Mini project report on

Cricket Metrics: Advanced Dashboard for Performance Analysis

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

UE22CS351A – DBMS Project

Submitted by:

Rishit Rastogi	**PES2UG22CS446**
Rudransh Das	**PES2UG22CS466**

Under the guidance of
Prof. Shilpa S
Assistant Professor
PES University

AUG - DEC 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

Cricket Metrics: Advanced Dashboard for Performance Analysis

is a bonafide work carried out by

Rishit Rastogi	**PES2UG22CS446**
Rudransh Das	**PES2UG22CS466**

In partial fulfilment for the completion of fifth semester DBMS Project (UE22CS351A) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2024 – DEC. 2024. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature
Prof. Shilpa S
Assistant Professor

DECLARATION

We hereby declare that the DBMS Project entitled **Cricket Metrics: Advanced Dashboard for Performance Analysis** has been carried out by us under the guidance of **Prof. Shilpa S, Associate Professor** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester AUG – DEC 2024.

Rishit Rastogi	**PES2UG22CS446**
Rudransh Das	**PES2UG22CS466**

ABSTRACT

Cricket Metrics: Advanced Dashboard for Performance Analysis is a DBMS intended to provide substantial cricket analytics. It is meant for cricket lovers, analysts, and sports professionals to provide them with detailed information about players, teams, matches, and venues. The system utilizes the methods of web scraping to retrieve and update cricket data from online sources in which information relating to player statistics, match details, and conditions of the stadiums are accessed so that the fetched data remains both accurate and updated.

The structured, optimized format with a MySQL database supports efficient querying and analysis of this data. With the use of Python in data processing, the system allows users to conduct specific queries, generate custom reports, and then make an analysis of trends within the performance metrics. Important entities in the database would be modeled in a well-defined schema, along with an ER diagram, supporting complex relationships within cricket data.

It is intuitive and allows searching filters, visualizations, as well as customizable views. This makes it accessible not only to technical users but to users in general. Being initially a command-line tool, this can easily be extended to be either a web or desktop application.

In a nutshell, "Cricket Metrics" aims to enhance cricket analytics by providing reliable, scalable, and user-friendly access to performance data so that informed decisions can be made and further analyses done in the cricket world.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
1.	INTRODUCTION	5
2.	PROBLEM DEFINITION WITH USER REQUIREMENT SPECIFICATIONS	7
3.	LIST OF SOFTWARES/TOOLS/PROGRAMMING LANGUAGES USED	10
4.	ER MODEL	12
5.	ER TO RELATIONAL MAPPING	13
6.	DDL STATEMENTS	14
7.	DML STATEMENTS (CRUD OPERATION SCREENSHOTS)	18
8.	QUERIES (JOIN QUERY, AGGREGATE FUNCTION QUERIES AND NESTED QUERY)	19
9.	STORED PROCEDURE, FUNCTIONS AND TRIGGERS	20
10.	FRONT END DEVELOPMENT (FUNCTIONALITIES/FEATURES OF THE APPLICATION)	21
REFERENCES/BIBLIOGRAPHY		23

INTRODUCTION

The "**Cricket Metrics: Advanced Dashboard for Performance Analysis**" project solves the demand to have structured data about cricket that can be accessed easily. In the present digital world, it is critical for all fans, analysts, and professionals to be furnished with micro-accurate sports analytics on any given player, team, or match outcomes. Its access and management are complicated due to its scattered nature across multiple platforms and for ongoing updates. Our solution addresses this very problem. It will provide a centralized platform with easy storage, retrieval, and analysis of cricket data.

This system sits on top of a solid MySQL database and ingests data coming from a Python script reading directly out of a JSON file. Thus, by automating the population of this data into the database, we ensure consistency in the database and can easily feed new data into it. The Python script reads from the JSON file detailing precise statistics, populates tables with players, teams, matches, and other necessary metrics as the basis on which to base analyses.

Other key functionalities of the "Cricket Metrics" project include the ability to view bowlers' statistics above the rest, including his name and nationality and wickets, retrieve best teams, where it displays each team's name, matches played, and matches won, search by the name of the players including quick access into every player's ID, name, and nationality.

It further comprises a team stats update function where users can input a team ID for updating the relevant statistics within the database such that data continues to reflect the most up-to-date information without the need for hand editing. It is designed to be very interactive and easily accessible to enable users to explore data and analyze performance trends straight from the database.

In the final analysis, it is a well-of-useful tool in cricket analytics and gathers many data points within a single frame and supports deep performance analysis. Such an approach brings better accessibility to data but also allows for multiple queries, which adds more value to any stakes who hold interests in cricket analytics.

PROBLEM DEFINITION WITH USER REQUIREMENT SPECIFICATIONS

The aim of the project was to provide a centralized accessible cricket database for the masses, analysts, and professionals. Data is scattered in so many sources, and hence the users find it challenging to get any actionable insight from players, teams, and matches. This project has solved the problem by creating a strong MySQL database populated with data through a Python script reading a JSON file. It aggregates crucial stats onto one page for view and calculation, where users can view top bowlers, search by player name, and refresh team statistics in a click.

User Requirement Specification :

1) **Data Ingestion:**

- The system should automate the process of populating the MySQL database by reading data from a JSON file.
- A Python script reads structured cricket data in JSON format, including player and team statistics, match results, and performance metrics, and inserts it into respective database tables. This ensures efficient, organized data storage, enabling consistent data retrieval and analysis.
- Streamline database setup and updates without requiring manual data entry, maintaining accuracy and scalability.

2) **Top Bowlers View:**

- Users can view a list of top-performing bowlers, focusing on their total wickets taken.
- The system retrieves data on bowlers from the database, displaying their names, nationalities, and total wickets. Users can easily access insights into the highest wicket-takers and their backgrounds.

- Provide users with quick access to essential performance metrics of bowlers to help identify top players.

3) Top Teams View:

- Displays a list of top teams with their cumulative career achievements.
- Shows each team's name, total matches played, and matches won to allow users to analyze team performance over time.
- Helps users compare team successes, providing valuable insights for fans, analysts, and stakeholders.

4) Player Search:

- Users can search for players by name to quickly find relevant information.
- The system retrieves and displays the player's ID, name, and nationality, making it easy to identify specific players in the database.
- This feature is useful for quickly referencing player details or for detailed player-based analysis.

5) Update Team Stats:

- Allows users to update statistics for a specific team by entering a team ID.
- Updates data such as matches played and won in the database, keeping team information current.
- Ensures that the database reflects the latest team performance metrics without manual record modifications.

6)Data Consistency:

- All data should be consistently stored and retrieved from the MySQL database.
- Ensures the accuracy of data during both insertion (from JSON) and retrieval.
- Provides reliable information for analysis, supporting valid insights across all functionalities.

LIST OF SOFTWARES/TOOLS/PROGRAMMING LANGUAGES USED

1) Python

- Python serves as the core programming language for this project, handling data ingestion, processing, database connections, and various functionalities.
- Its ease of use, extensive libraries, and versatility make it ideal for data-driven applications, enabling efficient scripting and smooth integration with databases, data visualization tools, and web frameworks.

2) Streamlit

- Streamlit is used to create the graphical user interface (GUI) for the project, allowing users to interact with the database through a web-based dashboard to view and analyze cricket metrics.
- Streamlit simplifies web development with Python by offering easy-to-use components for UI creation, supporting rapid prototyping, real-time interactivity, and seamless integration with data libraries.

3) PyMySQL

- PyMySQL connects Python to the MySQL database, facilitating data transfer between the application and the database for storing and retrieving cricket data.
- As a lightweight and easy-to-use library, PyMySQL enables reliable, fast, and secure connections to MySQL, making database operations seamless and accessible.

4) JSON Library

- The JSON library in Python is used to read data from JSON files, which contain structured cricket data to be ingested into the database.
- JSON is widely used for data exchange due to its lightweight format and readability, allowing easy data parsing, conversion, and integration into the database.

5) **Pandas**

- Pandas provides powerful data manipulation and analysis capabilities, making it essential for data processing, filtering, and statistical computations within the project.
- It simplifies handling large datasets, enabling efficient data wrangling and preparation for visualization or further analysis, enhancing the overall analytical capacity of the application.

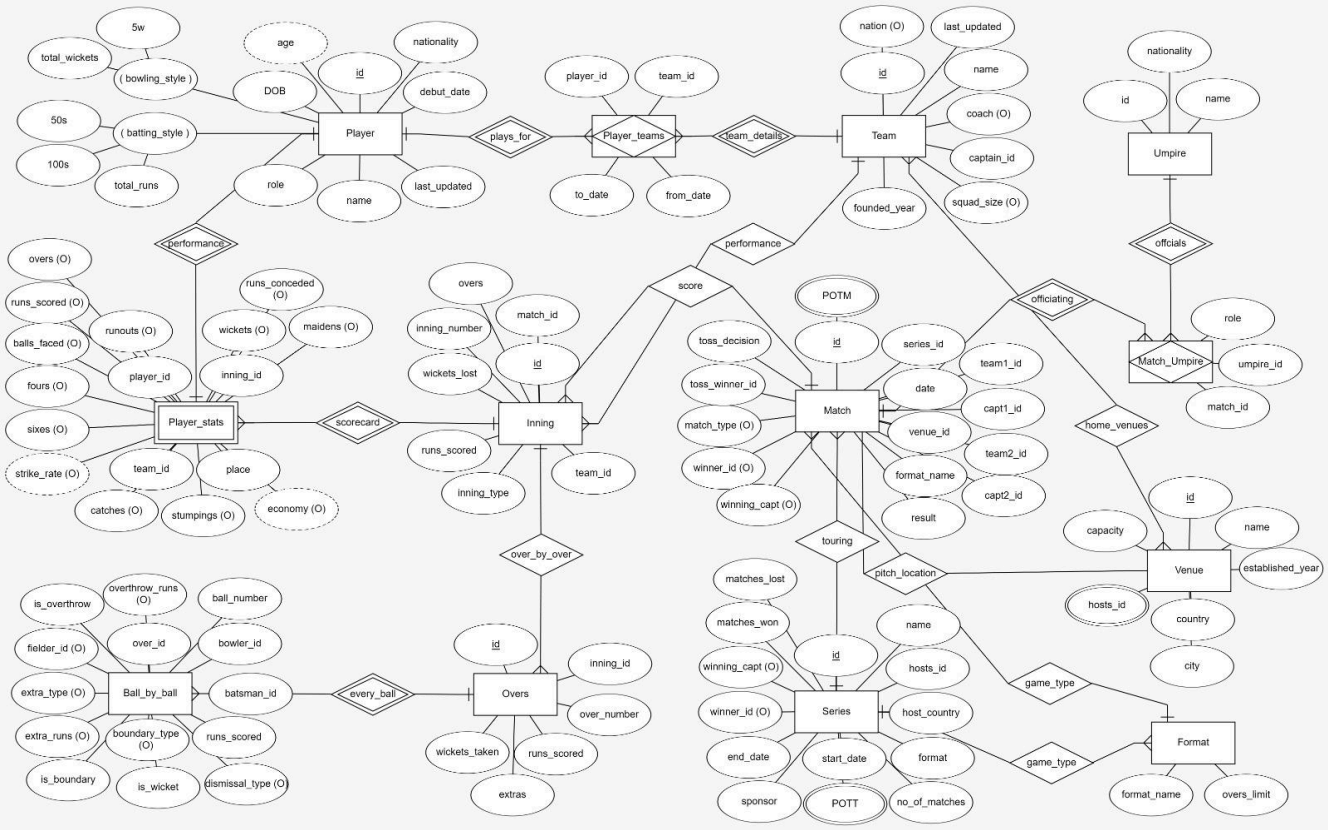
6) **Matplotlib**

- Matplotlib is used to create visualizations, including graphs and charts, to illustrate cricket metrics and statistical trends for better analysis.
- With its robust plotting functions, Matplotlib enables detailed, customizable visual representations, helping users interpret data insights effectively.

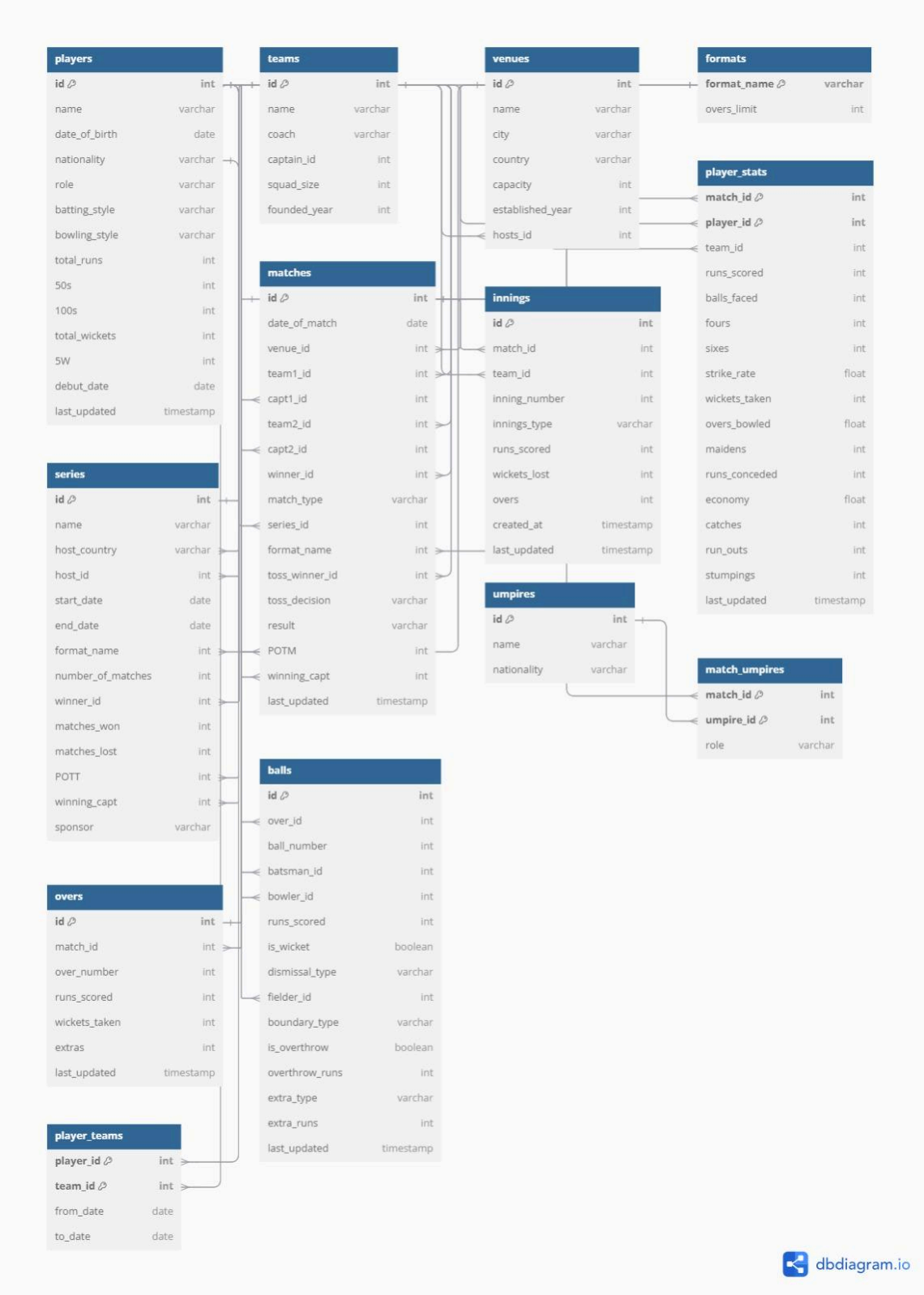
7) **MySQL**

- MySQL serves as the project's database, where cricket data is stored in a structured format for consistent retrieval and analysis.
- Known for its reliability, scalability, and wide community support, MySQL ensures efficient storage, management, and querying of complex data structures, providing a solid foundation for data analysis.

ER MODEL


Player
5w
total_wickets
(bowling_style)
age
id
nationality
DOB
debut_date
50s
(batting_style)
100s
total_runs
role
name
last_updated
plays_for
Player_teams
player_id
team_id
to_date
from_date
team_details
Team
nation (O)
last_updated
id
name
coach (O)
captain_id
squad_size (O)
founded_year
Umpire
nationality
id
name
offcials
officiating
Match_Umpire
role
umpire_id
match_id
performance
performance
overs (O)
runs_scored (O)
balls_faced (O)
fours (O)
sixes (O)
strike_rate (O)
runouts (O)
runs_conceded (O)
wickets (O)
maidens (O)
player_id
inning_id
team_id
place
catches (O)
stumpings (O)
economy (O)
Player_stats
scorecard
Inning
overs
inning_number
match_id
id
wickets_lost
runs_scored
team_id
inning_type
score
POTM
id
toss_decision
toss_winner_id
match_type (O)
winner_id (O)
winning_capt (O)
Match
series_id
date
venue_id
format_name
result
team1_id
capt1_id
team2_id
capt2_id
home_venues
id
capacity
name
Venue
established_year
hosts_id
country
city
over_by_over
touring
pitch_location
matches_lost
matches_won
winning_capt (O)
winner_id (O)
end_date
sponsor
id
Series
name
hosts_id
host_country
start_date
POTT
format
no_of_matches
game_type
game_type
Format
format_name
overs_limit
is_overthrow
overthrow_runs (O)
ball_number
fielder_id (O)
over_id
bowler_id
extra_type (O)
Ball_by_ball
batsman_id
extra_runs (O)
boundary_type (O)
runs_scored
is_boundary
is_wicket
dismissal_type (O)
every_ball
id
Overs
inning_id
over_number
wickets_taken
runs_scored
extras

ER TO RELATIONAL MAPPING



DDL STATEMENTS

The DDL statements used for this project are :

CREATE TABLE `players` (`id` int PRIMARY KEY, `name` varchar(255), `date_of_birth` date, `age` int, `nationality` varchar(255), `role` varchar(255), `batting_style` varchar(255), `bowling_style` varchar(255), `total_runs` int, `50s` int, `100s` int, `total_wickets` int, `5W` int, `debut_date` date, `last_updated` timestamp);

CREATE TABLE `teams` (`id` int PRIMARY KEY, `name` varchar(255), `nation` text, `coach` varchar(255), `captain_id` int, `squad_size` int, `founded_year` int, `last_updated` timestamp);

CREATE TABLE `venues` (`id` int PRIMARY KEY, `name` varchar(255), `city` varchar(255), `country` varchar(255), `capacity` int, `established_year` int, `hosts_id` int);

CREATE TABLE `matches` (`id` int PRIMARY KEY, `date_of_match` date, `venue_id` int, `team1_id` int, `capt1_id` int, `team2_id` int, `capt2_id` int, `winner_id` int, `match_type` varchar(255), `series_id` int, `format_name` varchar(255), `toss_winner_id` int, `toss_decision` varchar(255), `result` varchar(255), `POTM` int, `winning_capt` int);

CREATE TABLE `innings` (`id` int PRIMARY KEY, `match_id` int, `team_id` int, `inning_number` int, `innings_type` varchar(255) COMMENT 'All-out, Over finished, Declared or Follow-on', `runs_scored` int, `wickets_lost` int, `overs` int);

CREATE TABLE `player_stats` (`inning_id` int, `player_id` int, `team_id` int, `place` varchar(255), `runs_scored` int, `ball_by_ball_faced` int, `fours` int, `sixes` int, `strike_rate` float, `wickets_taken` int, `overs_bowled` float, `maidens` int, `runs_conceded` int, `economy` float, `catches` int, `run_outs` int, `stumpings` int, PRIMARY KEY (`inning_id`, `player_id`));

CREATE TABLE `overs` (`id` int PRIMARY KEY, `inning_id` int, `over_number` int, `runs_scored` int, `wickets_taken` int, `extras` int);

CREATE TABLE `ball_by_ball` (`over_id` int, `ball_number` int, `batsman_id` int, `bowler_id` int, `runs_scored` int, `is_wicket` boolean, `dismissal_type` varchar(255) COMMENT 'Type of dismissal: Bowled, Caught, LBW, Stumped, Run Out, etc.', `fielder_id` int COMMENT 'The fielder involved in the dismissal', `is_boundary` boolean, `boundary_type` varchar(255) COMMENT 'Indicates if the ball went for a boundary: 4, 6, None', `is_overthrow` boolean, `overthrow_runs` int, `extra_type` varchar(255) COMMENT 'Type of extra: No Ball, Wide, Leg Bye, Byes, Penalty, etc.', `extra_runs` int COMMENT 'Runs scored due to extra', PRIMARY KEY (`over_id`, `ball_number`));

CREATE TABLE `player_teams` (`player_id` int, `team_id` int, `from_date` date, `to_date` date, PRIMARY KEY (`player_id`, `team_id`));

tables Examples :

1) ball by ball

over_id	ball_number	batsman_id	bowler_id	runs_scored	is_wicket	dismissal_type	fielder_id	is_boundar
1	1	1	20	0	0	NULL	NULL	0
1	2	1	20	0	0	NULL	NULL	0
1	3	1	20	1	0	NULL	NULL	0
1	4	2	20	2	0	NULL	NULL	0
1	5	2	20	0	0	NULL	NULL	0
1	6	2	20	3	0	NULL	NULL	0
2	1	2	21	0	0	NULL	NULL	0

2) innings :

id	match_id	team_id	inning_number	innings_type	runs_scored	wickets_lost	overs
1	1	1	1	Batting	168	6	20
2	1	2	2	Batting	172	5	20
3	2	1	1	Batting	173	10	20
4	2	2	2	Batting	176	8	20
5	3	1	1	Batting	187	6	20
6	3	2	2	Batting	146	10	18
7	4	3	1	Batting	169	5	20
8	4	4	2	Batting	129	10	20

3) Matches:

id	date_of_match	venue_id	team1_id	capt1_id	team2_id	capt2_id	winner_id	match_type	series_id	format_name	toss_winner_id	
1	2017-02-17	1	1	NULL	2	NULL	2	NULL	NULL	NULL	2	fi
2	2017-02-19	2	1	NULL	2	NULL	2	NULL	NULL	NULL	2	fi
3	2017-02-22	3	1	NULL	2	NULL	1	NULL	NULL	NULL	2	fi
4	2016-09-05	4	4	NULL	3	NULL	3	NULL	NULL	NULL	3	b
5	2016-06-18	5	6	NULL	5	NULL	6	NULL	NULL	NULL	5	fi
6	2016-06-20	5	6	NULL	5	NULL	5	NULL	NULL	NULL	6	b
7	2016-06-22	5	6	NULL	5	NULL	5	NULL	NULL	NULL	6	fi

4) Overs :

id	inning_id	over_number	runs_scored	wickets_taken	extras
1	1	0	6	0	0
2	1	1	7	0	0
3	1	2	6	0	0
4	1	3	8	0	0
5	1	4	16	0	0
6	1	5	4	0	1
7	1	6	6	0	0
8	1	7	8	0	1

5) Player Stats :

inning_id	player_id	team_id	place	runs_scored	ball_by_ball_faced	fours	sixes	strike_rate	wickets_taker
1	15	1	NULL	NULL	NULL	NULL	NULL	NULL	1
1	17	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL
1	18	1	NULL	NULL	NULL	NULL	NULL	NULL	0
1	19	1	NULL	NULL	NULL	NULL	NULL	NULL	1
1	20	1	NULL	NULL	NULL	NULL	NULL	NULL	2
1	21	1	NULL	NULL	NULL	NULL	NULL	NULL	1
1	22	1	NULL	NULL	NULL	NULL	NULL	NULL	1

6) player_teams :

player_id	team_id	from_date	to_date
1	1	2024-11-06	NULL
2	1	2024-11-06	NULL
3	1	2024-11-06	NULL
4	1	2024-11-06	NULL
5	1	2024-11-06	NULL
6	1	2024-11-06	NULL
7	1	2024-11-06	NULL
7	19	2024-11-06	NULL

7)teams :

id	name	nation	coach	captain_id	squad_size	founded_year	last_updated	total_runs	total_wins
1	Australia	Aus	NULL	NULL	NULL	NULL	2024-11-06 11:54:00	736205	120
2	Sri Lanka	Sri	NULL	NULL	NULL	NULL	2024-11-06 12:38:24	736205	93
3	Hong Kong	Hon	NULL	NULL	NULL	NULL	2024-11-06 12:54:52	736205	44
4	Ireland	Ire	NULL	NULL	NULL	NULL	2024-11-06 12:55:46	736205	70

8)Venues :

	id	name	city	country	capacity	established_year
▶	1	Melbourne Cricket	Melbourne Cricket Ground	NULL	NULL	NULL
	2	Simonds Stadium, South Geelong	Victoria	NULL	NULL	NULL
	3	Adelaide Oval	Unknown	NULL	NULL	NULL
	4	Bready Cricket Club, Magheramason	Londonderry	NULL	NULL	NULL
	5	Harare Sports Club	Unknown	NULL	NULL	NULL
	6	McLean Park	Napier	NULL	NULL	NULL
	7	Bay Oval	Mount Maunganui	NULL	NULL	NULL
	8	Eden Park	Auckland	NULL	NULL	NULL

9)Players :

	id	name	date_of_birth	age	nationality
▶	1	AJ Finch	NULL	NULL	Australia
	2	M Klinger	NULL	NULL	Australia
	3	TM Head	NULL	NULL	Australia
	4	MC Henriques	NULL	NULL	Australia
	5	AJ Turner	NULL	NULL	Australia
	6	JP Faulkner	NULL	NULL	Australia
	7	TD Paine	NULL	NULL	Australia
	8	DJ Cumming	NULL	NULL	Australia

DML STATEMENTS

```
ALTER TABLE innings MODIFY id INT AUTO_INCREMENT;
ALTER TABLE overs MODIFY id INT AUTO_INCREMENT;
ALTER TABLE umpires MODIFY id INT AUTO_INCREMENT;
ALTER TABLE players MODIFY id INT AUTO_INCREMENT;
ALTER TABLE teams MODIFY id INT AUTO_INCREMENT;
ALTER TABLE venues MODIFY id INT AUTO_INCREMENT;
ALTER TABLE matches MODIFY id INT AUTO_INCREMENT;
ALTER TABLE series MODIFY id INT AUTO_INCREMENT;
-- Add for other tables as necessary
SET FOREIGN_KEY_CHECKS = 1;
```

```
-- Enable foreign key checks after truncation
SET FOREIGN_KEY_CHECKS = 1;

ALTER TABLE matches MODIFY POTM varchar(255);
ALTER TABLE matches DROP FOREIGN KEY matches_ibfk_10;
alter table matches add won_by varchar(255);
alter table matches drop total_wins;
alter table teams add total_runs int;
alter table teams add total_wins int;
alter table player_stats add average float;
```

```
-- Truncate each table
TRUNCATE TABLE ball_by_ball;
TRUNCATE TABLE overs;
TRUNCATE TABLE innings;
TRUNCATE TABLE matches;
TRUNCATE TABLE teams;
TRUNCATE TABLE players;
TRUNCATE TABLE venues;
TRUNCATE TABLE formats;
TRUNCATE TABLE umpires;
TRUNCATE TABLE match_umpires;
TRUNCATE TABLE player_stats;
TRUNCATE TABLE player_teams;
TRUNCATE TABLE series;
```

QUERIES

```sql
SELECT m.id AS match_id, m.date_of_match, t1.name AS team1_name, t2.name AS team2_name,
       t3.name AS winner_name, m.result
FROM matches m
JOIN teams t1 ON m.team1_id = t1.id
JOIN teams t2 ON m.team2_id = t2.id
LEFT JOIN teams t3 ON m.winner_id = t3.id;
```

```sql
SELECT i.id AS inning_id, m.id AS match_id, m.date_of_match, t.name AS team_name,
       i.runs_scored, i.wickets_lost, i.overs
FROM innings i
JOIN matches m ON i.match_id = m.id
JOIN teams t ON i.team_id = t.id;
```

```sql
SELECT player_id, (runs_scored * 100.0 / ball_by_ball_faced) AS strike_rate
FROM player_stats
WHERE player_id = (
    SELECT player_id
    FROM player_stats
    ORDER BY (runs_scored * 100.0 / ball_by_ball_faced) DESC
    LIMIT 1
);
```

```sql
SELECT t.name, t.total_wins
FROM teams t
WHERE t.total_wins > (
    SELECT AVG(total_wins)
    FROM teams
);
```

STORED PROCEDURE, FUNCTIONS AND TRIGGERS

```
CREATE FUNCTION get_bowling_average(player_id INT)
RETURNS FLOAT
DETERMINISTIC
BEGIN
    DECLARE runs_conceded INT;
    DECLARE wickets_taken INT;
    DECLARE average FLOAT;

    SELECT SUM(runs_conceded), SUM(wickets_taken)
    INTO runs_conceded, wickets_taken
    FROM player_stats
    WHERE player_id = player_id;

    IF wickets_taken = 0 THEN
        RETURN NULL;  -- No wickets taken
    ELSE
        SET average = runs_conceded / wickets_taken;
        RETURN average;
    END IF;
END;
```

1)

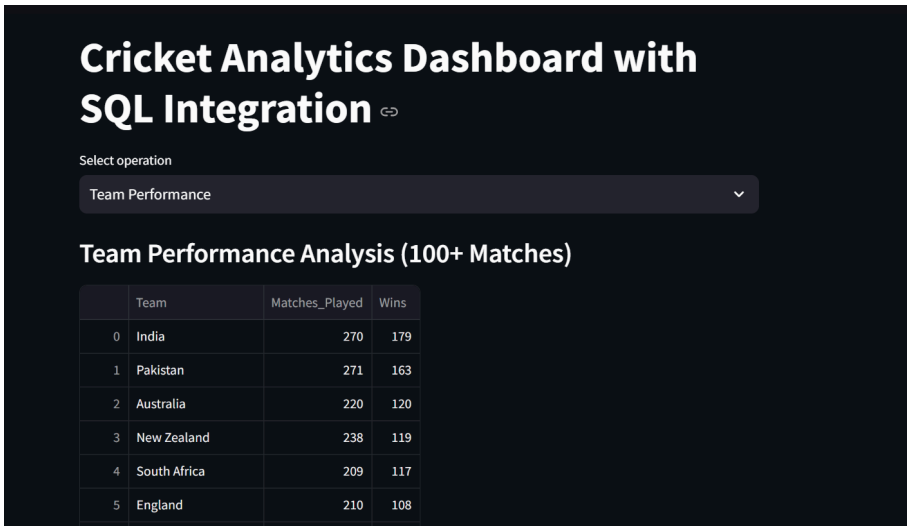
2)

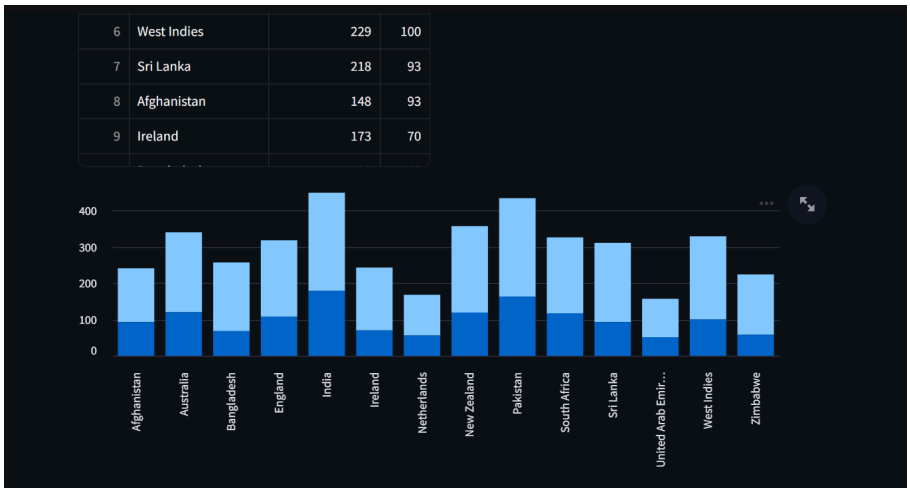
```
CREATE TRIGGER before_team_update
BEFORE UPDATE ON teams
FOR EACH ROW
BEGIN
    SET NEW.last_updated = NOW();
END //
```

3)

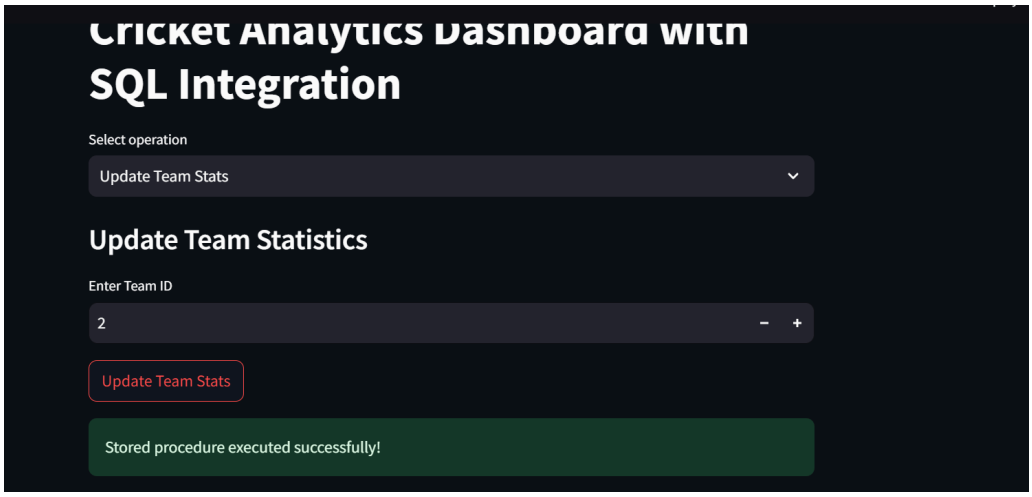
```
DELIMITER //
CREATE PROCEDURE update_team_stats(IN team_id INT)
BEGIN
    DECLARE total_runs INT DEFAULT 0;
    DECLARE total_wins INT DEFAULT 0;

    SELECT SUM(runs_scored) INTO total_runs
    FROM innings
    WHERE team_id = team_id;

    SELECT COUNT(*) INTO total_wins
    FROM matches
    WHERE winner_id = team_id;

    UPDATE teams
    SET last_updated = NOW(), total_runs = total_runs, total_wins = total_wins
    WHERE id = team_id;
END //
```

FRONT END DEVELOPMENT

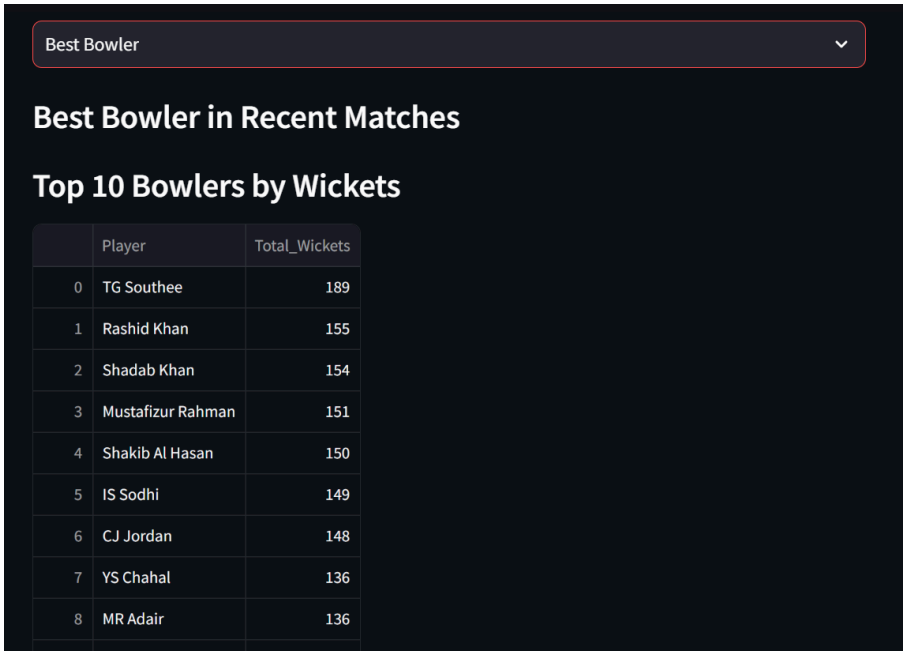




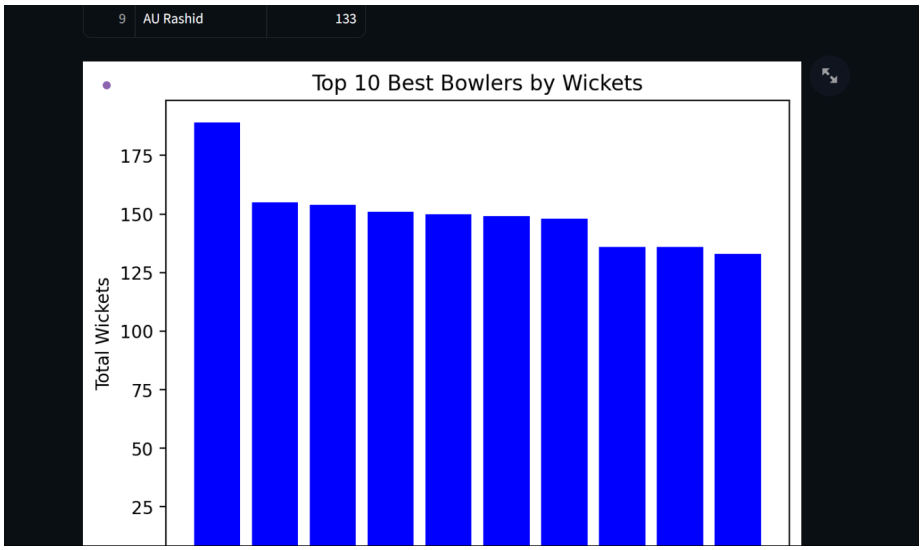
2)Updating team stats



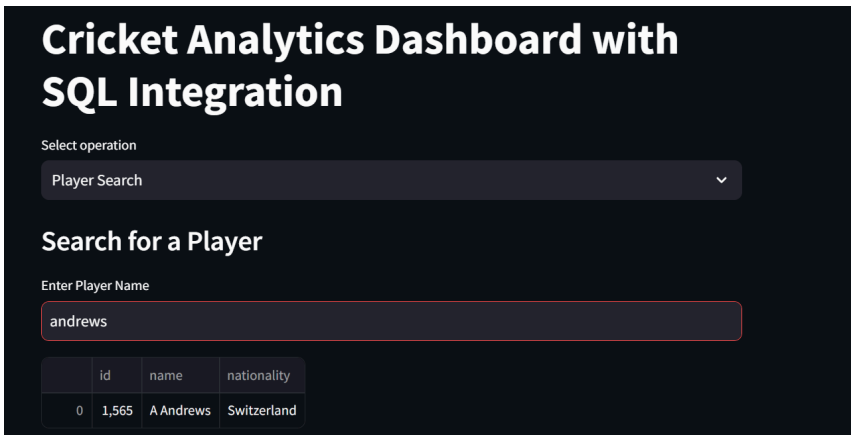
3) Best Bowler :



	Player	Total_Wickets
0	TG Southee	189
1	Rashid Khan	155
2	Shadab Khan	154
3	Mustafizur Rahman	151
4	Shakib Al Hasan	150
5	IS Sodhi	149
6	CJ Jordan	148
7	YS Chahal	136
8	MR Adair	136
9	AU Rashid	133



4) Player Search :



	id	name	nationality
0	1,565	A Andrews	Switzerland

BIBLIOGRAPHY

1) MySQL Documentation. (n.d.). MySQL 8.0 Reference Manual. Oracle. Retrieved from https://dev.mysql.com/doc/refman/8.0/en/

- The official MySQL documentation provides detailed information on database setup, management, and SQL queries, which was essential for setting up and optimizing the database.

2) Streamlit Documentation. (n.d.). Streamlit Documentation. Retrieved from https://docs.streamlit.io/

- Streamlit's official documentation offers a comprehensive guide to building interactive web applications with Python, which supported the development of the GUI for this project.

3) PyMySQL Documentation. (n.d.). PyMySQL Documentation. Retrieved from https://pymysql.readthedocs.io/

- This source details how to use PyMySQL to connect Python applications to a MySQL database, which was critical in managing database connections and queries within the project.

4) JSON Documentation. (n.d.). The JSON Data Interchange Format. ECMA International. Retrieved from https://www.json.org/json-en.html

- JSON's official documentation provides the specifications for this lightweight data-interchange format, facilitating data extraction and transformation for ingestion into the MySQL database.