mealmateAI

Introduction

MealMateAI is a microservices-based meal planning application designed to provide personalized recipe recommendations and weekly meal plans. It integrates modern web technologies with Retrieval-Augmented Generation (RAG) to ensure that generated plans are accurate, diverse, and tailored to user preferences. By combining a large recipe dataset with semantic search, classification, and AI-powered summaries, MealMateAI delivers meal plans that align with dietary needs and cooking styles.

This report describes the overall system, including its architecture, main components, and implementation details. It describes the AI techniques applied to recipe recommendations, and possible directions for extending the system in the future.

System Overview

MealMateAI is an AI-powered meal planning and recipe recommendation system built with a modern microservices architecture. The system is designed to be scalable, efficient, and user-friendly, making it suitable for both individual users and enterprise deployment in the food, nutrition, and wellness domains.

The platform integrates advanced AI/ML technologies with robust backend services to deliver personalized meal plans, recipe discovery, and automated grocery list generation. It ensures modularity and maintainability through independently deployed microservices and leverages containerization for scalability.

Core Components

- **Frontend**: Built with **React 18, TypeScript, and Material-UI**, providing an intuitive and responsive user interface. Features include recipe search, drag-and-drop meal planning, user preference management, and automated grocery list generation.

- **Backend**: Implemented with **FastAPI microservices** in Python. Each service is independently deployable and responsible for a dedicated business function such as user management, recipe processing and meal planning.,

- **API Gateway**: Developed with **Express.js**, it manages authentication, request routing, rate limiting, and monitoring while exposing a unified entry point for clients.

- **Databases**:

    - **MySQL** for user authentication and preferences

    - **PostgreSQL** for recipe and meal plan storage

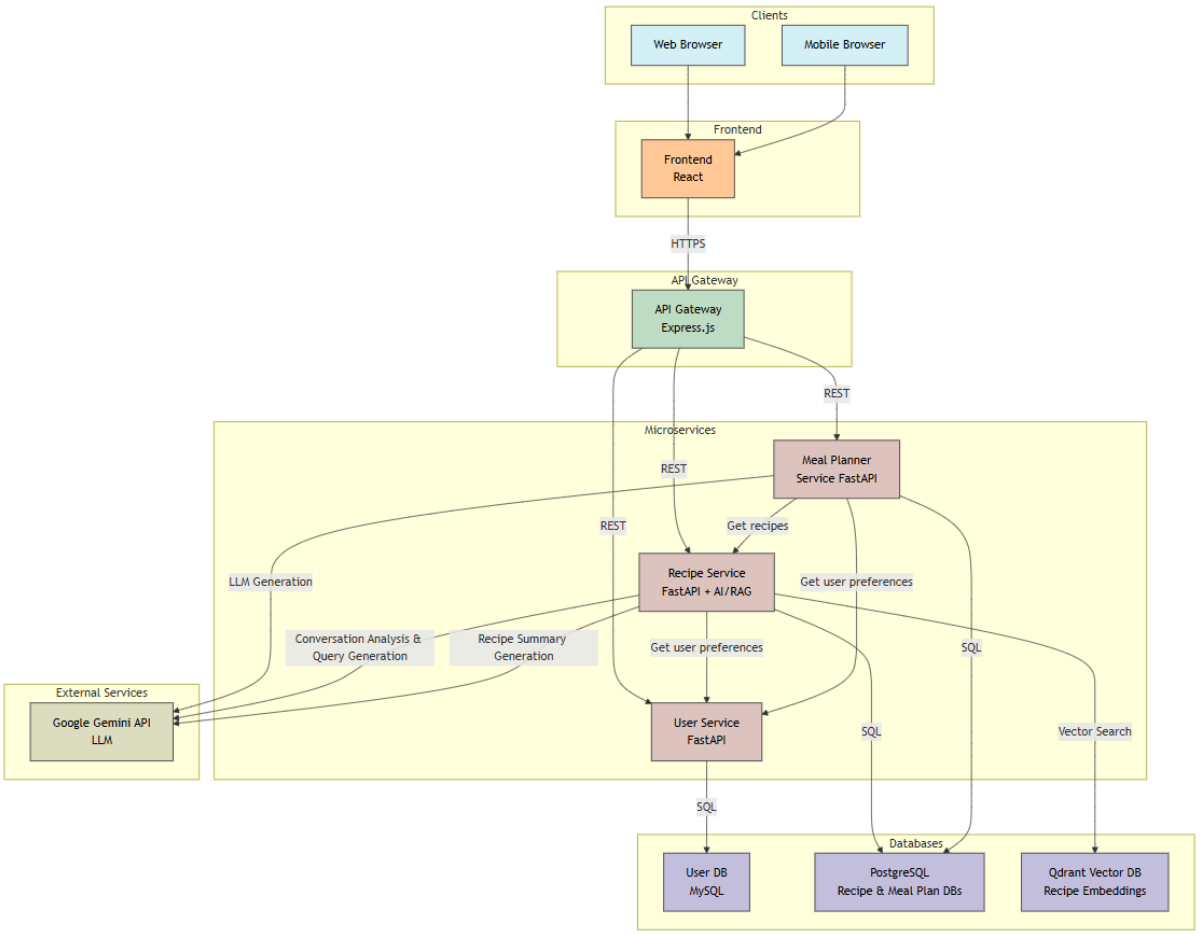
    - **Qdrant Vector Database** for semantic recipe search using AI embeddings

    - **MongoDB** for chat conversation history and context persistence

- **AI/ML Components**:
    - **Google Gemini LLM** for natural language understanding and personalized meal plan generation
    - **all-mpnet-base-v2 SentenceTransformers** for generating 768-dimensional embeddings used in semantic recipe retrieval

- **Infrastructure**: All services are **containerized with Docker**, orchestrated with Docker Compose, and fronted by **NGINX** for routing and load balancing. JWT-based authentication ensures security across the system.

Architecture



The architecture of **MealMateAI** is structured as a **multi-layered microservices system** that integrates AI-driven services with robust backend infrastructure. Each component is modular, independently deployable, and communicates via secure APIs, ensuring scalability, resilience, and ease of maintenance.

Client Layer

The **Client Layer** consists of components such as the web browser and mobile browser, serving as the primary entry points for end users. Through these interfaces, users can access the system using a responsive web application or mobile browser. This layer communicates with the frontend via HTTPS, ensuring secure data transfer between the client and the system.

Frontend Layer

The **Frontend Layer** is built with **React**, using **TypeScript** and **Material-UI** to implement the user interface for meal planning, recipe exploration, grocery list generation, and profile management. It is responsible for handling user interactions and visualizations, offering features such as drag-and-drop meal scheduling and real-time search suggestions. All API requests from this layer are routed exclusively through the API Gateway, ensuring consistent and secure communication with the backend service

API Gateway Layer

The **API Gateway Layer**, built with **Express.js**, serves as the unified entry point for all client requests. Its responsibilities include routing requests to the appropriate microservices, enforcing security through JWT-based authentication, request validation, and rate limiting, as well as providing observability with logging, monitoring, and health check endpoints. By decoupling the frontend from backend services, the API Gateway simplifies communication, and supports easier scaling of the system.

Microservices Layer

This layer implements the core business logic of MealMateAI, with each microservice designed for a single aspect of the services.

User Service (FastAPI + MySQL)

The **User Service**, developed with **FastAPI** and backed by **MySQL**, is responsible for managing user registration, authentication, and profile updates. It securely stores user-specific information such as dietary restrictions, allergies, and cuisine preferences, ensuring personalized experiences. Additionally, this service provides preference data to other services, supporting meal planning and recipe retrieval tailored to individual user needs.

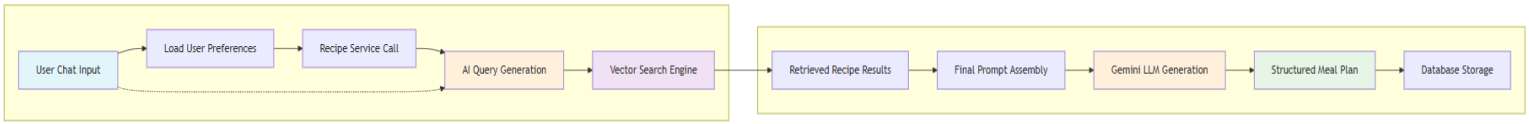
Recipe Service (FastAPI + PostgreSQL + Qdrant + Gemini)

The **Recipe Service**, built with **FastAPI** and integrated with **PostgreSQL**, **Qdrant**, and **Google Gemini**, is designed to handle recipe storage, retrieval, and intelligent search. Recipes are stored and managed in PostgreSQL, while Qdrant serves as the vector database for semantic recipe search using embeddings generated by all-mpnet-base-v2. To enhance user experience, the service leverages the Google Gemini API for conversation

analysis to infer user intent, optimized query generation for vector searches, and AI-generated recipe summaries that provide better representations in the embedding space.

Meal Planner Service (FastAPI + PostgreSQL + Gemini)

The **Meal Planner Service**, developed with **FastAPI** and powered by **PostgreSQL** and **Google Gemini**, orchestrates Retrieval-Augmented Generation (**RAG**) to deliver personalized meal planning. It retrieves user preferences from the User Service and recipe data from the Recipe Service, then compiles this context to send to the Gemini LLM for generating customized multi-day meal plans. Once created, the structured meal plans are stored and managed in PostgreSQL for persistence and future access.



Databases

The **Databases** consist of multiple specialized data stores that support different aspects of the system.

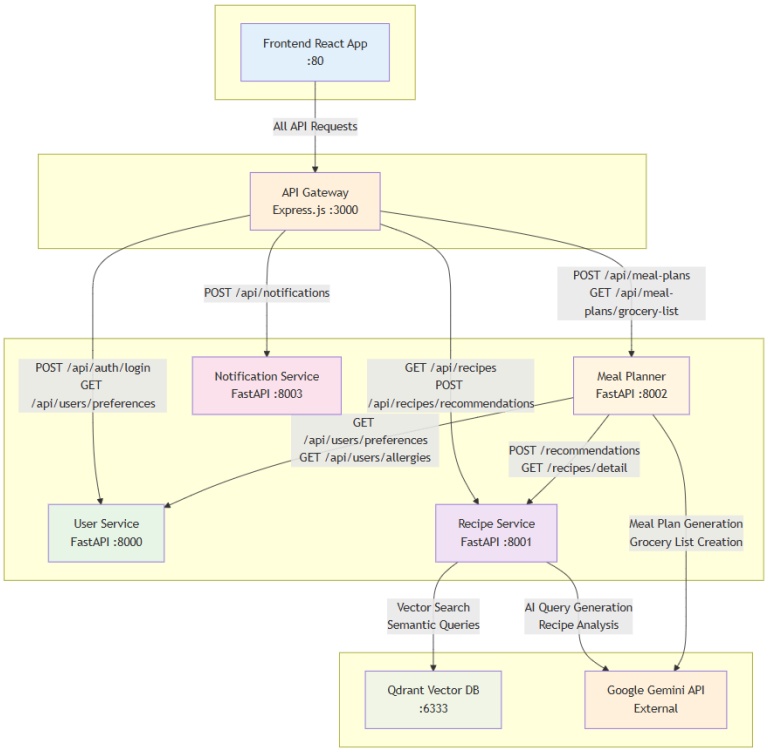
- The **User Database**, implemented in MySQL, manages user accounts, authentication data, and personal preferences.
- The **Recipe and Meal Plan Databases**, built on PostgreSQL, serve as the central repositories for recipes, meal plans, and their associated metadata.
- The **Qdrant Vector Database** is optimized for semantic recipe embeddings, enabling efficient cosine similarity searches to power intelligent and context-aware recipe retrieval.

External Services

Google Gemini API (LLM)

The **Google Gemini API (LLM)** provides advanced natural language understanding to support user interactions across the platform. It generates recipe summaries, optimized search queries, and fully personalized meal plans, enhancing both usability and relevance. Working together with internal services such as the Recipe Service and Meal Planner Service, Gemini enables AI-powered personalization that tailors the system's outputs to each user's preferences and needs.

Inter-Service Communication



Features

MealMateAI provides a comprehensive suite of features that combine AI-driven personalization with a scalable microservices foundation. The following are the most important capabilities that define the system's value for end users and enterprise deployment.

User Management & Personalization

- **Secure authentication with JWT:** safe logins and consistent sessions.
- **Flexible preference storage:** dietary restrictions, allergies, and cuisines.

AI-Powered Recipe Discovery

- **Semantic search:** natural language queries using vector embeddings.
- **Conversational AI:** interprets preferences from user chat history.
- **Smart query generation:** across 8 specialized recipe collections.

Intelligent Meal Planning

- **RAG-based personalized multi-day meal plan generation**.
- **Dietary compliance:** allergies, restrictions, and nutritional balance.
- **Dynamic adjustments:** swapping meals and regenerating plans.
- **Automated grocery list creation.**

Recipe Collection Management

- **AI-generated recipe summaries: used to embed the recipes.**.

Performance & Real-Time Processing

- **Fast vector search** for ~10k of recipes .
- [put real performance here]

Data Integration & APIs

- **AI integration:** via Google Gemini API for external LLM tasks.

User Experience

- **Feature 21:** Modern, responsive UI built with React + Material-UI.
- **Interactive browsing:** drag-and-drop scheduling.

Security, Monitoring & Deployment

- **Containerized deployment**: Docker and horizontal scaling.
- **Comprehensive logging, error tracking, and analytics**..

Implementation Details

Implementation: MealMateAI is built as a containerized microservices system for portability, scalability, and modular deployment.

Hosting & Deployment: Hosted in a Dockerized environment, orchestrated via Docker Compose for development/staging.

Infrastructure Management: NGINX serves as a reverse proxy and load balancer; each microservice runs in its own Docker container.

Database Integration:

- MySQL for user accounts, authentication, and preferences.
- PostgreSQL for recipes, metadata, and meal plans with advanced indexing.
- Qdrant Vector DB for semantic search using recipe embeddings.

AI Integration: Google Gemini API and all-mpnet-base-v2 embeddings provide AI-powered functionalities (meal planning, semantic search, conversation analysis), orchestrated via backend microservices.

Security: JWT-based authentication, input validation, rate limiting and secure storage of sensitive credentials via environment variables.

Performance Optimizations: Asynchronous APIs using FastAPI for high-throughput processing.

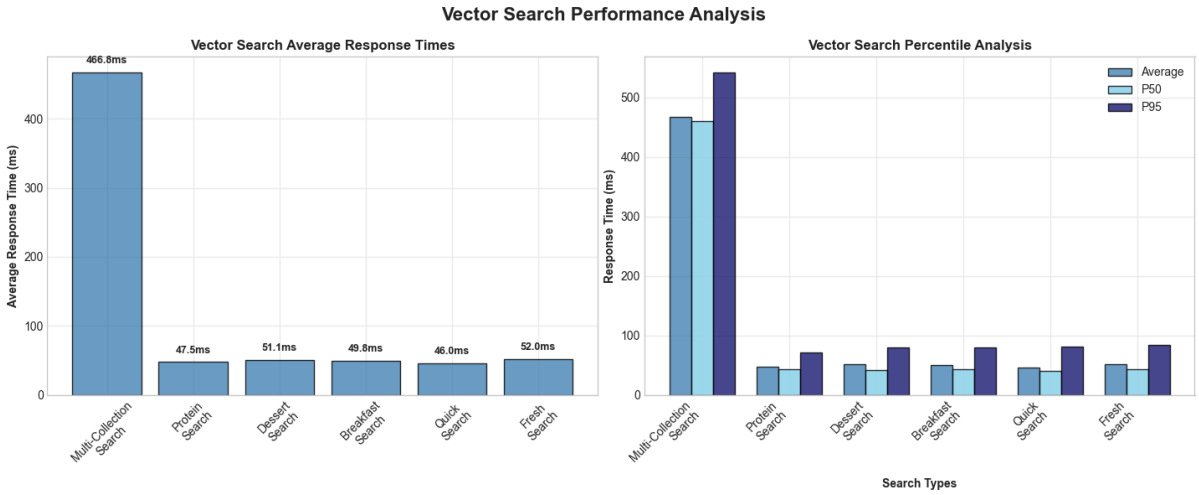
Monitoring & Logging: Health endpoints (/health) on each service, structured logging for traceability and analytics.

Scalability: Services can scale horizontally independently to handle spikes in AI-powered requests without affecting others.

Performance and Benchmarking

MealMateAI Performance Summary Table

Category	Operation	Avg (ms)	Min (ms)	Max (ms)	P50 (ms)	P95 (ms)	P99 (ms)	Tests
Database Operations	Health Check	21.0	7.5	85.2	14.9	52.5	78.7	15
	Database Recipes	28.9	17.7	93.0	23.4	49.6	84.3	15
	Database Collections	60.3	20.2	421.4	25.2	226.0	382.4	12
	Database Recipe Detail	17.2	7.4	90.2	8.9	55.1	83.2	10
Vector Search	Vector Multi Search	466.8	338.5	829.4	460.1	541.8	771.9	20
	Vector Protein Search	47.5	38.5	100.5	43.0	70.8	94.5	15
	Vector Dessert Search	51.1	37.9	81.5	42.4	80.0	81.2	15
	Vector Breakfast Search	49.8	38.5	80.0	42.9	79.8	80.0	15
	Vector Quick Search	46.0	36.7	82.6	39.8	81.0	82.3	15
	Vector Fresh Search	52.0	40.0	90.2	43.7	83.4	88.9	15
User Operations	Authentication	830.2	527.9	1694.9	650.4	1436.1	1643.1	10
	User Preferences Get	7.5	5.7	15.4	6.6	12.0	14.7	10
	User Preferences Update	18.2	9.8	43.2	12.8	37.2	42.0	5
AI Operations	Ai Recommendations	15.1s	10.2s	23.2s	14.6s	23.1s	23.2s	10
	Ai Meal Planning	17.1s	13.8s	20.8s	17.8s	20.5s	20.8s	5
Concurrent Load	Concurrent Vector	1551.2	361.1	1808.9	1695.8	1766.1	1800.3	20
	Concurrent Database	50.4	26.3	61.1	52.8	57.9	60.4	20



Use Cases

MealMateAI is designed to support diverse scenarios where personalized meal planning, recipe discovery, and dietary adaptation are valuable. Because it integrates semantic search, AI-powered summaries, and an extensible microservices backend, the system can be applied in multiple domains:

MealMateAI is applicable in various domains, including:

- **Individual Consumers:** Provides weekly meal plans tailored to tastes, time constraints, and nutritional goals, enabling personalized nutrition, time savings through automatic recipe retrieval, and discovery via semantic search.

- **Health & Fitness Applications:** Integrates as a backend recommendation engine for wellness or fitness platforms, supporting diet-aligned planning, enhancing user engagement, and enabling clinical dietary plans with extensions such as diabetes-friendly options.

- **Food Services and Retail:** Assists restaurants, catering services, and grocery platforms with recipe recommendations using seasonal ingredients, automated menu planning for institutions, and integration with product catalogs for streamlined grocery lists.

- **Educational Contexts:** Serves as a teaching tool for microservices architecture, demonstrating retrieval-augmented generation workflows, and providing practical examples for applied NLP and ML in data science projects.

Future Enhancements

While MealMateAI already provides a functional meal planning and recipe recommendation platform, there are multiple opportunities to extend its capabilities and improve user experience:

Future Enhancements for MealMateAI:

- **Mobile Support:** Dedicated mobile app (React Native/Flutter) for on-the-go meal planning, notifications, and recipe browsing.
- **Advanced Personalization:** Integrate richer user data (nutritional goals, allergies, health conditions) and link with wearable devices for dynamic recommendations.
- **Notifications:** Push notifications, SMS/WhatsApp alerts, and weekly meal plan summaries.
- **Shopping List & Grocery Integration:** Auto-generated shopping lists and direct ordering via grocery partnerships.
- **Expanded AI Capabilities:** Recipe adaptation for dietary restrictions, nutritional analysis, and optimized meal plan generation.
- **Improved Semantic Search:** Hybrid keyword-semantic search, multilingual support, and fine-tuned culinary embeddings.
- **Collaboration Features:** Shareable meal plans, co-created recipes, and curated collections for social engagement.
- **Security & Privacy:** Multi-factor authentication, secure storage for sensitive data, and role-based access control.

Conclusion