**Chiikawa小组总结**

**一.项目概况与小组成员**

**小组名称：**Chiikawa

**小组成员：**俞乐楠、黄浚航、徐文彬、刘子凯、赵会洋  
**项目名称：**ChiikaCode

**项目介绍：**ChiikaCode是一个智能、高效的代码辅助工具，集成代码解释、注释、自动补全等功能。通过采用检索增强生成（RAG）技术，ChiikaCode 能够从已有代码库中检索并生成高质量的代码，提供高效的代码生成、解释和优化建议，进而提升开发者的工作效率、帮助开发者编写更具可读性和可维护性的代码，减少冗余代码并避免常见的编码错误。

**二.项目背景与目标**

随着人工智能技术的发展，生成式模型已广泛应用于软件开发领域。本项目旨在为开发者提供一个智能化的代码生成助手，覆盖软件开发生命周期的各个阶段，包括需求分析、设计、编码、测试、部署和运维。通过结合检索增强生成（RAG）等技术我们构建了一个全面的工程级代码生成与智能助手系统。

本项目的主要目标是：

* 提高软件开发效率，减少开发过程中的重复性任务。
* 提供智能化的代码生成与优化，自动化部分代码编写与调试。
* 支持与现有代码库的集成，优化开发流程。

**三.开发内容**

项目的开发内容涵盖了多个方面，核心模块包括：

**1. 代码整库分析**

为支持高效的代码生成，项目首先实现了对开源代码库的深入理解。通过对代码库中的文件、函数和类进行分析，系统能够自动识别代码中的结构、模式以及潜在的功能模块，为后续的代码生成和优化提供支持。

**2. 文档分析增强**

为了支持开发者在文档中的需求挖掘与理解，项目融合了文档知识库与知识图谱，利用检索与推理增强文档分析能力。该模块能够帮助开发者从文档中快速提取关键信息，并提供文档内容的自动推荐和查询功能，从而提升开发者在开发中的效率。

**3. 代码定制化交互式生成**

本模块通过多轮对话与用户需求总结，根据现有的开源代码库（特别是机器学习与深度学习领域的开源代码）生成和优化代码。通过自然语言处理技术与深度学习模型，系统能够自动根据用户需求生成定制化的代码，并进行迭代优化。

**4. 工具库与知识库集成**

ChiikaCode支持多种工具库与知识库的集成，能够通过统一的API调用，结合不同的开源工具和框架进行操作。该模块的设计目标是确保系统能够灵活地与外部工具集成，并增强工具调用的能力。

**四.实现的功能**

ChiikaCode项目的核心功能包括：

**1.基于LLM与Embedding的离线私有部署**  
 支持开源LLM模型的离线部署，确保即使在没有网络连接的环境下也能进行高效推理与计算。

**2.统一API调用与模型接入扩展**  
 提供统一的API接口，支持接入多种开源大模型与模型API，增强系统的灵活性与兼容性。

**3.代码补全与优化建议**  
 通过自动补全与代码建议功能，系统能够根据现有代码库为开发者提供代码补充建议，并对现有代码进行优化。

**4.自动生成代码注释与修复**  
 利用自然语言处理技术，系统能够自动为代码生成注释，并修复代码中的潜在问题，帮助开发者提升代码的可读性与质量。

**5.文档分析与知识图谱增强**  
 通过文档分析和知识图谱技术，系统能够帮助开发者快速理解技术文档并在项目中应用。

**五.运行环境与技术栈**

为了确保系统的高效性和可扩展性，我们在以下技术栈上进行了开发：

* **语言：**TypeScript、Python、Javascript、json
* **大语言模型：**ollama3.2、qwen：7b、ChatGPT
* **工具库：**streamlit、langchain、uvicron
* **部署与运行环境：**Windows、WSL2（Ubuntu22.04LTS）

**六.总结**

ChiikaCode项目是基于生成式模型和检索增强生成（RAG）技术的智能代码生成系统，旨在提高开发效率、提升代码质量并优化开发者体验。通过对代码库的深度分析、智能化生成与优化，ChiikaCode成功实现了多项功能，包括代码补全、代码修复、注释生成、文档分析等。项目的实施不仅展示了生成式模型在软件开发中的应用潜力，也为开发者提供了一个高效的工具，帮助他们更好地完成开发任务。