

Advancing Nutrition Science Through Geminiai

A Gen AI-powered initiative by Google Research

A Research Project by Google Research – Health AI Division

Submitted by:

Ashutosh Gunjal

Palak Jethwani

Mohith P

Saiyam jain

Under the Guidance of: Sai Kiran

June 2025

SmartInternz

Hyderabad, Telangana



1. Acknowledgement

We would like to express our sincere gratitude to our mentor for their invaluable guidance and support throughout this project. Their insights and encouragement played a pivotal role in shaping our understanding and execution of the work.

We are equally grateful to our institution and faculty for providing the resources and environment conducive to learning and innovation.

This project, *Advancing Nutrition Science Through GeminiAI*, would not have been possible without the combined efforts of our entire team. Each member contributed significantly, demonstrating dedication, creativity, and technical excellence.

We also acknowledge the use of powerful platforms and tools such as Google Gemini, Firebase, the USDA FoodData Central API, and open-source technologies like Flask and React, which collectively enabled us to bring this project to life.

2. Executive Summary

Advancing Nutrition Science Through GeminiAI is a collaborative initiative that leverages the capabilities of Generative AI to transform personal nutrition management. Our project, **NutriGen**, is an AI-driven web application designed to serve as a personal nutrition assistant, offering users personalized meal plans, detailed nutritional insights, and real-time virtual coaching.

The application combines data from the USDA FoodData Central API, secure user management via Firebase, and AI-powered responses using Google Gemini models. Built using a modern tech stack (Flask, React, Firebase, Tailwind), NutriGen ensures scalability, security, and user-friendliness.

Through tailored recommendations and interactive features, NutriGen empowers individuals to make informed dietary choices, promoting healthier lifestyles with AI as a constant companion. This project showcases the convergence of health science and advanced artificial intelligence in delivering real-world impact.

3. Introduction

NutriGen is a sophisticated, AI-driven web application engineered to serve as a personal nutrition assistant. The core mission of the project is to empower users to

make informed and healthy dietary choices by leveraging the advanced capabilities of Google's Generative AI.

The application delivers a suite of tools including instant, comprehensive data on macronutrients (protein, fat, carbohydrates), micronutrients (vitamins, minerals), and caloric content for a vast array of food items. By offering features like tailored meal planning, dynamic nutritional analysis, and a virtual nutrition coach, NutriGen aims to make nutritional science accessible, understandable, and actionable for everyone.

4. Project Overview

This section outlines the foundational knowledge required for the project and provides a high-level view of its structural organization.

4.1 Prior Knowledge

To fully grasp the architecture and implementation of NutriGen, a foundational understanding of the following technologies is essential:

- **Backend Development:** Proficiency in Python and the Flask micro-framework is necessary to understand the server-side logic, API creation, and business rule implementation.
- **Frontend Development:** Knowledge of JavaScript/TypeScript, React, and the Vite build tool is crucial. Experience with state management using Redux Toolkit, routing with React Router, and styling with TailwindCSS is also required.
- **Database and Authentication:** Familiarity with Google Firebase, specifically Firestore (a NoSQL database) for data storage and Firebase Authentication for user management, is key.
- **API and AI Integration:** A basic understanding of RESTful API principles, JWT-based authentication, and the fundamentals of prompt engineering for interacting with large language models (LLMs) like Google Gemini is needed.

4.2 Project Structure

The project is organized into two main directories, `client` and `server`, representing a clear separation of concerns between the frontend and backend.

- **server/ (Backend):**
 - Built with **Flask**, this is the application's engine.
 - `app.py`: The main application file that defines all API endpoints for authentication, meal planning, nutrition insights, and AI chat.

- `firebase_config.py`: Manages the connection to the Firebase Admin SDK and handles the creation and verification of custom JSON Web Tokens (JWTs) for secure communication.
 - `AI/`: This directory contains the logic for interacting with the Google Generative AI models.
 - `mealPlanner.py`: Generates personalized 7-day meal plans using a detailed, structured prompt.
 - `chat.py`: Powers the virtual coach, using user health data to provide tailored responses.
 - `NutriInsights.py`: A dedicated module for communicating with the external USDA FoodData Central API to fetch nutritional information.
- **client/ (Frontend):**
 - A modern single-page application (SPA) built with **React** and **Vite**.
 - `src/pages/`: Contains the main view components for each route, such as `DashboardPage.jsx`, `MealPlannerPage.jsx`, and `VirtualCoachPage.jsx`.
 - `src/components/`: Holds reusable UI elements like `Button.jsx`, `Card.jsx`, and `Input.jsx`.
 - `src/app/`: The core of the client-side state management, built with **Redux Toolkit**. It includes slices for `authSlice.js`, `mealPlanSlice.js`, etc., which manage application state and asynchronous API calls.
 - `src/api/`: Contains service modules (`authService.js`, `mealPlanService.js`, etc.) that use **Axios** to communicate with the Flask backend APIs.
 - `src/routes/`: Manages application navigation with `AppRoutes.jsx`, defining public and private routes.
-

5. Literature Survey

5.1 Existing Problem

Modern lifestyles present significant challenges to maintaining a healthy diet. The primary issues identified are:

- **Complexity of Meal Planning:** Many individuals struggle to create healthy, satisfying, and varied meal plans that align with their specific dietary needs,

allergies, health conditions, and taste preferences. This often leads to repetitive, nutritionally imbalanced diets.

- **Lack of Nutritional Transparency:** While understanding the nutritional content of food is crucial for making healthy choices, this information is often inaccessible or difficult to interpret. Consumers need a simple way to get detailed information about macronutrients, micronutrients, and calories for their meals.
- **High Cost of Expert Guidance:** Access to personalized advice from a certified nutritionist is often expensive and time-consuming, placing it out of reach for many. This creates a barrier for individuals seeking to improve their health through diet.

5.2 Proposed Solution

NutriGen is designed to directly address these challenges by leveraging AI to provide a cost-effective, accessible, and personalized solution.

- **Tailored Meal Planning:** The application generates a complete seven-day meal plan, including recipes and grocery lists, based on a user's detailed profile. The AI ensures the plan is nutritionally balanced, enjoyable, and respects all user-specified restrictions.
- **Dynamic Nutritional Insights:** NutriGen provides instant nutritional breakdowns for any food item. Users can search for foods and receive detailed information, empowering them to make conscious decisions and track their intake against their health goals.
- **Virtual Nutrition Coaching:** The application democratizes access to nutritional expertise through an AI-powered virtual coach. This interactive feature provides personalized advice, answers questions, and offers support, helping users stay motivated and achieve long-term health improvements.

6. Ideation and Brainstorming

The core idea behind NutriGen was to create a single, unified platform that addresses the primary friction points in personal nutrition management. The brainstorming process centered on a few key principles:

- **Personalization as a Priority:** Generic advice is less effective. The platform had to be built around the user's unique profile, including their physical attributes, lifestyle, goals, and allergies. This led to the detailed registration process.
- **AI as the Engine:** To provide deep personalization at scale, Generative AI was chosen as the core technology. Instead of rule-based systems, an LLM could

handle complex, nuanced user requirements for both meal planning and conversational coaching.

- **Data-Driven Insights:** To be credible, the application needed to be backed by reliable data. The USDA FoodData Central database was selected as the source for all nutritional information.
 - **Seamless User Experience:** The interface needed to be intuitive and engaging. This led to the design of distinct sections for the Dashboard, Meal Planner, Insights, and Coach, all accessible from a clean navigation bar. Features like a meal-logging streak were added to encourage consistent user engagement.
-

7. Map and Canvas

7.1 User Flow Map

The user's journey through the application is structured logically, as defined by the application's routes.

1. New User

- Landing Page → Register Page (fills out profile and credentials) → Dashboard (upon successful registration)

2. Returning User

- Landing Page → Login Page → Dashboard

3. Authenticated User Journey

From the Dashboard, the user can:

- Log a meal to update their streak
- Navigate to the Meal Planner to generate or view their weekly plan
- Navigate to Insights to search for food nutrition data
- Navigate to the Virtual Coach for AI-powered advice
- Navigate to their Profile to view their health details

> All private routes are protected and will redirect to Login if the user is not authenticated.

7.2 LEAN CANVAS

Problem	Solution	Key Metrics	Unique Value Proposition
1. Meal planning is difficult and time-consuming. 2. Nutritional information is hard to access. 3. Expert advice is expensive.	An AI-powered web app providing: 1. Personalized meal plans. 2. Instant food nutrition analysis. 3. 24/7 virtual coaching.	- User engagement (daily active users) - Meal plan generation rate - Number of food lookups - Meal logging streak length	Your all-in-one AI nutrition assistant. Get hyper-personalized meal plans, instant food insights, and 24/7 coaching to achieve your health goals effortlessly.
Unfair Advantage	Channels	Customer Segments	Cost Structure & Revenue
- Advanced prompt engineering for highly tailored outputs. - Integration of user data into a single, cohesive AI context. - Use of a reliable, comprehensive food database (USDA).	- Web application (direct access) - Search engine optimization (organic traffic) - Social media marketing	- Health-conscious individuals - People with specific dietary goals (weight loss, muscle gain) - Users with dietary restrictions (allergies, vegetarian)	Cost: Server hosting, external API fees (Google AI, USDA). Revenue : (Currently free) Potential for premium subscription model.

8. Requirement Analysis

8.1 Functional Requirements

- **F1. User Authentication:**

Users must be able to register with their credentials and a detailed health profile. They must be able to log in and log out. The system must keep users logged in across sessions.

- **F2. Secure Access:**

All user-specific data and features must be protected and accessible only after successful authentication.

- **F3. Profile Management:**
Users must be able to view their registered name, email, and health details on a profile page.
- **F4. Meal Plan Generation:**
The system must generate a personalized 7-day meal plan based on the user's stored health profile.
- **F5. Meal Plan Viewing:**
Users must be able to view their generated meal plan, organized by day and meal type.
- **F6. Food Nutrition Search:**
Users must be able to search for any food item and receive a list of matching results from the USDA database.
- **F7. Detailed Food Analysis:**
Users must be able to select a food item from the search results to view its detailed nutritional information, including macronutrients and micronutrients, presented with a pie chart.
- **F8. Virtual Coaching:**
Users must be able to have a text-based conversation with an AI nutrition coach. The chat must support conversation history.
- **F9. Meal Logging & Streak:**
Users must be able to log that they've eaten a meal for the day. The system must track and display their daily logging streak.
- **F10. Theme Personalization:**
Users must be able to toggle between light and dark themes to enhance comfort and accessibility during app usage.

8.2 Non-Functional Requirements

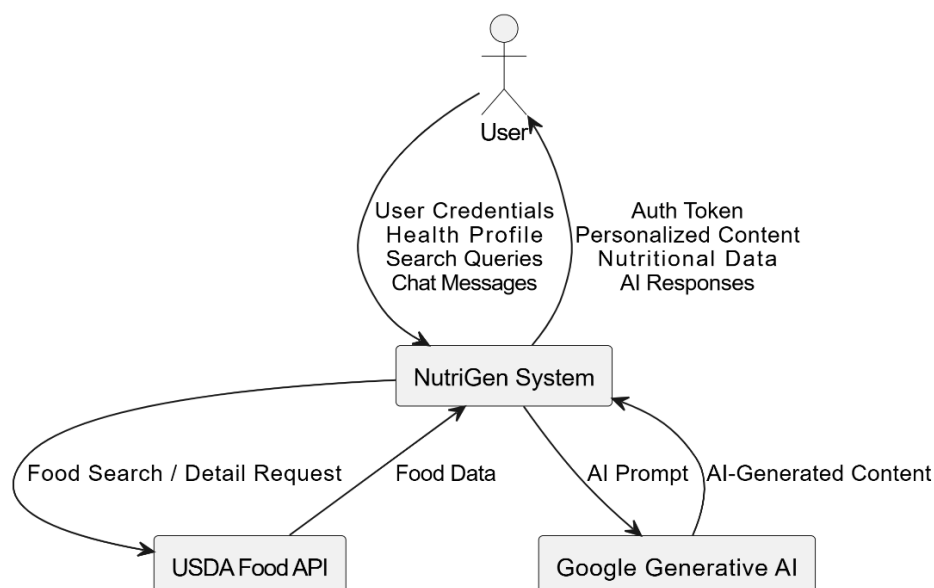
- **NF1. Security**
JWT-based authentication must secure APIs. Sensitive data (API keys, credentials) must be stored in environment variables and protected using best practices.
- **NF2. Performance**
The system must be responsive. API calls and AI processing should run asynchronously with loading indicators. Data must be cached when possible.
- **NF3. Usability**
The user interface must be clean, intuitive, responsive, and mobile-friendly. Accessibility standards (like color contrast and keyboard navigation) must be met.
- **NF4. Reliability**
The system must gracefully handle network failures, timeouts, and exceptions with clear, user-friendly error messages.

- **NF5. Scalability**
The app should handle increasing numbers of users without degradation. Firebase Firestore and Functions must be used efficiently to scale horizontally.
- **NF6. Maintainability**
The codebase should follow modular design, with clear separation of concerns and documentation for ease of future updates.
- **NF7. Compatibility**
The app should work across modern browsers (Chrome, Firefox, Safari, Edge) and be tested on different screen sizes.
- **NF8. Internationalization (i18n)**
The app should be prepared to support multiple languages and localization formats (dates, units, etc.).
- **NF9. Analytics Support**
The system should log user activity (meal generation, searches, streaks) for analytics, while maintaining data privacy.
- **NF10. Backup & Recovery**
Cloud Firestore's backup features should be used to ensure user data can be recovered in the event of failure or corruption.

9. Project Design - Data Flow Diagrams

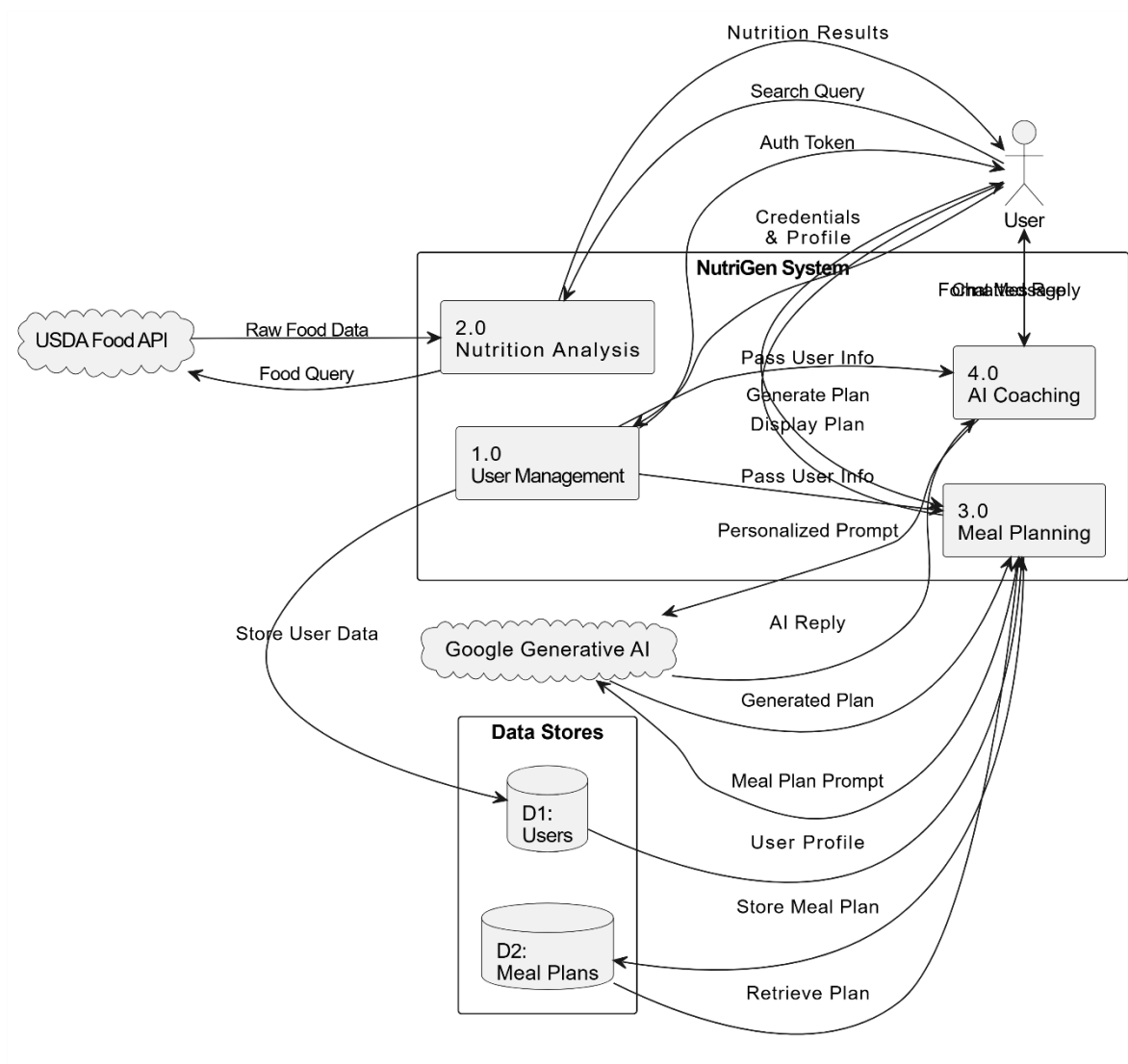
Level 0: Context Diagram

This diagram shows the overall context of the NutriGen system, with the user as the primary external entity.



Level 1: Data Flow Diagram

This diagram breaks down the NutriGen system into its major processes.



10. Technical Architecture

The NutriGen application is built on a modern, decoupled architecture with a React frontend and a Flask backend.

- **Frontend (Client):**
 - **Framework: React 18** with **Vite** as the build tool for a fast development experience.

- **State Management: Redux Toolkit** is used for centralized and predictable state management. It simplifies API calls and caching logic with `createAsyncThunk`.
 - **Routing: React Router v6** manages client-side navigation, including protected routes for authenticated users.
 - **Styling: TailwindCSS** provides a utility-first CSS framework for rapid UI development. A custom theme is defined in `tailwind.config.js` for brand consistency. Dark mode is supported via a `ThemeContext`.
 - **API Communication: Axios** is used for making HTTP requests to the backend, with an interceptor configured to automatically attach the JWT authorization header to protected requests.
 - **Animations: Framer Motion** is used to add fluid animations and transitions, enhancing the user experience.
 - **Backend (Server):**
 - **Framework: Flask**, a lightweight Python web framework, serves the REST API.
 - **Authentication:** A custom token-based authentication system is implemented. On login/register, a **JWT** is generated using the `PyJWT` library and a secret key. This token is then required for accessing protected API endpoints, where it is verified by a custom `@requires_auth` decorator.
 - **Database: Google Firestore** is used as the NoSQL database, accessed via the `firebase-admin` SDK. It stores user documents (containing profiles and health details) and meal plans.
 - **AI Integration: LangChain** and the `langchain-google-genai` library provide a high-level interface to communicate with Google's Gemini models. The backend constructs detailed prompts, sends them to the AI, and parses the JSON or text response.
 - **External APIs:** The `requests` library is used to interact with the **USDA FoodData Central API** for fetching nutritional information.
 - **Deployment:** The application is configured to run with `Gunicorn`, a production-ready WSGI server. CORS is handled by the `Flask-Cors` extension to allow requests from the frontend domain.
-

11. User Story

- **As a new user**, I want to register with my health details (age, weight, goal, allergies) so that the application can provide me with personalized nutritional advice and meal plans.

- **As a busy professional**, I want to generate a 7-day vegetarian meal plan for muscle gain so that I can shop and cook efficiently without having to plan every meal myself.
 - **As a health-conscious individual**, I want to quickly search for "1 cup of quinoa" to understand its nutritional content so I can make an informed decision about my lunch.
 - **As a user trying to lose weight**, I want to ask the virtual coach, "What are some healthy low-calorie snack options?" so I can get instant ideas without consulting a professional.
 - **As a user building a new habit**, I want to log my meals daily and see my progress streak so that I stay motivated on my health journey.
-

12. Project Planning & Task Scheduling

This table outlines a possible division of labor for the project's development.

Phase	Task	Team Member	Status
1. Setup & Design	Initialize project structure, set up Git, design database schema.	Team Lead	Completed
	Design UI/UX mockups and define component library.	Frontend Dev	Completed
2. Backend Dev	Implement Firebase integration and authentication (register, login, JWT).	Backend Dev	Completed
	Create API endpoints for meal plan generation and retrieval.	Backend Dev	Completed
	Implement AI prompt engineering for the meal planner and chat coach.	Backend Dev	Completed
	Develop USDA API integration for nutrition search.	Backend Dev	Completed
	Implement streak and meal logging logic and endpoints.	Backend Dev	Completed
3. Frontend Dev	Set up Redux store and slices (auth, mealPlan, coach).	Frontend Dev	Completed
	Build core UI components (Button, Card, Input) and layout (Navbar, Footer).	Frontend Dev	Completed
	Develop authentication pages (Login, Register) with form handling.	Frontend Dev	Completed

	Build Dashboard, Meal Planner, Insights, and Coach pages.	Frontend Dev	Completed
4. Integration & Test	Connect frontend components to backend APIs via Redux thunks.	Full Stack	Completed
	Write unit tests for backend logic and API endpoints.	Backend Dev	Completed
	Perform manual end-to-end testing of all user stories.	QA/Team	Completed
5. Deployment	Configure server for production (Gunicorn, environment variables).	Team Lead	Completed
	Deploy frontend and backend to a hosting service (e.g., Render, Vercel).	Team Lead	Completed

13. Reports - Feature Summaries

- **Landing Page:** Serves as the marketing front door, outlining the app's value proposition with clear calls-to-action for registration and login.
 - **Dashboard:** The user's home base after logging in. It displays a welcome message, a meal-logging streak tracker to encourage engagement, and quick-action cards to navigate to the main features.
 - **Meal Planner Page:** Allows users to generate a new 7-day meal plan with a single click. It presents the plan in a clean, day-by-day view with a summary of total daily nutritional values.
 - **Insights Page:** A powerful tool for on-demand nutritional information. Users can search for any food, select from a list of results, and view a detailed breakdown of its macro- and micronutrients, complete with a visual pie chart.
 - **Virtual Coach Page:** An interactive chat interface where users can ask nutrition-related questions. It maintains conversation history and provides helpful, context-aware answers based on the user's profile.
 - **Registration Page:** A comprehensive form that collects not only user credentials but also all the necessary health details (age, weight, goals, allergies) required for personalization.
-

14. Coding Explanation & Code Snippets

14.1 Backend

1. Secure API Endpoint with a Decorator (`app.py`)

This decorator ensures that only authenticated users with a valid JWT can access certain routes. It checks for the `Authorization` header, splits out the bearer token, and verifies it using the function from `firebase_config.py`.

Python:

```
# v2 finale/server/app.py
def requires_auth(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        auth_header = request.headers.get('Authorization')
        if not auth_header or not auth_header.startswith('Bearer '):
            raise AuthError({
                'code': 'authorization_header_missing',
                'description': 'Authorization header is expected.'
            }, 401)

        token = auth_header.split(' ')[1]
        try:
            payload = verify_custom_token(token)
            request.current_user = payload
        except Exception as e:
            raise AuthError({
                'code': 'invalid_token',
                'description': 'The token is invalid or expired.'
            }, 401)

        return f(*args, **kwargs)
    return decorated

@app.route('/api/me', methods=['GET'])
@requires_auth
def get_current_user():
    # This route is now protected
    user_id = request.current_user['uid']
    # ...
```

2. AI Prompt Engineering for Meal Plans (AI/mealPlanner.py)

This snippet shows the detailed prompt used to instruct the Gemini model. It includes placeholders for user data, a clear explanation of the required calculations (BMR), and a strict definition of the desired JSON output format. This ensures the AI's response is structured and predictable.

Python:

```

# v2 finale/server/AI/mealPlanner.py
def generate_meal_plan(name, age, gender, height, weight, diet_preference,
goal, activity_level, allergies):
    prompt = f"""
You are a certified AI nutritionist.

Your task is to create a personalized 7-day meal plan... based on the
following input:
- Name: {name}
- Age: {age}
- Gender: {gender}
# ... other user details ...
- Calories: Calculate the ideal daily calorie intake using this logic:
    - **Base Calories (BMR estimate)**:
        - Male: `10 × {weight} + 6.25 × {height} - 5 × {age} + 5`
        - Female: `10 × {weight} + 6.25 × {height} - 5 × {age} - 161`
# ... other instructions ...

STRICT JSON STRUCTURE (required):
{{
  "mealPlan": {{
    "Sunday": {{
      "Breakfast": {{
        "name": "", "ingredients": "", "portionSize": "",
        "calories": 0, "protein": 0, "carbs": 0, "fat": 0
      }},
      ...
    }}
  }}
}}

DO NOT:
- Include markdown
- Include explanations
"""

    # ... model invocation ...
    return response

```

14.2 Frontend

1. Asynchronous Authentication with Redux (app/features/authSlice.js)

This async thunk handles the user login process. It calls the `authService`, and upon a successful response, it updates the application state to mark the user as

authenticated and stores their data. Redux Toolkit automatically handles the pending, fulfilled, and rejected action types.

JavaScript:

```
// v2 finale/client/src/app/features/authSlice.js
import { createSlice, createAsyncThunk } from '@reduxjs/toolkit';
import authService from '../../api/authService';

export const login = createAsyncThunk(
  'auth/login',
  async ({ email, password }, { rejectWithValue }) => {
    try {
      const { user } = await authService.login(email, password);
      return { user };
    } catch (error) {
      return rejectWithValue(error.message || 'Login failed');
    }
  }
);

// ... in extraReducers
builder
  .addCase(login.pending, (state) => {
    state.isLoading = true;
    state.error = null;
  })
  .addCase(login.fulfilled, (state, action) => {
    state.isLoading = false;
    state.isAuthenticated = true;
    state.user = action.payload.user;
  })
  // ...
```

2. Handling User Input and API Calls (pages/InsightsPage.jsx)

This function from the Insights page demonstrates how the frontend captures user input, sets a loading state, calls the nutrition search API, and handles the response or error.

JavaScript:

```
// v2 finale/client/src/pages/InsightsPage.jsx
const handleSearchSubmit = async (e) => {
  e.preventDefault();
  if (!query.trim()) return;
```



```
setIsLoading(true);
setError(null);

try {
  const results = await searchFoodAPI(query.trim()); // API call
  setSearchResults(results);
} catch (err) {
  setError(err.message);
  setSearchResults([]);
} finally {
  setIsLoading(false);
}
};
```

15. Training - Prompt Engineering

In the context of NutriGen, AI "training" is not about fine-tuning a model but about **expert prompt engineering**. The application's intelligence is derived from crafting highly detailed and structured prompts that guide the pre-trained Google Gemini models to produce the desired output.

The most prominent example is in the `mealPlanner.py` module. The prompt is engineered with several key components:

1. **Role-Playing:** The prompt begins by assigning a role to the AI: "You are a certified AI nutritionist". This sets the context and tone for the response.
2. **Explicit Instructions:** The task is clearly defined: "create a personalized 7-day meal plan".
3. **Data Injection:** User-specific data (age, weight, goal, etc.) is dynamically inserted into the prompt using f-strings.
4. **In-Prompt Logic:** The AI is given the exact formulas to calculate the user's daily calorie needs based on their BMR, activity level, and goals. This offloads complex calculations to the LLM and ensures the plan is based on scientific principles.
5. **Strict Output Formatting:** The prompt demands the output be "pure JSON" and provides a precise, non-negotiable schema. This is critical for the backend to reliably parse the AI's response and save it to the database.
6. **Negative Constraints:** The prompt explicitly states what *not* to do (e.g., "Do not include markdown", "Do not include explanations"), which prevents the AI from adding conversational fluff that would break the JSON parsing.

A similar, though simpler, approach is used in `chat.py`, where the user's health profile is prepended to the chat history to give the AI context for personalized, conversational replies.

16. Testing

While automated test files were not included, a robust testing strategy for this project would involve the following layers:

- **Backend Testing:**
 - **Unit Tests:** Using a framework like `pytest` to test individual functions, such as the BMR calculation logic or the parsing functions in `NutriInsights.py`.
 - **Integration Tests:** Testing the interaction between different backend components, such as ensuring the `/generate-meal-plan` route correctly fetches user data from Firestore before calling the AI service.
 - **API Endpoint Tests:** Using a tool like Postman or writing test scripts to call each API endpoint (`/login`, `/register`, `/me`, etc.) with valid and invalid data/tokens to verify correct responses, status codes, and error handling.
- **Frontend Testing:**
 - **Component Tests:** Using a framework like React Testing Library to render individual components (e.g., `Button`, `Card`) and assert that they display correctly and respond to user interaction.
 - **Integration Tests:** Testing how multiple components work together. For example, testing the `LoginPage` to ensure that clicking the "Sign In" button dispatches the correct Redux action.
 - **End-to-End (E2E) Tests:** Using a tool like Cypress or Playwright to simulate a full user journey, such as registering, logging in, generating a meal plan, and logging out.
- **Manual Testing Use Cases:**
 - **Registration:** Attempt to create an account with a password that does not match the confirmation. Verify that an error is shown.
 - **Authentication:** Log in with correct credentials and confirm navigation to the dashboard. Log out and try to access `/dashboard` directly, verifying a redirect to `/login`.
 - **Meal Planner:** For a user with a "gluten" allergy, generate a meal plan and manually inspect the results to ensure no gluten-containing ingredients (like wheat) are present.

- **Insights:** Search for a common food like "banana" and verify results are displayed. Click a result and confirm the nutrition details and pie chart load correctly.
-

17. Advantages and Disadvantages

Advantages

- **Hyper-Personalization:** The use of Generative AI allows for a level of personalization in meal planning and coaching that is difficult to achieve with traditional software.
- **Cost-Effective:** NutriGen provides services that are analogous to those of a professional nutritionist but at a fraction of the cost, making it highly accessible.
- **24/7 Availability:** The AI coach and other tools are available anytime, providing instant support whenever the user needs it.
- **All-in-One Platform:** It consolidates multiple health tools (meal planner, nutrition database, coach) into a single, cohesive application.
- **Engaging User Experience:** Features like the meal-logging streak, interactive charts, and a clean UI are designed to keep users motivated and engaged.

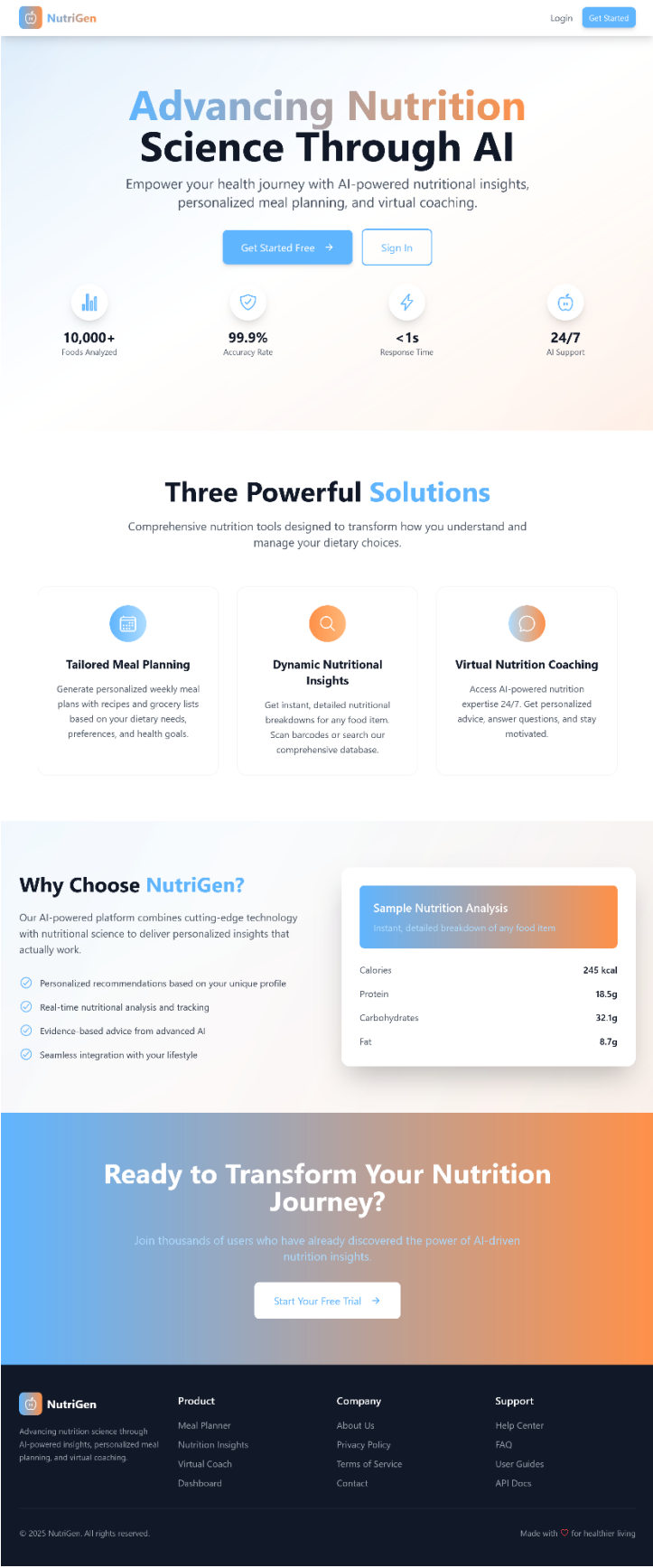
Disadvantages

- **AI Reliability:** As with any LLM, the AI's responses can occasionally be generic, factually incorrect, or may require regeneration to get the best result. It does not replace professional medical advice.
 - **API Dependency:** The application's core functionality is heavily dependent on external services (Google AI, USDA API). Any downtime, rate limiting, or cost changes from these services would directly impact NutriGen.
 - **Data Accuracy:** The accuracy of the nutritional data is entirely dependent on the quality of the data provided by the USDA API.
 - **Lack of Professional Oversight:** While the AI is powerful, it lacks the nuanced judgment and accountability of a certified human nutritionist, which is critical for users with serious medical conditions.
-

18. Output - User Interface

The user interface is designed to be clean, modern, and intuitive, with key information presented in visually appealing cards and charts.

- **Landing Page:** The first point of contact for users, it effectively communicates the app's value through bold typography, benefit lists, and feature cards.





Create Your Account

Join NutriGen and start your nutrition journey

Health Information

Full Name *

 Enter your full name

Age *

Enter your age


Height (cm) *

Enter your height


Weight (kg) *

Enter your weight


Gender

 Select gender


Diet Preference

 Select diet

Goal

 Select goal

Activity Level

 Select activity level

Allergies

☐ Gluten

☐ Dairy

☐ Nuts

☐ Soy

☐ Eggs

☐ None

Account Creation

Email Address *

 Enter your email

Password *

 Create a strong password



Confirm Password *

 Confirm your password



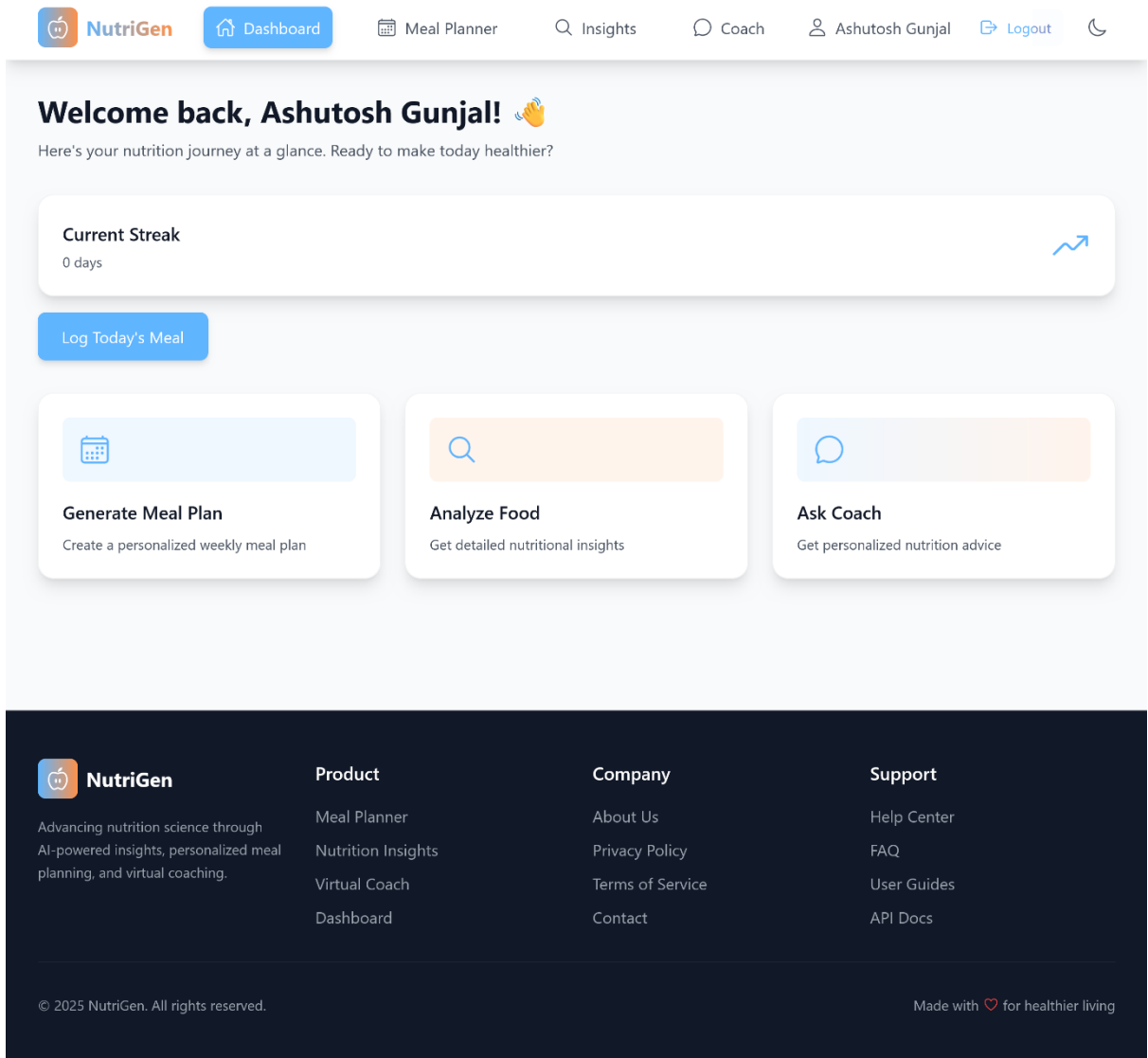
☐ I agree to the [Terms of Service](#) and [Privacy Policy](#)

Create Account

Already have an account? [Sign in here](#)

[← Back to Homepage](#)

- **Dashboard Page:** Upon logging in, the user is greeted with a personalized welcome message. The central features are a card displaying their current meal-logging **streak** and quick-action cards that provide easy navigation to the main sections of the app.



- **Insights Page:** This page features a prominent search bar. After a search is performed, results are displayed in a grid of cards. Clicking a result transitions the view to a detailed analysis, which includes a **nutrition pie chart** for macronutrients and a list of micronutrients, providing a comprehensive overview at a glance.

NutriGen

Dashboard

Meal Planner

Insights

Coach

Ashutosh Gunjal

Logout

Nutritional Insights

Search any food to get an instant, detailed nutritional breakdown. Make smarter choices, effortlessly.

dominos

Search

dominos

< New Search

DOMINO'S 14" Cheese Pizza,
Crunchy Thin Crust

Calories
1247 kcal

Protein
12.33 g

Carbs
28.18 g

Fat
15.1 g

Macronutrient Breakdown

Micronutrient Details

Alanine	0.395
Amino Acids	0
Arginine	0.505
Ash	2.8
Aspartic Acid	0.966
Calcium, Ca	292.0
Carbohydrates	0
Cholesterol	27.0
Copper, Cu	0.005

NutriGen

Advancing nutrition science through AI-powered insights, personalized meal planning, and virtual coaching.

Product

Meal Planner

Nutrition Insights

Virtual Coach

Dashboard

Company

About Us

Privacy Policy

Terms of Service

Contact

Support

Help Center

FAQ


User Guides

API Docs

© 2025 NutriGen. All rights reserved.

Made with for healthier living

- **Meal Planner Page:** This page features a horizontal day selector, allowing the user to switch between days of the week. For the selected day, it shows a summary of total calories and macros, followed by individual cards for Breakfast, Lunch, Dinner, and Snack, each detailing the meal's name and ingredients.

 **NutriGen**

[Dashboard](#)


[Meal Planner](#)


[Insights](#)

[Coach](#)

[Ashutosh Gunjal](#)

[Logout](#)





Meal Planner

Here is your personalized weekly meal plan. You can generate a new one at any time.

Generating...

Sun15

Mon16

Tue17

Wed18

Thu19

Fri20

Sat21

Wednesday


Calories0kcal

Protein0ggrams

Carbs0ggrams

Fat0ggrams

Loading your meal plan...

 **NutriGen**

Advancing nutrition science through AI-powered insights, personalized meal planning, and virtual coaching.

Product

[Meal Planner](#)[Nutrition Insights](#)[Virtual Coach](#)[Dashboard](#)


Company

[About Us](#)[Privacy Policy](#)[Terms of Service](#)[Contact](#)

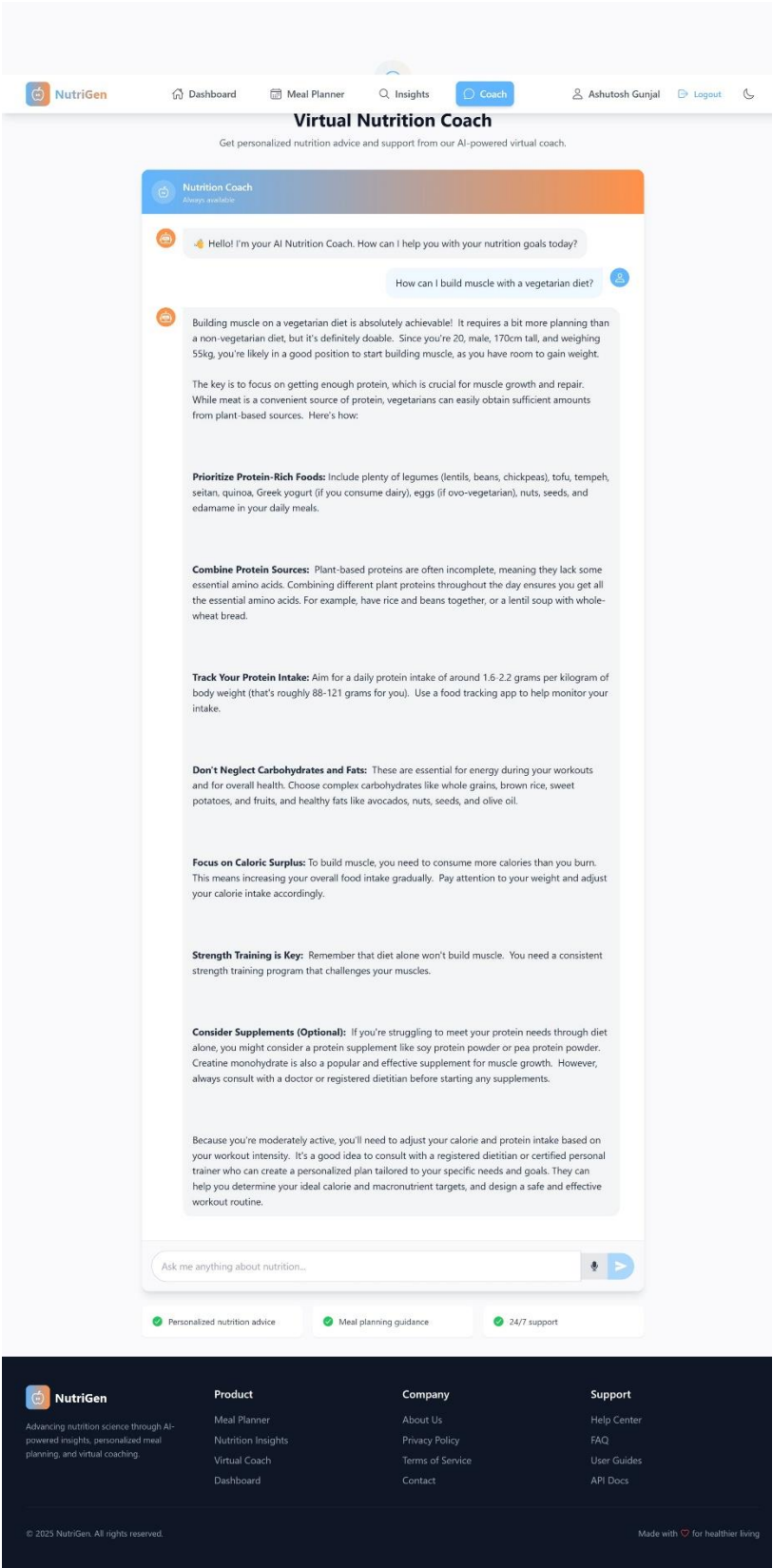
Support

[Help Center](#)[FAQ](#)[User Guides](#)[API Docs](#)

© 2025 NutriGen. All rights reserved.

Made with  for healthier living

- Virtual Coach Page:** This page presents a familiar chat interface with message bubbles for the user and the AI coach. It includes suggested prompts to help users start the conversation and supports voice input for hands-free interaction.



19. Future Scope

The current platform provides a strong foundation that can be extended with several high-impact features:

- **Mobile Application:** Develop a native or cross-platform (e.g., React Native) mobile app for on-the-go access to coaching and food logging.
- **Fitness Tracker Integration:** Integrate with APIs from Apple Health, Google Fit, and other fitness trackers to automatically sync activity levels, weight, and other health metrics, further enhancing personalization.
- **Image-to-Nutrition Analysis:** Allow users to take a photo of their meal, and use a computer vision model to identify the food items and estimate their nutritional content automatically.
- **Advanced Grocery List Management:** Enhance the grocery list feature with the ability to check off items, categorize them by store aisle, and integrate with online grocery delivery services.
- **Community and Social Features:** Introduce a community section where users can share their own healthy recipes, progress, and support one another, fostering a more engaging and supportive ecosystem.
- **Barcode Scanning:** Implement barcode scanning functionality in the mobile app to allow users to instantly retrieve nutritional information for packaged food products.

20. Conclusion

The NutriGen project successfully demonstrates the integration of a modern frontend stack (React, Redux) with a robust Python backend (Flask) to deliver a powerful, AI-driven health and nutrition platform. By leveraging Google's Generative AI, the application effectively addresses common challenges in personal nutrition management, offering highly personalized meal plans, on-demand nutritional insights, and accessible virtual coaching. The well-structured code, clear separation of concerns, and secure authentication mechanism provide a solid foundation for a scalable and user-centric product. NutriGen stands as a strong proof-of-concept for how AI can be practically applied to empower individuals in their journey toward better health.

21. References

1. Google Research. (2024, June 11). *Advancing personal health and wellness insights with AI*. Retrieved from <https://research.google/blog/advancing-personal-health-and-wellness-insights-with-ai/>
2. U.S. Department of Agriculture. (2024). *FoodData Central*. Retrieved from <https://fdc.nal.usda.gov/>
3. Firebase. (2024). *Firebase Authentication & Firestore documentation*. Retrieved from <https://firebase.google.com/docs>
4. Python Software Foundation. (2024). *Flask web development documentation*. Retrieved from <https://flask.palletsprojects.com/>
5. OpenAI. (2023). *Prompt engineering guide for large language models*. Retrieved from <https://platform.openai.com/docs/guides/prompt-engineering>
6. Google AI. (2024). *Gemini API overview and SDK*. Retrieved from <https://ai.google.dev/>
7. Jones, M., Bradley, J., & Sakimura, N. (2015). *JSON Web Token (JWT)*. Internet Engineering Task Force (IETF). Retrieved from <https://jwt.io/introduction>