Beyond the numbers : A Deep Dive into Sports Performance Metrics

Data 228-11 : Project Report

Team name : Big Data Battalion

Deekshita Prakash Savanur (016597815)

Gouri Benni (016285698)

Uzair Riyaz Pachhapure (016285698)

Aboli Wankhade (016078998)

ABSTRACT:

Cricket is an extremely popular activity that is played all over the world. It requires a variety of abilities and tactics, and player performance is crucial to a team's success. In recent years, various performance indicators have been developed as a result of the growing curiosity in employing data analytics to more accurately assess cricket player performance. The project focuses on using big data and cloud computing to examine cricket players' metrics for performance. To extract useful insights and trends in the performance of cricket teams and players based on various matches in different series are calculated beyond the conventional measurements. The project will make use of data from a number of sources such as cricket player profiles and match statistics. The main goal of the project is to offer a thorough and in-depth study of the players' performance metrics that goes beyond the standard measures like batting average, bowling average, and strike rate. The analysis will explore numerous elements that influenced a player's performance, including the opposing team statistics and the game's circumstances.

Acknowledgements:

The mentorship provided by Professor Andrew H. Bond and Teaching Assistant, Venkatsai Deedkonda has been essential in enhancing our understanding of the subject and has undeniably played a vital role in the completion of this work. We are truly grateful for the opportunity to learn from such dedicated and inspiring educators.

Table of Contents

Chapter 1 Introduction

1.1 Project goals and objectives:

The primary objective of our project is to meticulously analyze the statistical data related to cricket teams and their players. By examining various performance metrics, we aim to uncover significant correlations and trends that impact a team's success and the development of individual players. The analysis is carried out using tools like Python and Tableau, and it employs multiple AWS services such as S3, Glue, and Redshift. To enhance our in-depth analysis, we intend to create an accessible website that displays the insights and results gathered from our investigation. This interactive platform will allow fans, experts, and other stakeholders to explore the intricacies of cricket performance, fostering a more informed and engaged community.

1.2 Problem and motivation:

The issue at hand is the insufficient integration of abundant cricket data for teams and players, making it challenging to identify essential performance factors and patterns. This knowledge gap hinders fans, coaches, and analysts from making well-informed decisions and deepening their involvement in the sport. Our goal is to utilize cutting-edge analytical tools and techniques to develop a user-friendly platform that facilitates exploration, visualization, and comprehension of cricket performance metrics, ultimately nurturing a better-informed and engaged cricket community.

1.3 Data description:

We have obtained our project data from CricSheet, an all-encompassing cricket data repository that includes information from 2003 to 2022. This wide-ranging dataset encompasses various aspects of the sport and features granular ball-by-ball data, allowing us to dive deeper into the nuances of cricket. By studying this expansive collection of data, we can discover patterns and tendencies that impact the performance of teams and individual players. Moreover, the

CricSheet dataset provides a solid base for our research, as it has been carefully curated and updated over time, ensuring the accuracy and dependability of our insights.

Dataset link: https://cricsheet.org/downloads/

1.4 Project results and deliverables:

The project aims to provide a user-friendly website with interactive visualizations and comprehensive analysis of cricket data. In-depth Data Examination and Insight Discovery and Detailed analysis of the stored cricket data uncovers trends, patterns, and crucial elements that influence team and player performance. These insights provide valuable information for cricket enthusiasts, athletes, coaches, and other stakeholders. Tableau Public is utilized to generate visually appealing, interactive visualizations and dashboards that showcase the analyzed data and insights in an easily digestible format. These visualizations allow users to explore and customize their view of the data, catering to individual interests.

CHAPTER 2 Project Background and Related Work

2.1 Background and Technologies used

The foundation of the Cricket Analytics project lies in the growing interest and necessity for data-based insights within the realm of sports. As one of the world's most popular sports, cricket produces extensive data through individual performance, team statistics, and match outcomes. This abundance of information presents an opportunity to extract valuable knowledge that can benefit enthusiasts, athletes, coaches, and other involved parties in gaining a deeper understanding of the game and making well-informed choices.

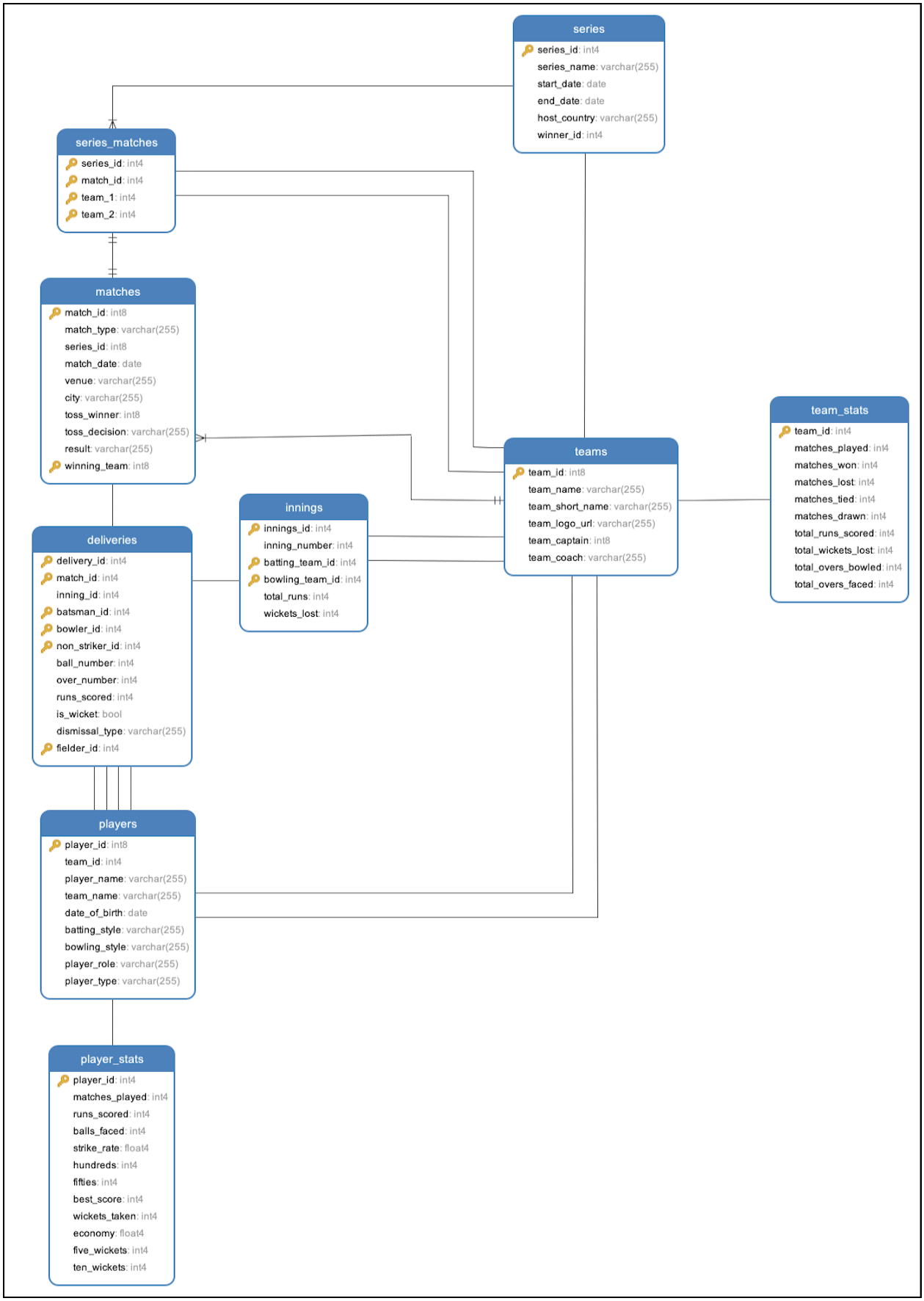
In the past, cricket analysis primarily depended on simple metrics such as averages and strike rates. However, recent progress in technology, data analytics, and machine learning has paved the way for more sophisticated analysis and predictive models. The goal of the Cricket Analytics project is to tap into this potential by developing an all-encompassing platform that merges data gathering, examination, visualization, and forecasting capabilities.

Tools used throughout the project:

NAME	USAGE
Python	Data Cleaning and Data Transforming
AWS S3	Storing raw data
AWS Glue	ETL Process
AWS Redshift	Analytical Queries
Tableau	Diverse Big Data Analytical Results

2.2 ER Diagram or Data Model:

A data model offers a structured way to represent data elements, their connections, and limitations. It is crucial for designing databases, organizing data, and establishing a plan for effective data management. In cricket analysis, a data model is vital for categorizing various statistics, performance measures, and match information to detect trends, gain insights, and make well-informed decisions.


series
series_id: int4
series_name: varchar(255)
start_date: date
end_date: date
host_country: varchar(255)
winner_id: int4
series_matches
series_id: int4
match_id: int4
team_1: int4
team_2: int4
matches
match_id: int8
match_type: varchar(255)
series_id: int8
match_date: date
venue: varchar(255)
city: varchar(255)
toss_winner: int8
toss_decision: varchar(255)
result: varchar(255)
winning_team: int8
teams
team_id: int8
team_name: varchar(255)
team_short_name: varchar(255)
team_logo_url: varchar(255)
team_captain: int8
team_coach: varchar(255)
team_stats
team_id: int4
matches_played: int4
matches_won: int4
matches_lost: int4
matches_tied: int4
matches_drawn: int4
total_runs_scored: int4
total_wickets_lost: int4
total_overs_bowled: int4
total_overs_faced: int4
innings
innings_id: int4
inning_number: int4
batting_team_id: int4
bowling_team_id: int4
total_runs: int4
wickets_lost: int4
deliveries
delivery_id: int4
match_id: int4
inning_id: int4
batsman_id: int4
bowler_id: int4
non_striker_id: int4
ball_number: int4
over_number: int4
runs_scored: int4
is_wicket: bool
dismissal_type: varchar(255)
fielder_id: int4
players
player_id: int8
team_id: int4
player_name: varchar(255)
team_name: varchar(255)
date_of_birth: date
batting_style: varchar(255)
bowling_style: varchar(255)
player_role: varchar(255)
player_type: varchar(255)
player_stats
player_id: int4
matches_played: int4
runs_scored: int4
balls_faced: int4
strike_rate: float4
hundreds: int4
fifties: int4
best_score: int4
wickets_taken: int4
economy: float4
five_wickets: int4
ten_wickets: int4

Our data model consists of 9 tables, each focusing on distinct aspects of cricket games, teams, players, and related statistics. Each table has primary keys (unique identifiers for each entry) and foreign keys (references to primary keys in other tables) that create relationships and preserve referential integrity between tables.

- Players table holds data about individual cricket players, comprising unique player_id, team_id, player_name, team_name, date_of_birth, batting_style, bowling_style, player_role, and player_type.
- Teams table stores details about cricket teams, including team_id, team_name, team_short_name, team_logo_url, team_captain (player_id), and team_coach.
- Matches table records information related to individual cricket matches, such as match_date, venue, city, toss_winner (team_id), toss_decision, result, and winning_team (team_id).
- Innings table captures data about each innings in a cricket match, encompassing innings_id, inning_number, batting_team_id, bowling_team_id, total_runs, and wickets_lost.
- The Delivery table contains information about every delivery (ball) in a cricket match.
- Player stats table holds individual player performance data, including matches_played, runs_scored, balls_faced, strike_rate, hundreds, fifties, best_score, wickets_taken, economy, five_wickets, and ten_wickets.
- Team_stats table records team-level performance data, such as team_id, matches_played, matches_won, matches_lost, matches_tied, matches_drawn, total_runs_scored, total_wickets_lost, total_overs_bowled, and total_overs_faced.
- Series table stores information about cricket series, containing series_id, series_name, start_date, end_date, host_country, and winner_id (team_id).
- Series_matches table links cricket matches to their respective series. It comprises series_id, match_id, team_1 (team_id), and team_2 (team_id).

Each table includes primary keys, foreign key constraints for ensuring referential integrity, and unique constraints to avoid duplicate entries. The data model is structured to accommodate a wide range of information about cricket matches, teams, players, and their achievements across various series.

2.3 Literature Survey:

The research paper titled "Exploring the Impact of Team-Centric Causal Attributions and Team Performance on Group Confidence: A Hierarchical Analysis" delves into the effects of team-focused causal explanations and overall team performance on the shared belief in a team's capabilities. By utilizing a hierarchical analytical method, the study examines the connections between these elements at both individual and team tiers. This research emphasizes the significance of comprehending team members' perspectives on the reasons behind their team's triumphs and setbacks, as well as the influence of the team's performance on their collective confidence in their ability to excel as a unit. By investigating these associations, the study offers valuable contributions to the domains of sports psychology and team cohesion, suggesting potential approaches to improve team performance and cultivate a more robust sense of shared confidence within team members.

Chapter 3 System Requirements and Analysis

3.1 Domain and Business requirements

Domain: The domain requirements involve acquiring a thorough understanding of the sport's unique complexities, regulations, and the organization of various cricket competitions. Additionally, it includes grasping common cricket data formats and metrics, such as batting averages, strike rates, and bowling economy rates. This knowledge is crucial for creating an analytics platform customized to address the distinct aspects of cricket.

Business: The business requirements revolve around the broader goals and desired results that the sports analytics platform aims to accomplish for the organization. These requirements will

encompass offering insights to improve team performance, boosting fan engagement via interactive visualizations, equipping decision-makers with comprehensive data analysis, and securing a competitive advantage through advanced sports analytics.

3.2 Customer-oriented requirements

Meeting user-focused requirements is essential to provide a seamless, informative, and engaging experience for a diverse group of users, including team administrators, coaches, players, analysts, and enthusiasts. These requirements aim to cater to the specific needs and preferences of the intended audience.

Key user-focused requirements for this project are:

User-friendly interface: Developed an easy-to-use and navigable interface, enabling users to efficiently access pertinent data, visualizations, and insights.

Customization options: Allowed users to personalize their experience by choosing their favorite teams, players, or match types, displaying customized content and suggestions.

Engaging visualizations: Presented interactive and visually captivating data representations that help users gain a deeper comprehension of cricket dynamics.

Safe and secure experience: Implemented strong user authentication and data security measures, such as single sign-on (SSO), to guarantee a safe user experience.

3.3 System Function requirements:

Data acquisition and storage: The system will adeptly import and store large quantities of data from various sources, like cricsheet, while maintaining a structured and easily accessible format. It will also safeguard data integrity and reduce the chances of data loss or damage.

Data manipulation and conversion: The platform will be proficient in carrying out intricate ETL (Extract, Transform, Load) operations, transforming raw data into an appropriate format for analysis and visualization. Additionally, it will integrate seamlessly with services such as Amazon Redshift and AWS Glue.

Sophisticated analytics: Incorporated powerful analytical tools and techniques to extract significant insights from the data, enabling better decision-making for teams, coaches, and players.

User administration: The platform offers user authentication and access management features, including single sign-on (SSO), to ensure only authorized users can access the system and its resources.

3.4 System Behaviour Requirements

System Behavior Requirements outline the expected performance and interaction of a system with its environment, guaranteeing it satisfies user needs and adheres to vital quality standards. In the context of the Sports Analytics project, these requirements are essential for delivering a smooth user experience and consistent system functionality.

Key system behavior requirements for this Sports Analytics project encompass:

Promptness: The system will rapidly process requests and execute actions, offering users instant data analysis and visualization with minimal waiting time.

Dependability: The platform will be sturdy and trustworthy, ensuring data accuracy, consistency, and availability for users. It should also reduce the likelihood of system failures or downtime.

User-friendliness: The system is easy to navigate and interact with, catering to users with varying levels of expertise and providing clear instructions, tooltips, and support resources when necessary.

3.5 System Performance and non-functional requirements

Scalability: The system will be able to accommodate increased data volume, user traffic, and resource demands without compromising its performance or responsiveness. It is designed with the ability to grow and adapt to the evolving needs of the sports analytics domain.

Responsiveness: The system will provide fast and efficient processing of user requests, delivering real-time data analysis and visualization with minimal latency. It will optimize data retrieval, processing, and display to offer users a seamless experience.

3.6 Technology and Resource requirement

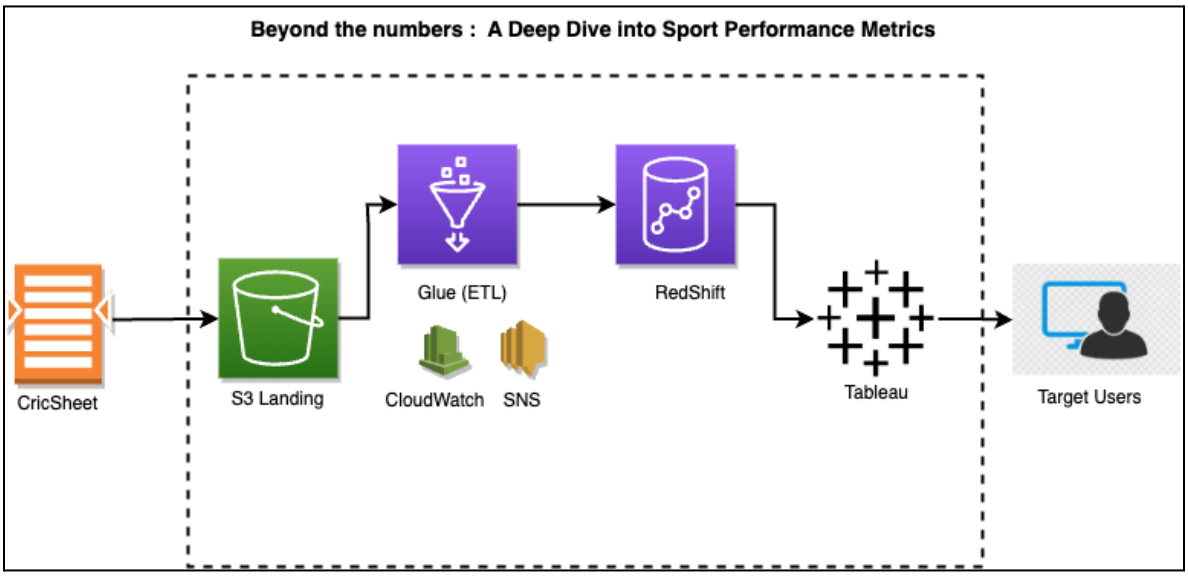
Hardware: Suitable server infrastructure is there for hosting the application, managing computational resources, and storing the data. The hardware infrastructure will be adaptable, allowing for seamless growth as the project expands.

Software: The system has various software components, including database management systems (for example, Amazon Redshift), ETL tools (such as AWS Glue), web servers, and application development frameworks (like Django).

Analytics and visualization tools: This will offer users valuable insights and engaging visualizations, the system will integrate advanced analytics tools (such as Tableau) capable of processing and presenting data in an accessible format.

Chapter 4 System Design

4.1 System architecture design



Overall, the architecture is built to offer customers a secure, scalable, and flexible system that makes it simple for them to obtain and analyze cricket player performance indicators. For Data Source, the project's main data source is the Cricsheet website, which offers JSON files with

statistics on cricket players. To be used for future processing and storage, the data is placed into an Amazon S3 bucket. Using AWS Glue to transfer the data from S3 to Amazon Redshift, an ETL process is carried out to convert the JSON files into a format appropriate for analysis. Amazon Redshift is used as the project's data warehouse. It has nine tables that house all of the analysis-related columns. Tableau Public is used to produce interactive dashboards and visualizations that let users explore and examine data kept in Amazon Redshift.

4.2. System data and database design

Our project Cricket Analytics utilizes data from the Cricsheet website and includes an ETL workflow, database structure, and data visualization with Tableau Public and Django, here's a description of the system data and database design:

System Data Components:

Data Origin: Cricket data files obtained from the Cricsheet website (in JSON format)

Data Storage: AWS S3 bucket employed for holding raw data files

ETL Workflow: Python code to execute the ETL process utilizing AWS Glue

Data Repository: Amazon Redshift for retaining transformed data

Database Design:

Table 1: *Matches* - Holds data about individual matches, including match ID, date, location, participating teams, and match outcome.

Table 2: Players - Stores players' data, such as player ID, name, team, date of birth, batting style, and bowling style.

Table 3: Innings - Includes detailed innings id, innings number, batting team, bowling team, total runs scored and wickets lost.

Table 4: Match Innings - This table contains inning id, match id and team batted.

Table 5: Series - This is the main table that contains detailed information of series id, series name, match id, team names who played in the match, match name, year of the match/series,

series start date and series end date, match type, toss winner, winning team and venue of the match.

Table 6: Deliveries - This table contains the detailed ball-to-ball data of the matches. The unique delivery id for each delivery, match id, inning id, batsman name, bowler name, non-striker name, ball number, over number, runs scored, extra runs, wicket status, dismissed player, and mode of dismissal and fielder name.

Table 7: Series matches - This contains the particular series id and match id along with the team names that played in that particular match.

Table 8: Series - Includes the series id, series name, season in year, match id, series start date, series end date and winning team.

Table 9: Teams - Includes unique team id for each team, team name, team short name, team logo, team captain, and team coach.

Project Workflow:

1. Acquire data from Cricsheet and store it in an S3 bucket.
2. Use Python code and AWS Glue for transforming JSON files and loading data into Amazon Redshift.
3. Run SQL scripts to establish required tables in Amazon Redshift.
4. Employ Tableau Public for crafting visualizations and interactive dashboards using data from Amazon Redshift.
5. Build a Django-based website that integrates Tableau dashboards and enables user authentication with SSO. Host the website on an AWS EC2 instance.

This layout offers a complete system for examining cricket data, presenting insights visually, and sharing them via an interactive web platform.

4.3 System design problems, solutions, and patterns

In terms of scalability, the system should be able to grow as data volumes increase and be built to handle enormous amounts of data efficiently. Utilizing tools built for handling large-scale data processing and storage, such as Amazon S3, AWS Glue, and Amazon Redshift helped with this. In terms of processing of data, the system should be capable of quickly processing massive amounts of data and converting it to the necessary format. This is done by ETL solutions like AWS Glue, which can automate and improve the efficiency of the data transformation process. In terms of availability, the system is built to guarantee that data is always accessible and that any downtime is kept to a minimum. This is done by Amazon EC2, which offers web applications great availability and scalability. In terms of performance, the system is built to ensure that data is handled fast and efficiently and supplied to end users. Amazon Redshift, which is designed to process massive volumes of data quickly, will help with this. In terms of data visualization, the system is created to offer end users the ability to see data in an understandable manner. Tableau, which offers strong data visualization capabilities and is linked with AWS services like Amazon Redshift, is used to accomplish this.

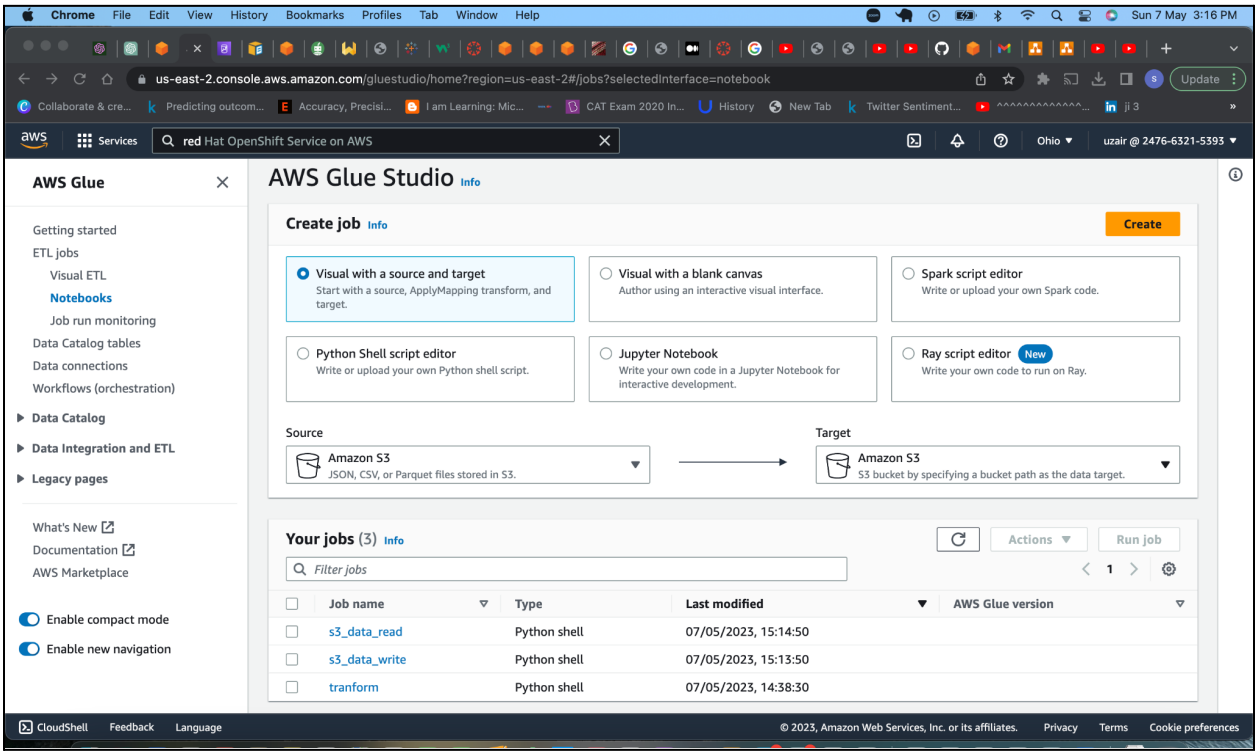
Chapter 5 System Implementation

5.1 System Implementation Summary:

Cricsheet is a highly-regarded and all-inclusive cricket data repository, offering detailed and current match data. This makes it an excellent asset for sports analysts, and data-driven applications alike. We have harnessed cricsheet data to our project which has uniform information encompassing a variety of cricket match aspects, such as player achievements, team rankings, game outcomes, and others. We are utilizing Amazon Web Services (AWS) to handle and store the cricsheet data effectively. We are achieving this by directly transferring the data source into an Amazon S3 (Simple Storage Service) bucket.

5.1.1 ETL Process:

In AWS Glue, data transformation is a crucial component of the ETL (Extract, Transform, Load) process. This phase involves processing and converting raw data into the desired format before transferring it to the target storage, such as a database or data warehouse.



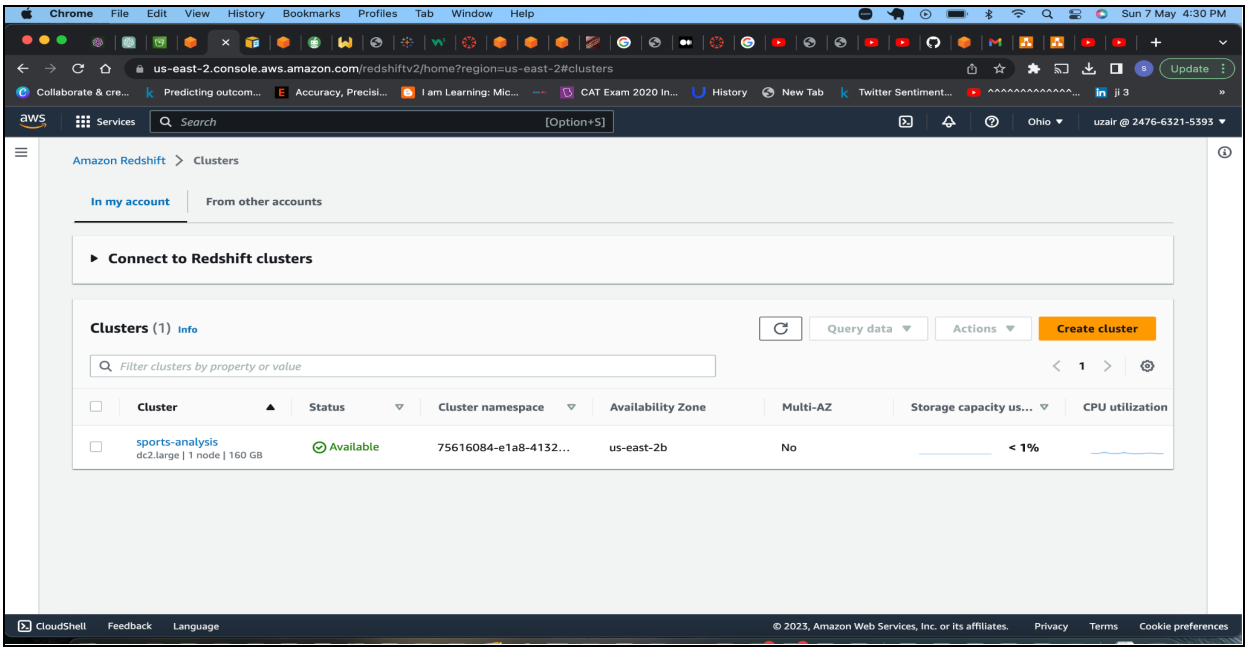
We started with setting up a Glue job using the AWS Management Console and we configured the job, specified the source and target data repositories, and the programming language (Python or Scala) to be employed for the ETL script. Then, AWS Glue is connected to source data from Amazon S3. Created a connection to our data source to directly access data using the AWS Glue DynamicFrame API.

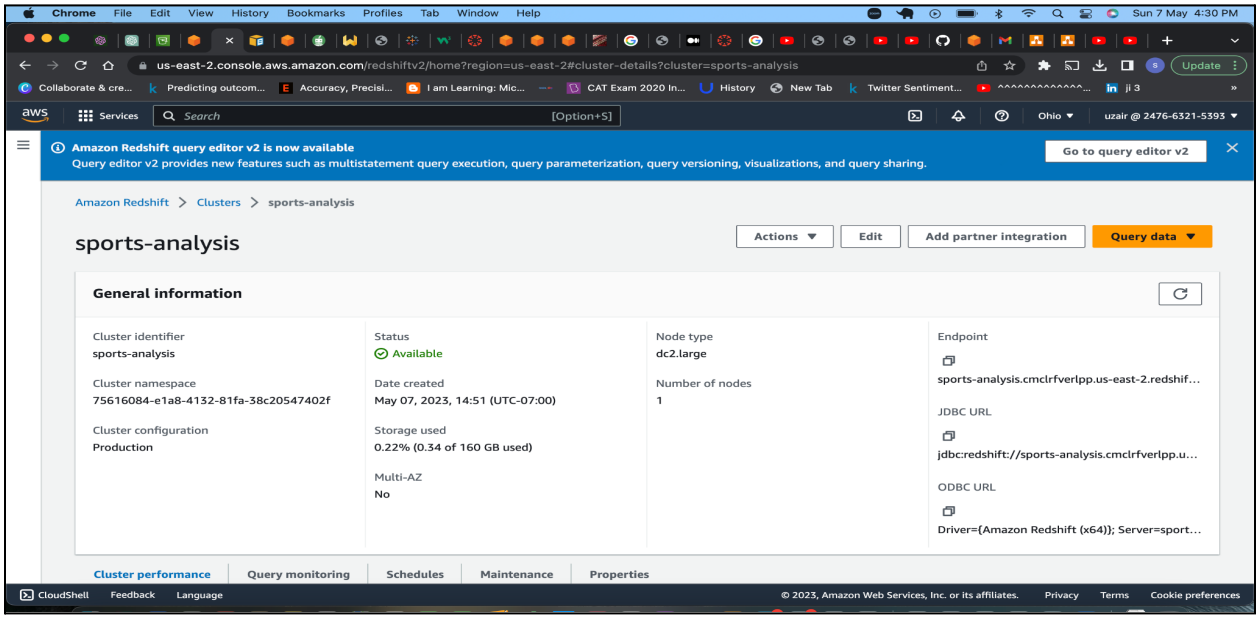
In Data transformation, we did several operations that will alter the data to meet our specific needs. First we did Filtering where we eliminated all unneeded rows based on required conditions. Then, we flattened the nested data where we used the Relationalize transformation available in AWS Glue to convert the nested structure into a relational format which is available in the aws glue.transforms package, to the nested data within the DynamicFrame. We provided

two parameters: a unique identifier for the root table and the DynamicFrame containing the nested information. The transformation will examine the nested structure and generate 9 new tables for each level of nesting. These new tables will preserve primary and foreign key connections to retain the relationships between the original nested data elements. This process flattens the nested JSON data and creates multiple tables with primary and foreign key relationships. Throughout the ETL process, we had to enhance or clean our data to improve the quality and incorporate valuable information. This involved eliminating null values, duplicates, or outliers, as well as adding new columns with computed values or data from other sources. AWS Glue automatically deduces the schema of our source data. However, we have made adjustments or rectified the schema if there are discrepancies or conflicts. Lastly, we transferred the transformed data into the target destination, Amazon Redshift, which is a supported data store. AWS Glue efficiently manages the loading process and can also partition the data for optimized performance.

5.1.2 Creating data model in Amazon Redshift Query Editor-

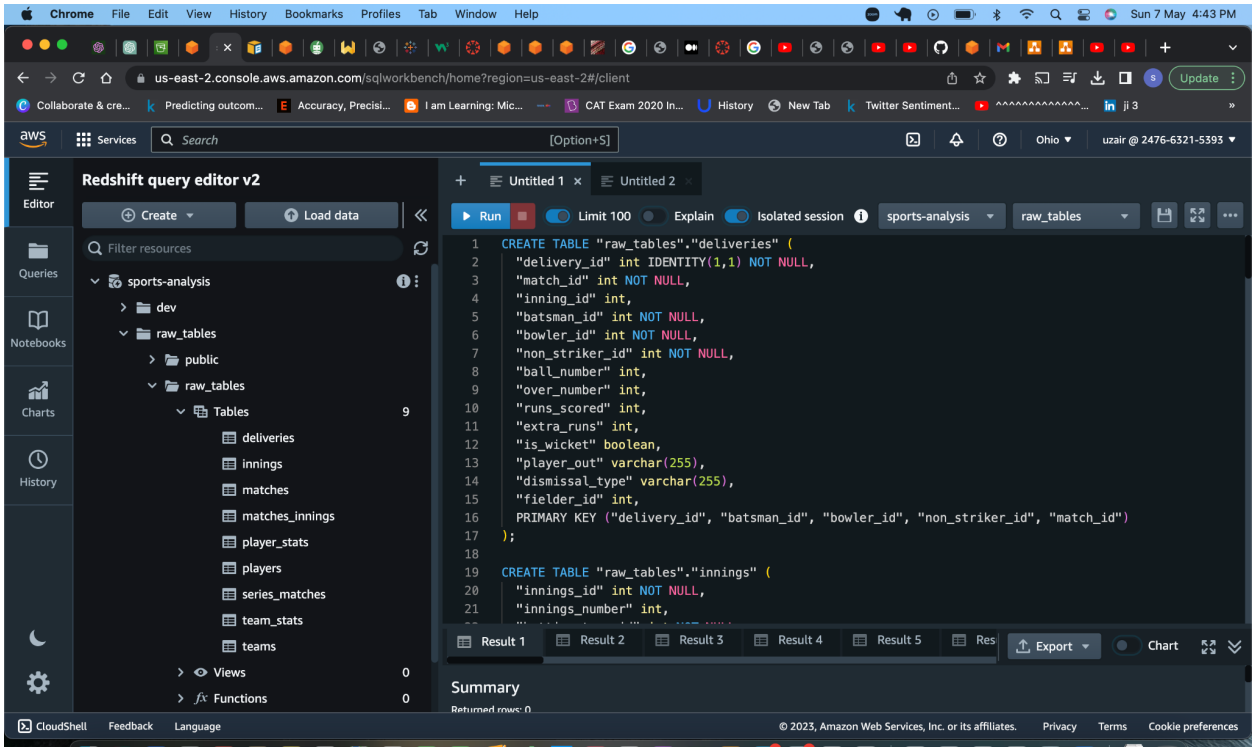
We set up an Amazon Redshift cluster by entering the essential data, including the cluster identification, node type, number of nodes, and the master user's login details.





Redshift's Query Editor was used to develop a data model for cricket once the cluster had been set up. The data model is created by using SQL script in Amazon Redshift. The table structure, including column names and data types, was specified by the SQL scripts. The create TABLE

SQL statement was used for building the tables.



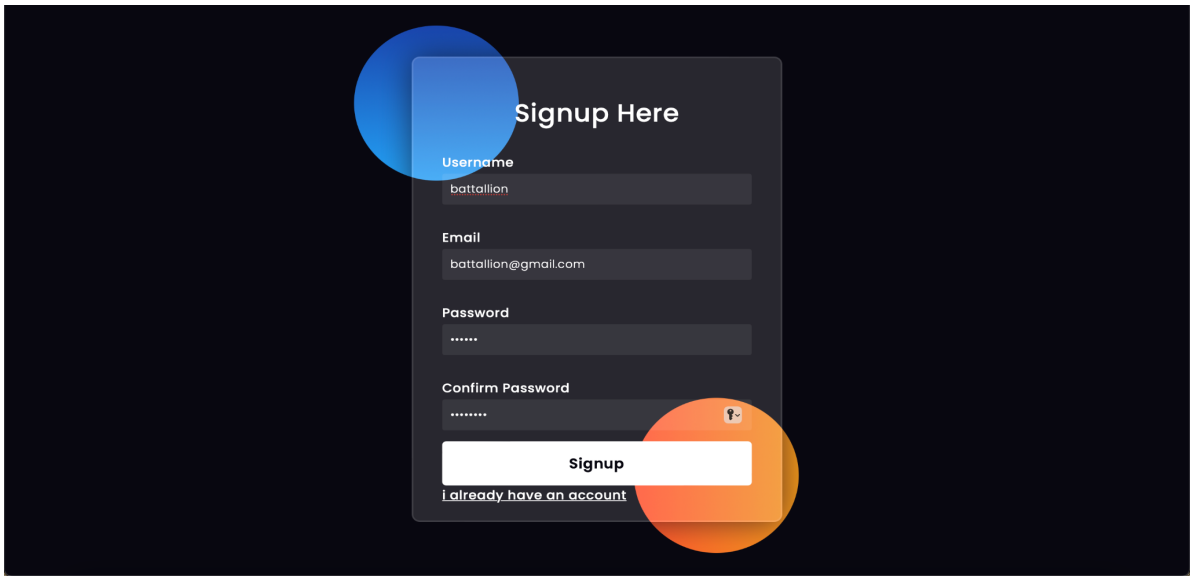
All of the tables we created using Amazon Redshift are displayed in the screenshot above. We created tables for deliveries, innings, teams, matches, players, and series. At the end, all foreign key constraints are provided. Tables are connected to one another by relationships.

Web Application

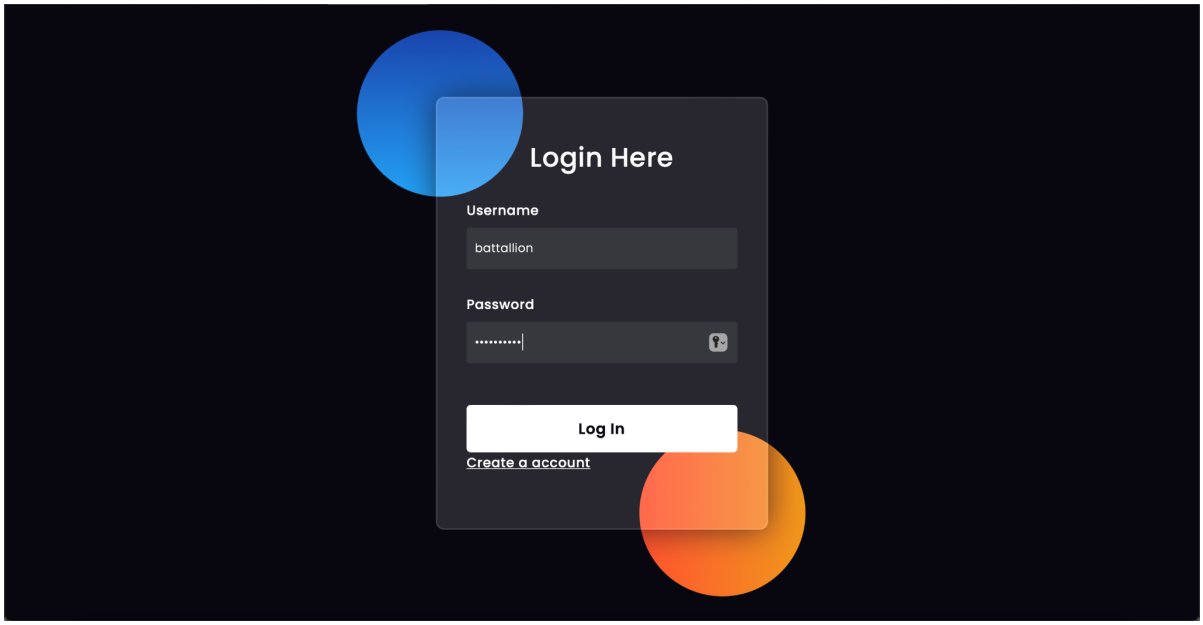
Our website for sports analytics was built using the Django web framework and includes multiple pages like Series, Players, Matches, and Visualizations. Each page shows the information that will be fetched from the backend. Our website allows users to easily access the cricket data such as match winners, series match winners, Player statistics etc. The series tab provides the information on cricket series such as the series name, series start date, series end date, venue and series winner team. The player tab shows the players statistics where users will be able to see the player performance in different series or matches. Finally the visualization tab will provide some graphs of each series and each match.

For the security of our website we have created an SSO page that helps users to create login credentials using username, email id and password. After creating the login details, the user can login in to access the information. We have hosted our web application.

Website sign up page (SSO) -

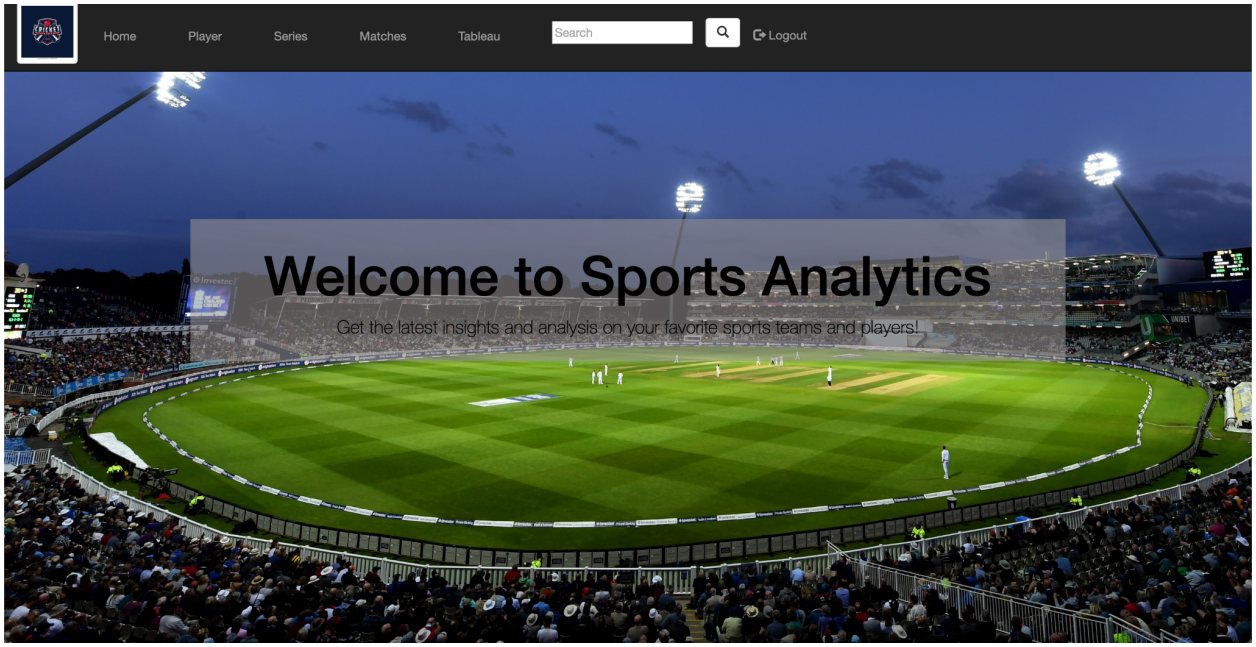


The above image is the signup page where users can sign up choosing username, email and password.



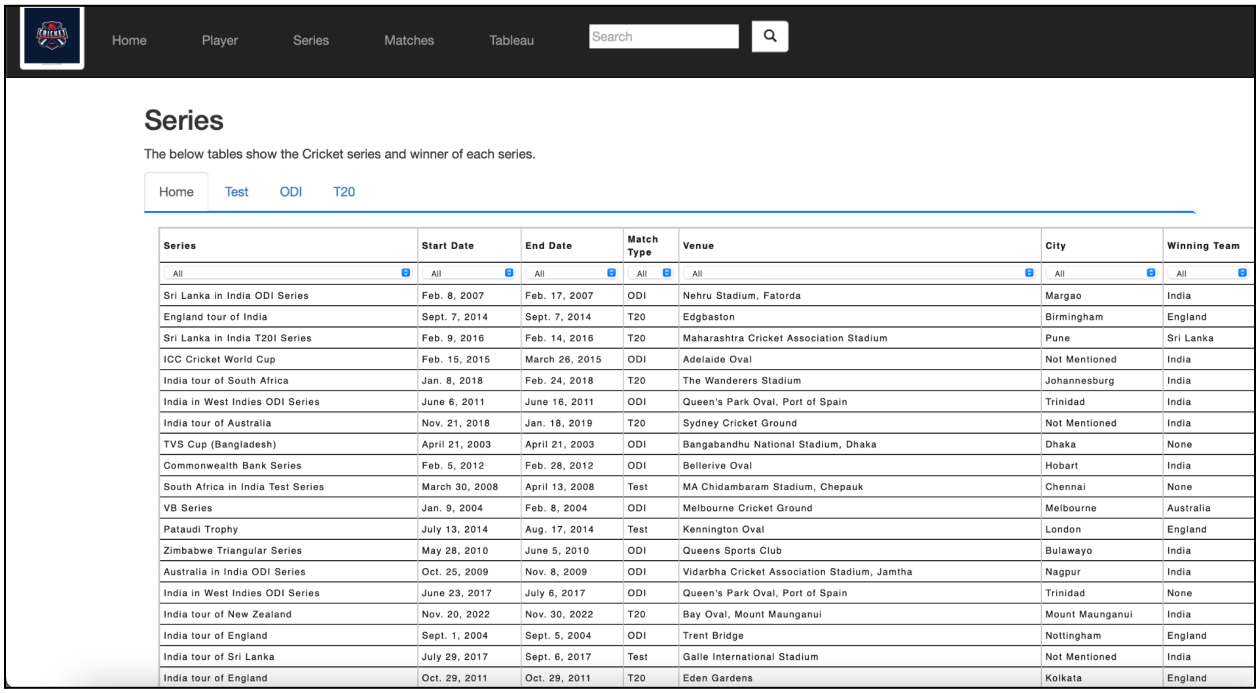
The above image is the login page where users can login to their account.

Website Home Page -



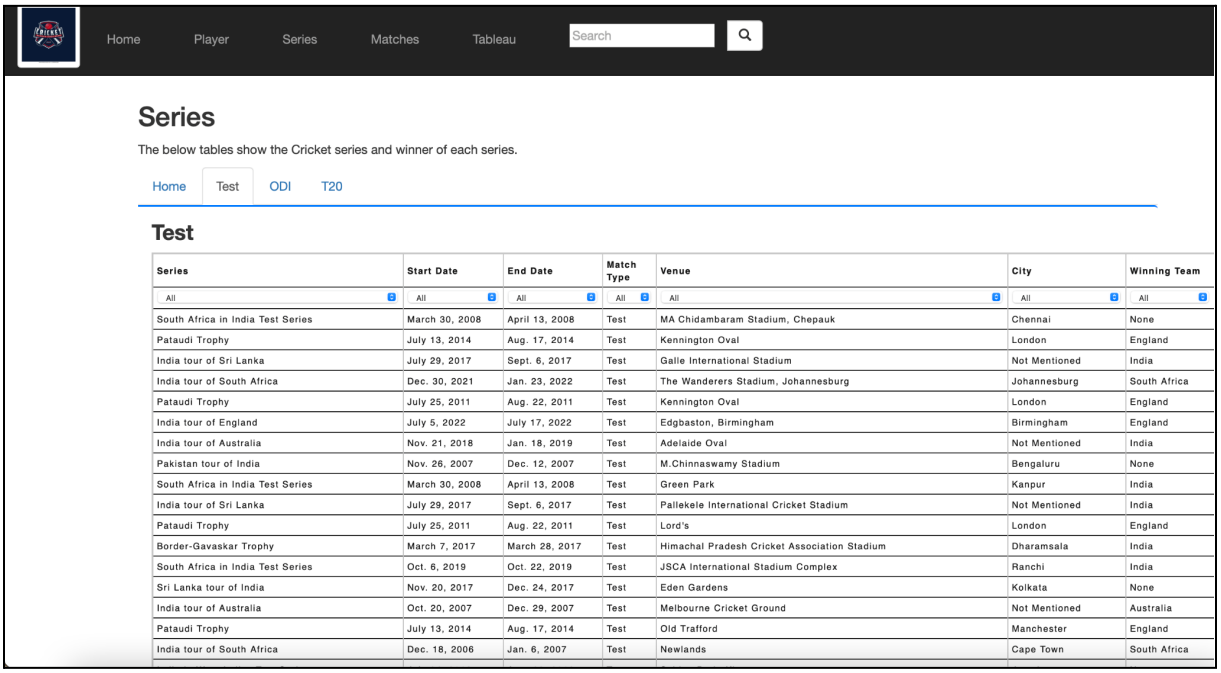
The above image is our website home page where we have a navigation bar. In the navigation bar we have multiple tabs such as home, player, series, matches and visualization.

Website Series Tab -



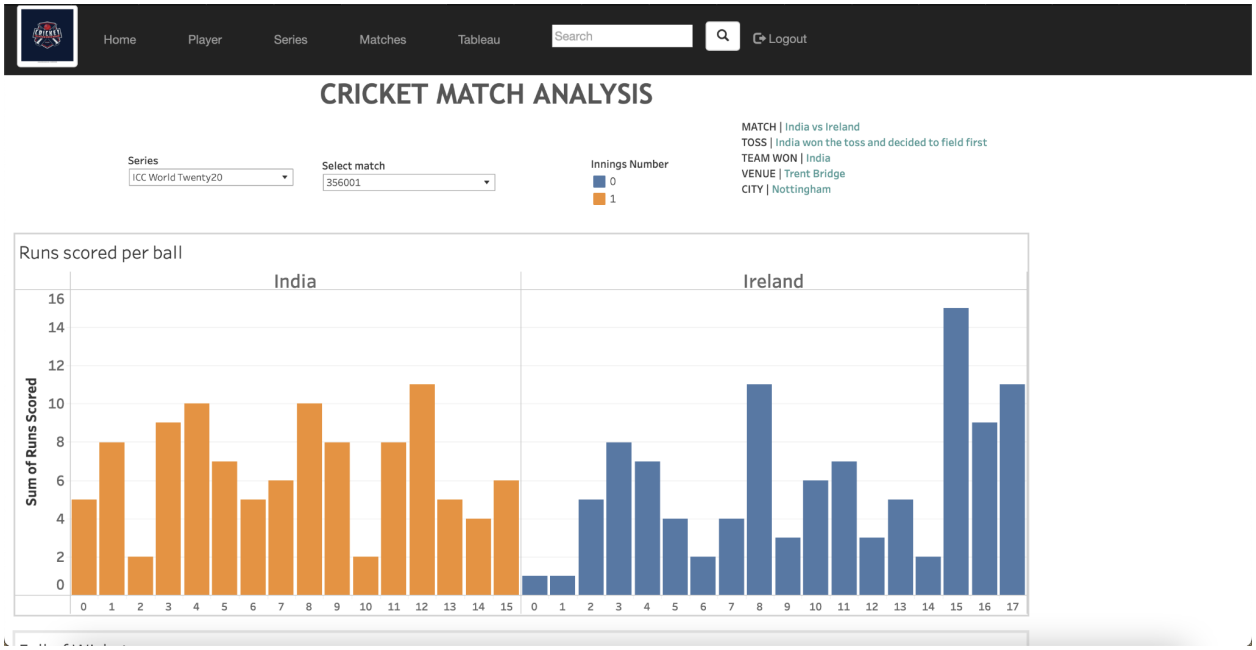
Series	Start Date	End Date	Match Type	Venue	City	Winning Team
All	All	All	All	All	All	All
Sri Lanka in India ODI Series	Feb. 8, 2007	Feb. 17, 2007	ODI	Nehru Stadium, Fatorda	Margao	India
England tour of India	Sept. 7, 2014	Sept. 7, 2014	T20	Edgbaston	Birmingham	England
Sri Lanka in India T20I Series	Feb. 9, 2016	Feb. 14, 2016	T20	Maharashtra Cricket Association Stadium	Pune	Sri Lanka
ICC Cricket World Cup	Feb. 15, 2015	March 26, 2015	ODI	Adelaide Oval	Not Mentioned	India
India tour of South Africa	Jan. 8, 2018	Feb. 24, 2018	T20	The Wanderers Stadium	Johannesburg	India
India in West Indies ODI Series	June 6, 2011	June 16, 2011	ODI	Queen's Park Oval, Port of Spain	Trinidad	India
India tour of Australia	Nov. 21, 2018	Jan. 18, 2019	T20	Sydney Cricket Ground	Not Mentioned	India
TVS Cup (Bangladesh)	April 21, 2003	April 21, 2003	ODI	Bangabandhu National Stadium, Dhaka	Dhaka	None
Commonwealth Bank Series	Feb. 5, 2012	Feb. 28, 2012	ODI	Bellerive Oval	Hobart	India
South Africa in India Test Series	March 30, 2008	April 13, 2008	Test	MA Chidambaram Stadium, Chepauk	Chennai	None
VB Series	Jan. 9, 2004	Feb. 8, 2004	ODI	Melbourne Cricket Ground	Melbourne	Australia
Pataudi Trophy	July 13, 2014	Aug. 17, 2014	Test	Kennington Oval	London	England
Zimbabwe Triangular Series	May 28, 2010	June 5, 2010	ODI	Queens Sports Club	Bulawayo	India
Australia in India ODI Series	Oct. 25, 2009	Nov. 8, 2009	ODI	Vidarbha Cricket Association Stadium, Jamtha	Nagpur	India
India in West Indies ODI Series	June 23, 2017	July 6, 2017	ODI	Queen's Park Oval, Port of Spain	Trinidad	None
India tour of New Zealand	Nov. 20, 2022	Nov. 30, 2022	T20	Bay Oval, Mount Maunganui	Mount Maunganui	India
India tour of England	Sept. 1, 2004	Sept. 5, 2004	ODI	Trent Bridge	Nottingham	England
India tour of Sri Lanka	July 29, 2017	Sept. 6, 2017	Test	Galle International Stadium	Not Mentioned	India
India tour of England	Oct. 29, 2011	Oct. 29, 2011	T20	Eden Gardens	Kolkata	England

In the series tab we have fetched the data from the backend where we have shown the tables that contain Series, start date, end date, match type, venue, city and winning team. In the same page we have put sub tabs where tables of different match types are shown. We have also added filters in the tables.



Series	Start Date	End Date	Match Type	Venue	City	Winning Team
All	All	All	All	All	All	All
South Africa in India Test Series	March 30, 2008	April 13, 2008	Test	MA Chidambaram Stadium, Chepauk	Chennai	None
Pataudi Trophy	July 13, 2014	Aug. 17, 2014	Test	Kennington Oval	London	England
India tour of Sri Lanka	July 29, 2017	Sept. 6, 2017	Test	Galle International Stadium	Not Mentioned	India
India tour of South Africa	Dec. 30, 2021	Jan. 23, 2022	Test	The Wanderers Stadium, Johannesburg	Johannesburg	South Africa
Pataudi Trophy	July 25, 2011	Aug. 22, 2011	Test	Kennington Oval	London	England
India tour of England	July 5, 2022	July 17, 2022	Test	Edgbaston, Birmingham	Birmingham	England
India tour of Australia	Nov. 21, 2018	Jan. 18, 2019	Test	Adelaide Oval	Not Mentioned	India
Pakistan tour of India	Nov. 26, 2007	Dec. 12, 2007	Test	M.Chinnaswamy Stadium	Bengaluru	None
South Africa in India Test Series	March 30, 2008	April 13, 2008	Test	Green Park	Kanpur	India
India tour of Sri Lanka	July 29, 2017	Sept. 6, 2017	Test	Pallekele International Cricket Stadium	Not Mentioned	India
Pataudi Trophy	July 25, 2011	Aug. 22, 2011	Test	Lord's	London	England
Border-Gavaskar Trophy	March 7, 2017	March 28, 2017	Test	Himachal Pradesh Cricket Association Stadium	Dharamsala	India
South Africa in India Test Series	Oct. 6, 2019	Oct. 22, 2019	Test	JSCA International Stadium Complex	Ranchi	India
Sri Lanka tour of India	Nov. 20, 2017	Dec. 24, 2017	Test	Eden Gardens	Kolkata	None
India tour of Australia	Oct. 20, 2007	Dec. 29, 2007	Test	Melbourne Cricket Ground	Not Mentioned	Australia
Pataudi Trophy	July 13, 2014	Aug. 17, 2014	Test	Old Trafford	Manchester	England
India tour of South Africa	Dec. 18, 2006	Jan. 6, 2007	Test	Newlands	Cape Town	South Africa

When navigated to the Series Tab, the above table will be displayed when we select the Test subtab, where you can filter out Series, Start Date, End Date, Match Type, Venue, and City.



When navigated to the Tableau tab in the navigation bar, the above dashboard is displayed which describes various cricket matches with filter options, where you can select the series name and match and the hue describes the Innings number. After selecting the options, brief description is displayed.

Chapter 6 Visualization

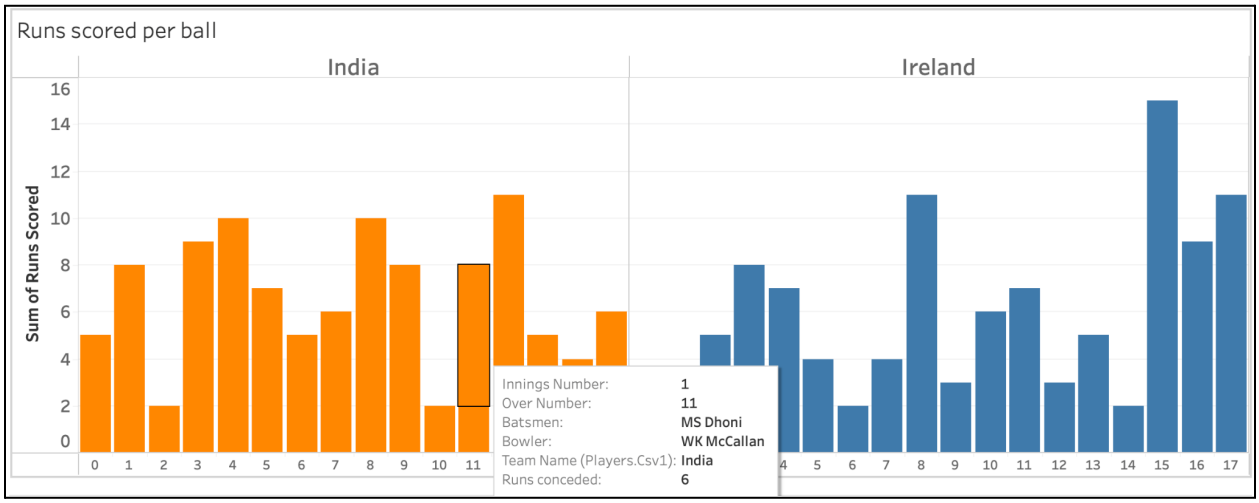
6.1 Data Visualization Using Tableau:

The processed information was subsequently integrated into Tableau Desktop via Redshift drivers for enhanced visualization. Tableau Desktop serves as a data visualization and analytics platform that facilitates the creation of interactive dashboards and promotes data-driven storytelling. By utilizing Tableau, we were able to delve deeper into the pet adoption dataset and elevate our analysis. The findings from the Tableau-based examination are discussed in the following sections.

6.2 Tableau Desktop connecting with Redshift:

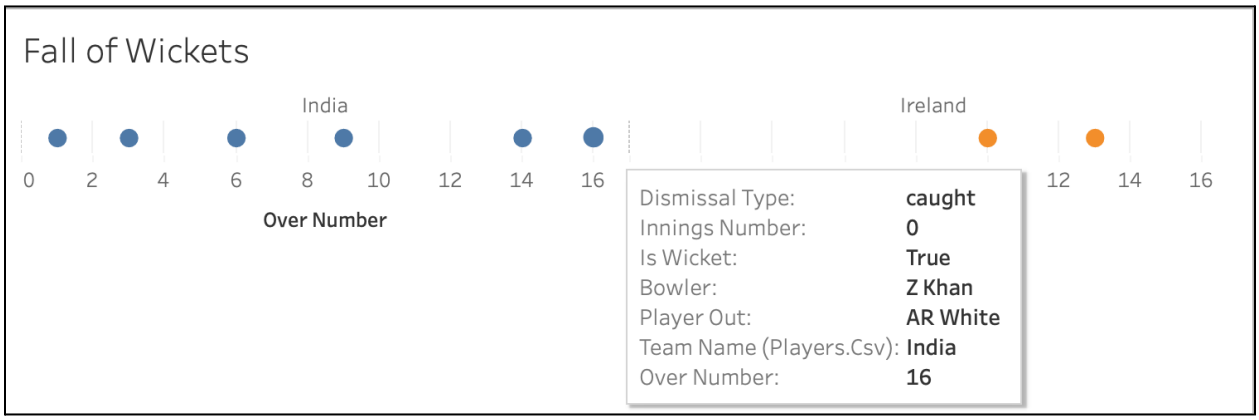
Our dataset was successfully linked to AWS Redshift through Tableau's connectors, establishing a visualization pipeline and enabling Tableau to directly access data from Redshift clusters. This connection is depicted in Figure 11 below. We incorporated several datasets by specifying the VPC subnet ID as the host and supplying user information.

Insight 1: Runs scored per ball



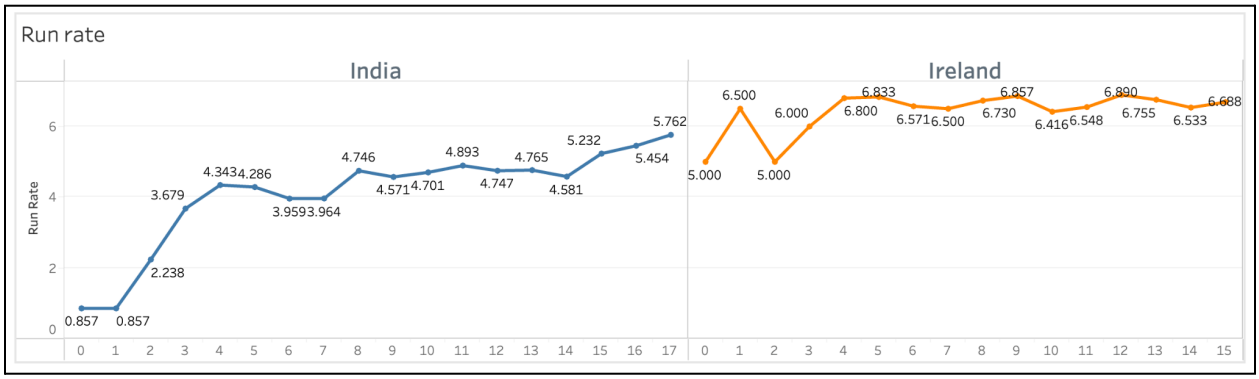
In this visualization we have shown the series name where we can select the series from the dropdown and the corresponding matches that happened in that series from, which we can select from the select match dropdown. Once the selection has been made we can see the match details like the teams played, toss winner, winning team and the venue of the match. The bar chart shows the runs scored by each team against total overs according to the innings number. In the label we can see the innings number, over number, batsman name, batting team, bowler name and the runs conceded in the particular over.

Insight 2: **Fall of wickets**



In this graph we can see the fall of wickets of each team against the over number. Here blue is represented by team India and orange is represented by team Ireland. In the label we can see the mode of dismissal, innings number, the wicket status, bowler name, batsman that got out, team name and the over number.

Insight 3: **Run rate**



In this line graph we can see the run rate pattern of each team against the over number. Run rate is cumulative runs scored per over. For instance, in Ireland innings 5 runs were scored in the first over hence the run rate was 5. In the second over, 8 runs were scored, hence the run rate increased from 5 to 6.50. In Tableau we took running total of runs as progressing through the innings and created a calculated field to calculate the run rate.

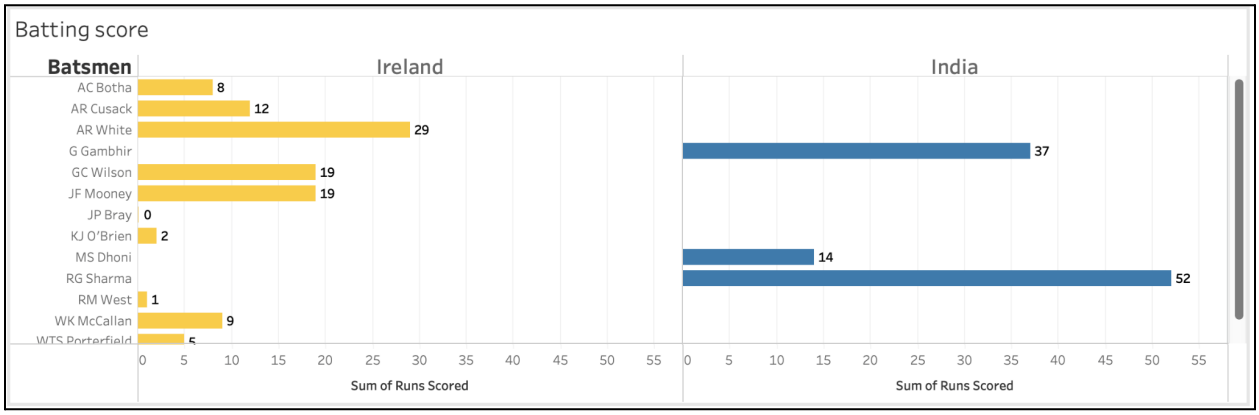
Insight 4: Bowler scoresheet

Bowler scoresheet

Bowler	Overs bowled India	Runs conceded India	Wickets India	Economy India	Overs bowled Ireland	Runs conceded Ireland	Wickets Ireland	Economy Ireland
AR Cusack					2	13	0	6.50
Harbhajan Sin..	4	25	1	6.25				
I Sharma	3	18	0	6.00				
IK Pathan	3	21	0	7.00				
KJ O'Brien					2	17	0	8.50
PP Ojha	4	18	3	4.50				
RM West					4	21	1	5.25
WB Rankin					4	28	0	7.00
WK McCallan					4	27	1	6.75
Yuvraj Singh	1	4	0	4.00				
Z Khan	3	18	4	6.00				

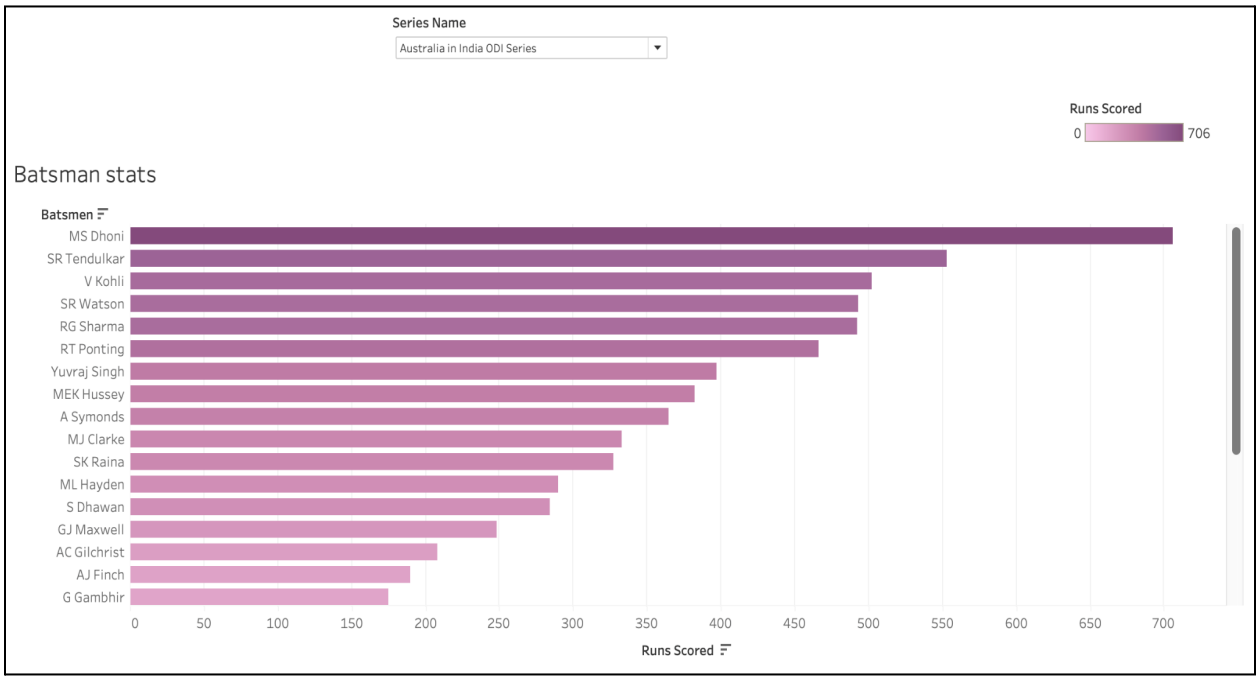
In this scoresheet we can see the bowler statistics and details of each bowler of both teams. We can see the total overs bowled by the bowler, runs conceded and the number of wickets taken by that bowler and the economy.

Insight 5: Batting score



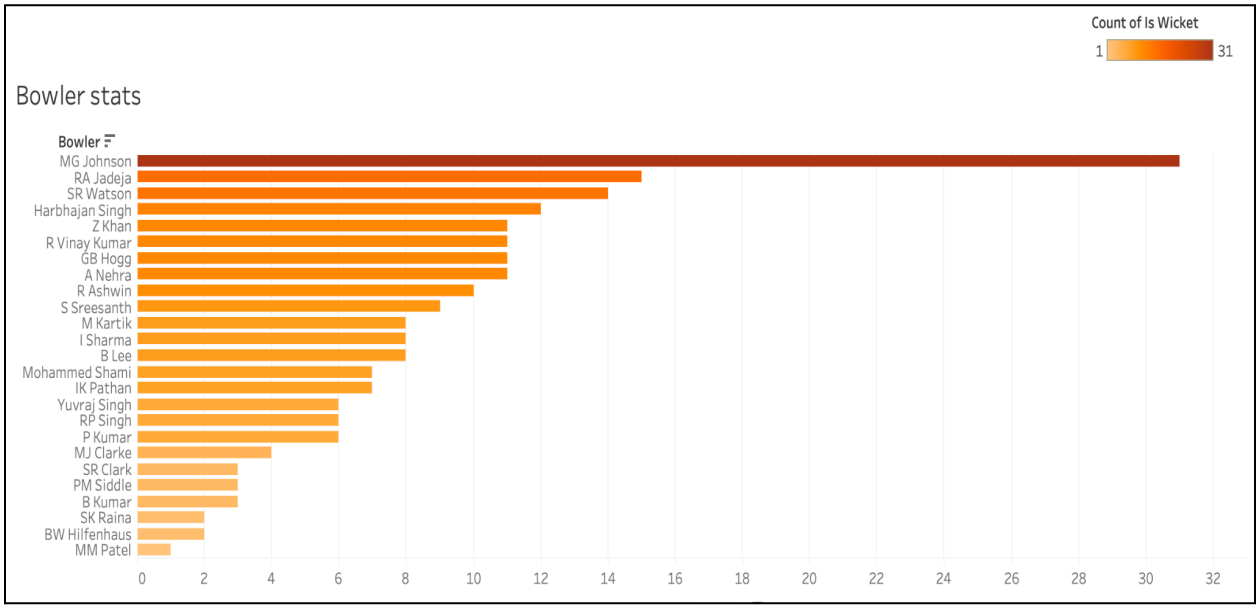
In this bar graph we see the statistics and details of the batsmen of both the teams. The label shows the runs scored by a particular batsman against the over number.

Insight 6: Batsman stats by most runs scored



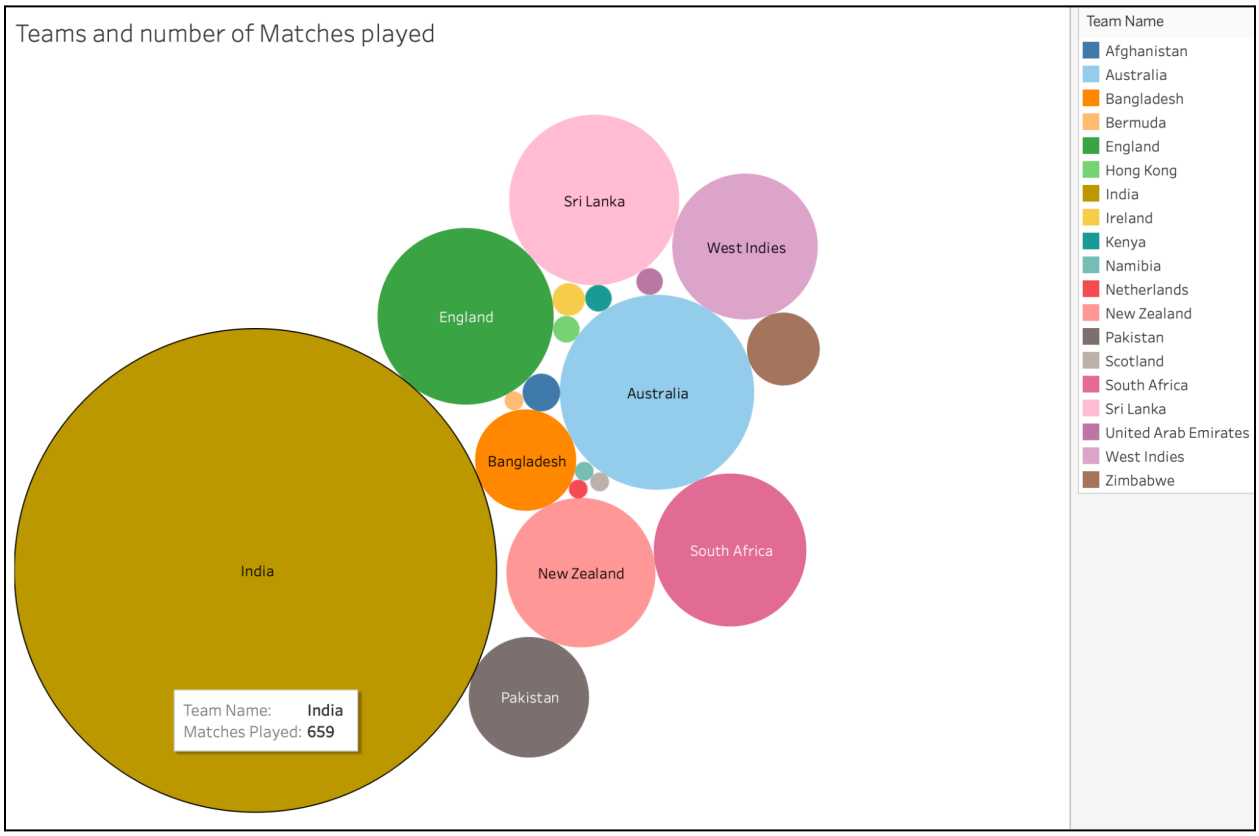
In this bar graph we can see the batsmen statistics who scored the most to least runs in a particular series. The data is displayed amongst all the team players who played in that particular series.

Insight 7: Bowler stats by most wickets taken



In this bar graph we can see the bowler statistics who took the most wickets in a particular series. The data is displayed amongst all the team players that played in that particular series

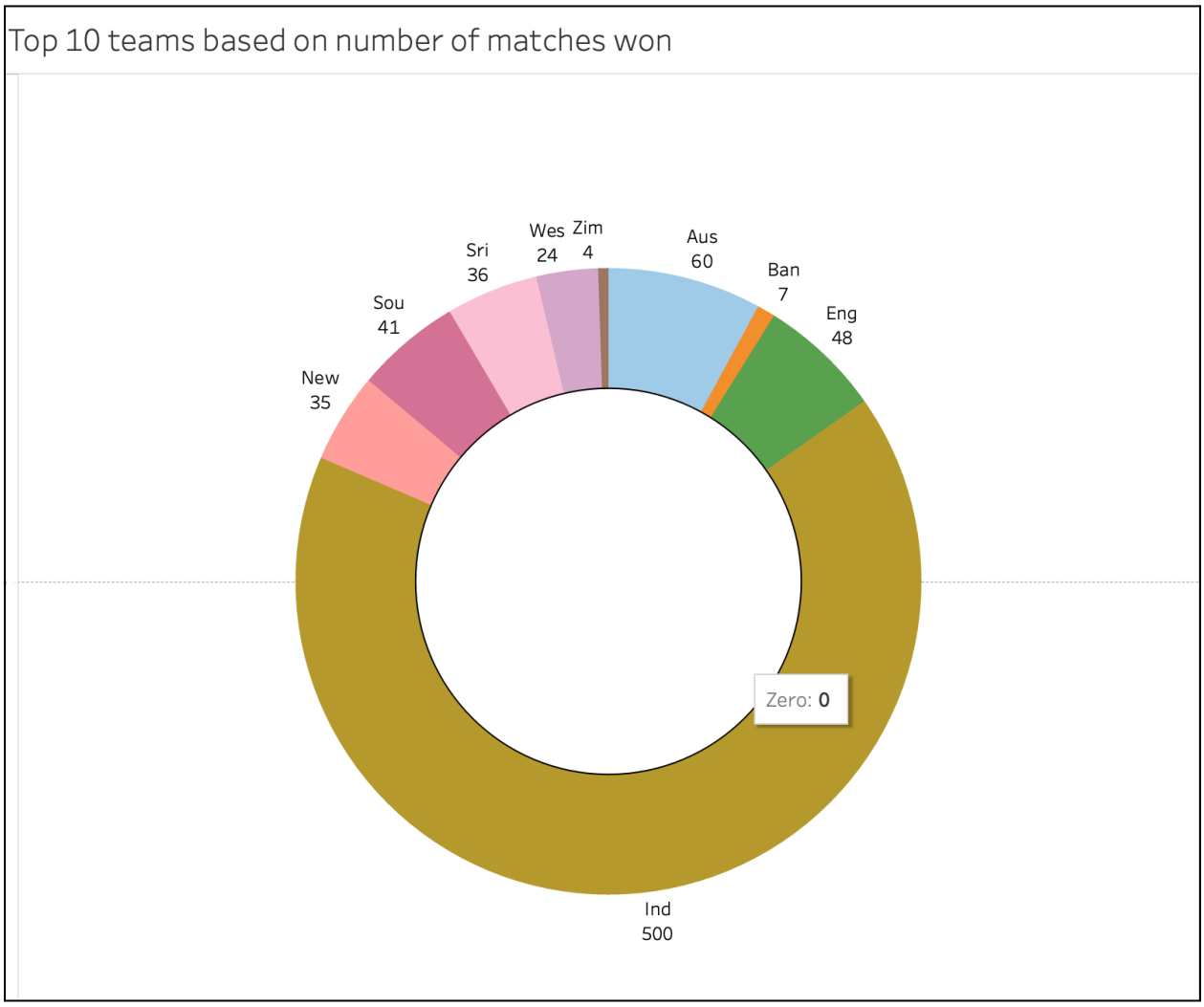
Insight 8: Teams and number of matches played



The bubble chart created showcases the total matches played by each cricket team in various series and test matches. By representing the data through bubbles, the chart allows for easy comparison between teams based on their overall participation in cricket competitions. The size of each bubble corresponds to the number of matches played by the respective team.

The chart reveals that India has participated in the most matches among all teams. By studying the chart, users can compare the performances of different teams and analyze factors contributing to their achievements or shortcomings.

Insight 9: **Top 10 teams based on number of matches won**



Above donut chart displays the number of wins of the teams in our dataset. Here we can see that India won the most giving a total count of 500 wins while Australia has 60 wins, England has 48 wins, South Africa has 41 wins and so on.

6.12 **Website URL:** https://sports-analysis.herokuapp.com/

Chapter 7 GitHub and Collaboration

7.1. GitHub

The complete source code and finalized documentation will be made available on the GitHub repository.

GitHub Link: https://github.com/uzair1996/sports_analytics.git

Conclusion and Future Work:

Conclusion: In summary, the developed project demonstrates a comprehensive sports analytics solution that merges data handling, processing, visualization, and web development to deliver valuable cricket insights. By utilizing various technologies like Amazon S3, AWS Glue, Amazon Redshift, Tableau Public, Django, and EC2 instances, an integrated system has been created that presents users with an informative and engaging experience.

Future Work: Integrating more data sources or adding real-time data streams would enrich the platform, providing users with up-to-date and diverse information and incorporating machine learning and artificial intelligence methods could enable predictive and prescriptive analytics on the platform. This would empower users to forecast future trends, simulate scenarios, and make data-driven decisions.Creating a mobile app would make the platform available to a larger audience, allowing users to access data and visualizations on-the-move while also enabling location-based features or augmented reality experiences.

References:

Dithurbide, L., Sullivan, P., & Chow, G. (2009). Examining the Influence of Team-Referent Causal Attributions and Team Performance on Collective Efficacy: A Multilevel Analysis. *Small Group Research*, *40*(5), 491–507. **https://doi.org/10.1177/1046496409340328**