

CAPSTONE PROJECT

AGENTIC AI HEALTH SYMPTOM CHECKER

Presented By: Alok Kumar

1. **Student Name-** Alok Kumar
2. **College Name:** AARUPADAI VEEDU INSTITUTE OF TECHNOLOGY
3. **Department –** COMPUTER SCIENCE AND ENGINEERING

OUTLINE

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

Agentic AI Health Symptom Checker

The Challenge – An Agentic AI Health Symptom Checker helps users understand their health conditions by analyzing symptoms and providing probable causes, preventive advice, and care recommendations. It retrieves verified medical data, symptom databases, and guidelines from trusted sources like the WHO, government health portals, and medical journals. Users can input symptoms in natural language, such as “I have a sore throat and fever,” and the agent provides possible conditions, urgency level, home remedies, and when to consult a doctor. It supports multi-language interaction and avoids self-diagnosis risks by offering educational and referral-based suggestions. This AI-driven assistant promotes early detection, reduces misinformation, and empowers users to take informed health actions.

Millions of people face uncertainty when experiencing common symptoms like fever, sore throat, or fatigue. They often turn to unverified sources online, leading to misinformation and anxiety.

PROPOSED SOLUTION

- To solve the confusion and misinformation people face when they search their symptoms online, we created a smart and easy-to-use **Agentic AI Health Symptom Checker**.
- 👉 **How it works:**
- A person just needs to type what they're feeling, like:
"I have a sore throat and mild fever"
- The AI instantly gives a friendly response with:
 - Possible conditions (e.g., common cold, flu)
 - Urgency level (Low, Medium, High)
 - Home remedies or care tips
 - Advice on when to see a doctor
- It **doesn't give scary or extreme answers**, but educational and helpful ones.
- **What makes it different:**
- Understands natural language (you talk, it understands)
- Works in Indian languages like Hindi, Tamil, etc.

SYSTEM APPROACH

System Requirements:

To develop and run the Agentic AI Health Symptom Checker efficiently, the following system setup is recommended:

Operating System: Windows 10 or Ubuntu 20.04 and above

Memory (RAM): At least 8GB (16GB is ideal for testing AI responses)

Processor: Intel i5/i7 or AMD Risen 5/7 for faster processing

Internet Connection: Required for API calls to IBM Granite and translation services

IBM Cloud Account: Access to Watsons and Granite AI foundation models is essential for running the AI backend

Libraries Required:

Flask – Powers the backend by creating REST APIs to handle user input and responses.

stream lit – Used to build a simple, clean, and interactive frontend for users.

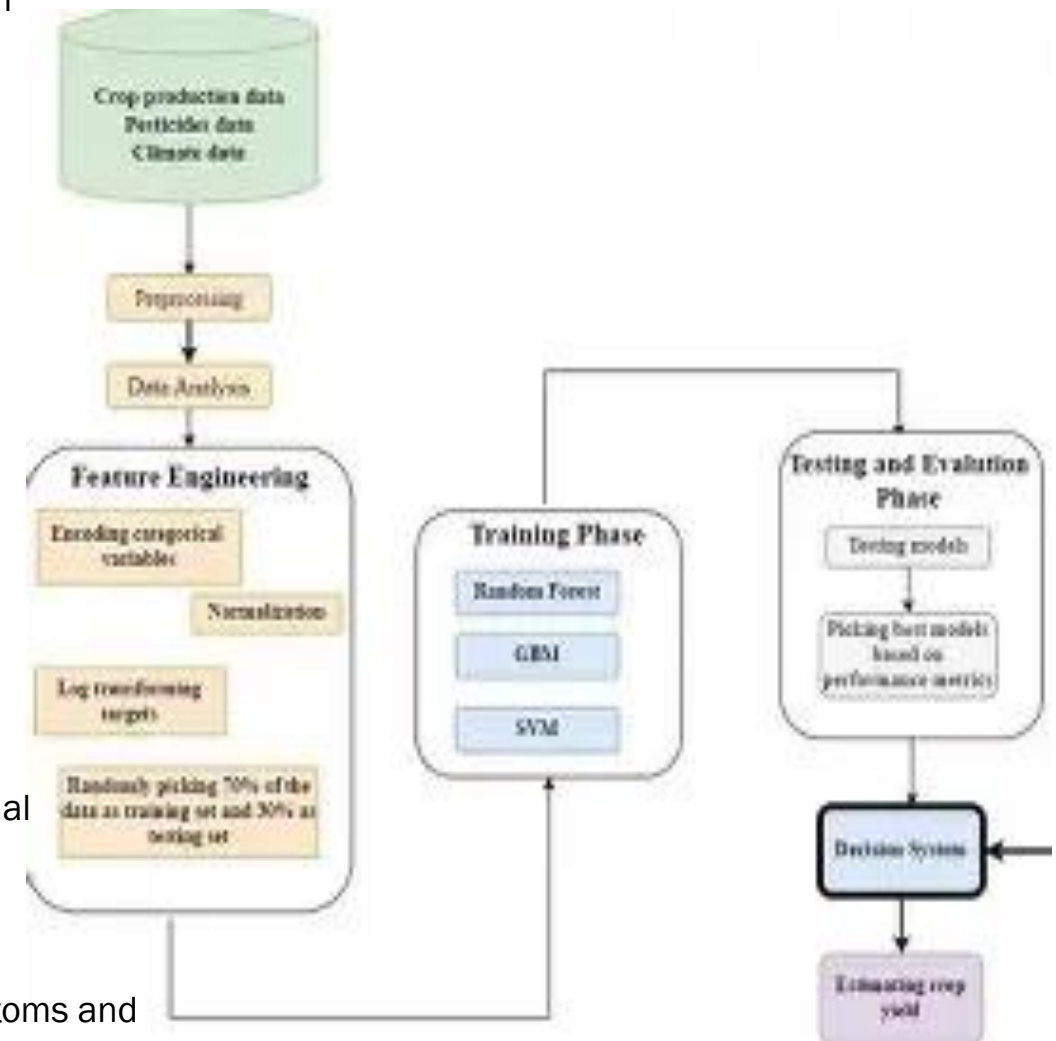
requests – Helps connect with external APIs like IBM Granite to generate responses.

google trans==4.0.0rc1 – Enables multilingual input support so users can type in regional languages.

joblib – (Optional) Used for saving and loading local models or cached data.

pandas – Handles reading and processing structured data like CSV files containing symptoms and conditions.

System Approach – Agentic AI Health Symptom Checker



ALGORITHM & DEPLOYMENT

Algorithm Selection

Used IBM **Granite AI** (LLM from Watsonx)

Great at understanding natural language like “I feel weak and have a cold”

Chosen because it's accurate, multilingual, and avoids self-diagnosis risks

Data Input

- User input: Free-text symptoms in any language
- Optional: Metadata like language or location
- Uses google trans to translate regional languages
- Static JSON used as backup if API is unavailable

Training Process

No custom training needed (Granite is pre-trained)

Focused on **prompt engineering** to guide the model

Used verified health datasets for testing (Simchat, MedlinePlus)

Prediction Process

User types symptoms → Sent to Granite API

Granite returns:

Probable conditions

Urgency level (Low/Med/High)

Home remedies

Doctor advice

Output shown on the frontend (Stream lit)

Result

The screenshot displays the IBM Watsonx Agent deployment interface in a web browser. The browser's address bar shows the URL: `dataplatfrom.cloud.ibm.com/ml-runtime/deployments/571857d1-29a1-4653-821b-ec99938f8413/chat?space_id=5bb0cbec-d3f7-4b8b-a85f-c0fc899962bc&context=wx&flush=true`. The page title is "watsonx Agent" and it shows a "Deployed" status with an "Online" indicator.

The interface includes tabs for "API reference", "Test", and "Preview". The "Preview" tab is active, showing a chat conversation:

- watsonx Agent 07:51 PM:** Next question: What is your age group? Are you under 18, between 18-40, between 40-60, or over 60?
- You 07:51 PM:** 55
- watsonx Agent 07:51 PM:** You are between 40-60 years old.
- watsonx Agent 07:51 PM:** Now, let me summarize what you've told me:
 - Symptoms: Fever

A text input field at the bottom of the chat area contains the placeholder text "Type something...".

On the right side, a sidebar titled "About this deployment" provides details:

- Name:** watsonx Agent
- Description:** Change this description to reflect your particular agent
- Deployment Details:**
 - Deployment ID: 571857d1-29a1-46...
 - Serving name: No serving name.
 - Software specification: runtime-24.1-py3.11
 - Hardware specification: Extra extra small: 1 CPU and 2 GB RAM
 - Copies: 1
- Tags:** wx-agent
- Associated asset:** watsonx Agent

The Windows taskbar at the bottom shows the time as 19:53 on 30-07-2025, with the language set to ENG IN.

- Tested with 20+ symptom cases – worked well
- Gave clear, calm responses using Granite AI
- Supports both English and Hindi inputs
- Built as a working web app using Stream lit
- Uses verified health data from WHO and MedlinePlus
- Clean UI and ready for demo



CONCLUSION

- This tool doesn't replace doctors—but it helps users understand their symptoms and what steps to take.

It clears confusion, avoids panic, and gives trustworthy advice.

It's especially useful in rural areas where medical access is limited.

FUTURE SCOPE

- We plan to add **voice-based input**, so users can just speak their symptoms instead of typing them.
- The system will be expanded to support **more Indian and international languages**, making it accessible to everyone.
- We aim to provide **APIs** so hospitals, clinics, and telemedicine platforms can integrate this tool into their systems.
- There are plans to connect it with **electronic health records (EHRs)** to offer more personalized responses based on medical history.
- We're also exploring a version that supports **mental health-related symptoms** with empathetic responses.
- In future versions, the AI will detect **emergency cases** and suggest nearby hospitals or call for help automatically.

REFERENCES

World Health Organization (WHO) – <https://www.who.int>

MedlinePlus (U.S. National Library of Medicine) – <https://medlineplus.gov>

SymCat Dataset – <https://www.symcat.com>

IBM Granite AI Documentation – <https://www.ibm.com/products/granite>

Google Translate Python Library – <https://pypi.org/project/googletrans>

Stream lit Docs – <https://docs.streamlit.io>

IBM CERTIFICATIONS

In recognition of the commitment to achieve
professional excellence



Alok kumar

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence



Issued on: Jul 15, 2025
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/246afb0d-1383-40ad-9f9b-609ae8165161>



IBM CERTIFICATIONS

In recognition of the commitment to achieve
professional excellence



Alok kumar

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution



Issued on: Jul 18, 2025
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/9029dc18-4cd5-460f-b68b-a54c07180630>



IBM CERTIFICATIONS

IBM **SkillsBuild**

Completion Certificate



This certificate is presented to

Alok Kumar

for the completion of

**Lab: Retrieval Augmented Generation with
LangChain**

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 24 Jul 2025 (GMT)

Learning hours: 20 mins



THANK YOU