Introduction Slide

Team : SYNAPSE

A Ramanuj Patro (Team Leader) , Pranav Pratap Singh , Sudhanshu Singh

Ramanuj (Backend) , Pranav (data handling) , Sudhanshu (frontend) .
We are ready to tackle problem all coders face , and give them a less tension free code time.

Project Overview

AI-Powered Code Refactoring Assistant

Developers spend a significant amount of time maintaining and refactoring code. Existing tools provide generic recommendations and fail to adapt to project-specific coding styles.
BUT

Now They can easily see whats happening with there code and take a quick action.

Key Features

- Learning coding patterns from existing repositories
- Detection of code smells and anti-patterns
- Context-aware refactoring suggestions
- Seamless IDE integration
- Feedback-driven improvement loop

Target Users

Individual developers, software teams, open-source maintainers

Problem Statement

Build an AI-powered refactoring assistant that learns from a codebase and delivers personalized, context-aware improvement suggestions.

Proposed Solution

Synapse provides a safe, data-driven environment for modernization.

1. AI-Driven Refactoring: Powered by **Google Gemini 2.0, it understands context, variable flow, and architectural patterns across **any language (Python, JS, Java, C++).

2. *Quantifiable Metrics*: We don't just say "it's better." We prove it.

   - 📉 *Complexity Reduction*: Measures Cyclomatic Complexity before and after.

   - ⚡ *Performance Impact*: Predicts Big O changes (e.g., O(n) -> O(1)).

   - 🛡️ *Risk Score*: Estimates the probability of regression bugs.

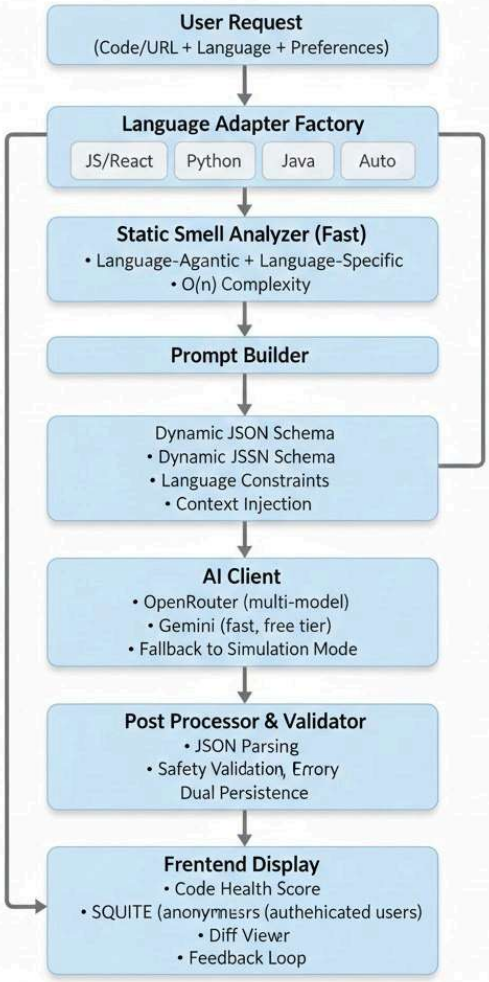
Technical Overview & Architecture

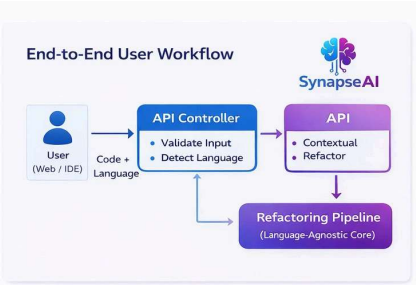
1. The "Brain" (AI Layer)
- *Model*: Google Gemini 2.0 Flash (via @google/generative-ai).
- *Why?*: Chosen for its massive context window and speed, allowing it to analyze full files, not just snippets.

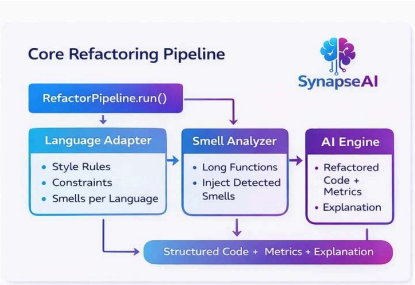
2. The "Safety Net" (Parsing Layer)
- *Engine*: @babel/parser (for JS/TS) & Regex Heuristics (for Python/Others).
- *Function: Validates that code structure is sound *before and after AI processing, preventing "hallucinated" syntax errors.

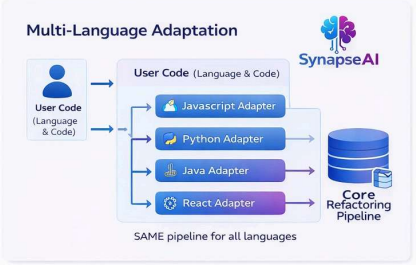
3. The "Experience" (Frontend)
- *Stack*: React, Vite, Framer Motion.
- *Visuals: A specialized **Diff Viewer* allows developers to review changes line-by-line using a side-by-side comparison, popularized by tools like GitHub.
- *HUD*: A "Head-Up Display" for engineering metrics (Risk, Complexity, Lines Saved)

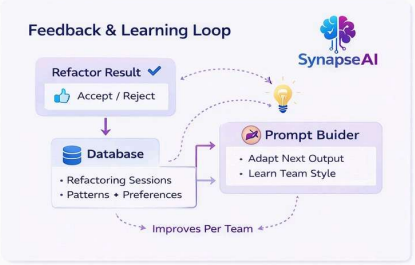
- *Frontend*: React 18, Vite, Framer Motion, PrismJS, React Diff Viewer.
- *Backend*: Node.js, Express, SQLite (for history tracking).
- *AI*: Google Gemini SDK.
- *DevOps*: Docker ready, Strict Linting.

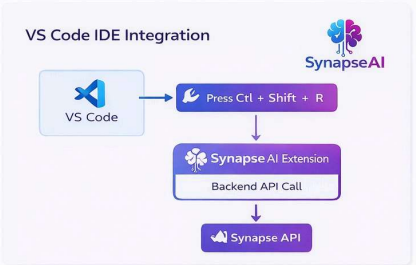

User Request
(Code/URL + Language + Preferences)
Language Adapter Factory
JS/React
Python
Java
Auto
Static Smell Analyzer (Fast)
• Language-Agantic + Language-Specific
• O(n) Complexity
Prompt Builder
Dynamic JSON Schema
• Dynamic JSSN Schema
• Language Constraints
• Context Injection
AI Client
• OpenRouter (multi-model)
• Gemini (fast, free tier)
• Fallback to Simulation Mode
Post Processor & Validator
• JSON Parsing
• Safety Validation, Errory
Dual Persistence
Frentend Display
• Code Health Score
• SQUITE (anonymnesrs (authehicated users)
• Diff Viewer
• Feedback Loop

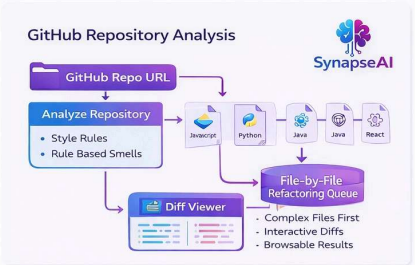

End-to-End User Workflow
SynapseAI
User
(Web / IDE)
Code +
Language
API Controller
• Validate Input
• Detect Language
API
• Contextual
• Refactor
Refactoring Pipeline
(Language-Agnostic Core)

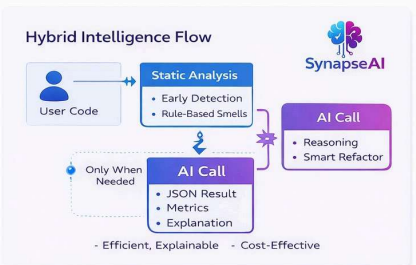

Core Refactoring Pipeline
SynapseAI
RefactorPipeline.run()
Language Adapter
• Style Rules
• Constraints
• Smells per Language
Smell Analyzer
• Long Functions
• Inject Detected Smells
AI Engine
• Refactored Code + Metrics
• Explanation
Structured Code + Metrics + Explanation

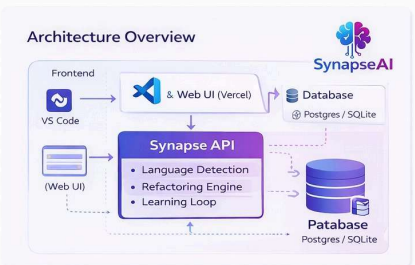

Multi-Language Adaptation
SynapseAI
User Code
(Language
& Code)
User Code (Language & Code)
Javascript Adapter
Python Adapter
Java Adapter
React Adapter
Core
Refactoring
Pipeline
SAME pipeline for all languages


Feedback & Learning Loop
SynapseAI
Refactor Result
Accept / Reject
Database
• Refactoring Sessions
• Patterns • Preferences
Prompt Buider
• Adapt Next Output
• Learn Team Style
Improves Per Team


VS Code IDE Integration
SynapseAI
VS Code
Press Ctrl + Shift + R
Synapse AI Extension
Backend API Call
Synapse API


GitHub Repository Analysis
SynapseAI
GitHub Repo URL
Analyze Repository
• Style Rules
• Rule Based Smells
Javascript
Python
Java
Java
React
File-by-File
Refactoring Queue
Diff Viewer
- Complex Files First
- Interactive Diffs
- Browsable Results


Hybrid Intelligence Flow
SynapseAI
User Code
Static Analysis
• Early Detection
• Rule-Based Smells
AI Call
• Reasoning
• Smart Refactor
Only When
Needed
AI Call
• JSON Result
• Metrics
• Explanation
- Efficient, Explainable
- Cost-Effective

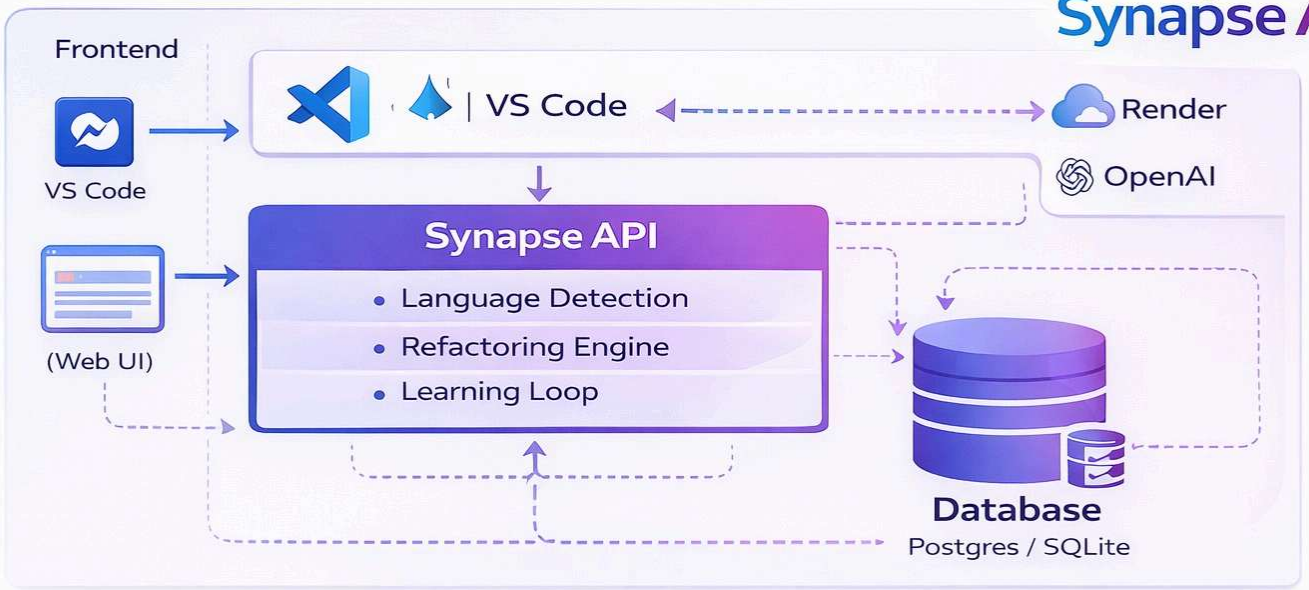

Architecture Overview
SynapseAI
Frontend
VS Code
& Web UI (Vercel)
Database
Postgres / SQLite
(Web UI)
Synapse API
• Language Detection
• Refactoring Engine
• Learning Loop
Patabase
Postgres / SQLite

React Kolkata

ELITE CODERS
WINTER OF CODE

Architecture Overview

Synapse AI



How is your project unique?

We are not just providing error and correction to a particular files of code.

We are also encouraging that there coding reasoning and concepts that gets better every time to make them more pro at there work .

This is not just copy pasting from Chatgpt , We tell them where they have done mistakes , which concepts to revise and remember and save them for future use . Tell them the complexity , risk score , How much the code improved (comparison).

Overall Giving them Insight of What have to done to make it more efficient .

Real World Impact

- Coders can code efficiently .
- Time saved which they can put it on other important project .
- Improving Coding reasoning which will eventually help in making Good features and Projects .
- People find it easy to get all the past error history and learnings to upskill themselves.
- Encouraging more Language learning .
- Doesn’t need to go here and there for Projects .

- SUSTAINABLE ON A LARGE SCALE: Over 47 millions Worldwide software developers which make up to a huge population

UI / Design / Prototype Images or Screenshots