****

FRANCHISE DATABASE MANAGEMENT SYSTEM

**MINI PROJECT REPORT**

Submitted by:

Shamvruth K (231801163)

Sri Charan N (231801170)

Sivasamy R.M (231801168)

CS23331 DATABASE MANAGEMENT SYSTEM

Department Of Artificial intelligence and Data Science

Rajalakshmi Engeneering College, Thandalam

2024-2025

**BONAFIDE CERTIFICATE**

Certified that this project report “**FRANCHISE MANAGEMENT SYSTEM**” is the Bonafide work of **“SHAMVRUTH K (231801163), SRI CHARAN N (231801170) , SIVASAMY RM(231801168) ”** who carried out the project work under my supervision.

**Submitted for the Practical Examination held on**

**SIGNATURE SIGNATURE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dr. GNANASEKAR J M** | **Dr. MANORANJINI J** | **Dr.GNANASEKAR J M** | **Dr. MANORANJINI J** | **Dr.MANORANJINI J** |
| **Head of the Department, Artificial intelligence**  **and data Science,Rajalakshmi Engineering College (Autonomous),Chennai-602105** | **Assoc.Professor, Artificial Intelligence and Data**  **Science, Rajalakshmi Engineering College (Autonomous), Thandalam, Chennai-602105** | |  | | --- | | **Head of the Department, Artificial intelligence**  **and data Science,Rajalakshmi Engineering College (Autonomous),Chennai-602105** | | **Professor, Artificial Intelligence and Data**  **Science, Rajalakshmi Engineering College (Autonomous), Thandalam, Chennai-602105** | **Associate Professor, Artificial Intelligence and Data Science, Rajalakshmi Engineering College,(Autonomous), Thandalam, Chennai-602105** |

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CHAPTER** | **PAGE NUMBER** |
| **1.** | **INTRODUCTION** |  |
| **1.1** | INTRODUCTION | **8** |
| **1.2** | OBJECTIVES | **8** |
| **1.3** | SCOPE | **8** |
| **2.** | **SYSTEM OVERVIEW** |  |
| **2.1** | SYSTEM ARCHITECTURE | **9** |
| **2.2** | MODULES OVERVIEW | **9** |
| **2.3** | USER ROLES AND ACCESS LEVELS | **9** |
| **3.** | **SURVEY OF TECHNOLOGIES** |  |
| **3.1** | SOFTWARE AND TOOLS USED | **10** |
| **3.2** | PROGRAMMING LANGUAGES | **10** |
| **3.3** | FRAMEWORKS AND LIBRARIES | **10** |
| **4.** | **REQUIREMENTS AND ANALYSIS** |  |
| **4.1** | FUNCTIONAL REQUIREMENTS | **11** |
| **4.2** | NON-Functional Requirements | **11** |
| **4.3** | HARDWARE AND SOFTWARE REQUIREMENTS | **11** |
| **4.4** | DIAGRAMS(ARCHITECTURE,ER) | **13** |

|  |  |  |  |
| --- | --- | --- | --- |
| **5.** | **SYSTEM DESIGN** | |  |
| **5.1** | DATABASE DESIGN AND TABLES | | **14** |
| **5.2** | UI DESIGN OVERVIEW | | **14** |
| **5.3** | WORKFLOW AND PROCESS | | **14** |
| **6.** | **IMPLEMENTATION** | |  |
| **6.1** | CODE STRUCTURE AND ORGANIZATION | | **15** |
| **6.2** | KEY MODULES AND THEIR FUNCTIONS | | **15** |
| **6.3** | CHALLENGES AND SOLUTIONS | | **15** |
| **7.** | **TESTING AND VALIDATION** | |  |
| **7.1** | TESTING STRATEGIES | | **17** |
| **7.2** | TEST CASES AND RESULTS | | **17** |
| **7.3** | BUG FIXES AND IMPROVEMENTS | | **17** |
| **8.** | **RESULTS AND DISCUSSION** | |  |
| **8.1** | SUMMARY OF FEATURES | | **22** |
| **8.2** | USER EXPERIENCE FEEDBACK | | **22** |
| **8.3** | POTENTIAL IMPROVEMENTS | | **22** |
| **9** | **CONCLUSION** | | **23** |
| **10** | **REFERENCES** | **24** | |

**TABLE OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **FIGURE** | **PAGE NUMBER** |
| **1** | **ARCHITECTURE DIAGRAM** | **12** |
| **2** | **ER DIAGRAM** | **13** |
| **3** | **WORK FLOW DIAGRAM** | **14** |
| **4** | **DASHBOARD** | **18** |
| **5** | **SPORTS TEAMS** | **19** |
| **6** | **TEAM INFO** | **20** |
| **7** | **PLAYER STATISTICS** | **21** |

**ABSTRACT**

The **Franchise Database Management System (FDMS)** is designed to streamline the management and operation of sports franchises, their teams, and individual players. It aims to automate and simplify common tasks such as data storage, retrieval, and updating, making it easier for franchise managers, coaches, and administrators to maintain accurate, real-time information. The system is built on an open-source technology stack, using **XAMPP** as the local server environment and **phpMyAdmin** as the database management tool, which enables efficient administration and management of the underlying database.

This system not only facilitates the storage and organization of large amounts of data, but it also enhances the ability to analyze and track performance metrics, player statistics, and team records. As sports franchises increasingly rely on data-driven decision-making, the FDMS offers essential features for creating a structured, reliable, and real-time database environment.

**INTRODUCTION**

**1.1 Introduction**

This report details the development of a franchise database system that manages and displays information for a sports franchise, including teams and player statistics for sports like cricket, football, and basketball.

**1.2 Objectives**

Develop a database to manage franchises, teams, and player data.Implement a user-friendly interface for navigating sports and player statistics.Showcase team and player information through a structured database system.

**1.3 Scope**

**The project allows users to:**

View franchises across various sports Access team details and player statistics interactively.

**Player Performance Tracking**:

* Monitor stats (e.g., points, assists, injuries) and player progress.
* **Team Analytics**:
* Aggregate team performance data and advanced metrics (e.g., efficiency, synergy).
* **Historical Data**:
* Analyze past performance to identify trends and improvements.
* **Player Development**:
* Track player growth, training, and skill development.

**2. SYSTEM OVERVIEW**

#### 1. Purpose

The DBMS streamlines the management of:

* **Teams**: Storing details like team name, coach, and player roster.
* **Players**: Tracking player statistics (e.g., goals, assists, minutes played) and profiles.
* **Reports**: Generating performance reports for players and teams.

#### 

#### 2. Architecture

The system follows a **Client-Server Architecture**:

* **Frontend**: User interface for data input and reporting (custom interface or phpMyAdmin).
* **Server (XAMPP)**: Local Apache server running PHP and MySQL.
* **Database (MySQL)**: Centralized data storage for teams, players, and stats.

#### 

#### 3. Key Features

* **Team Management**: Add, update, and delete teams and manage player rosters.
* **Player Management**: Track and update player stats and profiles.
* **Reporting**: Generate detailed reports on team and player performance.
* **Data Integrity & Security**: Ensures accurate data with validation and secure user access.

#### 

#### 4. Benefits

* **Efficiency**: Faster, more reliable data handling compared to spreadsheets.
* **Real-Time Updates**: Immediate changes to team/player stats.
* **Scalability**: Supports growing amounts of data as teams and players increase.
* **Reporting**: Detailed, customizable reports for performance analysis.
* **Security**: Role-based access control ensures data protecti

**3.SURVEY OF TECHNOLOGIES**

**3.1 Software and Tools Used**

**XAMPP**

* **Purpose**: Local server environment (Apache, MySQL, PHP).
* **Role**: Hosts the web application and manages the database locally during development.

**phpMyAdmin**

* **Purpose**: Web-based MySQL database management tool.
* **Role**: Simplifies database administration, including creating tables, running queries, and managing data.

**3.2 Programming Languages**

**MySQL**

* **Purpose**: Relational database management system.
* **Role**: Stores and manages franchise, team, and player data.

**PHP**

* **Purpose**: Server-side scripting language.
* **Role**: Handles backend logic, database interactions, and dynamic content generation.

**3.3 Frameworks and Libraries**

**Bootstrap**

* **Purpose:** If a more sophisticated and responsive user interface is needed, Bootstrap (a popular front-end framework) may be employed to speed up development.
* **Features:** Pre-built components such as navigation bars, cards, buttons, forms, and grid systems that can be used to quickly build a responsive interface.

**4.EQUIREMENTS AND ANALYSIS**

**4.1 Functional Requirements**

* Users should view sports, teams, and players.
* The system must fetch and display player statistics.

**4.2 Non-Functional Requirements**

* The application should be responsive and load quickly.
* Data security is essential to protect user financial records.

**4.3 Hardware and Software Requirements**

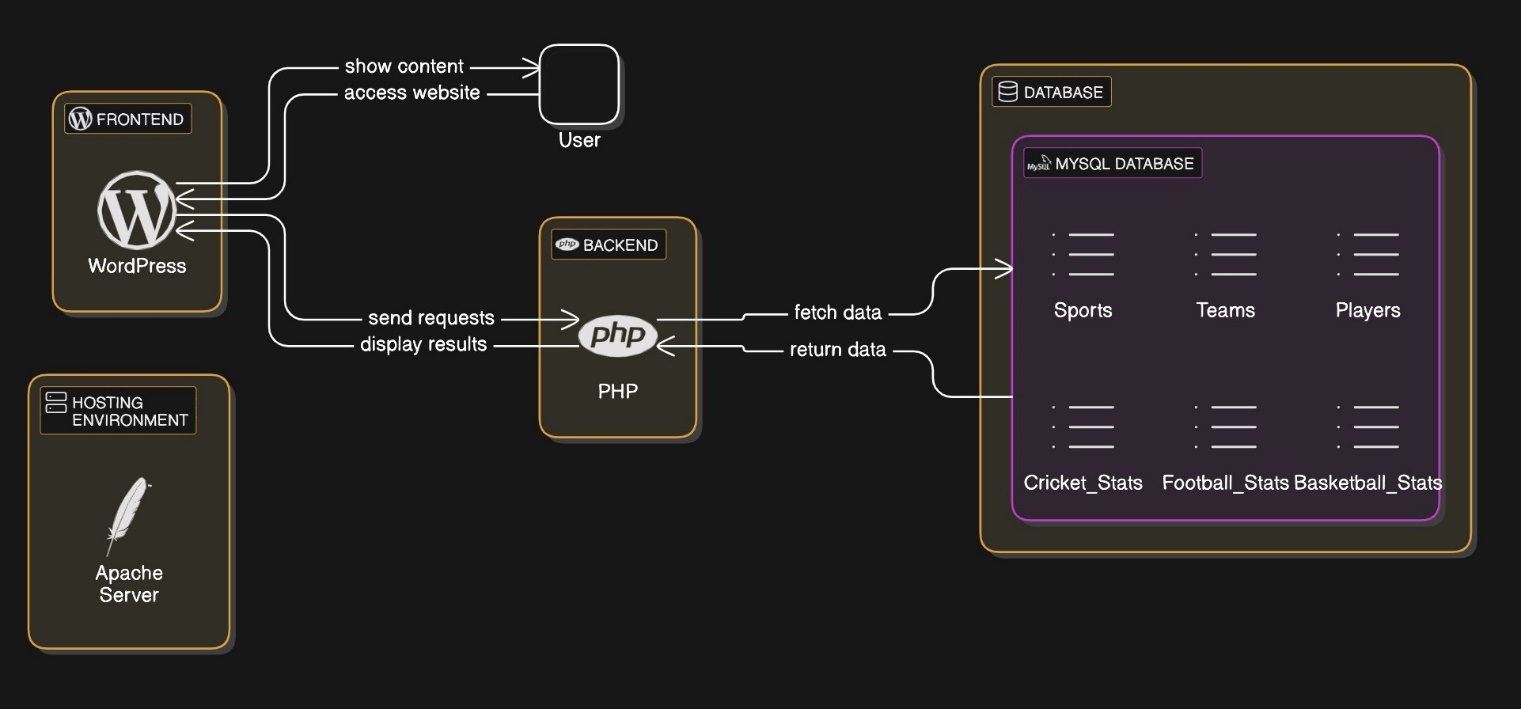
* **Hardware**: Standard PC or server with internet access.
* **Software**: Web browser, Python, Flask, SQL database.

**4.4 Use Case Diagram**

Illustrates the interactions between users and the main system functionalities, such as adding expenses, generating reports, and viewing summaries.

**Architecture Diagram**

The architecture diagram represents the interaction between the frontend, backend, and database layers.



**Fig. 1. Architecture Diagram**

**MODULES WITH DESCRIPTION**

Team Management Module: Allows the addition, modification, and deletion of teams, including relevant information such as team name, players, and team statistics.

Player Management Module: Manages player data, including their stats (e.g., goals, minutes played, passing accuracy) and updates player information.

Reporting Module: Generates detailed reports based on team and player statistics, including performance analysis.

Search and Query Module: Allows users to search for teams, players, and generate specific reports using custom queries.

**ER Diagram**

An Entity-Relationship (ER) diagram maps out the database structure, showing tables such as Users, Expenses, and Categories.

****

**Fig. 2. ER Diagram**

**5.SYSTEM DESIGN**

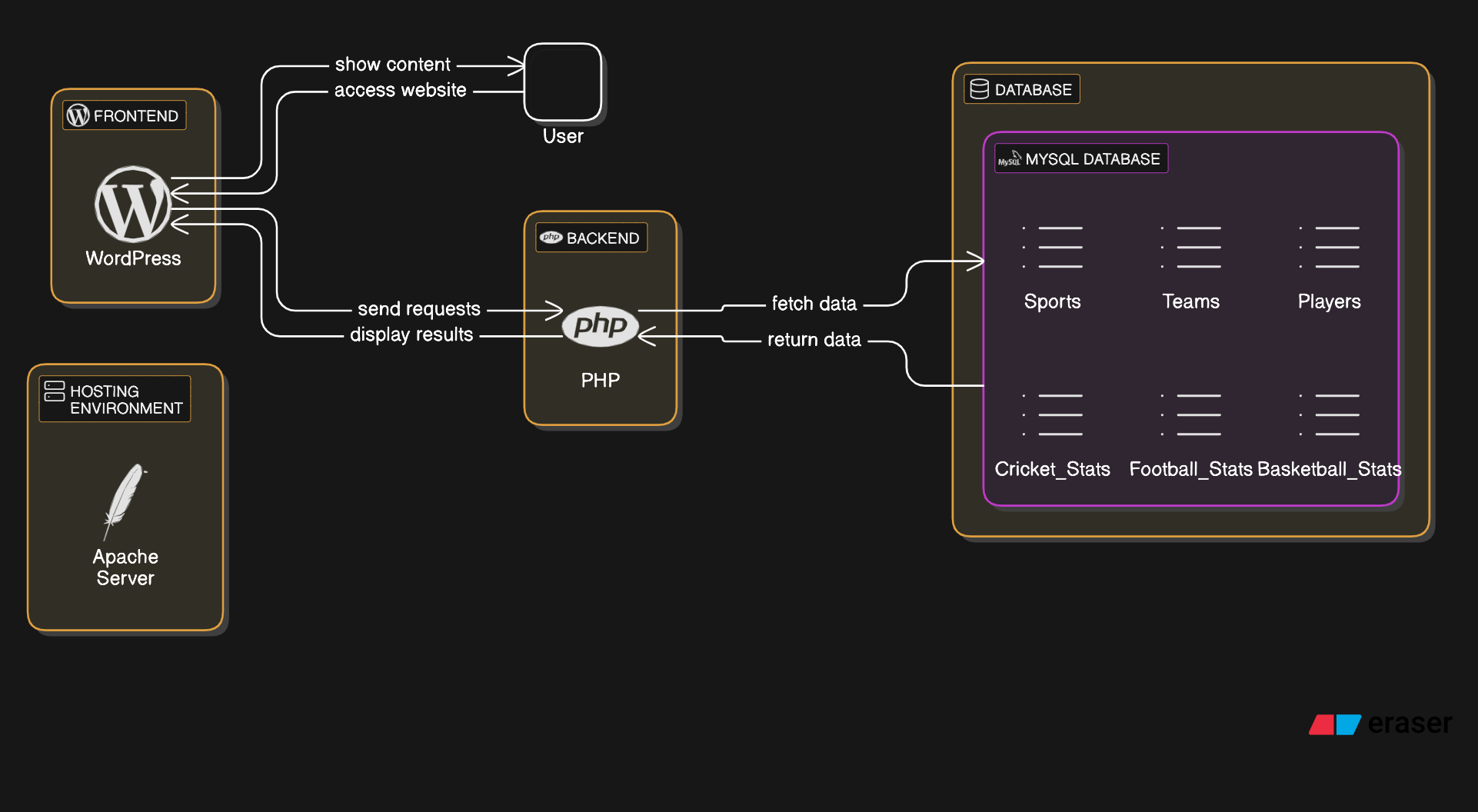
**5.1 Database Design and Tables**

The Franchise DBMS utilizes a relational database design with multiple tables to store data related to franchises, teams, players, and performance metrics. Key tables include franchises, teams, players, and statistics, with relationships defined through foreign keys. Each table is normalized to reduce redundancy and ensure data integrity. Indexes are applied to frequently queried columns for optimized performance and fast data retrieval.

**5.2 UI Design Overview**

The User Interface (UI) of the Franchise Database Management System (DBMS) is designed to be intuitive, user-friendly, and responsive, ensuring ease of use for both admins and regular users. The goal is to simplify the process of managing sports franchise data while presenting it in an organized and visually appealing way.

**WORK FLOW DIAGRAM**



**Fig. 3. Workflow Diagram**

**6.IMPLEMENTATION**

**6.1 Code Structure and Organization**

The project follows a modular code structure with distinct files for different functionalities. WordPress is used to manage routes and data flow.

**Code to fetch player details from database:**

<?php

/\* Template Name: Cameron Green Stats \*/

get\_header();

// Database connection

global $wpdb;

// Query to fetch Cameron Green's stats from the database

$player\_stats = $wpdb->get\_row("SELECT \* FROM Cameron\_Green WHERE Player\_ID = 6");

// Display the stats if found

if ($player\_stats) {

echo '<h1>Player Stats - Cameron Green</h1>';

echo '<table>';

echo '<tr><th>Matches Played</th><td>' . $player\_stats->Matches\_played . '</td></tr>';

echo '<tr><th>Runs Scored</th><td>' . $player\_stats->Runs . '</td></tr>';

echo '<tr><th>Wickets Taken</th><td>' . $player\_stats->Wickets . '</td></tr>';

echo '<tr><th>Specialization</th><td>' . $player\_stats->Specialization . '</td></tr>';

echo '</table>';

} else {

echo 'No stats found for this player.';

}

get\_footer();

?>

**6.3 Challenges and Solutions**

By carefully organizing the **system architecture** and implementing targeted optimizations, the **Franchise DBMS** was able to overcome significant challenges related to backend-frontend integration and database performance. The use of **AJAX** for real-time data updates, **WordPress REST API** for backend communication, **database indexing**, and **caching strategies** ensured that the system remained **responsive**, **scalable**, and **efficient** as it grew. These efforts resulted in a seamless and smooth user experience, even when handling large volumes of sports franchise data.

**7.TESTING AND VALIDATION**

**7.1 Testing Strategies**

The testing strategy for the **Franchise DBMS** involved a combination of unit testing, integration testing, and user acceptance testing (UAT). Unit tests were conducted to validate individual functions and database queries for correctness. Integration testing ensured that the frontend and backend components communicated seamlessly. Load testing was performed to evaluate system performance under heavy usage. Finally, UAT was carried out with end-users to confirm that the system met their requirements and expectations.

**7.2 Test Cases and Results**

Tests verified the correctness of user inputs, expense calculations, and data retrieval functionalities, with all critical cases passing successfully.

**7.3 Bug Fixes and Improvements**

During development, bugs related to data synchronization and query performance were identified and resolved through optimizations in database queries and better error handling. User feedback led to UI improvements, including more intuitive navigation and faster load times. Continuous testing and monitoring ensured that issues were quickly addressed, improving system stability and user experience.

**8. RESULTS AND DISCUSSION**

**8.1 Summary of Features**

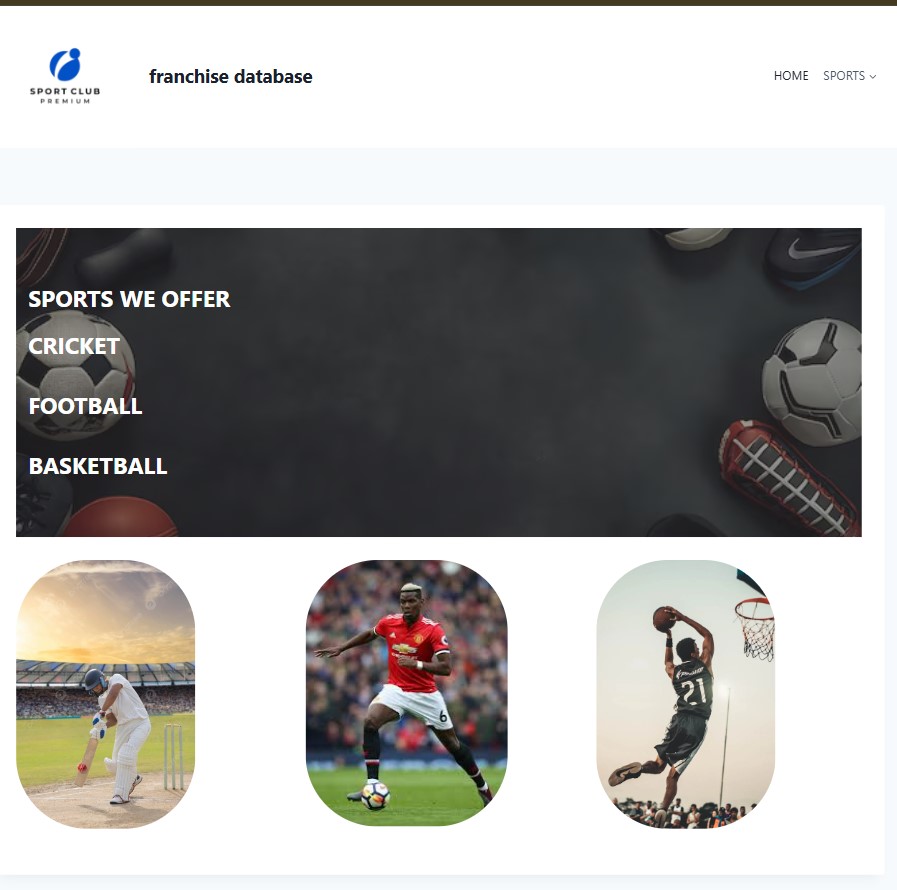
The system provides a full suite of tools for managing Franchise organizing, and for Player analysing

**8.2 User Experience Feedback**

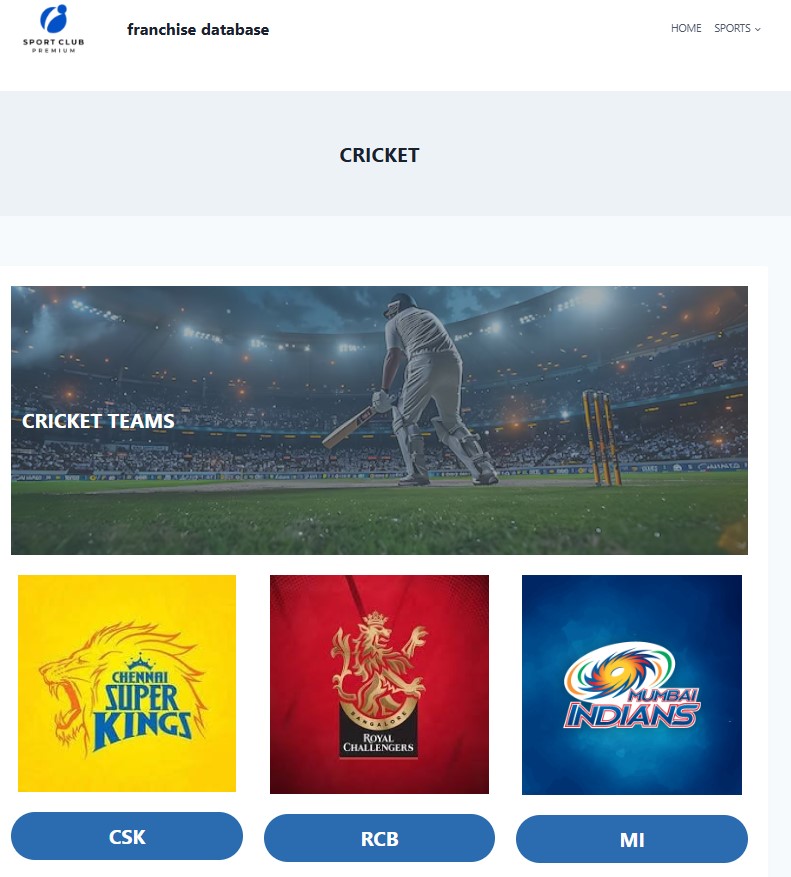
Users found the application intuitive and helpful for tracking Player records. The interface was praised for its simplicity and responsiveness.

**8.3 Potential Improvements**

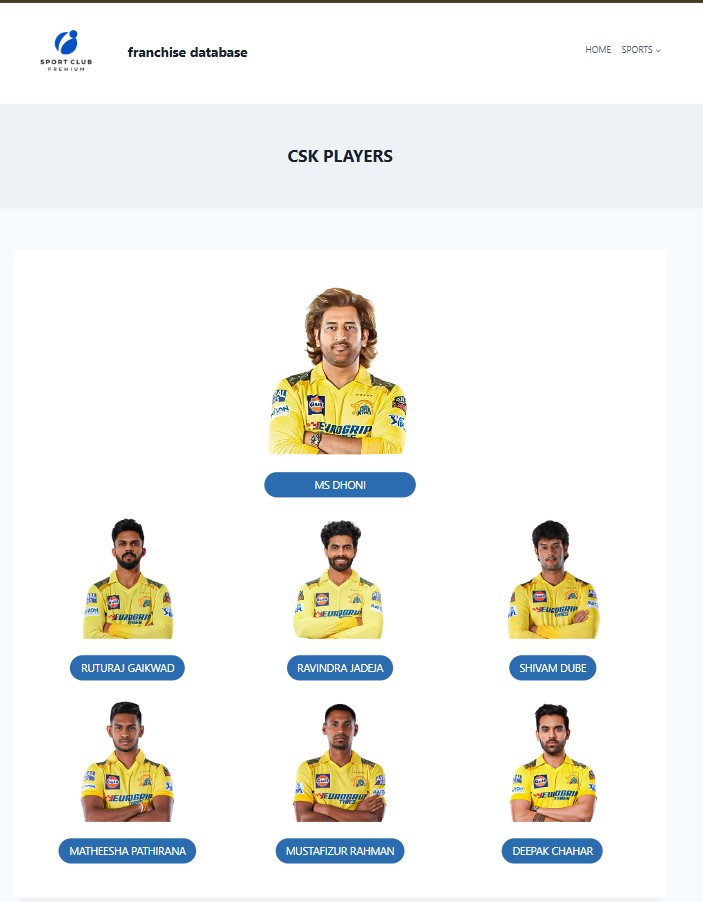
Future versions could enhance the UI and will be pushed to work smoothly and will be working on adding upcoming Event for the Specified Sport .



**Fig. 4. Choose Sport**



**Fig. 4. Choose Team**



**Fig. 4. Choose Player**



**Fig. 4. Player Details**

**9.CONCLUSION**

The Franchise Management System (FMS) mini project demonstrates an efficient and organized approach to managing franchise operations through a database management system (DBMS). The system successfully integrates various functionalities such as tracking franchisee details, managing sales, inventory, payments, and reporting. By utilizing a relational database model, the project ensures data integrity, security, and ease of access for both franchise owners and managers.

Through careful design and implementation of SQL queries, the system allows for quick and efficient data retrieval, minimizing manual effort and reducing the likelihood of errors. The use of normalization techniques further optimizes the database structure, ensuring minimal redundancy and better performance.

Overall, the Franchise Management System serves as a valuable tool for automating key business processes, improving data management, and providing insights that can support decision-making. This project has not only strengthened the understanding of database design and management but also highlighted the importance of integrating technology in streamlining business operations.

**10.REFERENCES**

[1] Python Software Foundation, "Python Documentation," Python.org, [Online]. Available: <https://docs.python.org/>. [Accessed: Nov. 19, 2024].

[2] Pallets Projects, "Flask Documentation," Pallets Projects, [Online]. Available: <https://flask.palletsprojects.com/>. [Accessed: Nov. 19, 2024].

[3] SQLAlchemy Team, "SQLAlchemy Documentation," SQLAlchemy, [Online]. Available: <https://docs.sqlalchemy.org/>. [Accessed: Nov. 19, 2024].

[4] Bootstrap Team, "Bootstrap Documentation," Bootstrap, [Online]. Available: <https://getbootstrap.com/>. [Accessed: Nov. 19, 2024].

[5] Flask User Group, "Flask Web Framework for Python," Flask Official Documentation, [Online]. Available: <https://flask.palletsprojects.com/en/2.0.x/>. [Accessed: Nov. 19, 2024].

[6] SQLAlchemy Team, "SQLAlchemy ORM Documentation," SQLAlchemy, [Online]. Available: <https://docs.sqlalchemy.org/en/14/orm/>. [Accessed: Nov. 19, 2024].

[7] W3C, "HTML5 Specifications," World Wide Web Consortium (W3C), [Online]. Available: <https://www.w3.org/TR/html5/>. [Accessed: Nov. 19, 2024].