Sports Analytics Management System

DBMS Mini-Project Report

Submitted By:

Navika Srikanth – PES1UG23CS385

Nikhil Thomas Sojan – PES1U23CS393

Section - G

1. Introduction

This project implements a complete Sports Analytics Management System designed to manage teams, players, matches, scores, and injuries while providing analytical insights through a web-based dashboard. The system includes authentication, administrative controls, data visualization, and automated audit logging. It is built using a modern tech stack and emphasizes clean data management and analytical depth.

2. Objectives

- To design and implement a relational database for sports data management.
- To build a web interface allowing users to view and analyze sports statistics.
- To implement administrative features for CRUD operations on all entities.
- To integrate analytics such as xG, score flow, team comparison radar charts, MVP prediction, and play-by-play timelines.
- To implement a secure authentication system.
- To maintain a complete audit log for all database changes.

3. Technologies Used

Frontend / UI

- Streamlit (Python-based web framework)
- Plotly (data visualization)
- Custom CSS integrated for a dark theme

Backend

- Python (business logic, API layer, Streamlit pages)
- MySQL (database, procedures, triggers, views)

Database Layer

- Stored Procedures for CRUD

- Triggers for validation and audit logging
- Views for analytics and summaries
- Full-Text Search and Window Functions

Tools

- Git, GitHub
- VS Code
- SQLAlchemy for database connections
- bcrypt for password hashing

4. System Architecture

The system follows a modular structure:

4.1 Database Layer

Contains core tables:

- Users
- Teams
- Players
- Matches
- Scores
- Injuries
- Team_Match
- Audit_Log

Includes additional components:

- Views: player summary, injury summary, match play-by-play, score flow, xG, team stats, key moments, MVP ranking
- Stored Procedures for each table (add, update, delete)
- Triggers for validation and automatic audit logging

4.2 Application Layer

Organized into Streamlit pages:

- **Players** – Search, filters, analytics, CRUD
- **Teams** – View and manage teams
- **Matches** – Search, timeline, score flow chart, xG plots, radar comparison, MVP prediction
- **Scores** – Add/update scoring events
- **Injuries** – Manage injuries, active injury filtering
- **Admin Panel** – Complete system-level CRUD
- **Authentication** – Login & Signup with hashed passwords

4.3 Common Module

Handles:

- Database engine connection (SQLAlchemy)
- Stored procedure calls
- Theming and UI consistency
- Authentication guard
- Password hashing and verification
- Modal rendering

5. Key Features

5.1 Authentication System

- User registration and login
- Password hashing with bcrypt
- Role-based access control (admin and viewer)
- Sidebar user identity and logout functionality

5.2 Admin Panel

Allows admins to:

- Add, update, delete teams
- Add, update, delete players
- Add, update, delete matches
- Add, update, delete scores
- Add, update, delete injuries
- View full audit logs of all changes

5.3 Match Analytics

- Scoring timeline
- Cumulative score-flow visualization
- Momentum-based turning point estimation
- Expected Goals (xG) per player
- Heatmap of scoring minutes
- Team comparison radar charts
- MVP prediction based on goals, xG, and key moments
- Play-by-play breakdown

5.4 Injury Management

- Track injury type, dates, expected recovery
- Automatic detection of “active” injuries
- Team-wise and status-wise filtering
- Clean card-based layout for viewing injuries

5.5 Player & Team Insights

- Full-text search on players
- Total points, position, nationality, last match date
- Player age calculation using SQL function
- Team performance ranking using window functions

5.6 Audit Logging

All CRUD operations across major tables are automatically recorded:

- Table changed
- Record ID
- Action (INSERT, UPDATE, DELETE)
- Timestamp
- Actor identity

6. Data Flow

1. User logs in via Streamlit interface.
2. Auth guard validates session and applies role-based access.
3. Admin actions trigger stored procedures.
4. Procedures perform data operations and write to Audit_Log.
5. Views and analytics models compute results dynamically.
6. Streamlit renders results through interactive visualizations.

7. Conclusion

The Sports Analytics Management System is a comprehensive DBMS mini-project integrating database design, authentication, analytics, and a modern web interface. It demonstrates the use of stored procedures, triggers, views, and SQL window functions, along with Python-driven UI development. The system provides real-time insights into player performance, match trends, and team analytics while maintaining a secure and auditable data workflow.