



# HACKATHON BUDDY

## An AI-Based Collaborative Platform for Smart Hackathon Team Formation and Project Management

Mrs. Sushama Sachin Takmare<sup>1</sup>, Arya Gaikwad<sup>2</sup>, Abhishek Patil<sup>2</sup>, Komal Walke<sup>2</sup>, Nilesh Bhatt<sup>2</sup>

Guide, Department of Data Science, D Y Patil College of Engineering and Technology, Kolhapur, India<sup>1</sup>

Department of Data Science, D Y Patil College of Engineering and Technology, Kolhapur, India<sup>2</sup>

**ABSTRACT:** Hackathons and innovation competitions create opportunities for rapid problem solving, interdisciplinary collaboration, and prototype development within limited time. However, one of the major challenges in such events is forming effective teams with compatible skills, interests, and roles. Existing hackathon platforms mainly support registration and project submission, but they provide limited assistance in strategic team assembly, often resulting in skill mismatches and inefficient collaboration. To address this issue, this paper proposes Hackathon Buddy, a cross-platform team formation and collaboration system that connects participants through intelligent matching logic. The platform considers factors such as technical skills, domain preferences, expertise level, and preferred roles to recommend suitable teammates and project opportunities. Users can create detailed profiles highlighting both hard and soft skills, choose their innovation areas, and either initiate new project ideas or join existing ones. By enabling smarter team formation and easier collaboration, the proposed system aims to improve participant engagement, strengthen team compatibility, and maximize the overall effectiveness of hackathon outcomes.

### I. INTRODUCTION

Hackathons are time-bound, collaborative events where participants usually students, developers, and domain experts—work in teams to design, prototype, and present technological solutions within a short duration, typically 24–72 hours. These events focus on solving specific problem statements, often related to software, apps, or real-world challenges, and are hosted by organizations, universities, and companies to encourage rapid innovation and skill development. Beyond coding, hackathons emphasize teamwork, creativity, and quick decision-making, making them important platforms for experiential learning and professional exposure. A major issue in such events is the reliance on informal networking and manual team selection. Participants often spend valuable time searching for teammates with complementary skills, similar interests, and suitable roles. This process frequently leads to mismatched teams, incomplete skill coverage, and reduced project effectiveness. The problem becomes Identify applicable funding agency here. If none, delete this. even more serious in large-scale or virtual hackathons, where participants may not have the opportunity to interact directly before team formation. Existing hackathon management platforms mainly focus on registration, project submission, and presentation workflows. Although these systems support event administration, they provide only limited assistance in strategic team assembly. Features such as skill-based matching, role synchronization, and collaborative project discovery are often missing, which makes the team-building process less efficient and less inclusive. To address these limitations, this paper proposes Hackathon Buddy, a cross-platform solution for intelligent team formation and project collaboration. The system uses matching logic based on participant skills, domain expertise, proficiency levels, and preferred roles to suggest suitable teammates and projects. It also allows users to create detailed profiles, propose ideas, recruit collaborators, and join ongoing initiatives. By combining intelligent matching with real-time collaboration features, the proposed system aims to improve team compatibility, reduce manual effort, and enhance the overall hackathon experience

### II. LITERATURE SURVEY

Hackathons have become an important platform for innovation, rapid problem solving, and collaborative software development. They provide opportunities for students, developers, designers, and professionals to work together within a limited time period to create technological solutions. Although hackathons encourage creativity and teamwork, one of the major challenges faced by participants is forming effective teams with balanced technical and non-technical skills. Many participants struggle to identify suitable teammates, especially in large-scale or virtual

**Volume 15, Issue 5, May 2026****|DOI: 10.15680/IJRSET.2026.1505144|**

hackathons where direct interaction is limited. Trainer and Herbsleb studied the behavior and dynamics of hackathon teams and observed that participants often select teammates based on existing friendships or limited knowledge of each other's abilities. This frequently leads to unbalanced teams, communication issues, and inefficient project execution. Their research highlighted the need for better systems that support structured and skill-oriented team formation instead of relying only on informal networking methods. Falk and Nolte explored the broader impact of hackathons and emphasized that successful collaboration depends not only on technical expertise but also on communication, coordination, and participant motivation. Their findings showed that many existing hackathon platforms focus mainly on registration and submission processes while providing minimal support for collaboration and team-building activities. This creates difficulties for participants who are searching for compatible teammates or suitable project opportunities. Several studies have also discussed the increasing role of artificial intelligence and recommendation systems in collaborative environments. Recommendation systems are widely used in modern digital platforms to analyze user preferences and provide personalized suggestions. Linden, Smith, and York introduced item-to-item collaborative filtering techniques that became highly effective in recommendation-based applications. Such methods can be adapted for hackathon platforms to recommend teammates based on shared interests, complementary skills, and preferred roles. Recent advancements in artificial intelligence have further improved intelligent recommendation capabilities. Sajja et al. explained how generative AI technologies can support innovation events by assisting with idea generation, collaboration, and decision-making. Their research suggests that AI-driven systems can reduce manual effort and improve the efficiency of team management processes during hackathons. Research on real-time collaboration systems has also contributed to the development of modern teamwork platforms. Real-time communication technologies such as WebSockets enable instant data exchange, live chat, and synchronized updates between users. These technologies are commonly used in collaborative applications to improve coordination and communication among team members working remotely. Cloud-based architectures and scalable web technologies are also widely adopted in collaborative platforms. Technologies such as React.js, Node.js, and MongoDB provide flexible and scalable environments for managing user interactions, project data, and communication services. These technologies support responsive user interfaces, efficient backend processing, and secure data management, making them suitable for hackathon management systems. Despite these advancements, most existing hackathon platforms still provide limited intelligent support for participant matching and collaborative project management. Current systems primarily focus on event administration rather than improving team compatibility and participant interaction. There remains a significant gap in platforms that combine AI-based recommendation logic, real-time communication, and collaborative project discovery within a single environment. To address these limitations, the proposed Hackathon Buddy system introduces an AI-based collaborative platform designed specifically for smart hackathon team formation and project management. The system integrates user profile analysis, recommendation techniques, real-time collaboration tools, and project-sharing features to improve the overall hackathon experience. By combining intelligent matching with communication and coordination support, the proposed platform aims to create balanced teams, reduce manual effort, and enhance productivity during hackathon events.

### III. METHODOLOGY

The proposed system is designed to support hackathon participants from registration to team formation and project collaboration. The methodology focuses on collecting user information, analyzing compatibility factors, and using them to recommend suitable teammates and project opportunities. The overall workflow is intended to reduce manual effort and improve the speed and quality of team building during hackathons. The first step is user profile creation. Each participant provides essential details such as name, skills, interests, preferred role, experience level, and domain preference. This information forms the basis for matching and allows the system to understand the user's technical and collaborative profile. A detailed profile helps the platform identify participants with complementary abilities. After profile creation, the system applies a matching process to find suitable teammates or projects. The recommendation logic compares user attributes such as skill overlap, role compatibility, and common interest areas. Participants interested in similar domains or projects are ranked and suggested according to their match quality. This helps users discover relevant collaborators without manually searching through large lists of profiles. The platform also supports idea sharing and project discovery. Users can post new project ideas, browse existing initiatives, and invite others to join their teams. This two-way interaction ensures that both project creators and participants looking for teams can connect through a single environment. It also increases the chances of forming complete and balanced teams. For collaboration, the system includes communication features such as chat, notifications, and project updates. These tools allow participants to discuss ideas, assign responsibilities, and coordinate tasks in real time. The

collaboration layer ensures that team members can remain connected throughout the hackathon event. The implementation uses a layered architecture with a React.js frontend, a Node.js backend, Python-based recommendation logic, and MongoDB for data storage. This structure supports scalability, maintainability, and smooth data flow between modules. The combination of these technologies allows the platform to manage user interactions, team recommendations, and collaboration features efficiently. Overall, the methodology aims to create an intelligent and user-friendly hackathon support system. By combining profile analysis, recommendation logic, and real-time communication, the platform improves team formation and enhances the collaborative experience for participants.

#### IV. SYSTEM DESIGN

Hackathon Buddy is designed as a structured platform to support hackathon participants throughout the full lifecycle of team formation, communication, code collaboration, and data management. The architecture shown in the diagram follows a clear request flow that begins with the user and moves through multiple layers of processing before data is stored permanently. This layered design helps the system remain secure, scalable, and easy to maintain. The process starts when a user opens the web or mobile application. At this stage, the user may register, log in, search for teammates, join a team, communicate with others, or manage project-related activities. Since hackathons usually involve fast-paced teamwork, the platform must provide a smooth and responsive interface that allows participants to act quickly without unnecessary delay. The user layer is therefore the entry point for all actions in the system. After accessing the application, the next important stage is authentication. In this layer, the system verifies the identity of the user before allowing access to protected functions. Authentication services such as Firebase Auth or Auth0 can be used to generate an access token and ensure secure login. This step is necessary because hackathon platforms often store sensitive information such as user profiles, team details, project files, and chat records. By validating each user, the system prevents unauthorized access and improves trust among participants. Once the user is verified, the request is passed to the API gateway. The API gateway acts as the central routing layer of the entire application. Its main role is to receive requests from the client, direct them to the correct backend service, and ensure secure communication between different modules. It also supports file upload and retrieval, especially for project documents, source files, and other shared resources. By separating request handling from application logic, the gateway improves organization and allows the platform to scale more efficiently. The next layer is the application services layer, which contains the main business functions of Hackathon Buddy. This layer manages user profiles, settings, team formation, role-based matching, chat features, and notifications. It is one of the most important parts of the system because it directly supports the hackathon experience. For example, users can describe their technical skills, preferred roles, and areas of interest, and the system can use this information to suggest suitable teammates. This helps participants create balanced teams faster and reduces the difficulty of finding the right collaborators during a short event. In addition to team management, the application services layer also handles real-time chat and notifications. These features are important because hackathon teams need to coordinate quickly and share updates without delay. Notifications can inform users about invitations, team acceptance, messages, or project activity. This improves interaction and keeps all team members informed about ongoing progress. The architecture also includes a real-time communication layer based on WebSocket connections. Unlike traditional request-response communication, WebSockets allow continuous two-way data exchange between the client and server. This makes it possible to support live chat, instant updates, and synchronized collaboration. For hackathon participants, this is especially valuable because they often need to discuss ideas, assign tasks, and solve problems in real time. The live sync of code and chat helps reduce communication gaps and improves teamwork. Another major component is the code execution engine. This module allows users to run code inside an isolated sandbox environment. The purpose of this layer is to provide safe and controlled execution without affecting the main system. Since code submitted by users may contain errors or risky operations, sandboxing is essential for both security and resource management. It ensures that each execution is limited in terms of CPU, memory, and file access, preventing one user's code from interfering with others. This feature is highly useful in hackathon environments where participants need to test their ideas quickly and repeatedly. The final major layer is the data layer, which is responsible for persistence and long-term storage. This layer stores users, teams, chat logs, project metadata, results, and other important records in a cloud database such as Firestore or SQL. It also stores project files, source code, dependencies, and generated outputs in cloud storage. Keeping this information in a persistent layer ensures that data is not lost when sessions end or devices change. It also allows users to revisit their work later, continue development, and review previous activities. The overall flow of the system can be summarized as: User → Authentication → API Gateway → Application Services → WebSockets → Sandbox Execution → Database/Storage. This flow clearly shows how a request moves through the platform from initial access to final storage. Each layer has a distinct responsibility, and

together they create a complete ecosystem for hackathon participation and collaboration. From an IEEE perspective, this architecture is effective because it separates concerns into different modules. Such separation improves security, maintainability, and scalability. It also makes it easier to upgrade individual components without disturbing the whole system. For example, authentication services can be changed without affecting team matching logic, and the data storage system can be expanded without modifying the user interface. Overall, the proposed architecture supports the main goals of Hackathon Buddy: secure login, intelligent team formation, real-time collaboration, safe code execution, and reliable data storage. By combining these components in a layered flow, the platform provides an efficient and practical solution for hackathon participants and organizers. This makes the system suitable for both small college hackathons and larger multiteam innovation events.

**Hackathon Platform – Request Lifecycle Flow**

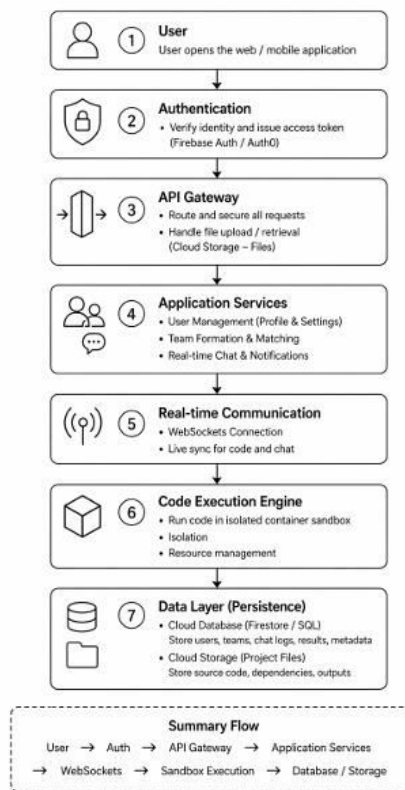


Fig. 1. System Flow Diagram

## V. RESULT

The developed Hackathon Buddy Platform prototype, which includes the Login Module, User Dashboard, and Real-Time IDE, was successfully implemented and tested. The system effectively delivers the primary functionalities required for user authentication, workspace interaction, and live coding within a collaborative setup. The frontend layer, implemented using React.js, offers a clean and user-friendly interface. The authentication process was designed with proper input validation and secure state handling, ensuring that users are redirected smoothly to the dashboard after successful login. The dashboard presents relevant user details along with accessible workspace options, allowing users to navigate the system without difficulty. The Real-Time IDE, built using the Monaco Editor, supports essential programming features such as code editing, syntax highlighting, and real-time updates. The editor performs reliably by accurately capturing user input and maintaining a stable session state. The layout of the IDE is structured to include a dedicated coding area along with optional output or logging sections, improving usability and readability. Overall system performance remained stable during regular usage, with no significant errors or interruptions observed. Error messages were displayed appropriately, aiding in usability and debugging. In conclusion, the prototype successfully

## Volume 15, Issue 5, May 2026

**|DOI: 10.15680/IJIRSET.2026.1505144|**

meets the basic requirements of a Hackathon Buddy platform. It provides secure login functionality, an organized dashboard interface, and a functional real-time coding environment. The results indicate that the system is reliable and can serve as a foundation for further improvements such as collaborative editing, team formation, project management features, and execution environments.

### VI. CONCLUSION

Hackathon Buddy is developed to address the common difficulty of forming effective teams during hackathon events. The project brings together user profiling, team matching, idea sharing, and collaboration features in a single platform, making the hackathon process more organized and efficient. The system helps participants find teammates based on skills, interests, roles, and domain preferences, which reduces manual effort and improves team compatibility. It also supports communication and project coordination, allowing users to work together more smoothly during innovation events. Overall, the project shows that a smart collaboration platform can improve both participation and productivity in hackathons. Hackathon Buddy provides a practical foundation for better team formation and can be further enhanced in the future with more advanced recommendation, collaboration, and event-management features.

### VII. RESULT IMAGES

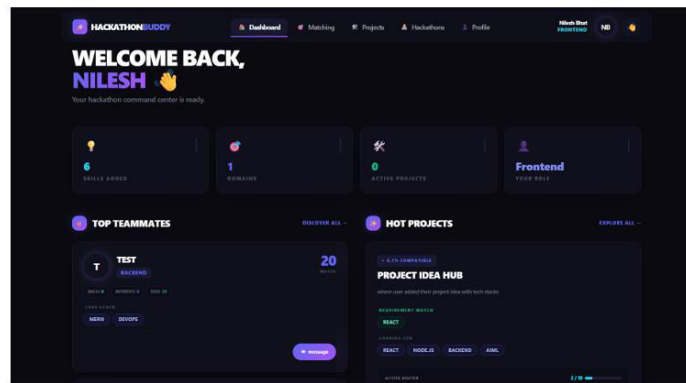


Fig. 2. Login Page

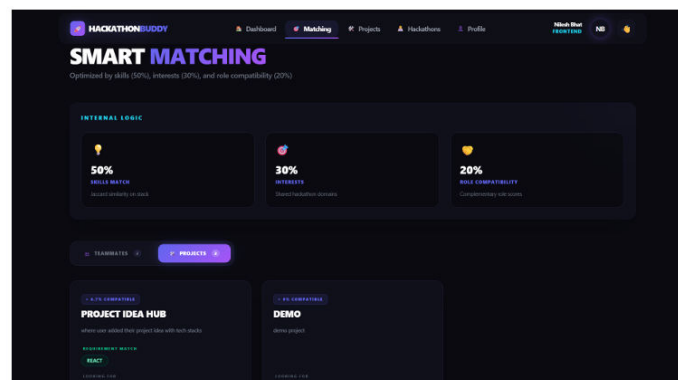


Fig. 3. Dashboard Interface

**Volume 15, Issue 5, May 2026**

**|DOI: 10.15680/IJIRSET.2026.1505144|**

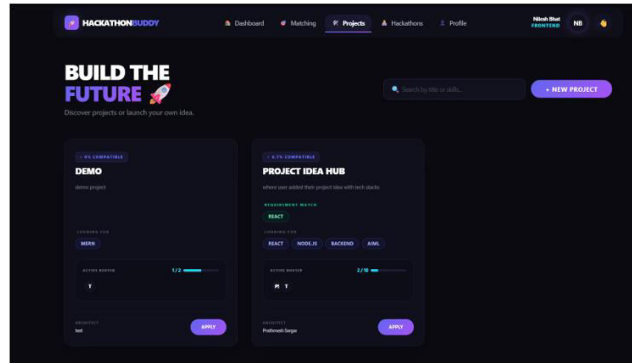


Fig. 4. Project Workspace

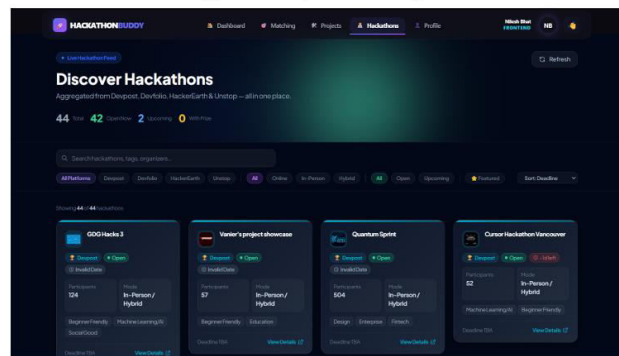


Fig. 5. Real-Time IDE

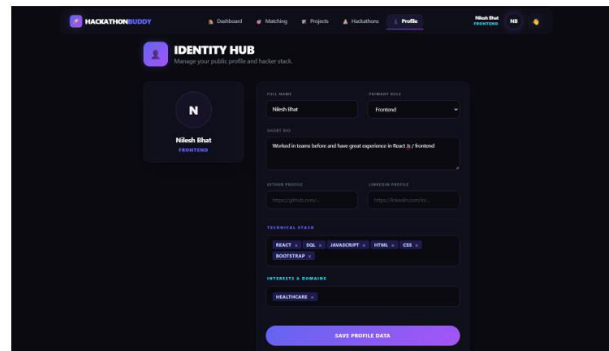


Fig. 6. Team Matching Interface

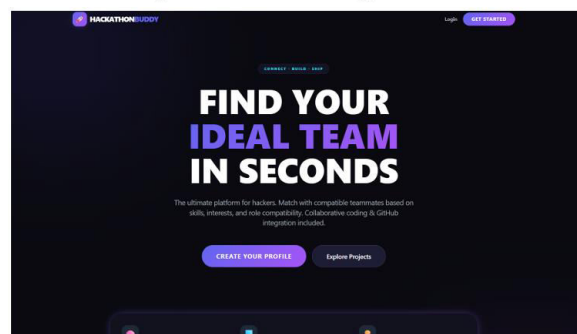


Fig. 7. Final Output Screen

**REFERENCES**

- [1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting”, in Proc. SIGGRAPH, pp. 417–424, 2000.
- [2] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting.”, IEEE Transactions on Image Processing, vol. 13, no.9, pp. 1200–1212, 2004.
- [3] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, Stanley Osher, “Simultaneous Structure and Texture Image Inpainting”, IEEE Transactions On Image Processing, vol. 12, No. 8, 2003.
- [4] Yassin M. Y. Hasan and Lina J. Karam, “Morphological Text Extraction from Images”, IEEE Transactions On Image Processing, vol. 9, No. 11, 2000
- [5] Eftychios A. Pnevmatikakis, Petros Maragos “An Inpainting System For Automatic Image Structure-Texture Restoration With Text Removal”, IEEE trans. 978-1-4244-1764, 2008
- [6] S.Bhuvaneshwari, T.S.Subashini, “Automatic Detection and Inpainting of Text Images”, International Journal of Computer Applications (0975 – 8887) Volume 61– No.7, 2013
- [7] Aria Pezeshk and Richard L. Tutwiler, “Automatic Feature Extraction and Text Recognition from Scanned Topographic Maps”, IEEE Transactions on geosciences and remote sensing, VOL. 49, NO. 12, 2011
- [8] Xiaoqing Liu and Jagath Samarabandu, “Multiscale Edge-Based Text Extraction From Complex Images”, IEEE Trans., 1424403677, 2006
- [9] Nobuo Ezaki, Marius Bulacu Lambert , Schomaker , “Text Detection from Natural Scene Images: Towards a System for Visually Impaired Persons” , Proc. of 17th Int. Conf. on Pattern Recognition (ICPR), IEEE Computer Society, pp. 683-686, vol. II, 2004
- [10] Mr. Rajesh H. Davdal, Mr. Noor Mohammed, “ Text Detection, Removal and Region Filling Using Image Inpainting”, International Journal of Futuristic Science Engineering and Technology, vol. 1 Issue 2, ISSN 2320 – 4486, 2013
- [11] Uday Modha, Preeti Dave, “ Image Inpainting-Automatic Detection and Removal of Text From Images”, International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622 Vol. 2, Issue 2, 2012
- [12] Muthukumar S, Dr.Krishnan .N, Pasupathi.P, Deepa. S, “Analysis of Image Inpainting Techniques with Exemplar, Poisson, Successive Elimination and 8 Pixel Neighborhood Methods”, International Journal of Computer Applications (0975 – 8887), Volume 9, No.11, 2010



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