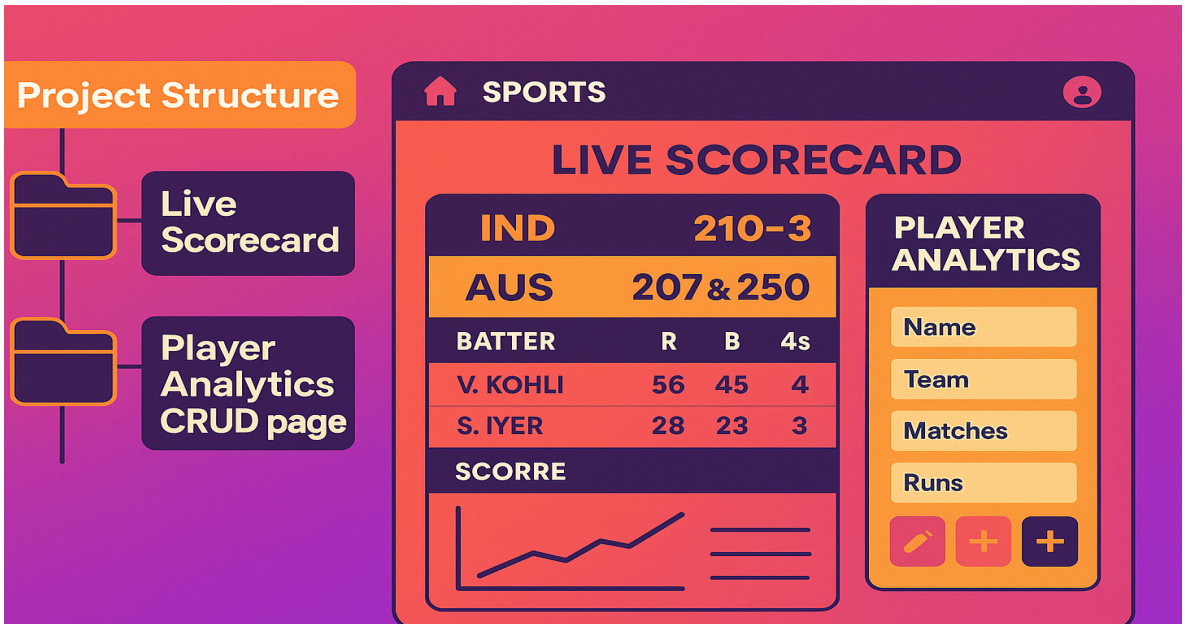
Project Title	**🏏 Cricbuzz LiveStats: Real-Time Cricket Insights & SQL-Based Analytics**
Skills take away From This Project	**Python • SQL • Streamlit • JSON • REST API**
Domain	**Sports Analytics**



🎯 Problem Statement

Build a comprehensive cricket analytics dashboard that integrates live data from the Cricbuzz API with a SQL database to create an interactive web application. The platform should deliver:

- ⚡ Real-time match updates
- 📊 Detailed player statistics
- 🔍 SQL-driven analytics
- 🛠️ Full CRUD operations for data management

💼 Business Use Cases

1. 📺 Sports Media & Broadcasting

- Real-time match updates for commentary teams
- Player performance analysis for pre-match discussions
- Historical data trends for match predictions

2. 🎮 Fantasy Cricket Platforms

- Player form analysis and recent performance tracking
- Head-to-head statistics for team selection
- Real-time score updates for fantasy leagues

3. 📈 Cricket Analytics Firms

- Advanced statistical modeling and player evaluation
- Performance trend analysis across different formats
- Data-driven insights for team management

4. 🎓 Educational Institutions

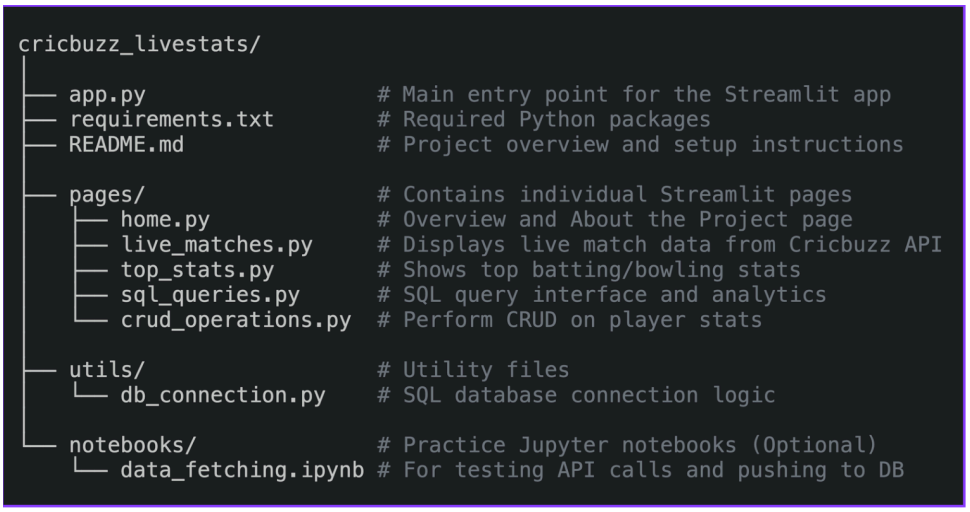
- Teaching database operations with real-world data
- SQL practice with engaging cricket datasets
- API integration and web development learning

5. 🎲 Sports Betting & Prediction

- Historical performance analysis for odds calculation
- Player form and momentum tracking

- Venue-specific performance insights

🏗️ Approach



```
cricbuzz_livestats/
│
├── app.py                  # Main entry point for the Streamlit app
├── requirements.txt        # Required Python packages
├── README.md               # Project overview and setup instructions
│
├── pages/                  # Contains individual Streamlit pages
│   ├── home.py             # Overview and About the Project page
│   ├── live_matches.py     # Displays live match data from Cricbuzz API
│   ├── top_stats.py        # Shows top batting/bowling stats
│   ├── sql_queries.py      # SQL query interface and analytics
│   └── crud_operations.py  # Perform CRUD on player stats
│
├── utils/                  # Utility files
│   └── db_connection.py    # SQL database connection logic
│
└── notebooks/              # Practice Jupyter notebooks (Optional)
    └── data_fetching.ipynb # For testing API calls and pushing to DB
```

Key Development Steps

1. 🌐 **API Integration**
   - Utilize Cricbuzz Cricket API via REST
   - Implement Python's requests library for data fetching
   - Handle real-time match data, player stats, and series information
2. 🖥️ **Interactive Dashboard**
   - Multi-page web application using Streamlit
   - Live scorecards and statistics visualization
   - Custom SQL query interface
   - Administrative CRUD operations
3. 🗄️ **SQL Database Integration**
   - Database-agnostic design (PostgreSQL, MySQL, SQLite)
   - Centralized connection handling in `utils/db_connection.py`
   - Optimized queries for performance
4. ⚙️ **CRUD Operations**

- Form-based UI for data manipulation
- Add, update, and delete player records

🧮 SQL Practice Questions (25 Complete Queries)

Beginner Level (Questions 1-8)

Question 1 Find all players who represent India. Display their full name, playing role, batting style, and bowling style.

Question 2 Show all cricket matches that were played in the last 30 days. Include the match description, both team names, venue name with city, and the match date. Sort by most recent matches first.

Question 3 List the top 10 highest run scorers in ODI cricket. Show player name, total runs scored, batting average, and number of centuries. Display the highest run scorer first.

Question 4 Display all cricket venues that have a seating capacity of more than 50,000 spectators. Show venue name, city, country, and capacity. Order by largest capacity first.

Question 5 Calculate how many matches each team has won. Show team name and total number of wins. Display teams with most wins first.

Question 6 Count how many players belong to each playing role (like Batsman, Bowler, All-rounder, Wicket-keeper). Show the role and count of players for each role.

Question 7 Find the highest individual batting score achieved in each cricket format (Test, ODI, T20I). Display the format and the highest score for that format.

Question 8 Show all cricket series that started in the year 2024. Include series name, host country, match type, start date, and total number of matches planned.

Intermediate Level (Questions 9-16)

Question 9 Find all-rounder players who have scored more than 1000 runs AND taken more than 50 wickets in their career. Display player name, total runs, total wickets, and the cricket format.

Question 10 Get details of the last 20 completed matches. Show match description, both team names, winning team, victory margin, victory type (runs/wickets), and venue name. Display most recent matches first.

Question 11 Compare each player's performance across different cricket formats. For players who have played at least 2 different formats, show their total runs in Test cricket, ODI cricket, and T20I cricket, along with their overall batting average across all formats.

Question 12 Analyze each international team's performance when playing at home versus playing away. Determine whether each team played at home or away based on whether the venue country matches the team's country. Count wins for each team in both home and away conditions.

Question 13 Identify batting partnerships where two consecutive batsmen (batting positions next to each other) scored a combined total of 100 or more runs in the same innings. Show both player names, their combined partnership runs, and which innings it occurred in.

Question 14 Examine bowling performance at different venues. For bowlers who have played at least 3 matches at the same venue, calculate their average economy rate, total wickets taken, and number of matches played at each venue. Focus on bowlers who bowled at least 4 overs in each match.

Question 15 Identify players who perform exceptionally well in close matches. A close match is defined as one decided by less than 50 runs OR less than 5 wickets. For these close matches, calculate each player's average runs scored, total close matches played, and how many of those close matches their team won when they batted.

Question 16 Track how players' batting performance changes over different years. For matches since 2020, show each player's average runs per match and average strike rate for each year. Only include players who played at least 5 matches in that year.

Advanced Level (Questions 17-25)

Question 17 Investigate whether winning the toss gives teams an advantage in winning matches. Calculate what percentage of matches are won by the team that wins the toss, broken down by their toss decision (choosing to bat first or bowl first).

Question 18 Find the most economical bowlers in limited-overs cricket (ODI and T20 formats). Calculate each bowler's overall economy rate and total wickets taken. Only consider bowlers who have bowled in at least 10 matches and bowled at least 2 overs per match on average.

Question 19 Determine which batsmen are most consistent in their scoring. Calculate the average runs scored and the standard deviation of runs for each player. Only include players who have faced at least 10 balls per innings and played since 2022. A lower standard deviation indicates more consistent performance.

Question 20 Analyze how many matches each player has played in different cricket formats and their batting average in each format. Show the count of Test matches, ODI matches, and T20 matches for each player, along with their respective batting averages. Only include players who have played at least 20 total matches across all formats.

Question 21 Create a comprehensive performance ranking system for players. Combine their batting performance (runs scored, batting average, strike rate), bowling performance (wickets taken, bowling average, economy rate), and fielding performance (catches, stumpings) into a single weighted score. Use this formula and rank players:

- Batting points: (runs_scored × 0.01) + (batting_average × 0.5) + (strike_rate × 0.3)
- Bowling points: (wickets_taken × 2) + (50 - bowling_average) × 0.5) + ((6 - economy_rate) × 2)
- Fielding points: (catches × 3) + (stumpings × 5)

Rank the top performers in each cricket format.

Question 22 Build a head-to-head match prediction analysis between teams. For each pair of teams that have played at least 5 matches against each other in the last 3 years, calculate:

- Total matches played between them
- Wins for each team
- Average victory margin when each team wins
- Performance when batting first vs bowling first at different venues
- Overall win percentage for each team in this head-to-head record

Question 23 Analyze recent player form and momentum. For each player's last 10 batting performances, calculate:

- Average runs in their last 5 matches vs their last 10 matches
- Recent strike rate trends
- Number of scores above 50 in recent matches
- A consistency score based on standard deviation

Based on these metrics, categorize players as being in "Excellent Form", "Good Form", "Average Form", or "Poor Form".

Question 24 Study successful batting partnerships to identify the best player combinations. For pairs of players who have batted together as consecutive batsmen (positions differ by 1) in at least 5 partnerships:

- Calculate their average partnership runs
- Count how many of their partnerships exceeded 50 runs
- Find their highest partnership score
- Calculate their success rate (percentage of good partnerships)

Rank the most successful batting partnerships.

Question 25 Perform a time-series analysis of player performance evolution. Track how each player's batting performance changes over time by:

- Calculating quarterly averages for runs and strike rate
- Comparing each quarter's performance to the previous quarter
- Identifying whether performance is improving, declining, or stable
- Determining overall career trajectory over the last few years
- Categorizing players' career phase as "Career Ascending", "Career Declining", or "Career Stable"

Only analyze players with data spanning at least 6 quarters and a minimum of 3 matches per quarter.

📊 Query Implementation Tips

- **Beginner Queries**: Focus on basic SELECT, WHERE, GROUP BY, and ORDER BY operations
- **Intermediate Queries**: Utilize JOINs, subqueries, and aggregate functions
- **Advanced Queries**: Implement window functions, CTEs, and complex analytical calculations
- **Performance**: Add proper indexing for frequently queried columns
- **Data Validation**: Include NULL checks and data quality validations

🎯 Expected Results

The final output of the *Cricbuzz LiveStats* project will deliver a comprehensive and user-friendly cricket analytics dashboard. The expected deliverables include:

📌 Page-wise Features

1. Live Match Page

- Users can view ongoing matches from Cricbuzz API
- Detailed scorecards, batsmen/bowler info, match status, and venue details shown in real-time

2. Top Player Stats Page

- Display top batting and bowling stats fetched via the Cricbuzz API
- Includes categories like most runs, highest score, most wickets, etc., visualized cleanly

3. SQL Queries & Analytics Page

- Integrates 25+ advanced SQL queries on the player and match database
- Shows tabular outputs directly in Streamlit for user interaction

4. CRUD Operations Page

- Full Create, Read, Update, Delete operations on player and match stats using form-based UI
- Useful for learners to understand database manipulation

5. Home Page

- Describes the project, tools used, instructions, and navigation between pages
- Provides a link to project documentation and folder structure info

Technical Tags:

`Python` `Streamlit` `SQL Database` `REST API` `Database Connector` `pandas` `requests` `Data Analytics` `Web Development` `Database Management` `Sports Technology`

Project Deliverables:

- **Source Code**: Complete Python application (`main.py`)
- **Database Schema**: SQL table structure and sample data
- **Documentation**: Project setup instructions and API key configuration
- **Requirements**: Dependencies list (streamlit, database connector, requests, pandas)
- **Demo**: Working dashboard with all four modules functional
- **SQL Practice**: Complete 25 SQL questions (easy, medium, hard levels) for analytics module

Project Guidelines:

- **Coding Standards**: Follow PEP 8 Python style guidelines
- **Error Handling**: Implement proper exception handling for API calls and database operations
- **Security**: Secure database credentials and API keys
- **Modularity**: Separate functions for different operations
- **Documentation**: Clear comments and docstrings for all functions
- **Version Control**: Use Git for code versioning and collaboration

Timeline:

The project should be completed and submitted **within 14 days** from the date it is assigned.