

Smart Diet Planning System

Project Documentation & Technical Report

Submitted by: [Your Names Here]

Date: January 2026

1. Project Overview

The Smart Diet Planning System is a web-based application designed to generate personalized nutrition plans using Artificial Intelligence and Machine Learning. Unlike generic diet apps, this system tailors meals based on specific medical conditions (Diabetes, Hypertension, CVD, PCOS, Obesity) while considering user biometrics (Height, Weight, BMI, Age).

2. Technology Stack

Component	Technology Used	Purpose
Language	Python 3.x	Core logic and backend processing.
Backend	Flask	Web framework to handle API routes and server logic.
Frontend	HTML5, CSS3, JavaScript	User Interface, Charts, and Dynamic Interactions.
Database	SQLite3	Storing User credentials and Diet History.
ML Models	Scikit-Learn (Random Forest)	Classifying foods as Safe/Unsafe based on nutrients.
Generative AI	Google Gemini API	AI Chatbot for nutritional queries.
Reporting	ReportLab	Generating downloadable PDF diet plans.

3. Implementation Details

3.1 Backend Logic (`app.py`)

The backend is built using Flask. The core logic resides in the `generate_smart_daily_plan` function. Key algorithms implemented:

- Disease Filtering:** Uses ML models to filter the dataset. If a user has Diabetes, the system checks the 'diabetes_label' column predicted by our Random Forest model.
- Obesity Logic:** If 'Obesity' is selected, the system automatically subtracts 500 calories from the TDEE (Total Daily Energy Expenditure) to create a caloric deficit, capped at a minimum of 1200 kcal for safety.
- Dynamic Portion Scaling:** The system calculates a scaling factor based on the target calories. For Diabetics, portions are strictly capped (0.8x - 1.2x) to prevent blood sugar spikes. For Obesity, portions are reduced (down to 0.7x).
- Database Management:** SQLite is used to store 'Users' (Authentication) and 'Diets' (History). The system saves Height, Weight, and BMI along with the generated JSON plan.

3.2 Machine Learning (`train_models.py`)

We trained 5 separate Random Forest Classifiers for each disease.

Features Used: Calories, Protein, Carbs, Fat, Sugar, Sodium, GI.

Target: Labeled data (Recommended / Not Recommended).

Performance: The models achieve high accuracy in filtering unsafe foods.

3.3 Frontend Logic (*index.html*)

The frontend is a dynamic Single Page Application (SPA) feel.

Chart.js: Used to visualize the nutrient breakdown (Protein, Carbs, Fat) for every meal card.

History Mode: Users can view past approved diets. A sophisticated 'Flagging System' was implemented in JS to pause real-time updates while the user is viewing history.

Doctor Workflow: Users submit plans via API. Doctors view them in a separate panel (Expert Panel) to Approve/Reject.

4. Key Features & Innovations

- **BMI Auto-Calculation:** Height and weight inputs automatically calculate BMI and Daily Calorie Needs.
- **Smart Portion Control:** The algorithm renames food items (e.g., 'Large Portion') based on the calculated scale factor.
- **PDF Generation:** A custom PDF report is generated using the ReportLab library, organizing the weekly plan into a printable format.
- **AI Chatbot:** Integrated Google Gemini API to answer general nutrition questions.

5. Conclusion

This project successfully integrates medical constraints with nutritional data. By combining Machine Learning for food safety and a rule-based algorithm for meal assembly, it provides a safe, personalized, and scalable solution for diet planning.