

---

HACKATHON: PHARMA KNOWLEDGE ASSISTANT — ESA

---

Submitted by:

**Team 40**

C Harsha	PES2UG21CS130
Harshan	PES2UG21CS190
Nishank Koul	PES2UG21CS342
Pragya Das	PES2UG21CS381
Urvashi Bhargava	PES2UG21CS579



## **Table of Contents:**

### **1. Introduction**

#### **1.1 Objective of the Hackathon**

#### **1.2 Problem Statement**

### **2. Project Scope**

#### **2.1 Features to be Implemented**

### **3. Dataset Description**

#### **3.1 Initial Sample Dataset Details**

#### **3.2 Dataset Expansion and Preparation**

### **4. High-Level Requirements**

#### **4.1 Core Features Overview**

### **5. Architecture Design**

#### **5.1 Agent-Based Framework Overview**

### **6. Implementation Steps**

#### **6.1 Dataset Preparation**

#### **6.2 Question Answering using RAG**

#### **6.3 Recommender System Implementation**

#### **6.4 Alternatives Generator Implementation**

#### **6.5 Summarizer Implementation**

#### **6.6 Agent-Based Framework Design**

#### **6.7 Sample Questions and Outputs**

#### **6.8 GUI Design and Implementation**

### **7. Evaluation Metrics**

#### **7.1 Accuracy of Question Answering**

#### **7.2 Recommender System Performance**

### **8. Submission Evaluation Criteria**

#### **8.1 Individual Feature Contributions**

#### **8.2 Team Performance**

# 1. Introduction

## 1.1 Objective of the Hackathon

The Pharma Knowledge Assistant aims to revolutionize the way pharmaceutical queries are addressed by leveraging cutting-edge **Large Language Model (LLM) application patterns**. These include:

1. **Retrieval-Augmented Generation (RAG)**: Enhances response accuracy by retrieving relevant context from structured datasets, ensuring precise and fact-based answers to user queries.
2. **Prompt Engineering**: Tailors the interactions between the user and the LLM, enabling context-aware, natural language responses for complex pharmaceutical queries.
3. **Agent-Based Designs**: Decomposes the overall application into interconnected, task-specific nodes, ensuring modularity, reusability, and efficient handling of multi-step tasks.

The hackathon's goal is to create a **user-friendly, intelligent assistant** capable of answering medication-related queries, generating recommendations, summarizing key information, and suggesting alternatives. The assistant empowers users, including healthcare professionals and the public, to make informed decisions by providing accurate, context-aware information in real-time.

## 1.2 Problem Statement

The Pharma Knowledge Assistant hackathon challenges participants to develop an **Agent-Based Pharma Knowledge Assistant** that leverages Large Language Models (LLMs) to process and respond to user queries about pharmaceutical products. The system should integrate advanced AI methodologies such as Retrieval-Augmented Generation (RAG), prompt engineering, and agent-based designs to enable functionalities like question answering, medication recommendations, alternative suggestions, and concise summarization of drug details. Participants must design an agentic framework where features are implemented as interconnected nodes, enabling efficient multi-step query resolution and context-sharing. The assistant aims to provide accurate, user-friendly, and

context-aware responses, empowering users to make informed decisions about medications while demonstrating the practical application of AI in healthcare.

## 2. Project Scope

The Pharmaceutical Knowledge Assistant aims to provide an interactive, intelligent solution for managing pharmaceutical queries through an **agent-based framework** that leverages cutting-edge LLM technologies. The scope of the project encompasses the following components:

### 2.1 Features to be implemented

#### 1. Dataset Preparation and Expansion

**Objective:** Enhance the existing pharmaceutical dataset by expanding it beyond the initial 10 JSON files using web scraping.

**Scope:**

- a. Include key details for medications, such as:
  - i. Name
  - ii. Uses
  - iii. Composition
  - iv. Side Effects
  - v. Contraindications
  - vi. Dosage
- b. Use a Python-based scraper to collect information from trusted sources (e.g., Microlabs USA).
- c. Ensure uniformity and consistency in the schema of the expanded dataset.

#### 2. Question Answering with Retrieval-Augmented Generation (RAG)

**Objective:** Implement a RAG model using LangChain for accurately handling user queries.

**Scope:**

- a. Ingest and index the expanded dataset for efficient retrieval.
- b. Use vector representations (e.g., with Sentence Transformers) for similarity-based searches.
- c. Integrate with a vector database (e.g., FAISS) for fast and scalable context retrieval.

### 3. **Medication Recommender System**

**Objective:** Suggest medications based on user-reported symptoms and conditions.

**Scope:**

- a. Filter results based on symptom-condition mapping.
- b. Use **Nearest Neighbor Search** to identify relevant matches.
- c. Generate personalized medication recommendations, including their uses, dosages, and side effects.

### 4. **Alternatives Generator**

**Objective:** Provide users with alternative medications based on therapeutic equivalence.

**Scope:**

- a. Cluster medications using HDBSCAN or similar algorithms to group items by therapeutic use.
- b. Suggest alternative options with similar benefits but distinct compositions.

### 5. **Summarization of Medication Information**

**Objective:** Provide concise, user-friendly summaries of detailed medication data.

**Scope:**

- a. Generate section-wise summaries (e.g., Uses, Dosage, Side Effects).
- b. Combine section summaries into a complete overview for quick user reference.

### 6. **Agent-Based Framework Integration**

**Objective:** Design a modular framework that integrates RAG, recommenders, alternatives generators, and summarizers.

**Scope:**

- a. Ensure seamless communication between nodes using Langgraph.
- b. Enable shared context for handling multi-step queries efficiently.

### 7. **Streamlit-Based GUI**

**Objective:** Create a user-friendly interface for interacting with the Pharma Knowledge Assistant.

**Scope:**

- a. Design a simple chatbot interface using Streamlit.
- b. Allow users to input queries and view results dynamically.
- c. Include interactive elements like buttons for recommendations and alternatives.

### 8. **Performance Metrics Evaluation**

**Objective:** Assess the system's efficiency and accuracy to refine its functionality.











**Scope:**

- a. Evaluate accuracy, response relevance, and user satisfaction.
- b. Track response time and throughput to ensure optimal performance.

This project aims to integrate these features cohesively to deliver an intelligent and intuitive solution for pharmaceutical information management.

### 3. Dataset Description

#### 3.1 Initial Sample Dataset Details

microlabs_usa 10 items	
Name	
	Acetazolamide Extended-Release Capsules.json
	Amlodipine Besylate and Olmesartan Medoxomil Tablets.json
	Amoxicillin and Clavulanate Potassium for Oral Suspension, USP....
	Amoxicillin and Clavulanate Potassium Tablets, USP.json
	Amoxicillin Capsules, USP.json
	Aspirin and Extended-Release Dipyridamole Capsules.json
	Atorvastatin Calcium Tablets.json
	Bimatoprost Ophthalmic Solution.json
	Celecoxib capsules.json
	Chlordiazepoxide Hydrochloride and Clidinium Bromide Capsule...



```

def extract_product_listing():
    base_url = "https://www.microlabsusa.com/products/"
    response = requests.get(base_url)
    soup = BeautifulSoup(response.text, "html.parser")

    # Step 1: Navigate to the <body>
    body_tag = soup.find("body")
    if not body_tag:
        raise ValueError("No <body> tag found in the HTML.")

    # Step 2: Find the div with id 'page-container' inside the body
    page_container_div = body_tag.find("div", id="page-container")
    if not page_container_div:
        raise ValueError("No <div> with id 'page-container' found.")

    # Step 3: Find the div with id 'et-boc' inside 'page-container'
    et_boc_div = page_container_div.find("div", id="et-boc")
    if not et_boc_div:
        raise ValueError("No <div> with id 'et-boc' found.")

    # Step 4: Find the div with id 'et-main-area' inside 'et-boc'
    et_main_area_div = et_boc_div.find("div", id="et-main-area")
    if not et_main_area_div:
        raise ValueError("No <div> with id 'et-main-area' found.")

    # Step 5: Find the div with class 'container' inside 'et-main-area'
    container_div = et_main_area_div.find("div", class_="container")
    if not container_div:
        raise ValueError("No <div> with class 'container' found.")

    # Step 6: Find the div with class 'inner_container' inside 'container'
    inner_container_div = container_div.find("div", class_="inner_container")
    if not inner_container_div:
        raise ValueError("No <div> with class 'inner_container' found.")

```

Ln 85, Col 26   Spaces: 4   UTF-8   CRLF

## 4. High-Level Requirements

### 4.1 Core Features Overview

The Pharma Knowledge Assistant is designed to address pharmaceutical queries using an **agent-based framework** that integrates multiple functionalities to enhance accuracy, usability, and interactivity. The core features include:

#### 1. Question Answering



- a. Enable natural language queries, such as "What are the side effects of Paracetamol?"
- b. Use **Retrieval-Augmented Generation (RAG)** to fetch relevant information from the dataset and generate accurate, context-aware responses.

## **2. Medication Recommender System**

- a. Suggest appropriate medications based on user-reported symptoms or conditions.
- b. Provide personalized recommendations, ensuring relevance and adherence to contraindications, while explaining the reasoning behind the suggestions.

## **3. Alternatives Generator**

- a. Offer substitute medications with similar therapeutic benefits but different compositions.
- b. Use clustering techniques to group medications by therapeutic use and recommend relevant alternatives to the user.

## **4. Summarization**

- a. Generate concise and user-friendly summaries of complex medication information, including uses, dosage instructions, and side effects.
- b. Present summaries in a format suitable for quick reference and easy comprehension.

## **5. Search Agent**

- a. Handle queries outside the scope of the dataset by integrating a search agent that fetches information from external sources.
- b. Ensure context-sharing with other nodes to maintain coherent and seamless query handling.

## **6. Streamlit-Based Chatbot**

- a. Provide a graphical user interface for users to interact with the assistant.
- b. Support features such as chat history, interactive queries, and visual presentation of results.

# **5. High-Level Requirements**

## **5.1 Agent-Based Framework Overview**

The Pharma Knowledge Assistant employs an **agent-based framework** to modularize and streamline the implementation of its functionalities. The framework consists of interconnected nodes, each dedicated to a specific task, such as question answering, recommendations, or summarization.

Key aspects of the framework include:

- **Multi-Node Integration:** Each feature, such as the RAG-based question answering or the alternatives generator, is implemented as a separate node.
- **Context Sharing:** Nodes share context to ensure consistent and coherent responses across multi-step queries. For example, a user's query about a medication's side effects may trigger the question-answering node followed by the recommender node if alternatives are needed.
- **Router Nodes:** Central routing nodes evaluate user queries and dynamically determine the appropriate feature nodes to invoke based on query intent.

## 5.2 Visualization of Architecture

The architecture of the Pharma Knowledge Assistant can be visualized as a directed graph:

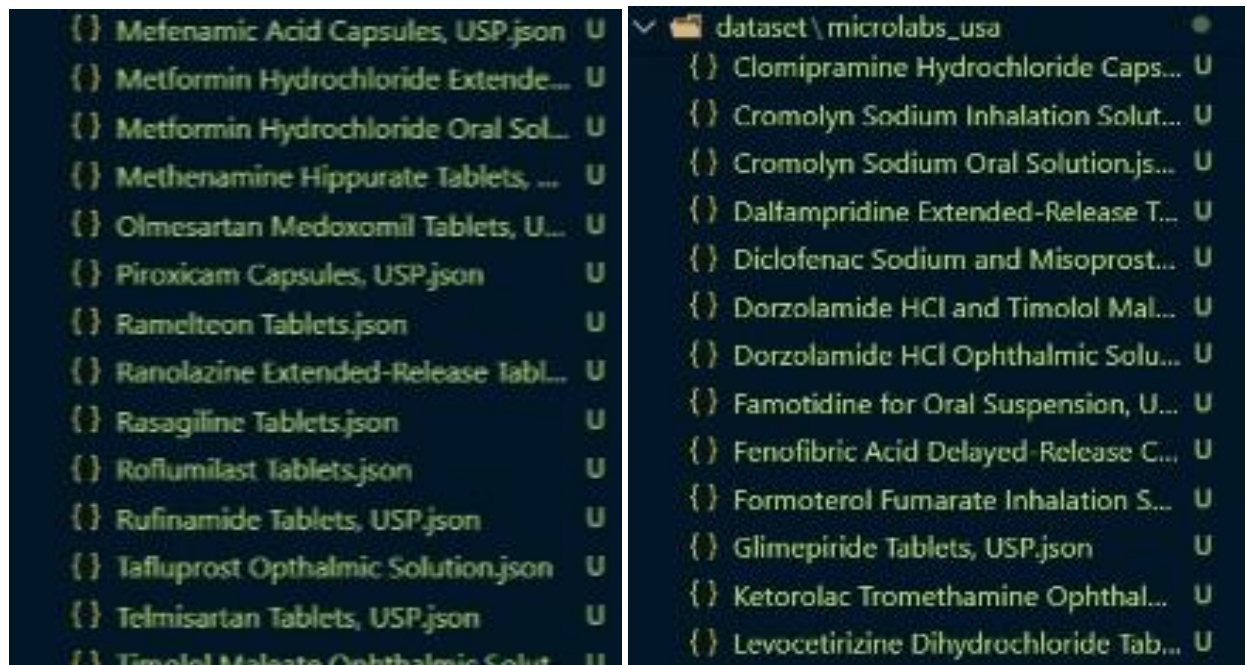
1. **Input Node:** Captures user queries and initiates processing.
2. **Router Node:** Analyzes the query to identify intent and invokes relevant feature nodes.
3. **Feature Nodes:**
  - a. **RAG Node:** Retrieves information and generates responses for queries within the dataset.
  - b. **Recommender Node:** Suggests medications based on symptoms or conditions.
  - c. **Alternatives Generator Node:** Provides substitute medication options.
  - d. **Summarizer Node:** Condenses detailed information into concise summaries.
  - e. **Search Agent Node:** Fetches answers from external sources if data is unavailable locally.
4. **Output Node:** Formats and presents the final response to the user through the Streamlit GUI.

## 6. Implementation Steps

### 6.1 Dataset Preparation

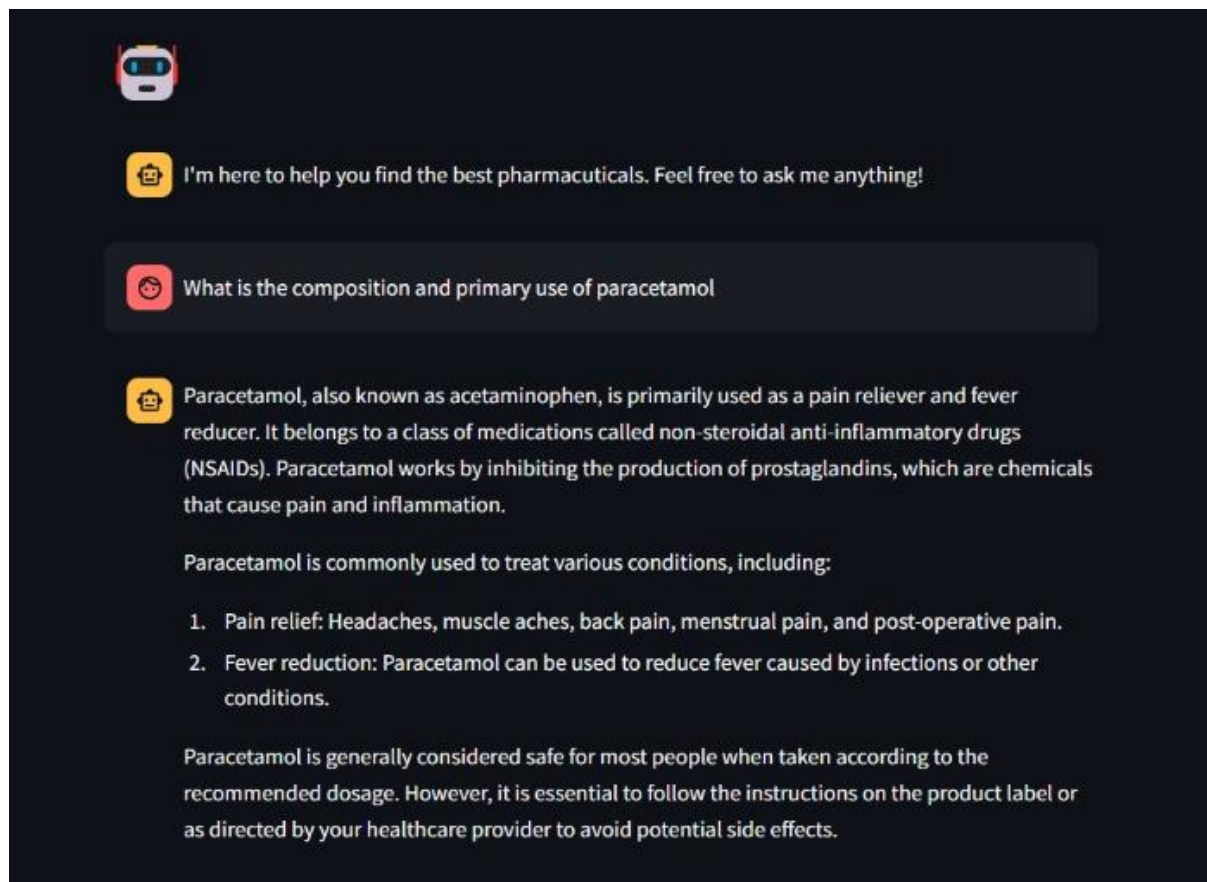
The dataset preparation begins with **expanding the initial dataset** of pharmaceutical products by scraping additional data from reliable sources using the provided `web_scraper.py` script. The scraping process targets key

product information, such as drug names, uses, composition, side effects, dosage, contraindications, and more. The data is then formatted into a consistent **JSON schema**, ensuring uniformity across all entries. Additionally, the dataset is cleaned to handle missing or inconsistent data, making it ready for ingestion into the system. After validation and testing, the expanded dataset is stored and integrated into the application for further processing.



## 6.2 Question Answering using RAG

The **Retrieval-Augmented Generation (RAG)** model is implemented using **LangChain** to handle user queries related to pharmaceutical products. The process begins by **ingesting the dataset** into the system using a **JSON loader**. To enable efficient retrieval, the data is converted into **vector embeddings** using tools like **Sentence Transformers** and stored in a **vector database** (e.g., FAISS). When a user query is received, the RAG model retrieves relevant documents from the vector store and combines this information with the query to generate an accurate and context-aware response. The **LLM server** (such as Ollama or LMStudio) is used to process the query, and the result is evaluated for accuracy, ensuring the quality of the answer.



### 6.3 Recommender System Implementation

The **Medication Recommender System** uses an **agent-based approach** combined with RAG to recommend suitable medications based on user inputs. The system processes user-reported **symptoms** or **conditions** and **filters medications** that match. The **Nearest Neighbor Search** algorithm is employed to identify the most relevant medications based on the user's context. The LLM generates personalized recommendations, explaining the reasoning behind the selection. The system also ensures that contraindications are taken into account, providing **warnings** if needed.

# Pharma Recommendation Chatbot



I'm here to help you find the best pharmaceuticals. Feel free to ask me anything!



Can i take Ibuprofen if i have a history of stomach ulcers



Ibuprofen belongs to a class of medications called nonsteroidal anti-inflammatory drugs (NSAIDs). NSAIDs can irritate the stomach lining, which may worsen stomach ulcers. If you have a history of stomach ulcers, it's best to avoid taking Ibuprofen.

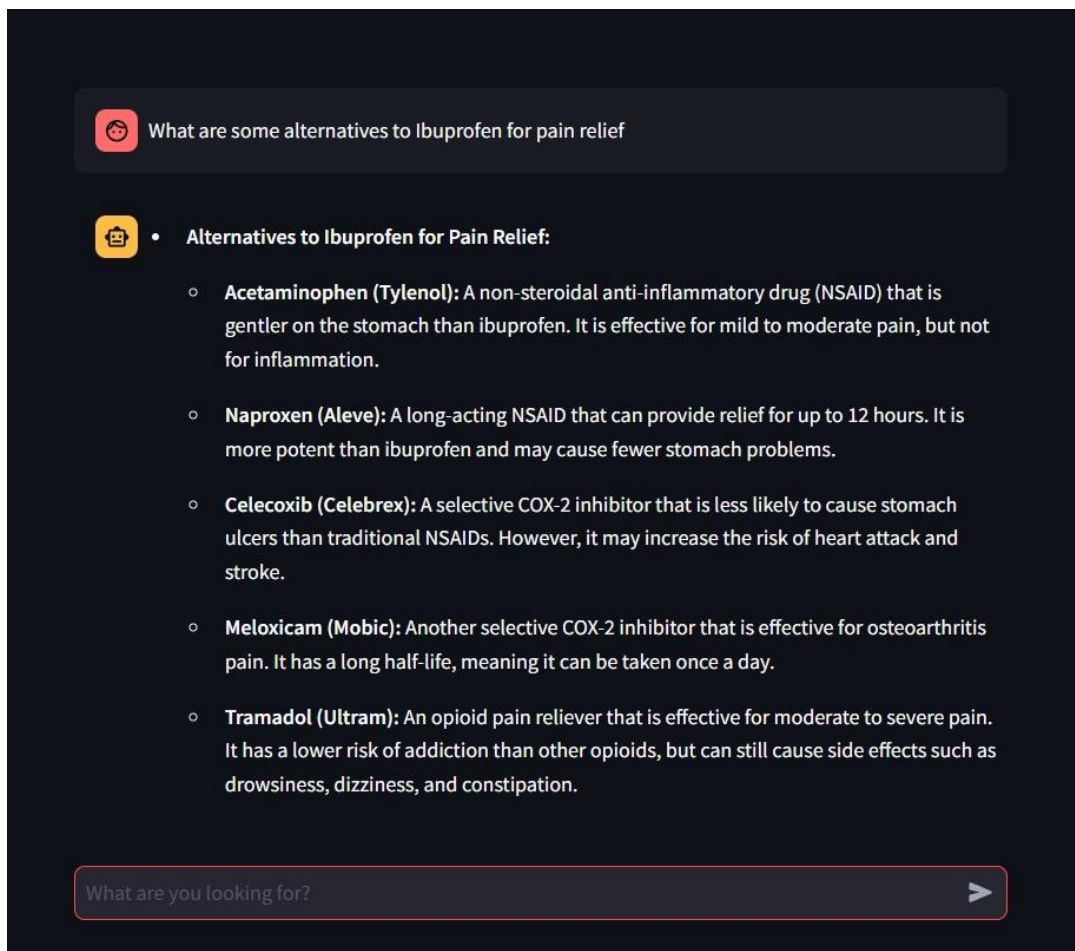
Alternative pain relievers that are less likely to cause stomach problems include acetaminophen, naproxen, or celecoxib. Always consult your doctor or pharmacist for the best advice on which medication is right for you.

What are you looking for?



## 6.4 Alternatives Generator Implementation

The **Alternatives Generator** is designed to suggest alternative medications within the same therapeutic category but with different compositions. The implementation begins by **clustering medications** based on their **therapeutic uses** using **embedding-based clustering** techniques such as **HDBSCAN**. Each medication's details are converted into vectors using **Sentence Transformers**, enabling similarity-based clustering. When a user queries for alternatives, the system retrieves relevant medications from the same cluster, offering choices that provide similar therapeutic benefits but vary in composition, dosage, or formulation. Alternatives are presented along with key information such as side effects and contraindications, helping the user make informed decisions.



## 6.5 Summarizer Implementation

The **Summarizer** condenses detailed pharmaceutical product information into shorter, more digestible summaries. The implementation begins by modifying the **RAG pipeline** to handle summarization tasks. Instead of simply retrieving answers, the system is tasked with summarizing specific sections of the product details, such as **uses, dosage, side effects, and contraindications**. These sections are summarized independently using prompts tailored for each aspect. After generating individual section summaries, the system combines them into a complete and concise overview of the medication. This summary is optimized for user comprehension and quick reference, ensuring clarity and brevity.



# Pharma Recommendation Chatbot



I'm here to help you find the best pharmaceuticals. Feel free to ask me anything!



Summarize the details of Amoxicillin in a paragraph



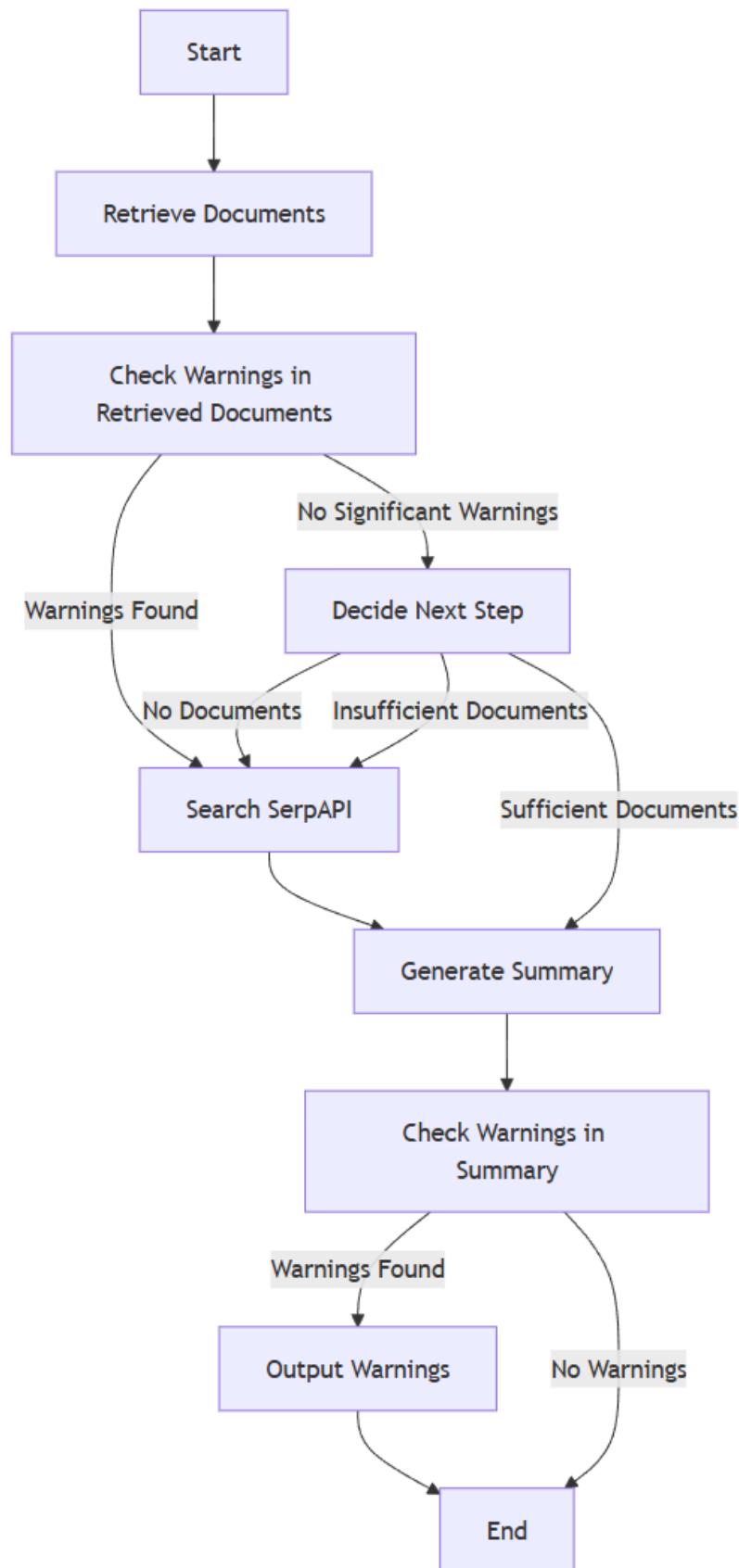
Amoxicillin is a penicillin-type antibiotic used to treat bacterial infections. It works by interfering with the bacteria's ability to build their cell walls, leading to their death. Amoxicillin is effective against a wide range of bacteria, including those that cause pneumonia, bronchitis, skin infections, and urinary tract infections. It is typically taken orally, and the dosage and duration of treatment will vary depending on the type and severity of the infection. Amoxicillin is generally well-tolerated, but side effects can include nausea, vomiting, diarrhea, and skin rash. It is essential to take Amoxicillin exactly as directed by your doctor and to complete the full course of treatment, even if you start to feel better, to ensure that the infection is fully cleared.

What are you looking for?



## 6.6 Agent-Based Framework Design

The **Agent-Based Framework** is the backbone of the Pharma Knowledge Assistant, allowing all individual features to function cohesively. The framework is designed using **Langgraph**, where each feature (e.g., question answering, recommender system, alternatives generator, summarizer) is represented as a **node** in the graph. **Router nodes** evaluate incoming queries and determine the appropriate feature nodes to activate based on the context. **Context sharing** between nodes ensures seamless transitions and coherent responses, even for multi-step queries. The agent-based framework also allows for the easy addition of new features or tools, making the system modular and scalable. The framework is designed to ensure that all components interact efficiently while maintaining flexibility and responsiveness.





# Pharma Recommendation Chatbot



I'm here to help you find the best pharmaceuticals. Feel free to ask me anything!



I have a headache and a history of stomach ulcers. Can u recommend a safe medication?



I understand you're experiencing a headache and have a history of stomach ulcers. It's important to be cautious when choosing medications.

NSAIDs, like ibuprofen, can irritate the stomach lining and increase the risk of ulcers. Therefore, it's not advisable to take ibuprofen in your case.

Instead, I recommend acetaminophen. It's a non-steroidal analgesic that is less likely to cause stomach problems.

Additionally, you can try non-medicinal pain relief methods like applying a cold compress to your head or taking a warm bath.

If your headache persists or worsens, please consult a healthcare professional.

What are you looking for?



what dosage should i take for acetaminophen



The recommended dosage of acetaminophen varies depending on the individual's age, weight, and medical condition. For adults and children 12 years and older, the usual dose is 325 to 1,000 milligrams (mg) every 4 to 6 hours as needed for pain or fever. Do not take more than 3,000 mg of acetaminophen per day.

If you have any questions about the dosage of acetaminophen you should take, please consult with your doctor or pharmacist.

## 6.7 Sample Questions and Outputs

The system's performance is evaluated by testing a range of sample questions. Examples include:

- **"What is the composition and primary use of Paracetamol?"**

*Expected Output:* The system retrieves relevant information from the dataset, providing a comprehensive answer on the composition (active ingredient) and primary use (e.g., pain relief and fever reduction) of Paracetamol.

*Our Output:* Paracetamol, also known as acetaminophen, is primarily used as a pain reliever and fever reducer. It belongs to a class of medications called analgesics and antipyretics. Paracetamol is commonly used to treat mild to moderate pain, such as headaches, backaches, and muscle aches. It is also effective in reducing fever.

Paracetamol's mechanism of action involves inhibiting the production of prostaglandins, which are chemicals that play a role in pain and inflammation. By reducing prostaglandin levels, paracetamol can effectively alleviate pain and fever.

- **"Can I take Ibuprofen if I have a history of stomach ulcers?"**

*Expected Output:* The system suggests an alternative medication like **acetaminophen**, explaining why **Ibuprofen** is not recommended due to its potential effects on the stomach lining.

*Our Output:* Ibuprofen, a nonsteroidal anti-inflammatory drug (NSAID), can worsen stomach ulcers due to its ulcerogenic effects. It can irritate the stomach lining, increasing the risk of bleeding and further damage. Considering your history of stomach ulcers, it's crucial to avoid Ibuprofen. Alternative pain relievers that are less likely to irritate the stomach include acetaminophen (Tylenol) or COX-2 inhibitors like celecoxib (Celebrex) or rofecoxib (Vioxx). If you must take Ibuprofen, do so with caution and under medical supervision. Your doctor may recommend taking it with food or a stomach protectant to minimize the risk of stomach upset. Additionally, avoid alcohol consumption while taking Ibuprofen, as it can further irritate the stomach.

- **"What are the side effects of Amoxicillin?"**

*Expected Output:* The system provides a concise list of common side effects (e.g., nausea, rash) based on the data set.

*Our Output:* Amoxicillin is a penicillin-type antibiotic used to treat bacterial infections. It works by killing or stopping the growth of bacteria. Amoxicillin is effective against a wide range of bacteria, including those that cause pneumonia, bronchitis, skin infections, and urinary tract infections. It is taken orally, usually every 8 or 12 hours. The usual dose for adults is 250 to 500 mg every 8 hours or 500 to 875 mg every 12 hours. The usual dose for

children is 20 to 40 mg/kg/day divided into three doses. Amoxicillin is generally well-tolerated, but side effects can include diarrhea, nausea, vomiting, and rash. It is important to take amoxicillin exactly as prescribed and to finish the entire course of treatment, even if you feel better. Stopping treatment early can increase the risk of developing antibiotic resistance.

## 6.8 GUI Design and Implementation

The **GUI** is developed using **Streamlit**, providing an intuitive and interactive interface for users to interact with the Pharma Knowledge Assistant. The front end allows users to **input queries** via text fields, and the system displays the generated responses in a clean, user-friendly format. The **chat history** feature enables users to track their previous queries and responses, ensuring continuity in interactions. The interface is designed to be responsive, ensuring compatibility across devices. The screenshots above showcase the UI of our project.

## 7. Evaluation Metrics

### 7.1 Accuracy of Question Answering

The accuracy of question answering is crucial for ensuring that the system provides correct, contextually relevant responses to user queries. We utilize **Trulens** to track and evaluate the accuracy of the system's responses. This helps in monitoring how well the system answers questions about pharmaceutical products and whether the retrieved data matches the expected results.

**Metric:** Accuracy of Response

**Trulens Use:** Trulens helps track how often the agent provides the correct answers by comparing the actual response with the expected output. This can be done using the **TrulensTracker** API. The system logs the accuracy over time, allowing continuous evaluation and improvements.

**Code Example:**

```
from trulens.eval import TrulensTracker

tracker = TrulensTracker() tracker.track("response_accuracy",
response, expected_answer)
```

**Goal:** Achieve high accuracy by ensuring the agent consistently provides the correct responses to user queries, particularly in handling drug-related question

## 7.2 Recommender System Performance

The performance of the **Recommender System** is evaluated by measuring its **precision**, **recall**, and **F1 score**. These metrics help assess how effectively the system recommends medications based on user symptoms or conditions. The **Trulens** tool tracks precision and recall, which are critical for evaluating recommendation systems, ensuring that the recommended medications are relevant and helpful.

**Metric:** Precision and Recall

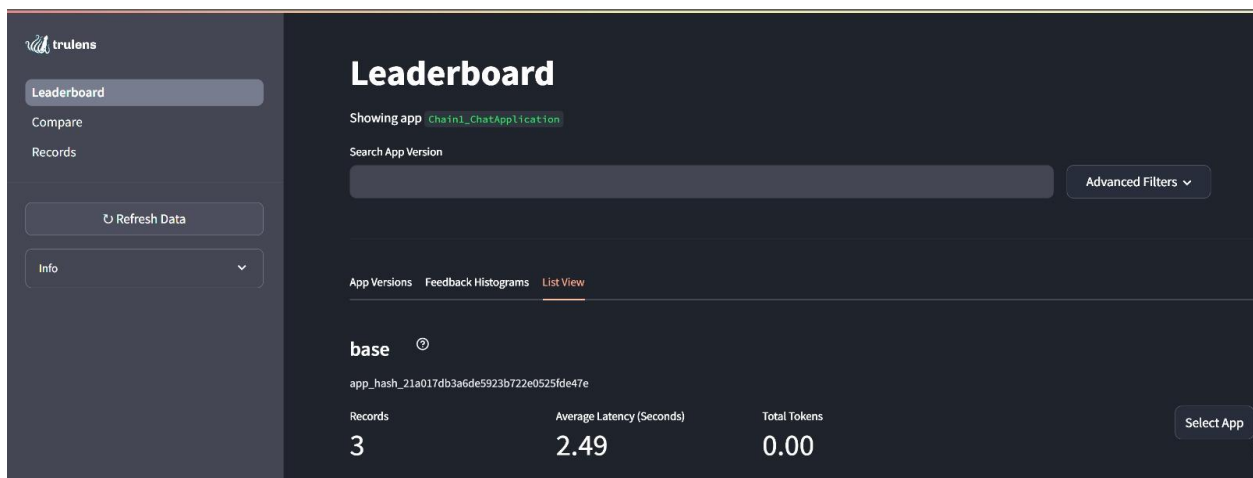
**Trulens Use:** Precision measures the proportion of recommended medications that are relevant, while recall tracks how many of the relevant medications are being suggested. Trulens allows us to log these metrics for each recommendation and adjust the system to balance them for optimal performance.


**Code Example:**

```
tracker.track_precision_recall("recommendation_precision",
recommendations, relevant_results)
```

**Goal:** Maximize both **precision** and **recall**, ensuring the system recommends medications that are not only relevant but also cover a wide range of possible therapeutic options.

## Dashboard:





Leaderboard
Compare
Records

Refresh Data
Info

base
What are ingredients of the product

The inactive ingredients include hydroxypropyl cellulose, talc, microcrystalline cellulose, sodium lauryl sulfate, D&C Red No. 28, titanium dioxide, shellac, alcohol, isopropyl alcohol, butyl alcohol, propylene glycol, ammonia, ferrous ferric oxide, and...

1 to 3 of 3

app\_hash\_21a017db3a6de5923b722e0525fde47e / record\_hash\_fb064ba880893d57b3df01f68848ceeb

record\_hash\_fb064ba880893d57b3df01f68848ceeb

Record Input
Record Output

Total tokens (#)
0

Total cost (USD)
\$0.00

Latency (s)
2.52086s
↑ 0.0279s

```

import streamlit as st
from openai import OpenAI
from trulens_eval import Tru
from trulens_eval.feedback.provider import OpenAI as TruProvider
from trulens_eval import Feedback
from trulens_eval.app import App
import numpy as np

# Initialize TruLens
tru = Tru()
provider = TruProvider()

# Set up the Streamlit App
st.title("Medical Q&A with Llama-3 and TruLens Evaluation")

# Connect to Llama-3 hosted locally (e.g., LMStudio or Ollama)
client = OpenAI(base_url="http://localhost:1234/v1", api_key="lm-studio")

# Initialize the chat history
if "messages" not in st.session_state:
    st.session_state.messages = []

# Display the chat history (only user and assistant messages)
for message in st.session_state.messages:
    if message["role"] != "system": # Hide the system message from frontend
        with st.chat_message(message["role"]):
            st.markdown(message["content"])

# Accept user input
if prompt := st.chat_input("Ask your medical question:"):
    # Add user message to chat history
    st.session_state.messages.append({"role": "user", "content": prompt})
    with st.chat_message("user"):
        st.markdown(prompt)

    # System message to set the context for medical queries
    system_message = {
        "role": "system",
        "content": (
            "You are a medical assistant specializing in pharmaceutical products. "
            "Answer user queries about composition, usage, dosage, side effects, and other "
            "information related to medicines. If a query is outside your knowledge or unrelated to medical products, "
            "respond with: 'I can only provide information about pharmaceutical products.'"
        )
    },

```

## 8. Submission Evaluation Criteria

## 8.1 Individual Feature Contributions

Each team member contributed to different core aspects of the Pharma Knowledge Assistant, ensuring a collaborative approach to building the system:

- **Harshan:** Worked on **dataset web scraping**, where he expanded the initial dataset by scraping pharmaceutical product details from reliable sources using the `web_scraper.py` script. His efforts ensured that the dataset was comprehensive and consistent, covering a wide range of pharmaceutical products with all relevant information such as compositions, side effects, and dosages.
- **Harsha:** Focused on **RAG-based Question Answering (Q/A)**. He implemented the RAG model using LangChain to enable the system to answer user queries related to pharmaceutical products. He integrated the system with the dataset, ensuring that the model accurately retrieved and generated relevant information for user queries.
- **Urvashi:** Took the lead in developing the **summarization** feature. She modified the RAG pipeline to condense detailed pharmaceutical information into concise, user-friendly summaries. This feature allows users to quickly grasp the essential details of medications, such as their uses, side effects, and dosages.
- **Pragya:** Implemented the **recommendation system**. She developed the logic to recommend suitable medications based on user symptoms and conditions, ensuring that the system could suggest alternatives or relevant drugs while considering contraindications and side effects.
- **Nishank:** Led the integration of **Trulens** for **evaluation**. He tracked key metrics such as accuracy, precision, recall, F1 score, and response time to assess the performance of the system. His work ensured that the model's performance was continuously evaluated and fine-tuned for optimal accuracy and responsiveness.

## 8.2 Team Performance

The team worked collaboratively to integrate their individual contributions into a cohesive system. Each member brought their expertise to the table, and their individual efforts were critical to the overall success of the project:

- The dataset, created by **Harshan**, served as the foundation for all subsequent features, ensuring the system had accurate and extensive information to process.
- **Harsha's** work on RAG-based Q/A provided the core functionality of the system, allowing it to answer complex pharmaceutical queries effectively.

- **Urvashi's** summarization feature added significant value by ensuring that users could quickly obtain concise information about medications, enhancing user experience.
- **Pragya's** recommendation system made the assistant more personalized by suggesting medications based on specific user needs and conditions.
- **Nishank's** work on evaluating the system using **Trulens** ensured continuous performance monitoring, allowing for adjustments and improvements in real-time.

Overall, the team demonstrated effective collaboration and problem-solving skills, with each member contributing their expertise to deliver a fully functional and well-optimized Pharma Knowledge Assistant.