A Mini-Project Report on

CRICKET TOURNAMENT

Submitted by

AMULYA K M(19GACSE002)

ANUJA NAMRATHA (19GACSE004)

5th Semester, CSE

Under the guidance of

Dr. Champa H N
Professor
Dept. of CSE
UVCE

Mrs. Suhasini B
Guest Faculty
Dept. of CSE
UVCE

Department of Computer Science and Engineering

UNIVERSITY VISVESVARAYA COLLEGE OF ENGINEERING
K.R. Circle, Bengaluru- 560001

MARCH 2021-2022

Bangalore University

UNIVERSITY VISVESVARAYA COLLEGE OF ENGINEERING

K.R. Circle, Bengaluru- 560001

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that **ANUJA NAMRATHA (19GACSE004)** has successfully completed the Mini- Project work entitled **CRICKET TOURNAMENT,** in partial fulfillment for the requirement of the **Database Management Systems Laboratory** of 5th semester Computer Science and Engineering during the academic year 2021- 2022.

Dr. Champa H N	**Mrs. Suhasini B**	**Dr. Dilip Kumar S M**
Professor	**Guest Faculty**	**Professor & Chairperson**
Dept. of CSE	**Dept. of CSE**	**Dept. of CSE**
UVCE	**UVCE**	**UVCE**

Examiners:

1. …… **2. ……**

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task is incomplete without the mention of people who made it possible, whose constant guidance and encouragement crown all efforts and success.

I am grateful to acknowledge the hon'ble Vice Chancellor **Dr. Venugopal K R,** Bangalore University, for his valuable suggestions and relentless support, which has sustained me throughout the course of the project.

I express my gratefulness to **Dr. H N Ramesh**, Principal, UVCE who has been leading our college with a brighter vision in technical education.

I express my profound gratitude to **Dr. Dilip Kumar S M,** Chairperson, Department of Computer Science and Engineering, UVCE who has been a constant source of inspiration to the students work.

With extreme honor I express my deep sense of gratitude to my guide **Dr. Champa H N**, Professor, Department of Computer Science and Engineering, UVCE for her valuable guidance and supervision in this course of project. We also thank **Mrs. Suhasini B,** Guest Faculty, Department of Computer Science and Engineering, UVCE for her suggestions and support.

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 provided me to successfully complete this project.

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

-ANUJA NAMRATHA

19GACSE004

ABSTRACT

The project represents the **'Cricket Tournament**'. It provides information of various cricket matches, in which all the major countries participate. It also provides us with information about the various players participating in the tournament and provide the analysis of their performance

The aim of the project is to provide the complete information of the International Cricket statistics. The information is available Country wise, and Player wise. This project is very use for Cricket match broadcasters to get information quickly. Actually, it is not easy to do this process manually because it would become very hectic. Hence it is recommended to automate the process by developing the relevant software as the world is moving from manual working to information and technology era where computerization becomes important in all walks of life.

The proposed system works on a client-server architecture. A desktop application built on NodeJS; framework communicates with an SQL based database through APIs written using Express JS. MySQL which is an open-source Relational Database Management System is used to manage the database. Fundamental database concepts such as Views, Triggers and Indices are used where applicable for efficiency in performance and modularity. Reusable components ensure data abstraction and resemble an Object-Oriented style of programming.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. INTRODUCTION TO CRICKET TOURNAMENT

The main aim of the project is to computerize the maintenance of a Cricket tournament. Cricket Tournament is a Web based application, which is developed to display the information of a cricket tournament which involves the information of players, team, coach, captain, umpires and it also includes the details of each and every match played during a specific interval of time. The main objective of the Project the project is to provide the complete information of the International Cricket statistics. The information is available Country wise, and Player wise. This project is very use for Cricket match broadcasters to get information quickly. The project is desired to have the following features:

- The users must be able to login and see the player details of each player in the database and team performance in each match.
- Administrator has all the privileges of the user but has the authority to add and remove data from the database like time and venue of the match which the user cannot do.
- The program must display the Score board which contains the rank of each team.
- The program should also display the name of the Umpire for the current match.
- The database contains details of **players, coaches** and **umpires** among others. All the useful information about the entire World Cup can be found here.

Data Requirements:

Entities:

1. **Team** is an entity type which has many attributes like Team Name which uses the data type varchar. Every team has been given a Team ID which is the primary key which is of data type varchar. Team Ranking, Number of Batsmen and Number of Bowlers are of the data type number. There is another attribute - Wicketkeeper which is of multivalued type and accepts varchar data type. Primary key cannot have null value.
2. **Players** is an entity type which has an attribute – Player Name which is of the data type varchar. It has a primary key, Player ID, which cannot have null value. It has a foreign key, Team ID which is the primary key of the entity, Team. There is a complex attribute, Number of matches played, which comprises of Number of Test Matches, Number of T20 Matches,

number of World Cup Matches and Number of ODIs.

3. **Batsman** is an entity type which has the attributes – Number of sixes hit, Number of Fours hit, the batting average, and the total runs scored. All of these attributes are of the data type number.
4. **Bowler** is an entity type which has the attribute – type of batsman with varchar data type. It also includes number of wickets and economy which are of the data type number.
5. **Umpire** is an entity type which has the attributes name and country of origin of data type varchar. The primary key of this is Umpire Id which is of varchar data type. It also has an attribute Number of matches of data type number.
6. **Coach** is an entity type with a foreign key, Team ID, which is a primary key of entity type, Team. It has a primary key, Coach ID, of data type varchar. It also has another attribute of data type varchar, Name.
7. **Captain i**s an entity type with a primary key, Captain ID of data type varchar. It has two foreign keys,Player id from table Players and Team ID from table Team. Number of years of captaincy and Number of wins are also attributes of this table of data type number.
8. **Matches** is an entity type with a primary key, match ID, of varchar data type. It has attributes like Team1 Name, Team2 Name, Stadium, Winner Team and Loser Team of data type varchar. Match date is an attribute which uses the datatype date. Match time is an attribute which is of the data type time.

Relationships:

Cricket player plays in team (N-1)

A cricket player can play in only one team but a team can have many players in it but a team must have players in it. So, the relationship becomes (N-1).

Coach manages team(1-N)

Coach can manage a single team, but each team can have many coaches (like batting coach, fielding coach, bowling coach). But it is compulsory for a team to have a coach. So, the relationship is 1-N.

Team plays match(M-N) Team can play many matches and a match can be played by two teams. So, the relationship is M-N.

Matches are umpired by Umpire(M-N)

An umpire can umpire in many matches and a match can have two umpires. So, the relationship is M-N.

Team headed by a Captain (1-1)

A team has 1 captain and a captain is from single team only. So the relationship is 1-1.

Functional Requirements:

1) **VIEWER**

System must allow users to login if they enter the correct login id and password. The users must be able to see the player details of each player in the database. Scores of each match must be visible. Match date and venue should be displayed on the login if the users seek for it. System should display the complete roster of a team including the captain and the players playing in the top 11 and the current rank of the team. The details of the coach must also be available to the users. Referees and their details are also important as the players and the viewers want to see the best referees managing their team's match. Each player's statistics should also be available like total runs, number of matches played etc. System should display data on each match which has been scored in the duration of the entire tournament. System should allow fixtures to be searched and the date should also be available.

BASIC ANALOGY:

- View the website with a browser.
- View all teams.
- View all players of a team.
- View all batsmen in the tournament.
- View all bowlers in the tournament.
- View statistics of a player (all time).
- View coach details.
- View umpire details.
- View Match details.
- View ranking of each team

View Player information per match:

a) Number of matches
b) Total runs

c) Total wickets

View all match details:

- Team 1
- Team 2
- Umpire
- Winner
- Date
- Time
- Stadium
- Rank of teams after match

2) ADMINISTRATOR

Administrator is in charge of creating the website which is used to access the database. Administrator has all the privileges of the user but has the authority to add and remove data from the database which the user cannot do. Administrator is responsible for creating different user accounts and assigning the id and password. Administrators are the one who generate the fixtures and update them in the database. They should be allowed to enter the team name of home and away teams. He should have the authority to enter and modify the match details like time and venue in case the need to be changed. If any player has been punished for bad behavior or other reasons and cannot play in the World Cup anymore the administrator should be able to delete the data from the database. The rank of every team must keep being modified after each match. After a team is eliminated or disqualified the administrator should be able to delete the entire team's record.

BASIC ANALOGY:

- Create website.
- Generate login ID for viewer.
- Design website.
- Display different menus.
- Create World Cup.
- Display Team Name.
- Display Team Captain.
- Display Team Squad.

View Player information per match:

- Number of matches
- Total runs
- Total wickets

REMOVAL OF OLD DATA:

- If any team gets disqualified, then their data needs to be removed from the database.
- If a player gets injured during the World Cup and is unable to play further, then their data needs to be removed from the database.
- If any match gets cancelled due to unforeseen circumstances, then the particular match details should be removed.

MODIFICATION OF DATA:

- After every match the existing ranks of every team should be modified
- After every match, the statistics of every player should be updated.
- Due to unfavorable weather conditions, a match might get delayed. Hence, the match timings need to be changed.

RETRIEVAL OF DATA:

View information of every Team:

Before the start of a new match, we have to retrieve the Team record like: a) Team Name b) Number of Batsman c) Number of Bowlers d) Wicketkeeper e) Number of Wins f) Number of Losses g) Names of Players

View information of every Match:

After every match, we have to retrieve the Match details like: a) First Team Name b) Second Team Name c) Umpire d) Winner e) Loser

View Score Board: After every match, we have to retrieve the ranking order of teams: a) Rank of each team b) Team name iv) View the captain of each team: During the toss, we need to retrieve the data of the captain a) Name of captain b) Number of wins under his captaincy c) Years of captaincy

1.2. 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.

3.1.1 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.

3.1.2 Advantages of using the DBMS approach

- Controlling the redundancy.
- Restricting unauthorized access.
- Providing persistent storage for program objects.

CHAPTER 2

LITERATURE REVIEW

2.1. STUDY OF EXISTING SYSTEM

- Needs a lot of working staff and extra attention on all the records.
- In existing system, there are various problems like keeping records of a particular players performance, a specific captain details, umpire details of a particular match etc.
- Finding out details regarding any information is very difficult, as the user has to go through all the bulky databases manually.
- Major problem was the lack of security.
- Client satisfaction is not up to the mark.

2.2. PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations.

It has got following features

- Ensure data accuracy.
- Records are efficiently maintained by DBMS.
- DBMS also provides security for the information.
- Any person across the world, having internet can access this service.
- Match status can be accessed very easily.
- User can update the details easily.
- Minimum time needed for the various processing.
- Better Service.
- Minimum time required.

2.3. SOFTWARE USED

A software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate. It should only specify the external

behavior of the system. For the successful, efficient and problem free designing any project or program, the system should meet some requirements.

2.3.1. Hardware Requirements

- Processor: Intel core Duo 2.0GHz or more.
- RAM: 1GB or more.
- Hard disk: 80GB or more
- Monitor: 15” CRT or LCD monitor
- Keyboard: Normal or Multimedia
- Mouse: Compatible mouse

2.3.2. Software Requirements:

- Operating System: Platform independent. Windows used.
- Database: MySQL.
- Front-End: HTML, CSS, JS, Bootstrap
- Back-end: NodeJS (A JavaScript based framework)

2.4. MYSQL

MySQL is an Oracle-backed open-source 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 open-source 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.)

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.

SQL uses the terms table, row, and column for relation, tuple, and attribute, respectively.

The SQL commands for data definition are CREATE, ALTER and DROP.

2.4.1 CREATE

This command is used to create 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 (ATR1 TYP1 CONST1, ATR2 TYP2 CONST,…)

2.4.2 ALTER

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

Example: ALTER TABLE STUDENT ADD NAME VARCHAR (12)

2.4.3 DROP

If a whole schema is not needed any more, the DROP SCHEMA command can be used. There are two drip behavior 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 it has no elements in it; otherwise, the DROP command will not be executed.

SYNTAX: DROP TABLE STUDENT CASCADE

2.4.4 Statements in SQL

Following are the important statements used in SQL.

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

2.4.5 Aggregate Functions in SQL

Following aggregate functions are provided by the SQL.

- **COUNT** - Returns number of tuples.
- **SUM** - Returns sum of entries in a column.
- **MAX** - Returns Maximum value from an entire column.
- **MIN** - Returns Minimum value from an entire column.
- **AVG** - Returns Average of all the entries in a column.

2.4.6. Constraints in SQL

Following constraints are provided by the SQL.

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

2.5. NodeJS

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the Chrome V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web - application development around a single programming language, rather than different languages for server-side and client-side scripts.

Though .js is the standard filename extension for JavaScript code, the name "Node.js" doesn't refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

2.5.1 Features of Node.js

- **Asynchronous and Event Driven** − All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
- **Very Fast** − Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- **Single Threaded but Highly Scalable** − Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a singlethreaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
- **No Buffering** − Node.js applications never buffer any data. These applications simply output the data in chunks.
- **License** − Node.js is released under the MIT license

2.5.2 Where to Use Node.js?

Following are the areas where Node.js is proving itself as a perfect technology partner.

- I/O bound Applications
- Data Streaming Applications
- Data Intensive Real-time Applications (DIRT)
- JSON APIs based Applications
- Single Page Applications

2.5.3 NPM

These are libraries built by the awesome community which will solve most of your generic problems. npm (Node package manager) has packages you can use in your apps to make your development faster and efficient.

2.5.4 REQUIRE

Require does three things:

- It loads modules that come bundled with Node.js like file system and HTTP from the Node.js API.
- It loads third-party libraries like Express and Mongoose that you install from npm.
- It lets you require your own files and modularize the project.
- Require is a function, and it accepts a parameter “path” and returns module.exports.

2.5.5 NODE MODULES

A Node module is a reusable block of code whose existence does not accidentally impact other code. one can write your own modules and use it in various application. Node.js has a set of built-in modules which you can use without any further installation.

2.5.6. Express.js (APIs that connect to database)

- Express.js or simply Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. Express is a back-end web application framework for Node.js released as a free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js. Express is relatively minimal with many features available as plugins. Express is the back-end component of the MEAN stack together with the MongoDB database software and Angular JS front-end framework. Express provides mechanisms to:
- Write handlers for requests with different HTTP verbs at different URL paths (routes).
- Integrate with "view" rendering engines in order to generate responses by inserting data into templates.
- Add additional request processing "middleware" at any point within the request handling pipeline.

A web application traditionally waits for HTTP requests from the web browser (or other client). When a request is received the application works out what action is needed based on the URL pattern and possibly associated information contained in POST data or GET data.

Depending on what is required it may then read or write information from a database or perform other tasks required to satisfy the request. The application will then return a response to the web browser, often dynamically creating an HTML page for the browser to display by inserting the retrieved data into placeholders in an HTML template.

Some merits of Express.js include:

- It makes Node.js web application development fast and easy.
- It is easy to configure and customize.
- It allows the definition of routes for the application based on HTTP methods and URLs.
- It allows the definition of error handling middleware.
- It makes it easy to serve static files and resources of the application.
- It allows the creation of a REST API server. It is easy to connect to databases such as MongoDB, Redis and MySQL.

CHAPTER 3

PROPOSED WORK

3.1. ENTITY RELATIONSHIP DIAGRAM:

An Entity relationship diagram shows the relationships of entity sets stored in databases. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

An entity relationship diagram is a snapshot of data structure. An entity relationship diagram shows entities (tables) in a database and relationships between tables within that database. For a good database design it is essential to have an entity relationship diagram.

There are three basic elements in entity relationship diagram

- Entities are the things for which we want to store information. An entity is a person, place, thing or event.
- Attributes are the data we want to collect for an entity.
- Relationships describe the relations between the entities.

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An entity-relationship diagram is a conceptual and representational model of data used to represent the entity framework infrastructure.

The elements of an entity-relationship diagram are:

- Entities
- Relationships
- Attributes

Steps involved in creating an entity-relationship diagram include:

- Identifying and defining the entities
- Determining all interactions between the entities
- Analyzing the nature of interactions/determining the cardinality of the relationship
- Creating the entity-relationship diagram

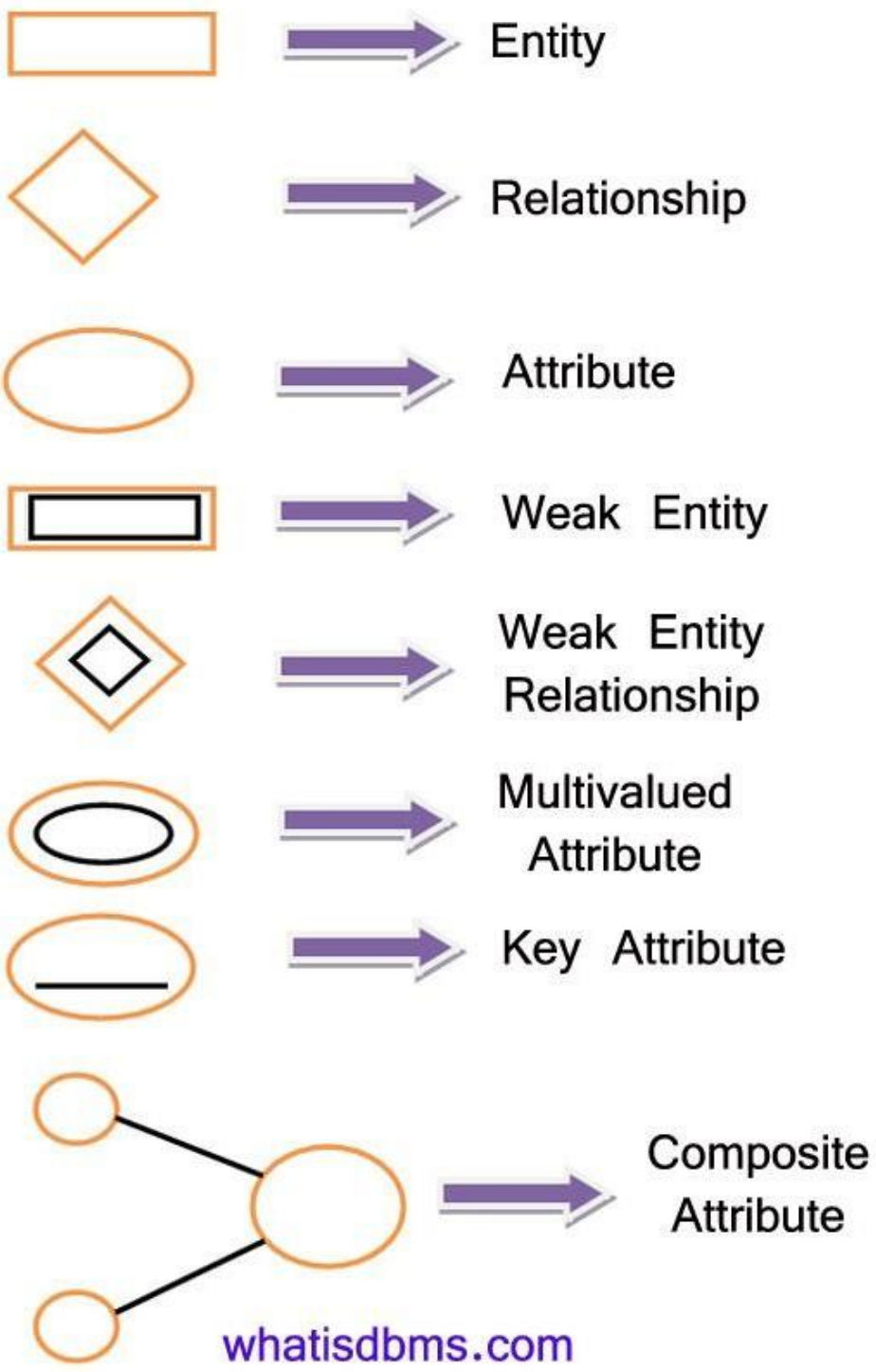

Figure 5.1 Notations for ER Diagram

ER DIAGRAM:

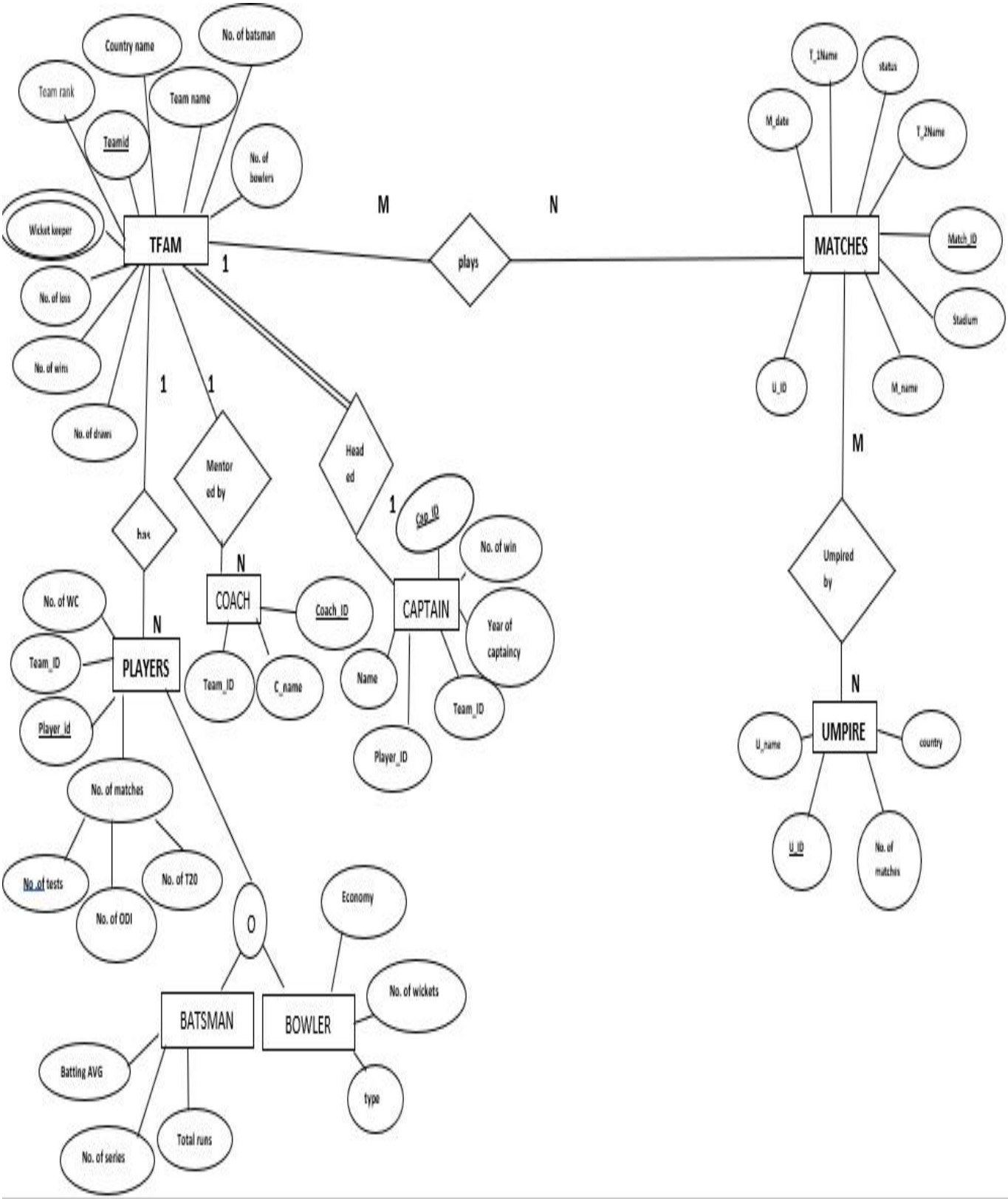

Fig 5.2.ER diagram

Fig 5.2 shows the ER diagram for the Cricket Tournament. The entities are TEAM, PLAYERS, MATCHES, COACH, CAPTAIN AND UMPIRE.

The attributes of TEAM ARE Team_id, no_of_wins, no_of_loss, no_of_draws, team_rank, no_of_batsman, no_of_bowlers, country_name, team_name.where team_id serves as a primary key.

The attributes of MATCHES are T1_name,T2_name,match_id,status,stadium,m_date,m_time.where match_id serves as a primary key.

The attributes of COACH are c_name,coach_id,team_id.Where coach_id serves as a primary key and the team_id is the foreign key from the TEAM to COACH.

The attributes of CAPTAIN are cap_id,name,team_id,no_of_wins,year_of _captaincy.Where cap_id is the primary key and the team_id is the foreign key from the TEAM.

The attributes of UMPIRE are u_name,u_id,no_of_matches,country.Where u_id is the primary key.

The attributes of PLAYERS are player_id,no_of_ODI,no_of_T20,no_of_test,team_id.The PLAYERS entity also has an overlapping inheritance where the BATSMAN and BOWLERS inherit from the PLAYERS through player_id.

BATSMAN has the attributes batting_avg,no_of_series,total_runs.

BOWLER has the attributes no_of_wickets,type,Economy.

3.2. RELATIONAL DATABASE SCHEMA

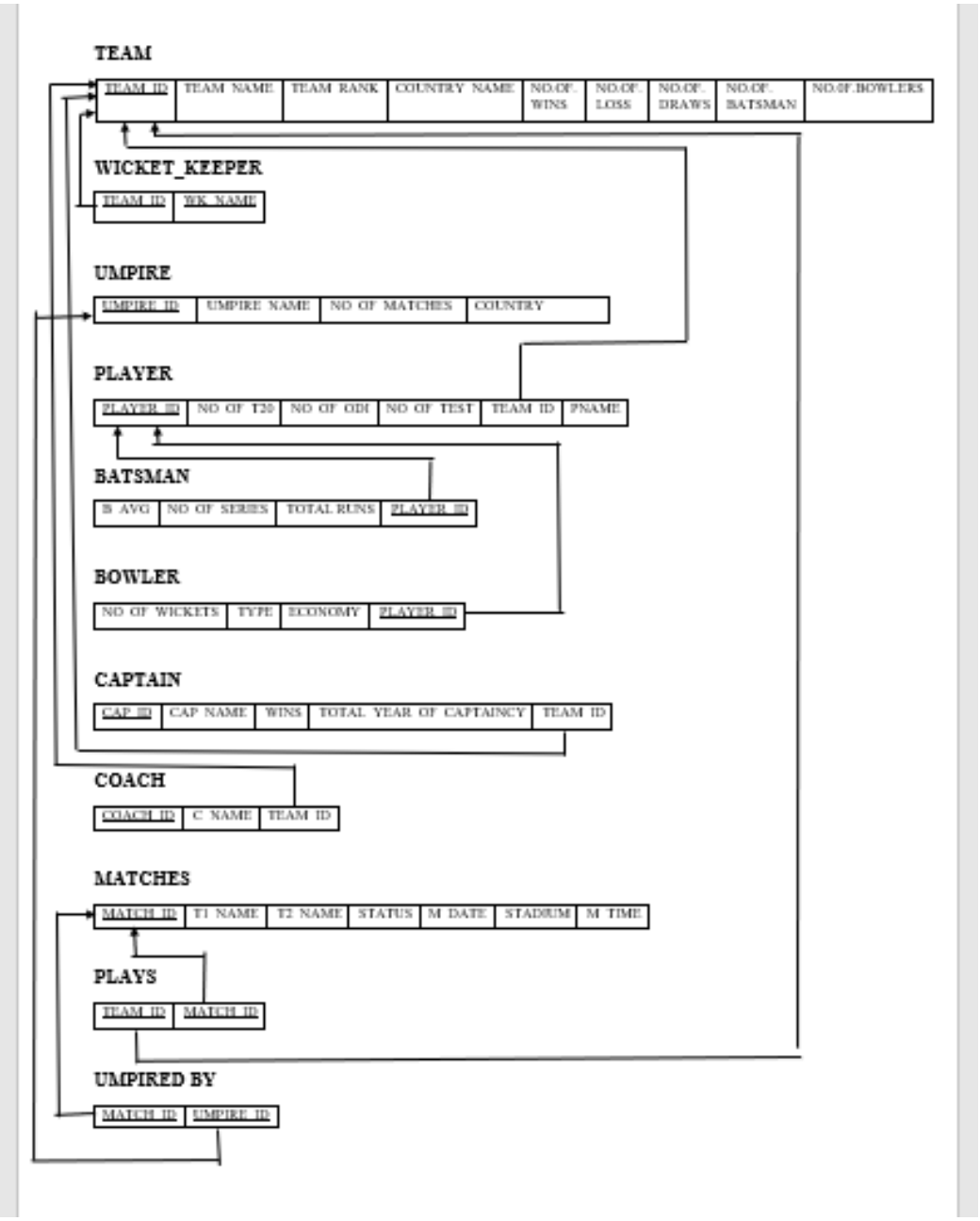

Fig.5.3. relational database schema.

Mapping of 1:1 Relationship type:

For each 1:1 relationship type R can be migrated to any participating entity types. This relationship type ensures that each user in the database can lodge one complaint. There is 1:1 relationship between TEAM to CAPTAIN.

Mapping of 1:N Relationship type:

For each regular binary 1:N relationship type R, identify the relation S that represents the participating entity type at the N-side of the relationship type. Include as the foreign key in S the primary key of relation T that represents other entity type in R. We have to do this because each entity instance on N side is related to at most one entity instance on 1-side of relationship type There is 1:N relationship from TEAM to PLAYERS and TEAM to COACH .

Mapping of M:N Relationship type:

For each M:N relationship type R, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity type, their combination will form the primary key of S. Also include any simple attributes of m: n relationship type.

3.3. NORMALISATION:

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.

Normalization is used for mainly two purposes,

- Eliminating redundant(useless) data.
- Ensuring data dependencies make sense i.e data is logically stored.

1. **First Normal Form** – If a relation contains composite or multi-valued attribute, it violates first normal form or a relation is in first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if every attribute in that relation is singled valued attribute.
2. **Second Normal Form** – To be in second normal form, a relation must be in first normal

form and relation must not contain any partial dependency. A relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table. Partial Dependency – If the proper subset of candidate key determines non-prime attribute, it is called partial dependency.

3. **Third Normal Form** – A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form. A relation is in 3NF if at least one of the following condition holds in every non-trivial function dependency X –> Y

   1. X is a super key.
   2. Y is a prime attribute (each element of Y is part of some candidate key).

   Transitive dependency – If A->B and B->C are two FDs then A->C is called transitive dependency.

CHAPTER 4

RESULTS AND SCREENSHOTS

This is the landing page of the Cricket Tournament website. It contains four buttons that will redirect to the details of the database.

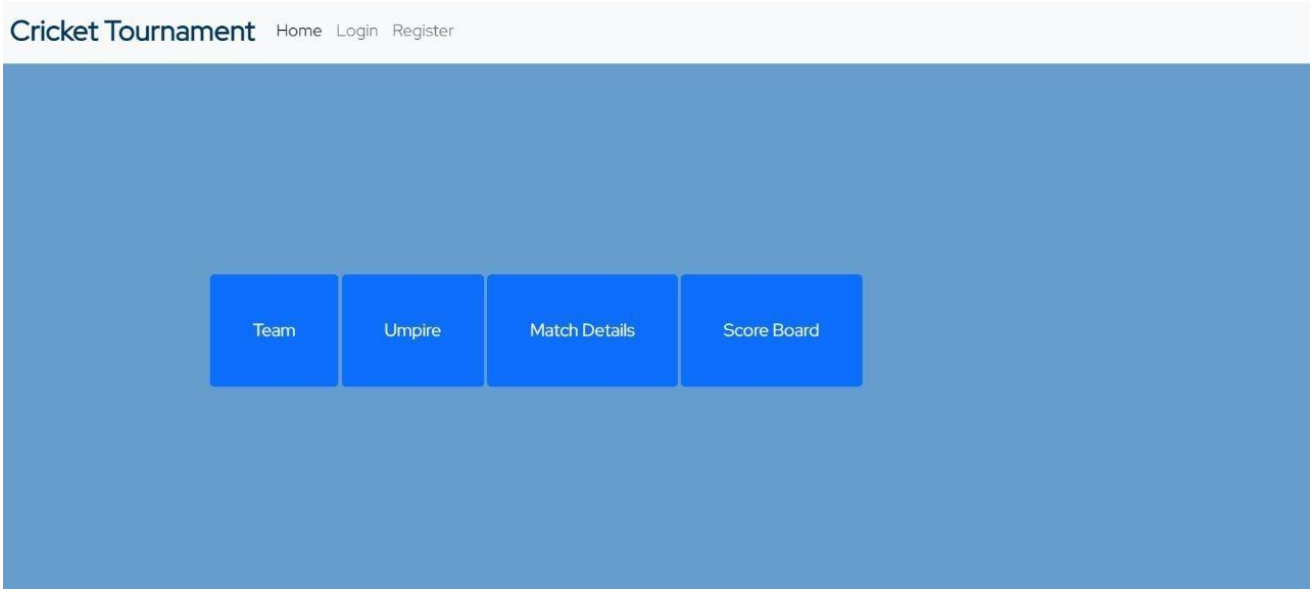

Fig.6.1. Home page

This page displays the details of all the teams that have participated in the World Cup Tournament. The details include the team_id, No. of Batsmen and Bowlers along with their rank.

TeamId	Country	No. Of Batsmen	No. Of Bowlers	Team Rank	Action
TEAMID0001	NewZealand	9	6	1	View
TEAMID0002	England	8	5	2	View
TEAMID0003	South Africa	5	7	5	View
TEAMID0004	West Indies	7	7	8	View
TEAMID0005	INDIA	7	6	4	View
TEAMID0006	PAKISTHAN	6	7	6	View
TEAMID0007	AUSTRALIA	9	6	3	View
TEAMID0008	SRI LANKA	7	7	9	View

Fig6.2. Cricket Team View

The All-rounder table is a description of all the players who are both batsmen and bowlers. It contains players from all the countries participating in the World Cup Tournament

ALL ROUNDER TABLE

Player Id	Batting Average	No. Of Series	Total Runs	No. Of Wickets	Bowling Type	Economy
PLAYER0005	28.43	50	2445	103	Right-arm medium	5.71
PLAYER0007	27.87	42	3010	13	Right-arm medium	5.45
PLAYER0008	23.04	119	2051	182	Left-arm orthodox	4.23
PLAYER0021	25.57	59	5329	310	Right-arm offbreak	4.23
PLAYER0035	19.96	147	877	134	Right-arm medium	6.02
PLAYER0048	17.46	110	520	28	Left-arm orthodox	7.41
PLAYER0050	23.07	58	1777	110	Right-arm fast	6.68
PLAYER0065	31.57	57	4862	466	left-arm orthodox	4.93
PLAYER0076	27.94	138	1325	99	left-arm-orthodox	4.89
PLAYER0077	33.9	142	11867	218	right-arm-offbreak	4.67
PLAYER0079	24.02	144	1009	149	right-arm-leg-break	5.12
PLAYER0092	30.05	155	1562	43	right-arm-medium	6.12
PLAYER0095	32.41	118	5413	92	right-arm-offbreak	5.67

Fig.6.3. All-rounder table

The admin page contains all the edit controls that the administrator of the Association will be provided. The admin can make changes to the details of the team.

Cricket Tournament Home Edit

Team Table

All Rounder

TeamId	Country	No. Of Batsmen	No. Of Bowlers	Team Rank	Action
TEAMID0001	NewZealand	9	6	1	Edit Team
TEAMID0002	England	8	5	2	Edit Team
TEAMID0003	South Africa	5	7	5	Edit Team
TEAMID0004	West Indies	7	7	8	Edit Team
TEAMID0005	INDIA	7	6	4	Edit Team
TEAMID0006	PAKISTHAN	6	7	6	Edit Team
TEAMID0007	AUSTRALIA	9	6	3	Edit Team
TEAMID0008	SRI LANKA	7	7	9	Edit Team

Fig.6.4. Admin page

This page is form where the administrator of the association gives the revised details of the team players. The submit button makes these changes in the database

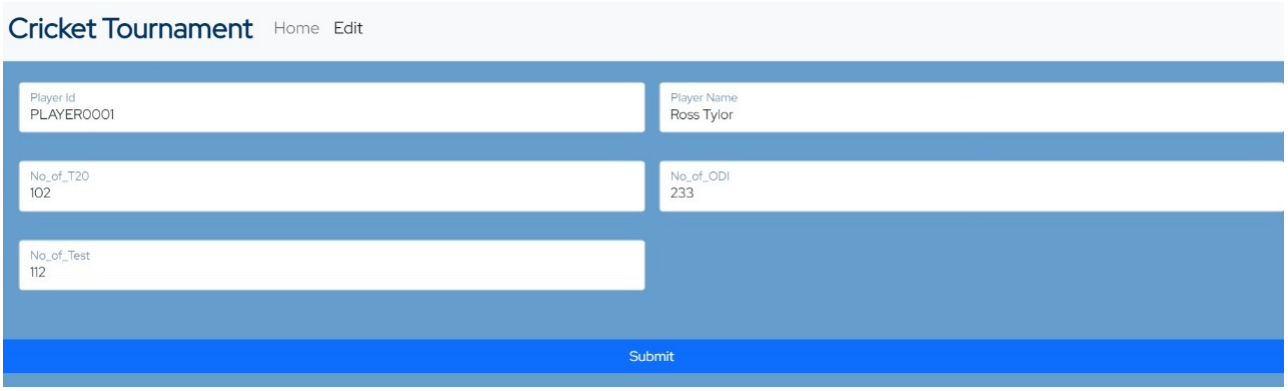

Fig.6.5. Edit page

CONCLUSION AND FUTURE WORK

CONCLUSION

CRICKET TOURNAMENT provides information of various cricket matches, in which all the major countries participate. It also provides us with information about the various players participating in the tournament and provide the analysis of their performance

The aim of the project is to provide the complete information of the International Cricket statistics. The information is available Country wise, and Player wise. This project is very use for Cricket match broadcasters to get information quickly.

It will be able to check anything related to cricket anytime. Paper work and manual work is reduced through this system, the system is user friendly and easy to use.

I hope that user would not only enjoy this system, but also get satisfaction from finding how each and every feature of cricket tournament is implemented.

FUTURE WORK

- Mobile application can be developed to reach a wider audience.
- The project is still mature and fully enthusiastic. Any requirements, this project is completed but still, the project allows to update and modify some modules.
- We always want to implement something more. This project is completed when you watch, but we want to implement more things.

BIBLIOGRAPHY

[1] https://www.magicbricks.com/

[2] https://www.landlordstudio.com

[3] Elmasri and Navathe, Fundamentals of Database System, Sixth Edition, AddisonWesley, 2011

[4] https://www.w3schools.com/html

[5] https://www.w3schools.com/css

[6] https://www.tutorialspoint.com/mysql

[7] https://getbootstrap.com/