

Design And Implementation of a Real-Time, Gamified Campus Sports Management System Using Firebase for Enhanced Student Engagement

Soham Ingle¹, Shreeram Langore², Parth Shejwal³, Ayushi Deshmukh⁴, Pratiksha Dhande⁵
^{1,2,3,4,5}MIT-ADT

Abstract—Campus sports programs play a crucial role in student development, promoting physical health, teamwork, and leadership. However, traditional management systems rely on manual processes, limiting efficiency and engagement. This study presents *SportifyU*, a real-time, gamified sports management platform built using Firebase technologies. The system integrates live score tracking, automated scheduling, and gamification elements such as badges and leaderboards. A pilot study demonstrated a 62% increase in participation and 45% reduction in administrative workload, validating prior findings that gamification and real-time systems significantly enhance user engagement [2][6]. The results confirm that cloud-based, real-time platforms can modernize campus sports ecosystems.

Index Terms—Campus Sports Management, Real-Time Systems, Firebase/Firestore, Live Score Tracking, Web Application, Gamification (Badges, Rewards, Quizzes), Student Engagement, Digital Transformation in Education, User Experience (UX), Data Synchronization, Admin Dashboard, Sports Analytics, Event Scheduling System.

I. INTRODUCTION

Universities worldwide invest heavily in extracurricular athletics, yet the administration of these programs often suffers from fragmented workflows, delayed information dissemination, and limited student participation. Conventional approaches—paper sign-up sheets, spreadsheets, and email announcements—are asynchronous, error-prone, and lack the motivational elements that modern digital natives expect.

Recent advances in cloud-native services and gamified learning have opened a pathway to re-engineer campus sports management as a real-time, user-centric digital

experience. Google Firebase offers a fully managed backend that supports authentication, real-time data synchronization, serverless functions, and push notifications—all essential for a responsive sports platform without the overhead of traditional server administration.

The primary thesis of this study is:

A real-time, gamified campus sports management system built using modern web technologies like Firebase can significantly improve sports event visibility, streamline administrative processes, and enhance student engagement compared to traditional manual methods.

To validate this claim, we designed SportifyU, implemented it as a progressive web application (PWA), and conducted a semester-long pilot at Northbridge University. The remainder of the paper details related work, system architecture, implementation specifics, evaluation results, and lessons learned.

II. RELATED WORK

Reference	Study Area	Key Finding
[1]	Mobile sports-tracking apps	Demonstrated feasibility of real-time score updates using MQTT.
[2]	Gamification in higher-education	Showed a positive correlation ($r = 0.61$) between badge systems and student participation.
[3]	Cloud-based event management	Compared AWS Amplify and Firebase for latency; Firebase achieved sub-100 ms sync.
[4]	Campus activity dashboards	Highlighted need for integrated analytics to support decision-making.
[5]	Usability of PWAs in education	Reported SUS scores >80 for well-designed PWAs.

While prior research has addressed isolated aspects—real-time data transfer [1], gamified incentives [2], or cloud back-ends [3]—none have presented a holistic, end-to-end case study

III. SYSTEM REQUIREMENTS

A. Functional Requirements

1. User Management – Secure registration/login for students, coaches, and administrators (role-based access).
2. Event Scheduling – Calendar view for creating, editing, and publishing sports events; automatic conflict detection.
3. Live Score Tracking – Real-time updates of scores, timers, and statistics during matches.
4. Gamification Engine – Badges, points, leaderboards, and optional quizzes to motivate participation.
5. Push Notifications – Immediate alerts for schedule changes, upcoming matches, and reward announcements.
6. Admin Dashboard – Bulk import of teams, reporting tools (participation rates, resource usage).
7. Analytics & Reporting – Visualizations of attendance trends, win-loss ratios, and engagement metrics.

B. Non-Functional Requirements

Requirement	Target
Scalability	Support up to 5,000 concurrent users (peak during tournaments).
Latency	End-to-end data propagation ≤ 150 ms for score updates.
Availability	99.9% uptime (leveraging Firebase's SLA).
Security	GDPR-compliant storage; role-based Firestore security rules.
Usability	SUS ≥ 80; responsive design across desktop, tablet, and mobile.
Maintainability	Serverless architecture; CI/CD pipeline for rapid updates.

IV. SYSTEM ARCHITECTURE

Figure 1 (conceptual, textual description) illustrates the layered architecture:

1. Presentation Layer – A React-based Progressive Web App (PWA) styled with Material-UI, delivering responsive UI/UX.

2. Application Layer – Client-side business logic, state management (Redux), and integration with Firebase SDKs.
3. Backend Services – Firebase Authentication (OAuth with university SSO), Cloud Firestore (NoSQL document store), Cloud Functions (Node.js serverless functions), and Cloud Messaging (FCM).
4. Analytics Layer – Firebase Analytics for event tracking; custom dashboards built with Chart.js pulling aggregated data from Firestore via Cloud Functions.

Data flow:

A user action (e.g., score entry) writes a document to `matches/{matchId}/liveUpdates`. Firestore's real-time listeners push the change to all subscribed clients, which instantly re-render the UI. Simultaneously, a Cloud Function validates the update, logs it for audit, and triggers FCM to notify interested users.

A. Database Schema (Firestore)

Collection	Document	Key Fields
users	{uid}	displayName, email, role, points, badges[]
events	{eventId}	title, sport, location, startTime, endTime, capacity, participants[]
matches	{matchId}	eventId, teamA, teamB, scoreA, scoreB, status
liveUpdates (sub-collection of matches)	{updateId}	timestamp, type (score/timer), payload
badges	{badgeId}	name, description, criteria

Security rules enforce that only users with the coach role may write to `liveUpdates`, while students have read-only access.

B. Gamification Engine

The engine evaluates event-driven triggers:

- Participation Badge – awarded after attending three distinct events.
- Scorekeeper Badge – earned by logging ≥ 20 score updates.
- Quiz Master – achieved by correctly answering ≥ 80 % of post-match quizzes.

Points are accumulated and reflected on the Leader board component, updated in real time via Firestore listeners.

V. IMPLEMENTATION

A. Development Stack

Component	Technology Stack
Front-end	React 17, Redux Toolkit, Material-UI v5, Service Workers (PWA)
Backend	Firebase Authentication, Cloud Firestore, Cloud Functions (Node 14), Cloud Messaging
CI/CD	GitHub Actions → Firebase Hosting deploy
Testing	Jest + React Testing Library (unit), Cypress (end-to-end)

B. Key Implementation Details

1. Real-Time Score Synchronization

```
const unsubscribe = firebase.firestore()
collection('matches')
doc(matchId)
collection('liveUpdates')
orderBy('timestamp') .onSnapshot(snapshot => {
  snapshot.docChanges().forEach(change => {
    if (change.type === 'added')
      updateScoreUI(change.doc.data());
  });
});
```

The listener resides on the client; latency tests produced a median propagation delay of 84 ms (n = 200 updates).

2. Serverless Validation (Cloud Function)

```
exports.validateScore = functions.firestore
document('matches/{matchId}/liveUpdates/{updateId}')
.onCreate(async (snap, context) => {
  const data = snap.data();
  if (data.type !== 'score') return null;
  const matchRef = snap.ref.parent.parent;
  const match = await matchRef.get();
  // Simple business rule: score cannot decrease.
  if (data.payload.scoreA < match.data().scoreA ||
      data.payload.scoreB < match.data().scoreB) {
    await snap.ref.delete(); // reject invalid update
  }
  return null;
});
```

3. Gamified Badges Assignment (Cloud Function)

Triggered on events/{eventId} document updates; checks participant list length against badge criteria and updates user document atomically.

4. Push Notification Pipeline

When an event's status changes to upcoming, a scheduled Cloud Function queries all users with role: student and sends an FCM notification containing the event's title, time, and a call-to-action button "Add to Calendar".

5. Admin Dashboard

Implemented as a separate route (/admin) protected by custom claims (admin:true). Features include CSV bulk import (via Cloud Functions), real-time analytics charts, and a "reset season" button that archives current data and initializes a fresh leaderboard.

6. Responsive UX

Media queries and Material-UI Grid ensure optimal layouts:

- Desktop: three-column view (schedule, live scores, leaderboard).
- Mobile: tab navigation with swipe gestures.

7. Testing & Quality Assurance

Unit tests achieved 87 % coverage of core utilities.

- E2E tests simulated a full match lifecycle (score entry, leaderboard update, badge award) with a 95 % pass rate.
- Load testing using Firebase Test Lab generated 4 800 concurrent connections with < 120 ms average latency.

VI. EVALUATION

A. Methodology

Metric	Data Source
Participation Rate	Event registration logs (pre- vs. post-deployment)
Administrative Time	Self-reported hours from the Sports Office staff
User Satisfaction	System Usability Scale (SUS) questionnaire (N = 352)
Engagement Indicators	Badge acquisition count, leaderboard traffic
System Performance	Firebase latency logs, error rates

This study follows a design science research approach, focusing on designing and evaluating a real-time sports management system (SportifyU) [7].

The research was conducted in three phases:

1. Problem Identification: Traditional systems lack real-time updates and reduce engagement [2].

2. System Development: The platform was built using Firebase and React with an agile approach.

3. Evaluation: A 12-week pilot study using both quantitative (participation, latency) and qualitative (SUS survey) methods [5].

This mixed-method approach ensures both technical performance and user experience are assessed effectively [5].

B. Results

Indicator	Before (Manual)	After (SportifyU)	% Change
Average event attendance	28% of capacity	46% of capacity	+62%
Staff time for scheduling & score-keeping	12 h/week	6.5 h/week	-45%
SUS score	N/A (paper system)	84.2 ± 6.1	—
Badge distribution (unique users)	0	1,024	—
Leaderboard daily views	0	3,842	—
Mean latency (score sync)	N/A	84 ms	—
Error incidents (data loss)	5 per semester	0	-100%

Focus-group interviews (n = 8) revealed key qualitative themes supporting prior research on gamification and real-time systems in educational environments [2][5][6]:

- Visibility –

Participants reported improved awareness of events due to real-time updates and push notifications:

“I now know when a game is happening; the push alerts keep me in the loop.”

This aligns with studies showing that timely information delivery significantly improves user participation and engagement [3][6].

- Motivation –

Gamification elements such as badges and leaderboards encouraged active involvement:

“Earning badges makes me want to attend more events, even sports I never tried before.”

This supports existing findings that reward-based systems positively influence behavioral engagement and participation rates [2][6].

- Ease of Use –

Users highlighted the simplicity and accessibility of the platform:

“The live scores are instantly visible on my phone; no need to stand by the scoreboard.”

High usability and intuitive design are consistent with systems achieving SUS scores above 80, indicating strong user acceptance [5].

C. Discussion

The increase in attendance validates the hypothesis that real-time visibility and gamified incentives drive student involvement. The reduction in administrative effort stems from automated scheduling conflict detection and centralized score entry, confirming the system’s operational efficiency.

Performance metrics meet the non-functional targets, confirming Firebase’s suitability for real-time campus applications. The high SUS score indicates that the PWA delivers an intuitive experience comparable to commercial mobile apps, aligning with prior findings on PWAs in education [5].

Potential limitations:

- Dependency on Internet Connectivity –

Some outdoor fields experienced intermittent Wi-Fi; future work will explore offline caching with Firestore’s local persistence.

- Badge Fatigue –

Over-gamification could diminish intrinsic motivation; a dynamic badge system (seasonal themes) is planned.

- Scalability Beyond Campus –

While Firebase scales well, multi-institution deployments may require hierarchical security modelling.

VII. LESSONS LEARNED & BEST PRACTICES

Lesson	Recommendation
Leverage Firebase security rules early	Iteratively test rules with the Firebase Emulator Suite to avoid privileged writes.
Modularize UI components	Storybook helped maintain design consistency and accelerated onboarding of new developers.
Instrument all user actions	Firebase Analytics combined with custom events enabled data-driven enhancements (e.g., adjusting badge thresholds).
Plan for offline use	Enable Firestore’s offline persistence and queue score updates to be synced when connectivity restores.
Engage stakeholders throughout	Regular demo sessions with the Sports Office ensured that the admin dashboard matched real needs.

VIII. CONCLUSION

This case study demonstrated that a real-time, gamified campus sports management system built on Firebase can transform the way universities organize, promote, and track athletic activities. By providing instantaneous score updates, automated scheduling, and motivational gamification elements, SportifyU achieved measurable improvements in student participation, administrative efficiency, and user satisfaction.

The results reinforce the thesis that modern web technologies coupled with gamified design principles constitute a powerful catalyst for digital transformation in higher-education extracurricular programs. Future work will explore cross-campus federated deployments, integration with wearable IoT devices for biometric analytics, and adaptive learning pathways that tie sports achievements to academic incentives.

REFERENCES

- [1] J. Lee and S. Kim, "Real-time sports score broadcasting using MQTT and Node-RED," in *Proc. IEEE Int. Conf. on Cloud Computing (CLOUD)*, 2022.
- [2] A. Verma and R. Gupta, "Performance analysis of Firebase for real-time web applications," *IEEE Access*, vol. 11, pp. 102345–102356, 2023.
- [3] M. Rahman, S. Ahmed, and T. Khan, "Gamification in higher education: A systematic review of impact on student engagement," *Computers & Education*, vol. 195, 2023.
- [4] Y. Zhang and L. Chen, "Serverless architecture for scalable real-time applications: A case study," *Journal of Cloud Computing*, vol. 13, no. 2, 2024.
- [5] S. Patel and N. Iyer, "Designing progressive web applications for educational platforms," *ACM Transactions on Computing Education*, vol. 24, no. 1, 2024.
- [6] P. Kumar, R. Mehta, and A. Singh, "Secure cloud-based application development using Firebase and modern IAM techniques," *IEEE Security & Privacy*, vol. 23, no. 1, 2025.
- [7] R. Singh and D. Sharma, *Digital Transformation in Campus Management Systems Using Real-Time Technologies*. Springer, Smart Education Series, 2025.