NutriAI

AI-Powered Personalized Nutrition Assistant

Intelligent meal recommendations · AI coaching · Body composition tracking

Flask + PostgreSQL

Groq / LLaMA 3.3-70B

Recommendation System

500K+ Recipes

By: **Hasan Al Mousawi & Ali Zalghout**

Supervisor: Dr. Kassem Danach

Lebanese University · Bachelor of Data Science · 2025–2026

Project Overview

What is NutriAI?

A full-stack web application delivering intelligent, personalised nutritional guidance through the fusion of AI, machine learning, and modern web technologies.

Core goals:
• Generate personalised meal recommendations based on fitness goals, dietary preferences, and nutritional needs.
• Provide an AI-driven conversational assistant for food & exercise logging.
• Track body composition and adapt targets via InBody analysis.
• Visualise weekly nutrition patterns through a rich analytics dashboard.

Tech Stack

Backend	Python · Flask
Database	PostgreSQL
ML Models	K-NN · TF-IDF · KMeans
AI / LLM	Groq API · LLaMA 3.3-70B
Frontend	HTML · CSS · JavaScript
APIs	Pexels · Groq

Data Collection & Dataset

522K+	1.4M+	370K+	7
Raw Recipes	User Reviews	After Cleaning	Dietary Tags

Food.com Dataset

Source: Kaggle Food.com Recipes & Reviews

Fields include: Name, Calories, Protein, Fat, Carbs, Fibre, Sugar, Saturated Fat, Cholesterol, Sodium, Ingredients, Cooking Time, User Ratings, Images

Challenges: Missing values, broken image URLs, inconsistent ingredient units, community noise, insufficient features.

Data Engineering Fixes

- Meal-type Labels
- goal scores
- Dietary Tags

Meal types assigned are:
Breakfast · Lunch · Dinner · Snack

7 dietary tags engineered:
Vegan, Vegetarian , Gluten-Free
Dairy-Free, Nut-Free, High-Protein,
Low-Carb

Data Preprocessing Pipeline

01

Structural Cleaning

Remove null RecipeId, zero-serving rows

02

Quality Filtering

Calorie bounds (<15 or >1500 kcal/serving), phantom-nutrition & bad macro-energy ratios

03

Feature Engineering

Dietary tags, goal scores (lose/maintain/build), meal-type keyword pass

04

Meal-Type kNN Vote

k=15 TF-IDF nearest-neighbour vote classifies recipes keywords missed

05

Duplicate Removal

De-duplicate near-identical recipe names to boost diversity

06

Unit Prediction

Regex + lookup table assigns units, converts to per-serving quantities

07

Per-Serving Validation

Drop recipes with absurd per-serving amounts or any quantity >10

08

TF-IDF Training

12K-vocab, fit on a 40K-recipe sample

09

Review Cleaning

Filter reviews to valid RecipeIds

Hybrid Recommendation Engine

Content-Based (TF-IDF)

Vectorises ingredients,name & keywords
Cosine similarity against user preference profile
Weight: 65% in final hybrid blend

Popularity Bonus / Soft Preference Boost

Popularity Bonus: boosts highly rated and frequently reviewed recipes.
Preference Boost: increases scores for recipes matching user preferences.

Goal Score Boosting

Lose · Maintain · Build Muscle
Pre-computed scores boost relevant macros
Guaranteed dietary-constraint compliance

MMR (Maximal Marginal Relevance)

- Re-ranks recommendations by balancing high relevance with diversity to avoid repetitive recipes.
- Used in systems like YouTube and Spotify to keep recommendations diverse and less repetitive.

Score = 2.0 × (0.65 × Content + 0.35 × CF) + 0.9 × Goal + Popularity + Prefs | Dietary filters first, MMR diversifies

Hybrid Recommendation Engine

Collaborative Filtering (CF)

Collaborative Filtering recommends recipes using user rating behavior patterns rather than recipe ingredients or text.

User-Based Collaborative Filtering (2 Tiers)

Tier 1 — NutriAI User-Based CF (Profile Matching + Cosine Similarity): recommends from users with similar profiles.
Tier 2 — Food.com User-Based CF (User Similarity + Cosine Similarity): recommends from users with similar tastes.

Item Based CF (K-NN + cosine-similarity)

Recommends recipes similar to recipes you already liked based on user rating patterns.

7 Recommendation Pathways

- ✓ Hybrid (Content + CF)
- ✓ Cold Start (New Users)
- ✓ Content-Based (Pure TF-IDF)
- ✓ Similar User (2-Tier)
- ✓ History / Previous Meals
- ✓ Ingredient Search
- ✓ Dashboard Auto-Suggest

Score = 2.0 × (0.65 × Content + 0.35 × CF) + 0.9 × Goal + Popularity + Prefs | Dietary filters first, MMR diversifies

Recommendation Pipeline Flow

Dietary Filters First → Score & Blend → MMR Diversifies → Top-N Final Results

AI Conversational Assistant

Powered by: Groq API · LLaMA 3.3-70B model · Avg response 0.8–2.5 s

Exercise Logging

Natural language activity detection.
MET-based calorie burn estimation.
Stored as negative-calorie meal_log entry.

Food Logging

Describe any meal in plain text.
LLM estimates full macro breakdown.
Instant confirmation + log update.

Macros-Precise Suggestion

Ask for meals hitting exact macros.
LLM builds options to match your targets.
Ready to log with one tap.

Ingredient-Based Suggestions

List what's in your kitchen.
LLM suggests meals you can make right now.
No extra shopping required.

General Chat

Personalised dietary advice.
Remaining calorie budget reporting.
Contextual AI chef assistant mode.

InBody Body Composition Analysis

01

Input Collection

Weight, body fat %, muscle mass, BMR, and visceral fat supplied by user

02

Field Validation

All fields checked for physiologically plausible ranges before processing

03

Trend Analysis

Compare current scan against previous scan & 30-day calorie logs

04

Adherence Score

Checks logging consistency – distinguishes tracking gaps from diet failures

05

Verdict Classification

4 categories: Insufficient Data, On Track, Ubove Target, Under Target

06

Groq AI Narrative

LLaMA generates personalised 3–4 sentence assessment + target adjustment

What Drives the Verdict

🎯 **Adherence Score** — calorie adherence (35%) + protein adherence (35%) + logging consistency (30%)

⚖️ **Body-Comp Override** — weight gain ≥4kg, or ≥0.75kg/week trending up, forces an above_target verdict regardless of calories

🚹 **Goal-Aware Logic** — the same calorie gap means something different for "lose," "maintain," and "build" goals

⚠️ **Contradiction Flags** — a separate list (not a verdict) surfacing mismatches like claimed vs. logged intake

Frontend Interface

🔑 Login / Register

Dark split-panel with animated gradient orbs. · Tabbed Sign In / Create Account. · Full profile collected at registration (no onboarding step).

🍴 Meals Page

Recommendation carousels: Suggested, Content, Similar Users, Previous Meals, Ingredient Search. · Detail panel with portion scaler. · Real-time macro tracker sidebar.

📈 Dashboard Page

5 parallel API calls on load. · Calorie chart · Macro donut · 30-day heatmap · Streaks · AI insights. · Time-of-day personalised greeting.

⚖️ User Metrics Panel

InBody test form + BMI check-in. · Weekly calorie bar chart & BMI trend line. · Day-by-day expandable meal history log.

👤 Profile Page

Two-column responsive form grid. · Dietary chip toggles + favourite ingredients. · 7-day profile update reminder banner.

Meals Page

✨ Suggested for Today

Always-visible top pick, refreshed daily via the hybrid recommendation engine.

💖 Recommendation Tabs

Your Favourites · Similar Users · Previous Meals — three switchable carousels.

🔍 Recipe Detail Panel

Full nutrition per serving, ingredient list, and step-by-step instructions.

⚖️ Portion Scaler

5 preset portions + custom input, live macro recalculation before logging.

💬 Embedded AI Chat

Floating chatbot widget for natural-language food & exercise logging, scoped to this page.

Profile Page

📝 Personal Details Form

Age, gender, height, weight, max cook time — two-column responsive grid.

🏃 Activity Level Selector

5 tiers, sedentary to very active (1.2–1.9×), feeds BMR/TDEE calculation.

🥗 Dietary Preference

Vegetarian, Vegan , Dairy-Free, Nut-Free, Low-Carb, High-Protein.

🌟 Favourite Ingredients

Free-text field feeding directly into cold-start & similar-user scoring.

⏰ 7-Day Update Reminder

Dynamic banner shown once profile data is 7+ days old, for accurate targets.

Dashboard & Analytics

📊 Weekly Calorie Chart

Bar chart of 7-day calorie totals.
Green = on-target · Coral = over-budget.
Dashed reference line at daily target.

🍩 Macro Donut Chart

Protein / Fat / Carbs proportion today.
Centre label shows total calories consumed.
Colour-coded consistently across the app.

📅 30-Day Calendar Heatmap

Colour-coded grid: calorie % of target.
Identify weekend under-logging patterns.
Hover tooltips for exact daily values.

🔥 Streaks & Achievements

Consecutive logging streak counter.
Badges: 'Dedicated Logger', 'Perfect Week', 'Protein Consistent'.
Motivates daily engagement.

⬜ 7-Day Logging Heatmap

Quick visual of last 7 days.
Checkmark = at least one meal logged.
Today's circle pulses to orient user in time.

🤖 AI Insights Card

Groq-generated 2–3 sentence observation.
Personalised to goal, macros & history.
Loads asynchronously – no blocking.

Coaching & Client Oversight

🔑 Separate Coach Accounts

Own table, own login, own JWT role ("coach"). · Never confusable with a regular user token. · coach_token_required decorator guards every coach route.

🔎 Find & Subscribe

User searches/browses the coach roster. · Request sits pending until the coach accepts. · Zero visibility into user data before acceptance.

📋 Client Dashboard

Roster of accepted clients with attention flags. · Read-only meal history & InBody/BMI progress tabs. · Private coach-only notes per client.

🎯 Test Requests

Coach requests a fresh InBody test or BMI check-in, with an optional note. · Client sees it in a "requests from your coach" inbox. · One submission auto-marks it fulfilled.

🔒 Privacy & Control

User can revoke a coach's access at any time. · Access is instantly cut off — no lingering visibility. · No messaging layer: users coordinate with coaches however they normally would.

System Architecture & Backend API

Frontend

- HTML · CSS · JS (Vanilla)
- 5 pages: Login, Meals, Dashboard, Metrics, Profile
- Chart.js dashboards · Fetch API

Flask API (app.py)

- 30+ REST endpoints
- JWT auth (token_required decorator)
- Groq integration · psycopg2 queries

PostgreSQL + Cache

- 8 tables: users, user_profiles, meal_log, user_ratings…
- Pickle DATA_CACHE (K-NN CF & TF-IDF in memory)
- image_cache table for Pexels URL caching

Key API Endpoints

`/api/auth/login` `/api/meals/recommend` `/api/chatbot` `/api/inbody/submit` `/api/dashboard/insights` `/api/meals/image`

Security Architecture

01 bcrypt Password Hashing

Cost factor 12 → ~250 ms per hash computation.
Constant-time comparison prevents timing attacks.
Passwords never stored in plaintext or reversible form.

02 JWT Authentication

72-hour token expiry with HS256 signing.
Algorithm whitelist prevents confusion attacks.
User existence re-verified on every protected request.

03 Parameterised SQL Queries

All 30+ endpoints use psycopg2 %s placeholders.
Centralised q() helper enforces the pattern.
SQL injection completely eliminated at the architecture level.

04 CORS Configuration

Flask-CORS applied globally to all API routes.
Dev: all origins permitted for local development.
Prod: should restrict to specific domain.

Conclusion & Future Work

✅ What We Built

A production-quality, full-stack AI nutrition app integrating LLMs, classical ML, and modern web development.

Hybrid recommendation engine combining K-NN Collaborative Filtering + TF-IDF content based scoring

AI chatbot with 5 intent types for natural language food & exercise logging.

InBody analysis with 4 verdict categories and personalised AI narratives.

Complete dashboard analytics with streaks, heatmaps, and AI insights.

Security-first architecture: bcrypt, JWT, parameterised SQL, CORS.

🚀 Future Work

→ **USDA FoodData API**
Replace LLM-estimated nutrition with validated database values

→ **Computer Vision Logging**
Photo → food recognition → automatic macro estimation

→ **Native Mobile App**
iOS/Android with push notifications, HealthKit/Google Fit

→ **Arabic & French Support**
Multi-language chatbot & UI for Lebanese users

→ **Dietitian Review Layer**
Human-in-the-loop for InBody target adjustments