* **Abstract** 
  + The Medical Diagnosis System is a Fuzzy Logic-based AI system designed for diagnosing diseases such as Diabetes Disease, Heart Disease, PCOD Disorder, Thyroid Disorder, and Anxiety Disorder.
  + It processes uncertain and imprecise medical data, mimicking human reasoning to make diagnostic decisions.
  + The system analyses symptoms, medical history, and test reports to provide risk assessment and early detection, improving healthcare outcomes with rule-based decision-making.
  + Additionally, the system includes a file upload feature, allowing patients to upload medical files (such as reports), based on which the system predicts the likelihood of the presence of specific diseases.
* **Introduction**
* **Background of the problem**

Accurate medical diagnosis often faces challenges due to uncertainty, incomplete information, and variability in patient symptoms. Traditional diagnostic methods may struggle to handle such complexities effectively. Fuzzy logic provides a flexible approach to model this uncertainty, offering a more human-like reasoning process for supporting medical decisions.

* **Purpose of the project**

The purpose of this project is to design and develop a Medical Diagnosis System using Fuzzy Logic that assists in the identification and risk assessment of diabetes, thyroid disorders, anxiety, PCOD, and heart diseases. The system aims to enhance diagnostic support by processing clinical data through fuzzy inference. It allows for both manual symptom input and automated analysis through uploaded patient files, enabling broader usability and convenience.

* **Scope and limitations**

The system evaluates selected health parameters to predict the risk levels for specific diseases using fuzzy rule-based logic.  
**Limitations** include:

* The system acts as a supportive tool and does not replace professional medical consultation.
* Results depend on the quality and accuracy of the input data.
* Fuzzy models are based on predefined rules and may require updates as medical knowledge evolves
* **Literature Review**

**1. Existing Research Work**

* **Diabetes Prediction**

The PIMA Indian Diabetes Dataset is widely used. Smith et al. (2019) applied Decision Trees, Random Forest, and SVM, achieving over 80% accuracy. Deep learning models like neural networks improved precision with larger datasets.

* **Thyroid Disorder Detection**

Zhang et al. (2020) applied ensemble methods like XGBoost and AdaBoost on UCI thyroid datasets, significantly improving detection of hyperthyroidism and hypothyroidism.

* **Heart Disease Prediction**

The Cleveland Heart Disease dataset is a standard benchmark. Logistic Regression, Gradient Boosting, and Neural Networks (Patel and Prajapati, 2018) have achieved up to 90% accuracy in heart disease prediction.

* **PCOD Detection**

Due to limited datasets, recent studies use hormonal and lifestyle data. Kumar et al. (2021) used Random Forests on health parameters, detecting PCOD with about 87% accuracy.

* **Anxiety Detection**

AI in mental health diagnostics is growing. NLP techniques classify anxiety from patient responses. Lee et al. (2021) used BERT models on forum data for high-accuracy anxiety detection.

**2. Similar Projects**

* **IBM Watson Health**

IBM’s Watson Health integrates large datasets for disease diagnosis and treatment recommendations, illustrating AI’s healthcare potential, although not disease-specific.

* **UCI ML Repository Projects**

Open-source projects using UCI datasets (diabetes, thyroid, heart disease) apply algorithms like k-NN, Naive Bayes, and Deep Neural Networks to compare model performances.

* **Disease Prediction Android Applications**

Apps like Ada Health and Babylon use user-reported symptoms and demographics to predict diseases, but they are generalized and do not specifically predict PCOD or anxiety.

* **Ensemble Disease Prediction Models**

Recent projects combine multiple datasets for multi-disease prediction. Rao and Singh (2022) developed a CNN-LSTM hybrid model to classify diabetes, heart disease, and thyroid disorders.

* **System Requirements**

# **Software Requirements**

|  |  |
| --- | --- |
| Component | Details |
| Operating System | Windows 10/11, macOS, or Linux (Ubuntu/Fedora) |
| Programming Language | Python 3.8 or higher |

# **Hardware Requirements**

|  |  |  |
| --- | --- | --- |
| Component | Minimum | Recommended |
| Processor | Dual-core 2.0 GHz | Quad-core Intel i5 / AMD Ryzen 5 or better |
| RAM | 4 GB | 8 GB or more |
| Storage | 1 GB free space | 5 GB+ SSD storage |
| Internet | Required for downloading libraries | Required for online dataset & Streamlit UI |
| Graphics | Not required | Integrated GPU sufficient |

* **Methodology**
* **Steps followed in the project**

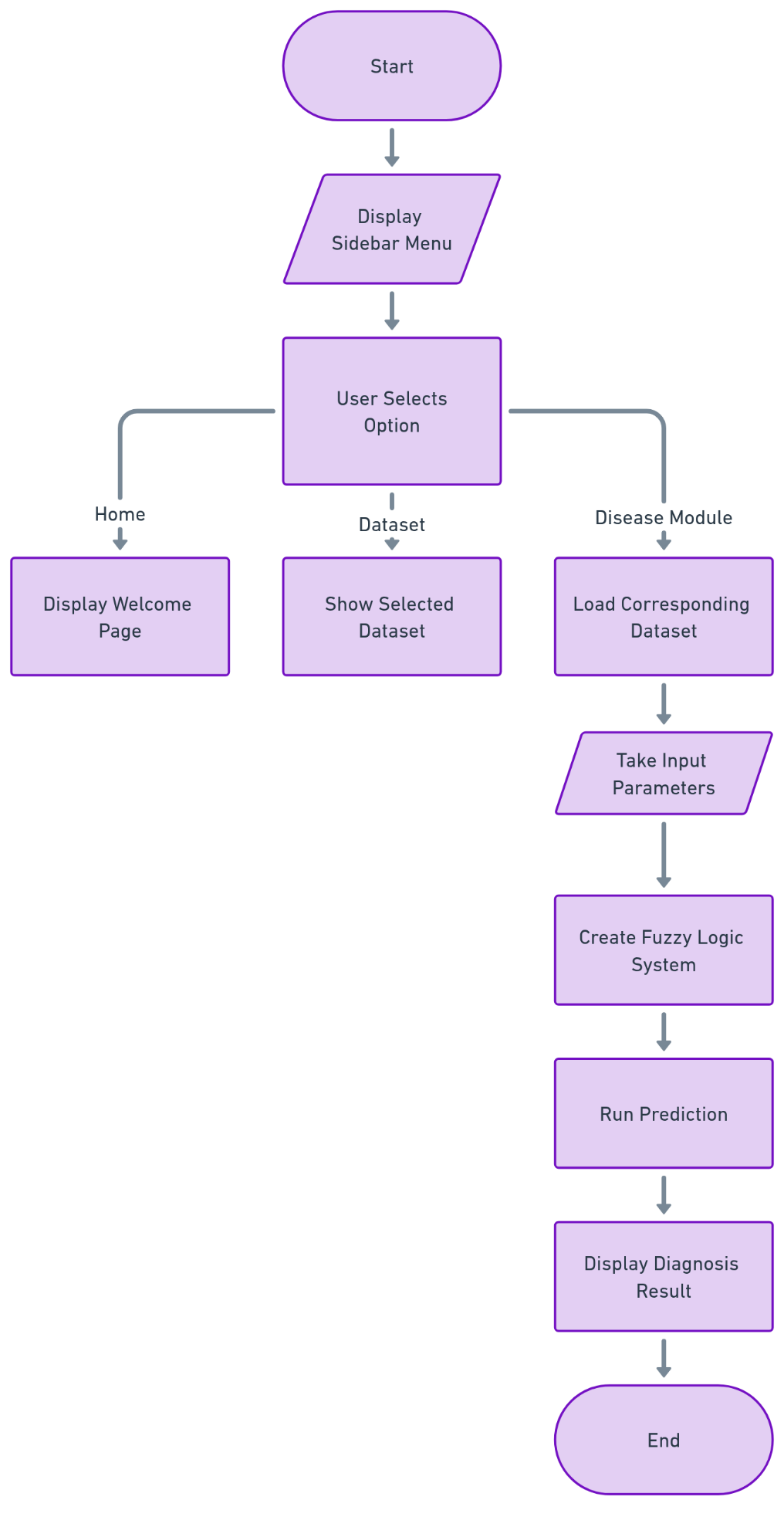
1. **Problem Identification**  
   Identified the need for a medical diagnosis system that can assess multiple diseases using fuzzy logic and provide early diagnosis support.
2. **Dataset Collection**  
   Collected publicly available datasets for the following diseases:
   * Diabetes
   * Heart Disease
   * Thyroid
   * PCOD
   * Anxiety
3. **Data Preprocessing**
   * Removed missing or irrelevant values
   * Renamed columns for clarity
   * Normalized or standardized data as needed
   * Selected the most relevant features (input parameters)
4. **Fuzzy Logic System Design**
   * Defined fuzzy input variables (e.g., Glucose, Blood Pressure, etc.)
   * Defined fuzzy output variable (e.g., Diagnosis or Risk Level)
   * Created membership functions (Poor, Average, Good or Low, Medium, High)
   * Formulated fuzzy rules (IF-THEN logic)
   * Simulated and tested the fuzzy control systems
5. **Frontend Development using Streamlit**
   * Designed an interactive user interface for each disease module
   * Created sidebar navigation using streamlit-option-menu
   * Collected input values from the user dynamically
   * Displayed diagnosis results with simple outputs (e.g., Yes/No, Risk Level)
6. **Backend Integration**
   * Linked user input to fuzzy control system
   * Processed the input using scikit-fuzzy library
   * Displayed real-time diagnosis results
7. **Dataset Display Feature**
   * Added a “Dataset” tab to allow users to view raw data
   * Used st.dataframe() to display tabular data for all five diseases
8. **Testing and Validation**
   * Ran multiple test cases using known data to verify outputs
   * Ensured accuracy and reliability of fuzzy results
9. **Final Integration**
   * Combined all modules (Diabetes, Heart, Thyroid, PCOD, Anxiety) into a single Streamlit app
   * Ensured a consistent and user-friendly design
10. **Deployment and Execution**
    * Final version launched locally using the command:
    * streamlit run app.py

* **Algorithms, flowcharts, or diagrams**
* **Algorithms**

The Fuzzy Inference System (FIS) consists of:

1. Fuzzification – Converts numeric inputs into fuzzy variables.
2. Fuzzy Rule-Based System – Uses expert-defined IF-THEN rules to determine medical risk.
3. Inference Engine – Applies fuzzy logic rules to evaluate risk levels.
4. Defuzzification – Converts fuzzy outputs into a Health risk.

* **Diagram :** System flow Diagram



* **Tools and technologies used**

|  |  |
| --- | --- |
| Tools and Technology | |
| IDE / Code Editor | Visual Studio Code / Jupyter Notebook / Anaconda Navigator |
| Web Framework | Streamlit |
| Required Python Packages | numpy, pandas, scikit-fuzzy, streamlit, streamlit-option-menu |
| Browser | Google Chrome / Mozilla Firefox / Microsoft Edge |
| Optional Tools | Excel / Google Sheets (for viewing CSV files) |

* **Implementation & Development**
* **Description of coding, database, or system development**

The medical diagnosis system was developed using **Python**, leveraging its powerful libraries and fuzzy logic capabilities to predict five key health conditions: **Heart Disease**, **Diabetes**, **Thyroid Disorders**, **PCOD**, and **Anxiety**.

* **Coding and Algorithm**

We used **fuzzy logic algorithms** to manage the uncertainty in medical data and make reliable predictions based on user inputs such as age, symptoms, and lifestyle. Key libraries include:

* **NumPy**, **Pandas** – Data handling
* **scikit-fuzzy** – Fuzzy logic implementation
* **Matplotlib**, **Seaborn** – Data visualization
* **Flask** – For web deployment (optional)

Each condition uses a dedicated fuzzy model with rules based on clinical research.

* **Database**

Data was sourced from public online databases such as:

* **UCI Repository** – Heart disease, diabetes
* **Kaggle** – Thyroid, PCOD, and anxiety

Data was pre-processed and used to train/test the system without including any personal information.

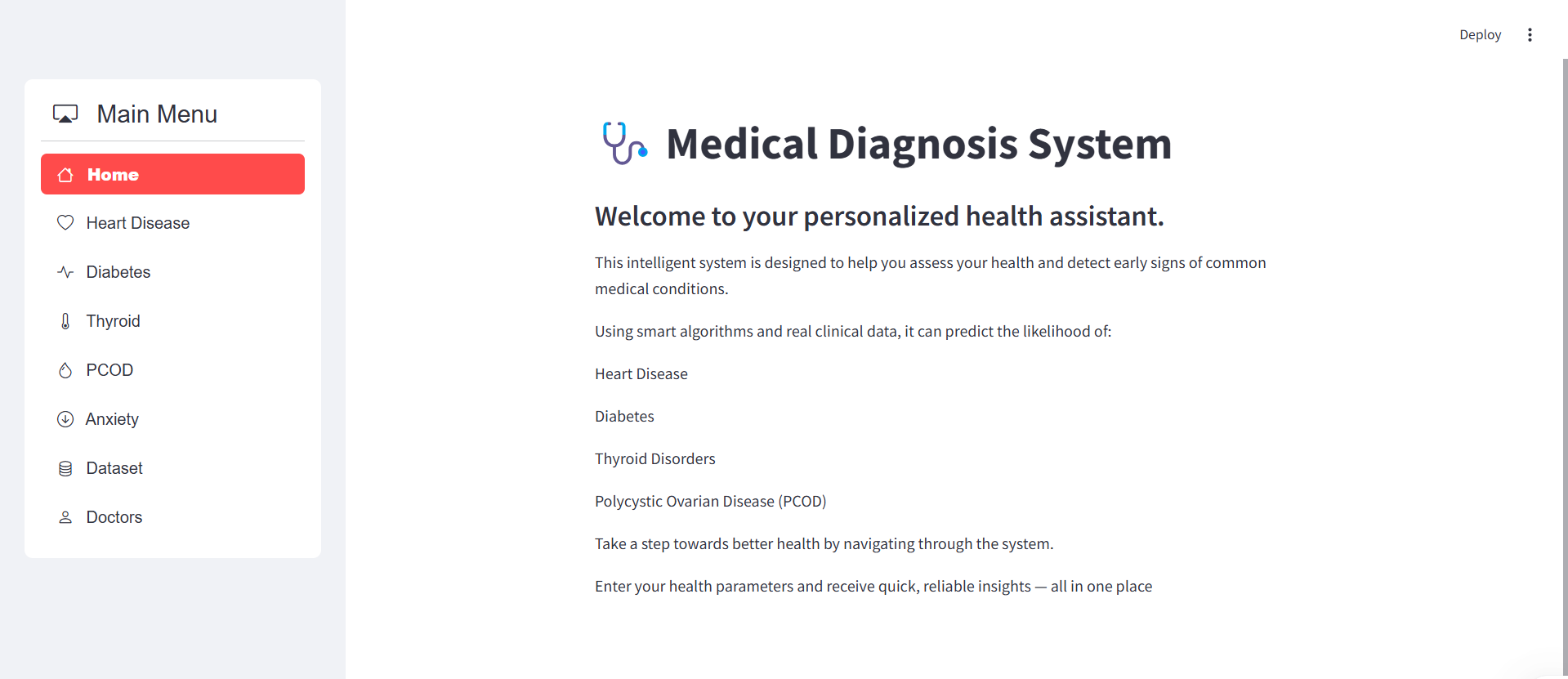
* **System Overview**

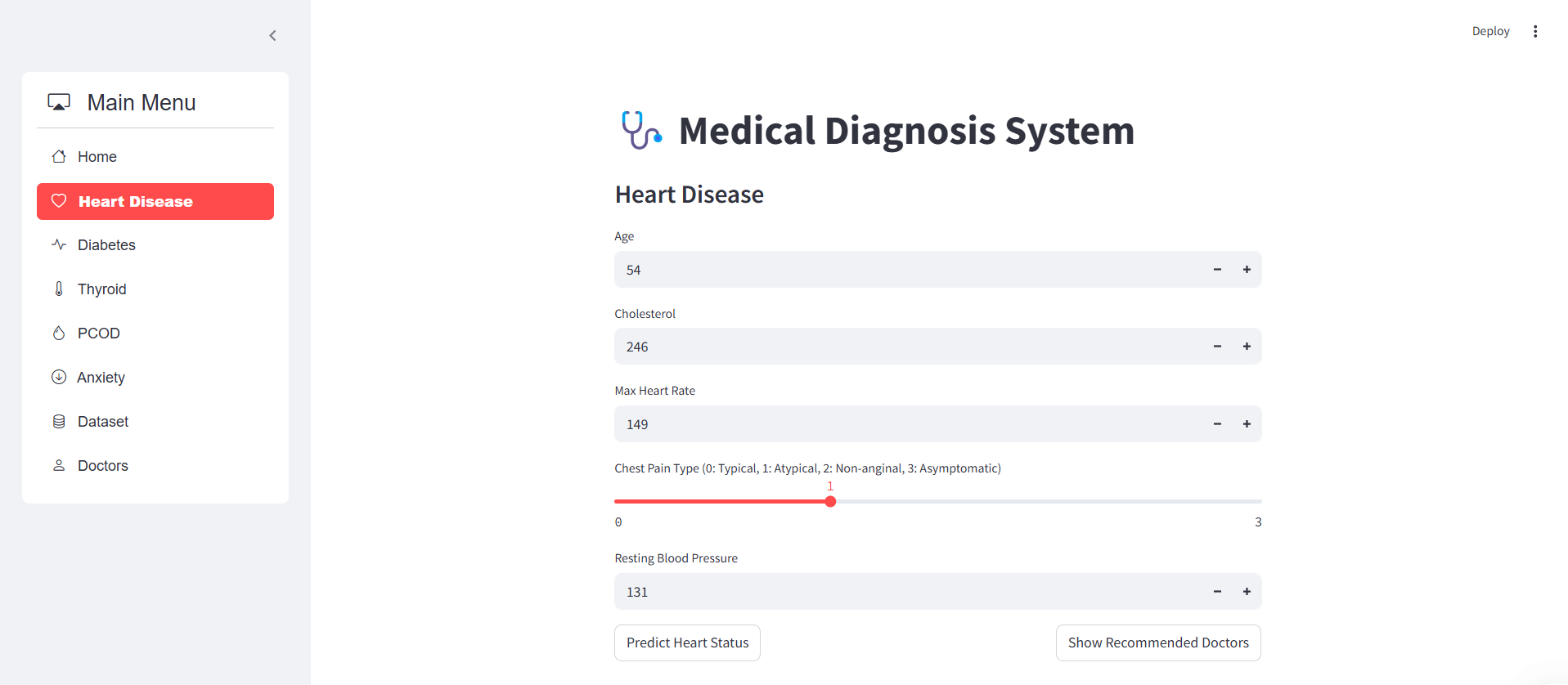
Components include:

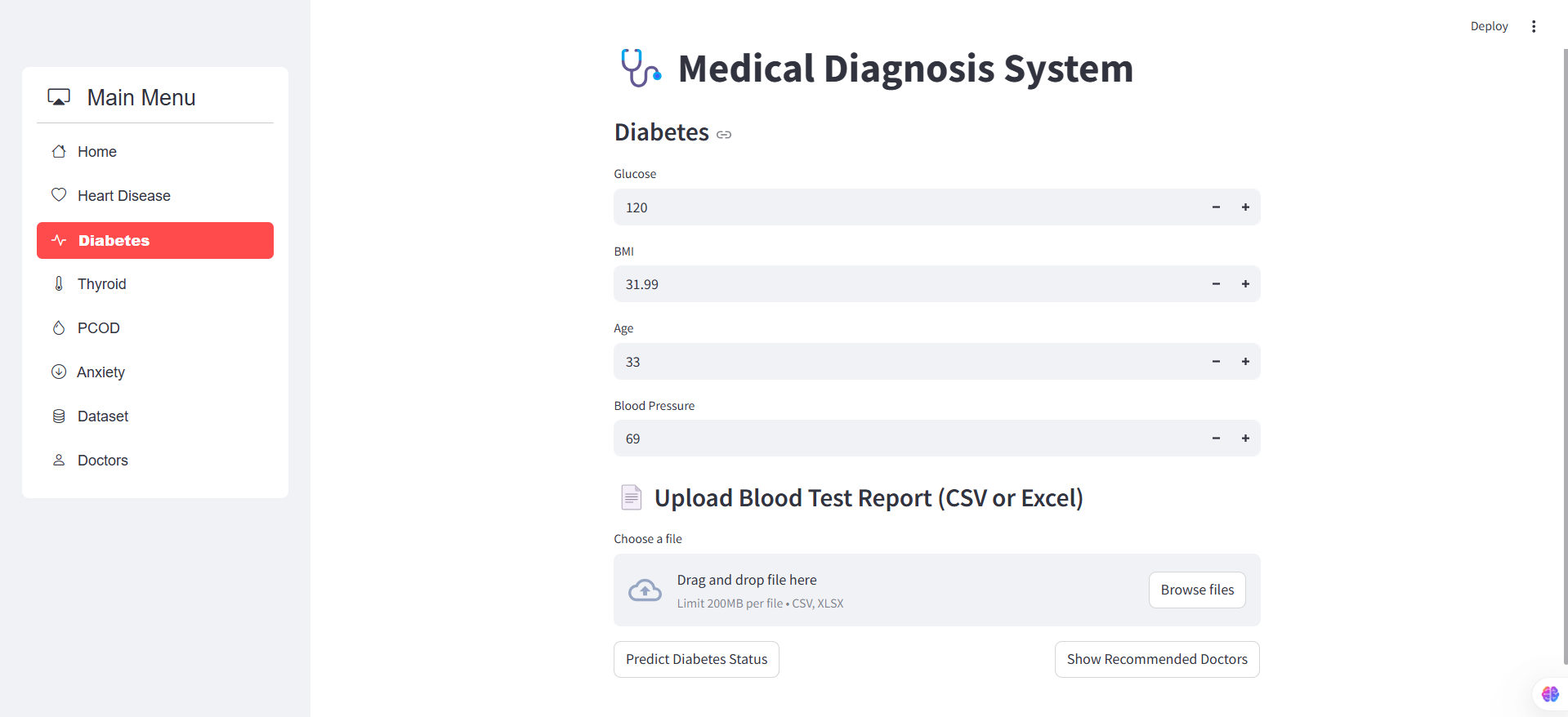
1. **User Input Interface**
2. **Fuzzy Inference Engine**
3. **Prediction Output Display**
4. **Database Access Layer**

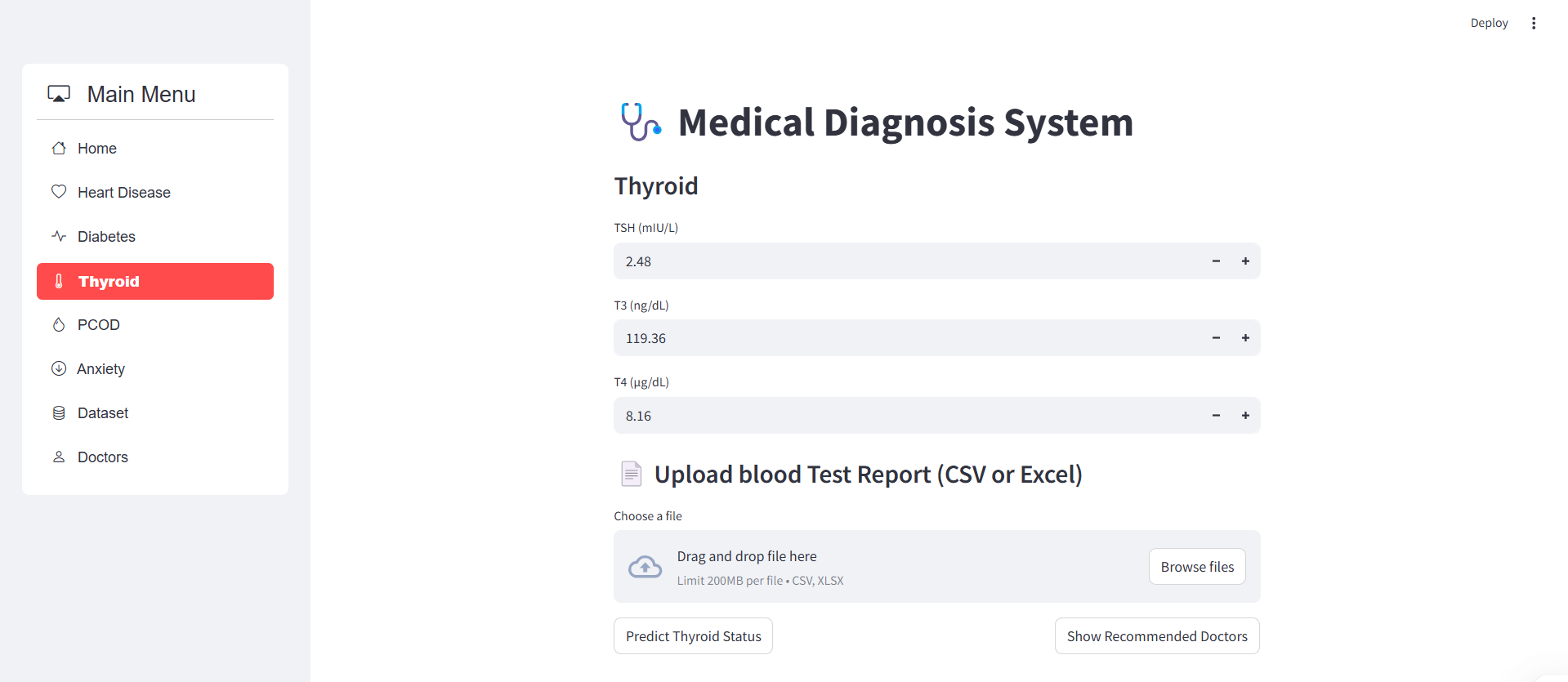
The modular design supports future expansion and integration with machine learning models.

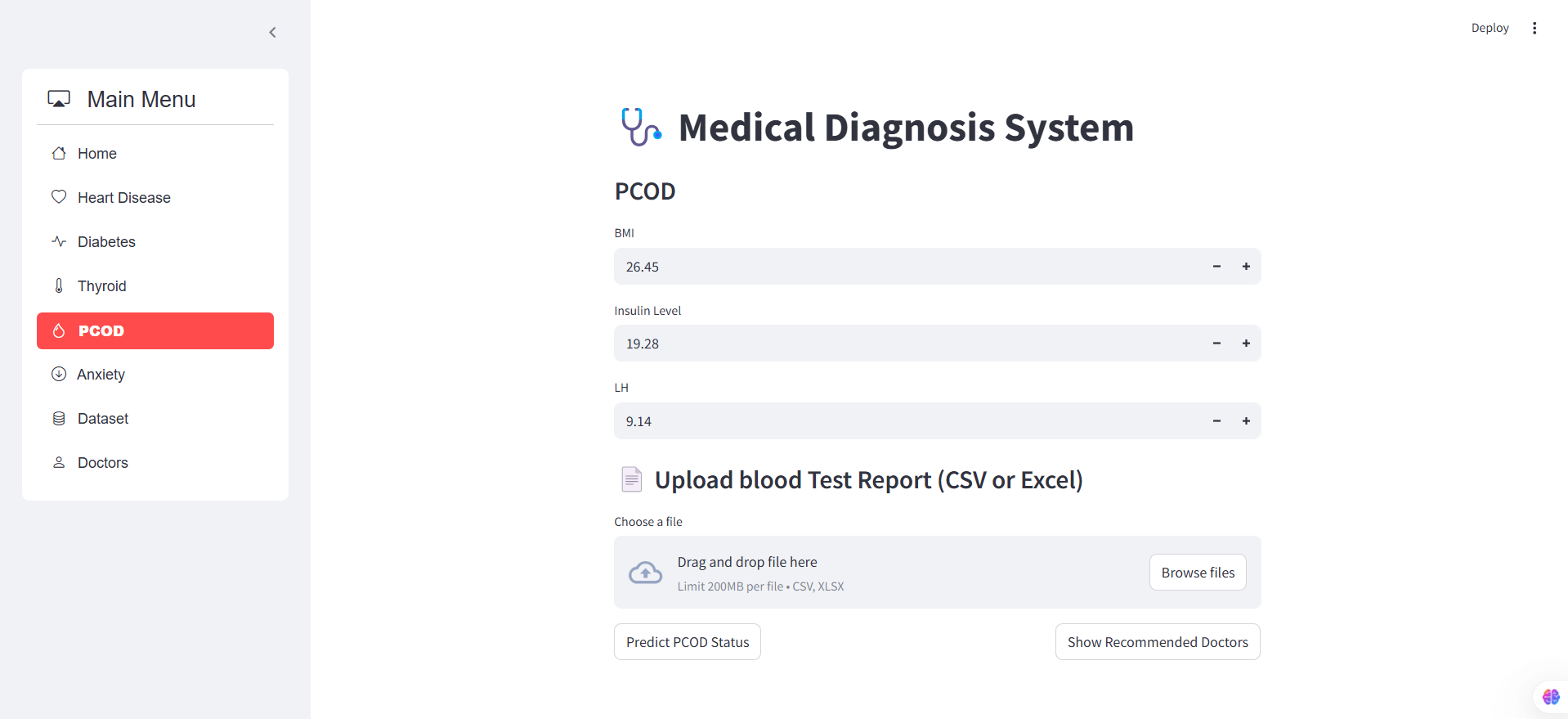
* **Screenshots of the user interface**

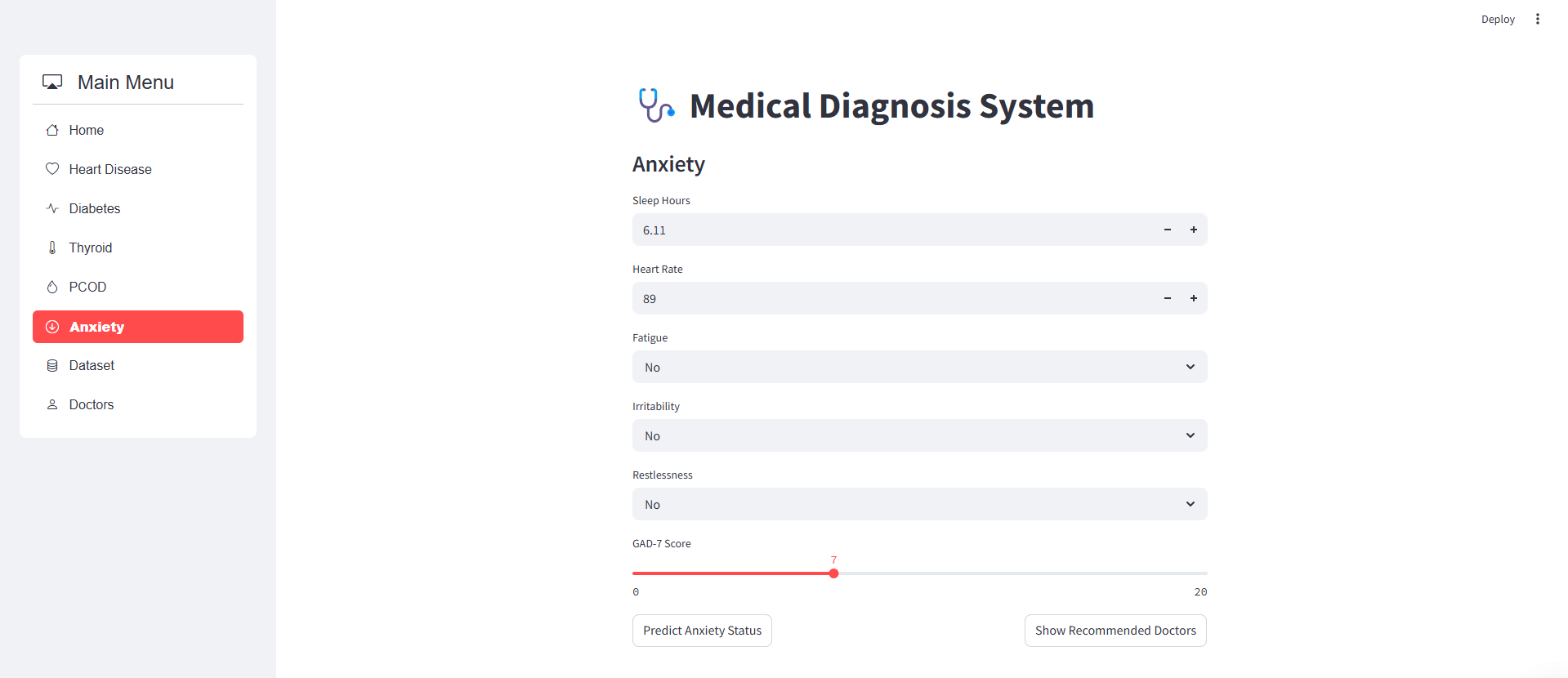
****

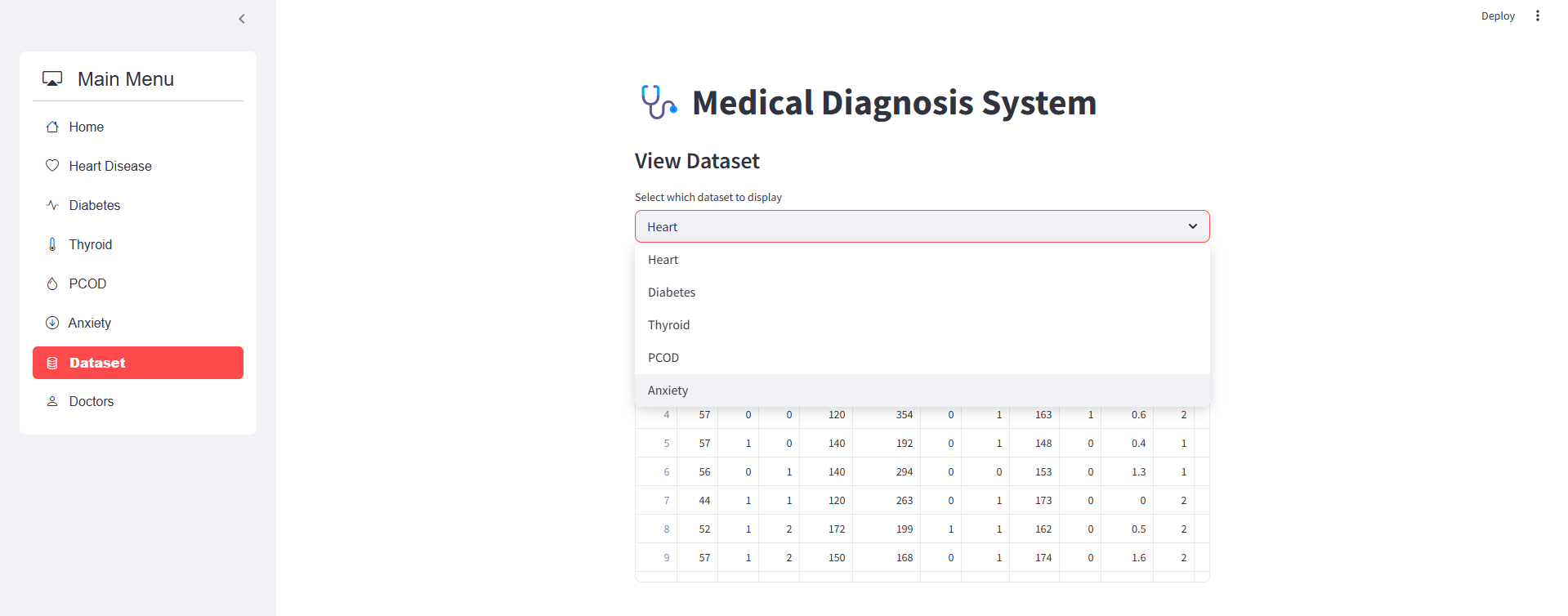
****

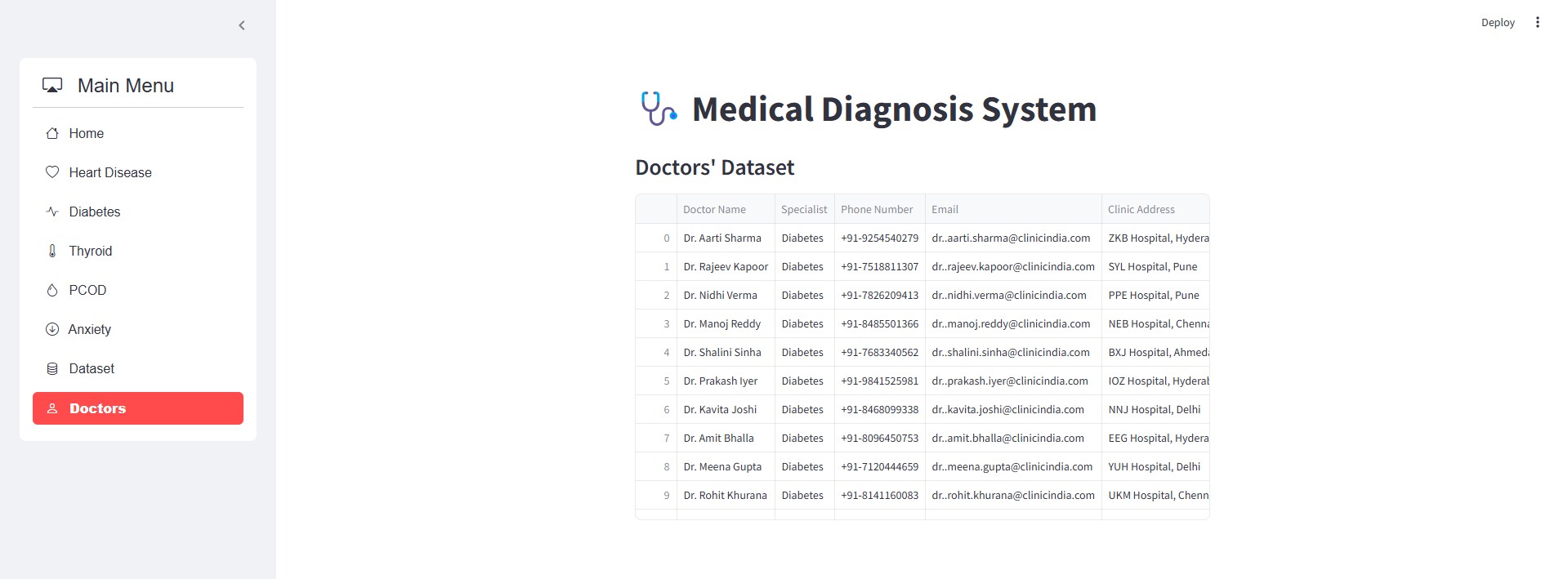
****

****

****

****

****

****

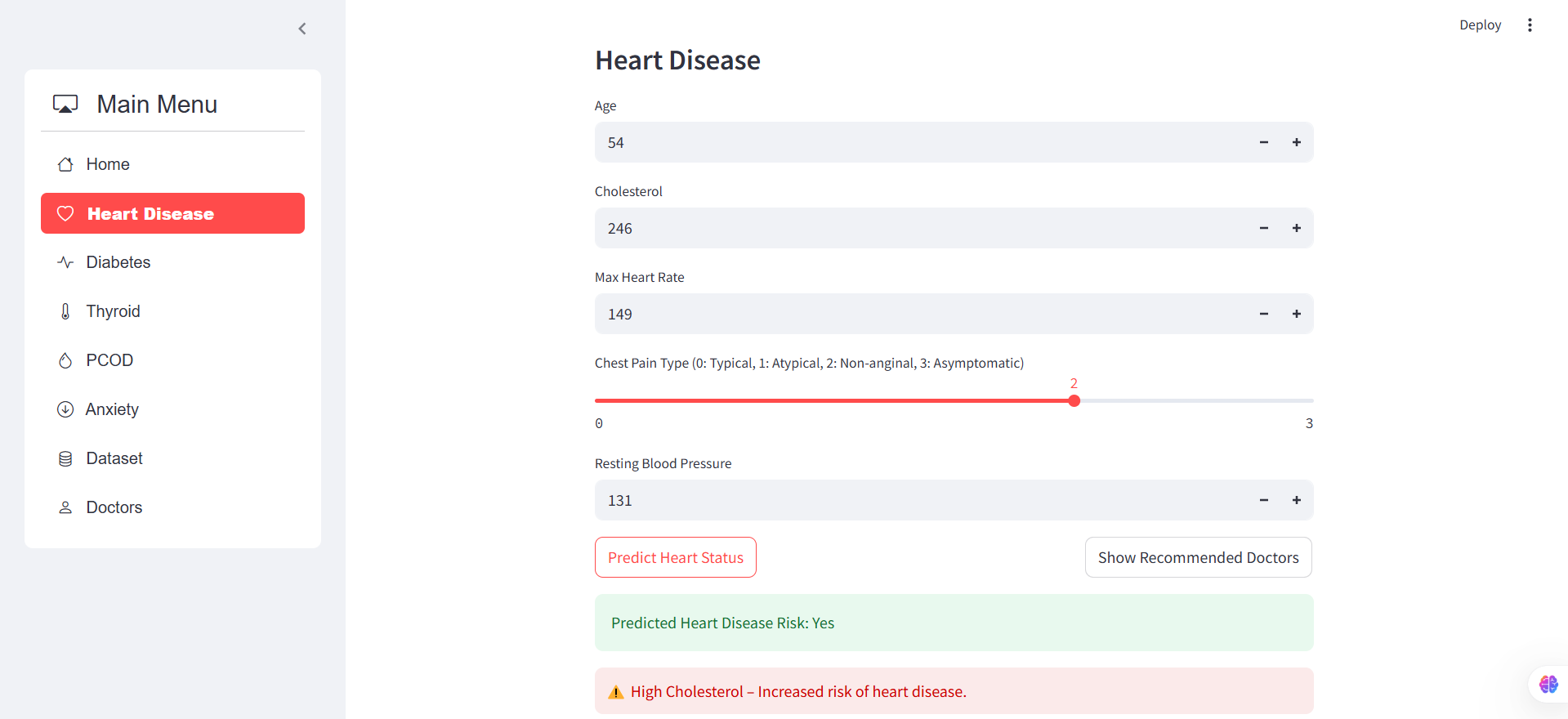
* **Testing & Results**

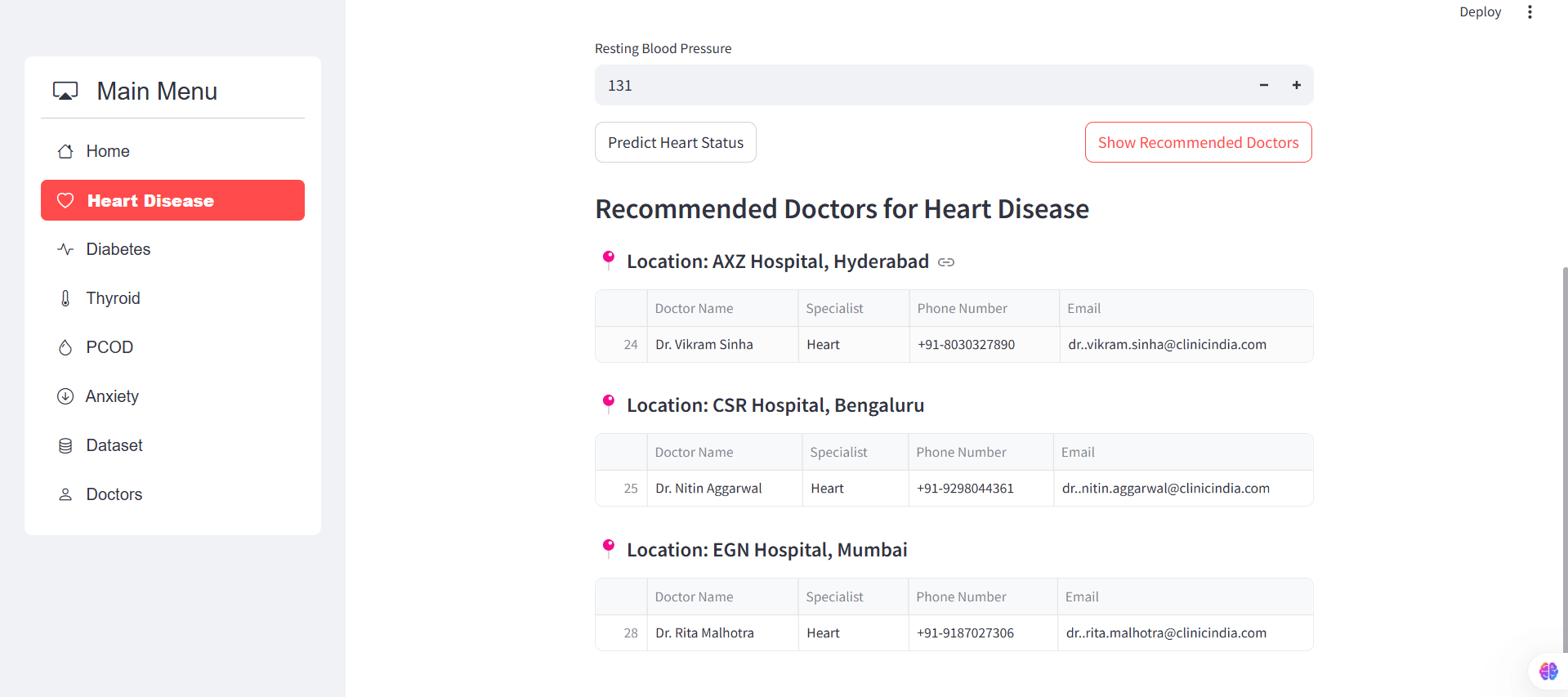
To ensure the reliability and accuracy of the medical diagnosis system, several testing strategies were employed throughout development. The system predicts five health conditions—**Heart Disease**, **Diabetes**, **Thyroid Disorders**, **PCOD**, and **Anxiety**—using Python-based fuzzy logic models.

* **Testing Strategies**
* **Unit Testing:**  
  Individual components such as data preprocessing functions, fuzzy logic modules, and input validation routines were tested independently to verify correctness and robustness.
* **Integration Testing:**  
  Ensured smooth interaction between modules, including the user interface, fuzzy inference engine, and database layer.
* **System Testing:**  
  The complete system was tested end-to-end to validate predictions across all five medical conditions. Multiple test scenarios were used, including normal, borderline, and extreme input cases.
* **Data Validation Testing:**  
  The datasets from **UCI** and **Kaggle** were cross-checked for missing or inconsistent values. Preprocessing steps were tested to ensure clean and normalized input to the fuzzy logic models.
* **Results**

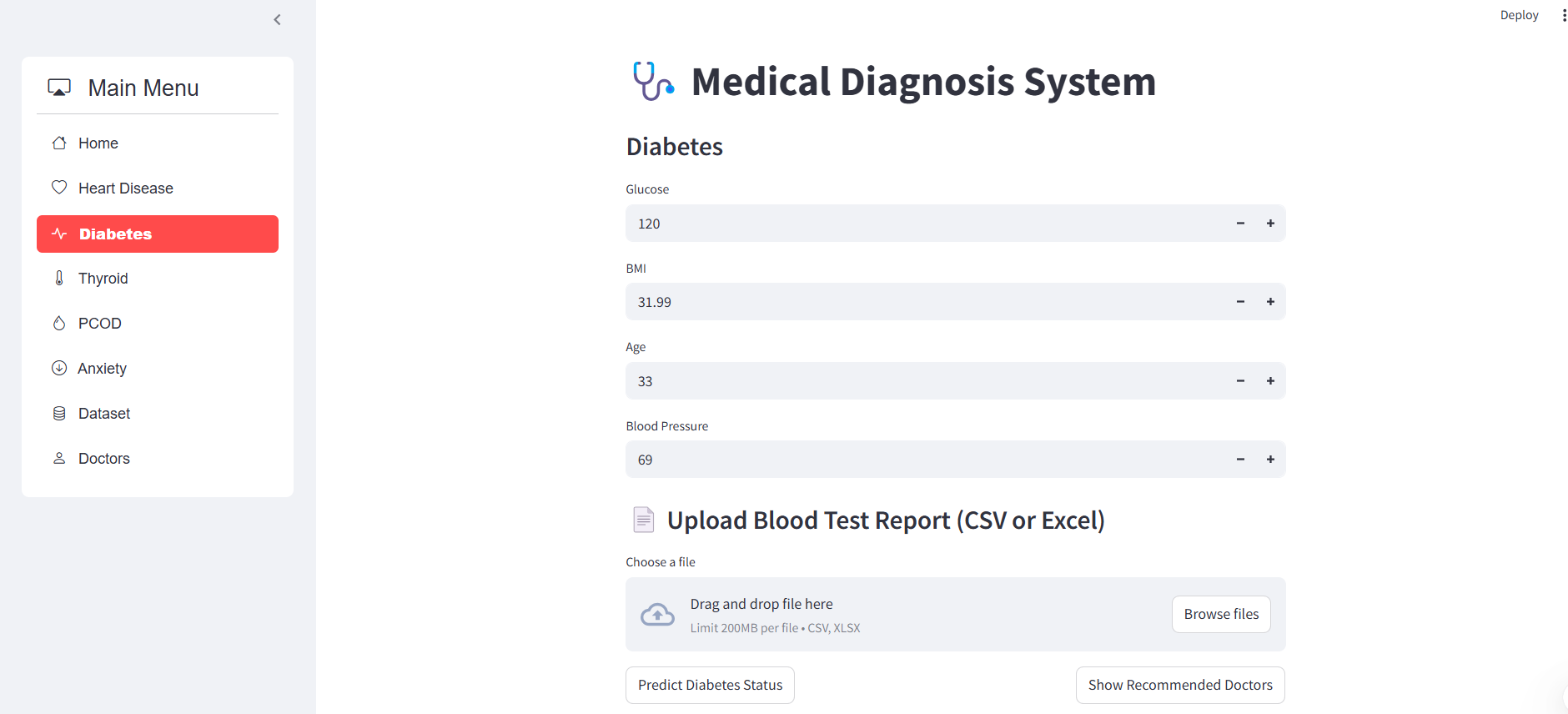
1. Heart Disease

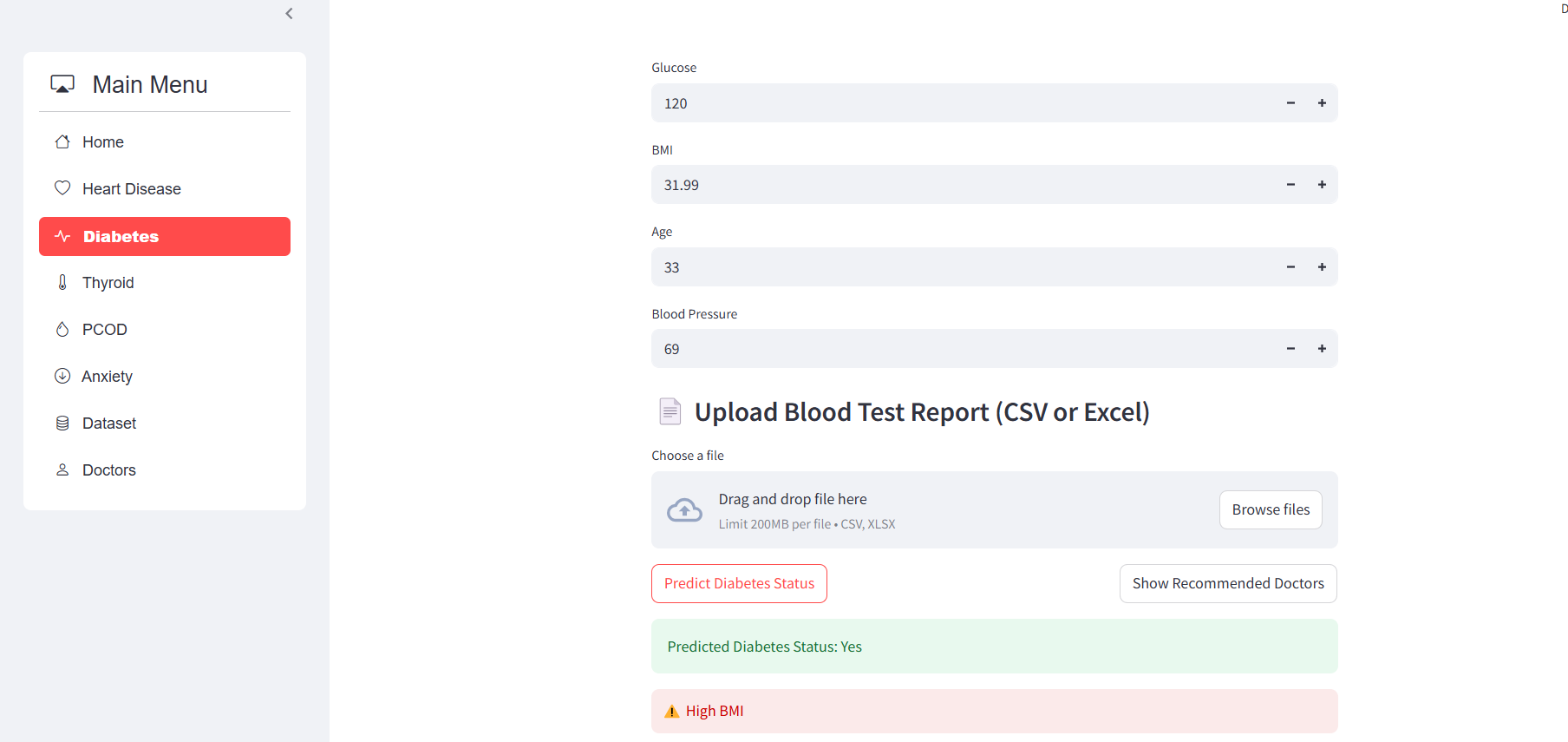




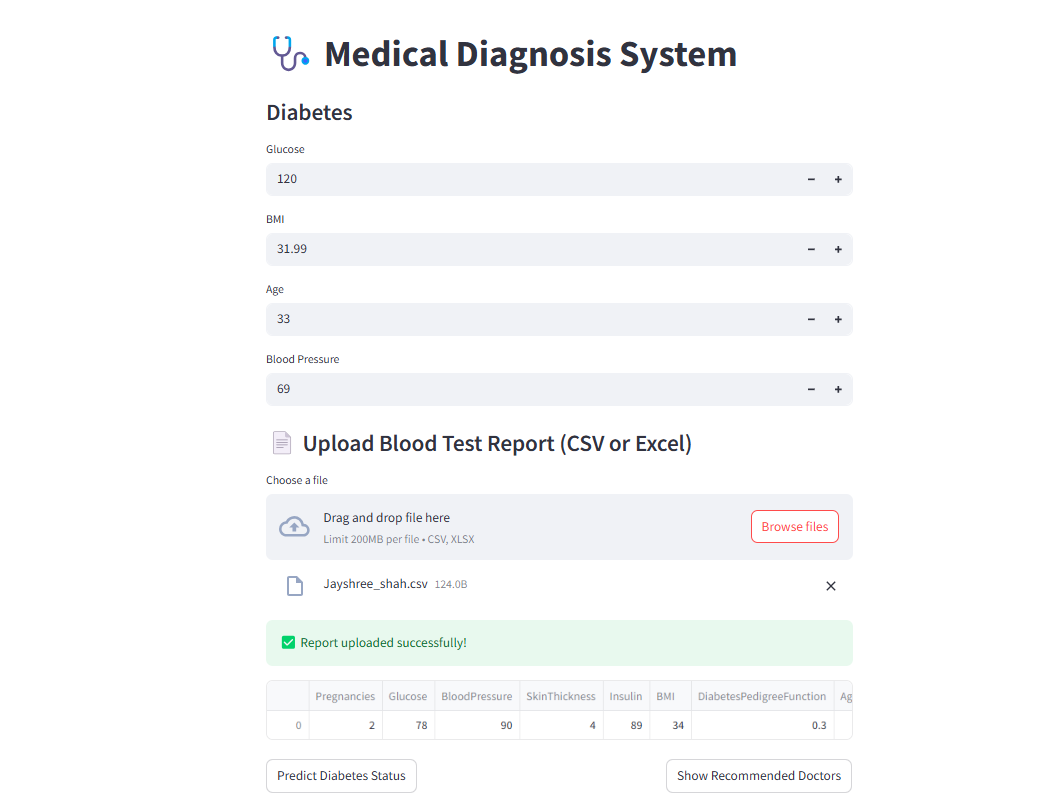


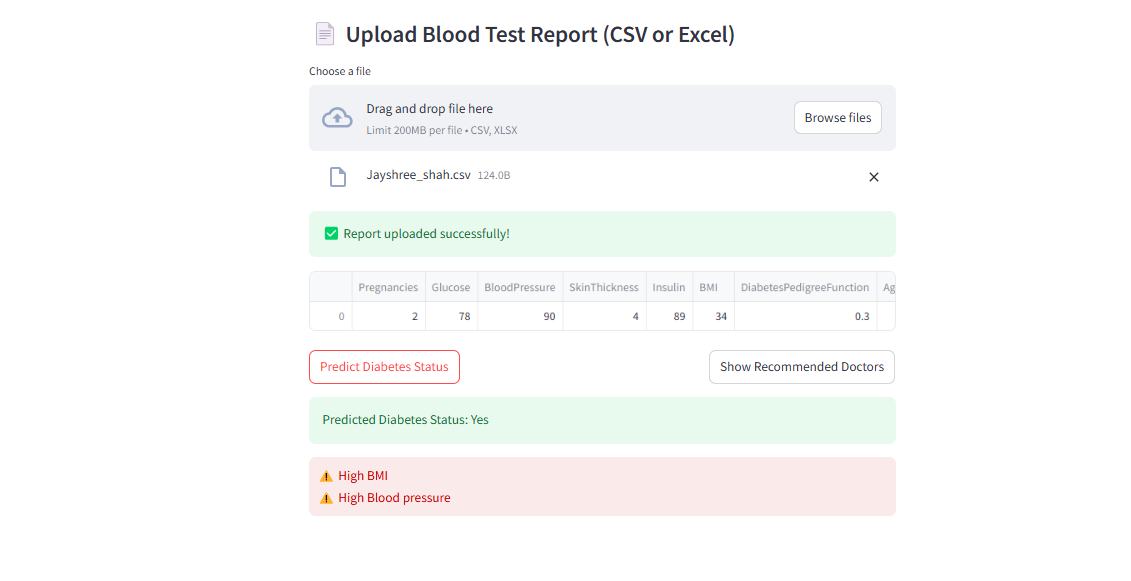
1. Diabetes





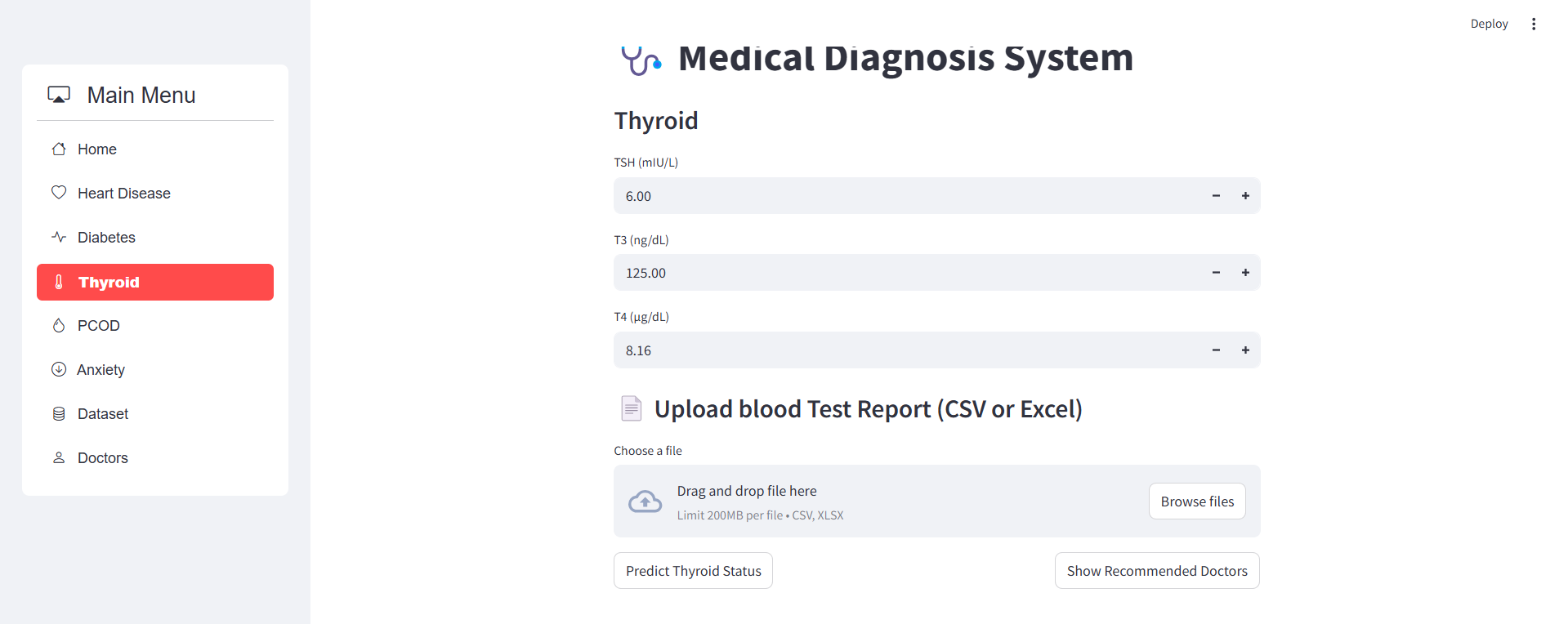
With Report Upload

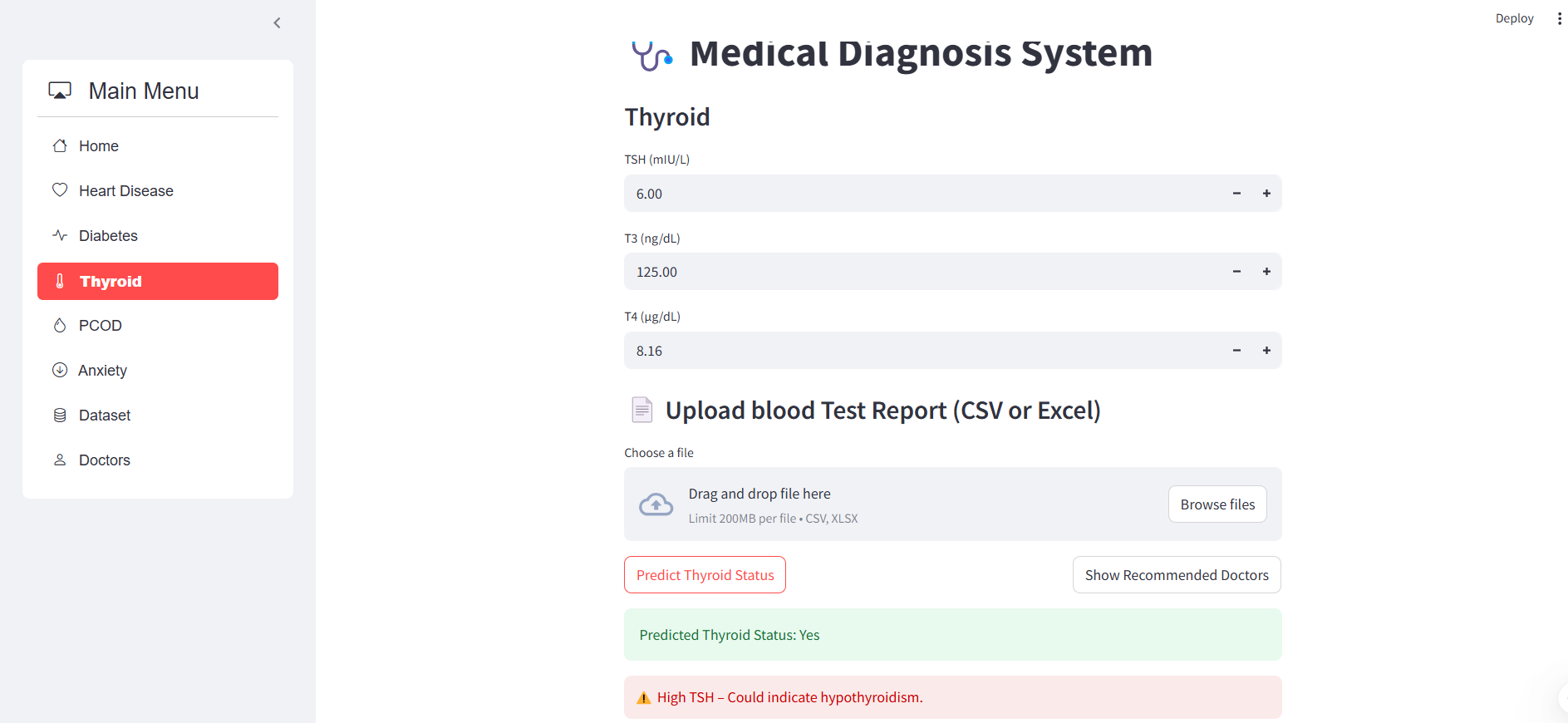




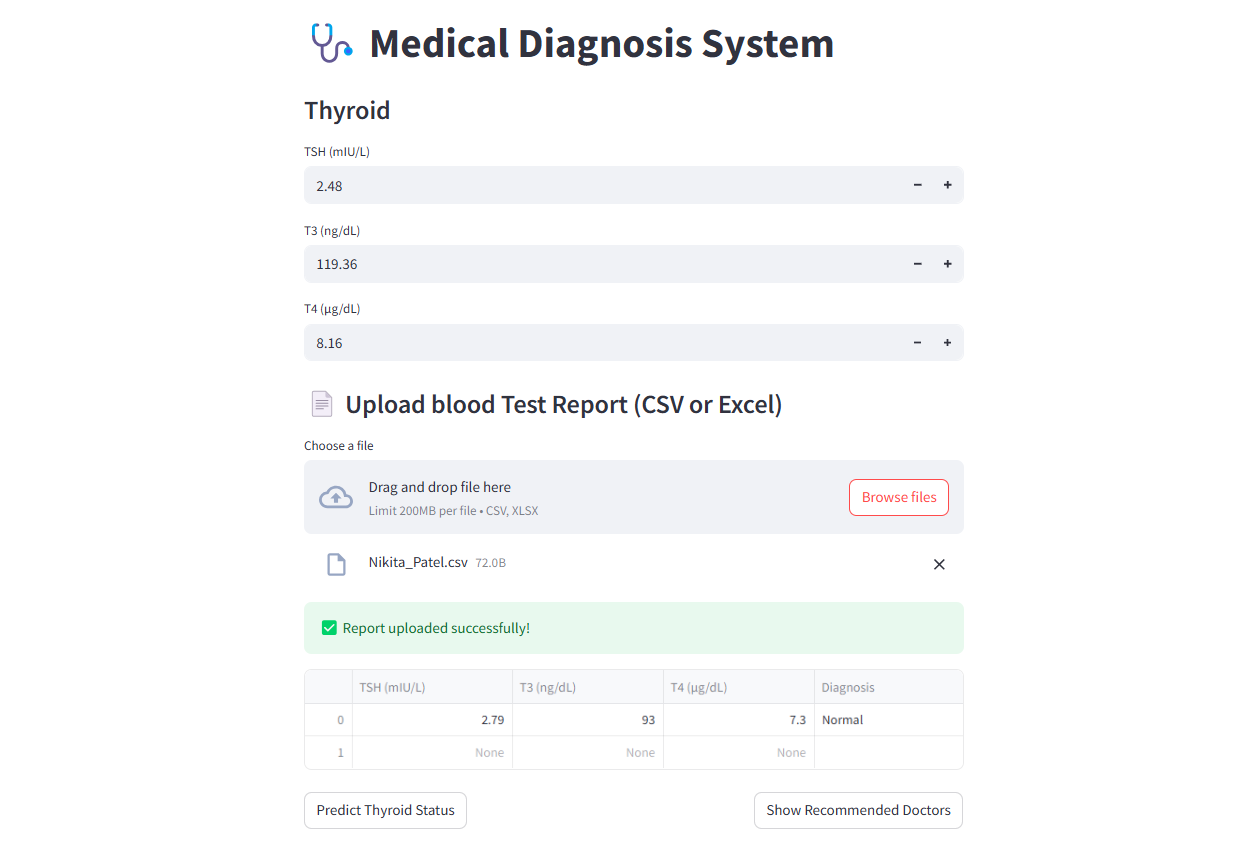


1. Thyroid

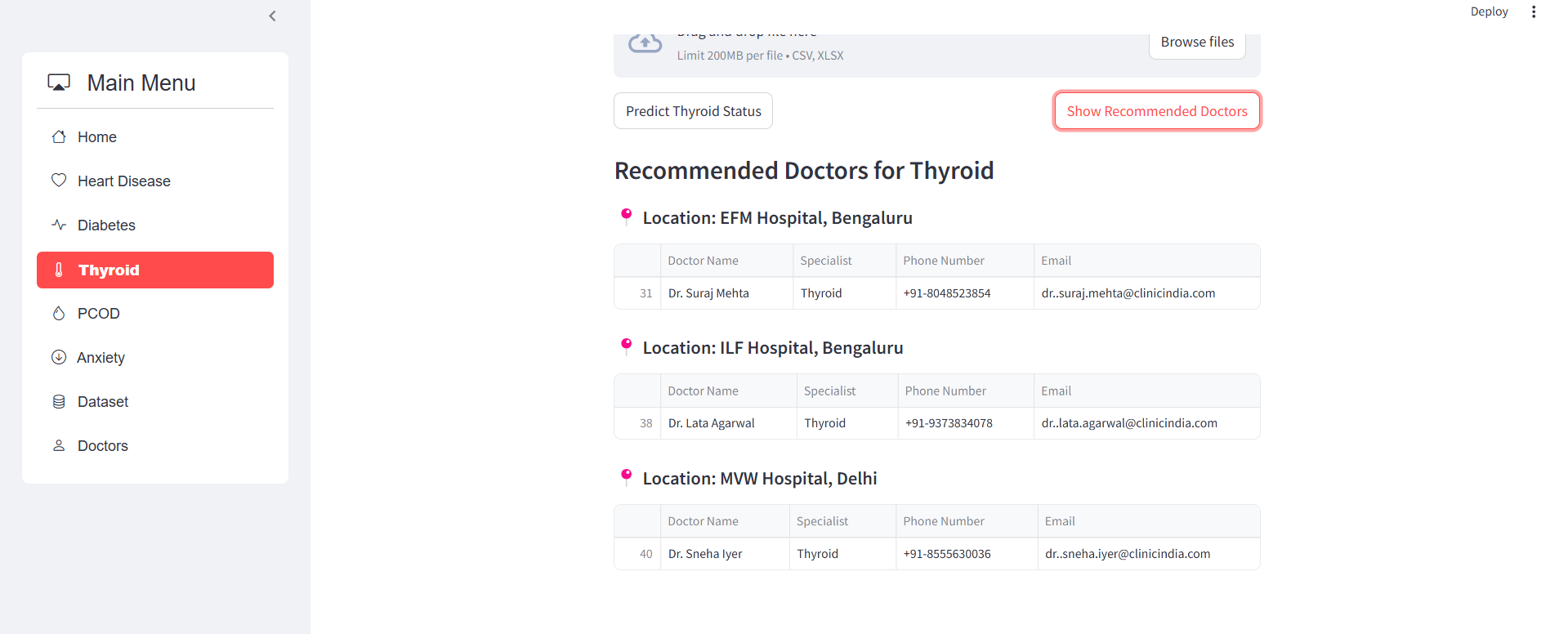




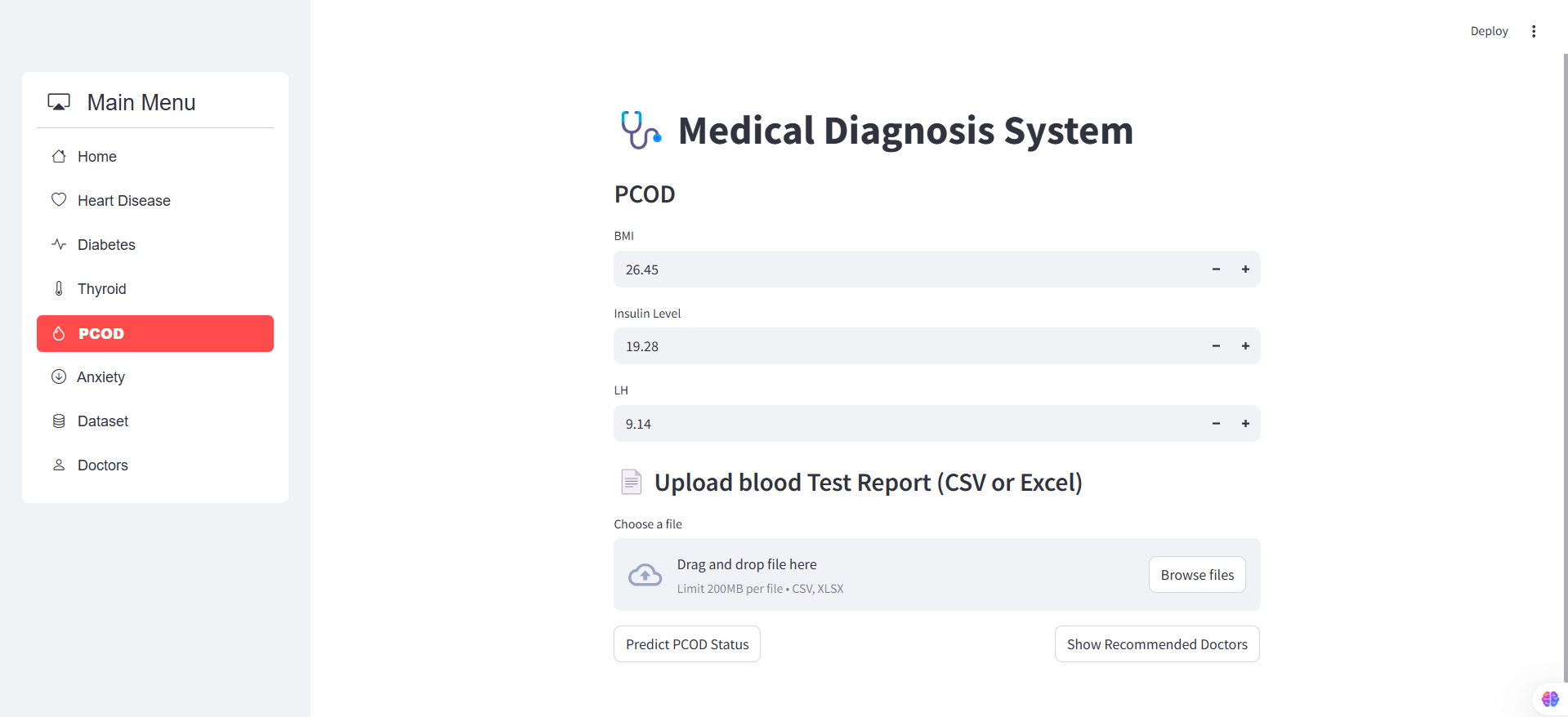
With Report Upload

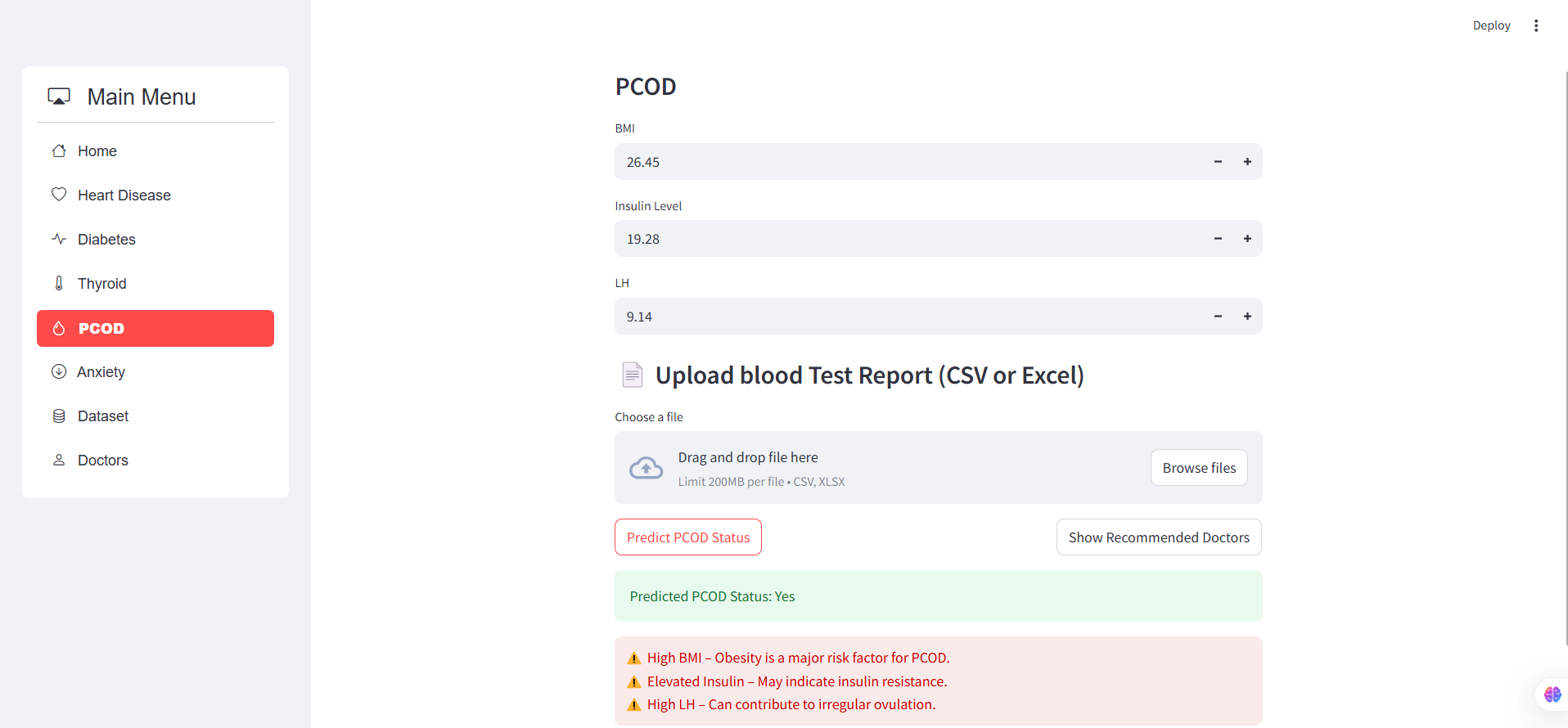




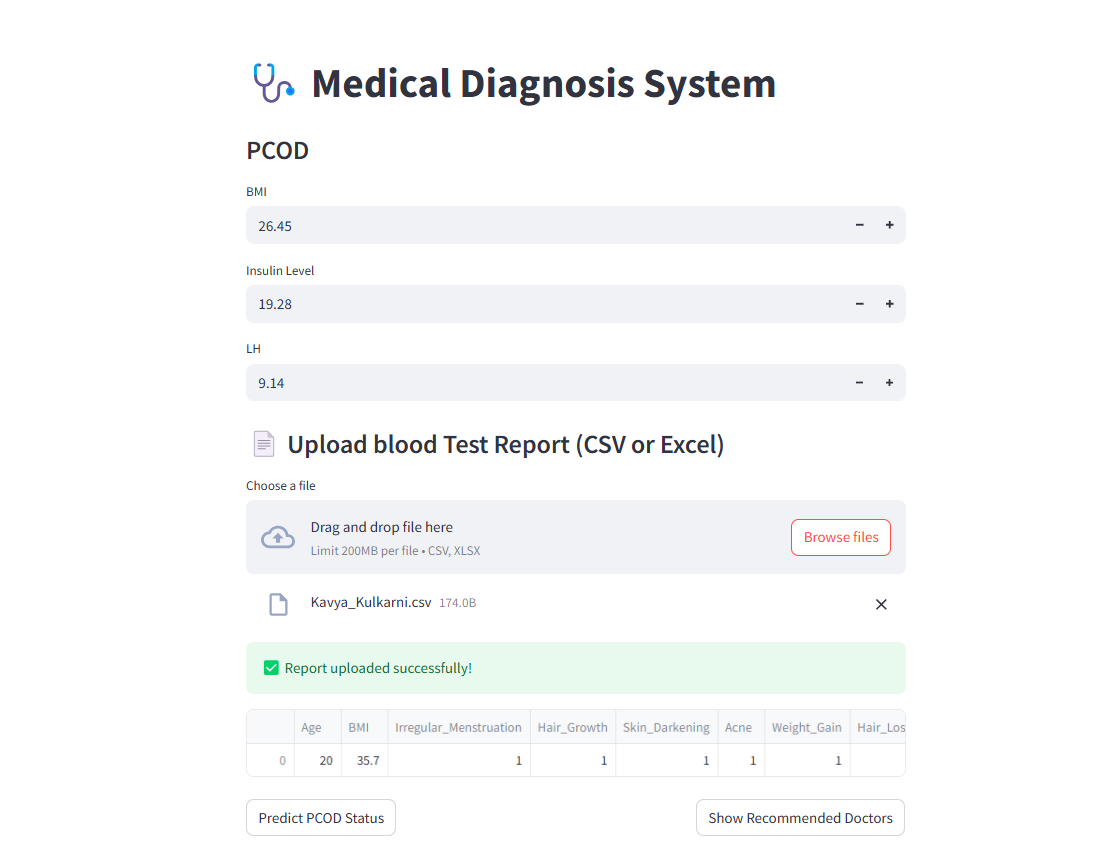


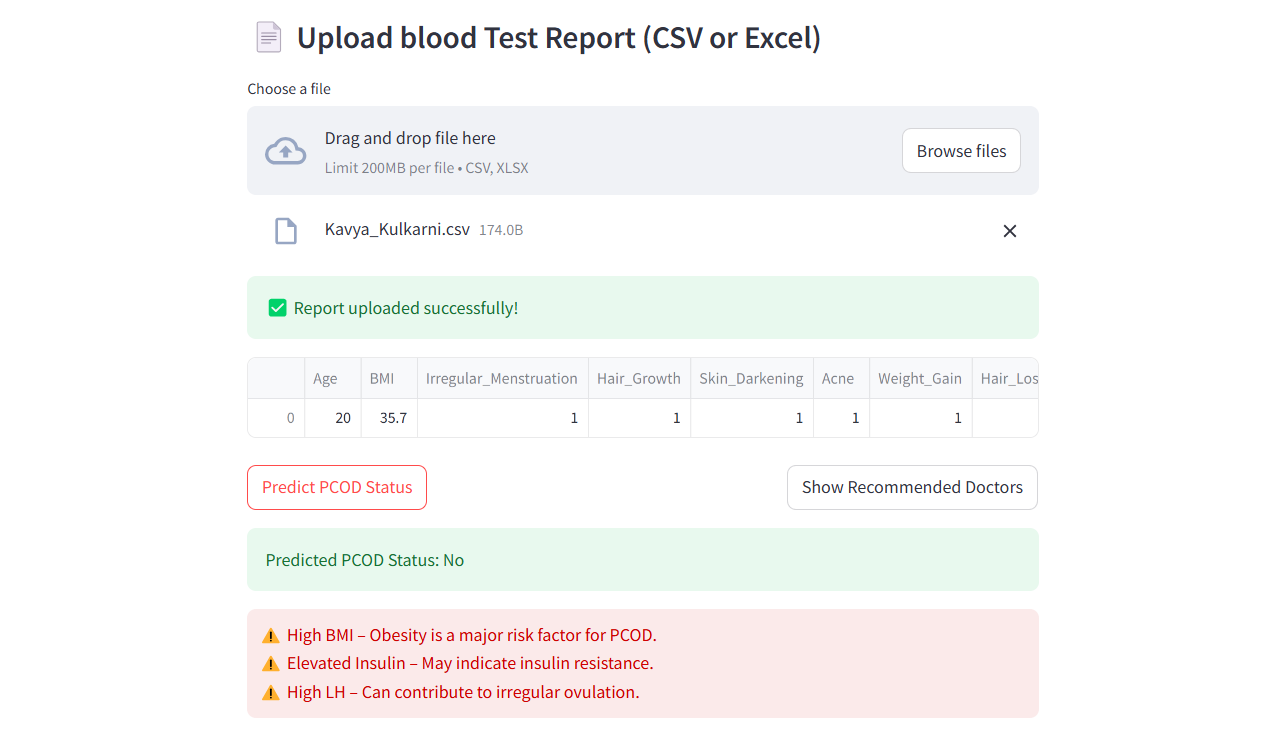
1. PCOD

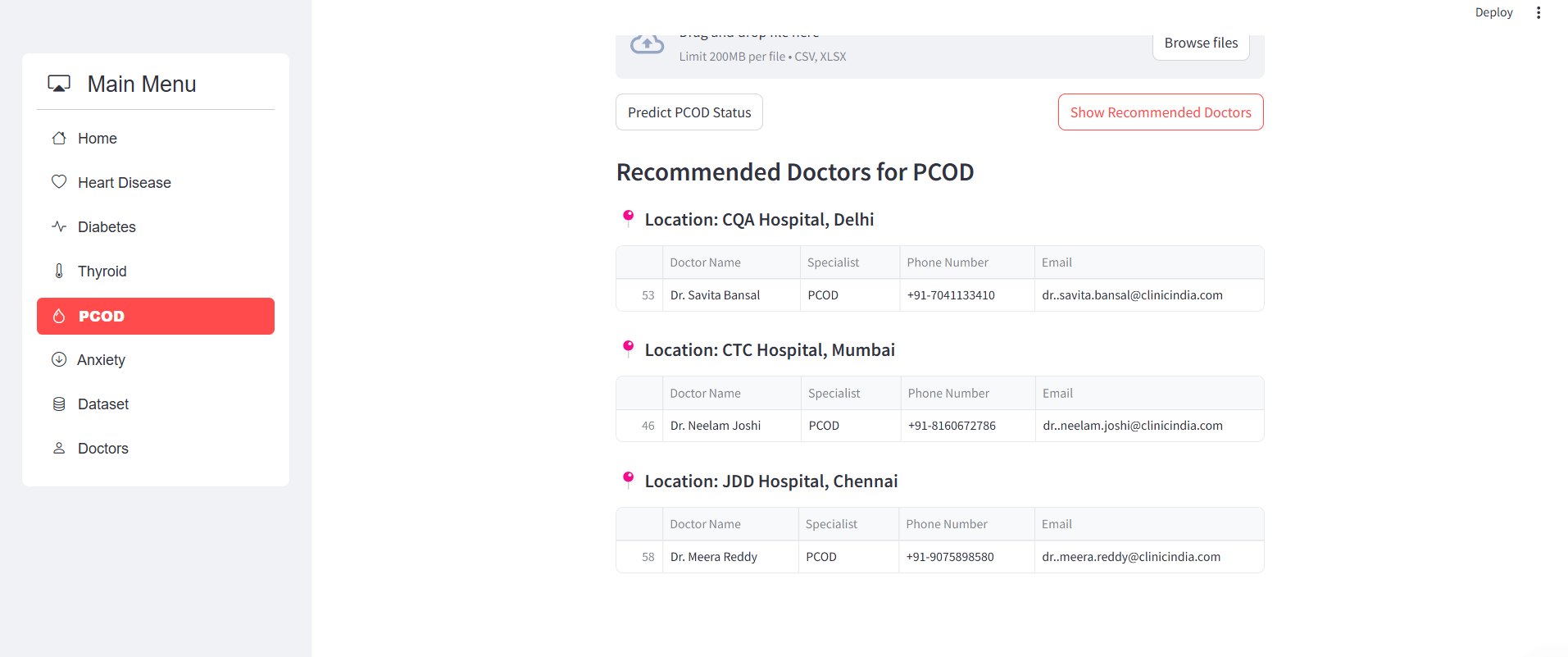




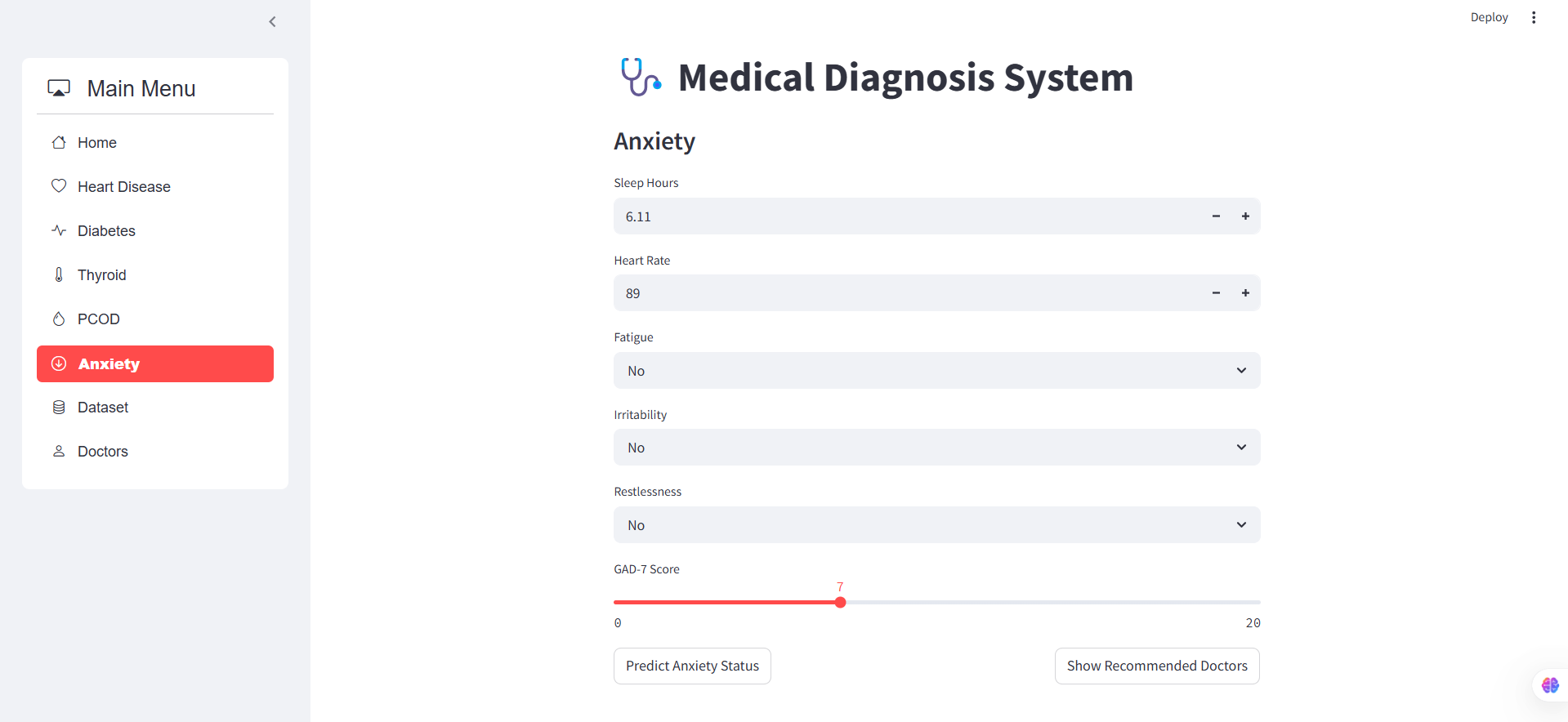
With Report Upload

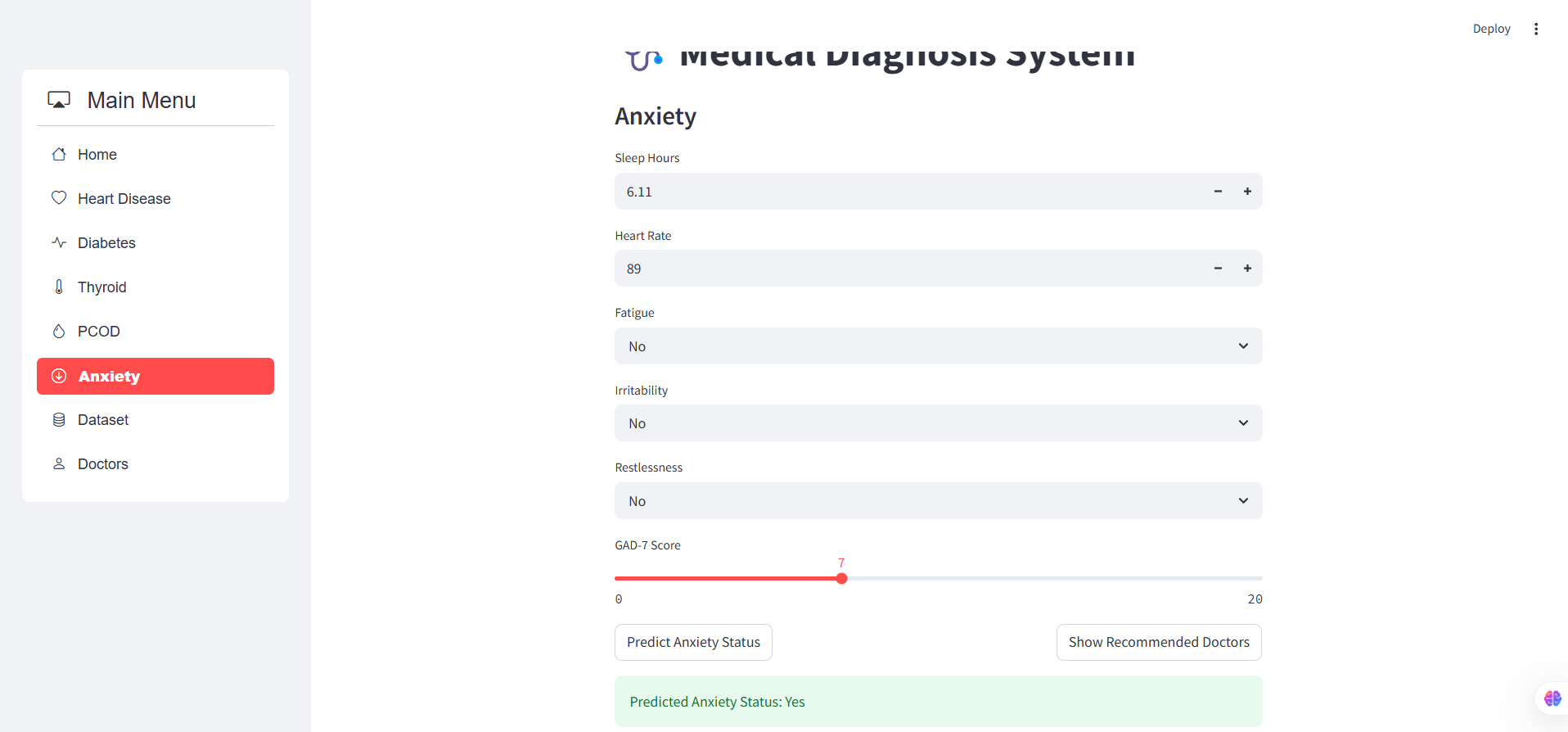


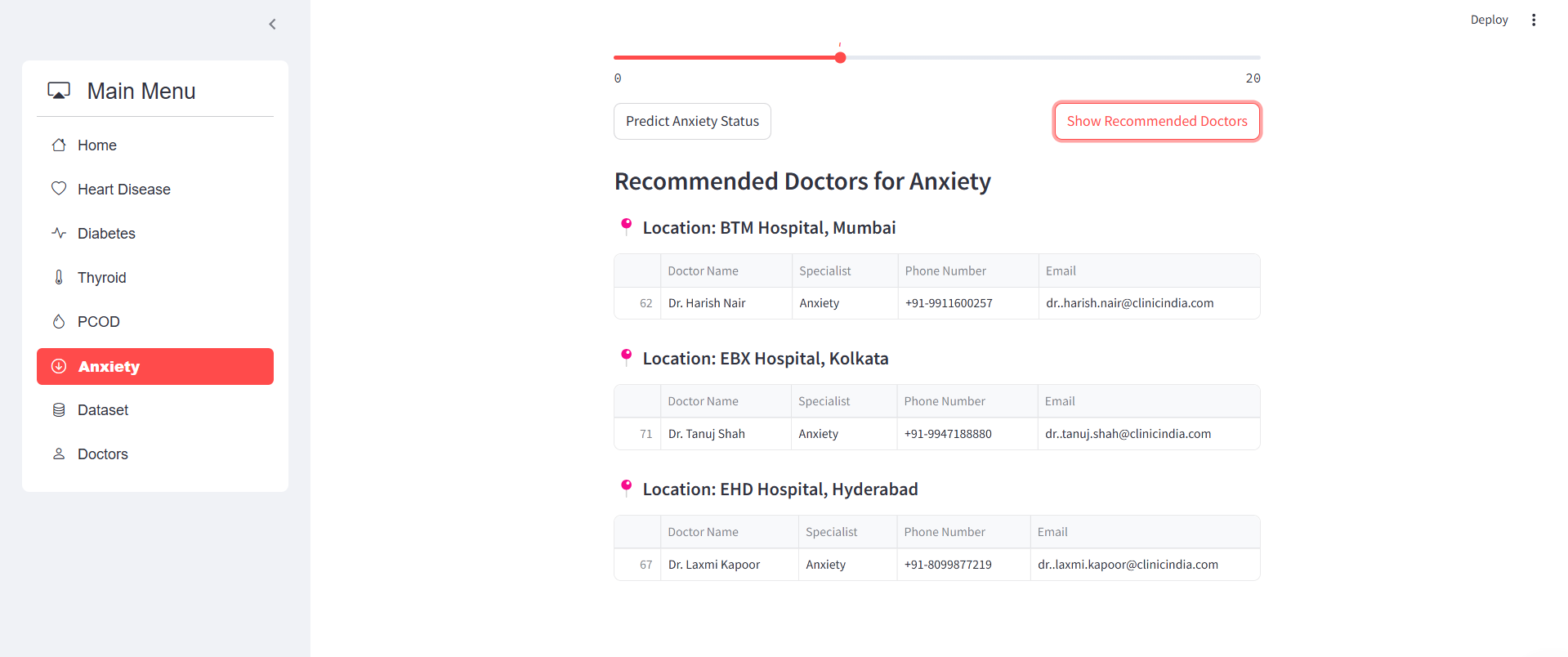




1. Anxiety







* **Discussion & Challenges**
* **Problems Faced and Solutions**
* **Data Issues:** Public datasets were often incomplete or imbalanced, especially for PCOD and anxiety.  
  **Solution:** Applied data cleaning, imputation, and oversampling (e.g., SMOTE) to improve dataset quality.
* **Multi-Disease Fuzzy Logic Integration:** Each condition required unique fuzzy rules and parameter ranges.  
  **Solution:** Developed separate fuzzy systems for each disease and integrated them under a common interface.
* **Overlapping Symptoms:** Shared symptoms caused diagnostic confusion.  
  **Solution:** Introduced rule-based prioritization to differentiate conditions.
* **Accuracy Limitations:** Pure fuzzy logic had constraints in achieving high precision.  
  **Solution:** Explored hybrid approaches with machine learning for future enhancement.
* **Possible Improvements**
* **Real-Time Data:** Integrate wearable or live health data for better predictions.
* **More Diseases:** Expand coverage to include additional conditions.
* **Better UI:** Develop a user-friendly interface using Flask/Django.
* **Hybrid Models:** Combine fuzzy logic with ML models for improved accuracy.
* **Clinical Testing:** Validate system outcomes with healthcare professionals.
* **Conclusion**
* **Summary of Findings**

We developed a medical diagnosis system using Python and fuzzy logic that predicts five major health conditions: diabetes, heart disease, thyroid disorders, PCOD, and anxiety. The system utilizes publicly available datasets to interpret user symptoms through a rule-based fuzzy inference approach. Our results show that fuzzy logic is effective in handling the uncertainties and overlaps often present in medical diagnosis, providing a practical tool for early screening and awareness.

* **Future Scope of the Project**

The project offers several opportunities for enhancement, both from a technical and user experience perspective:

* **Video Consultation:** Integrate video conferencing capabilities to connect patients with healthcare professionals directly from the platform.
* **Diet and Lifestyle Plans:** Offer AI-generated personalized diet recommendations and lifestyle improvement plans based on diagnosis results.
* **Symptom Tracker:** Add daily or weekly symptom logging features for long-term monitoring.
* **Interactive Chatbot:** Implement a virtual assistant to guide users through symptom input and answer health-related queries.
* **User Profile and Reports:** Allow users to download and share detailed diagnostic reports with doctors.
* **Multi-language Support:** Enhance accessibility by adding support for regional languages and voice-based inputs.
* **References**
* Google. (n.d.). *Google search*. Retrieved April 29, 2025, from [https://www.google.com](https://www.google.com/)
* Wikipedia contributors. (n.d.). *Wikipedia, the free encyclopedia*. Wikipedia. Retrieved April 29, 2025, from [https://www.wikipedia.org](https://www.wikipedia.org/)
* Kaggle. (n.d.). *Datasets for data science and machine learning*. Retrieved April 29, 2025, from [https://www.kaggle.com](https://www.kaggle.com/)
* Mayo Clinic. (n.d.). *Symptom Checker*. Retrieved April 29, 2025, from <https://www.mayoclinic.org/symptom-checker/select-symptom/itt-20009075>
* Patient.info. (n.d.). *Symptom Checker*. Retrieved April 29, 2025, from <https://patient.info/symptom-checker>
* **Appendix**

