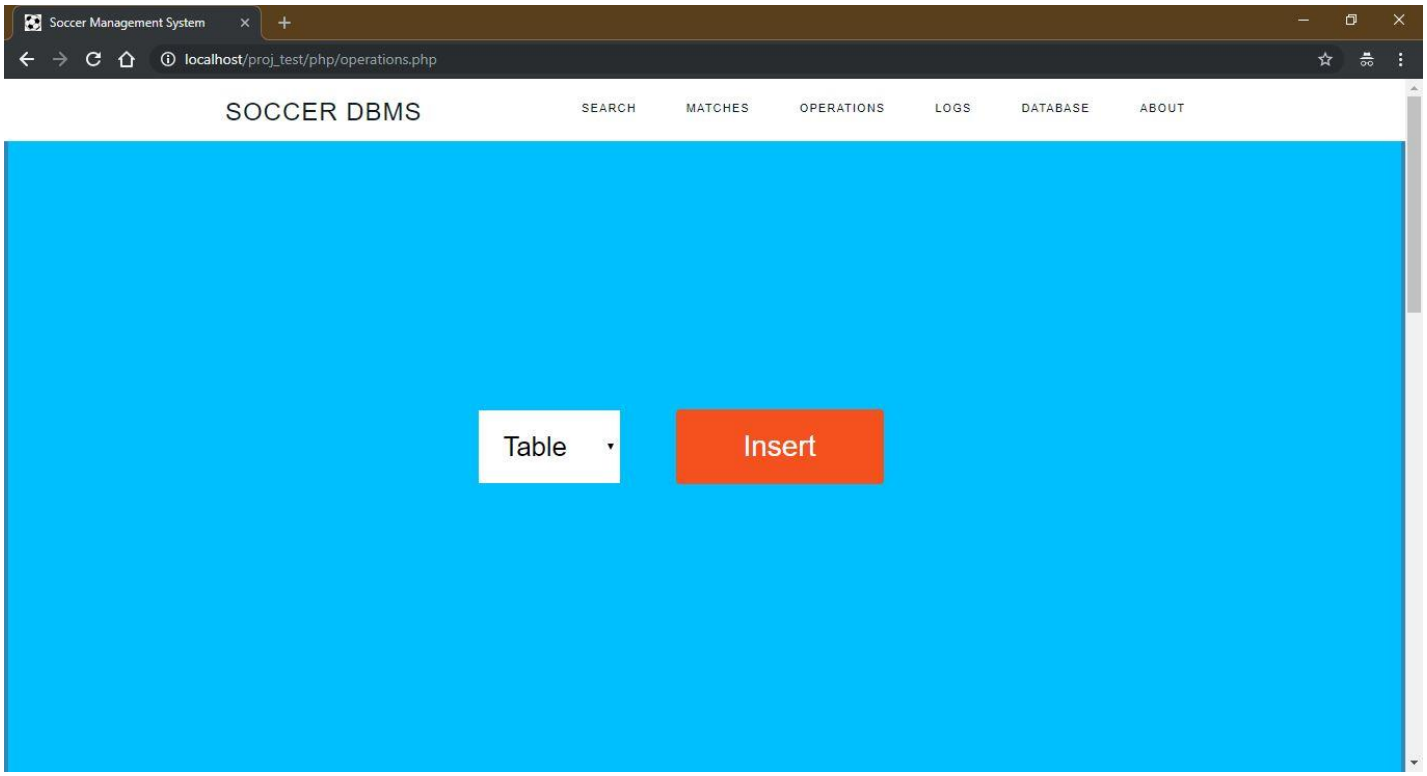


Fig 6.4 Operation page to select the table in which insertion should be done

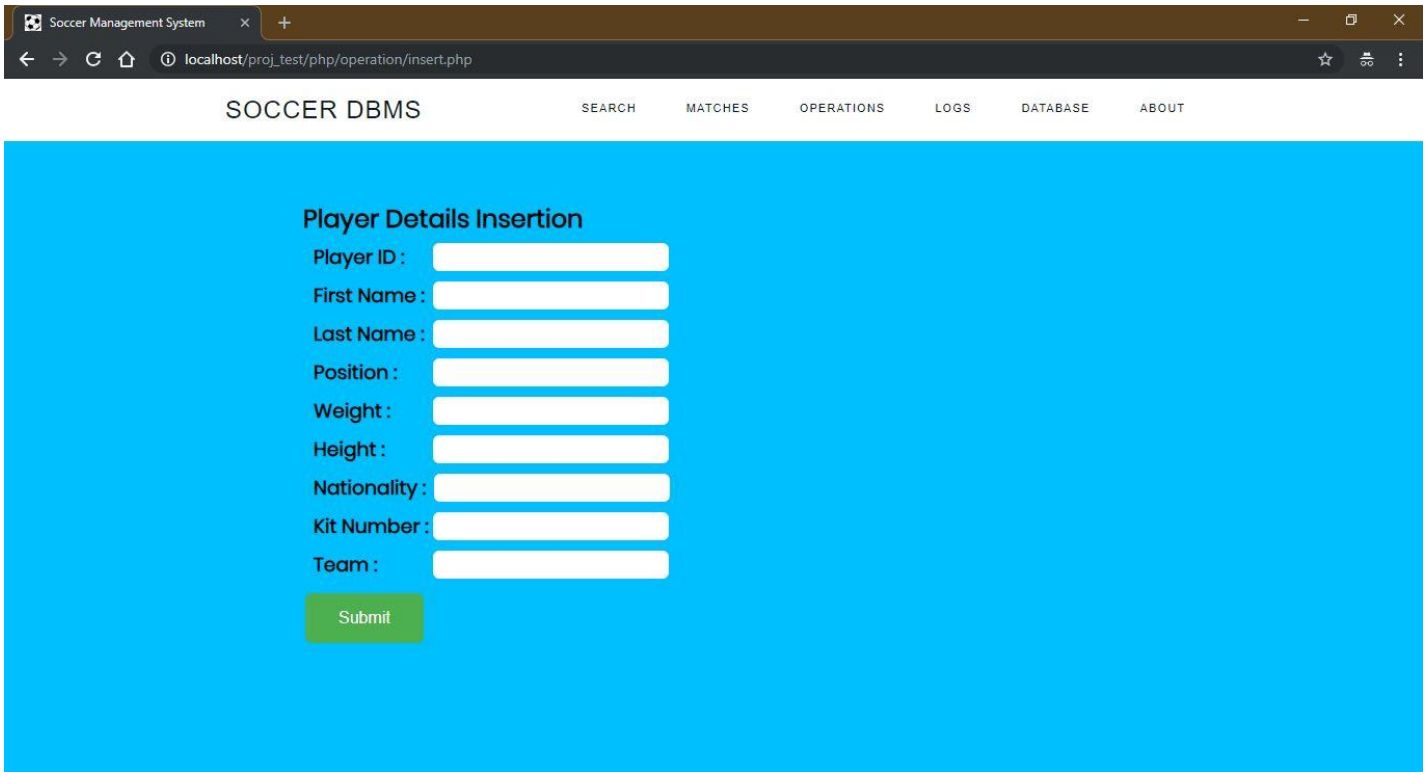


Fig 6.5 From to insert new record

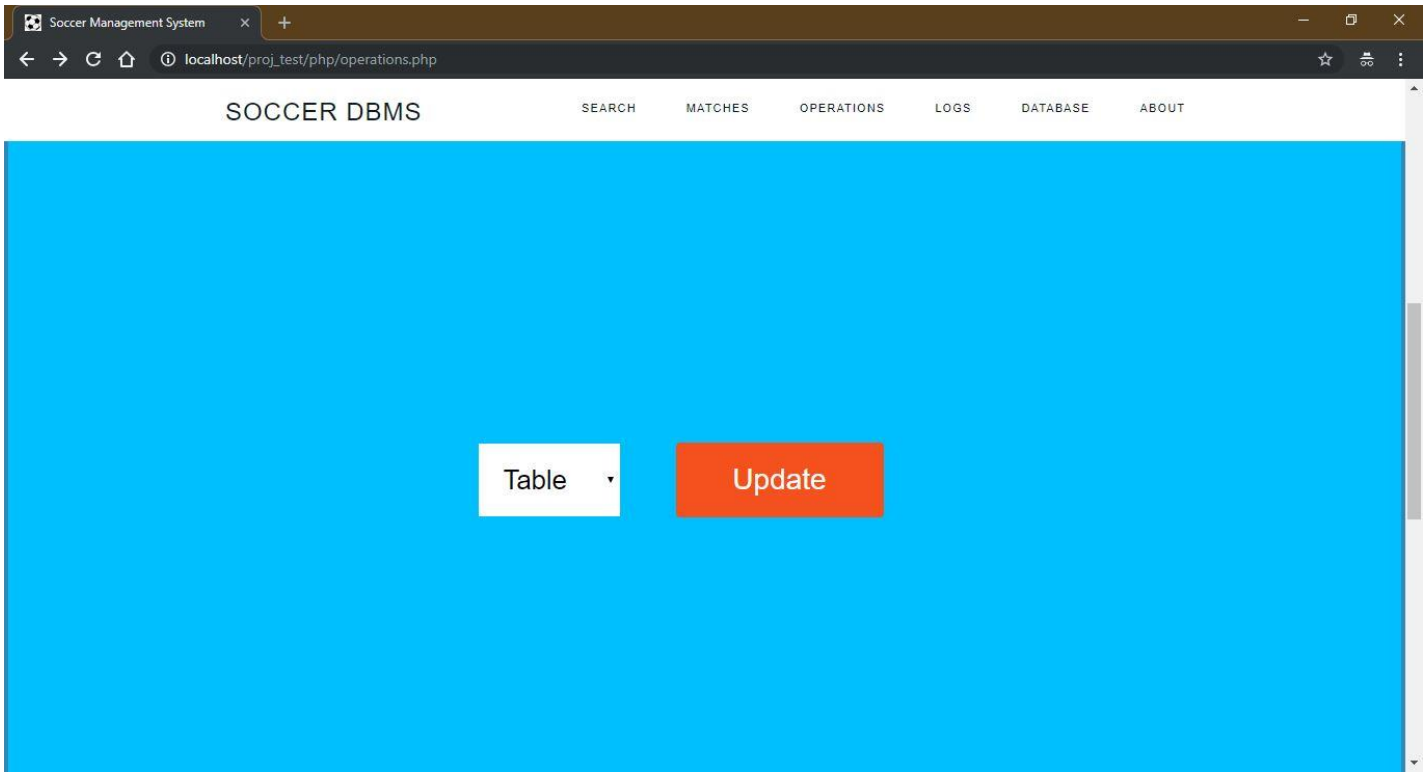


Fig 6.6 Operation page to select the table in which update should be done

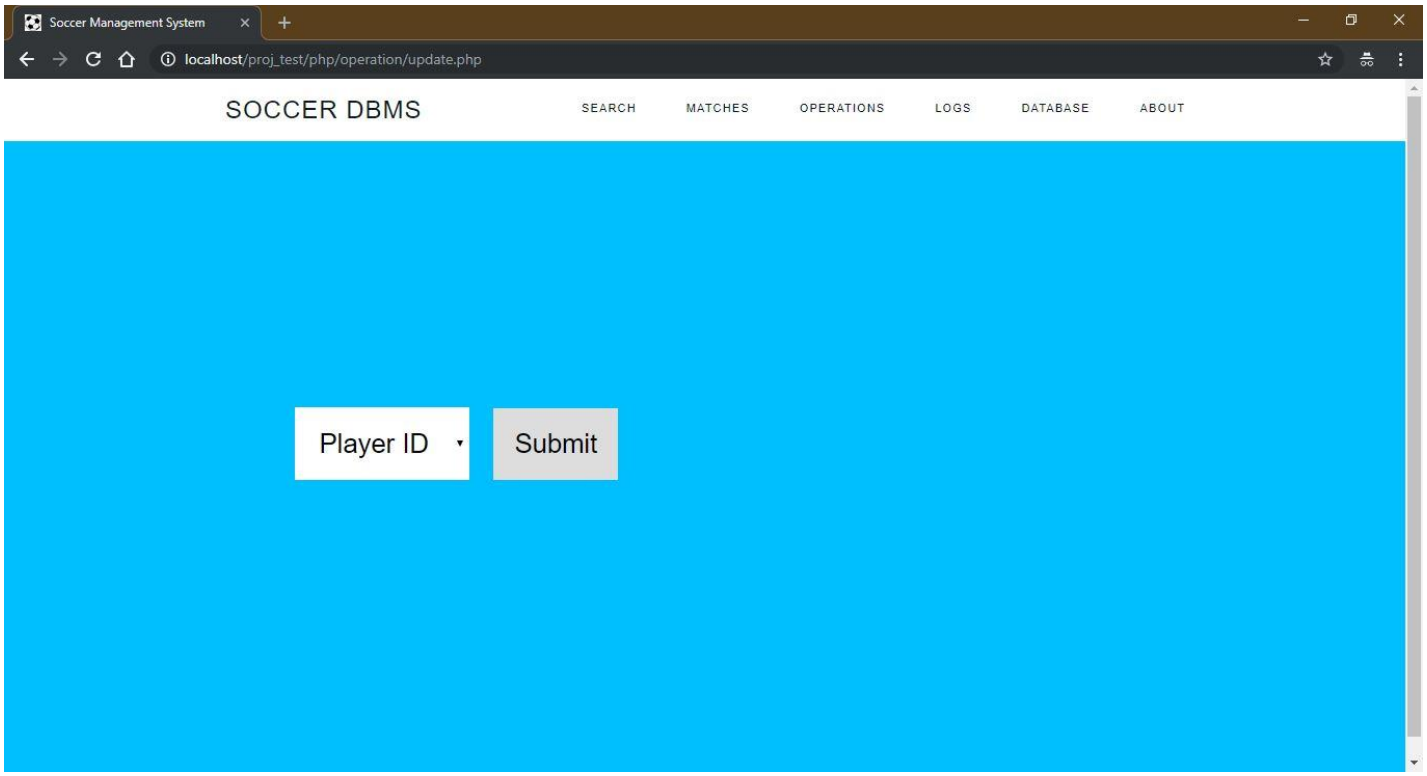


Fig 6.7 Selecting the details of the record to be modified

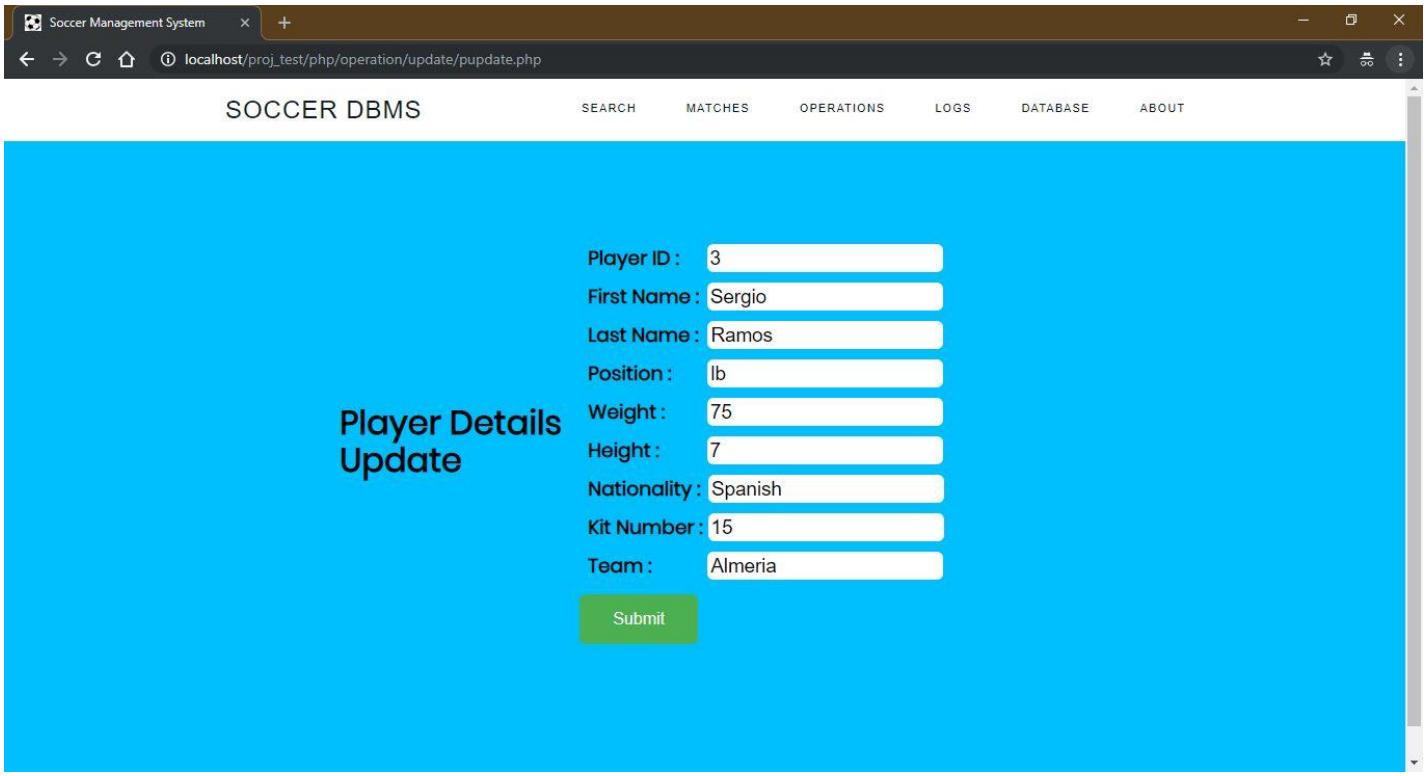


Fig 6.8 Already existing record for update

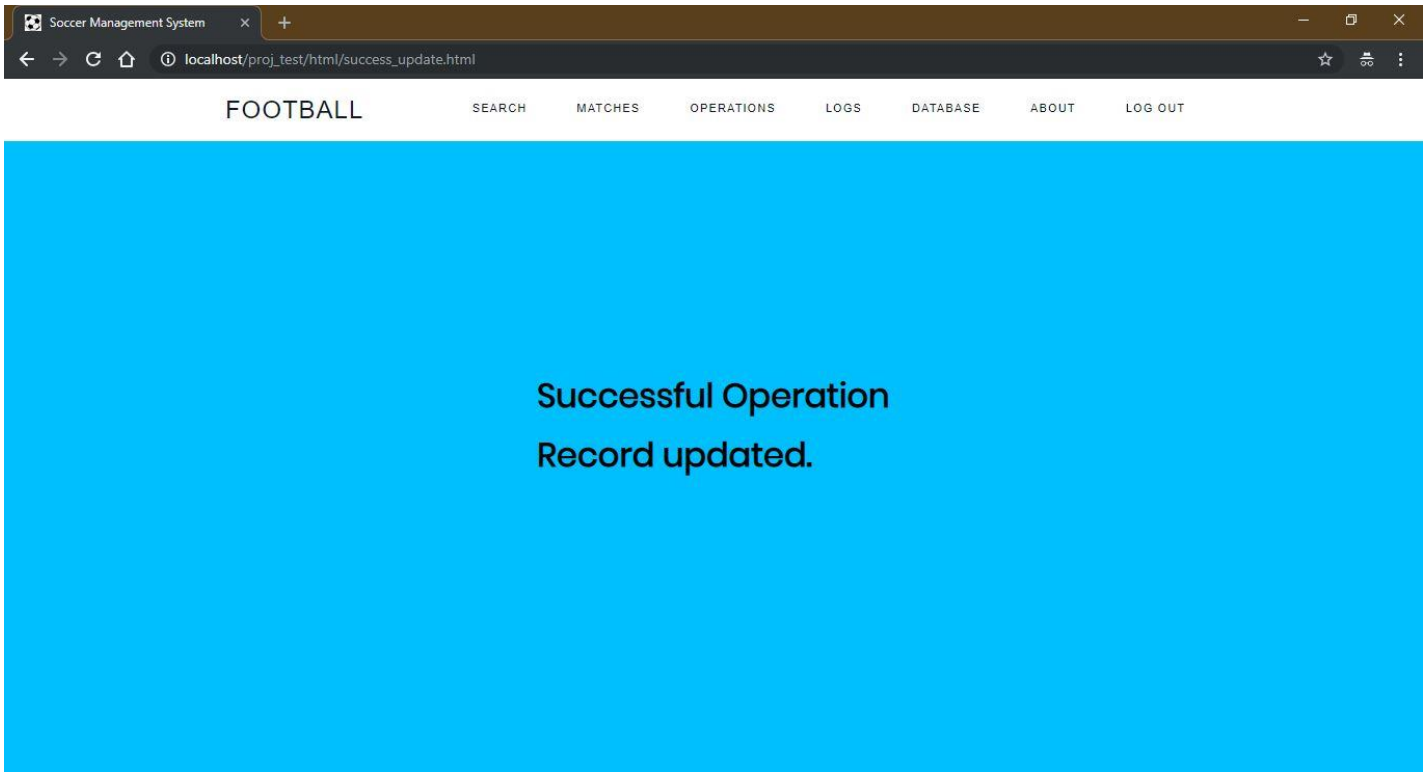


Fig 6.9 After successful operation

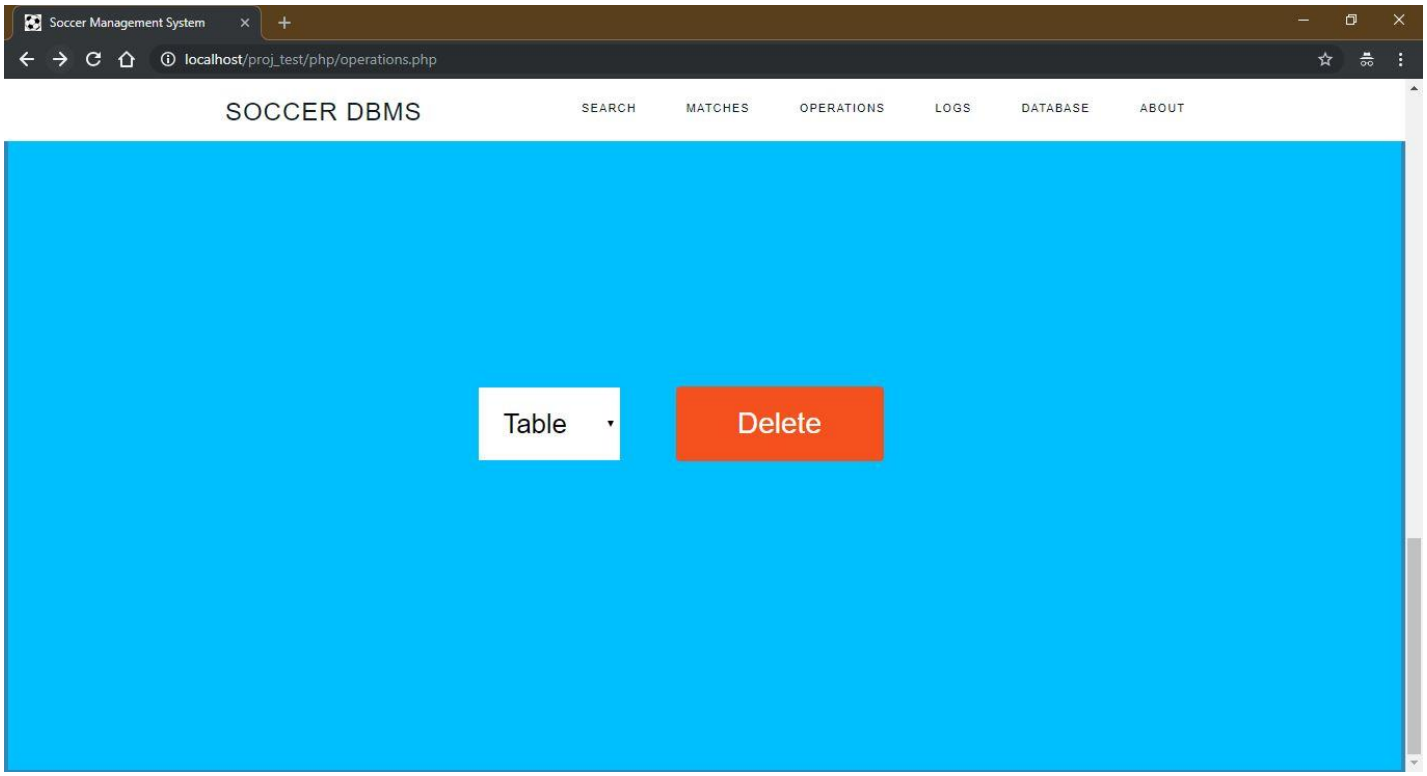


Fig 6.10 Operation page to select the table in which deletion should be done

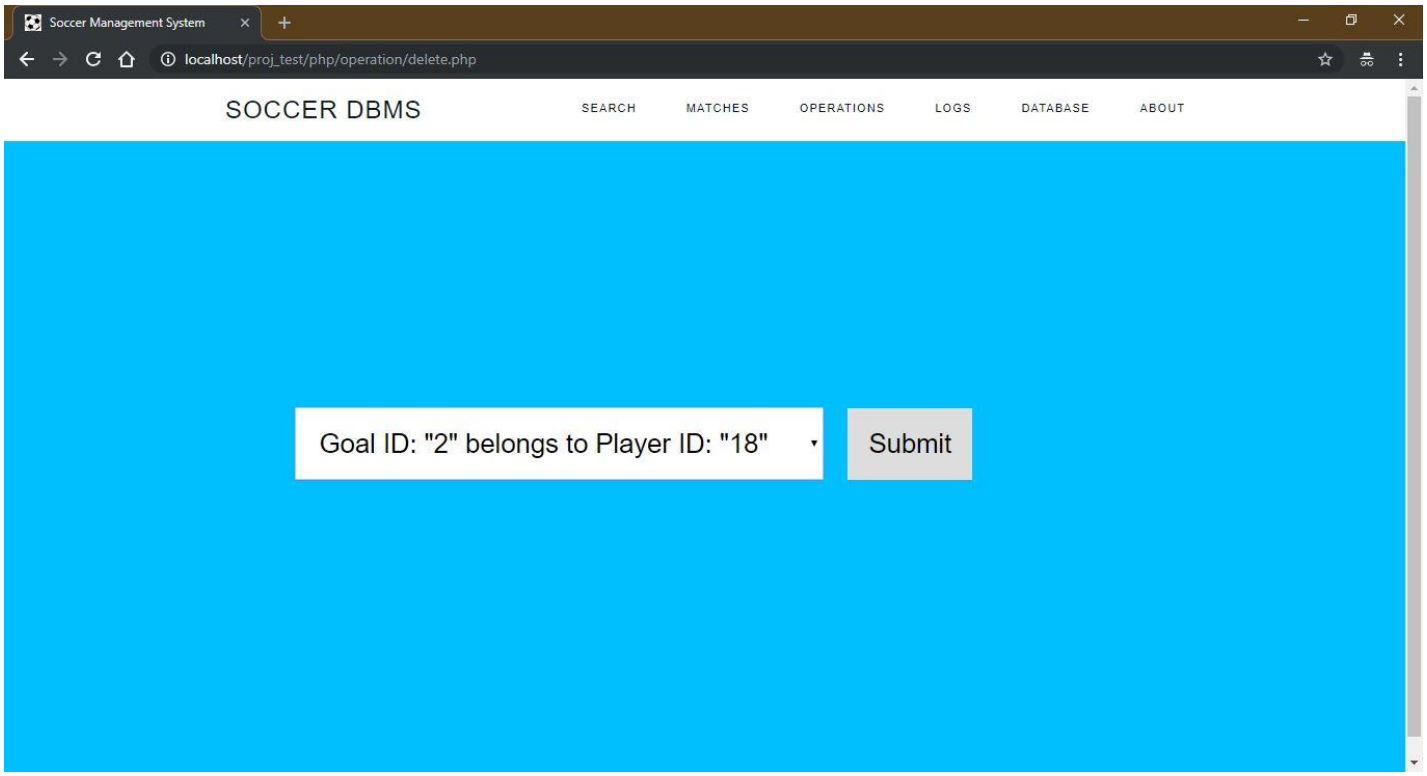
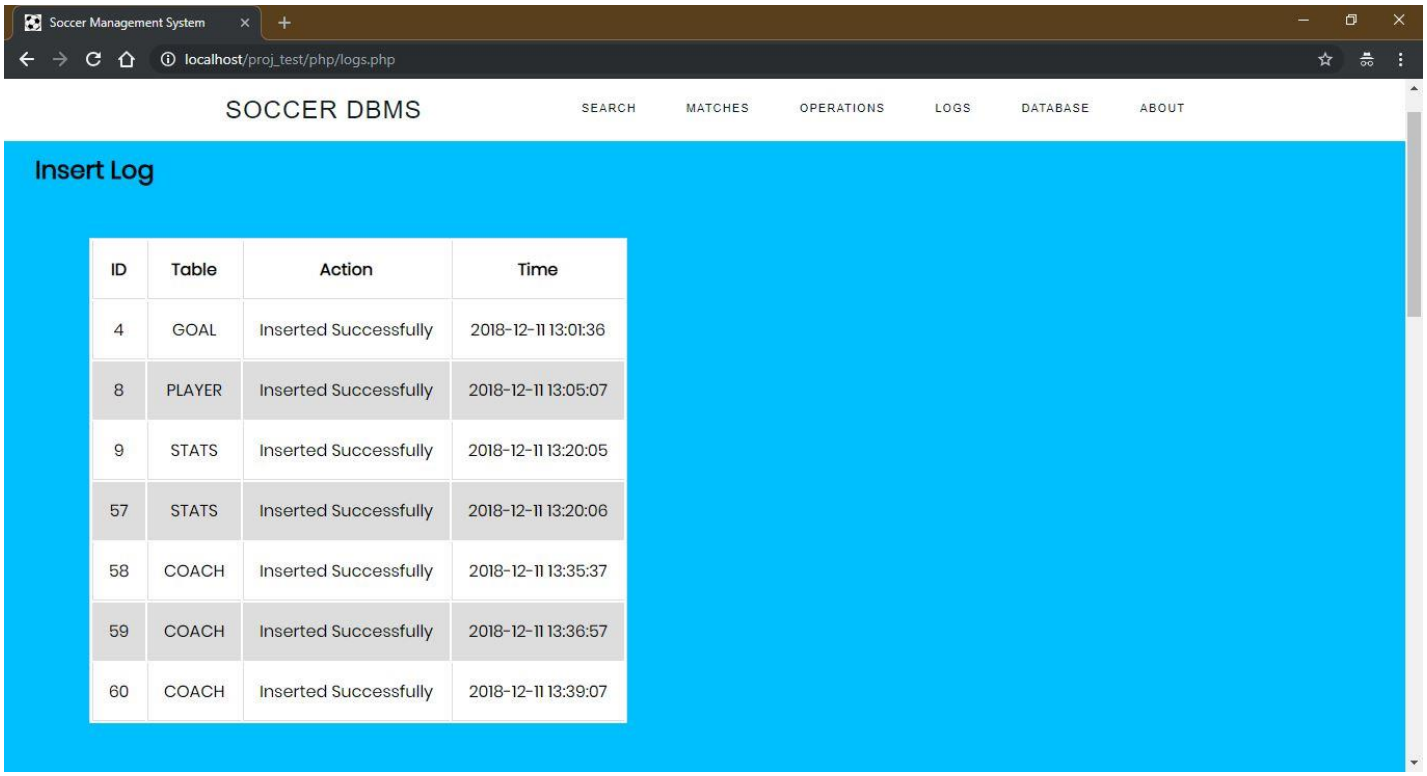


Fig 6.11 Selecting the details of the record to be deleted



ID	Table	Action	Time
4	GOAL	Inserted Successfully	2018-12-11 13:01:36
8	PLAYER	Inserted Successfully	2018-12-11 13:05:07
9	STATS	Inserted Successfully	2018-12-11 13:20:05
57	STATS	Inserted Successfully	2018-12-11 13:20:06
58	COACH	Inserted Successfully	2018-12-11 13:35:37
59	COACH	Inserted Successfully	2018-12-11 13:36:57
60	COACH	Inserted Successfully	2018-12-11 13:39:07

Fig 6.12 Insertion Log

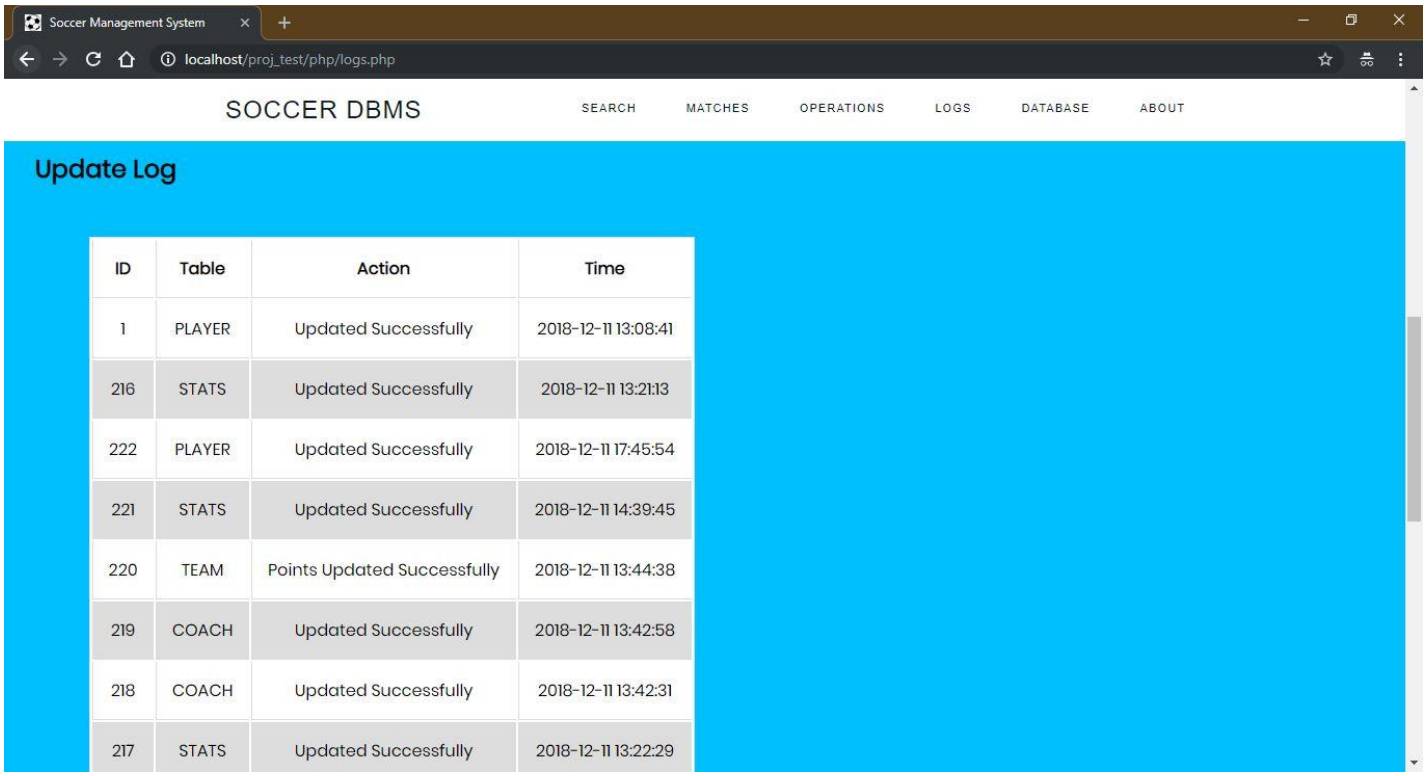


Fig 6.13 Update Log

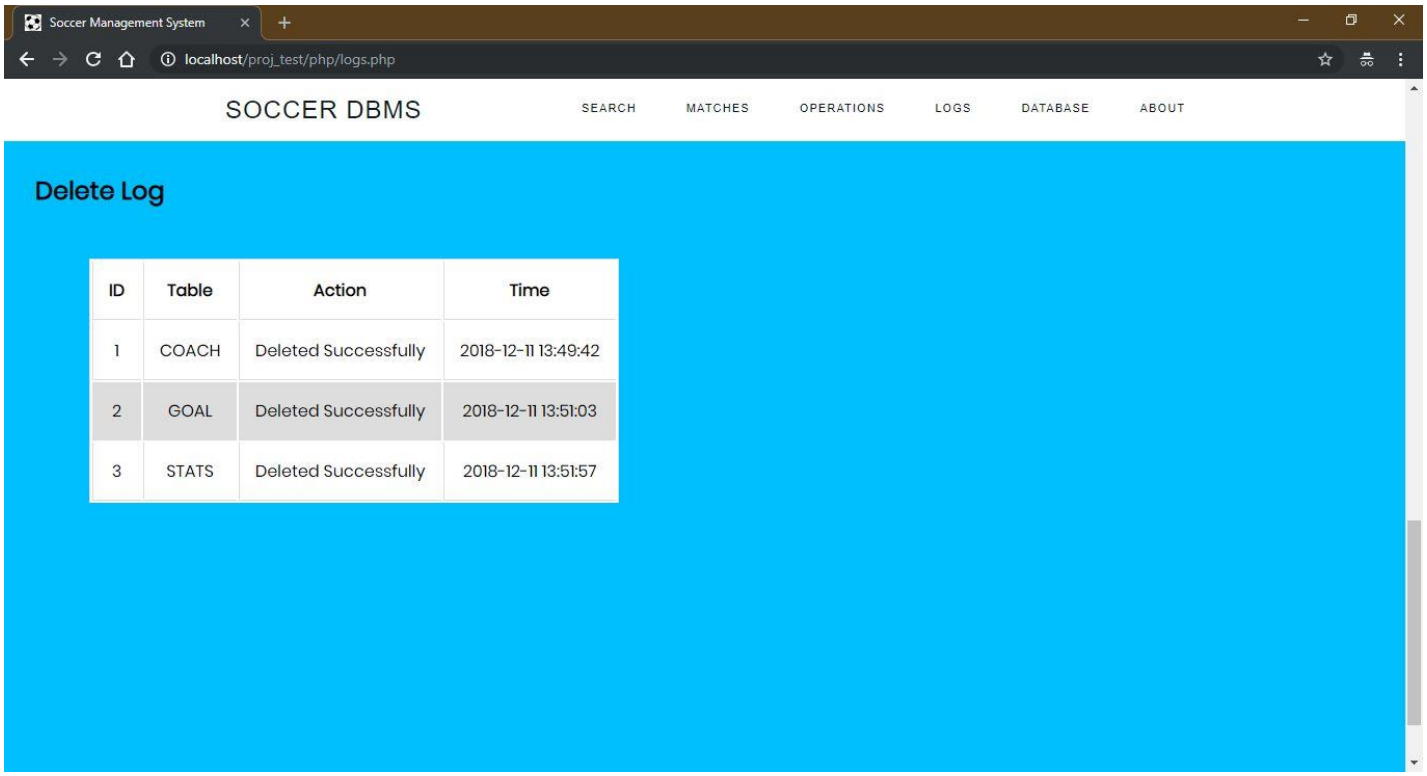


Fig 6.14 Deletion Log

Chapter 7

CONCLUSION & FUTURE SCOPE

In conclusion, it is safe to say that a web-application such as this one, which is built using a secure, swift and robust and a firm enough backend, is close to ideal for managing soccer database. An application such as this one reduces time consumption when it comes to data entry and retrieval. It also reduces storage space in terms of keeping files in extra storage rooms and minimizes the usage of paper. This application also reduces the margin for error when it comes to entering details. Hence, to conclude, the application is a boon to modern day sports database management and will remain so with the right amount of database integrity.

Future Enhancements:

This application has the potential to become a common sight in most database system. With the right alteration and more enhancements to the aesthetics of the application, people would not need to think twice before using it. The application has its many advantages as mentioned above and makes things easier for everyone. By applying concepts of data science and machine learning, various predictions are possible. This reduces a lot of the work and contributes to the research and development that goes on in this field.

BIBLOGRAPHY

[1] Database System Model, Languages, Design and Application Programming, Ramez Elmasri and ShamKant B. Navathe, 7th edition, 2017, Pearson.

[2] Database Management System, Ramakrishnan, and Gehrke, 3rd edition, 2014, McGrawHill.

Websites Referred:

[1] https://stackoverflow.com/

[2] https://www.w3schools.com/

[3] https://www.youtube.com/

[4] https://www.colorlib.com/