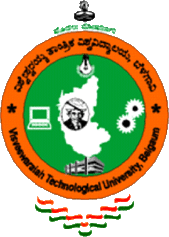
## VISVESVARAYA TECHNOLOGICAL UNIVERSITY BELAGAVI-590018, KARNATAKA



### A MINI PROJECT REPORT

ON

## “FIFA 18 PLAYER MANAGEMENT SYSTEM”

Submitted in partial fulfillment of requirements for the award of 5th semester,

### BACHELOR OF ENGINEERING IN

**COMPUTER SCIENCE & ENGINEERING**

Submitted By: **Praveen kumar (USN: 1MJ16CS105)**

Under the Guidance of

### Mrs. G . Sivagama Sundari Associate Professor,

#### Department of Computer Science & Engineering



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING MVJ COLLEGE OF ENGINEERING**

**BENGALURU-67**

**2018-19**

**MVJ COLLEGE OF ENGINEERING**

**Channasandra, Near ITPB, Bangalore-67 Department of Computer Science and Engineering**



***Certificate***

This is to certify that the mini project entitled **"FIFA 18 Player Management System”** is a bona fide work carried out by **Praveen Kumar (1MJ16CS105),** a bona fide student of MVJ College of Engineering, in partial fulfillment for the award of degree of Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University, Belagavi during the year 2018-19. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the Report. The mini Project Report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said degree.

|  |  |  |
| --- | --- | --- |
| **S ig n a tu r e o f th e Gu id e** |  | **S ig n a tu r e o f t h e HO D** |

#### Signature of the Examiners

***............................................ .........................................***

#### Internal External

**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 18 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 18 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.

# ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

I express my sincere gratitude to our Principal **Dr. Nagaraj Sitaram**, & Vice Principal

**Mrs. M. Brindha**, MVJ College of Engineering for providing facilities.

I wish to place on record my grateful thanks to **Mrs. I Manimozhi**, Head of the Department, Computer Science and Engineering, MVJ College of Engineering, Bangalore for providing encouragement and guidance.

I consider it a privilege and honour to express my sincere gratitude to my guide **Mrs. G Sivagama Sundari,** Associate Professor, Department of Computer Science & Engineering for their valuable guidance throughout the tenure of this seminar work and whose support and encouragement made this work possible.

I wish to thank the faculty of Computer Science and Engineering department whose suggestions have enabled me to surpass many of the seemingly impossible hurdles.

Thank you.

# 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](#_TOC_250001)

[Bibliography 62](#_TOC_250000)

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **Sl. No.** | **Figures** | **PAGE NO.** |
| 2.1 | Memory consumption by brackets web editor | 2 |
| 2.2 | Memory consumption by chrome | 2 |
| 2.3 | Project size on disk | 3 |
| 2.4 | Wamp web server size on disk | 3 |
| 2.5 | Memory consumption by local server (Wamp) | 3 |
| 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 |
| 5.1 | Search input for stored procedures | 54 |
| 5.2 | Search result using stored procedure from frontend | 54 |
| 5.3 | Search result using stored procedure from backend | 55 |
| 5.4 | Selecting table for insertion | 55 |
| 5.5 | Successful insert instance page | 56 |
| 5.6 | Selecting table to modify record | 56 |
| 5.7 | Modifying records in real time | 57 |
| 5.8 | Selecting categories for deleting record | 57 |
| 5.9 | Successful deletion instance | 58 |
| 5.10 | Delete triggers table | 58 |

|  |  |  |
| --- | --- | --- |
| 5.11 | Insert triggers table | 58 |
| 5.12 | Update triggers table | 59 |
| 5.13 | Player club table | 59 |
| 5.14 | Player salary table | 59 |
| 5.15 | Player position table | 60 |
| 5.16 | Player stats table | 60 |
| 5.17 | Player’s table | 60 |

**LIST OF TABLES**

**Sl. No. Table Name Page No.**

* 1. Player table structure 7
  2. Player stats table structure 7
  3. Salary table structure 8
  4. Position table structure 8
  5. Club table structure 8
  6. Player club table 59
  7. Player salary table 59
  8. Player position table 60
  9. Player stats table 60
  10. Player’s table 60

**Chapter 1**

**INTRODUCTION**

The project titled “**FIFA 18 Player management system**” is player management software for monitoring and accessing players based on their FIFA 18 PC/XBOX 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 18 Management system allows easy maintenance record of such skilled youth talent.

**Chapter 2**

**SYSTEM REQUIREMENT**

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. WAMPP web server
4. Brackets web editor

### 2.1 Software and Executables memory size:

þÿ

Fig 2.1 Memory consumption by brackets web editor

þÿ

Fig 2.2 Memory consumption by chrome

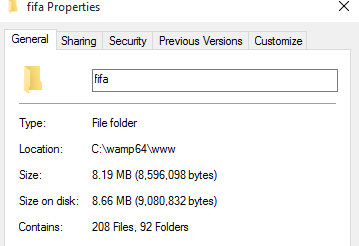


Fig 2.3 Project size on disk

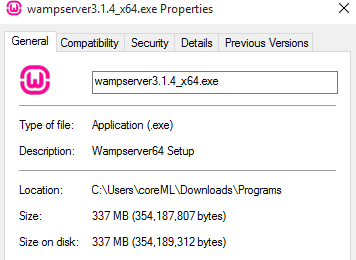


Fig 2.4 Wamp web server size on disk

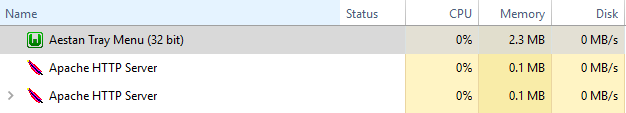


Fig 2.5 Memory consumption by local server (Wamp)

## Chapter 3

**DESIGN OF THE PROJECT**

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 18 player management system provides effortless player management system to help users to analyse, improve, train and access plays on a daily basis.

#### Description 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 procedures** 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 thus aids in building a strong positional team. It also helps in monitoring player growth.

* 1. **ER Diagram:**

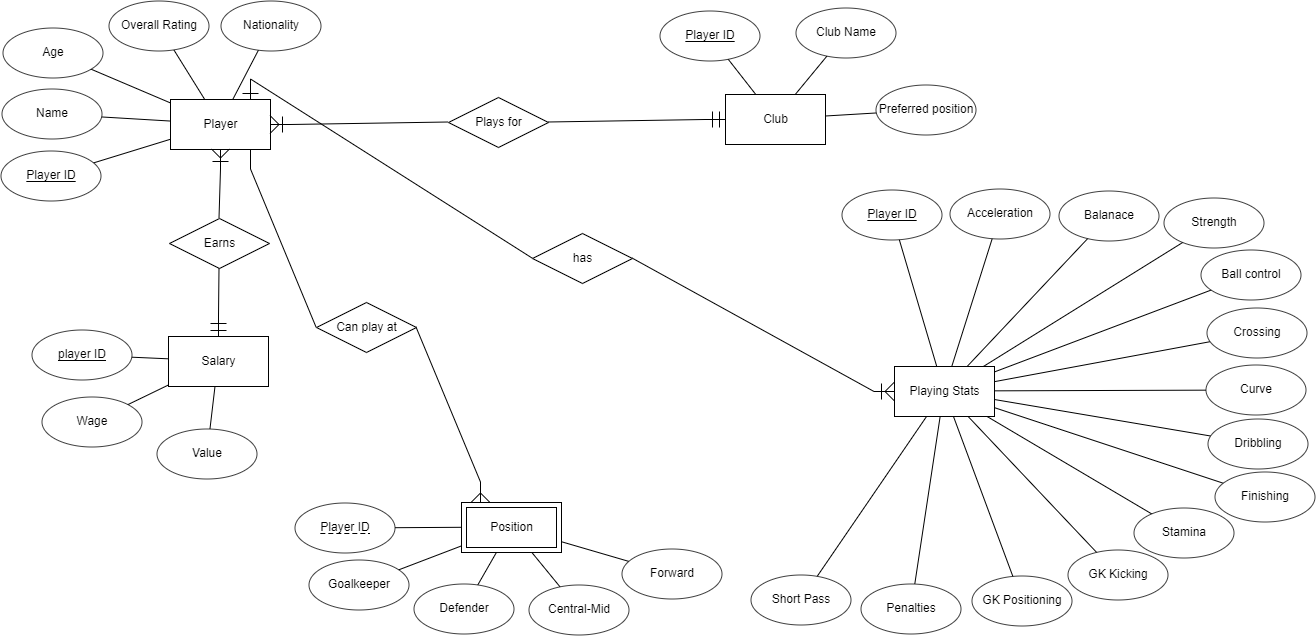


Fig 3.1 ER Diagram

### Schema Diagram:

#### Player

**Playing Stats**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Player ID | Name | Age | Overall rating | Nationality |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | |
|  | **Position**  Player ID  **Salary** | Goalkeeper | Defender | Central-Mid | | Forward |
|  | Player ID | | Wage | | Value | |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Player ID | Acceleration | Balance | Ball control | Curve | Crossing | Dribbling | Finishing |

|  |  |  |
| --- | --- | --- |
| **Club** |  |  |
| Player ID | Club | Preferred position |

Fig 3.2 Relational schema

### Table Structures:

#### Player details

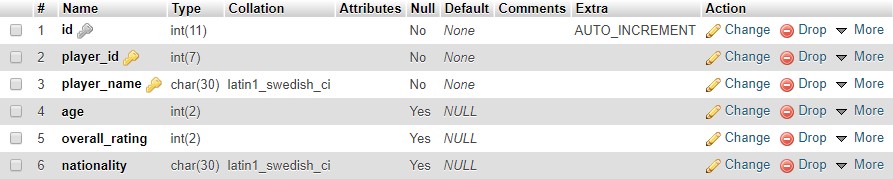


Fig 3.3 Player table structure

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.

#### Stats details

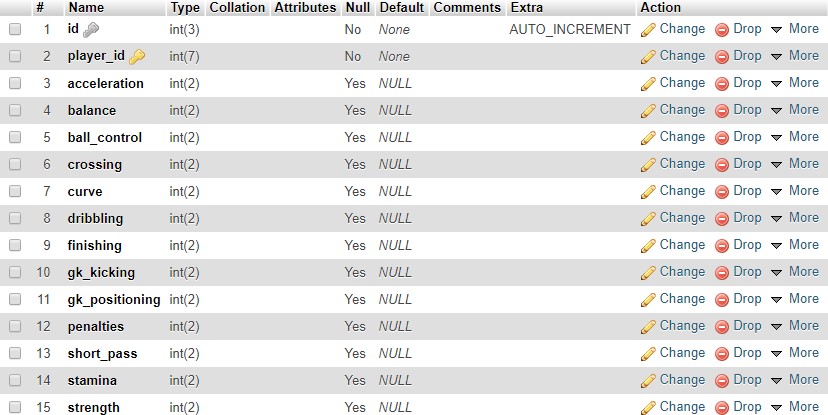


Fig 3.4 Player stats table structure

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.

#### Salary details

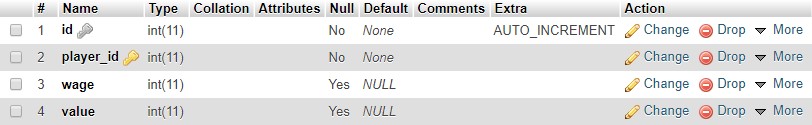


Fig 3.5 Salary table structure

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.

#### Position details

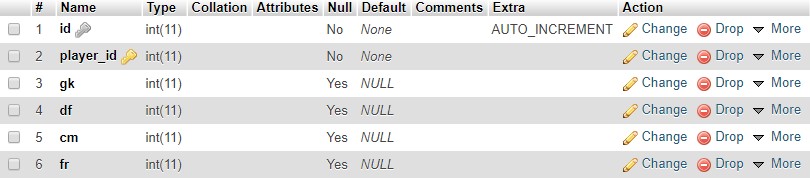


Fig 3.6 Position table structure

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.

#### Club details

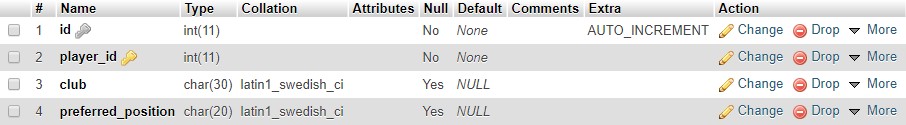


Fig 3.7 Club table structure

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.

#### Delete logs

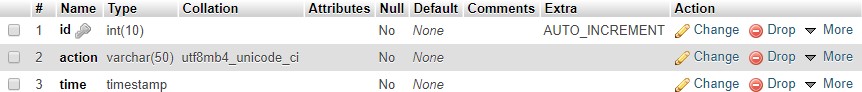


Fig 3.8 Delete logs trigger structure

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.

#### Update logs

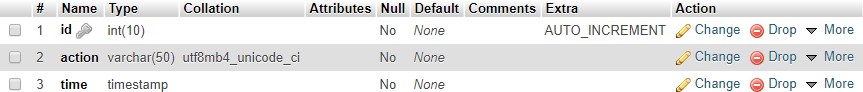


Fig 3.9 Update logs trigger structure

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.

#### Insert logs

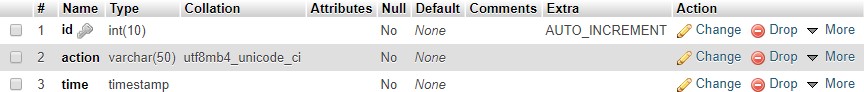


Fig 3.10 Insert logs trigger structure

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.

#### Stored procedures

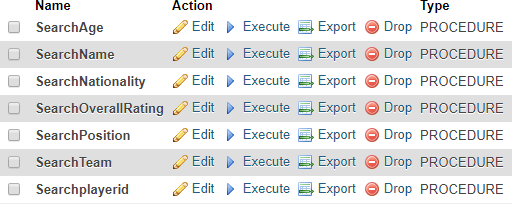


Fig 3.11 Stored procedures structure

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**

### Stored procedures:

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

The result of stored procedure is shown in Fig 5.2.

**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="../../../../report/project\_report.html">Report</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;");

$spfornationality = mysqli\_query($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\_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 ($choices == 'NATIONALITY' && !ctype\_digit(strval($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

$result = mysqli\_query($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();

?>

### Inserting new records:

The insert page allows users to select the table to insert values into [Fig 5.4]. It then asks users to input required data columns for the particular table and upon successful insertion, a new page is displayed [Fig 5.5] 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="../../report/project\_report.html">Report</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 "<input id=\"input-5\" type=\"text\" placeholder=\"Player name=\"pdribbling\"/>"; | Dribbling\" | required |
| echo "<label for=\"input-5\">"; |  |  |
| echo " <span class=\"label-text\">Dribbling</span>"; |  |  |
| echo " <span class=\"nav-dot\"></span>"; |  |  |
| echo "</label>"; |  |  |
| echo "<input id=\"input-5\" type=\"text\" placeholder=\"Player name=\"pfinishing\"/>"; | Finishing\" | required |

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="../../../report/project\_report.html">Report</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="../../../report/project\_report.html">Report</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:index.html");

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

}

$conn->close();

?>

### 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 [Fig 5.6]. 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 time [Fig 5.7].

Given 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').Tabledit({ 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(textStatus); console.log(jqXHR);

},

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

console.log(textStatus); console.log(errorThrown);

},

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="../../../../report/project\_report.html">Report</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="panel panel-default">

<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-table-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-edit.php");

}

else if($choices=='PLAYER STATS'){ header("Location:test\_player\_stats/inline-table-edit.php");

}

else {

header("Location:index.html");

}

?>

### Deleting existing records:

The delete page allows users to delete data based on categories like age, nationality, player ID, name and overall rating [Fig 5.8]. Choosing from different categories provides better user-database interface. Upon deletion, a new page is displayed with appropriate message [Fig 5.9] and upon failing, a new page with error information and possible solutions.

Given below is a code snippet of implementation of delete page. It is created using PHP and MySQL.

<?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=='BY AGE'){

echo "<form action=\"php\_submit\_delete/delete\_byage.php\" method=\"post\">";

echo "<input id=\"input-1s\" type=\"text\" placeholder=\"PLAYER AGE\" required autofocus name=\"page\"/>";

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

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

echo "</label>";

echo "<button type=\"submit\" onclick=\"php\_submit\_delete/delete\_byage.php\" class=\"signup-button\">DELETE</button>";

echo "<p class=\"tip\">Press Tab</p>";

// echo "<div class=\"signup-button\">INSERT</div>"; echo "</form>";

}

else if($choices=='BY NAME'){

echo "<form action=\"php\_submit\_delete/delete\_byname.php\" method=\"post\">";

echo "<input id=\"input-1s\" type=\"text\" placeholder=\"PLAYER NAME\" required autofocus name=\"pname\"/>";

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

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

echo "</label>";

echo "<button type=\"submit\" onclick=\"php\_submit\_delete/delete\_byname.php\" class=\"signup-button\">DELETE</button>";

echo "<p class=\"tip\">Press Tab</p>";

// echo "<div class=\"signup-button\">INSERT</div>"; echo "</form>";

}

else if($choices=='BY PLAYER ID'){

echo "<form action=\"php\_submit\_delete/delete\_byplayerid.php\" method=\"post\">";

echo "<input id=\"input-1s\" type=\"text\" placeholder=\"PLAYER ID\" 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 "<button type=\"submit\" onclick=\"php\_submit\_delete/delete\_playerid.php\" class=\"signup-button\">DELETE</button>";

echo "<p class=\"tip\">Press Tab</p>";

// echo "<div class=\"signup-button\">INSERT</div>"; echo "</form>";

}

else if($choices=='BY OVERALL RATING'){

echo "<form action=\"php\_submit\_delete/delete\_byoverallrating.php\" method=\"post\">";

echo "<input id=\"input-1s\" type=\"text\" placeholder=\"OVERALL RATING\" required autofocus name=\"poverall\_rating\"/>";

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

echo "<span class=\"label-text\">PLAYER OVERALL RATING</span>"; echo "<span class=\"nav-dot\"></span>";

echo "</label>";

echo "<button type=\"submit\" onclick=\"php\_submit\_delete/delete\_byoverallrating.php\" class=\"signup-button\">DELETE</button>";

echo "<p class=\"tip\">Press Tab</p>";

// echo "<div class=\"signup-button\">INSERT</div>"; echo "</form>";

}

else if($choices=='BY NATIONALITY'){

echo "<form action=\"php\_submit\_delete/delete\_bynationality.php\" method=\"post\">";

echo "<input id=\"input-1s\" type=\"text\" placeholder=\"PLAYER NATIONALITY\" required autofocus name=\"pnationality\" />";

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

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

echo "</label>";

echo "<button type=\"submit\" onclick=\"php\_submit\_delete/delete\_bynationality.php\" class=\"signup-button\">DELETE</button>";

echo "<p class=\"tip\">Press Tab</p>";

// echo "<div class=\"signup-button\">INSERT</div>"; echo "</form>";

}

else { header("Location:index.html");

}

?>

<?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);

}

$spname = $\_POST["pname"];

$spages = $\_POST["page"];

$spnationality = $\_POST["pnationality"];

$spid = $\_POST["pid"];

$spoverallrating = $\_POST["poverall\_rating"];

?>

<?php

$sql = "DELETE FROM personal\_details WHERE player\_name=\"$spname\"";

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

echo "<h3 style=\"color:white;text-align:center; font-family:cursive;\"> DELETED 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;

}

$sql = "DELETE FROM personal\_details WHERE age=\"$spages\"";

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

echo "<h3 style=\"color:white;text-align:center; font-family:cursive;\"> DELETED 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;

}

$sql = "DELETE FROM personal\_details WHERE nationality=\"$spnationality\"";

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

echo "<h3 style=\"color:white;text-align:center; font-family:cursive;\"> DELETED 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;

}

$sql = "DELETE FROM personal\_details WHERE overall\_rating=\"$spoverallrating\"";

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

echo "<h3 style=\"color:white;text-align:center; font-family:cursive;\"> DELETED 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;

}

$sql = "DELETE FROM personal\_details WHERE player\_id=\"$spid\"";

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

echo "<h3 style=\"color:white;text-align:center; font-family:cursive;\"> DELETED 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();

?>

### 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>";

}

## Chapter 5

**SCREENSHOTS**

### Stored procedures:

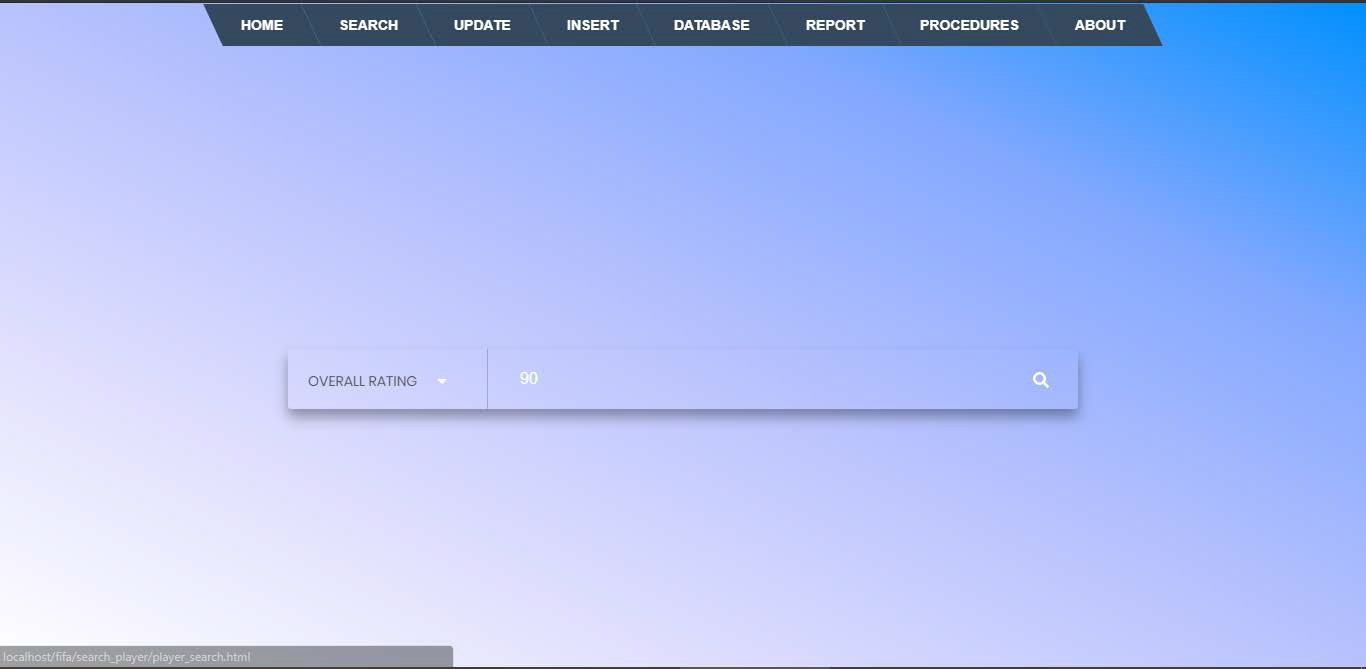


Fig 5.1 Search input for stored procedures.

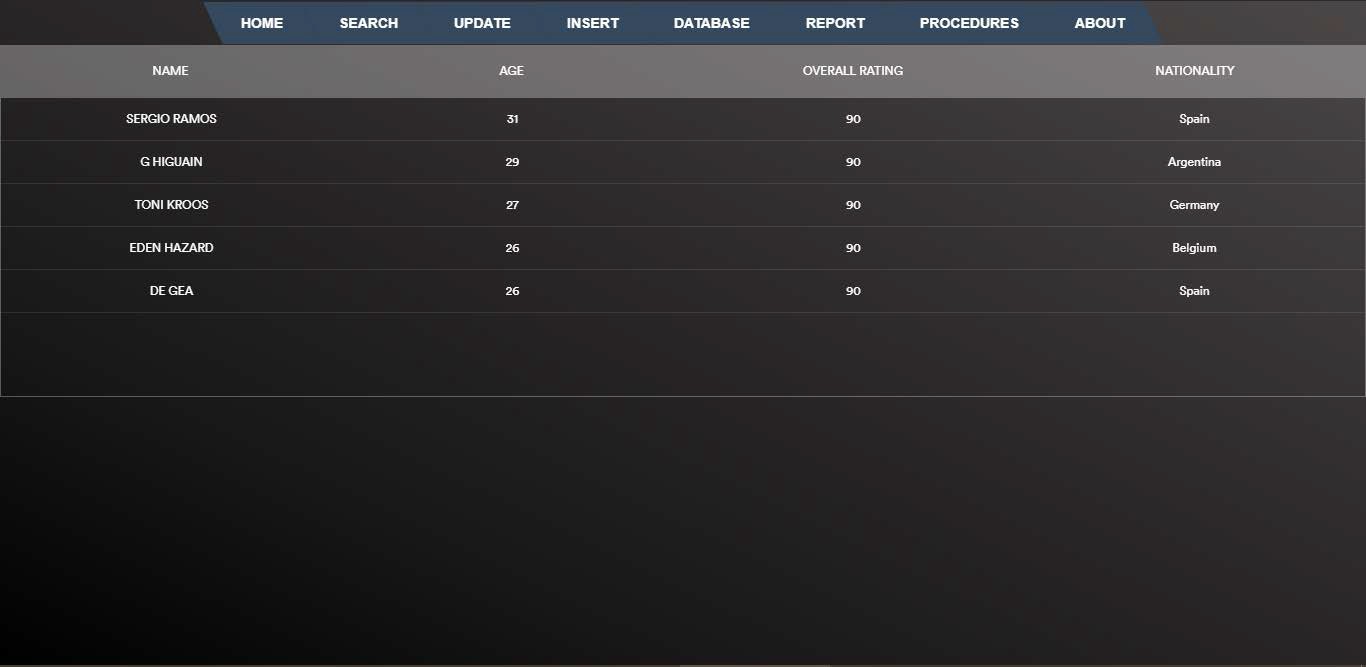


Fig 5.2 Search result using stored procedure from frontend.

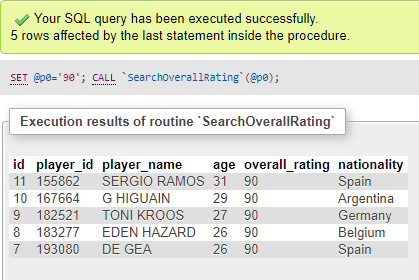


Fig 5.3 Search result using stored procedure from backend.

### Inserting new records:

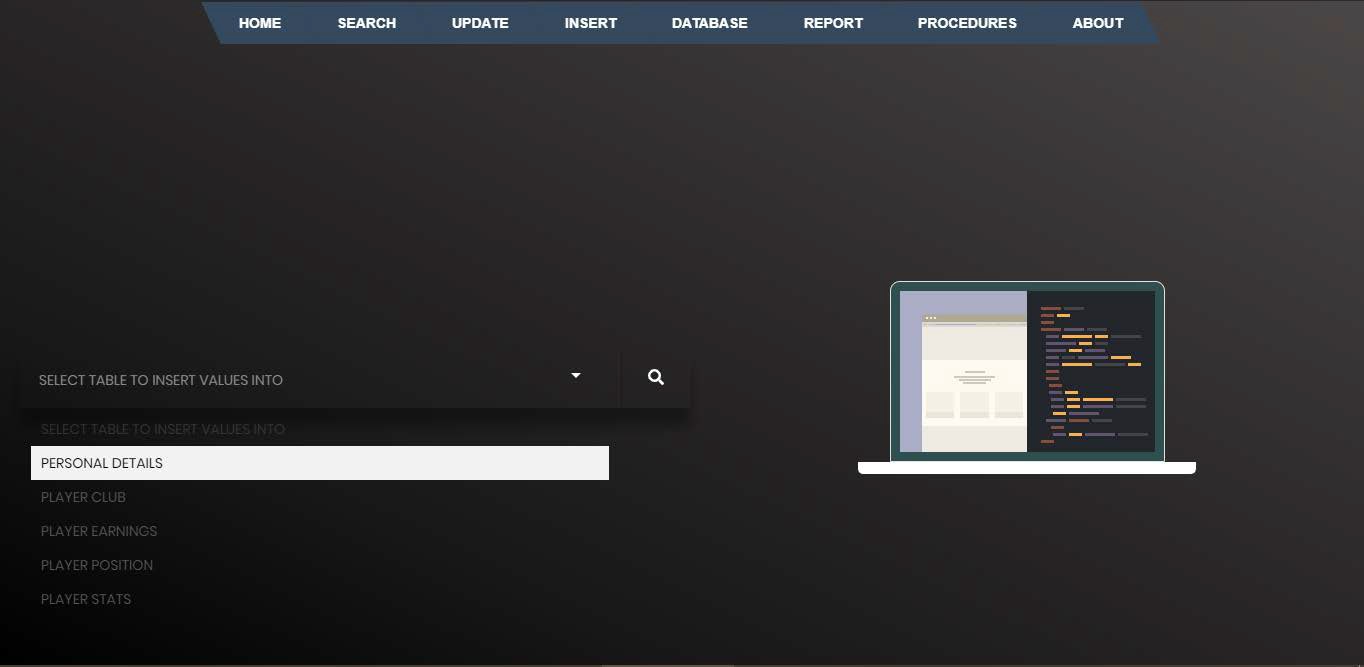


Fig 5.4 Selecting table for insertion

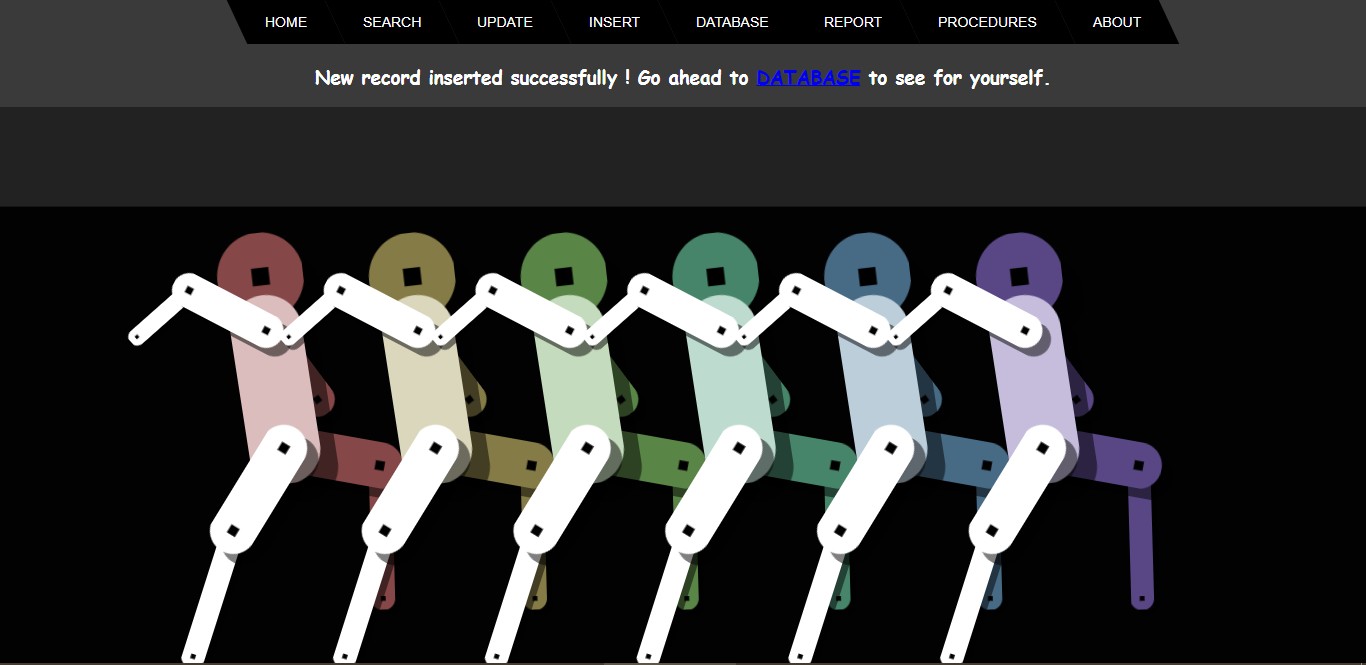


Fig 5.5 Successful insert instance page

### Update existing records:

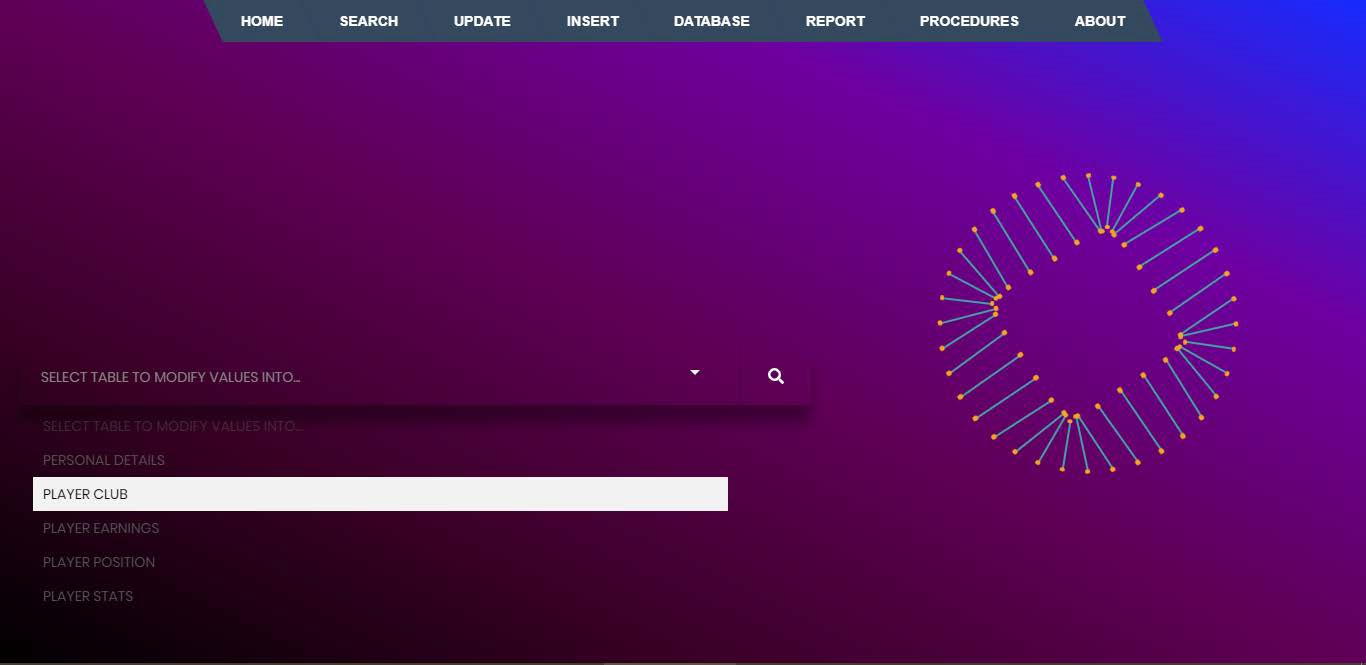


Fig 5.6 Selecting table to modify record

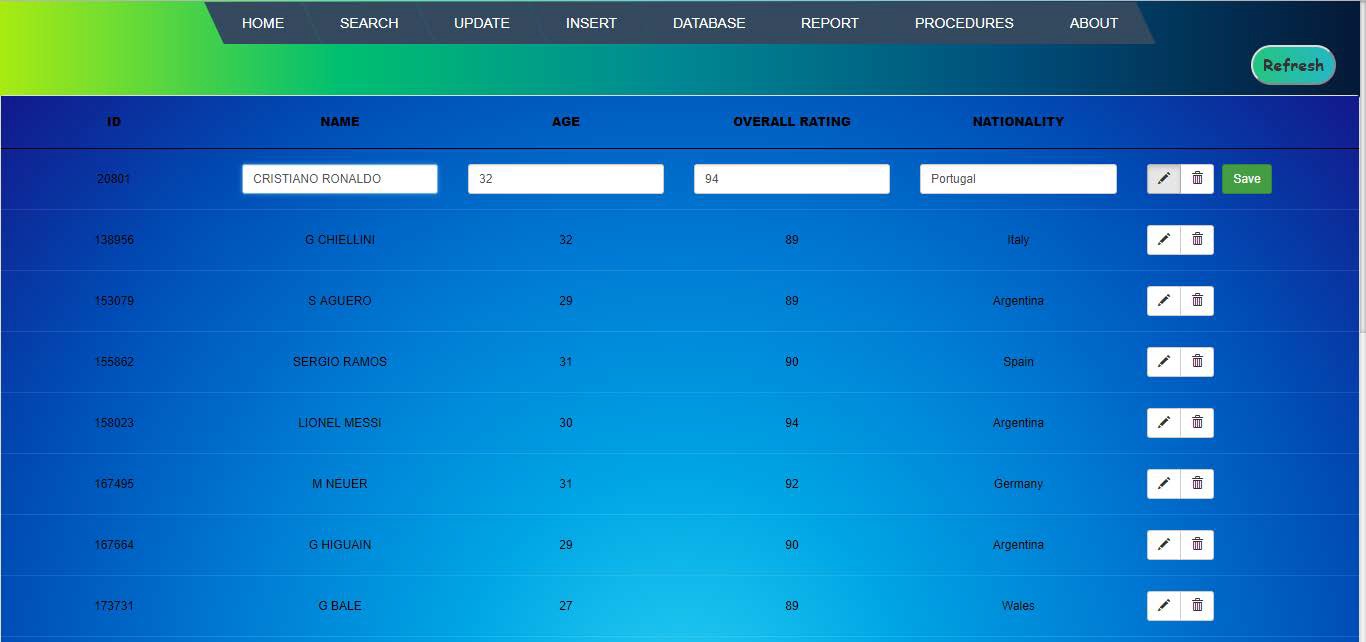


Fig 5.7 Modifying records in real time

### Deleting records:

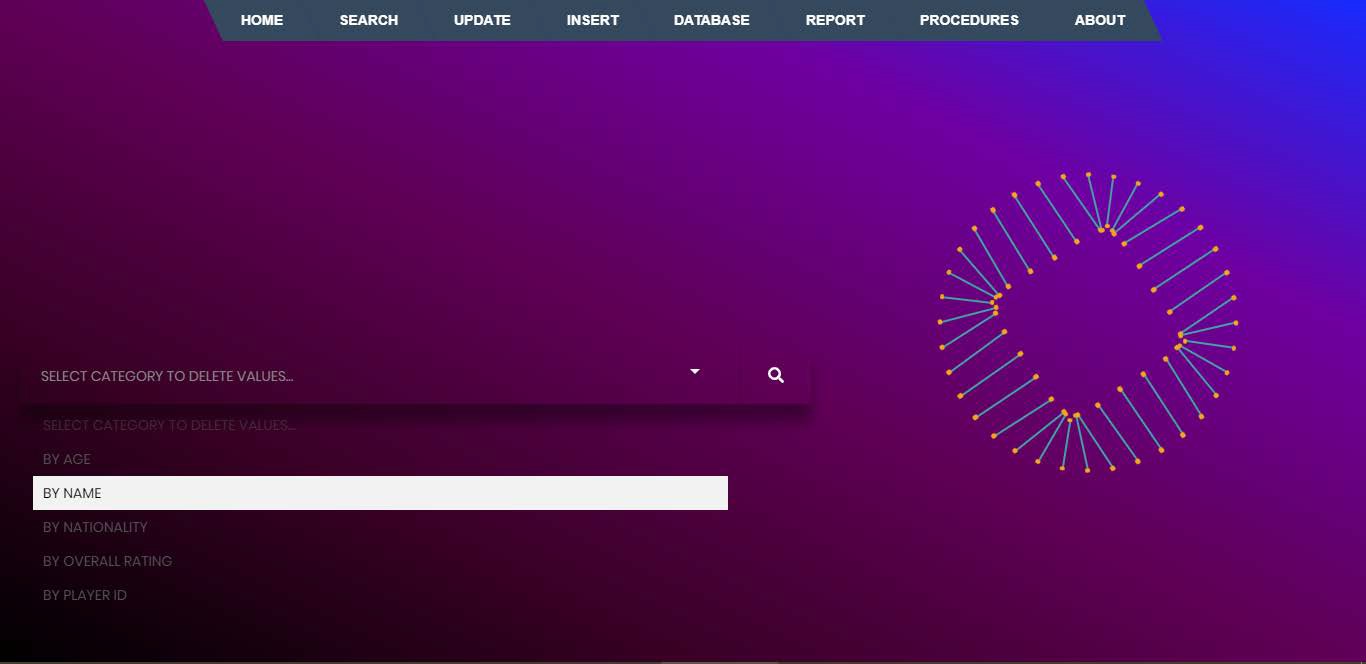


Fig 5.8 Selecting categories for deleting record

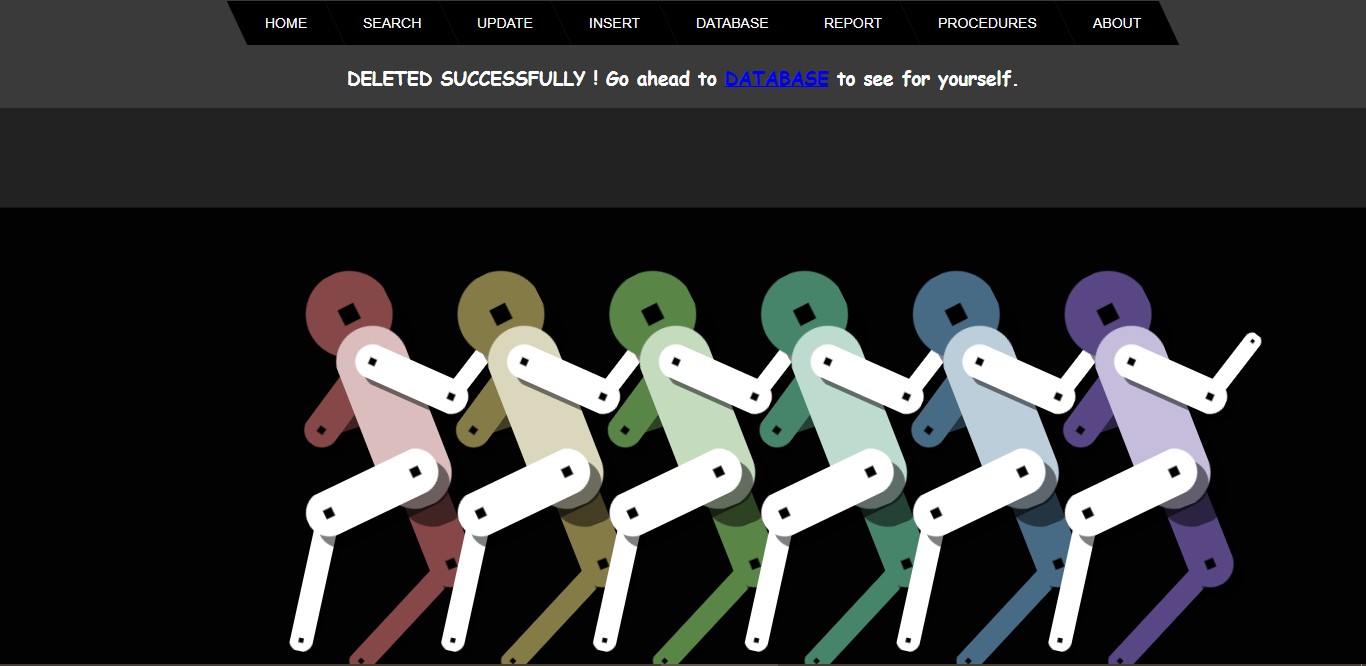


Fig 5.9 Successful deletion instance

### Triggers:



Fig 5.10 Delete triggers table



Fig 5.11 Insert triggers table



Fig 5.12 Update triggers table

### Database tables:

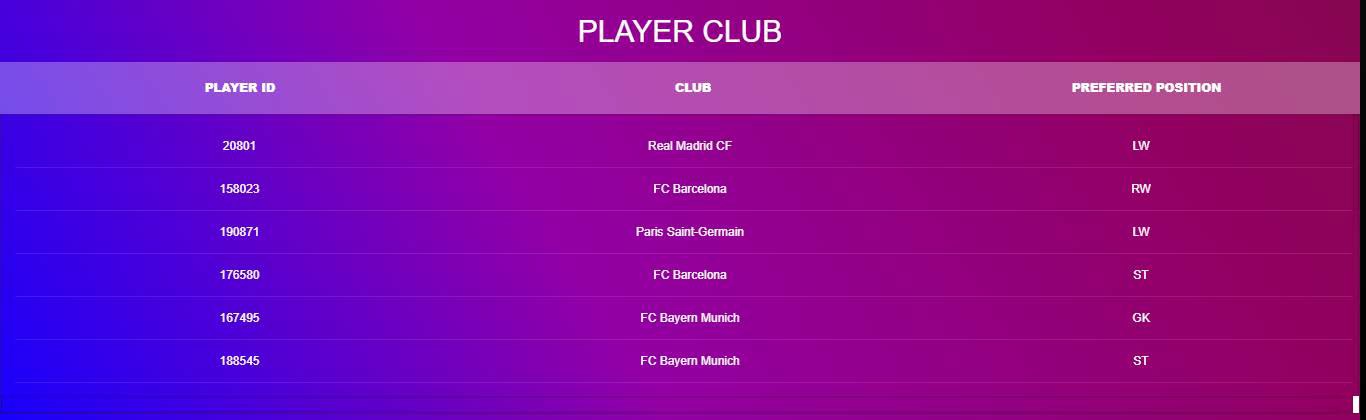


Fig 5.13 Player club table

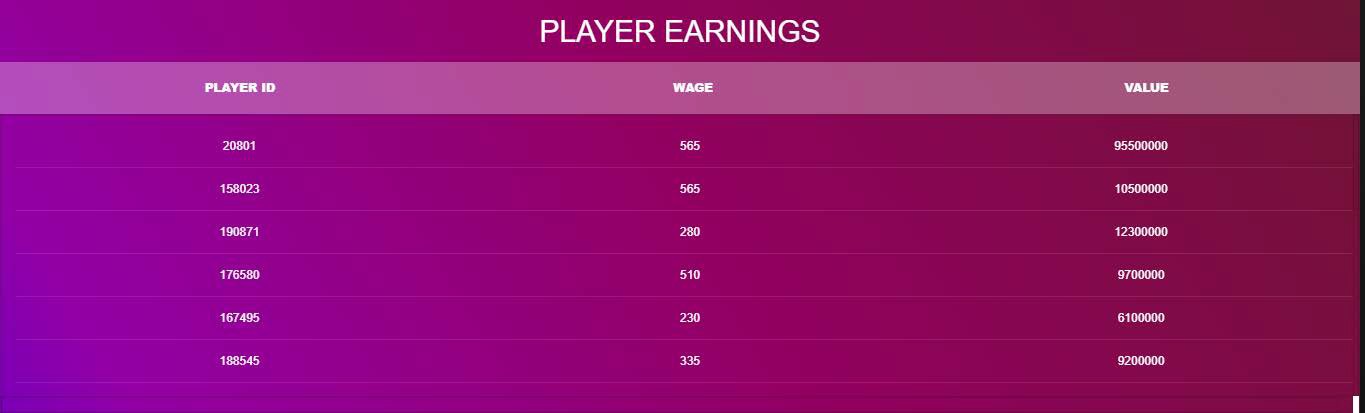


Fig 5.14 Player salary table

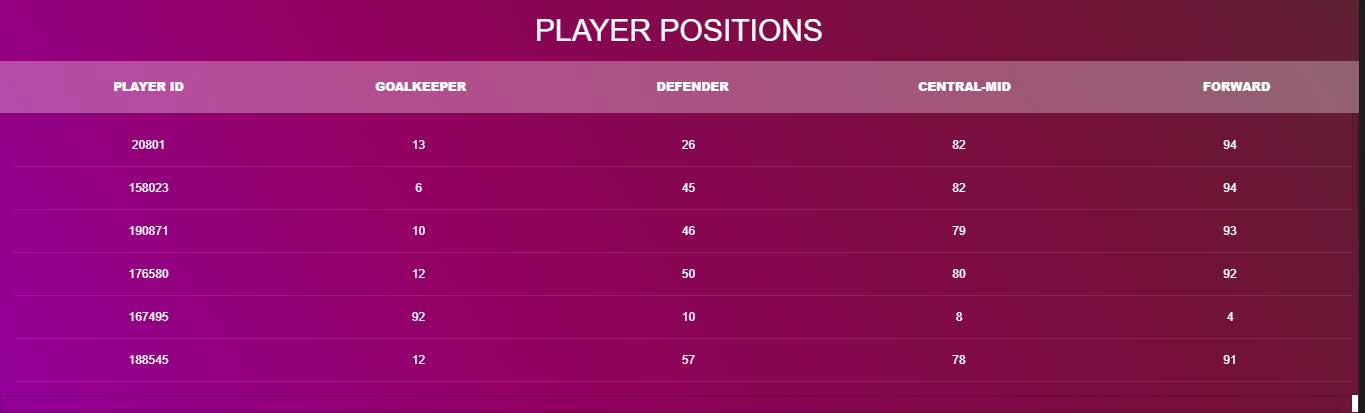


Fig 5.15 Player position table



Fig 5.16 Player stats table

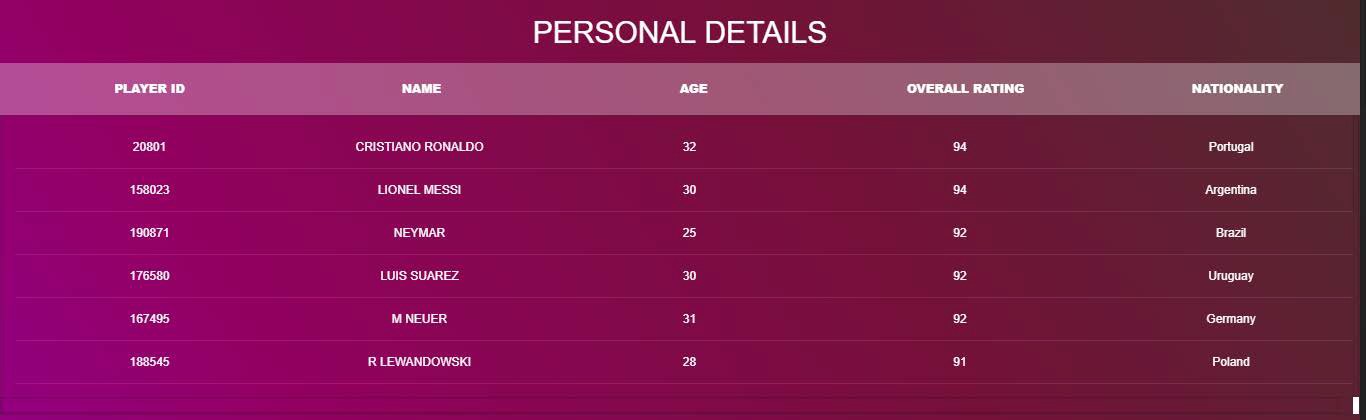


Fig 5.17 Player’s table

# 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=insert%2Btable)
* [https://stackoverflow.com/search?q=view+table](https://stackoverflow.com/search?q=view%2Btable)
* [https://stackoverflow.com/search?q=search+from+table](https://stackoverflow.com/search?q=search%2Bfrom%2Btable)
* [www.quora.com](http://www.quora.com/)
* [https://www.codeproject.com/search.aspx?q=php+code+for+radio+buttons&x=0&y= 0&sbo=qa](https://www.codeproject.com/search.aspx?q=php%2Bcode%2Bfor%2Bradio%2Bbuttons&amp;x=0&amp;y=0&amp;sbo=qa)
* [https://www.codeproject.com/Articles/38808/Overview-of-SQL-Server-database- Triggers](https://www.codeproject.com/Articles/38808/Overview-of-SQL-Server-database-Triggers)
* [https://www.techrepublic.com/article/determine-when-to-use-stored-procedures-vs- sql-in-the-code/](https://www.techrepublic.com/article/determine-when-to-use-stored-procedures-vs-sql-in-the-code/)
* <https://codepen.io/>
* <https://www.uplabs.com/>
* [https://www.ibm.com/support/knowledgecenter/en/SS6NHC/com.ibm.swg.im.dashdb](https://www.ibm.com/support/knowledgecenter/en/SS6NHC/com.ibm.swg.im.dashdb.sql.ref.doc/doc/c0004100.html)

[.sql.ref.doc/doc/c0004100.html](https://www.ibm.com/support/knowledgecenter/en/SS6NHC/com.ibm.swg.im.dashdb.sql.ref.doc/doc/c0004100.html)

* <https://www.w3schools.com/html/default.asp>
* <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](https://www.ibm.com/support/knowledgecenter/en/SSEPEK_10.0.0/apsg/src/tpc/db2z_storedprocedure.html)

[\_storedprocedure.html](https://www.ibm.com/support/knowledgecenter/en/SSEPEK_10.0.0/apsg/src/tpc/db2z_storedprocedure.html)