**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**JNANASANGAMA, BELAGAVI - 590018**

****

# Mini Project Report

# on

CRICKET PLAYER STATS AND PREDICTION SYSTEM

*Submitted in partial fulfillment for the award of degree of*

**Bachelor of Engineering**

**in**

**Artificial Intelligence and Machine Learning**

Submitted by

|  |  |
| --- | --- |
| USN | Name |
| 1RN21AI047 | Dhruva N U |
| 1RN21AI043 | Darshan V V |

**

**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. **Dhruva N U** USN **1RN21AI047** and **Darshan V V** USN **1RN21AI043** 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**

The Cricket Database Management System (DBMS) is a web-based platform designed to revolutionize the experience of cricket enthusiasts by offering an interactive and comprehensive tool for team management, match simulation, and statistical analysis.

The system utilizes Tailwind CSS for streamlined and user-friendly web development, providing a visually appealing and responsive interface. Users can dynamically create teams, simulate matches, and analyze player performances through intuitive controls.

The project incorporates a robust database structure, ensuring efficient data management for players, teams, and matches. Security features, including user authentication and data validation, contribute to a secure environment.

Future enhancements include player transfers, tournament modes, advanced analytics, and a notification system, enhancing the system's functionality and user engagement. The Cricket DBMS represents a valuable resource for both casual fans and enthusiasts, offering a unique blend of entertainment and analytical exploration within the realm of cricket.

**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

**Dhruva N U-1RN21AI047**

**Darshan V V-1RN21AI043**

TABLE OF CONTENTS

**Chapter No.**  **CONTENTS** **PAGE NO.**

Abstract i

Acknowledgement ii

Table of Contents iii

Figure Names iv

1 Introduction

1.1 Overview of project 1

1.2 Motivation 1

1.3 Problem Definition 2

1.4 Objectives 2

2 System Requirements

2.1 Software and Hardware 5

3 Process

3.1 Methodology 6

3.2 Block diagram 7

3.3 Schema Diagram 8

4 Code

4.1 Sql Code 9

4.2 Front-end Code 13

5 Web Page

5.1 User Interface 18

6 Conclusion and future enhancement 20

**TABLE OF FIGURES**

**Fig No. FIGURE NAME Page No**

Fig. 3.1 Entity Relation Diagram 6

Fig. 3.2 Schema Diagram 7

Fig. 5.1 Home Page 16

Fig. 5.2 Compare Players 16

Fig. 5.3 Insert/Select Players 17

Fig. 5.4 Team Player Strength 17

Fig. 5.5 Individual Player Details 18

**Chapter 1**

INTRODUCTION

* 1. **Overview of Database Management System**

The cricket database management system (DBMS) project revolves around the meticulous design and implementation of a comprehensive cricket-oriented database. The system incorporates three primary entities: Players, Teams, and Matches, each characterized by a set of attributes capturing detailed information such as player statistics, team details, and match particulars.

Functionality is centered on the dynamic creation of teams through the addition of players, smartly dividing them based on specified criteria. The system facilitates match simulations and offers a robust mechanism to compare team performances, calculating and presenting various statistics. The user-friendly web interface allows seamless interaction, enabling users to view player details, team compositions, and match outcomes. Security measures, including user authentication and data validation, contribute to a robust system.

Future enhancements, such as player transfers, tournament modes, and advanced analytics, showcase the scalability and potential for continued development. The project's technological foundation includes a specified database management system (e.g., MySQL) and a web development stack (e.g., HTML, CSS, JavaScript, Node.js, Flask). Overall, this cricket DBMS provides a captivating platform for enthusiasts to delve into the strategic and statistical dimensions of cricket, offering a rich and interactive experience for users interested in team dynamics and player performances.

* 1. **Motivation**

The inspiration behind creating the cricket database project is to bring the excitement of cricket to fans in a fun and interactive way. The project aims to go beyond basic match stats and offer a platform where users can form their own teams, simulate matches, and analyze player performances. The goal is to make cricket more engaging, allowing fans to actively participate in the sport they love. Whether you're a casual fan or someone who enjoys digging into the details of the game, this project provides a user-friendly space to explore and enjoy cricket from a new perspective. The satisfaction comes from knowing that users can have a great time creating teams, playing out matches, and delving into the strategic side of cricket in an accessible and enjoyable manner.

* 1. **Problem Definition**

The problem addressed by the cricket database management system (DBMS) project is the need for a comprehensive and user-friendly platform that allows cricket enthusiasts to actively engage with the sport beyond traditional match statistics. The existing landscape often lacks tools that empower users to create teams, simulate matches, and analyze player performances in a dynamic and interactive way. This project aims to fill this gap by providing a solution that enables users to easily add players, form teams, simulate matches, and assess team and player statistics. The lack of such a platform limits the interactive and participatory aspects of cricket fandom, and the project seeks to address this by offering a cohesive system that caters to both casual fans and those interested in a more analytical exploration of the game. The problem definition, therefore, revolves around enhancing the user experience in cricket-related activities and creating a platform that fosters a deeper and more engaging connection with the sport

* 1. **Key objectives**

The objectives for the cricket database management system (DBMS) project are as follows:

* **User Engagement:**
  + Create a user-friendly interface to encourage active participation and engagement from cricket enthusiasts.
  + Facilitate user interaction through features such as team creation, player additions, and match simulations.
* **Data Management:**
* Develop a robust database structure to efficiently store and manage player details, team information, and match statistics.
* Ensure data integrity and security through proper validation mechanisms.
* **Team Formation:**
  + Implement algorithms for smart team formation based on predefined criteria such as player statistics, skills, and roles.
* **Match Simulation:**
  + Enable users to simulate cricket matches, providing a realistic experience with dynamic outcomes.
  + Calculate and present match statistics, including runs scored, wickets taken, and overall team performance.
* **Comparison and Analysis:** 
  + Allow users to compare team performances and analyze player statistics to determine the strengths and weaknesses of each team.
  + Provide insightful analytics for users interested in a deeper understanding of the game.
* **Scalability:**
  + Design the system to handle an increasing number of players, teams, and matches over time.
  + Ensure the system can accommodate the addition of new players and teams seamlessly.
* **User Authentication and Security:**
  + Implement secure user authentication mechanisms to control access to sensitive features.
  + Incorporate data validation to prevent errors and ensure the integrity of the information stored in the database.
* **User Education and Enjoyment:**
  + Provide educational elements within the platform to enhance users' understanding of cricket strategies and statistics.
* **Future Enhancements:**
  + Lay the foundation for future enhancements, such as player transfers, tournament modes, and advanced statistical tools.
  + Create a platform that can evolve to meet the changing needs and interests of cricket enthusiasts.
* **Technological Integration:**
  + Utilize appropriate database management system technologies and web development stacks to create a seamless and efficient user experience.
* These objectives collectively aim to address the identified problem, providing a comprehensive and enjoyable platform for cricket enthusiasts to actively participate, analyze, and engage with the sport.

**Chapter 2**

SYSTEM REQUIREMENTS

* 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: Flask, Python
* Backend Database: MySQL
* Database Management System: Main.py
* 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

* 1. **Methodology**

Here's a simplified version of the methodology for the cricket database management system (DBMS) project:

**Understand Needs:**Talk to users and stakeholders to understand what they want from the cricket DBMS. Identify key features and how users will interact with the system.

**DesignDatabase:** Create a blueprint for storing player, team, and match information efficiently.

**Build User Interface:**Design an easy-to-use web page where users can add players, form teams, and simulate matches.

**Create Backend:**Develop the logic that makes everything work behind the scenes, including team formation and match simulation.

**Develop Frontend:**Build the visible parts of the web page using HTML, CSS, and JavaScript.

**Set Up Database:** Put the designed database into action using a Database Management System (e.g., MySQL).

**Ensure Security:**Make sure the system is secure by adding user authentication and data validation.

**Test Thoroughly:**Check that everything works as expected and fix any issues found during testing.

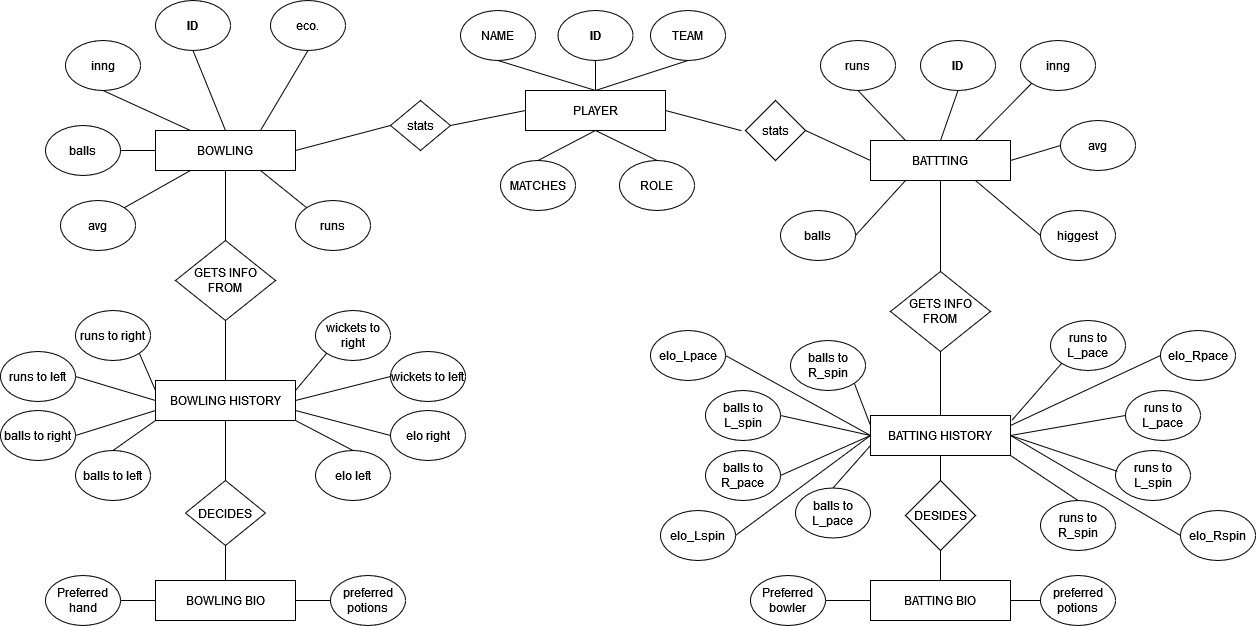
**Educate Users:**Include helpful information in the system to guide users on how to use it effectively.

**Deploy the System:**Make the system available online so users can access it.

**Gather Feedback:**Ask users for their thoughts and make improvements based on their feedback.

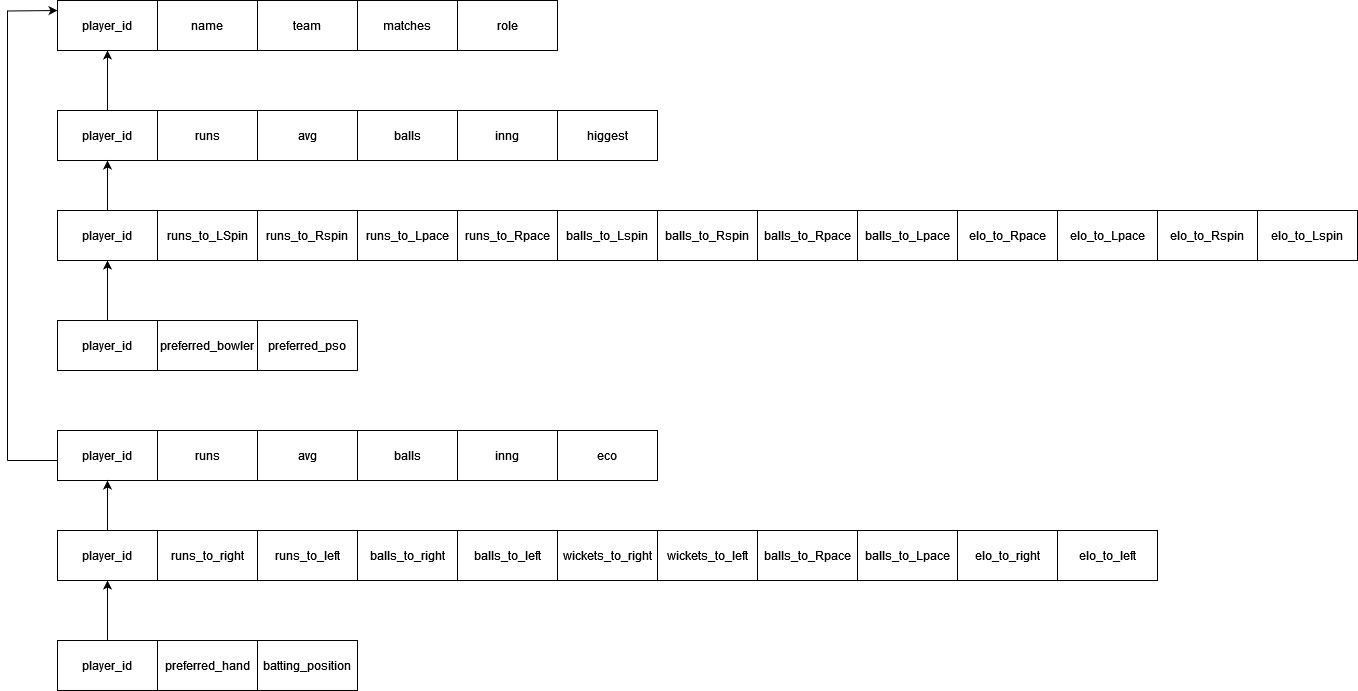
**Document Everything:** Create clear instructions for how the system works for future reference.

* 1. **Entity Relation Diagram**



### Fig. 3.1 Entity-Relationship Diagram

* 1. **Schema Diagram**



### Fig. 3.2 Schema Diagram

**Chapter 4**

CODE

* 1. **SQL CODE**

CREATE DATABASE player\_stats

--@block

CREATE TABLE team(

team\_name VARCHAR(10) PRIMARY KEY

);

--@block

CREATE TABLE player(

player\_id INTEGER PRIMARY KEY,

player\_name VARCHAR(20),

No\_of\_matches INTEGER,

player\_role ENUM('batsman','bowler','all-rounder','wicketkeeper'),

bowlingType ENUM('Lspin','Lpace','Rspin','Rpace'),

battingType ENUM('right','left'),

team\_name VARCHAR(10) REFERENCES team(team\_name)

);

--@block

CREATE TABLE battingStats(

player\_id INTEGER REFERENCES player(player\_id) ON DELETE CASCADE,

inngs INTEGER,

runs INTEGER,

SR DECIMAL(3,2),

battingAvg DECIMAL(3,2),

best INTEGER,

PRIMARY KEY (player\_id)

);

--@block

CREATE TABLE bowlingStats(

player\_id INTEGER REFERENCES player(player\_id) ON DELETE CASCADE,

inngs INTEGER,

wickets INTEGER,

runs\_conceded INTEGER,

economy DECIMAL(4,2),

best\_bowling\_figures VARCHAR(10),

PRIMARY KEY (player\_id)

);

--@block

CREATE TABLE battingbio(

player\_id INTEGER REFERENCES player(player\_id) ON DELETE CASCADE,

prefered\_bowler ENUM('Lspin','Lpace','Rspin','Rpace'),

prefered\_position ENUM('top','middle','lower','tail'),

PRIMARY KEY (player\_id)

);

CREATE TABLE battingHistory(

player\_id INTEGER REFERENCES player(player\_id) ON DELETE CASCADE,

runs\_to\_Lspin INTEGER,

runs\_to\_Rspin INTEGER,

runs\_to\_Rpace INTEGER,

runs\_to\_Lpace INTEGER,

out\_to\_Lspin INTEGER,

out\_to\_Rspin INTEGER,

out\_to\_Rpace INTEGER,

out\_to\_Lpace INTEGER,

balls\_Lspin INTEGER,

balls\_Rspin INTEGER,

balls\_Rpace INTEGER,

balls\_Lpace INTEGER,

batting\_avg\_Lspin DECIMAL(5,2),

batting\_avg\_Rspin DECIMAL(5,2),

batting\_avg\_Rpace DECIMAL(5,2),

batting\_avg\_Lpace DECIMAL(5,2),

elo\_rating\_Lspin DECIMAL(6,2),

elo\_rating\_Rspin DECIMAL(6,2),

elo\_rating\_Rpace DECIMAL(6,2),

elo\_rating\_Lpace DECIMAL(6,2),

PRIMARY KEY (player\_id)

);

--@block

CREATE TABLE bowlingbio (

player\_id INTEGER REFERENCES player(player\_id) ON DELETE CASCADE,

prefered\_batting\_hand ENUM('left', 'right'),

prefered\_position ENUM('new', 'first\_change', 'second\_change', 'death'),

PRIMARY KEY (player\_id)

);

CREATE TABLE bowlingHistory (

player\_id INTEGER REFERENCES player(player\_id) ON DELETE CASCADE,

runs\_conceded\_to\_left INTEGER,

runs\_conceded\_to\_right INTEGER,

wickets\_against\_left INTEGER,

wickets\_against\_right INTEGER,

balls\_bowled\_to\_left INTEGER,

balls\_bowled\_to\_right INTEGER,

strikeRate\_right DECIMAL(5,2),

strikeRate\_left DECIMAL(5,2),

bowling\_avg\_against\_left DECIMAL(5,2),

bowling\_avg\_against\_right DECIMAL(5,2),

elo\_rating\_against\_left DECIMAL(6,2),

elo\_rating\_against\_right DECIMAL(6,2),

PRIMARY KEY (player\_id)

);

--@block -- to clear all the tables

DELETE FROM player;

DELETE FROM battingbio;

DELETE FROM battingHistory;

DELETE FROM battingStats;

DELETE FROM bowlingbio;

DELETE FROM bowlingstats;

DELETE FROM bowlinghistory

--@block --to view all the tables

SELECT \* FROM player;

SELECT \* FROM battingbio;

SELECT \* FROM battingHistory;

SELECT \* FROM battingStats;

SELECT \* FROM bowlingbio;

SELECT \* FROM bowlingstats;

SELECT \* FROM bowlinghistory;

--@block

DELETE FROM player where player\_id = 23;

DELETE FROM battingbio where player\_id = 23;

DELETE FROM battingHistory where player\_id = 23;

DELETE FROM battingStats where player\_id = 23;

DELETE FROM bowlingbio where player\_id = 23;

DELETE FROM bowlingstats where player\_id = 23;

DELETE FROM bowlinghistory where player\_id = 23;

--@block

SELECT player\_id FROM player WHERE player\_name = 'Shubam Gill';

* 1. **FRONT-END CODE**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Team Comparison</title>

<!-- Include Bootstrap CSS -->

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<style>

body {

font-family: Arial, sans-serif;

margin: 20px;

}

h1 {

text-align: center;

margin-bottom: 30px;

}

.progress-wrapper {

margin-bottom: 20px;

}

.progress-label {

font-weight: bold;

margin-bottom: 5px;

}

.progress-bar {

border-radius: 10px;

}

.player-list {

list-style: none;

padding-left: 20px;

cursor: pointer;

/\* Add cursor pointer for better UX \*/

}

.gauge {

width: 100%;

max-width: 250px;

font-family: "Roboto", sans-serif;

font-size: 32px;

color: #004033;

position: fixed;

bottom: 20px;

/\* Adjust this value to set the distance from the bottom \*/

left: 50%;

transform: translateX(-50%);

}

.gauge\_\_body {

width: 100%;

height: 0;

padding-bottom: 50%;

background: #b4c0be;

position: relative;

border-top-left-radius: 100% 200%;

border-top-right-radius: 100% 200%;

overflow: hidden;

}

.gauge\_\_fill {

position: absolute;

top: 100%;

left: 0;

width: inherit;

height: 100%;

background: #009578;

transform-origin: center top;

transform: rotate(0.25turn);

transition: transform 0.2s ease-out;

}

.gauge\_\_cover {

width: 75%;

height: 150%;

background: #ffffff;

border-radius: 50%;

position: absolute;

top: 25%;

left: 50%;

transform: translateX(-50%);

}

</style>

</head>

<body>

<h1>Team Comparison Results</h1>

<div class="container">

<div class="row">

<!-- Team 1 Balance Progress Bar -->

<div class="col-md-6">

<div class="progress-wrapper">

<h2 class="progress-label">Team 1 Balance: {{ paras[0] }}%</h2>

<div class="progress">

<div class="progress-bar bg-success" role="progressbar" style="width: {{ paras[0] }}%;"

aria-valuenow="{{ paras[0] }}" aria-valuemin="0" aria-valuemax="100">{{ paras[0] }}%</div>

</div>

</div>

<h2>Players for Team 1:</h2>

<ul class="player-list">

{% for player in paras[3] %}

<li onclick="sendPlayerName('{{ player }}')">{{ player }}</li>

{% endfor %}

</ul>

</div>

<!-- Team 2 Balance Progress Bar -->

<div class="col-md-6">

<div class="progress-wrapper">

<h2 class="progress-label">Team 2 Balance: {{ paras[1] }}%</h2>

<div class="progress">

<div class="progress-bar bg-success" role="progressbar" style="width: {{ paras[1] }}%;"

aria-valuenow="{{ paras[1] }}" aria-valuemin="0" aria-valuemax="100">{{ paras[1] }}%</div>

</div>

</div>

<h2>Players for Team 2:</h2>

<ul class="player-list">

{% for player in paras[4] %}

<li onclick="sendPlayerName('{{ player }}')">{{ player }}</li>

{% endfor %}

</ul>

</div>

<form action="{{url\_for('handle\_player\_click')}}" method="post" id='forID'><input type="hidden" value=""

name='player' id="frm"></form>

<!-- Gauge -->

<div class="gauge">

<div class="gauge\_\_body">

<div class="gauge\_\_fill"></div>

<div class="gauge\_\_cover"></div>

</div><br>

<button onclick='someFun("{{paras}}")'>click for info</button>

</div>

</div>

</div>

<!-- Include Bootstrap JS (optional) -->

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<script>

const gaugeElement = document.querySelector(".gauge");

const parasValue = gaugeElement.getAttribute("data-paras");

const parasValueAsNumber = parseFloat(parasValue);

var val;

var per;

function setGaugeValue(gauge, value) {

if (value < 0 || value > 1) {

return;

}

gauge.querySelector(".gauge\_\_fill").style.transform = `rotate(${value / 2}turn)`;

gauge.querySelector(".gauge\_\_cover").textContent = `${Math.round(value \* 100)}`;

}

function someFun(parameter) {

val = parameter[2]

per = (val / 10)

console.log(val);

setGaugeValue(gaugeElement, per);

}

const variableName = document.getElementById('frm')

const variableform = document.getElementById('forID')

function sendPlayerName(playerName) {

variableName.value = playerName

variableform.submit()

}

</script>

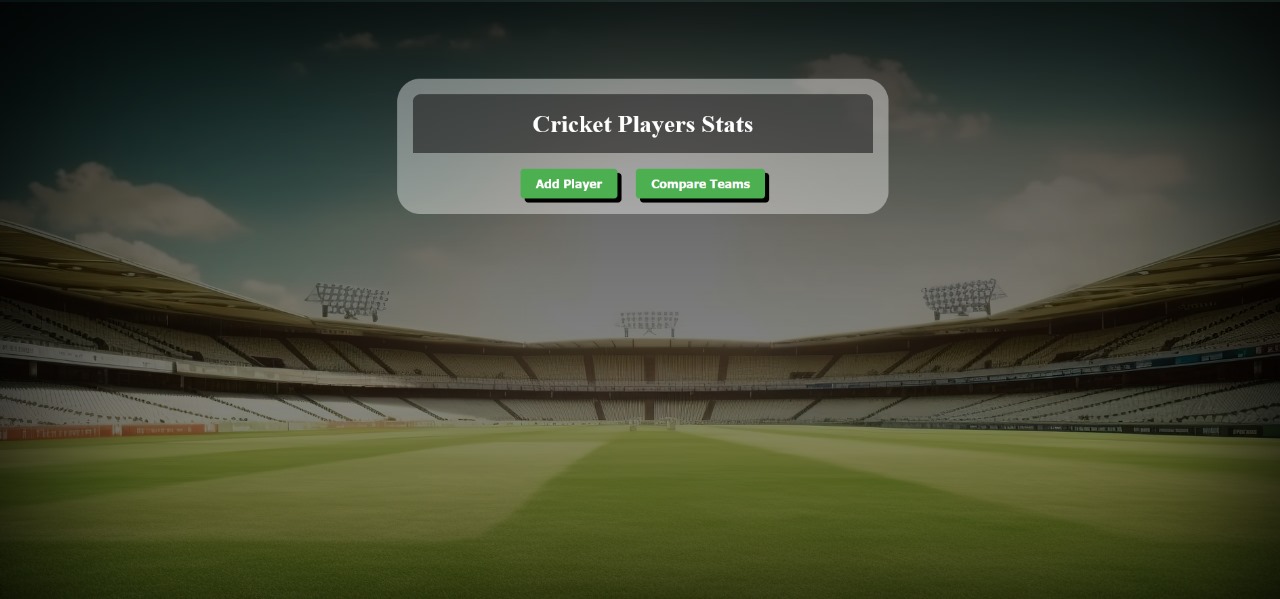
</body>

</html>

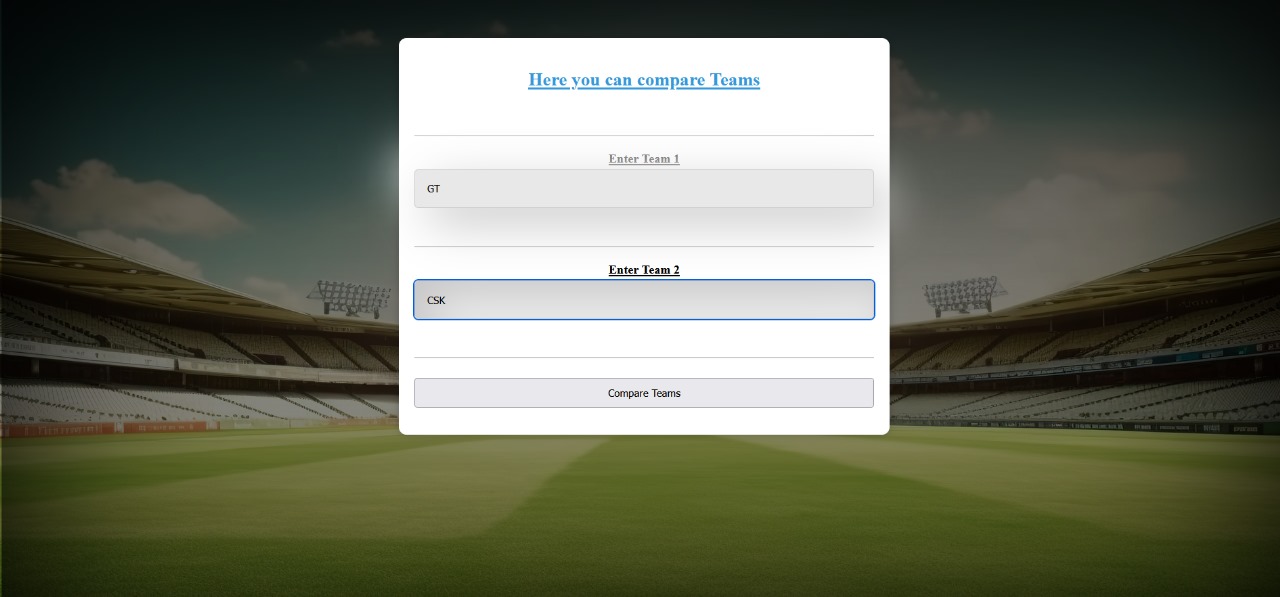
**CHAPTER 5**

**WEB PAGES OUTPUT**

**5.1 USER INTERFACES**



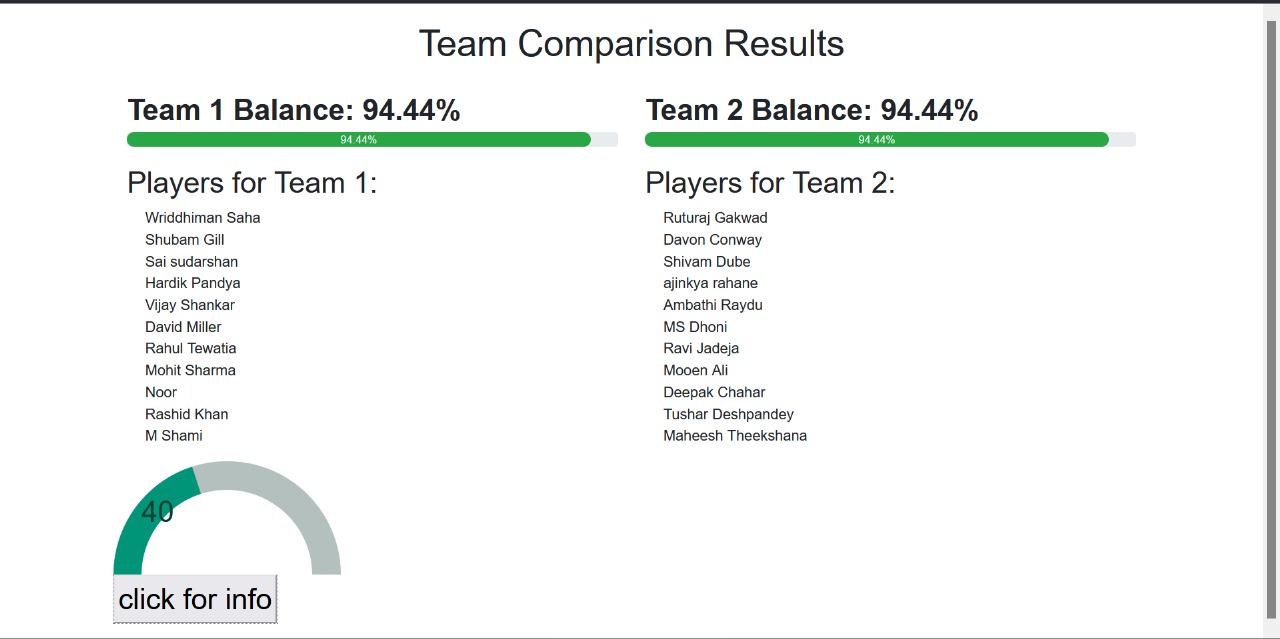
### Fig. 5.1 Home Page



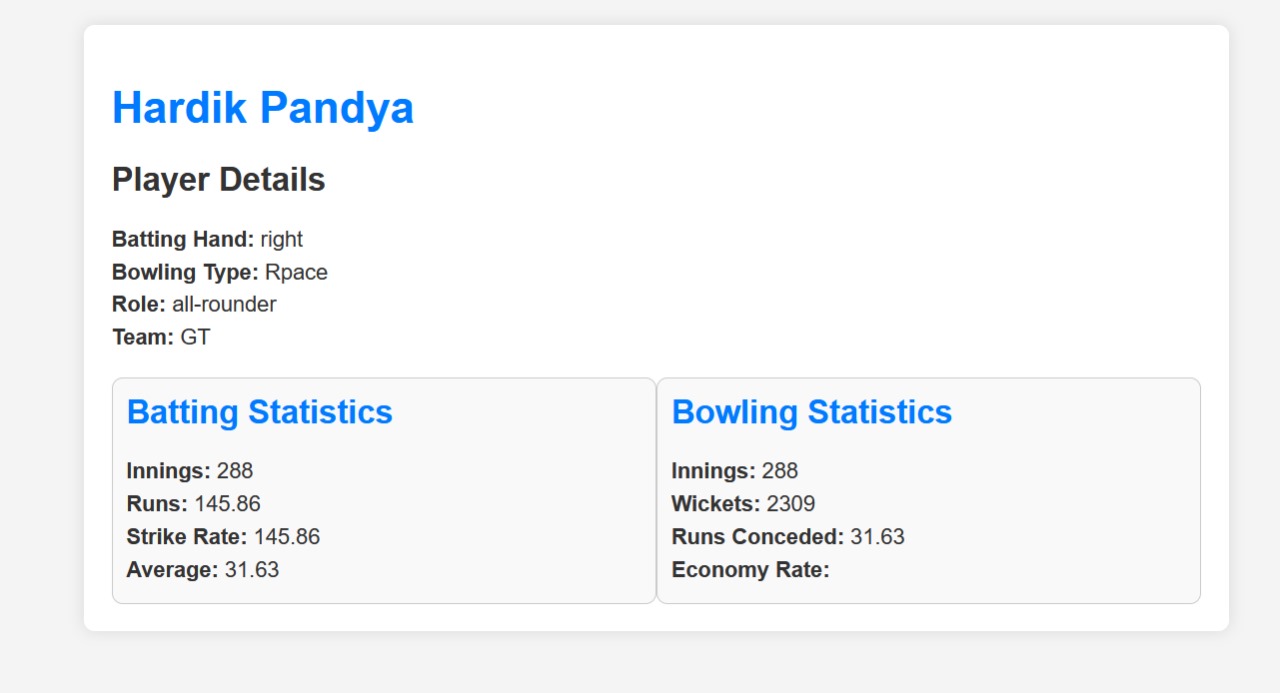
### Fig.5.2 Compare Players

### 

### Fig. 5.3 Insert/Select Players



### Fig. 5.4 Team Players Strength



### Fig. 5.5 Individual Player Details

**Chapter 6**

CONCLUSION AND FUTURE ENHANCEMENT

**Conclusion:**

In conclusion, our analysis of cricket player statistics provided valuable insights into individual performance. We identified top-performing batsmen based on batting averages and strike rates. Additionally, bowlers with impressive wicket counts were highlighted. Moving forward, players can focus on improving their key metrics to enhance overall team performance.

**Future Enhancements:**

* **Ongoing matches Predictive Modeling**:
  + Implement machine learning models to predict player performance based on historical data.
* **Dynamic Visualizations**:
  + Create interactive dashboards or web applications to visualize player stats.
  + Allow users to filter data by format, team, or time period.
* **Team Analysis**:
  + Extend the analysis to team-level performance.
  + Explore team dynamics, win-loss ratios, and player contributions to team victories.
* **Player Profiles**:
  + Develop detailed profiles for each player, including career highlights, milestones, and memorable performances.
  + Include visualizations and infographics for a comprehensive view.
* **Mobile Responsiveness:**
* Ensure the web interface is fully responsive for optimal user experience on various devices.
* Implement a mobile-friendly design for users on smartphones and tablets.
* **Live Match Updates:**
* Integrate real-time updates for.

REFERENCES

1. "Database Systems: Design, Implementation, & Management" by Carlos Coronel, Steven Morris, and Peter Rob
2. "Beginning Node.js" by Basarat Ali Syed
3. "Learning Flask, MySQL & Python: With jQuery, CSS & HTML5" by Robin Nixon
4. "Web Development with Node and Express: Leveraging the JavaScript Stack" by Ethan Brown
5. "A Survey of Web Development Frameworks for Node.js" by Atif Mashkoor and Imran Rashid
6. "A Comparative Study of SQL and NoSQL Databases" by Pravin Mishra and Prachi Pandey
7. "An Introduction to Modern Front-End Development" by Daniel C. Younger and Timothy J. McLaughlin
8. "Web-Based Database Access" by Nenad Jukic, Susan Vrbsky, and Svetlozar Nestorov
9. Cricket official website (www.iplcricket.com) for information on ipl, teams, and players
10. Microsoft documentation for Windows OS(docs.microsoft.com/en-us/windows/)
11. Google Chrome documentation (developer.chrome.com/docs)
12. Brackets documentation (brackets.io/docs/)
13. WampServer documentation (www.wampserver.com/en/documentation.php)
14. "Modern Web Development: Understanding Full Stack Development" by JetBrains
15. Case studies from companies or organizations that have implemented similar player management systems or sports management software