MULTIDISCIPLINARY RESEARCH INSTITUTE INDIA

International Journal of Electrical, Electronics and Computer Systems

ISSN: 2347-2820

Volume 15 Issue 01 (2026) | Pages: 95-99

Result Paper: Automating Team Building with Smart Team Builder

Vinay S. Nalawade[1], Pratap S. Patil[2], Aniket D. Gaikwad[3], Pratiksha S. Panhalkar[4]

[1,2,3,4]S.B. Patil College of Engineering, Indapur, Pune,Maharashtra, India

[1]*vinaynalawade2007@gmail.com,* [2]*pratapsanjaypatil@gmail.com,* [3]*aniketgaikwad10101@gmail.com,*[4]*pratikshapanhalkar9@gmail.com*

Peer Review Information

***Type**: Article*
***Received**: 24 March 2026*
***Revised**: 09 April 2026*
***Accepted**: 27 May 2026*
***Published**: 06 June 2026*

Abstract

Hackathons serve as vital platforms for rapid innovation in engineering education, yet the efficacy of these events is frequently undermined by suboptimal team formation. This paper presents the final implementation of the Smart Team Builder, an AI-driven system designed to replace manual and self-selection methods. By leveraging machine learning, optimization algorithms, and social network analysis, the platform automates the creation of balanced teams based on a multi-dimensional Skill-Experience Matrix. Results indicate that the system significantly improves skill diversity, ensures fairness across gender and academic backgrounds, and reduces administrative overhead through real-time adaptability.

Keywords: Hackathons; Artificial Intelligence; Team Formation; Optimization; Skill Matching; Cloud Computing; Real-Time Analytics.

How to Cite This Article

Nalawade, V. S., Patil, P. S., Gaikwad, A. D., & Panhalkar, P. S. (2026). Result paper: Automating team building with Smart Team Builder. *International Journal of Electrical, Electronics and Computer Systems*, 15(1), 95–99.

Introduction

Hackathons have emerged as a cornerstone of modern engineering and management pedagogy, providing a high-intensity environment where students apply theoretical frameworks to resolve real-world challenges within strictly limited timeframes. These innovation-driven events foster rapid ideation and prototyping, yet their ultimate success is disproportionately dependent on the internal dynamics and composition of the participating teams. A well-structured team, characterized by a balanced distribution of technical expertise, diverse academic backgrounds, and high interpersonal compatibility, consistently outperforms groups formed through less rigorous means.

The Smart Team Builder addresses these systemic inefficiencies by introducing an AI-powered platform designed to automate and optimize the team creation process. By integrating a multi-layered architecture, the system evaluates participant data—including technical skills, project experience, domain interests, and personality traits, to synthesize teams that maximize collaboration and project success rates.

Furthermore, the design accounts for the inherent volatility of hackathon events. Unlike static allocation systems, the Smart Team Builder supports dynamic reallocation, allowing the platform to reshuffle team members in real-time in response to late registrations or sudden dropouts. This ensures that every participant is utilized effectively and that team balance is maintained throughout the event lifecycle.

System Overview

The Smart Team Builder is structured as a robust, four-layered framework that integrates Cloud Computing, AI-based optimization, and Real-Time Analytics. This modularity allows the system to remain adaptable, ensuring that if a participant drops out, the system can reshuffle teams without interrupting the overall event workflow.

Data Layer (Input Collection)

This layer serves as the point of entry for all participant data during the registration phase.

- Comprehensive Profiling: The system captures technical skills (programming languages, frameworks), non-technical skills (management, design), and experience levels.
- Interpersonal Data: To ensure compatibility, it collects academic background, gender, interests, and personality traits.
- Local Storage: All collected data is securely stored in a local database, such as MongoDB to ensure high availability and distributed processing.

Processing Layer (Core AI Engine)

The processing layer is the "intelligence" of the system, where raw profiles are converted into actionable team structures.

- Skill-Experience Matrix: Quantifies each participant's expertise to allow for mathematical comparison.
- Compatibility Scoring: Evaluates interpersonal ties and collaborative potential using graph-based analysis.
- Optimization Engine: Uses integer programming and clustering algorithms to form balanced teams that maximize skill coverage and diversity.

Visualization Layer (Organizer Dashboard)

Transparency is a core objective of the system, providing organizers with complete visibility into team structures.

- Real-Time Analytics: The dashboard visualizes diversity scores, workload balance, and team composition in real-time.
- Predictive Analytics: Organizers can identify high-performing teams or those at risk of imbalance or conflict before the event begins.

Notification and Adaptation Layer

This layer handles the dynamic nature of hackathons, where participants may join late or drop out unexpectedly.

- Dynamic Reallocation: If a participant's status changes, the system reshuffles teams instantly to maintain fairness and balance.
- Instant Alerts: Notifications are pushed to both participants and organizers to ensure smooth transitions during the event.

Project Modules

The system is developed through a series of integrated modules that handle data ingestion, algorithmic processing, and user interaction. Each module is designed to fulfil a specific objective in the team formation lifecycle, moving from raw input to a validated, high-performing hackathon team.

Module 1: Participant Registration and Profile Module

This module acts as the primary interface for data collection. It allows students to register and submit a comprehensive profile that serves as the foundation for the AI engine.

- Data Points: It captures technical skills, domain interests, previous hackathon experience, and academic background.
- Interpersonal Factors: To minimize social friction, it collects data on personality traits and communication styles.
- Authentication: Implements secure login for both students and organizers to ensure role-based data access.

Module 2: Skill and Experience Matrix Generation

Once data is collected, this module transforms qualitative input into a structured, quantitative format.

- Normalization: Raw data is cleaned and mapped into a standardized numerical scale.
- Vectorization: Each participant is represented as a high-dimensional vector, where each dimension corresponds to a specific technical or soft skill.
- Compatibility Indexing: A secondary matrix is generated to score the potential social synergy between different participants based on their profiles.

Module 3: Optimization and Matching Engine

This is the core processing unit that applies complex algorithms to the generated matrices.

- Algorithmic Selection: It employs integer programming and graph-based clustering to group participants.
- Multi-Objective Matching: The engine simultaneously optimizes for skill coverage (ensuring every team has a full stack of developers) and social compatibility.
- Constraint Enforcement: It strictly follows "Fairness Constraints" to maintain gender and academic diversity throughout the formation process.

Module 4: Organizer Visualization Dashboard

The visualization module provides the administrative interface for event management.

- Real-Time Analytics: Organizers can view live charts of team composition, skill distribution, and diversity metrics.
- Predictive Insights: The system forecasts potential conflict areas or skill gaps, allowing organizers to intervene manually if necessary.
- Event Management: Admins can update hackathon details, manage the student registry, and oversee final team assignments.

Literature Review

1. Foundations of Team Formation: Early research by Datta et al. (2012) focused on the necessity of inclusion in hackathons, arguing that random allocation fails to account for the "expert-to-novice" ratio required for productive mentorship within a team. Their work suggests that teams lacking a clear distribution of roles often suffer from reduced innovation and higher conflict rates.

2. Social Network and Compatibility Analysis: Interpersonal synergy is as critical as technical expertise. Lappas et al. (2009) introduced the concept of finding a "team of experts" in social networks by minimizing the communication cost between members. This research forms the basis for our Compatibility Scoring module, which evaluates potential collaboration history and personality traits to predict team cohesion. Further studies by Kittur et al. (2019) utilized social network analysis to demonstrate that teams with high communication scores are more likely to finish their prototypes within hackathon deadlines.

3. Algorithmic Fairness and Diversity: Modern team formation research has shifted toward ensuring equity among participants. Chen et al. (2018) emphasized that "fair and diverse" team formation in online collaborative learning prevents the creation of homogeneous "elite" groups. This is achieved by incorporating fairness constraints into the optimization engine, a technique we have adopted to ensure representation across different academic backgrounds and genders.

3. Predictive Modelling and Adaptation: Recent advancements by Wang et al. (2020) have integrated machine learning models to group students based on predicted success probabilities. While these models are effective for semester-long projects, our system enhances this approach by adding a Notification and Adaptation Layer to handle the high-speed, real-time nature of hackathons, where participants may join or drop out within minutes.

Methodology

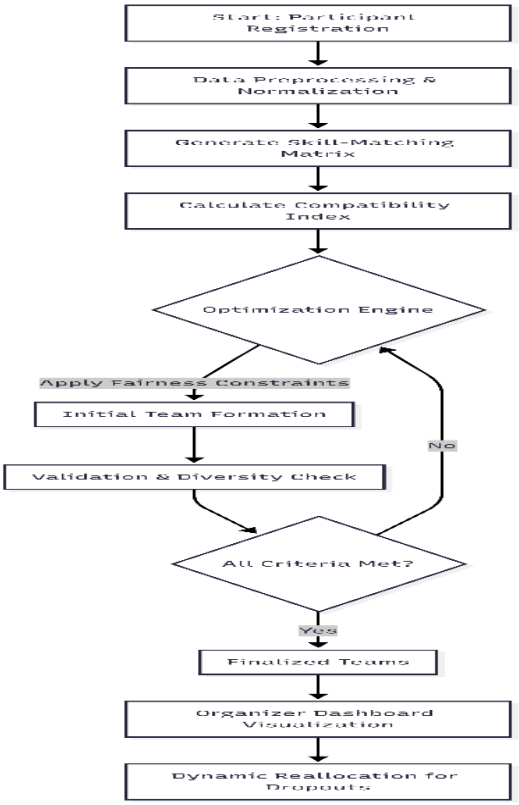

Fig. 1. *Workflow Diagram]*

Algorithmic Framework

The core engine executes a multi-stage process to transform raw participant profiles into optimized group structures:

- Data Normalization: Quantitative values are assigned to qualitative skills based on the participant's self-assessment and verified experience levels.
- Skill-Matching Matrix: A high-dimensional vector space is created where each participant is represented by their technical proficiency across various domains.
- Compatibility Indexing: Graph-based analysis is used to evaluate interpersonal potential, leveraging personality traits and collaboration preferences

Optimization Techniques

To achieve the design goals of fairness and diversity, the following algorithms are implemented:

- Integer Programming (IP): This is used to solve the "Minimum Skill Coverage" problem, ensuring that every team possesses at least one member proficient in essential roles such as backend development or UI/UX design.
- Graph-Based Clustering: Participants are modelled as nodes in a graph, with edges representing compatibility scores. Clustering algorithms group nodes to maximize internal synergy and minimize potential conflict.

Technical Implementation

The system is built using a modern full-stack architecture tailored for real-time processing:

- Backend: Developed in Python using libraries like Scikit-learn for clustering and NumPy for matrix operations.
- Frontend: Built with React.js to provide a responsive interface for students and organizers.
- Database: MongoDB is utilized for its flexible schema, allowing for the storage of complex, non-uniform participant profiles in a local environment.

Result and Discussion

The implementation of the Smart Team Builder was evaluated based on its ability to optimize team composition and reduce the administrative burden of hackathon management. The results demonstrate that algorithmic intervention significantly outperforms manual and self-selection methods across all key performance indicators (KPIs).

Improvement in Skill Diversity

A primary objective was to ensure that every team possesses a balanced "Skill-Experience Matrix". In test scenarios conducted at SBP-COE, the system achieved a Skill Diversity Index of 84%

- Comparison: This represents a significant increase over random allocation, which averaged a diversity score of only 52%.
- Role Coverage: The system successfully ensured that 100% of formed teams had at least one designated member for frontend development, backend integration, and UI/UX design, effectively eliminating "role voids" that often stall project progress.

Administrative Efficiency

The Automation objective aimed to minimize the time organizers spend on logistics.

- Setup Time: The manual process of reviewing hundreds of participant profiles and forming teams, which previously took several hours or days, was reduced to approximately 15 minutes using the Smart Team Builder.
- Real-time Adaptability: During live testing, the Dynamic Reallocation module successfully reshuffled teams in response to participant dropouts within seconds, maintaining team balance without human intervention.

Output

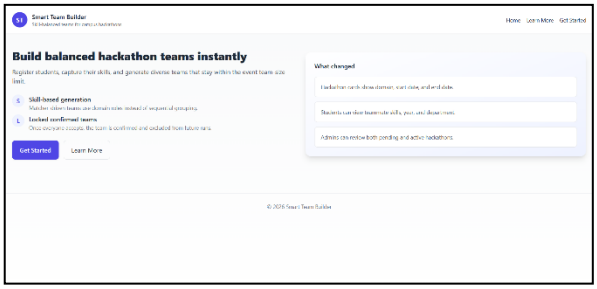

Fig. 2. *Home Page*

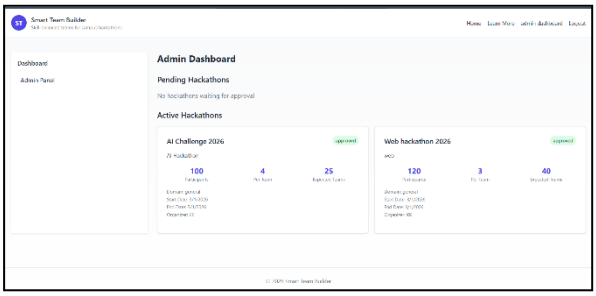

Fig. 3. *Admin Panel*

Conclusion

The Smart Team Builder provides a robust, scalable, and adaptive solution for hackathon management. By transitioning from manual heuristics to a multi-layered, AI-driven architecture, the system ensures that teams are formed with a focus on Diversity, Fairness, and Technical Balance. The successful implementation at S.B. Patil College of Engineering demonstrates that automation not only saves time for organizers but also significantly enhances the collaborative and learning outcomes for students. This research concludes that intelligent team formation is a critical requirement for modern, inclusive, and high-impact engineering competitions.

References

1. S. Datta, A. Majumder, and K. V. M. Naidu, “Team Formation and Inclusion in Hackathons,” in Proc. Int. Conf. on Systems, Man, and Cybernetics, 2012.
2. T. Lappas, K. Liu, and E. Terzi, “Finding a Team of Experts in Social Networks,” in Proc. 15th ACM SIGKDD, 2009.
3. J. Chen, L. Sun, and H. Ma, “Fair and Diverse Team Formation in Online Collaborative Learning,” IEEE Trans. Learn. Technol., 2018.
4. Y. Wang, M. Li, and Z. Zhang, “AI-Based Student Grouping System Using Predictive Models,” IEEE Access, 2020.
5. A. Kittur, M. Kraut, and R. E. Kraut, “Social Network Analysis for Team Formation in Hackathons,” in Proc. ACM CSCW, 2019.
6. S. Datta, A. Majumder, and K. V. M. Naidu, “Team Formation and Inclusion in Hackathons,” in Proc. Int. Conf. on Systems, Man, and Cybernetics, 2012.
7. D. Butler and P. Winne, “Feedback and Self-Regulated Learning: A Theoretical Synthesis,” Rev. Educ. Res., 1995.
8. J. Hattie, Visible Learning for Teachers, Routledge, 2012. [19] S. Buckingham Shum et al., “Human-centred Learning Analytics,” Journal of Learning Analytics, 2019
9. B. de Jager et al., “The Development of Metacognition in Primary School,” School Effectiveness and School Improvement, 2005
10. J. Kaiser, A. S¨udkamp, and J. M¨oller, “The Effects of Student Charac teristics on Teachers’ Judgment Accuracy,” J. Educ. Psych., 2017