Project Report

AI-Powered Medical Assistant Chatbot for Preliminary Health Advice

Group Members:
Deep Patoliya
Dhruv Shah
John Miller
Vedant Kadam

Table of Contents

Goal of the Project – 3

How the System works – 3
- **User Interaction**
- **Query Processing Pipeline**
- **Information Retrieval via RAG Framework**
- **Response Generation**
- **Backend Architecture**

Strengths and Weaknesses of the Project - 5
- **Strengths**
- **Weaknesses**

API Usage (which LLMs and how they are used) - 5

Data used, data handling processes – RAG usage, other usage - 7
- **Primary Data Sources**
- **Data Preprocessing**
- **Embedding Generation**
- **Metadata Attachment**

Vector Database Usage - 8

Functions / Tools Used - 8

Data privacy, bias in AI models, and the broader societal impact of project - 10
- **Data Privacy**
- **Bias in AI Models**

Reflection on how their project aligns with ethical AI practices and measures taken to mitigate potential ethical issues - 11
- **User Privacy and Data Security**
- **Transparency and Source Citation**
- **Mitigating Bias in AI Models**
- **Broader Societal Impact**
- **Proposed Additional Measures**

Individual Section - 13

Goal of the project:

The goal of this project is to develop a robust and reliable medical chatbot that leverages Retrieval-Augmented Generation (RAG) technology to provide accurate, evidence-based answers to medical queries. Designed to ensure source transparency, personalized user experiences through account authentication, and domain-specific optimization using state-of-the-art biomedical models, this system aims to improve access to credible medical information while maintaining user privacy and adhering to ethical AI principles. Additionally, the chatbot includes a daily health tracker for individuals with chronic conditions, enhancing their ability to monitor vital signs such as heart rate and blood pressure. This tool aims to be a supplemental resource that enhances user knowledge and supports informed decision-making regarding health issues. Ultimately, the project strives to empower users with trustworthy knowledge, bridging gaps in health literacy and supporting informed decision-making.

How the system works:

The system integrates several advanced technologies to create a user-friendly, efficient, and reliable RAG-based medical chatbot. The architecture is designed to ensure personalized interaction, domain-specific accuracy, and source transparency. The following sections outline the detailed workings of the system:

- **User Interaction**
  - **Account Authentication**:
    - New users can create an account, providing basic credentials for registration.
    - Returning users can log in to access their profiles and view their previous conversations with the chatbot, ensuring a seamless, personalized experience.
    - Secure authentication protocols (e.g., hashed passwords and token-based sessions) safeguard user data and privacy.
  - **Query Input**:
    - Users interact with the chatbot through a simple and intuitive interface, entering medical queries in natural language.

- **Query Processing Pipeline**
  - **2.1 Natural Language Processing (NLP)**
    - The user's query is tokenized and preprocessed for optimal embedding generation.
    - NLP techniques ensure that the query is correctly understood in a medical context, capturing its intent and specificity.
  - **2.2 Query Embedding**
    - The processed query is embedded using **PubMed BERT**, a language model fine-tuned on biomedical data.
  - **Why PubMed BERT?**

        - It ensures high relevance and accuracy in vector representations of medical queries, as it is optimized for domain-specific language.

- **Information Retrieval via RAG Framework**
    - **3.1 Vector Database (FAISS-Based)**
        - The system utilizes a **FAISS-based vector database** for storing and retrieving embeddings of medical knowledge.
        - The database contains embeddings generated from the **Gale Encyclopedia of Medicine** and a curated dataset of approximately 200,000 medicines, including their uses and associated details.
    - **How it Works**:
        - The query embedding is compared against the stored embeddings in the FAISS database using similarity search algorithms (e.g., cosine similarity).
        - The most relevant documents or entries are retrieved based on similarity scores.
    - **3.2 Document Context Retrieval**
        - The retrieved documents provide relevant context for answering the query. These documents include:
            - Medical explanations from the **Gale Encyclopedia of Medicine**.
            - Specific details about medicines, their uses, and any related information from the medicines dataset.
- **Response Generation**
    - **4.1 Generative Model (BIOMISTRAL)**
        - The retrieved context is passed to the **BIOMISTRAL** language model to generate a coherent and contextually accurate response.
        - **Why BIOMISTRAL?**
            - BIOMISTRAL is fine-tuned for medical applications, ensuring that responses are precise, clinically relevant, and aligned with the input query.
        - The model synthesizes the retrieved context with the user's query, creating an informative and user-friendly answer.
    - **4.2 Source Transparency**
        - The chatbot appends the source(s) of the information in its response to ensure credibility and enable users to verify the details independently.
            - For example, responses include citations like:
                - "Source: Gale Encyclopedia of Medicine" and
                - "Source: Medicines Dataset, Entry: [Medicine Name]"
- **Backend Architecture**
    - **6.1 Data Handling**
        - **Data Sources**:
            - **Gale Encyclopedia of Medicine**: A comprehensive knowledge base of medical conditions, treatments, and terminologies.
            - **Medicines Dataset**: Contains detailed information on over 200,000 medicines and their uses.
        - **Preprocessing**:

        - The raw data is cleaned, normalized, and embedded using **PubMed BERT** to create vector representations.
        - Metadata, including source information, is linked to each embedding for retrieval transparency.
    - **6.2 Vector Storage (FAISS)**
        - FAISS serves as the vector database, efficiently handling:
            - Storage of embeddings for all preprocessed data.
            - Real-time similarity searches during query processing.
            - Scalability to accommodate additional medical data as needed.

Strengths and Weaknesses of project:

Strengths:

- **Advanced Technology Integration:**
  The chatbot leverages cutting-edge technologies such as the Llama 2 model and FAISS for data retrieval, enhancing its ability to process and respond to medical queries accurately and swiftly. This robust technological backbone ensures reliable performance and high-quality user interactions.

- **Comprehensive Medical Knowledge:**
  With a medical knowledge retrieval system that includes source attribution, the chatbot has access to a vast and reliable medical database. This allows it to provide well-informed responses, which can improve user trust and satisfaction.

- **User-Friendly Interface:**
  The use of Streamlit for the chat interface ensures that the system is accessible and easy to use.

Weaknesses:

- **Dependence on Accurate User Input:**
  The effectiveness of the chatbot is heavily reliant on the accuracy of the information provided by users. Miscommunication or incorrect data entry by users can lead to inappropriate responses, potentially affecting the reliability of health advice.

- **Limited Scope of Medical Advise:**
  While the chatbot can handle general medical queries and symptom analysis, its ability to provide comprehensive medical advice is limited. It cannot replace professional medical diagnosis or treatment, which might be expected by some users.

- **Technical Challenges in Scalability:**
  Scaling the chatbot to handle a large number of concurrent users while maintaining performance and accuracy can be challenging. Technical limitations, such as those associated with server capacity or algorithm efficiency, could affect service quality as user numbers grow.

API Usage (which LLMs and how they are used):

Large Language Models(LLMs):

Llama 2 (7B quantized model)

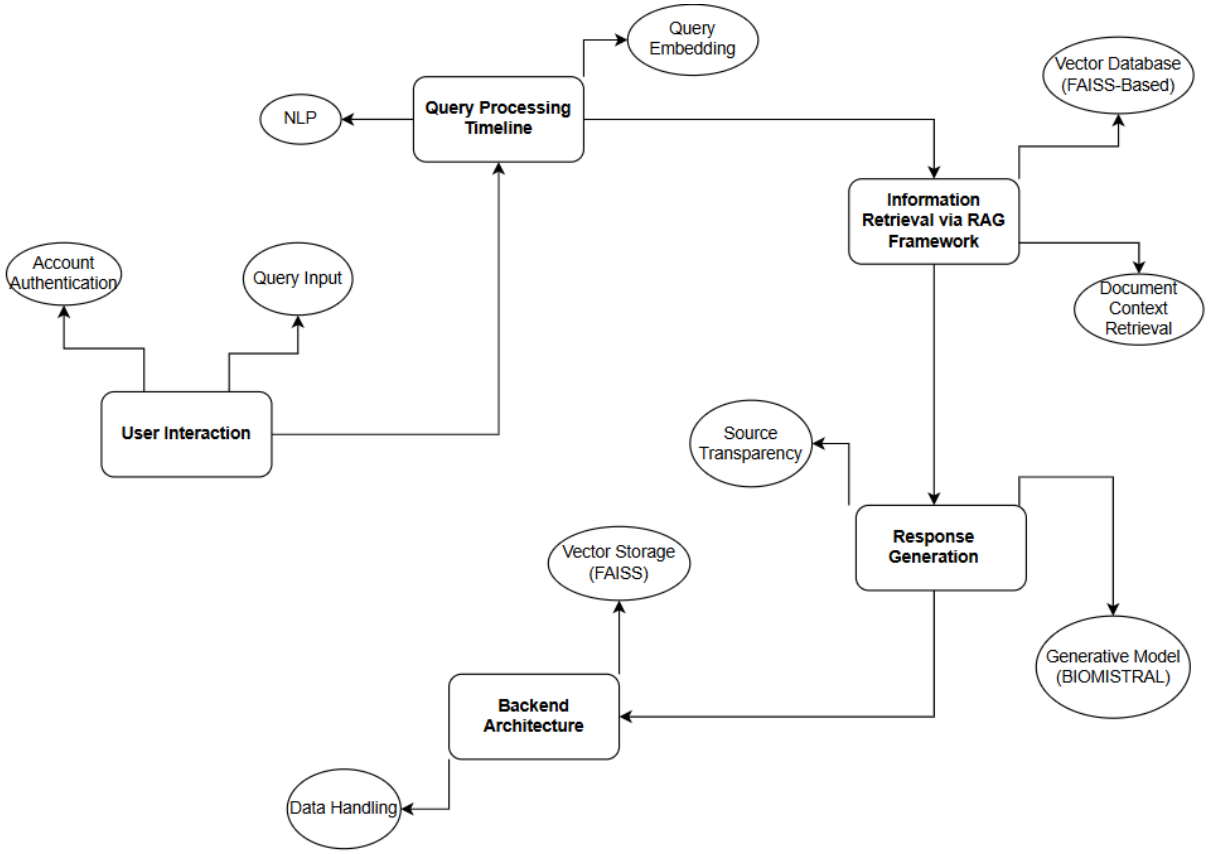
Usage: This model is central to the RetrievalQA system. It processes and interprets user-input symptoms or questions, retrieving relevant medical information based on the context provided. The Llama 2 model is particularly adept at understanding natural language, which allows it to provide accurate and relevant responses to users' medical queries.
Integration: The Llama 2 API is called whenever a user inputs a query. The system sends the query to the API, which then processes the text and returns information or answers extracted from its training data and integrated medical databases.

APIs and Tools:

- **Streamlit:**
    - **Usage:** Streamlit is used to build the interactive chat interface where users input their medical questions and symptoms. It provides a user-friendly environment that makes the chatbot accessible to a wide range of users, regardless of their technical skills.
    - **Integration:** The system uses Streamlit to render the frontend interface directly. It connects the backend processing, handled by Llama 2 and FAISS, with the user interface, ensuring that all components work together seamlessly.

- **Langchain:**
    - **Usage:** Langchain is employed for system orchestration and document handling. It helps in managing the flow of data between different components of the system, ensuring that user queries are processed efficiently and accurately.
    - **Integration:** Langchain acts as the middleware that integrates the LLM outputs and document retrieval functionalities with the Streamlit interface. It ensures that the data flow within the system is smooth and that the responses generated by the LLMs are correctly formatted and displayed to the user.

- **Usage Flow:**



Data used, data handling processes – RAG usage, other usage:

Data Used:

- **Primary Data Sources:**
  - Gale Encyclopedia of Medicine:
    - A comprehensive repository of medical knowledge, covering diseases, conditions, treatments, and medical terminologies.
    - Used as the foundational knowledge base for the chatbot to retrieve contextually relevant medical information.
  - Medicines Dataset:
    - Contains approximately 200,000 entries detailing:
      - Medicines, their uses, and side effects.
    - Provides domain-specific granularity for queries related to pharmaceuticals.

- **Data Preprocessing:**
  - Cleaning and Normalization:
    - Text from the datasets is preprocessed to remove inconsistencies, such as typos, redundant data, and irrelevant sections.

        - All text is converted into a normalized format for tokenization and embedding generation.
- E**mbedding Generation:**
    - Each data entry is converted into a high-dimensional vector using PubMed BERT, a language model optimized for biomedical text.
    - These embeddings represent the semantic meaning of the text, enabling efficient similarity searches.
- **Metadata Attachment:**
    - Metadata, such as source citations (e.g., Gale Encyclopedia, Medicine Dataset), is linked to each embedding to ensure traceability.
    - Contextual tags (e.g., "Disease," "Medication") are added for better organization and retrieval accuracy.

Vector Database Usage:

The system incorporates a **FAISS-based vector database** to handle the efficient storage and retrieval of embeddings, a critical component of the Retrieval-Augmented Generation (RAG) framework.

Why FAISS?

Facebook AI Similarity Search (FAISS) is a specialized library designed for similarity search and clustering of dense vectors. Key reasons for using FAISS include:

- **Speed**: Optimized for fast similarity searches, even with millions of vectors.
- **Scalability**: Supports large datasets with minimal memory and computational overhead.
- **Customizability**: Allows fine-tuning of search parameters to balance speed and accuracy.

Functions / tools used:

Category	Tool/Function	Purpose	Why Used
NLP Tools	PubMed BERT	Generates domain-specific embeddings and powers the QA pipeline.	Fine-tuned for biomedical text, ensuring contextual understanding and precise answers.
	Sentence Transformers	Encodes knowledge base and user queries into vector representations for semantic search.	Provides efficient and accurate embedding generation for similarity searches.

	Hugging Face Transformers	Provides access to pretrained models (PubMed BERT).	Facilitates seamless integration and fine-tuning of machine learning models.
	FAISS	Handles high-dimensional vector storage and similarity searches.	Optimized for large-scale data retrieval with fast query execution.
Frontend Tools	Streamlit	Provides the chatbot's user interface and manages user sessions.	Enables easy, interactive, and responsive web-based interfaces.
		- Sidebar and session management	Allows persistent user sessions and intuitive navigation for past conversations.
Backend Tools	SQLite	Stores user data, conversation history, and chat messages.	Lightweight and reliable database for small-scale projects.
	bcrypt	Hashes passwords for secure user authentication.	Ensures secure storage and comparison of sensitive user credentials.
Data Handling	pdfplumber	Extracts text content from PDFs.	Allows easy integration of structured data from the Gale Encyclopedia of Medicine.
	pandas	Processes and integrates CSV data for the knowledge base.	Efficiently manipulates tabular datasets for embedding and storage.
Authentication	hash_password	Hashes passwords securely before storing them in the database.	Protects user credentials against unauthorized access.
	check_password	Verifies user-entered passwords against stored hashes.	Ensures secure and accurate user login verification.

	create_user	Adds new users to the database with hashed passwords.	Supports the signup functionality.
	authenticate_user	Authenticates users by checking their credentials.	Supports secure user login.
Chatbot Features	answer_question	Combines semantic search and QA pipeline to produce answers.	Ensures contextually relevant and specific responses to user queries.
	Conversation Management	Tracks user-specific conversations and links them to stored chat histories.	Supports continuity and personalization of user interactions.
	Message History	Saves and retrieves individual messages with timestamps and sender details.	Allows users to revisit past interactions and improve system usability.

- **Data privacy, bias in AI models, and the broader societal impact of project:**

Data Privacy:

Importance: Data privacy is critical for ensuring user trust, especially in sensitive domains like healthcare.

- Measures Taken:
  - User Authentication: Secure authentication implemented using hashed passwords (bcrypt) and token-based sessions.
  - Data Storage:
    - User credentials and chat histories are stored securely in a relational database (SQLite).
    - Passwords are hashed before storage, ensuring sensitive information is not stored in plain text.
  - Session Management: Streamlit's session state ensures temporary session data does not persist beyond the current interaction.

Bias in AI Models:

Importance: In the medical domain, biases in AI models can lead to inaccurate responses or exclusionary practices, potentially affecting user trust and outcomes.

- Sources of Bias:

- Model Training Data:
  - The models (e.g., PubMed BERT) were trained on biomedical data, which may contain biases in how certain diseases, treatments, or populations are represented.

- Knowledge Base:
  - The Gale Encyclopedia of Medicine and drug dataset may reflect biases inherent in the source material, such as a lack of representation of certain demographics or conditions.

Reflection on how their project aligns with ethical AI practices and measures taken to mitigate potential ethical issues:

Alignment with Ethical AI Practices:
This project is designed to prioritize ethical AI principles, ensuring user safety, transparency, and equitable access to information. Below is a reflection on how the project aligns with ethical AI practices, along with measures taken to address potential issues:

- **User Privacy and Data Security**
  - **Alignment**:
    - The project adheres to data privacy principles by implementing secure authentication and storage mechanisms.
    - Passwords are hashed using bcrypt, ensuring sensitive user credentials are not stored in plaintext.
    - Conversations are linked to authenticated user accounts but stored securely to maintain privacy.
  - **Measures Taken**:
    - Use of token-based session management (Streamlit session state) to limit data persistence during interactions.
    - Restricting direct database access to authorized processes only, preventing unauthorized access to sensitive data.
    - Future deployment plans include implementing HTTPS to encrypt data transmission.
- **Transparency and Source Citation**
  - **Alignment**:
    - Responses generated by the chatbot include citations of the source material, allowing users to verify the information independently.
    - The knowledge base is constructed from credible sources like the **Gale Encyclopedia of Medicine** and a curated dataset of drug information, ensuring accuracy.
  - **Measures Taken**:
    - Attaching metadata (e.g., source details) to every piece of information stored in the vector database.
    - Displaying source citations in chatbot responses to enhance user trust and mitigate misinformation risks.
- **Mitigating Bias in AI Models**
  - **Alignment**:

        - Domain-specific models like **PubMed BERT** and fine-tuned QA pipelines are used to reduce generalization errors and improve accuracy in medical contexts.
        - Careful selection of datasets (e.g., Gale Encyclopedia and drug conditions dataset) ensures reliable coverage of medical topics.
    - **Measures Taken**:
        - Testing the chatbot against diverse queries, including conditions affecting underrepresented groups, to identify potential biases.
    - **Proposed Actions**:
        - Expand the dataset to include global medical knowledge for better inclusivity.

- **Broader Societal Impact**
    - **Alignment**:
        - The chatbot democratizes access to medical knowledge, enabling users to make informed decisions and improve health literacy.
        - Designed as a supplementary tool, not a replacement for professional healthcare, it includes disclaimers to prevent over-reliance.
    - **Measures Taken**:
        - Disclaimers explicitly warn users that the chatbot is not a substitute for medical consultation.
        - Feedback mechanisms are planned to allow users to report inaccuracies or misleading information.
        - Efforts to address the digital divide by optimizing the chatbot for lightweight deployment and mobile compatibility.

- **Proposed Additional Measures**
    - **Periodic Model and Dataset Review**:
        - Regularly evaluate the AI models and datasets for outdated or incomplete information.
        - Incorporate diverse datasets to address potential biases and improve accuracy.
    - **Feedback Loop for Continuous Improvement**:
        - Enable users to provide feedback on responses, allowing developers to address inaccuracies or gaps.
        - Use feedback to fine-tune retrieval and response generation processes.
    - **Data Minimization and Anonymization**:
        - Only store user data essential for providing the service.
        - Anonymize chat histories where possible to reduce the risk of data misuse.

Individual Section:

Deep Patoliya

What You Did (Estimated ~30%)

I played a key role in designing and implementing the query processing pipeline, integrating PubMed BERT for domain-specific embeddings and FAISS for efficient vector similarity searches. This ensured accurate retrieval of medical information. I also led data preprocessing, cleaning, normalizing, and embedding data from the Gale Encyclopedia of Medicine and the medicines dataset into high-dimensional vectors. Additionally, I contributed to system testing, validating chatbot responses against diverse medical queries to ensure robustness and accuracy. These contributions accounted for approximately 30% of the project's workload.

Challenges Faced During the Project

A major challenge was integrating PubMed BERT with the FAISS-based retrieval system, ensuring they worked seamlessly together. Another hurdle was maintaining embedding precision when dealing with noisy or inconsistent data from the raw datasets. Debugging issues in a pipeline with multiple interconnected components added further complexity.

How They Were Overcome

To address these challenges, I employed an incremental testing approach, embedding smaller subsets of data initially to isolate and resolve issues before scaling to the full dataset. Enhanced preprocessing techniques were used to filter irrelevant data, standardize formats, and improve embedding accuracy. Collaboration with team members was critical in overcoming bottlenecks, as their input helped identify and resolve issues effectively.

Lessons Learned About Working in a Team

This project highlighted the importance of effective communication within a team, especially when debugging across interconnected systems. I also learned to value delegation, trusting teammates to handle tasks independently while focusing on my responsibilities. Collaborative brainstorming and constructive feedback further improved the system's performance.

Application in Future Professional Settings

I plan to apply structured debugging techniques in future projects, testing components individually before integration. Transparent communication and proactive collaboration will remain priorities to ensure alignment and task clarity in cross-functional teams. The emphasis on thorough data preparation during this project has reinforced its importance, and I will carry these principles forward to tackle future challenges in data-intensive AI projects.

Dhruv Shah

What You Did (Estimated ~25%)

I contributed significantly to the project by designing and implementing the frontend interface using Streamlit, ensuring an intuitive and seamless user experience. This included creating a user-friendly environment for interactions and viewing past conversations. I also focused on establishing secure user authentication, implementing hashed password storage and managing session states to safeguard user data. Additionally, I handled database integration, managing user profiles, storing conversation histories, and linking these to backend processes. These tasks accounted for roughly 25% of the project workload, bridging the gap between backend operations and user-facing functionalities.

Challenges Faced During the Project

One key challenge was designing a responsive and intuitive user interface to accommodate diverse user needs while maintaining simplicity. Implementing secure yet lightweight authentication protocols without compromising performance also posed difficulties. Additionally, ensuring smooth integration between the frontend and backend required overcoming issues related to the interconnected nature of the system.

How They Were Overcome

To address UI challenges, I adopted an iterative development approach, regularly refining the design based on feedback from teammates and potential users. For authentication, I leveraged Streamlit's session state for efficient session management and bcrypt for secure password hashing. Collaboration with backend developers helped align both ends of the system, ensuring seamless data flow and functionality.

Lessons Learned About Working in a Team

This project underscored the importance of aligning frontend and backend development early to avoid potential misalignment. Regular communication with team members helped manage interdependencies effectively. I also learned to coordinate overlapping tasks by setting clear timelines and adapting to changing requirements with flexibility. Constructive feedback was vital for delivering a cohesive and user-friendly system.

Application in Future Professional Settings

I plan to continue using iterative design and user feedback loops to create adaptable and user-centric interfaces. The focus on security will remain a priority, ensuring robust authentication systems in future projects. Additionally, I intend to emphasize early collaboration between frontend and backend teams to streamline workflows and reduce inefficiencies. These lessons will enhance my ability to contribute effectively to complex projects in professional environments.

Vedant Kadam

What You Did (Estimated ~25%)

I focused on improving the Retrieval-Augmented Generation (RAG) framework, specifically optimizing retrieval accuracy and handling document context retrieval. My work included refining the BIOMISTRAL response generation system, fine-tuning it to handle ambiguous or insufficiently detailed queries effectively. This ensured the chatbot could deliver coherent and contextually accurate answers even with limited input. Additionally, I worked on attaching metadata to embeddings to enhance source transparency and allow users to trace the origins of the chatbot's responses. These tasks accounted for approximately 20% of the project workload and were critical to ensuring the system's reliability and user trust.

Challenges Faced During the Project

One significant challenge was fine-tuning BIOMISTRAL to generate coherent responses when retrieved documents lacked sufficient relevance or detail. Balancing performance and accuracy in similarity search algorithms for high-dimensional embeddings also proved complex, as it required maintaining speed without sacrificing precision.

How They Were Overcome

I implemented a fallback mechanism, triggering additional query expansion when retrieval results were of low relevance. This approach improved document retrieval for ambiguous queries. To address performance and accuracy tradeoffs, I tested various FAISS indexing techniques, ultimately selecting a configuration that provided an optimal balance between speed and precision. Iterative benchmarking and adjustments ensured both retrieval accuracy and response quality.

Lessons Learned About Working in a Team

This project emphasized the value of cross-functional collaboration, particularly in resolving complex issues spanning multiple components. Input from frontend and data processing teams proved crucial for identifying bottlenecks and implementing effective solutions. I also learned the importance of adaptability, as project requirements and task scopes often evolved during development, requiring flexibility to maintain progress.

Application in Future Professional Settings

In future projects, I will apply fallback mechanisms and iterative testing to handle ambiguity and enhance system robustness. This experience reinforced the importance of interdisciplinary collaboration, which I plan to encourage in professional settings for solving complex challenges more efficiently. Additionally, I will prioritize transparency and traceability in AI systems, ensuring that users can trust and verify outputs. These lessons have prepared me to contribute effectively to dynamic, team-oriented environments.

John Miller

What You Did (Estimated ~20%)

I led the development of the ethical AI framework, focusing on transparency, privacy, and bias mitigation in datasets and model behavior. This involved evaluating and refining data sources to ensure fair representation of diverse medical conditions. I conducted extensive testing of chatbot responses, emphasizing inclusivity and identifying gaps for underrepresented groups or conditions. Additionally, I developed the source citation system, linking embeddings to their metadata to provide concise, user-friendly references. These efforts accounted for about 20% of the project's workload and were crucial for ensuring the chatbot's credibility and ethical integrity.

Challenges Faced During the Project

A key challenge was identifying and addressing inherent biases in the Gale Encyclopedia of Medicine and the medicines dataset, which had limitations in representing underrepresented conditions or demographics. Another challenge was ensuring transparency by providing clear source citations without overwhelming users with overly technical details. Balancing accessibility and depth in chatbot explanations was particularly challenging.

How They Were Overcome

To mitigate bias, I conducted manual audits of the datasets and used feedback loops to refine and improve their inclusivity. Diverse query testing helped identify areas needing better representation. For transparency, I created a simplified source attribution system, displaying concise yet clear citations within chatbot responses. This ensured users could verify information without being burdened by excessive technical details.

Lessons Learned About Working in a Team

I learned the importance of balancing technical and ethical priorities through collaboration and regular feedback from teammates. Brainstorming sessions were essential for addressing challenges and aligning on shared goals. Additionally, I recognized the value of thorough documentation, which ensures that ethical considerations are traceable and reproducible. Working with diverse roles highlighted the need to integrate ethical practices at every stage of development.

Application in Future Professional Settings

This experience has underscored the importance of embedding ethical considerations as a core element in AI design. In future projects, I will advocate for user feedback mechanisms to enhance system fairness and inclusivity continually. I plan to prioritize transparent communication of system strengths and limitations, fostering user trust. These lessons will

guide me in creating AI solutions that are both technically robust and ethically responsible, ensuring positive societal impact.