Cloud Computation Project - Phase 6

Project Name: *FoodRecommendation Cloud API*

New recommendation use case:
Proposed by 58235

We propose implementing a **recipe recommendation feature** that allows users to receive suggestions for similar recipes based on a recipe they input, using machine learning algorithms to do so. This adds value to the application by helping users discover new recipes they might like.

Implementation steps:

- A new API endpoint (`/recommend`) will be added.
- Users can send a recipe title and ingredients.
- The system will return a list of similar recipes from the database, with some added randomness for variety.
- Each recommendation will include the full recipe (ingredients, directions, etc.).

How:

- The system will compare recipes based on their titles and main ingredients.
- It will look for similarities using a machine learning algorithm and return the most relevant matches.
- To make results more interesting, some randomness will be added.

Planned Improvements:

- This feature will be deployed as a new service in our cloud environment.
- It will be integrated with our existing backend and BigQuery recipe dataset.
- Future improvements could include user personalization or smarter suggestions using AI.

Why It Matters:

- Helps users quickly find related recipes they might enjoy.
- Makes the app smarter and more interactive.
- Adds a real use case that connects different parts of the system (data + logic + API).

New recommendation architectural implementation:
Proposed by 58235

To make our application more scalable and easier to maintain, we are planning to separate it into multiple independent services, unsimilar from the previous phases. Each service (such as searching and recommending recipes) will be placed in its own container. An API Gateway will handle external requests and route them to the correct service.

What Will Be Done:

- Each core service (e.g., search, recommend, health check) will be **containerized** using Docker.
- A **central API Gateway** will be created to expose a single entry point to users.
- The gateway will forward requests to the correct internal service.
- Each service will continue to interact with the database (BigQuery) independently.

Planned Improvements:

- **Modularization**: Services are now independent, making the system easier to test and update.
- **Scalability**: Each service can scale individually depending on usage.
- **Maintainability**: Easier to isolate and debug issues in one part of the system without affecting others.
- **Cloud Readiness**: This approach aligns with cloud-native best practices and prepares us for deployment on GKE.

Architecture Overview

The new structure includes:

- **API Gateway** (single entry point)
- **Search Service** (for exact ingredient search)
- **Recommendation Service** (for recipe similarity)
- **Health Check Route** (for monitoring)