**CodeSensei: An AI-Powered Coding Assistant for Beginners**

**Abstract**

The ability to understand and write code is a crucial skill in the modern digital landscape. However, for beginners and non-programmers, the learning curve can be steep due to complex syntax and abstract programming concepts. CodeSensei is an AI-powered coding assistant designed to simplify programming education. By leveraging state-of-the-art large language models (LLaMA 3.2 via Ollama and Meta LLaMA-4-Scout-17B via Groq API), intuitive UI designs, and multiple interaction modes, CodeSensei provides real-time code explanations, code generation, and educational support to ease users into the world of coding. This report integrates and compares two separate implementations of CodeSensei, detailing their methodologies, technologies, and performance.

**1. Introduction**

Programming is an essential skill across various industries, yet the journey of learning to code is often intimidating for newcomers. Traditional platforms are typically geared towards experienced developers and lack beginner-friendly tools. CodeSensei was developed to bridge this gap by creating an intelligent, interactive assistant that helps users understand and write code through natural language interactions.

This project explores two distinct approaches to building CodeSensei, comparing a local deployment using LLaMA 3.2 (Ollama) with a cloud-based deployment using Meta LLaMA-4-Scout-17B (Groq API). Both systems feature multi-mode support, intuitive UI, and responsive design.

**2. Literature Review**

Several AI-driven code assistants exist today, such as GitHub Copilot and Amazon CodeWhisperer. These tools target professional developers, focusing on productivity rather than education. Research has shown that tailored explanations, simplified language, and visual interfaces can significantly aid code comprehension for beginners. Recent advancements in large language models (LLMs) like OpenAI's GPT series and Meta's LLaMA have opened up new possibilities for educational coding tools. CodeSensei builds upon this foundation, emphasizing educational interaction over productivity automation.

**3. Objectives**

* Explain complex code in beginner-friendly language.
* Analyze and interpret code submitted by users.
* Generate code from natural language prompts.
* Provide a simple, intuitive, and accessible interface.
* Offer multiple interaction modes for different use cases.

**4. Methodology**

**4.1 Shared Functional Features**

|  |  |
| --- | --- |
| Mode | Description |
| Explain | Breaks down code into understandable segments and explains logic clearly. |
| Generate | Converts user prompts into working code snippets in supported languages. |
| Ask | Answers programming-related questions. |
| Auto | Provides real-time, context-aware suggestions during typing. |

**4.2 System 1: Local Deployment (Ollama + LLaMA 3.2)**

* **Frontend**: React (Vite), Pure CSS
* **Backend**: FastAPI
* **Model**: LLaMA 3.2 (via Ollama)
* **Deployment**: Localhost
* **Special Features**:
  + AST parsing for enriched code explanation
  + Modular components (App.jsx, vite.config.js)
  + Light/Dark theme toggle

**Architecture Overview:**

User → React Frontend → FastAPI Backend → Ollama (LLaMA 3.2)

**4.3 System 2: Cloud-Based (Groq API + Meta LLaMA-4-Scout-17B)**

* **Frontend**: HTML, CSS, JavaScript
* **Backend**: FastAPI
* **Model**: Meta LLaMA-4-Scout-17B via Groq API
* **Deployment**: Localhost / Docker-compatible VPS
* **Special Features**:
  + Real-time API calls to Groq
  + Navigation Bar with Upload/Dashboard
  + .env for secure API key handling

**Architecture Overview:**

User → HTML/CSS/JS Frontend → FastAPI → Groq API (LLaMA-4)

**5. Results & Discussion**

**5.1 Comparative Results**

|  |  |  |
| --- | --- | --- |
| 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) |
| Code Generation | Reliable for basics | More fluent and accurate |
| Setup Complexity | Requires Ollama, local install | Easy API integration |
| UI/UX | Modern (React-based) | Minimalist (HTML/CSS/JS) |
| Deployment Scalability | Limited (localhost) | Scalable (cloud-ready) |

**5.2 User Feedback**

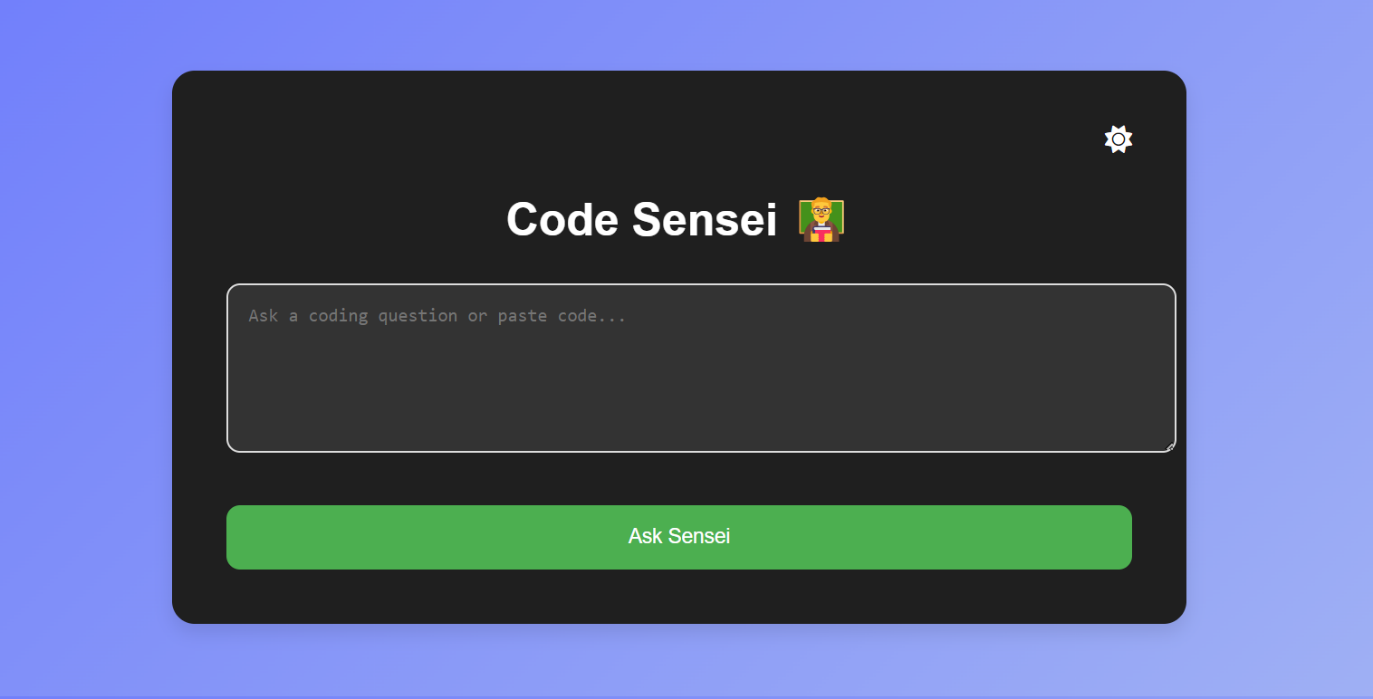
* Both versions received praise for beginner-friendly explanations.
* Users appreciated the Light/Dark mode toggle and clean interface.
* The React-based version felt slightly more responsive.

**6. Conclusion**

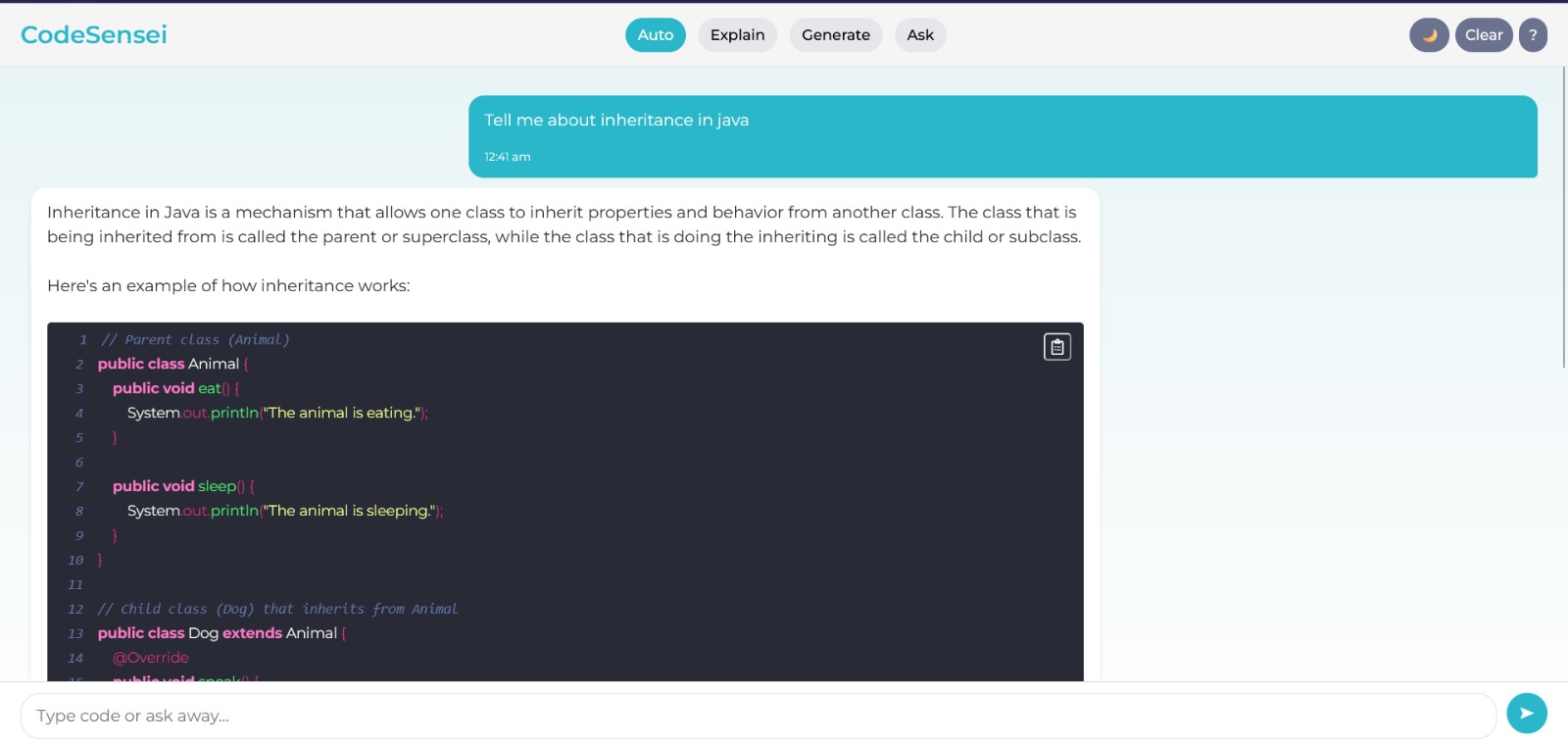
CodeSensei proves that with modern AI models and well-designed interfaces, even non-programmers can begin their coding journey with confidence. Both versions achieved the core objectives of code explanation, generation, and query resolution. While the local Ollama version benefits from fast inference and AST integration, the Groq-powered cloud version offers higher-quality language output and easier deployment.

CodeSensei highlights how AI can make programming more inclusive and accessible, marking a significant step toward democratizing code education.

* Local (LLaMA 3.2)



Cloud (Meta LLaMA-4 via Groq)



**7. Future Work**

* Expand language support to include Java, C++, and more.
* Implement user accounts with history and personalized learning paths.
* Introduce voice command support for hands-free queries.
* Add mobile-responsive Progressive Web App (PWA) features.
* Explore integration with other LLMs like GPT-4 Turbo or Mistral 7B.

**Acknowledgments**

We thank the open-source communities behind FastAPI, React, Ollama, Groq, and Meta AI for their valuable tools and documentation that made CodeSensei possible.

**One-Line Summary:** *"CodeSensei — Making Coding Simple, One Line at a Time."*

**References :**

* FastAPI Documentation: <https://fastapi.tiangolo.com/>
* Groq API Docs: <https://groq.dev/docs>
* Meta LLaMA: <https://ai.meta.com/llama/>
* Smith et al., 2022. "Teaching Code with NLP Interfaces." Journal of Educational AI