📄 Bilingual Food Recommendation Chatbot - Technical Documentation

# 1. Overview

This chatbot provides food recommendations based on natural language queries in English or French. It connects to a MongoDB database containing restaurant and product data, performs fuzzy keyword matching, and returns top relevant dishes ranked by rating.

# 2. Technologies Used

• Python – Core programming language

• Flask – API server for chatbot endpoint

• MongoDB – Stores restaurant, product, and user data

• Pandas – Used for data transformation and merging

• PyMongo – Python client for MongoDB

• Flask-CORS – Enables cross-origin access

• difflib – Fuzzy matching for keywords

• re / unicodedata – Text normalization and keyword extraction

# 3. File Structure

recommendation\_systemv1/

├── chatbot\_api.py # Flask API endpoint

├── chatbot\_handler.py # Core logic for recommendations

├── data\_loader.py # MongoDB connection and collection loader

├── config.yaml # MongoDB URI

# 4. data\_loader.py – MongoDB Connection Helper

• Loads configuration from YAML

• Connects to MongoDB

• Returns references to all major collections

# 5. chatbot\_handler.py – Chatbot Logic

• Combines English and French stopwords

• Normalizes and tokenizes input using regex and Unicode

• Loads data directly from MongoDB (not CSV)

• Performs fuzzy matching using difflib.get\_close\_matches

• Merges Products with Restaurents to fetch ratings and names

• Deduplicates keywords and trims extra spaces in restaurant names

# 6. chatbot\_api.py – Flask API

• Provides a /chatbot endpoint

• Accepts POST request with JSON containing user\_id and message

• Calls handler and returns recommendations

# 7. config.yaml – MongoDB URI

mongo\_uri: "mongodb://localhost:27017"

# 8. Data Flow Summary

1. User sends message to /chatbot

2. Flask API parses input and calls handler

3. Handler extracts keywords and queries MongoDB

4. Fuzzy match performed and top results sorted by rating

5. JSON response returned with dishes and restaurants

# 9. Language Support

• Combined English and French stopword filtering

• Normalizes accented characters (e.g., crème -> creme)

• Extracts meaningful keywords for both languages

# 10. Example Queries

Input: Je veux une pizza au fromage

→ Keywords: ["pizza", "fromage"]

Input: I want a spicy burger

→ Keywords: ["spicy", "burger"]

# 11. Future Improvements (Optional)

• Add language detection and translation

• Include dish images

• Personalize recommendations by user history

• Add zone-based filtering or delivery time