

 **CodeMatic: A Learning Platform for Student** 

**Team Name :** Code Crew

**Leader Name :** Abhishek Namdeo

**Submission Date:** 05/09/2025

**College Address:** Oriental College of Technology

**<CODE>  
CREW>**

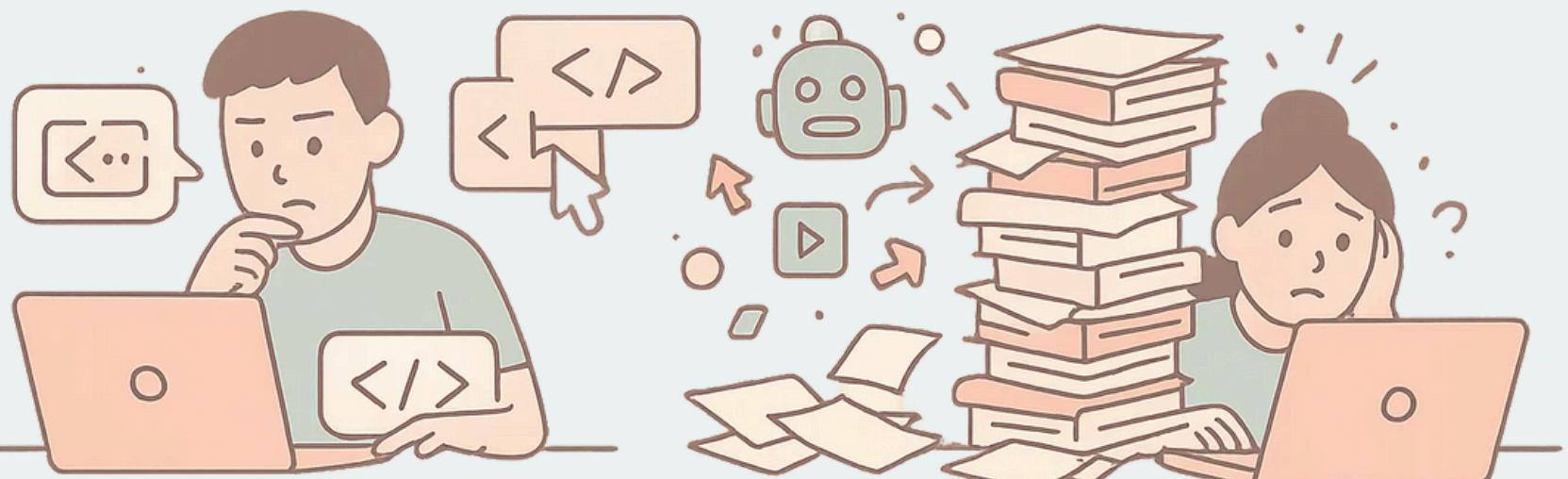
## Area Overview

### **Problem:**

- Current AI tools mostly give only raw code without explanation.
- Students copy–paste instead of understanding the concepts.
- Rural learners face language and comprehension barriers.
- Docs and tutorials are often confusing and English-heavy.
- Need for a learning-focused AI tool that explains step by step.

### **Importance of Area:**

- Makes coding understandable and accessible for all students.
- Bridges the language and learning gap in rural education.
- Helps students develop logic and problem-solving skills, not just coding.
- Supports schools, colleges, and labs with AI-powered learning.
- Contributes to quality education and digital empowerment.



← → Q Problem Context

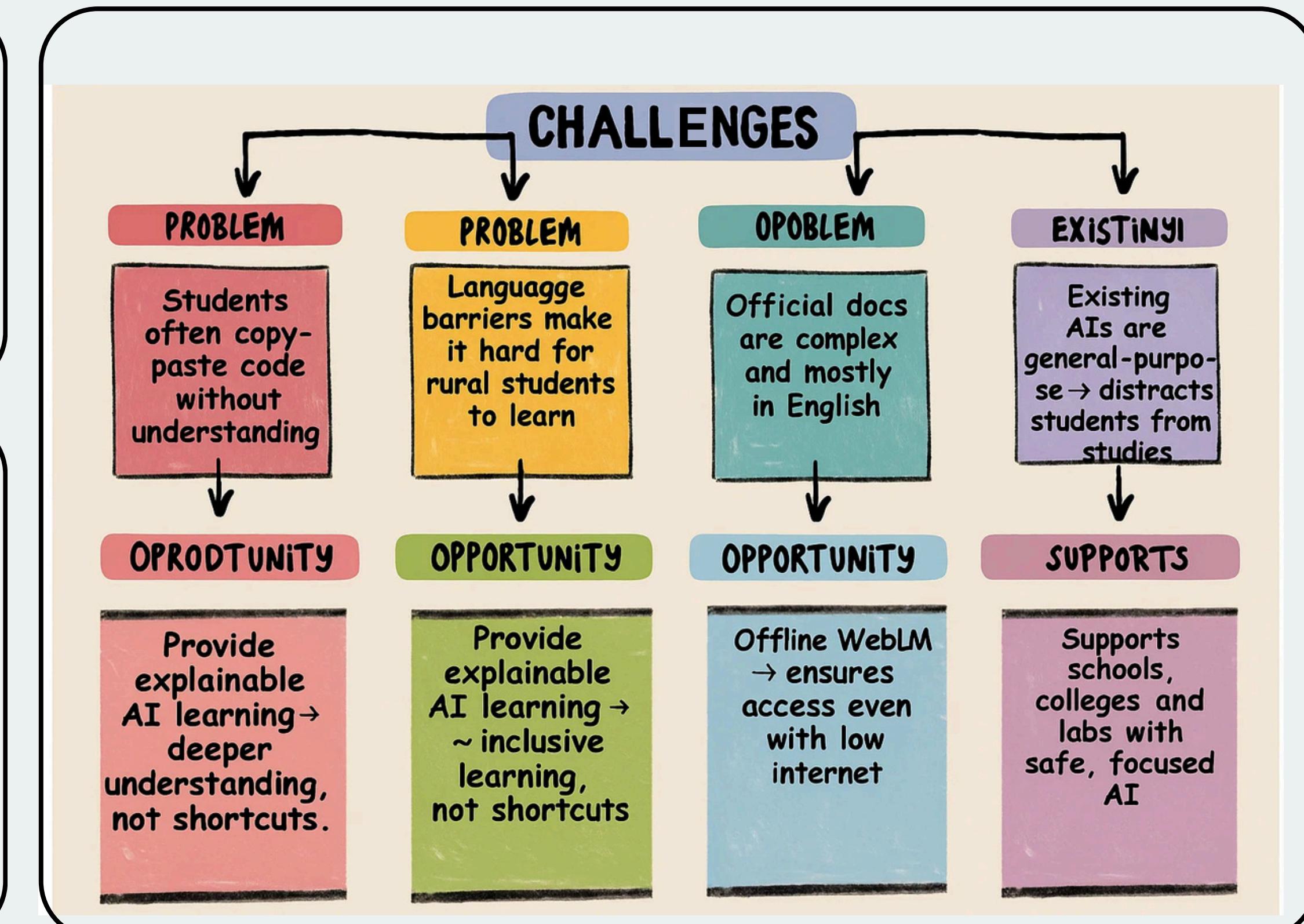
## Challenges And Opportunity

### Challenges:

- Students often copy–paste code without understanding.
- Language barriers make it hard for rural students to learn.
- Official docs are complex and mostly in English.
- Internet availability is limited in many classrooms/labs.
- Existing AIs are general-purpose → distract students from studies.

### Opportunities:

- Provide explainable AI learning → deeper understanding, not shortcuts.
- Multilingual + voice support → inclusive learning for rural areas.
- Gamification → keeps students motivated and consistent.
- Offline WebLLM → ensures access even with low internet.
- Supports schools, colleges, and labs with safe, focused AI.

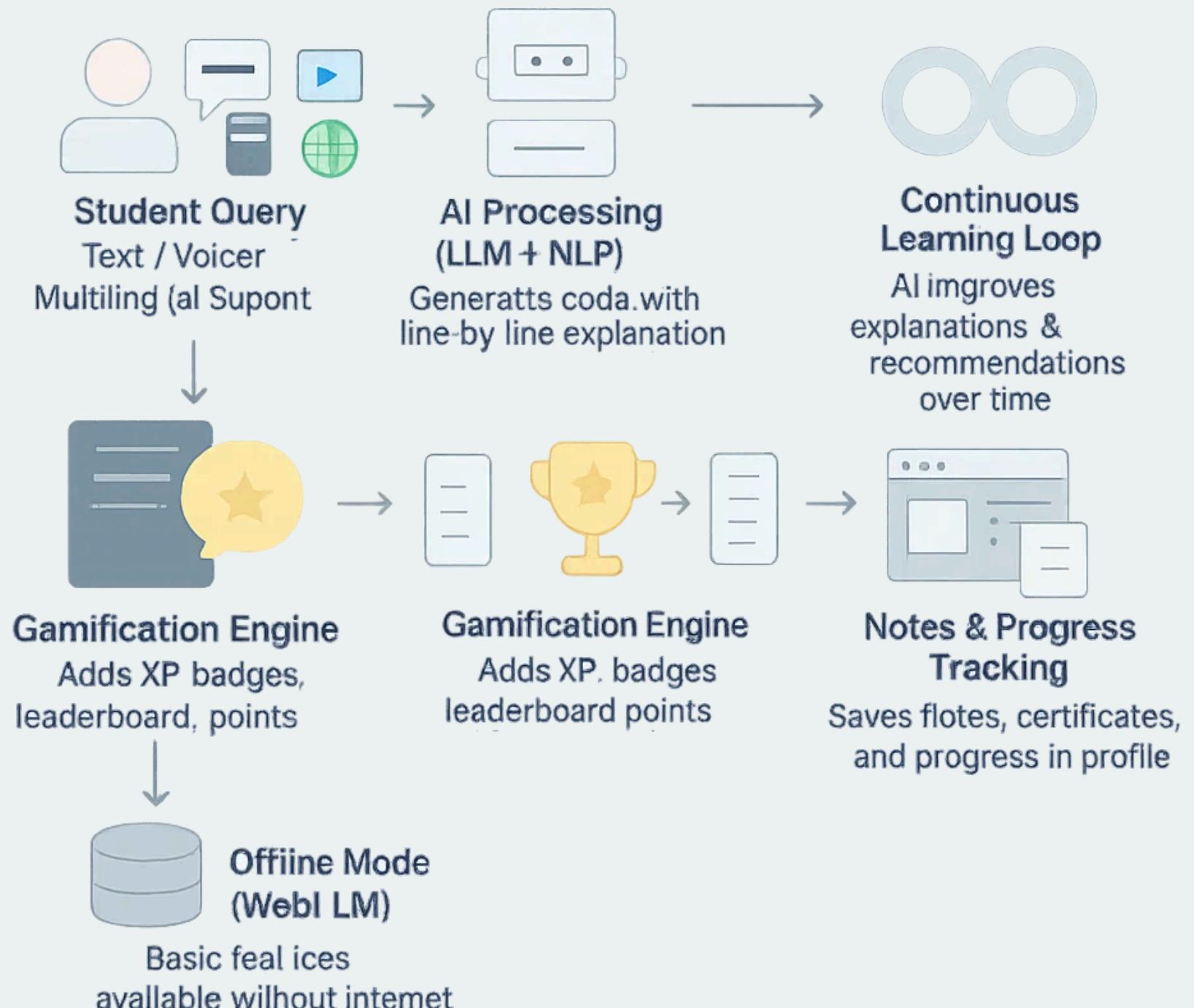


← → Q solution Proposed

## Preliminary Solution

### Preliminary Solution Concept:

- An AI-powered learning platform focused only on education.
- Provides code + line-by-line explanation for better understanding.
- Auto-links to relevant official docs (MDN, React, etc.) for clarity.
- Includes multilingual and voice support for rural students.
- Gamified learning system (XP, badges, leaderboard, certificates).
- Partial offline mode with WebLLM for use in labs and low-internet areas.



← → Q The Focus

## KEY FEATURES



Code with line-by-line explanation



Docs integration  
(MDN, React, W3, etc.)



Multilingual +  
voice support



Gamification → XP,  
badges, leaderboard,  
certificates

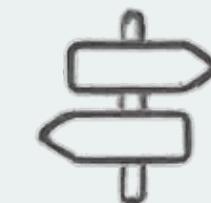


Offline mode (WebLLM  
for labs and rural areas)

## KEY FUNCTIONALITIES



24/7 assistance for  
instant coding help



Personalized learning  
paths based on progress



Text-to-speech & translation  
for accessibility



Quiz & challenge system  
to reinforce learning



Secure profile management  
for students & teachers

← → Q Technology Used

## Target Users And Expected Use Cases

### Target Users

- School and college students learning coding.
- Rural learners needing multilingual and voice support.
- Teachers and tutors for classroom and lab use.
- Educational institutions ensuring focused AI learning.
- Self-learners and beginners practicing at home.

### Expected Use Cases

- Student asks for HTML form → gets code + explanation.
- Teacher uses app in lab for guided coding practice.
- Rural student listens to code explanation in local language.
- Learner plays challenges → earns XP, badges, certificates.
- Lab with poor internet uses offline WebLLM features.



← → Q Q

## Data Requirements And Privacy Considerations

### Data Requirements:

- Basic user details (name, class, progress) for tracking.
- Learning activity data (queries, scores, notes) for personalization.
- Performance metrics (time spent, quiz results) to improve learning.

### Privacy Considerations:

- All data stored with encryption and secure servers.
- No sharing of user data with third parties.
- Option for users to control or delete their data anytime.
- Strictly focused on education-only use, no distractions.



← → Q Q

## AI Technologies and Methods

### AI Technologies and Methods

- Natural Language Processing (NLP): Understands student queries in text and voice.
- Code LLM (Language Model): Generates code with line-by-line explanations.
- Machine Learning: Personalizes learning paths based on student progress.
- Text-to-Speech & Translation: Converts explanations into voice and multiple languages.
- Gamification Engine: Tracks XP, badges, and leaderboard points.
- WebLLM (Offline AI): Provides basic features without internet connectivity.

### Tech Stack

- Frontend- Html, css, js, react js ,redux
- Backend - Node js, express ,Socket.io
- Database - MongoDB, Pinecone,redis
- Api - Gemini 2.5 flash
- Tool - postman, Figma
- Deployment - Vercel, Render

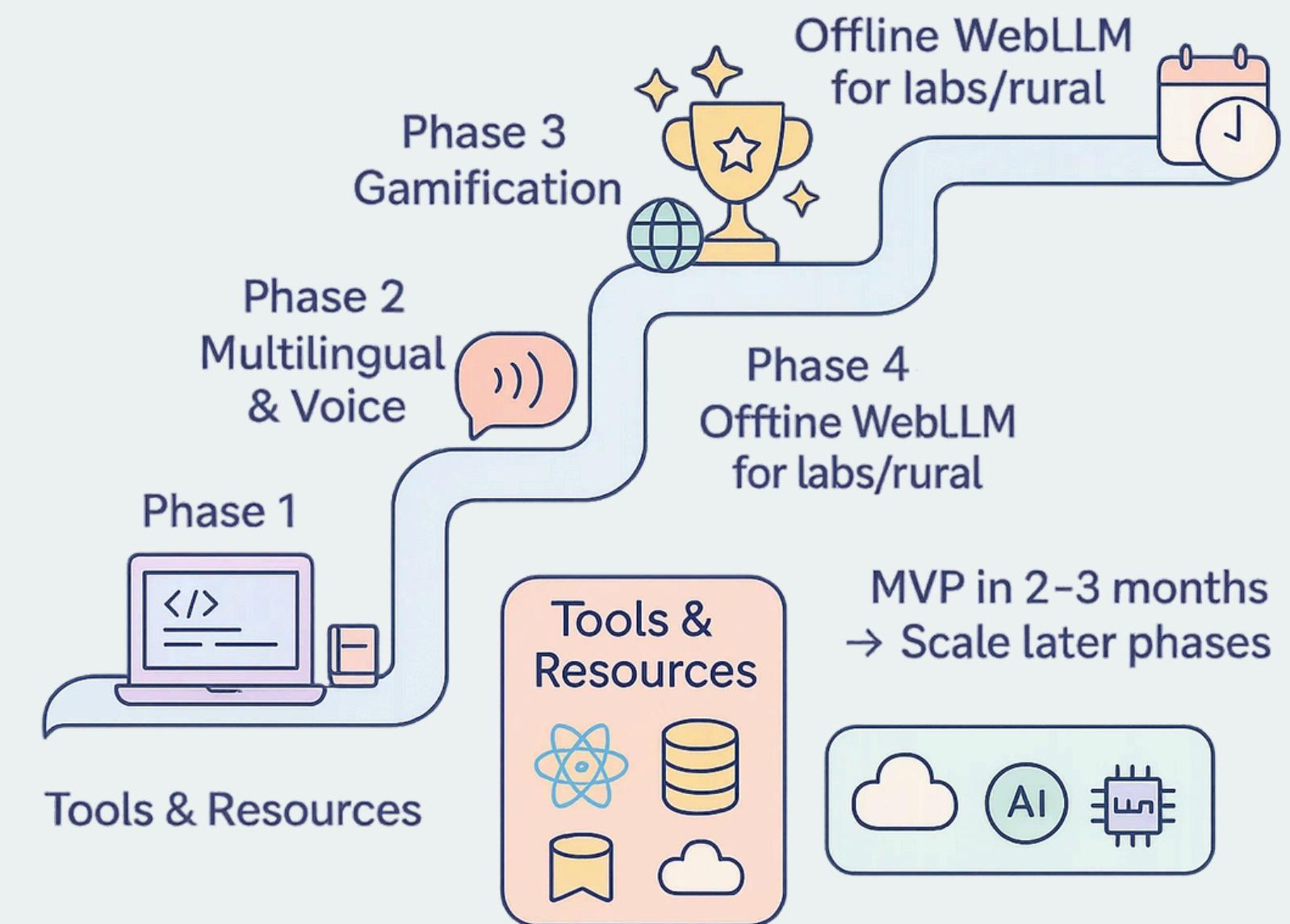


← → Q Q

## Implementation Approach;

### Implementation Approach;

- Phase 1: Build MVP with code + explanation + docs integration.
- Phase 2: Add multilingual and voice support features.
- Phase 3: Introduce gamification with XP, badges, and certificates.
- Phase 4: Enable offline WebLLM for labs and rural usage.
- Tools & Resources: React/Next.js, WebLLM, Firebase/SQL, cloud hosting.
- Timeline: Start with MVP in 2–3 months, scale features in later phases.



← → Q Q

## Evaluation Metrics

### Evaluation Metrics

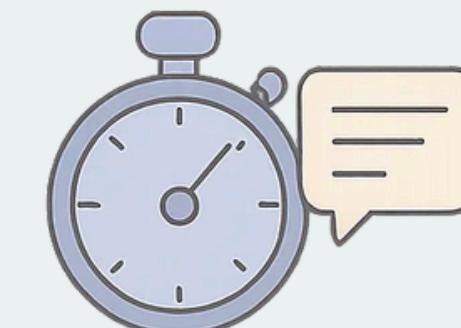
- Student Learning Improvement → measured through quiz scores and certificate completion.
- Engagement Levels → tracked via XP points, time spent, and leaderboard activity.
- Accuracy of Explanations → AI responses checked for correctness and clarity.
- Response Time → speed of AI answers to student queries.
- Accessibility Impact → number of students using multilingual and voice features.
- Adoption Rate → institutions (schools, labs, colleges) actively using the app.



Student Learning Improvement



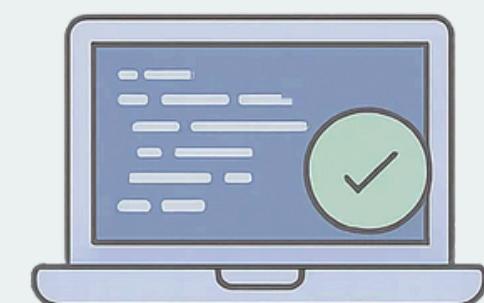
Engagement Levels



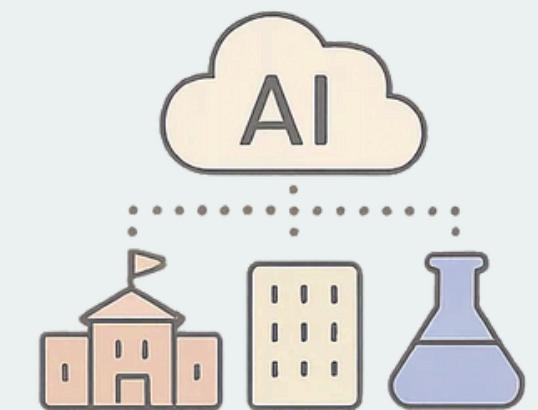
Response Time



Accessibility Impact



Accuracy of Explanations



Adoption Rate