Tecnvirons Pvt LTD

AI Food Agent Assignment: "FoodieBot" - Database-Driven Conversational Fast Food System

Deadline : - 16/9/25

The Challenge Overview

Create an intelligent conversational AI system that works with pre-generated fast food product data stored in a database. The bot should analyze customer conversations, calculate interest scores, and recommend products from your database in real-time.

Assignment Structure (3 Phases)

PHASE 1: Product Data Generation & Database Setup

Step 1: Generate 100 Fast Food Products using Grok

Use Grok API (or alternative AI) to generate a comprehensive fast food product dataset:

Required Product Data Structure:

json

```json
{
  "product_id": "FF001",
  "name": "Spicy Fusion Dragon Burger",
  "category": "Burgers",
  "description": "Korean gochujang meets American beef with kimchi slaw and sriracha aioli",
  "ingredients": ["beef patty", "gochujang sauce", "kimchi", "brioche bun", "sriracha aioli"],
  "price": 12.99,

  "calories": 680,

  "prep_time": "8-10 mins",

  "dietary_tags": ["spicy", "fusion", "contains_gluten"],

  "mood_tags": ["adventurous", "comfort", "indulgent"],

  "allergens": ["gluten", "soy", "dairy"],

  "popularity_score": 85,

  "chef_special": true,

  "limited_time": false,

  "spice_level": 7,

  "image_prompt": "korean-fusion burger with kimchi and gochujang sauce on brioche bun"

}
```

Product Categories to Generate (10 each):

- Burgers (classic, fusion, vegetarian)
- Pizza (traditional, gourmet, personal)
- Fried Chicken (wings, tenders, sandwiches)
- Tacos & Wraps (mexican, fusion, healthy)
- Sides & Appetizers (fries, onion rings, etc.)
- Beverages (sodas, shakes, specialty drinks)
- Desserts (ice cream, cookies, pastries)
- Salads & Healthy Options
- Breakfast Items (all-day breakfast)
- Limited Time Specials

Step 2: Database Implementation

Choose ONE database option:

- SQLite: Simple, no setup required
- PostgreSQL: Free tier on Railway/Neon/Supabase
- MongoDB: Free Atlas tier
- Firebase: Google's free tier

PHASE 2: Conversational AI with Interest Scoring

Conversational Intelligence Engine

Build an AI system that:

Core Conversation Features:

- **Natural dialogue about food preferences**
- **Extract dietary restrictions, budget, mood, cravings**
- **Calculate real-time interest scores (0-100%)**
- **Maintain conversation context and memory**
- **Handle complex food-related queries**

Interest Score Calculation Logic:

python

```python
# Interest Score Factors
ENGAGEMENT_FACTORS = {
    'specific_preferences': +15,  # "I love spicy Korean food"
    'dietary_restrictions': +10,  # "I'm vegetarian"
    'budget_mention': +5,         # "Under $15"
    'mood_indication': +20,       # "I'm feeling adventurous"
    'question_asking': +10,       # "What's the spice level?"
    'enthusiasm_words': +8,       # "amazing", "perfect", "love"
    'price_inquiry': +25,         # "How much is that?"
    'order_intent': +30,          # "I'll take it", "Add to cart"
}


NEGATIVE_FACTORS = {
    'hesitation': -10,            # "maybe", "not sure"
    'budget_concern': -15,        # "too expensive"
    'dietary_conflict': -20,      # Product doesn't match restrictions
    'rejection': -25,             # "I don't like that"
    'delay_response': -5,       # Long response time
}
```

Database Query Integration:

- **Search products by category, dietary tags, price range**
- **Filter by mood tags and preferences**
- **Rank products by popularity and compatibility**
- **Real-time product availability checks**

Example Conversation with Database Integration:

Bot: "Welcome to FoodieBot! What's your food mood today? "

User: "I want something spicy and adventurous but under $10"

[Database Query: SELECT * FROM products WHERE 'spicy' IN (mood_tags) AND 'adventurous' IN (mood_tags) AND price < 10]

[Interest Score: 75% - Specific preferences + budget indication]

Bot: "Perfect! I found 3 amazing spicy adventures under $10. Our 'Fire Dragon Tacos' are super popular - Korean chili beef with jalapeño lime crema for $8.99. Want to hear more?"

User: "That sounds perfect! What's the spice level?"

[Interest Score: 90% - Enthusiasm + specific question]

[Database Query: SELECT spice_level FROM products WHERE name = 'Fire Dragon Tacos']

Bot: "It's a solid 7/10 on our spice scale - definitely has a kick but won't destroy your taste buds! The Korean chili gives it a smoky heat. Ready to add it?"

PHASE 3: Smart Recommendation & Analytics System (25%)

Database-Driven Recommendation Engine

Create intelligent product matching:

Recommendation Algorithms:

1. **Preference Matching: Match conversation keywords to product tags**
2. **Collaborative Filtering: "Customers who liked X also liked Y"**
3. **Mood-Based Filtering: Map customer emotions to product mood_tags**
4. **Budget Optimization: Find best value within price range**
5. **Dietary Intelligence: Strict filtering for restrictions/allergens**

Interest-Driven Product Selection:

Real-Time Analytics Dashboard

Track conversation and database metrics:

Conversation Analytics:

- **Interest score progression graphs**
- **Most effective conversation paths**
- **Drop-off points identification**
- **Average conversation duration**
- **Conversion rate by interest score ranges**

Product Analytics:

- **Most recommended products**
- **Highest converting products**
- **Products by interest score correlation**
- **Category performance analysis**
- **Price sensitivity analysis**

Technical Requirements

Required APIs & Tools:

- **Grok API: For initial product generation**
- **Database: One of SQLite/PostgreSQL/MongoDB/Firebase**
- **LLM API: Choose from Groq, Hugging Face, Gemini, or Ollama**
- **Web Framework: Flask/FastAPI for backend API**

System Architecture:

[User Input] → [Conversation AI Agent] → [Interest Score Calculator]

↓ ↓ ↓

[Database Query Engine] → [Recommendation System] → [Response Generator]

↓ ↓ ↓

[Analytics Tracker] → [Database Logger] → [User Interface]

Database Performance Requirements:

- Sub-100ms query response time
- Proper indexing on searchable fields
- Connection pooling for concurrent users
- Database migration scripts
- Data backup strategy

Interface & Demo Requirements

User Interface (Choose One):

- Streamlit: Chat interface with live analytics
- Web App: HTML/CSS/JS with database connectivity
- API + Frontend: Separate backend/frontend architecture

Must Include:

- Live chat interface with database-driven responses
- Real-time interest score display (0-100%)
- Product cards pulled from database with images
- Conversation history stored in database
- Analytics dashboard showing database queries and results
- Product search/filter functionality
- Database admin panel to view/edit products

Database Visualization:

- Show current database queries being executed
- Display product match scores in real-time
- Conversation-to-database query mapping
- Product recommendation ranking visualization

Evaluation Criteria

Database Design & Implementation (25%)

- **Proper schema design and normalization**
- **Efficient querying and indexing**
- **Data integrity and validation**
- **Query optimization and performance**

AI Product Generation Quality (20%)

- **Creativity and realism of Grok-generated products**
- **Data consistency and completeness**
- **Proper categorization and tagging**
- **Variety and uniqueness across 100 products**

Conversational Intelligence (25%)

- **Natural conversation flow with database integration**
- **Accurate interest scoring throughout conversation**
- **Context retention and memory management**
- **Smart product matching from database**

System Architecture & Performance (20%)

- **Clean code structure and documentation**
- **Database connection management**
- **Real-time performance optimization**
- **Error handling and edge cases**

Analytics & Insights (10%)

- **Meaningful conversation analytics**
- **Product performance tracking**
- **Interest score correlation analysis**
- **Business intelligence dashboard**

Deliverables

1. Complete System Package

2. Product Generation Documentation

3. System Demonstration

Ready to build the most sophisticated database-driven conversational AI food system? Let's make data-driven food recommendations that customers can't resist!