
CAPSTONE PROJECT

AGENTIC AI HEALTH SYMPTOM CHECKER

Presented By:
JAHNAVI PENMETHSA
St. Martin's Engineering College
Computer Science and Engineering (AI & ML)

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach**
- **Algorithm & Deployment**
- **Result**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

With the rapid adoption of AI in healthcare, there is a growing need for accessible, accurate, and reliable health information systems that empower individuals to make informed decisions without promoting self-diagnosis.

Millions of people, especially in rural and underserved areas, lack immediate access to healthcare professionals and often rely on unverified online sources, leading to misinformation and delayed treatment.

The challenge is to develop an **Agentic AI Health Symptom Checker** using **IBM Cloud Lite** and **IBM Granite LLM** with a RAG pipeline to provide **safe, multilingual, and verified** health guidance, promoting early awareness and timely medical consultation.

PROPOSED SOLUTION

- The proposed system is an **Agentic AI Health Symptom Checker** that allows users to describe their symptoms in natural language and receive educational, non-diagnostic health information.
- It uses RAG (Retrieval-Augmented Generation) to fetch trusted content from WHO, Government Health Portals, and verified medical databases.
- Supports multilingual conversations for inclusivity.
- Provides:
 - Probable conditions (with disclaimers)
 - Preventive advice
 - Urgency level
 - When to consult a doctor
 - Home remedies (if applicable)
- Powered by IBM Granite LLM for natural conversation.
- Hosted and deployed on IBM Cloud Lite Services for free-tier accessibility.

SYSTEM APPROACH

System Requirements

- a. Hardware Requirements
 - **Processor:** Intel i5/i7 or AMD equivalent
 - **RAM:** Minimum 8 GB (Recommended 16 GB)
 - **Storage:** 20 GB free space

- b. Software Requirements
 - **Operating System:** Windows 10/11, macOS, or Linux
 - **IBM Cloud Lite Account** with access to:
 - IBM Granite LLM
 - IBM Watson Discovery
 - IBM Watson Assistant
 - IBM Cloud Object Storage
 - **Web Browser:** Chrome / Edge / Firefox

SYSTEM APPROACH

Libraries / Tools Required

- **Watson Discovery**
 - Used for RAG (Retrieval-Augmented Generation) by ingesting verified health documents.
 - Provides search and retrieval capabilities via natural language queries.
- **Watson Assistant**
 - Creates the conversational interface (chatbot) for users.
 - Manages intents, entities, and dialog flows.
- **IBM Granite LLM**
 - Generates contextual and natural-sounding responses.
 - Works with retrieved data to provide safe educational guidance.
- **Cloud Object Storage**
 - Stores source documents and datasets.
 - Used by Watson Discovery for ingestion.
- **Language Translator Service (Optional)**
 - Adds multilingual support to handle queries in multiple languages.

ALGORITHM & DEPLOYMENT

Algorithm Steps:

- **Input Acquisition** – User describes symptoms via chatbot interface.
- **Language Processing** – Detect language → Translate to English if required.
- **Information Retrieval (RAG)**
 - Query sent to IBM Watson Discovery with vector similarity search.
 - Retrieve relevant documents from WHO / government health portals.
- **LLM Processing (Granite)**
 - Retrieved context is fed to Granite for summarization and generation.
 - Ensure medical disclaimers are appended automatically.
- **Urgency Detection** – Simple symptom severity scoring using predefined rules (e.g., high fever + chest pain = urgent).
- **Output Generation** – Conditions, preventive advice, home remedies, referral recommendations.
- **Output Delivery** – Return to frontend in multilingual format.

ALGORITHM & DEPLOYMENT

Deployment Steps

- Create **Watson Discovery** instance → upload verified datasets.
- Create **Watson Assistant** → integrate with Granite API.
- Host Flask backend on **IBM Cloud Code Engine**.
- Serve frontend from **IBM Cloud Object Storage (Static Website)**.
- Connect Watson Assistant to frontend via WebSocket or REST API.

RESULT

IBM watsonx

?

🔔

Jahnavi Penmethsa's Accou... ▾

Dallas ▾

JP

Create a project

Start with a new, blank project or select from where to import an existing project.

+ New

📁 Local file

📄 Sample

Define details

Name

MediQuery

Description (optional)

MediQuery is an AI-powered health symptom checker that provides safe, multilingual, and verified medical guidance using IBM Cloud's Watson Discovery, Granite LLM, and RAG technology. It empowers users with early awareness, preventive advice, and referral recommendations, reducing misinformation and improving healthcare accessibility.

Tags (optional)

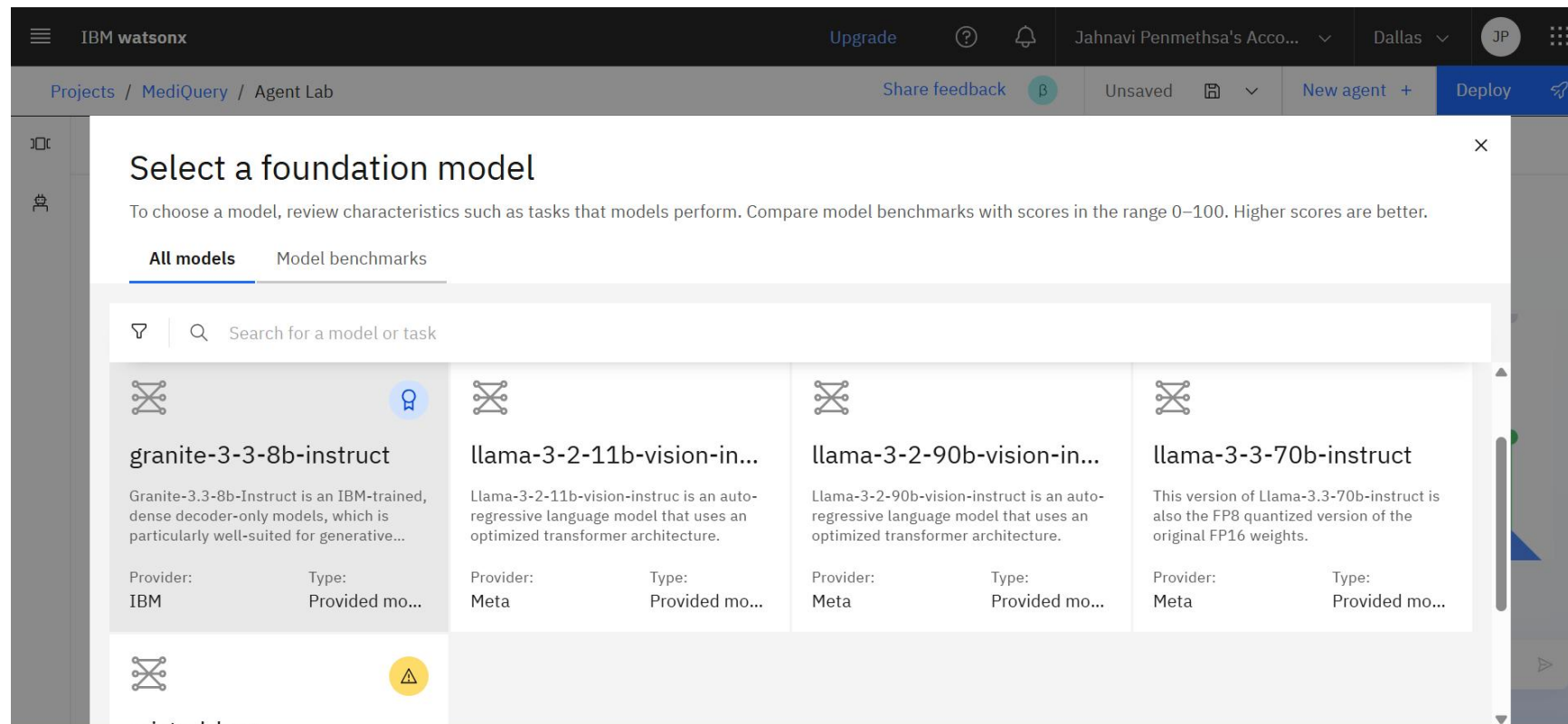
Cancel

Create

Project Definition and Overview

Introduces the project title “MediQuery” and outlines its purpose, scope, and objectives

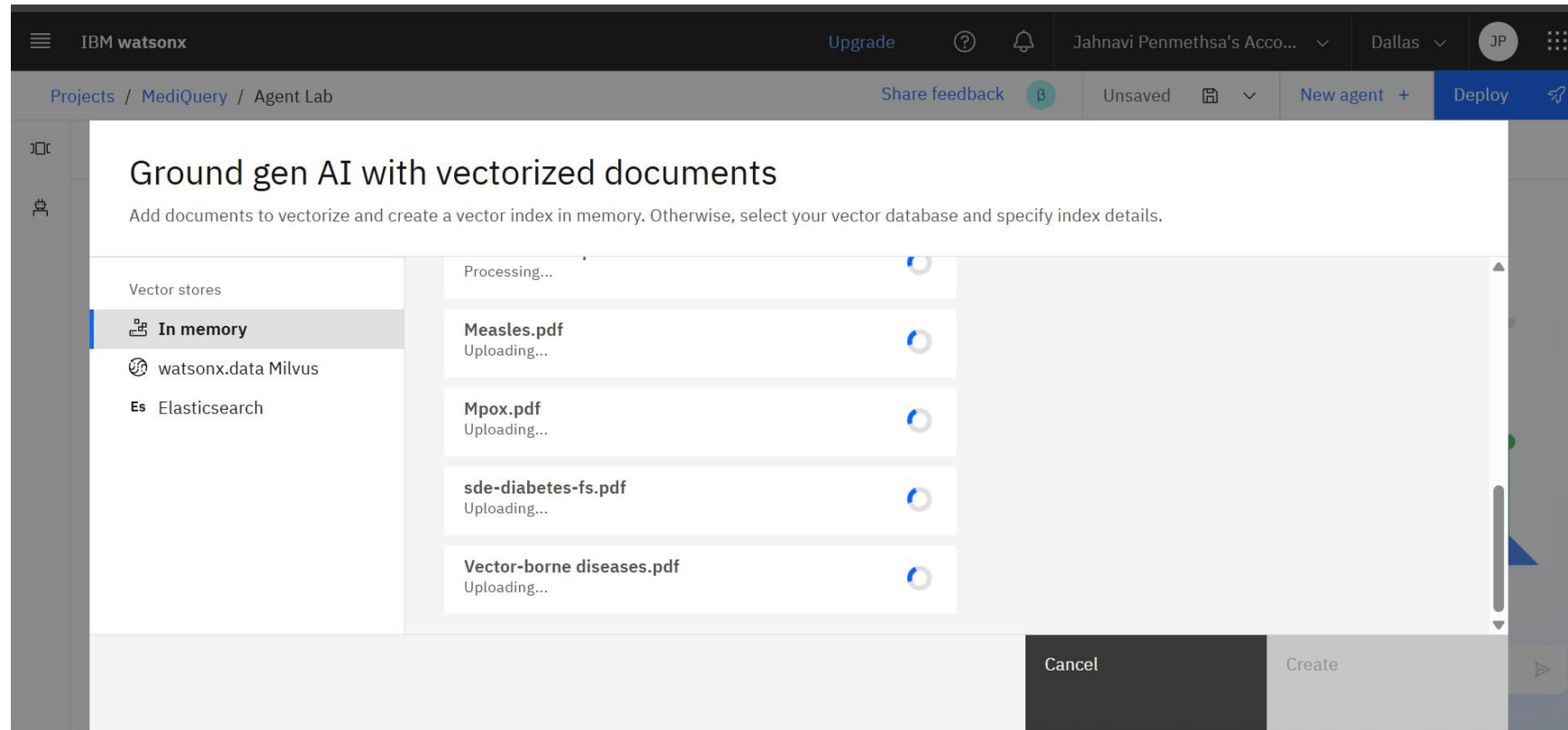
RESULT



Foundation Model Selection

Selection of IBM Granite model granite-3-3-8b-instruct to power the conversational AI capabilities

RESULT



Knowledge Base Preparation

Uploading verified WHO and government health documents to enable Retrieval-Augmented Generation

RESULT

The screenshot displays the IBM watsonx Agent Lab interface. The top navigation bar includes the IBM watsonx logo, an 'Upgrade' button, a help icon, a notification bell, and user information for 'Jahnvi Penmethsa's Acco...' with a location dropdown set to 'Dallas'. The breadcrumb trail shows 'Projects / MediQuery / Agent Lab'. The main interface is split into two panels. The left panel, titled 'Build', contains sections for 'Setup', 'Configuration', and 'Knowledge'. Under 'Configuration', the 'Framework' is set to 'LangGraph' and the 'Architecture' is set to 'ReAct'. The 'Instructions' section is highlighted with a blue border and contains the text: 'You are MediQuery, an AI-powered health assistant. Use verified medical PDFs (WHO and government sources) as your primary knowledge base, and retrieve additional information from trusted sites in real time when needed. Provide clear, multilingual, educational health guidance on symptoms, possible conditions, treatments, and prevention. Do not give...'. A link for 'Advanced configuration' is visible. The right panel, titled 'Agent preview', shows a chat window for 'MediQuery Agent 05:10 PM' with a welcome message: 'Welcome to MediQuery Agent. MediQuery – Trusted health answers from verified sources, in real time.' Below the message is a flowchart diagram and a text input field with the placeholder 'Type something...'.

Agent Configuration

Defining operational instructions for the MediQuery agent to ensure accurate, and educational responses

RESULT

The screenshot displays the IBM watsonx Agent Lab interface. The top navigation bar includes the IBM watsonx logo, an 'Upgrade' button, a help icon, a notification bell, and a user profile for 'Jahnvi Penmethsa's Acco...'. The breadcrumb trail shows 'Projects / MediQuery / Agent Lab'. The main interface has a 'Build' tab and an 'Agent preview' tab. The chat window shows a conversation where the user asks, 'I have mild fever and body aches, should I be worried?'. The MediQuery Agent responds with a detailed medical explanation, advising that mild fever and body aches can be due to common causes like the flu or stress, but also could indicate more serious conditions like dengue or COVID-19. The agent advises not to panic, to monitor symptoms, and to seek medical attention if they worsen or are accompanied by severe symptoms like difficulty breathing or chest pain. The chat window also includes a text input field with the placeholder 'Type something...' and a send button.

Initial Agent Testing

Evaluating the agent's basic performance by providing a sample health-related prompt

RESULT

The screenshot displays the IBM watsonx interface. The top navigation bar includes the 'IBM watsonx' logo, an 'Upgrade' button, a help icon, a notification bell, and a user profile dropdown for 'Jahnvi Penmethsa's Acco...'. The main header shows the project path 'Projects / MediQuery / Agent Lab' and buttons for 'Share feedback', 'Unsaved', 'New agent +', and 'Deploy'. The left sidebar is titled 'Build' and contains sections for 'Setup', 'Configuration', 'Knowledge', and 'Vector index'. The 'Vector index' section is expanded, showing a list of vector indices: 'Diarrhoeal disease', 'Ebola disease', 'Mpox', 'diabetes' (highlighted with a blue border), and 'Vector-borne diseases'. The right pane, titled 'Agent preview', shows a chat interface for the 'MediQuery Agent' with a timestamp of '05:23 PM'. The chat message reads: 'Welcome to MediQuery Agent' followed by 'MediQuery – Trusted health answers from verified sources, in real time.' Below the message is a diagram of a network graph with a magnifying glass icon over a green node. At the bottom of the chat interface is a text input field labeled 'Type something...' with a send button.

RAG Implementation

Integrating RAG by linking one of the uploaded documents from the vector index for contextual retrieval

RESULT

The screenshot displays the IBM watsonx Agent Lab interface. The top navigation bar includes the IBM watsonx logo, an 'Upgrade' button, a help icon, a notification bell, and user information for 'Jahnavi Penmethsa's Acco...' with a location of 'Dallas'. Below this, a breadcrumb trail shows 'Projects / MediQuery / Agent Lab', along with 'Share feedback', a beta badge, 'Unsaved', and buttons for 'New agent +' and 'Deploy'.

The main interface is split into two panels. The left panel, titled 'Build', contains sections for 'Setup', 'Configuration', 'Knowledge', and 'Tools'. Under 'Knowledge', a 'Vector index' section shows a dropdown menu with 'diabetes' selected. A note below states: 'If you want to add a vector index to the agent, clear the agent preview chat.' The right panel, titled 'Agent preview', shows a chat history. The user's query is 'how can i differentiate type-1 diabetes from type-2?'. The agent's response explains the complexity of differentiating Type 1 and Type 2 diabetes and lists three key differences: 1. **Age of Onset**: Type 1 develops in childhood or adolescence, while Type 2 typically manifests in adults. 2. **Body Mass Index (BMI)**: Type 2 is often associated with overweight or obesity, while Type 1 is not as common. 3. **Insulin Production**: In Type 1, the body produces little or no insulin. A text input field at the bottom of the chat area contains the placeholder text 'Type something...'.

RAG Query Testing

Assessing the RAG pipeline's effectiveness by submitting a test query and reviewing the retrieved context

RESULT

The screenshot displays the IBM watsonx Agent Lab interface. The top navigation bar includes the IBM watsonx logo, an 'Upgrade' button, a help icon, a notification bell, and user information for 'Jahnavi Penmethsa's Acco...'. The main breadcrumb trail shows 'Projects / MediQuery / Agent Lab'. The left sidebar contains a 'Build' tab and a list of sections: Setup, Configuration, Knowledge, Vector index, and Tools. The 'Vector index' section is expanded, showing a dropdown menu with 'diabetes' selected. Below this, a note states: 'If you want to add a vector index to the agent, clear the agent preview chat.' The right pane, titled 'Agent preview', displays a list of retrieved context items:

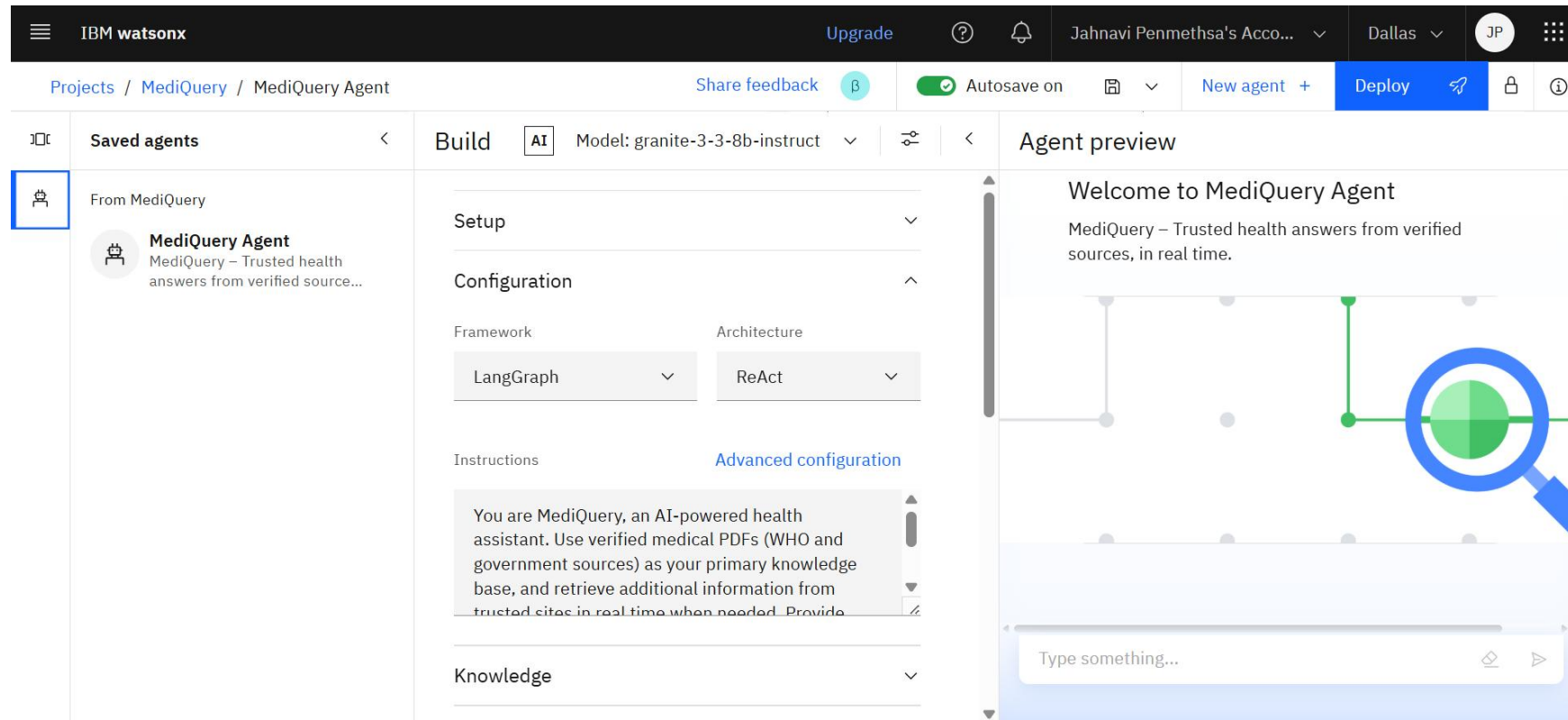
- 3. **Insulin Production:** In Type 1 diabetes, the body produces little or no insulin. In Type 2 diabetes, the body either resists the effects of insulin or doesn't produce enough insulin to maintain healthy blood glucose levels.
- 4. **Symptoms Onset:** In Type 1 diabetes, symptoms usually develop over a short period, such as weeks. In Type 2 diabetes, symptoms may change gradually and can be less noticeable.
- 5. **Family History:** Type 2 diabetes tends to run in families, suggesting a genetic component. While Type 1 diabetes also has a genetic link, it's not as strong as with Type 2.

Below the list, a disclaimer reads: 'Please note that these are general guidelines and exceptions exist. For an accurate diagnosis, consult with a healthcare professional. They can conduct tests to measure blood glucose levels, assess insulin production, and determine the type of diabetes.' At the bottom of the preview pane is a text input field with the placeholder 'Type something...' and a send button.

RAG Query Testing

Assessing the RAG pipeline's effectiveness by submitting a test query and reviewing the retrieved context

RESULT



Agent Deployment

Saving and finalizing the MediQuery agent for continued use and future enhancements

CONCLUSION

- The **Agentic AI Health Symptom Checker** demonstrates how AI + RAG can make reliable health education accessible to all.
- By retrieving knowledge from verified medical sources, the system promotes early awareness, reduces the spread of misinformation, and enhances accessibility for users across diverse backgrounds.
- While challenges such as maintaining accuracy in multilingual responses and ensuring strict adherence to credible sources were encountered, the solution proves effective in bridging the healthcare information gap.
- With future enhancements like voice-enabled interaction, offline functionality, and integration with wearable devices, the system holds strong potential to further empower communities with timely and trustworthy health guidance.

FUTURE SCOPE

- Integration with wearable devices (Fitbit, Apple Watch) for real-time symptom monitoring.
- Expansion to include mental health awareness modules.
- Voice-based interaction for low-literacy populations.
- Offline mobile app for rural areas with limited internet.
- Integration with telemedicine platforms for direct doctor consultation.
- AI-powered risk prediction models using medical history.

REFERENCES

- World Health Organization (WHO) – *Fact Sheets on Diseases and Conditions*
 - <https://www.who.int/news-room/fact-sheets>
- Indian Council of Medical Research (ICMR) – *Public Health Information*
 - <https://www.icmr.gov.in>
- IBM Granite LLM Documentation
 - <https://www.ibm.com/granite>
- IBM Watson Discovery Documentation
 - <https://cloud.ibm.com/docs/discovery>

IBM CERTIFICATIONS



IBM CERTIFICATIONS

In recognition of the commitment to achieve
professional excellence



Jahnavi Penmethsa

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution



Issued on: Jul 21, 2025
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/c9510ec9-4416-4029-a7fe-49b5040473c3>



IBM CERTIFICATIONS

IBM **SkillsBuild**

Completion Certificate



This certificate is presented to
Jahnavi Penmethsa

for the completion of

**Lab: Retrieval Augmented Generation with
LangChain**

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 23 Jul 2025 (GMT)

Learning hours: 20 mins



THANK YOU