PROJECT REPORT

Rule-Based Medical Expert System

Intern Name: Syed Muneeb Haider

Internship Company: Syntecxhub

Domain: Artificial Intelligence (AI)

Project Type: Web-Based Expert

Rule-Based Medical Expert System

1. Introduction

This project was developed as part of my internship at **Syntacxhub**. The purpose of this project is to design and implement a **Rule-Based Expert System** that simulates human expert decision-making using predefined rules and logical inference.

The system focuses on medical diagnosis based on user-provided symptoms. It demonstrates how expert systems can assist in decision support by applying **if–then rules**, a **facts base**, and **forward chaining inference**.

This project helped me strengthen my understanding of Python, logical reasoning, backend development, and professional software structuring in an industry environment.

2. Objectives of the Project

The main objectives of this project are:

- To design a simple but effective **rule-based inference engine**
- To accept **user-provided facts (symptoms)** through a web interface
- To infer conclusions using **forward chaining**
- To support **multi-step inference**
- To display **reasoning steps** for transparency
- To follow **clean, professional, and industry-level coding practices**

3. Problem Statement

Manual diagnosis based on symptoms requires experience and structured reasoning. The goal of this project is to simulate that reasoning process using a computer-based expert system that:

- Applies predefined medical rules
- Matches user symptoms with those rules
- Produces a diagnosis based on logical inference
- Explains how the conclusion was reached

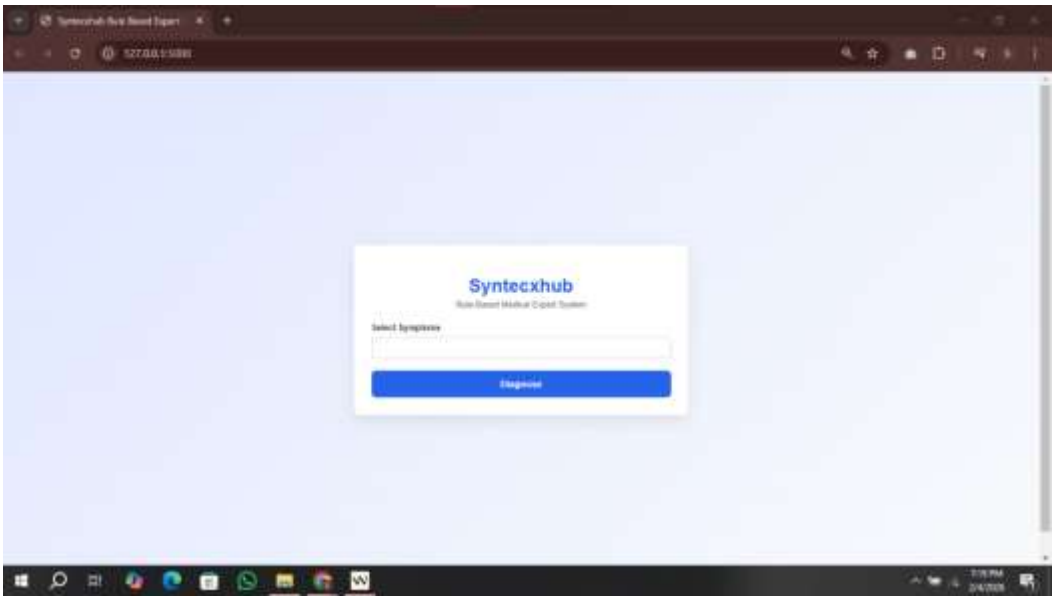
4. System Architecture Overview

The system follows a simple expert system architecture:

1. **Knowledge Base** – Stores medical rules

2. **Facts Base** – Stores user-selected symptoms
3. **Inference Engine** – Applies forward chaining
4. **Explanation Module** – Logs reasoning steps
5. **User Interface** – Web-based input/output

Interface:



5. Knowledge Base Design

The knowledge base consists of a collection of rules. Each rule represents a medical condition.

Rule Structure

- **IF** a set of symptoms is present
- **THEN** infer a specific disease

Example (conceptual):

IF fever AND fatigue AND body aches
THEN Flu

Each rule is stored as a dictionary containing:

- Disease name
- List of associated symptoms

6. Facts Base

The facts base consists of symptoms selected by the user through the web interface. These symptoms are treated as known facts and passed to the inference engine for reasoning.

The system dynamically compares these facts with the rules in the knowledge base.

7. Inference Engine (Forward Chaining)

The system uses **forward chaining**, which means:

- Reasoning starts from known facts (user symptoms)
- Rules are evaluated one by one
- Matching symptoms are counted
- Conclusions are inferred progressively

Multi-Step Inference

The inference process includes:

1. Matching user symptoms with each rule
2. Counting the number of matching symptoms
3. Evaluating all rules
4. Selecting exact or partial matches
5. Determining confidence level (severity)

This ensures the system does not jump directly to a conclusion but follows a logical reasoning path.

8. Explanation and Reasoning Logs

One of the key features of an expert system is transparency.

This system logs each inference step, such as:

- Which rule was checked
- How many symptoms matched

These logs are displayed to the user, allowing them to understand **why** a particular diagnosis was suggested.

9. User Interface Design

The user interface is designed to be:

- Simple
- Clean

- Professional
- Easy to understand

UI Features

- Multi-select symptom input
- Submit button for diagnosis
- Diagnosis display section
- Reasoning log section

The UI intentionally avoids complexity to keep focus on expert system logic rather than visual effects.

10. Technologies Used

Backend

- **Python** – Core logic and inference engine
- **Flask** – Web framework for routing and rendering

Frontend

- **HTML** – Structure
- **CSS** – Styling and layout
- **Jinja2** – Template rendering

11. Features Implemented

- Rule-based reasoning
- Facts-based inference
- Forward chaining algorithm
- Multi-step decision-making
- Explanation logs
- Web-based user interaction
- Clean project structure

12. Professional Coding Practices

The project follows industry-level practices such as:

- Separation of concerns (logic, UI, styling)
- Clean and readable code
- Minimal and meaningful comments
- Modular design

- Scalable rule structure

13. Learning Outcomes

Through this project and my internship at **Syntacxhub**, I gained:

- Practical understanding of expert systems
- Experience with forward chaining inference
- Improved Python and Flask skills
- Better problem-solving and logical thinking
- Exposure to professional software development standards

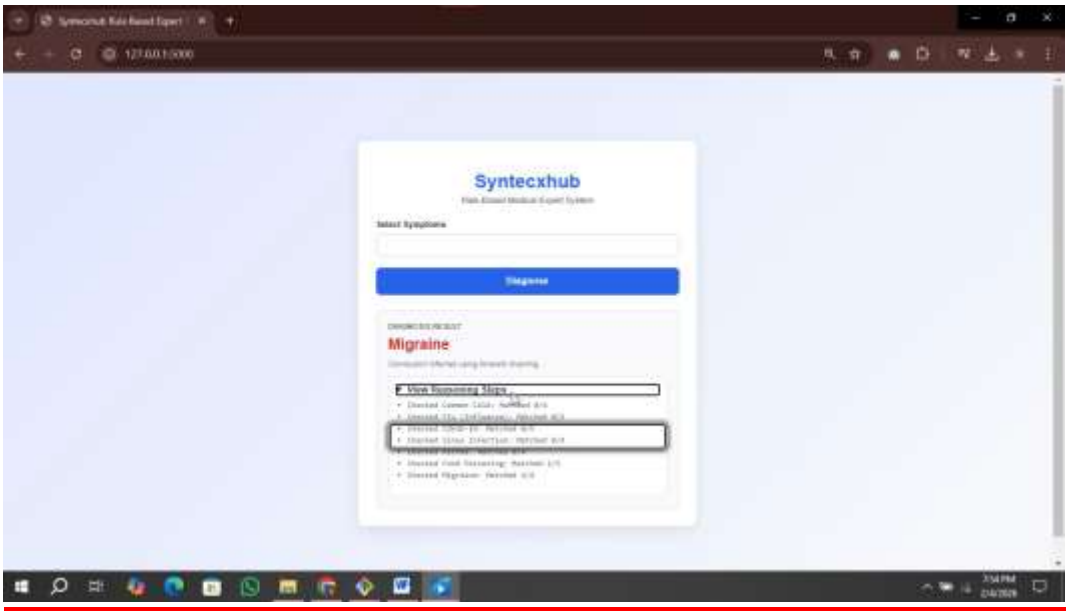
14. Role of Syntacxhub

I am grateful to **Syntacxhub** for providing me the opportunity to work on this project. The internship environment allowed me to:

- Apply theoretical knowledge to real projects
- Learn industry-relevant practices
- Enhance my technical and analytical skills
- Gain confidence in backend and logic-based development

This project reflects the skills and knowledge I developed during my time at Syntacxhub.

Result_Visualization



15. Conclusion

The Rule-Based Medical Expert System successfully meets all the objectives of a standard expert system. It implements forward chaining, multi-step inference, and transparent reasoning in a clean and professional manner.

This project demonstrates my ability to design logical systems, work with structured rules, and build professional software solutions, aligning well with industry expectations.

Special Thanks To Syntecxhub