



(REVIEW ARTICLE)



Ultimate cricket experience: Dynamic web app for a real-time scoring system in university cricket

Habid Hossen, Md. Shahin Shiddik Shuvon, Jubaida Begum Barsha, Arshek Ahmed Chy and Abhijit Pathak *

Department of Computer Science and Engineering, BGC Trust University Bangladesh, Chattogram, Bangladesh.

World Journal of Advanced Research and Reviews, 2023, 19(02), 1269–1280

Publication history: Received on 17 July 2023; revised on 24 August 2023; accepted on 27 August 2023

Article DOI: <https://doi.org/10.30574/wjarr.2023.19.2.1721>

Abstract

Cricket, a globally popular team sport, involves eleven players and encompasses both batting and bowling skills, with the objective of scoring runs and dismissing the opposition's batsmen. The sport has complex rules and demands skill, technique, and teamwork. Cricket enjoys a significant following worldwide and has produced iconic players. It has evolved with shorter formats and tournaments, attracting a global audience. The "Ultimate Cricket Experience: Dynamic Web App for Real-Time Scoring System in University Cricket" project aims to develop a web application specifically designed for university cricket sports. The main goal of the project is to develop a dynamic web application that provides real-time cricket scorecards for cricket enthusiasts, offering an immersive and engaging experience across all active cricket matches. By providing a user-friendly platform that makes it simple for users to stay up to date with live scores, statistics, and important match highlights, this program strives to satisfy the ardent love of cricket fans. The strategic implementation of the MERN stack, combining MongoDB, Express.js, React.js, and Node.js, ensures a robust, scalable, and modern solution for developing the real-time scorecard system. This combination of technologies offers a robust and scalable solution for developing A Real-time Cricket Scorecard System. The key features of the web application include a live scorecard that provides real-time updates on ongoing matches, and comprehensive match details such as team information, player statistics, and match schedules. The platform also offers university sports news, sports event information, and other relevant details to keep users informed and engaged with the cricketing activities within the university.

Keywords: MERN Stack; Real-time Cricket Scorecards; University Cricket Sports; Sports News; Dynamic Web Application; MongoDB

1. Introduction

The origins of Cricket trace back to the 16th century in England. Over five centuries, the sport has evolved significantly, advancing through various stages. Beginning with notches on sticks, the art of cricket scoring has now transitioned to encompass PC software and mobile applications. The total number of runs scored, and the number of wickets taken are two important factors in the cricket scoring system. In the world of professional cricket, two scorers—one from each team—are normally chosen to painstakingly track runs scored, wickets taken, and, when necessary, the number of overs bowled. However, the umpire holds the authority to declare runs and wickets lost, after which the scorer documents these values. The updated score is subsequently projected on large screens visible to both players and the crowd. Although their roles may be defined as the simple documentation of runs, wickets, and overs, the reality is far more intricate, encompassing the management of a range of game statistics. Cricket scoring encompasses two methodologies: manual and computerized. The manual approach involves the scorer using a preprinted scorecard, colloquially referred to as the scorebook, to log details of each ball bowled in the match. The advent of cricket scoring software aimed to alleviate the intricacies associated with traditional methods. However, even in this evolved process, complete

* Corresponding author: Abhijit Pathak

computerization is not yet achieved. The scorer is still required to press buttons to record ball-by-ball occurrences. Carrying out this action for each delivery can prove monotonous and repetitive. The automation of this scoring process could drastically diminish the burden on scorekeepers, freeing them from the need to interact with the screen after every delivery and allowing them to concentrate solely on other vital game statistics. Automation has the potential to truncate the time dedicated to this laborious task and concurrently mitigate the risk of errors when updating the scoreboard.

The "Ultimate Cricket Experience: Dynamic Web App for Real-Time Scoring System in University Cricket" project is firmly anchored in a multi-faceted motivation. This motivation is driven by an ardent desire to revolutionize how university cricket sports are experienced and embraced. Through the harmonious fusion of accessible information, heightened user experiences, comprehensive sports coverage, and the embrace of technological innovations, the project aspires to elevate university cricket sports to new heights, fostering a sense of community, engagement, and celebration among students and cricket enthusiasts.

In summary, the project objectives encapsulate a multifaceted approach aimed at transforming the landscape of university cricket sports:

- Robust Technological Foundation
- Convenient Access Hub
- Engaging User Experience
- Deepened User Understanding
- Real-Time Updates
- Holistic Sports Coverage
- Engagement and Belonging
- Cultural Revitalization
- Technological Advancements
- User-Centric Design

Collectively, these objectives converge to reshape the way university cricket sports are experienced, celebrated, and embraced. The project strives to create an innovative platform that engages users, deepens their connection with cricket, and fosters a vibrant community of sports enthusiasts. Through technology, comprehensive insights, and an immersive journey, the project endeavors to redefine the boundaries of sports engagement and inspire a shared passion for the sport among students and cricket enthusiasts. Upon successful completion, the web application is poised to usher in a new era of engagement and connection within the realm of university cricket sports. This dynamic application will serve as an efficient and user-friendly platform, providing users with instantaneous access to live match scores, up-to-the-minute university sports news, comprehensive sports event details, intricate match specifics, and real-time scores. This transformative project aims to redefine the very fabric of how university cricket sports are perceived, experienced, and celebrated. The project embarks on an ambitious journey to bridge the gap between the realms of university cricket sports and cutting-edge technology. By harnessing the potential of the MERN stack and embracing contemporary web development methodologies, this project aspires to reshape the very contours of user experiences. The result is a platform that not only enhances engagement but also cultivates a profound sense of belonging among students, faculty members, and cricket aficionados alike. This endeavor strives to usher university cricket sports into the digital era, where the sport is not just celebrated but also experienced and cherished in novel and transformative ways.

2. Literature Review

The literature review section serves the purpose of offering a comprehensive overview of existing research, studies, and pertinent projects about cricket, sports technology, and web applications. This segment endeavors to illuminate the current landscape within the realm of university cricket sports, while also delving into the realm of possibilities brought forth by the introduction of a real-time cricket scorecard system. Through this analysis, the review aims to shed light on the multifaceted benefits that such a system could bring in augmenting the overall experience of individuals passionate about cricket.

University cricket sports, a breeding ground for personal growth and social integration, hold immense value. However, the lack of accessible and timely information poses a substantial impediment to the engagement and enjoyment of ardent cricket enthusiasts. It is against this backdrop that the proposed project aims to make a meaningful intervention. Through the implementation of a real-time cricket scorecard system, the project endeavors to surmount these obstacles and usher in a holistic platform tailored to the unique needs of cricket enthusiasts within university settings. The field

of real-time sports scorecard systems, particularly in the context of cricket, has witnessed significant advancements in recent years. The emergence of technology has significantly transformed the landscape of sports engagement and spectatorship. Several studies and works have contributed to this field by exploring the integration of computer vision, machine learning, and gesture recognition techniques. These studies examine various developments that have contributed to the evolution of real-time cricket scorecard systems and their impact on user engagement and experience. Numerous studies emphasize the importance of real-time updates and interactivity in sports engagement. In the context of university sports, Vaishnavi K. Nair et. al introduced a system that identifies pitch events through the recognition of umpire gestures, subsequently adjusting the scoreboard. The model is specifically trained to recognize four umpire gestures: Six, No Ball, Wide, and Out [1]. Dhanusha T. John et. al proposed employing a pre-trained model named Caffe to extract skeleton points from supplied images. These points are utilized to construct skeleton images, which are subsequently trained using Deep Neural Networks (DNNs). The system's performance could be impacted by the umpire's gesture consistency, and its adaptability would require ongoing updates to accommodate rule alterations or variations [2]. On the other hand, Md. Asif Shahjalal et. al proposed a method and a functional prototype for automating umpire decisions through the interpretation of hand gestures. This automation would eliminate the need for manual scorecard updates, resulting in a significant reduction in game duration. While this approach effectively handles static gestures like "out," "no," "six," and "wide," it exhibits lower efficiency for dynamic gestures such as "four" [3]. K. Kalai Selvan et al created a dynamic cricket scoreboard that provides real-time updates of match scores along with commentary. Given the evolving nature of the computer field, the system can be continually enhanced to incorporate technical advancements and improvements [4]. Additionally, Rameshwari A. Lokhande et. al proposed a model featuring two techniques, both of which are realized through either a Linear Regression Classifier or a Q-Learning-based decision tree approach. An ensemble module is developed by employing various learning algorithms, ensuring applicability across all cricket match formats, including T20, ODI, and Test cricket [5]. Medha Wyawahare et. al proposed a system that can identify different umpire signals within images and subsequently update the scorecard automatically. The model is trained on a dataset of 6000 images categorized into six classes: Six, Four, Out, No ball, wide ball, and a no-action class. While the system effectively identifies the five specified umpire signals, it falls short in recognizing uncommon signals like leg byes and a dead ball. Additionally, since there are no distinct umpire signals for singles and doubles, manual scorecard updates are necessary for those instances [6]. Furthermore, Eeshan Mundhe et. al created a web application that conducts predictive analysis during a live T-20 match. This analysis aims to forecast the ultimate score and even predict the match's victor before it commences. Although the application's potential can be broadened to encompass One Day International and Test matches, this expansion would require comprehensive historical data from all past games across formats, which is presently unavailable [7]. P.N.P. Fernando et al. discussed the creation and evolution of an information system tailored for One-Day International (ODI) cricket. Additionally, after achieving a complete database update, the system could serve multiple functions, including generating diverse cricket literature, magazines, and souvenirs. While existing literature highlights the evolution and impact of real-time cricket scorecard systems, there are still research gaps to be addressed. These include investigating the effectiveness of machine learning predictions, exploring new ways to integrate augmented reality for an immersive experience, and studying the psychological and emotional aspects of fan engagement in the context of real-time sports updates [8]. Sanjay Joshi's research spanning 2001 to 2021 offers a comprehensive examination of Machine Learning (ML) applications in cricket. The study underscores ML's pivotal role in enhancing player performance analysis, deciphering game dynamics, and refining strategic decision-making. Prominent ML techniques like SVM, Random Forest, and Naive Bayes are scrutinized using accuracy measures based on the confusion matrix. Although the study effectively addresses cricket challenges through ML, it also acknowledges the untapped potential of advanced learning methods. In presenting a holistic overview of ML's influence, the paper acknowledges certain constraints in data selection [9]. Real-time updates, interactive elements, and user-centered design concepts can all be combined to boost community development, fan satisfaction, and engagement. However, additional research is required to examine the system's effects over a longer time frame and to pinpoint any potential issues with its adoption and implementation in the setting of universities. By offering creative solutions to improve the real-time cricket viewing experience and further reunite fans with their beloved sport, this paper contributes to the changing landscape.

3. Methodology

The methodology for the project involves a systematic approach to ensure the successful development and implementation of the proposed web application. The following steps outline the methodology:

- Project Planning and Requirement Gathering:
 - Define the scope, objectives, and deliverables of the project.
 - Identify key stakeholders, including cricket enthusiasts, students, faculty, and administrators.
 - Gather detailed requirements through surveys, interviews, and discussions to understand user needs and preferences.

- **Technology Selection and Framework Design:**
 - Choose the appropriate technology stack, including MongoDB for the database, Express.js for the backend framework, React.js for the front end, and Node.js for server-side scripting.
 - Design the overall architecture of the web application, considering factors like scalability, modularity, and security.
- **Database Design and Schema Creation:**
 - Design the database schema to store match data, user information, scores, and other relevant details.
 - Implement data models and relationships using MongoDB to ensure efficient data storage and retrieval.
- **Backend Development:**
 - Develop the backend components using Express.js and Node.js to handle API requests and business logic.
 - Implement real-time functionality to update scores and statistics as matches progress.
 - Set up user authentication and authorization mechanisms to ensure data security and access control.
- **Frontend Development:**
 - Create the user interface using React.js, focusing on a user-centric design approach.
 - Develop interactive components for live scores, match details, statistics, and news.
 - Implement responsive design to ensure a seamless experience across various devices.
- **Real-time Data Integration:**
 - Utilize WebSocket or other real-time communication technologies to provide instant updates on live match scores and events.
 - Integrate APIs or data sources to fetch real-time scores and statistics from ongoing cricket matches.
- **Testing and Quality Assurance:**
 - Conduct thorough testing, including unit testing, integration testing, and user acceptance testing.
 - Identify and rectify any bugs, errors, or usability issues to ensure a seamless user experience.
- **Deployment and Hosting:**
 - Deploy the web application on a suitable hosting platform, ensuring scalability and performance.
 - Configure domain settings, security protocols, and SSL certificates for secure data transmission.
- **User Training and Documentation:**
 - Provide user documentation and guides to facilitate seamless navigation and interaction with the web application.
 - Offer training sessions for administrators and content managers to manage and update the platform effectively.
- **Launch and User Engagement:**
 - Launch the web application and promote its availability among the target audience.
 - Gather user feedback and suggestions to continuously improve the platform's features and functionalities.
- **Continuous Maintenance and Upgrades:**
 - Regularly monitor the application's performance, security, and user engagement.
 - Implement updates, enhancements, and new features based on user feedback and emerging technologies.

By following this comprehensive methodology, the project aims to successfully develop and deploy the web application, creating an immersive and engaging platform for cricket enthusiasts in university settings.

3.1. Functional Requirements

Functional requirements cover the particular traits and skills that a system or application must have to effectively meet the needs of its users. These specifications outline not only what the system must do, but also how it must act. They are typically derived from sources such as user stories, use cases, and business requirements. Examples of functional requirements include tasks like user authentication, data validation, data manipulation, calculations, data storage, reporting, and any other precise actions the system is expected to perform.

3.2. Live Match Score and Details

In this section, users, spanning from regular users to administrators, are granted the ability to effortlessly access real-time live match scores and comprehensive match details without the necessity of authentication or sign-up. The system is tasked with offering up-to-the-minute updates on runs scored, wickets taken, overs bowled, and the required run rate for the enjoyment of all users.

3.3. Match Information

Accessible to both normal users and administrators, this section provides a window into comprehensive match information. This encompasses past matches, upcoming matches, and the overall match schedule. Pertinent details such

as the teams involved, venues, timings, and participating players are laid out for users without the hassle of authentication or sign-up requirements.

3.4. Sport News

Delightfully, users, whether they be regular users or administrators, are granted the privilege of diving into the realm of sports news and updates pertinent to university cricket sports. The system is endowed with the capacity to showcase well-sourced articles, news updates, and engaging narratives within its confines.

3.5. Sport Events

A harmonious blend of functionality and accessibility, this section allows users—ranging from regular users to administrators—to access a wealth of information regarding upcoming sports events. Cricket enthusiasts can uncover insights into match schedules, participating teams, event particulars, timings, venues, and the composition of team line-ups—all without the constraint of authentication or sign-up.

3.6. Admin Privileges

Herein, the admin role emerges as a key player, endowed with exclusive access to a plethora of additional features and capabilities. Administrative users are empowered to usher in new matches, update match scores in a granular ball-by-ball manner, publish sports news articles, introduce sports events, eliminate matches that no longer hold relevance, and even strike down redundant sports news and events. The admin's authority extends to bestowing these administrative privileges upon other deserving users. Notably, it is acknowledged that while the initial iteration of the Real-time Scorecard System operates without the need for authentication or sign-up, future iterations will integrate these features in tandem with the evolution of new functionalities.

3.7. System Design

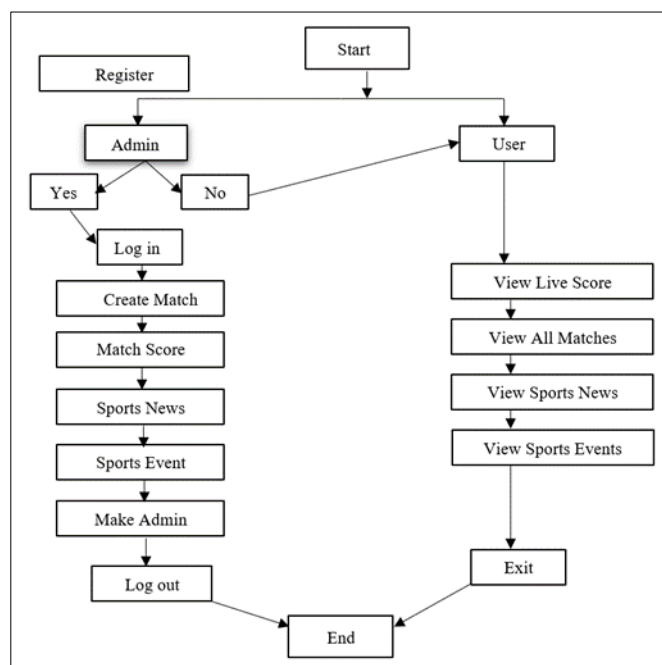


Figure 1 Flowchart of the System

Initially, users or administrators can access the website from their devices. Upon arrival, they can seamlessly input real-time cricket scoreboard queries that are shown in Figure 1. Once they locate the dynamic web application, a simple tap initiates it. A pivotal choice then emerges, presenting two distinct paths: 1) Admin and 2) User. Upon selecting the Administrator option, a subsequent decision between "Yes" and "No" arises. Opting for "Yes" prompts the input of essential login details, login as admin must be registered with some information like university name, university's website and email, etc. while "No" redirects to the user alternative. Subsequently, the Admin gains exclusive privileges, including match creation, real-time score updates during matches, provision of sports news and events, and the

authority to designate others as administrators. Culminating the interaction, the Administrator can elegantly exit the web application via a secure logout process.

On the other hand, choosing the User route effortlessly leads to live game score exploration, diverse game options, and access to game-related news and events, all achievable with a simple tap. Concluding the user experience, a straightforward "Exit" button gracefully concludes the application interaction.

3.8. Use Case Diagram of the System

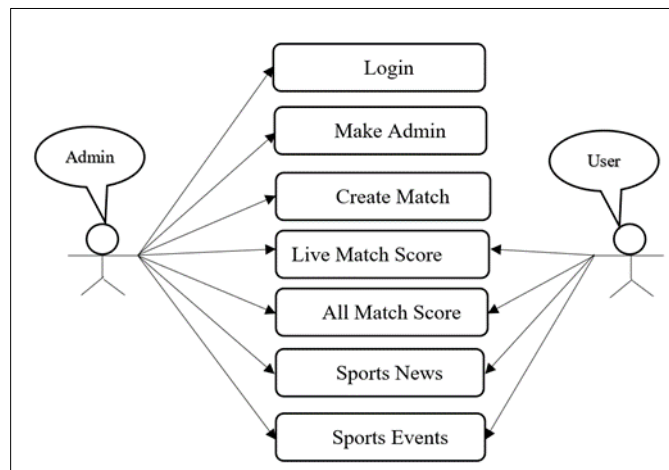


Figure 2 Use Case Diagram of the System

In Figure 2, The "Administrator" has exclusive access to actions like login, creating admins, creating matches, live match scores, all match scores, publishing sports news, and publishing sports events.

The "User" represents the app's regular users and can perform actions like viewing live match scores, all match scores, sports news, and sports events.

The arrows connecting the actors to use cases represent the interactions or actions they can perform within the system. This diagram provides an overview of the different actors and their interactions with the app's functionalities

3.9. ER Diagram

This project's Entity-Relationship (ER) diagram visually illustrates the interconnections between different entities in the system. It portrays the data flow and relationships within the web application. The ER diagram is designed based on the project's goals and methodology, and it can be summarized as follows:

The "Player" entity represents active cricket participants with attributes such as PlayerID, Name, Type, Runs scored, and Strike rate. These metrics assess players' skills and contributions. "Team" denotes competing groups with attributes like TeamID, Name, Captain, and Type. Captains lead and strategize. "Match" embodies specific matches with attributes like MatchID, Name, Toss, Venue, and Date. "Format" indicates match type (Test, ODI, T20), and "Result" records outcomes.

The "Admin" entity manages the application. AdminID, Name, Email, and Role are attributes. Admins oversee matches, news, and functionality. "Sports News" holds articles on cricket and sports events. NewsID, Title, Content, Date, and Source are attributes. It keeps users informed. "Sports Event" covers university sports with EventID, Title, Date, Location, and Description attributes. Users plan participation.

These entities create the app's foundation, organizing data on players, teams, matches, admins, news, and events. Their interactions shape the user experience, facilitating real-time scorecard updates, news, and event information.

In this web application, the relationships between entities and their associated attributes form the foundation of the system. These interconnected elements are essential for organizing data flow and interactions within the platform. Users play a central role by following matches, players, and news items, establishing a "Follows" relationship with respective entities. This personalized approach tailor updates to users' interests, boosting engagement. Cricket matches

thrive on player interactions. The "Participates In" relationship captures the involvement of multiple players from different teams in a match. Additionally, each match's scores, including runs and wickets, are connected through the "Has Scores" relationship for accurate recording and presentation. Contextualization is vital, with news and events linked to matches or players through the "Relates To" relationship, giving users broader insights. Entity attributes provide structured data storage. Users have attributes like UserID, Username, Email, and Password, while matches feature MatchID, Date, Location, Format, and Result. Players are defined by PlayerID, Name, Age, and Role, and scores have ScoreID, Runs, Wickets, Overs, and MatchID. News articles hold NewsID, Title, Content, and Date, and events include EventID, Title, Date, and Location. This interwoven network of relationships and attributes is depicted in the ER diagram, guiding the development of a user-friendly web application with real-time scorecards and engaging sports coverage. By embracing these elements, this project aims to create an immersive platform for cricket enthusiasts.

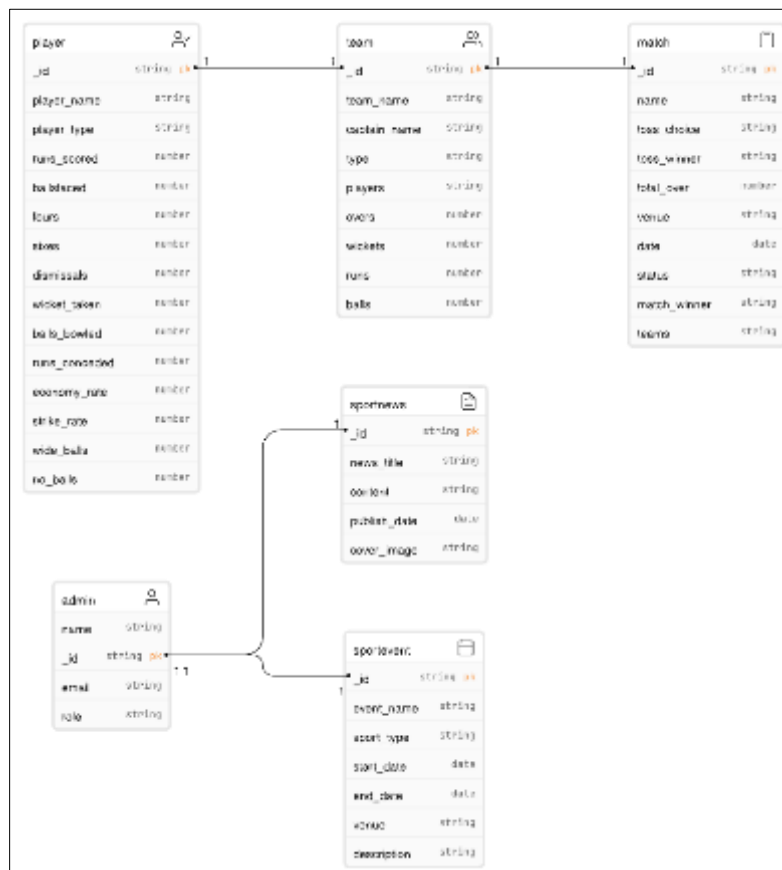


Figure 3 ER Diagram of the System

3.10. Agile Model

The development of this web application will follow an agile methodology, enabling iterative and incremental development cycles. The project team will leverage the MERN stack, incorporating MongoDB as the database, Express.js as the web application framework, React.js as the front end, and Node.js as the runtime environment.

In the Agile Model, the project would be approached through iterative and incremental development cycles. The project tasks would be divided into smaller, manageable units known as "user stories." According to the goals of the project, each user story would focus on a certain feature or functionality of the online application. Collaboration and communication between the team, stakeholders, and end users would be crucial throughout the development process. The development tasks would be organized into time-bound iterations called "sprints," typically lasting 1 to 4 weeks. At the beginning of each sprint, a planning meeting would be conducted to prioritize and select user stories for implementation during that iteration. During a sprint, the development team would work on designing, developing, testing, and integrating the selected user stories. Regular daily stand-up meetings would be held to discuss progress, address challenges, and ensure alignment among team members. At the end of the sprint, a "sprint review" would take place to showcase the completed user stories to stakeholders and gather feedback. The Agile Model places a strong emphasis on adaptability. Based on the feedback received during sprint reviews, the project's direction can be adjusted and refined. This iterative approach allows for flexibility in accommodating changing requirements, technological

advancements, and evolving user needs. Agile's iterative development process would be advantageous to the project since it would allow for ongoing improvement of the web application, frequent updates based on user feedback, and the delivery of additional value with each sprint. With this strategy, the project is kept in line with user expectations, maintains a high level of involvement, and encourages cooperation between the team and stakeholders.

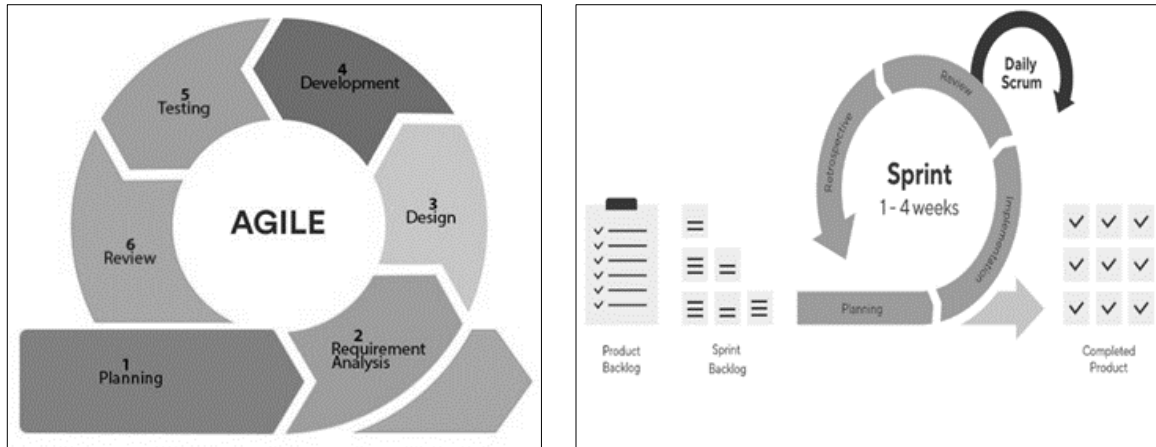


Figure 4 Agile Model

3.11. Hardware Requirements

The hardware requirements for the Real-time Scorecard System are contingent on user load, scalability needs, and performance expectations. The system can be hosted on servers or cloud platforms. Essential hardware components include the operating system (compatible with the MERN stack), a capable processor (Intel Core or better), sufficient memory (4GB minimum), and ample hard disk space (5GB minimum). These specifications facilitate efficient system performance, handling complex tasks, managing data, and accommodating multiple concurrent processes.

3.12. Software Requirements

In the software development process, several tools and software enhance productivity, streamline workflows, and promote collaboration. Notable tools include Visual Studio Code, a versatile source code editor; Postman, an API testing and management platform; MongoDB Compass, a GUI for working with MongoDB; web browsers for compatibility and standards adherence; package managers like NPM or Yarn for managing dependencies; and Git for version control and collaboration. Compatibility and proper configuration are crucial for ensuring the seamless operation of the system.

4. Project Implementation and Description

The implementation phase involves translating the project design and requirements into a functional system. It involves creating the essential parts, putting them together, and launching the application. An outline of the system implementation process is provided below:

- **Development Environment Setup:** Set up the development environment with the required tools and technologies, such as Node.js, React.js, and the MERN stack.
- **Backend Development:** Develop the server-side components using Node.js and Express.js. Implement APIs for handling user registration, authentication, fetching live match scores, retrieving sports news, managing events, and other functionalities as per the project requirements. Integrate MongoDB as the database using Mongoose for data storage and retrieval.
- **Frontend Development:** Build the user interface using React.js and the Material-UI library. Create components for displaying live match scores, sports news, event information, match details, and user interactions. Implement routing using React Router DOM to handle navigation and rendering of different views.
- **Integration and Testing:** Integrate the backend and frontend components, ensuring proper communication and functionality between them. Conduct integration testing to verify that the system components work together as expected. Validate the flow of data, user interactions, and real-time updates.
- **Deployment:** Prepare the application for deployment to a hosting environment, such as a cloud platform or a web server. Verify the user interactions, real-time updates, and data flow. Deploy the backend and frontend components, ensuring they are accessible to users.

- **Iterative Development:** Continuously iterate and improve the system based on user feedback, addressing any bugs or enhancements required.

4.1. Testing

The Real-time Scorecard System's quality and dependability must be ensured through testing. Alpha testing, beta testing, white-box testing, black-box testing, unit testing, integration testing, performance testing, security testing, and regression testing are among the main testing tasks.

Alpha testing involves controlled testing by the development team to uncover defects and usability issues before external testing. The focus is on validating system behavior and performance. Beta testing engages a limited group of external users to gather real-world feedback on usability and functionality. White-box testing examines internal structure and logic, utilizing techniques like code coverage analysis and unit testing. On the other hand, black-box testing concentrates on the system's inputs, outputs, and observable behavior, utilizing methods such as functional and system testing. Unit testing hones in on individual code components, ensuring they function as intended. Integration testing ensures seamless interaction between different system parts, validating data flow. Performance testing assesses responsiveness, stability, and scalability under different loads. Security testing aims to identify system vulnerabilities and weaknesses, safeguarding user data. Regression testing verifies existing functionalities remain unaffected after system changes. These testing methodologies collectively contribute to a comprehensive quality assurance strategy for the Real-time Scorecard System, enhancing its reliability and user satisfaction.

4.2 Project Snapshots

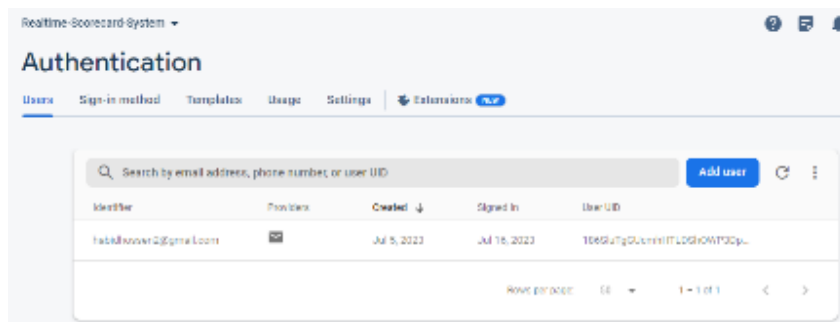
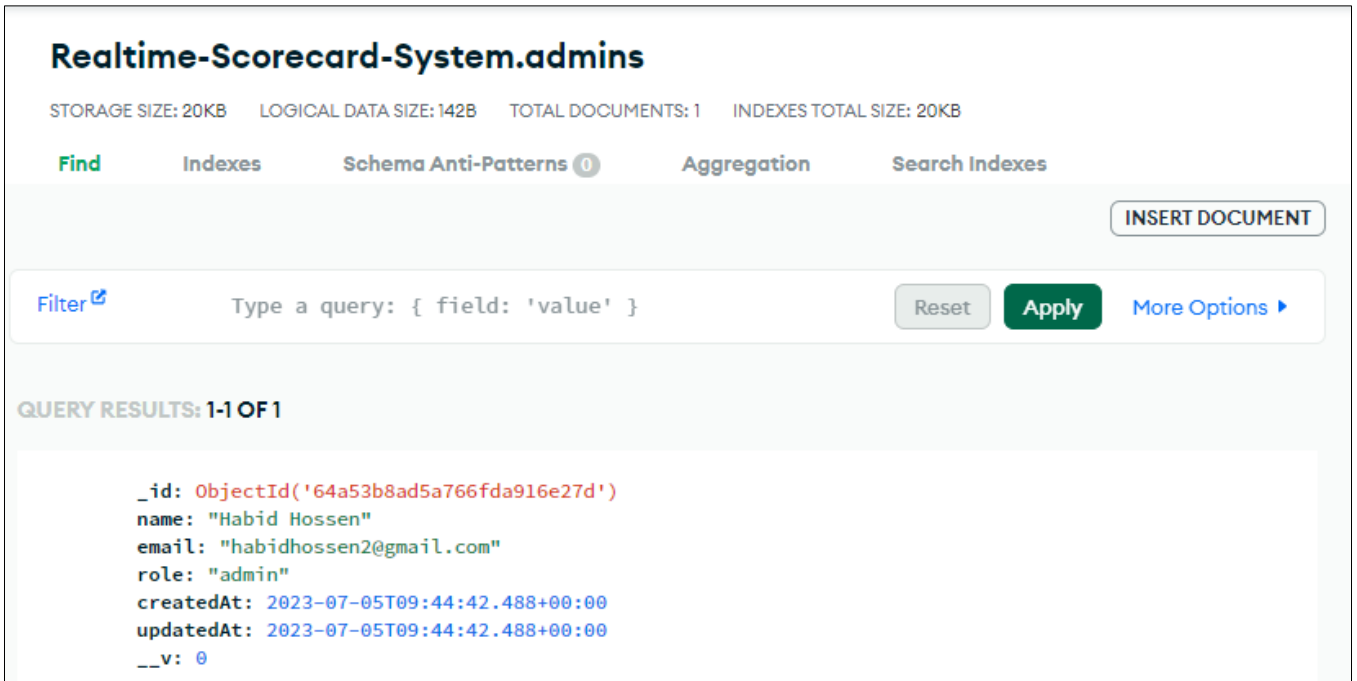


Figure 5 Authentication View

Figure 5 depicts the authentication format for administrators, displaying crucial user authentication details such as identifier, authentication provider, creation and sign-in dates, and associated user ID. The identifier serves as a unique user reference. The admin interface includes an "Add user" option to simplify user addition. Admins use the search bar to find and add new users by email, phone number, or user UID. This feature streamlines user identification and addition. Figure 6 (a) visually represents the systematic approach to managing user authentication and expanding the user base in the system.



a



b

Figure 6 (a,b) Admin View for Database Creation

In Figure 6 (b), the Database creation format within the admin interface of the Real-time Scorecard System is visually depicted. This figure acts as a guide for administrators, showcasing the schema and setup of the database. It illustrates tables, relationships, and attributes, providing a blueprint for how user data, performance metrics, and other pertinent information are organized and stored. These visuals aid administrators in understanding the backend architecture, enabling informed decision-making and effective system management.

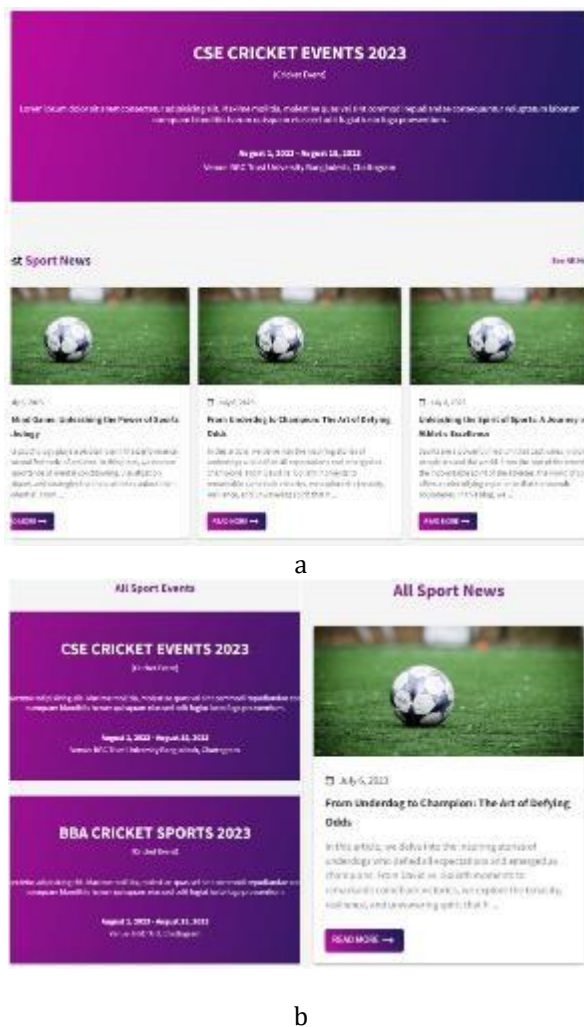


Figure 7 (a, b) Home page and Events View

Figure 7(a) provides a comprehensive view of the Front-end Home page, divided into five key segments. The first section guides users to ongoing live matches, enabling real-time engagement. The snapshot also presents recent live match results, ensuring timely updates. Detailed scorecards for recent matches enhance users' understanding of game statistics. Upcoming sports events are highlighted, keeping users informed about upcoming matches and tournaments. Additionally, the snapshot features a dedicated stream of the latest sports news, catering to users' need for current updates. Figure 7(b) offers a comprehensive display of all sports events, serving as a central hub for users to access information about a wide range of sporting activities.

5. Conclusion

"Ultimate Cricket Experience: Dynamic Web App for Real-Time Scoring System in University Cricket" introduces an innovative dynamic web application for real-time cricket scorecards, elevating the viewing experience for cricket enthusiasts. It offers live match updates, university sports news, detailed match information, and comprehensive statistics. Challenges like data accuracy, real-time updates, user engagement, scalability, and performance were effectively tackled during development. Leveraging technologies like React.js, Material UI, Express.js, Node.js, MongoDB, and Mongoose, the platform boasts real-time updates, user-friendly interfaces, efficient data storage, and seamless server-client communication. This implementation enables users to follow live cricket matches, access in-depth statistics, stay informed about university sports, and engage with the cricket community, enhancing the experience for university cricket fans. Looking ahead, the project envisions refining the user interface through continuous enhancements and user feedback integration. Advanced algorithms and visualization techniques will enrich statistical analysis capabilities while introducing social interaction and gamification elements that will foster community engagement. Extending accessibility through a mobile app will cater to smartphone users. Integration with third-party APIs will enhance features, and ongoing testing and performance optimization will ensure a seamless user experience. In summary, the successful execution of the project opens doors for future improvements in user interfaces, statistical

analysis, social components, mobile app development, API integration, and performance refinement. These strides will further benefit university cricket enthusiasts and nurture a thriving cricket community. In conclusion, while this project has significantly enhanced the cricket viewing experience, upcoming endeavors will concentrate on overcoming limitations, embracing technological advancements, and catering to a diverse user base. The application serves as a foundation for a more immersive, data-driven, and interactive cricket fandom, heralding a fresh era of sports engagement.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

- [1] Vaishnavi K. Nair, Raakhi Rachel Jose, Parvathy B. Anil, Minnu Tom, Lekshmy P.L.; ISSN: 2319-6386 (Online), Volume-6 Issue-7, May 2020. Automation of Cricket Scoreboard by Recognizing Umpire Gestures Crossref DOI link: <https://doi.org/10.35940/ijisme.G1235.056720> Published Online: 2020-05-15 International Journal of Innovative Science and Modern Engineering (IJISME).
- [2] Dhanusha T. John, Kavya S. Kumar, Vaishak T. Nair, P. Visakh, B. R. Poorna; July-2019 www.ijresm.com | ISSN (Online): 2581-5792 Cricket Scoreboard Automation using Umpire Gestures International Journal of Research in Engineering, Science and Management Volume-2, Issue-7.
- [3] Md. Asif Shahjalal, Zubaer Ahmad, Rushrukh Rayan, Lamia Alam 2017 3rd International Conference on Electrical Information and Communication Technology (EICT) An Approach to Automate the Scorecard in Cricket with Computer Vision and Machine Learning Date Added to IEEE Xplore: 01 February 2018 ISBN Information: INSPEC Accession Number: 17557499 DOI: 10.1109/EICT.2017.8275204 Publisher: IEEE Conference Location: Khulna, Bangladesh.
- [4] K. Kalai Selvan; Dr. V. Kavitha; Dr. N. Revathy; A. Sivaramakrishnan; IJCSMC, Vol. 8, Issue. 9, September 2019, pg.100 – 105 “Cricket Scorecard System” Available Online at www.ijcsmc.com International Journal of Computer Science and Mobile Computing ISSN 2320–088X Impact Factor: 6.199.
- [5] Rameshwari A. Lokhande and Pramila M. Chawan 16 July 2018. Prediction of Live Cricket Score and Winning International Journal of Trend in Research and Development, Volume 5(4), ISSN: 2394-9333 www.ijtrd.com.
- [6] Medha Wyawahare(B), Amol Dhanawade, Shreyas Dharyekar, Asavari Dhole, and Mugdha Dhopade Automating Scorecard and Commentary Based on Umpire Gesture Recognition Crossref DOI link: https://doi.org/10.1007/978-3-031-23092-9_13 Published Online: 2023-01-11 Published Print: 2022 Update policy: https://doi.org/10.1007/springer_crossmark_policy.
- [7] Eeshan Mundhe, Ishan Jain, Sanskar Shah; Oct 29-30, 2021 Live Cricket Score Prediction Web Application using Machine Learning 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON) | 978-1-6654-2503-2/21/\$31.00 ©2021 IEEE | DOI: 10.1109/SMARTGENCON51891.2021.9645855.
- [8] P.N.P. Fernando and G.N. Wikramanayake; 20 May 2014. Internet-Based Information System for ODI Cricket See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/216361330>.
- [9] Sanjay Joshi, Volume:05/Issue:04/April-2023 Impact Factor- 7.868 www.irjmets.com A SYSTEMATIC REVIEW OF MACHINE LEARNING APPLICATIONS IN CRICKET DOI: <https://www.doi.org/10.56726/IRJMETS37721> e-ISSN: 2582-5208 International Research Journal of Modernization in Engineering Technology and Science (Peer-Reviewed, Open Access, Fully Refereed International Journal).