Codexia

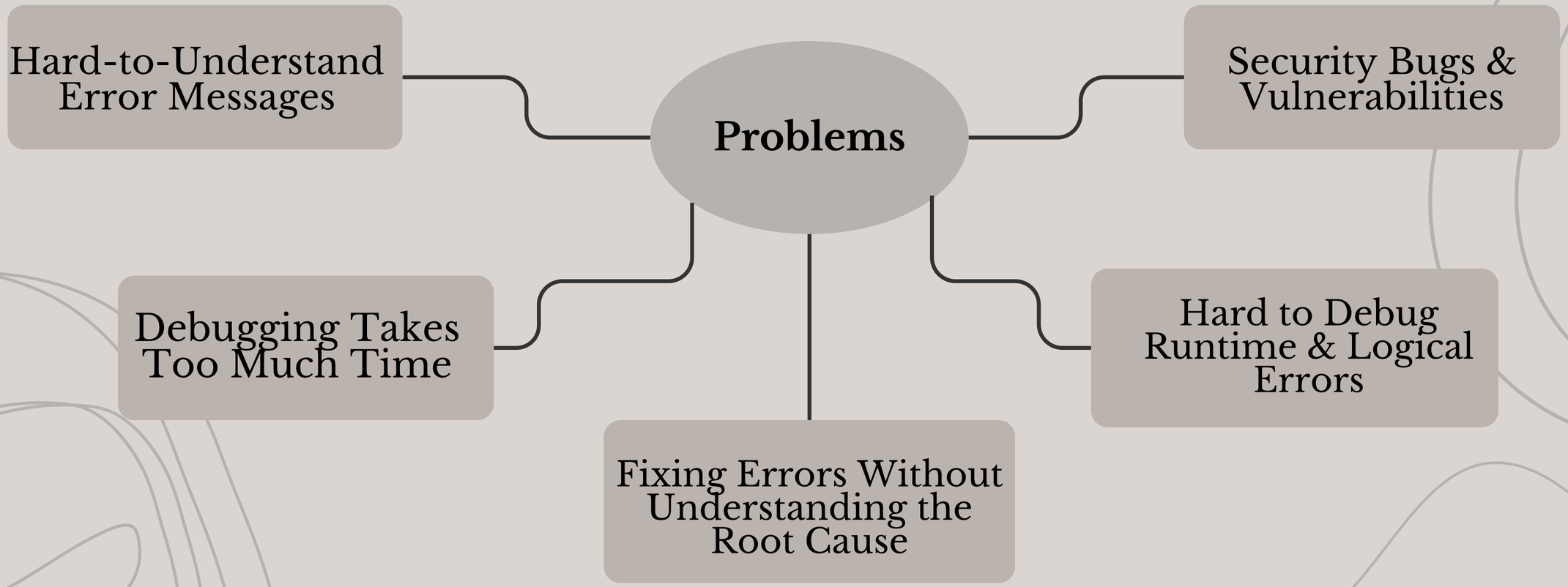
The Future of Flawless Coding

By Team Catalyst

Pranay Savdekar
Smruti Singh
Anjali Soni
Diya Golait

Our Problem Statement

AI powered real-time bug detection tool that helps developers to find and fix error in code.



🎬 The Story: "The Developer’s Nightmare"

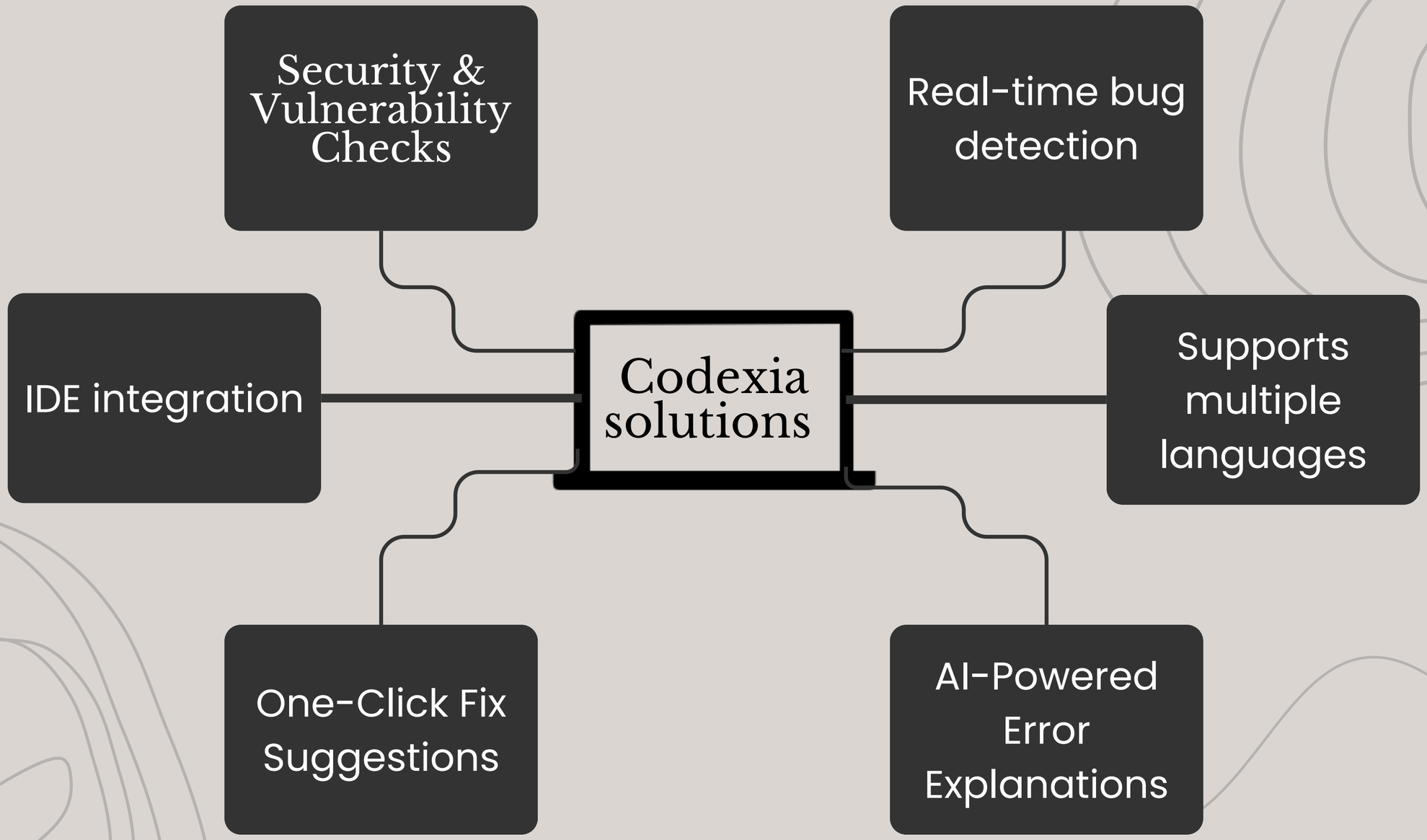
Meet Rahul, a college student passionate about coding. He spends hours debugging while learning how to code
He writes a simple Python script, but suddenly... ERROR!
The error message is cryptic: TypeError: 'NoneType' object is not subscriptable
Rahul has no clue what it means. He tries Googling it, but the solutions are confusing.
Hours later, he finally fixes it—by trial and error.

The question Rahul thought was
What if debugging was simpler? What if AI could detect, explain, and fix bugs instantly?

Existing Apps

Existing Platform	What it does	What it lacks
GitHub Copilot	1. AI-powered code autocomplete. 2. Suggests code snippets in real time. 3. Detects basic syntax errors.	1. Doesn't detect deep logic errors. 2. No runtime bug detection. 3. Can generate incorrect/insecure code
SonarLint	1. Real-time static analysis for security, reliability, and maintainability. 2. Works as a plugin inside IDEs.	1. No runtime bug detection. 2. Can slow down large projects. 3. No AI-powered auto-fixing
Google Cloud Debugger	1. AI-driven debugging for deployed cloud applications. 2. Captures snapshots of running code to detect bugs.	1. Not real-time for local development. 2. Only works for cloud apps. 3. Limited to Java, Go, Node.js, Python, C#.

Razer Wyvrn's AI QA Copilot	1. Automated AI bug testing integrated with game development engines like Unreal and Unity. 2. Enhances bug detection by up to 25% and reduces QA time by 50%. and Generates detailed QA reports for developers.	1. Primarily focused on game development; may not be suitable for other software types. 2. Relatively new; long-term effectiveness and adoption are yet to be fully assessed.
Google's Jules AI Agent	1. Experimental AI-powered agent assisting developers in fixing coding errors. 2. Integrates with GitHub workflows for Python and JavaScript tasks. 3. Aims to automate bug fixes, allowing focus on larger projects.	1. Currently in early development; prone to mistakes. 2. Does not automatically spot bugs; requires direction to pre-identified issues.
Codacy	1. Cloud based static analysis 2. Automated code reviews	1. Limited language support 2. Not Real-Time


Security & Vulnerability Checks
Real-time bug detection
IDE integration
Codexia solutions
Supports multiple languages
One-Click Fix Suggestions
AI-Powered Error Explanations

Technologies

Frontend

Next.js
TypeScript

Backend

FastAPI
Tenserflow

AI/ML Model

Hugging Face Transformers
Large Language Models
Abstract Syntax Trees parsing

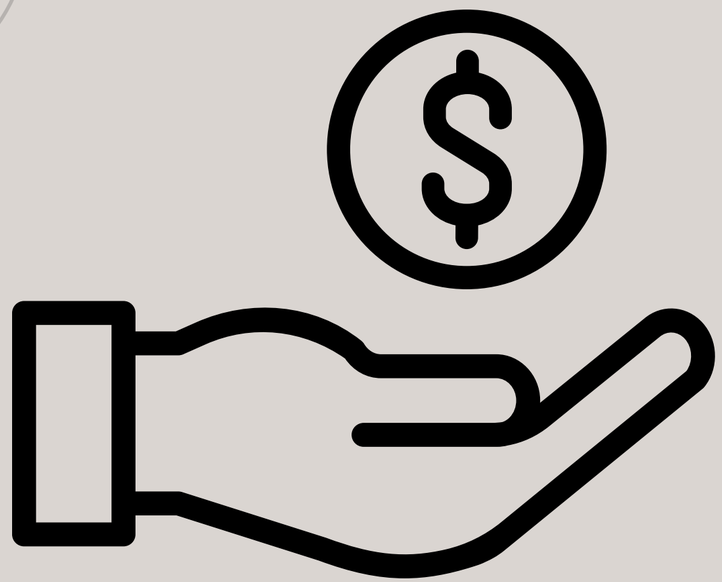
Database & Storage

MongoDB
Cloud Storage

Deployment

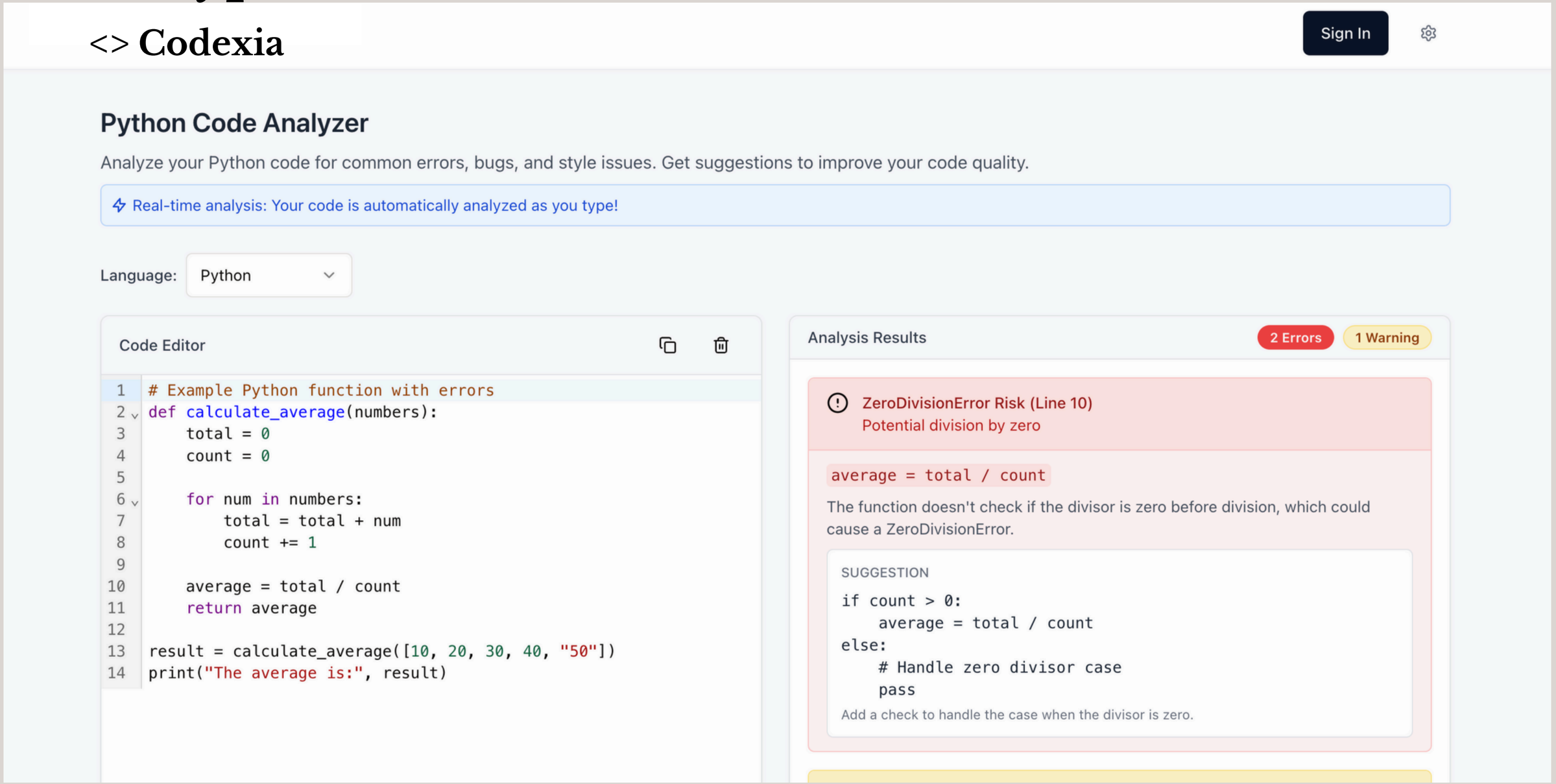
Amazon Web Services

Revenue Model



- Subscription Model
- White-Label Licensing
- IDE Plugin Sales
- For Enterprises

Prototype


<> Codexia
Sign In
Python Code Analyzer
Analyze your Python code for common errors, bugs, and style issues. Get suggestions to improve your code quality.
Real-time analysis: Your code is automatically analyzed as you type!
Language:
Python
Code Editor
1 # Example Python function with errors
2 def calculate_average(numbers):
3 total = 0
4 count = 0
5
6 for num in numbers:
7 total = total + num
8 count += 1
9
10 average = total / count
11 return average
12
13 result = calculate_average([10, 20, 30, 40, "50"])
14 print("The average is:", result)
Analysis Results
2 Errors
1 Warning
ZeroDivisionError Risk (Line 10)
Potential division by zero
average = total / count
The function doesn't check if the divisor is zero before division, which could cause a ZeroDivisionError.
SUGGESTION
if count > 0:
average = total / count
else:
# Handle zero divisor case
pass
Add a check to handle the case when the divisor is zero.

Thank You