**Semester Project Report**

**Project Title: Eczema diagnosis system**

**Presenter: Walid khan Ahmed, Zain Ul Abidin, Muneer**

**Reg No:22MDBCS274,302,273s**

**Problem Statement:**

Eczema, also known as atopic dermatitis, is a chronic skin condition characterized by inflammation, redness, and itching, affecting millions of people worldwide. Early and accurate diagnosis is critical to managing symptoms and improving patients' quality of life. However, diagnosing eczema often requires expert dermatological knowledge, which can be inaccessible or expensive in underserved areas.

Healthcare providers face challenges such as:

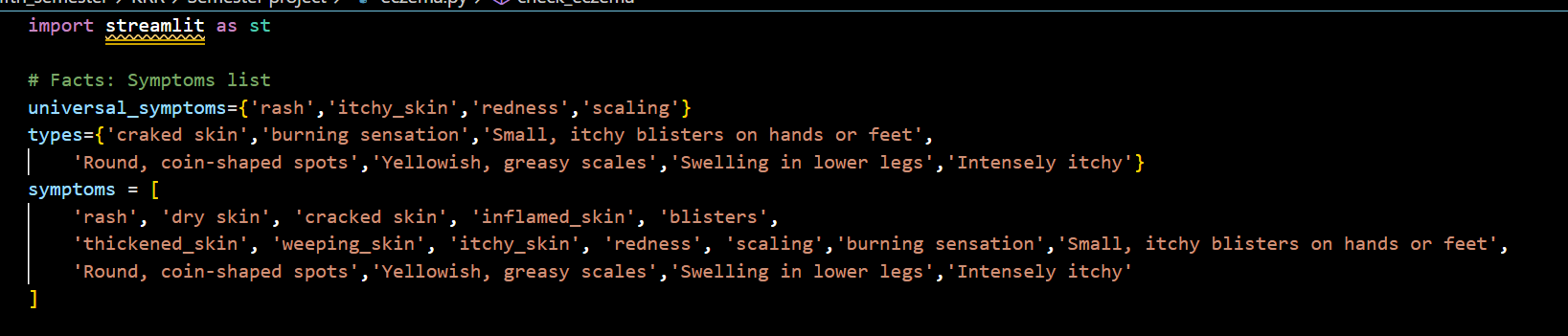
* Limited availability of dermatologists in rural and remote areas.
* Time-consuming and subjective diagnosis processes based on visual assessment and patient history.

To address these challenges, there is a need for an **automated Eczema Diagnosis System** that leverages machine learning and image processing to assist healthcare providers in diagnosing eczema accurately and efficiently. Such a system would analyze skin images, identify patterns indicative of eczema, and provide a preliminary diagnosis to aid clinicians. However we use KRR to implement such system for now.

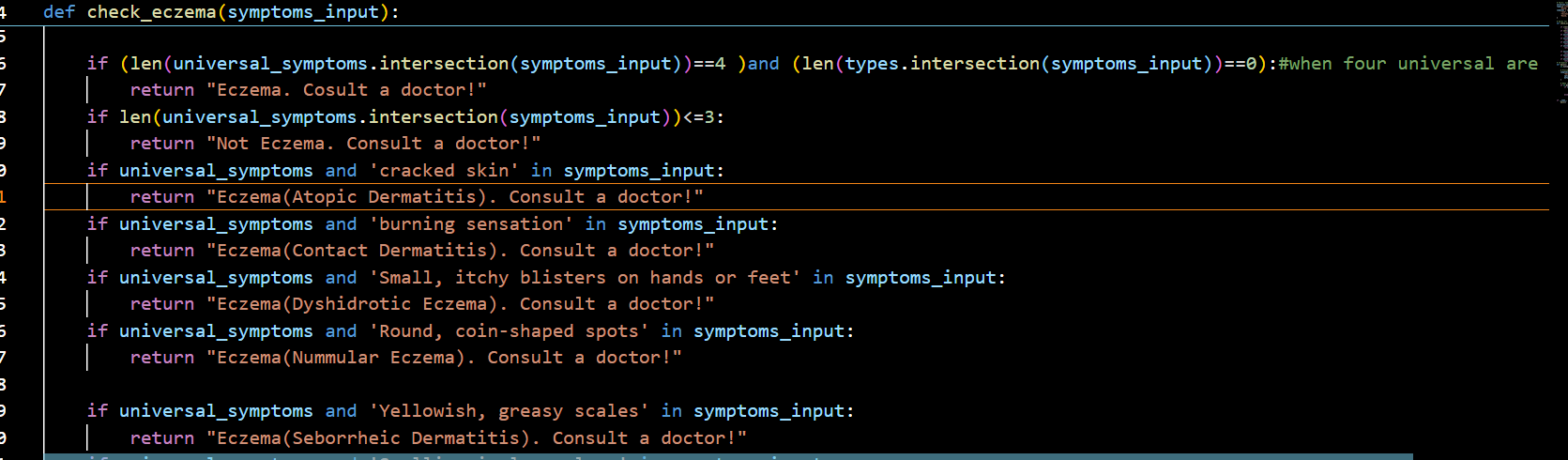
**Knowledge representation techniques used:**

Thetechniques used in the eczema diagnosis system is Declarative representation of knowledge and rule production.

Declarative representation:



Rule Production:



**System architecture and reasoning process:**

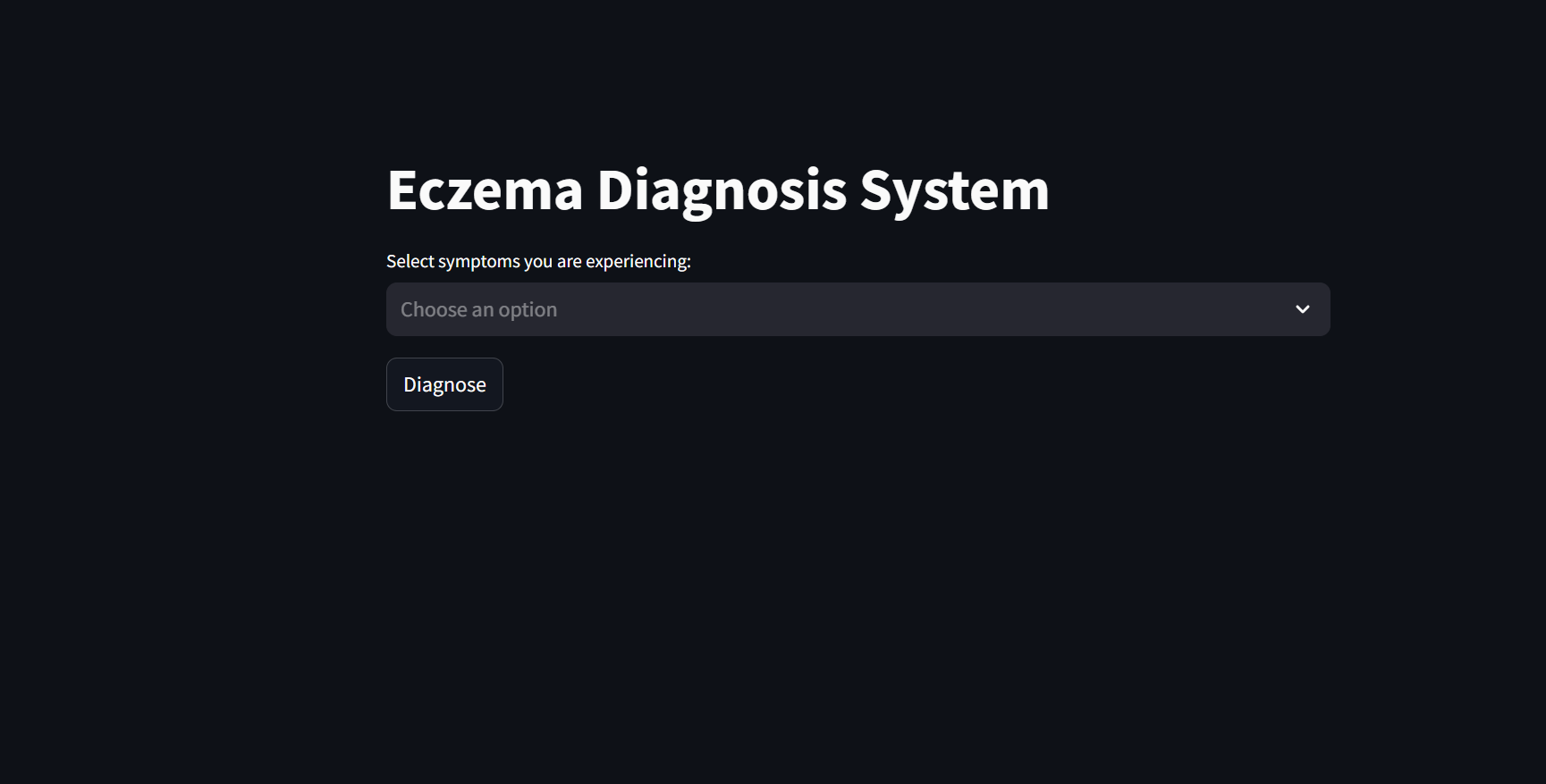
**System Architecture:**

1. **User Interface Layer:**

**Tool**: Streamlit provides an interactive UI for the user to input symptoms through a multiselect widget.

**Functionality**:

* + - Users select symptoms from a predefined list.
    - The input symptoms are submitted to the backend for diagnosis when the "Diagnose" button is clicked.



1. **Knowledge Base:**

**Facts**:

* + - Universal symptoms and specific types of symptoms are predefined in sets:

python

Copy code

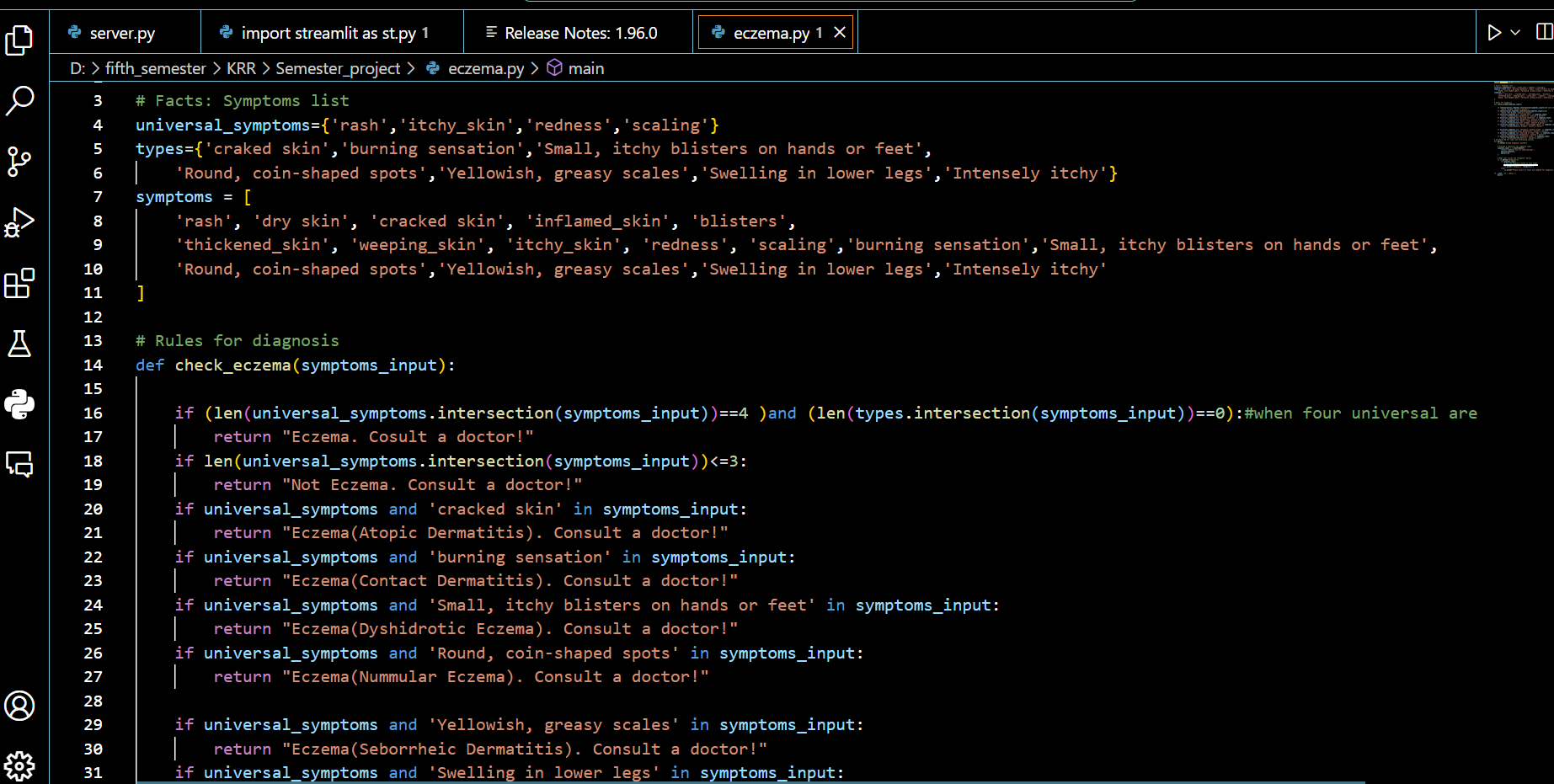
universal\_symptoms = {'rash', 'itchy\_skin', 'redness', 'scaling'}

types = {'cracked skin', 'burning sensation', ...}

* + - A comprehensive list of symptoms serves as the domain of discourse.

**Rules**:

* + - Encoded in the check\_eczema() function using if-else conditions.
    - These rules determine the type of eczema (or absence thereof) based on symptoms provided by the user.



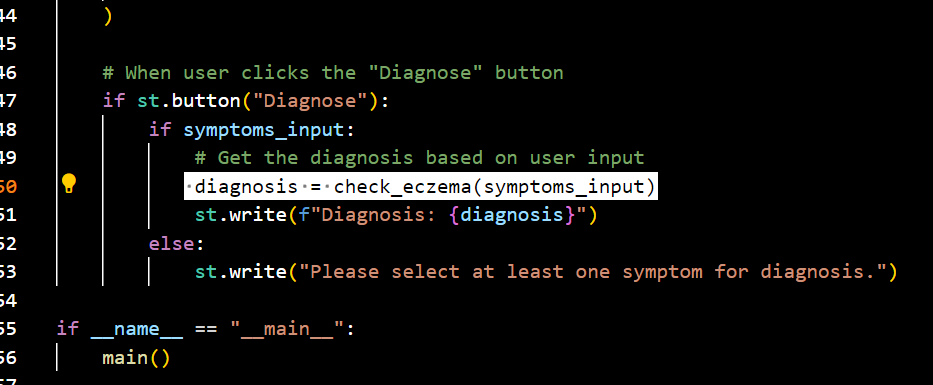
1. **Reasoning Engine:**

**Logic Processor**:

* + - The check\_eczema() function processes input symptoms and applies rules.
    - The reasoning is based on set operations (e.g., intersection) to match user-provided symptoms with predefined rules.

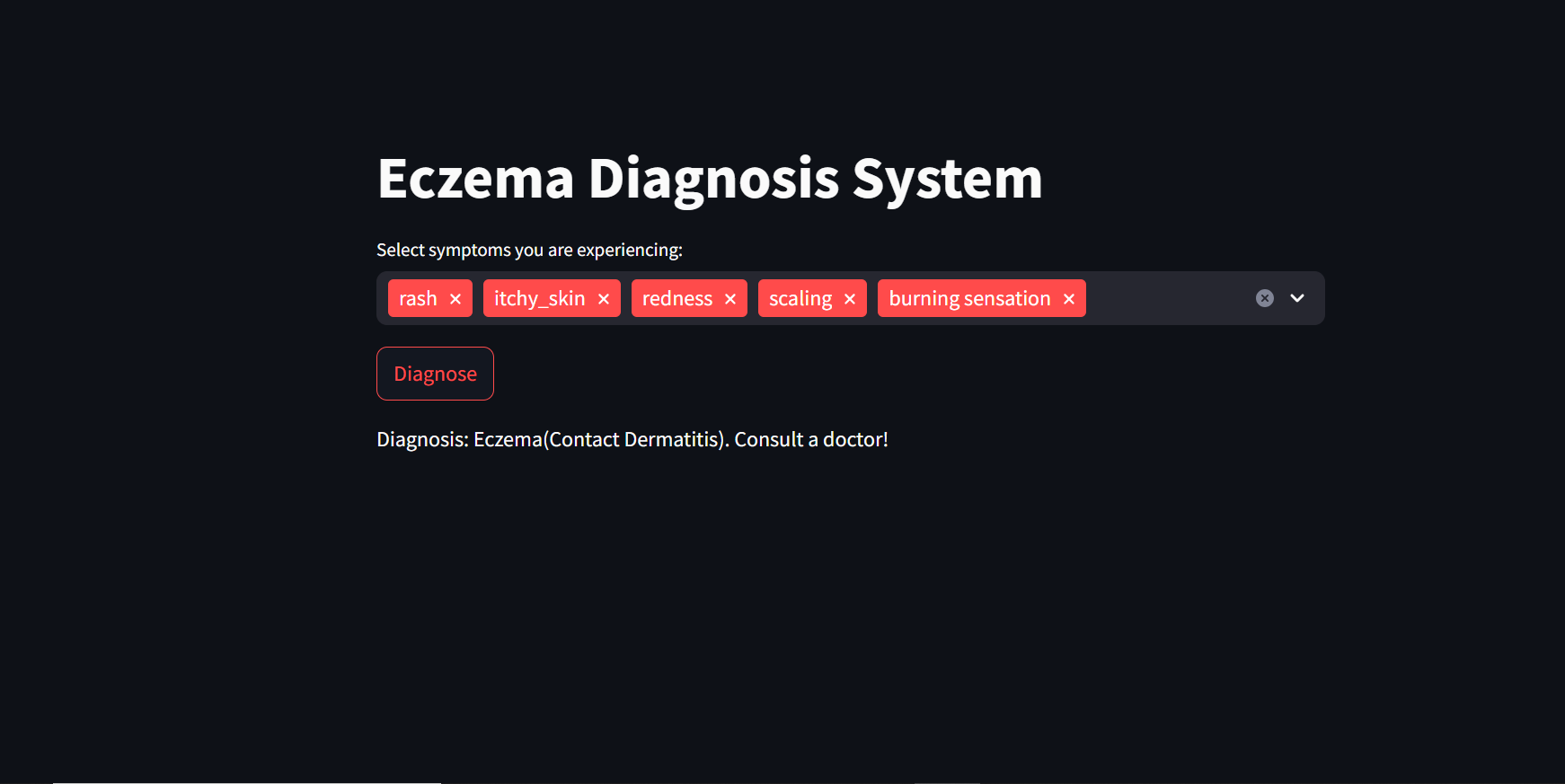
**Inference Mechanism**:

* + - Rule-based reasoning (forward chaining): Sequentially checks rules to find a match for the input.



1. **Output Layer:**

The diagnosis result is displayed to the user on the Streamlit app interface.



**Reasoning Process(Forward Chaining):**

1. **Input Acquisition:**

User selects symptoms through a UI (st.multiselect), which are passed as a list (symptoms\_input).

1. **Matching Symptoms:**

**Intersection Operation**:

* + - The system checks if user-inputted symptoms match with:
      * Universal symptoms (universal\_symptoms).
      * Specific types of symptoms (types).

Example:

python

Copy code

if universal\_symptoms and 'burning sensation' in symptoms\_input:

return "Eczema(Contact Dermatitis). Consult a doctor!"

**3: Rule Evaluation:**

**Sequential Rule Checking**:

* + - Each rule is evaluated in order of priority, using if-else conditions.
    - For example:
      * If exactly 4 universal symptoms are present and no specific type is matched:

python

Copy code

if (len(universal\_symptoms.intersection(symptoms\_input)) == 4) and (len(types.intersection(symptoms\_input)) == 0):

return "Eczema. Consult a doctor!"

1. **Diagnosis Determination:**

* The first rule satisfied determines the output (single-deterministic reasoning).
* If no rules match, the system defaults to the fallback case:

python

Copy code

if len(universal\_symptoms.intersection(symptoms\_input)) <= 3:

return "Not Eczema. Consult a doctor!"

1. **Result Delivery:**

The diagnosis result is displayed in the UI as text output.

**Challenges faced and solutions implemented:**

1. Finding the universal symptoms that must be there in a person having eczema solution implemented is that I did a research by conforming from a cousin doctor and a research on the internet.
2. Initially I implemented the system in a way it only detected eczema but later I realized there are types of eczema which should also be implemented and then by doing a research I was able to do it.
3. Initially I tried to use backward chaining by asking questions from the user but then I realized I should use forward chaining by letting user input symptoms.