

# ***HACKNITE'25***

*TEAM NAME : CODE BYTES*

*TEAM CODE : HACK205B*

*TEAM MEMBERS NAME: MANYA M*

*MEGHANA T B*

## **PROBLEM NAME: HEALTHCARE**

### **• PROBLEM STATEMENT:**

*Develop an advanced, explainable AI framework capable of the early, accurate, and causal detection of potential ADRs(Adverse Drug Reactions) by integrating and analyzing complex, heterogeneous data sources including early post-market surveillance, unstructured patient feedback, and evolving social media discourse, while overcoming challenges of data sparsity, bias, and the need for transparent, clinically actionable insights.*

### **ADR- ADVERSE DRUG REACTION**

- It is a harmful response to medication that at normal.
- ADR may range from mild symptoms to life-threatening conditions.
- The phrase "Prevention is better than cure" perfectly aligns with this problem statement because detecting adverse drug reactions (ADRs) early is a preventive approach to safeguarding patient health. Instead of waiting for serious side effects to occur and then trying to treat them, the goal is to proactively identify risks using advanced AI.

### ***Simple explanation of the problem statement:***

We want to create a smart AI system that can **find harmful side effects of medicines early.**

- The data from different places, like reports from patient feedback and even social media posts.
- Solve tricky issues like unfair biases in data, data sparsity and making sure the results are easy for doctors to trust and act on.
- The goal is to help healthcare professionals.

### **CREATIVITY:**

- **Using Many Data Sources:** Instead of looking at just one type of data, they combined reports from

doctors, patient feedback, and social media to get a full picture of medicine side effects.

- **Finding the Real Cause:** They made sure the system doesn't just show patterns but tells if the medicine is

truly causing the side effect, not just a coincidence.

- **Easy to Understand:** The system explains its findings clearly so doctors can trust and use it quickly.
- **Fixing Missing Data:** They added tools to handle cases where there isn't enough data, so the system

still works well.

- **Removing Bias:** They made sure the results are fair and not influenced by uneven or faulty data.
- **Quick and Useful:** The system provides results that doctors can act on immediately to prevent harm.

## **Explanation of code:**

Enhanced Disease Diagnosis System, the system is designed to collect user inputs (e.g., symptoms, age, and lifestyle risk factors), detect possible diseases based on the symptoms, and display actionable insights, including disease severity, prevention tips, and diagnosis history.

### **1. Purpose**

The primary purpose of this code is to serve as a web-based tool for diagnosing diseases based on user inputs (symptoms and lifestyle risks). The tool provides:

- Disease detection and severity classification.
- Tailored health tips and insights.
- Visualization of disease severity using a bar chart.
- Tracking of diagnosis history for future reference.

## **2. Functional Overview**

### **Input Fields**

- **Name Field (id="name"):** Collects the user's name for personalized interactions.

- **Symptoms Field (id="symptoms"):** Accepts a comma-separated list of symptoms (e.g., "Fatigue, Fever, Sweating").
- **Age Field (id="age"):** Captures the user's age for context in the diagnosis.
- **Gender Dropdown (id="gender"):** Allows the user to specify their gender as Male, Female, or Other.
- **Lifestyle Risk Factors (id="risk"):** Collects additional health-related inputs, such as smoking or high blood pressure.

### Buttons

- **Diagnose Button (id="diagnoseBtn"):** Triggers the analysis process to detect a disease and display the results.
- **Show History Button (id="showHistoryBtn"):** Displays a history of previously detected diseases and the corresponding symptoms.

### Outputs

- **Result Div (id="result"):** Displays the detected disease name, severity, and brief information about the disease.
- **Health Tips Div (id="healthTips"):** Provides actionable prevention or recovery tips specific to the detected disease.
- **Severity Chart (id="severityChart"):** Uses to display a bar chart representing the severity of the detected disease.
- **History Div (id="history"):** Lists previously diagnosed diseases with symptoms for user reference.

Enhanced Disease Diagnosis System. The system is designed to collect user inputs (e.g., symptoms, age, and lifestyle risk factors), detect possible diseases based on the symptoms, and display actionable insights, including disease severity, prevention tips, and diagnosis history.

## **1. Purpose**

The primary purpose of this code is to serve as a web-based tool for diagnosing diseases based on user inputs (symptoms and lifestyle risks). The tool provides:

Disease detection and severity classification.

Tailored health tips and insights.

Visualization of disease severity using a bar chart.

Tracking of diagnosis history for future reference.

## **2. Functional Overview**

### **Input Fields**

Name Field (id="name"): Collects the user's name for personalized interactions.

Symptoms Field (id="symptoms"): Accepts a comma-separated list of symptoms (e.g., "Fatigue, Fever, Sweating").

Age Field (id="age"): Captures the user's age for context in the diagnosis.

Gender Dropdown (id="gender"): Allows the user to specify their gender as Male, Female, or Other.

Lifestyle Risk Factors (id="risk"): Collects additional health-related inputs, such as smoking or high blood pressure.

### **Buttons**

Diagnose Button (id="diagnoseBtn"): Triggers the analysis process to detect a disease and display the results.

Show History Button (id="showHistoryBtn"): Displays a history of previously detected diseases and the corresponding symptoms.

### **Outputs**

Result Div (id="result"): Displays the detected disease name, severity, and brief information about the disease.

Health Tips Div (id="healthTips"): Provides actionable prevention or recovery tips specific to the detected disease.

Severity Chart (id="severityChart"): Uses Chart.js to display a bar chart representing the severity of the detected disease.

History Div (id="history"): Lists previously diagnosed diseases with symptoms for user reference.

### **3. Features**

#### Symptom-to-Disease Mapping

The JavaScript object symptomToDisease maps symptoms to predefined diseases:

Pneumonia:

Symptoms: Fatigue, Fever, Sweating, Shaking Chills, etc.

Severity: High

Tips: Stay hydrated, rest adequately, and avoid cold exposure.

Flu:

Symptoms: Fever, Cough, Sore Throat, Fatigue.

Severity: Moderate

Tips: Maintain hygiene, rest, and stay hydrated.

Allergy:

Symptoms: Sneezing, Itching, Swelling.

Severity: Low

Tips: Avoid allergens and take antihistamines as needed.

#### Dynamic Severity Chart

The severity of the detected disease (Low, Moderate, High) is visualized as a bar chart using Chart.js.

Example:

Low Severity: Green bar dominates.

Moderate Severity: Orange bar is prominent.

High Severity: Red bar dominates.

History Tracking

Diagnosed diseases and corresponding symptoms are stored in the history array.

The diagnosis history is displayed in the history section when the "Show History" button is clicked.

#### **4. Code Workflow**

Input Collection:

The user provides their details (name, symptoms, age, gender, and lifestyle risk factors).

Symptoms are matched against the predefined symptomToDisease object to detect potential diseases.

Disease Detection:

If a match is found for the input symptoms, the system retrieves:

Disease name.

Severity level.

Detailed information about the disease.

Prevention tips.

If no match is found, it displays a message indicating no disease detected.

Chart Generation:

Generates a bar chart dynamically using Chart.js to visualize the severity of the detected disease.

History Display:

The system maintains a record of previously detected diseases and their symptoms.

The history is displayed dynamically when the user clicks the "Show History" button.

## **5. Strengths**

User-Friendly Interface:

Clean layout with intuitive inputs and clearly defined sections for outputs.

Responsive design ensures compatibility with different screen sizes.

Actionable Insights:

Provides disease-specific prevention tips tailored to the detected severity.

Dynamic Data Visualization:

The inclusion of a severity bar chart enhances data interpretability.

History Tracking:

Users can track their previously diagnosed diseases and symptoms easily.

## **Extensibility:**

The symptomToDisease object can be expanded to include additional diseases and symptoms for improved diagnostic accuracy.

## **6. Limitations**

Static Symptom Mapping:

The detection logic relies on a predefined set of symptoms and diseases, which limits flexibility and scalability.

No Backend Integration:

The system operates entirely on the front end. It lacks a backend for handling large-scale data or connecting to real-world medical databases.

Limited Scope:

The system does not integrate advanced algorithms like machine learning models or Natural Language Processing (NLP) for deeper insights.

No Personalization:



Outputs are generic and do not consider specific user data like medical history, genetic predispositions, or environmental factors.

## **7. Recommendations for Improvement**

Integration of Machine Learning:

Use AI models for symptom analysis and ADR detection for more accurate and dynamic results.

Example: Train a neural network on large medical datasets.

Backend Support:

Add a backend (e.g., Node.js or Python Flask) to connect to databases or integrate real-time medical APIs.

NLP for Advanced Symptom Analysis:

Use Natural Language Processing to interpret unstructured patient feedback or symptoms entered in free text format.

User Authentication:

Include a login system to personalize the experience and secure user data.

Real-Time Data Sources:

Integrate social media monitoring and live medical databases for dynamic disease and ADR tracking.

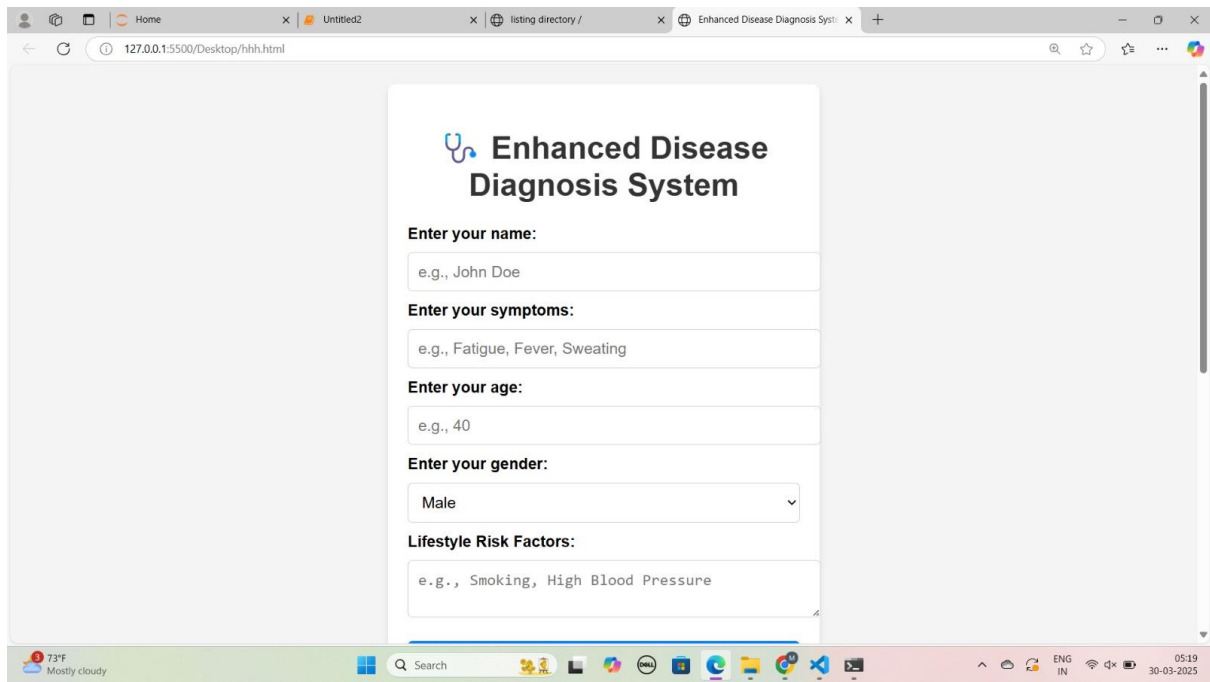
## **8. Practical Applications**

Disease Risk Assessment:

Helps users identify potential health risks based on lifestyle and symptoms.

Health Awareness:

Provides actionable insights for users to improve their health or seek medical attention



e.g., Smoking, High Blood Pressure

**Diagnose**

**Show History**

Please fill in your details and click "Diagnose" to see your results.

Your previous diagnoses will appear here.



# Enhanced Disease Diagnosis System

**Enter your name:**

anu

**Enter your symptoms:**

fever, cough, sore throat, fatigue

**Enter your age:**

45

**Enter your gender:**

Female



**Lifestyle Risk Factors:**

High blood pressure


Diagnose

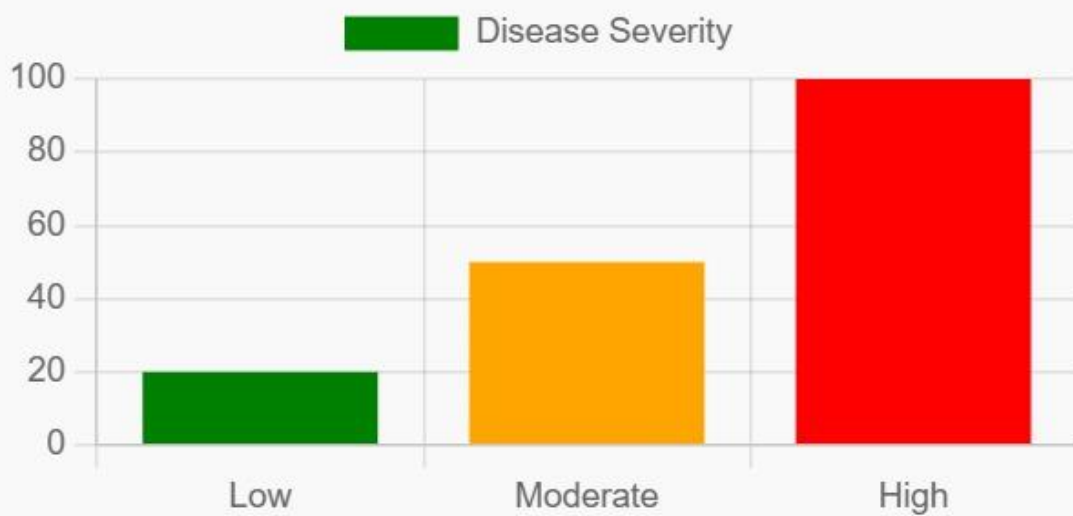
Show History

**Disease Detected:** Pneumonia

**Details:** Pneumonia inflames the lungs' air sacs. Immediate medical attention is recommended.

**Severity:** High

 **Prevention Tips:** Stay hydrated, rest adequately, and avoid cold exposure.



**1. Symptoms:** fever, cough, sore throat, fatigue, Detected Disease: Pneumonia