

Jawaharlal Nehru Technological University Hyderabad
University College of Engineering , Science &
Technology



Real-time/Field-Based Research Project Report

On

HowzThat! - Celebrating Local Cricket Heroes

Submitted by

Kotla Prakhyath Reddy (22011A6609)

Veerabathini Bhanu Prasad (22011A6610)

Shaik Sohail (22011A6616)

Kalloji Pavan Kumar (22011A6617)

**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

CERTIFICATE

This is to certify that the project work entitled “**HowzThat! - Celebrating Local Cricket Heroes**” is being submitted by

Kotla Prakhyath Reddy (22011A6609), Veerabathini Bhanu Prasad (22011A6610), Shaik Sohail (22011A6616), Kalloji Pavan Kumar (22011A6617) in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering Artificial Intelligence and Machine Learning** , during the academic year 2023-2024.

Professor and Head, CSE

(DR.K P Supreethi)

Project Supervisor

(Ms. Praveena N)

Assistant Professor(c)

Submitted for Autonomous End Semester Examination Mini Project
Viva - Voce on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

Abstract

Local cricket tournaments, despite nurturing future stars, often go unnoticed compared to professional leagues. HowzThat! bridges this gap by offering a user-friendly web application for cricket fans. Users can enjoy real-time score updates, stay updated with comprehensive match schedules, and delve into in-depth player statistics.

The application leverages a robust tech stack. Python handles core backend functionality, while Django acts as the development framework. MySQL ensures scalable data storage. WebSockets enable real-time score updates. Standard web technologies like HTML, CSS, and JavaScript provide structure, styling, and user interaction.

This application goes beyond just following local cricket. HowzThat! fosters a sense of community by celebrating the exceptional talent often unseen on the grand stage. By keeping fans engaged and shining a light on the true spirit of cricket at the grassroots level, HowzThat! ensures that local heroes get the recognition they deserve.

ACKNOWLEDGEMENT

We extend our heartfelt appreciation to all those who have contributed to the development and success of our cricket website.

We would like to express our deepest gratitude to the faculty members whose guidance and support have been invaluable throughout the development process. Their expertise, mentorship, and encouragement have provided us with the necessary guidance and resources to navigate the complexities of app development and meteorological analysis

I would like to acknowledge the invaluable information provided by the official website of the International Cricket Council (ICC), the ESPNcricinfo website, and the Cricbuzz website. These resources proved to be a rich source of cricket data and insights.

Additionally, I am grateful for the existence of open-source technologies like Python, Django, MySQL and WebSockets. These tools provided a powerful foundation for building this web application.

Last but not least, we extend our sincere appreciation to our users for their support and feedback. Their engagement and enthusiasm have been vital in driving the ongoing improvement of our application.

TABLE OF CONTENTS

Abstract.....	3
ACKNOWLEDGEMENT.....	4
TABLE OF CONTENTS.....	5
1. INTRODUCTION.....	7
1.1 Motivation :.....	7
1.2 Problem Statement :.....	7
1.3 Project Objectives:.....	8
2. Requirement Analysis.....	9
2.1 Software Requirements:.....	9
2.2 Hardware Requirements.....	9
3. UML DIAGRAMS.....	10
3.1 CLASS DIAGRAM.....	10
3.2 USE CASE DIAGRAM.....	11
3.3 SEQUENCE DIAGRAM.....	12
3.4 SCHEMA.....	13
4. Project Design and Development.....	14
4.1 Linear vs. Iterative Development.....	14
4.2 Development Methodology.....	14
4.2.1 Step-by-Step Breakdown:.....	15
4.2.2 Benefits of Our Methodology.....	16
5. Tools and Technologies.....	17
5.1 Python.....	17
5.2 Django.....	17
5.3 HTML, CSS, and JavaScript.....	18
5.4 MySQL.....	19
5.5 Visual Studio Code (VS Code).....	19
5.6 Git and GitHub.....	20
5.7 WebSockets.....	20
6. Project Phases.....	22
7. Testing.....	24
7.1 Unit Testing.....	24
7.2 Integration Testing.....	25
7.3 Functional Testing.....	25
7.4 Usability Testing.....	26
7.5 Performance Testing.....	26
7.6 Regression Testing.....	27
7.7 User Acceptance Testing (UAT).....	27

8. User Guide.....	29
8.1 Register and Create a New Account:.....	29
8.2 Login to Your Account:.....	29
8.3 View Matches List:.....	30
8.4 View Live Scores and Player Statistics:.....	30
8.5 Search Player Statistics:.....	31
8.6 Host a New Match:.....	32
8.7 Select Team Players:.....	32
8.8 Coin Toss and Match Start:.....	33
8.9 Update Match Based on Live Feed:.....	33
8.10 Match Completion and Summary:.....	34
9. Conclusion.....	35
10. REFERENCES.....	36
11. APPENDIX.....	36

1. INTRODUCTION

1.1 Motivation :

In the world of cricket, where major tournaments and matches enjoy widespread attention and dedicated apps, local cricket matches often remain in the shadows. This disparity leaves a gap in recognizing and celebrating the remarkable achievements of local players, hindering the growth and appreciation of grassroots cricket. Our project, "HowzThat!," aims to bridge this gap by providing a platform that empowers local cricket communities and showcases the true spirit of the game.

Competitive Edge : "HowzThat!" distinguishes itself by focusing on the underserved segment of local cricket. While existing apps cater to major events, our platform will specifically address the needs of local tournaments and matches, providing a unique and tailored experience for players, organizers, and fans at the grassroots level.

1.2 Problem Statement :

The lack of attention and recognition for local cricket matches poses a significant challenge. Talented players in local communities often go unnoticed, their accomplishments uncelebrated. This not only impacts the morale of players but also hinders the overall development and popularity of cricket at the grassroots level. Additionally, the absence of a centralized platform makes it difficult for fans to follow local matches, access live scores, and engage with their local cricket community.

1.3 Project Objectives:

The primary objectives of "HowzThat!" are:

- To develop a user-friendly web application that enables real-time tracking of live cricket scores for local matches.
- To create a platform that allows local players to showcase their skills and achievements, fostering a sense of community and recognition.
- To provide cricket enthusiasts with easy access to information about local matches, schedules, and player statistics.
- To promote the growth and development of cricket at the grassroots level by increasing visibility and engagement.
- To help cricket team selectors scout for local cricket players and find talents from local tournaments.

2. Requirement Analysis

2.1 Software Requirements:

Live score updates: Real-time updates of scores for ongoing matches.

Match schedules and fixtures: Detailed schedules and information about upcoming and ongoing tournaments.

Team and player profiles: In-depth profiles of teams and players, including statistics and match history.

User registration and authentication: Allowing users to create accounts and log in to personalize their experience.

Data storage and retrieval: Efficiently storing and retrieving match data, player information, and user preferences.

Error handling and validation: Implementing mechanisms to handle errors and validate user input.

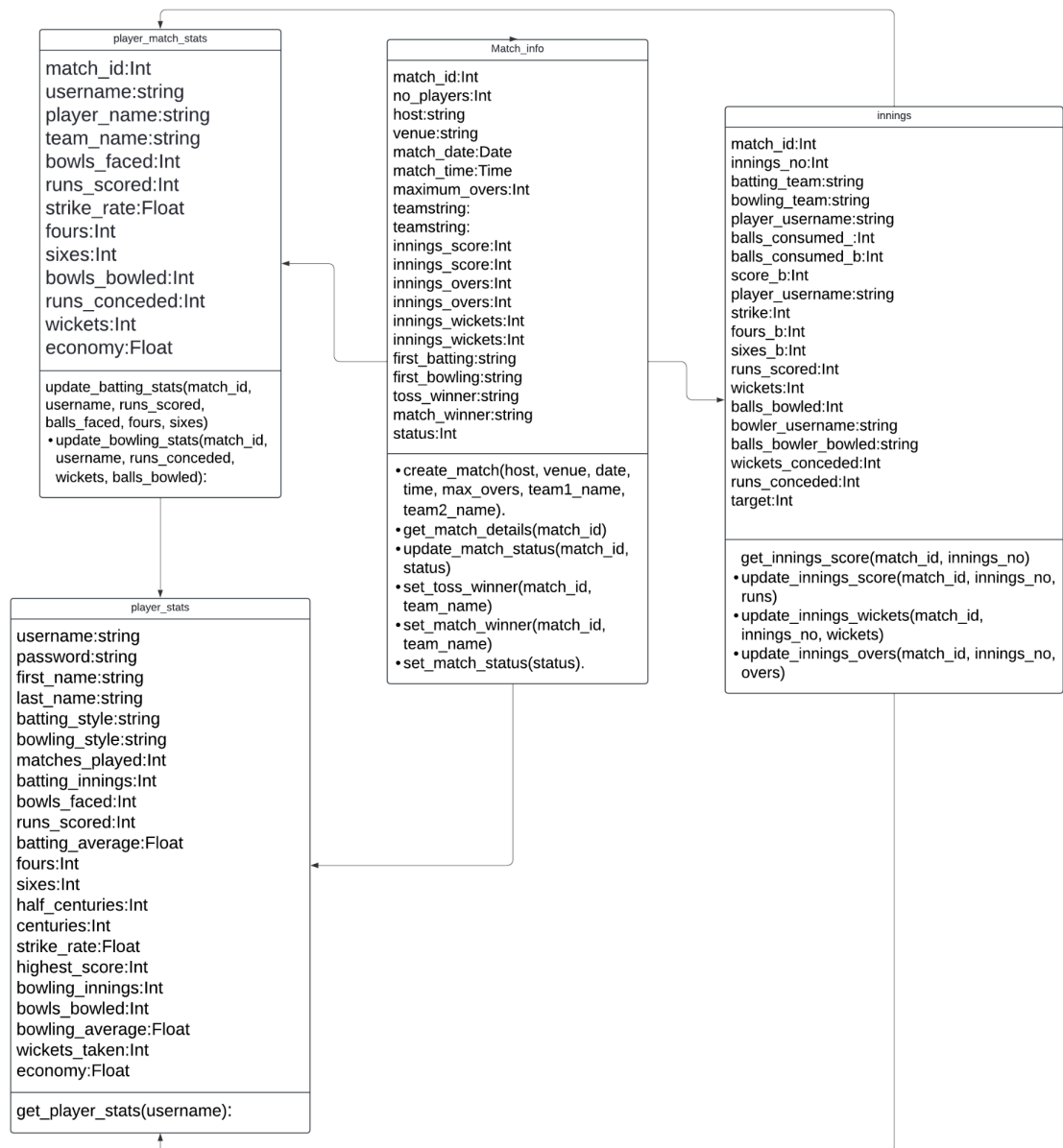
Security: Protecting user data and ensuring the application is secure from vulnerabilities.

2.2 Hardware Requirements

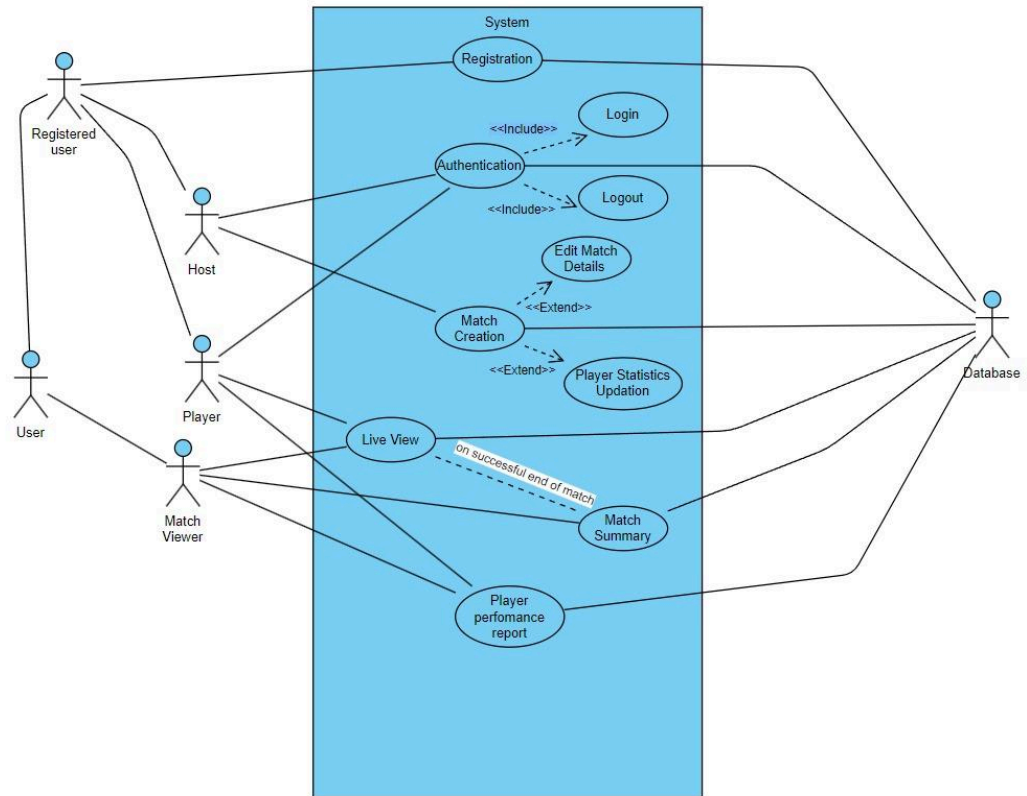
- **OPERATING SYSTEM :** WINDOWS , MACOS , LINUX , ANDROID , IOS
- **APPLICATIONS :** ANY BROWSER
- **RAM :** 2GB
- **PROCESSOR :** INTEL CORE I3 OR EQUIVALENT
- **Active Internet connection.**

3. UML DIAGRAMS

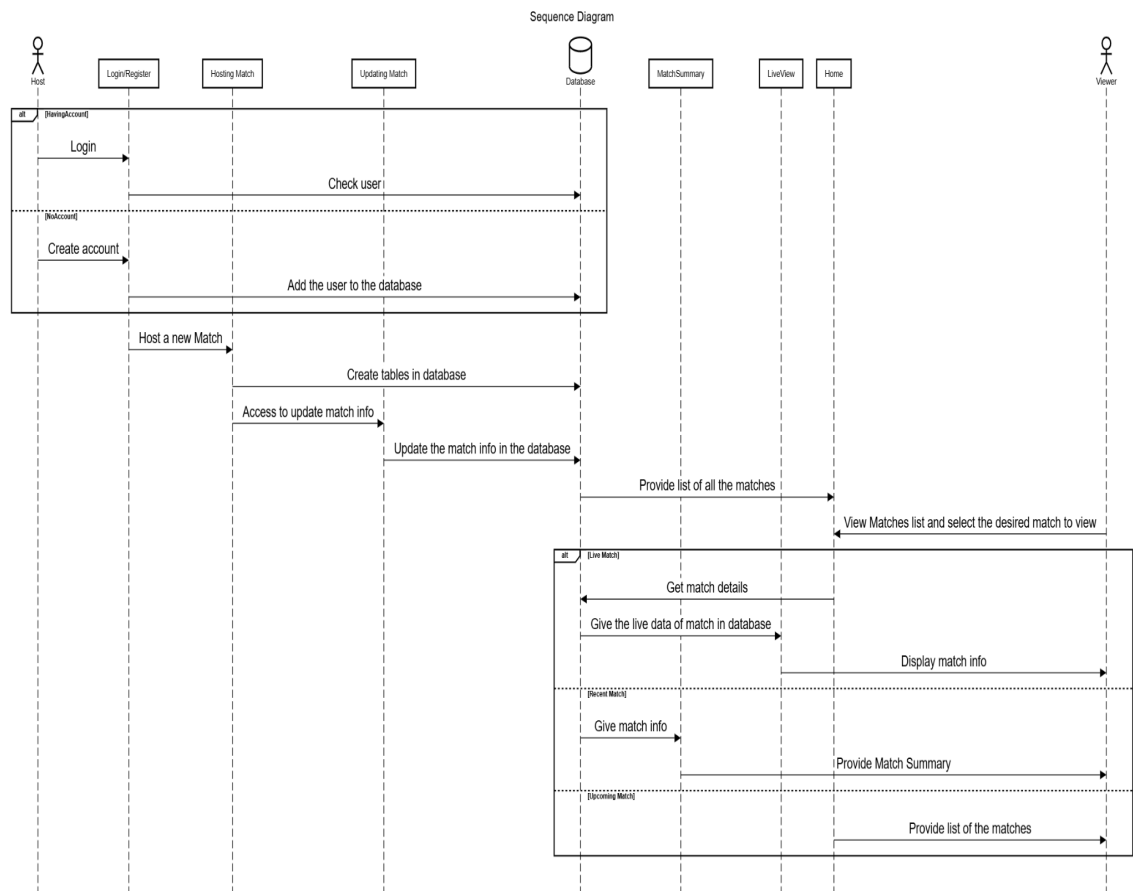
3.1 CLASS DIAGRAM



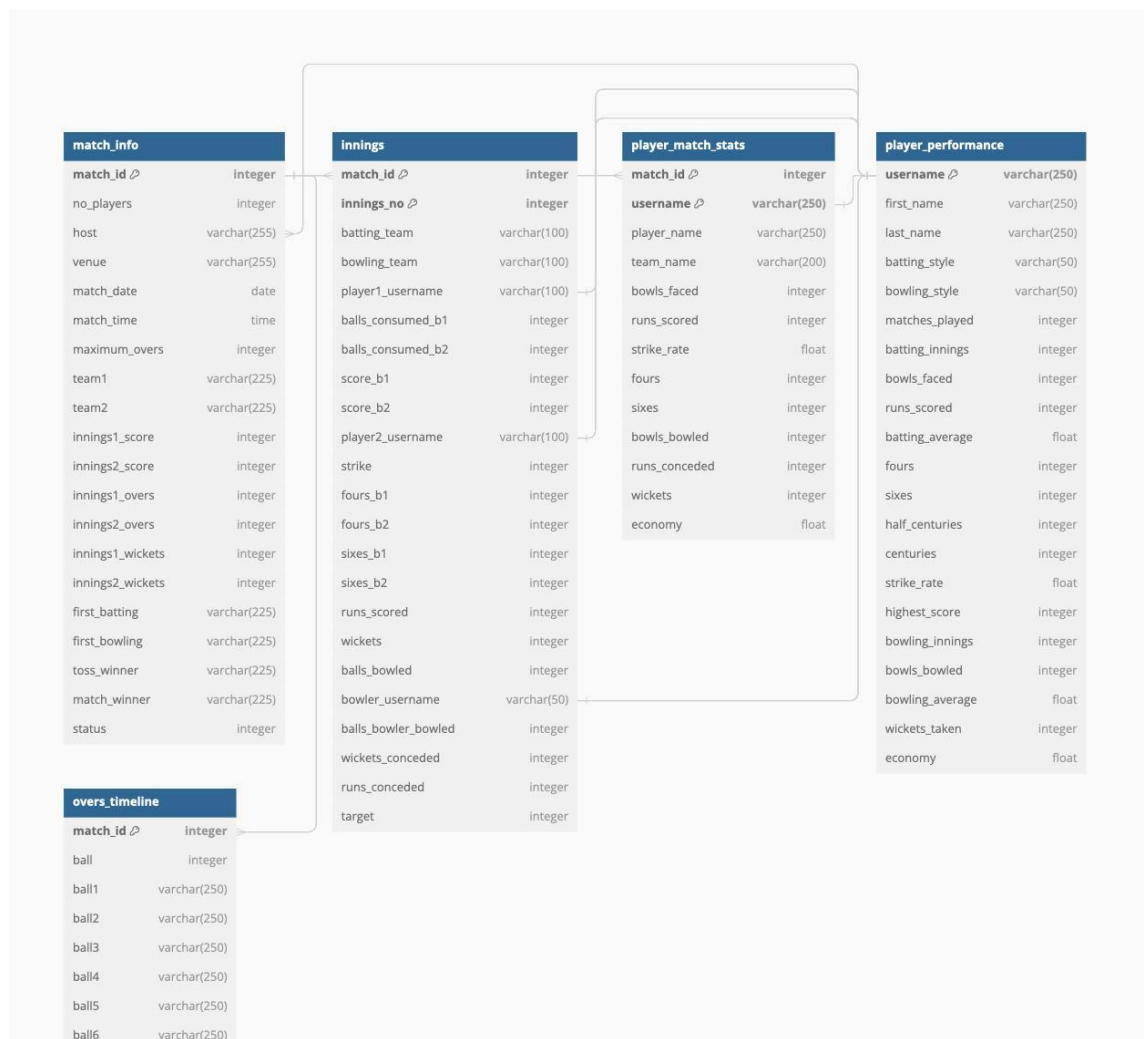
3.2 USE CASE DIAGRAM



3.3 SEQUENCE DIAGRAM



3.4 SCHEMA



4. Project Design and Development

Our project is an innovative web application that allows users to host live cricket matches, with real-time score updates. Using Django and Channels, we created a dynamic and responsive platform where hosts can update match information and users can view live scores seamlessly. The development process involved dividing the project into distinct parts, completing each part systematically, and refining them through continuous testing and integration. This methodology provides a clear roadmap of our development process, showcasing our systematic approach and iterative improvements.

4.1 Linear vs. Iterative Development

Our project can be categorized as an iterative development process. Although we initially divided the project into different parts and completed each one sequentially, we incorporated continuous testing and refinement after each part was completed. This iterative aspect allowed us to identify and address issues early on, ensuring that each component was functional and integrated smoothly into the larger project.

4.2 Development Methodology

We employed a hybrid approach that combines elements of both Waterfall and Agile methodologies. The project's division into discrete phases—creating HTML pages, implementing core functionalities, integrating web pages, and styling—mirrors the structured, sequential nature of the Waterfall model. However, our approach also aligns with Agile principles due to its iterative cycles of development and testing.

4.2.1 Step-by-Step Breakdown:

1. Creating HTML Pages (Basic Layout)

- **Description:** We began by creating all the HTML pages required for the project, focusing on the basic layout without any styling.
- **Benefit:** This provided a structural framework for the project, making it easier to visualize the overall design and plan subsequent steps.

2. Implementing Core Functionalities

- **Description:** We implemented core functionalities one by one, such as the login page, hosting a match page, and live score page.
- **Benefit:** This incremental approach allowed us to focus on developing and perfecting each functionality individually, ensuring that each component worked correctly before moving on to the next.

3. Integrating Web Pages

- **Description:** After implementing the core functionalities, we integrated all the web pages to form a cohesive, single project.
- **Benefit:** Integrating the components early in the process helped us identify integration issues and ensured that all parts worked harmoniously together.

4. Styling the Web Pages

- **Description:** Finally, we styled the web pages to enhance their visual appeal and user experience.
- **Benefit:** By focusing on styling after integrating the functionalities, we ensured that the core functions were not disrupted and could be enhanced visually without compromising functionality.

4.2.2 Benefits of Our Methodology

1. Enhanced Focus and Clarity

- Dividing the project into distinct parts provided a clear roadmap and milestones. This structured approach ensured that we remained focused on specific goals and avoided being overwhelmed by the project's complexity.

2. Early Detection of Issues

- Continuous testing after completing each part allowed us to detect and address issues early in the development process. This iterative refinement minimized the risk of major issues arising later in the project.

3. Improved Integration

- By integrating the web pages after developing core functionalities, we ensured that all components worked together smoothly. This stepwise integration helped us identify and resolve compatibility issues early on.

4. User-Centric Design

- Focusing on styling and user experience after integrating the functionalities ensured that the core functions remained intact. This approach allowed us to enhance the user interface without compromising the application's performance.

5. Flexibility and Adaptability

- The hybrid methodology provided the structure of the Waterfall model while allowing the flexibility of Agile.

5. Tools and Technologies

Our project, which allows users to host live cricket matches and view real-time scores, utilizes a robust stack of tools and technologies. A key highlight is the use of WebSockets to make our web pages dynamic, eliminating the need to reload pages when updating scores and other match information. Below, we provide an in-depth look at each technology and tool employed in our development process.

5.1 Python

Python is the backbone of our project. We chose Python for its simplicity, readability, and extensive library support, which enabled rapid development and prototyping.

Advantages:

- Comprehensive standard library that provided solutions for various tasks.
- Strong community support, ensuring readily available resources and troubleshooting.

5.2 Django

Django is our chosen web framework, leveraging Python's capabilities to build a robust and scalable web application.

Key Features:

- **Channels and WebSockets:** We used Django Channels to implement WebSockets, making our web pages dynamic. This was crucial for real-time score updates, allowing the host to update match information without requiring users to reload the page.
- **Authentication:** Django's built-in authentication system was used to manage user logins securely.

- **Template Tags:** We utilized Django template tags to insert dynamic content into our HTML pages efficiently.

Advantages:

- High-level framework that promotes rapid development and clean, pragmatic design.
- Secure, with a built-in authentication system that simplifies user management.
- Supports WebSockets through Django Channels, which is essential for real-time applications like ours.

5.3 HTML, CSS, and JavaScript

We built the web pages using **HTML**, **CSS**, and **JavaScript**, ensuring a dynamic and visually appealing user interface.

HTML:

- Structured the content of our web pages.
- Created the basic layout for the application.

CSS:

- Styled the web pages to make them visually appealing.
- Ensured a responsive design that works across different devices and screen sizes.

JavaScript:

- Added interactivity to the web pages.
- Implemented WebSockets to dynamically update the page content in real-time, a critical feature for displaying live scores without requiring page reloads.

Advantages:

- HTML provides a standardized structure, making the pages accessible and SEO-friendly.
- CSS ensures a consistent and attractive look and feel across the application.
- JavaScript, combined with WebSockets, enables real-time updates, enhancing user experience by providing instant feedback and dynamic content.

5.4 MySQL

We used **MySQL** as our database management system to store and manage all the data related to users, matches, and scores.

Advantages:

- Reliable and well-established relational database system.
- High performance and scalability, suitable for handling large volumes of data.
- Strong support for complex queries, ensuring efficient data retrieval and manipulation.

5.5 Visual Studio Code (VS Code)

Visual Studio Code (VS Code) was our integrated development environment (IDE) of choice for this project.

Advantages:

- Lightweight yet powerful, with a plethora of extensions to enhance productivity.

- Excellent support for Python and Django development, with features like IntelliSense, debugging, and Git integration.
- Customizable interface, allowing us to tailor the environment to our development needs.

5.6 Git and GitHub

We used **Git** for version control and **GitHub** for code repository hosting, collaboration, and project management.

Advantages:

- Git provides a robust version control system, enabling us to track changes, revert to previous states, and collaborate effectively.
- GitHub offers a platform for remote collaboration, issue tracking, and code reviews, essential for maintaining code quality and managing the project efficiently.

5.7 WebSockets

A standout feature of our project is the use of WebSockets to create dynamic web pages. WebSockets enable real-time, bidirectional communication between the server and clients, which is crucial for applications requiring live updates, such as our live cricket match platform.

Advantages:

- **Real-Time Updates:** Users receive live score updates instantly, without needing to refresh the page. This enhances the user experience by providing seamless, up-to-date information.
- **Efficient Data Transfer:** WebSockets maintain a single, persistent connection, reducing the overhead of repeatedly opening and closing connections. This efficiency is particularly beneficial for frequent updates.

- **Responsive Interaction:** WebSockets allow for quick and responsive interactions, making the application feel more interactive and engaging.

The combination of these tools and technologies, especially the integration of WebSockets through Django Channels, enabled us to build a robust, dynamic, and user-friendly web application. Each component played a crucial role in the development process, contributing to the overall functionality, performance, and user experience of our live cricket match hosting platform. Our choice of tools ensured a smooth development workflow, facilitated real-time updates, and provided a solid foundation for future enhancements and scalability.

6. Project Phases

1. Initiation

- We started by exploring different project ideas and eventually decided to develop a platform for live cricket match hosting.

2. Planning

- To get started, we researched various technologies suitable for the project. Considering the need for real-time functionality, we opted for Django Channels for the backend.
- We then discussed user requirements like account login, access to cricket history, and clear presentation of match information.
- After creating a high-level project overview, we met with our faculty to get their insights and feedback. Their guidance helped us shape the project direction.
- We then divided the workload among our four-person team based on everyone's strengths.
- Finally, we created a basic layout for the project pages, outlining how data would be presented using HTML, CSS, and JavaScript.

3. Execution (Development Phase)

- We collaboratively determined the attributes to be included in the project's database, ensuring it could efficiently store match information and user details.
- Since we were new to Django Channels, the two of us focused on learning and implementing this technology to enable real-time functionality in the platform.

- We then started developing the project's backend, likely involving features like user authentication, data storage and retrieval, and real-time updates.
- To catch issues early on, we implemented small-scale testing throughout the backend development process.

4. Monitoring & Control (Testing and Refinement Phase)

- Once the backend was complete, we focused on refining the frontend (user interface) with CSS for styling and adding transitions for a smoother user experience.
- To simulate real-world scenarios, we conducted internal testing using live cricket matches. This helped us identify any bugs or errors in the platform's functionality.
- Based on the internal testing results, we made necessary changes to the code to rectify identified errors and ensure the platform functioned as intended.

7. Testing

Testing was a critical component of our project development process, ensuring that each functionality worked correctly and the overall application provided a seamless user experience. We adopted a multi-faceted testing approach, incorporating various methods to validate the functionality, performance, and user interaction of our live cricket match hosting platform. This section details the testing strategies and methodologies we employed at different stages of the project.

Testing Methodologies

1. **Unit Testing**
2. **Integration Testing**
3. **Functional Testing**
4. **Usability Testing**
5. **Performance Testing**
6. **Regression Testing**
7. **User Acceptance Testing (UAT)**

7.1 Unit Testing

Unit testing was the foundational step in our testing process. We conducted unit tests for each core functionality as it was implemented.

Approach:

- **Isolated Testing:** Each unit or component was tested in isolation to ensure it performed its specific function correctly. For instance, the login functionality was tested separately from the hosting a match feature.

- **Sample Data:** We used sample data to validate the expected outcomes of each unit. This included various inputs to test normal cases, edge cases, and invalid inputs.
- **Tools:** Django's built-in testing framework was utilized to write and run unit tests. We created test cases for views, models, and forms.

7.2 Integration Testing

Integration testing was conducted after individual units were verified to ensure they functioned correctly when combined.

Approach:

- **Sequential Integration:** We integrated components sequentially as they were completed. For example, after implementing the login page and hosting a match page, we tested their interaction.
- **Interaction Testing:** Tested the interaction between different modules, such as ensuring that only logged-in users could access the match hosting page.
- **Database Integration:** Validated that data was correctly stored and retrieved from the MySQL database when different components interacted.

7.3 Functional Testing

Functional testing focused on verifying that the application behaved as expected and met the specified requirements.

Approach:

- **Requirement-Based Testing:** Tested each functionality against the project requirements. For example, the live score page was tested to ensure it displayed real-time updates correctly.

- **End-to-End Scenarios:** Simulated complete user scenarios, such as logging in, hosting a match, and viewing live scores, to ensure each functionality worked as part of the overall system.
- **Edge Cases:** Considered various edge cases to ensure robustness, such as testing the login functionality with incorrect credentials or attempting to host a match with incomplete data.

7.4 Usability Testing

Usability testing was conducted to ensure the application was user-friendly and intuitive.

Approach:

- **User Feedback:** Invited a group of users to interact with the application and provide feedback on the user interface and experience.
- **Observation:** Monitored users as they navigated the application to identify any usability issues or areas of confusion.
- **Surveys and Interviews:** Collected detailed feedback through surveys and interviews to understand user satisfaction and areas for improvement.

7.5 Performance Testing

Performance testing was essential to ensure the application could handle real-time updates and multiple users simultaneously.

Approach:

- **Load Testing:** Simulated multiple users accessing the application simultaneously to test its performance under load.
- **Stress Testing:** Pushed the application beyond its expected limits to identify breaking points and ensure stability.
- **Response Time:** Measured the response time for real-time updates,

- that WebSockets provided the necessary performance for live score updates.

7.6 Regression Testing

Regression testing was conducted to ensure that new changes did not introduce bugs into the existing functionality.

Approach:

- **Re-testing:** After fixing bugs or adding new features, we re-tested the existing functionalities to ensure they still worked correctly.
- **Automation:** Used automated tests to quickly validate the entire application after each change.
- **Comprehensive Test Suite:** Maintained a comprehensive suite of tests covering all functionalities to catch regressions early.

7.7 User Acceptance Testing (UAT)

User Acceptance Testing was the final step before deployment, ensuring the application met user expectations and business requirements.

Approach:

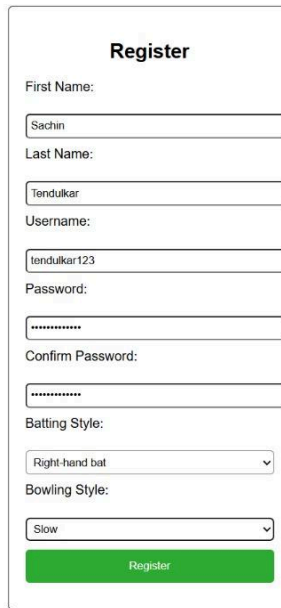
- **Real-World Scenarios:** Tested the application using real-world scenarios, including using live matches from the T20 World Cup 2024 as reference data.
- **Feedback Loop:** Gathered feedback from users and stakeholders, making final adjustments based on their input.
- **Pilot Testing:** Conducted a pilot test with a small group of users to validate the application in a controlled environment before full deployment.

Our comprehensive testing approach ensured that each part of the project was thoroughly validated before integration, and the entire application was rigorously tested before deployment. Unit testing and integration testing provided a strong foundation, while functional, usability, performance, regression, and user acceptance testing ensured the application was robust, user-friendly, and met all specified requirements. This multifaceted testing strategy, combined with continuous feedback and refinement, enabled us to deliver a high-quality, reliable application for hosting live cricket matches with real-time score updates.

8. User Guide

8.1 Register and Create a New Account:

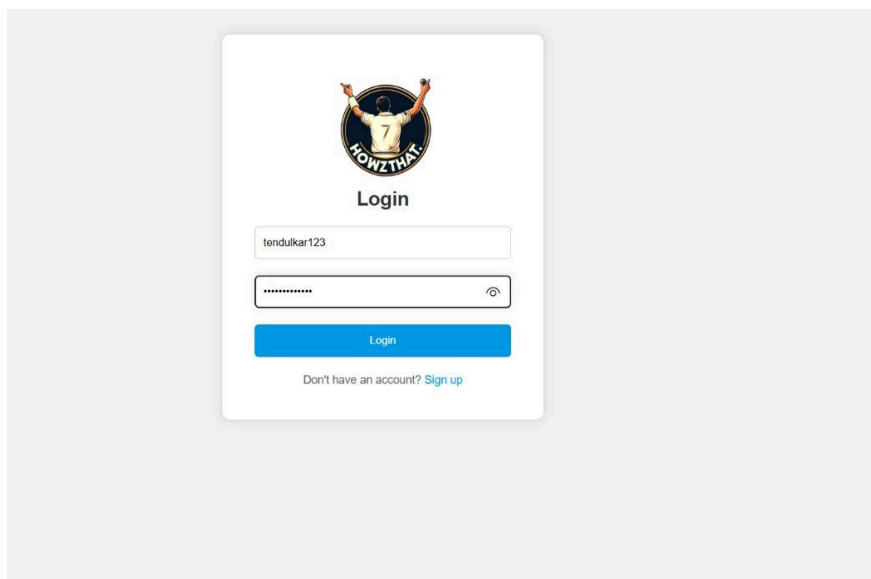
- Sign up to create your account.



A registration form titled "Register" with the following fields: First Name (Sachin), Last Name (Tendulkar), Username (tendulkar123), Password (masked with asterisks), Confirm Password (masked with asterisks), Batting Style (Right-hand bat), and Bowling Style (Slow). A green "Register" button is at the bottom.

8.2 Login to Your Account:

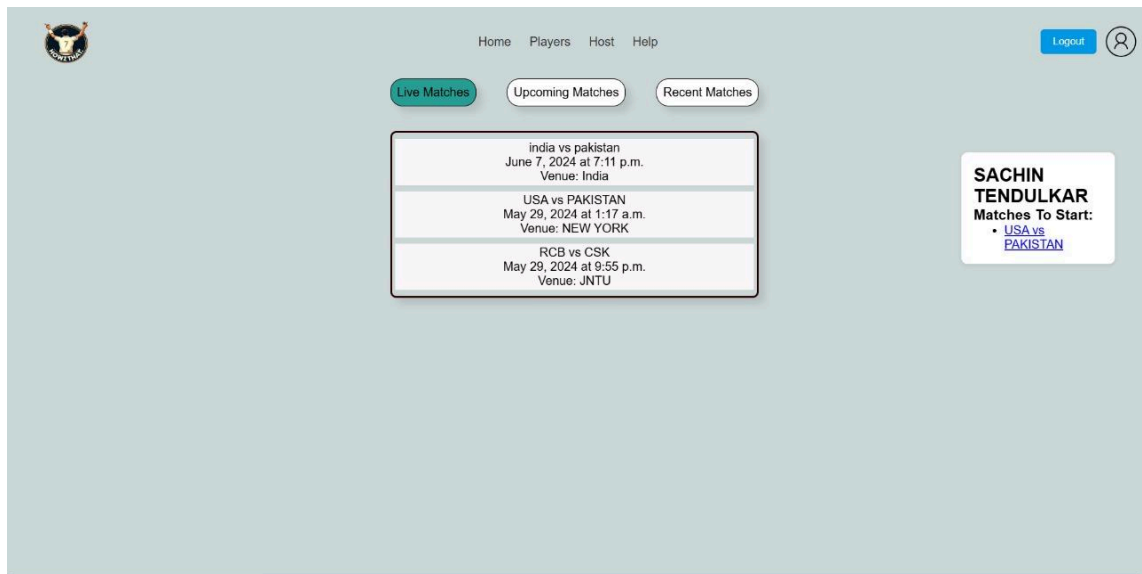
- Access your account to host new matches or continue registered matches.



A login form titled "Login" with the "HOWZTHAT" logo at the top. It contains fields for Username (tendulkar123) and Password (masked with asterisks). A blue "Login" button is below the fields. At the bottom, it says "Don't have an account? [Sign up](#)".

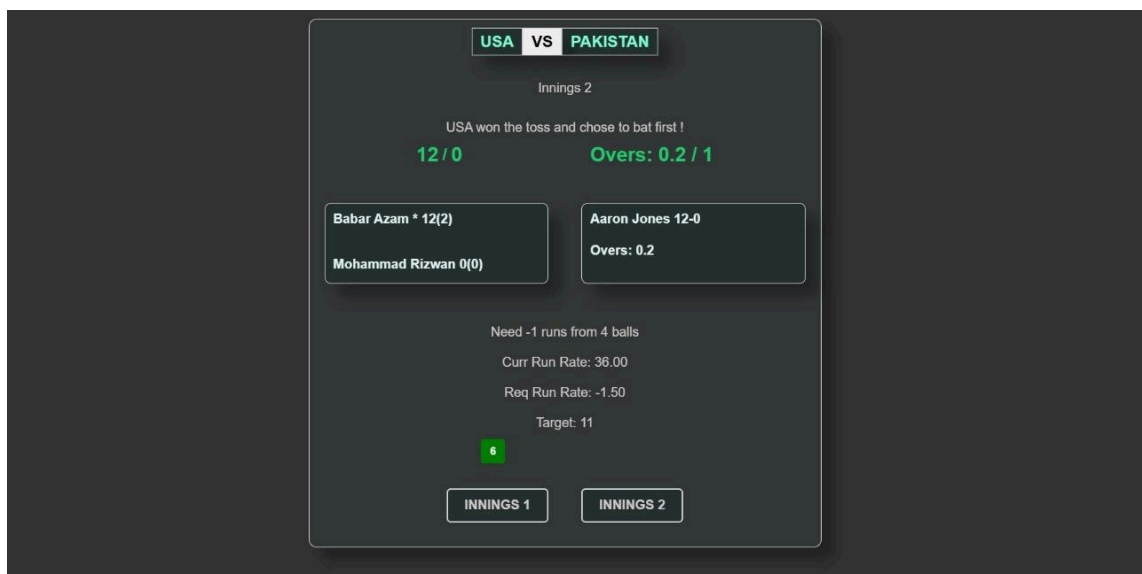
8.3 View Matches List:

- On the home page, browse the list of matches. Select a match to spectate or host.



8.4 View Live Scores and Player Statistics:

- Check live scores and real-time player statistics for the selected match.



Mohammad Rizwan 0(0)	Overs: 0.2
Need -1 runs from 4 balls Curr Run Rate: 36.00 Req Run Rate: -1.50 Target: 11	
6	
INNINGS 1	INNINGS 2

Batting					
Batsman	Runs	Balls	4s	6s	SR
Aaron Jones	4	3	0	0	133.33
Gajanand Singh	6	3	0	0	133.33

Bowling				
Bowler	Overs	Runs	Wickets	Economy
Babar Azam	1.0	10	0	10

8.5 Search Player Statistics:

- In the Statistics section, search for a player by name to view their career stats.

Search by username or name

Search

Babar Azam

Matches Played: 16

Batting Stats

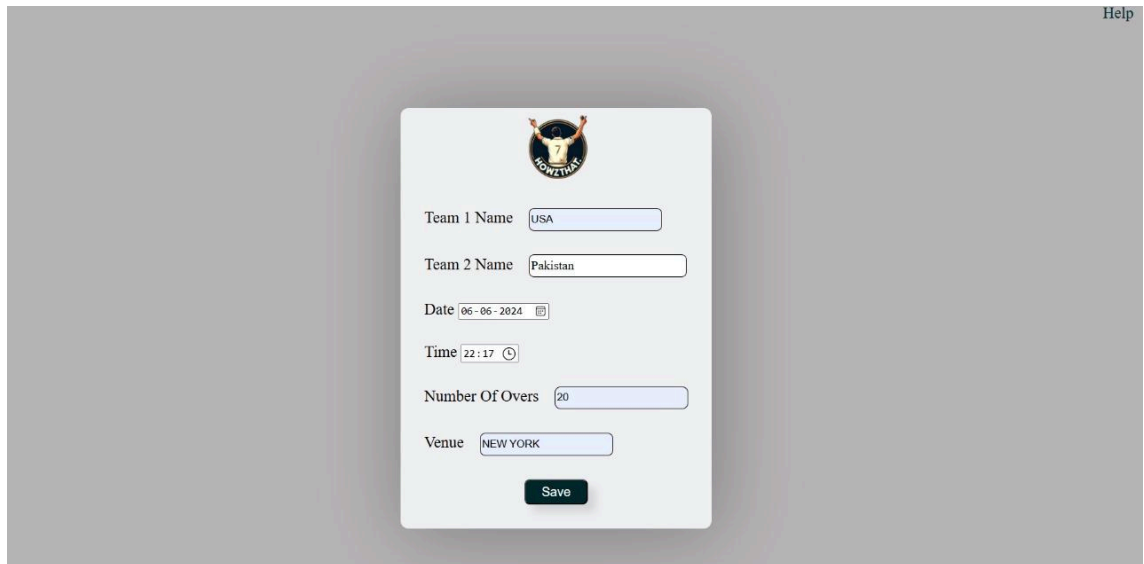
Runs Scored	Average	Fours	Sixes	Highest Score	Strike Rate	50	100
390	24.375	16	28	64	102.0	1	0

Bowling Stats

No. of Wickets	Bowling Average	Economy
18	15.5	7.8428887569061

8.6 Host a New Match:

- Use the 'Host' option in the navigation bar to enter match details and start a new match.



Help

KONTAK

Team 1 Name

Team 2 Name

Date

Time

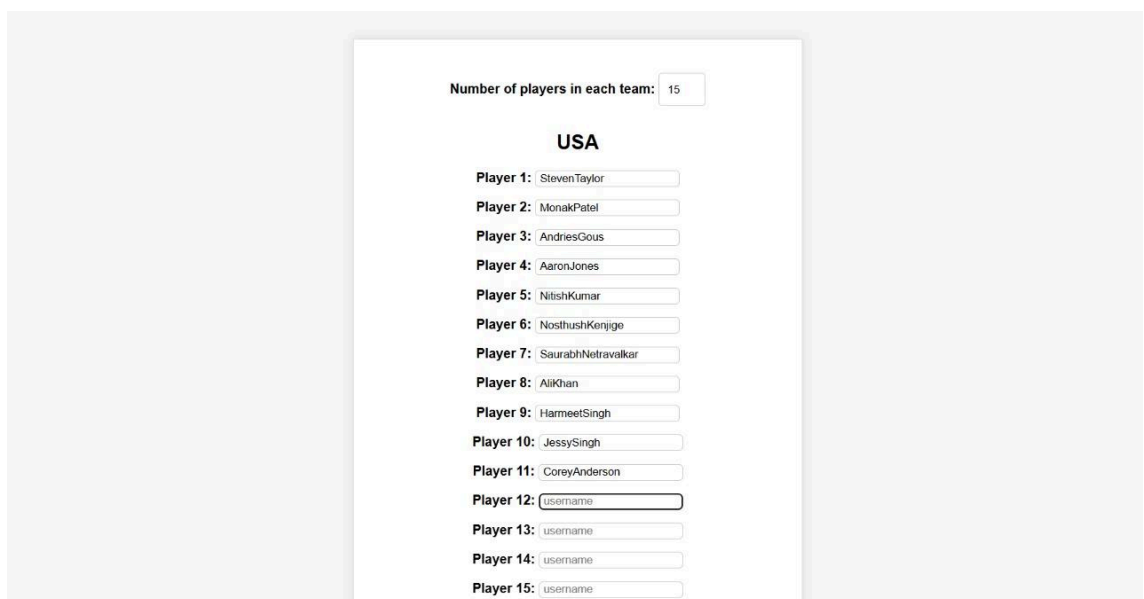
Number Of Overs

Venue

Save

8.7 Select Team Players:

- Choose players for each team by entering their usernames. Ensure the usernames match the actual usernames of the players.



Number of players in each team:

USA

Player 1:

Player 2:

Player 3:

Player 4:

Player 5:

Player 6:

Player 7:

Player 8:

Player 9:

Player 10:

Player 11:

Player 12:

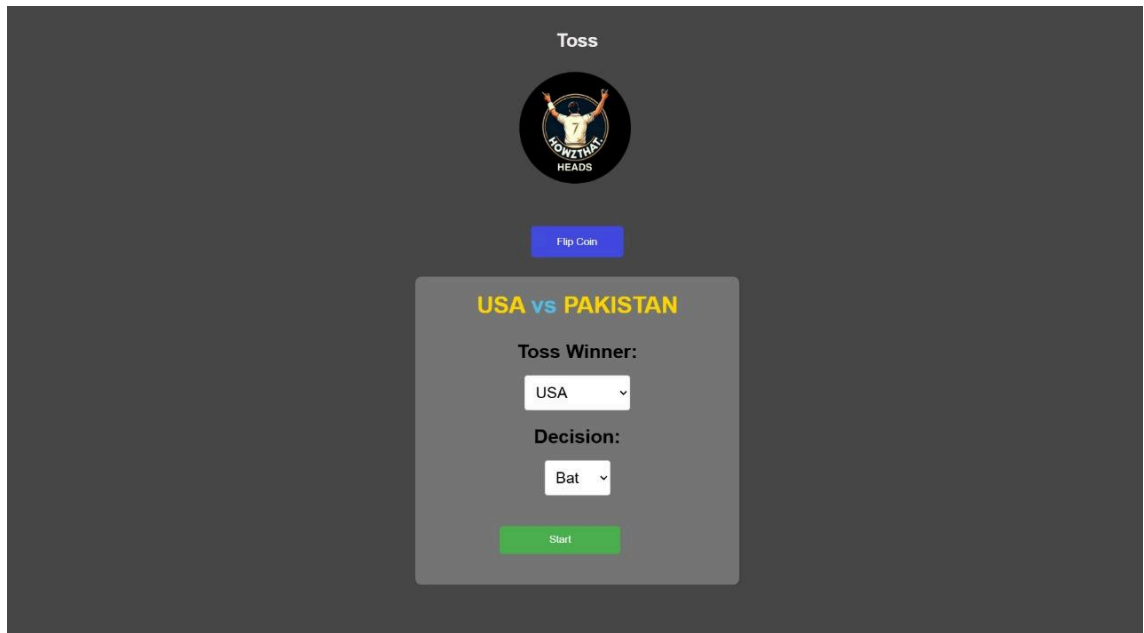
Player 13:

Player 14:

Player 15:

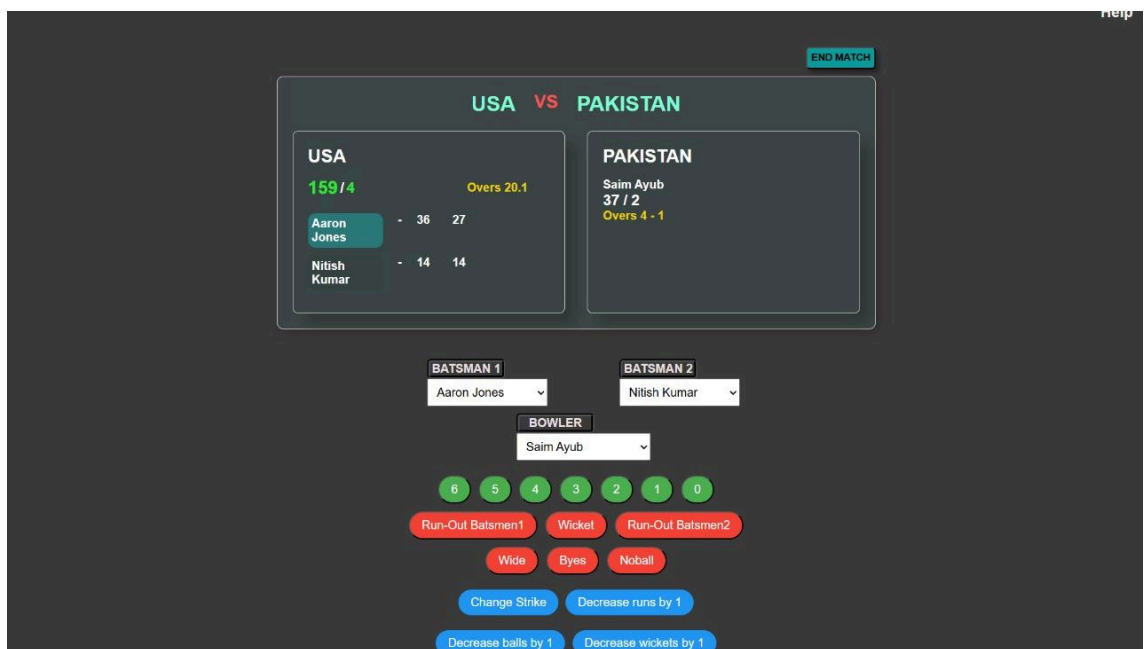
8.8 Coin Toss and Match Start:

- Flip the coin to decide the toss and start the match based on the outcome.



8.9 Update Match Based on Live Feed:

- Continuously update the match details based on the live feed.



8.10 Match Completion and Summary:

- Once the match ends, the page will redirect to the Match Summary.

USA

159/7

(20.0)

PAKISTAN

158/3

(20.0)

USA WON BY 3 WICKETS

Innings 1

Innings 2

Innings 2 Batting

Player	Runs	Balls	4s	6s	SR
Aaron Jones	36	26	1	0	138.0
Nitish Kumar	36	26	2	2	138.0
Steven Taylor	12	16	1	0	75.0
Andries Gous	35	26	4	1	134.0
Monank Patel	50	38	7	1	131.0

Innings 2 Bowling

Player	Overs	Runs	Wickets	Econ
Iftikhar Ahmed	1.0	10	0	10.0
Shadab Khan	3.0	27	0	9.0

9. Conclusion

In conclusion, the "HowzThat!" project successfully addresses the need for a platform dedicated to local cricket matches. By providing real-time score updates, comprehensive match information, and player statistics, the application empowers local cricket communities and fosters a deeper appreciation for the sport at the grassroots level. The user-friendly interface and mobile optimization ensure accessibility for cricket enthusiasts on any device, promoting engagement and participation.

The development of "HowzThat!" involved the utilization of cutting-edge web technologies, including Python, Django, MySQL, Redis, React.js, and WebSocket. These technologies enabled the creation of a dynamic and responsive application capable of delivering real-time updates and a seamless user experience.

The project's success can be attributed to the collaborative efforts of frontend developers, backend engineers, and cricket data experts. Their combined expertise ensured the delivery of a high-quality product that accurately reflects the nuances of local cricket matches.

"HowzThat!" is poised to make a significant impact on the local cricket scene. By shining a spotlight on grassroots talent, fostering community engagement, and providing a platform for celebration, the application is set to revolutionize how local cricket is experienced and appreciated. As the application continues to evolve and expand its features, it has the potential to become an indispensable tool for cricket enthusiasts and players alike, further solidifying its position as a game-changer in the world of local cricket.

10. REFERENCES

- ICC(International Cricket Council).
- Django Documentation.
- MYSQL Documentation.
- Python Documentation

11. APPENDIX

SOURCE CODE : <https://github.com/prax2712/Cricket-app>