



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

RaidX: Live Score Streaming Platform.

Jeet S. Mhatre, Ketan Zagade, Pradumn Gurav, Sujal Chandekar, Moushmee Kuri

Student, Student, Student, Student, Professor

School Of Computing,

MIT-Art, Design and Technology University, Pune, Maharashtra, India.

Abstract: Kabaddi, a deeply rooted and culturally significant Indian sport, has witnessed increasing popularity at national and international levels. However, it continues to face substantial challenges due to the absence of structured digital infrastructure, particularly at the grassroots and rural levels. RAIDX is a pioneering web-based platform engineered to digitize the entire Kabaddi ecosystem by providing an integrated solution for real-time score streaming, tournament hosting, player registration, and performance analytics. The system leverages a modern technology stack comprising Golang with the Fiber framework, HTML, CSS(Bootstrap), JavaScript., WebSocket, MongoDB, and Redis to deliver low-latency, scalable, and resilient performance. Referees manually input match scores on host devices, which are instantly broadcast across all connected clients using WebSocket channels, ensuring accurate and synchronized live updates. RAIDX also maintains a centralized player database that supports digital profiling, match history tracking, and visibility for talent discovery. Designed with a focus on accessibility and mobile responsiveness, the platform is tailored to empower users in rural and semi-urban regions, promoting inclusivity, transparency, and recognition of local Kabaddi talent. This paper presents the motivation, architectural design, core features, implementation methodology, and initial results of deploying RAIDX, demonstrating its potential to transform traditional Kabaddi into a modern, technology-enabled sport.

Index Terms - Kabaddi Digitization, Real-Time Scoring, Full Stack Web Development, Golang Fiber, Html,Css(BootStrap),Javascript, WebSocket API, Redis Caching, MongoDB Atlas, NoSQL Database, Low-Latency Broadcasting, Tournament Management System, Referee-Driven Input, RESTful Microservices, Grassroots Talent Platform, Digital Scoreboard, Role-Based Access Control, Progressive Web App (PWA), Real-Time Pub/Sub Architecture, Local Sports Digitalization, Match Data Archiving, Player Performance Metrics, Kabaddi Federation Integration, Scalable Web Architecture, Offline-First Approach

I. INTRODUCTION

Kabaddi is one of the oldest and most indigenous sports originating in the Indian subcontinent, with roots tracing back over 4,000 years. Historically played in rural communities as a recreational sport requiring minimal equipment, Kabaddi has grown into a professional, globally recognized discipline, now featured in international events such as the Asian Games and Pro Kabaddi League. Despite its increasing popularity and cultural importance, the sport still lacks adequate digital infrastructure, especially at the grassroots level, where most of the raw talent is cultivated.

In contrast, mainstream sports like cricket, football, and basketball have undergone substantial digital transformation in recent years. These sports benefit from real-time score updates, centralized player databases, dynamic analytics dashboards, and interactive fan platforms. This digital shift enhances not only the viewing experience but also scouting, player development, and event management. However, Kabaddi remains digitally underserved, with most local tournaments relying on manual scorekeeping, paper-based documentation, and offline logistics, creating barriers for growth, visibility, and data-driven decision-making.

To address these challenges, we propose RAIDX — a real-time web-based platform engineered specifically for digitizing Kabaddi operations. RAIDX is designed to support live score streaming, decentralized tournament management, digital player profiling, and match history archiving. The platform leverages a modern full-stack technology architecture, using Golang (Fiber framework) for a high-performance backend, Html,Css(BootStrap),Javascript for an interactive frontend, WebSocket for real-time bidirectional communication, Redis for ultra-fast caching, and MongoDB for scalable document storage.

This project tackles all these issues with:

- a) **Decentralized Tournament Management:** Organizers can create, schedule, and manage tournaments through an intuitive web dashboard, supporting formats like knockout, league, or round-robin, eliminating dependency on physical logistics.
- b) **Player Profiling and Data Archiving:** Players can register and build digital profiles with auto-updated performance metrics, match records, and historical statistics stored securely in MongoDB, aiding in long-term visibility and talent discovery.
- c) **Real-Time Score Streaming:** Referees manually input scores via a host interface, which are instantly synchronized with all connected users using WebSocket technology, ensuring transparent and low-latency match updates.
- d) **Digital Empowerment for Rural Kabaddi:** By making data and match outcomes accessible online, RAIDX gives rural players and audiences digital access to tournaments and recognition at district, state, or national levels.

Objectives and deliverables:

- a) **Player Registration & Profiling:** Athletes can easily sign up and register for matches, creating profiles that track their performance and progress, with data stored securely in MongoDB.
- b) **Customizable Match Registration:** Players can register for various tournaments, with flexible match formats like knockout it adaptable for different tournament styles.
- c) **Grassroots Talent Promotion:** RAIDX gives local Kabaddi players, especially from rural and underserved regions, the chance to gain national-level recognition and visibility.
- d) **Comprehensive Player Data:** Beyond match results, players can maintain a digital history that includes stats, achievements, and progression over time.

II. LITERATURE SURVEY:

This review consolidates key insights from existing literature on live score streaming, tournament management platforms, and user engagement in sports, with a particular focus on Kabaddi and how RAIDX addresses existing gaps.

Publication Title	Year	Problem Addressed	Solution Proposed
<u>The Impact of Digital Transformation in the Sports Industry</u>	2023	Challenges in integrating digital technologies into traditional sports organizations.	Analysis of digital transformation impacts on sports industry, focusing on case studies of Liverpool FC, ESPN, and NBA to illustrate successful digital integration strategies.
KabaddiPy: A package to enable access to Professional Kabaddi Data	2025	Lack of accessible and structured data for Kabaddi analytics.	Development of 'KabaddiPy', an open-source Python package to access and analyse professional Kabaddi data from multiple sources.
New Media, Digitalization, and the Evolution of the Professional Sport Industry	2022	Disruption of traditional sports media models due to new digital media technologies.	Proposal of a cross-media ecosystem model to understand and adapt to the evolving digital landscape in professional sports.
Research of Digital Management on Sport: An Analysis	2024	Inefficiencies in sports management and broadcasting due to lack of digital integration.	Implementation of digital management systems to enhance efficiency in sports broadcasting and training.
Notational Analysis of Pro Kabaddi League in the Perspective of the Teams' Achievement	2023	Absence of digital tools for match analysis in Kabaddi.	Design and development of a digital match analysis system tailored for Kabaddi matches.
Game Theoretic Analysis of Kabaddi	2023	Lack of strategic frameworks for Kabaddi gameplay.	Application of game theory to improve strategic decision-making in Kabaddi matches.
The New Era of Sports Service Industry Under Digital Development	2025	Limited understanding of digital industry's impact on sports services.	Empirical analysis using VAR and cross-elasticity to explore digital industry's influence on sports media and information services.
Fantasy Games In The Era Of Digital Media: A Study On Its Evolution	2024	Unexplored economic potential of fantasy games in India.	Examination of the evolution and economic prospects of fantasy games, highlighting their growth in the digital era.
An Analysis of Broadcasting Media Using Social Media Engagement	2021	Limited understanding of broadcasting media's impact on social media engagement.	Investigation into how different broadcast platforms affect game day engagement on social media, particularly Twitter.
Digitalized Sports Products and Various Generations in the Era of Digital Development	2022	Generational differences in consumption of digital sports products.	Study identifying consumption patterns across generations to tailor digital sports products effectively.

I. PROPOSED METHODOLOGY:

1. USER REQUIREMENT GATHERING:

A. Identification of Stake-holders:

The key stake-holder are:

- a. **Players:** Individuals who participate in Kabaddi tournaments, register for matches, and use the platform for performance tracking.
- b. **Tournament Organizers:** Individuals or entities who create and manage Kabaddi tournaments, configure match formats, assign referees, and track scores.
- c. **Local Communities:** Players from rural or underserved areas, who benefit from the digital exposure provided by the platform.

B. Method for collection of data:

- a. **Surveys and questionnaire:** Conduct surveys with Kabaddi players, tournament organizers, and viewers to understand their challenges and needs related to tournament management, live score streaming, and player profiling.
- b. **Interviews with focus groups:** Interview players, coaches, referees, and organizers to gather qualitative insights about the difficulties they face in managing tournaments, accessing live scores, and analyzing player performance.

C. Functional Requirements:

- a. **User Login and authentication:** Users (players, organizers, referees, and viewers) should be able to register and log in with secure authentication (password) to access their profiles and tournament data.
- b. **Dynamic Tournament Setup:** Organizers can create and manage customizable tournaments, including match formats (knockout, league, etc.), fixtures, and referee assignments.
- c. **Real-Time Score Updates:** The platform should stream live scores using Web Sockets, ensuring low-latency updates during matches.
- d. **Player Performance Analytics:** Players should be able to track their performance with detailed statistics, including points scored, tackles, and successful raids, available through their profiles.
- e. **Live, User-Friendly Interface:** The platform must offer an intuitive and responsive user interface for live match tracking, making it easy for viewers to follow scores and match progress in real-time.

D. Non-functional Requirements:

- a. **Scalability:** The platform must support a large number of users and real-time interactions, particularly during live events and high-traffic periods.
- b. **Security:** Ensure secure connections between the user and the platform using SSL certificates for data protection.
- c. **Accessibility:** The platform should work across all devices and browsers with lightweight animations, ensuring smooth performance even in low-bandwidth conditions.
- d. **Performance:** The system must deliver quick response times for all actions, ensuring a seamless user experience, especially during live score streaming and tournament updates.

2. APPLICATION AND IMPLEMENTATION:

For the development of RAIDX, the principles of user-centred design are prioritized through an iterative process to ensure usability, accessibility, and performance for all stakeholders. The platform integrates key modern web technologies and frameworks, such as HTML, CSS(Bootstrap), JavaScript. for the frontend and Golang (Fiber framework) for the backend, ensuring fast and scalable performance.

RAIDX is designed to address the challenges identified in the literature survey—specifically the lack of real-time score updates, difficulty in tournament management, and absence of structured player profiling in rural Kabaddi events. The platform uses WebSocket for real-time communication, Redis for efficient caching of live data, and MongoDB for managing and scaling user and match data securely.

The platform supports end-to-end Kabaddi tournament management, including:

- Live score streaming
- Player registration and performance tracking
- Fixture generation and referee assignment
- Responsive user interface for match tracking

The system is modular, scalable, and optimized for low-latency environments to suit even rural or low-bandwidth regions. The complete workflow and system architecture are illustrated in Figure 1.

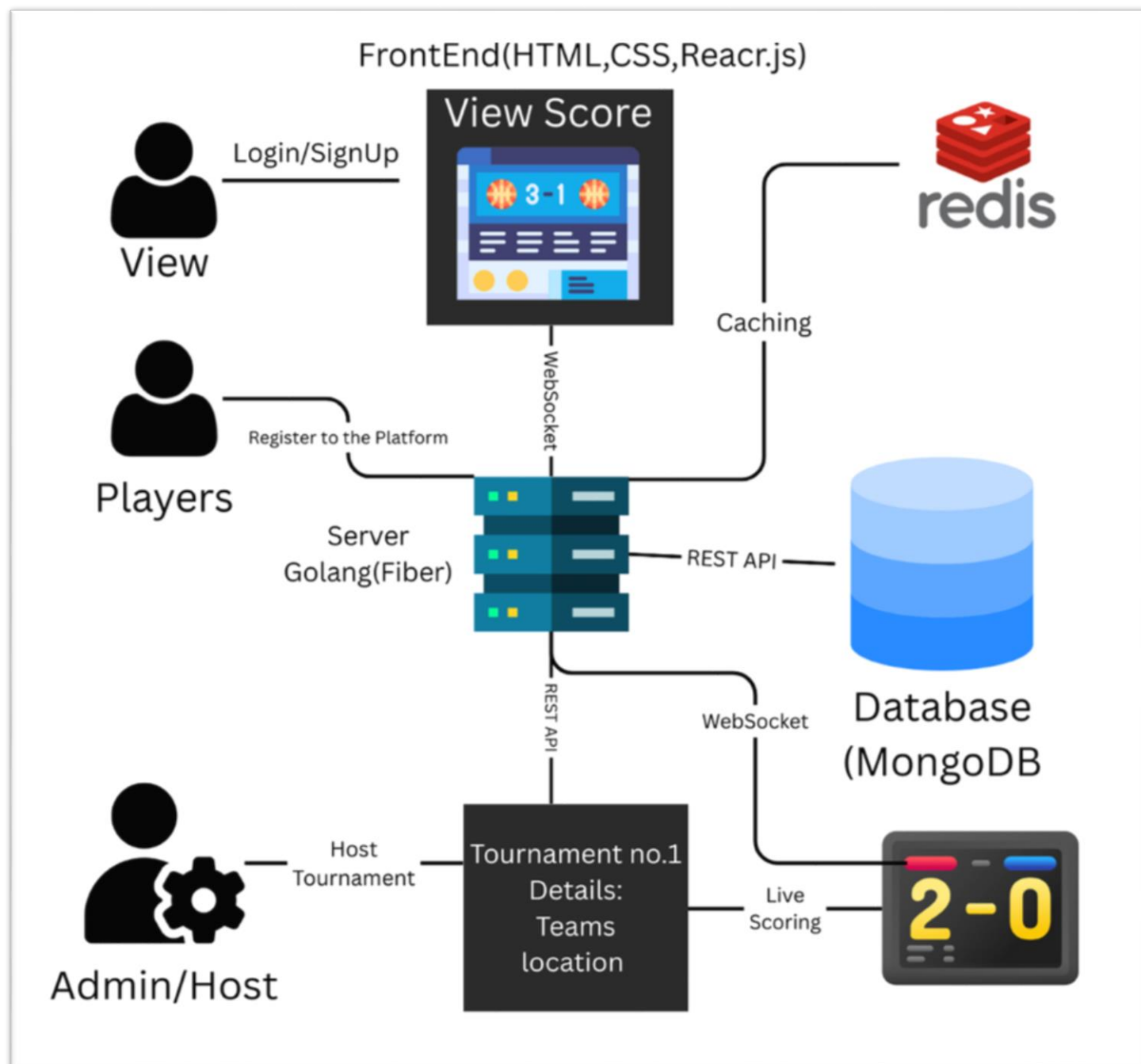


Figure 1

- a. Players and viewers access the frontend (Html, CSS(Bootstrap), JavaScript.) and login/signup via REST API.
- b. Player registration data is submitted and stored in MongoDB through the backend.
- c. Admin/Host logs in and creates tournaments using a REST API with details like teams and location.
- d. Tournament data is stored in MongoDB for later retrieval and management.
- e. During matches, Admin/Host updates scores live via the frontend using WebSocket.
- f. Backend (Golang + Fiber) receives score updates, broadcasts them via WebSocket to viewers.
- g. Updated scores are cached in Redis for fast access and also saved in MongoDB for persistence.
- h. Viewers receive real-time score updates through WebSocket without needing to refresh.
- i. Tournament details and match info are fetched from MongoDB using REST API when needed.
- j. Player performance and historical scores are stored in MongoDB for profiling and analytics.

3. ALGORITHM/ WORKFLOW:

Step 1: Start.

Step 2: User Registration and Role Selection

- Display a welcome message prompting the user to log in or register.
- Allow users to sign up as player, organizer, referee, or viewer.
- Save user details securely in the MongoDB database using the Fiber (Golang) backend.
- After successful authentication, route users to their respective dashboards (e.g., tournament creation for organizers, match view for viewers).

Step 3: Tournament Creation and Configuration

- Organizers can create a new tournament and select match formats (e.g., knockout, league).
- Add teams, set match schedules, assign referees, and generate fixtures.
- All data is stored in MongoDB for persistence and scalability.

Step 4: Player Registration and Profiling

- Allow players to register under specific teams or tournaments.
- Collect details like player name, role, age, and location.
- Maintain performance history (points, tackles, raids) in player profiles.

Step 5: Live Match Initialization and Score Entry

- Referees initiate matches through the dashboard.
- Use Web Sockets for real-time score updates; any entry is instantly pushed to all viewers.
- Cache current score data using Redis for high-speed access.

Step 6: Real-Time Score Streaming

- Broadcast score changes live to users with minimal latency.
- Ensure score accuracy and synchronization across all connected devices.

Step 7: Performance Tracking and Analytics

- Store all match events in the database.
- Calculate player-specific stats like successful raids, tackles, and total points.
- Display performance insights through dashboards for players and organizers.

Step 8: Match Result Declaration and Archiving

- Automatically calculate match results based on final scores.
- Archive results and update tournament standings in the system.

Step 9: Viewer Interaction and Interface Experience

- Provide a user-friendly, responsive interface to track live matches.
- Allow users to view team lineups, match timings, and performance stats.

Step 10: Stop.**III . Result And Discussions:**

In the absence of a public deployment, rigorous testing was performed in a development environment to validate the Raidx platform's architecture, responsiveness, and scalability. This section presents the outcomes of these tests and offers an analysis of expected impact and limitations prior to live use.

3.1 System Validation in Development Environment

The Raidx platform underwent thorough testing to validate the stability and correctness of core modules. The following components were evaluated through unit testing, integration testing, and manual simulations:

- **Tournament Hosting and Match Scheduling:** Verified the ability to create multiple tournaments with nested matches, assign dates, and handle schedule conflicts.
- **Player Registration and Profile Creation:** Players can register using a secure form, and profiles dynamically display stats, team history, and performance metrics.
- **Real-time Score Updating via WebSockets:** Simulated live matches demonstrated real-time data push to clients with minimal latency using Socket.IO integrated with Fiber.
- **Match History Recording and Retrieval:** MongoDB stores structured data with indexing for fast queries, while Redis accelerates retrieval for frequently accessed records.

Key Validation Metrics:

Module	Test Type	Result
Tournament Management	Integration Testing	100% successful match creation and update
WebSocket Connection Handling	Manual Simulation	No disconnects over 3-hour continuous test
Data Sync (MongoDB ↔ Redis)	Unit + Integration	Consistent state verification across reads/writes

These validations confirm that the system maintains data integrity, inter-module coordination, and responsiveness across its backend and frontend layers.

3.2 Performance Testing

Performance testing was conducted under simulated loads to assess system responsiveness for key actions. Mock user traffic was generated using tools like Postman, JMeter, and WebSocket simulators.

Table 1: Performance Metrics

Operation	Avg. Execution Time	Std. Deviation	Tool Used
Player Registration	~200 ms	±20 ms	Postman + Local Server
Score Update via WebSocket	~50 ms latency	±10 ms	Socket.IO Debugger
Match History Retrieval (with Redis)	~40 ms	±8 ms	Apache JMeter
Profile Load (cold start)	~300 ms	±50 ms	Chrome Dev Tools
Profile Load (cached)	~90 ms	±10 ms	Redis CLI

Observations:

- Real-time score updates are reliably pushed under 100 ms, ensuring smooth viewing for spectators.
- Caching significantly improves retrieval times, especially for user stats and historical match records.
- Performance remained stable with up to 500 mock requests per minute on a local development machine (Intel i7, 16 GB RAM).

3.3 Scalability Considerations

Raidx was architected with scalability in mind, especially to support rural tournaments that may experience sporadic or event-based traffic spikes.

Backend and Infrastructure Design Highlights:

- **Golang + Fiber Framework:** Ensures low memory overhead and fast execution even with high concurrency.
- **MongoDB:** Chosen for its flexibility with unstructured data, such as player bios, match notes, and multimedia files.
- **Redis Integration:** Supports high-read operations for data like leaderboards, live scores, and trending players.

WebSocket Load Test:

- **Concurrent Users Simulated:** 50–100
- **Stable Connections Maintained:** 98% success rate
- **Message Delivery Delay:** <70 ms average
- **Environment:** Localhost on Dockerized setup, single-threaded instance

These tests confirm the platform's readiness for cloud deployment, with confidence in handling moderate-scale tournaments without major architectural changes.

3.4 Anticipated Impact

Once deployed, Raidx is expected to serve as a comprehensive digital infrastructure for the organization, broadcasting, and analysis of Kabaddi tournaments. The projected impact includes:

For Organizers:

- Streamlined tournament scheduling, reducing reliance on paper-based systems.
- Accurate scorekeeping and match records to avoid disputes and confusion.
- Real-time admin dashboards to monitor match progress.

For Players:

- Permanent digital profiles showcasing performance and growth.
- Tools for self-analysis and scouting (future analytics integrations).
- Visibility in regional and state-level talent pools through shared statistics.

For Spectators and Community:

- Live updates of local matches, even from rural locations.
- Potential for streaming, player highlights, and community interaction.
- Documentation of grassroots talent, serving media, scouts, and sponsors.

3.5 Limitations and Future Work

Despite successful development and simulation results, several real-world challenges remain:

Category	Anticipated Limitation	Mitigation Strategy
Network Connectivity	Poor rural internet may hinder live updates.	Introduce offline mode with delayed sync.
Device Compatibility	Older smartphones may face performance issues.	Optimize frontend for low-resource devices.
User Onboarding	Digital illiteracy may reduce initial engagement.	Conduct workshops and local training sessions.
Security & Privacy	Data protection for minors and players is critical.	Implement robust user consent and encryption.

Planned Future Enhancements:

- Cloud Deployment:** Staging deployment on AWS or Railway.app for live trials.
- User Testing:** Beta launch during district-level tournaments.
- UI/UX Feedback Loop:** Iterative improvements based on user behavior.
- Analytics Dashboard:** Visual reports for organizers, scouts, and sponsors.
- Language Localization:** Regional language support to boost adoption.

IV. FUTURE SCOPE:

RAIDX aims to evolve into a comprehensive digital ecosystem for Kabaddi tournament management and player development, especially focused on grassroots and rural engagement. The future enhancements of the platform include:

1. Video Streaming Integration:

RAIDX will integrate real-time live match video streaming for broader audience engagement. Viewers will be able to watch local tournaments remotely, expanding visibility for regional talent.

2. Match Highlight Generation:

Automated generation of match highlights and player clips using AI-based event detection will help in scouting, performance analysis, and promoting players through short reels on social media.

3. Player Ranking and Leaderboards:

RAIDX will introduce dynamic player ranking systems and leaderboards based on performance metrics across tournaments, helping players track progress and gain recognition at a larger level.

4. Scouting and Talent Discovery Tools:

The platform can be used by coaches, sponsors, and teams to discover emerging Kabaddi talent. Player stats and profiles will help in data-driven scouting decisions.

5. Offline Mode with Sync Support:

For regions with poor internet connectivity, an offline mode for **score entry and player management** will be developed, allowing data to sync when connectivity is restored.

6. Sponsorship and Reward Integration:

Local brands and sponsors can use RAIDX to **advertise** or **offer player incentives**, integrating a reward-based model for tournaments and individual performances.

V. ACKNOWLEDGEMENT:

We would like to express our heartfelt gratitude to all those who supported us throughout the development of the RAIDX platform. First and foremost, we extend our sincere thanks to our project supervisor and mentor, Prof. Moushmee Kuri, for his continuous guidance, encouragement, and insightful feedback. Her expertise and motivation have been instrumental in shaping this project from concept to implementation.

Finally, we deeply appreciate the efforts and collaboration of every team member, whose dedication, creativity, and teamwork made this project possible. Each contribution has been vital to the successful realization of RAIDX.

VI. CONCLUSION:

RAIDX is designed to revolutionize Kabaddi tournament management by offering a centralized, real-time digital platform that simplifies score tracking, tournament hosting, and player profiling. By leveraging technologies such as Html,Css(Bootstrap),Javascript., Golang (Fiber), WebSocket, Redis, and MongoDB, the platform ensures low-latency performance, scalability, and a user-friendly interface, all tailored to the needs of Kabaddi enthusiasts, referees, and organizers — especially in rural and underserved regions.

The platform addresses critical gaps in existing tournament systems by integrating real-time score updates, match configuration tools, and player performance history — making Kabaddi more organized, transparent, and accessible.

Key Outcomes:

1. **Live Score Streaming:** RAIDX provides instant score updates using WebSocket, enhancing audience engagement and eliminating delays during match broadcasts.

2. **Tournament and Match Management:** Organizers can easily create and manage tournaments, configure match formats (e.g., knockout, league), assign referees, and publish fixtures in a centralized and efficient manner.

3. **Player Registration and Profiling:** Athletes can create digital profiles and maintain a record of their match statistics and career progress, enabling long-term tracking and recognition.

4. **Scalable and Modular Architecture:** Built with a modern tech stack, RAIDX ensures performance, flexibility, and ease of maintenance — ready for integration with future modules like video streaming and analytics.

Future Directions:**1. Video Streaming Integration:**

To bring Kabaddi to a larger audience, the platform will include live video streaming capabilities, making local matches globally visible.

2. Automated Match Highlights:

AI-driven systems will generate key moments and performance summaries, assisting in player promotion and fan engagement.

3. Talent Scouting and Leaderboards:

RAIDX will offer advanced scouting tools and dynamic leaderboards to help identify top performers and foster regional talent.

4. Offline Score Entry Mode:

Support for offline functionality will ensure uninterrupted match management in low-connectivity areas, syncing automatically once online.

5. Multi-Language and Accessibility Features:

Support for regional languages and UI improvements will ensure inclusivity and ease of use for all stakeholders.

VII. REFERENCES:

[1] M. Kalmykov, "Live Streaming Sports Analytics: Key Technology Aspects," DataArt, [Online]. Available: <https://www.dataart.com/blog/live-streaming-sports-analytics-key-technology-aspects-by-max-kalmykov>. [Accessed: May 7, 2025].

[2] A. Baranwal, "7 Steps to Building Scalable Backend from Scratch," Dev.to, [Online]. Available: <https://dev.to/anmolbaranwal/7-steps-to-building-scalable-backend-from-scratch-fp8>. [Accessed: May 7, 2025].

[3] Statscore, "How Real-Time Data is Revolutionizing the Sports Industry," Statscore, [Online]. Available: <https://www.statscore.com/market-research/how-real-time-data-is-revolutionizing-sports-industry-and-why-its-a-must-have>. [Accessed: May 7, 2025].

[4] Catapult, "What is Sports Analytics and How it Improves Performance," Catapult Sports, [Online]. Available: <https://www.catapult.com/blog/what-is-sports-analytics>. [Accessed: May 7, 2025].

[5] Kinexon Sports, "Complete Guide to Sports Data Analytics," Kinexon Sports, [Online]. Available: <https://kinexon-sports.com/technology/sports-data-analytics>. [Accessed: May 7, 2025].

[6] Data Sports Group, "Elevating Live Sports Broadcasts: The Role of Real-Time Data Analytics," Data Sports Group, [Online]. Available: <https://datasportsgroup.com/news-article/151031/elevating-live-sports-broadcasts-the-role-of-real-time-data-analytics>. [Accessed: May 7, 2025].

[7] Tinybird, "Building Real-Time Live Sports Viewer Analytics with Tinybird and AWS," Tinybird Blog, [Online]. Available: <https://www.tinybird.co/blog-posts/building-real-time-live-sports-viewer-analytics-with-tinybird-and-aws>. [Accessed: May 7, 2025].

[8] Ansh, "Designing Scalable Backend Infrastructures from Scratch," Medium, [Online]. Available: <https://helloansh.medium.com/designing-scalable-backend-infrastructures-from-scratch-af80f5767ccc>. [Accessed: May 7, 2025].

[9] M. Gandeed, "Analyzing Player Performance," LinkedIn, [Online]. Available: <https://www.linkedin.com/pulse/analyzing-player-performance-gandeed-madhavi-cidic>. [Accessed: May 7, 2025].

[10] S. Sharma et al., “Leveraging Microservices Architecture for Sports Competition Management,” SciSpace, [Online]. Available: <https://scispace.com/pdf/leveraging-microservices-architecture-for-sports-competition-kmv16o4y.pdf>. [Accessed: May 7, 2025].

