DevAssistAI: An AI Pair Programmer

Prajwal Hosahalli Dayananda, Adithya Seesanabilu Nagaraj, Jahnavi Bollineni,
Sinduja Kuna, and Nidhishree Talastha
Advisor: Dr. Zesheng Chen
Computer Science Department

PURDUE UNIVERSITY FORT WAYNE

Introduction

Software development is a complex and iterative process requiring efficient coding practices, debugging, and deployment. As AI technology advances, integrating **Large Language Models (LLMs)** into development environments presents an opportunity to streamline coding workflows. DevAssistAI is an AI-powered **pair programming** assistant designed as a Visual Studio Code (VS Code) extension that enhances developer productivity. It captures user inputs via a **chatbot** UI, integrates multiple LLMs for code generation, debugging, and execution, and automates file creation and code insertion based on LLM-generated outputs. Furthermore, DevAssistAI extends its functionality by executing terminal commands, detecting and correcting errors, and ensuring seamless interactions between AI and human developers. This project explores the integration of AI-driven **automation** in software development, addressing common challenges such as ambiguous code instructions, error handling, and deployment management. The ultimate goal is to provide an intelligent, interactive **assistant** that improves coding efficiency, reduces debugging time, and facilitates smoother project workflows.

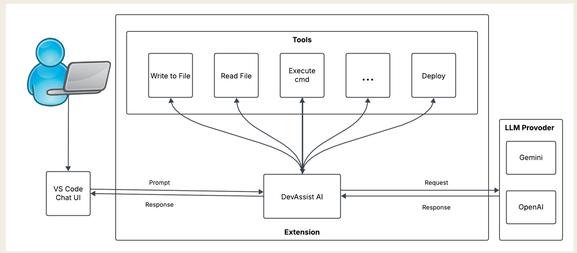
Motivation

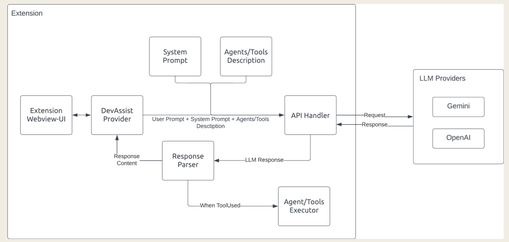
The complexity of software development has increased the demand for AI-assisted tools to enhance **productivity**. DevAssist AI automates repetitive tasks, allowing developers to focus on problem-solving. It integrates LLM-powered tools like Write to File, Execute Command, Read File, List Files, Ask Follow-Up Question, and Attempt Completion for efficient interactions. It also enables real-time debugging and cloud deployment to Google Cloud Platform (GCP), with future support for AWS, Azure, and Heroku. By leveraging LLMs like GPT, Claude, and Gemini, DevAssistAI improves code quality, error detection, and deployment automation, streamlining the software development lifecycle.

Related Work

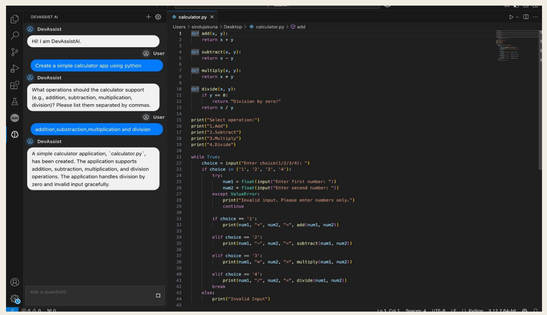
AI-driven coding assistants like GitHub Copilot, Tabnine, and Amazon CodeWhisperer help developers with code completion, debugging, and optimization. Copilot, powered by OpenAI Codex, excels in inline completions but lacks interactive debugging, command execution, and deployment—features DevAssistAI includes. Tabnine offers local processing for privacy but lacks chatbot functionality and AI-driven debugging. CodeWhisperer is AWS-focused, limiting multi-cloud flexibility. DevAssistAI integrates OpenAI API (GPT-4), Google Gemini API, and other LLMs, overcoming rate limits and task complexity issues. It enables code generation, file management, error detection, and deployment automation, enhancing real-time developer assistance.

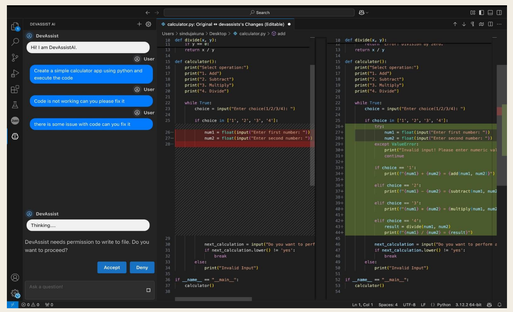
Architecture

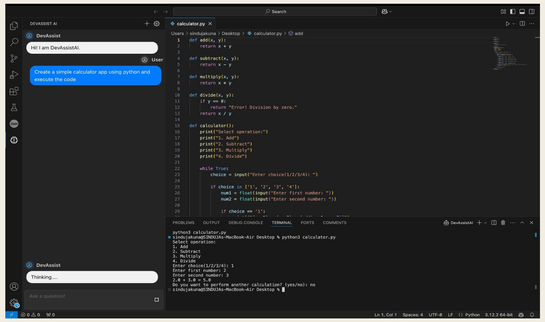


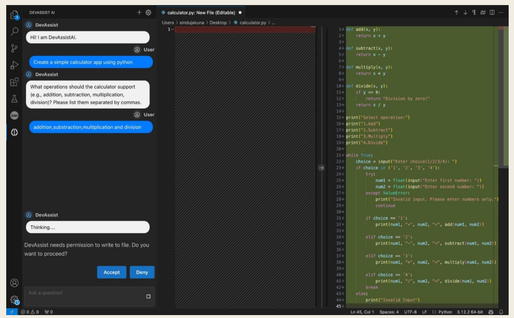


Results









Methodology

DevAssistAI was developed as a Visual Studio Code extension with an interactive chatbot interface that processes user commands in natural language. The system was implemented in multiple phases. In the initial phase, a structured prompt system was designed to guide the AI in generating accurate and actionable responses. Core functionalities such as file creation, command execution, and code generation were integrated. The second phase focused on enhancing capabilities by adding error detection, file reading, auto-completion, and cloud deployment support.

The backend system processes user inputs by mapping them to predefined actions using a large language model (LLM). DevAssistAI supports two AI-powered backends—Google Gemini and OpenAI GPT-4, allowing flexibility in response generation. Execution tools were developed to handle file operations, command-line execution, and deployment tasks while ensuring user approval for critical actions. The tool was designed to deploy applications to Google Cloud Platform (GCP) by automating configuration and authentication processes. Extensive testing was conducted to refine response accuracy, performance, and error handling. Logs and automated troubleshooting workflows were integrated to improve reliability. DevAssitAI continues to evolve with optimizations in AI response handling, user interface improvements, and extended deployment capabilities.

Conclusion

In conclusion, DevAssistAI is an innovative AI-powered VS Code extension designed to enhance developer productivity by providing real-time code suggestions, error detection, and debugging assistance. By leveraging LLMs and intelligent automation, our solution streamlines the development process, reducing time spent on debugging and improving code quality. This project aims to showcase the potential of AI in software development.

References

Google Gemini API Documentation – Google AI. [Online]. Available: https://ai.google.dev/gemini-api/docs
OpenAI GPT-4 API Documentation – OpenAI. [Online]. Available: https://platform.openai.com/docs/
Visual Studio Code API Documentation – Microsoft. [Online]. Available: https://code.visualstudio.com/api