Phase 1 Documentation: Recommendation System Refactor

# ✅ Overview

In Phase 1, we performed a full architectural refactor of the food delivery recommendation system to improve modularity, maintainability, and correctness. We separated the system into collaborative, content-based, and fallback engines, fixed MongoDB integration, and restored proper ID ↔ name mapping.

# ✅ Key Tasks Completed

• Split the system into three engine modules: CollaborativeEngine, ContentEngine, FallbackEngine.

• Built a RecommendationOrchestrator to coordinate between the engines.

• Updated CollaborativeEngine to pull ratings directly from MongoDB Reviews collection.

• Ensured CollaborativeEngine works with both user IDs and user names.

• Corrected MongoDB field mapping for Reviews, Orders, Products collections.

• Updated fallback engine to use user\_preferences collection correctly.

• Added detailed debug logging to trace why fallback layers activate.

• Updated api.py and main\_runner.py to wire everything correctly.

• Verified outputs with multiple users, confirming collaborative and fallback recommendations work.

# ✅ Files Updated

• collaborative\_engine.py

• content\_engine.py

• content\_based\_product\_recommender.py

• fallback\_engine.py

• orchestrator.py

• api.py

• main\_runner.py

# ✅ Current Status

The system is now modular, with proper MongoDB data integration, and produces collaborative recommendations when user vectors and neighbor ratings are available. It cleanly falls back to preference or global recommendations when needed.

# 🚀 Ready for Phase 2

We are now ready to enter Phase 2, which will focus on algorithmic improvements, advanced filtering (e.g., time decay), sentiment integration, performance optimizations, and new recommendation features.