



# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.699**

**Volume 15, Issue 3, March 2026**

# ML-Powered Collaborative Event Management

**K. Sukeerthi<sup>1</sup>, E. Ganesh<sup>2</sup>, M. Ruthvick<sup>3</sup>, V. Archana<sup>4</sup>, M. Hyder Ali<sup>5</sup>, L. Sharath<sup>6</sup>**

Assistant Professor, Department of CSE (Data Science), ACE Engineering College, Hyderabad, India<sup>1</sup>

IV B.Tech, Department of CSE (Data Science), ACE Engineering College, Hyderabad, India<sup>2,3,4,5,6</sup>

**ABSTRACT:** Managing and participating in college-level virtual events, such as hackathons, often involves multiple complex steps including event creation, team formation, communication, submission tracking, and feedback collection [1]. This project aims to develop a web-based platform that simplifies and automates these processes for both organizers and participants. The platform allows organizers to create events with rules, deadlines, and team structures, while participants can register individually or in groups. An ML-powered team matching algorithm automatically forms balanced teams based on participants skills and interests. Each team has access to a private chat with file sharing for seamless collaboration, and the platform manages submission tracking, evaluation, scoring, and post-event networking. By integrating ML-based team formation, real-time collaboration, and structured event management, It provides a user-friendly ecosystem that reduces administrative effort, ensures fair evaluation, and enhances the overall experience for virtual college events.

**KEYWORDS:** Machine Learning, Event Management System, Team formation Algorithm, Virtual Hackathon, Real time collaboration.

## I. INTRODUCTION

Managing hackathons and college technical events involves registrations, team formation, communication, submissions, and evaluation. However, most existing systems offer only basic registration and scheduling, with no automation or collaboration features. Organizers must manually create teams, track deadlines, and manage submissions, while participants rely on external apps like WhatsApp and Google Drive for communication and file sharing. This leads to confusion, delays, and increased workload. There is a need for an intelligent, unified platform that can automate event workflows, form balanced teams using ML, support real-time collaboration, manage submissions with deadlines, ensure fair evaluation. Therefore, there is a need for an intelligent platform

## II. EXISTING SYSTEM

Existing event management systems primarily handle basic tasks such as participant registration, event scheduling, and result publication. They allow organizers to manage events and participants but provide limited support for real-time collaboration. Communication between organizers and participants is generally handled through emails or static message boards. Most systems offer only basic administrative support without any intelligent automation or personalized recommendations.

### A. Drawbacks of Existing System

- 1.Organizers manually handle event setup and team registrations
- 2.Participants join events using simple online forms or spreadsheets
- 3.Teams are formed manually or through basic rule-based methods.
- 4.Communication and file sharing happen through external platforms (e.g., WhatsApp, Google Drive).

## III. PROPOSED MODEL

The proposed system is a web-based platform for managing college hackathons and technical events. It uses machine learning for intelligent team formation, supports real-time communication, automates submissions with deadlines. Post-event feedback and networking enhances engagement while reducing organizer workload.

### A. Advantages of the Proposed System

- 1.ML-based balanced team formation.

- 2. Real-time collaboration with chat and file sharing.
- 3. Automated submissions with deadline enforcement.

#### IV. METHODOLOGY

##### A. Event Creation & Participant Registration:

The system must allow organizers to create events with: event name, description, rules, deadlines, and team size limits. Participants should be able to register either individually or as a pre-formed team. The system must support role-based access (Organizer / Participant).

##### B. ML-Based Team Formation:

The system must analyze participant data (skills, interests, experience) using Scikit-Learn models. When individuals register without a team, the system should automatically group them into balanced teams. The algorithm must ensure fairness by distributing skills evenly across teams.

##### C. Real-Time Communication (Chat):

Each team must have access to a private chat room. The system must support real-time messaging using Web Sockets or Socket.IO. Files such as code, documents, or PDFs must be up loadable within the chat.

##### D. Submission Handling:

Participants must be able to upload project files (documents, code, reports). The system must enforce submission deadlines and prevent uploads after the deadline.

##### E. Evaluation & Scoring:

Organizers should be able to review uploaded submissions. The system must allow evaluators to assign scores or comments manually.

#### V. SYSTEM ARCHITECTURE

The system architecture follows a modular, layered design where each component handles a specific stage of the event workflow. User actions—such as registration, event creation, team communication, or file submission—are received through the web interface and passed to the backend via REST APIs. After input is received, the data processing layer validates user details and prepares participant skill data for the ML Team Formation module, which uses Scikit-Learn to generate balanced teams. All event information, team assignments, chat messages, and submission metadata are stored in the MySQL database. Real-time collaboration is handled by the Communication Module using Web Sockets/Socket.IO, enabling private team chat and file sharing. The Submission Module manages project uploads, enforces deadlines, and maintains version history. Organizers access an Evaluation Module to review submissions and assign scores. The output—team details, chat updates, event dashboards, and results—is rendered back to users through the frontend. Each module operates independently.

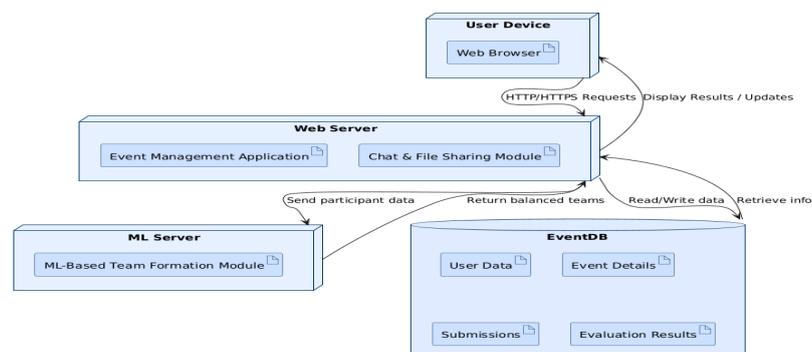
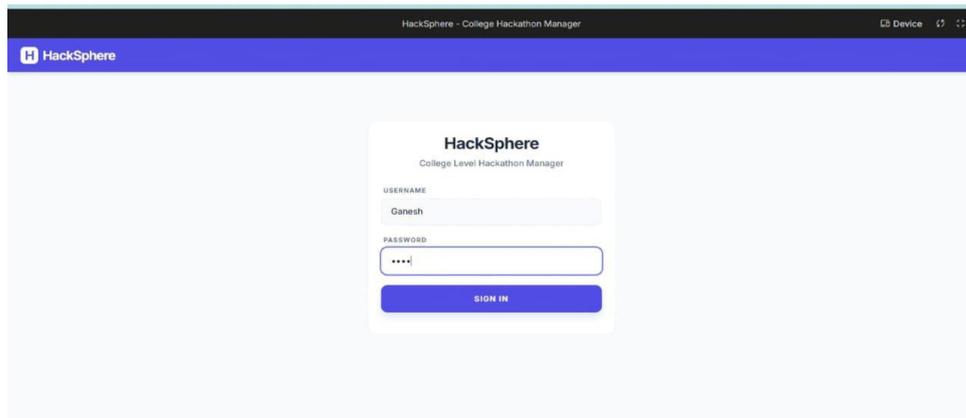


Fig 1: System Architecture

## VI. RESULTS AND OUTPUT

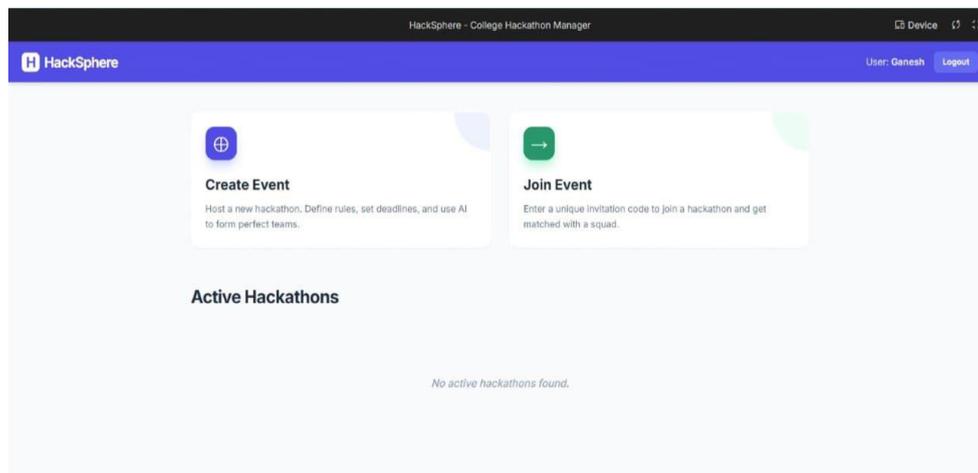
### Step 1: User Registration

The Figure-2, system allows new users to create an account through the registration interface by entering details such as name, email, mobile number, gender, and password. Once the information is submitted, the system securely stores the user details in the database and creates a new account for accessing the application.



**Fig 2: Register Page**

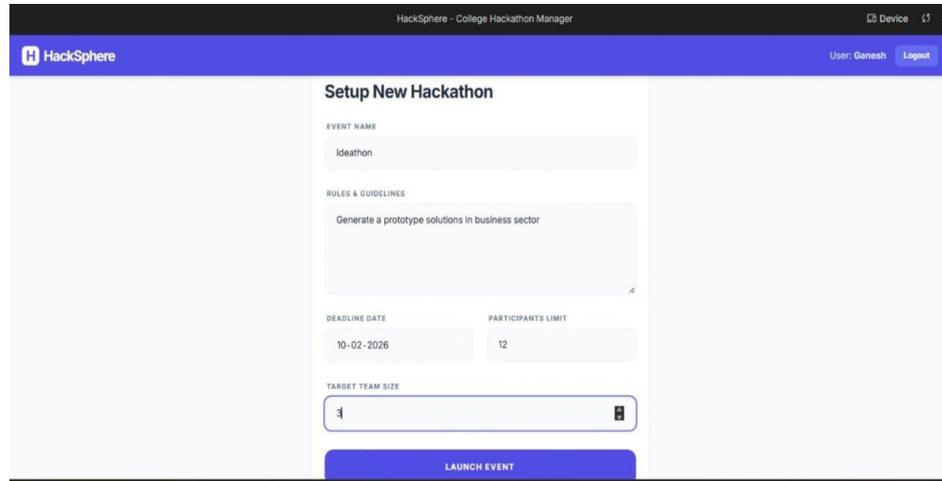
### Step 2: Home Page



**Fig 3: Home Page**

### Step 3: Event creation

In Figure-4, organizer can create an hackathon by defining rules and deadlines ,participants and team size.

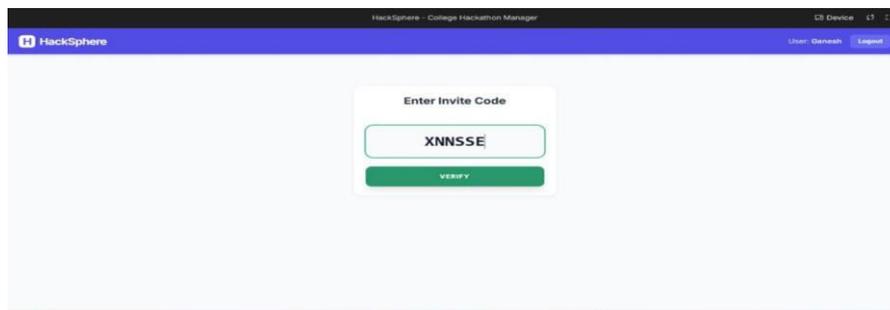


The screenshot shows the 'Setup New Hackathon' interface. It includes a header with the HackSphere logo and user information (User: Ganesh, Logout). The main form has the following fields: 'EVENT NAME' (Ideathon), 'RULES & GUIDELINES' (Generate a prototype solutions in business sector), 'DEADLINE DATE' (10-02-2026), 'PARTICIPANTS LIMIT' (12), and 'TARGET TEAM SIZE' (4). A 'LAUNCH EVENT' button is at the bottom.

**Fig 4: Event creation**

**Step 4: Entering event**

In Figure-4, Participants can enter into the Hackathon by entering the code of the hackathon a unique code is generated when an organizer created an event.

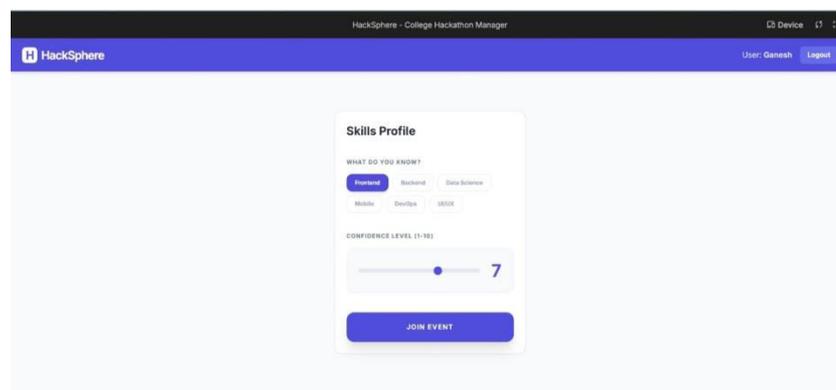


The screenshot shows the 'Enter Invite Code' interface. It features a text input field containing 'XNNSSE' and a green 'VERIFY' button below it. The header includes the HackSphere logo and user information (User: Ganesh, Logout).

**Fig 5: Entering event**

**Step 5: skill set:**

Participants. Have to enter the skills they are good at by giving rating to themselves



The screenshot shows the 'Skills Profile' interface. It has a header with the HackSphere logo and user information (User: Ganesh, Logout). The main form asks 'WHAT DO YOU KNOW?' and lists skills: Frontend, Backend, Data Science, Mobile, DevOps, and WEBX. Below this is a 'CONFIDENCE LEVEL (1-10)' slider set to 7. A 'JOIN EVENT' button is at the bottom.

**Fig 6: Skill set**

### Step 6: Event Page:

After entering the event using unique code this is how event page look like.

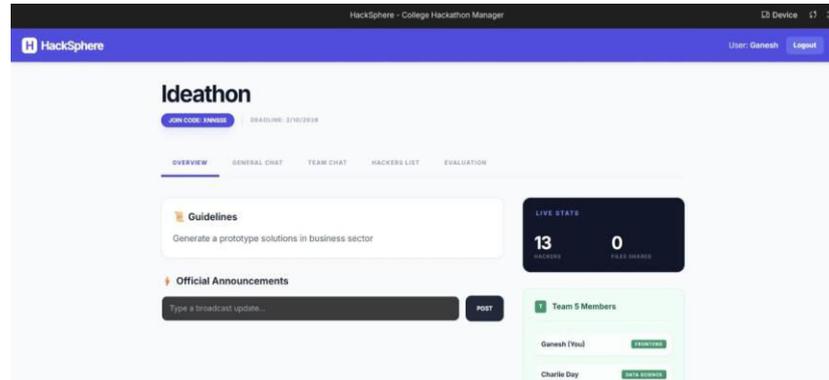


Fig 7: Event Page

## VII. CONCLUSION AND FUTURE SCOPE

The ML-Powered Collaborative Event Management Platform is a necessary solution to the inefficiencies and difficulties associated with managing hackathons and college technical events through traditional methods. Existing systems often provide only basic features like registration and scheduling, lacking the crucial automation and real-time collaboration tools needed for complex events. This forces organizers to manually manage tasks such as team formation, tracking deadlines, and managing submissions, while participants are left to rely on external, scattered applications like WhatsApp and Google Drive for essential communication and file sharing, leading to delays and confusion. The core motivation for developing this new platform is the growing reliance on virtual hackathons, digital submissions, and collaborative tools, which demands a more unified and intelligent system.

This proposed web-based platform introduces an **intelligent, unified approach** by leveraging Machine Learning for its central function: **ML-Based Team Formation**. Using algorithms like K-Means and KNN from Scikit-Learn, the system analyzes participant skills, interests, and experience to automatically group individuals into balanced and fair teams, ensuring skill diversity and strength distribution. Beyond intelligence, the platform significantly enhances collaboration by providing a **Real-Time Communication Module** using Web sockets or Socket.IO, which includes team-specific chat rooms and secure file sharing. Furthermore, the system automates the event lifecycle by enforcing submission deadlines and managing version control.

## REFERENCES

- [1] A. A. Fowler, "Virtual Hackathons as a Collaborative Innovation Model: Design, Execution, and Outcomes," IEEE Transactions on Engineering Management, vol. 68, no. 6, pp. 1864–1876, Dec. 2021. doi: 10.1109/TEM.2020.2989480
- [2] J. V. Nickerson and C. de la Torre, "Remote Collaboration in Virtual Hackathons: Tools, Challenges, and Team Dynamics," Proc. ACM Conference on Computer Supported Cooperative Work (CSCW), 2020, pp. 1–12.
- [3] A. R. Mishra and S. Gupta, "A Web-Based Event Management and Team Coordination System for Technical Fests," International Journal of Computer Applications, vol. 177, no. 41, pp. 25–30, Feb. 2020.
- [4] M. Hosseini et al., "Hackathons as Experiential Learning Platforms: A Systematic Review," Proc. IEEE Frontiers in Education Conference, 2019. doi: 10.1109/FIE43999.2019.9028622
- [5] R. Kumar and K. Srinivasan, "Intelligent Team Allocation Using K-Means Clustering for Academic Project Groups," International Journal of Advanced Computer Science and Applications, vol. 10, no. 5, pp. 170–177, 2019.
- [6] H. Maxwell and D. R. Foss, "Evaluating Machine Learning Approaches for Instructor-Assisted Team Formation," Journal of Educational Data Mining, vol. 12, no. 1, pp. 1–22, 2020.



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details