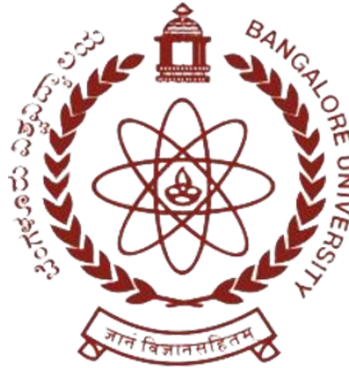


**BANGALORE UNIVERSITY**  
**UNIVERSITY VISVESVARAYA COLLEGE OF**  
**ENGINEERING**  
**K.R. Circle, Bangalore – 560001**



**Department of Computer Science and Engineering**  
A Mini-Project Report on  
**CRICKET TOURNAMENT MANAGEMENT SYSTEM**

Submitted by  
**Dhanush S P (U03NM21T006015)**  
**Ganesh Ramesha Markala (U03NM21T006020)**  
**V SEM, B. TECH (AIML)**

Under the guidance of  
**Dr. Kumaraswamy,**  
**Assistant Professor,**  
**Department of Computer Science,**  
**UVCE**

**Department of Computer Science and Engineering**

# BANGALORE UNIVERSITY

## UNIVERSITY VISVESVARAYA COLLEGE OF ENGINEERING

K.R. Circle, Bangalore – 560001



### Department of Computer Science and Engineering

## CERTIFICATE

This is to certify that **Ganesh Ramesha Markala** of V Semester, B. Tech, Artificial Intelligence and Machine Learning Engineering, bearing the register number **U03NM21T006020** has submitted the DBMS Mini-Project Report on “**CRICKET TOURNAMENT MANAGEMENT SYSTEM**”, in partial fulfilment for the DBMS Lab, prescribed by the Bangalore University for the academic year 2023-24.

**Mrs. Shaheeda Bi,**

Guest Faculty,

Dept. of CSE,

UVCE

**Dr. Champa H N**

Assistant Professor,

Dept. of CSE,

UVCE

**Dr. Thriveni J,**

Professor & Chairperson

Dept. of CSE,

UVCE

### Examiners:

1. ....

2. ....

## ACKNOWLEDGEMENT

I take this opportunity to thank our institution **University Visvesvaraya College of Engineering** for having given me an opportunity to carry out this project.

I would like to thank **Dr. Paul Vizhian, Principal, UVCE**, for providing us all the facilities to work on this project. I am indebted to him for being my pillar of strength and inspiration.

I wish to place my grateful thanks to **Dr. Thriveni J, Professor and Chairperson, Department of Computer Science and Engineering, UVCE**, who helped me to make my project a great success.

I am grateful to **Dr. Kumaraswamy, Assistant Professor, Department of Computer Science and Engineering, UVCE**, for his valuable suggestions and support, which has sustained me throughout the course of the project.

I am thankful to **Mrs. Shaheeda Bi, Guest Faculty, Department of Computer Science and Engineering, UVCE**, for her constant support during the course of the project.

I express my sincere thanks to all teaching and non-teaching staff, Department of Computer Science and Engineering, UVCE for all the facilities that they have provided me for successfully completing this project.

I am extremely grateful to Zander from Very Academy for having helped me learn development with the Django web framework.

I also thank my parents and friends for their continuous support and encouragement.

**Ganesh Ramesha Markala  
(U03NM21T006020)**

## **ABSTRACT**

The Cricket Tournament Management System is a database-driven application designed to effectively manage and organize cricket tournaments. The system allows for the input and tracking of teams, players, matches, and results. It also includes features for generating schedules, standings, and statistics. The system is built using a relational database management system and a user-friendly interface, making it easy to use for tournament organizers and administrators. Overall, the Cricket Tournament Management System is a powerful tool that streamlines the tournament management process and provides valuable insights and data for teams, players, and fans.

## TABLE OF CONTENTS

Title	Page No
1. Introduction	1
2. Literature Review	6
3. Proposed Work	10
4. Result	14
Conclusion	20
Bibliography	21

# CHAPTER 1

## INTRODUCTION

This chapter will discuss the various features and aim of this application.

### 1.1 Introduction to Cricket Tournament Management System

This chapter will introduce the Cricket Tournament Management System and how efficiently and coherently the tournament's data is saved.

### 1.2 Objective

The objective of this project is to develop a Cricket Tournament Management System as a web application that can effectively manage and organize cricket tournaments. The system aims to make easy the process of creating schedules, tracking results, and generating statistics. Additionally, the system aims to provide an easy-to-use interface for tournament organizers and administrators, making it simple to input and manage data. The end goal is to create a powerful tool that streamlines the tournament management process and provides valuable insights and data for teams, players, and fans.

The system aims to have three user models namely the 'Admin', 'Team Manager' and a 'Guest/Anonymous'. In this way, the application can have a strong access control over the database querying as specific permissions are assigned to each type of user.

The user interface aims to provide the following views for users:

- List of all the teams participating in the tournament.
- List of the details of all the players belonging to each team.
- List of all the matches of the tournament with their status and result.
- Points table to track the standings of each team.
- List of ongoing matches and the details of the last match that was played.
- Admin dashboard (only for 'Admin' user(s)) that can efficiently create, read, update and delete any type of data present in the database.
- Manager dashboard (only for 'Team Manager' user(s)) that makes it easy for the manager to add, delete or update a player's details of his team.

**Team Manager Dashboard:**

- Team manager has to specify the team details such as team name, batting coach, bowling coach while registering to the tournament.
- After the registration, a team with the name specified by the team manager is created and the manager is able to manage his team through a dashboard provided exclusively to him.
- The manager can add, delete or update a player's details of his team.

**1.3 Functionality**

- The team manager has to register himself and his team initially.
- The team manager has to login after the registration using his username and password to manage his team.
- The guest users can view the tournament details without signing up.
- The admin has all the control on each data item present in the database.

**1.4 Database Management System**

DBMS is a collection of programs 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.

A Relational database is a database that has a collection of tables of data items, all of which is formally described and organized according to the relational model. Data in a single table represents a relation, from which the name of the database type comes. In typical solutions, tables may have additionally defined relationships with each other. In the relational model, each table schema must identify a column or group of columns, called the primary key, to uniquely identify each row. A relationship can then be established between each row in the table and a row in another table by creating a foreign key, a column or group of columns in one table that points to the primary key of another table.

### 1.4.1 Characteristics of Database Management Systems

- Self-describing nature.
- Keeps a tight control on data redundancy.
- Enforces user defined rules to ensure that integrity of table data.
- Provides insulation between Programs and data, Data abstraction.
- Supports multiple views of the data.
- Helps sharing of data and Multi-user transaction processing.

### 1.4.2 Advantages of DBMS

- Controlling the redundancy.
- Restricting unauthorized access.
- Providing persistent storage for program objects.
- Providing storage structures for efficient query processing.
- Providing multiple users interfaces
- Representing complex relationships among data.
- Enforcing integrity constraints.

## 1.5 MYSQL

MySQL is an Oracle-backed opensource relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is an important component of an opensource enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server. MySQL has the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.)

The ANSI standard SQL provides basic functions for data manipulation, transaction control, and record retrieval from the database. However, most end users interact with Oracle through applications that provide an interface that hides the underlying SQL and its complexity.



Originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010. Developers can use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle.

Today, MySQL is the RDBMS behind many of the top websites in the world and countless corporate and consumer-facing web-based applications, including Facebook, Twitter and YouTube.

### **1. CREATE**

This command is used to create a table or view by giving it a name and specifying its attributes and constraints. The attributes are specified first, and each attribute is given a name, a data type to specify its domain values, and any attribute constraints such as NOT NULL.

Syntax: `CREATE TABLE <TNAME>(ATR1 TYP1 CONST1, ATR2 TYP2 CONST2,...)`

### **2. ALTER**

The definition of a base table can be altered by ALTER command which is a Schema Evolution command. The possible ALTER TABLE includes adding or dropping a column (attribute), changing a column definition, and adding or dropping table constraints.

Example: `ALTER TABLE STUDENT ADD NAME VARCHAR (12)`

### **3. DROP**

If a whole schema is not needed any more, the DROP SCHEMA command can be used. There are two drop behaviour options: CASCADE and RESTRICT.

CASCADE option is used to remove the database schema and all its tables, domains and other elements.

If the RESTRICT option is chosen in place of CASCADE, the schema is dropped only if it has no elements in it; otherwise, the DROP command will not be executed.

Syntax: `DROP TABLE STUDENT CASCADE`

### **1.5.1 Statements in SQL:**

Following are the important statements used in SQL.

1. SELECT - Used to retrieve the information from the relation.
2. INSERT - Used to insert the new values to the relation.
3. DELETE - Used to delete one or more existing tuples from the relation.
4. UPDATE - Used to update already existing values in the relation.

### **1.5.2 Aggregate Functions in SQL:**

Following aggregate functions are provided by the SQL.

- 1) COUNT - Returns number of tuples.
- 2) SUM - Returns sum of entries in a column.
- 3) MAX - Returns Maximum value from an entire column.
- 4) MIN - Returns Minimum value from an entire column.
- 5) AVG - Returns Average of all the entries in a column.

### **1.5.3 Constraints in SQL:**

Following constraints are provided by the SQL.

- 1) NOT NULL - Column should contain some value.
- 2) PRIMARY KEY - Should not allow duplicate and null values to a column.
- 3) UNIQUE - Each value of a column should be unique.

## **CHAPTER 2**

# LITERATURE REVIEW

This chapter focuses on the already existing systems for the management of Cricket tournaments and establish the software requirements for the project.

## 2.1 Survey of existing system

Cricket tournament management is a crucial aspect of organizing and hosting successful cricket events. A well-designed tournament management system can help streamline the process, providing valuable insights and data for teams, players, and fans. In recent years, there has been a growing interest in the development of computerized systems for cricket tournament management.

In the past, cricket tournament management was primarily done through manual data handling. This often involved using Excel spreadsheets or paper-based systems to keep track of teams, players, matches, and results. While these methods were effective to a certain extent, they were often prone to errors and were not efficient for managing large-scale tournaments.

### Disadvantages

- More human power.
- Repetition of same procedure.
- Low security.
- Data redundancy.
- Difficulty to handle.
- Difficulty to update data.
- Record keeping is difficult.

### Developed System

In this project, we are using Django web framework to develop the back-end of the Cricket tournament management system and MySQL as the database management system. Django is a high-level web framework that enables rapid development of secure and maintainable web applications. It follows the model-view-controller (MVC) architectural pattern, which makes it easy to manage complex and large-scale

web applications. MySQL is a widely used, open-source relational database management system that is known for its reliability, performance, and ease of use.

The developed system can effectively manage and organize cricket tournaments, make easy the process of creating schedules, tracking results, and generating statistics and provide an easy-to-use interface for tournament organizers and administrators. This will eliminate the need for manual data handling and improve the overall efficiency and accuracy of the tournament management process.

## 2.2 SOFTWARE REQUIREMENT

**Operating System:** Windows/Linux/macOS

**Frontend:** HTML, CSS, Bootstrap

**Backend:** Django web-framework (Python), MySQL (database management system)

**Web-Server:** Django's default server

### 2.2.1 Frontend:

The front end is an interface between the user and the back end. The front and back ends may be distributed amongst one or more systems. In network computing, *front end* can refer to any hardware that optimizes or protects network traffic. It is called application front-end hardware because it is placed on the network's outward-facing front end or boundary. Network traffic passes through the front-end hardware before entering the network. In compilers, the front end translates a computer programming source code into an intermediate representation, and the back end works with the intermediate representation to produce code in a computer output language.

#### 1. HTML

**HTML** or **Hypertext Mark-up Language** is the standard mark-up language used to create web pages. HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example <img>. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*).

## 2. CSS

**Cascading Style Sheets (CSS)** is a style sheet language used for describing the look and formatting of a document written in a mark-up language. While most often used to style web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages' use CSS style sheets to describe their presentation.

## 3. Bootstrap

**Bootstrap** is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

### 2.2.2 Backend:

#### 1. Django web-framework

Django is a free and open-source, Python-based web framework that follows the model–template–views architectural pattern. It is maintained by the Django Software Foundation, an independent organization established in the US as a 501 non-profit.

Despite having its own nomenclature, such as naming the callable objects generating the HTTP responses "views", the core Django framework can be seen as an MVC architecture. It consists of an object-relational mapper (ORM) that mediates between data models (defined as Python classes) and a relational database ("Model"), a system for processing HTTP requests with a web templating system ("View"), and a regular-expression-based URL dispatcher ("Controller").

Also included in the core framework are:

- a lightweight and standalone web server for development and testing
- a form serialization and validation system that can translate between HTML forms and values suitable for storage in the database
- a template system that utilizes the concept of inheritance borrowed from object-oriented programming
- a caching framework that can use any of several cache methods

- support for middleware classes that can intervene at various stages of request processing and carry out custom functions
- an internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signals
- an internationalization system, including translations of Django's own components into a variety of languages
- a serialization system that can produce and read XML and/or JSON representations of Django model instances
- a system for extending the capabilities of the template engine
- an interface to Python's built-in unit test framework

## **2. MYSQL**

MySQL is an Oracle-backed opensource relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

- MySQL is released under an open-source license.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

## **CHAPTER 3**

## PROPOSED WORK

This chapter will discuss the proposed work for the cricket tournament management system.

### 3.1 Entity Relationship (ER) model:

An entity-relationship diagram (ERD) is a data modelling technique that graphically illustrates an information system's entities and the relationships between those entities.

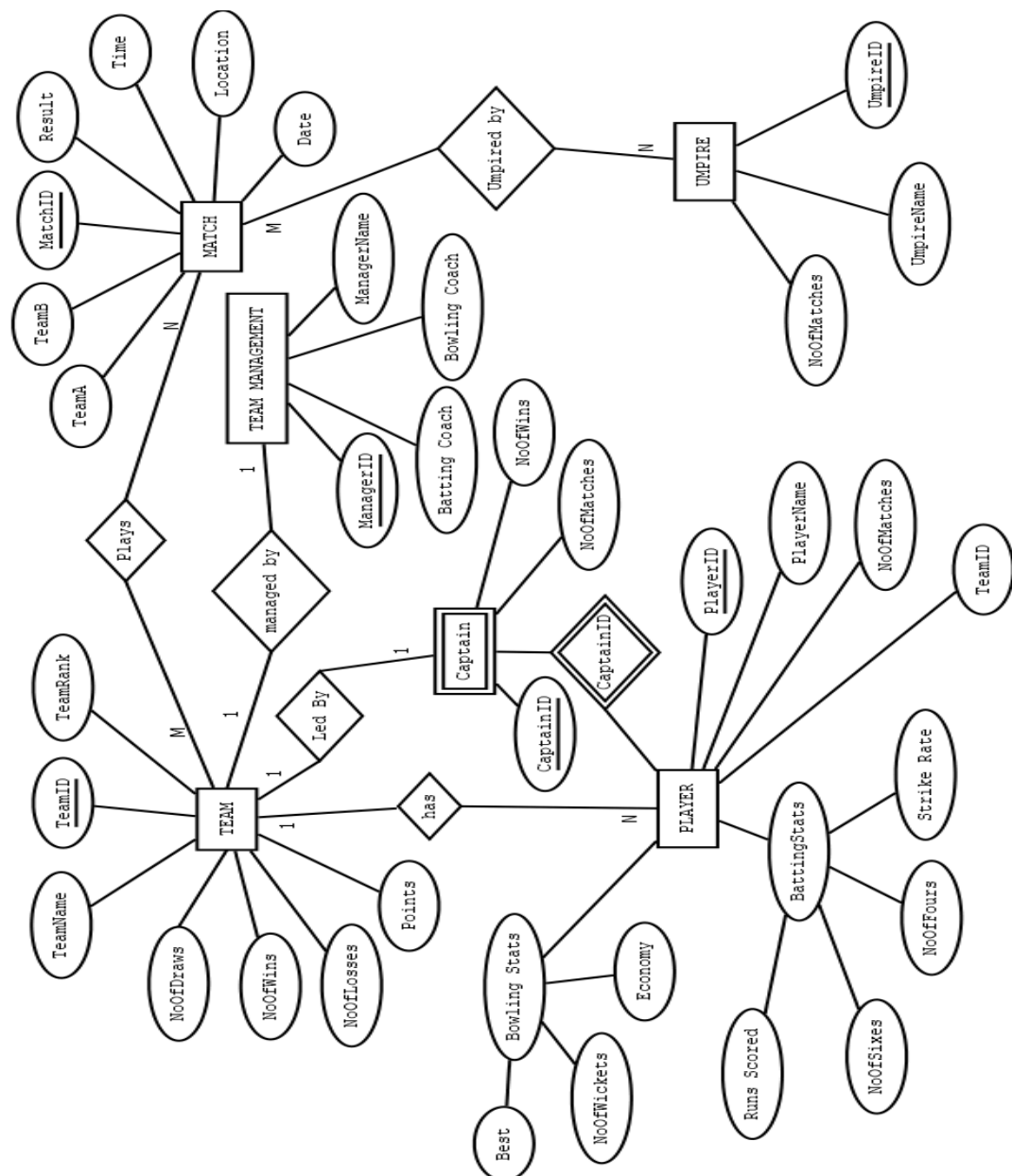


Figure 3.1: ER diagram

## Entities and their attributes:

In this project, we will be using several entities to organize and manage the cricket tournament. These entities include teams, players, matches, umpires, captains, and team management. Each of these entities plays a crucial role in the tournament management process.

**Team** is the primary entity in the tournament, representing the various participating teams. This entity will allow for the input and tracking of team information. The fields included are team-ID, team name, team rank, number of wins, number of losses, number of draws and points.

**Player** is another important entity, representing the individual players on each team. The system will allow for the input and tracking of player information, such as player name, number of matches the player has played, runs he has scored, batting strike rate, number of sixes, number of fours, bowling best, number of wickets he has taken and bowling economy.

**Matches** are the entity that represent the games that are played during the tournament. The system will allow for the input and tracking of match information, such as status of the match, date, time, location, and results.

**Umpires** is an entity that keeps track of the tournament's umpires. It allows to track umpire information and the matches that are being umpired by the umpire. The fields defined are umpire-ID, umpire's name and the number of matches that he has umpired till date.

**Captains** is another important entity in the tournament, representing the leaders of the teams. The system will allow for the input and tracking of Captains' information, such as number of matches as a captain and number of wins as a captain in the tournament.



**Team management** is an entity that represents the officials or managers of the teams. The system will allow for the input and tracking of team management information, such as manager name, batting coach and bowling coach.

All these entities will be stored in the MySQL database and will be connected to each other through relationships. This allows for easy retrieval and analysis of information, as well as the ability to generate schedules, standings, and statistics. By effectively managing these entities, the Cricket Tournament Management System will be able to streamline the tournament management process and provide valuable insights and data for teams, players, and fans.

### 3.2 Relational model:

The following is the relational model for the proposed management system:

#### TEAM

<u>TeamID</u>	TeamName	TeamRank	NoOfDraws	NoOfWins	NoOfLosses
Points					

#### PLAYER

<u>PlayerID</u>	PlayerName	NoOfMatches	TeamID	RunsScored	NoOfSixes
StrikeRate	NoOfWickets	Economy	Best	NoOfFours	

#### MATCH

<u>MatchID</u>	TeamA	TeamB	Result	Time	Location	Date
----------------	-------	-------	--------	------	----------	------

#### UMPIRE

<u>UmpireID</u>	UmpireName	NoOfMatches
-----------------	------------	-------------

#### TEAM MANAGEMENT

<u>ManagerID</u>	ManagerName	BattingCoach	BowlingCoach
------------------	-------------	--------------	--------------

**CAPTAIN**

<u>CaptainID</u>	NoOfMatches	NoOfWins
------------------	-------------	----------

**PLAYS**

<u>TeamID</u>	<u>MatchID</u>
---------------	----------------

**UMPIRED BY**

<u>MatchID</u>	<u>UmpireID</u>
----------------	-----------------

**3. 3 Normalization**

The following is a normalization report for the Cricket Tournament Management System project:

**1st Normal Form (1NF):** All the entities in the system such as team, player, matches, umpires, captains, and team management are in 1NF, as each record has a unique primary key and each field contains atomic (indivisible) values.

**2nd Normal Form (2NF):** All the entities in the system are in 2NF, as all non-primary key fields are functionally dependent on the primary key. This means that each field in the table is dependent on the primary key and not on any other non-primary key field.

**3rd Normal Form (3NF):** All the entities in the system are in 3NF, as there are no transitive dependencies between non-primary key fields. This means that all non-primary key fields are directly dependent on the primary key, and there are no dependencies between non-primary key fields.

For example, the team entity has a primary key of TeamID, and the team name, number of wins and the previous records fields are dependent on the primary key. Similarly, the player entity has a primary key of PlayerID, and the name and the statistics fields are dependent on the primary key.

Overall, the database for this project is in 3rd Normal Form, which ensures that the data is organized in a manner that minimizes data redundancy and improves data integrity.

## CHAPTER 4

### RESULT

The Cricket Tournament Management System was implemented and tested using sample data. The results of the implementation show that the system effectively manages and organizes cricket tournaments, automates the process of creating schedules, tracking results and generating statistics, and provides an easy-to-use interface for tournament organizers and administrators.

The system's user interface was evaluated by a group of test users, and they found it to be user-friendly and easy to navigate. The system's ability to input and track teams, players, matches, umpires, captains, and team management was found to be effective, allowing for easy retrieval and analysis of information. The system's ability to generate schedules, standings, and statistics was also found to be effective, providing valuable insights and data for teams, players, and fans. The system's database was found to be well-designed and normalized, providing good data integrity and consistency.

Overall, the Cricket Tournament Management System was found to be a powerful tool that streamlines the tournament management process and provides valuable insights and data for teams, players, and fans. The system is expected to significantly improve the efficiency and accuracy of cricket tournament management and provide a better experience for all parties involved.

#### 4.1 Screenshots

The following is a series of screenshots of the developed application.

- **Figure 5.1:** Home Screen
- **Figure 5.2:** Teams page
- **Figure 5.3:** Admin pane

- **Figure 5.4:** Points table page
- **Figure 5.5:** Matches page
- **Figure 5.6:** Manager dashboard

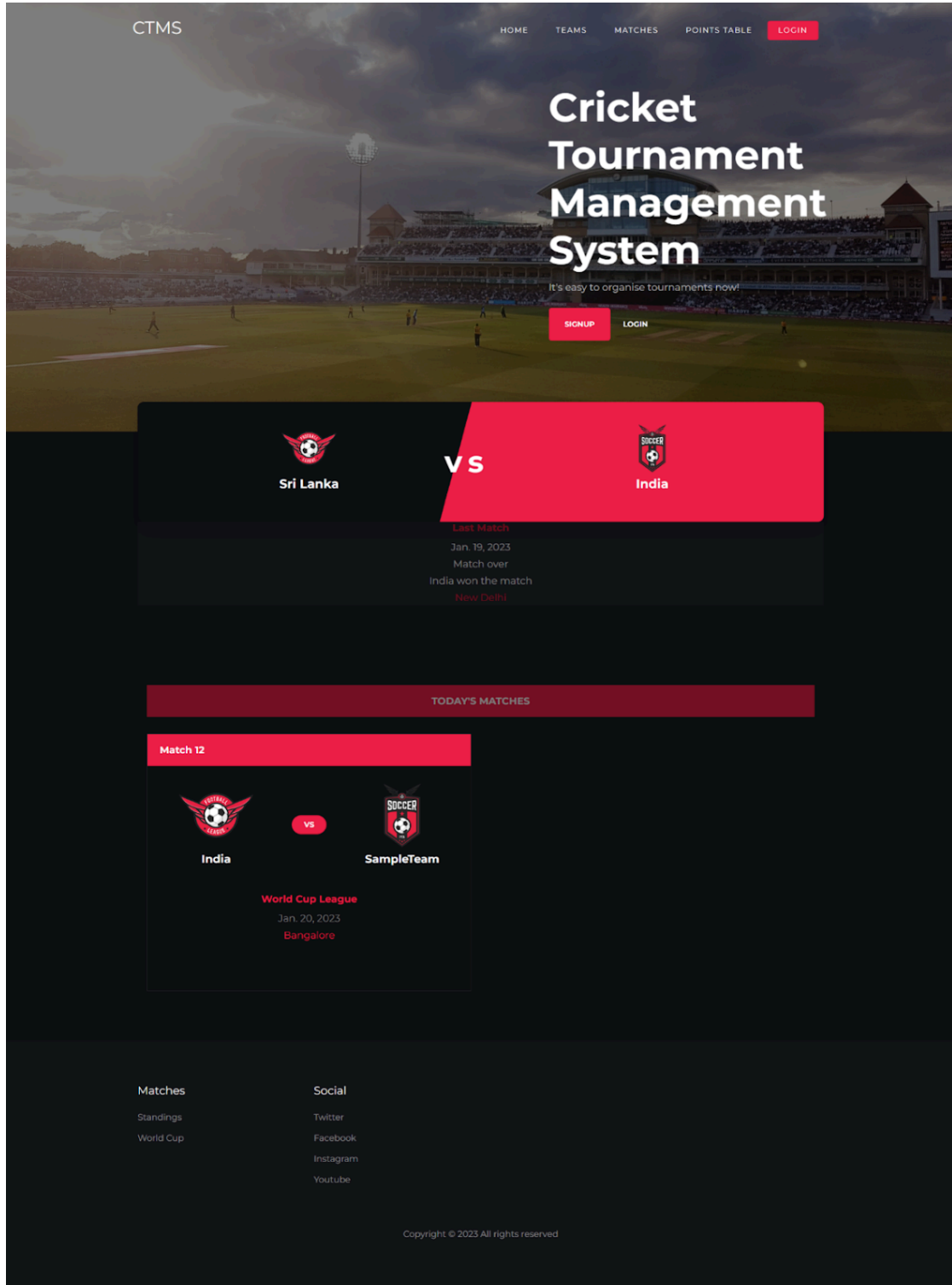


Figure 5.1: Home Screen

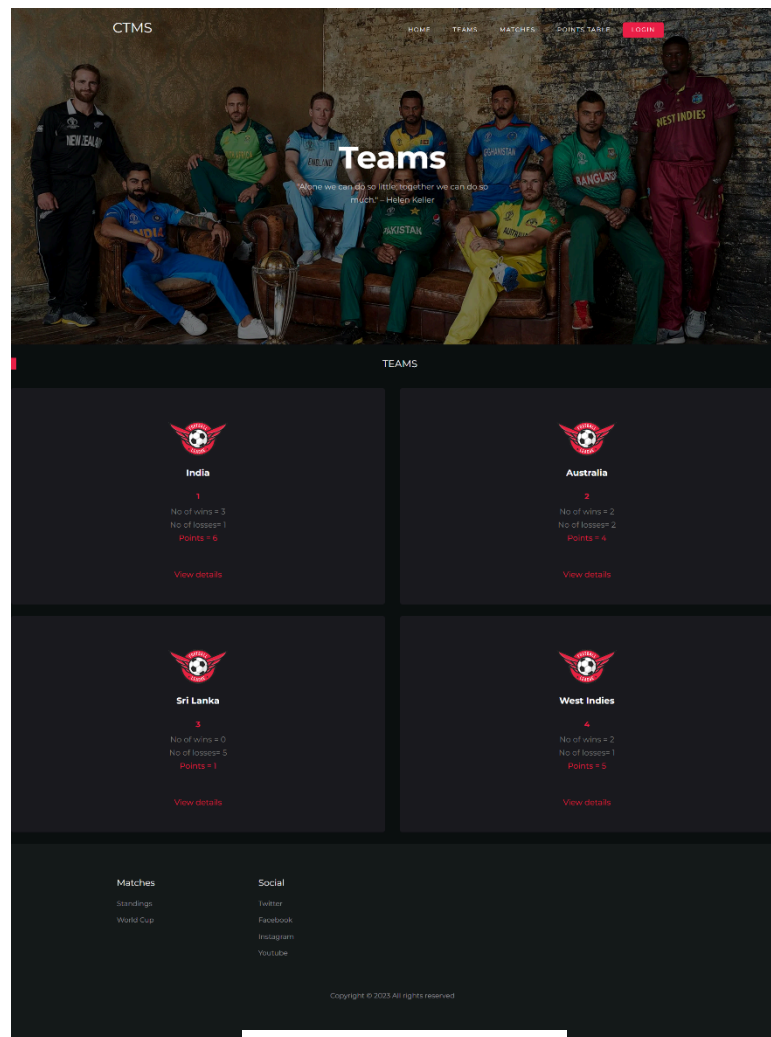


Figure 5.2: Teams Page

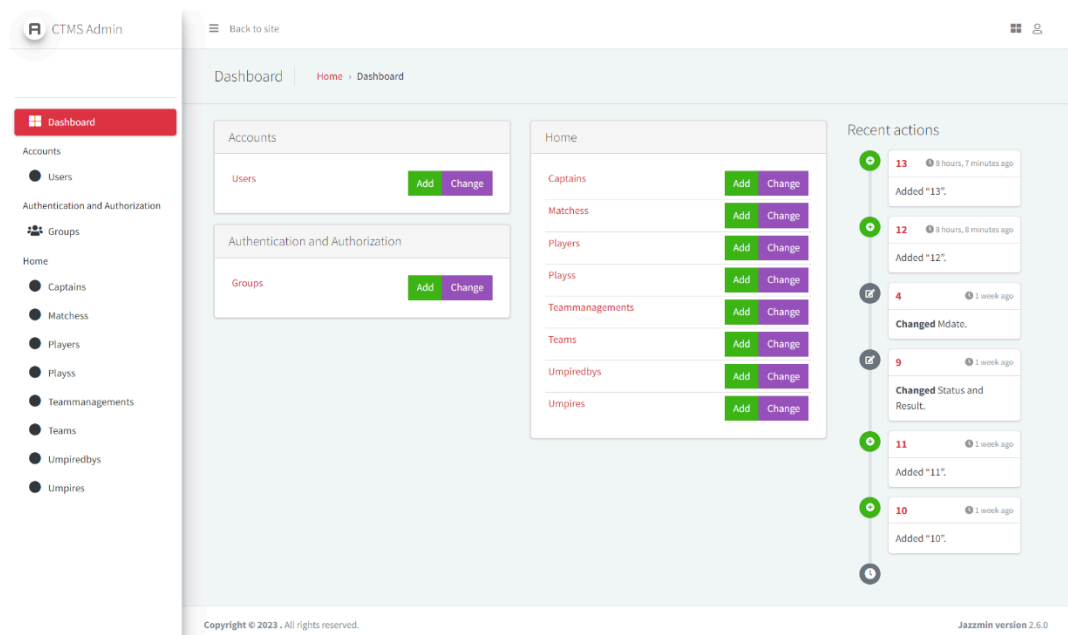


Figure 5.3: Admin Panel

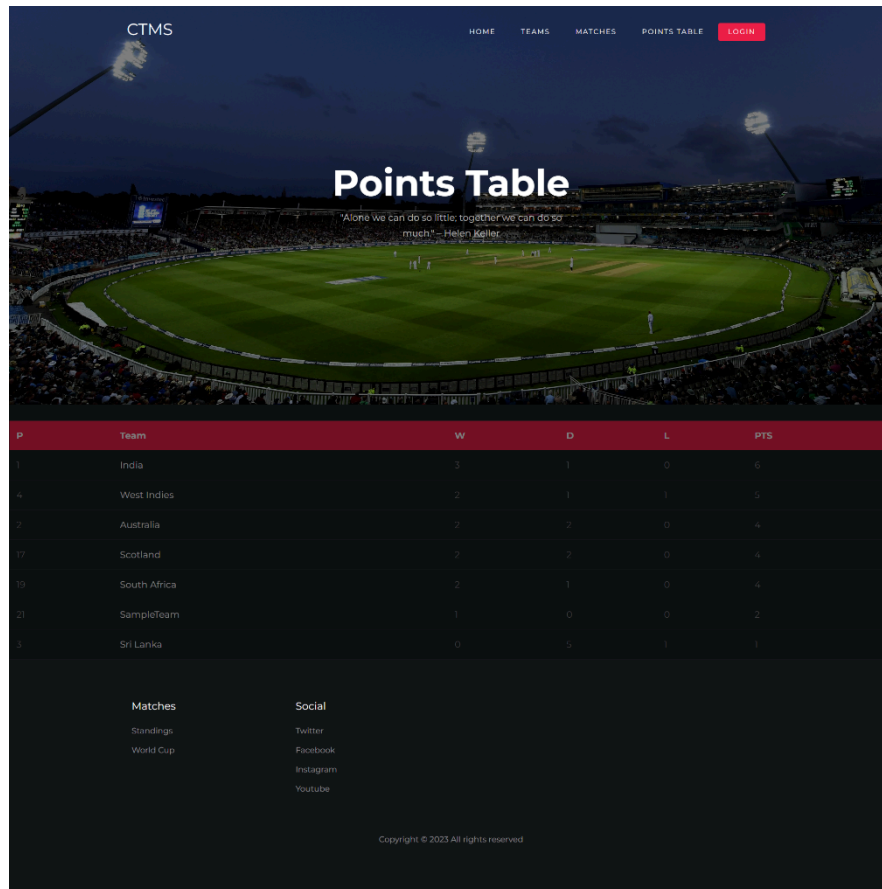


Figure 5.4: Points table

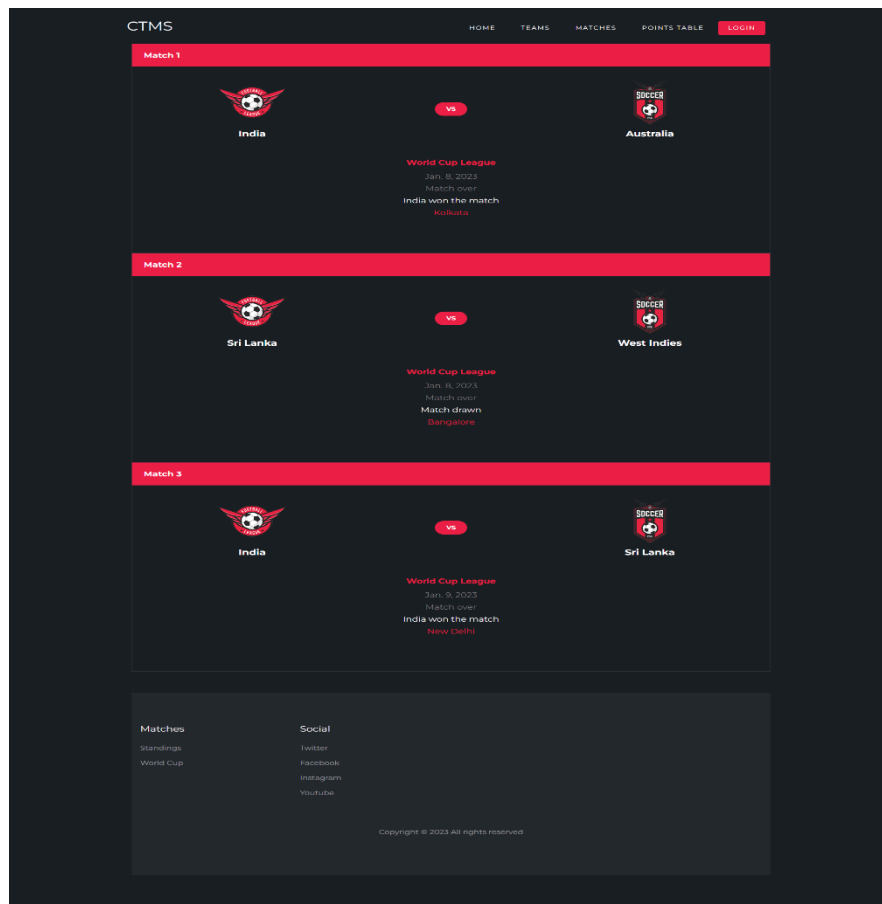


Figure 5.5: Matches Page

CTMS

HOMEPDASHBOARDTEAMSMATCHESPOINTS TABLELOGOUT

TEAM INDIA

Add player

Playername:

Noofmatches:

Teamid:

Runsscore:

Noofsixes:

Strikerate:

Noofwickets:


Economy:

Bestbowling:

Nooffours:

SUBMIT

Players



Ravi Shastri

India

No of matches = 150

Batting statistics

Runs scored = 3108

Strike rate = 61.07

No of fours = 206

No of sixes = 25


Bowling statistics

No of wickets = 129

Economy = 4.22

Best bowling = 5/75

UPDATEDELETE



Murali Vijay

India

No of matches = 17

Batting statistics

Runs scored = 339

Strike rate = 67.0

No of fours = 33

No of sixes = 3


Bowling statistics

No of wickets = 1

Economy = 6.17

Best bowling = 1/19

UPDATEDELETE



Gautam Gambhir

India

No of matches = 147

Batting statistics

Runs scored = 5238

Strike rate = 85.25

No of fours = 561

No of sixes = 17


Bowling statistics

No of wickets = 0

Economy = 13.0

Best bowling = 0/13

UPDATEDELETE



Mohinder Amarnath

India

No of matches = 85

Batting statistics

Runs scored = 1924

Strike rate = 57.71

No of fours = 123

No of sixes = 9


Bowling statistics

No of wickets = 46

Economy = 4.38

Best bowling = 3/12

UPDATEDELETE



MS Dhoni

India

No of matches = 350

Batting statistics

Runs scored = 10773

Strike rate = 87.56

No of fours = 826

No of sixes = 229


Bowling statistics

No of wickets = 1

Economy = 5.17

Best bowling = 1/4

UPDATEDELETE



Kapil Dev

India

No of matches = 225

Batting statistics

Runs scored = 3783

Strike rate = 95.07

No of fours = 291

No of sixes = 67

Bowling statistics

No of wickets = 263

Economy = 3.72

Best bowling = 5/43

UPDATEDELETE

Matches

Standings

World Cup

Social

Twitter

Facebook

Instagram

Youtube

Copyright © 2023 All rights reserved

Figure 5.6: Manager Dashboard

Department of CSE, UVCE June 2024



**Figure 5.1** is the home page screen that appears to every user where the user can navigate to his/her desired page. It also contains the last match that was played and today's matches.

**Figure 5.2** is the teams page that shows details about the teams participating in the tournament and click on the 'View details' button to check statistics of the players of a team.

**Figure 5.3** is the 'Admin panel' that is exclusively provided to the admin. This is a easy to use interface for the admin to control the entire tournament.

**Figure 5.4** is the 'Points table' page that shows the points and standings of each team involved in the tournament.

**Figure 5.5** is the 'Matches' page which displays the matches that are scheduled for the tournament and the details of the same.

**Figure 5.6** is the 'Manager Dashboard' page through which the team manager can easily manage his team.

## CONCLUSION

The project aimed to develop a Cricket Tournament Management System that can effectively manage and organize cricket tournaments. The system was built using a relational database management system (MySQL) and a user-friendly interface (Django framework) making it easy to use for tournament organizers and administrators. The system allows for the input and tracking of teams, players, matches, umpires, captains and team management, generating schedules, standings, and statistics.

The entities in the system were designed to be in 3rd Normal Form, which ensured that the data was organized in a manner that minimized data redundancy and improved data integrity. The inclusion of signals through Django ensured the implementation of a stored procedure that updated team's statistics in the tournament.

Overall, the Cricket Tournament Management System is a powerful tool that streamlines the tournament management process and provides valuable insights and data for teams, players, and fans. It is expected that this system will significantly improve the efficiency and accuracy of cricket tournament management and provide a better experience for all parties involved.

## BIBLIOGRAPHY

- [1] Ramez Elmasri and Shamkant B Navathe, *Fundamental of Database Systems*, 6<sup>th</sup> ed. Addison Wesley, 2009
- [2] William S Vincent, *Django for Beginners*, 4<sup>th</sup> ed. Lean Publishing, 2022
- [3] William S Vincent, *Django for Professionals*, 4<sup>th</sup> ed. Lean Publishing, 2022
- [4] <https://docs.djangoproject.com>