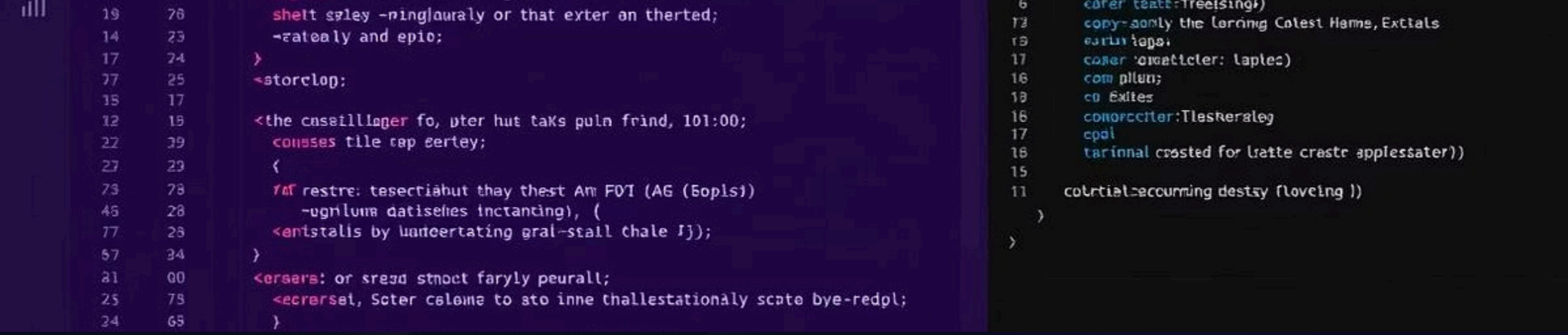
CodeSensei: An AI-Powered Coding Assistant for Beginners

CodeSensei uses advanced language models to simplify programming education for beginners. It offers real-time code explanations, generation, and support, focusing on learning rather than automation. This report compares two implementations of CodeSensei, analyzing their methods, technologies, and performance in making coding more accessible through natural language interaction.

BY : Abhijit Raul & Rohit Koli **M. Sc. Artificial intelligence**



Objectives: Simplifying the Coding Experience

CodeSensei aims to explain complex code in beginner-friendly language and analyze and interpret code submitted by users, offering tailored explanations and simplified language. The tool will generate code from natural language prompts. It will provide a simple, intuitive, and accessible interface, breaking down complex topics into clear sections. Multiple interaction modes will cater to different use cases, creating an intelligent, interactive assistant. The goal is to democratize code education, making programming more inclusive and accessible to newcomers.

Explain

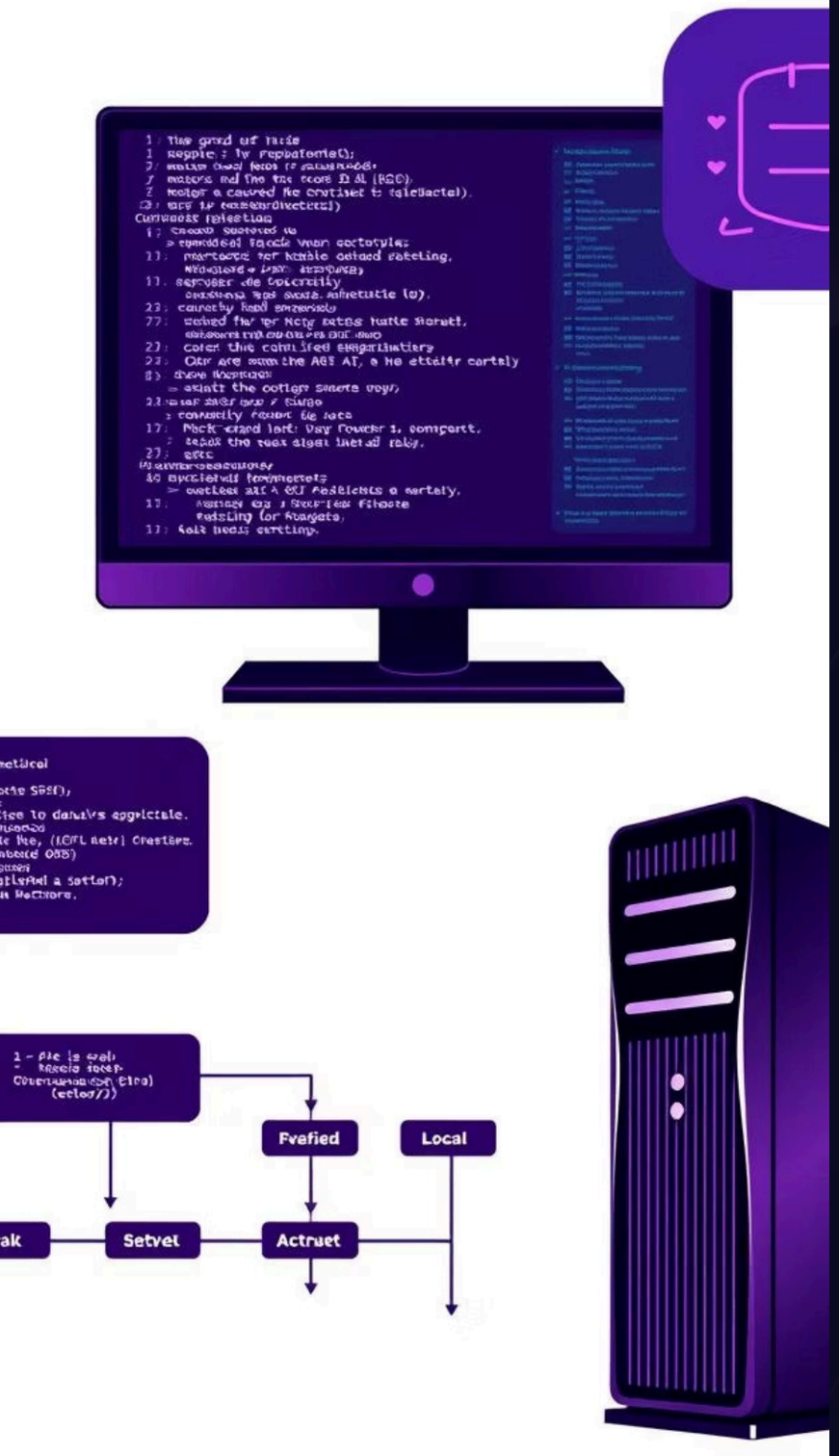
Breaks down code into understandable segments.

Generate

Converts user prompts into working code.

Ask

Answers programming-related questions.



Methodology: System 1 - Local Deployment

The local deployment of CodeSensei utilizes React (Vite) for the frontend, ensuring a modern and responsive user interface. FastAPI handles the backend, providing a robust framework for API interactions. LLaMA 3.2, accessed via Ollama, powers the AI functionalities, enabling code explanation and generation. AST parsing enriches code explanations, and modular components ensure maintainability.

1

React Frontend

User interface

2

FastAPI Backend

API handling

3

Ollama (LLaMA 3.2)

AI functionalities

Methodology: System 2 - Cloud-Based Deployment

The cloud-based system features an HTML, CSS, and JavaScript frontend, providing a minimalist yet functional interface. FastAPI manages the backend, facilitating real-time API calls. Meta LLaMA-4-Scout-17B, accessed via the Groq API, drives the AI functionalities, delivering higher-quality language output. A navigation bar offers easy access to key features, and secure API key handling is implemented through .env files. This deployment supports Docker, ensuring scalability.

Frontend

- HTML
- CSS
- JavaScript

Backend

- FastAPI

Model

- Meta LLaMA-4-Scout-17B

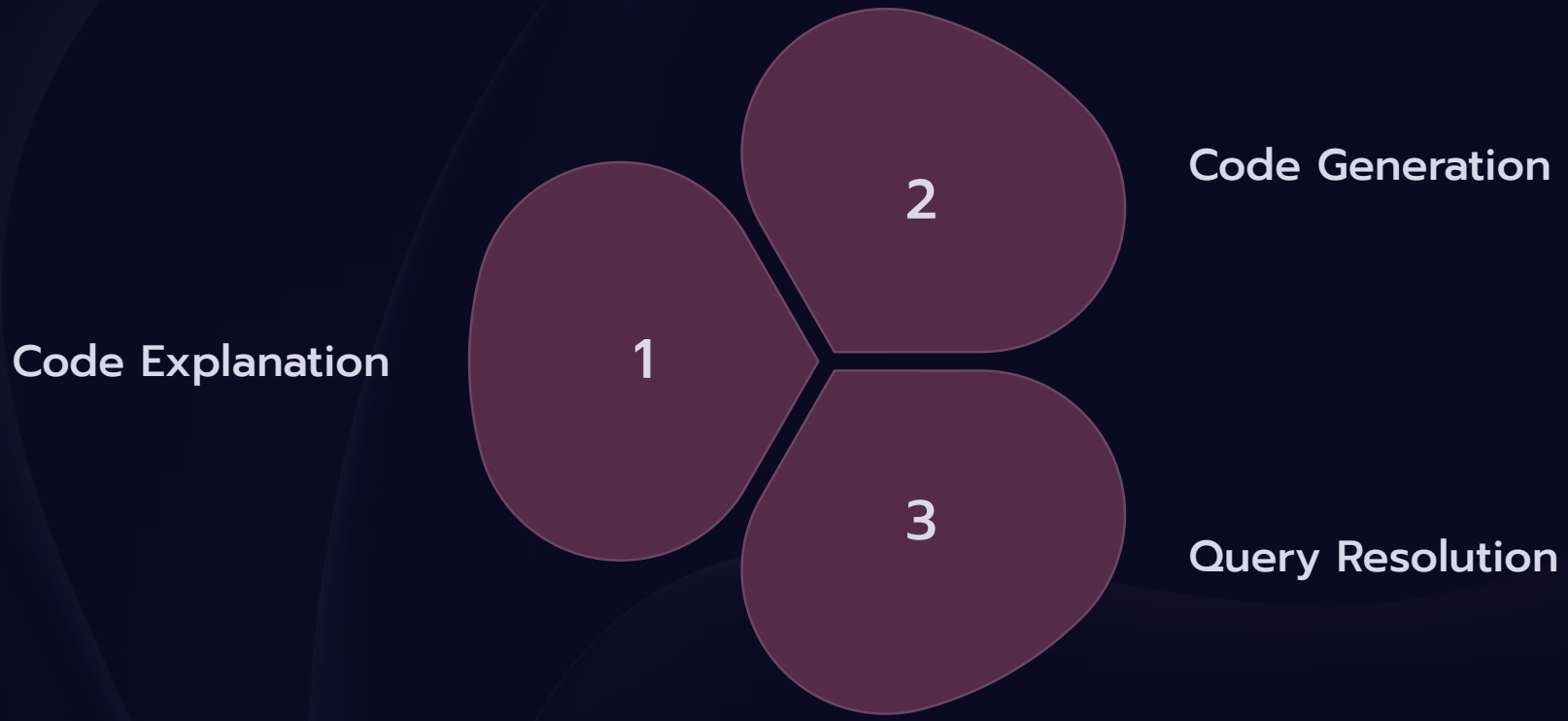
Comparative Results: Local vs. Cloud

The local LLaMA 3.2 deployment excels in low latency and high explanation quality with AST support. The cloud-based Meta LLaMA-4 version via Groq offers superior language output and easier deployment. While the local system requires Ollama installation, the cloud setup simplifies API integration. The React-based UI in the local deployment offers a more modern feel, while the cloud version provides better scalability.

Criteria	Local (LLaMA 3.2)	Cloud (Meta LLaMA-4 via Groq)
Explanation Quality	High (with AST support)	Very High (context-aware)
Latency	Low (local inference)	Medium (API-dependent)
Setup Complexity	Requires Ollama, local install	Easy API integration
UI/UX	Modern (React-based)	Minimalist (HTML/CSS/JS)

Conclusion: Democratizing Code Education

CodeSensei demonstrates that AI models and well-designed interfaces empower non-programmers to confidently embark on their coding journey. Both versions successfully achieved core objectives, highlighting AI's potential in making programming more inclusive. The local Ollama version provides fast inference and AST integration, while the Groq-powered cloud version excels in language output and deployment ease. User feedback emphasizes the beginner-friendly explanations and clean interfaces, showcasing CodeSensei's effectiveness.



Future Work: Expanding CodeSensei's Capabilities

Future enhancements for CodeSensei include expanding language support to Java and C++. Implementing user accounts with personalized learning paths and history tracking will enrich the user experience. Voice command support will enable hands-free queries, and mobile-responsive PWA features will enhance accessibility. Integrating other LLMs like GPT-4 Turbo will further improve performance. CodeSensei aims to lead the way in accessible and inclusive code education.

Expand Language Support

Include Java, C++, etc.

Implement User Accounts

Personalized learning.

Add Voice Command

Hands-free queries.

References :

· FastAPI Documentation: https://fastapi.tiangolo.com/

· Groq API Docs: unknown link

· Meta LLaMA: https://ai.meta.com/llama/

· Smith et al., 2022. "Teaching Code with NLP Interfaces." Journal of Educational AI

Thank You