# LOVELY PROFESSIONAL UNIVERSITY

****

**Project Name:** Health Risk Advisor Using Fuzzy Logic

|  |
| --- |
| **Student Name: R DEEPAK** |
| **Student’s Reg. no: 12314444** |
| **Student’s Roll no: 55(K23AJ)** |

|  |
| --- |
| **Student Name: JAN MOHAMMAD** |
| **Student’s Reg. no: 12317720** |
| **Student’s Roll no: 54(K23AJ)** |

|  |
| --- |
| **Student Name: A ROHITH KUMAR** |
| **Student’s Reg. no: 12308444** |
| **Student’s Roll no: 53(K23AJ)** |

|  |
| --- |
| **Student Name: KULDIP DHAR** |
| **Student’s Reg. no: 12323736** |
| **Student’s Roll no: 56(K23AJ)** |

**ACKNOWLEDGEMENT**

First and foremost I’d like to thank my guide MS: **SHEVETA** for being so helpful and patient throughout this project. Their insights and support really helped shape the work

I also want to thank my friends, classmates for cheering me on especially when I was stuck or doubting myself. This project wouldn’t be complete without them

**ABSTRACT:**

This project is all about creating a simple tool that can give people a heads-up about their health based on a few common symptoms. It’s not meant to be a replacement for a doctor but more of a friendly suggestion tool using fuzzy logic

Fuzzy logic is useful when things aren’t black and white Health symptoms are often like that - someone might feel a bit tired, have a slight fever, or just feel “off.” Instead of saying “yes” or “no,” fuzzy logic helps say things like “maybe” or “somewhat.”

I built this system to take inputs like body temperature, Heart rate, and whether someone has a cough, and then suggest if their health risk is low, medium, or high.

**1. INTRODUCTION:**

We have all had days when we didn’t feel our best but were not sure if we should go to the doctor or just rest This system is designed to help in those moments - not to diagnose.

The system uses fuzzy logic which is a way of thinking that’s more like how humans make decisions - especially when the information isn’t perfect, Instead of being super technical or complicated, this tool is made to be as simple and helpful as possible.

**2. OBJECTIVES:**

* Build a basic system that can suggest health risk levels.
* Use fuzzy logic to deal with symptoms that are vague.
* Categorize health risk as low, moderate, or high.
* Make it simple and easy to use, even for non-technical people.

**3. BACKGROUND:**

Fuzzy logic has been around for a while - it was introduced by Lotfi Zadeh in the 1960s.

In healthcare, fuzzy logic has already been used in things like diabetes prediction, heart monitoring, and even in some fitness apps. What we have done here is just take that idea and apply it in a simpler way, using only a few basic symptoms.

**4. HOW IT WORKS:**

**Here is how the system works:**

1. The user enters a few simple details:
   * Body temperature
   * Fatigue level
   * Heart rate
   * Whether they have a cough or throat pain
2. The system turns these values into fuzzy categories like “slightly high temperature” or “high fatigue.”
3. A set of rules checks what combination of symptoms the person has.
4. Based on that, the system gives a risk level:
   * Low Risk
   * Moderate Risk
   * High Risk

It also gives a short message - like “You seem fine” or “You should probably see a doctor”

**5. BUILDING THE SYSTEM:**

**Technologies Used:**

* **Python 3**  
  Python served as the core development language for both backend logic and integration. It is widely used in AI and machine learning applications.
* **scikit-fuzzy (skfuzzy)**  
  This is a Python library specifically for building fuzzy logic systems. It made it easier to define fuzzy variables, membership functions, and inference rules in a structured and readable manner.
* **Flask**  
  Flask is a lightweight Python web framework used to create the backend of the web application. It handles routing, HTTP requests, and integrates the fuzzy logic system with the frontend.
* **HTML/CSS & Bootstrap**  
  HTML and CSS were used for the user interface. Bootstrap, a CSS framework, was used to give a clean and responsive layout to the form, ensuring a better user experience.
* **JavaScript (Minimal Use)**  
  JavaScript was optionally used to handle client-side interactivity and enhance the responsiveness of the UI if needed

**System Workflow Summary:**

1. User inputs **Temperature**, **Heart Rate**, and **Cough Level** through the web interface.
2. Flask routes this data to the backend Python script.
3. The **fuzzy logic engine** (via scikit-fuzzy) processes the input data based on predefined fuzzy rules and membership functions.
4. The computed fuzzy output is **defuzzified** and mapped to a **risk category** (Low, Medium, High).
5. The result is sent back and displayed on the web interface.

**6. SAMPLE RESULTS**

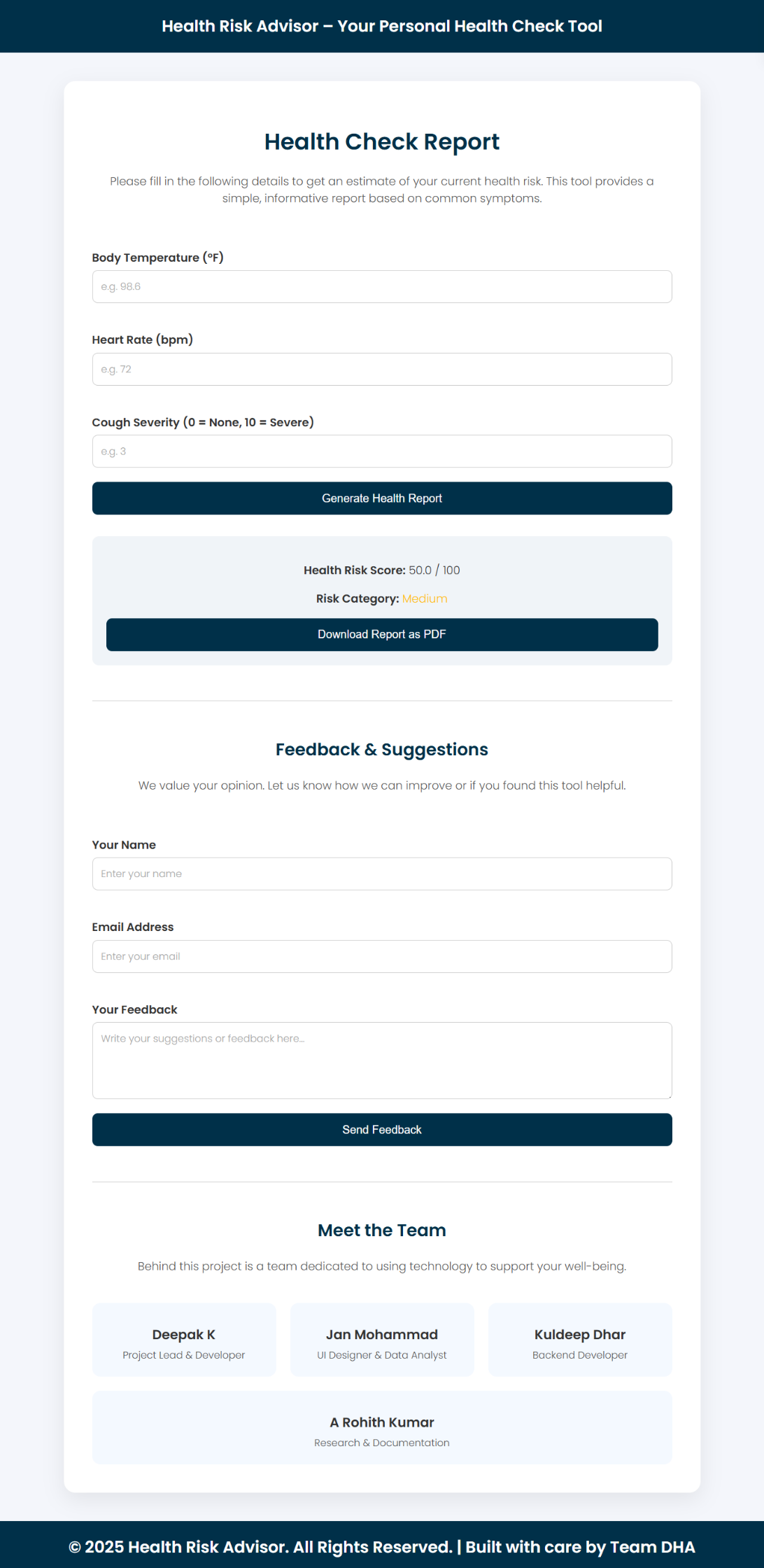
**Here are a few example cases:**

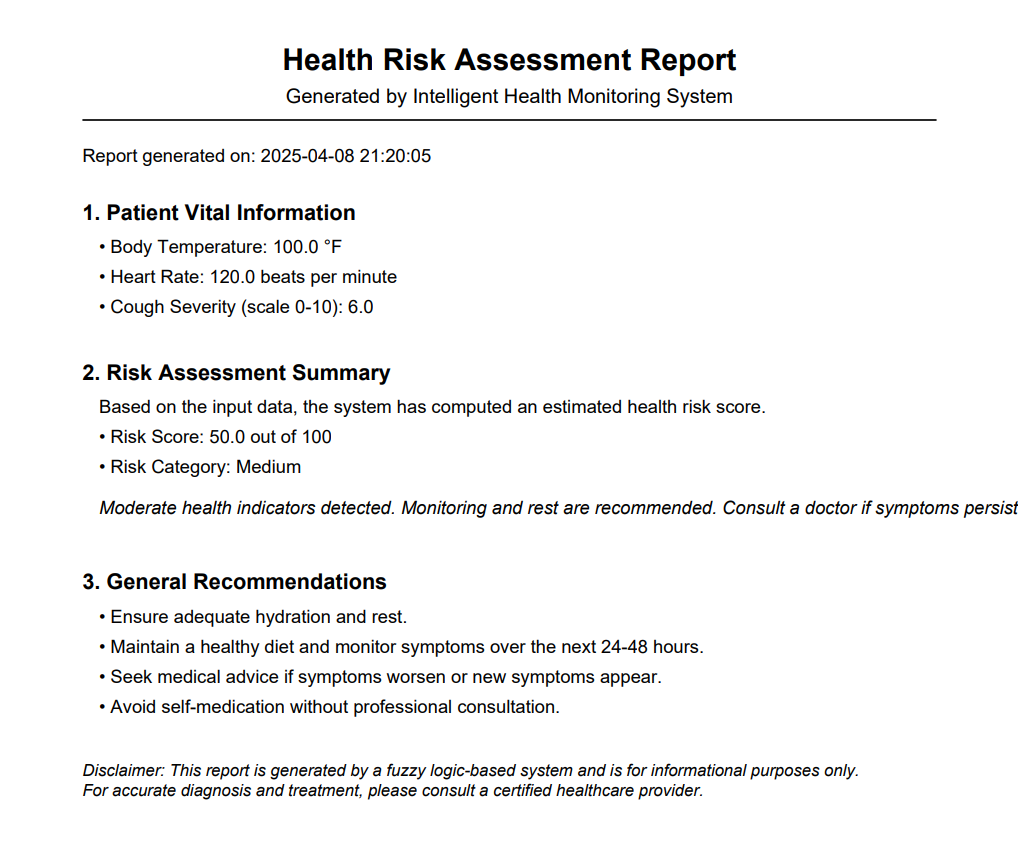
|  |  |  |  |
| --- | --- | --- | --- |
| Temp | Fatigue | Cough | Output |
| 101°F | high | Yes | High Risk |
| |  | | --- | | 98.5°F |  |  | | --- | |  | | medium | No | Moderate Risk |
| 97.6°F | low | No | Low Risk |

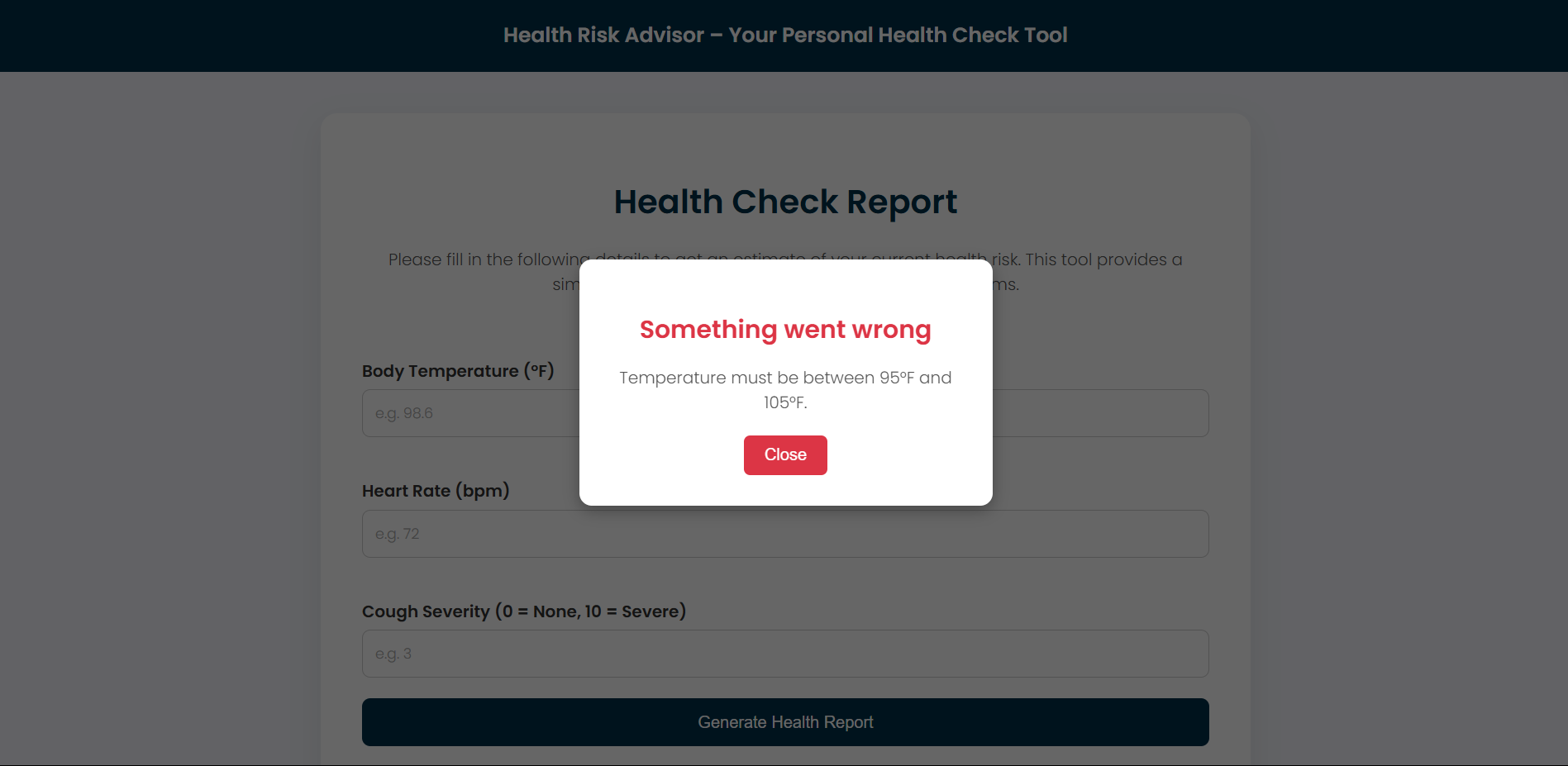
Even small differences can change the output slightly, which is what makes fuzzy logic feel more natural than hard if this thenthat rules.

**Flow chart:**

**SCREESHOTS:**

****

****

**Error For Incorrect Input**

**7. CONCLUSION**

Working on this project taught me a lot - not just about fuzzy logic, but also about how simple tools can actually be very useful.

The Health Risk Advisor is just a basic version now, but I can see it being used in health apps or even smart watches one day. It’s not meant to be a final answer, but more of a helpful nudge in the right direction**.**

**8. REFERENCES**

1. Zadeh, L. A., “Fuzzy Sets,” Information and Control, 1965
2. Python scikit-fuzzy documentation
3. Health articles and symptom guidelines for rule development
4. Discussions and papers on fuzzy logic in medical applications