

## Review Article

## Open Access

## FoodVisor: An AI-Powered Food Label Analysis System for Ingredient Interpretation and Personalized Dietary Recommendations

Amogha CH, Padmapriya R\* and Ajina A

Department of AI & ML, M S Ramaiah Institute of Technology, Bangalore, India

### ABSTRACT

FoodVisor is an AI-powered food label analysis system that offers personalized ingredient interpretation and dietary recommendations. Users input health data securely (AES-128 encryption), and scan product labels using OCR or barcodes. A hybrid AI model combining Retrieval-Augmented Generation (RAG) and a fine-tuned LLaMA 2 analyzes ingredients in context with user allergies and medical conditions. Integrated APIs support nutritional logging and product verification. FoodVisor also works via browser extension for online shopping, bridging ingredient transparency with personalized dietary safety through advanced NLP, OCR, and cloud deployment.

### \*Corresponding author

Padmapriya R, Department of AI & ML, M S Ramaiah Institute of Technology, Bangalore, India.

**Received:** January 08, 2026; **Accepted:** January 12, 2026; **Published:** January 25, 2026

**Keywords:** AI in Healthcare, Food Safety, Retrieval-Augmented Generation (RAG), LLaMA 2, Optical Character Recognition (OCR), Advanced Encryption Standard (AES-128)

### Introduction

In today's health-conscious world, food labels play a crucial role in guiding consumer choices. However, many labels can be intentionally misleading-products often claim to be "sugar-free" or "healthy" while including hidden sugars listed under unfamiliar names like maltodextrin, invert syrup, or fructose solids. Furthermore, complex chemical names or food codes such as E-numbers and INS additives are used, which the average consumer may not recognize or understand, masking potential health risks behind technical jargon.

This lack of clarity can be particularly dangerous for individuals with medical conditions or allergies. While some solutions aim to address allergens, few systems cater to the broader spectrum of both common and rare medical conditions that require dietary vigilance. There is a pressing need for intelligent systems that go beyond generic nutrition tracking and educate users about exactly what they are consuming-not just what the label claims.

FoodVisor addresses these gaps by offering a real-time, AI-powered food label analysis system that personalizes food safety insights based on each user's health profile. Users can securely upload personal details and health records-protected using AES-128 CBC encryption-to receive tailored feedback when analyzing food products. Ingredient data is extracted via OCR or barcode scans and processed by a hybrid Retrieval-Augmented Generation (RAG) and fine-tuned LLaMA 2 model. RAG ensures up-to-date contextual knowledge by referencing live FSSAI and other authoritative sources, while the LLaMA model, trained on a certified dataset, provides in-depth interpretation for unknown ingredients.

Additionally, FoodVisor integrates with APIs such as Bon Happtee and OpenFoodFacts to enhance meal logging and ensure product authenticity. Whether shopping in-store or online, users can simply scan a product barcode via web app or browser extension to get a full analysis based on their individual medical needs.

By combining AI, domain knowledge, and secure user profiling, FoodVisor empowers consumers to make informed, medically safe food choices-bridging the critical gap between food transparency and personalized nutrition.

### Related Works

Recent advances in artificial intelligence (AI) and classical methods have significantly improved food safety analysis, dietary recommendations, and food label processing. This section reviews key works aligned with FoodVisor's technologies, including Optical Character Recognition (OCR), Retrieval-Augmented Generation (RAG), and AI-driven food analysis, grouped by methodological or topical relevance.

Johnson and Smith leveraged large language models (LLMs) to extract chemical hazards from food safety literature, showing generative AI's potential for risk assessment [1]. Kuppasamy et al. integrated AI with contaminant analysis to enhance quality monitoring, while Thapa et al. reviewed AI's role in automating safety protocols [2,3]. These studies underscore AI's effectiveness in mitigating contamination risks, a core focus of FoodVisor's pipeline.

Sharma and Verma and Brown and Davis achieved over 90% accuracy in text extraction using Tesseract OCR, with Wilson and Thompson optimizing performance through image preprocessing [4-6]. Grubert and Gao extended this by training custom vision models for nutrition label recognition [7].

Gupta et al. and Lee and Wang developed RAG for healthcare AI, improving the accuracy of generative models [8,9]. Patel and Kumar applied RAG to structured knowledge retrieval, which is critical for FoodVisor's ingredient analysis [10]. Nguyen and Chen combined LLAMA-2 and convolutional neural networks (CNNs) for multimodal recipe generation [11].

Amiri et al. used reinforcement learning for meal planning, achieving 92% user acceptance, while Ali et al. employed federated learning for secure health data processing [12,13]. Zhang et al. and Kim and Park implemented AES-256 encryption, informing FoodVisor's data security measures [14,15]. Foundational LLM methodologies were proposed in while Roopa et al. focused on AI for additive detection [16-18].

While prior work excels in modular solutions (e.g., OCR [11]–[13], RAG [5]–[7], LLMs [1]), it often lacks real-time multimodal integration. FoodVisor addresses this gap by unifying OCR, RAG, and GEMINI models into a scalable pipeline, overcoming limitations in data fusion and scalability noted in [6,19,20].

### Proposed Work

The FoodVisor system is designed as an intelligent food safety and nutritional analysis tool that personalizes dietary decisions by integrating Optical Character Recognition (OCR), Retrieval-Augmented Generation (RAG), and fine-tuned Large Language Models (LLMs). The proposed system enables users to assess the safety of packed food items based on individual health profiles while also providing detailed nutritional breakdowns. It accepts inputs from physical food labels, barcodes, or online product pages, processes them through a modular pipeline, and delivers customized feedback in real time.

Figure 1 presents a high-level workflow, outlining the user journey from registration to food safety evaluation and nutritional feedback. Figure 2 illustrates the complete system architecture, highlighting the modular design and interaction between components.

#### A. User Registration and Profile Setup

The system begins with a secure onboarding process where users provide essential health details such as age, weight, height, gender, activity level, and dietary preferences. Additionally, users can upload medical records and allergy reports in PDF or image formats. For convenience, FoodVisor also allows natural language entries for example, users may type phrases like "I am diabetic."

All sensitive health data is encrypted using the AES-128 encryption standard in Cipher Block Chaining (CBC) mode with PKCS7 padding to ensure complete confidentiality. This encrypted information including uploaded documents, natural language inputs, and structured health data is stored securely in a MySQL relational database.

Once the health profile is set up, the user is ready to receive fully personalized food safety analysis and dietary recommendations tailored to their medical context.

#### Data Ingestion and OCR Module

FoodVisor accepts data through both image-based and structured digital sources. Users can scan packed food packaging using their web camera, or utilize the web extension to extract barcodes from online product listings. Barcode inputs are mapped to product metadata such as ingredient lists and nutrition facts via the OpenFoodFacts API. In cases where a physical label is scanned, the OCR engine extracts textual information which is further processed using an NLP pipeline to isolate the ingredients.

This hybrid ingestion approach ensures that both physical and online packed food items can be analyzed efficiently. Once the ingredient list is extracted, each ingredient is passed individually into the hybrid AI engine for further evaluation.

#### Ingredient Safety Analysis Engine

This module forms the core intelligence of FoodVisor and is responsible for analyzing ingredients for potential health risks. It employs a hybrid AI approach that combines a Retrieval-Augmented Generation (RAG) system with a fine-tuned LLAMA 2 model, ensuring comprehensive and up-to-date analysis.

Once ingredients are extracted from packed food items, they are passed one by one into the RAG system. This component references a dynamic document store containing regulatory data from sources such as the FDA, WHO, FSSAI, USDA, and peer-reviewed scientific literature. The documents are embedded and stored using FAISS, and new updates from these organizations can be added to the document store without retraining the model, making RAG highly dynamic and adaptive to real-time changes.

If a safety-related match is found, the RAG system generates a context-specific explanation using a Gemini Flash LLM. However, if RAG is unable to provide reliable information, the system falls back to a fine-tuned LLaMA 2 model (7B parameters). This model has been trained on a curated dataset of food safety cases using QLoRA and PEFT techniques. It is capable of independently analyzing unfamiliar or ambiguous ingredients and classifying them as "safe" or "risky" based on known risk patterns, regulatory bans, and health impact assessments.

#### Personalized Verdict Generation

Based on the ingredient evaluation and the user's encrypted medical history, the system checks for contraindications, allergens, and chronic condition triggers. Each ingredient is flagged accordingly, and a final verdict is generated with visual indicators and LLM-backed explanations tailored to the user's health profile.

#### Macronutrient Analysis Module

Beyond safety evaluation, FoodVisor also offers comprehensive nutritional monitoring. Individuals can analyze the macronutrient composition of any food item—including regional specialties and global cuisines, such as Indian home-made dishes. Nutritional data for foods is retrieved via the Bon Happtee API, ensuring accurate and diverse dietary insights.

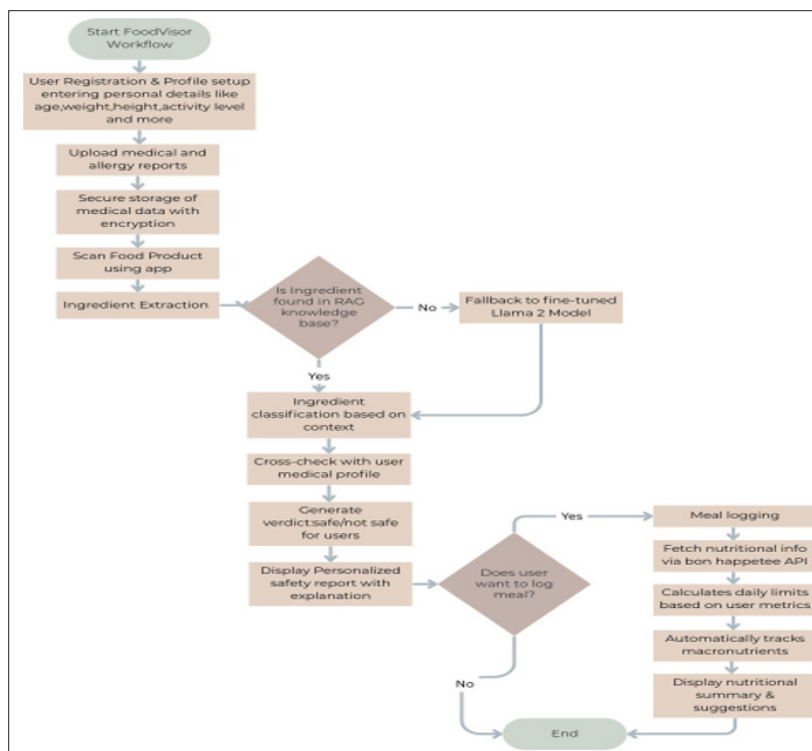


Figure 1: End-To-End Workflow of the Foodvisor System from Input to Output

The system calculates users' daily nutritional requirements using the Mifflin-St Jeor Equation, then identifies potential imbalances by comparing their intake against personalized dietary needs. Based on this analysis, FoodVisor provides tailored recommendations, suggesting healthier alternatives or adjustments to align with the user's health goals and preferences.

## Methodology

### User Profile Construction and Secure Data Handling

The user profile setup pipeline in FoodVisor is designed to intake both structured and unstructured health information, while ensuring secure, persistent storage through robust encryption and schema design. The architecture integrates deterministic LLM-based parsing, OCR processing for medical reports, and AES-128-CBC encryption for sensitive fields.

**Structured and Unstructured Input Data Processing:** The user registration flow begins with the intake of both structured and unstructured health-related inputs. Structured fields such as age, weight, height, gender, activity level, and dietary preferences are directly mapped to columns in the `user_data` table within the MySQL database. These fields are inserted as-is without any preprocessing.

In contrast, unstructured inputs such as free-form text entries like "I am allergic to peanuts and eggs" or "Diagnosed with PCOS and hypothyroidism." require semantic parsing to extract clinically relevant entities. These entries are passed to a Gemini-based LLM pipeline, where a Named Entity Recognition (NER)-style prompt identifies and classifies key medical entities. For example:

- 'Diagnosed with PCOS and hypothyroidism.' → {'medical\_conditions': ['PCOS', 'Hypothyroidism']}
- 'I have an egg allergy.' → {'allergies': ['Egg']}

This natural language processing (NLP) mechanism ensures that unstructured sentences are normalized into a consistent, structured schema suitable for downstream storage and analytics.

Additionally, users can upload multiple PDF or image-based medical reports, such as diagnostic summaries or allergy profiles. These documents are parsed using Tesseract OCR, which extracts raw visible text. However, OCR outputs often lack syntactic structure or consistency. Therefore, the extracted text is also passed through the same LLM-based structuring pipeline used for free text. This ensures that both typed sentences and scanned reports converge into a common schema, allowing unified handling of all unstructured health data.

This dual-pipeline strategy ensures comprehensive capture and standardization of health metadata, supporting both digital and scanned document sources and enabling robust personalization and safety assessment features later in the system.

**Data Encryption and Secure Storage:** To enforce end-to-end data confidentiality, parsed medical and allergy information is encrypted using symmetric encryption before persistence. The current implementation employs the cryptography.fernet library, which internally utilizes AES-128 in Cipher Block Chaining (CBC) mode, augmented with PKCS7 padding and HMAC-SHA256 for integrity validation.

- **Key Initialization:** A 32-byte secret key is pulled from the environment variable ENCRYPTION\_KEY and passed to the Fernet cipher initializer.
- **Data Encryption:** Text entries produced by the clean\_and\_structure\_extracted\_data() routine are encoded to UTF-8 bytes and encrypted via cipher.encrypt().
- **Storage:** The encrypted payloads are stored as BLOB fields in the health\_data table within the MySQL database. Empty inputs are safely handled with zero-byte encryption.

A dedicated retrieval endpoint /health\_data supports on-demand decryption and serves results in structured JSON format.

**MySQL Schema Design and Storage Policy:** All user- related data is stored in a MySQL relational database. Sensitive medical and allergy information is securely encrypted, while non-sensitive fields (e.g., demographics, activity level, maintenance calories, etc.) are stored in plaintext for fast access.

The health\_data table schema is outlined as follows:

```
CREATE TABLE health_data (
id INT AUTO_INCREMENT PRIMARY KEY,
```

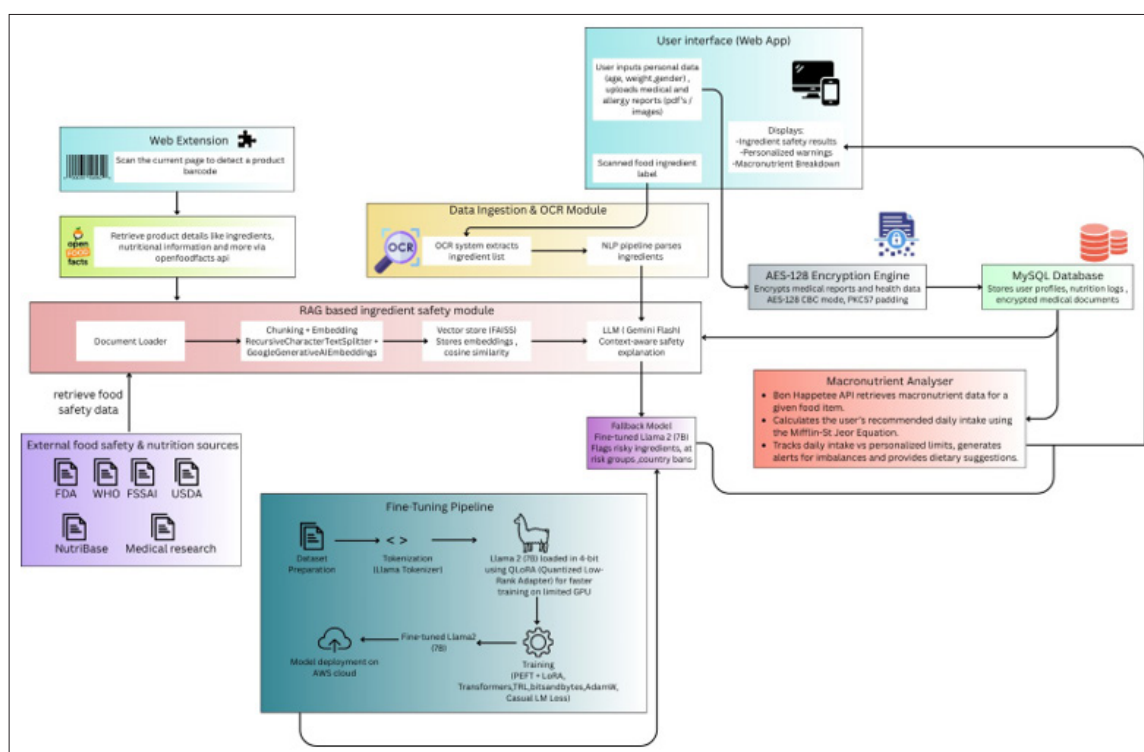


Figure 2: System Architecture Diagram Showing Component Interactions and Data Flow

```
user_id INT NOT NULL,
simplified_allergy BLOB,
simplified_medical BLOB,
created_at TIMESTAMP DEFAULT
CURRENT_TIMESTAMP,
FOREIGN KEY (user_id)
REFERENCES user_data(id));
```

- user\_id enforces relational linkage to the core user profile.
- simplified\_allergy and simplified\_medical store AES-encrypted BLOBs.
- Timestamps enable versioning and chronological sorting of updates.

This schema enforces strong referential integrity while preserving data confidentiality where required. All transient files used during processing are deleted immediately post-ingestion to minimize attack surfaces.

With secure user profiles established and personalized dietary parameters defined, the system is now primed to process visual inputs. We next outline the OCR and data ingestion pipeline, which extracts structured nutritional information from food labels and barcodes.

### OCR and Data Ingestion Pipeline

This section details the end-to-end optical character recognition (OCR) and data ingestion architecture designed for the food

safety application. The pipeline handles complex food label and barcode inputs by integrating advanced pre-processing, OCR-based extraction, post-processing using large language models, and standardized data formatting. All stages are implemented in a Pythonic backend environment, using OpenCV, pyzbar, Tesseract, and Gemini-1.5-flash.

**Image Pre-processing:** To ensure robust and accurate OCR in real-world conditions-such as low lighting, skewed orientations, or noisy backgrounds-the system implements an extensive image enhancement pipeline within the /processImageAdjustments endpoint. The pipeline is built using OpenCV (cv2) and operates on images received as base64-encoded strings via JSON.

The pre-processing steps are as follows:

- **Image Decoding:** The image\_to\_base64 function decodes base64 strings into a NumPy array using np.frombuffer, which is then converted to a color image using cv2.imdecode.
- **Brightness and Contrast Adjustment:** Poor lighting and contrast are rectified by applying linear intensity transformation through cv2.convertScaleAbs. The transformation is parameterized as:

$$\text{output} = \alpha \cdot \text{image} + \beta \quad (1)$$

where  $\alpha$  adjusts contrast and  $\beta$  controls brightness. These parameters are user-defined or default to  $\alpha = 1, \beta = 0$ .

- **Cropping:** If provided, the image is cropped to a region of interest using array slicing based on frontend input: img[y:y+height, x:x+width].
- **Rotation (Skew Correction):** Skewed text is corrected via affine rotation. A rotation matrix is calculated using cv2.getRotationMatrix2D, and the transformation is applied with cv2.warpAffine:

$$M = \begin{bmatrix} \cos \theta & -\sin \theta & (1 - \cos \theta) \cdot c_x + \sin \theta \cdot c_y \\ \sin \theta & \cos \theta & -\sin \theta \cdot c_x + (1 - \cos \theta) \cdot c_y \end{bmatrix} \quad (2)$$

where  $\theta$  is the user-specified rotation angle and (cx, cy) is the image center.

The final adjusted image is stored temporarily and optionally returned to the frontend in base64 format for real-time preview.

**Barcode Decoding:** Barcode recognition and product data retrieval are handled in the /processBarcode end-point using the pyzbar library. The decoding sequence includes:

- **Grayscale Conversion:** The base64-decoded image is converted to grayscale using cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY) to improve detection fidelity.
- **Barcode Extraction:** pyzbar.decode is applied to extract barcode symbology (e.g., EAN-13, UPC-A) and corresponding data.
- **External API Integration:** The decoded barcode is queried against the OpenFoodFacts API to retrieve product metadata including name, brand, ingredients, and nutrients.
- **Structured Formatting:** Relevant fields are normalized into a JSON object, isolating macronutrient values (e.g., sugars\_100g, proteins\_100g, energy-kcal\_100g) for downstream safety checks.

**Text Post-processing:** Raw OCR output is susceptible to spelling inconsistencies, special character artifacts, and lexical noise. To

address this, we utilize the Gemini-1.5-flash LLM to normalize ingredient lists through contextual spell correction and formatting. An example is given below:

**Raw OCR:** “sugar, salt, E100, citric acid”

**Post-Processed:** “sugar, salt, E100, citric acid”

This step ensures that subsequent ingredient analysis modules operate on clean, semantically accurate text inputs. The LLM operates at low temperature ( $T = 0.3$ ) to maximize determinism, and is constrained by a 2000-token limit to maintain throughput.

The clean, structured output from this ingestion pipeline forms the foundation for intelligent food analysis. In the subsequent section, we present our ingredient analysis module, which evaluates health and safety risks using a hybrid AI system.

### Ingredient Safety Analysis (Hybrid AI Engine)

This module is designed to deliver fast, regulation-aware, and medically informed safety assessments of food ingredients. It combines a high-recall Retrieval-Augmented Generation (RAG) system, powered by Gemini Flash, with a fine-tuned LLaMA 2 model trained using QLoRA and PEFT. This hybrid approach blends policy-aligned factual retrieval with generative reasoning to produce comprehensive, user-friendly explanations.

**RAG Pipeline Architecture:** The core of the pipeline is a Retrieval-Augmented Generation (RAG) architecture that utilizes Google Generative AI Embeddings to encode regulatory documents and ingredients queries. These embeddings are stored in a FAISS (Facebook AI Similarity Search) vector index optimized for high-speed retrieval using L2 (Euclidean) distance similarity ranking. The L2 distance between a query vector  $q$  and document vector  $d$  is computed as:

$$L2(q, d) = \sqrt{\sum_{i=1}^n (q_i - d_i)^2} \quad (3)$$

where  $n$  is the embedding dimension, and  $q_i, d_i$  represent the  $i$ -th components of the query and document vectors respectively. This metric enables efficient nearest-neighbor search in the high-dimensional embedding space.

- **Document Corpus:** Official publications from WHO, FSSAI, FDA, and Codex Alimentarius form the primary corpus. These documents are converted to machine-readable format, chunked (max 512 tokens), embedded using GoogleGenerativeAIEmbeddings, and indexed using FAISS.
- **Real-Time Update Mechanism:** New regulations are automatically processed and embedded into the FAISS store weekly. The ingestion pipeline involves PDF parsing, chunking, and embedding. This mechanism is the primary motivation for using RAG, allowing updates to reflect new regulations without retraining the LLaMA model.
- **Query Processing:** Ingredient safety queries (e.g., “Is E171 safe?”) are converted into embedding vectors and used to retrieve top-k relevant document chunks. These chunks are then passed to a generative LLM for response synthesis.

**Gemini Flash LLM Role and Prompt Engineering:** The Gemini-1.5-Flash model is deployed as the generative reasoning layer on top of the retriever. It serves multiple roles:

- **Contextual Synthesis:** Synthesizes up to 8 retrieved chunks, distilling ingredient-specific safety data from complex

regulatory language.

- **Compliance Enforcement:** All outputs explicitly cite regulation sources and align with verified guidelines.
- **Prompt Strategy:** A structured prompt template is employed with a low temperature ( $T = 0.3$ ) to ensure deterministic, factual outputs:

Using the provided regulatory context excerpts, please formulate a comprehensive and accurate response to the user's query.

#### Adhere to the following guidelines:

1. Base your response strictly on the given context
2. If the context does not contain sufficient information to answer the question definitively, respond: "Insufficient regulatory evidence available."

**Question:** {input}

**Context Excerpts:** {context}

**Response:**

**LLaMA 2 Fine-tuning with QLoRA and PEFT:** The second layer of the system is a generative reasoning model that delivers descriptive, health-educational outputs when ingredient is unavailable in RAG vector-store. The base model used is NousResearch/llama-2-7b-chat-hf, fine-tuned using QLoRA on curated domain-specific data.

**Dataset Construction:** A structured CSV dataset of 7,000 ingredient entries was manually curated after referring to various regulatory sources (FDA, WHO, FSSAI) and validated by a certified dietitian. Each entry includes:

- Regulatory status (e.g., "banned by EU")
- Population-specific safety concerns (e.g., diabetics, pregnant individuals)
- Toxicological profile (e.g., carcinogenicity, allergenicity)

**Format:** <s>[INST] Give safety assessment about the food ingredient 'Corn Syrup'. Highlight if it's harmful for any group of people, and mention if it's banned in any countries for toxic nature. [/INST] High in empty calories; linked to obesity, insulin resistance, and fatty liver disease when consumed excessively. Safe in small amounts for healthy individuals but avoid regular use. Not recommended for diabetics or those with metabolic disorders. </s>

This structured approach enables the model to generate informative, nuanced content about both common and uncommon ingredients.

**QLoRA Training Setup:** Training was performed on an NVIDIA T4 GPU using Low-Rank Adaptation with 4-bit quantization for efficient fine-tuning.

#### LoRA Configuration

- rank = 64
- alpha = 16
- dropout = 0.1

#### Training Arguments:

- Epochs: 2
- Batch size: 8
- Gradient accumulation: 2 (effective batch size 16)
- Learning rate:  $1 \times 10^{-4}$
- Mixed Precision: FP16
- Gradient checkpointing: Enabled
- Weight decay: 0.001
- Trainer: SFTTrainer from HuggingFace TRL trainer = SFTTrainer(model=model, train\_dataset=ds, peft\_

config=peft\_cfg, args=training\_args)

The fine-tuned LLaMA 2 model is deployed on an AWS cloud instance equipped with NVIDIA L40S GPUs, enabling fast inference latency (~7s). The model weights are publicly available at padmapriya/foodvisor.

**Flagging Logic for Risk Classification:** Unlike binary classifiers, our fine-tuned model provides granular, descriptive safety assessments:

#### Examples:

- "Safe for all except individuals with diabetes."
- "Extremely harmful for everyone due to its carcinogenic properties."
- "Conditionally approved in the USA; banned in the EU."

The outputs are designed to educate consumers with medically relevant rationales rather than providing binary verdicts. This nuanced approach enables more informed dietary decisions.

#### Verdict Generation and Personalization Logic

This component constitutes the heart of the system's decision-making pipeline, combining authoritative regulatory insights with personalized health context to generate comprehensive, adaptive food ingredient safety assessments. The decision logic is operationalized through a hierarchical evaluation architecture leveraging a hybrid AI model composed of a Retrieval-Augmented Generation (RAG) pipeline and a fine-tuned LLaMA 2 language model.

**Hierarchical Decision Framework:** The system initiates safety assessment with the RAG pipeline, which indexes and retrieves content from trusted regulatory sources (e.g., FSSAI, WHO, FDA) using a FAISS vector store. If a query matches an entry marked as banned or restricted, the pipeline immediately assigns the "Highly Risky" tag, citing the regulatory document as the source. No user-specific context is applied in this case, as such bans are universally applicable. For instance, querying "Titanium Dioxide" yields: "Highly risky: Banned in food products by the EU due to potential genotoxicity (Source: EFSA, 2021)."

If no restriction is found, the query is passed to a fine-tuned 7B parameter LLaMA 2 model. This model incorporates user health data such as diabetes, nut allergies, or kidney disease dynamically injected via prompts using structured records from a MySQL database. It enables personalized reasoning that reflects individual dietary risks.

**Verdict Classification Schema:** To aid interpretability and UX design, the system broadly categorizes outputs under three flexible, human-readable tags. However, these tags are supported by detailed, descriptive content generated by the LLaMA 2 model, often going beyond strict classification:

**Fully Safe for Everyone:** This tag is used when an ingredient has no associated health risks, contraindications, or regulatory limitations for any demographic or health condition. Both RAG and LLaMA 2 confirm the absence of risk. Example: "Malic Acid is fully safe for all populations. No known adverse effects or usage restrictions (Source: FDA)."

**Safe Except for Specific Conditions:** This tag denotes ingredients that are generally safe but may present risks to individuals with specific health profiles. The LLaMA 2 model tailors the output

using health data from the MySQL database. The output often includes quantitative safe limits for consumption (e.g., grams per day), improving user clarity and helping with portion control. For example:

A user with diabetes querying “Corn Syrup” receives: “Safe except for specific conditions: Limit to less than 10g/day due to insulin resistance risk in diabetics.”

**A healthy user receives:** “Safe to consume in moderation: No known issues for healthy individuals.”

Additionally, if an ingredient is correlated with long-term health risks such as inflammation, hyperactivity, or metabolic syndrome, even in the absence of current illness, the model issues precautionary notes to encourage moderation. Example: “Sodium benzoate is safe in approved doses but has been linked to hyperactivity in children. Limit intake in sensitive populations.”

**Highly Risky:** If the RAG pipeline retrieves evidence of a regulatory ban, the ingredient is immediately labeled highly risky. This overrides personalization, as the restriction is considered universal. Example: “Raspberry ketone is banned by FSSAI due to toxicity concerns (Source: FSSAI, July 2018).”

Thus, the system’s output schema is not rigidly bounded by these three labels but rather uses them as high-level summaries. Each verdict is augmented with explanatory insights, safe consumption thresholds, and secondary risk indicators to offer a multi-dimensional safety perspective.

**Output Formatting:** The final safety assessment is structured in a JSON object for frontend compatibility and clear user interpretation. The output comprises:

- **Ingredient:** Name of the queried compound.
- **Functional Description:** Its primary use in food (e.g., sweetener, emulsifier, colorant).
- **General Safety Assessment:** Evaluation for the population at large.
- **Personalized Safety Assessment:** Modified evaluation based on user-specific health factors.
- **Recommended Daily Limit:** Quantitative guidance (e.g., “Less than 10g/day for diabetic individuals”).
- **Reasoning:** Concise explanation citing biomedical or regulatory rationale.
- **Citation:** Source of truth (e.g., “FSSAI July 2018”).

This format facilitates transparency and empowers end-users to make informed decisions about their food intake. By incorporating both universal risk classification and individualized dietary safety, the system ensures that safety assessments are scientifically grounded, regulatory-compliant, and health-contextualized.

### Macronutrient Retrieval and Dietary Recommendation

This module enables personalized macronutrient tracking by retrieving nutritional values for ingredients and meals, computing user-specific macronutrient requirements, and detecting deviations from recommended intake levels. The system supports dietary decision-making by providing real-time insights into carbohydrate, protein, and fat consumption based on user input and health profiles.

**Bon Happendee API:** Request Structure and Response Parsing: The Bon Happendee Food and Nutrition Data API is used to obtain detailed macronutrient profiles (carbohydrates, proteins, fats) for

individual ingredients and dishes, spanning Indian and global cuisines. The API features smart alias-based recognition to handle regional variations in food naming—for example, both dosa and dosai yield equivalent results—enhancing coverage across diverse dietary inputs.

**Request Structure:** The API is accessed via HTTP POST requests to <https://api.bonhappendee.com/v1/nutrition>. Each request includes a JSON payload with the queried food item, optional portion size, and regional context. An example payload is:

```
{
  "query": "brown rice", "portion_size": "100g", "region": "India"
}
```

**Response Parsing:** The API returns a structured JSON response with caloric values and macronutrient content. For instance:

```
{
  "status": "success", "data": {
    "name": "brown rice", "portion_size": "100g",
    "calories": 111, "macronutrients": {
      "carbohydrates": 23.0,
      "proteins": 2.6,
      "fats": 0.9}}
}
```

Responses are parsed using Python’s json library. Macronutrient values and calorie data are extracted and cached in an SQLite database, reducing latency and minimizing redundant API calls. The system supports batch processing of meal components (e.g., “brown rice, lentils, spinach”) and aggregates macronutrient profiles for composite dishes.

### Mifflin-St Jeor Computation: Formula and Parameters

User-specific caloric needs are calculated using the Mifflin-St Jeor equation, widely recognized for its accuracy in estimating Basal Metabolic Rate (BMR). This serves as the foundation for determining individualized macronutrient thresholds.

BMR Formula:

• **Male**  
 $BMR = (10 \times \text{weight}[\text{kg}] + (6.25 \times \text{height}[\text{cm}]) - (5 \times \text{age}[\text{years}]) + 5 \quad (4)$

• **Female**  
 $BMR = (10 \times \text{weight}[\text{kg}] + (6.25 \times \text{height}[\text{cm}]) - (5 \times \text{age}[\text{years}]) - 161 \quad (5)$

**TDEE Calculation:** Total Daily Energy Expenditure (TDEE) is computed by applying an activity factor:

- Sedentary:  $BMR \times 1.2$
- Lightly active:  $BMR \times 1.375$
- Moderately active:  $BMR \times 1.55$
- Very active:  $BMR \times 1.725$

User demographic and fitness information—such as age, height, weight, gender, and activity level—is retrieved from the /userprofile endpoint. For instance, a 30-year-old female weighing 65 kg, with a height of 165 cm and moderate activity level would yield:

$$BMR = 1370.25 \text{ kcal/day}$$

$$TDEE = 1370.25 \times 1.55 = 2123.89 \text{ kcal/day}$$

TDEE values are further used to derive macronutrient distribution goals under standard dietary ratios (e.g., 50% carbohydrates, 30% proteins, 20% fats), unless otherwise specified by user-defined

fitness goals. All calculations are implemented using NumPy within the Flask backend and executed in real time (~50ms), with caching enabled for repeat queries.

**Imbalance Detection Logic:** The imbalance detection logic compares the macronutrient composition of user-reported meals with their daily recommended limits, highlighting over- or under-consumption in real time.

**Target Derivation:** Macronutrient limits are computed from the user’s TDEE. For a daily intake of 2123.89 kcal, the targets would be:

- Carbohydrates:  $(2123.89 \times 0.5) \div 4 = 265.49$  g
- Proteins:  $(2123.89 \times 0.3) \div 4 = 159.29$  g
- Fats:  $(2123.89 \times 0.2) \div 9 = 47.20$  g

**Actual Intake Calculation:** Macronutrient values for consumed food items are aggregated. For example, a meal of 200g brown rice and 100g lentils provides:

- Carbohydrates:  $46g + 25.7g = 71.7$  g
- Proteins:  $5.2g + 9.0g = 14.2$  g
- Fats:  $1.8g + 0.3g = 2.1$  g

**Deviation Flagging:** Deviations are flagged if the intake of any macronutrient exceeds or falls below the target by more than 20%. For the above example:

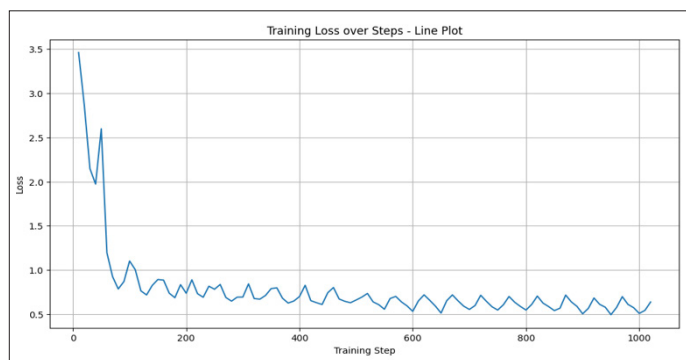
- **Carbohydrates:**  $71.7 \div 265.49 = 27\% \rightarrow$  within acceptable range
- **Proteins:**  $14.2 \div 159.29 = 8.9\% \rightarrow$  flagged as low
- **Fats:**  $2.1 \div 47.2 = 4.4\% \rightarrow$  flagged as low

This feedback is computed and returned to the user via the frontend. Configuration for threshold percentages and default TDEE values is stored in a JSON configuration file.

## Results and Discussion

### Fine-tuning Loss Curve

The training loss obtained during LLaMA 2 fine-tuning is shown in Figure 3, which depicts a steep decline in early steps followed by gradual convergence. The final training loss reached 0.7785 over 1024 steps, with a total runtime of approximately 48 minutes (2890.68 seconds). The process achieved 5.67 train samples per second, 0.354 steps per second, and a total of  $4.99 \times 10^{16}$  FLOPs, indicating efficient and stable model convergence.



**Figure 3:** Training Loss Curve for Llama 2 Fine-Tuning Showing Convergence Over 1024 Steps

### Evaluation of LLaMA 2-Based Classifier on Unseen Ingredients

Although the LLaMA 2 model was fine-tuned on a dataset with descriptive safety assessments (e.g., risks, conditions, and country-level bans), we opted to evaluate it using binary classification

outputs (“safe” or “risky”) during testing. This decision was driven by the fact that free-form descriptive answers are subjective and hard to quantify, making it infeasible to compute objective performance metrics like accuracy, precision, recall, or F1-score. In contrast, binary classification enables clear, measurable comparisons using confusion matrices and classification reports. Furthermore, using a common response format allows for fair benchmarking between the base and fine-tuned models, avoiding ambiguity in judging answer correctness. We constructed a small unseen test set containing ingredients not present in the training data. The goal was to assess whether the model could generalize its understanding of food safety concepts to new entries. The baseline (non-finetuned) LLaMA 2 model often produced uncertain or incomplete responses sometimes explicitly saying “unsure,” despite being prompted to classify ingredients.

A comparative summary of both models’ classification performance is provided in Table 1.

The fine-tuned model demonstrates strong generalization, flagging risky ingredients with higher recall and F1-score while completely eliminating uncertain outputs. The base model, lacking domain alignment, showed hesitation (labeling a few samples as “unsure”) and lower sensitivity to risky items. Thus, fine-tuning significantly enhanced the model’s robustness and decision confidence.

This evaluation confirms the effectiveness of the fine-tuned LLaMA 2 model.

**Table 1: Model Performance Comparison with Key Metrics**

Metric	Base Model	Fine-tuned Model
Accuracy	94.95%	98.99%
Precision (risky)	100.0%	100.0%
Recall (risky)	83.3%	95.0%
F1 Score (risky)	90.9%	97.44%
Misclassifications (risky)	5 (3 wrong, 2 unsure)	1

### Real-World Evaluation: Using RAG + LLaMA 2

To evaluate the personalized safety classification capabilities of our hybrid model, we tested it on the packaged food product Cadbury Gems, configuring the user profile as pregnant with heart disease. Figure 4 illustrates a subset of the output generated by our system, showcasing the retrieved and generated safety assessments for four representative ingredients.

The system effectively contextualized each ingredient against the specified medical conditions. For instance:

- Sugar was flagged for its association with gestational complications and its potential to worsen cardiovascular risk.
- Hydrogenated oils were identified as hazardous due to trans fats, which elevate LDL cholesterol and are especially problematic during pregnancy.
- Whey permeate powder was marked with caution, considering its lactose content and potential digestive impact, which is common during pregnancy.
- E102 (Tartrazine) was strongly discouraged, with the system referencing EU warning requirements and international bans due to its possible reproductive toxicity.

These responses reflect the system’s ability to integrate retrieved evidence with condition-specific reasoning to generate targeted and informative health warnings.

## Personalized Meal Logging and Macronutrient Tracking

To demonstrate the system's dynamic logging and nutrition tracking capabilities, we configured a test scenario with a user profile: a 60-year-old female, weighing 68 kg, 160 cm tall, with a lightly active lifestyle and a goal of maintaining overall health. Based on these parameters, the system computes personalized daily macronutrient goals using standard nutritional guidelines.

Figure 5 illustrates the interactive interface used for food logging and macronutrient tracking. The user logs food items such as white rice (cooked) and sambhar, and the system automatically:

- Retrieves nutritional information (calories, carbohydrates, fats, and proteins) from a food database
- Updates consumed quantities in real time
- Visualizes macronutrient progress through a radial plot segmented by macronutrient category

This functionality enables users to make informed dietary decisions by monitoring their nutritional intake against personalized goals. For instance, the system shows that the user has consumed 27g of carbohydrates, helping them track remaining allowances effectively. The integration of contextualized logging with a responsive dashboard supports practical health management by simplifying meal planning.

## Conclusion and Future Scope

This paper presented a hybrid AI system that leverages retrieval-augmented generation (RAG) alongside a fine-tuned LLaMA 2 model for ingredient-level food safety classification. The system demonstrated strong performance in identifying risky ingredients from unseen inputs, significantly improving over the base model. Additionally, it enables personalized macronutrient tracking by adapting to user profiles and accommodates regional food variations effectively.

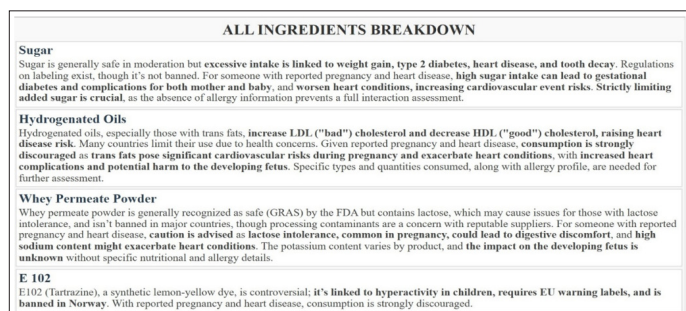


Figure 4: Result Snapshots of Ingredient-Specific Safety Warnings Generated by Our AI System

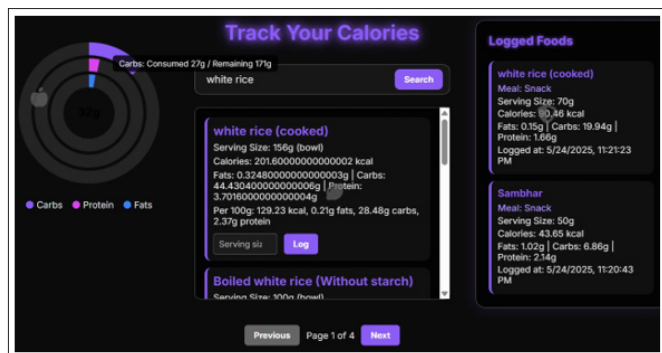


Figure 5: Macronutrient Tracking Dashboard

The integrated pipeline provides a reliable, user-friendly solution for real-time dietary assessment, bridging both health safety and nutritional awareness in a single platform.

Looking ahead, the platform's scope will expand beyond food to include cosmetics, pharmaceuticals, and personal care products, offering personalized health warnings across diverse categories. To tackle barcode counterfeiting, planned features include verification of FSSAI registration numbers, font analysis on packaged labels, and layout-based checks to detect forged products. These advancements aim to evolve the system into a comprehensive consumer safety assistant, empowering users with trusted, real-time insights across all daily-use product domains.

## Acknowledgment

We gratefully acknowledge Dr. Hema Arvind, certified dietician, for her expert validation of our food safety dataset used in fine-tuning the LLaMA 2 model. Her domain expertise was instrumental in ensuring the accuracy and reliability of our safety classifications. This project was also supported by the Ramaiah Institute of Technology Alumni Funds, whose contribution helped bring it to fruition.

This work was supported by the Ramaiah Institute of Technology Alumni Funds.

## References

1. Johnson M, Smith D (2024) AI-driven extraction of chemical food safety hazards from scientific literature using large language models. *J. Food Saf* 12:45-60.
2. Kuppusamy S, Kumar R, Gupta A, Zhang P (2023) Integrating AI in food contaminant analysis: Enhancing quality and environmental protection. *J. Environ. Sci. Health B* 8: 112-130.
3. Samad S, Rahman T, Ibrahim M (2023) Smartphone apps for tracking food consumption and recommendations: Evaluating artificial intelligence-based functionalities, features, and quality of current apps. *J. Health Inform* 15: 201-220.
4. Amiri M, Rad FS, Li J (2024) Delighting palates with AI: Reinforcement learnings triumph in crafting personalized meal plans with high user acceptance. *Trends Food Sci. Technol* 5: 34-52.
5. Gupta A, Sharma B, Wang C, Chen D (2023) Enhancing medical AI with retrieval-augmented generation. *Artif. Intell. Med* 7: 89-104.
6. Lee B, Wang C (2023) Retrieval-augmented generation for generative artificial intelligence in healthcare. *J. Healthcare Inform. Res* 9: 145-162.
7. Patel R, Kumar S (2024) Using retrieval-augmented generation to build context for data-driven stories. *Appl. Artif. Intell* 11: 77-95.
8. (2023) Meta AI Research, LLaMA 2: Open foundation and fine-tuned chat models. arXiv preprint <https://arxiv.org/abs/2307.09288>.
9. Nguyen T, Chen L (2024) Integrating convolutional neural networks and Meta LLaMA2 for recipe generation. *Front. Nutr* 6: 88-105.
10. Roberts J (2023) Fine-tuning LLaMA 2: A practical guide. *IEEE Trans. Neural Netw. Learn. Syst* 4: 56-72.
11. Sharma K, Verma P (2023) Implementation of Tesseract OCR and bounding box for text extraction on food nutrition labels. *J. Food Eng* 10: 23-40.
12. Brown L, Davis M (2023) Optical character recognition and Tesseract to read nutritional facts. *Food Sci. Nutr* 7: 155-170.
13. Wilson R, Thompson S (2024) Improved Tesseract optical

- character recognition performance on food labels. *Trends Food Sci. Technol* 9: 67-84.
14. Zhang H, Liu Y, Chen W (2023) A selective encryption algorithm based on AES for medical information systems. *J. Med. Syst* 12: 45-62.
  15. Martinez E, Lopez F (2024) Secure medical data communication: A comprehensive approach with AES encryption. *Cybersecurity* 8: 33-50.
  16. Kim D, Park Y (2023) Improving Firebase BaaS service security in counseling chat applications: AES-256 and CBC approach for end-to- end encryption. *J. Healthcare Inform. Res* 11: 78-95.
  17. Ali AA, Khan B, Ahmed C, Wilson D (2024) Securing electronic health records using blockchain-enabled federated learning for IoT-based smart healthcare. *J. Healthcare Inform. Res* 8: 134-152.
  18. Grubert O, Gao L (2023) Recognition of nutrition facts labels from mobile images. *IEEE Trans. Image Process* 32: 5678-5692.
  19. RNK, Ameena BB, Siddiqua A (2023) Analysis of food additives in packaged food materials using AI and ML. *Int. J. Food Sci. Technol* 58: 2105-2120.
  20. Thapa A, Shrestha B, Gurung C, Rai D (2023) A comprehensive review on artificial intelligence assisted technologies in food industry. *Trends Food Sci. Technol* 135: 1-18.

**Copyright:** ©2026 Padmapriya R, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.