



# HACKS.10

## Problem Statements

### 1. Entrepreneurship — AI Co-Founder for Student Startups

Student ideas stall for lack of structure; build an AI co-founder that helps teams move from idea to a testable MVP. The tool should help refine the problem statement, identify the earliest customer segment, sketch a simple business model, and auto-generate a one-page investor pitch and a beginner financial canvas (basic revenue, cost, runway). Unique features to demo: a live “idea validation” module that suggests 3 low-cost experiments (surveys, landing pages, paid ads) and a co-founder matching suggestion engine that finds complementary skillsets within the campus. Deliver a prototype that generates a concise, actionable startup plan from a short user prompt.

### 2. Sales & Marketing — Virtual Campaign Lab with Optimizer

Marketing budgets are precious; give founders a safe place to practice. Build a virtual campaign lab where teams input product details, target demographics, channel preferences, and budget, then receive a simulated campaign outlook: projected reach, engagement, and conversion likelihood. The lab should include an optimizer that proposes channel splits and a simple creative scoring tool (NLP evaluation of ad text) to compare variants. Teams should show at least one simulated campaign and an “optimize” suggestion that reallocates budget for better predicted ROI. Optional integrations (mocked) with ad platforms are fine — the point is predictive rehearsal, not live ad spend.





# HACKS.10

## Problem Statements

### 3. Agentic AI — Campus Workflow Assistant

Colleges run on countless small processes that eat time: assignment collection, quiz creation, timetable changes, peer-mentoring coordination, faculty approvals, and notification cascades. The challenge is to design a practical agentic application that goes beyond a Q&A bot and can act on behalf of users by autonomously executing defined workflows: for example, generate a quiz from a lecture transcript and schedule it on the LMS, create and distribute a revision pack, or coordinate a study group based on calendars and availability. Teams should show an interface for defining “action templates” (what the agent is allowed to do), an execution log for transparency, and at least two automated end-to-end demo flows (e.g., lecture → summary → auto-quiz published; absence request → approval workflow). Focus on safe, auditable actions and a lightweight, realistic integration (mocked or via public APIs) rather than enterprise-grade deployment — deliverable is a working prototype that proves the agent can make decisions and carry out tasks within policy constraints.

### 4. Web3 for Alumni & Revenue — Decentralized Alumni Marketplace

Alumni networks are full of goodwill but often fail to be sustainable or useful over time. Design a blockchain-backed alumni platform where verified alumni and students can connect, offer mentorship, post paid workshops, or sell services (resume reviews, mock interviews, micro-courses), and where participation/reputation is transparently tracked. Revenue features should be built into the model: paid listings, micro-payments for office hours, sponsored job posts from companies, and optional premium subscriptions for verified mentors. Teams should demonstrate identity verification (mock DigiLocker / email/KYC flow), a simple on-chain ledger of contributions, and two monetization flows (e.g., paid booking and a subscription). The goal is a trustworthy, low-friction space where alumni engagement self-sustains through real transactions and measurable impact.





# HACKS.10

## Problem Statements

### 5. Campus Navigator + SLMs — Freshers' Guided Onboarding

Freshers' apps are usually just maps and FAQs; make something that actually teaches new students how to be part of campus life. The task is to design a campus navigator that combines geolocated directions with micro Structured Learning Modules (SLMs) — short, scaffolded lessons about campus systems (library usage, exam rules, hostel guidelines, lab safety) delivered as interactive micro-modules with quick checks and badges. Add peer-contributed “campus hacks” and mentor micro-sessions so new students can both learn and get human help. Teams should demonstrate a guided pathway (e.g., a “first week” SLM sequence), one interactive quiz-badge, and a map for locating help points or mentors. Make it low-friction, offline-resilient (caching SLMs), and built for rapid adoption by beginners.

### 6. AI for Education — Personalized Study Companion with Concept Maps

Students waste time on generic materials; build a study companion that adapts and explains. The challenge is to create an AI system that ingests a student's past scores or short self-assessments, builds a lightweight concept graph for a subject, and then generates a personalized study plan that includes short micro-lectures, adaptive practice questions, and a spaced-repetition schedule. The unique feature is a visual, auto-generated concept map that shows weak spots and how topics connect, with “explain at level X” controls (overview → detail → ultra-simple). Teams should show at least one subject flow: student input → concept map → targeted practice → progress update. Keep the AI integration seamless and transparent; a small, useful prototype that demonstrably improves study focus is the deliverable.





# HACKS.10

## Problem Statements

### 7. FinTech — Seamless AI Expense Buddy for Students

Students need money sense, not spreadsheets. Build a smart expense companion that automatically parses receipts, categorizes expenses, predicts future cashflows, and suggests micro-saving nudges — all powered by AI so categorization and insights feel instant and human. The required twist: seamless, contextual AI prompts — for example, when the app sees repeated late-night food expenses it proactively suggests a “meal budget” and an automated weekly saving plan, or it drafts a short message to roommates splitting a bill. Teams should implement AI-driven receipt parsing, at least one proactive nudge feature, and a simple “what-if” simulator (e.g., how saving X per week changes outcomes). The experience must feel like the app is thinking about the user’s money and offering concrete, automatic help.

### 8. Cybersecurity — Student Digital Identity Vault and Selective Disclosure

Storing certificates and sharing them with recruiters is messy and risky. The challenge is to build a secure student vault that supports verifiable credentials and selective disclosure: students should be able to prove a claim (degree, grade, certificate) to a verifier without revealing extraneous data. Teams should show a simple user vault UI, an issuance/verification demo (a mock university issues a credential), and a selective-share flow where a recruiter receives only what they need via a time-limited QR or link. Bonus: include audit trails, simple revocation, and an easy recruiter experience so sharing feels seamless. Emphasize privacy, ease of use, and a short but solid demonstration of verifiable trust without raw document exposure.





# HACKS.10

## Problem Statements

### 9. 3D Micro-Simulators — Career Micro-Experiences (WebGL)

Instead of AR/VR, create lightweight, browser-based 3D micro-simulators that let students try day-in-the-life tasks: a short 3D interaction module for roles like product designer (wireframing task), retail manager (stocking/planogram), or lab technician (safety workflow). The aim is to let students understand job tasks through interactive decision points and quick performance feedback — all delivered via low-poly WebGL so it runs on normal laptops/browsers. Teams should build one 3D micro-experience, a mini scoring rubric, and an exportable “3D resume snippet” students can share. Keep scenes focused and short (2–5 minutes) so prototypes are feasible in 36 hours but still convey meaningful career exposure.

### 10. Civic Sprint — Community Repair, Sponsorship & Transparency

Municipal complaint portals are transactional; reimagine one that turns small civic problems into community sprints. The platform should let citizens log issues with photo + location, allow neighborhood groups or student volunteers to pick tasks, and enable local businesses to sponsor fixes or provide micro-grants. Add a progress timeline and a transparency dashboard showing outcomes and budgets. Teams should demo a flow from complaint → volunteer/team assignment → sponsored resolution → outcome report; incorporate a simple reputation or incentive mechanism for citizens and sponsors. The goal is faster grassroots repair with civic pride and local accountability, not heavyweight municipal integration.





# HACKS.10

## Problem Statements

### 11. Farmer Marketplace — India-First, Farmer-Centric Trade Portal

Make a marketplace built for Indian farmers' realities: local language interface, mandi price aggregation, MSP alerts, and collective listings so smallholders can pool produce and access better rates. The system should support both direct listings and bid/auction flows, show historical price trends per district, and offer templated contracts for advance sales. Teams should include a WhatsApp/SMS interface for listing and bidding (so farmers without smartphones can participate), a basic buyer dashboard, and a demo of a collective sell flow where multiple small producers list a combined lot. Emphasize practical adoption — low complexity, clear farmer benefits, and a path to fairer pricing.

### 12. Open Innovation — Hack for Bharat (Wildcard)

This track is intentionally open: pick any real Indian problem and design a software-first solution that shows clear local impact. What matters is evidence of problem understanding, a concrete demo, and a plausible rollout plan. Examples could include a "safe commute" app for night-shift workers that coordinates escorts and shares verified routes, a small-business digitalization kit for neighbourhood kiranas, or a platform for women artisans to receive pre-orders and micro-credit. Teams must articulate the user, the pain they solve, a working prototype, and an adoption strategy tailored to India. Be bold, local, and impact-oriented.