# ABSTRACT

The Railway Reservation System facilitates the passengers to enquire about the trains available on the basis of source and destination, Booking and Cancellation of rickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers.

This project contains Introduction to the Railways reservation system. It is the computerized system of reserving the seats of train seats in advanced. It is mainly used for long route. Online reservation has made the process for the reservation of seats very much easier than ever before.

In our country India, there are number of counters for the reservation of the seats and one can easily make reservations and get tickets. Then this project contains entity relationship model diagram based on railway reservation system and introduction to relation model. There is also design of the database of the railway reservation system based on relation model. Example of some SQL queries to retrieves data from rail management database.

# TABLE OF CONTENTS

**Sl. No. CHAPTERS PAGE NO.**

1. [Introduction 1](#_bookmark0)
2. [System Requirement 3](#_bookmark1)
3. [Design of the project 5](#_bookmark2)
4. [Implementation 10](#_bookmark3)
5. [Snapshots 26](#_bookmark4)
6. [Conclusion 33](#_bookmark5)
7. [Bibliography 34](#_bookmark6)

# LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Tab. No.** | **Table Name** | **Page No.** |
| Tab 3.1 | Train table structure | 9 |
| Tab 3.2 | Schedule Table structure | 9 |
| Tab 3.3 | Reservation table structure | 10 |
| Tab 3.4 | User table structure | 10 |

**Tab 3.5 Inquires table structure 10**

# LIST OF FIGURES

**Fig. No. Figure Name Page No.**

|  |  |  |  |
| --- | --- | --- | --- |
| Fig 1.1  Fig 2.1 | Components of a Database Management System  Memory Consumption By Bracket Web Editor | 4 | 2 |
| Fig 2.2 | Memory Consumption By Bracket Web Editor | 4 |  |
| Fig 2.3 | Project Size On Disk | 5 |  |
| Fig 2.4 | XAMMP Size On Disk | 5 |  |
| Fig 2.5 | Memory Consumption By Local Server (XAMMP) | 5 |  |
| Fig 3.1 | E R Diagram | 7 |  |
| Fig 3.2 | Schema Diagram | 8 |  |
| Fig 5.1 | Admin Login | 28 |  |
| Fig 5.2 | Selection of table for insertion | 28 |  |
| Fig 5.3 | Successful insert instance Table | 29 |  |
| Fig 5.4 | Selecting table to modify | 30 |  |
| Fig 5.5 | Modifying record in Realtime | 30 |  |
| Fig 5.6 | Selecting a categories for deleting a record | 31 |  |
| Fig 5.7 | Successful delete instance Table | 31 |  |
| Fig 5.8 | Player club table | 32 |  |
| Fig 5.9 | Player salary table | 32 |  |
| Fig 5.10 | Player stats table | 32 |  |
| Fig 5.11 | Player’s table | 33 |  |
| Fig 5.12 | Trigger | 33 |  |

**Chapter 1**

# INTRODUCTION

## Introduction To DBMS:

A database has the following implicit properties:

* A database represents some aspect of the real world, sometimes called the mini world or the universe of discourse. Changes to the mini world are reflected in the database.
* A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
* A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

A database management system (DBMS) is a computerized system that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications. Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalogue or dictionary; it is called meta-data. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world, and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

## Advantages Of DBMS:

A DBMS manages data and has many benefits. These are:

* + - * **Data independence:** Application programs should be as free or independent as possible from details of data representation and storage. DBMS can supply an abstract view of the data for insulating application code from such facts.
      * **Efficient data access:** DBMS utilizes a mixture of sophisticated concepts and techniques for storing and retrieving data competently. This feature becomes important in cases where the data is stored on external storage devices.
      * **Data integrity and security:** If data is accessed through the DBMS, the DBMS can enforce integrity constraints on the data.
      * **Data administration:** When several users share the data, integrating the administration of data can offer significant improvements. Experienced professionals understand the nature of the data being managed and can be responsible for organizing the data representation to reduce redundancy and make the data to retrieve efficiently.

## Components Of DBMS:

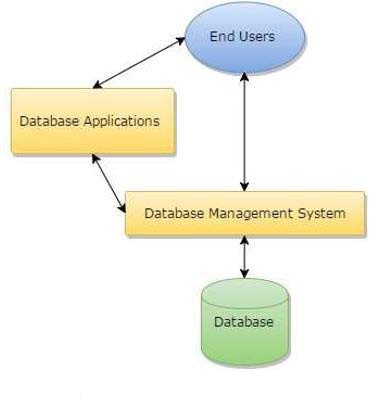


Fig 1.1 Components of a Database Management System

## Introduction To RAILWAY MANAGEMENT SYSTEM:

The project titled “**RAILWAY MANAGEMENT SYSTEM** ” is a ticket booking manager which helps customers to book their tickets online. Railway reservation management refers to the process of booking and managing train tickets for passengers. It involves creating and maintaining a database of available seats on trains, as well as handling the booking and payment process for passengers.

Railway reservation systems can be computerized or manual, and may be operated by the railway company itself or by third-party travel agencies. In many cases, railway reservation systems are integrated with other systems, such as ticketing and scheduling systems, to provide a seamless experience for passengers.

**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. XAMPP web server
4. Brackets web editor

## 2.1 Software and Executables memory size:



Fig 2.1 Memory consumption by brackets web editor



Fig 2.2 Memory consumption by chrome

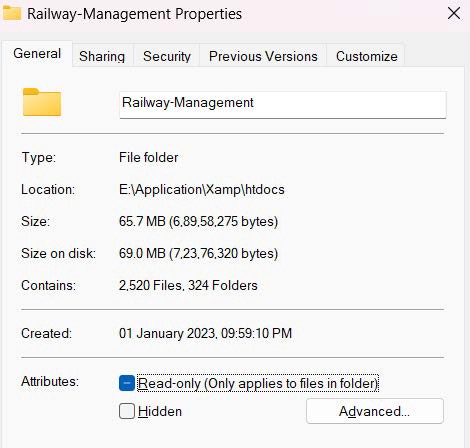


Fig 2.3 Project size on disk

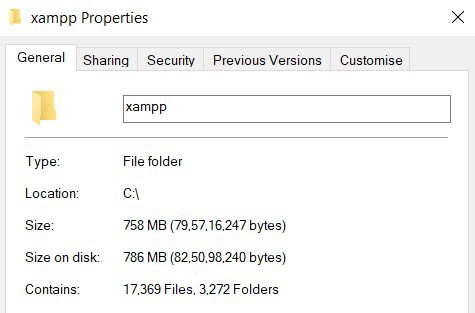


Fig 2.4 xampp web server size on disk

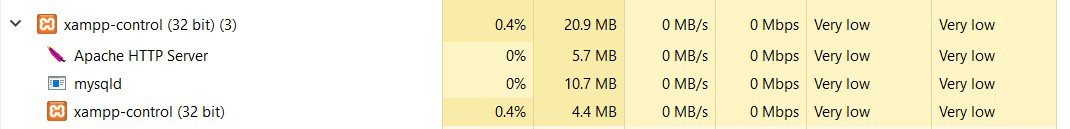


Fig 2.5 Memory consumption by local server (xampp)

**Chapter 3**

# DESIGN OF THE PROJECT

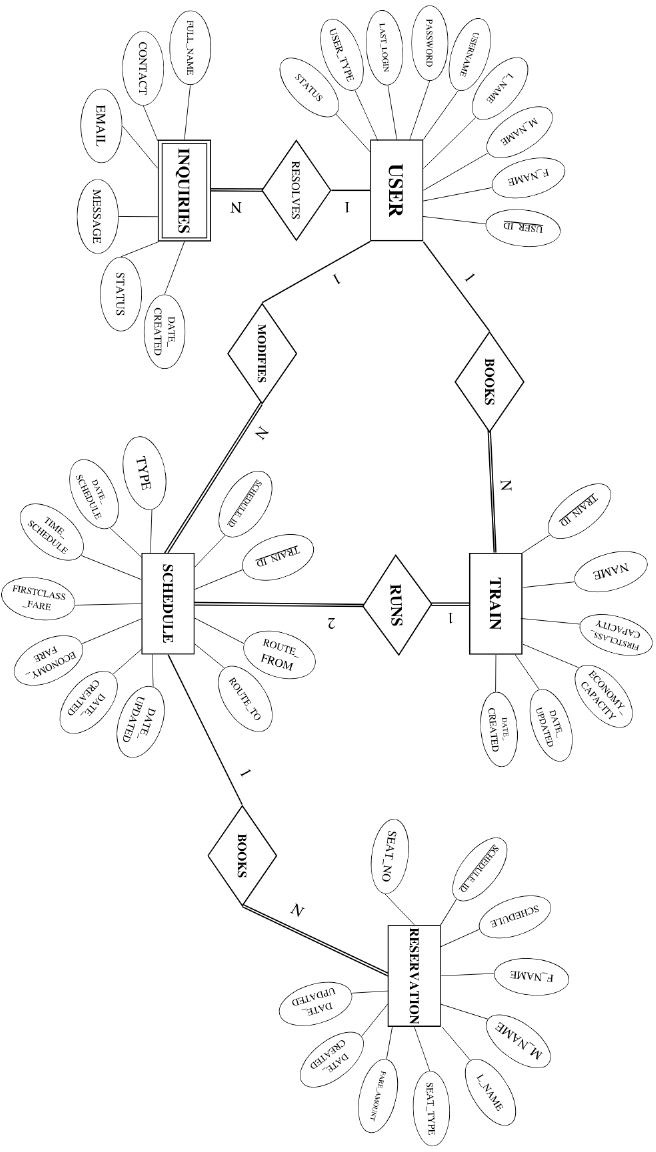
A railway management system is a system that is used to manage a railway network, including the infrastructure, trains, and personnel. It manages the train ticket bookings and seats for customers through a website or an app.

### Description of the Project:

A railway booking management system is a system that is used to manage the reservation and ticketing process for railway travel. It is typically used by railway companies to handle the sale of tickets to passengers, as well as the management of seat assignments and other details related to railway travel.

The system is usually accessed by customers through a website or mobile app, which allows them to search for and book tickets for their desired travel dates and routes. The system is also used by railway employees to manage the availability of seats on trains, as well as to handle ticketing inquiries and other customer service tasks.

* 1. **ER Diagram:**



**Dept of CSE , SJBIT 2022-2023** 6

## Schema Diagram:

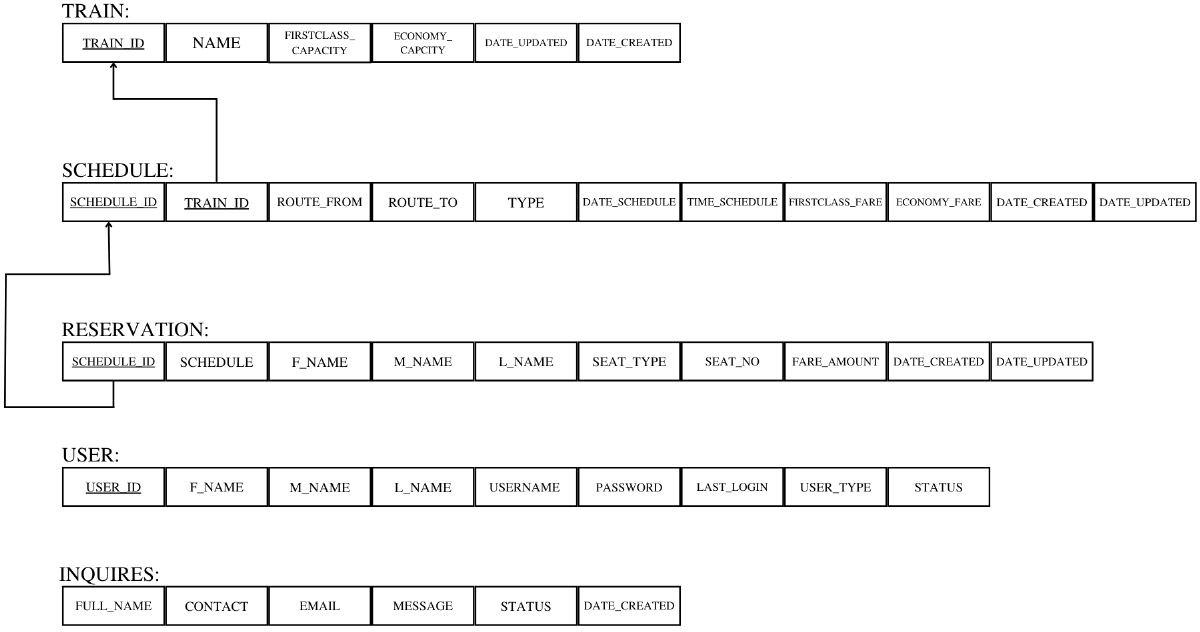


Fig 3.2 Relational schema

## Table Structures:

### train details



Tab 3.1 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.

* + 1. schedule details



Tab 3.2 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.

### reservation details



Tab 3.3 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.

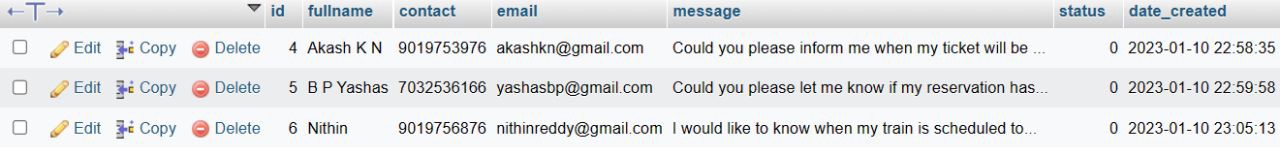
### User details



Tab 3.4 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.

### inquiries details



Tab 3.4Inquiries table structure

**Chapter 4**

# IMPLEMENTATION

## Technologies Used In Building The Project

### HTML

HTML stands for Hypertext Markup Language, it is the standard markup language for creating web pages and web applications. With Cascading Style Sheets(CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receives HTML documents from a Web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages, with HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create standard documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, return using angle brackets. Browsers do not display the HTML tags, but use them to interpret the content of the page.

### CSS

**C**ascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

### PHP

PHP is a servlet-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in I994,

the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor. PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may beany type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

### SQL

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database Management system(RDBMS), or for stream processing in a relational data stream management system (RDSMS). In comparison to older read/write APIs like [SAM or VSAM, SQL offers two main advantages : first, it introduced the concept of accessing many records with one single command; and second, it eliminates the need to specify how to reach a record, e.g. with or without an index. definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages for Edgar F Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large SharedDataBanks".91Despitenot entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SQL became a standard of the American National Standards Institute(ANSI) in 1986 and of the International Organization for Standardization(ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence .of such standards, most SQL code is not completely portable among different data base systems without adjustments.

## Create Table:

SQL comprises both data definition and data manipulation languages. Using the data definition properties of SQL, one can design and modify database schema. **CREATE:** Creates new databases, tables and views from RDBMS. **Syntax:** CREATE TABLE table\_name(attributes list with their databytes);

CREATE TABLE `train\_list` (

`id` int(30) NOT NULL,

`code` varchar(100) NOT NULL,

`name` text NOT NULL,

`first\_class\_capacity` float NOT NULL DEFAULT 0,

`economy\_capacity` float NOT NULL DEFAULT 0,

`delete\_flag` tinyint(1) NOT NULL DEFAULT 0,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp(),

`date\_updated` datetime DEFAULT NULL ON UPDATE current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `schedule\_list` (

`id` int(30) NOT NULL,

`code` varchar(100) NOT NULL,

`train\_id` int(30) NOT NULL,

`route\_from` text NOT NULL,

`route\_to` text NOT NULL,

`type` tinyint(1) NOT NULL DEFAULT 1 COMMENT '1 = daily, 2= One-Time Schedule',

`date\_schedule` date DEFAULT NULL,

`time\_schedule` time NOT NULL,

`first\_class\_fare` float NOT NULL DEFAULT 0,

`economy\_fare` float NOT NULL DEFAULT 0,

`delete\_flag` tinyint(1) NOT NULL DEFAULT 0,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp(),

`date\_updated` datetime DEFAULT NULL ON UPDATE current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `reservation\_list` (

`id` int(30) NOT NULL,

`seat\_num` varchar(50) NOT NULL,

`schedule\_id` int(30) NOT NULL,

`schedule` datetime NOT NULL,

`firstname` text NOT NULL,

`middlename` text NOT NULL,

`lastname` text NOT NULL,

`seat\_type` tinyint(1) NOT NULL DEFAULT 1 COMMENT '1=First Class, 2 = Economy',

`fare\_amount` float NOT NULL DEFAULT 0,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp(),

`date\_updated` datetime DEFAULT NULL ON UPDATE current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `users` (

`id` int(50) NOT NULL,

`firstname` varchar(250) NOT NULL,

`middlename` text DEFAULT NULL,

`lastname` varchar(250) NOT NULL,

`username` text NOT NULL,

`password` text NOT NULL,

`avatar` text DEFAULT NULL,

`last\_login` datetime DEFAULT NULL,

`type` tinyint(1) NOT NULL DEFAULT 0,

`status` int(1) NOT NULL DEFAULT 1 COMMENT '0=not verified, 1 = verified',

`date\_added` datetime NOT NULL DEFAULT current\_timestamp(),

`date\_updated` datetime DEFAULT NULL ON UPDATE current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `message\_list` (

`id` int(30) NOT NULL,

`fullname` text NOT NULL,

`contact` text NOT NULL,

`email` text NOT NULL,

`message` text NOT NULL,

`status` tinyint(1) NOT NULL DEFAULT 0,

`date\_created` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

## Code Snippets:

### CONFIG – TO ESTABLISH CONNECTION

<?php

// session\_start();

//initilizing variables...

$dbServername = "localhost";

$dbUsername = "root";

$dbPassword = "";

$dbName = "orrs\_db";

$reg\_errors = array();

$log\_errors = array();

//connect to database..

$db = mysqli\_connect($dbServername, $dbUsername, $dbPassword, $dbName) or die("!Could not Connect to Database!");

### Admin Login:

<?php session\_start();

include('database.php');

// //Login user... if(isset($\_POST['login\_user'])) {

## Inserting new records:

The insert page allows users to select the table to insert values into [Fig 5.1.1]. It then asks users to input required data columns for the particular table and upon successful insertion, a new page is displayed [Fig 5.1.2] 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.

<?php

## 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 5.2.1]. 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 5.2.2].

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

## 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 userdatabase 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.

## 4.6 Trigger

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in database. The trigger is most used for maintaining the integrity of the information on the database. For example, when a new record is added to the patient table, if he needs room, room table is automatically updated. Triggers can also be used to log historical data, for example keeping track of patient and their medical history.

CREATE TRIGGER `username` BEFORE INSERT ON `users` FOR EACH ROW SET

NEW.username=UPPER(NEW.username) ;

**Chapter 5**

# SNAPSHOTS

## 5.1 Admin login:

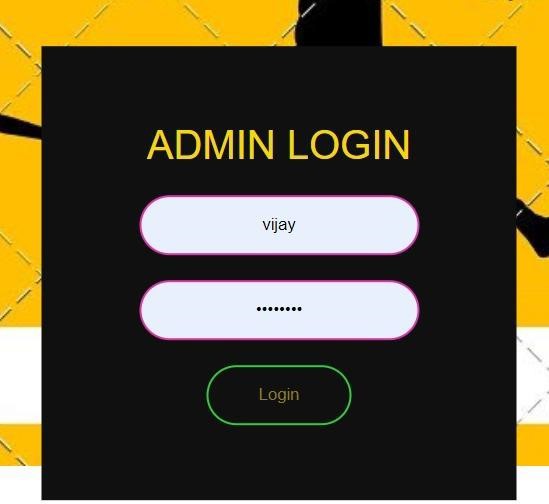


Fig 5.1 Admin Login

## Inserting new records:

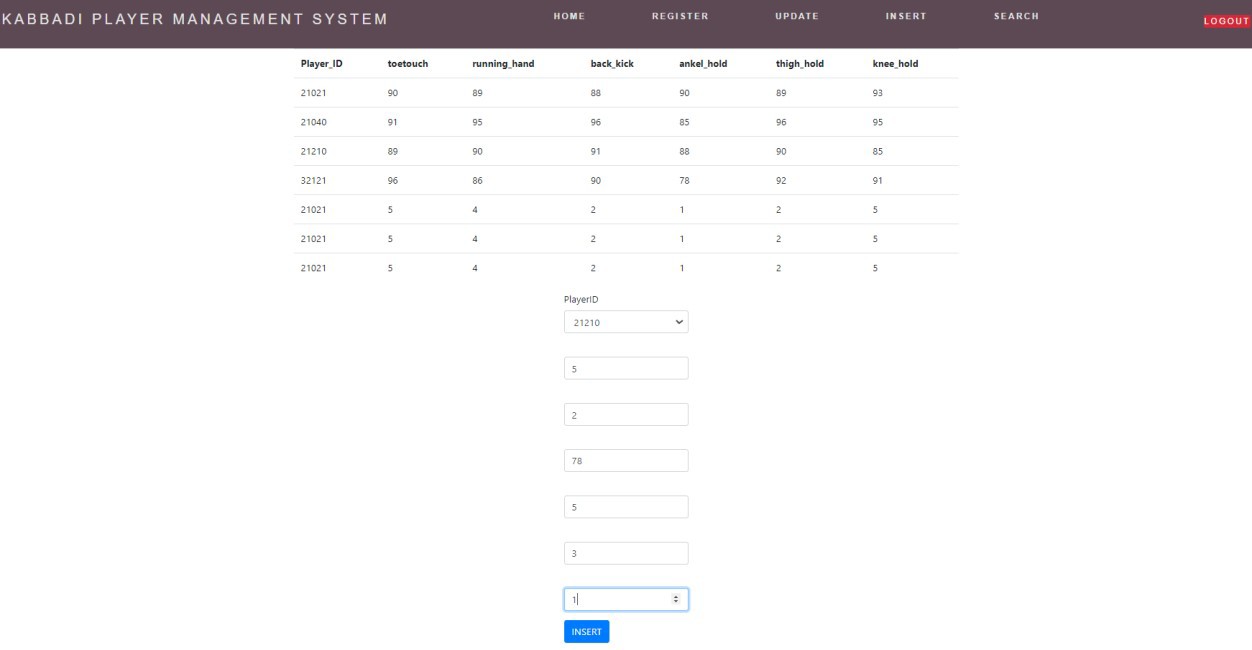


Fig 5.2 Selecting table for insertion

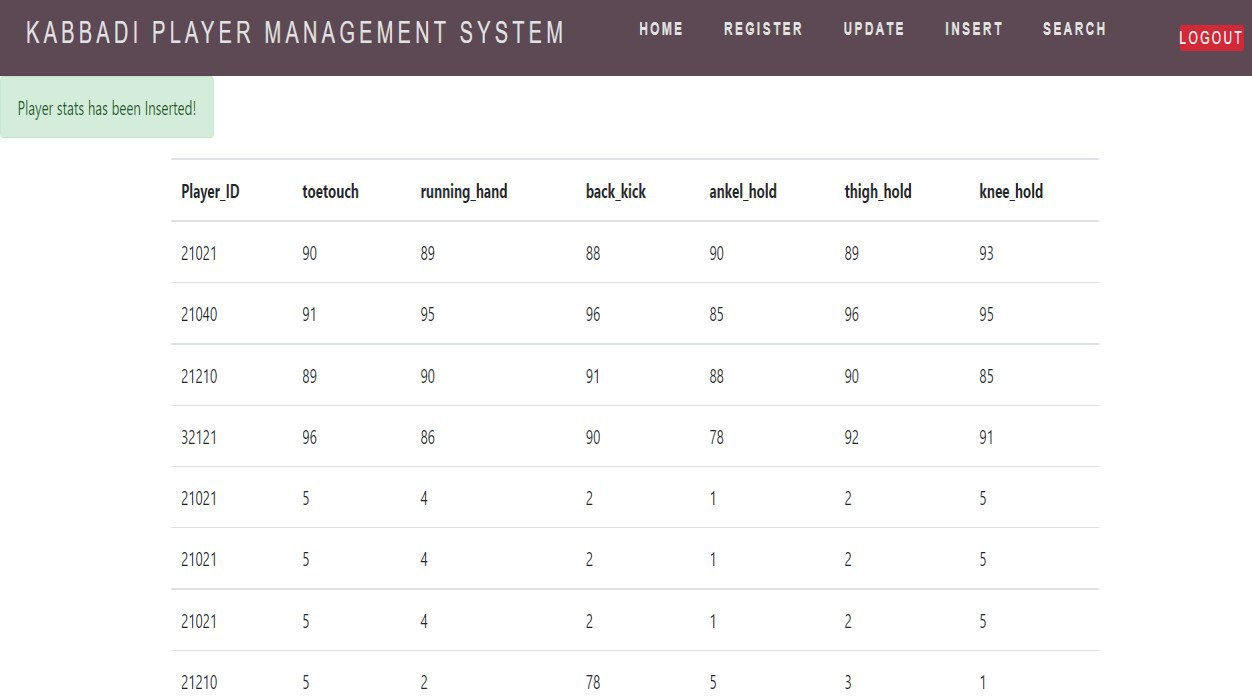


Fig 5.3 Successful insert instance page

## Update existing records:



Fig 5.4 Selecting table to modify record



Fig 5.5 Modifying records in real time

## Deleting records:



Fig 5.6 Selecting categories for deleting record

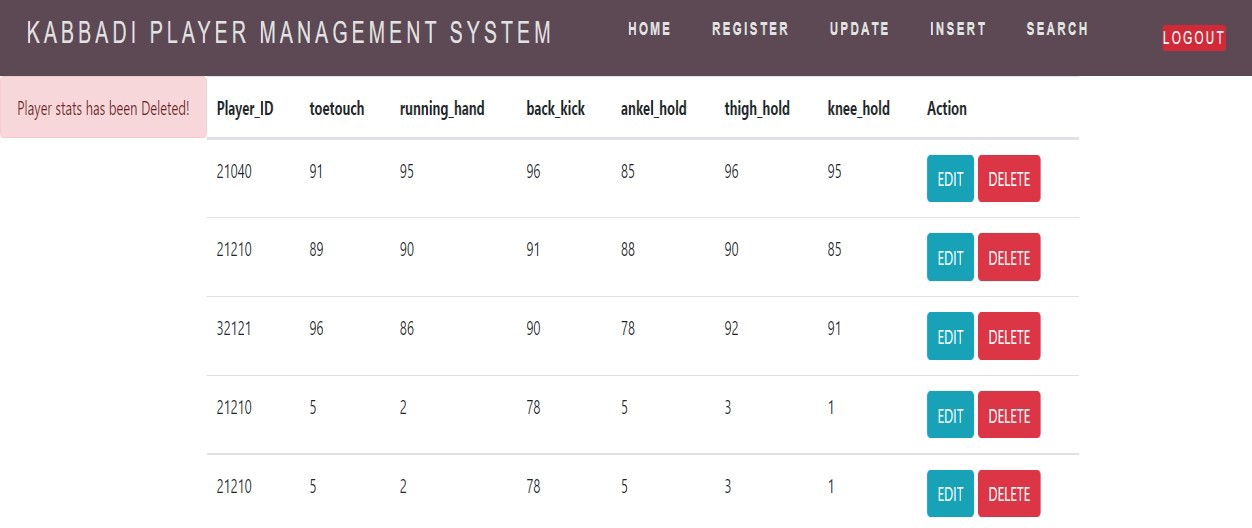


Fig 5.7 Successful deletion instance

## Database tables:

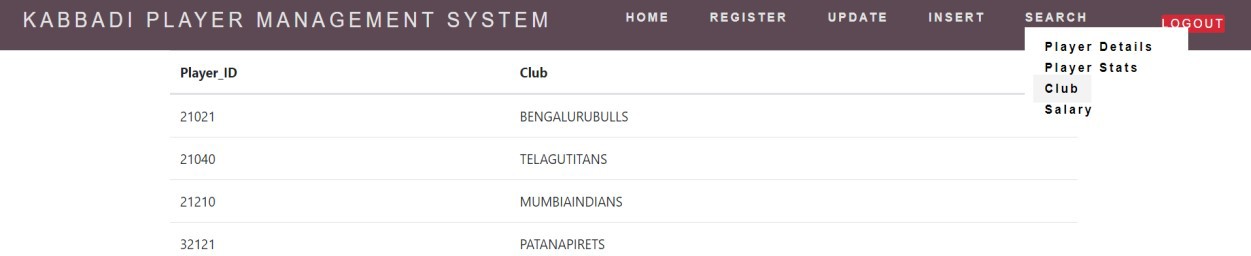


Fig 5.8 Player club table



Fig 5.9 Player salary table



Fig 5.10 Player stats table

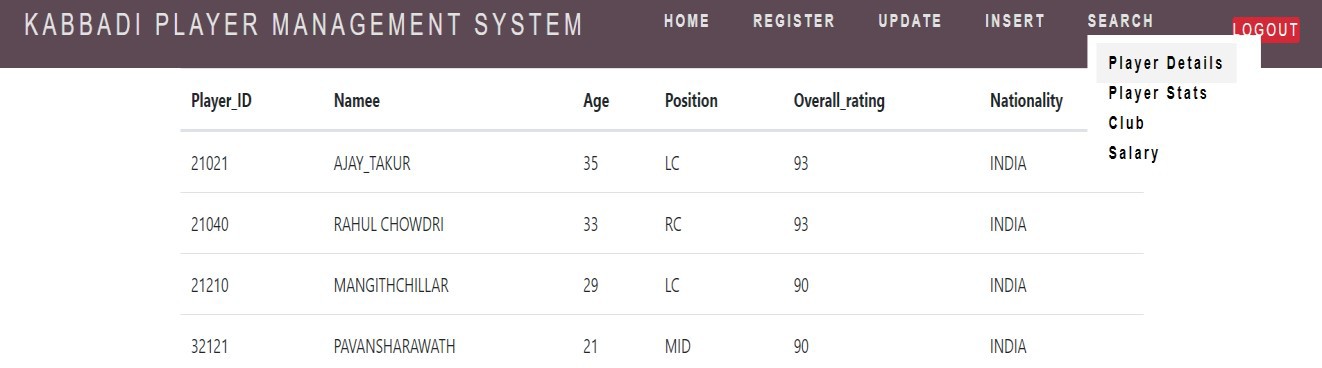


Fig 5.11 Player’s table

## Trigger:

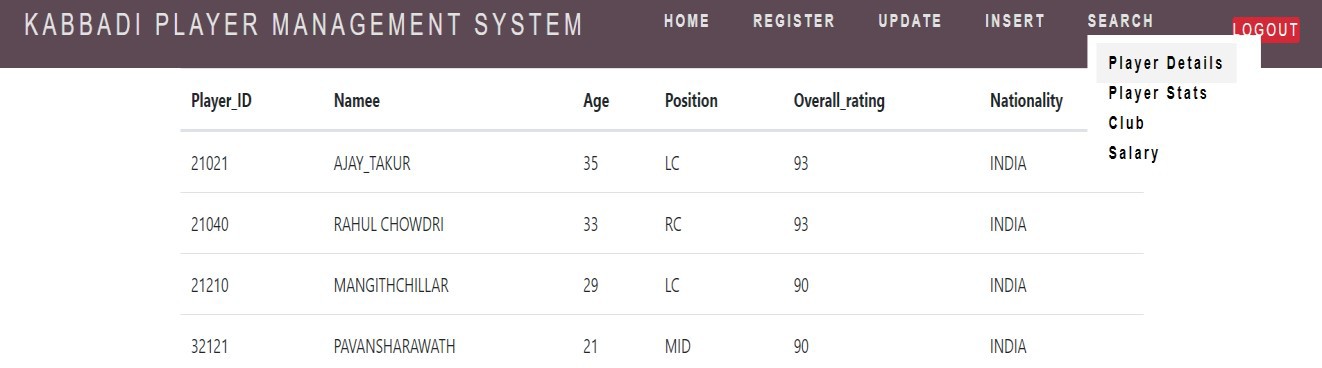


Fig 5.12 Trigger

# 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,McGrawHill.
    3. Laura Thompson & Luke Welling ,”PHP & MySQL Web development”, QUE; 1st

edition (30 March 2001)

* + 1. Stack overflow (https://stackoverflow.com)
    2. Codeproject (https://www.codeproject.com)
    3. Techrepublic (https://www.techrepublic.com)
    4. [Codepen (https://codepen.io)](https://stackoverflow.com/search?q=insert%2Btable)
    5. [Uplabs (https://www.uplabs.com)](https://stackoverflow.com/search?q=view%2Btable)
    6. Ibm [(https://www.ibm.com)](https://stackoverflow.com/search?q=search%2Bfrom%2Btable)
    7. W3schools (https://www.w3schools.com) 11.