

PROJECT ABSTRACT

Hackathon Management Platform

A full-stack web platform running hackathons end to end: event discovery, team registration, project submission, weighted-criteria judging and a live provisional leaderboard, with organizer, hacker and judge roles. Delivered built-to-order with complete source code, database schema, seed data, report, PPT and viva kit.

01. Title

Hackathon Management Platform

02. Introduction / Background

Hackathons have become the default way colleges and companies spot engineering talent, yet most are still run on a patchwork of Google Forms, spreadsheets and WhatsApp groups. Registrations get lost, judging criteria are applied inconsistently across judges, and participants never see where they stand. The engineering gap is a single system that owns the whole lifecycle: event publishing with timelines, team registration, project submission, structured judging and transparent results. This project builds exactly that as a full-stack web application — a discovery portal for hackers, a detail page per hackathon with timeline and prizes, and an organizer dashboard where judges score submissions against weighted criteria and a provisional leaderboard updates as scores come in.

03. Problem Statement

Running a hackathon on forms and spreadsheets breaks down at scale: team sizes violate event rules silently, submissions arrive in five different formats, and judging collapses to a single gut-feel number per judge that cannot be compared or audited. The project addresses this by modeling every role and workflow explicitly — organizers publish events with criteria weights, hackers register validated teams and submit in one format, judges score the same four criteria for every submission — so the leaderboard is a deterministic function of published weights, and the whole journey is demoable in the viva.

04. Objectives

- Build a hackathon discovery portal with search, dates, mode, prize pools and theme tags.
- Implement per-event detail pages: timeline milestones, prize breakdown and judging-criteria weights.
- Support team registration with member details validated against event team-size rules.
- Provide project submission with description, demo video link and repository URL before the deadline.
- Implement structured judging: four weighted criteria (innovation 30, technical 30, impact 20, demo 20) instead of a single score.

- Compute a live provisional leaderboard from the published weights as judges submit.
- Enforce role-based access for organizer, hacker and judge across every workflow.

05. Proposed Methodology

The system is built as a React single-page app over a Node.js/Express REST API backed by PostgreSQL. Development proceeds workflow by workflow: (1) event CRUD with timeline and prize models; (2) discovery listing with filters; (3) team registration with server-side team-size validation; (4) submission intake with deadline enforcement; (5) judge assignment and criterion scoring; (6) the leaderboard aggregator that combines criterion scores by event weights. JWT authentication with role guards protects every route, and realistic seed data (six hackathons, a 342-team flagship event) makes every screen demoable immediately.

06. System Architecture / Working

The organizer creates a hackathon — name, dates, mode, prize breakdown, timeline milestones and criteria weights. Hackers discover events on the portal, form teams of 2–4 and register before the deadline. At submission time teams upload project details, demo link and repository. The organizer assigns judges; each judge scores assigned submissions against the four weighted criteria. The leaderboard aggregates scores per the criteria weights (total out of 100) and shows provisional rankings as judging progresses; when all scores are in, results are finalized with the prize distribution. The demo's organizer dashboard shows submissions, judging progress and the sorted provisional leaderboard.

07. Hardware / Software Requirements

- React frontend (single-page app)
- Node.js + Express REST API
- PostgreSQL relational schema with migrations and seed data
- JWT authentication with role-based route guards
- REST API documentation
- Dockerized deployment
- Git version control
- Report PDF, PPT and viva Q&A; documents

08. Expected Outcomes

- A working multi-role platform covering the full hackathon lifecycle.
- Deterministic, auditable scoring from published criteria weights.
- A live provisional leaderboard that updates as judges submit.
- A complete report, PPT and viva Q&A; on full-stack architecture and scoring logic.

09. Applications

- College technical festivals running internal hackathons.
- Corporate hiring and innovation challenges.
- Community and open-source sprint events.
- Final-year extensions into peer voting or sponsor dashboards.

10. Future Scope

- Automatic judge-to-submission allocation.
- Payment-gateway integration for paid hackathons.
- WebSocket push for true real-time leaderboard updates.
- Email/SMS notification wiring for deadlines and results.