PES UNIVERSITY
CSE DEPARTMENT

LLM COURSE: ESA

Anantharaman Palacode Narayana Iyer

HACKATHON: PHARMA KNOWLEDGE ASSISTANT — ESA

📅 Due date: Dec 4th, 2024 before 5 pm IST 📅

Contents

1 Introduction 3
2 Project Scope 4
3 Dataset Description 4
4 Platform Requirements 5
5 High Level Requirements 5
6 Architecture 7
7 Implementation Steps 8
7.1 Preparing the Datasets 8
7.2 Develop a RAG Application for Question Answering 10
7.3 Implement a Recommender 10
7.4 Implement an Alternatives generator 11
7.5 Implement a Summarizer 11
7.6 Implementing an Agent Based Framework 11
7.7 Sample Questions 12
7.8 Build a GUI based Chatbot 12
7.9 Metrics Evaluation 13
8 Submission Evaluation Criteria 13

9 References 13

10 Project Rubric 14

Disclaimer

In this hackathon, we will be scrapping the web in order to build datasets. We declare that this was done solely for the academic purposes and we don't intend to violate any copyright or other legal requirements.

Some features to be developed during this project might appear like providing medical advice or emulating some functions of qualified healthcare personnel. We declare that any output of this assistant needs to be validated by qualified professionals and should not be treated as a genuine medical advice.

Usage of LLMs

You are permitted to use any generative AI tool, such as ChatGPT for the purpose of implementing this hackathon problem statement.

Individual Student Grading Process

Each team's contribution should approximately be in linear proportion of individual efforts, with the expectation that all team members contribute equally to the overall outcome. While you will work as a team of five students, individual performance will be the primary basis for grading, though the overall team's performance will also factor into the evaluation.

To ensure accountability, each student must declare at the start of the hackathon the specific features they will work on. This can be indicated in the provided Excel sheet using feature codes such as i, j, k, etc., corresponding to the features listed under *High-Level Requirements*. For example, selecting feature 1 means you will develop *"Answer user queries about pharmaceutical products in natural language,"* as specified in the requirements section.

Assessment will consider several factors, including your proposed design, the quality of your code, and a demonstration of the implemented feature.

1 Introduction

In this 6-hour hackathon, your challenge is to build an intelligent pharma knowledge assistant that leverages cutting-edge Large Language Model (LLM) application patterns. The assistant should help users query and analyze pharmaceutical product data in natural language in a user friendly way. The problem is designed to incorporate multiple LLM application patterns such as Agent-based design, Retrieval-Augmented Generation (RAG), and Prompt Engineering. You are not required to perform any finetuning for this project.

You are required to generate a custom dataset by web scrapping. To help you with this, we have provided a starter code and also a sample dataset of 10 products. You are required to expand this dataset to include all the products described in the given web site. The dataset contains a number of pharmaceutical products, including their names, descriptions, compositions, usage instructions, side effects, and more. Your solution should demonstrate creativity and technical sophistication in solving the tasks outlined in this problem statement. Please see the details of dataset in the following section: Dataset Description.

2 Project Scope

The scope of this hackathon includes using any prompt engineering techniques including ReAct, both naive and advanced RAG techniques, and building the end-to-end workflow using Langgraph agentic framework and GUI. You are required to use Langgraph and Langchain for agent design and orchestration and Streamlit for the GUI as these were part of our course.

The scope doesn't include using other LLM programming techniques such as Finetuning. You are required to implement chat history and also GUI for all features. The scope includes features (such as Question Answering, Summarization, etc.) pertaining to all the products described in the website. For instance, there could be a question like: "What are the side effects of Amoxycillin?" as this product is part of our dataset.

3 Dataset Description

You are provided with the starter code that implements a web scrapper for the system of web pages that covers the dataset. You are also provided a sample dataset of 10 products. Each product is described in a JSON file. For instance, the file "Amoxicillin Capsules, USP.json" contains the product details of Amoxycillin Capsules, USP. While most fields (keys) of the JSON for every products are same, there can be some differences between the JSON fields of each product.

For example, the fields of Amoxycillin capsules, USP are the following:

- HIGHLIGHTS OF PRESCRIBING INFORMATION
- Table of Contents
- 1 INDICATIONS AND USAGE
- 2 DOSAGE AND ADMINISTRATION
- 3 DOSAGE FORMS AND STRENGTHS
- 4 CONTRAINDICATIONS
- 5 WARNINGS AND PRECAUTIONS
- 6 ADVERSE REACTIONS
- 7 DRUG INTERACTIONS
- 8 USE IN SPECIFIC POPULATIONS
- 10 OVERDOSAGE
- 11 DESCRIPTION
- 12 CLINICAL PHARMACOLOGY
- 13 NONCLINICAL TOXICOLOGY
- 14 CLINICAL STUDIES
- 15 REFERENCES
- 16 HOW SUPPLIED/STORAGE AND HANDLING
- 17 PATIENT COUNSELING INFORMATION

- PACKAGE LABEL.PRINCIPAL DISPLAY PANEL
- INGREDIENTS AND APPEARANCE
- product_name

Before starting your design/coding, please review a few JSON files to get an idea of the way the products are described. The features that need to be developed are closely associated with the dataset.

4 Platform Requirements

- **Operating system:** Windows 11, Linux, **or** macOS
- **GPU:** For macOS, M2 and above silicon, or NVIDIA CUDA supported GPUs for Windows with > 12 GB RAM. You can also use Google Colab. Based on memory and speed requirements, obtain either T4 or A100 GPU.
- **IDE:** PyCharm, VS Code, or any other Python development system with Anaconda environment.
- **Frameworks:** Ollama, LMStudio.
- **LLM:** Llama 3 Family, Google gemma, Microsoft Phi 3 Family.
- **Web Browser:** Google Chrome.

5 High Level Requirements

Your objective is to build an **"Agent Based"** Pharma Knowledge Assistant that can:

1. Answer user queries about pharmaceutical products in natural language.
   For example:
   User Query: *"What is the composition and primary use of Paracetamol?"*
   Assistant Response: "Paracetamol, also known as acetaminophen, primarily contains the active ingredient acetaminophen. It is commonly used to relieve pain and reduce fever. Paracetamol is suitable for mild to moderate pain such as headaches, toothaches, and muscle aches. However, it should be used with caution as excessive doses can cause liver damage. Always follow the dosage instructions provided or consult your healthcare provider for safe use."

2. Generate recommendations or warnings based on user-provided symptoms, conditions, or medication combinations.
   For example:
   User Query: *"Can I take Ibuprofen if I have a history of stomach ulcers?"*
   Assistant Response: "Ibuprofen is a nonsteroidal anti-inflammatory drug (NSAID) that can irritate the stomach lining and may worsen your condition if you have a history of stomach ulcers. It is generally not recommended for individuals with this condition. You could consider acetaminophen as an alternative for pain relief, as it is less likely to irritate the stomach. Please consult your healthcare provider for personalized advice and to ensure safe

use."

3. Summarize product details for user-friendly explanations.
   For example
   User Query: *"Summarize the details of Amoxicillin."*
   Assistant Response: "Amoxicillin is an antibiotic used to treat a variety of bacterial infections, including respiratory tract infections, ear infections, and urinary tract infections. It works by inhibiting the growth of bacteria. Amoxicillin is usually taken orally, with or without food, as directed by a healthcare provider. Common side effects may include nausea, diarrhea, or a mild skin rash. Serious allergic reactions are rare but can occur. Always follow the prescribed dosage and consult your doctor if you experience any unusual symptoms."

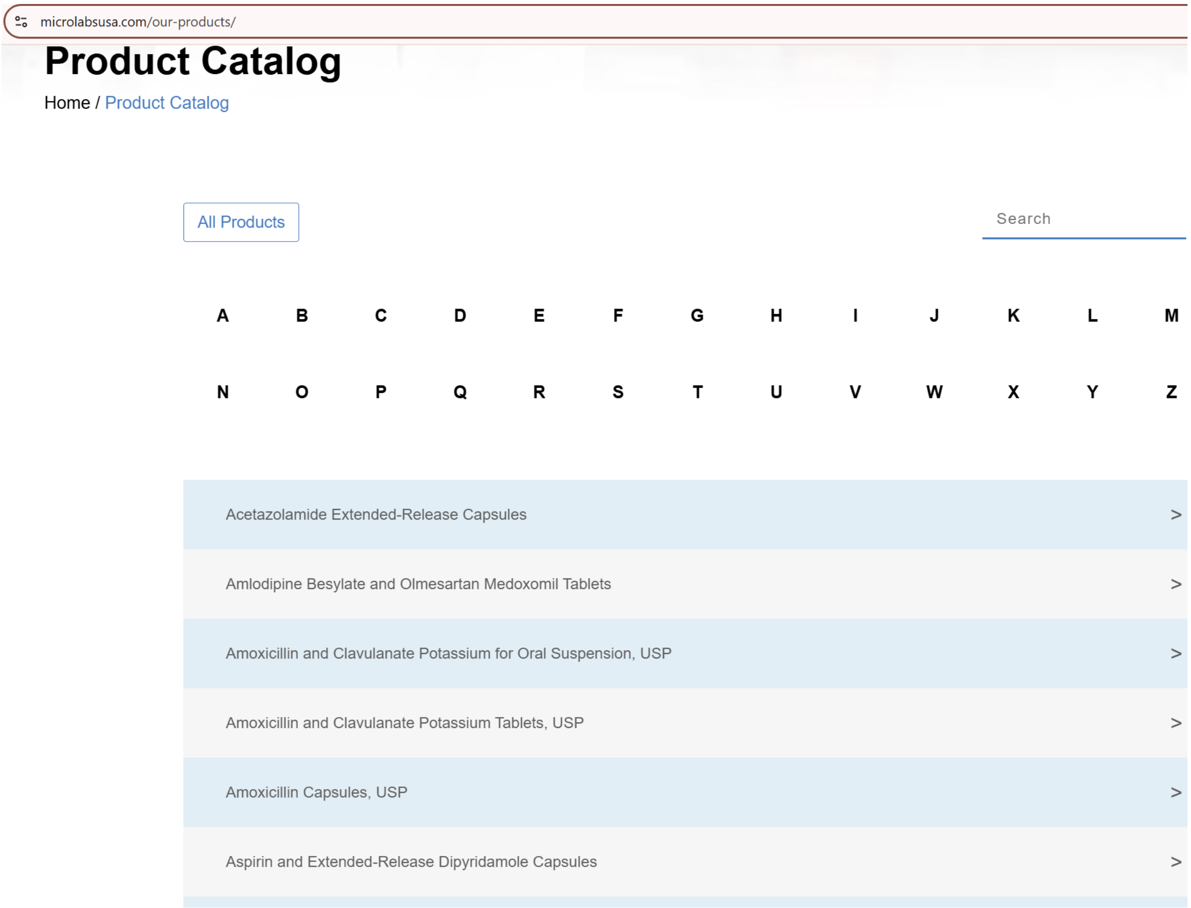
4. Provide references to relevant products using Retrieval-Augmented Generation.
   For example:
   User Query: *"What are some alternatives to Ibuprofen for pain relief?"*
   Assistant Response: "Based on your query, here are some alternatives to Ibuprofen for pain relief:

   - **Acetaminophen (Paracetamol):** Suitable for mild to moderate pain and fever reduction. It is gentler on the stomach compared to NSAIDs like Ibuprofen.
   - **Naproxen:** Another NSAID that can be used for pain relief, but it should be taken with caution and food to minimize stomach irritation.
   - **Aspirin (low dose):** Effective for pain relief, though it may not be ideal for those with certain conditions such as ulcers.

     Please note that the suitability of these alternatives depends on your specific health conditions. Always consult a healthcare provider for personalized recommendations."

5. Utilize Agent-based design for interactive and task-driven problem solving.
   For example:
   User Query: *"I have a headache and a history of stomach ulcers. Can you recommend a safe medication?"*
   Assistant Interaction (Multi-Step):

   (a) **Agent Action 1: Analyze Query** The agent identifies the user's symptoms (*headache*) and condition (*history of stomach ulcers*).
   (b) **Agent Action 2: Retrieve Relevant Data** The agent searches the dataset for medications used to treat headaches and filters out drugs that may irritate the stomach lining (e.g., NSAIDs such as Ibuprofen).
   (c) **Agent Action 3: Evaluate and Recommend** Based on the analysis, the agent recommends medications that are less likely to cause stomach irritation.



microlabsusa.com/our-products/

Product Catalog

Home / Product Catalog

All Products

Search

A B C D E F G H I J K L M

N O P Q R S T U V W X Y Z

Acetazolamide Extended-Release Capsules >

Amlodipine Besylate and Olmesartan Medoxomil Tablets >

Amoxicillin and Clavulanate Potassium for Oral Suspension, USP >

Amoxicillin and Clavulanate Potassium Tablets, USP >

Amoxicillin Capsules, USP >

Aspirin and Extended-Release Dipyridamole Capsules >

Figure 1: Base web page of **Micro Labs USA**

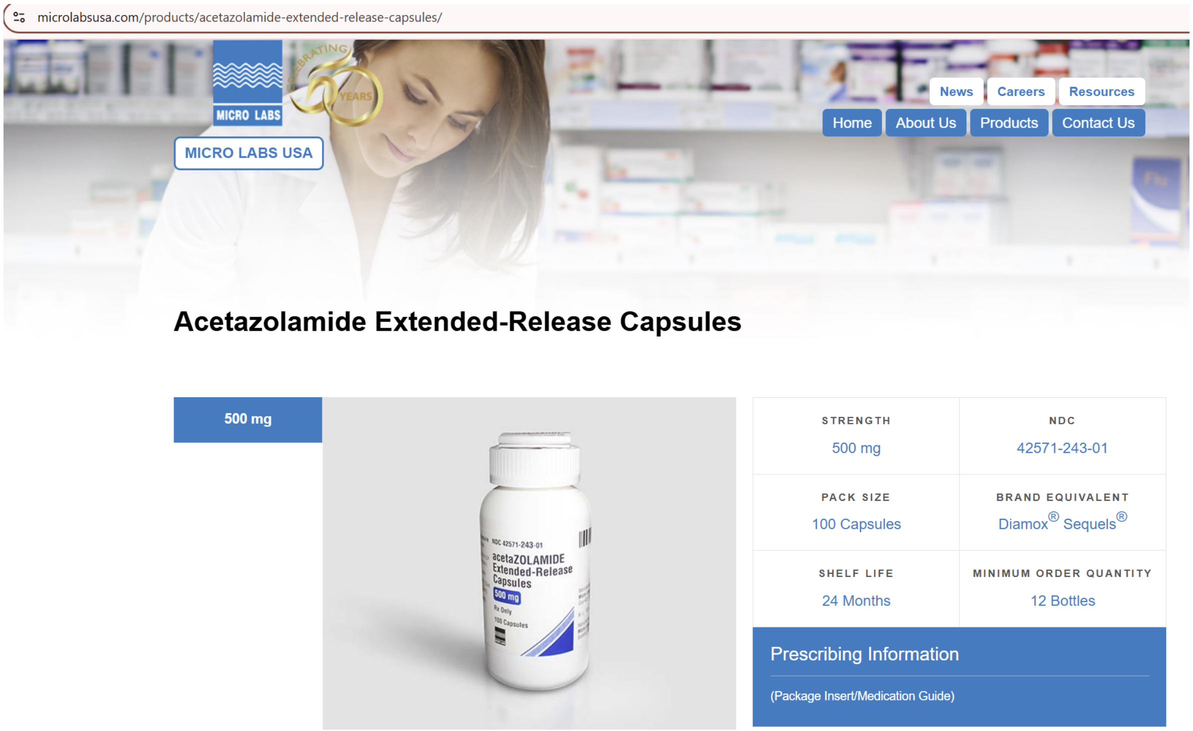
(d) **Response to User:** "For your headache, I recommend acetaminophen as it is effective for pain relief and less likely to aggravate stomach ulcers. Please consult your healthcare provider for confirmation and safe dosage instructions."

Follow-Up Query: *"What dosage should I take for acetaminophen?"*
Agent Response: "The standard dosage for acetaminophen is typically 500–1000 mg every 4–6 hours as needed, with a maximum limit of 4000 mg per day for adults. However, exact dosages may vary based on your medical history and condition. Please confirm with your healthcare provider."

6. Develop a search tool based agent that can be invoked when the user query can't be answered from the available products in the dataset. For example, the user queries about the side effects of a particular drug which is not answerable by RAG, use an agent to fetch the answer from internet search.

6 Architecture

You are required to design the complete architecture as an Agentic application, where individual features such as RAG based Question Answering, Summariza-



Acetazolamide Extended-Release Capsules

STRENGTH	NDC
500 mg	42571-243-01
PACK SIZE	BRAND EQUIVALENT
100 Capsules	Diamox® Sequels®
SHELF LIFE	MINIMUM ORDER QUANTITY
24 Months	12 Bottles

Prescribing Information

(Package Insert/Medication Guide)

Figure 2: Product example web page of **Micro Labs USA**

tion, Search tools etc. are nodes of the Langgraph.

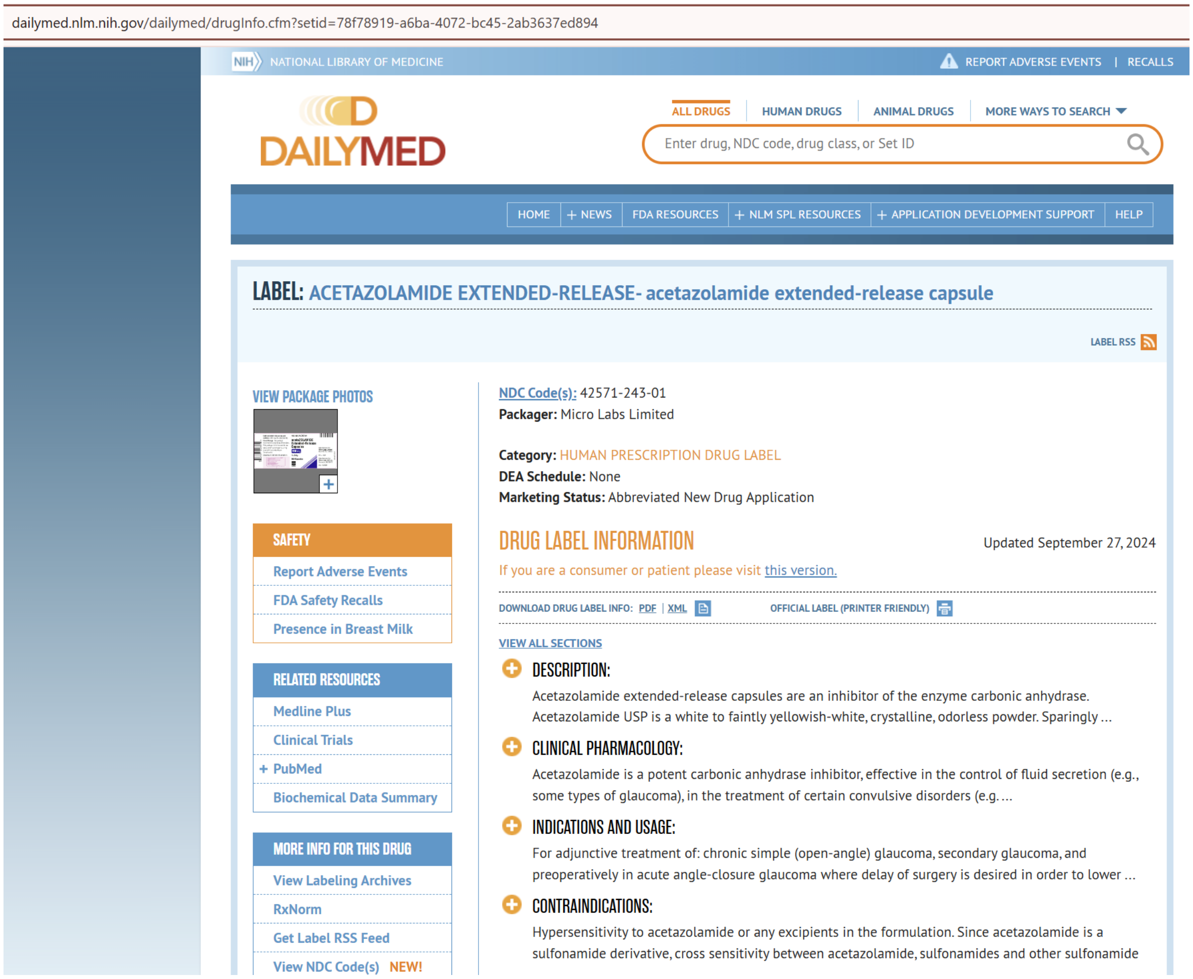
You should report a graphical visualization of your implementation in your documentation and during your demo.

7 Implementation Steps

Please refer the Figure 1 that provides visualization of the base URL. You can view at `https://www.microlabsusa.com/our-products/`

7.1 Preparing the Datasets

- You are provided with the following starter material:
  1. **<product-name.json>** This JSON is already a preprocessed form of scrapped content of prescribing information web page. We will have one such JSON for every product. You can OPTIONALLY perform any additional data cleaning or any other processing to enhance the metadata content. You are given a sample of 10 such JSONs.
  2. **Starter code: web_scrapper.py** This file implements the necessary code to take the list of URLs from a python list **URLS** and for each such url that corresponds to a product, retrieve the prescribing information. This is further written in JSON format to a file.



dailymed.nlm.nih.gov/dailymed/drugInfo.cfm?setid=78f78919-a6ba-4072-bc45-2ab3637ed894

LABEL: ACETAZOLAMIDE EXTENDED-RELEASE- acetazolamide extended-release capsule

NDC Code(s): 42571-243-01
Packager: Micro Labs Limited

Category: HUMAN PRESCRIPTION DRUG LABEL
DEA Schedule: None
Marketing Status: Abbreviated New Drug Application

DRUG LABEL INFORMATION

Updated September 27, 2024

If you are a consumer or patient please visit this version.

DESCRIPTION:

Acetazolamide extended-release capsules are an inhibitor of the enzyme carbonic anhydrase. Acetazolamide USP is a white to faintly yellowish-white, crystalline, odorless powder. Sparingly ...

CLINICAL PHARMACOLOGY:

Acetazolamide is a potent carbonic anhydrase inhibitor, effective in the control of fluid secretion (e.g., some types of glaucoma), in the treatment of certain convulsive disorders (e.g. ...

INDICATIONS AND USAGE:

For adjunctive treatment of: chronic simple (open-angle) glaucoma, secondary glaucoma, and preoperatively in acute angle-closure glaucoma where delay of surgery is desired in order to lower ...

CONTRAINDICATIONS:

Hypersensitivity to acetazolamide or any excipients in the formulation. Since acetazolamide is a sulfonamide derivative, cross sensitivity between acetazolamide, sulfonamides and other sulfonamide

Figure 3: Prescribing Information example web page of **Micro Labs USA**

- Review the datasets mentioned above, optionally enhance them as required. Come up with an uniform schema that fits all the JSONs.
- Note that you have been given dataset for only 10 products. Your **first task** is to use the starter code to expand the dataset to cover all the products mentioned in the base url and other links that mentions other products: A to Z.

7.2 Develop a RAG Application for Question Answering

- Answer user queries about pharmaceutical products in natural language. You should develop it using LangChain and use Streamlit for GUI.
- Ingest the dataset which is in JSON form. You can try different hyperparameters, vector stores and loading methods. Since the information is already structured as JSON, you can use JSON loader for RAG. You can also try splitting each JSON file as multiple JSONs, where each file contains 1 section, e.g. Contraindications.
- Choose suitable LLM, run it as a server under either Ollama or LMStudio. Write a client code to access the LLM.
- Write the code for implementing RAG using Langchain features
- Perform a simple evaluation of the quality of your RAG for question answering.
- Note that this RAG based feature should be one or more nodes in your Agent based application. This is not a standalone feature. It should be invoked through an agent graph and should share the context with other nodes of the graph.

After you complete the above, your implementation should be able to answer queries like: "What is the composition and primary use of Paracetamol?" as discussed under the section: High Level Requirements.

7.3 Implement a Recommender

Based on the kind of input from the user, you should execute the corresponding feature node in the agentic graph. Here you implement a *Recommender or Warnings* feature using preferably agent approach, though you can do this with RAG also. If you consider a query like: *Can I take Ibuprofen if I have a history of stomach ulcers?*, you need to answer it with some reasoning or web search, where you have to determine whether the given medicine causes stomach ulcers and recommend alternatives if needed.

- Modify the RAG implementation to implement product recommendation
- Perform a quick evaluation and verify the performance of the RAG
- Implement the same using agents
- Evaluate and choose the right approach
- Implement this feature such that it is integrated with the larger Agent application. This is to be invoked using the agent graph and should share the context with other nodes of the graph.

7.4 Implement an Alternatives generator

If the user requires suggestion on alternatives, you can implement a specialized agent or RAG in a way similar to the previous approach.

7.5 Implement a Summarizer

- Modify the RAG implementation to perform summarization. This involves modifying the prompt appropriately.
- You can implement this by creating short summaries of each section and you can combine this to form a complete summary
- You can also come up with any other creative methods to implement summarization that may not need RAG.
- Implement this feature such that it is integrated with the Agent application. This is to be invoked using the agent graph and should share the context with other nodes of the graph.

7.6 Implementing an Agent Based Framework

Your overall project should be based on an Agent based application pattern, where the application is decomposed in to several distinct functionalities implemented as nodes of a graph. This should also include tools like search tool.

- **Develop the Specifications:** Given the High Level Requirements, identify the feature requirements in more detail, identify the owner in the team who will develop the feature.
- **Graph Design:** Based on the feature specifications perform the design. Identify the nodes and edges, start node, tools, etc.
- **Context Design:** Based on the specifications, develop the context data structure. For example, if you are implementing a search tool you may identify fields such as "search input", "search results" etc., where one of the nodes fills up the search input, search tools writes to the search results and a downstream node consumes the search results.
- **Implement the functions:** Each node needs to be implemented using a function or a method of a class. Develop these functions
- **Implement Router nodes:** This project involves making lot of decisions. For example, if there is a query: "Which diseases are treated by Amoxycillin?" a router needs to invoke a RAG based question answering node. If that doesn't provide a good answer, it needs to try a web search. Similarly, if the question pertains to some analysis or reasoning, it is necessary to invoke corresponding nodes that may be specialized agents. A router node is one that uses the context to make a decision and invokes the right node.
- **Unit Test:** Test the individual nodes and ensure their functionality is implemented correctly.
- **Integrate:** Set up start nodes, Compile the graph. Give some input and test the whole application in an integrated way. Check all features of every team member.

7.7 Sample Questions

In this sub section we provide some sample questions that can be used to test the project. The faculty will use an elaborate set of questions to verify your features.

Example Questions and Suitable LLM Techniques

Question 1: Explain why aspirin and extended-release dipyridamole capsules are contraindicated in patients with known hypersensitivity. Provide examples of specific conditions or reactions that justify this contraindication.
Suitable LLM Technique: Retrieval-Augmented Generation (RAG)
Reason: The question requires retrieving specific medical contraindications and examples from a knowledge base or document repository to provide an accurate and fact-based response.

Question 2: Discuss the potential risks of administering aspirin to a child or teenager with a viral infection. Why is this practice discouraged, and what condition might develop as a result?
Suitable LLM Technique: Agent-Based Framework
Reason: This question involves task-driven reasoning where the agent must identify the risks (e.g., Reye syndrome), validate the query using multiple tools or knowledge sources, and provide a coherent explanation.

Question 3: Volini is a widely known pain relief gel that is used for various conditions. Based on your understanding of pharmaceutical products, discuss:

- The type of active ingredients typically found in over-the-counter pain relief gels.
- How such products function to alleviate pain and inflammation.
- Precautions a user should take when applying pain relief gels like Volini.

Additional Notes: Volini is not included in the dataset provided for the hackathon. Your answer should demonstrate reasoning and synthesis of general pharmaceutical knowledge.

Suitable LLM Technique: Agent-Based Framework with Reasoning Capabilities
Reason: Since Volini is not in the dataset, an agent can decompose the query into tasks such as understanding the category of pain relief gels, identifying typical active ingredients (e.g., diclofenac, menthol), and generating a synthesized response by reasoning over general knowledge and user-specified requirements.

7.8 Build a GUI based Chatbot

In this step, you are required to implement a chatbot using Streamlit library. This allows the user to ask a question through a GUI interface and get the response from the LLM, display the results in graphical form (charting) as needed.

Refer: `https://github.com/streamlit/llm-examples/blob/main/Chatbot.py` and `https://docs.streamlit.io/develop/tutorials/llms/build-conversational-apps`

7.9 Metrics Evaluation

You are required to use TruLens and evaluate your work using metrics that include: Context Relevance, Answer Relevance, Groundedness and Comprehensiveness. You can also add your own metrics to this list as needed.

8 Submission Evaluation Criteria

Submissions will be evaluated based on the following criteria:

- **Accuracy:** Comprehensiveness of Dataset Creation.
- **Relevance:** The questions from the faculty and relevance of answers.
- **Creativity in Design:** Innovative and effective use of several design techniques
- **Demoe Experience:** The product's ability to engage users, provide clear explanations, and handle follow-up questions.
- **Efficiency:** The product's ability to handle complex queries while minimizing errors and maintaining the flow of conversation.

Deliverables

You must submit:

- A working LLM-powered Agent application capable of providing the features as discussed in this problem statement.
- A report documenting the design and key results.
- A demonstration of the product.
- Source code
- Any videos of your work
- Shared link that contains the deliverales

9 References

- Class notes and slides
- Langgraph tutorials: `https://langchain-ai.github.io/langgraph/tutorials/introduction/`
- Agentic RAG using Langgraph: `https://langchain-ai.github.io/langgraph/tutorials/rag/langgraph_agentic_rag/`
- Langchain Tutorials `https://python.langchain.com/docs/tutorials/`
- Micro Labs USA products: `https://www.microlabsusa.com/our-products/`
- Ollama: `https://ollama.com`
- LMStudio: `https://lmstudio.ai`

10 Project Rubric

Please use the rubric given below to evaluate your work:

Task	**Completion %**
Dataset created as required in this problem statement	
Specifications done	
Top level agentic framework with context designed	
RAG nodes for the necessary features done	
Streamlit based GUI implemented and is used to provide inputs and view the outputs	
Agent nodes for different features implemented	
Search tool implemented	
Final integration completed	
Evaluation against metrics performed and reported	
Final Demo done	