Architecture Design

Health Care Application

Written By	Shreyash Virendra Chawda
Document Version	***0.1***
Last Revised Date	22-02-2024

Version	Date	Author	Comments
0.1	21-02-2024	Shreyash Virendra Chawda	Architecture, building, Deployment

- # Introduction

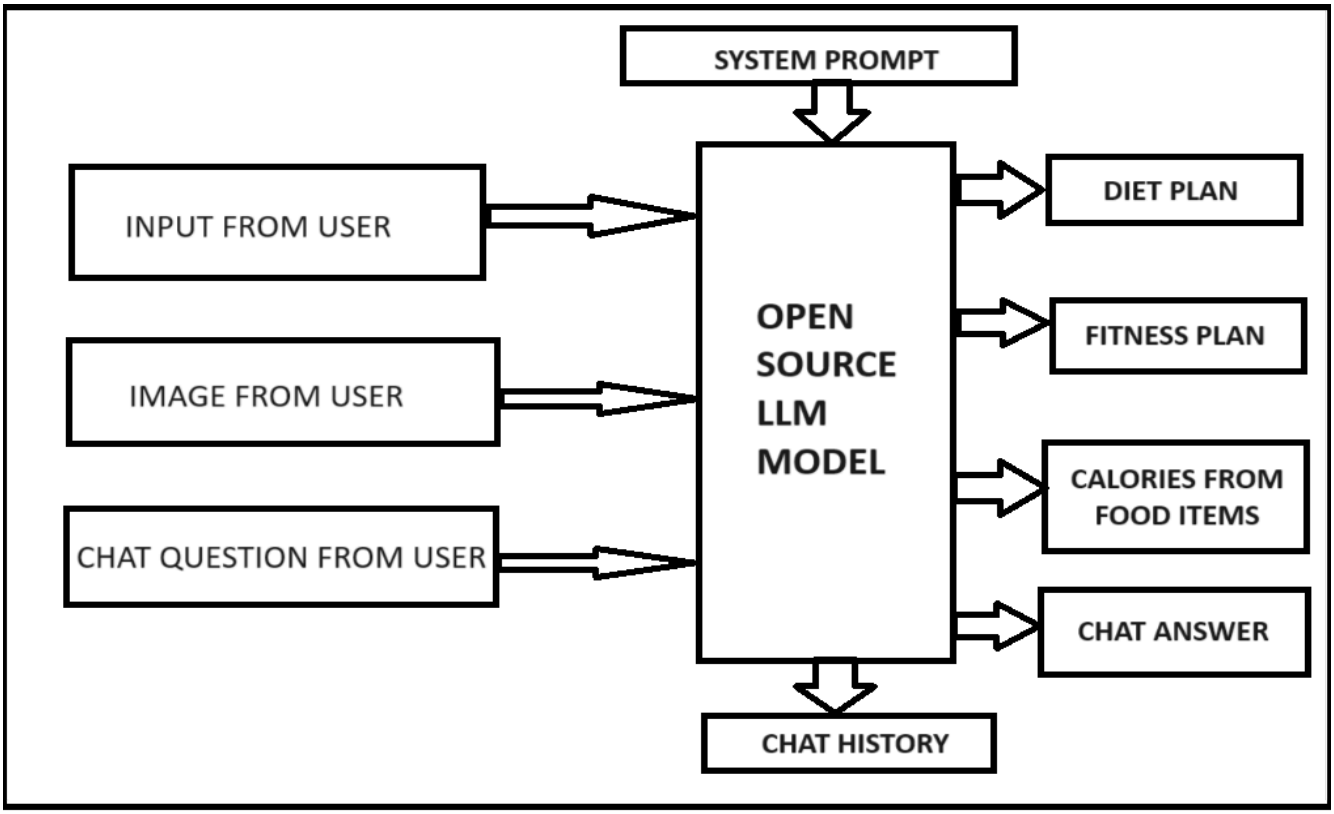
  - ## What is a Low-Level design document?

    The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Health Consulting Application . LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

  - ## Scope

    Low-level design (LLD) is a component-level design process that follows a step-by-Step refinement Architecture process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design.

- # Architecture



- # Architecture Description

Open Source LLM & LIM

Models	Input	Output
Gemini		
• Gemini Pro	Text	Text
• Gemini Pro Vision	Text and images	Text

At their most basic level, large language models (LLMs) are like sophisticated autocomplete applications. Given input text ("You can lead a horse to water,"), LLMs output text that's statistically likely to follow ("but you can't make it drink"), based on patterns learned from their training data.

Prompt Design

Prompt design is the process of creating prompts that elicit the desired response from language models. Writing well structured prompts is an essential part of ensuring accurate, high quality responses from a language model. This page introduces some basic concepts, strategies, and best practices to get you started in designing prompts.

Context Prompt/System Prompt

Context can be one of the following:

- Instructions that specify how the model should behave.
- Information that the model uses or references to generate a response.

Input Prompt

An input is the text in the prompt that you want the model to provide a response for, and it's a required content type. Inputs can be a question that the model answers (question input), a task the model performs (task input), an entity the model operates on (entity input), or partial input that the model completes or continues (completion input).

Question Input Prompt

A question input is a question that you ask the model that the model provides an answer to.

- ## Model Building

  Gemini is a family of models built with multimodal use cases in mind.It is a family of generative AI models developed by Google DeepMind that are designed for multimodal use cases.A multimodal model is a model that is capable of processing information from multiple modalities, including images, and text.The Gemini API gives access to the gemini-pro-vision and gemini-pro models.

- ## API

  API is created using streamlit framework as it is an open-source Python library that allows you to create web applications for data science and machine learning with minimal effort.

  Streamlit Overview:

  1. Purpose:

     - Streamlit simplifies the process of turning data scripts into interactive web applications.

  2. Key Features:

     - Rapid Prototyping: Quickly create interactive apps with only a few lines of code.
     - Widgets: Easily add widgets like sliders, buttons, and text inputs for user interaction.
     - Data Display: Visualize data with charts, tables, and images.
     - Intuitive Syntax: Requires minimal knowledge of web development.

- ## Deployment

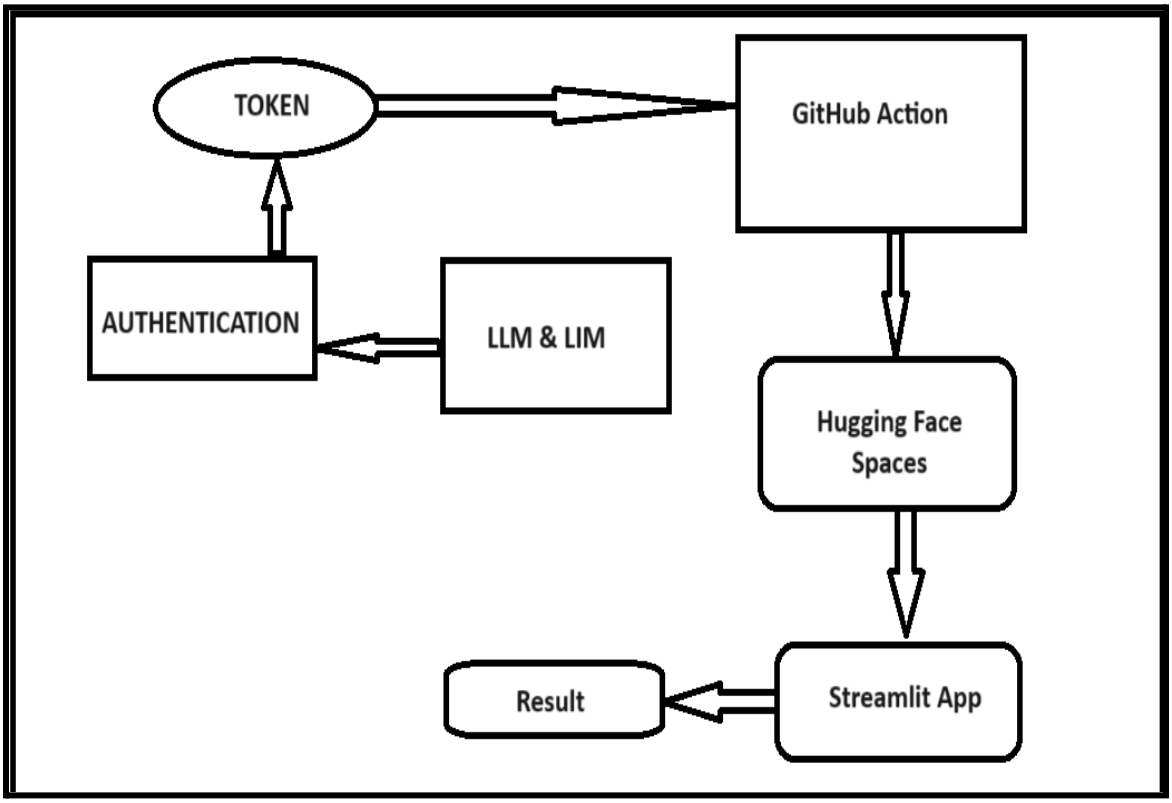
  We will be deploying the model to HuggingFace spaces. Also GitHub Action is used for creating CI/CD Pipeline.

  Workflows for executing whole pipeline starts with pushing code to GitHub with GitHub Actions It is deployed to Hugging Face Spaces from where Application is executing.

  Hugging Face has Access Key as a Token which we have to configure in GitHub as a secret key and according to that configuration CircleCI (CI/CD) pipeline executed.

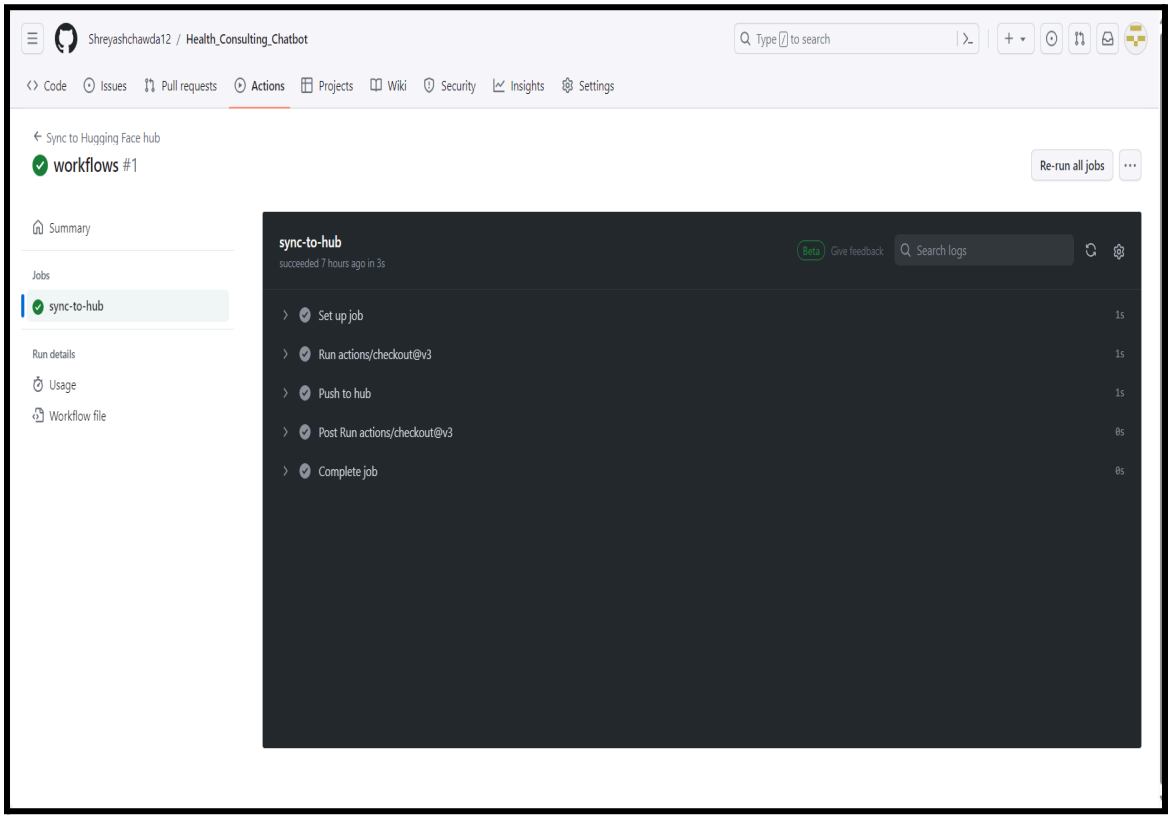
  Also LLM Model contains Secret Access Key which is given to Hugging Face as a secret variable. Hugging Face spaces is integrated with GitHub as similar repository space where we upload our code using GitHub Action

- Architecture Deployment



- GitHub Secret Action Variable From Hugging Face Spaces.

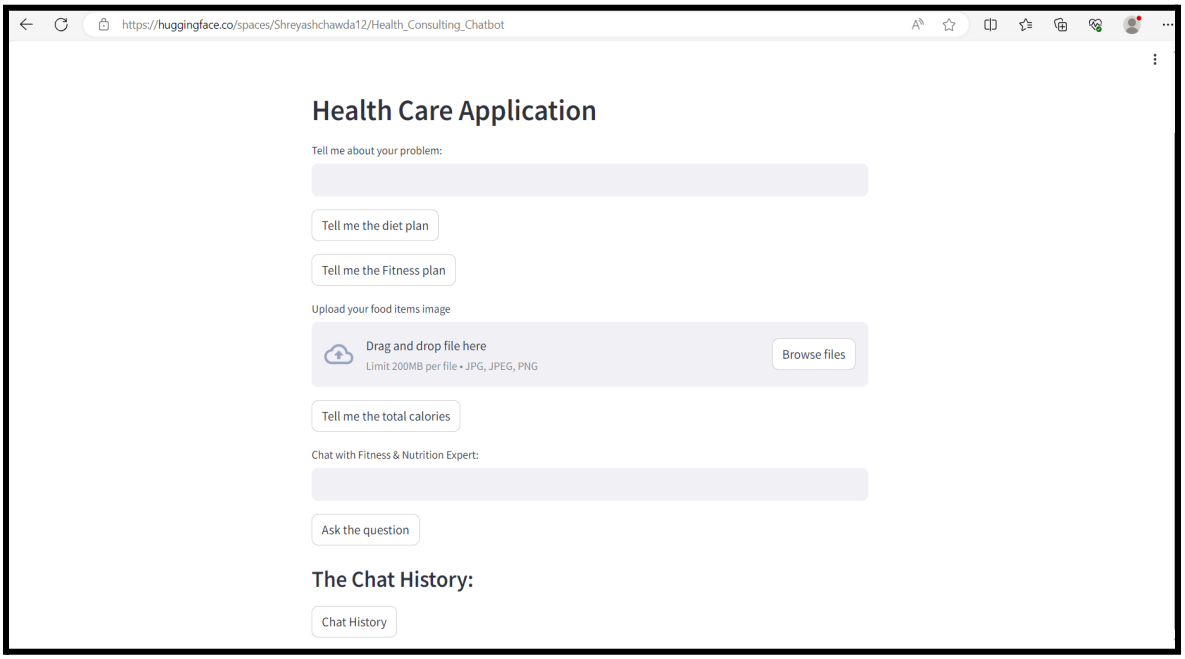
- GitHub Action with CICD Pipeline known as CircleCI



- Hugging Face Secret Variable

Final Application

- Front End Interface using Streamlit Framework and Hugging Face Deployment using GitHub
- Application is divided into 4 parts:
    - Based on Input user will give by sharing there problem related to health it will give a Diet Plan and Fitness Plan according to their work routine.
    - By uploading image of our food items which we eat daily it will give the proper ratio of calorie content and healthy or unhealthy food also recommend alternatives to particular unhealthy food items.
    - If we want to discuss detail about particular health problem there is chat available which gives good advice.
    - The conversation in a chat is recorded and stored as chat history for more reliability.



Application URL:- [Health Consulting Chatbot - a Hugging Face Space by Shreyashchawda12]

GitHub URL:- [https://github.com/Shreyashchawda12/Health_Consulting_Chatbot.git]