MedExpert: Prolog-Based Medical Diagnosis Expert System

```
% ======================================
%        MedExpert: Prolog Medical Expert System
%        Author: Maria (SEO + Technical Writer)
%        Goal: Demonstrate advanced Prolog techniques
%             to diagnose diseases based on symptoms
% ======================================

% SECTION 1: Knowledge Base - Symptoms and Diseases

% Disease Definitions:
disease(cold, viral, low).
disease(flu, viral, medium).
disease(covid19, viral, high).
disease(pneumonia, bacterial, high).
disease(asthma, chronic, medium).
disease(tuberculosis, bacterial, high).

% Symptoms per disease:
symptom(cold, sneezing).
symptom(cold, runny_nose).
symptom(cold, sore_throat).
symptom(cold, headache).
symptom(flu, fever).
symptom(flu, chills).
symptom(flu, sore_throat).
symptom(flu, headache).
symptom(covid19, fever).
symptom(covid19, cough).
symptom(covid19, shortness_of_breath).
symptom(covid19, fatigue).
symptom(covid19, loss_of_taste).
symptom(pneumonia, cough).
symptom(pneumonia, chest_pain).
symptom(pneumonia, shortness_of_breath).
symptom(pneumonia, fever).
```

```
symptom(asthma, shortness_of_breath).
symptom(asthma, wheezing).
symptom(asthma, chest_tightness).
symptom(asthma, coughing).
symptom(tuberculosis, weight_loss).
symptom(tuberculosis, night_sweats).
symptom(tuberculosis, fever).
symptom(tuberculosis, persistent_cough).

% SECTION 2: Inference Rules
:- dynamic has_symptom/2.

possible_disease(User, Disease) :-
   disease(Disease, _, _),
   forall(symptom(Disease, Symptom), has_symptom(User, Symptom)).

ranked_diseases(User, RankedList) :-
   findall(Disease-Count, (
      disease(Disease, _, _),
      findall(Symptom, (symptom(Disease, Symptom), has_symptom(User, Symptom)),
MatchList),
      length(MatchList, Count),
      Count > 0
   ), Results),
   sort(2, @>=, Results, RankedList).

high_risk_disease(Disease) :-
   disease(Disease, _, high).

% SECTION 3: Diagnosis Engine
diagnose(User) :-
   ranked_diseases(User, Ranked),
   format('\n Diagnosing based on your symptoms...\n\n'),
   display_diseases(Ranked),
   (   member(Top-_, Ranked),
      high_risk_disease(Top)
   ->  format('  Potential high-risk condition: ~w~n', [Top])
   ;   true
   ),
   format(' Diagnosis Complete.\n').
```

```
display_diseases([]) :-
    write('No matching diseases found. Try adding more symptoms.\n').
display_diseases([D-C|T]) :-
    format('□ ~w (matched symptoms: ~w)~n', [D, C]),
    display_diseases(T).

% SECTION 4: User Simulation & Input
simulate_user(User) :-
    assert(has_symptom(User, fever)),
    assert(has_symptom(User, cough)),
    assert(has_symptom(User, fatigue)),
    assert(has_symptom(User, shortness_of_breath)),
    assert(has_symptom(User, loss_of_taste)),
    diagnose(User),
    cleanup_user(User).

cleanup_user(User) :-
    retractall(has_symptom(User, _)).

% SECTION 5: Debugging & Performance
enable_debugging :-
    trace.

disable_debugging :-
    notrace.

:- dynamic memo_disease_check/3.

memoized_possible_disease(User, Disease, Count) :-
    memo_disease_check(User, Disease, Count), !.

memoized_possible_disease(User, Disease, Count) :-
    findall(Symptom, (symptom(Disease, Symptom), has_symptom(User, Symptom)),
MatchList),
    length(MatchList, Count),
    assertz(memo_disease_check(User, Disease, Count)).

% SECTION 6: CLI Simulation
start_session :-
```

```
    write('Welcome to MedExpert: Prolog Medical Diagnosis\n'),
    write('Enter your name: '), read(User),
    input_symptoms(User),
    diagnose(User),
    cleanup_user(User).

input_symptoms(User) :-
    write('Enter symptoms one by one. Type "done." to finish.\n'),
    read_symptoms(User).

read_symptoms(User) :-
    read(Input),
    (   Input == done
    ->  write('□  Symptoms collected\n')
    ;   assertz(has_symptom(User, Input)),
        read_symptoms(User)
    ).

% SECTION 7: Sample Queries
% ?- start_session.
% ?- simulate_user(john).
% ?- enable_debugging, simulate_user(maria), disable_debugging.
% ?- simulate_user(ali), memo_disease_check(ali, Disease, Count).
```