📝 代码文档生成器

🟡 难度等级：中级

⏳ 开发周期：4–5 天
🎯 课程主题：Transformers、BERT/GPT、模型微调

🧠 问题陈述

创建一个智能系统，能够自动为代码仓库生成全面的文档，文档应包含以下内容：

- 函数说明
- 参数解释
- 使用示例

该系统需利用微调后的语言模型来理解代码并进行自然语言生成。

✅ 关键学习成果

- 学习如何将 Transformer 应用于代码相关任务
- 微调 CodeBERT / GPT 等模型处理代码数据
- 理解代码的语义与结构
- 生成清晰、结构化且实用的文档

📦 可用数据集

- **CodeSearchNet**：600 万个带文档的函数，涵盖 6 种编程语言
- **The Stack**：支持 300+ 种语言的大规模开源代码数据集
- **GitHub Repositories**：包含高质量文档的精选仓库（如 awesome-python）
- **CodeBERT Datasets**：用于代码理解与生成的预训练数据

🛠️ 技术栈

- **Python**
- **Transformers**（CodeBERT、GPT）
- **Hugging Face Datasets**

- **Gradio 或 Streamlit**：用于构建交互界面
- （可选）**GitHub API**：实现实时仓库代码获取

🚀 建议开发流程

1. 收集与预处理代码数据

- 使用如 CodeSearchNet 或 GitHub 仓库作为数据来源
- 解析函数、类、参数及已有的 docstring（文档字符串）

2. 准备训练数据

- 将“代码 + docstring”对转为输入输出格式
- 使用适配的分词器进行编码（如 CodeBERT 或 GPT-2 的 tokenizer）

3. 微调语言模型

- 选择一个预训练模型（如 CodeBERT 或 GPT-2）
- 在准备好的数据集上进行微调，实现代码摘要或文档生成

4. 构建生成管道

- 输入：原始代码或函数块
- 输出：生成的 docstring 或 Markdown 文档
- （可选）添加提示词控制输出结构

5. 构建用户界面

- 使用 Streamlit 或 Gradio 让用户上传代码文件或粘贴代码片段
- 以整洁的格式展示生成的文档

6. 🌟 进阶功能（可选）

- 使用 GitHub API 获取公共仓库中的代码文件
- 自动为每个文件或模块生成文档