Complete RAG-based AI Code Generator application with the following components:

1. **Main Application (src/main.py)**:

* Modern GUI using PyQt6
* Two main tabs: Code Generation and Code Analysis
* Asynchronous operations using worker threads
* Progress indication and error handling

1. **RAG System (src/rag/rag\_manager.py)**:

* Vector database using Chroma
* OpenAI integration for code generation
* Context-aware code generation
* Code storage and retrieval

1. **Code Analysis (src/code\_analysis/analyzer.py)**:

* Comprehensive code analysis using multiple tools
* Security analysis with Bandit
* Code quality checks with Pylint
* Code formatting with Black
* Structural analysis using AST

1. **Utilities (src/utils/helpers.py)**:

* File handling utilities
* JSON processing
* Code analysis result formatting
* Temporary file management

To get started with the application:

1. Create a virtual environment and install dependencies:

bash

Apply to requirements...

Run

   python -m venv venv

   venv\Scripts\activate  # On Windows

   pip install -r requirements.txt

1. Create a .env file with your OpenAI API key (as described in the README)
2. Run the application:

bash

Apply to requirements...

Run

   python src/main.py

The application provides a user-friendly interface for:

* Generating code based on natural language requirements
* Analyzing existing code for vulnerabilities and improvements
* Getting detailed code analysis reports
* Storing and retrieving code snippets using RAG