

GFG HACKOVATION 2026: MENTOR & EVALUATOR HANDBOOK

Role: Mentor, Technical Guide, and Auditor.

Phase 1: The Mentoring Round (2:00 PM - 3:00 PM)

Focus: Architecture & Feasibility

At this stage, teams should have a clear plan. They shouldn't have a finished product (that's suspicious), but they should have the skeleton.

Questions to Ask: 1. **The Approach:** "Walk me through your database schema. How are you handling [Specific Feature]?" 2. **The Twist:** "Have you seen the Twist for your problem statement? How are you planning to integrate it?" 3. **The Blocker:** "Is there any specific error or logic you are stuck on right now?"

Your Job: * If they are overcomplicating (e.g., trying to use Microservices for a simple MVP), advise them to simplify. * If they are stuck on a bug, point them to the right documentation or logic. **Do not write the code for them.**

Phase 2: Mid-Hack Evaluation (10:00 PM - 11:00 PM)

Focus: Progress Check & Integrity (Anti-Cheating)

This is the most important round. You are checking for "**Vibe Coding**" (Blindly copying AI code) and **Pre-Built Projects**.

The Rubric (Pass/Flag): Mark teams on a simple 1-5 scale. Flag any team with a 'Red Flag'.

Criteria	What to Look For	Red Flags
Git Activity	Look at the commit history. Are there regular, small commits?	"Initial Commit" containing 50 files at once. No commits since morning.
Code Comprehension	Ask a specific question about a function.	Student stammers, looks at teammate, or cannot explain the logic of a file they "wrote."
Progress	Is there a basic UI or Backend connectivity working?	A fully polished app with zero bugs (suspiciously fast) OR Zero progress (just a PPT).

Criteria	What to Look For	Red Flags
The Twist	Have they started coding the logic for the Twist?	Ignoring the Twist completely to focus on the pre-planned idea.

The “Vibe Check” Questions (To Catch Cheaters): 1. *“Open `utils.js` (or any helper file). Explain what this specific regex/function does.”* 2. *“Show me the commit where you fixed the last bug you encountered.”* 3. *“Delete this line of code. What will break?”* (If they don’t know, they didn’t understand it).

Phase 3: General Guidelines for Mentors

1. **Be Encouraging:** For first-year teams, be lenient on code quality but strict on effort. If they are struggling, encourage them to complete *one* feature perfectly rather than a broken full app.
2. **Be Neutral:** If you are mentoring a friend’s team, please swap with another mentor to avoid conflict of interest.
3. **Reporting:** If you suspect a pre-built project, do not confront them aggressively. Note the Team Name and inform the Core Team immediately. We will run a deeper audit.