AI Nutrition Project

File Documentation

Generated: 2025-12-30 11:57

Table of Contents

Root Files

■ main.py (src/)

Purpose: Main FastAPI application entry point

The core application file that defines all REST API endpoints for the AI Nutrition system.

Key Features:

- Authentication endpoints (login, register)

- Virtual Coach chat API with RAG integration

- Analytics endpoints (health score, trends, insights)

- Food scanning and logging APIs

- Medical report upload with OCR

- Exercise Guidance AI

- Fix My Meal (Clinical Nutrition AI)

Technologies: FastAPI, Pydantic, async/await

Key Endpoints:

- POST /api/coach/chat - AI-powered nutrition chat
- GET /api/analytics/score - Health score calculation
- POST /api/food/scan - YOLO-based food recognition
- POST /api/upload/medical-report - OCR medical report parsing
- GET /api/analytics/feature-importance - Feature selection analysis

■ requirements.txt

Purpose: Python dependencies

Lists all required packages: FastAPI, uvicorn, python-dotenv, PyJWT, ultralytics (YOLO), etc.

■ Dockerfile

Purpose: Container configuration

Docker configuration for containerized deployment of the application.

■ docker-compose.yml

Purpose: Multi-container orchestration

Defines services, networks, and volumes for Docker deployment.

Analytics Module (src/analytics/)

■ analytics_service.py

Purpose: Health analytics and scoring

Computes health metrics and insights from user meal data.

Features:

- Health score calculation (0-100)

- Daily/weekly nutrient trends

- Pattern detection (late-night eating, sugar spikes)

- Actionable insights generation

- Meal log storage and retrieval

■ feature_selection.py

Purpose: ML Feature Selection Analysis

Analyzes nutritional features to determine importance for health predictions.

Techniques Used:

- Pearson correlation analysis between nutrients

- Variance threshold filtering

- Feature importance scoring for health conditions

- Median-based missing value imputation

- Duplicate detection and removal

Health Conditions Analyzed:

- diabetes_risk (based on sugar/carbs)

- hypertension_risk (based on sodium)

- obesity_risk (based on calories/fats)

- heart_health_risk (based on fats)

Services Module (src/services/)

■ llm_service.py

Purpose: LLM Integration (Ollama/Gemma)

Provides AI-powered responses using local LLM models.

Features:

- Ollama API integration

- System prompts for nutrition coaching

- RAG context injection

- Fallback handling when LLM unavailable

■ rag_service.py

Purpose: Retrieval-Augmented Generation

Retrieves relevant context for AI responses.

Data Sources:

- User medical profiles (conditions, allergens, medications)

- Meal history (recent logs)

- Food nutrition database (Indian Food CSV)

- Real-time food context from scans

■ yolo_service.py

Purpose: YOLO Food Recognition

Computer vision for food detection.

Features:

- YOLOv8/v11 model integration

- Food item classification

- Confidence scoring

- Integration with nutrition lookup

■ usda_service.py

Purpose: USDA Nutrition Database

External API integration for USDA FoodData Central nutrition lookups.

Rules Module (src/rules/)

🟩 engine.py

Purpose: Medical Rule Engine (Safety Layer)

Deterministic safety rules that ALWAYS override AI suggestions.

Rule Categories:

- Allergen detection (BLOCK severity)

- Diabetes rules (sugar, glycemic index, fiber-to-carb ratio)

- Hypertension rules (sodium monitoring)

- Obesity rules (calorie density, saturated fat)

Severity Levels: ALLOW → WARN → ALERT → BLOCK

Models Module (src/models/)

🟩 food.py

Purpose: Food and Nutrition Data Models

Pydantic/dataclass models for food items.

Classes:

- NutritionInfo (calories, protein, carbs, fat, sugar, fiber, sodium)

- Food (food_id, name, serving_size, nutrition, allergens)

- FoodCategory (enum)

🟩 user.py

Purpose: User Profile Models

User health profile definitions.

Classes:

- UserProfile (user_id, conditions, allergens, daily_targets)

- HealthCondition (enum: DIABETES, HYPERTENSION, OBESITY)

- DailyTargets (calorie/nutrient limits)

- DailyIntake (current day's consumption)

🟩 conversation.py

Purpose: Chat Conversation Models

Models for chat history, messages, and conversation context.

🟩 feedback.py

Purpose: User Feedback Models

Models for collecting user feedback on AI responses.

🟩 analytics_models.py

Purpose: Analytics Data Models

Models for health scores, trends, patterns, and insights.

OCR Module (src/ocr/)

■ parser.py

Purpose: Medical Report Parser

Extracts medical data from uploaded reports.

Extracts:

- Glucose levels (blood sugar)

- Cholesterol values (HDL, LDL, total)

- Blood pressure readings

- Health conditions detection

■ service.py

Purpose: OCR Service Orchestrator

Coordinates OCR processing, PDF handling, and result storage.

■ food_recognition.py

Purpose: Food Label OCR

Extracts nutrition facts from food packaging labels.

■ error_handler.py

Purpose: OCR Error Handling

Robust error handling for OCR processing failures.

Coach Module (src/coach/)

🟩 virtual_coach.py

Purpose: AI Virtual Nutrition Coach

Context-aware nutrition coaching.

Features:

- Personalized dietary advice

- Food safety evaluation

- Meal suggestions

- Rule engine integration for medical safety

Intelligence Module (src/intelligence/)

🟩 recipe_generator.py

Purpose: AI Recipe Generation

Generates healthy recipes based on user preferences and restrictions.

🟩 meal_fixer.py

Purpose: Clinical Meal Analysis

Analyzes meals and suggests improvements.

Features:

- Problem detection based on health conditions

- REMOVE/REDUCE/REPLACE suggestions

- Healthier alternatives

🟩 router.py

Purpose: Intelligence Request Router

Routes requests to appropriate AI services.

Auth Module (src/auth/)

🟩 auth_service.py

Purpose: Authentication Service

JWT-based authentication, token generation and verification.

🟩 database.py

Purpose: Database Operations

SQLite database for users, medical profiles, and uploads.

Feedback Module (src/feedback/)

🟩 feedback_service.py

Purpose: User Feedback Collection

Collects and stores user feedback on AI responses for improvement.

Frontend (static/)

■ login.html

Purpose: Login Page

User authentication interface.

■ register.html

Purpose: Registration Page

New user registration form.

■ dashboard.html

Purpose: Main Dashboard

Health score, trends, insights, and AI coach interface.

■ food-scan.html

Purpose: Food Scanner

Camera-based food scanning with YOLO detection.

■ upload.html

Purpose: Document Upload

Medical report upload for OCR processing.

Tests (tests/)

■ test_rule_engine.py

Purpose: Rule Engine Tests

Unit tests for medical safety rules.

■ test_feature_selection.py

Purpose: Feature Selection Tests

Tests for correlation, variance, and feature importance.

■ test_intelligence.py

Purpose: Intelligence Module Tests

Tests for meal fixer and recipe generator.

■ test_phase3.py

Purpose: Integration Tests

End-to-end tests for complete workflows.

Data (data/)

■ Indian_Food_Nutrition_Processed.csv

Purpose: Nutrition Database

Indian food nutrition dataset with 1016 food items.

Columns:

- Dish Name
- Calories (kcal)
- Carbohydrates (g)
- Protein (g)
- Fats (g)
- Free Sugar (g)
- Fibre (g)
- Sodium (mg)
- Calcium (mg)
- Iron (mg)
- Vitamin C (mg)
- Folate (µg)