Table of Contents

Medical Diagnosis Expert System

1. Acknowledgement

I would like to express my deepest gratitude to **Miss Dua Agha** for her constant guidance, encouragement, and support throughout this project. Her lectures on *Expert Systems Fundamentals* and *Inference Mechanisms* provided the conceptual foundation that enabled me to design and implement this intelligent system successfully.
I am also thankful to my classmates and institution for providing the resources to complete this work.

2. Abstract

This project presents an **Expert System for Medical Diagnosis** built using **Prolog**. The system simulates a doctor's reasoning process by asking the user about symptoms and providing a likely disease diagnosis along with confidence level, severity, treatment, and preventive measures.
It uses a **knowledge base** of facts and rules for five common diseases: **Flu, Malaria, Typhoid, COVID-19, and Heart Attack**.
The inference engine applies **forward-chaining reasoning**, evaluating the user's symptoms and inferring the most probable illness. This project demonstrates the practical application of AI for decision support and preventive healthcare awareness.

3. Introduction

3.1 Expert System

An **Expert System** is an AI program that emulates the decision-making ability of a human expert. It uses logical reasoning and a knowledge base to solve problems requiring human expertise.

3.2 Components of an Expert System

Knowledge Base: Stores expert knowledge as facts and rules.

Inference Engine: Applies logical reasoning to derive conclusions.

User Interface: Communicates with the user to collect information and display results.

Knowledge Acquisition Module: Collects and updates domain knowledge.

3.3 Objective of the Project

To design a **rule-based medical expert system** that:

- Identifies probable diseases based on user-reported symptoms,
- Calculates confidence levels for accuracy,
- Suggests treatments and preventive actions.

4. Methodology

4.1 Knowledge Acquisition

Medical knowledge such as symptoms, treatments, and preventions was gathered for five diseases (Flu, Malaria, Typhoid, COVID-19, Heart Attack) from verified sources.

4.2 Knowledge Representation

Facts and rules were encoded in **Prolog** using first-order logic.
Example:

```
has_symptoms(flu, [fever, cough, sore_throat, runny_nose, body_ache]).
```

4.3 Inference Mechanism

The system applies **forward chaining (data-driven reasoning)** to infer the best diagnosis by:

- Matching user input with known disease symptoms.
- Calculating confidence percentage.
- Selecting the disease with the highest match.

4.4 Implementation

The expert system was implemented in **Prolog** for its strong reasoning and pattern-matching capabilities.

4.5 Testing

Various user inputs were tested to validate accuracy and logical correctness of the results.

5. Description of Diseases

Disease	Description	Symptoms	Severity	Treatment	Prevention
Flu (Influenza)	A viral respiratory illness causing mild to moderate symptoms.	fever, cough, sore throat, runny nose, body ache	2 (Mild)	Rest, warm fluids, paracetamol, vitamin C, avoid strenuous activity	Wash hands, keep warm, avoid crowded places, get flu vaccine annually
Malaria	A parasitic disease transmitted through mosquito bites.	fever, sweating, nausea, headache, body ache	5 (Severe)	Doctor visit, anti-malaria medicine, hydration, rest	Mosquito nets, avoid mosquitoes, insect repellents, drain stagnant water
Typhoid	A bacterial infection spread through contaminated food or water.	fever, constipation, nausea, vomiting, headache	4 (Moderate –Severe)	Antibiotics, clean water, soft food, electrolyte drinks, avoid raw food	Boiled water, hygiene, avoid street food, vaccine if traveling to risk areas
COVID-19	A respiratory viral disease caused by SARS-CoV-2.	fever, cough, tiredness, loss of smell, headache	5 (Severe)	Isolate, rest, fluids, doctor if breathing issue, fever reducers, monitor oxygen	Mask, sanitize, avoid crowds, vaccination, maintain distance

Heart Attack	A critical medical emergency due to blocked blood flow to the heart.	chest pain, shortness of breath, left arm pain, sweating, dizziness	5 (Critical)	EMERGENCY: Call ambulance, chew aspirin, CPR if needed, hospital care	Exercise, avoid oily food, reduce stress, maintain healthy weight, regular checkups

6. Facts and Rules Used

6.1 Facts

These represent stored medical knowledge.
Examples:

```
symptom(fever).
disease(flu).
severity(covid, 5).
```

6.2 Rules

Rules describe logical relationships and inference steps.

Symptom Matching Rule

Confidence Calculation

Disease Selection Rule

7. Inference Reasoning

7.1 Concept

Inference means deriving new facts from known facts using reasoning rules, two main types exist:

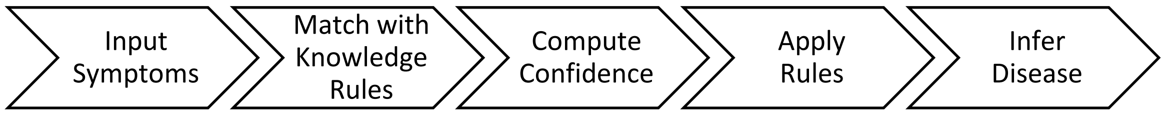
- **Forward Chaining (Data-Driven):** Start from facts → apply rules → reach conclusion.
- **Backward Chaining (Goal-Driven):** Start from a hypothesis → verify if facts support it.

7.2 Example Reasoning Flow

User gives symptoms: [fever, headache, nausea]
System checks matches:

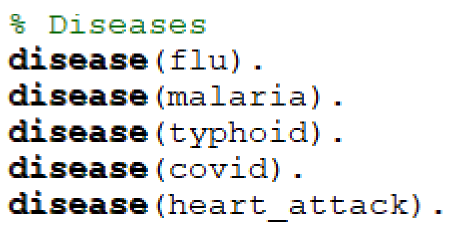
- Malaria → 3 matches
- Typhoid → 2 matches
- Flu → 1 match
  Inferred Disease: *Malaria (Highest Confidence)*

7.3 Rule-Based Reasoning Flow



8. Program Code and Explanation

8.1 Diseases Section

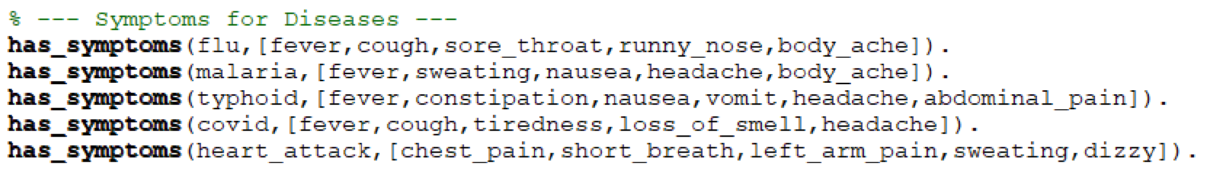


```
% Diseases
disease(flu).
disease(malaria).
disease(typhoid).
disease(covid).
disease(heart_attack).
```

Explanation:
This section lists all the **diseases** the system can diagnose.
Each line like **disease(flu).** defines a distinct disease fact.

- These are used later in the rules like **find_best_disease** to iterate through all possible diseases.
- It forms the **domain knowledge** for the diagnosis process.

8.2 Disease–Symptom Relationship

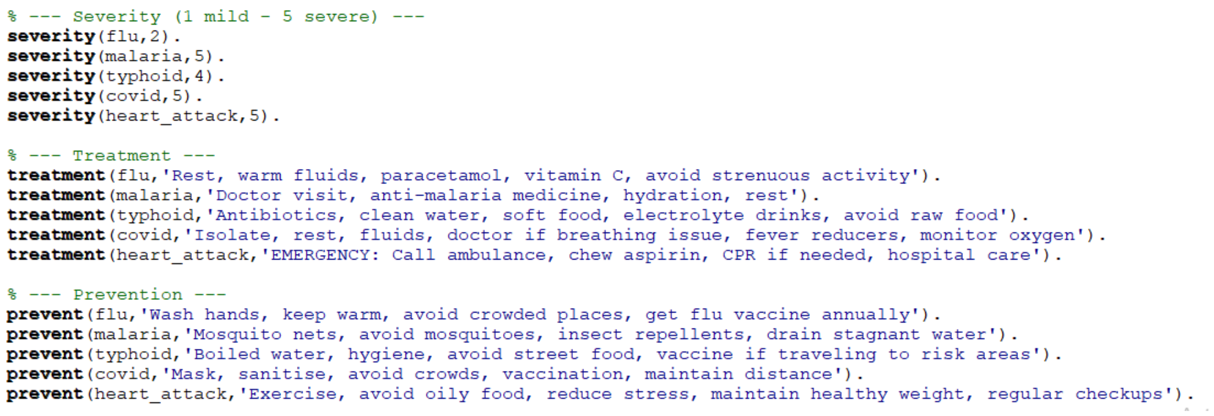


```
% --- Symptoms for Diseases ---
has_symptoms(flu,[fever,cough,sore_throat,runny_nose,body_ache]).
has_symptoms(malaria,[fever,sweating,nausea,headache,body_ache]).
has_symptoms(typhoid,[fever,constipation,nausea,vomit,headache,abdominal_pain]).
has_symptoms(covid,[fever,cough,tiredness,loss_of_smell,headache]).
has_symptoms(heart_attack,[chest_pain,short_breath,left_arm_pain,sweating,dizzy]).
```

Explanation:
This part defines which symptoms belong to which disease using the predicate **has_symptoms.**
For example,
has_symptoms(flu,[fever,cough,sore_throat,runny_nose,body_ache]).tells the system which symptoms are commonly associated with flu.

- The system uses this mapping when comparing user input with known disease symptoms.
- It helps compute the **confidence percentage** later.

8.3 Severity, Treatment, and Prevention

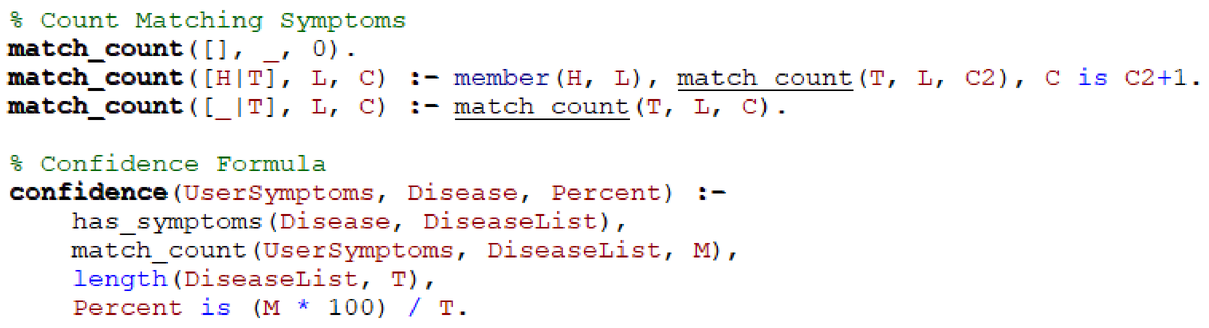


```
% --- Severity (1 mild - 5 severe) ---
severity(flu,2).
severity(malaria,5).
severity(typhoid,4).
severity(covid,5).
severity(heart_attack,5).

% --- Treatment ---
treatment(flu,'Rest, warm fluids, paracetamol, vitamin C, avoid strenuous activity').
treatment(malaria,'Doctor visit, anti-malaria medicine, hydration, rest').
treatment(typhoid,'Antibiotics, clean water, soft food, electrolyte drinks, avoid raw food').
treatment(covid,'Isolate, rest, fluids, doctor if breathing issue, fever reducers, monitor oxygen').
treatment(heart_attack,'EMERGENCY: Call ambulance, chew aspirin, CPR if needed, hospital care').

% --- Prevention ---
prevent(flu,'Wash hands, keep warm, avoid crowded places, get flu vaccine annually').
prevent(malaria,'Mosquito nets, avoid mosquitoes, insect repellents, drain stagnant water').
prevent(typhoid,'Boiled water, hygiene, avoid street food, vaccine if traveling to risk areas').
prevent(covid,'Mask, sanitise, avoid crowds, vaccination, maintain distance').
prevent(heart_attack,'Exercise, avoid oily food, reduce stress, maintain healthy weight, regular checkups').
```

Explanation:
These facts store the **severity level (1–5)**, **treatment**, and **prevention advice** for each disease.
For example:

- **severity(covid,5).** means COVID-19 is very severe.
- **treatment(flu,'Rest, warm fluids, paracetamol, vitamin C, avoid strenuous activity').** defines the remedy.
- **prevent(malaria,'Mosquito nets, avoid mosquitoes, insect repellents, drain stagnant water').**provides preventive measures.

8.4 Matching and Confidence Rules



```
% Count Matching Symptoms
match_count([], _, 0).
match_count([H|T], L, C) :- member(H, L), match_count(T, L, C2), C is C2+1.
match_count([_|T], L, C) :- match_count(T, L, C).

% Confidence Formula
confidence(UserSymptoms, Disease, Percent) :-
    has_symptoms(Disease, DiseaseList),
    match_count(UserSymptoms, DiseaseList, M),
    length(DiseaseList, T),
    Percent is (M * 100) / T.
```

Explanation:
These rules handle logic and computation:

- **match_count** counts how many user symptoms match a disease's known symptoms.
- **confidence** calculates how confident the system is in its diagnosis using the formula:

$$\text{Confidence} = \frac{\text{Matches}}{\text{Total Symptoms}} \times 100$$

8.5 Disease Selection and Comparison

```
% --- Find Best Disease ---
find_best_disease(UserSymptoms, BestDisease, BestConfidence) :-
    confidence(UserSymptoms, flu, C1),
    confidence(UserSymptoms, malaria, C2),
    confidence(UserSymptoms, typhoid, C3),
    confidence(UserSymptoms, covid, C4),
    confidence(UserSymptoms, heart_attack, C5),
    Pairs = [C1-flu, C2-malaria, C3-typhoid, C4-covid, C5-heart_attack],
    max_member(BestConfidence-BestDisease, Pairs),
    BestConfidence > 0.

% --- Collect All Possible Symptoms ---
all_symptoms(AllSymptoms) :-
    findall(S, (has_symptoms(_, L), member(S, L)), L1),
    sort(L1, AllSymptoms).
```

Explanation:
This rule loops through all diseases using **findall**, calculates confidence for each, and keeps track of the disease with the highest match.

- It represents the **inference engine's decision-making** process.

8.6 Asking User Symptoms

```
% --- Ask User Symptoms ---
ask([], []).
ask([S|T], [S|A]) :-
    write('Do you have '), write(S), write('? (yes/no): '),
    read(Response),
    Response == yes, !,
    ask(T, A).
ask([_|T], A) :-
    ask(T, A).
```

Explanation:
This rule handles **user interaction**.

- It displays each symptom as a yes/no question.
- The user's "yes" responses are stored in **UserSymptoms** which are later used for diagnosis.

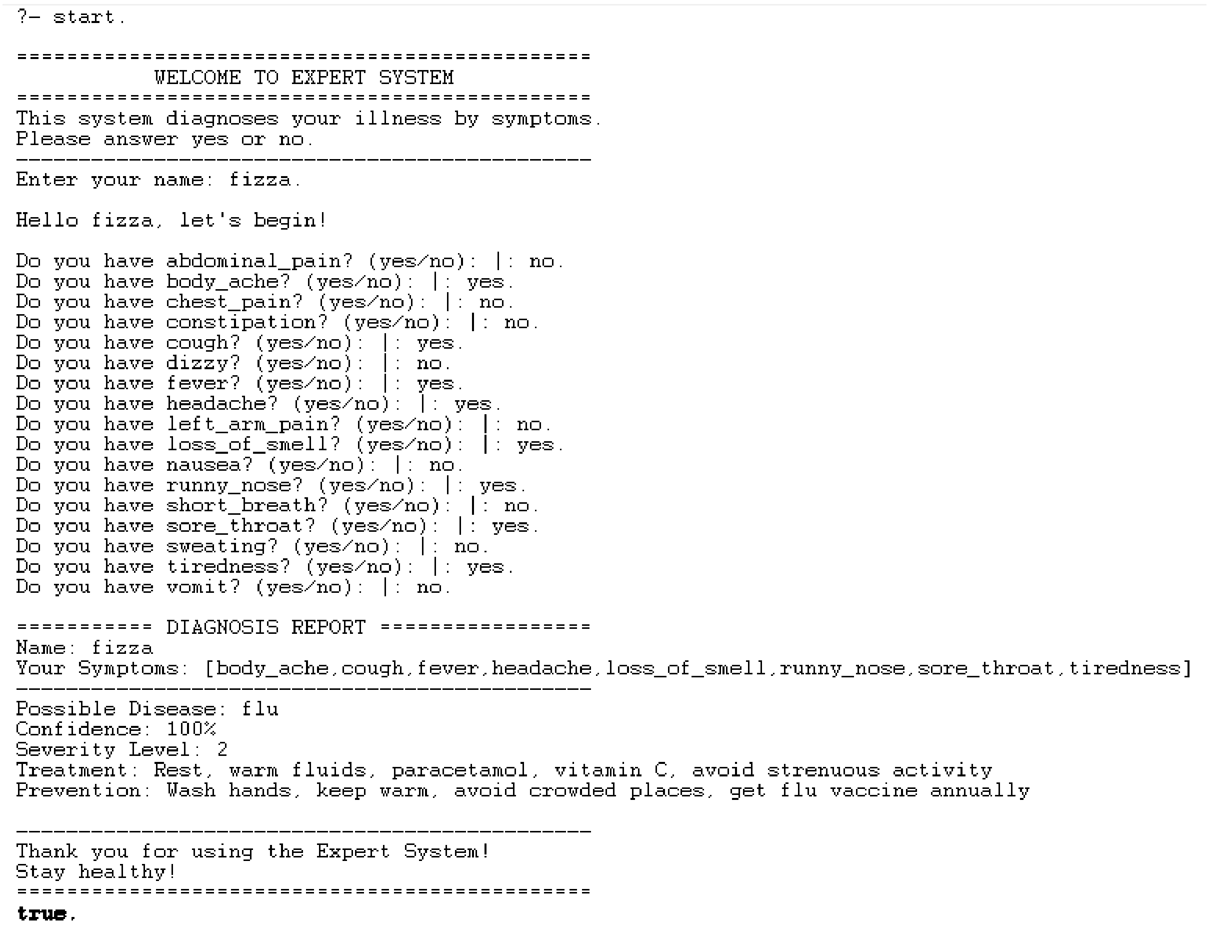
8.7 Start Rule (Main Program)

```
% --- Start the Expert System ---
start :-
    nl, write('=============================================='), nl,
    write('           WELCOME TO EXPERT SYSTEM           '), nl,
    write('=============================================='), nl,
    write('This system diagnoses your illness by symptoms.'), nl,
    write('Please answer yes or no.'), nl,
    write('----------------------------------------------'), nl,

    write('Enter your name: '), read(Name),
    nl, write('Hello '), write(Name), write(', let''s begin!'), nl, nl,

    all_symptoms(AllSymptoms),
    ask(AllSymptoms, UserSymptoms),

    find_best_disease_start(UserSymptoms, Name).

find_best_disease_start(UserSymptoms, Name) :-
    find_best_disease(UserSymptoms, Best, Conf), !,
    nl, write('=========== DIAGNOSIS REPORT ================='), nl,
    write('Name: '), write(Name), nl,
    write('Your Symptoms: '), write(UserSymptoms), nl,
    write('----------------------------------------------'), nl,
    severity(Best, Sev),
    treatment(Best, Treat),
    prevent(Best, Prev),
    write('Possible Disease: '), write(Best), nl,
    write('Confidence: '), write(Conf), write('%'), nl,
    write('Severity Level: '), write(Sev), nl,
    write('Treatment: '), write(Treat), nl,
    write('Prevention: '), write(Prev), nl,
    nl, write('----------------------------------------------'), nl,
    write('Thank you for using the Expert System!'), nl,
    write('Stay healthy!'), nl,
    write('==============================================='), nl.

find_best_disease_start(_, _) :-
    nl, write('----------------------------------------'), nl,
    write('No matching disease. Consult a doctor.'), nl,
    write('----------------------------------------'), nl.
```

Explanation:
This is the **main entry point** for the system.
It:

1. Greets the user.
2. Collects name and symptoms.
3. Runs diagnosis logic.
4. Displays disease name, confidence %, severity, treatment, and prevention.

9. Query



```
?- start.

=============================================
            WELCOME TO EXPERT SYSTEM
=============================================
This system diagnoses your illness by symptoms.
Please answer yes or no.
---------------------------------------------
Enter your name: fizza.

Hello fizza, let's begin!

Do you have abdominal_pain? (yes/no): |: no.
Do you have body_ache? (yes/no): |: yes.
Do you have chest_pain? (yes/no): |: no.
Do you have constipation? (yes/no): |: no.
Do you have cough? (yes/no): |: yes.
Do you have dizzy? (yes/no): |: no.
Do you have fever? (yes/no): |: yes.
Do you have headache? (yes/no): |: yes.
Do you have left_arm_pain? (yes/no): |: no.
Do you have loss_of_smell? (yes/no): |: yes.
Do you have nausea? (yes/no): |: no.
Do you have runny_nose? (yes/no): |: yes.
Do you have short_breath? (yes/no): |: no.
Do you have sore_throat? (yes/no): |: yes.
Do you have sweating? (yes/no): |: no.
Do you have tiredness? (yes/no): |: yes.
Do you have vomit? (yes/no): |: no.

=========== DIAGNOSIS REPORT =================
Name: fizza
Your Symptoms: [body_ache,cough,fever,headache,loss_of_smell,runny_nose,sore_throat,tiredness]
---------------------------------------------
Possible Disease: flu
Confidence: 100%
Severity Level: 2
Treatment: Rest, warm fluids, paracetamol, vitamin C, avoid strenuous activity
Prevention: Wash hands, keep warm, avoid crowded places, get flu vaccine annually

---------------------------------------------
Thank you for using the Expert System!
Stay healthy!
=============================================
true.
```

10. Challenges & their Solutions

Challenge	Description	Cause	Solution
Repeated Symptom Questions	Some symptoms (e.g., chest pain, dizziness) were being asked multiple times during execution.	The symptom-asking logic treated “yes” and “no” responses separately, causing Prolog to re-ask if input didn’t match exactly.	Simplified the symptom-asking logic to ensure each symptom is prompted only once, recording the response correctly or skipping it.
Infinite Loop in Symptom Questions	Certain symptoms were asked repeatedly, creating an infinite loop and preventing the system from progressing.	Backtracking in Prolog caused the same symptom to be re-evaluated multiple times if user input didn’t match expectations.	Revised the logic so each symptom is asked only once and unexpected input is handled gracefully without repeating questions.
Program Running Repeatedly on Its Own	After finishing, the system restarted automatically without user input, causing repeated executions.	The main start predicate or recursion inside the symptom-asking logic allowed Prolog to backtrack and re-run the program.	Adjusted program flow and termination conditions to stop execution after completing the diagnosis, preventing automatic repeats.
Symptoms Stored as Separate Facts	Symptoms were stored individually per disease instead of as a group.	Checking multiple individual facts for each disease increased redundancy and complexity, making confidence calculation and symptom collection harder.	Restructured symptoms to be grouped as a list per disease, simplifying queries, confidence calculation, and symptom management.
Confidence Level Accuracy	The confidence percentage often came out incorrect or above 100%, leading to unreliable diagnosis.	Repeated symptoms or integer division errors caused miscalculations.	Corrected the formula to (Matched * 100) / Total for proper percentage calculation. Also, duplicate symptoms were removed using the sort() function in symptom collection.

11. Conclusion

The **Medical Diagnosis Expert System** successfully demonstrates how knowledge-based reasoning can simulate a doctor’s diagnostic process.
It provides a **preliminary diagnostic suggestion** with treatment and prevention guidance. While not a replacement for professional medical care, it highlights the potential of **AI in healthcare decision support systems**.