# **Code Copilot**

## **Project Overview**

This project is a **Code Copilot** web application built using Streamlit that leverages the **Google Gemini API** to analyze and provide insights on code snippets. The application allows users to input code, and then it analyzes the code context (specifically focusing on the surrounding 5 lines before and after the cursor) and generates suggestions or insights.

## **Features**

1. **Interactive Code Analysis**:
   * Users can input their code directly into the text area provided.
   * The application intelligently splits the input into lines and analyzes a code context centered around the middle of the input.
   * The analyzed context includes 5 lines before and after the cursor's position.
2. **Google Gemini API Integration**:
   * The app sends a prompt to the Google Gemini API containing the code context and retrieves suggestions or analysis for the code.
   * The content generated by the API is displayed in the application.
3. **User-Friendly Interface**:
   * A simple and intuitive interface built with Streamlit makes the tool easy to use.
   * The application displays the API response, making it easy to debug and understand the output.
4. **Environment Variable Management**:
   * The API key is securely loaded from a .env file, ensuring sensitive information is not hardcoded into the application.
5. **Additional Features Sidebar**:
   * The sidebar offers an easy way to extend the functionality with additional features such as content analysis, documentation generation, or further code insights.

## **How It Works**

1. **Environment Setup**:
   * The application begins by loading environment variables, particularly the Google Gemini API key, from a .env file.
2. **User Input**:
   * Users enter their code into a text area in the main interface.
3. **Context Extraction**:
   * The code is split into lines, and the context for analysis is extracted. This context includes the middle line (where the cursor is assumed to be) and 5 lines before and after it.
4. **Prompt Creation & API Call**:
   * A prompt is created using the extracted code context and sent to the Google Gemini API via an HTTP POST request.
5. **Response Handling**:
   * The application processes the API's response and displays the generated suggestions or analysis back to the user.
6. **Error Handling**:
   * The application includes error handling for HTTP errors and unexpected API response formats, ensuring that users are informed of any issues.

## **Future Enhancements**

* **Advanced Code Analysis**: Extend the API calls to analyze entire code files, detect potential errors, and suggest improvements.
* **Multi-Language Support**: Enhance the copilot to support multiple programming languages.
* **Code Documentation Generation**: Add functionality to automatically generate documentation for code snippets.
* **Voice Input/Output**: Integrate voice input/output for hands-free operation.
* **User Authentication**: Add user authentication for personalized experiences.