VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

DBMS Mini Project Report
On

NBA PLAYER MANAGEMENT SYSTEM

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in

Artificial Intelligence and Machine Learning

Submitted by

USN	Name
1RN21AI054	HARSHITH S GOWDA
1RN22AI400	AJAY KUMAR T A

RNS INSTITUTE OF TECHNOLOGY
(**AICTE** Approved, **VTU** Affiliated and **NAAC 'A'** Accredited)
(**UG programs** – CSE, ECE, ISE, EIE and EEE are **Accredited** by **NBA** up to **30.6.2025)**
Channasandra, Dr. Vishnuvardhan Road, Bengaluru - 560 098

Department of AI & ML

2023 – 2024

RNS INSTITUTE OF TECHNOLOGY

(**AICTE** Approved, **VTU** Affiliated and **NAAC 'A'** Accredited)
(**UG programs** – CSE, ECE, ISE, EIE and EEE are **Accredited** by **NBA** up to **30.6.2025)**
Channasandra, Dr. Vishnuvardhan Road, Bengaluru - 560 098

Department of AI & ML

CERTIFICATE

Certified that the Mini-Project entitled **NBA PLAYER MANAGEMENT SYSTEM** carried out by Mr./Ms. **Harshith S Gowda** USN **1RN21AI054** and **Ajay Kumar T A** USN **1RN22AI400** a bonafide student of V Semester BE, **RNS Institute of Technology** in partial fulfillment for the Bachelor of Engineering in AI & ML ENGINEERING of the **Visvesvaraya Technological University**, Belagavi during the year 2023-24. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report. The Project report has been approved as it satisfies the academic requirements in respect of Database Management System with Mini Project Laboratory prescribed for the said Degree.

Course Teacher
Ms. Pooja M
Assistant Professor
Department of AI & ML
RNSIT, Bengaluru

HOD
Dr. Harsha S
Department of AI & ML
RNSIT, Bengaluru

Name & Signature

Examiner 1:

Examiner 2:

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 'NBA 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 basketball world. The software also allows users to view the whole list of players, teams and basketball 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. NBA 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

At the very onset, we would like to place on record our gratitude to all those people who have helped us in making this project work a reality. Our Institution has played a paramount role in guiding us in the right direction.

We would like to profoundly thank **Sri. Satish R Shetty**, Chairman, RNS Group of Institutions. Bangalore for providing such a healthy environment for the successful completion of this project work.

We would like to thank **Dr. M K Venkatesha**, Director, RNSIT, Bengaluru for providing the necessary facilities to carry out this work

We would like to express my gratitude to **Dr. Ramesh Babu H S** Principal, RNSIT, Bengaluru for his constant encouragement.

We are extremely grateful to **Dr. Harsha S**, Head of the Department of Artificial Intelligence and Machine Learning, for having accepted to guide me in the right direction with all his wisdom.

We would like to express our sincere thanks to our Course Teacher **Ms. Pooja M** Assistant Professor, Department of Artificial Intelligence and Machine Learning for her constant encouragement that motivated us for the successful completion of this project work.

Last but not the least. we are thankful to all the teaching and non-teaching staff members of the Artificial Intelligence and Machine Learning Department for their encouragement and support throughout this work.

SIGNATURE

Harshith S Gowda-1RN21AI054

Ajay Kumar T A-1RN22AI400

TABLE OF CONTENTS

TABLE OF FIGURES

Chapter 1

INTRODUCTION

1.1 Overview of Database Management System

To analyze and access players on a daily basics can be hectic and huge effort demanding task. To help users, managers, staffs and scouting agents in basketball world, NBA 20 player management system provides effortless player management system to help users to analyze, improve, train and access plays on a daily basis.

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 provides 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 triggers 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.

1.2 Motivation

The motivation behind the NBA Player Management System project stems from the need for efficient player organization in the face of a growing global basketball community. By centralizing player data and providing a user-friendly interface, the system aims to enhance decision-making processes for coaches and team managers.

Furthermore, it facilitates talent scouting, international collaboration, and data-driven insights, offering basketball organizations a competitive advantage in player development and strategic planning. Embracing technology ensures adaptability to modern sports management practices, fostering the continued growth and success of the sport.

1.3 Problem Definition

The NBA Player Management System project seeks to address the complex challenge of effectively managing the ever-expanding roster of basketball players across the globe. With the sport's popularity surging internationally, basketball organizations face a multitude of logistical and administrative hurdles in tracking player data, coordinating team rosters, and making informed decisions. Currently, there lacks a centralized platform that comprehensively stores and organizes player information, hindering efficient player management.

Key issues include the cumbersome process of manually updating player records, the difficulty in accessing comprehensive player profiles, and the lack of standardized systems for talent scouting and team selection. This fragmentation not only complicates administrative tasks but also hampers strategic decision-making processes for coaches and team managers.

Furthermore, the absence of a unified database limits international collaboration and inhibits the ability to harness valuable insights from player statistics and performance data. These challenges underscore the urgent need for a robust player management solution that streamlines data management, enhances user accessibility, and provides actionable insights for optimizing player performance and team dynamics.

The NBA Player Management System project aims to tackle these issues head-on by developing a desktop-based application with a user-friendly interface, a robust backend database, and comprehensive features for player tracking, team management, and data analysis. By doing so, the project seeks to revolutionize basketball management practices, empowering organizations to navigate the complexities of player management in today's dynamic and competitive sports landscape.

1.4 Key objectives

Efficiency Enhancement: Develop a centralized platform to streamline player data management, reducing manual effort and time spent on administrative tasks.

Comprehensive Player Profiles: Create detailed player profiles encompassing personal information, club history, nationality, technical skills, and performance statistics for informed decision-making.

User-Friendly Interface: Design an intuitive web-based GUI that facilitates easy navigation and accessibility for coaches, team managers, and other stakeholders.

Real-Time Updates: Implement functionalities for instant updates to player records, ensuring data accuracy and timeliness in roster management.

Talent Scouting Support: Provide tools for talent identification and scouting, enabling teams to discover and recruit promising players globally.

Data Analysis Capabilities: Incorporate features for analyzing player statistics and performance metrics to derive actionable insights for team strategy and player development.

International Collaboration: Foster collaboration between basketball organizations worldwide by enabling seamless sharing of player data and insights.

Adaptability and Scalability: Develop a flexible and scalable system capable of accommodating the evolving needs of basketball management and accommodating future expansions.

Enhanced Decision Making: Empower coaches and team managers with data-driven insights for optimizing player selection, team composition, and game strategies.

Competitive Advantage: Provide basketball organizations with a competitive edge through efficient player management, strategic decision-making, and talent optimization.

Chapter 2
SYSTEM REQUIREMENTS

2.1 Software and Hardware

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.

Software Requirements:

- Operating System: Windows 10, 64-bit
- Frontend: HTML, CSS, JavaScript
- Server-side Language: PHP
- Backend Database: MySQL
- Database Management System: phpMyAdmin
- Web Server: WampServer 64-bit
- Web Browser: Google Chrome
- Internet Connectivity
- Integrated Development Environment (IDE): Visual Studio Code

Hardware Requirements:

- Processor: AMD64 and Intel EM64T, minimum 1.8 GHz clock speed
- Screen Resolution: Minimum 1024 x 768
- RAM: Minimum 2 GB
- Disk Space: Minimum 15 GB

Chapter 3

DESIGN OF THE PROJECT

3.1 Methodology

The structured software development process commences with Requirements Analysis, where the project team collaborates closely with stakeholders to grasp their needs and expectations. They articulate the issue the software aims to address and establish the essential functions the software system must execute to resolve it. This stage yields a comprehensive requirements document.

Subsequently, the System Design phase ensues. System designers and architects utilize the requirements to craft the software system. This involves conceptualizing the software architecture, database structure, user interfaces, and system interactions. The outcome of this phase is a design blueprint that steers the subsequent steps.

The Implementation phase marks the initiation of coding by developers, following the design blueprint. They construct the software by integrating various components of the system, such as the database, server components, and user interfaces. The outcome of this phase is a software product poised for testing.

The Testing phase verifies that the software operates as anticipated and aligns with the requirements. This encompasses unit testing, integration testing, system testing, and acceptance testing. Any detected bugs and issues are addressed and resolved. The result is a thoroughly tested software product.

Once the software product successfully completes the testing phase, it undergoes Deployment to a live environment for end-user utilization. This deployment may occur gradually or through a comprehensive launch, contingent upon the chosen deployment strategy. Following deployment, the software necessitates maintenance to rectify newly identified bugs, enhance performance, or introduce new features. This continuous process persists throughout the software's lifecycle.

The system will be developed utilizing a programming language suitable for web-based applications, such as Python, Ruby, or Java. This ensures cross-platform accessibility across various operating systems, including Windows, macOS, Linux, and mobile platforms. The selection of the programming language hinges on project-specific requirements, including performance criteria, security considerations, and the expertise of the development team.

3.2 Schema Diagram

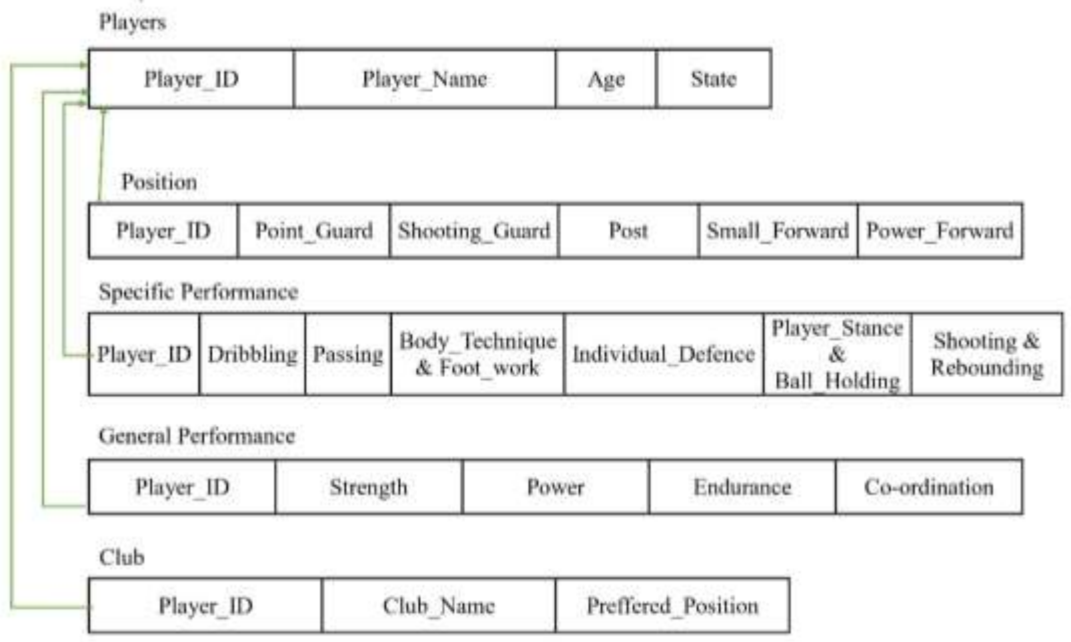


Fig. 3.1 Schema Diagram For NBA Management System

3.3 Entity Relation Diagram

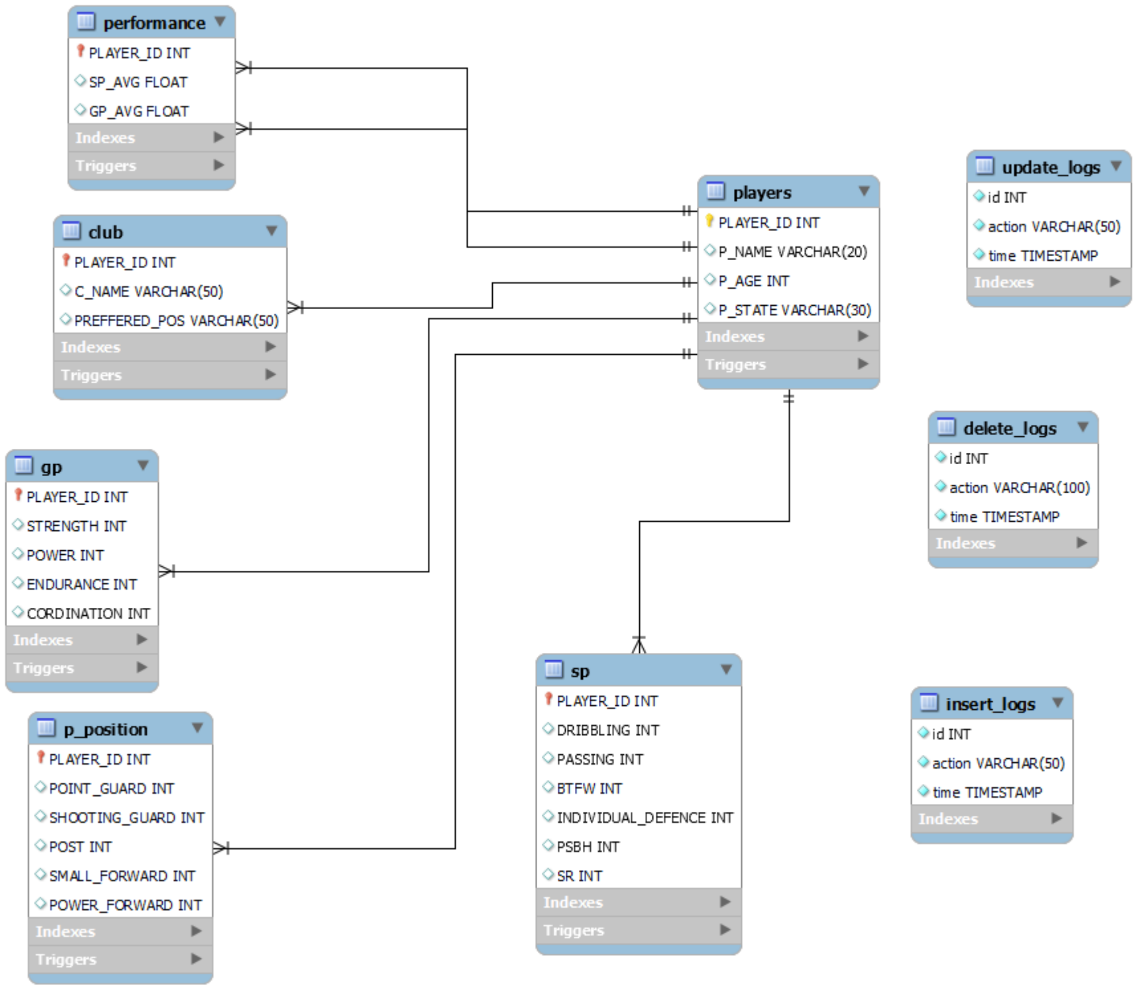
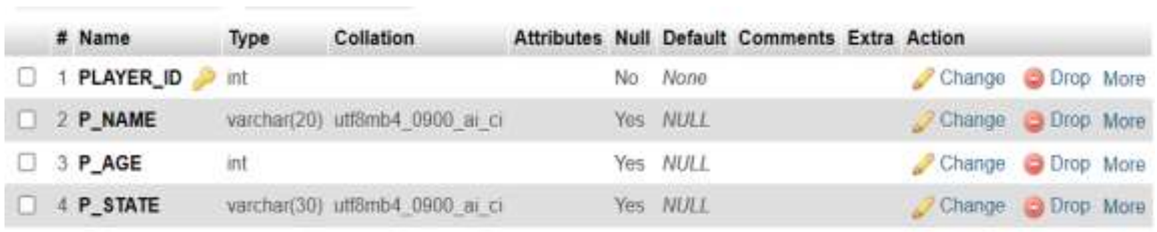


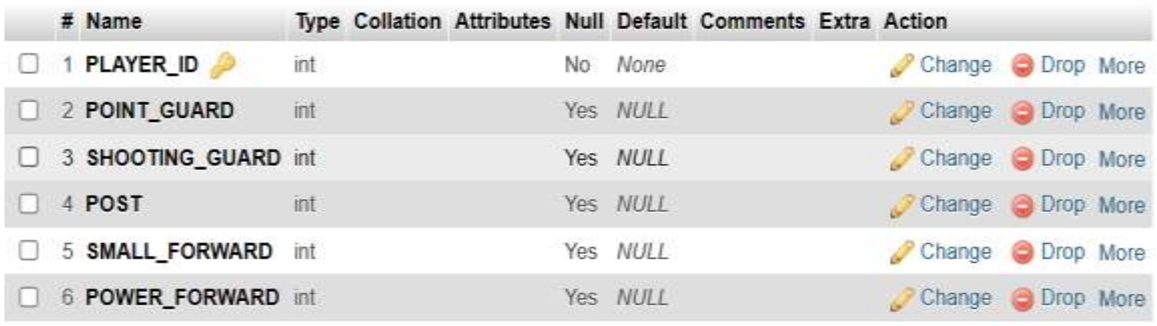
Fig. 3.2 Entity Relationship Diagram For NBA Management System

3.4 Table Structures



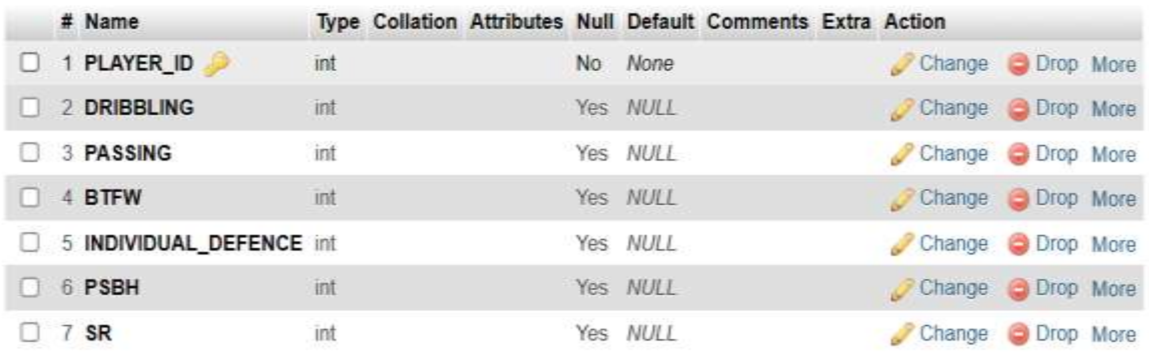
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	PLAYER_ID	int			No	None			Change Drop More
2	P_NAME	varchar(20)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
3	P_AGE	int			Yes	NULL			Change Drop More
4	P_STATE	varchar(30)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More

Fig.3.3.players Table



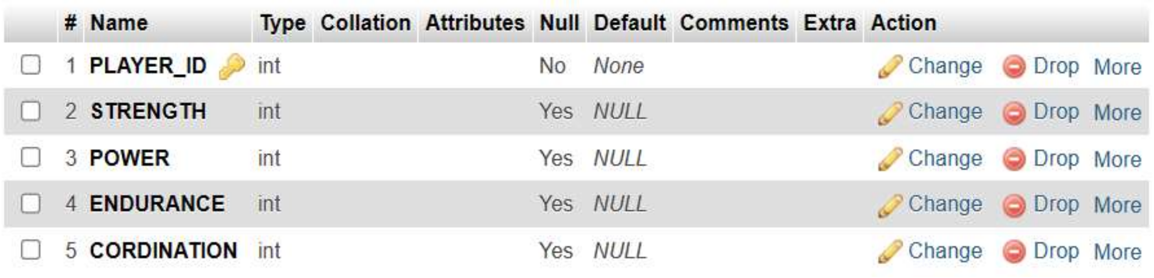
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	PLAYER_ID	int			No	None			Change Drop More
2	POINT_GUARD	int			Yes	NULL			Change Drop More
3	SHOOTING_GUARD	int			Yes	NULL			Change Drop More
4	POST	int			Yes	NULL			Change Drop More
5	SMALL_FORWARD	int			Yes	NULL			Change Drop More
6	POWER_FORWARD	int			Yes	NULL			Change Drop More

Fig. 3.4 Players Position



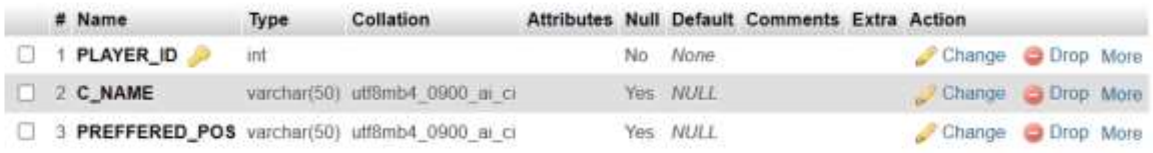
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	PLAYER_ID	int			No	None			Change Drop More
2	DRIBBLING	int			Yes	NULL			Change Drop More
3	PASSING	int			Yes	NULL			Change Drop More
4	BTFW	int			Yes	NULL			Change Drop More
5	INDIVIDUAL_DEFENCE	int			Yes	NULL			Change Drop More
6	PSBH	int			Yes	NULL			Change Drop More
7	SR	int			Yes	NULL			Change Drop More

Fig. 3.5 Specific Performance



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	PLAYER_ID	int			No	None			Change Drop More
2	STRENGTH	int			Yes	NULL			Change Drop More
3	POWER	int			Yes	NULL			Change Drop More
4	ENDURANCE	int			Yes	NULL			Change Drop More
5	CORDINATION	int			Yes	NULL			Change Drop More

Fig. 3.6 General Performance



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	PLAYER_ID	int			No	None			Change Drop More
2	C_NAME	varchar(50)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
3	PREFFERED_POS	varchar(50)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More

Fig. 3.7 Club Details

Chapter 4

CODE

4.1 SQL Code

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1:3306
-- Generation Time: Feb 27, 2024 at 10:49 AM
-- Server version: 8.0.36
-- PHP Version: 8.2.13

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `nba_management`
--

DELIMITER $$
--
-- Procedures
--
DROP PROCEDURE IF EXISTS `SearchAge`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `SearchAge` (IN `page`
INT(11))   SELECT P_NAME,P_AGE,P_STATE FROM PLAYERS P WHERE P.P_AGE =
page$$

DROP PROCEDURE IF EXISTS `SearchAge1`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `SearchAge1` (IN `page`
INT(11))   SELECT P_NAME,P_AGE,P_STATE FROM PLAYERS P WHERE P.P_AGE =
page$$

DROP PROCEDURE IF EXISTS `SearchName`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `SearchName` (IN `page`
VARCHAR(30))   SELECT * FROM personal_details WHERE player_name = page$$
```

```
DROP PROCEDURE IF EXISTS `SearchNationality`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `SearchNationality` (IN `page`
VARCHAR(30))   SELECT * FROM personal_details WHERE
personal_details.nationality=page$$

DROP PROCEDURE IF EXISTS `SearchOverallRating`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `SearchOverallRating` (IN
`page` INT(11))   SELECT * FROM personal_details WHERE
personal_details.overall_rating = page$$

DROP PROCEDURE IF EXISTS `Searchplayerid`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `Searchplayerid` (IN `page`
INT(11))   SELECT * FROM personal_details WHERE player_id = page$$

DROP PROCEDURE IF EXISTS `SearchPosition`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `SearchPosition` (IN `page`
VARCHAR(11))   SELECT pd.player_name, pd.overall_rating,
od.preferred_position, p.gk, p.df, p.cm, p.fr FROM personal_details pd,
other_details od, position p WHERE od.preferred_position = page AND
p.player_id = od.player_id AND p.player_id = pd.player_id GROUP BY
pd.player_id$$

DROP PROCEDURE IF EXISTS `SearchTeam`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `SearchTeam` (IN `page`
VARCHAR(30))   SELECT
pd.player_name,pd.overall_rating,pd.age,pd.nationality,od.club FROM
personal_details pd,other_details od WHERE od.club = page AND pd.player_id
= od.player_id ORDER BY pd.player_id$$

DELIMITER ;

-- --------------------------------------------------------

--
-- Table structure for table `club`
--

DROP TABLE IF EXISTS `club`;
CREATE TABLE IF NOT EXISTS `club` (
  `PLAYER_ID` int NOT NULL,
  `C_NAME` varchar(50) DEFAULT NULL,
  `PREFFERED_POS` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`PLAYER_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```sql
--
-- Dumping data for table `club`
--

INSERT INTO `club` (`PLAYER_ID`, `C_NAME`, `PREFFERED_POS`) VALUES
(1, 'mountain', 'Post guard'),
(2, 'sea', 'POWER FORWARD');

--
-- Triggers `club`
--
DROP TRIGGER IF EXISTS `delete_trigger_club`;
DELIMITER $$
CREATE TRIGGER `delete_trigger_club` AFTER DELETE ON `club` FOR EACH ROW
INSERT INTO delete_logs(action,time) VALUES("deleted successfull from the
table club",now())
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `insert_trigger_club`;
DELIMITER $$
CREATE TRIGGER `insert_trigger_club` AFTER INSERT ON `club` FOR EACH ROW
INSERT INTO  insert_logs VALUES(null,"inserted successfull in club",now())
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `update_trigger_club`;
DELIMITER $$
CREATE TRIGGER `update_trigger_club` AFTER UPDATE ON `club` FOR EACH ROW
INSERT INTO update_logs(action,time) VALUES("insertion is successfull in
club",now())
$$
DELIMITER ;

-- --------------------------------------------------------

--
-- Table structure for table `delete_logs`
--

DROP TABLE IF EXISTS `delete_logs`;
CREATE TABLE IF NOT EXISTS `delete_logs` (
  `id` int NOT NULL AUTO_INCREMENT,
  `action` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci
NOT NULL,
  `time` timestamp NOT NULL,
  UNIQUE KEY `id` (`id`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=22 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

--
-- Dumping data for table `delete_logs`
--

INSERT INTO `delete_logs` (`id`, `action`, `time`) VALUES
(14, 'Deleted sucessfully in Players Table', '2024-02-24 03:54:20'),
(15, 'Deleted sucessfully in Players Table', '2024-02-24 03:54:20'),
(16, 'Deleted sucessfully in Players Table', '2024-02-24 03:54:20'),
(17, 'Deleted sucessfully in Players Table', '2024-02-25 12:18:38'),
(18, 'deleted the position values successfull', '2024-02-25 13:27:41'),
(19, 'deleted successfull from the table specific performance', '2024-02-
25 14:15:35'),
(20, 'deleted successfull from the table general performance', '2024-02-25
14:35:55'),
(21, 'deleted successfull from the table club', '2024-02-25 14:36:27');

-- --------------------------------------------------------

--
-- Table structure for table `gp`
--

DROP TABLE IF EXISTS `gp`;
CREATE TABLE IF NOT EXISTS `gp` (
  `PLAYER_ID` int NOT NULL,
  `STRENGTH` int DEFAULT NULL,
  `POWER` int DEFAULT NULL,
  `ENDURANCE` int DEFAULT NULL,
  `CORDINATION` int DEFAULT NULL,
  PRIMARY KEY (`PLAYER_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Dumping data for table `gp`
--

INSERT INTO `gp` (`PLAYER_ID`, `STRENGTH`, `POWER`, `ENDURANCE`,
`CORDINATION`) VALUES
(1, 8, 8, 8, 8),
(2, 8, 8, 8, 7);
```

```
--
-- Triggers `gp`
--
DROP TRIGGER IF EXISTS `GP_AVG_INSERT`;
DELIMITER $$
CREATE TRIGGER `GP_AVG_INSERT` AFTER INSERT ON `gp` FOR EACH ROW BEGIN
    DECLARE gp_avg FLOAT;
    SET gp_avg = (NEW.STRENGTH + NEW.POWER + NEW.ENDURANCE +
NEW.CORDINATION) / 4;
    INSERT INTO PERFORMANCE (PLAYER_ID, GP_AVG) VALUES (NEW.PLAYER_ID,
gp_avg)
    ON DUPLICATE KEY UPDATE GP_AVG = gp_avg;
END
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `delete_trigger_gp`;
DELIMITER $$
CREATE TRIGGER `delete_trigger_gp` AFTER DELETE ON `gp` FOR EACH ROW
INSERT INTO delete_logs(action,time) VALUES("deleted successfull from the
table general performance",now())
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `insert_trigger_gp`;
DELIMITER $$
CREATE TRIGGER `insert_trigger_gp` AFTER INSERT ON `gp` FOR EACH ROW
INSERT INTO  insert_logs VALUES(null,"inserted successfull in general
performance",now())
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `update_trigger_gp`;
DELIMITER $$
CREATE TRIGGER `update_trigger_gp` AFTER UPDATE ON `gp` FOR EACH ROW
INSERT INTO update_logs(action,time) VALUES("insertion is successfull in
general performance ",now())
$$
DELIMITER ;

-- --------------------------------------------------------

--
-- Table structure for table `insert_logs`
--

DROP TABLE IF EXISTS `insert_logs`;
CREATE TABLE IF NOT EXISTS `insert_logs` (
  `id` int NOT NULL AUTO_INCREMENT,
  `action` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci
NOT NULL,
  `time` timestamp NOT NULL,
  UNIQUE KEY `id` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=24 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

--
-- Dumping data for table `insert_logs`
--

INSERT INTO `insert_logs` (`id`, `action`, `time`) VALUES
(6, 'inserted sucessfully in players table', '2024-02-23 15:36:18'),
(7, 'inserted sucessfully in players table', '2024-02-23 15:36:53'),
(8, 'inserted sucessfully in players table', '2024-02-23 15:37:12'),
(9, 'inserted into position successfull', '2024-02-23 15:37:31'),
(10, 'inserted into position successfull', '2024-02-23 15:37:44'),
(11, 'inserted into position successfull', '2024-02-23 15:38:03'),
(12, 'inserted successfull in specific performance', '2024-02-23
15:38:52'),
(13, 'inserted successfull in specific performance', '2024-02-23
15:39:11'),
(14, 'inserted successfull in specific performance', '2024-02-23
15:39:27'),
(15, 'inserted successfull in general performance', '2024-02-23
15:40:07'),
(16, 'inserted successfull in general performance', '2024-02-23
15:40:18'),
(17, 'inserted successfull in general performance', '2024-02-23
15:41:19'),
(18, 'inserted successfull in club', '2024-02-23 15:41:54'),
(19, 'inserted successfull in club', '2024-02-23 15:42:28'),
(20, 'inserted successfull in club', '2024-02-23 15:43:09'),
(21, 'inserted sucessfully in players table', '2024-02-25 12:12:44'),
(22, 'inserted sucessfully in players table', '2024-02-25 12:13:06'),
(23, 'inserted sucessfully in players table', '2024-02-25 12:26:17');

-- --------------------------------------------------------

--
-- Table structure for table `performance`
--

DROP TABLE IF EXISTS `performance`;
```

```sql
CREATE TABLE IF NOT EXISTS `performance` (
  `PLAYER_ID` int NOT NULL,
  `SP_AVG` float DEFAULT NULL,
  `GP_AVG` float DEFAULT NULL,
  PRIMARY KEY (`PLAYER_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Triggers `performance`
--
DROP TRIGGER IF EXISTS `delete_trigger_per`;
DELIMITER $$
CREATE TRIGGER `delete_trigger_per` AFTER DELETE ON `performance` FOR EACH
ROW INSERT INTO delete_logs(action,time) VALUES("Deleted sucessfully in
performance table",now())
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `insert_trigger_per`;
DELIMITER $$
CREATE TRIGGER `insert_trigger_per` AFTER INSERT ON `performance` FOR EACH
ROW INSERT INTO insert_logs VALUES(null,"inserted sucessfully in
performance table",NOW())
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `update_trigger_per`;
DELIMITER $$
CREATE TRIGGER `update_trigger_per` AFTER UPDATE ON `performance` FOR EACH
ROW INSERT INTO update_logs (action,time) VALUES ("AUTO UPDATED
SUCESSFULLY IN PERFORMANCE TABLE",NOW())
$$
DELIMITER ;

-- --------------------------------------------------------

--
-- Table structure for table `players`
--

DROP TABLE IF EXISTS `players`;
CREATE TABLE IF NOT EXISTS `players` (
  `PLAYER_ID` int NOT NULL,
  `P_NAME` varchar(20) DEFAULT NULL,
  `P_AGE` int DEFAULT NULL,
  `P_STATE` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`PLAYER_ID`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Dumping data for table `players`
--

INSERT INTO `players` (`PLAYER_ID`, `P_NAME`, `P_AGE`, `P_STATE`) VALUES
(1, 'AJAY', 25, 'KARNATAKA'),
(2, 'SRI', 22, 'TAMILNADU');

--
-- Triggers `players`
--
DROP TRIGGER IF EXISTS `delete_trigger_players`;
DELIMITER $$
CREATE TRIGGER `delete_trigger_players` AFTER DELETE ON `players` FOR EACH
ROW INSERT INTO delete_logs(action,time) VALUES("Deleted sucessfully in
Players Table",now())
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `insert_trigger_players`;
DELIMITER $$
CREATE TRIGGER `insert_trigger_players` AFTER INSERT ON `players` FOR EACH
ROW INSERT INTO insert_logs VALUES(null,"inserted sucessfully in players
table",NOW())
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `update_trigger_players`;
DELIMITER $$
CREATE TRIGGER `update_trigger_players` AFTER UPDATE ON `players` FOR EACH
ROW INSERT INTO update_logs (action,time) VALUES ("UPDATED SUCESSFULLY IN
PLAYERS",NOW())
$$
DELIMITER ;

-- --------------------------------------------------------

--
-- Table structure for table `p_position`
--

DROP TABLE IF EXISTS `p_position`;
CREATE TABLE IF NOT EXISTS `p_position` (
  `PLAYER_ID` int NOT NULL,
  `POINT_GUARD` int DEFAULT NULL,
  `SHOOTING_GUARD` int DEFAULT NULL,
  `POST` int DEFAULT NULL,
  `SMALL_FORWARD` int DEFAULT NULL,
  `POWER_FORWARD` int DEFAULT NULL,
  PRIMARY KEY (`PLAYER_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Dumping data for table `p_position`
--

INSERT INTO `p_position` (`PLAYER_ID`, `POINT_GUARD`, `SHOOTING_GUARD`,
`POST`, `SMALL_FORWARD`, `POWER_FORWARD`) VALUES
(1, 9, 9, 9, 8, 9),
(3, 7, 7, 7, 7, 5);

--
-- Triggers `p_position`
--
DROP TRIGGER IF EXISTS `delete_trigger_position`;
DELIMITER $$
CREATE TRIGGER `delete_trigger_position` AFTER DELETE ON `p_position` FOR
EACH ROW INSERT INTO delete_logs(action,time) VALUES ("deleted the
position values successfull",now())
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `insert_trigger_position`;
DELIMITER $$
CREATE TRIGGER `insert_trigger_position` AFTER INSERT ON `p_position` FOR
EACH ROW INSERT INTO insert_logs VALUES(null,"inserted into position
successfull",now())
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `update_trigger_position`;
DELIMITER $$
CREATE TRIGGER `update_trigger_position` AFTER UPDATE ON `p_position` FOR
EACH ROW INSERT INTO update_logs(action,time) VALUES("update the position
values successfull",now())
$$
DELIMITER ;

-- --------------------------------------------------------

--
-- Table structure for table `sp`
```

```sql
--

DROP TABLE IF EXISTS `sp`;
CREATE TABLE IF NOT EXISTS `sp` (
  `PLAYER_ID` int NOT NULL,
  `DRIBBLING` int DEFAULT NULL,
  `PASSING` int DEFAULT NULL,
  `BTFW` int DEFAULT NULL,
  `INDIVIDUAL_DEFENCE` int DEFAULT NULL,
  `PSBH` int DEFAULT NULL,
  `SR` int DEFAULT NULL,
  PRIMARY KEY (`PLAYER_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

--
-- Dumping data for table `sp`
--

INSERT INTO `sp` (`PLAYER_ID`, `DRIBBLING`, `PASSING`, `BTFW`,
`INDIVIDUAL_DEFENCE`, `PSBH`, `SR`) VALUES
(1, 8, 8, 8, 8, 8, 8),
(2, 8, 8, 8, 8, 7, 8);

--
-- Triggers `sp`
--
DROP TRIGGER IF EXISTS `SP_AVG_INSERT`;
DELIMITER $$
CREATE TRIGGER `SP_AVG_INSERT` AFTER INSERT ON `sp` FOR EACH ROW BEGIN
    DECLARE sp_avg FLOAT;
    SET sp_avg = (NEW.DRIBBLING + NEW.PASSING + NEW.BTFW +
NEW.INDIVIDUAL_DEFENCE + NEW.PSBH + NEW.SR) / 6;
    INSERT INTO PERFORMANCE (PLAYER_ID, SP_AVG) VALUES (NEW.PLAYER_ID,
sp_avg)
    ON DUPLICATE KEY UPDATE SP_AVG = sp_avg;
END
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `delete_trigger_sp`;
DELIMITER $$
CREATE TRIGGER `delete_trigger_sp` AFTER DELETE ON `sp` FOR EACH ROW
INSERT INTO delete_logs(action,time) VALUES("deleted successfull from the
table specific performance",now())
$$
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS `insert_trigger_sp`;
DELIMITER $$
CREATE TRIGGER `insert_trigger_sp` AFTER INSERT ON `sp` FOR EACH ROW
INSERT INTO  insert_logs VALUES(null,"inserted successfull in specific
performance",now())
$$
DELIMITER ;
DROP TRIGGER IF EXISTS `update_trigger_sp`;
DELIMITER $$
CREATE TRIGGER `update_trigger_sp` AFTER UPDATE ON `sp` FOR EACH ROW
INSERT INTO update_logs(action,time) VALUES("insertion is successfull in
specific performance ",now())
$$
DELIMITER ;

-- --------------------------------------------------------

--
-- Table structure for table `update_logs`
--

DROP TABLE IF EXISTS `update_logs`;
CREATE TABLE IF NOT EXISTS `update_logs` (
  `id` int NOT NULL AUTO_INCREMENT,
  `action` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci
NOT NULL,
  `time` timestamp NOT NULL,
  UNIQUE KEY `id` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

--
-- Dumping data for table `update_logs`
--

INSERT INTO `update_logs` (`id`, `action`, `time`) VALUES
(1, 'UPDATED SUCESSFULLY IN PLAYERS', '2024-02-25 12:13:33'),
(2, 'UPDATED SUCESSFULLY IN PLAYERS', '2024-02-25 12:26:45'),
(3, 'update the position values successfull', '2024-02-25 13:27:28'),
(4, 'update the position values successfull', '2024-02-25 13:27:38'),
(5, 'update the position values successfull', '2024-02-25 14:12:58'),
(6, 'insertion is successfull in specific performance ', '2024-02-25
14:29:49'),
(7, 'insertion is successfull in general performance ', '2024-02-25
14:35:52'),
(8, 'insertion is successfull in club', '2024-02-25 14:36:24');
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

4.2 Front-End Code

```
<!DOCTYPE html>
<html lang="en" >

<head>
  <meta charset="UTF-8">
  <title>Update Player</title>


  <link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet' type='text/css'>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min
.css">

      <link rel="stylesheet" href="css/style.css">
      <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/meyer-
reset/2.0/reset.min.css">
      <link rel='stylesheet' href='css/34b729901a37198f5d0414728.css'>
     <link
href='https://fonts.googleapis.com/css?family=Love+Ya+Like+A+Sister'
rel='stylesheet' type='text/css'>
     <link href='https://fonts.googleapis.com/css?family=Codystar'
rel='stylesheet' type='text/css'>
         <link href="css/menu.css" rel="stylesheet" type="text/css">
           <script
src="https://ajax.googleapis.com/ajax/libs/webfont/1.4.7/webfont.js"
type="text/javascript"></script>
         <link href="https://fonts.googleapis.com/css?family=Poppins"
rel="stylesheet" />

</head>

<body style="background: linear-gradient(to right top, #000623, #0e284f,
#194881, #1b6cb7, #0092f0);">
```

```
    <canvas class="fireworks">

    </canvas>

  <svg id="game" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1000 400"
overflow="visible">
    <linearGradient id="ArcGradient" >
            <stop offset="0"  stop-color="#fff" stop-opacity=".2"/>
            <stop offset="50%" stop-color="#fff" stop-opacity="0"/>
    </linearGradient>
    <path id="arc" fill="none" stroke="url(#ArcGradient)" stroke-width="4"
d="M100,250c250-400,550-400,800,0"  pointer-events="none"/>
    <defs>
        <g id="arrow">
            <line x2="60" fill="none" stroke="#888" stroke-width="2" />
            <polygon fill="#888" points="64 0 58 2 56 0 58 -2" />
            <polygon fill="#88ce02" points="2 -3 -4 -3 -1 0 -4 3 2 3 5 0"
/>
        </g>
    </defs>
    <g id="target">
        <path fill="#FFF" d="M924.2,274.2c-21.5,21.5-45.9,19.9-52,3.2c-
4.4-12.1,2.4-29.2,14.2-41c11.8-11.8,29-18.6,41-14.2
C944.1,228.3,945.7,252.8,924.2,274.2z" />
        <path fill="#F4531C" d="M915.8,265.8c-14.1,14.1-30.8,14.6-36,4.1c-
4.1-8.3,0.5-21.3,9.7-30.5s22.2-13.8,30.5-9.7
C930.4,235,929.9,251.7,915.8,265.8z" />
        <path fill="#FFF" d="M908.9,258.9c-8,8-17.9,9.2-21.6,3.5c-3.2-4.9-
0.5-13.4,5.6-19.5c6.1-6.1,14.6-8.8,19.5-5.6
C918.1,241,916.9,250.9,908.9,258.9z" />
        <path fill="#F4531C" d="M903.2,253.2c-2.9,2.9-6.7,3.6-8.3,1.7c-
1.5-1.8-0.6-5.4,2-8c2.6-2.6,6.2-3.6,8-2
C906.8,246.5,906.1,250.2,903.2,253.2z" />
    </g>
        <g id="bow" fill="none" stroke-linecap="round" vector-effect="non-
scaling-stroke" pointer-events="none">
            <polyline fill="none" stroke="#ddd" stroke-linecap="round"
points="88,200 88,250 88,300"/>
            <path fill="none" stroke="#88ce02" stroke-width="3" stroke-
linecap="round" d="M88,300 c0-10.1,12-25.1,12-50s-12-39.9-12-50"/>
    </g>
    <g class="arrow-angle"><use x="100" y="250" xlink:href="#arrow"/></g>
    <clipPath id="mask">
        <polygon opacity=".5" points="0,0 1500,0 1500,200 970,290 950,240
925,220 875,280 890,295 920,310 0,350" pointer-events="none"/>
```

```
    </clipPath>
    <g class="arrows" clip-path="url(#mask)"  pointer-events="none">
    </g>
    <g class="miss" fill="#aaa" opacity="0" transform="translate(0, 100)">
        <path d="M358 194L363 118 386 120 400 153 416 121 440 119 446 203
419 212 416 163 401 180 380 160 381 204"/>
        <path d="M450 120L458 200 475 192 474 121"/>
        <path d="M537 118L487 118 485 160 515 162 509 177 482 171 482 193
529 199 538 148 501 146 508 133 537 137"/>
        <path d="M540 202L543 178 570 186 569 168 544 167 546 122 590 116
586 142 561 140 560 152 586 153 586 205"/>
        <path d="M595,215l5-23l31,0l-5,29L595,215z M627,176l13-70l-41-0l-
0,70L627,176z"/>
    </g>
    <g class="bullseye" fill="#F4531C" opacity="0">
        <path d="M322,159l15-21l-27-13l-32,13l15,71l41-14l7-32L322,159z
M292,142h20l3,8l-16,8 L292,142z M321,182l-18,9l-4-18l23-2V182z"/>
        <path d="M340 131L359 125 362 169 381 167 386 123 405 129 392 183
351 186z"/>
        <path d="M413 119L402 188 450 196 454 175 422 175 438 120z"/>
        <path d="M432 167L454 169 466 154 451 151 478 115 453 113z"/>
        <path d="M524 109L492 112 466 148 487 155 491 172 464 167 463 184
502 191 513 143 487 141 496 125 517 126z"/>
        <path d="M537 114L512 189 558 199 566 174 533 175 539 162 553 164
558 150 543 145 547 134 566 148 575 124z"/>
        <path d="M577 118L587 158 570 198 587 204 626 118 606 118 598 141
590 112z"/>
        <path d="M635 122L599 198 643 207 649 188 624 188 630 170 639 178
645 162 637 158 649 143 662 151 670 134z"/>
        <path d="M649,220l4-21l28,4l-6,25L649,220z M681,191l40-79l-35-
8L659,184L681,191z"/>
    </g>
    <g class="hit" fill="#ffcc00" opacity="0" transform="translate(180, -
80) rotate(12) ">
        <path d="M383 114L385 195 407 191 406 160 422 155 418 191 436 189
444 112 423 119 422 141 407 146 400 113"/>
        <path d="M449 185L453 113 477 112 464 186"/>
        <path d="M486 113L484 130 506 130 481 188 506 187 520 131 540 135
545 119"/>
        <path d="M526,195l5-20l22,5l-9,16L526,195z M558,164l32-44l-35-9l-
19,51L558,164z"/>
    </g>
<!--    <line x1= "875", y1= "280", x2= "925", y2= "220" stroke="red"/>

<circle class="point" r="7" fill="purple" opacity=".4"/> -->
```

```
</svg>
    <svg xmlns="http://www.w3.org/2000/svg" version="1.1" class="goo">
      <defs>
        <filter id="goo">
          <feGaussianBlur in="SourceGraphic" stdDeviation="10"
result="blur" />
          <feColorMatrix in="blur" mode="matrix" values="1 0 0 0 0  0 1 0
0 0  0 0 1 0 0  0 0 0 19 -9" result="goo" />
          <feComposite in="SourceGraphic" in2="goo"/>
        </filter>
      </defs>
    </svg>
        <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>

  </ul>
<div>
    <form style="align-items: center; margin-top: 20%;">
    <span class="button--bubble__container">
      <a href="php/modify.html" class="button button--bubble">
        MODIFY DATA
      </a>
      <span class="button--bubble__effect-container">
        <span class="circle top-left"></span>
        <span class="circle top-left"></span>
        <span class="circle top-left"></span>

        <span class="button effect-button"></span>

        <span class="circle bottom-right"></span>
        <span class="circle bottom-right"></span>
        <span class="circle bottom-right"></span>
      </span>
    </span>
    <span class="button--bubble__container">
        <a href="php/delete.html" class="button button--bubble">
          DELETE DATA
        </a>
        <span class="button--bubble__effect-container">
```

```
          <span class="circle top-left"></span>
          <span class="circle top-left"></span>
          <span class="circle top-left"></span>

          <span class="button effect-button"></span>

          <span class="circle bottom-right"></span>
          <span class="circle bottom-right"></span>
          <span class="circle bottom-right"></span>
        </span>
      </span>
    </form>
    </div>
    <defs>
        <radialGradient cx="50%" cy="0%" fx="50%" fy="0%" r="50%"
id="radialGradient-1">
          <stop stop-color="#329FFF" offset="0%"></stop>
          <stop stop-color="#206EFF" offset="100%"></stop>
        </radialGradient>
        <radialGradient cx="50%" cy="0%" fx="50%" fy="0%" r="50%"
id="radialGradient-2">
          <stop stop-color="#FF7894" offset="0%"></stop>
          <stop stop-color="#FF324A" offset="100%"></stop>
        </radialGradient>
        <radialGradient cx="50%" cy="0%" fx="50%" fy="0%" r="100%"
id="radialGradient-3">
          <stop stop-color="#FF7894" offset="0%"></stop>
          <stop stop-color="#FF324A" offset="100%"></stop>
        </radialGradient>
        <radialGradient cx="50%" cy="0%" fx="50%" fy="0%" r="100%"
id="radialGradient-4">
          <stop stop-color="#359FFC" offset="0%"></stop>
          <stop stop-color="#206EFF" offset="100%"></stop>
        </radialGradient>
        <radialGradient cx="50%" cy="0%" fx="50%" fy="0%" r="50%"
id="radialGradient-5">
          <stop stop-color="#5FFFD2" offset="0%"></stop>
          <stop stop-color="#31FFA6" offset="100%"></stop>
        </radialGradient>
      </defs>
      <g stroke="none" stroke-width="1" fill="none" fill-rule="evenodd">
        <rect id="dot-js" x="80" y="352" width="32" height="32"
rx="16"></rect>
      </g>
```

```
<script src='js/4032af61ca0478304ab7b31b7.js'></script>

  <script
src='https://cdnjs.cloudflare.com/ajax/libs/gsap/1.19.1/TweenMax.min.js'><
/script>
<script src='http://s3-us-west-
2.amazonaws.com/s.cdpn.io/16327/MorphSVGPlugin.min.js'></script>
<script
src='https://cdnjs.cloudflare.com/ajax/libs/gsap/1.17.0/plugins/CSSPlugin.
min.js'></script>
<script
src='https://cdnjs.cloudflare.com/ajax/libs/gsap/1.17.0/easing/EasePack.mi
n.js'></script>
<script
src='https://cdnjs.cloudflare.com/ajax/libs/gsap/1.17.0/TweenLite.min.js'>
</script>
<script
src='http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js'></s
cript>
<script
src='https://cdnjs.cloudflare.com/ajax/libs/gsap/latest/TimelineLite.min.j
s'></script>



    <script  src="js/index.js"></script>




</body>

</html>
```

CHAPTER 5

WEB PAGES OUTPUT

5.1 User Interfaces

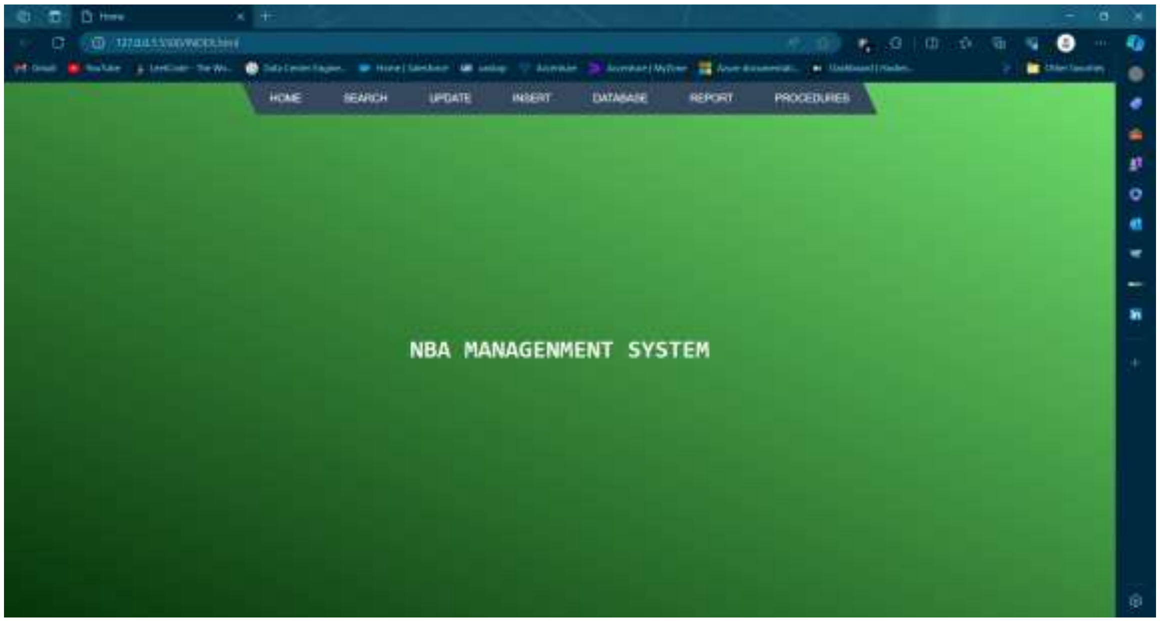


Fig. 5.1 Home Page

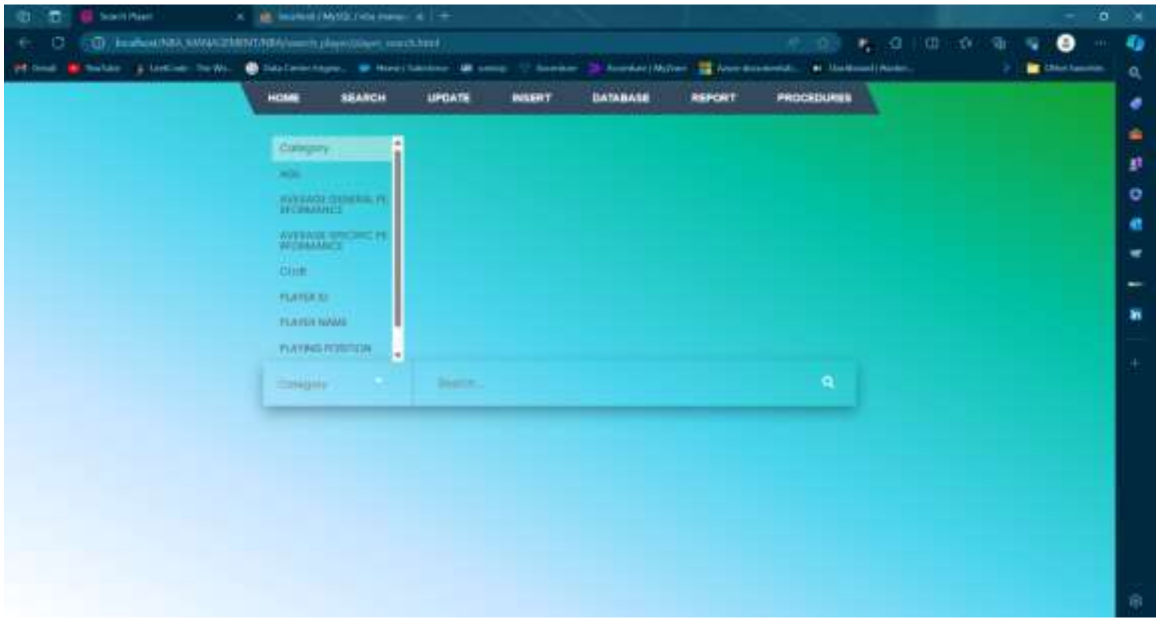


Fig.5.2 Search Players

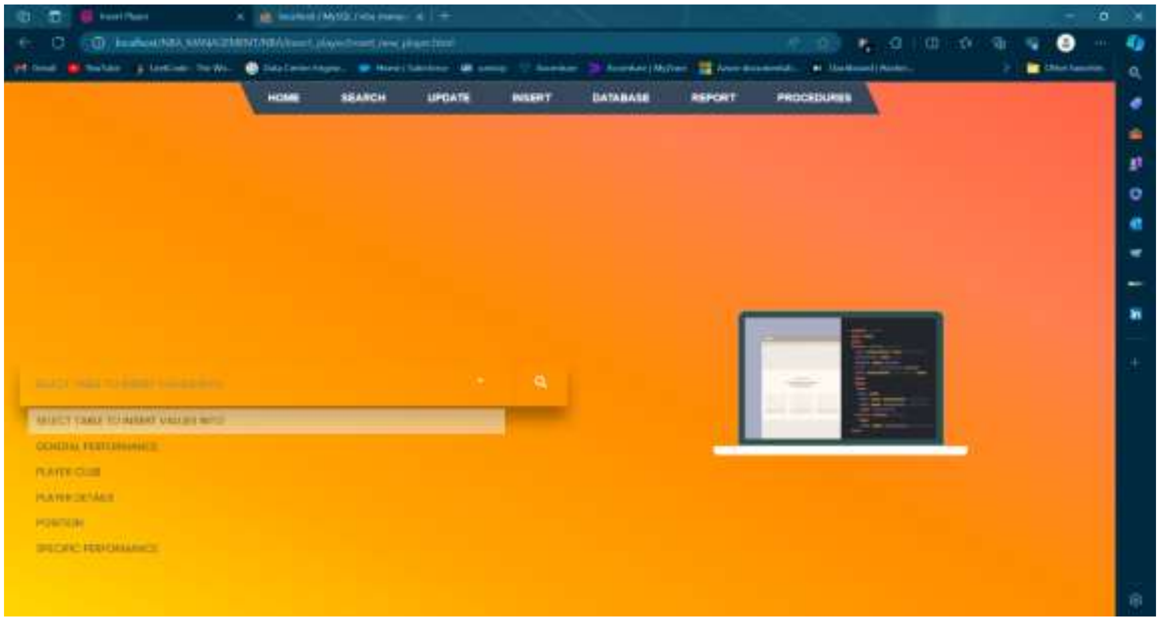


Fig. 5.3 Insert Players

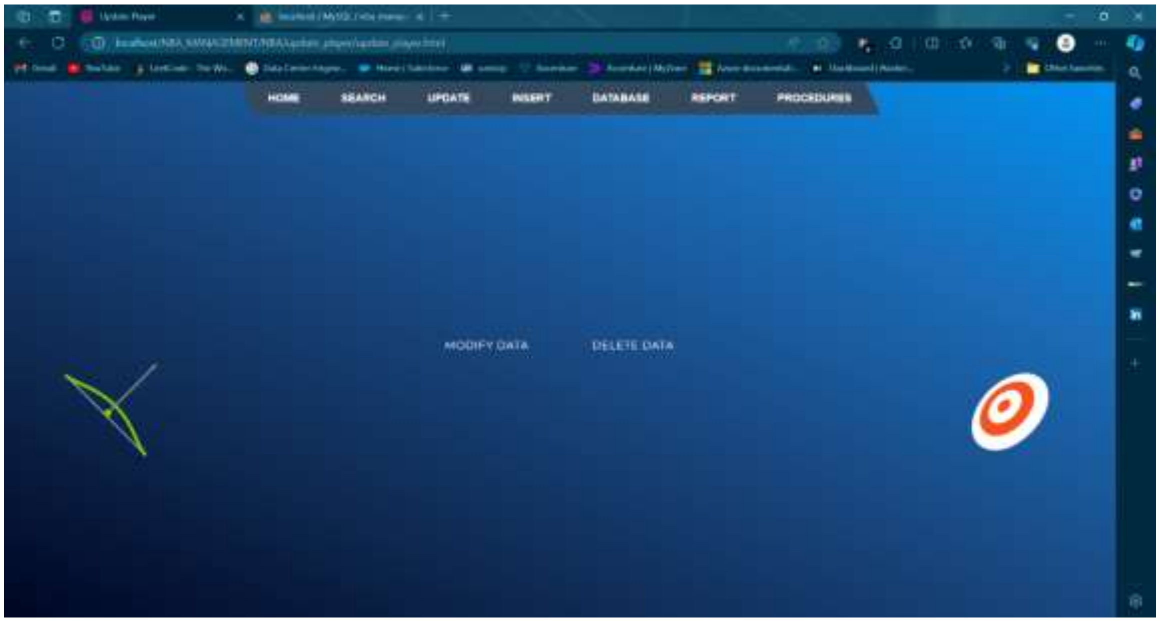


Fig. 5.4 Update Players

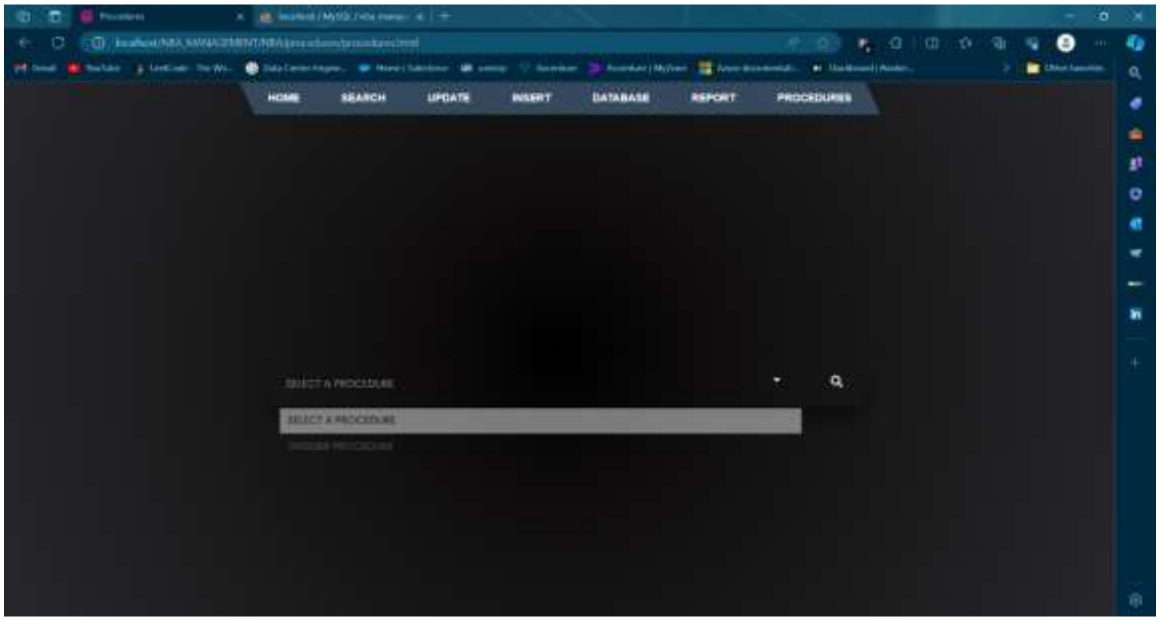


Fig. 5.5 Procedure

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENT

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 basketball 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 Enhancements:

Integration with Live Game Data: Incorporate real-time integration with live game data feeds to provide coaches with immediate insights into player performance, game trends, and opponent analysis during matches.

Player Injury Tracking and Management: Develop a comprehensive module for tracking player injuries, rehabilitation progress, and medical history to better manage player health and reduce injury risks.

Salary and Contract Management: Implement features for managing player contracts, salaries, and financial incentives, including contract negotiation tools and salary cap management functionalities.

Video Analysis Tools: Integrate video analysis tools for reviewing game footage, analyzing player movements, and identifying areas for improvement in technique and strategy.

International Player Database: Expand the player database to include international players from various leagues around the world, providing a global talent pool for scouting and recruitment.

Player Performance Benchmarking: Develop tools for benchmarking player performance against league averages, historical data, and performance metrics of top-performing players in similar roles.

Customizable Reporting and Analytics: Enable users to generate custom reports and analytics dashboards tailored to their specific needs, allowing for deeper insights and data visualization.

Integration with Player Development Programs: Partner with player development programs and academies to integrate training resources, skill assessments, and developmental pathways for aspiring players within the system.

Gamification for Player Engagement: Incorporate gamification elements to incentivize player engagement with the system, such as achievement badges.

REFERENCES

[1] "Database Systems: Design, Implementation, & Management" by Carlos Coronel, Steven Morris, and Peter Rob

[2] "Learning PHP, MySQL & JavaScript: With CSS & HTML" by Robin Nixon

[3] "Web Development with Node and Express: Leveraging the JavaScript Stack" by Ethan Brown

[4] "A Comparative Study of SQL and NoSQL Databases" by Pravin Mishra and Prachi Pandey

[5] "An Introduction to Modern Front-End Development" by Daniel C. Younger and Timothy J. McLaughlin

[6] "Web-Based Database Access" by Nenad Jukic, Susan Vrbsky, and Svetlozar Nestorov

[7] MDN Web Docs (developer.mozilla.org) for HTML, CSS, and JavaScript reference

[8] W3Schools (www.w3schools.com) for tutorials on web development technologies

[9] MySQL documentation (dev.mysql.com/doc/) for MySQL reference

[10] phpMyAdmin documentation (docs.phpmyadmin.net/) for phpMyAdmin reference

[11] WampServer documentation (www.wampserver.com/en/) for WampServer reference

[12] NBA official website (www.nba.com) for information on basketball, teams, and players

[13] Google Chrome documentation (developer.chrome.com/docs)

[14] WampServer documentation (www.wampserver.com/en/documentation.php)