AI for Bharat Hackathon

Powered by aws

Team Name : The Sentinels

Team Leader Name : Abhinav Khare

Problem Statement : [Student Track] AI for Learning & Developer Productivity

Brief about the Idea:

70% of Indian CS Students:

❌ Copy-paste AI code (Copilot/ChatGPT) without understanding

❌ Fail technical interviews (40% pass rate)

❌ Become "prompt engineers," not real coders

No tool prevents this during learning.

Why Current Solutions Fail?

❌ GitHub Copilot: Zero verification

❌ LeetCode: Tests after damage done

❌ Professors: Can't detect AI usage

❌ ChatGPT: No learning enforcement

Result: 1.5M students graduate unprepared

CodeGuardian Solution

Intercepts AI code → Forces explanation → Grades with AWS Bedrock → Unlocks if understood

1. Detect AI code paste

2. "Explain in your words" (text/voice)

3. AWS Bedrock scores 0-100

4. ≥70% = unlock code

5. ELI5 Guided Learning Mode
(If students fail to explain after retries, CodeGuardian teaches the concept in simple terms with examples and verifies understanding before unlocking code.)

6. Track AI dependency weekly

USP of the proposed solution

Real-Time Learning Guardrail
First tool using AWS Bedrock to verify understanding before execution

Bharat-First Design
Hindi voice support

Measurable Outcomes
AI dependency 70%→30% + 40% interview pass rate improvement

ELI5 Guided Learning Mode
If students fail to explain after retries, CodeGuardian teaches the concept in simple terms

No Shaming, Pure Gamification
Progressive unlocks build confidence, not guilt

Production-Ready Architecture
7 AWS services + <₹5/student/month on Free Tier

Unlike LeetCode (post-test) or Copilot (no verification)

List of features offered by the solution

It is always better to add a few visual representations (drawings/sketches/illustrations etc.) to your presentation, it adds to the power through Powered by 7 AWS Services:

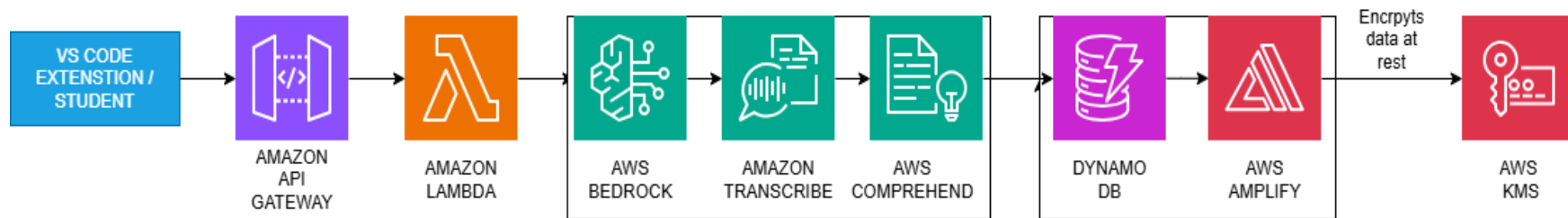Bedrock (grading) ← Lambda (orchestration) ← API Gateway ← VS Code

↓

Transcribe (Hindi voice) + Comprehend (plagiarism)

↓

DynamoDB (progress) + Amplify (dashboard) + KMS (security)

Process flow diagram or Use-case diagram

Add a flow diagram or a use case diagram or an architecture diagram.

VS CODE EXTENSTION / STUDENT → AMAZON API GATEWAY → AMAZON LAMBDA → AWS BEDROCK → AMAZON TRANSCRIBE → AWS COMPREHEND → DYNAMO DB → AWS AMPLIFY → Encrpyts data at rest → AWS KMS

Mock Example:

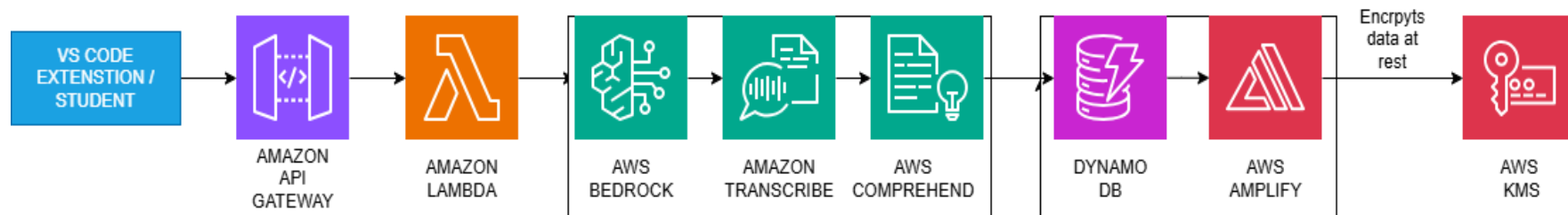VS Code Extension:

```
🔒 Code locked - Explain!

   def add(a, b):
        return a + b

[Text /Voice 🎤 ]
[Submit ] →
```

Score: 85% ✅ Code Unlocked!

Architecture diagram of the proposed solution:

Technologies to be used in the solution:

Frontend (VS Code Extension)

- TypeScript + VS Code Extension API

- React for modal UI

- Webpack build system

Backend (AWS Serverless)

- AWS Lambda (Node.js 18.x)

- Amazon API Gateway

- AWS KMS (encryption)

AI/ML Services

- Amazon Bedrock (Claude 3.5 Sonnet)

- Amazon Transcribe (Hindi/English)

- Amazon Comprehend (plagiarism)

Data & Dashboard

- Amazon DynamoDB (NoSQL)

- AWS Amplify (React dashboard)

Estimated Implementation Cost ‹ ₹5/student/month (AWS Free Tier)

Service	Monthly Cost	Free Tier Limit
Lambda	₹0.25	1M requests
Bedrock	₹1.50	25K inferences
DynamoDB	₹0.75	25 GB storage
Transcribe	₹0.80	60 min/month
Amplify	₹0.50	5 GB hosting

Total→ ₹3.80 (Scales to 100K students)