COMPANY PROFILE

Company Name: EZ Trainings and Technologies Pvt. Ltd.

Introduction:

EZ Trainings and Technologies Pvt. Ltd. is a dynamic and innovative organization dedicated to providing comprehensive training solutions and expert development services. Established with a vision to bridge the gap between academic learning and industry requirements, we specialize in college trainings for students, focusing on preparing them for successful placements. Additionally, we excel in undertaking development projects, leveraging cutting-edge technologies to bring ideas to life.

Mission:

Our mission is to empower the next generation of professionals by imparting relevant skills and knowledge through specialized training programs. We strive to be a catalyst in the career growth of students and contribute to the technological advancement of businesses through our development projects.

Services:

College Trainings:

- Tailored training programs designed to enhance the employability of students.
- Industry-aligned curriculum covering technical and soft skills.
- Placement assistance and career guidance.

Development Projects:

- End-to-end development services, from ideation to execution.
- Expertise in diverse technologies and frameworks
- . • Custom solutions to meet specific business needs.

Locations: Hyderabad | Delhi NCR

At EZ Trainings and Technologies Pvt. Ltd., we believe in transforming potential into excellence

ABSTRACT

- The Cricket Match Results Tracker (CMRT) is an automated tool developed in Python for efficiently tracking and analysing cricket match results.
- In the dynamic world of cricket, where matches occur across various formats and tournaments, CMRT serves as a comprehensive platform for capturing, processing, and interpreting match data.
- CMRT offers a user-friendly interface for inputting match details such as team names, scores, venues, match formats, and player performances.
- Leveraging Python's libraries including Pandas, Matplotlib, and Requests, CMRT employs advanced algorithms for data processing, enabling users to generate comprehensive match summaries, statistical analyses, and visual representations of match outcomes.
- Key features of CMRT include real-time data retrieval through web scraping techniques, historical data storage for trend analysis, player performance metrics calculation, and customizable reporting options.
- Additionally, CMRT allows for comparative analysis between teams, players, and venues, facilitating deeper insights into match dynamics and strategic trends.
- CMRT is designed to be scalable and adaptable, catering to the needs of cricket enthusiasts, analysts, coaches, and administrators.
- By providing an automated and intuitive solution for cricket match analysis, CMRT aims to enhance decision-making processes and improve understanding of cricket match outcomes in both professional and recreational context.

INTRODUCTION OF THE PROJECT

- The Cricket Match Result Tracker (CMRT) emerges as a comprehensive solution to fulfill this need.
- With each match generating a wealth of data, it becomes imperative to have a structured approach to gather and interpret this information
- CMRT provides a user-friendly interface to input essential match details such as teams, scores, venues, player performances, and match formats.
- Utilizing advanced algorithms, CMRT processes the collected data to generate comprehensive match summaries, player statistics, team performance metrics, and trend analyses.
- CMRT offers visual representations of match data through graphs, charts, and heatmaps, facilitating easier interpretation and understanding of match outcomes and tree
- CMRT stores historical match data, allowing users to track performance trends over time, conduct comparative analyses, and gain insights into long-term patterns.
- Users can receive real-time updates on ongoing matches, enabling them to stay informed about match progress, outcomes, and key events as they happen.
- CMRT is highly customizable, allowing users to tailor reports, analyses, and visualizations according to their specific requirements and preferences.

MODULE DESCRIPTION

- Data Input Module: This module handles the input of match details such as teams, scores, venues, player performances, and match formats. It includes functionality for manual data entry and may incorporate features for automated data retrieval from online sources.
- Data Storage Module: Responsible for storing the inputted match data in a structured format. This module may use databases or file systems to efficiently manage and retrieve the data.
- Data Processing Module: Processes the stored match data to generate meaningful insights and analyses. It calculates player statistics (e.g., runs scored, wickets taken, batting averages, bowling averages), team performance metrics (e.g., win-loss records, run rates), and other relevant metrics.
- Visualization Module: Presents the processed data in visual formats such as graphs, charts, and tables. This module aids users in interpreting and understanding match outcomes, player performances, and strategic trends more effectively.
- Historical Data Management Module: Manages the storage and retrieval of historical match data. It allows users to access past match results, track performance trends over time, conduct comparative analyses, and gain insights into long-term patterns.
- Real-time Updates Module: Provides real-time updates on ongoing matches, including match progress, outcomes, and key events. This module ensures that users stay informed about the latest developments as they occur.
- User Management Module: Handles user authentication, access control, and permissions management. This module ensures that only authorized users have access to sensitive match data
- Reporting Module: Generates customizable reports summarizing match outcomes, player performances, and team statistics. Users can customize the content and format of the reports based on their preferences.

- Notification Module: Optionally, provides notification alerts for important match events, updates, or changes in match status. This module keeps users informed in real-time, even when they are not actively using the tracker.
- Integration Module: Facilitates integration with external data sources and APIs, allowing users to augment the tracker's capabilities with additional data sources and functionalities.

ALGORITHM

Here's the algorithm for the given code:

- Import necessary libraries: stream lit, j son, OS, and mat plot lib.pyplot.
- Define functions for loading and saving match data from/to a JSON file.
- Define functions for adding a match, displaying match results, displaying team performance graph, updating match data, searching for a match, and deleting a particular match data.
- Create the main Streamlit UI with a title.
- Load existing match data from a JSON file.
- Add a section to add a new match, which includes input fields for team names, scores, Man of the Match, and Highest Wicket Taker. Upon clicking the "Add Match" button, the match data is added to the match _data dictionary and saved to the JSON file.
- Add a section to display match results. This section displays the team names, scores, Man of the Match, and Highest Wicket Taker for each match.
- Add a section to display the team performance graph. This section aggregates the scores for each team across all matches and plots a bar graph showing the total runs scored by each team.
- Add a section to search for a match based on a search query. Upon clicking the "Search" button, the code searches for the match containing the search query and displays its details if found.
- Add a section to update match data. This section allows users to select a match to update and provides input fields to modify the team names, scores, Man of the Match, and Highest Wicket Taker. Upon clicking the "Update Match" button, the match data is updated and saved to the JSON file.
- Add a section to delete a particular match data. This section allows users to select a match to delete. Upon clicking the "Delete Match" button, the selected match data is deleted from the match _data dictionary and saved to the JSON file.

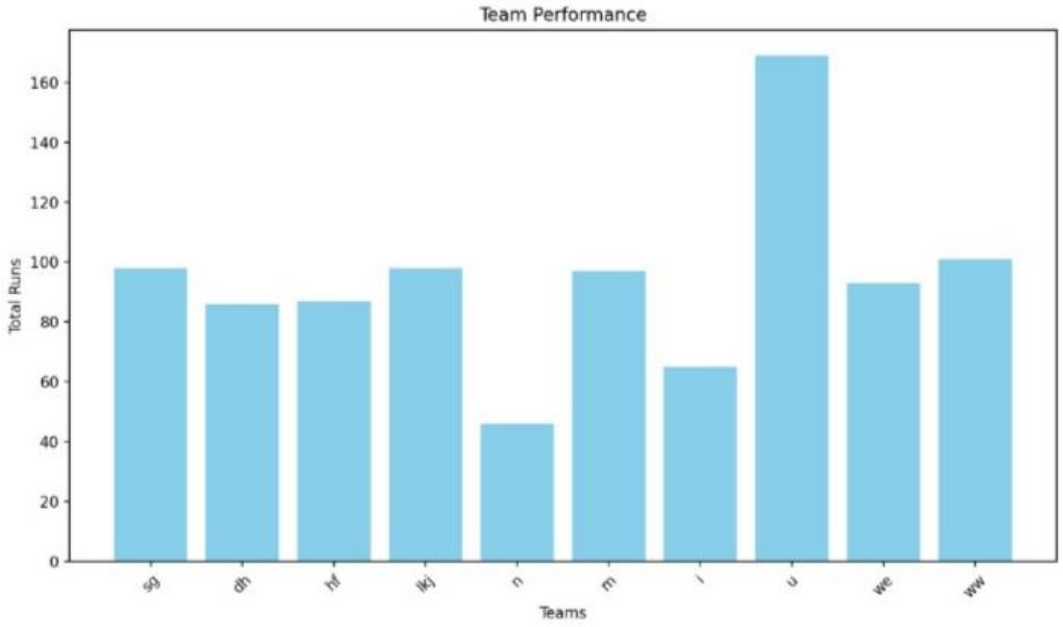
This algorithm outlines the functionality and flow of the given code, allowing users to interact with the Stream lit app to add, display, update, search, and delete match data, as well as visualize team performance through a graph.

OUTPUT


Match Results Tracker
Team 1
Team 2
Score 1
0
Score 2
0
Man of the Match
Highest Wicket Taker

Display Results

Team Performance Graph



Search for a match

Search

Select match to update:

sg vs dh

Team 1 - Update

sg

Team 2 - Update

dh

Score 1 - Update

98

Score 2 - Update

86

Score 1 - Update

98 − +

Score 2 - Update

86 − +

Man of the Match - Update

kjh

Highest Wicket Taker - Update

nmb

Update Match

Select match to delete:

sg vs dh

Delete Match

Conclusion

In conclusion, the Cricket Match Result Tracker developed using Python provides a robust and user-friendly solution for capturing, analysing, and interpreting cricket match data. Leveraging Python's versatile libraries and functionalities, the tracker efficiently handles data input, processing, visualization, and historical data management. With real-time updates, customizable reporting, and integration capabilities, it offers a comprehensive platform for cricket enthusiasts, analysts, and professionals to gain valuable insights into match outcomes, player performances, and strategic trends. By empowering users with actionable insights and facilitating informed decision-making, the Python-based Cricket Match Result Tracker enhances the understanding and appreciation of the game while providing a valuable resource for improving team performance and strategic planning.

REFERENCE

- https://www.bing.com/chat
- https://gemini.google.com/app
- https://chat.openai.com/