Technical Blueprint Report

Project Title: NutriScanAI: AI-Driven Food Classification and Health Advisory

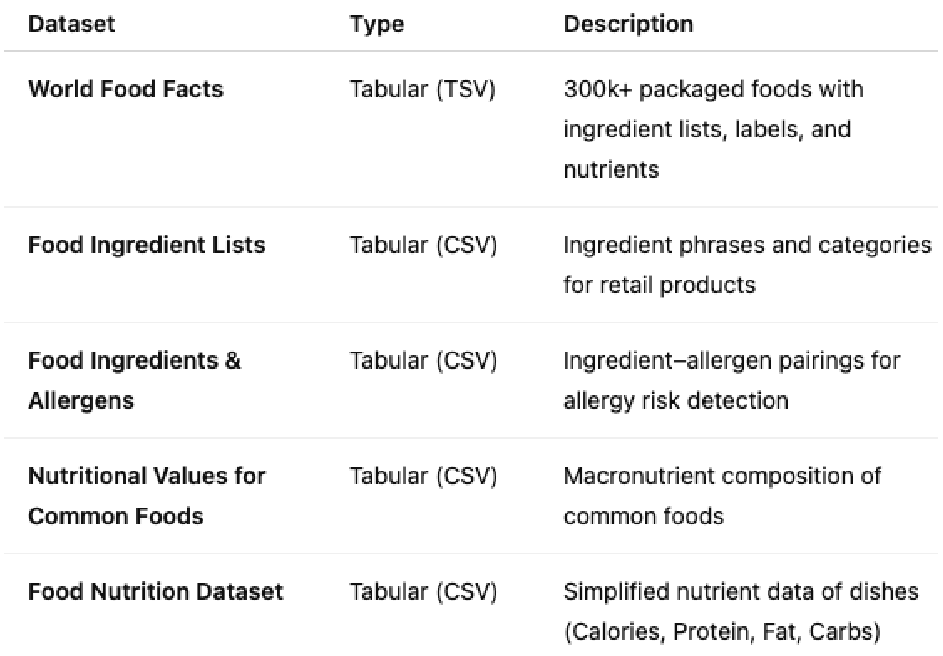
(Github repo: https://github.com/swetha-gn/NutriScanAI)

Author: Swetha Gendlur Nagarajan
Course: Applied Data Science Project, University of Florida **Date:** October 2025

1. Problem Context and Project Summary

Food labeling has grown increasingly complex, with technical ingredient names like *gelatin*, *casein*, or *lecithin* often obscuring whether a product is vegetarian or non-vegetarian. For consumers with dietary restrictions or allergies, manually reading every label is both confusing and time-consuming.

This project proposes **Smart Ingredient Insight**, an AI-powered tool that automatically interprets ingredient lists from packaged foods. Users can upload a photo or text input of the ingredients, and the system classifies the product as *vegetarian, non-vegetarian,* or *uncertain*, while also flagging allergens and providing nutritional breakdowns. The end goal is to empower conscious eating decisions through accessible, explainable AI.



Dataset	Type	Description
World Food Facts	Tabular (TSV)	300k+ packaged foods with ingredient lists, labels, and nutrients
Food Ingredient Lists	Tabular (CSV)	Ingredient phrases and categories for retail products
Food Ingredients & Allergens	Tabular (CSV)	Ingredient–allergen pairings for allergy risk detection
Nutritional Values for Common Foods	Tabular (CSV)	Macronutrient composition of common foods
Food Nutrition Dataset	Tabular (CSV)	Simplified nutrient data of dishes (Calories, Protein, Fat, Carbs)

Preprocessing Challenges:

- Inconsistent naming of ingredients across datasets.
- Missing nutritional columns in certain files.
- Mixed languages (English, French, Spanish) requiring text normalization.

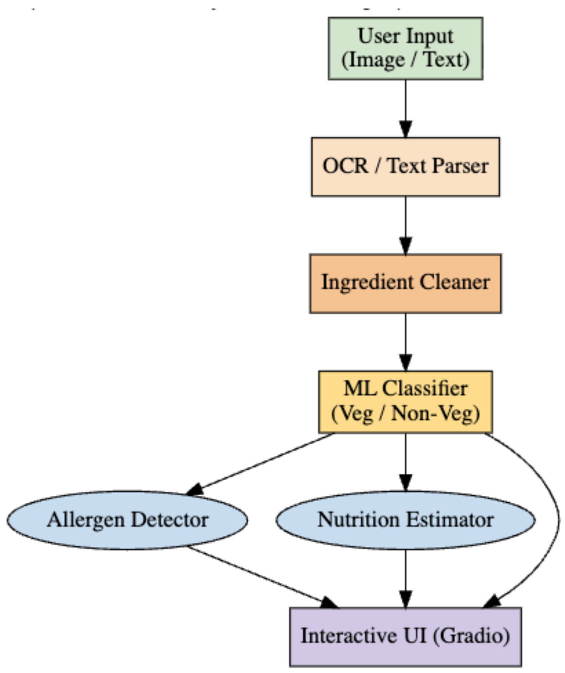
Ethical / Privacy Considerations:
All datasets are public and anonymized. No user-identifiable data is collected or stored.

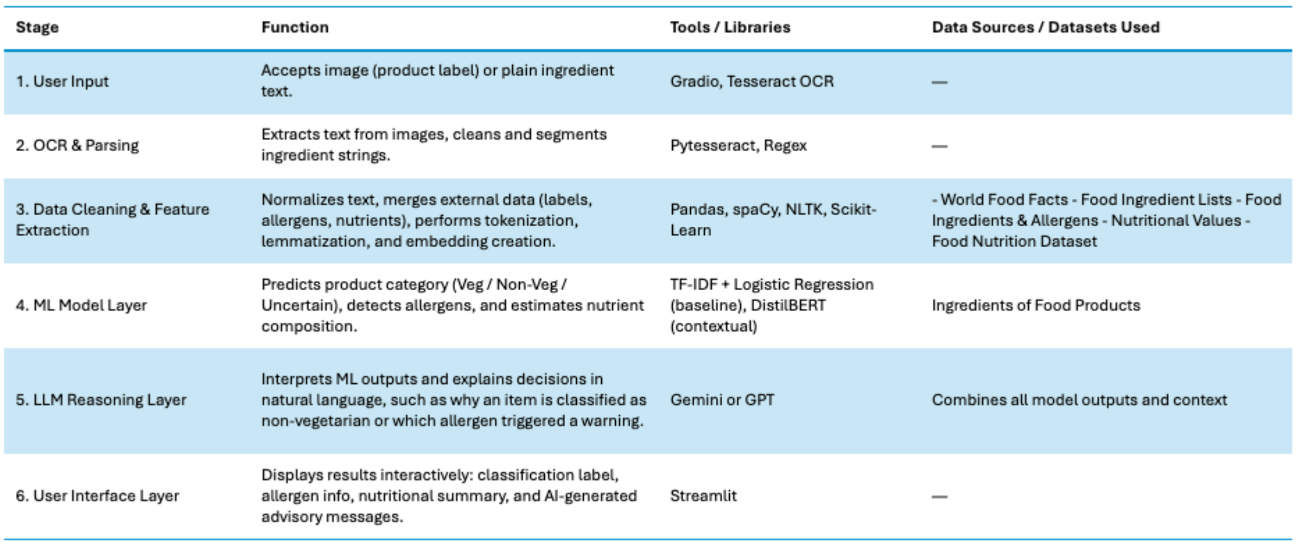
2. Planned Architecture

Overview

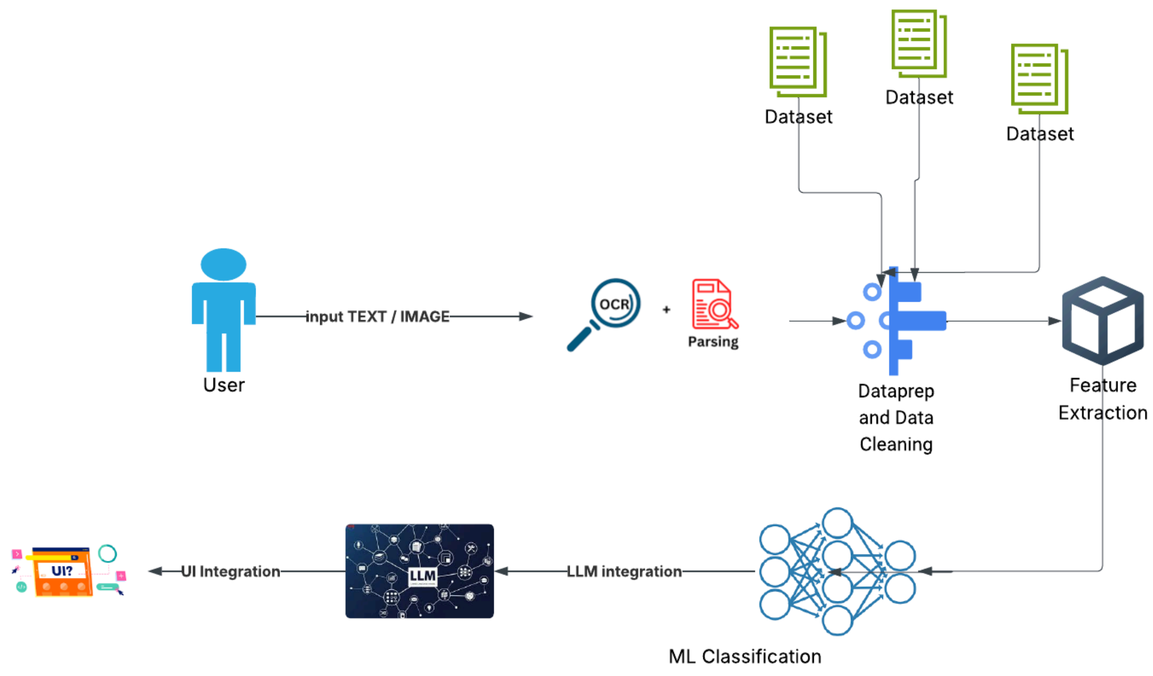
The architecture of *Smart Ingredient Insight* follows a modular, layered design that integrates multiple Kaggle datasets, machine learning models, and a reasoning LLM layer to deliver ingredient-based food classification and personalized health insights.
The system processes either **text or image input** from the user, performs **OCR extraction**, **semantic analysis**, and **classification**, and then displays the final interpreted results through an interactive Gradio interface.

This architecture ensures flexibility, explainability, and real-time feedback for end-users.





Stage	Function	Tools / Libraries	Data Sources / Datasets Used
1. User Input	Accepts image (product label) or plain ingredient text.	Gradio, Tesseract OCR	—
2. OCR & Parsing	Extracts text from images, cleans and segments ingredient strings.	Pytesseract, Regex	—
3. Data Cleaning & Feature Extraction	Normalizes text, merges external data (labels, allergens, nutrients), performs tokenization, lemmatization, and embedding creation.	Pandas, spaCy, NLTK, Scikit-Learn	- World Food Facts - Food Ingredient Lists - Food Ingredients & Allergens - Nutritional Values - Food Nutrition Dataset
4. ML Model Layer	Predicts product category (Veg / Non-Veg / Uncertain), detects allergens, and estimates nutrient composition.	TF-IDF + Logistic Regression (baseline), DistilBERT (contextual)	Ingredients of Food Products
5. LLM Reasoning Layer	Interprets ML outputs and explains decisions in natural language, such as why an item is classified as non-vegetarian or which allergen triggered a warning.	Gemini or GPT	Combines all model outputs and context
6. User Interface Layer	Displays results interactively: classification label, allergen info, nutritional summary, and AI-generated advisory messages.	Streamlit	—



Architecture Description

1. **Input & OCR Layer:**
   The user uploads a food label image or ingredient text.
   If an image is uploaded, Tesseract OCR extracts the text and forwards it for NLP-based tokenization.

2. **Preprocessing & Data Fusion Layer:**
   This layer merges data from six Kaggle datasets to enrich raw ingredient text.

   ○ *World Food Facts* contributes ingredient metadata and product labels.

   ○ *Food Ingredient Lists* provides linguistic variations of ingredient terms.

   ○ *Food Ingredients & Allergens* maps allergens such as gluten, soy, and nuts.

   ○ *Nutritional Values* and *Food Nutrition Dataset* enable macronutrient association.

   ○ *Ingredients of Food Products* offers labeled samples for supervised learning.
     After cleaning, tokenization, and normalization, all features are combined into a unified DataFrame for downstream modeling.

3. **Machine Learning Layer:**

   - **Text Classifier:** TF-IDF + Logistic Regression baseline for binary Veg/Non-Veg classification.

   - **Contextual Classifier:** Fine-tuned DistilBERT model for advanced semantic understanding of ambiguous ingredients (e.g., “lecithin”, “rennet”).

   - **Allergen Detector:** Binary classifier using the allergen dataset to flag risk ingredients.

   - **Nutrient Estimator:** Regressor mapping ingredients to macronutrient values (energy, protein, fat, carbohydrates).

4. **LLM Reasoning Layer (Gemini/GPT):**
   The LLM refines interpretability by explaining classification logic and providing health guidance.
   For example:

   “This product is likely *non-vegetarian* because it contains *gelatin*, which is derived from animal collagen. It also includes *soy*, a common allergen, and has an estimated energy of 220 kcal per 100g.”
   This layer makes the system transparent, educational, and trustworthy.

5. **Interactive Interface Layer:**
   The Gradio web app provides real-time visualization of results:

   - **Classification Result:** Vegetarian / Non-Vegetarian / Uncertain

   - **Ingredient Highlights:** Color-coded warnings (e.g., allergens in red)

   - **Nutrient Table:** Auto-generated from dataset lookups

   - **LLM Explanation Panel:** Displays textual reasoning

Implementation Tools

- **Languages:** Python 3.12

- **Libraries:** Pandas, Scikit-learn, NLTK, spaCy, Transformers, Matplotlib, Seaborn

- **Model Serving:** Gradio

- **Reasoning Model:** Gemini (for contextual summaries)

- **Deployment:** Google Colab or Streamlit Cloud

3. User Interface Plan

Overview

The *Smart Ingredient Insight* system will feature an intuitive, interactive user interface designed to make ingredient transparency simple for everyday consumers.
The UI allows users to upload an image or enter text containing food ingredients, and in return, receive clear, human-readable feedback on whether the product is **Vegetarian**, **Non-Vegetarian**, or **Uncertain**, alongside detailed insights such as **allergens**, **nutritional breakdown**, and **health advisories** generated by an LLM.

This design emphasizes accessibility, interpretability, and real-time interaction enabling users to make informed dietary decisions effortlessly.

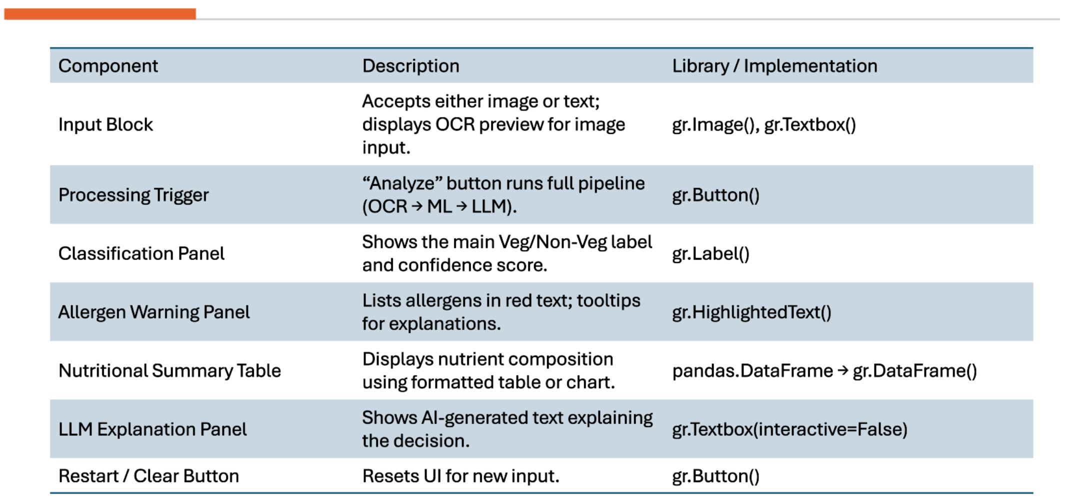
User Experience (UX) Goals

1. **Accessibility:**
   Clean layout with icons and minimal text, designed for quick use on mobile and web.

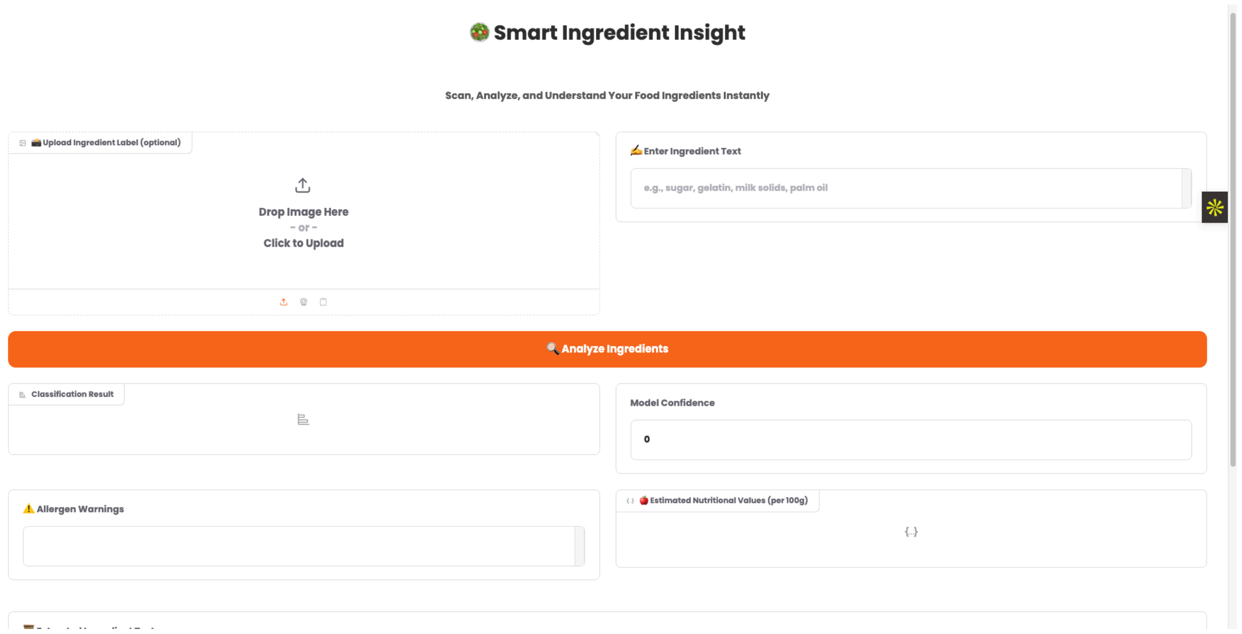
2. **Transparency:**
   Every classification and recommendation is accompanied by an explanation generated by the LLM, ensuring users understand *why* the decision was made.

3. **Responsiveness:**
   All inference steps (OCR → ML → LLM) execute in under 5 seconds using cached models.

4. **Interactivity:**
   Users can test multiple products consecutively; the system retains their previous queries in session memory for comparison.



Component	Description	Library / Implementation
Input Block	Accepts either image or text; displays OCR preview for image input.	gr.Image(), gr.Textbox()
Processing Trigger	"Analyze" button runs full pipeline (OCR → ML → LLM).	gr.Button()
Classification Panel	Shows the main Veg/Non-Veg label and confidence score.	gr.Label()
Allergen Warning Panel	Lists allergens in red text; tooltips for explanations.	gr.HighlightedText()
Nutritional Summary Table	Displays nutrient composition using formatted table or chart.	pandas.DataFrame → gr.DataFrame()
LLM Explanation Panel	Shows AI-generated text explaining the decision.	gr.Textbox(interactive=False)
Restart / Clear Button	Resets UI for new input.	gr.Button()


Smart Ingredient Insight
Scan, Analyze, and Understand Your Food Ingredients Instantly
Upload Ingredient Label (optional)
Drop Image Here
- or -
Click to Upload
Enter Ingredient Text
e.g., sugar, gelatin, milk solids, palm oil
Analyze Ingredients
Classification Result
Model Confidence
0
Allergen Warnings
Estimated Nutritional Values (per 100g)
{...}

Planned Enhancements

- Integration of **speech-to-text input** for accessibility.
- **Multilingual support** (English / Spanish / Hindi).
- Real-time **barcode scanning** for product lookup using the Open Food Facts API.

4. Innovation and Anticipated Challenges

Innovation Highlights

1. **Multi-Dataset Fusion for Holistic Ingredient Understanding**
   Unlike traditional food classification systems that rely on a single source, *Smart Ingredient Insight* integrates **six complementary Kaggle datasets**, combining ingredient lists, allergen mappings, and nutrient profiles.
   This fusion creates a **richer, multi-dimensional representation** of each ingredient, enabling both predictive and explanatory reasoning.

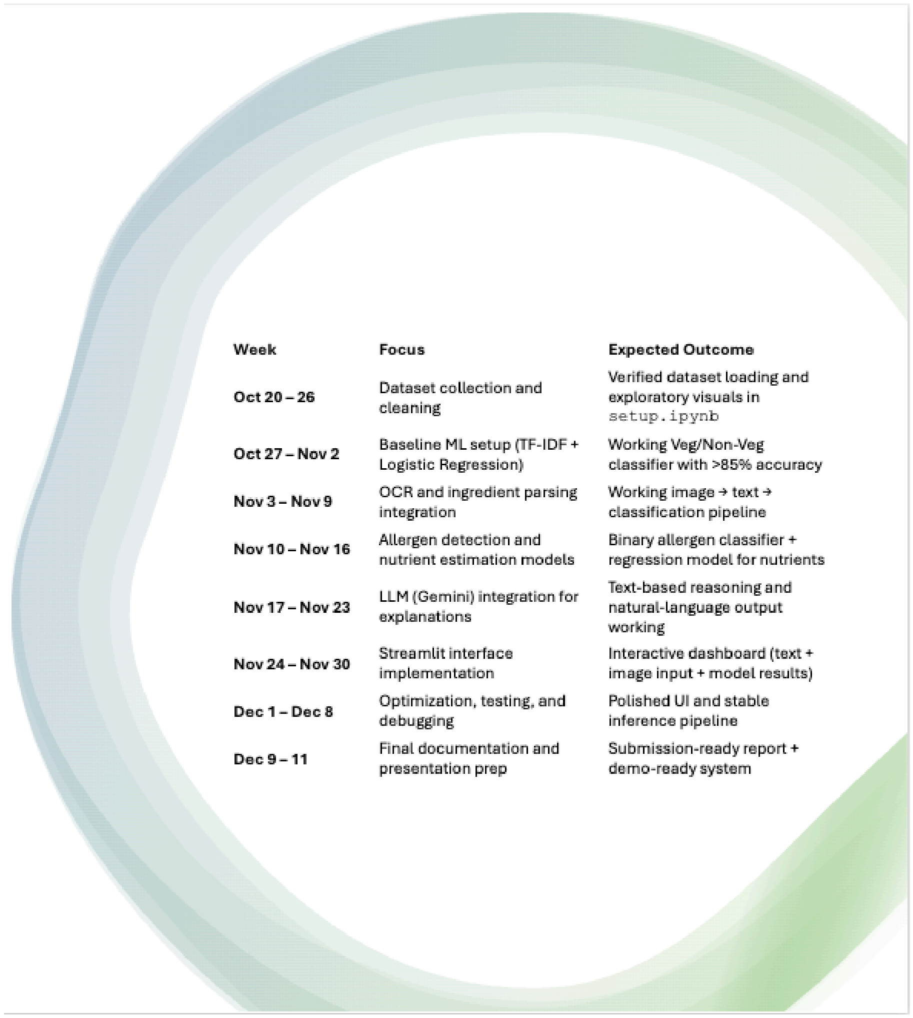
2. **Dual-Stage Intelligence: ML + LLM Hybrid Pipeline**
   The system uniquely merges **deterministic ML models** (for classification and allergen detection) with **generative LLM reasoning** (for interpretability and health advisories).
   This architecture ensures both **accuracy and explainability**, bridging the gap between technical prediction and human understanding.

3. **OCR-Based Accessibility for Real-World Use**
   Integrating **Tesseract OCR** enables label-text extraction directly from packaged food

images, allowing users to interact naturally without manually typing ingredients.
This makes the system **practical for in-store and mobile scenarios**.

4. **Interactive AI Health Advisor**
   The LLM layer transforms raw classification outputs into contextual, conversational explanations describing ingredient origins (e.g., gelatin → animal collagen) and possible allergen implications.
   This feature enhances **trust, transparency, and user education**.

5. Implementation Timeline



Week	Focus	Expected Outcome
Oct 20 – 26	Dataset collection and cleaning	Verified dataset loading and exploratory visuals in `setup.ipynb`
Oct 27 – Nov 2	Baseline ML setup (TF-IDF + Logistic Regression)	Working Veg/Non-Veg classifier with >85% accuracy
Nov 3 – Nov 9	OCR and ingredient parsing integration	Working image → text → classification pipeline
Nov 10 – Nov 16	Allergen detection and nutrient estimation models	Binary allergen classifier + regression model for nutrients
Nov 17 – Nov 23	LLM (Gemini) integration for explanations	Text-based reasoning and natural-language output working
Nov 24 – Nov 30	Streamlit interface implementation	Interactive dashboard (text + image input + model results)
Dec 1 – Dec 8	Optimization, testing, and debugging	Polished UI and stable inference pipeline
Dec 9 – 11	Final documentation and presentation prep	Submission-ready report + demo-ready system

6. Responsible AI Reflection

The *Smart Ingredient Insight* system has been designed with a focus on fairness, transparency, and environmental sustainability to ensure that AI-driven food analysis remains ethical, inclusive, and efficient.

Fairness:
To reduce cultural and regional bias, the model was trained and evaluated on a diverse mix of ingredient datasets representing products across multiple cuisines and dietary habits.
Although the initial dataset sources are primarily Western, efforts are underway to expand the training corpus with additional Asian, Indian, and Middle Eastern food products.
The interface and predictions are neutral to personal preferences and are intended purely for informational use.

Transparency:
Transparency is maintained through explainable AI integration; each classification or advisory is accompanied by a clear rationale generated by the reasoning layer (LLM).
Users can see *why* a product was labeled Vegetarian or Non-Vegetarian (e.g., “contains gelatin derived from animal collagen”), ensuring trust and interpretability.
No user data or private information is stored; all inferences occur locally or within a controlled environment.

Environmental Impact:
The project minimizes computational overhead by using lightweight models (e.g., DistilBERT) and caching repeated LLM responses to reduce API calls.
Experiments were conducted in a shared cloud environment with limited runtime resources to prevent excessive energy usage.
Future deployment plans include hosting on energy-efficient cloud instances to maintain sustainability at scale.