Healthcare Planning Assistant Agent

Course Name: Agentic AI

Institution Name: Medicaps University – Datagami Skill Based Course

Student Name(s) & Enrolment Number(s):

Sr no	Student Name	Enrolment Number
1.	Meet Solanki	EN22CS301595
2.	Mayuri Patidar	EN22CS301593
3.	Mokshika Yadav	EN22CS301614
4.	Nirdesh Sharma	EN22CS301655
5.	Pankaj	EN22CS301679
6.	Palak Shahdadpuri	EN22CS301677

Group Name: 11D

Project Number: AAI-22

Industry Mentor Name:

University Mentor Name: Nishant Shrivastav

Academic Year:2026

1. PROBLEM STATEMENT & OBJECTIVES:-

1.1 PROJECT STATEMENT

The healthcare ecosystem today faces a growing gap between the demand for immediate health information and the availability of professional medical consultation. With the rapid expansion of digital platforms, individuals increasingly rely on search engines and social media to interpret their symptoms. However, the information available online is often fragmented, non-contextual, and sometimes misleading. This lack of structured and personalized guidance can result in unnecessary panic, incorrect assumptions about illnesses, delayed treatment, or inappropriate self-medication. In many cases, individuals struggle to differentiate between minor health concerns and serious medical conditions, which further complicates decision-making and increases anxiety.

Access to healthcare services is also unevenly distributed. In rural or underdeveloped areas, medical facilities and specialists may not be readily available. Even in urban regions, long waiting times, high consultation costs, and limited appointment availability can discourage people from seeking timely advice. As a result, individuals often delay consulting healthcare professionals, relying instead on incomplete or unverified information sources. This situation highlights the need for a reliable, easily accessible preliminary advisory system that can provide structured health-related insights based on user-reported symptoms.

Another significant challenge lies in the lack of organized symptom interpretation tools that present information in a clear and user-friendly format. Many existing digital tools provide generic responses without contextual reasoning or structured categorization. Users require a system that not only suggests possible health conditions but also offers precautionary measures, general medication awareness, and nutritional recommendations in a logically organized manner. The absence of such an integrated solution limits the effectiveness of current digital healthcare assistance platforms.

Furthermore, with the advancement of Artificial Intelligence and Large Language Models, there exists an opportunity to design intelligent systems capable of reasoning over symptom inputs and generating coherent, structured, and context-aware health guidance. However, most available AI tools are not specifically structured for healthcare advisory workflows and may lack modularity, transparency, and maintainability in their design. There is a need for a layered, modular AI-driven system that ensures secure API communication, flexible model integration, and privacy-focused architecture without storing sensitive user data.

1. AI-based healthcare advisory prototype
2. Symptom-driven health condition analysis

3. Structured output generation (conditions, precautions, medication, nutrition)
4. Layered modular architecture design
5. LLM orchestration using LangChain
6. Integration with Google Gemini and Groq APIs

1.2 PROJECT OBJECTIVES

The objective of the Healthcare Planning Assistant is to design and develop an intelligent, AI-driven healthcare advisory system capable of analyzing user-provided symptoms and generating structured, easy-to-understand health insights. The system is intended to function as a preliminary guidance tool that helps users gain better awareness of their health concerns before seeking professional medical consultation. By leveraging advanced Large Language Models, the assistant interprets symptom descriptions and produces organized outputs that include possible health conditions, precautionary measures, general medication awareness, and personalized nutritional suggestions.

A key objective of the project is to ensure that the generated information is presented in a clear, structured, and logically separated format so that users can easily understand and interpret the recommendations. Rather than providing scattered or generic responses, the system is designed to follow a sequential reasoning approach where each stage of output builds upon the previous one. This enhances contextual understanding and improves the reliability of advisory responses.

Key Objectives

1. **Symptom-Based Analysis**
   To design a system that accepts user symptoms and analyzes them using AI models to identify possible health conditions.

2. **Structured Health Guidance**
   To generate organized outputs divided into sections such as conditions, precautions, medication suggestions, and nutrition plans for better clarity.

3. **LLM Integration**
   To integrate advanced Large Language Models like Google Gemini and Groq using LangChain for intelligent reasoning.

4. **Modular Architecture Design**
   To implement a layered system architecture that ensures maintainability, flexibility, and future scalability.

5. **Sequential Prompt Chaining**
   To enable step-by-step reasoning where each response builds on the previous output or contextual accuracy.
6. **Security and Privacy Focus**
   To ensure secure API handling and avoid storing any sensitive user health data.

7. **Vendor Flexibility**
   To create an abstraction layer that allows switching between different LLM providers without major code changes.

8. **Practical AI Implementation**
   To demonstrate real-world application of AI in healthcare advisory systems while maintaining ethical and advisory boundaries.

1.3 PROJECT SCOPE

In Scope

The Healthcare Planning Assistant project includes the design and development of a web-based AI-driven healthcare advisory prototype. The system accepts user-provided symptoms and processes them using Large Language Models to generate structured health insights. It provides preliminary guidance in the form of possible health conditions, precautionary measures, general medication awareness, and nutritional recommendations. The application is built using Python and Streamlit for the user interface, with LangChain handling prompt orchestration and integration with Google Gemini and Groq APIs. The architecture follows a layered modular design to ensure maintainability and scalability. The system also includes secure API key handling and does not store any user health data, maintaining a privacy-focused approach. The project demonstrates sequential prompt chaining and structured output formatting within a local prototype environment.

Out of Scope

The project does not provide clinical diagnosis, medical prescriptions, emergency healthcare services, or real-time health monitoring. It does not integrate with hospital management systems, Electronic Health Records (EHR), insurance platforms, wearable health devices, or government healthcare databases. The system does not store patient history or maintain long-term user data. Regulatory compliance certifications, production-grade cloud deployment, and large-scale multi-user infrastructure are not included in the current prototype. The application is strictly advisory in nature and is not intended to replace professional medical consultation or certified healthcare services.

2. PROPOSED SOLUTION:-

2.1 KEY FEATURES

1. Interactive Symptom Input Interface

The system has a simple and user-friendly web page where users can type their symptoms easily. It is designed so that anyone can use it without technical knowledge.

2. Dynamic Model Selection

Users can choose which AI model they want to use, like Google Gemini or Groq. This helps in comparing performance and gives flexibility to use different AI services.

3. Context-Aware AI Reasoning

The system understands the connection between different steps. It does not give random answers. Instead, it remembers previous responses and gives logical and connected health suggestions.

4. Real-Time Response Generation

Once the user enters symptoms, the system quickly processes the information and shows results instantly. There is no delay or manual work needed.

5. Clear Section-Based Output Display

The results are shown in different sections like possible diseases, precautions, medicines, and diet plan. This makes the output easy to read and understand.

6. Error Handling Mechanism

If the user enters wrong information or if there is an API problem, the system shows proper error messages instead of stopping suddenly. This makes the system reliable.

7. Session State Management

The system remembers user inputs during the current session, but it does not store the data permanently. This protects user privacy.

8. Vendor-Independent Architecture

The system is designed in such a way that we can change the AI model easily without changing the main code. This makes the system flexible for the future.

9. Lightweight Prototype Design

The project runs on a local system and does not need heavy servers or databases. It is simple and easy to deploy.

10. Expandable Framework

The system can be improved in the future by adding features like login system, database, cloud hosting, analytics dashboard, and more advanced healthcare tools.

2.2 OVERALL ARCHITECTURE / WORKFLOW

2.2.1 SYSTEM ARCHITECTURE

The Healthcare Planning Assistant follows a Layered Architecture Model.
This architecture divides the system into different layers, where each layer performs a specific function.
This design improves maintainability, scalability, and flexibility.

The system mainly consists of four layers:

1. Presentation Layer
2. Application Logic Layer
3. LLM Orchestration Layer
4. External AI Service Layer

1. Presentation Layer (User Interface Layer) - The Presentation Layer acts as the user interface of the system and is developed using Streamlit. It is responsible for collecting user input such as symptoms, API key, and model selection (Gemini or Groq). It also displays structured health guidance results and error messages. This layer does not perform any heavy processing. Its main purpose is to provide a simple, interactive, and user-friendly interface that communicates with the backend system.

2. Application Logic Layer (Backend Processing Layer) -The Application Logic Layer is implemented using Python and acts as the core controller of the system. It validates the user input, manages session state, controls the workflow, formats prompts, handles responses, and manages errors. After receiving symptoms from the user interface, this layer checks whether the input is valid, prepares structured prompts, and sends them to the AI orchestration layer. Once the AI response is received, it organizes the output into clear sections such as possible conditions, precautions, medication awareness, and nutrition advice before sending the final structured result back to the user interface. This layer is considered the main processing unit of the system.

3. LLM Orchestration Layer (AI Management Layer) - The LLM Orchestration Layer manages communication between the backend and external AI models. It uses tools like LangChain to structure prompts, manage context, and handle request-response cycles efficiently. This layer ensures that prompts are properly formatted and routed to the selected AI model. It also allows dynamic switching between different AI providers without modifying the core application logic, ensuring vendor independence and system flexibility.

4. External AI Service Layer (LLM APIs) - The External AI Service Layer (LLM APIs) includes external AI models such as Google Gemini API and Groq LLaMA models. This layer performs intelligent reasoning and natural language processing. It analyzes user symptoms, generates possible medical conditions, suggests precautions, provides medication awareness, and offers nutrition advice. The system sends formatted prompts through secure API calls to these AI services and receives AI-generated responses, which are then processed by the backend.

System Architecture Diagram (Text Representation)-

User

↓

Streamlit UI (Presentation Layer)

↓

Application Logic (Python Backend)

↓

LangChain (LLM Orchestration)

↓

Gemini / Groq API (External AI Service)

↓

Response back to UI

Working Flow of Architecture

In the overall working flow of the architecture, the user first enters symptoms through the interface. The backend validates the input and generates structured prompts using LangChain. The request is then sent to the selected AI model through secure API communication. The AI model processes the input and generates a response. Finally, the backend organizes the response into structured sections and displays the output clearly to the user.

This layered architecture ensures clean separation of responsibilities, efficient AI integration, and easy future enhancement of the Healthcare Planning Assistant system.

2.2.2 DATA FLOW DIAGRAM(DFD)

A Data Flow Diagram (DFD) is a graphical representation of how data moves within a system. It shows the flow of information between users, processes, and external systems. DFD focuses on data movement rather than system structure.

In the Healthcare Planning Assistant, the DFD explains how user symptoms are processed and how AI-generated responses are returned to the user.

DFD Components Used

The following standard DFD components are used:

1. External Entity → Represents users or external systems
2. Process → Represents operations performed on data
3. Data Flow → Shows movement of data
4. Data Store → Represents stored data (if any)

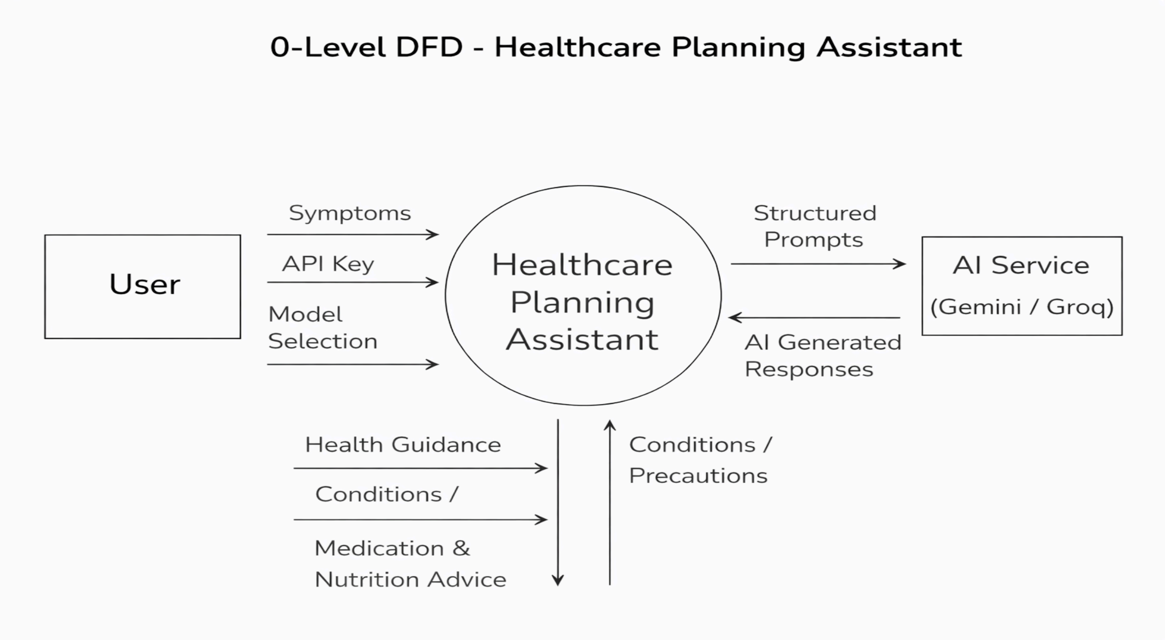
In this system, permanent data storage is not used, so the Data Store component is minimal.

Level 0 DFD - The Level 0 DFD represents the entire system as a single process.

External Entity: User

Main Process: Healthcare Planning Assistant System

Data Flow: 1) User provides symptoms as input
2) System processes input using AI
3) System returns health guidance output


0-Level DFD - Healthcare Planning Assistant
User
Symptoms
API Key
Model Selection
Healthcare Planning Assistant
Structured Prompts
AI Service (Gemini / Groq)
AI Generated Responses
Health Guidance
Conditions /
Medication & Nutrition Advice
Conditions / Precautions

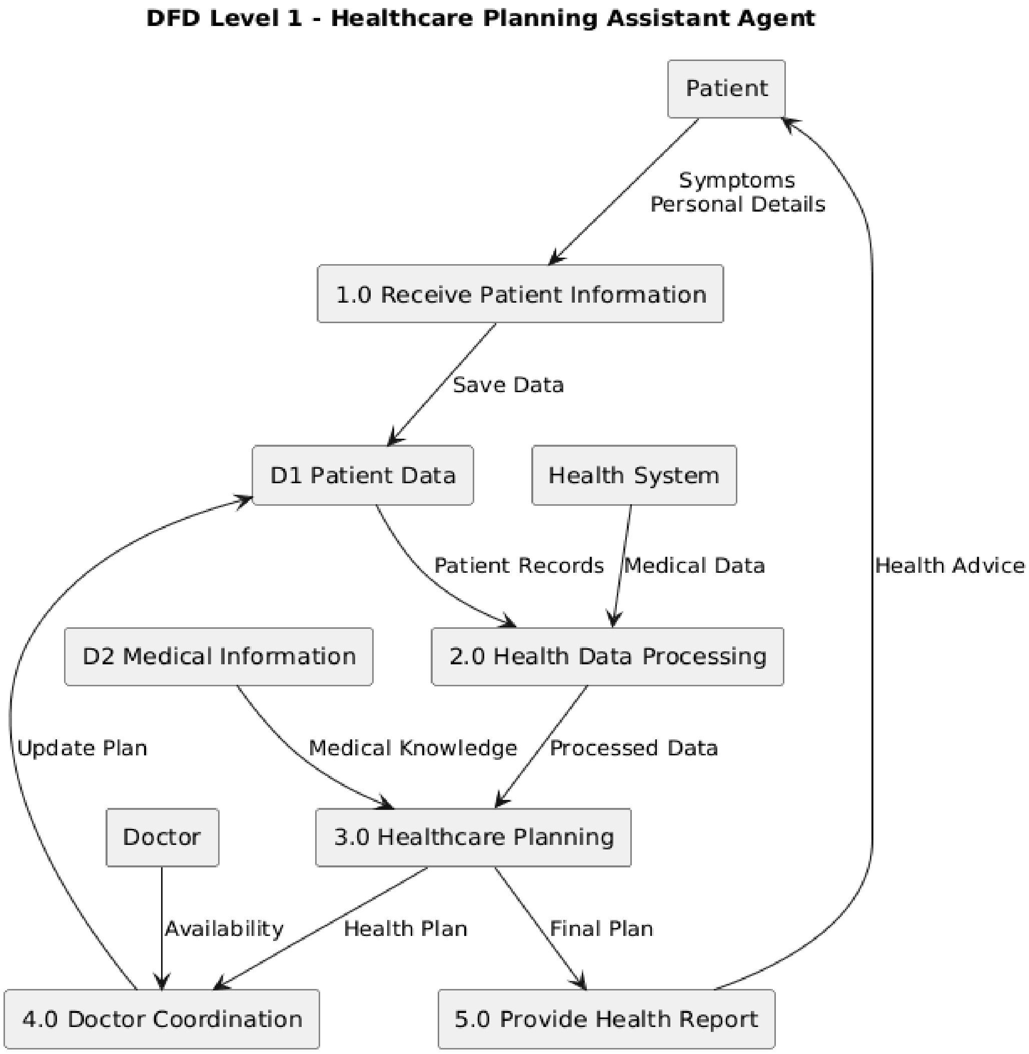
Working: - The Level 0 Data Flow Diagram, also known as the Context Diagram, represents the entire Healthcare Planning Assistant system as a single process. At this level, the internal working of the system is not shown. Instead, it focuses only on the interaction between the external entity and the system.

In this system, the external entity is the User. The user provides symptoms as input to the Healthcare Planning Assistant System. The system then processes the input using artificial intelligence models and generates structured health guidance as output. This output includes possible conditions, precautions, medication awareness, and nutrition suggestions. The generated response is then sent back to the user.

Thus, at Level 0, the system is treated as one single processing unit that receives symptoms and returns health recommendations. This level provides a high-level overview of the system's interaction with the user.

Level 1 DFD - The Level 1 DFD breaks the system into major internal processes.

External Entity: User



Working: - The Level 1 Data Flow Diagram expands the single process shown in Level 0 into multiple internal processes to explain how the system works in detail.

When the user enters symptoms into the system, the first step is the Input Collection Process, which captures the symptoms and other required information such as the API key and selected AI model. The collected data is then passed to the Input Validation Process, where the system checks whether the input is empty or invalid. If the input is invalid, an error message is displayed to the user.

If the input is valid, the data moves to the Prompt Generation Process. In this stage, the system formats the user's symptoms into structured prompts suitable for AI processing. These prompts are then sent to the AI Processing stage, where the selected AI model (Gemini or Groq) analyzes the symptoms and generates health-related responses.

Once the AI response is received, it is sent to the Response Formatting Process. Here, the system organizes the output into clear sections such as possible conditions, precautions, medication advice, and nutrition recommendations. Finally, the structured result is displayed to the user through the Output Display Process.

2.2.3 USE CASE DIAGRAM

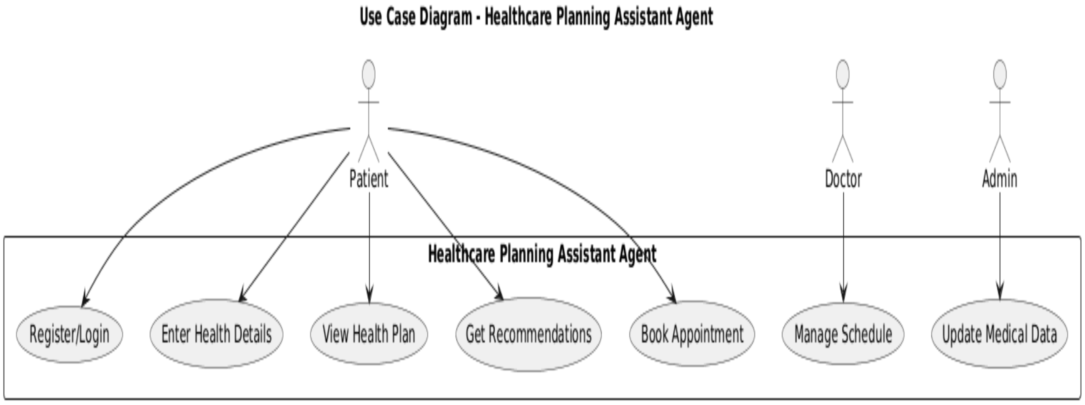
A Use Case Diagram represents the interaction between users (actors) and the system. It shows what functionalities the system provides and how the user interacts with them.

In the Healthcare Planning Assistant system, the Use Case Diagram explains how the user interacts with the system to receive healthcare guidance. It provides a high-level functional view of the system from the user's perspective. The diagram helps in identifying system requirements and understanding the main services offered by the application. It also simplifies communication between developers, stakeholders, and users by clearly showing system behavior and user involvement.

Primary Actor - User
The User interacts with the system to enter symptoms and receive health recommendations.

Main Use Cases

1. Enter Symptoms
2. Provide API Key
3. Select AI Model
4. Validate Input
5. Generate Health Analysis
6. View Health Recommendations



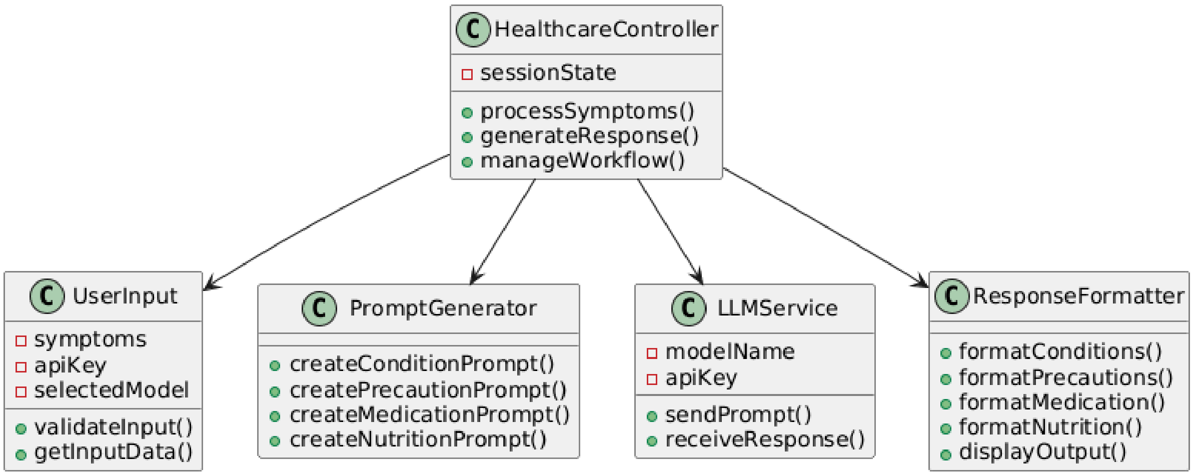
Working: - The user first enters symptoms into the system. The user also provides the API key and selects the AI model (Gemini or Groq). After submission, the system validates the input. If the input is valid, the system processes the symptoms using AI and generates health analysis results. Finally, the system displays structured health recommendations including possible conditions, precautions, medication awareness, and nutrition advice.

2.2.4 CLASS DIAGRAM

A Class Diagram represents the static structure of a system. It shows the classes, their attributes, methods, and relationships between them. It is mainly used to describe the object-oriented design of the system.

In the Healthcare Planning Assistant system, the Class Diagram explains how different components such as user input handling, AI processing, and response formatting are structured in the backend.

Class Diagram - Healthcare Planning Assistant



Working:

The Class Diagram shows how different classes work together in the Healthcare Planning Assistant system. The UserInput class collects and validates the symptoms entered by the user. The HealthcareController class manages the overall workflow of the system.

The PromptGenerator class creates structured prompts based on user symptoms. These prompts are sent to the AI model through the LLMService class, which handles communication with external AI APIs. Finally, the ResponseFormatter class organizes the AI response into structured sections and displays the output to the user.

Thus, the Class Diagram explains the internal structure and interaction between system components.

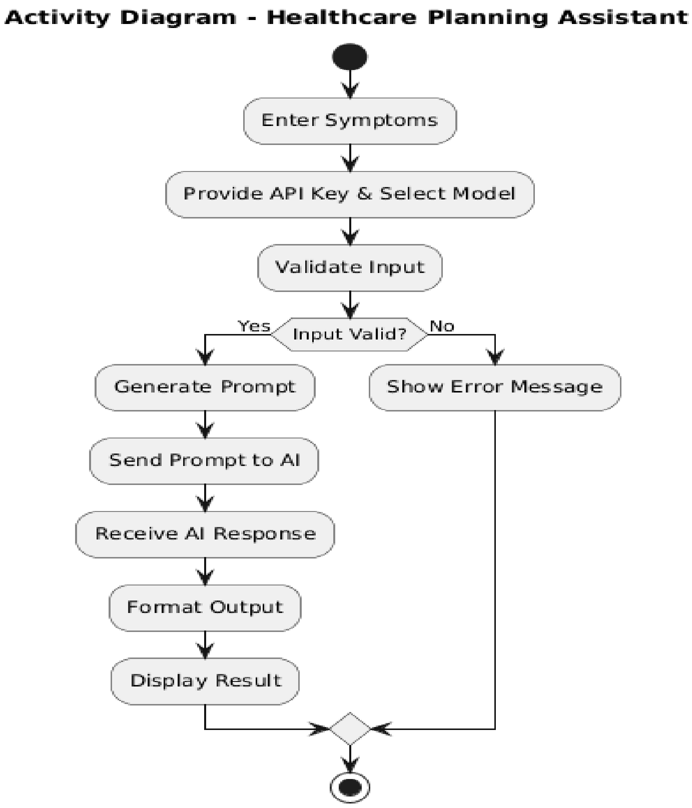
2.2.5 ACTIVITY DIAGRAM

An Activity Diagram represents the workflow of a system. It shows the sequence of activities, decision points, and the flow of control from start to end. It helps in understanding how the system performs tasks step-by-step.

In the Healthcare Planning Assistant system, the Activity Diagram explains how user input is processed and how healthcare guidance is generated.

Main Activities in the System:

1. Start
2. Enter Symptoms
3. Provide API Key & Select Model
4. Validate Input
5. Decision (Input Valid?)
6. Generate Condition Prompt
7. Send Prompt to AI
8. Receive AI Response
9. Generate Precaution, Medication & Nutrition Prompts
10. Format Response
11. Display Output
12. End



Working:- The activity begins when the user opens the system and enters symptoms. The user also provides the API key and selects the AI model. The system then validates the input. If the input is invalid, an error message is displayed, and the process stops.

If the input is valid, the system generates structured prompts and sends them to the AI model. The AI processes the request and returns responses. The system then formats the response into structured sections such as possible conditions, precautions, medication advice, and nutrition recommendations. Finally, the formatted result is displayed to the user, and the activity ends.

2.3 TOOLS AND TECHNOLOGIES USED

The Healthcare Planning Assistant system is developed using modern AI technologies and web-based tools. The following tools and technologies were used during the development of the project:

1. Python - Python is the primary programming language used for backend development. It is simple, powerful, and widely used in AI and machine learning applications. Python is used to handle input processing, workflow management, API integration, and response formatting in the system.

2.Streamlit - Streamlit is used to build the web-based user interface of the application. It allows quick development of interactive web applications using Python. Streamlit is responsible for:

1. Taking user input (symptoms, API key, model selection)
2. Displaying structured healthcare guidance
3. Managing session state

3.LangChain - LangChain is used as the LLM orchestration framework. It helps in managing prompts and maintaining context between multiple AI calls. LangChain is responsible for:

1. Creating structured prompt templates
2. Managing sequential prompt chaining
3. Handling AI model interaction

4. **Google Gemini API -** The Google Gemini API is used as one of the AI models in the system. It processes user symptoms and generates health-related guidance. It performs natural language understanding and intelligent response generation.

5.**Groq API (LLaMA Models) -** Groq API is used as an alternative AI model provider. It enables fast and efficient AI inference using LLaMA-based models. This provides flexibility in model selection and improves system scalability.

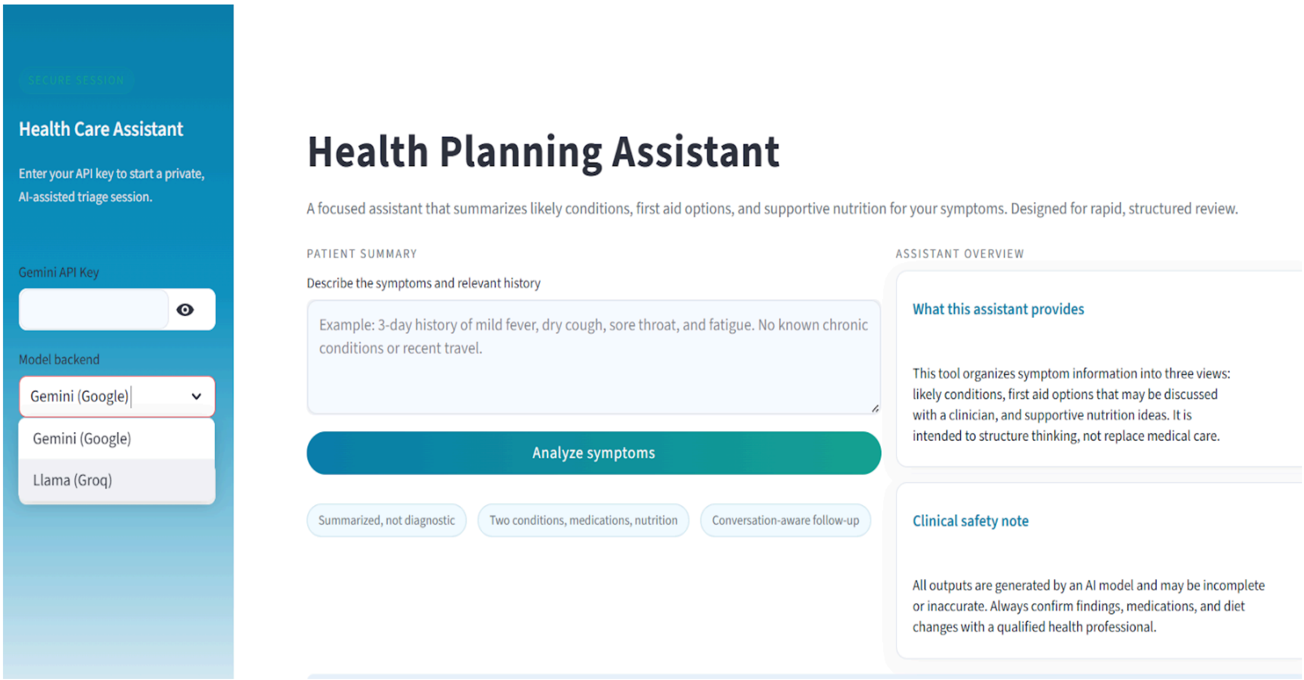
6. REST API Integration - The system communicates with external AI services using secure REST API calls. These APIs allow sending prompts and receiving responses over the internet securely.

7. VS Code - Visual Studio Code is used as the development environment. It provides coding support, debugging tools, and extension support for Python and AI libraries.

8. Git and GitHub - Git is used for version control, and GitHub is used for code repository management. These tools help in tracking changes, collaboration, and project management.

3. RESULTS AND OUTPUT

3.1 SCREENSHOTS / OUTPUT


SECURE SESSION
👁
Gemini (Google)
Gemini (Google)
Llama (Groq)
Analyze symptoms
Summarized, not diagnostic
Two conditions, medications, nutrition
Conversation-aware follow-up

Health Care Assistant

Enter your API key to start a private, AI-assisted triage session.

Gemini API Key

Model backend: Gemini (Google)

Health Planning Assistant

A focused assistant that summarizes likely conditions, first aid options, and supportive nutrition for your symptoms. Designed for rapid, structured review.

PATIENT SUMMARY

Describe the symptoms and relevant history

Example: 3-day history of mild fever, dry cough, sore throat, and fatigue. No known chronic conditions or recent travel.

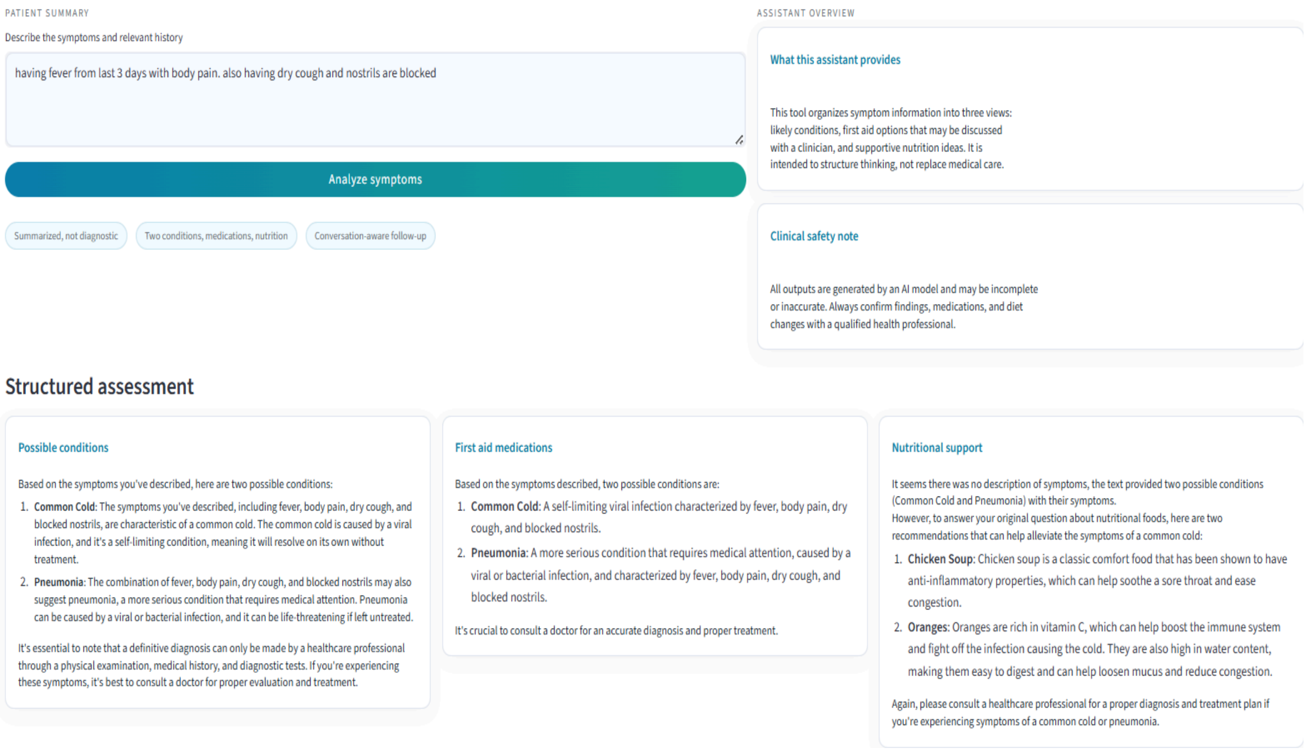
ASSISTANT OVERVIEW

What this assistant provides

This tool organizes symptom information into three views: likely conditions, first aid options that may be discussed with a clinician, and supportive nutrition ideas. It is intended to structure thinking, not replace medical care.

Clinical safety note

All outputs are generated by an AI model and may be incomplete or inaccurate. Always confirm findings, medications, and diet changes with a qualified health professional.


Analyze symptoms
Summarized, not diagnostic
Two conditions, medications, nutrition
Conversation-aware follow-up

PATIENT SUMMARY

Describe the symptoms and relevant history

having fever from last 3 days with body pain. also having dry cough and nostrils are blocked

ASSISTANT OVERVIEW

What this assistant provides

This tool organizes symptom information into three views: likely conditions, first aid options that may be discussed with a clinician, and supportive nutrition ideas. It is intended to structure thinking, not replace medical care.

Clinical safety note

All outputs are generated by an AI model and may be incomplete or inaccurate. Always confirm findings, medications, and diet changes with a qualified health professional.

Structured assessment

Possible conditions

Based on the symptoms you've described, here are two possible conditions:

1. **Common Cold**: The symptoms you've described, including fever, body pain, dry cough, and blocked nostrils, are characteristic of a common cold. The common cold is caused by a viral infection, and it's a self-limiting condition, meaning it will resolve on its own without treatment.
2. **Pneumonia**: The combination of fever, body pain, dry cough, and blocked nostrils may also suggest pneumonia, a more serious condition that requires medical attention. Pneumonia can be caused by a viral or bacterial infection, and it can be life-threatening if left untreated.

It's essential to note that a definitive diagnosis can only be made by a healthcare professional through a physical examination, medical history, and diagnostic tests. If you're experiencing these symptoms, it's best to consult a doctor for proper evaluation and treatment.

First aid medications

Based on the symptoms described, two possible conditions are:

1. **Common Cold**: A self-limiting viral infection characterized by fever, body pain, dry cough, and blocked nostrils.
2. **Pneumonia**: A more serious condition that requires medical attention, caused by a viral or bacterial infection, and characterized by fever, body pain, dry cough, and blocked nostrils.

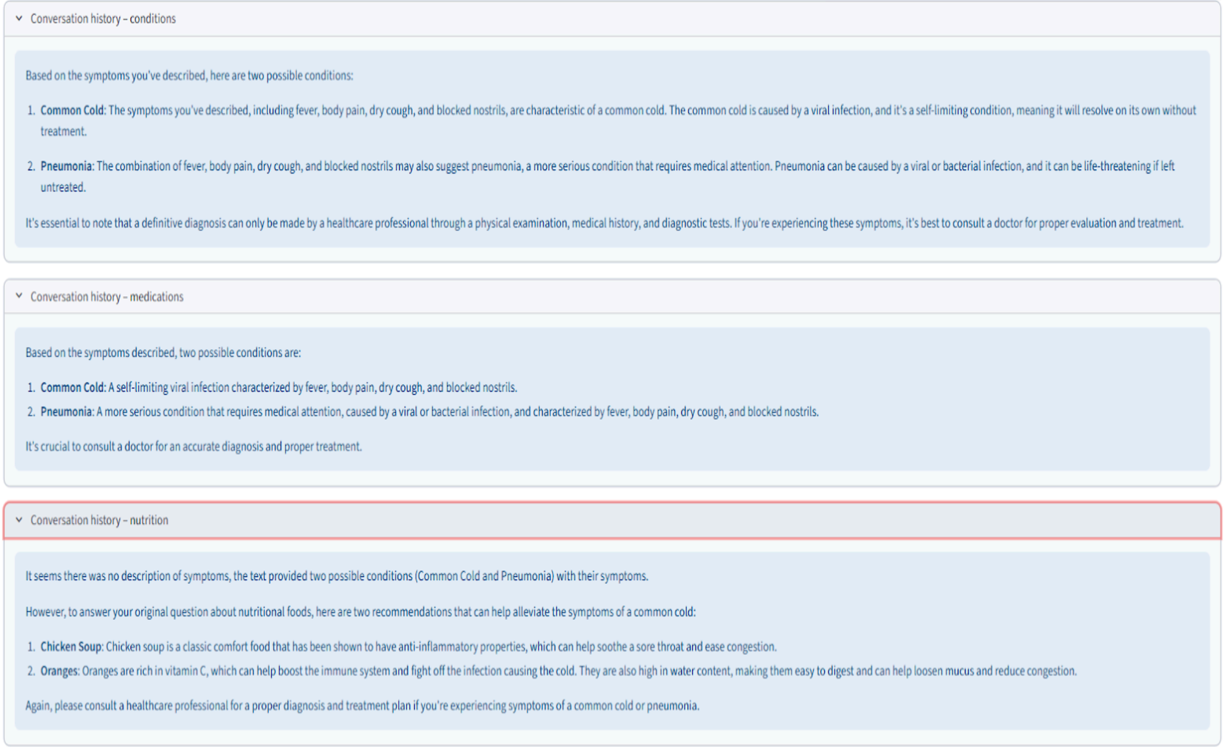
It's crucial to consult a doctor for an accurate diagnosis and proper treatment.

Nutritional support

It seems there was no description of symptoms, the text provided two possible conditions (Common Cold and Pneumonia) with their symptoms.
However, to answer your original question about nutritional foods, here are two recommendations that can help alleviate the symptoms of a common cold:

1. **Chicken Soup**: Chicken soup is a classic comfort food that has been shown to have anti-inflammatory properties, which can help soothe a sore throat and ease congestion.
2. **Oranges:** Oranges are rich in vitamin C, which can help boost the immune system and fight off the infection causing the cold. They are also high in water content, making them easy to digest and can help loosen mucus and reduce congestion.

Again, please consult a healthcare professional for a proper diagnosis and treatment plan if you're experiencing symptoms of a common cold or pneumonia.



Conversation history – conditions

Based on the symptoms you've described, here are two possible conditions:

1. **Common Cold**: The symptoms you've described, including fever, body pain, dry cough, and blocked nostrils, are characteristic of a common cold. The common cold is caused by a viral infection, and it's a self-limiting condition, meaning it will resolve on its own without treatment.
2. **Pneumonia**: The combination of fever, body pain, dry cough, and blocked nostrils may also suggest pneumonia, a more serious condition that requires medical attention. Pneumonia can be caused by a viral or bacterial infection, and it can be life-threatening if left untreated.

It's essential to note that a definitive diagnosis can only be made by a healthcare professional through a physical examination, medical history, and diagnostic tests. If you're experiencing these symptoms, it's best to consult a doctor for proper evaluation and treatment.

Conversation history – medications

Based on the symptoms described, two possible conditions are:

1. **Common Cold**: A self-limiting viral infection characterized by fever, body pain, dry cough, and blocked nostrils.
2. **Pneumonia**: A more serious condition that requires medical attention, caused by a viral or bacterial infection, and characterized by fever, body pain, dry cough, and blocked nostrils.

It's crucial to consult a doctor for an accurate diagnosis and proper treatment.

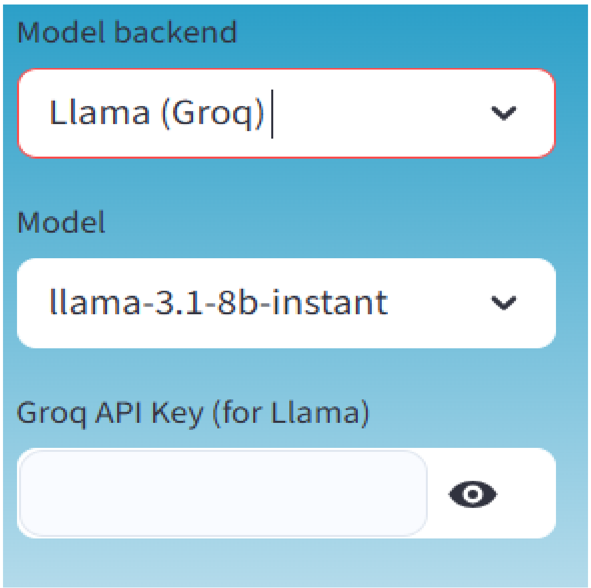
Conversation history – nutrition

It seems there was no description of symptoms, the text provided two possible conditions (Common Cold and Pneumonia) with their symptoms.

However, to answer your original question about nutritional foods, here are two recommendations that can help alleviate the symptoms of a common cold:

1. **Chicken Soup**: Chicken soup is a classic comfort food that has been shown to have anti-inflammatory properties, which can help soothe a sore throat and ease congestion.
2. **Oranges**: Oranges are rich in vitamin C, which can help boost the immune system and fight off the infection causing the cold. They are also high in water content, making them easy to digest and can help loosen mucus and reduce congestion.

Again, please consult a healthcare professional for a proper diagnosis and treatment plan if you're experiencing symptoms of a common cold or pneumonia.



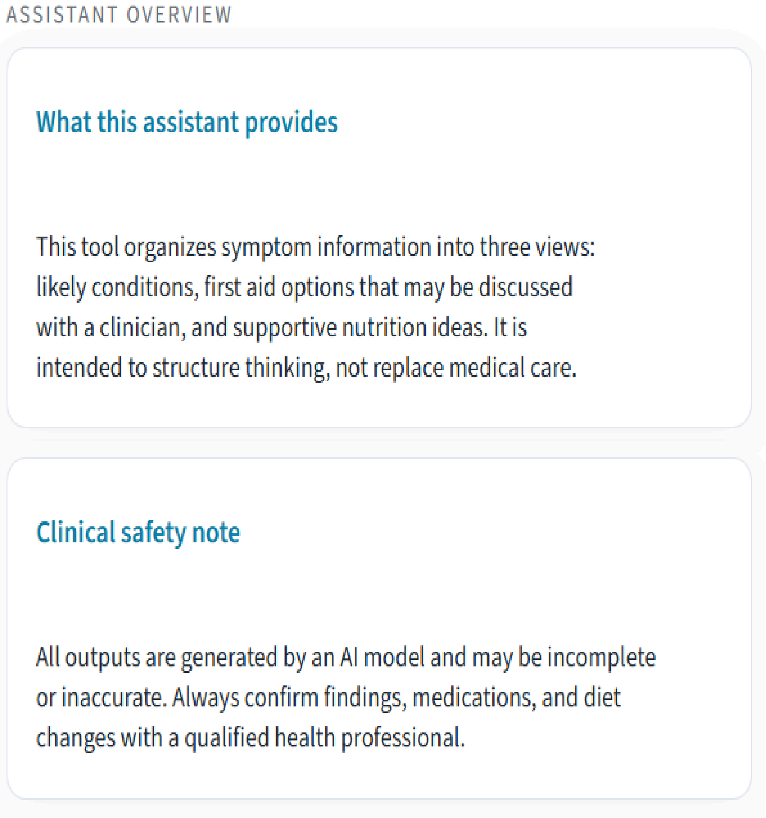
Model backend

Llama (Groq)

Model

llama-3.1-8b-instant

Groq API Key (for Llama)



ASSISTANT OVERVIEW

What this assistant provides

This tool organizes symptom information into three views: likely conditions, first aid options that may be discussed with a clinician, and supportive nutrition ideas. It is intended to structure thinking, not replace medical care.

Clinical safety note

All outputs are generated by an AI model and may be incomplete or inaccurate. Always confirm findings, medications, and diet changes with a qualified health professional.

3.2 REPORTS / DASHBOARDS / MODELS

1. Reports:- Reports provide structured outputs generated by the Healthcare Planning Assistant based on user symptoms and AI analysis.

Types of Reports Generated

1. Symptom Analysis Report
Displays the symptoms entered by the user and summarizes possible related conditions.

2. Health Guidance Report
Provides recommended precautions, lifestyle advice, and general care suggestions.

3. Medication & Nutrition Report
Suggests basic over-the-counter guidance and nutritional improvements (non-prescriptive).

4. AI Response Summary Report
Organizes AI-generated content into clear sections such as:

1.Possible Conditions
2.Precautions
3.Diet Recommendations
4.When to Consult a Doctor

Purpose of Reports

1. Provide clear and structured health information
2. Improve readability of AI responses
3. Help users understand recommendations easily
4. Maintain transparency of AI-generated advice

2. Dashboard:- The dashboard acts as the main interface for interaction.

Dashboard Features-

1.User Input Panel

1. Enter symptoms
2. Select AI model (Gemini / Groq)
3. Enter API key

2.Output Display Section

1. Structured health recommendations
2. Section-based results
3. Clear formatting

3.Error Notification Panel

1. Displays validation errors
2. Shows API connection issues

Dashboard Objectives-

1. Simple and clean user interface

2. Easy symptom submission
3. Real-time result visualization
4. Interactive user experience

3. Models Used in the Project

1. Gemini Model

Google's Large Language Model
1.Used for advanced health reasoning
2.Provides structured and detailed outputs

2. Llama Model

1.High-speed AI inference model

2.Generates quick responses

3. Suitable for real-time applications

3.3 KEY OUTCOMES

The Healthcare Planning Assistant project successfully demonstrates the integration of Artificial Intelligence with a web-based interface to provide intelligent and user-friendly healthcare guidance. The system shows how modern AI technologies can be used to assist individuals in understanding their health conditions and making better health-related decisions. Through this project, an intelligent assistant has been developed that can analyze symptoms, provide healthcare suggestions, and help users plan basic healthcare activities. The project highlights the practical implementation of AI-based decision-making systems in the healthcare domain and proves that such systems can support users in obtaining preliminary medical guidance in a simple and efficient way.

1. AI-Powered Symptom Analysis - The system is capable of analyzing user-entered symptoms using advanced AI models. It generates possible health conditions based on the provided information, showing effective implementation of Large Language Models (LLMs) in healthcare advisory applications.

2. Structured Healthcare Guidance - The system provides organized and structured output including:

1. Possible health conditions
2. Precautionary measures
3. Medication awareness
4. Nutrition recommendations

3. Modular and Scalable Architecture - The project successfully implements a layered and modular architecture. The separation of presentation layer, application logic, and AI orchestration ensures scalability, flexibility, and maintainability of the system.

4. Multi-Model AI Integration - The system supports multiple AI models such as Gemini and Groq. This demonstrates model flexibility and the ability to switch between AI providers without changing the core system structure.

5. User-Friendly Web Interface - Using Streamlit, the project provides an interactive and easy-to-use web interface, allowing users to input symptoms and receive instant AI-generated healthcare guidance.

6. Real-Time Response Generation - The system processes user input and generates responses in real time through secure API communication, ensuring efficient performance and quick output delivery.

7. Ethical and Advisory Design - The project is designed as a healthcare advisory tool rather than a diagnostic system. It promotes responsible AI usage by providing general guidance instead of medical prescriptions.

8. Practical Implementation of AI Concepts - The project demonstrates practical knowledge of:

1. Natural Language Processing
2. Prompt Engineering
3. API Integration
4. Web Application Development

5. Object-Oriented Design

4. CONCLUSION

The Healthcare Planning Assistant project successfully demonstrates the application of Artificial Intelligence in providing preliminary healthcare guidance based on user-entered symptoms. By integrating advanced Large Language Models (LLMs) with a user-friendly web interface, the system is capable of generating structured and meaningful health recommendations in real time.

The project effectively combines Python, Streamlit, LangChain, and AI APIs to create a modular and scalable architecture. The layered design ensures clear separation of responsibilities, making the system maintainable and flexible for future enhancements. The integration of multiple AI models such as Gemini and Groq further enhances adaptability and performance.

The system provides organized outputs including possible health conditions, precautionary measures, medication awareness, and nutrition suggestions. Although it does not replace professional medical diagnosis, it serves as a supportive advisory tool that promotes awareness and informed decision-making.

Overall, the Healthcare Planning Assistant highlights the potential of AI-driven applications in the healthcare domain and demonstrates how intelligent systems can improve accessibility to basic health guidance. The project reflects practical implementation of AI technologies and lays a strong foundation for future expansion into more advanced and personalized healthcare solutions.

5. Future Scope & Enhancements

The Healthcare Planning Assistant is currently designed as an AI-powered advisory system that provides users with symptom-based healthcare guidance and general recommendations. Although the system is effective in offering preliminary health advice, there is significant potential for future improvements that can enhance its accuracy, usability, intelligence, and real-world applicability. Future developments can transform the system from a simple advisory tool into a complete intelligent healthcare support platform. By incorporating advanced technologies such as artificial intelligence, real-time data processing, mobile integration, and secure medical databases, the Healthcare Planning Assistant can become a highly reliable and scalable solution for modern healthcare needs. The following sections describe the major areas where future enhancements can be implemented.

1. Integration with Electronic Health Records (EHR)

One of the most important future improvements is the integration of the Healthcare Planning Assistant with Electronic Health Record (EHR) systems. Currently, the assistant relies mainly on user-provided data such as symptoms and basic personal information. However, integration with EHR systems would allow the assistant to access detailed patient medical records, making the recommendations more personalized and accurate. This integration would allow the system to analyze past medical history, previous treatments, allergies, and chronic conditions. As a result, the system would become more intelligent and capable of providing context-aware healthcare suggestions.

Key Enhancements:

1. Personalized health recommendations based on medical history.
2. Access to complete patient medical records.
3. More accurate disease and symptom analysis.
4. Improved risk prediction and preventive care.
5. Better long-term health monitoring.

2. Real-Time Doctor Consultation Integration

In the future, the Healthcare Planning Assistant can be enhanced by integrating real-time communication features that allow users to connect directly with healthcare professionals. While the current system provides automated recommendations, some situations require professional medical advice. Real-time doctor consultation features would enable users to communicate with certified doctors through video calls, chat systems, or voice communication. This feature would be especially useful when the system detects serious symptoms that require immediate medical attention.

Key Enhancements:

1. Video consultation with certified doctors.
2. Online appointment booking system.
3. Chat-based interaction with healthcare experts.

4. Emergency consultation options.

3. Multi-Language Support

At present, the Healthcare Planning Assistant mainly supports English language interaction. However, many users may not be comfortable communicating in English. Future versions of the system can include multi-language support to make the system accessible to a larger population. The system can automatically detect the user's preferred language and provide responses accordingly. This would improve usability and ensure that healthcare information is easily understood by users from different regions.

Key Enhancements:

1. Support for multiple regional languages.
2. Automatic language detection.
3. Multilingual healthcare instructions.
4. Improved accessibility for rural users.
5. Better user experience.

4. AI Model Fine-Tuning for Healthcare

The current system uses general-purpose Artificial Intelligence models for generating healthcare advice. Although these models are powerful, they are not specifically trained for medical applications. In the future, specialized healthcare AI models can be developed by training them on medical datasets and healthcare knowledge bases. Fine-tuning AI models for healthcare would improve accuracy, reduce errors, and provide more reliable medical suggestions.

Key Enhancements:

1. Development of healthcare-specific AI models.
2. Training using medical datasets.
3. Improved understanding of medical terminology.

4. Better symptom analysis.
5. Higher recommendation accuracy.

5. Mobile Application Development

Currently, the Healthcare Planning Assistant is mainly designed as a web-based system. Future development can include mobile application support, allowing users to access the system through smartphones and tablets. Mobile applications would increase accessibility and convenience, as users could access healthcare guidance anytime and anywhere. Mobile apps would also support push notifications and reminders for better health management.

Key Enhancements:

1. Development of Android applications.
2. Development of iOS applications.
3. Progressive Web Application (PWA) support.
4. Easy access through mobile devices.
5. Health reminders and notifications.

6. Wearable Device Integration

Future versions of the Healthcare Planning Assistant can be integrated with wearable devices such as smartwatches and fitness trackers. These devices can continuously monitor health parameters such as heart rate, physical activity, and sleep patterns. By collecting real-time health data, the system can provide proactive healthcare recommendations and early warnings for potential health problems. This would help users maintain a healthier lifestyle.

Key Enhancements:

1. Integration with smartwatches.
2. Support for fitness bands.
3. Connection with health monitoring devices.

4. Real-time health data tracking.

5. Automatic health alerts.

7. Emergency Alert Detection System

The Healthcare Planning Assistant can be enhanced with an emergency detection system that identifies serious medical conditions. The system can analyze symptoms and detect emergency situations such as chest pain or breathing difficulty. When such situations are detected, the system can immediately alert the user and provide instructions for emergency care. This feature would increase safety and improve the reliability of the system.

Key Enhancements:

1. Detection of emergency symptoms.
2. Automatic emergency alerts.
3. Suggestions for nearby hospitals.
4. First-aid instructions.
5. Quick emergency response guidance.

8. Data Analytics and Reporting Dashboard

Future versions of the system can include advanced data analytics and reporting features. The system can store user health data over time and analyze patterns. This would allow users to track their health improvements and identify potential health risks. Visual reports and charts can help users understand their health condition more easily.

Key Enhancements:

1. Health trend analysis.
2. Symptom history tracking.
3. Graphical health reports.

4. Long-term health monitoring.

9. Security and Privacy Improvements

Healthcare data is highly sensitive, and protecting user information is very important. Future versions of the Healthcare Planning Assistant can include advanced security features to ensure data privacy and protection. Strong authentication and encryption techniques can be implemented to prevent unauthorized access. Compliance with healthcare regulations would also increase trust in the system.

Key Enhancements:

1. End-to-end data encryption.
2. Secure login authentication.
3. Role-based access control.
4. Data protection policies.
5. Compliance with healthcare standards.

10. AI Risk Scoring System

The Healthcare Planning Assistant can be improved by adding an AI-based risk scoring system. This system would analyze symptoms and classify the health condition into different risk levels. Users would be able to understand whether their condition is serious or not. This feature would help users take preventive measures at the right time and reduce health risks.

Key Enhancements:

1. Risk level classification (Low, Medium, High).
2. Severity prediction.
3. Preventive health scoring.
4. Early disease detection.