AIDRA
AI-based Diagnosis using RAG Architecture

By

AVISHEK MONDAL (CSE21019 / 679)
SOUVIK BAIDYA (CSE21088 / 748)
TAMAL MALLICK (CSE21099 / 759)

Bachelor Thesis submitted to

Indian Institute of Information Technology Kalyani

for the partial fulfillment of the degree of

Bachelor of Technology
in
Computer Science and Engineering

May, 2025

Certificate

This is to certify that the synopsis entitled **"AIDRA: AI-based Diagnosis using RAG Architecture"** is being submitted by Tamal Mallick, (**Enrollment No: CSE/21099/759**), Avishek Mondal (**Enrollment No: CSE21019/679**) and Souvik Baidya (**Enrollment No: CSE21088/748**), B.Tech., Indian Institute of Information Technology Kalyani, India, for the partial fulfillment of the requirements for the registration of the degree of Bachelor of Technology is an original research work carried by them under my supervision. The synopsis has fulfilled all the requirements as per the regulation of IIIT Kalyani and in my opinion, has reached the standards needed for submission. The works, techniques, and results presented have not been submitted to any other university or Institute for the award of any other degree or diploma.

Dr. Anirban Lakshman
Assistant Professor
Department of Mathematics
Indian Institute of Information Technology Kalyani
Kalyani - 741235, W.B., India.

Declaration

We hereby declare that the work being submitted in this thesis entitled,**"AIDRA: AI-based Diagnosis using RAG Architecture"**, submitted to Indian Institute of Information Technology, Kalyani in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering during the period from July, 2024 to November 2024 under the supervision of Dr. Anirban Lakshman, Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India, does not contain any classified information.

Tamal Mallick
Enrollment No.: CSE/21099/759
Indian Institute of Information Technology Kalyani
Kalyani - 741235, W.B., India.

Avishek Mondal
Enrollment No.: CSE/21019/679
Indian Institute of Information Technology Kalyani
Kalyani - 741235, W.B., India.

Souvik Baidya
Enrollment No.: CSE21088/748
Indian Institute of Information Technology Kalyani
Kalyani - 741235, W.B., India.

Acknowledgement

We hereby acknowledge our deep sense of gratitude to our supervisor **Dr. Anirban Lakshman**, Department of Mathematics, Indian Institute of Information Technology Kalyani, for providing us with adequate facilities, ways, and means by which we are able to do this work. We express our sincere gratitude to him, his valuable time, insightful suggestions, and constant support have been indispensable for our progress.

Finally, we would also like to thank our friends, college faculties and family members who in one way or another helped us in doing this work.

Tamal Mallick
Enrollment No.: CSE/21099/759
Indian Institute of Information Technology Kalyani
Kalyani - 741235, W.B., India.

Avishek Mondal
Enrollment No.: CSE/21019/679
Indian Institute of Information Technology Kalyani
Kalyani - 741235, W.B., India.

Souvik Baidya
Enrollment No.: CSE21088/748
Indian Institute of Information Technology Kalyani
Kalyani - 741235, W.B., India.

Abstract

This project, entitled **"AIDRA: AI-based Diagnosis using RAG Architecture"**, presents the development of an intelligent medical chatbot designed to assist users with accurate and trustworthy health-related information. Unlike traditional chatbots or wellness assistants focused solely on general advice or alternative therapies, this system leverages state-of-the-art Large Language Models (LLMs) integrated with Retrieval-Augmented Generation (RAG) to deliver context-aware, medically accurate responses [6, 7].

The chatbot sources its information from globally respected clinical references such as the *Oxford Handbook of Clinical Medicine*, *Harrison's Principles of Internal Medicine*, and *The Merck Manual* (referenced in the indexed dataset, not explicitly cited here due to lack of publication detail in your '.bib').

The core functionalities of the chatbot include analyzing user-described symptoms, suggesting relevant diagnostic tests, identifying possible conditions, and recommending appropriate medications when a confident match is found in the context. It also provides lifestyle and dietary advice tailored to specific conditions. To ensure efficient and relevant information retrieval, the project utilizes HuggingFace embeddings [8] and the Pinecone vector database [9], managed through the LangChain framework [7]. The system is deployed via a Flask web interface, offering users an interactive and responsive experience.

By combining trusted medical literature with advanced natural language processing, the chatbot aims to simulate the reasoning of a professional physician in non-critical scenarios, empowering users with reliable medical insights while clearly avoiding emergency diagnosis use cases.

Contents

1 Introduction 2
1.1 Background 2
1.2 Motivation 2
1.3 Problem Statement 2
1.4 Objectives 3
1.5 Scope of Work 4
2 Literature Review 6
2.1 Existing Chatbots in Healthcare 6
2.2 Generative AI and Medical Applications 6
2.3 Retrieval-Augmented Generation (RAG) 6
2.4 Embedding Models and Vector Databases 6
2.5 Prompt Engineering 7
3 System Design and Architecture 9
3.1 Workflow Overview 9
3.2 Architecture Diagram 10
3.3 Data Pipeline Flow 10
3.4 Technologies Used 11
3.5 Security Environment 11
4 Implementation 13
4.1 Data Processing 13
4.2 Code Overview 15
4.3 Embedding and Indexing Pipeline 17
4.4 Retrieval and Response Generation 17
4.5 Flask Application Flow 17
5 Results and Evaluation 20
5.1 Overview 20
5.2 Evaluation Methodology 20
5.3 Test Cases 20
5.4 Evaluation Results 21
5.5 Limitations Identified 25
6 Application and Future Work 28
6.1 Application 28
6.2 Future Work 29
7 Conclusion 32
8 Conclusion 32

Abbreviations Used

Abbreviation	**Full Form**
Artificial Intelligence and Machine Learning	
RAG	Retrieval-Augmented Generation
LLM	Large Language Model
STT	Speech-to-Text
TTS	Text-to-Speech
ML	Machine Learning
AI	Artificial Intelligence
NLP	Natural Language Processing
MLOps	Machine Learning Operations
General Healthcare	
FPG	Fasting Plasma Glucose
HbA1c	Hemoglobin A1c
CT	Computed Tomography
eGFR	Estimated Glomerular Filtration Rate
KUB	Kidneys, Ureters, and Bladder X-ray
MRI	Magnetic Resonance Imaging
WGS	Whole Genome Sequencing
CBC	Complete Blood Count
WHO	World Health Organization
CDC	Centers for Disease Control and Prevention
Technology and Infrastructure	
API	Application Programming Interface
GUI	Graphical User Interface
VM	Virtual Machine
SSL	Secure Sockets Layer
DB	Database
DevOps	Development and Operations

Table 1: List of Abbreviations Used in the Project

Chapter 1

1 Introduction

1.1 Background

The use of AI-driven chatbots in the healthcare domain is evolving rapidly, particularly with the advent of advanced large language models (LLMs). These models are capable of understanding and generating human-like responses, making them suitable for developing conversational agents. However, traditional healthcare chatbots have limitations in personalization, adaptability, and factual accuracy, especially in niche domains like alternative medicine. The concept of Retrieval-Augmented Generation (RAG) offers a promising solution by enriching LLMs with real-world knowledge from curated sources. This project, titled **AIDRA** leverages RAG to deliver a chatbot capable of answering user queries in the field of holistic and alternative medicine.‘

1.2 Motivation

Access to **quality healthcare** remains a **critical challenge** in many parts of the world. According to global health reports, at least **half of the population** lacks access to essential health services [1]. Moreover, over **1 billion people** are pushed into extreme poverty annually due to **out-of-pocket medical expenses**, especially in low- and middle-income countries [1] .

In countries like **India**, approximately **55 million people** fall below the poverty line each year due to unaffordable healthcare [2]. Even for **minor but untreated medical conditions**, the lack of timely consultation often leads to **severe complications or even death**.

Additionally, it is reported that **2.6 million people** die annually as a result of **unsafe medical practices** [3]. A significant portion of these deaths could be prevented if **reliable and early medical advice** were accessible to all.

While **AI-based solutions** are increasingly being used in healthcare, most **generic chatbots hallucinate** or provide **inaccurate information**, especially when queried for **specialized medical advice**. This motivated the development of a **Retrieval-Augmented Generation (RAG)-based Medical Chatbot**, which we have named the **AIDRA**. Our chatbot utilizes trusted medical references like *Harrison's Principles of Internal Medicine*, the *Oxford Handbook of Clinical Medicine*, and the *Merck Manual*, to ensure that the information it provides is **credible, contextual, and verifiable**.

This project aims to **democratize access** to trustworthy medical knowledge, especially for those who **cannot afford quality healthcare**, by providing **symptom-based diagnosis support**, **test suggestions**, and **prescription-level medical guidance** powered by **reliable clinical sources**.

1.3 Problem Statement

Despite their linguistic capabilities, general-purpose **large language models (LLMs)** struggle to offer **domain-specific**, context-aware, and explainable medical advice.

These models often respond with **hallucinated outputs** that may seem convincing but are not grounded in verified clinical data.

This limitation poses a significant risk in the medical field, where **accuracy**, **safety**, and source **verifiability** are non-negotiable. Furthermore, these models typically do not offer **traceability** of their responses, making it difficult for users to trust or validate the suggestions.

The challenge lies in building a chatbot system that:

- Can retrieve **real-time relevant content** from a trusted medical database;
- Contextualizes the patient's input (symptoms, test reports, etc.) before generating a response;
- Prescribes medications or suggests medical tests only when backed by **validated medical sources**.

Our solution, AIDRA, combines **retrieval-augmented generation** techniques with advanced LLMs to simulate the behavior of a virtual medical assistant. By leveraging a **vector database** for storing and querying reliable medical knowledge, the system ensures that its recommendations are grounded in **evidence** and **transparent** to the user.

1.4 Objectives

The primary goal of this project is to build an intelligent medical assistant that delivers reliable, context-aware, and explainable healthcare guidance.

- To create an **LLM-powered virtual medical specialist** capable of providing **reliable medication suggestions and prescriptions** based on trusted sources.
- To design a chatbot that utilizes **Retrieval-Augmented Generation (RAG)** to ensure **higher response accuracy** and **contextual relevance**.
- To integrate **LangChain**, **Gemini LLM**, and **Pinecone** into a complete **retrieval and generation pipeline** for intelligent medical dialogue.
- To provide users with a **simple and intuitive interface** for querying holistic and modern medical content.
- To ground all responses using a **curated dataset** derived from world-renowned medical references, including the *Gale Encyclopedia of Alternative Medicine*, the *Oxford Handbook of Clinical Medicine*, *Harrison's Principles of Internal Medicine*, and *The Merck Manual.*

1.5 Scope of Work

Functional Scope

The medical chatbot is designed to deliver **intelligent**, **context-aware responses** to user health-related queries. Its scope extends beyond general wellness or alternative therapies, offering **clinically grounded guidance**, including potential diagnoses, diagnostic test recommendations, and medication suggestions based on trusted clinical literature like the *Oxford Handbook of Clinical Medicine*, *Harrison's Principles of Internal Medicine*, and *The Merck Manual*.

While the system is **not intended for emergency use or life-threatening conditions**, it aims to simulate the reasoning and response behavior of a **professional doctor** in non-critical scenarios. The chatbot can:

- Analyze user-described **symptoms**,
- Suggest relevant **diagnostic tests**,
- Propose possible **conditions or diseases**,
- Recommend **medications** (only when confidence is high and backed by context),
- Provide **dietary and lifestyle advice** related to the diagnosed or suspected condition.

Technical Scope

The technical components that define the system include:

- **Document ingestion and preprocessing**,
- **Embedding generation and vector storage** using Pinecone,
- Implementation of **Retrieval-Augmented Generation (RAG)** with LangChain,
- Development of a **custom contextual memory system**,
- **Prompt engineering** to enforce structured, medically accurate outputs,
- **Web deployment** via a Flask-based interface for end-user interaction.

Chapter 2

2 Literature Review

This section presents a review of existing systems and research in the field of AI-driven medical chatbots, with a focus on their capabilities, limitations, and relevance to the development of the AIDRA system. It explores advancements in healthcare chatbots, generative AI models, and the integration of Retrieval-Augmented Generation (RAG) with vector databases. The review also includes a discussion on prompt engineering practices that help ensure medically accurate, trustworthy, and explainable outputs in conversational healthcare applications.

2.1 Existing Chatbots in Healthcare

Several healthcare chatbots, such as **Ada Health**, **Babylon Health**, and **HealthTap**, are widely used in digital healthcare [5]. These platforms primarily focus on **symptom analysis** and **rule-based triage** using structured datasets. While they are effective for **basic diagnostics**, their **general-purpose architecture** limits adaptability to **complex or domain-specific applications**, such as medically grounded prescription generation or handling of alternative therapies.

2.2 Generative AI and Medical Applications

Recent advancements in **Generative AI** models like **ChatGPT** (OpenAI), **Bard** (Google), and **Claude** (Anthropic) have shown remarkable progress in **natural language understanding** and **response generation**. However, these models often lack **domain grounding**, leading to **speculative or inaccurate responses** when queried for medical advice [6, 4]. In healthcare applications, **factual correctness**, **explainability**, and **source traceability** are critical — which general LLMs often fail to consistently deliver.

2.3 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) is a paradigm that enhances LLM performance by combining **document retrieval** with **generative response** [7]. Instead of depending solely on the model's internal parameters, RAG retrieves relevant context documents from a **vector store**, ensuring that outputs are **context-aware** and **factually grounded**. **LangChain**, a Python framework used in this project, simplifies this integration by offering modular components for **document loading**, **chunking**, **embedding**, **retrieval**, and **LLM orchestration** [7].

2.4 Embedding Models and Vector Databases

This project uses **HuggingFace's all-MiniLM-L6-v2**, a transformer-based sentence embedding model that captures **semantic similarity** between texts [8]. These embeddings are indexed in **Pinecone**, a high-speed, scalable vector database, allowing efficient **semantic search** to retrieve relevant chunks during inference [9]. This architecture forms the backbone of the AIDRA system, enabling reliable information grounding.

2.5 Prompt Engineering

Prompt engineering is pivotal for customizing LLM outputs in task-specific scenarios. In this project, a carefully crafted prompt instructs the **Gemini LLM** to behave like a professional doctor, **Dr. Sushruta**, drawing information only from trusted medical references. The prompt format ensures that answers are **context-restricted**, **medically relevant**, and delivered in a **structured JSON format**. This approach helps **minimize hallucination** and encourages **clinically appropriate response behavior** [4].

Chapter 3

3 System Design and Architecture

3.1 Workflow Overview

The architecture of AIDRA is designed around a modular **Retrieval-Augmented Generation (RAG)** pipeline. Each component is independently structured to allow for **debugging**, **modular development**, and **future enhancements**. The major components are:

- **Document Loader** – Loads medical documents in `.pdf` format.
- **Text Chunker** – Splits the documents into manageable text chunks.
- **Embedding Generator** – Converts text chunks into semantic vectors using a **transformer model** [8].
- **Vector Store (Pinecone)** – Stores and indexes vector embeddings for efficient retrieval [9].
- **Retriever** – Fetches top-k relevant chunks based on the user query.
- **Language Model (Gemini)** – Generates responses using retrieved context and prompt guidance.
- **Web Interface (Flask)** – Provides a user-friendly front-end for interaction.

These components together form a **semantic search** and **generation system**, capable of grounding responses in trusted medical literature.

3.2 Architecture Diagram

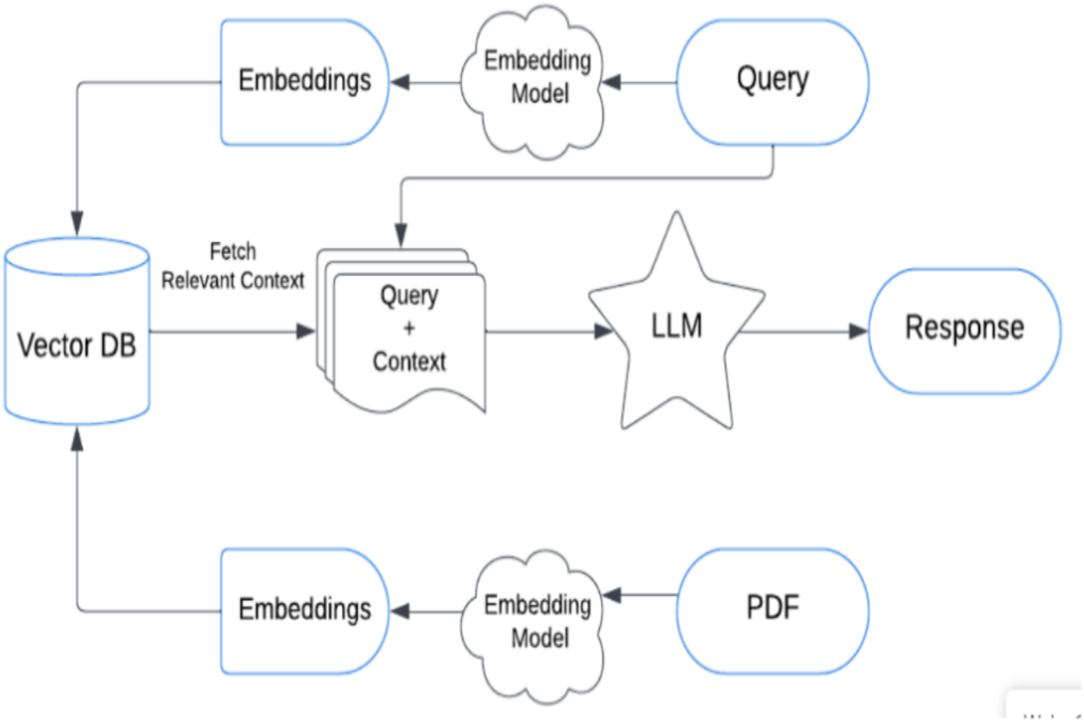


Figure 1: Architecture Diagram

3.3 Data Pipeline Flow

The core data flow of the system proceeds through the following steps:

1. Load medical documents (PDF format) into the system.
2. Split text into semantically coherent chunks using a **recursive text splitter**.
3. Generate embeddings for each chunk using **all-MiniLM-L6-v2** from **HuggingFace** [8].
4. Store embeddings in **Pinecone**, a high-speed vector database [9].
5. Upon a user query:
   (a) Use **retriever** to find the top-k most relevant chunks.
   (b) Inject retrieved chunks into a custom prompt designed for **Gemini LLM**.
6. **Gemini** processes the query and context to generate a clinically relevant response.
7. The response is displayed via a user interface built with **Flask** and **Jinja2**.

3.4 Technologies Used

Technology	**Purpose**
Python 3.x	Core backend programming language
Flask	Web application framework
Jinja2	HTML templating engine used by Flask
LangChain	Orchestrating Retrieval-Augmented Generation (RAG) pipelines
HuggingFace Transformers	Embedding generation using MiniLM model
Pinecone	Scalable vector similarity search
Gemini API	Large Language Model (LLM) for text generation
Azure Ubuntu Server	For hosting the service
SSL Certificate	Ensures secure communication (HTTPS) between the server and users
Gunicorn	WSGI HTTP server for serving the Flask app in production
Nginx	Reverse proxy server for load balancing, caching, and SSL termination

Table 2: Technologies and their purposes in the system

3.5 Security Environment

To ensure **security** and **scalability**, the following measures are implemented:

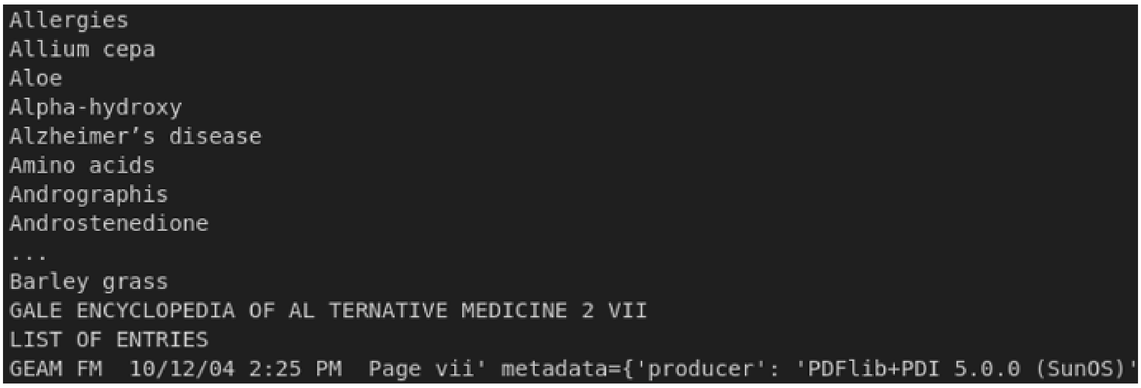
- **API keys and credentials** are stored in a `.env` file and loaded securely using the `python-dotenv` library [11].
- All **model access logic** is abstracted into backend modules to prevent direct exposure in the user interface.
- **User input** is sanitized before embedding or prompt injection to avoid prompt injection or code injection attacks [12].
- The system is **deployable locally or on cloud environments** (e.g., Render, Replit, or Heroku) with minimal changes.
- One version of the system is already **deployed on an Azure Ubuntu Server** to ensure scalability and reliability in a production environment [13].
- The system is **load balanced** with **Gunicorn**, ensuring scalability and performance under heavy traffic [14].
- **SSL certificates** have been added to ensure secure communication between the server and users, providing encrypted data transmission via HTTPS [15].

Chapter 4

4 Implementation

4.1 Data Processing

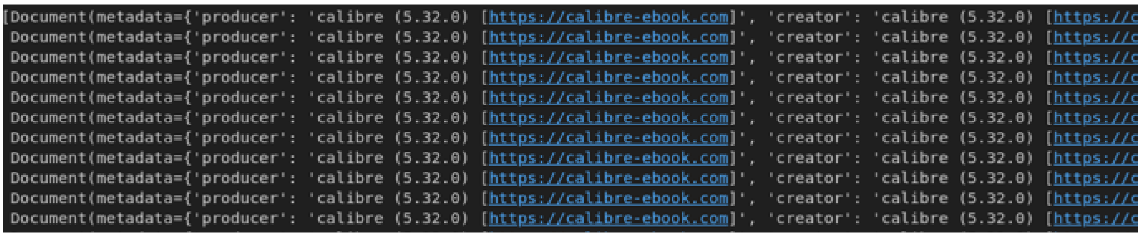
Raw extracted text



```
Allergies
Allium cepa
Aloe
Alpha-hydroxy
Alzheimer’s disease
Amino acids
Andrographis
Androstenedione
...
Barley grass
GALE ENCYCLOPEDIA OF AL TERNATIVE MEDICINE 2 VII
LIST OF ENTRIES
GEAM FM  10/12/04 2:25 PM  Page vii' metadata={'producer': 'PDFlib+PDI 5.0.0 (SunOS)'
```

Figure 2: Raw extracted text

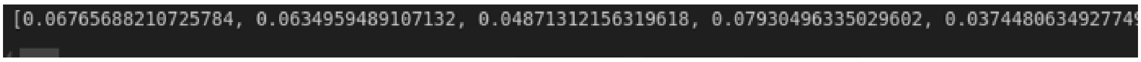
Text Chunks



```
[Document(metadata={'producer': 'calibre (5.32.0) [https://calibre-ebook.com]', 'creator': 'calibre (5.32.0) [https://c
 Document(metadata={'producer': 'calibre (5.32.0) [https://calibre-ebook.com]', 'creator': 'calibre (5.32.0) [https://c
 Document(metadata={'producer': 'calibre (5.32.0) [https://calibre-ebook.com]', 'creator': 'calibre (5.32.0) [https://c
 Document(metadata={'producer': 'calibre (5.32.0) [https://calibre-ebook.com]', 'creator': 'calibre (5.32.0) [https://c
 Document(metadata={'producer': 'calibre (5.32.0) [https://calibre-ebook.com]', 'creator': 'calibre (5.32.0) [https://c
 Document(metadata={'producer': 'calibre (5.32.0) [https://calibre-ebook.com]', 'creator': 'calibre (5.32.0) [https://c
 Document(metadata={'producer': 'calibre (5.32.0) [https://calibre-ebook.com]', 'creator': 'calibre (5.32.0) [https://c
 Document(metadata={'producer': 'calibre (5.32.0) [https://calibre-ebook.com]', 'creator': 'calibre (5.32.0) [https://c
 Document(metadata={'producer': 'calibre (5.32.0) [https://calibre-ebook.com]', 'creator': 'calibre (5.32.0) [https://c
 Document(metadata={'producer': 'calibre (5.32.0) [https://calibre-ebook.com]', 'creator': 'calibre (5.32.0) [https://c
 Document(metadata={'producer': 'calibre (5.32.0) [https://calibre-ebook.com]', 'creator': 'calibre (5.32.0) [https://c
```

Figure 3: Total Chunk size: 107476

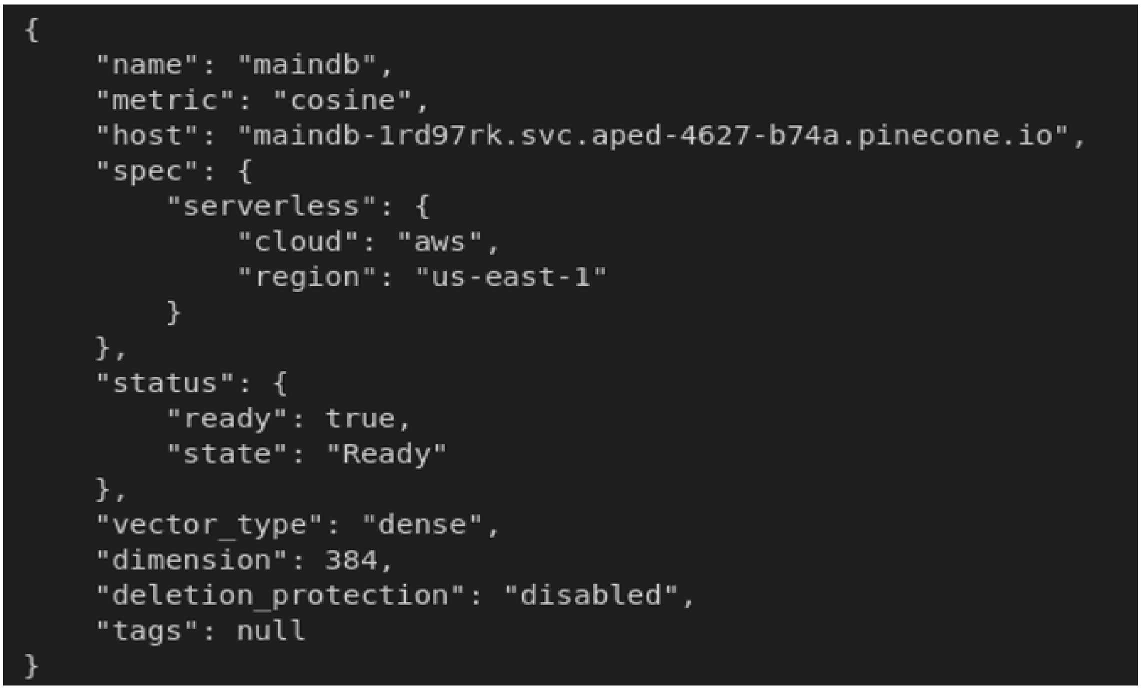
Vectorized Chunks



```
[0.06765688210725784, 0.0634959489107132, 0.04871312156319618, 0.07930496335029602, 0.0374480634927749
```

Figure 4: Total Length: 384

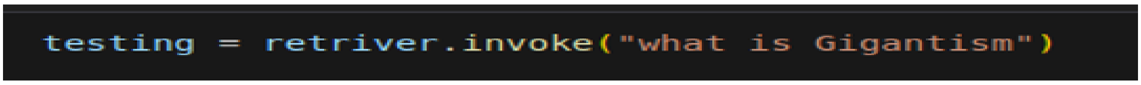
Vector Database



```
{
    "name": "maindb",
    "metric": "cosine",
    "host": "maindb-1rd97rk.svc.aped-4627-b74a.pinecone.io",
    "spec": {
        "serverless": {
            "cloud": "aws",
            "region": "us-east-1"
        }
    },
    "status": {
        "ready": true,
        "state": "Ready"
    },
    "vector_type": "dense",
    "dimension": 384,
    "deletion_protection": "disabled",
    "tags": null
}
```

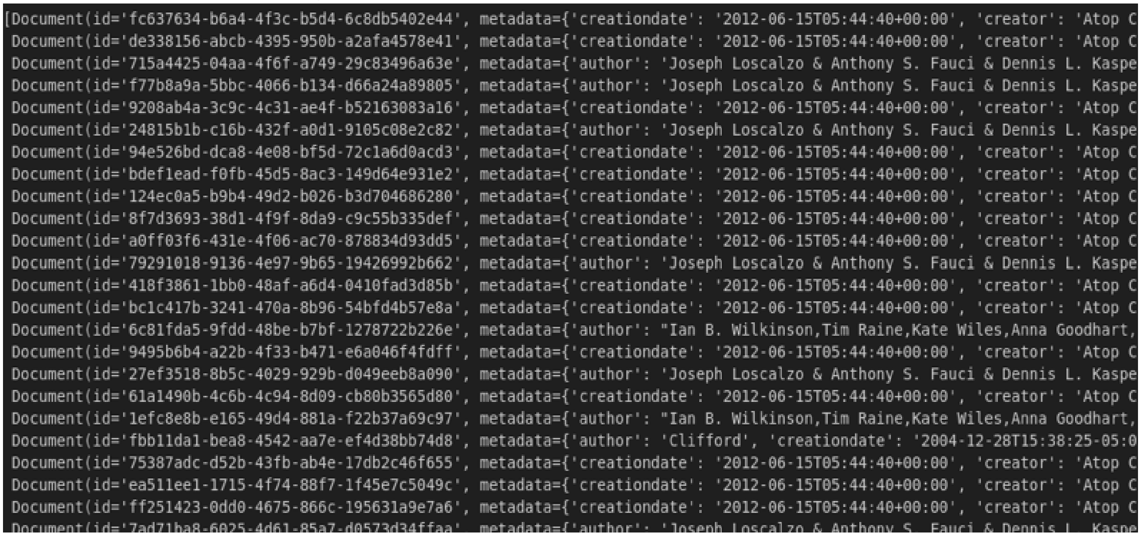
Figure 5: Vector Database

Raw Question and Retrieved data



```
testing = retriver.invoke("what is Gigantism")
```

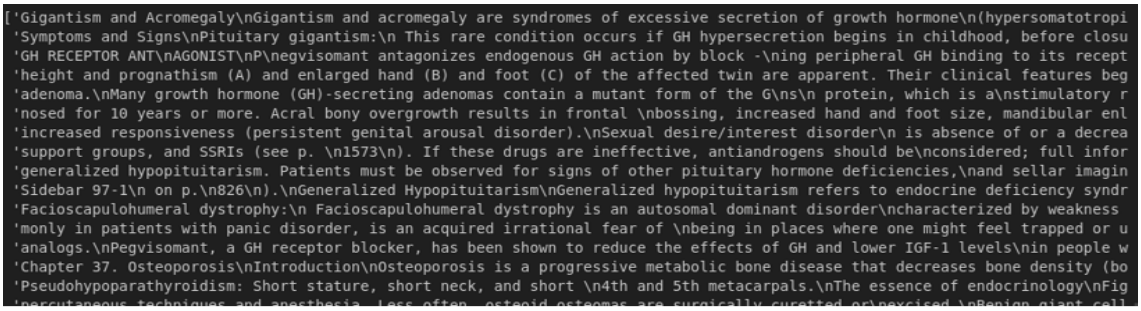
Figure 6: Quires for Information Retrieval



```
[Document(id='fc637634-b6a4-4f3c-b5d4-6c8db5402e44', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='de338156-abcb-4395-950b-a2afa4578e41', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='715a4425-04aa-4f6f-a749-29c83496a63e', metadata={'author': 'Joseph Loscalzo & Anthony S. Fauci & Dennis L. Kaspe
 Document(id='f77b8a9a-5bbc-4066-b134-d66a24a89805', metadata={'author': 'Joseph Loscalzo & Anthony S. Fauci & Dennis L. Kaspe
 Document(id='9208ab4a-3c9c-4c31-ae4f-b52163083a16', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='24815b1b-c16b-432f-a0d1-9105c08e2c82', metadata={'author': 'Joseph Loscalzo & Anthony S. Fauci & Dennis L. Kaspe
 Document(id='94e526bd-dca8-4e08-bf5d-72c1a6d0acd3', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='bdef1ead-f0fb-45d5-8ac3-149d64e931e2', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='124ec0a5-b9b4-49d2-b026-b3d704686280', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='8f7d3693-38d1-4f9f-8da9-c9c55b335def', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='a0ff03f6-431e-4f06-ac70-878834d93dd5', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='79291018-9136-4e97-9b65-19426992b662', metadata={'author': 'Joseph Loscalzo & Anthony S. Fauci & Dennis L. Kaspe
 Document(id='418f3861-1bb0-48af-a6d4-0410fad3d85b', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='bc1c417b-3241-470a-8b96-54bfd4b57e8a', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='6c81fda5-9fdd-48be-b7bf-1278722b226e', metadata={'author': "Ian B. Wilkinson,Tim Raine,Kate Wiles,Anna Goodhart,
 Document(id='9495b6b4-a22b-4f33-b471-e6a046f4fdff', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='27ef3518-8b5c-4029-929b-d049eeb8a090', metadata={'author': 'Joseph Loscalzo & Anthony S. Fauci & Dennis L. Kaspe
 Document(id='61a1490b-4c6b-4c94-8d09-cb80b3565d80', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='1efc8e8b-e165-49d4-881a-f22b37a69c97', metadata={'author': "Ian B. Wilkinson,Tim Raine,Kate Wiles,Anna Goodhart,
 Document(id='fbb11da1-bea8-4542-aa7e-ef4d38bb74d8', metadata={'author': 'Clifford', 'creationdate': '2004-12-28T15:38:25-05:0
 Document(id='75387adc-d52b-43fb-ab4e-17db2c46f655', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='ea511ee1-1715-4f74-88f7-1f45e7c5049c', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='ff251423-0dd0-4675-866c-195631a9e7a6', metadata={'creationdate': '2012-06-15T05:44:40+00:00', 'creator': 'Atop C
 Document(id='7ad71ba8-6025-4d61-85a7-d0573d34ffaa', metadata={'author': 'Joseph Loscalzo & Anthony S. Fauci & Dennis L. Kaspe
```

Figure 7: Raw Retrieved contextual data

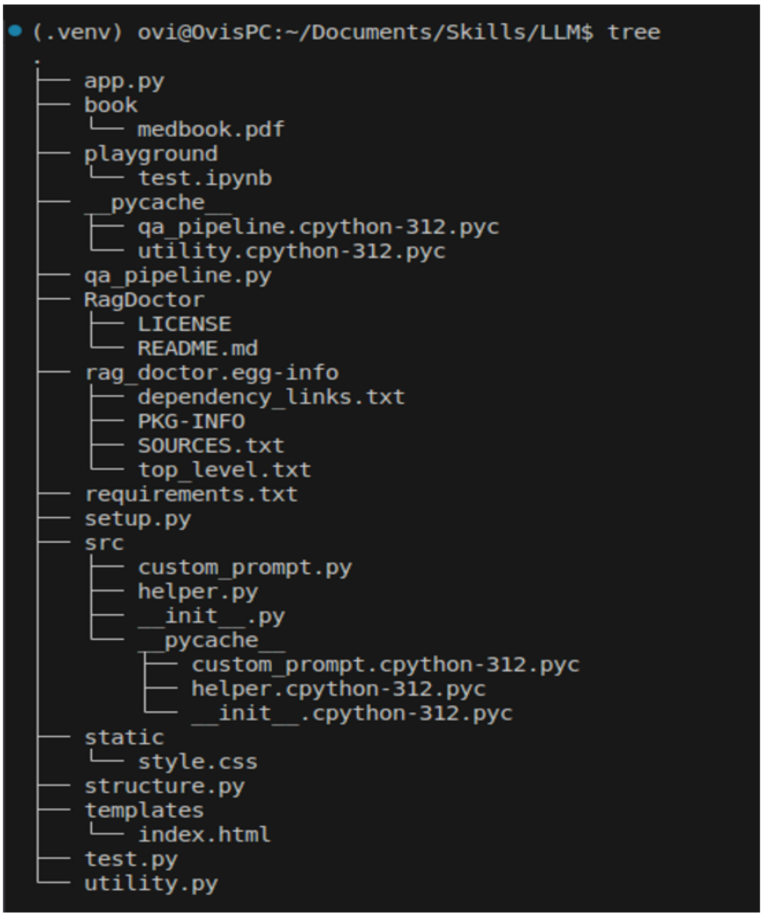
Clean Data



```
['Gigantism and Acromegaly\nGigantism and acromegaly are syndromes of excessive secretion of growth hormone\n(hypersomatotropi
 'Symptoms and Signs\nPituitary gigantism:\n This rare condition occurs if GH hypersecretion begins in childhood, before closu
 'GH RECEPTOR ANT\nAGONIST\nP\negvisomant antagonizes endogenous GH action by block -\ning peripheral GH binding to its recept
 'height and prognathism (A) and enlarged hand (B) and foot (C) of the affected twin are apparent. Their clinical features beg
 'adenoma.\nMany growth hormone (GH)-secreting adenomas contain a mutant form of the G\ns\n protein, which is a\nstimulatory r
 'nosed for 10 years or more. Acral bony overgrowth results in frontal \nbossing, increased hand and foot size, mandibular enl
 'increased responsiveness (persistent genital arousal disorder).\nSexual desire/interest disorder\n is absence of or a decrea
 'support groups, and SSRIs (see p. \n1573\n). If these drugs are ineffective, antiandrogens should be\nconsidered; full infor
 'generalized hypopituitarism. Patients must be observed for signs of other pituitary hormone deficiencies,\nand sellar imagin
 'Sidebar 97-1\n on p.\n826\n).\nGeneralized Hypopituitarism\nGeneralized hypopituitarism refers to endocrine deficiency syndr
 'Facioscapulohumeral dystrophy:\n Facioscapulohumeral dystrophy is an autosomal dominant disorder\ncharacterized by weakness
 'monly in patients with panic disorder, is an acquired irrational fear of \nbeing in places where one might feel trapped or u
 'analogs.\nPegvisomant, a GH receptor blocker, has been shown to reduce the effects of GH and lower IGF-1 levels\nin people w
 'Chapter 37. Osteoporosis\nIntroduction\nOsteoporosis is a progressive metabolic bone disease that decreases bone density (bo
 'Pseudohypoparathyroidism: Short stature, short neck, and short \n4th and 5th metacarpals.\nThe essence of endocrinology\nFig
 'percutaneous techniques and anesthesia. Less often, osteoid osteomas are surgically curetted or\nexcised.\nBenign giant cell
```

Figure 8: Context Without metadata will be passed to LLM

Project Directory Structure



```
● (.venv) ovi@OvisPC:~/Documents/Skills/LLM$ tree
.
├── app.py
├── book
│   └── medbook.pdf
├── playground
│   └── test.ipynb
├── __pycache__
│   ├── qa_pipeline.cpython-312.pyc
│   └── utility.cpython-312.pyc
├── qa_pipeline.py
├── RagDoctor
│   ├── LICENSE
│   └── README.md
├── rag_doctor.egg-info
│   ├── dependency_links.txt
│   ├── PKG-INFO
│   ├── SOURCES.txt
│   └── top_level.txt
├── requirements.txt
├── setup.py
├── src
│   ├── custom_prompt.py
│   ├── helper.py
│   ├── __init__.py
│   └── __pycache__
│       ├── custom_prompt.cpython-312.pyc
│       ├── helper.cpython-312.pyc
│       └── __init__.cpython-312.pyc
├── static
│   └── style.css
├── structure.py
├── templates
│   └── index.html
├── test.py
└── utility.py
```

Figure 9: Project Directory Structure

4.2 Code Overview

app.py

- Initializes the Flask application and loads all required environment variables.

- Sets up the retriever and custom prompt logic.
- Handles user `POST` requests, generates answers, and displays them through the chat interface.
- Controls the application flow between frontend and backend.

utility.py

Defines utility functions for:

- Loading PDF documents
- Text splitting and preprocessing
- Embedding creation
- Checking and managing Pinecone vector index
- Though not fully utilized in the current build, it provides support for modular scalability.

qa_pipeline.py

- Contains the core logic for integrating:
  - Document loading and chunking
  - Embedding generation with HuggingFace
  - Storage in Pinecone
  - Query retrieval and Gemini response synthesis via LangChain QA chain [7]

test.py

- Serves as a sandbox for testing new features, validating outputs, and running experiments on pipeline components.

playground/test.ipynb

- Jupyter Notebook for interactive debugging and observing response behaviors.
- Used for fine-tuning prompt templates and inspecting Gemini outputs during development.

4.3 Embedding and Indexing Pipeline

The document ingestion and embedding process consists of the following steps:

- Load PDFs using `PyPDFLoader` from LangChain [7].
- Split each document into overlapping chunks using recursive character-based splitters.
- Generate embeddings for each chunk using `all-MiniLM-L6-v2`, a transformer model from HuggingFace [8].
- Store the embeddings in Pinecone, a fast, cloud-native vector database [9].
- Index names and metadata are saved for retrieval mapping.

4.4 Retrieval and Response Generation

When a user submits a query:

- The retriever fetches the top-k most similar chunks from the Pinecone vector store [9].
- These chunks are passed to Gemini along with a custom prompt, which defines tone, scope, and answer structure.
- Gemini processes the input and generates a context-aware, medically grounded response.
- The response is sent back to the Flask frontend and rendered on the screen.
- This architecture ensures that the LLM only responds based on relevant, verified content, mitigating the risk of hallucination [4].

4.5 Flask Application Flow

The full interaction pipeline within the Flask app is:

- User submits a question via the web interface.
- The backend handles the request and routes it to the LangChain QA pipeline [7].
- The LangChain chain:
  - Retrieves relevant chunks,
  - Generates the prompt,
  - Queries Gemini for the final answer.
- The answer is returned and rendered on the `index.html` page using Jinja2.

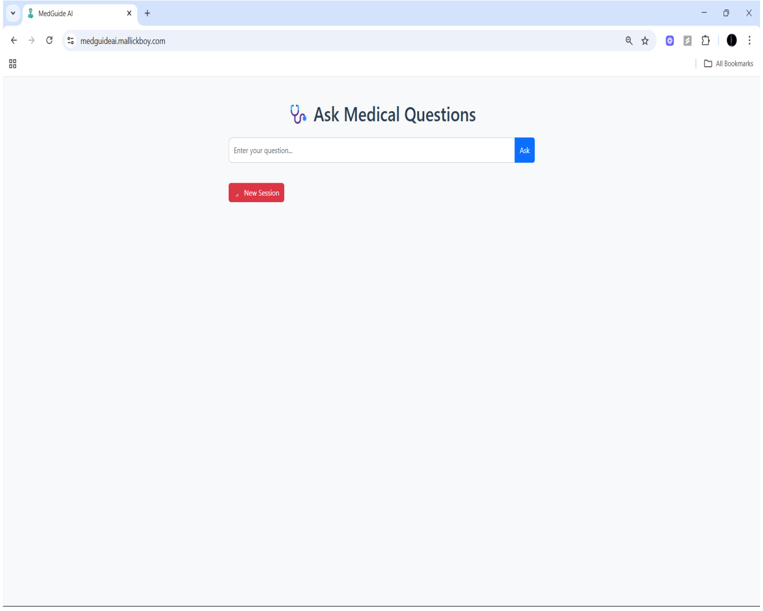
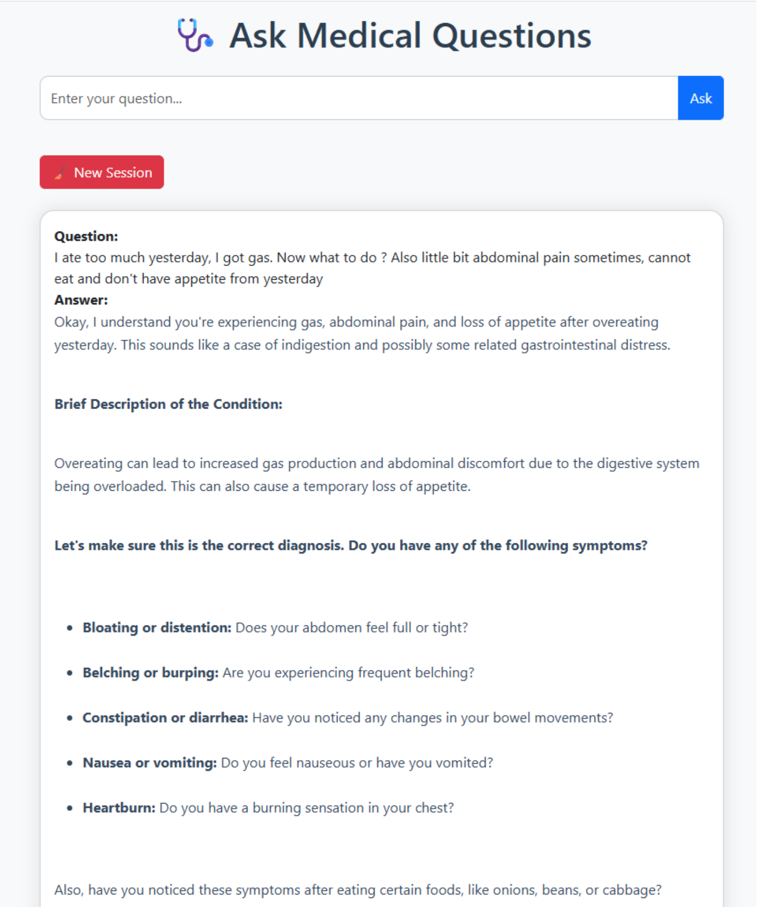


Figure 10: User Interface



Question:
I ate too much yesterday, I got gas. Now what to do ? Also little bit abdominal pain sometimes, cannot eat and don't have appetite from yesterday
Answer:
Okay, I understand you're experiencing gas, abdominal pain, and loss of appetite after overeating yesterday. This sounds like a case of indigestion and possibly some related gastrointestinal distress.

Brief Description of the Condition:

Overeating can lead to increased gas production and abdominal discomfort due to the digestive system being overloaded. This can also cause a temporary loss of appetite.

Let's make sure this is the correct diagnosis. Do you have any of the following symptoms?

- **Bloating or distention:** Does your abdomen feel full or tight?
- **Belching or burping:** Are you experiencing frequent belching?
- **Constipation or diarrhea:** Have you noticed any changes in your bowel movements?
- **Nausea or vomiting:** Do you feel nauseous or have you vomited?
- **Heartburn:** Do you have a burning sensation in your chest?

Also, have you noticed these symptoms after eating certain foods, like onions, beans, or cabbage?

Figure 11: Answering User Query

Chapter 5

5 Results and Evaluation

5.1 Overview

The evaluation was conducted using real-world-inspired patient queries that varied in symptom complexity, diagnostic requirements, and follow-up recommendations. The chatbot was tested for its ability to interpret symptoms, suggest possible diagnoses, recommend relevant lab tests, and provide prescriptions and lifestyle guidance [5].

Each query was mapped to a predefined evaluation level (Level 1 to Level 3) and assessed against the expected outcome criteria.

5.2 Evaluation Methodology

The system was evaluated on the following three competency levels:

- **Level 1: Basic health education / supplement queries**
  Objective: Provide general health information based on natural language queries.

- **Level 2: Preliminary diagnosis and test recommendation**
  Objective: Suggest possible causes and recommend initial diagnostic tests based on symptoms.

- **Level 3: Confirm diagnosis using test results and suggest treatment**
  Objective: Interpret lab test data, confirm diagnosis, suggest medications, and provide follow-up instructions.

Evaluation criteria included:

- Clinical relevance and accuracy
- Clarity and empathy in communication
- Diagnostic depth
- Test interpretation (if applicable)
- Treatment planning (if applicable)

5.3 Test Cases

Level 1 – General Medical Knowledge Questions

5.4 Evaluation Results

Level 1 Questions and Result

Q.No	Question	Summary of Chatbot Response	Evaluation
1	What are the common symptoms of iron deficiency anemia?	Listed fatigue, pallor, pica, glossitis, and more. Suggested relevant diagnostic tests (CBC, ferritin). Followed up with symptom check.	Accurate, informative, safe
2	What is the normal range for adult blood pressure?	Gave standard AHA values, noted stages of hypertension. Asked for user's BP reading and symptoms.	Aligned with guidelines, context-aware
3	How much water should an adult drink daily?	Suggested 2L/day, considered activity, diet, and health. Recommended urine output as marker.	Comprehensive, well-structured
4	What are the benefits of turmeric in Ayurveda?	Provided 8 Ayurvedic benefits: anti-inflammatory, detox, skin, digestion, dosha balance.	Holistic, culturally relevant, clear
5	What is the function of the liver in the human body?	Explained bile production, metabolism, detox, protein synthesis, sugar regulation, etc.	Concise yet thorough, medically accurate

Table 3: Evaluation of Level 1 Chatbot Responses

Level 2 Questions and Result

Q.No	**Question**	**Summary of Chatbot Response**	**Evaluation**
1	I have a cough and fatigue. What should I do?	Suggested possible viral or allergy-related causes, recommended rest, hydration, symptom monitoring. Asked if fever or chest pain was present.	Safe, encouraged follow-up care
2	I feel bloated after eating. Is that normal?	Explained common causes (gas, food intolerance, overeating), gave dietary suggestions (avoid carbonated drinks, eat slowly), recommended keeping a food diary.	Patient-friendly, well-reasoned
3	I can't sleep at night. Any natural remedies?	Provided sleep hygiene tips (reduce screen time, fixed sleep schedule, dark room), suggested herbal teas like chamomile, advised seeing a doctor if persistent.	Practical, evidence-based
4	I'm always tired even after sleeping. Why?	Listed possible causes (anemia, stress, sleep apnea, poor sleep quality), suggested lifestyle review, advised seeing a doctor if ongoing.	Good symptom mapping, cautious
5	What are some safe exercises for joint pain?	Suggested low-impact exercises (walking, swimming, yoga), stressed warming up, recommended consulting physiotherapist for persistent pain.	Tailored, safe, actionable

Table 4: Evaluation of Level 2 Chatbot Responses

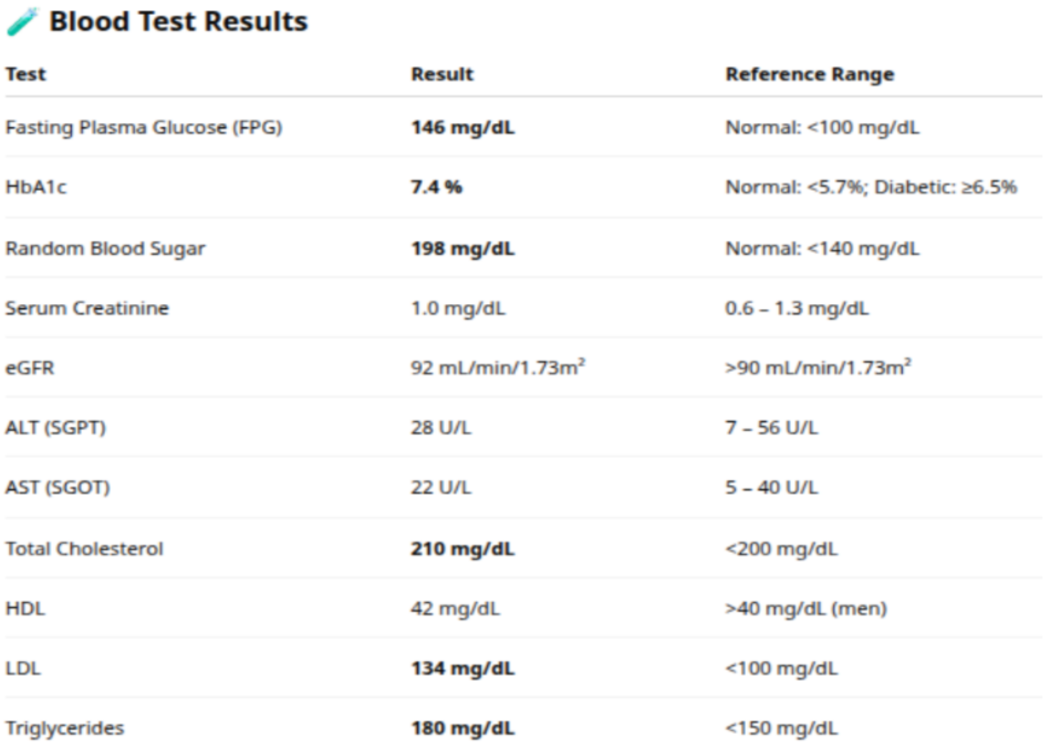
Level 3 Questions and Result

Test Case- 1 (Diabetes Mellitus Type 2)

Sample question to feed the chatbot: ”I'm a 48-year-old male. I've been very thirsty lately, urinating often, feeling tired all the time, and my vision seems blurry. I also noticed tingling in my feet. My fasting glucose is 146, HbA1c is 7.4%, and random sugar is 198. What might be going on?”

Suggested Tests by Chatbot:

- Fasting Plasma Glucose (FPG)
- HbA1c
- Random Blood Sugar
- Serum Creatinine
- eGFR
- ALT (SGPT)
- AST (SGOT)
- Total Cholesterol
- HDL
- LDL
- Triglycerides



🧪 **Blood Test Results**

Test	Result	Reference Range
Fasting Plasma Glucose (FPG)	**146 mg/dL**	Normal: <100 mg/dL
HbA1c	**7.4 %**	Normal: <5.7%; Diabetic: ≥6.5%
Random Blood Sugar	**198 mg/dL**	Normal: <140 mg/dL
Serum Creatinine	1.0 mg/dL	0.6 – 1.3 mg/dL
eGFR	92 mL/min/1.73m^2	>90 mL/min/1.73m^2
ALT (SGPT)	28 U/L	7 – 56 U/L
AST (SGOT)	22 U/L	5 – 40 U/L
Total Cholesterol	**210 mg/dL**	<200 mg/dL
HDL	42 mg/dL	>40 mg/dL (men)
LDL	**134 mg/dL**	<100 mg/dL
Triglycerides	**180 mg/dL**	<150 mg/dL

Figure 12: Blood Test Results

Evalution

Criteria	Details
Input Query	"I'm a 48-year-old male. I've been very thirsty lately, urinating often, feeling tired all the time, and my vision seems blurry. I also noticed tingling in my feet. My fasting glucose is 146, HbA1c is 7.4%, and random sugar is 198. What might be going on?"
Expected	Condition: Type 2 Diabetes Mellitus
Chatbot Diagnosis	Correctly identified as Type 2 Diabetes Mellitus based on symptoms and lab thresholds (FPG $\geq$ 126, HbA1c $\geq$ 6.5%)
Treatment Recommendation	Recommended Metformin 500mg, with proper dosage, precautions, and lifestyle advice
Additional Suggestions	Lifestyle guidance (diet, exercise, foot care), lab test suggestions (Lipid profile, Kidney function)
Limitations Observed	Minor phrasing issue on random sugar level interpretation; did not discuss contraindications of Metformin (e.g., renal function)
Result	Accurate, responsible, and context-aware response. Aligns with clinical guidelines. No hallucinated information or unsafe recommendations.

Table 5: Evaluation of Chatbot's Response for Diabetes Query

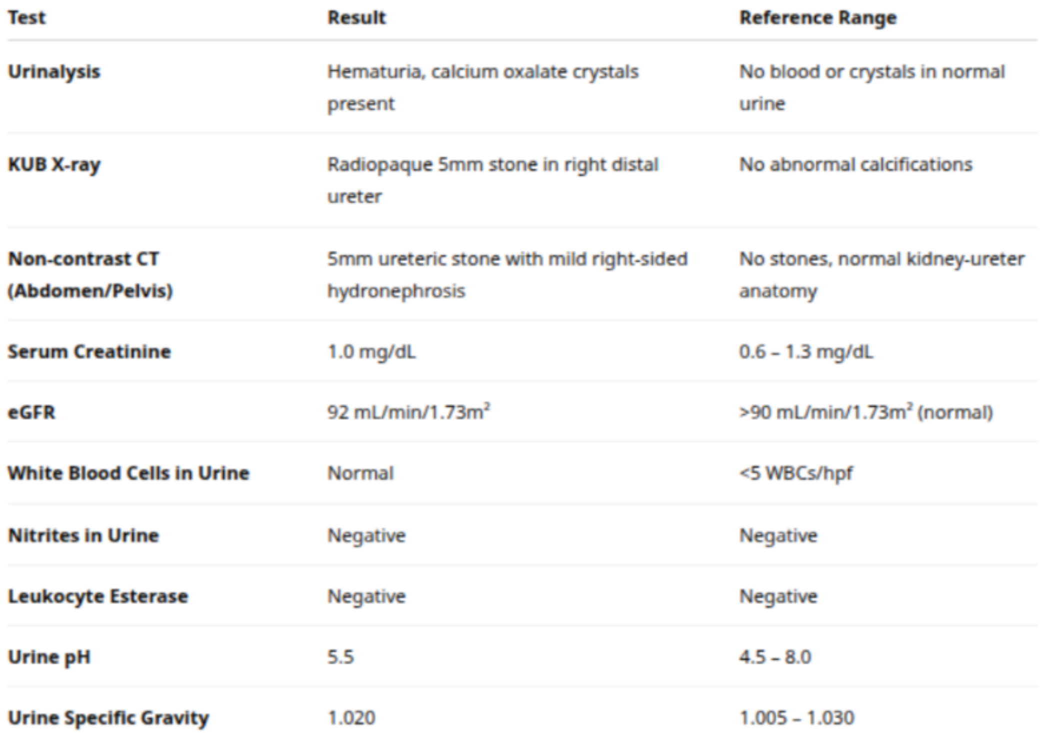
Test Case-2 (Kidney Stone)

Sample question to feed the chatbot: "Hi, I'm a 38-year-old male. I suddenly started experiencing severe pain on the right side of my lower back and abdomen. The pain comes in waves and sometimes radiates to my groin. I feel nauseated and have vomited twice. I've also noticed that I'm urinating more frequently, and today there was a small amount of blood in my urine. What should I do?"

Suggested Tests by Chatbot:

- Urinalysis – to detect hematuria, infection, and crystal type
- KUB X-ray – to locate radio-opaque stones
- Non-contrast CT scan – gold standard for detecting kidney stones and hydronephrosis
- Blood tests:
  - Serum Creatinine – to assess kidney function
  - eGFR – estimated glomerular filtration rate for overall kidney health

Test Report:



Test	Result	Reference Range
Urinalysis	Hematuria, calcium oxalate crystals present	No blood or crystals in normal urine
KUB X-ray	Radiopaque 5mm stone in right distal ureter	No abnormal calcifications
Non-contrast CT (Abdomen/Pelvis)	5mm ureteric stone with mild right-sided hydronephrosis	No stones, normal kidney-ureter anatomy
Serum Creatinine	1.0 mg/dL	0.6 – 1.3 mg/dL
eGFR	92 mL/min/1.73m²	>90 mL/min/1.73m² (normal)
White Blood Cells in Urine	Normal	<5 WBCs/hpf
Nitrites in Urine	Negative	Negative
Leukocyte Esterase	Negative	Negative
Urine pH	5.5	4.5 – 8.0
Urine Specific Gravity	1.020	1.005 – 1.030

Figure 13: Test Reports

5.5 Limitations Identified

- **Dependence on Provided Documents:** The chatbot's accuracy and reliability are dependent on the quality and scope of the documents in the knowledge base. Out-of-scope or ambiguous queries result in fallback messages or generic responses [9].

- **No Real-Time Symptom Diagnosis:** While the system can assist in understanding symptoms, it is not capable of real-time medical diagnosis or providing personalized health advice. It cannot replace professional healthcare consultation [6].

- **Performance Bottlenecks:** The performance of the system, especially in real-time queries, is influenced by network speed, with particular reference to Pinecone API latency and potential delays in retrieving relevant documents or embeddings [9].

- **Multilingual Limitation:** The system currently supports only English text processing, limiting its accessibility to a wider global audience.

- **Lack of Personalized Context:** The system does not maintain persistent user profiles or contextual history across sessions, making it unable to provide personalized responses based on prior interactions.

- **Limited Interaction Depth:** The chatbot is designed to handle specific medical queries but lacks the ability to engage in deeper, ongoing conversations about complex medical conditions or multi-step diagnostic processes.

- **Data Privacy and Security Concerns:** Although the chatbot does not store personal health information, any system handling sensitive health data must comply with data protection regulations such as HIPAA or GDPR, ensuring the confidentiality and security of user data [12].

- **Non-Specialized Medical Areas:** While the chatbot covers a wide range of general health topics, it is not specialized in niche medical areas (e.g., rare diseases) or real-time emergencies, which could result in incomplete or inaccurate information for specific queries.

- **User Input Limitations:** The system assumes user input in English and in a standard, clear format, which may limit its effectiveness in understanding unclear, misspelled, or informal queries.

- **Dependence on Third-Party Services for Response Shaping:** The chatbot's ability to generate responses is currently dependent on the ”gemini-2.0-flash” model for shaping the response using the user's question and the top-k most similar items retrieved. This reliance on third-party services introduces potential issues such as limited customization, service disruptions, or changes in the third-party API that may affect response accuracy and system performance.

- **No Integration with Healthcare Systems:** The system is standalone and does not integrate with real-time medical records, lab reports, or diagnostic equipment, which limits its ability to provide comprehensive diagnostic support.

Chapter 6

6 Application and Future Work

6.1 Application

The AIDRA chatbot is designed to address the growing need for reliable and accessible healthcare information, especially in the domains of holistic and alternative medicine. The system serves multiple real-world purposes:

- **Holistic Health Assistance:** The chatbot acts as a virtual health assistant, offering guidance on natural treatments, herbal remedies, and alternative therapies. It provides users with well-researched, credible responses sourced from reputable medical references like the Gale Encyclopedia of Alternative Medicine.

- **Accessibility for Remote and Underserved Communities:** With a focus on alternative medicine, the chatbot can be accessed globally, particularly by individuals who may not have access to specialized healthcare professionals. By offering immediate, context-aware responses, it assists users with wellness-related queries at their convenience.

- **Educational Tool:** The system serves as an educational platform for students, researchers, and enthusiasts in the fields of alternative medicine and holistic health practices. It provides reliable, evidence-based information to users exploring these topics.

- **System Deployment:** The chatbot has been deployed on a secure and scalable platform using an Azure VM (Ubuntu server) [13]. The system utilizes SSL for secure communication and has a custom domain, ensuring safe user interaction [15].

- **Integration with Health Management Platforms:** With minimal changes, the system can be integrated into larger health management platforms to offer complementary information on holistic approaches, diagnostic suggestions, and lifestyle modifications. This makes it a versatile tool for organizations in the health and wellness sector.

- **User Interaction:** The web-based interface built with Flask allows users to interact with the chatbot effortlessly. Its conversational style mimics a doctor-patient interaction, ensuring users feel comfortable querying sensitive health topics.

The chatbot system's real-time response generation, powered by the Gemini LLM, ensures that every query receives medically-grounded and contextually relevant information, making it an essential tool for anyone seeking alternative healthcare advice.

6.2 Future Work

The system demonstrates a robust proof-of-concept, but there are several areas where it can be improved or expanded to enhance functionality, accessibility, and user experience.

Multilingual Support

To make the system accessible to a wider global audience, future iterations can include support for multiple languages. This can be implemented using translation models such as:

- Google Translate API for real-time translation [16],
- MarianMT (by HuggingFace) for offline, multilingual machine translation [17].

Audio Input and Output

Accessibility can be significantly improved by integrating:

- Speech-to-Text (STT) using libraries like SpeechRecognition or Google's Web Speech API, allowing users to speak their symptoms or questions [18].
- Text-to-Speech (TTS) using tools like pyttsx3 or Google Cloud TTS, enabling the chatbot to speak out responses [19].

Enhanced UI/UX

The current UI is functional but basic. A more interactive and user-friendly interface can be created using modern frontend frameworks such as:

- ReactJS for dynamic rendering [20].

Persistent Chat History

Currently, chat sessions are stored temporarily. A long-term goal is to preserve user conversations across sessions using databases like:

- MongoDB (NoSQL, JSON-based documents) [21],
- PostgreSQL (relational database).

Broader Medical Scope

While the current system incorporates traditional and alternative medical literature, it can be extended to include modern (allopathic) sources such as:

- WHO Guidelines [22],
- CDC Publications [23].

Feedback Loop and Personalization

Introducing a user feedback mechanism will allow:

- Rating chatbot responses,
- Collecting user preferences and use cases,
- Retraining or fine-tuning the model based on interaction patterns.

Fine-Tuning and Model Optimization

To further enhance the system's response generation and adapt it to specific healthcare domains, fine-tuning the underlying language models with additional medical datasets could improve the accuracy and contextuality of the responses. Leveraging frameworks such as TensorFlow and Keras can facilitate model optimization for domain-specific tasks, improving both speed and performance [10].

Chapter 7

7 Conclusion

8 Conclusion

The **RAG Doctor** project successfully demonstrates how retrieval-augmented generation (RAG) can be integrated with a domain-specific knowledge base to build a reliable, LLM-powered chatbot. The focus on holistic and alternative medicine highlights how AI can support informational healthcare services beyond conventional methods.

Key Accomplishments

- Developed a working end-to-end LLM-based assistant with custom prompts [6].
- Embedded verified medical content using semantic vector representations [8].
- Integrated Google Gemini generative capabilities to produce fluent, informative answers [19].
- Built an intuitive and clean web interface for end-users [20].

The project shows that with proper architecture and grounded source material, generative AI can be used effectively to build helpful assistants without relying solely on model memory. It also emphasizes the importance of document grounding to reduce hallucination in critical domains like healthcare [7].

The complete source code and implementation details can be accessed at: `https://github.com/mallickboy/AIDRA`

Bibliography

[1] World Bank and World Health Organization (2017) *Half the world lacks access to essential health services, 100 million still pushed into extreme poverty because of health expenses*. Available at: `https://www.who.int/news/item/13-12-2017` (Accessed: 13 May 2025).

[2] Public Health Foundation of India (2018) *Catastrophic health expenditure in India: Who is at risk?* Available at: `https://www.phfi.org` (Accessed: 13 May 2025).

[3] World Health Organization (2019) *Patient safety fact file*. Available at: `https://www.who.int` (Accessed: 13 May 2025).

[4] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y. and Ishii, E. (2023) *Survey of hallucination in natural language generation. arXiv preprint*, arXiv:2302.03629. Available at: `https://arxiv.org/abs/2302.03629` (Accessed: 13 May 2025).

[5] Ada Health, Babylon Health and HealthTap (no date) *Digital healthcare platforms*. Available at: `https://ada.com`, `https://www.babylonhealth.com`, `https://www.healthtap.com` (Accessed: 13 May 2025).

[6] OpenAI (no date) *ChatGPT: Optimizing language models for dialogue*. Available at: `https://openai.com/chatgpt` (Accessed: 13 May 2025).

[7] LangChain (no date) *LangChain documentation*. Available at: `https://docs.langchain.com` (Accessed: 13 May 2025).

[8] Reimers, N. and Gurevych, I. (no date) *HuggingFace sentence transformers*. Available at: `https://www.sbert.net` (Accessed: 13 May 2025).

[9] Pinecone (no date) *Pinecone documentation*. Available at: `https://docs.pinecone.io` (Accessed: 13 May 2025).

[10] Géron, A. (2023) *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. 3rd edn. Sebastopol, CA: O'Reilly Media.

[11] Python-dotenv (no date) *Reads key-value pairs from a .env file and sets them as environment variables*. Available at: `https://pypi.org/project/python-dotenv/` (Accessed: 13 May 2025).

[12] OpenAI (2023) *Prompt Injection Attacks and Defenses.* Available at: `https://platform.openai.com/docs/guides/prompt-engineering/prompt-injection` (Accessed: 13 May 2025).

[13] Microsoft Azure (no date) *Deploy Python apps to Azure Virtual Machines.* Available at: `https://learn.microsoft.com/en-us/azure/developer/python/` (Accessed: 13 May 2025).

[14] Gunicorn (no date) *Green Unicorn: Production WSGI server for Python.* Available at: `https://gunicorn.org/` (Accessed: 13 May 2025).

[15] Mozilla (no date) *Server Side TLS: SSL Configuration Generator.* Available at: `https://ssl-config.mozilla.org/` (Accessed: 13 May 2025).

[16] Google (no date) *Google Translate API.* Available at: `https://translate.google.com` (Accessed: 13 May 2025).

[17] HuggingFace (no date) *MarianMT.* Available at: `https://huggingface.co/Helsinki-NLP/opus-mt` (Accessed: 13 May 2025).

[18] SpeechRecognition (no date) *SpeechRecognition Python Package.* Available at: `https://pypi.org/project/SpeechRecognition/` (Accessed: 13 May 2025).

[19] Google (no date) *Google Cloud Text-to-Speech.* Available at: `https://cloud.google.com/text-to-speech` (Accessed: 13 May 2025).

[20] React (no date) *ReactJS Framework.* Available at: `https://reactjs.org` (Accessed: 13 May 2025).

[21] MongoDB (no date) *MongoDB.* Available at: `https://www.mongodb.com` (Accessed: 13 May 2025).

[22] World Health Organization (no date) *WHO Guidelines.* Available at: `https://www.who.int` (Accessed: 13 May 2025).

[23] Centers for Disease Control and Prevention (no date) *CDC Publications.* Available at: `https://www.cdc.gov` (Accessed: 13 May 2025).