

# LLMエンジニア コーディング課題：RAGシステムの実装

## 課題概要

日本語RAG評価データセットを知識源とした、実践的なRAGシステムの構築を行っていただきます。本課題では、基本的なRAGシステムから発展的なAgenticRAGまでの実装を通じて、LLMエンジニアとしての総合的な技術力を評価いたします。

## 評価の目的・観点

本課題では以下の能力を総合的に評価いたします。

- RAGシステムの設計・実装能力
- LLMおよびRAGにおける分析能力
- Python及びオブジェクト指向プログラミングの理解度
- コンテナ・インフラ構築の基礎知識
- ドキュメント作成能力

## 実装要件

### 必須要件

#### 1. データセットと言語

- 対象データセット: [RAG-Evaluation-Dataset-JA](#)
- バックエンド言語: Python (メイン)
- フロントエンド言語: JavaScript / TypeScript (任意のフレームワーク可) / Pythonの軽量UIフレームワーク

#### 2. 開発環境・インフラ

- パッケージ管理: uv
- 実行環境: Docker + Docker Compose
- アーキテクチャ: マルチコンテナ構成
  - UIサーバー
  - アプリケーションサーバー
  - データベースサーバー

#### 3. RAGシステム実装

- ベースライン: ナイーブRAG (単純なテキスト分割、ベクトル検索のみ)
- 発展版: AgenticRAG (AgenticRAGの定義は世の中的にも確立されていないため、調査した上で独自に定義・実装してください)
- LLMモデル: OpenAI API (詳細は下記参照)

#### 4. コード品質・管理

- バージョン管理: Git (コミット履歴を含む.gitディレクトリを提出)

- **設計原則:** オブジェクト指向プログラミング

## 5. ドキュメント作成

- システム仕様書
- 環境構築手順書
- 実行手順書
- 精度評価レポート

## 提出物・提出方法

提出ファイル (zip形式)

- **ソースコード式:** 全プロジェクトファイル
- **Gitリポジトリ:** .gitディレクトリを含む (コミット履歴確認のため)
- **ドキュメント式:** 下記4点の文書
  - システム仕様書
  - 環境構築手順書
  - 実行手順書
  - 精度評価レポート

## 提出期限

課題送付日から **3週間以内** にご提出ください。

## 評価観点

- **実装品質:** コードの設計・可読性・保守性
- **システム設計:** アーキテクチャの妥当性
- **分析力:** LLMおよびRAGにおける分析能力
- **ドキュメント品質:** 理解しやすい説明・適切な粒度

## 注意事項・ガイドライン

### 外部リソースの利用について

- **既存コード:** 公開コードの利用可 (出典を明記)
- **LLM活用:** 使用可 (活用方法をドキュメントに記載)
- **質問対応:** 不明点は遠慮なくエージェント経由でお問い合わせください

### 実装のスコープについて

完璧な実装よりも、課題に取り組む姿勢や実装・分析時の考え方を重視します。

不完全な部分があっても、その内容をドキュメントに明記した上で提出してください。

### OpenAI API 利用について

#### API Key提供

- **OpenAI API Key:** 課題開始時にメールで提供いたします。

- **利用制限:** \$30相当のクレジットが課題期間中に利用可能です。追加で必要な場合は別途ご連絡ください。

## 利用可能モデル

以下のモデルのみご利用いただけます：

- **テキスト生成:** gpt-5-mini
- **埋め込み生成:** text-embedding-3-small

## 利用上の注意

- 提供されたAPI Keyは課題期間中のみ有効です
- 課題に関連しない用途での使用はご遠慮ください