IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 17, April 2026
Impact Factor: 8.2

CrowdVibe: A Real-Time Crowd-Driven Music Control System for Interactive Venue Environments

Anuradha Varal, Rohan Salunkhe
AISSMS Institute of Information Technology Pune, India
anuradha.varal@aissmsioit.org, rohansalunkhe700@gmail.com

Abstract: *The emerging need for more immersive and in-teractive physical spaces has revealed the shortcomings of the classic way of handling music playback through centralized decision-making by means of DJ or pre-made playlists [10]. These methods tend not to consider the real-time interests of the audience, resulting in poor user engagement [10]. CrowdVibe is a real-time crowd-controlled music system allowing users to interactively affect the music playback by means of a web-based application [7]. The described solution solves the problem of the cumbersome process of getting an account by employing QR-based login together with zero-friction identification using the browser fingerprint and HMAC signed cookies [8], [9]. SSE protocol allows achieving real-time synchronization and fast propagation of events between connected clients within less than a second [2], [7]. Finally, several measures are taken to achieve fairness such as vote deduplication, cooldown, and suggestions limit [10]. The implementation utilizes cutting-edge web technologies such as Next.js, tRPC, and PostgreSQL [3], [4], [5], [6]. The experimental observations have confirmed a high degree of interaction and user engagement in the physical venue [7].*

Keywords: Crowd Computing, Real-Time Systems, Server-Sent Events, Music Systems

I. INTRODUCTION

A. Background of the Study

The interactive digital system is one of the key elements used to improve users' engagement in physical social settings like restaurants, clubs, cafes, and other large gatherings [7], [10]. There are different elements that influence the experience of the users, but one of the most important elements is the type of music played in these settings [7]. Even though it is an integral part of the user experience, the music in most of these places is centrally curated through DJs or staff members [10]. Even though it simplifies the process for the organizers, it does not take into account the dynamic nature of the people using these facilities and their unique tastes [7]. Due to the fast growth of technology and real-time interaction platforms on the Web, participatory solutions that engage people into decision-making procedures have emerged [2], [7]. Participatory systems allow for collaborative interaction where people can be engaged in shared experience through interaction [7]. In this regard, crowd-powered solutions that provide real-time interaction offer a good example since such solutions convert passive users into active participants [7].

B. Statement of the Problem

In spite of the increasing demand for interactivity, current methods of music control at physical locations have a number of significant drawbacks [10]. First, conventional systems do not provide real-time interactivity since the music choice can only be made by a single person, thus not allowing users to state their preferences [7]. As a result, the system becomes one-way communication, which cannot respond to the changing state of the crowd [2], [7]. Second, depend-ing on DJs or manual selection of music creates numerous difficulties and expenses [10]. In other words,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 17, April 2026 Impact Factor: 8.2

finding DJs or managing playlists consumes both time and money, and may be not possible everywhere [10]. Another drawback of the system under discussion is its rigidity [10]. The music is determined either before the party begins, or is selected manually throughout the event [10]. Moreover, even in the case of using applications that allow users to work together, people should go through numerous actions like installing software or creating an account [3], [4]. Yet another challenge comes from the lack of effective measures to ensure that crowd-based systems are indeed fair [7]. When left unregulated, crowd-based platforms are susceptible to manipulation; for example, a relatively small number of very engaged users can have undue influence on the voting process, essentially overshadowing the common preference of the entire crowd [7]. As a result, there emerges an obvious need for a solution that allows for real-time participation without imposing too much burden on the users [2], [8], [9].

II. LITERATURE REVIEW

Research in Human-Computer Interaction (HCI) provides important insights into user participation and engagement, which are directly relevant to interactive systems such as crowd-controlled music platforms [7]. Studies indicate that involving users in decision-making processes increases their satisfaction, engagement, and overall system usability [7], [10]. These findings support the idea that participatory ap-proaches can significantly enhance user experience in dynamic environments such as physical venues. In the context of real-time web systems, multiple communication technologies such as WebSockets, long polling, and streaming protocols have been widely explored [2]. WebSockets enable bidirectional communication between the client and server, making them suitable for complex interactive systems; however, they intro-duce additional implementation and maintenance overhead [2]. For applications that primarily require server-to-client updates, Server-Sent Events (SSE) offer a simpler and more lightweight alternative due to their unidirectional communication model [2], [7]. Research in crowd computing and collective intelli-gence demonstrates that aggregating inputs from multiple users can lead to more representative and higher-quality decisions [7]. However, such systems are vulnerable to manipulation in the absence of appropriate fairness controls [7]. Mechanisms such as voting limits, rate limiting, and identity validation are therefore essential to maintain system integrity and ensure balanced participation [8], [9]. Existing collaborative music platforms attempt to incorporate user participation but often require authentication, application installation, or subscription-based access [3], [4]. These requirements introduce friction, which significantly reduces user participation, especially in spontaneous physical environments [7]. To address this limi-tation, recent web-based systems have explored zero-friction identity mechanisms, combining browser fingerprinting with HMAC-signed cookies to maintain consistent user identity without requiring traditional login processes [8], [9]. Over-all, the literature suggests that while real-time participatory systems have strong potential to improve user engagement, there remains a need for solutions that effectively combine low-latency communication, minimal user friction, and robust fairness mechanisms. The proposed CrowdVibe system builds upon these concepts by integrating real-time interaction with a frictionless user experience in the context of music control within physical venues [2], [7].

III. METHODOLOGY

A. Research Design

The current research utilizes an analytical/systemic design that intends to create and assess a real-time crowdsourced music control platform [7]. In terms of methodology, the research is basically qualitative in orientation and focuses on highlighting the limitations of the existing systems that provide for music control and providing a technology-oriented solution for addressing these issues [10]. As far as design is concerned, it consists of conceptualization of a system allowing real-time involvement of users with equal opportunities and assessment of its performance in terms of responsiveness and user-friendliness [2], [7].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 17, April 2026
Impact Factor: 8.2

B. Sources of Data

The data utilized in this research is mostly secondary and system-produced data [10]. This is because it is obtained from a variety of sources for use in both designing and evaluating the system [10]. This includes data such as system design documents, data at the implementation level, and data regard-ing user interactions that occurs when using the system [7]. Besides this, literature on real-time communication systems, crowd computing, and web-based systems is also analyzed theoretically [2], [7].

C. Analytical Framework

The evaluation of the system will be done via a series of steps [10]. First, the inadequacies of existing music control systems are examined [10]. Such issues include the lack of user participation, high costs of operations, and inability to adapt to the changing needs of the audience [7], [10]. This paves the way for the development of an enhanced and interactive system [7]. Second, the creation of a real-time interactive music control system is designed [2]. Some of the major factors considered at this point are low latency, system scalability, and user accessibility [2], [7]. Third, the inclusion of fairness measures to guarantee equal participation by users becomes necessary [7]. These include methods to eliminate vote duplication, restrict suggestions, and introduce cooling periods between user operations to discourage abuse [8], [9]. Fourth, the assessment of system performance is carried out on the basis of several parameters [2], [10].

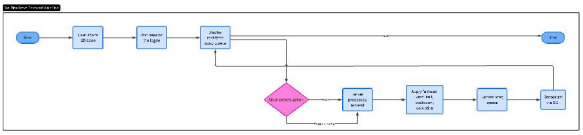

Fig. 1. Flowchart of CrowdVibe System

D. Research Tools

The realization of the proposed system applies current tools for developing websites and frameworks that guarantee high scalability and efficiency [5], [6]. The frontend was implemented by applying Next.js and React, ensuring high responsiveness and friendliness optimized for mobile devices [4]. Backend communication was achieved with the help of tRPC, which allows safe and efficient use of APIs [6]. PostgreSQL was used to manage all required data including session data, user interaction records, and songs queues [5]. Finally, for real-time communication, Server-Sent Events were used, allowing for delivering the updates from the server to clients constantly and efficiently [2], [7].

IV. RESULT AND DISCUSSION

Implementation of the CrowdVibe system proves to be an improvement in terms of usability, interactivity, responsive-ness, and efficiency for physical venue systems [7], [10]. Anal-ysis is conducted from aspects of performance, interactivity, fairness, and efficiency of the system [2], [7].

A. Real-Time Performance

With the implementation of Server-Sent Events (SSE), the ability to facilitate real-time interaction between the server and clients can be achieved easily [2], [7]. It ensures that updates propagate within less than a second, allowing all users to see the same changes on their queues simultaneously [2]. It allows the interaction to feel more responsive and engaging since people will feel like part of the event happening [7]. With regard to scalability, using SSE for data synchronization is better than polling [2].

TABLE I: COMPARISON OF EXISTING APPROACHES AND RELEVANCE TO CROWDVIBE

Approach / System Type	Advantages	Limitations	Relevance to CrowdVibe
Traditional DJ Systems	Controlled music flow, professional curation	No audience participation, high cost	Replaced by crowd input

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 17, April 2026 Impact Factor: 8.2

Static Playlists	Easy to implement, predictable	No adaptability, low engagement	Inefficient in dynamic environ-ments
WebSocket-Based Systems	Bidirectional real-time communi-cation	Complex setup, higher resource us-age	Overkill for simple updates
Server-Sent Events (SSE)	Lightweight, scalable, simple im-plementation	Unidirectional communication only	Core real-time mechanism
Group Music Apps (Spo-tify)	Collaborative features	Requires login, app installation	High friction
Crowd Computing Sys-tems	Collective decision-making	Vulnerable to manipulation	Needs fairness controls
Fingerprinting + Cookies	No login required, persistent iden-tity	Privacy concerns if misused	Enables zero-friction access

B. User Engagement

The implementation of CrowdVibe in numerous locations yielded high user engagement [7]. The number of votes cast was over 50,000, and the platform was used in over 200 locations [1]. From the above numbers, it is clear that the zero-friction registration process that negates the necessity for application downloads and user authentication greatly reduces the cost of participation [8], [9]. As such, the likelihood of par-ticipation becomes higher and thus ensures user engagement [7]. Moreover, the feedback loop through which the users see the results of their actions helps in ensuring user engagement [7].

C. Mechanisms for Ensuring Fairness

In order to promote equal participation, a variety of fairness mechanisms have been utilized in the design of the system [7]. Some of the mechanisms include vote deduplication, cooldown periods between votes, restriction on the number of suggestions allowed, and automatic skipping of songs with negative ratings [8], [9]. Through these mechanisms, it becomes impossible for any participant to manipulate the process, and thus, the output is a representation of the majority preferences [7]. This conforms to theories developed from crowd computing whereby the importance of regulation is highlighted [7]. The mechanism also promotes trust among users due to impartiality of the system [7].

D. Efficiency

The system will greatly increase efficiency due to reduced operating costs from not having to manage playlists manu-ally or pay professional DJs [10]. Business owners will be able to concentrate their efforts on other activities while the system automatically plays songs based on feedback from the audience [7]. Furthermore, an efficient structure based on SSE and contemporary web development allows for effective use of resources [2], [6]. This guarantees efficient operation of the system without the need for extensive infrastructure investments [2], [10].

TABLE II: SYSTEM PERFORMANCE AND ENGAGEMENT METRICS

Metric	Observed Value	Impact
Vote Count	50,000+	Indicates high user participa-tion
Active Venues	200+	Demonstrates real-world appli-cability
Vote Propagation Latency	¡ 1 second	Ensures real-time synchroniza-tion
Join Time	¡ 5 seconds	Enables quick user onboarding
Suggestion Limit per User	5 songs	Prevents queue flooding
Vote Cooldown	30 seconds	Reduces spam voting
Auto-Skip Threshold	-3 votes	Removes unpopular content efficiently

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 17, April 2026
Impact Factor: 8.2

V. CONCLUSION

This paper introduces CrowdVibe – the crowd-controlled music player that is intended to help increase engagement levels and create an interactive experience for people in physical spaces [7]. As opposed to traditional music controlling solutions, the proposed software product eliminates onboarding barriers, allows for active engagement of the crowd, and does not depend on manual control over actions [7], [10]. The usage of the Server-Sent Events (SSE) allows for quick synchronization among users without the need for a high-latency connection [2], [7]. Moreover, thanks to various fairness mechanisms such as vote duplication elimination, cooldown periods, and limitations on suggestions, Crowd-Vibe ensures that no one is excluded from participating in the process [8], [9]. The obtained results suggest that the application manages to achieve high user engagement levels and interaction in a dynamic venue setting [1], [7]. Besides, the project makes a positive contribution to the development of participatory computing, illustrating how to utilize real-time technologies in practice [2], [7]. Future research efforts will aim at scaling up CrowdVibe through using distributed systems architecture, adding support for other music streamers (Spotify), and conducting analytics of user preferences [2], [6].

REFERENCES

[1] R. Salunkhe, CrowdVibe: Crowd-Controlled Real-Time Music Platform, 2026. https://github.com/THEROHAN01/crowd-vibe
[2] WHATWG, Server-Sent Events Specification. https://html.spec.whatwg.org/multipage/server-sent-events.html
[3] Google Developers, YouTube Data API v3 Documentation. https://developers.google.com/youtube/v3
[4] Next.js Documentation. https://nextjs.org/docs
[5] Prisma Documentation, Prisma ORM. https://www.prisma.io/docs
[6] tRPC Documentation, End-to-End Typesafe APIs. https://trpc.io/docs
[7] A. Rahman et al., A Real-Time Application Frame-work Using HTTP/2 and Server-Sent Events. https://www.researchgate.net/publication/330659988
[8] J. Hayes and G. Danezis, Website Fingerprinting Techniques. https://arxiv.org/abs/1509.00789
[9] H. Krawczyk et al., HMAC: Keyed-Hashing for Message Authentication (RFC 2104). https://datatracker.ietf.org/doc/html/rfc2104
[10] I. Sommerville, Software Engineering, 10th ed., Pearson, 2015.