Dataflow Diagram

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

2.3. Dataflow Diagram

The HealthAI application's data flow depicts the interaction between its userfacing components, backend logic, LLM, and database, including the authentication process. graph TD

A[User (Web Browser)] -- Interaction --> B(Streamlit User Interface)

B -- 1. Sign-up/Login Credentials --> C{Python Application Backend (app.py)}

C -- 2. Authenticates/Registers --> D[MongoDB Database]

D -- 3. User Account Data (Hashed Passwords, User IDs) --> C

C -- 4. User Session State (Logged In, User ID) --> B

B -- 5. Health Queries (Symptoms, Disease, Vitals, Chat Message) --> C

B -- 6. Patient Data Input/Request (Name, Age, Gender, Patient ID) --> C

C -- 7. Retrieves Patient Profile (Linked to User ID) --> D

D -- 8. Patient Profile Data --> C

C -- 9. Formats LLM Prompt (with optional Patient Profile) --> E[Hugging Face Transformers Pipeline]

E -- 10. LLM Inference Request --> F(IBM Granite model)

F -- 11. Utilizes --> G(GPU/CPU Resources)

F -- 12. Generated Response (Text) --> E

E -- 13. LLM Response --> C

C -- 14. Stores Health Records (Input + LLM Response) --> D

D -- 15. Health Records Data --> C

C -- 16. Displays Results/Chat --> B

B -- 17. Presents to User --> A

Dataflow Diagram

Detailed Data Flow Description:

1. User Authentication:

- User (A) -> Streamlit UI (B): User inputs username and password for sign-up or login. Streamlit UI (B) -> Python Backend (C): Streamlit collects credentials and calls register_user or login_user function.
- Python Backend (C) -> MongoDB (D): The backend queries the users collection to check for existing usernames or verify credentials. Passwords are hashed using SHA256 before storage.
- MongoDB (D) -> Python Backend (C): Returns user account data (including _id for the user).
- Python Backend (C) -> Streamlit UI (B): On successful login, the st.session_state is updated to reflect logged_in=True, username, and the user_id (MongoDB _id of the user document). This triggers a UI re-render, revealing the main application functionalities.

2. HealthAI Functionalities (Post-Login):

- User (A) -> Streamlit UI (B): Once logged in, the user navigates through different sections (tabs/radios) and inputs data for Symptoms, Home Remedies, Treatment Plans, Health Analytics, or Patient Chat.
- Streamlit UI (B) -> Python Backend (C): Input data, including the current user_id from st.session_state and potentially a patient_id (if applicable), is sent to the corresponding backend function in app.py.
- Python Backend (C) -> MongoDB (D) (Patient Data Retrieval): For personalized features (Symptoms, Treatment Plans, Health Analytics), the backend retrieves the relevant patient's demographic data from the patients collection, crucially ensuring it belongs to the user_id of the logged-in user.
- MongoDB (D) -> Python Backend (C): Returns the patient's profile data.
- Python Backend (C) -> Hugging Face Pipeline (E): The backend dynamically constructs a detailed prompt for the LLM,

Dataflow Diagram

incorporating user inputs and, for personalized features, the retrieved patient's profile.

- Hugging Face Pipeline (E) -> IBM Granite model (F): The prepared promptis fed to the IBM Granite model.
- IBM Granite model (F) -> GPU/CPU Resources (G): The model performs its complex generative computations using the available hardware resources (ideally a GPU).
- IBM Granite model (F) -> Hugging Face Pipeline (E): The model returns the generated text response.
- Hugging Face Pipeline (E) -> Python Backend (C): The generated text is received by the app.py backend.
- Python Backend (C) -> MongoDB (D) (Health Record Storage): For Symptoms, Treatment Plans, and Health Analytics, the user's input and the AI's response are stored as a new document in the health_records collection, meticulously linked to the specific patient_id (and implicitly to the user_id via the patient).
- Python Backend (C) -> Streamlit UI (B): The processed AI response (and any status messages) is sent back to the Streamlit UI.

Dataflow Diagram

- Streamlit UI (B) -> User (A): The results are displayed to the user in the appropriate section of the we

Project Cycle Diagram