Advancing Nutrition Science through GeminiAI

Team Name: Dynamic DuAI

P.Gangothri(23wh1a0423)
S.Deena(23wh1a0426)
U.Sakshitha(23wh1a0443)

Phase-1: Brainstorming & Ideation

1.Objective: To develop a web-based application that provides users with detailed nutritional insights and generates personalized meal plans

2.Problem Statement: Many individuals struggle to create balanced, satisfying meal plans that align with their dietary needs, health conditions, and personal preferences. Without proper guidance, they may face nutritional imbalances or difficulty in maintaining a healthy diet.

3.Proposed Solution:

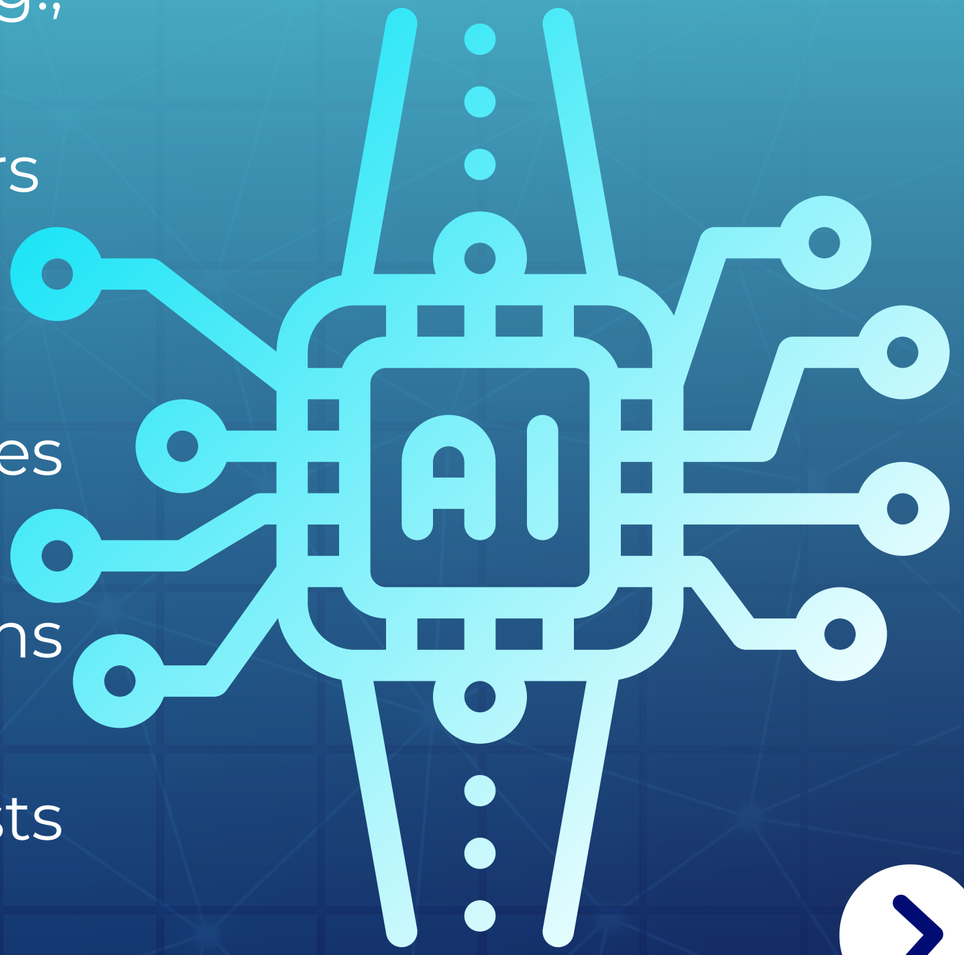
- Provide instant, comprehensive nutritional data on food item
- Generate personalized meal plans based on user inputs
- Offer recipes and grocery lists for a structured, convenient and nutritious diet.

Target Users:

- Health-conscious individuals
- People with dietary restrictions (e.g., diabetics, vegetarians, athletes)
- Fitness enthusiasts & weight managers
- Nutritionists and dietitians

Expected Outcome:

- Users make informed dietary choices with instant nutrition insights.
- AI-generated custom meal plans ensure balanced nutrition and variety.
- Easy access to recipes & grocery lists enhances meal preparation.



Phase-2: Requirement Analysis

Functional Requirements

- Users select dietary restrictions and meal types.
- Meal options are displayed with YouTube links.
- Generates a grocery list with nutritional info and saves it as a text fil

Technical Requirements

- Tech Stack: Python (web browser, file handling).
- Data Handling: Hardcoded meal and grocery list data in dictionaries
- Interface: CLI-based user input.

Constraints

- Limited meal options, manually updated.
- No personalization (calories, allergies).
- Runs only locally (no web or mobile version).

C**hallenges**

- Scalability: Needs API integration for meals & nutrition.
- Data Accuracy: Requires reliable nutrition sources.
- User Experience: CLI limits accessibility, a web UI is needed.

Phase-3: Project Design

System Architecture

- Client (CLI): Displays options, takes user input, shows meal details.
- Application Layer (Python): Handles logic, retrieves meal/grocery data, opens YouTube links, saves grocery list.
- External Services: Uses webbrowser for YouTube; future API integration for meals/nutrition.

User Flow

1. User selects dietary restriction → meal type → meal.
2. System displays meal details & opens YouTube link.
3. Grocery list (if available) is generated and saved.

UI/UX Considerations

- Simple Numbered Menu: Easy selection.
- Clear Prompts & Feedback: User-friendly instructions.
- Error Handling: Prevent invalid inputs.

Phase-4: Project Planning (Agile Methodologies)

Day 1: Core Functionality (MVP)

- Set up CLI-based meal selection.
- Store meal & grocery data in dictionaries.
- Implement YouTube link opening (webbrowser).
- Generate and save a grocery list (grocery_list.txt).
- Test user input handling.

Day 2: Refinements & Enhancements

- Improve UI (better prompts, formatting, emojis).
- Add error handling (invalid inputs, missing data).
- Expand meal options & grocery list details.
- Final testing & debugging.

Stretch Goal: Basic web UI (Flask).

Phase-5: Project Development

Technology Stack

- Python (CLI-based)
- Libraries: webbrowser (YouTube links), os (file handling)
- Data Storage: Hardcoded dictionaries (future: JSON/API)

Development Process (2-Day Agile Sprint)

Day 1: Build CLI, meal selection, YouTube links, grocery list generation.

Day 2: Improve UI, add error handling, expand meals, final testing.

Challenges & Fixes

Challenge	Fix
Hardcoded Data	Use JSON/API for scalability
CLI-Based UI	Future upgrade to Flask/Django
Invalid Input Crashes	Add input validation
Limited Nutrition Info	Integrate external API

Phase-6: Functional & Performance Testing

Functional Testing:

- Ensures correct meal selection, YouTube link opening, and grocery list generation.
- Handles invalid inputs gracefully and prevents crashes.
- Displays warnings for missing grocery data instead of failing.

Performance Testing:

- Meal selection and file generation complete in under 1 second.
- Memory usage remains low (<50MB), ensuring efficiency.

THANK YOU