Solution Requirement

Date	27 June 2025
Team ID	LTVIP2025TMID31381
Project Name	HealthAI
Marks	4 marks

2.2. Solution Requirements

2.2.1. Functional Requirements

- **User Authentication (Sign-in/Sign-up): (not added in present application)**
  - Users must be able to register a new account with a unique username and a password. Passwords will be hashed for storage.
  - Existing users must be able to log in with their credentials.
  - The application must maintain the user's logged-in state across interactions.
  - Only authenticated users can access personalized health functionalities.
- **Patient Profile Management:**
  - Logged-in users can create and add new patient profiles (Name, Age, Gender), linked to their user account.
  - Users can retrieve their associated patient profiles using the unique MongoDB ObjectId.
- **Symptoms Identifier (Scenario 1):**
  - Input: User's patient_id (selected or entered), free-text description of symptoms.
  - Process: The LLM analyzes the symptoms, optionally contextualized by the linked patient's profile.
  - Output: Potential conditions, their likelihood, and recommended next steps.
  - Persistence: Input symptoms and the AI's full prediction are stored in the patient's health records in MongoDB.
- **Home Remedies:**

Solution Requirement

- Input: Free-text disease name.
- Process: The LLM generates a list of safe and common natural home remedies.
- Output: Explanations for usage of each remedy.

➢ **Personalized Treatment Plans (Scenario 2):**

- Input: User's patient_id, diagnosed condition.
- Process: The LLM processes the condition, incorporating patient profile data to create a comprehensive, evidence-based general treatment plan.
- Output: Plan includes medication types (not specific drugs), lifestyle modifications, and follow-up testing suggestions.
- Persistence: The diagnosed condition and the generated plan are stored in the patient's health records.

➢ **Health Analytics Dashboard (Scenario 3):**

- Input: User's patient_id, free-text description of vital signs/health trends over time.
- Process: The LLM analyzes the textual data to identify potential health concerns and trends.
- Output: AI-generated insights and general improvement recommendations.
- Persistence: The vital signs description and the AI's analysis are stored in the patient's health records.

➢ **Patient Chat Interface (Scenario 4):**

- Input: Free-text medical queries.
- Process: The LLM engages in a conversational dialogue, maintaining context from previous messages.
- Output: Clear, empathetic responses with relevant medical facts, acknowledgement of AI limitations, and suggestions for professional medical advice.

2.2.2. Non-Functional Requirements

➢ **Security:**

- MongoDB connection credentials are secured via environment variables.

Solution Requirement

- User passwords are hashed using SHA256 before storage in MongoDB.
- Patient data is linked to the user account, ensuring access control (a patient's records can only be retrieved by the user_id who created them).
- All AI-generated responses include prominent disclaimers regarding their informational nature and the necessity of consulting a healthcare professional.

➢ **Performance:**

- LLM inference time should be minimized for a responsive user experience (leveraging GPU acceleration if available, e.g., via CUDA setup locally or cloudVM).
- Database operations (reads/writes) should be efficient. @st.cache_resource is used to optimize MongoDB connection and model loading, preventing repeated expensive operations.

➢ **Scalability:**

- MongoDB inherently supports horizontal scaling for data storage.
- Streamlit applications can be scaled by deploying on more powerful machines or using services like Streamlit Community Cloud for managed scaling.
- LLM inference scaling would involve optimized deployment on dedicated inference hardware or using model serving solutions.

➢ **Usability:**

- Streamlit's straightforward UI components provide an intuitive and easy-to-navigate interface.
- Clear instructions, input fields, and output areas are provided.
- Login/Sign-up flow is clearly separated, guiding the user through authentication first.

➢ **Reliability:**

- Robust error handling (try-except blocks) is implemented for database connection issues, invalid ObjectId formats, and LLM loading/inference failures, providing informative messages to the user.

Solution Requirement

- Input validation is performed to guide users on correct data entry.

- **Maintainability:**
  - The codebase is modular, with separate functions for database interactions, LLM calls, and UI rendering, improving readability and future modifications.
- **Ethical Considerations:**
  - The application explicitly and repeatedly disclaims that it is not a substitute for professional medical advice, mitigating risks of misinterpretation.
  - Privacy is considered through user authentication and data linking, though full HIPAA/GDPR compliance would require further enterprise-level security measures and legal consultation.