



HACKATHON GUIDELINES

How the Event Runs & What Every Team Must Submit

CHOOSE ANY ONE TOPIC :

1. AI FOR EVERYDAY PROBLEMS

Identify a real-world problem faced by **students, consumers, professionals, or businesses** and build an AI-powered solution that makes the experience smarter, faster, easier, or more accessible. Teams must demonstrate a clear customer need and build a business around the solution.

2. THE FUTURE OF WORK

Reimagine how people **learn, work, get hired, or grow their careers**. Identify an emerging challenge in employment, education, skills, recruitment, or workplace productivity and develop a solution with a clear target customer, market opportunity, and sustainable business model.

3. BUILD FOR A BETTER FUTURE

Choose a pressing real-world challenge around **sustainability, mobility, waste, energy, financial inclusion, or community needs** and turn it into a commercially viable solution. The idea should demonstrate both **meaningful impact and potential to scale**.



What Is This Hackathon?

A 24-hour, in-person build sprint where cross-functional teams design, build, and pitch a working prototype that solves a real problem statement — culminating in live demos to a judging panel.

24

Hours of build time

3-4

Members per team

5

Open problem tracks

₹3L+

Total prize pool

QUICK FACTS

- Mode: On-site, main campus auditorium
- Format: Offline hacking, offline judging
- Eligibility: Open to all undergrad & postgrad students
- Tech: Any language, framework, or platform
- Mentorship: Domain mentors available every 4 hours
- Registration: Team-based, via official event portal



How the 24 Hours Are Structured

Day 0, 6 PM	Day 1, 9 AM	Day 1, 6 PM	Day 2, 12 PM	Day 2, 2 PM	Day 2, 5 PM
Kickoff & Team Check-in Problem statements released, teams confirm rosters	Mentor Round 1 Validate your idea and approach with domain mentors	Progress Checkpoint Show working progress; pivot if the approach isn't working	Submission Deadline Code freeze — push repo and upload pitch deck	Live Demos & Judging 5-minute demo + 3-minute Q&A per team	Results & Awards Winners announced across all tracks

Late submissions are not accepted under any circumstance — code freeze is strict.



Team Formation & Ground Rules



Team size: 3 to 4 members

Cross-college teams are allowed; no solo entries



Original work only

Code and design must be built during the event window



Plagiarism = disqualification

Reused prior submissions will be removed from judging



One idea, one team

Each team submits exactly one project and one deck



Open-source libraries OK

Pre-built libraries and APIs are fine; pre-built products are not

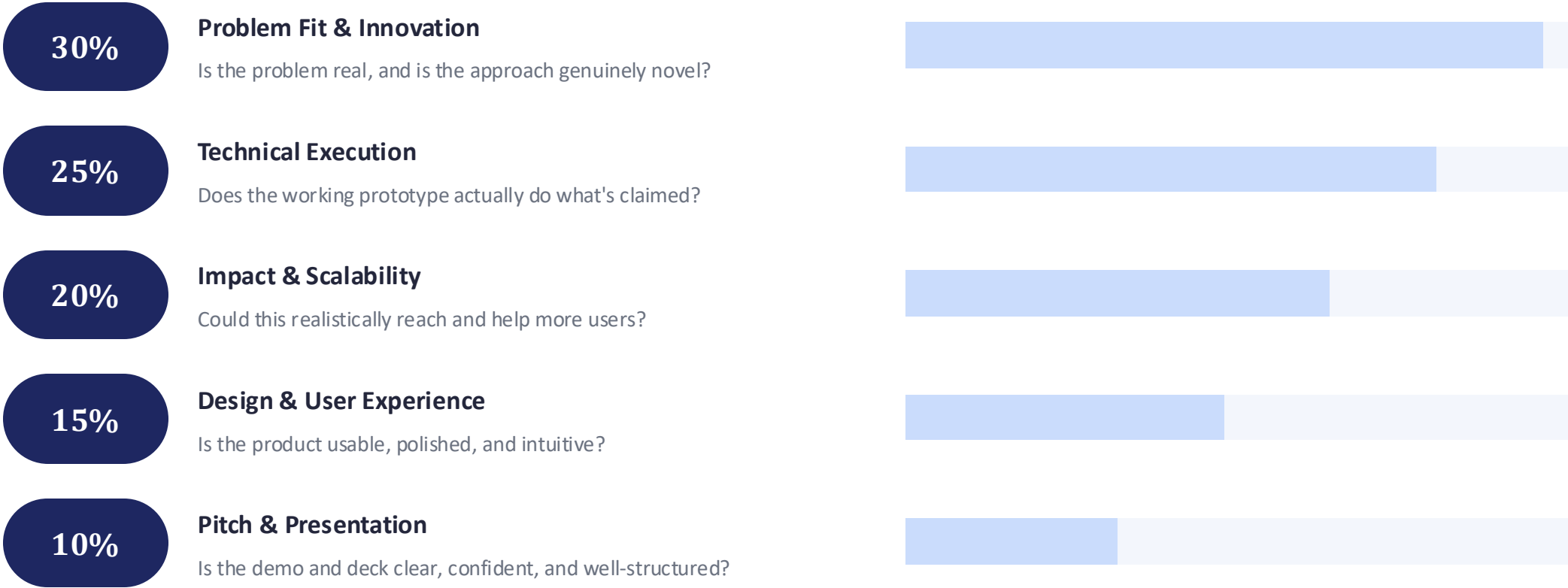


Respect the code of conduct

Harassment or unsportsmanlike conduct ends participation



How Submissions Are Scored





SUBMISSION REQUIREMENTS

Your Pitch Deck: Slide by Slide

The 10 slides every team must include, and what judges look for in each



Team Introduction

What This Slide Must Include

- Introduce all team members and their roles (dev, design, product, etc.)
- Mention relevant skills or past experience that apply to this build
- Name your team and give the project a working title
- Keep it to one slide — this isn't the place for long bios

PRO TIP

Judges remember teams that sound organized. A clear role split signals you can actually execute.



Problem Statement

What This Slide Must Include

- Clearly state the real-world problem you chose and why
- Explain who is affected and how, with a concrete example
- Back it up with a stat, quote, or scenario — not just an assertion
- Keep it to 3–4 crisp points; avoid vague or generic framing

PRO TIP

Judges should feel the pain point within 10 seconds of reading this slide.



Proposed Solution

What This Slide Must Include

- Describe what you built and how it solves the stated problem
- Walk through the core user flow in 3–4 steps
- Call out what makes this approach different from an obvious fix
- Use a diagram or screenshot wherever possible, not just text

PRO TIP

If a judge can't picture the product from this slide alone, add a visual.



Tech Stack & Architecture

What This Slide Must Include

- List the languages, frameworks, and APIs actually used
- Include a simple architecture diagram — client, server, data, integrations
- Flag anything technically ambitious you're proud of
- Be honest about what's mocked vs. fully functional

PRO TIP

A clean architecture diagram builds more technical credibility than a bullet list ever will.



Prototype / Demo Walkthrough

What This Slide Must Include

- Include screenshots or a short embedded clip of the working product
- Sequence visuals to match the actual demo you'll give live
- Label each screen so it makes sense without narration
- Note clearly if a feature is planned but not yet functional

PRO TIP

This slide is your safety net — if the live demo hiccups, it still proves the build works.



Innovation & Uniqueness

What This Slide Must Include

- State what makes your approach different from the obvious solution
- Reference any existing tools or approaches you're improving on
- Highlight any creative or unconventional use of technology
- Summarize your edge in one clear, bolded sentence

PRO TIP

If every team could describe your idea the same way, it isn't distinct enough yet.



Implementation Feasibility

What This Slide Must Include

- Confirm what's actually built and running vs. what's conceptual
- Note any real data, APIs, or devices your prototype connects to
- Explain what you'd need to take this from prototype to production
- Flag known limitations honestly rather than glossing over them

PRO TIP

Self-awareness about limitations reads as maturity, not weakness, to judges.



Impact & Scalability

What This Slide Must Include

- Estimate who benefits and how many people this could reach
- Explain how the solution would scale beyond the hackathon build
- Mention any social, environmental, or economic impact angle
- Ground estimates in a simple assumption, not a guess

PRO TIP

Show your math — a number without a visible assumption behind it looks made up.



Challenges & Learnings

What This Slide Must Include

- Share one or two real technical or scoping challenges you hit
- Explain how the team adapted or pivoted in response
- Note anything you'd do differently with more time
- Keep this brief and specific, not a general reflection

PRO TIP

Judges have built things too — an honest challenge story builds trust fast.



Future Roadmap & Ask

What This Slide Must Include

- Outline the next 2–3 features you'd build post-hackathon
- Mention any mentorship, resources, or partnerships you'd want
- Close with what you're asking of judges — feedback, funding, or next steps
- End on a confident, forward-looking note

PRO TIP

A concrete next step shows this project doesn't end when the hackathon does.



SLIDE 11

Thank You Slide

- Thank the judges, mentors, and organizers for their time
- Restate your project name and one-line tagline
- Include contact details — team lead name, email, phone
- Add a QR code or link to your GitHub repo if available
- Invite questions and open the floor for discussion

End with confidence — this is your last impression on the judges.