VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANASANGAMA, BELGAVI-590018

DBMS MINI PROJECT SYNOPSIS ON
"FIFA PLAYER MANAGEMENT SYSTEM"

Submitted in partial fulfilment of the requirements for the 5th Semester

INFORMATION SCIENCE AND ENGINEERING

Submitted by

NANDAN VASISTA BH	1BI20IS057
POOJA HARIHAR	1BI20IS061

Under the guidance of

Dr. S Mercy
Assistant Professor
Dept of ISE
BIT, Bangalore-04

Mrs. Anupama K.C
Assistant Professor
Dept of ISE
BIT, Bangalore-04

2022-23

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
BANGALORE INSTITUTE OF TECHNOLOGY
K. R. Road, V. V. Puram, Bengaluru-560004

ABSTRACT

Managing the ever increasing numbers of players in different parts of the world is a huge task. This project is aimed at developing a desktop-based application named ‘FIFA player management system’ for managing players using a robust database at the backend and a Web based GUI at the frontend.

The application will allow users to track complete details about a player starting from his personal details, going through club and nationality information to right down to his technicalities at each position in footballing world. The software also allows users to view the whole list of players, teams and footballing statistics at once, thereby helping them build their perspective. Users have the privilege to add new players to a particular team, and to modify their records when the player decides to retire. FIFA player management system also allows users to access players based on their rating other than their preferential position of playing thus guiding managers to build a strong positional team by selecting best rated player at each position. In conclusion, this application will come extremely handy in maintaining player spread across different teams and nations.

TABLE OF CONTENTS

Sl. No.	CHAPTERS	PAGE NO.
1	Introduction	1
2	System Requirement	2
3	Design of the project	4
4	Implementation	11
5	Screenshots	54
	Conclusion	61
	Bibliography	62

LIST OF FIGURES

Sl. No.	Figures	PAGE NO.
3.1	ER Diagram	5
3.2	Relational schema	6
3.3	Player table structure	7
3.4	Player stats table structure	7
3.5	Salary table structure	8
3.6	Position table structure	8
3.7	Club table structure	8
3.8	Delete logs trigger structure	9
3.9	Update logs trigger structure	9
3.10	Insert logs trigger structure	9
3.11	Stored procedures structure	10

CHAPTER 1

INTRODUCTION

Database is a collection of data and Management System is a set of programs to store and retrieve those data. Based on this one can define DBMS as a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.

1.1 What is the need of DBMS?

Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data. Storage: According to the principles of database systems, the data is stored in such a way that it acquires lot less space as the redundant data (duplicate data) has been removed before storage.

Fast Retrieval of data: Along with storing the data in an optimized and systematic manner, it is also important to retrieve the data quickly when needed. Database systems ensure that the data is retrieved as quickly as possible. The choice of a database product is often influenced by factors such as:

- the computing platform (i.e., hardware, operating system)
- the volume of data to be managed.
- the number of transactions required per second.
- existing applications or interfaces that an organization may have support for heterogeneous and/or distributed computing.
- Cost
- vendor support

1.2 Problem Statement

The project titled "**FIFA Player management system**" is player management software for monitoring and accessing players based on their FIFA Game ratings. This project is developed using HTML, CSS and JavaScript for front-end and PHP, MySQL for back-end, which focuses on basic operation like adding a new player, new statistics, searching players with detailed information and edit as they grow their skills.

This project is a web based application designed and developed to help user's access players and organize teams. This software is easy to use, and it features a familiar and well- thought-out attractive user interface, combined with strong searching, insertion, and deletion with procedure capabilities.

Analysing players have been a huge task performed by professional scouting agents who are spread around the world. From personal details to football technicalities, FIFA Management system allows easy maintenance record of such skilled youth talent.

CHAPTER 2
SYSTEM REQUIREMENTS

One of the most difficult tasks is that, the selection of the software, once system requirement is known is determining whether a software package fits the requirements. After initial selection further security is needed to determine the desirability of software compared with other candidates. This section first summarizes the application requirement question and then suggests more detailed comparisons.

Hardware Requirement

1. 32/64-bit processor
2. i3 or greater intel processor chip
3. 1.7 or more GHz processor

Software Requirement

1. Windows 7 or higher version OS
2. Google chrome v70.0.3538 or greater
3. XAMPP web server

To analyse and access players on a daily basics can be hectic and huge effort demanding task. To help users, managers, staffs and scouting agents in football world, FIFA player management system provides effortless player management system to help users to analyse, improve, train and access plays on a daily basis.

2.1Description of the Project:

This project consists of player details, which describes about player biodata such as age and nationality. It also consists of player stats which describes about players technical skills. It also consists of tables containing details such as player earnings, club information and preferred position of playing. It also provide a strong searching, updating, deleting and inserting operations with a user friendly web based UI.

The project also helps the users to keep track of the player details in a computerized way without any trouble. The project contains 7 stored procedure**s** and 3 triggers per table. Stored procedures are used in search engine. Every time the user searches through the database, a procedure is called and the results is collected and displayed for the user in a structured manner. It also has 3 trigger namely "**Insert, Delete and Update**" triggers assigned separately to each table. Whenever operations such as insert or delete or update is performed on any table, these triggers are automatically called, and the logs are captured into 3 separate tables, individually for each trigger. Hence use of triggers provides users to trace back all the latest as well as the oldest changes into any table at any point of time.

This project is a simple prototype of managing larger numbers of players across different nations with different skill sets and attributes. It helps to access players and aids in building a strong positional team. It also helps in monitoring player growth.

CHAPTER 3

DESIGN

ER MODEL

(ER Model) is a graphical approach to database design. It is a high-level data model that defines data elements and their relationship for a specified software system. An ER model is used to represent real-world objects.

Entities and Attributes

An Entity is a thing or object in real world that is distinguishable from surrounding environment. An Entity is an object of Entity Type and set of all entities is called as Entity set. Attributes are the properties which define the entity type. The section of the document explains the entities used in the project, their attributes and how they will work together. Basically, this is intended to make the design easier and understandable foreveryone.

3.1 ER diagram

An entity relationship model, also called an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems.

ER Diagram with relationships and cardinality ratio The cardinality or fundamental principle of one data aspect with respect to another is a critical feature. The relationship of one to the other must be precise and exact between each other in

order to explain how each aspect links together. In simple words Cardinality is a way to define the relationship between two entities.

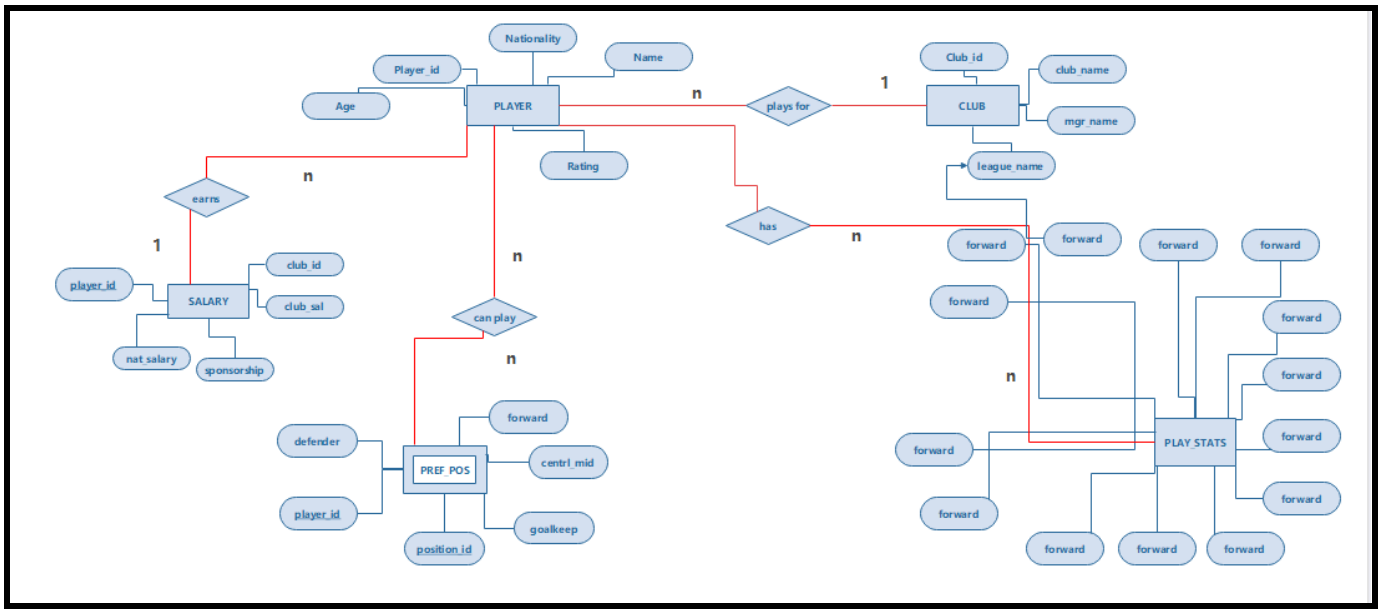


Fig 3.1 ER Diagram of FIFA management system

3.2 SCHEMA diagram

In any data model it is important to distinguish between the description of the database and the database itself. The description of a database is called the database schema, which is specified during database design and is not expected to change frequently. A displayed schema is called a schema diagram. A schema diagram displays only some aspects of a schema, such as the names of record types and data items, and some types of constraints.

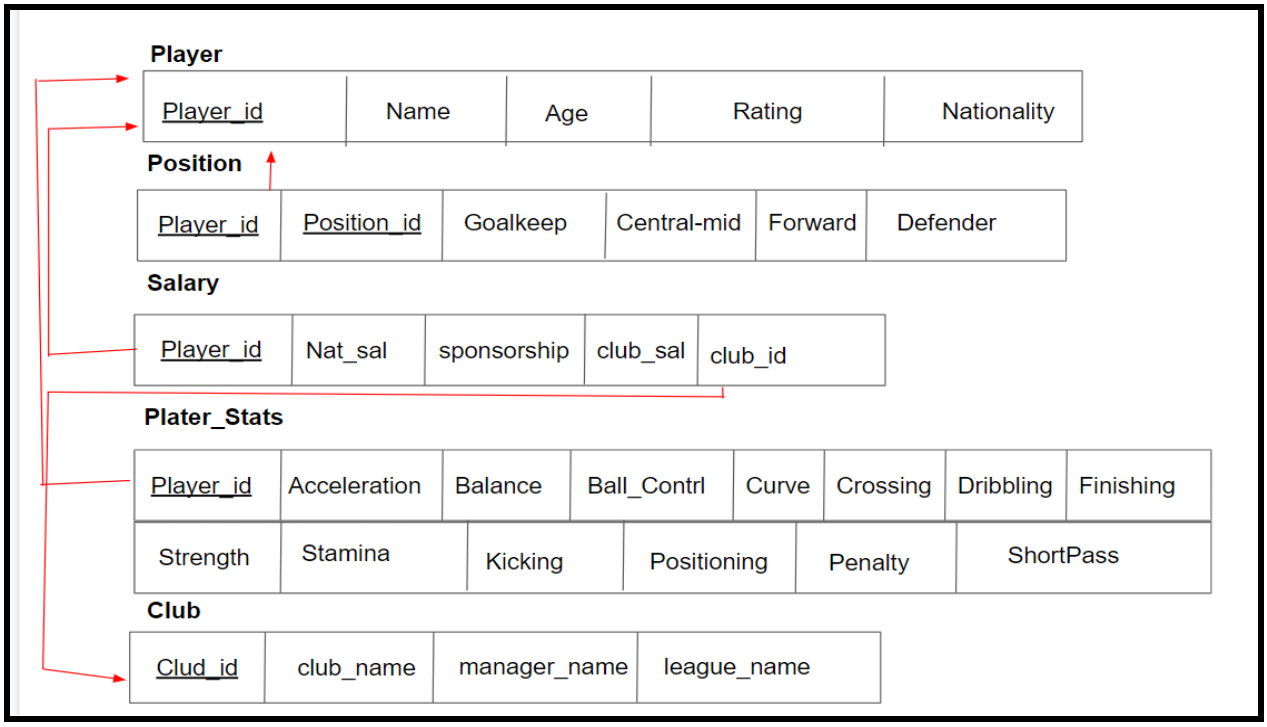


Fig 3.2 Schema Diagram of FIFA management system

3.3 Normalization

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion,

Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.

Normalization is used for mainly two purposes,

- Eliminating redundant(useless) data.
- Ensuring data dependencies make sense

First Normal form

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single(atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain.
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

Second Normal form

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. And, it should not have Partial Dependency

All the tables which are part of this project are in 2NF as they have at most one primarykey, so no partial dependency.

Third Normal form

A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

All the used tables satisfy both these conditions and hence are in 3NF.

3.4Table Structures:

3.4.1 Player details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	player_id	int(7)			No	None			Change Drop More
3	player_name	char(30)	latin1_swedish_ci		No	None			Change Drop More
4	age	int(2)			Yes	NULL			Change Drop More
5	overall_rating	int(2)			Yes	NULL			Change Drop More
6	nationality	char(30)	latin1_swedish_ci		Yes	NULL			Change Drop More

Fig 3.3 Player table structure

The player table consists of 6 columns. Player_id and player_name are primary keys and player_id have references of other tables as well. To insert data, player_id should exist in this table before inserting into other tables.

3.4.2 Stats details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(3)			No	None		AUTO_INCREMENT	Change Drop More
2	player_id	int(7)			No	None			Change Drop More
3	acceleration	int(2)			Yes	NULL			Change Drop More
4	balance	int(2)			Yes	NULL			Change Drop More
5	ball_control	int(2)			Yes	NULL			Change Drop More
6	crossing	int(2)			Yes	NULL			Change Drop More
7	curve	int(2)			Yes	NULL			Change Drop More
8	dribbling	int(2)			Yes	NULL			Change Drop More
9	finishing	int(2)			Yes	NULL			Change Drop More
10	gk_kicking	int(2)			Yes	NULL			Change Drop More
11	gk_positioning	int(2)			Yes	NULL			Change Drop More
12	penalties	int(2)			Yes	NULL			Change Drop More
13	short_pass	int(2)			Yes	NULL			Change Drop More
14	stamina	int(2)			Yes	NULL			Change Drop More
15	strength	int(2)			Yes	NULL			Change Drop More

Fig 3.4 Player stats structure

Player stats table consists of 14 attributes, among which, player_id is primary key and also has a foreign key reference to “player” table. It is designed to contain all the football technicalities of a player.

3.4.3 Salary details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	player_id	int(11)			No	None			Change Drop More
3	wage	int(11)			Yes	NULL			Change Drop More
4	value	int(11)			Yes	NULL			Change Drop More

Fig 3.5 Salary table structure

The salary table consists player_id as primary key and also have a foreign key reference to "player" table. It is designed to store player weekly wage and his current value in the market.

3.4.4 Position details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	player_id	int(11)			No	None			Change Drop More
3	gk	int(11)			Yes	NULL			Change Drop More
4	df	int(11)			Yes	NULL			Change Drop More
5	cm	int(11)			Yes	NULL			Change Drop More
6	fr	int(11)			Yes	NULL			Change Drop More

Fig 3.6 Position table structure

The position table also have player_id as primary key and a foreign key reference to "player" table. It is designed to store the positions a player can play, if so, then how well he does on a scale of rating from 0 to 99. It helps user to access player based on positional play and decide the best position for a player.

3.4.5 Club details

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	player_id	int(11)			No	None			Change Drop More
3	club	char(30)	latin1_swedish_ci		Yes	NULL			Change Drop More
4	preferred_position	char(20)	latin1_swedish_ci		Yes	NULL			Change Drop More

Fig 3.7 Club table structure

The club details table has club information and the preferred position of a player at that club. It also have player_id as primary key and also a foreign key reference on “player” table.

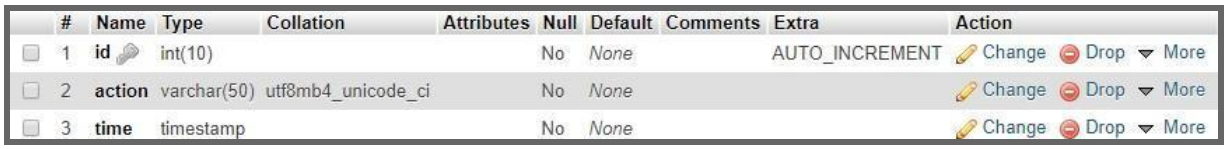
3.4.6 Delete logs

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	action	varchar(50)	utf8mb4_unicode_ci		No	None			Change Drop More
3	time	timestamp			No	None			Change Drop More

Fig 3.8 Delete logs trigger structure

The delete logs table consists of 3 columns. ID column is unique and set to auto increment. Action column contain the action along with table name. Time column contains the time at which the trigger was automatically invoked based on the action.

3.4.7 Update logs

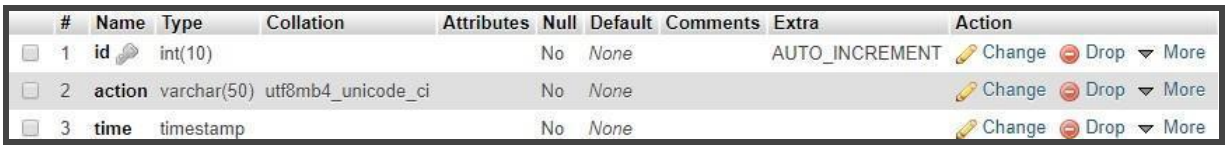


	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	action	varchar(50)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	3	time	timestamp			No	None			Change Drop More

Fig 3.9 Update logs trigger structure

The Update logs table consists of 3 columns. ID column is unique and set to auto increment. Action column contain the action along with table name. Time column contains the time at which the trigger was automatically invoked based on the action.

3.4.8 Insert logs



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	action	varchar(50)	utf8mb4_unicode_ci		No	None			Change Drop More
☐	3	time	timestamp			No	None			Change Drop More

Fig 3.10 Insert logs trigger structure

The Insert logs table consists of 3 columns. ID column is unique and set to auto increment. Action column contain the action along with table name. Time column contains the time at which the trigger was automatically invoked based on the action.

3.4.9 Stored Procedures

	Name	Action				Type
☐	SearchAge	Edit	Execute	Export	Drop	PROCEDURE
☐	SearchName	Edit	Execute	Export	Drop	PROCEDURE
☐	SearchNationality	Edit	Execute	Export	Drop	PROCEDURE
☐	SearchOverallRating	Edit	Execute	Export	Drop	PROCEDURE
☐	SearchPosition	Edit	Execute	Export	Drop	PROCEDURE
☐	SearchTeam	Edit	Execute	Export	Drop	PROCEDURE
☐	Searchplayerid	Edit	Execute	Export	Drop	PROCEDURE

Fig 3.11 Stored procedures structure

There are 7 stored procedures present inside search page on the web application. These are called whenever any search instance occur on the web page. The results of the stored procedures are then displayed on the UI in a tabular structure.

CHAPTER 4

IMPLEMENTATION

4.1 Stored Procedures

The Stored procedures are executed using PHP and MySQL in the application file. The result of stored procedure

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.

```
<section>
<ul class="menu cf">
   <li><a href="../../../../INDEX.html">Home</a></li>
   <li><a href="../../../../search_player/player_search.html">Search</a> </li>
   <li><a href="../../../../update_player/update_player.html">Update</a></li>
   <li><a href="../../../../insert_player/insert_new_player.html">Insert</a></li>
   <li><a href="../../../../database/database.php">Database</a></li>
   <li><a href="../../../procedures.html">Procedures</a></li>
   <li><a href="../../../../about/about.html">About</a></li>
</ul>
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "fifa";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
```

```
if ($conn->connect_error) {

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

}
$choices = $_POST['choices-single-defaul'];

$input_name = $_POST['input_search_keyword'];



#stored procedures

$spforage = mysqli_query($conn, "CREATE DEFINER=`root`@`localhost` PROCEDURE`SearchAge`(IN `page` INT(11)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT player_name,age,overall_rating,nationality FROM personal_details WHERE personal_details.age = page;");
```

explanation- The code uses the mysqli_query function to execute a SQL statement that creates a stored procedure named "SearchAge". The stored procedure takes one input parameter named "page" of type INT(11).The stored procedure is defined with a specific SQL security context "DEFINER" and it is set to the user root@localhost.The stored procedure performs a SELECT statement on the "personal_details" table. It selects the columns "player_name", "age", "overall_rating", and "nationality" from the "personal_details" table where the "age" column is equal to the input parameter "page".This procedure is defined as NOT DETERMINISTIC and CONTAINS SQL, this means that the results of this procedure are not guaranteed to be the same every time it is called, and it may contain SQL statements.In summary, this code creates a stored procedure named "SearchAge" which takes an input parameter of an age, and selects player's name, age, overall rating, nationality from the "personal_details" table where the age column is equal to the input age, and it is executed by root@localhost.

```
$spfornationality = mysqli_query($conn, "CREATE DEFINER=`root`@`localhost` PROCEDURE `SearchNationality`(IN `page` VARCHAR(30)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT * FROM personal_details WHERE personal_details.nationality=page;");
```

```
$spforoverallrating = mysqli_query($conn, "CREATE DEFINER=`root`@`localhost` PROCEDURE `SearchOverallRating`(IN `page` INT(11)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT * FROM personal_details WHERE personal_details.overall_rating = page;");
```

```
$spforteam = mysqli_query($conn, "CREATE DEFINER=`root`@`localhost` PROCEDURE`SearchTeam`(IN `page` VARCHAR(30)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT pd.player_name,pd.overall_rating,pd.age,pd.nationality,od.club FROM personal_details pd,other_details od WHERE od.club = page AND pd.player_id = od.player_id ORDER BY pd.player_id;");


$spforname = mysqli_query($conn, "CREATE DEFINER=`root`@`localhost` PROCEDURE `SearchName`(IN `page` VARCHAR(30)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT * FROM personal_details WHERE player_name = page");


$spforplayerid = mysqli_query($conn, "CREATE DEFINER=`root`@`localhost` PROCEDURE `Searchplayerid`(IN `page` INT(11)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT * FROM personal_details WHERE player_id = page;");


$spforposition = mysqli_query($conn, "CREATE DEFINER=`root`@`localhost` PROCEDURE `SearchPosition`(IN `page` VARCHAR(11)) NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER SELECT pd.player_name, pd.overall_rating, od.preferred_position, p.gk, p.df, p.cm, p.fr FROM personal_details pd, other_details od, position p WHERE od.preferred_position = page AND p.player_id = od.player_id AND p.player_id = pd.player_id GROUP BY pd.player_id;");



if ($choices == 'AGE' && ctype_digit(strval($input_name))) {
   $call = "CALL SearchAge('$input_name')";
?>
<div class="tbl-header">
<table cellpadding="0" cellspacing="0" border="0">
<thead><tr>
      <th>NAME</th>
      <th>AGE</th>
```

```
        <th>OVERALL RATING</th>
        <th>NATIONALITY</th></tr.
    </thead>
    </table>
    </div>
    <div class="tbl-content">
    <table cellpadding="0" cellspacing="0" border="0">
    <tbody>
    <?php
    $result =
    mysqli_query($conn, $call) ;
    if ($result) {
    // output data of each row
            while($row =
                $result->fetch_asso
                c()) { echo "<tr>";
            echo("<td>" . $row["player_name"] . "</td> <td>" . $row["age"] . "</td>
            <td>" .
            $row["overall_rating"] . "</td><td>"
                    .$row["nationality"]. "</td>"); echo
                    "</tr>";
        }
      }
```

```
        }
else if ($choices == 'NATIONALITY' && !ctype_digit(strval($input_name))) {
            $call = "CALL SearchNationality('$input_name')";
        ?>
        </tbody>
        </table>
        </div>
        <div class="tbl-header">
        <table cellpadding="0" cellspacing="0" border="0">
        <thead></tr>
              <th>NAME</th>
              <th>AGE</th>
              <th>OVERALL RATING</th>
              <th>NATIONALITY</th></tr>
        </thead>
        </table>
        </div>
<div class="tbl-content">
        <table cellpadding="0" cellspacing="0" border="0">
        <tbody>
        <?php
        $result =
        mysqli_query($conn, $call) ;
        if ($result) {
```

```
        while($row =
            $result->fetch_assoc()) {
            echo "<tr>";

        echo("<td>" . $row["player_name"] . "</td> <td>" . $row["age"] . "</td> <td>"
        .
        $row["overall_rating"] . "</td><td>"
                .$row["nationality"]. "</td>"); echo "</tr>";
        }
    }
}
else if ($choices == 'OVERALL RATING' && ctype_digit(strval($input_name))) {
    $call = "CALL SearchOverallRating('$input_name')";
?>
</tbody>
</table>
</div>
<div class="tbl-header">
<table cellpadding="0" cellspacing="0" border="0">
<thead><tr>
        <th>NAME</th>
        <th>AGE</th>
        <th>OVERALL RATING</th>
        <th>NATIONALITY</th></tr>
</thead>
</table>
```

```
</div>
<div class="tbl-content">
<table cellpadding="0" cellspacing="0" border="0">
<tbody>
<?php
$result =
mysqli_query($conn, $call) ;
if ($result) {
      while($row =
         $result->fetch_assoc()) {
         echo "<tr>";

      echo("<td>" . $row["player_name"] . "</td> <td>" . $row["age"] . "</td> <td>"
      .
      $row["overall_rating"] . "</td><td>"
            .$row["nationality"]. "</td>"); echo "</tr>";

   }
}
}
else if ($choices == 'PLAYER ID' && ctype_digit(strval($input_name))) {
   $call = "CALL Searchplayerid('$input_name')";
?>
</tbody>
</table>
</div>
```

```
<div class="tbl-header">
<table cellpadding="0" cellspacing="0" border="0">
<thead><tr>
    <th>NAME</th>
    <th>AGE</th>
    <th>OVERALL RATING</th>
    <th>NATIONALITY</th>
    </tr>
</thead>
</table>
</div>
<div class="tbl-content">
<table cellpadding="0" cellspacing="0" border="0">
<tbody>
<?php
$result =
mysqli_query($conn, $call) ;
if ($result) {
    while($row =
      $result->fetch_assoc()) {
      echo "<tr>";

    echo("<td>" . $row["player_name"] . "</td> <td>" . $row["age"] . "</td> <td>"
    .
    $row["overall_rating"] . "</td><td>"
```

```
            .$row["nationality"]. "</td>"); echo "</tr>";}

}
}
else if ($choices == 'PLAYER NAME' && !ctype_digit(strval($input_name))) {
   $call = "CALL SearchName('$input_name')";
?>
</tbody>
</table>
</div>
<div class="tbl-header">
<table cellpadding="0" cellspacing="0" border="0">

<thead>
</tr>
      <th>NAME</th>
      <th>AGE</th>
      <th>OVERALL RATING</th>
      <th>NATIONALITY</th>
      </tr>
</thead>
</table>
</div>
<div class="tbl-content">
<table cellpadding="0" cellspacing="0" border="0">
<tbody>
<?php
```

```
$result =
mysqli_query($conn, $call) ;
if ($result) {
      while($row =
         $result->fetch_assoc()) {
         echo "<tr>";

      echo("<td>" . $row["player_name"] . "</td> <td>" . $row["age"] . "</td> <td>"
      .
      $row["overall_rating"] . "</td><td>"
            .$row["nationality"]. "</td>"); echo "</tr>";

   }
}
}
else if ($choices == 'TEAM' && !ctype_digit(strval($input_name))) {
$call = "CALL SearchTeam('$input_name')";
?>
</tbody>
</table>
</div>

<div class="tbl-header">
<table cellpadding="0" cellspacing="0" border="0">
<thead>
<tr>
      <th>NAME</th>
```

```
            <th>AGE</th>
            <th>OVERALL RATING</th>
            <th>NATIONALITY</th>
            <th>TEAM</th>
            </tr>
        </thead>
</table>
        </div>
        <div class="tbl-content">
        <table cellpadding="0" cellspacing="0" border="0">
        <tbody>
        <?php
        $result =
        mysqli_query($conn, $call) ;
        if ($result) {
            while($row =
               $result->fetch_assoc()) {
               echo "<tr>";

            echo("<td>" . $row["player_name"] . "</td> <td>" . $row["age"] . "</td> <td>"
            .
        $row["overall_rating"] . "</td><td>" .$row["nationality"]. "</td><td>" .
        $row["club"] . "</td>");
            echo "</tr>";
          }
        }
```

```
}
else if ($choices == 'PLAYING POSITION' && !ctype_digit(strval($input_name)))
{

   $call = "CALL SearchPosition('$input_name')";

?>

</tbody>

</table>

</div>

<div class="tbl-header">

<table cellpadding="0" cellspacing="0" border="0">

<thead>

<tr>

  <th>NAME</th>

      <th>PLAYING POSITION</th>

      <th>OVERALL RATING</th>

      <th colspan="4">RATING AT OTHER POSITIONS</th>

      </tr>

      <tr>
      <th></th>

      <th></th>

      <th></th>

      <th>GOALKEEPER</th>

      <th>DEFENDER</th>

      <th>CENTER-MID</th>

      <th>FORWARD</th>

      <tr>
```

```
  </thead>
</table>
</div>
<div class="tbl-content">
<table cellpadding="0" cellspacing="0" border="0">
<tbody>
<?php
$result = ysqli_query($conn,
$call) ; if ($result) {
// output data of each row
      while($row =
         $result->fetch_assoc()) {
         echo "<tr>";
      echo("<td>" . $row["player_name"] . "</td> <td>" . $row["preferred_position"]
      . "</td>
<td>" . $row["overall_rating"] . "</td><td>" .$row["gk"]. "</td><td>" . $row["df"]
.
      "</td><td>" . $row["cm"] . "</td><td>" .
            $row["fr"] . "</td>"); echo "</tr>";
      }}}
   else {
      header("Location:index.html");
}
```

```
$conn->close();
    ?>
```

4.2Inserting new records:

The insert page allows users to select the table to insert values into the table. It then asks users to input required data columns for the particular table and upon successful insertion, a new page appears with appropriate message. Upon failure, another page is displayed with appropriate message and a possible solution.

Given below is the code snippet of the insert page which is execute in the application using PHP and MySQL.

```
    <ul class="menu cf">
    <li><a href="../../INDEX.html">Home</a></li>
    <li><a href="../../search_player/player_search.html">Search</a> </li>
    <li><a href="../../update_player/update_player.html">Update</a></li>
    <li><a href="../insert_new_player.html">Insert</a></li>
    <li><a href="../../database/database.php">Database</a></li>
    <li><a href="../../procedures/procedures.html">Procedures</a></li> <li><a
    href="../../about/about.html">About</a></li> </ul>
    <?php
$servername = "localhost";
    $username = "root";
    $password = "";
 // Create connection
    $conn = new mysqli($servername, $username, $password);
    // Check connection
if ($conn->connect_error) {
```

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

        }
        $choices = $_POST["choices-single-defaul"];
 ?>

        <?php

        if($choices=='PERSONAL DETAILS'){
 echo "<form action=\"php_submit/into_personal_details.php\" method=\"post\">";

 echo "<input id=\"input-1s\" type=\"text\" placeholder=\"PlayerID\" required autofocus
name=\"pid\"/>";

        echo "<label for=\"input-1\">";
 echo "<span class=\"label-text\">Player ID</span>"; echo "<span
class=\"nav-dot\"></span>";

echo "</label>";

 echo   "<input          id=\"input-2\"          type=\"text\"    placeholder=\"Player
Name\"          required name=\"pname\"/>";

        echo "<label for=\"input-2\">";

 echo "<span class=\"label-text\">Player Name</span>"; echo "<span
class=\"nav-dot\"></span>";

        echo "</label>";

        echo "<input id=\"input-3\" type=\"text\" placeholder=\"Age\" required

        name=\"page\"/>"; echo "<label for=\"input-3\">";

        echo " <span class=\"label-text\">Player

        Age</span>"; echo " <span

        class=\"nav-dot\"></span>";


        echo "</label>";
```

```
echo "<input id=\"input-4\" type=\"text\" placeholder=\"Overall Rating\" required name=\"poverallrating\"/>";

echo "<label for=\"input-4\">";

echo " <span class=\"label-text\">Overall Rating</span>"; echo " <span class=\"nav-dot\"></span>";

echo "</label>";

echo "<input id=\"input-5\" type=\"text\" placeholder=\"Nationality\"  required name=\"pnationality\"/>";

echo "<label for=\"input-5\">";

echo " <span class=\"label-text\">Nationality</span>";

echo " <span class=\"nav-dot\"></span>";

echo "</label>";

echo "<button type=\"submit\" onclick=\"php_submit/into_personal_details.php\" class=\"signup-button\">INSERT</button>";
```

```
echo "<p class=\"tip\">Press

Tab</p>"; echo "</form>";


}

else if($choices=='PLAYER CLUB'){

echo "<form action=\"php_submit/into_other_details.php\" method=\"post\">";

echo "<input id=\"input-1s\" type=\"text\" placeholder=\"PlayerID\" required

autofocus name=\"pid\"/>";


echo "<label for=\"input-1\">";
  echo "<span class=\"label-text\">Player ID</span>"; echo "<span
class=\"nav-dot\"></span>";

echo "</label>";

echo "<input id=\"input-2\" type=\"text\" placeholder=\"Player
Club\" required name=\"pclub\"/>";

echo "<label for=\"input-2\">";
echo "<span class=\"label-text\">Player Club</span>";

echo "<span

class=\"nav-dot\"></span>";

echo "</label>";

echo "<input id=\"input-3\" type=\"text\" placeholder=\"Preferred

Position\" required name=\"ppreferredposition\"/>";

echo "<label for=\"input-3\">";


echo " <span class=\"label-text\">Preferred
```

```
Position</span>"; echo " <span

class=\"nav-dot\"></span>";


echo "</label>";

 echo "<button type=\"submit\"
onclick=\"php_submit/into_other_details.php\"
class=\"signup-button\">INSERT</button>";

echo "<p class=\"tip\">Press

Tab</p>"; echo "</form>";}



else if($choices=='PLAYER POSITION'){

echo "<form action=\"php_submit/into_position.php\" method=\"post\">";

 echo "<input id=\"input-1s\" type=\"text\" placeholder=\"PlayerID\" required autofocus
name=\"pid\"/>";

echo "<label for=\"input-1\">";
 echo "<span class=\"label-text\">Player ID</span>"; echo "<span
class=\"nav-dot\"></span>";

echo "</label>";
echo "<input id=\"input-2\" type=\"text\" placeholder=\"Player@GoalKeeper\"
required name=\"pgk\"/>";

echo "<label for=\"input-2\">";
echo "<span class=\"label-text\">Player@Goalkeeper

Rating</span>"; echo "<span class=\"nav-dot\"></span>";


echo "</label>";

 echo   "<input          id=\"input-3\"          type=\"text\"
```

```
placeholder=\"Player@Defender\"          required name=\"pdf\"/>";

        echo "<label for=\"input-3\">";
 echo " <span class=\"label-text\">Player@Defender Rating</span>"; echo " <span
class=\"nav-dot\"></span>";

        echo "</label>";


        echo "<input id=\"input-4\" type=\"text\" placeholder=\"Player@Central-Mid\"
        required name=\"pcm\"/>";

        echo "<label for=\"input-4\">";

 echo " <span class=\"label-text\">Player@Central-mid Rating</span>"; echo " <span
class=\"nav-dot\"></span>";

        echo "</label>";

 echo   "<input          id=\"input-5\"          type=\"text\"
placeholder=\"Player@Forward\"          required name=\"pfr\"/>";

        echo "<label for=\"input-5\">";
 echo " <span class=\"label-text\">Player@Forward Rating</span>"; echo " <span
class=\"nav-dot\"></span>";

        echo "</label>";
 echo "<button type=\"submit\" onclick=\"php_submit/into_position.php\"
class=\"signup- button\">INSERT</button>";

        echo "<p class=\"tip\">Press

        Tab</p>"; echo "</form>";


        }


        else if($choices=='PLAYER STATS'){
        echo "<form action=\"php_submit/into_player_stats.php\" method=\"post\">";
```

```
echo "<input id=\"input-1s\" type=\"text\" placeholder=\"PlayerID\" required
autofocus name=\"pid\"/>";

echo "<label for=\"input-1\">";
echo "<span class=\"label-text\">Player ID</span>"; echo "<span
class=\"nav-dot\"></span>";
echo "</label>";
echo "<input id=\"input-2\" type=\"text\" placeholder=\"Player Acceleration\" required
name=\"pacceleration\"/>";
echo "<label for=\"input-2\">";

echo "<span
class=\"label-text\">Acceleration</span>";
echo "<span class=\"nav-dot\"></span>";

echo "</label>";
echo "<input id=\"input-3\" type=\"text\" placeholder=\"Player
Balance\" required name=\"pbalance\" />";
echo "<label for=\"input-3\">";
echo " <span class=\"label-text\">Balance</span>"; echo " <span
class=\"nav-dot\"></span>";
echo "</label>";

echo "<input id=\"input-4\" type=\"text\" placeholder=\"Player Ball Control\"
required name=\"pballcontrol\"/>";
echo "<label for=\"input-4\">";
```

```
        echo " <span class=\"label-text\">Ball

        Control</span>"; echo " <span

        class=\"nav-dot\"></span>";


        echo "</label>";

 echo   "<input          id=\"input-5\"          type=\"text\"          placeholder=\"Player
Crossing\"      required name=\"pcrossing\"/>";

        echo "<label for=\"input-5\">";
 echo " <span class=\"label-text\">Crossing</span>"; echo " <span
class=\"nav-dot\"></span>";

        echo "</label>";
 echo   "<input          id=\"input-5\"          type=\"text\"          placeholder=\"Player
Curve\"          required


          name=\"pcurve\"/>";

  echo "<label for=\"input-5\">";

  echo " <span class=\"label-text\">Curve</span>";

  echo " <span class=\"nav-dot\"></span>";

  echo "</label>";


          echo "<label for=\"input-5\">";

          echo " <span class=\"label-text\">Dribbling</span>";

          echo " <span class=\"nav-dot\"></span>";
```

```
echo "</label>";

echo "<label for=\"input-5\">";
echo " <span class=\"label-text\">Finishing</span>"; echo " <span
class=\"nav-dot\"></span>";

echo "</label>";

echo "<input id=\"input-5\" type=\"text\" placeholder=\"Player GK Kicking\" required
name=\"pgkk\"/>";

echo "<label for=\"input-5\">";


echo " <span class=\"label-text\">GK

kicking</span>"; echo " <span

class=\"nav-dot\"></span>";


echo "</label>";

echo "<input id=\"input-5\" type=\"text\" placeholder=\"Player GK Positioning\"

required name=\"pgkp\"/>";


echo "<label for=\"input-5\">";
echo " <span class=\"label-text\">GK Positioning</span>"; echo " <span
class=\"nav-dot\"></span>";

echo "</label>";

echo "<input id=\"input-5\" type=\"text\" placeholder=\"Player

Penalties\" required name=\"ppenalties\"/>";

echo "<label for=\"input-5\">";

echo " <span class=\"label-text\">Penalties</span>"; echo " <span
class=\"nav-dot\"></span>";
```

```
echo "</label>";

echo "<input id=\"input-5\" type=\"text\" placeholder=\"Player Short Pass\" required
name=\"pshortpass\"/>";

echo "<label for=\"input-5\">";
echo " <span class=\"label-text\">Short

pass</span>"; echo " <span

class=\"nav-dot\"></span>";echo "</label>";

echo "<input id=\"input-5\" type=\"text\"
placeholder=\"Player Stamina\" required name=\"pstamina\"/>";

echo "<label for=\"input-5\">";
 echo " <span class=\"label-text\">Stamina</span>"; echo " <span
class=\"nav-dot\"></span>";

echo "</label>"

echo "<input id=\"input-5\" type=\"text\"
placeholder=\"Player Strength\" required name=\"pstrength\"/>";

echo "<label for=\"input-5\">";


echo " <span

class=\"label-text\">Strength</span>";

echo " <span class=\"nav-dot\"></span>";

echo "</label>";


echo "<button type=\"submit\" onclick=\"php_submit/into_player_stats.php\"
class=\"signup- button\">INSERT</button>";
```

```
echo "<p class=\"tip\">Press

Tab</p>"; echo "</form>";


}

else if($choices=='PLAYER EARNINGS'){

echo "<form action=\"php_submit/into_salary.php\" method=\"post\">";


echo "<input id=\"input-1s\" type=\"text\" placeholder=\"PlayerID\" required

autofocus name=\"pid\" />";

echo "<label for=\"input-1\">";

 echo "<span class=\"label-text\">Player ID</span>"; echo "<span
class=\"nav-dot\"></span>";

echo "</label>";

 echo "<input id=\"input-2\" type=\"text\" placeholder=\"Player
Wages\" required name=\"pwage\"/>";

echo "<label for=\"input-2\">";


echo "<span class=\"label-text\">Player

Wage</span>"; echo "<span

class=\"nav-dot\"></span>";

echo "</label>";


echo "<input id=\"input-3\" type=\"text\"

placeholder=\"Player Value\" required name=\"pvalue\"/>";

echo "<label for=\"input-3\">";
```

```
echo " <span class=\"label-text\">Player

Value</span>"; echo " <span

class=\"nav-dot\"></span>";

echo "</label>";

echo "<button type=\"submit\" onclick=\"php_submit/into_salary.php\"
class=\"signup- button\">INSERT</button>";

echo "<p class=\"tip\">Press

Tab</p>"; echo "</form>"

}

else {

header("Location:index.html");

}?>
```

(Inside into_personal_details.php)

```
<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "fifa";

// Create connection

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

```
        // Check connection
        if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);}
        $spid = $_POST["pid"];
 $spname = $_POST["pname"];
 $spage = $_POST["page"];
 $spoverallrating = $_POST["poverallrating"];
        $spnationality = $_POST["pnationality"];
 ?>
<ul class="menu cf">
<li><a href="../../../INDEX.html">Home</a></li>
 <li><a href="../../../search_player/player_search.html">Search</a> </li>
 <li><a href="../../../update_player/update_player.html">Update</a></li>
<li><a href="../../insert_new_player.html">Insert</a></li>
        <li><a href="../../../database/database.php">Database</a></li>
 <li><a href="../../../procedures/procedures.html">Procedures</a></li>
<li><a href="../../../about/about.html">About</a></li>
 </ul>
 <?php
        $sql = "INSERT INTO $dbname.personal_details
        (player_id, player_name, age,overall_rating,nationality)
          VALUES ('$spid','$spname','$spage','$spoverallrating','$spnationality')";
        if ($conn->query($sql) === TRUE) {
        echo "<h3 id=\"result\" style=\"color:white; text-align:center;
```

```
font-family:cursive;\">New record inserted successfully !
Go ahead to <a
href=\"../../../database/database.php\">DATABASE</a> to see for
yourself.</h3>";
} else {
// header("Location:index.html");
echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
```

(Inside into_other_details.php)

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "fifa";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection\
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$spid = $_POST["pid"];
$spclub = $_POST["pclub"];
```

```
$sppreferredposition = $_POST["ppreferredposition"];
?>
<ul class="menu cf">
<li><a href="../../../INDEX.html">Home</a></li>
<li><a href="../../../search_player/player_search.html">Search</a> </li>
<li><a href="../../../update_player/update_player.html">Update</a></li>
<li><a href="../../insert_new_player.html">Insert</a></li>
<li><a href="../../../database/database.php">Database</a></li>
<li><a href="../../../procedures/procedures.html">Procedures</a></li>
<li><a href="../../../about/about.html">About</a></li>
</ul><?php
$sql = "INSERT INTO $dbname.other_details (player_id, club,
preferred_position) VALUES ('$spid','$spclub','$sppreferredposition')";
if ($conn->query($sql) === TRUE) {
echo "<h3 style=\"color:white;text-align:center; font-family:cursive;\">New record inserted successfully ! Go ahead to <a href=\"../../../database/database.php\">DATABASE</a> to see for yourself.</h3>";
} else {
header("Location:inde
x.html");
// echo "Error: " . $sql . "<br>" . $conn->error;
```

```
    }
    $conn->close();
?>
```

4.3 Updating existing records:

The update page allows users to look at a selected table and edit the table live on frontend. The user first selects the table he/she wants to modify. Upon selection, a new page is created with respective table. This table consists of editable rows that can be modified and result can be seen real timeGiven below is a .code snippet of implementation of update page. It is created using PHP, MySQL and JavaScript. Use of JavaScript makes the UI more flexible by providing real time data interaction.

```
<?php
include('db-connect.php');
$row1 = mysqli_query($con,"SELECT * FROM personal_details");
?><!DOCTYPE html>
<html>
<head>
<script src="jquery.tabledit.js"></script>
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/meyer-
reset/2.0/reset.min.css">
<link href="css/menu.css" rel="stylesheet" type="text/css">
<script type="text/javascript">
```

```
$(document).ready(function(){
$('#example1').T
abledit({        url:
'logic-edit-delete.
php', columns: {

identifier: [0, 'player_id'],
editable: [[1, 'player_name'], [2, 'age'],[3, 'overall_rating'],[4, 'nationality']]},
onDraw: function() {
console.log('onDraw()');},
onSuccess: function(data, textStatus,
jqXHR) { console.log('onSuccess(data,
textStatus, jqXHR)'); console.log(data);
console.log(tex
tStatus);
console.log(jq
XHR);

},
onFail: function(jqXHR, textStatus,
errorThrown) { console.log('onFail(jqXHR,
textStatus, errorThrown)');
```

```
        console.log(jqXHR);
        console.log(textSt
        atus);
        console.log(error
        Thrown);

        },
 onAlways: function() { console.log('onAlways()');},
        onAjax: function(action,
        serialize) {
        console.log('onAjax(action,
        serialize)');
        console.log(action);
        console.log(serialize);}});});

        </script>
        <body>
 <ul class="menu cf">
 <li><a href="../../../../INDEX.html">Home</a></li>
 <li><a href="../../../../search_player/player_search.html">Search</a> </li>
 <li><a href="../../../update_player.html">Update</a></li>
<li><a href="../../../../insert_player/insert_new_player.html">Insert</a></li>
<li><a href="../../../../database/database.php">Database</a></li>
```

```
<li><a href="../../../../procedures/procedures.html">Procedures</a></li>

<li><a href="../../../../about/about.html">About</a></li> </ul>

        <input TYPE="button" onClick="history.go(0)" VALUE="Refresh"
        class="input_style">

        <div class="tbl-header">
        <table cellpadding="0" cellspacing="0" border="0" id="example1">
        <tr><th>Id</th><th>Name</th><th>Age</th>

        <th>Overall

        Rating</th><th>Nationality</th></tr>

        <?php while($row = mysqli_fetch_assoc($row1)) {?>
<tr>

        <td><?php echo $row['player_id'];?></td>

        <td><?php echo $row['player_name']; ?></td>

        <td><?php echo $row['age']; ?></td>
<td><?php echo $row['overall_rating']; ?></td>

<td><?php echo $row['nationality']; ?></td></tr><?php } ?>

</table>

        </div>

</div>

$input = filter_input_array(INPUT_POST);

        if ($input['action'] === 'edit')
 {$sql = "UPDATE personal_details SET player_name ='" . $input['player_name'] . "',age
='" . $input['age'] . "',overall_rating ='" . $input['overall_rating'] . "', nationality='" .

        $input['nationality'] . "'" ." WHERE player_id='" . $input['player_id'] . "'";

 mysqli_query($con,$sql);

 }<?php
```

```
$servername = "localhost";
$username = "root";
$password = "";
$database = "fifa";
// Create connection
$conn = new mysqli($servername, $username, $password,$database);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);}
$choices = $_POST["choices-single-defaul"];
if($choices=='PERSONAL DETAILS'){
header("Location:test_personal_details/inline-table
-edit.php");}
   else if($choices=='PLAYER EARNINGS'){
      header("Location:test_player_salary/inline-ta
      ble-edit.php");}
else if($choices=='PLAYER POSITION'){
header("Location:test_player_position/inline-table
-edit.php");}
 else if($choices=='PLAYER CLUB'){
header("Location:test_player_club/inline-table-edi
```

```
t.php");}else if($choices=='PLAYER STATS'){

header("Location:test_player_stats/inline-table-ed

it.php");


}else {

    header("Location:index.html");}?
```

4.4 Triggers

The trigger page allows users to look at the database log of operations such as delete, update and insert.

Use: To improve data integrity, trigger can be used. When an action is performed on data, it is possible to check if the manipulation of the data concurs with the underlying business rules, and thus avoids erroneous entries in a table.Given below is a code snippet of implementation of trigger page. It is created using PHP, MySQL and HTML. Use of HTML makes the UI show database logs in a structured manner.

```
<?php
if($choices=='TRIGGER

PROCEDURE'){ echo "

<h1>INSERT TRIGGERS</h1>";


echo "<div class=\"tbl-header\">";

echo " <table cellpadding=\"0\" cellspacing=\"0\"

border=\"0\">"; echo "<thead>";

echo "<tr>";
echo " <th>ID</th>";


echo "
```

```
<th>ACTION</th>";

echo "

<th>TIME</th>";

echo "</tr>";

echo       "

</thead>"

;   echo   "

</table>";

echo

"</div>";


echo "<div class=\"tbl-content\">";

echo "<table cellpadding=\"0\" cellspacing=\"0\" border=\"0\">";

$sql = "SELECT * FROM insert_logs ORDER BY id";

$result =

$conn->query($sql); if

($result->num_rows

>0 ) {

while($row =

$result->fetch_assoc()) { echo

"<tbody>";


echo "<tr>";
```

```
echo "<td>" .$row["id"]. "</td>";
echo "<td>" .$row["action"]. "</td>";
echo "<td>" .$row["time"].

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


echo "</tbody>";}

}else {

echo "<h3 style=\"text-align:center; font-family:cursive;\">No changes yet !</h3>";

}

echo

"</table>";

echo

"</div><br>

";


echo " <h1>UPDATE TRIGGERS</h1>";

echo "<div class=\"tbl-header\">";

echo " <table cellpadding=\"0\" cellspacing=\"0\"

border=\"0\">"; echo "<thead>";


echo "<tr>";

echo " <th>ID</th>";

echo "

<th>ACTION</th>";

echo "
```

```
<th>TIME</th>";

echo "</tr>";

echo "

</thead>"

; echo "

</table>";

echo "</div>";
echo "<div class=\"tbl-content\">";

echo "<table cellpadding=\"0\" cellspacing=\"0\" border=\"0\">";

$sql = "SELECT * FROM update_logs ORDER BY id";

$result =

$conn->query($sql); if

($result->num_rows

>0 ) {

while($row =

$result->fetch_assoc()) { echo

"<tbody>";


echo "<tr>";

echo "<td>" .$row["id"]. "</td>";

echo "<td>" .$row["action"]. "</td>";

echo "<td>" .$row["time"].

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

```
        echo "</tbody>";
        }}else {
        echo "<h3 style=\"text-align:center; font-family:cursive;\">No changes yet !</h3>";
        }
        echo
        "</table>";
        echo
        "</div><br>
        ";

echo " <h1>DELETE TRIGGERS</h1>";
        echo "<div class=\"tbl-header\">";
        echo " <table cellpadding=\"0\" cellspacing=\"0\"
        border=\"0\">"; echo "<thead>";

        echo "<tr>";
        echo " <th>ID</th>";
        echo "
        <th>ACTION</th>";
        echo "
        <th>TIME</th>";
        echo "</tr>";
        echo       "
```

```
</thead>"

;   echo   "

</table>";

echo

"</div>";


echo "<div class=\"tbl-content\">";

echo "<table cellpadding=\"0\" cellspacing=\"0\" border=\"0\">";

$sql = "SELECT * FROM delete_logs ORDER BY id";

$result =

$conn->query($sql); if

($result->num_rows

>0 ) {

while($row =

$result->fetch_assoc()) { echo

"<tbody>";


echo "<tr>";

echo "<td>" .$row["id"]. "</td>";

echo "<td>" .$row["action"]. "</td>";

echo "<td>" .$row["time"].

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

"</tbody>";
```

```
        }
        }else {
        echo "<h3 style=\"text-align:center; font-family:cursive;\">No changes yet
        !</h3>";}
echo "</table>"; echo "</div><br>";

        }
```

CONCLUSION

This project is developed to nurture the needs of a user/scouting agent to monitor players and inspect their technicalities from every aspect on a football field. This is a computerized version of player management system which will benefit the players as well as the staff of a club.

In this entire process one can search player details, add new skilled players, Update ratings and view all the player statistics. The software takes care data and carefully stores all the player information. It provides security and encapsulation by the use of stored procedures.

FUTURE SCOPE

There is a future scope of this project is to help managers and club staffs to get out the best youth talent across the world. Features like predicting players rating based on their current performances and training sessions helps team staffs to judge players according to the club's needs.

BIBLIOGRAPHY

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, Mc-GrawHill.

Websites Referred:

- https://stackoverflow.com/search?q=insert+table
- https://stackoverflow.com/search?q=view+table
- https://stackoverflow.com/search?q=search+from+table
- www.quora.com
- https://www.w3schools.com/css/default.asp
- https://www.w3schools.com/php/default.asp
- https://www.ibm.com/support/knowledgecenter/en/SSEPEK_10.0.0/apsg/src/tpc/db2z_storedprocedure.html