**Capstone Project Concept Note and Implementation Plan**

**AI-Powered Clinical Decision Support System (CDSS) using Retrieval-Augmented Generation (RAG)  
  
Team Members**

1. Beza Mesfin Mekuria
2. Nahom Temesgen Nadew
3. Natnael Amsalu
4. Samuel Gashu Getahun
5. **Project Overview**

This project focuses on developing an **AI-Powered Clinical Decision Support System (CDSS)** using **Retrieval-Augmented Generation (RAG)** to enhance healthcare delivery. It aligns directly with **Sustainable Development Goal (SDG) 3: Good Health and Well-being** by improving diagnostic accuracy and supporting evidence-based medical decisions. Additionally, it contributes to **SDG 4: Quality Education** by providing continual learning opportunities for healthcare professionals, and **SDG 1: No Poverty** by reducing the financial burden caused by misdiagnoses and unnecessary treatments. The problem addressed is the high rate of diagnostic errors and limited access to real-time, context-aware clinical insights. This solution aims to bridge the knowledge gap in clinical environments, especially in under-resourced settings.

1. **Objectives**

* To design and implement an AI-powered CDSS that retrieves and synthesizes medical literature and patient data using RAG.
* To enhance diagnostic precision and reduce the rate of misdiagnosis in clinical settings.
* To improve the accessibility and usability of healthcare knowledge for both professionals and patients.
* To demonstrate the integration of deep learning models with medical datasets for real-time, context-aware recommendations.
* To support sustainable and scalable healthcare solutions through intelligent systems.

1. **Background**

Misdiagnosis remains a persistent issue in healthcare, contributing to patient harm and increased healthcare costs. Traditional decision support systems often lack real-time adaptability, are rule-based, or cannot integrate vast medical literature with patient-specific information. Existing AI models, while powerful, can generate hallucinated responses or lack domain-specific grounding. Recent advancements like RAG offer a promising approach by combining retrieval-based methods with generative models to improve the contextual relevance and accuracy of responses. The MIMIC-III dataset, which includes de-identified health data from real ICU patients, provides a rich source of clinical data for building robust diagnostic tools. Projects such as MedRAG and ClinicalRAG demonstrate the transformative potential of this approach but are still limited in accessibility and application breadth. This project aims to build upon these efforts and create a practical, scalable tool.

1. **Methodology**

Retrieval-Augmented Generation (RAG) is a methodology that combines retrieval-based and generative approaches in natural language processing (NLP) to improve the quality and relevance of generated responses. Here’s an overview of RAG:

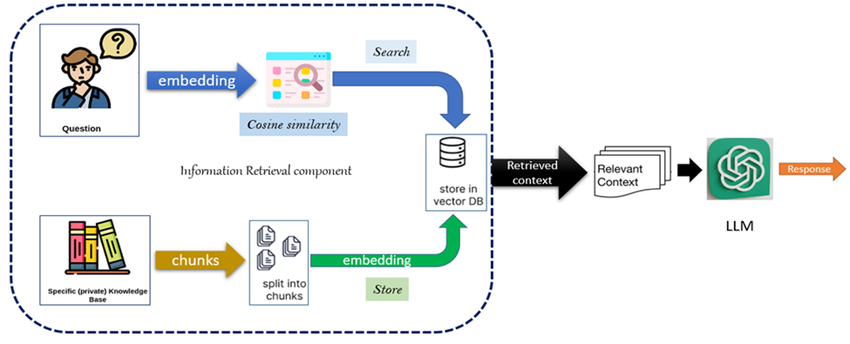
RAG integrates two main components:

1. **Retrieval Component**: This part involves retrieving relevant information from a large corpus of documents or data. It typically uses techniques like information retrieval (IR) to find pertinent text based on a query.
2. **Generation Component**: After retrieving relevant documents, a generative model (often a transformer-based model like GPT) is used to formulate coherent and contextually appropriate responses by synthesizing information from the retrieved documents.

**Workflow**

1. **Query Input**: The user provides a query or prompt.
2. **Information Retrieval**:
   * The retrieval system searches for a knowledge base or corpus to find relevant documents or snippets.
3. **Contextual Generation**:
   * The generative model takes the retrieved information and the original query to produce a response that is informed by the additional context.
4. **Output**: The final generated answer is presented to the user.
5. **Architecture Design Diagram**

High-Level Architecture Overview



**Component Descriptions**

1. **Question:** Interface for clinicians or patients to input symptoms or medical queries.
2. Embedding (Question): This question is then converted into a numerical representation called an embedding. This embedding captures the semantic meaning of the question.
3. Specific (private) Knowledge Base: You have a collection of documents or data that serve as your knowledge source.
4. Split into chunks: The knowledge base is often broken down into smaller, manageable pieces called chunks.
5. Embedding (Chunks): Each of these chunks is also converted into embeddings.
6. Store in vector DB: These chunk embeddings are stored in a vector database, which is optimized for efficient similarity searching.
7. Cosine Similarity: The embedding of the user's question is compared to the embeddings of all the chunks in the vector database using a similarity metric like cosine similarity. This helps identify the chunks that are most relevant to the question.
8. Search: The vector database is searched to retrieve the top-k most similar chunks based on the cosine similarity scores.
9. Retrieved Context: The most relevant chunks are retrieved as the "relevant context."
10. LLM (Large Language Model): The original question and the retrieved relevant context are fed into a large language model.
11. Response: The LLM then uses both the question and the retrieved context to generate a more informed and accurate response than it could have produced using only its pre-trained knowledge
12. **Data Sources**

This project will utilize medical textbooks, providing a structured knowledge base covering standardized medical information, such as treatment guidelines, disease descriptions, and symptomatology. The textbook component is useful for training systems to understand baseline medical standards and recommendations, which are critical for diagnostics and treatment decision support

1. **Literature review**Studies such as Lewis et al. (2020) on RAG, ClinicalRAG, and MedRAG show that hybrid retrieval-generation architectures significantly reduce hallucinations and increase diagnostic accuracy. Our system builds upon this research by adding domain-specific tuning and an explainability layer to make the tool reliable for clinical use.

**Implementation Plan**

1. **Technology Stack**

* **Programming Language:** Python
* **Libraries & Frameworks:**
  + Hugging Face sentence-transformers – Pretrained language models
  + Langchain - Provides the core functionality for building the RAG pipeline, including chains for document retrieval and question answering
  + Google Gemini API- used for generating responses based on retrieved documents and user input.
  + Pinecone – vector storage and retrieval, enabling similarity-based document search
  + NumPy, gensim text blob – text preprocessing and analytics
* **Web Framework:** Streamlit
* **Deployment:** Streamlit / Docker
* **Hardware:** GPU-enabled environment & Google colab

1. **Timeline**

**Task Distribution Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | Beza | Nahom | Natnael | Samuel |
| Literature Review |  |  | ✅ |  |
| Data Preprocessing |  | ✅ |  | ✅ |
| Model Development | ✅ |  | ✅ |  |
| Evaluation & Tuning | ✅ | ✅ |  |  |
| Frontend Integration |  |  | ✅ | ✅ |
| Documentation & Reporting | ✅ | ✅ |  | ✅ |

1. **Milestones**

* Completion of literature review and finalized architecture design.
* Preprocessed and indexed MIMIC-III dataset.
* Working RAG pipeline integrated with langchain\_pinecone and Hugging Face models.
* Initial model evaluation metrics met (e.g., F1-score, accuracy).
* Fully deployed prototype with chatbot interface.
* Final report and project presentation ready.

1. **Challenges and Mitigations**

|  |  |
| --- | --- |
| **Challenge** | **Mitigation Strategy** |
| Data Quality Issues | Apply robust preprocessing, missing value handling, and domain-specific cleaning. |
| Model Hallucination | Use RAG to ground outputs in retrieved documents and apply domain-tuned LLMs like BioGPT. |
| High Computational Demand | Use cloud-based GPUs or Google Colab Pro for training and evaluation. |
| Interpretation of Results | Collaborate with medical experts to validate outputs and fine-tune response logic. |

1. **Ethical Considerations**Given the use of sensitive healthcare data, this project adheres to strict data privacy standards. This response generated by this project is not a substitute for professional medical advice

**References**

* **Johnson, A.E.W., et al. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data.***
* **Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv preprint arXiv:2005.11401.***
* FAISS Documentation – <https://github.com/facebookresearch/faiss>
* LangChain Documentation – https://docs.langchain.com/