

Problem Statement

Agentic EdTech for Regional Language & Geography

1. Background

Educational technology has widened access to learning content across India, yet its benefits are not evenly shared. Most platforms are built in English or Hindi and assume a stable, high-speed internet connection and a capable personal device. Learners from regional-language backgrounds, particularly in rural and semi-urban areas, frequently do not meet these assumptions.

For these learners, language and geography become barriers to education. A student may understand concepts best in Odia or another regional language, may rely on a shared or low-end device, and may have slow or unavailable connectivity for long periods. Under such conditions, a platform that is formally available is still not practically usable.

Nextech 2.0 invites participants to address this gap by building agentic AI solutions that work in the learner's own language, remain usable with limited connectivity, and take meaningful action on the learner's behalf.

2. The Problem Statement

Design and build an agentic AI solution in the EdTech domain that improves access to quality education for learners across diverse linguistic and geographical backgrounds. The solution must have a genuinely agentic core — a system that can plan, decide, use tools and act toward an educational goal, not merely respond to a prompt. **Regional-language support, ease of use, offline compatibility and a well-chosen agentic framework earn bonus credit.**

2.1 The Problem in Context

- **Devices are often available.** ASER 2024, a survey of rural India, found that almost 90% of 14–16 year-olds report having a smartphone at home and over 80% know how to use one. Owning a device is no longer the only gap; what the device offers the learner is.
- **Language is already recognised as a barrier.** Odisha has run mother-tongue based multilingual education in tribal regions since 2007. Education authorities have developed over 300 textbooks and 2,500 supplementary reading materials in 21 tribal languages, and Project Samhati aims to reach about 2.5 lakh students in 1,450 primary schools.

The opportunity lies between these facts: learners already have devices, and mother-tongue support exists in print. The question is how to bring intelligent, personalised support in the learner's own language onto the devices they already have.

Sources: ASER 2024 (Pratham); Odisha School Education Programme Authority and Project Samhati reporting; S. Mohanty, MTB-MLE in Odisha (2015).

2.2 Reference Case: A Class 9 Student in Rural Odisha

An illustrative composite. The person is fictional, but the conditions reflect those described above.

Sunita is in Class 9 at a government school in a tribal-majority block of southern Odisha. She speaks Kui at home, is taught in Odia at school, and meets English mostly through textbooks. Her family has one smartphone, which her father uses during the day; she gets it for about an hour in the evening. Mobile signal in the village is weak and her data pack is limited. Her science teacher covers several classes and cannot answer every doubt.

When	What happens	Where it fails
In class	The teacher explains in Odia, with English technical terms. Sunita follows about half. In a crowded class there is no time for individual doubts.	Language; no individual attention
Evening, one hour on the shared phone	She opens a popular learning app. The interface is in English and the lessons are English or Hindi videos. Video buffers on the weak signal and the data pack runs low.	Language; connectivity
A doubt	She types a science question into a search engine. Typing Odia is slow, and the results are in English and pitched above her level.	Language; wrong level

When	What happens	Where it fails
Before tests	She memorises answers she does not understand. No one tracks what she knows or tells her what to study next.	No guidance; no follow-up

What this case asks of your solution

- **Understand her doubt** in her own language, by voice or text, and answer at her level. (Bonus: regional language)
- **Keep working when the signal drops**, and resume where she left off. (Bonus: offline)
- **Notice what she has not understood**, decide what she should study next and follow up without being asked. (Agentic core)
- **Be simple enough** for a first-time user on a shared, low-end phone. (Bonus: ease of use)

Your solution need not serve this exact learner. Choose your own user and problem, but show the same depth of understanding.

Other cases to reference

- **The multi-grade teacher.** A primary teacher in a remote school handles several classes at once, in a language many pupils speak only partly, with no way to track each child's progress or give individual attention.
- **The exam aspirant in a small town.** A Class 12 student in an Odia-medium school prepares for competitive exams far from the coaching hubs, with no affordable mentor, study plan or practice feedback in the language they think in.

3. EdTech Is Easy. Agentic EdTech Is Not.

A login, a dashboard, a quiz and a chatbot can be shipped in a day. That is not the challenge. The challenge is a system that decides what happens next, takes action and handles failure. A thin wrapper around an LLM API is not an agent: if removing the AI leaves the same app, the AI was decoration.

	Conventional EdTech app	Agentic EdTech system (what we want)
Behaviour	Waits for clicks; fixed screens	Pursues a goal; decides the next step itself
Role of AI	A chat box added on top	The engine: plans, calls tools, acts
Language	Translated labels and menus	Reasons and replies natively in Odia / Indic
Adaptation	Same content for everyone	Adapts to each learner's progress
When things fail	Errors out or stops offline	Recovers, falls back, syncs later

4. Evaluation Criteria

Evaluation has two tiers. Core criteria are must-haves: every team is assessed on them. Bonus criteria are add-ons that raise the score of a solution that already meets the core.

Core criteria (must-have)

#	Criterion	What evaluators look for
1	Problem Understanding	Clarity on who the learner is, what gap language and geography create for them, and why the chosen problem matters.
2	Technicalities	Soundness of the architecture, depth of the agentic implementation (tool calling, decision loops, state) and reliability of the working MVP.
3	Presentation	Clarity and confidence of the live demo and delivery, and how well the team answers questions.
4	Pitch Deck	Structure and storytelling: problem, solution, architecture, impact and feasibility, for industry CTOs and venture partners.

Bonus criteria (add-ons)

#	Criterion	What evaluators look for
+	Regional Language Support	Native-feeling Odia / Indic support — interface, content, translation, voice and AI assistance — not just translated labels.
+	Ease of Use & Accessibility	A student, teacher or admin with low digital confidence can use it without training. (Final wording to be confirmed by the organisers.)
+	Offline Compatibility	Stays useful on poor or intermittent connectivity, and fully offline where feasible — cached content, edge models, sync on reconnect.
+	Type of Agentic Framework	Deterministic or autonomous (see section 5), how well it fits the problem and how deeply it is implemented.

5. Choosing Your Agentic Approach

Bonus credit for the agentic framework depends on fit and depth. Two approaches are recognised, and neither is “better”. Use any framework (e.g. LangChain, AutoGen, CrewAI, LlamaIndex, Smolagents) and be ready to justify your choice.

Approach	How it works	Suits problems like
A. Deterministic Workflow	Predefined, workflow-based agent: fixed steps and rules, with AI inside defined nodes.	Predictable, auditable tasks. Easier to make reliable and to run offline.
B. Autonomous Workflow	An agent that decides which actions and tools to use, and in what order, to reach a goal — planning and self-correcting.	Open-ended tasks that need judgement, planning or multi-agent coordination.

6. What to Deliver

Three things are expected from every team:

Deliverable	What it must show	When / to whom
1. Working MVP	A live, end-to-end agent flow with real tool calls. Not static mock screens.	Live demo at final evaluation
2. Mid-evaluation checkpoint	Chosen problem and target users, agent architecture, a running skeleton of the agent loop, and any bonus features you plan to attempt.	Mandatory, during the sprint
3. Live pitch deck	Problem, users, solution, architecture, agentic approach, demo and impact, plus how any bonus features are addressed.	Presented live to industry CTOs and venture partners

What evaluators will check in your live demo

Problem understanding	You can state the user and the gap in a sentence, and the demo solves exactly that.
Technicalities	The agent’s decision trace: which tools it called, in what order, and why.
Bonus: language	Interact with the system in Odia or another Indic language, by text or voice.
Bonus: ease of use	A first-time user completes the main task without being guided.
Bonus: offline	Network switched off live; the fallback works and data syncs back later.
Bonus: framework	A clear reason for choosing deterministic or autonomous.

Questions every team should be ready to answer

- Who is your user, and what concretely changes for them?
- Bonus: why deterministic or autonomous, and what would break if you chose the other?
- Which tools does your agent call, and what happens when one fails?
- Bonus: what happens when the network drops mid-task?
- Bonus: how does the language experience go beyond translation?

The best submissions will not have the most screens. They will be the ones where an agent quietly does the work — in the learner’s own language, on a weak connection — and the learner simply learns.