

GID - 2

MEDICAL DIAGNOSIS SYSTEM

Submitted to : Naisargi Oza
Date : 5th May, 2025



**K. S. School of Business Management &
Information Technology**

Abstract

- ❑ The **Medical Diagnosis System** is a **Fuzzy Logic-based** AI system designed for diagnosing diseases like **Diabetes, Heart Disease, PCOD, Thyroid and Anxiety**.
- ❑ It processes **uncertain and imprecise medical data**, mimicking human reasoning to make diagnostic decisions.
- ❑ The system also gives recommendation for Doctors and uploading report to diagnose the result.
- ❑ The system analyzes **symptoms, medical history, and test reports** to provide **risk assessment and early detection**, improving healthcare outcomes with **rule-based decision-making**.

Library

The project utilizes:

- NumPy & Pandas** – Data processing and manipulation.
- SciKit-Fuzzy (skfuzzy)** – Fuzzy logic implementation.
- Streamlit** – Web-based UI for user interaction.
- Streamlit-Option-Menu** – Sidebar navigation for Home, Dataset, and Prediction views.

Algorithm

The **Fuzzy Inference System (FIS)** consists of:

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

Future Enhancement

- **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.

I.

MEDICAL DIAGNOSIS SYSTEM



Medical_Diagnosis_System.py 5 ×

```
1  import numpy as np
2  import pandas as pd
3  import skfuzzy as fuzz
4  import skfuzzy.control as ctrl
5  import streamlit as st
6  from streamlit_option_menu import option_menu
7
8
9 # --- Load datasets ---
10 Tabnine | Edit | Test | Explain | Document
11 def load_dataset(dataset_type):
12     try:
13         if dataset_type == "diabetes":
14             return pd.read_csv("diabetes.csv")
15         elif dataset_type == "heart":
16             return pd.read_csv("heart.csv")
17         elif dataset_type == "thyroid":
18             file_path = "thyroid_dataset_300_rows.csv"
19             df = pd.read_csv(file_path)
20             df.columns = df.columns.str.strip()
21             df.rename(columns={
22                 'TSH (mIU/L)': 'TSH',
23                 'T3 (ng/dL)': 'T3',
24                 'T4 (μg/dL)': 'T4'
25             }, inplace=True)
26             return df
27         elif dataset_type == "pcod":
28             return pd.read_csv("pcod.csv")
```

The image shows a screenshot of the PyCharm IDE. The title bar reads "Systemwith_3Disese". The left sidebar contains icons for file operations like Open, Save, Find, and Settings. The main editor window displays a Python script titled "Medical_Diagnosis_System.py". The code implements a fuzzy logic system for heart disease diagnosis using the Fuzzy Logic library (ctrl). It defines several antecedents (age, cholesterol, thalach, chest_pain, resting_bp) and a consequent (heart_risk), all with 3 membership functions each. It also handles dataset loading from CSV files for "pcod" and "anxiety".

```
25         return df
26     elif dataset_type == "pcod":
27         return pd.read_csv("pcod.csv")
28     elif dataset_type == "anxiety":
29         return pd.read_csv("anxiety_dataset_300_modified.csv")
30 except Exception as e:
31     st.error(f"Error loading {dataset_type} dataset: {e}")
32 return None
33
34
35 # --- Fuzzy Logic for Heart Disease ---
36 Tabnine | Edit | Test | Explain | Document
37 def create_fuzzy_heart(df):
38     age = ctrl.Antecedent(np.arange(df['age'].min(), df['age'].max() + 1, 1), 'age')
39     cholesterol = ctrl.Antecedent(np.arange(df['Cholesterol'].min(), df['Cholesterol'].max() + 1, 1), 'cholesterol')
40     thalach = ctrl.Antecedent(np.arange(df['thalach'].min(), df['thalach'].max() + 1, 1), 'thalach')
41     chest_pain = ctrl.Antecedent(np.arange(0, 4, 1), 'chest_pain')
42     resting_bp = ctrl.Antecedent(np.arange(df['trestbps'].min(), df['trestbps'].max() + 1, 1), 'resting_bp')
43     heart_risk = ctrl.Consequent(np.arange(0, 101, 1), 'heart_risk')
44
45     age.automf(3)
46     cholesterol.automf(3)
47     thalach.automf(3)
48     chest_pain.automf(3)
49     resting_bp.automf(3)
50     heart_risk.automf(3)
51
52     rules = [
53         Rule(age['Low'], cholesterol['Low'], thalach['Low'], chest_pain['No'], resting_bp['Low'], heart_risk['Low']),
54         Rule(age['Low'], cholesterol['Low'], thalach['Low'], chest_pain['Yes'], resting_bp['Low'], heart_risk['Low']),
55         Rule(age['Low'], cholesterol['Low'], thalach['Low'], chest_pain['No'], resting_bp['High'], heart_risk['Low']),
56         Rule(age['Low'], cholesterol['Low'], thalach['Low'], chest_pain['Yes'], resting_bp['High'], heart_risk['Low']),
57         Rule(age['Low'], cholesterol['Low'], thalach['Medium'], chest_pain['No'], resting_bp['Low'], heart_risk['Low']),
58         Rule(age['Low'], cholesterol['Low'], thalach['Medium'], chest_pain['Yes'], resting_bp['Low'], heart_risk['Low']),
59         Rule(age['Low'], cholesterol['Low'], thalach['Medium'], chest_pain['No'], resting_bp['High'], heart_risk['Low']),
60         Rule(age['Low'], cholesterol['Low'], thalach['Medium'], chest_pain['Yes'], resting_bp['High'], heart_risk['Low']),
61         Rule(age['Low'], cholesterol['Medium'], thalach['Low'], chest_pain['No'], resting_bp['Low'], heart_risk['Low']),
62         Rule(age['Low'], cholesterol['Medium'], thalach['Low'], chest_pain['Yes'], resting_bp['Low'], heart_risk['Low']),
63         Rule(age['Low'], cholesterol['Medium'], thalach['Low'], chest_pain['No'], resting_bp['High'], heart_risk['Low']),
64         Rule(age['Low'], cholesterol['Medium'], thalach['Low'], chest_pain['Yes'], resting_bp['High'], heart_risk['Low']),
65         Rule(age['Low'], cholesterol['Medium'], thalach['Medium'], chest_pain['No'], resting_bp['Low'], heart_risk['Low']),
66         Rule(age['Low'], cholesterol['Medium'], thalach['Medium'], chest_pain['Yes'], resting_bp['Low'], heart_risk['Low']),
67         Rule(age['Low'], cholesterol['Medium'], thalach['Medium'], chest_pain['No'], resting_bp['High'], heart_risk['Low']),
68         Rule(age['Low'], cholesterol['Medium'], thalach['Medium'], chest_pain['Yes'], resting_bp['High'], heart_risk['Low']),
69         Rule(age['Medium'], cholesterol['Low'], thalach['Low'], chest_pain['No'], resting_bp['Low'], heart_risk['Low']),
70         Rule(age['Medium'], cholesterol['Low'], thalach['Low'], chest_pain['Yes'], resting_bp['Low'], heart_risk['Low']),
71         Rule(age['Medium'], cholesterol['Low'], thalach['Low'], chest_pain['No'], resting_bp['High'], heart_risk['Low']),
72         Rule(age['Medium'], cholesterol['Low'], thalach['Low'], chest_pain['Yes'], resting_bp['High'], heart_risk['Low']),
73         Rule(age['Medium'], cholesterol['Low'], thalach['Medium'], chest_pain['No'], resting_bp['Low'], heart_risk['Low']),
74         Rule(age['Medium'], cholesterol['Low'], thalach['Medium'], chest_pain['Yes'], resting_bp['Low'], heart_risk['Low']),
75         Rule(age['Medium'], cholesterol['Low'], thalach['Medium'], chest_pain['No'], resting_bp['High'], heart_risk['Low']),
76         Rule(age['Medium'], cholesterol['Low'], thalach['Medium'], chest_pain['Yes'], resting_bp['High'], heart_risk['Low']),
77         Rule(age['Medium'], cholesterol['Medium'], thalach['Low'], chest_pain['No'], resting_bp['Low'], heart_risk['Low']),
78         Rule(age['Medium'], cholesterol['Medium'], thalach['Low'], chest_pain['Yes'], resting_bp['Low'], heart_risk['Low']),
79         Rule(age['Medium'], cholesterol['Medium'], thalach['Low'], chest_pain['No'], resting_bp['High'], heart_risk['Low']),
80         Rule(age['Medium'], cholesterol['Medium'], thalach['Low'], chest_pain['Yes'], resting_bp['High'], heart_risk['Low']),
81         Rule(age['Medium'], cholesterol['Medium'], thalach['Medium'], chest_pain['No'], resting_bp['Low'], heart_risk['Low']),
82         Rule(age['Medium'], cholesterol['Medium'], thalach['Medium'], chest_pain['Yes'], resting_bp['Low'], heart_risk['Low']),
83         Rule(age['Medium'], cholesterol['Medium'], thalach['Medium'], chest_pain['No'], resting_bp['High'], heart_risk['Low']),
84         Rule(age['Medium'], cholesterol['Medium'], thalach['Medium'], chest_pain['Yes'], resting_bp['High'], heart_risk['Low']),
85         Rule(age['High'], cholesterol['Low'], thalach['Low'], chest_pain['No'], resting_bp['Low'], heart_risk['Low']),
86         Rule(age['High'], cholesterol['Low'], thalach['Low'], chest_pain['Yes'], resting_bp['Low'], heart_risk['Low']),
87         Rule(age['High'], cholesterol['Low'], thalach['Low'], chest_pain['No'], resting_bp['High'], heart_risk['Low']),
88         Rule(age['High'], cholesterol['Low'], thalach['Low'], chest_pain['Yes'], resting_bp['High'], heart_risk['Low']),
89         Rule(age['High'], cholesterol['Low'], thalach['Medium'], chest_pain['No'], resting_bp['Low'], heart_risk['Low']),
90         Rule(age['High'], cholesterol['Low'], thalach['Medium'], chest_pain['Yes'], resting_bp['Low'], heart_risk['Low']),
91         Rule(age['High'], cholesterol['Low'], thalach['Medium'], chest_pain['No'], resting_bp['High'], heart_risk['Low']),
92         Rule(age['High'], cholesterol['Low'], thalach['Medium'], chest_pain['Yes'], resting_bp['High'], heart_risk['Low']),
93         Rule(age['High'], cholesterol['Medium'], thalach['Low'], chest_pain['No'], resting_bp['Low'], heart_risk['Low']),
94         Rule(age['High'], cholesterol['Medium'], thalach['Low'], chest_pain['Yes'], resting_bp['Low'], heart_risk['Low']),
95         Rule(age['High'], cholesterol['Medium'], thalach['Low'], chest_pain['No'], resting_bp['High'], heart_risk['Low']),
96         Rule(age['High'], cholesterol['Medium'], thalach['Low'], chest_pain['Yes'], resting_bp['High'], heart_risk['Low']),
97         Rule(age['High'], cholesterol['Medium'], thalach['Medium'], chest_pain['No'], resting_bp['Low'], heart_risk['Low']),
98         Rule(age['High'], cholesterol['Medium'], thalach['Medium'], chest_pain['Yes'], resting_bp['Low'], heart_risk['Low']),
99         Rule(age['High'], cholesterol['Medium'], thalach['Medium'], chest_pain['No'], resting_bp['High'], heart_risk['Low']),
100        Rule(age['High'], cholesterol['Medium'], thalach['Medium'], chest_pain['Yes'], resting_bp['High'], heart_risk['Low']),
101    ]
```

The image shows a screenshot of the PyCharm IDE. The title bar reads "Systemwith_3Disese". The left sidebar contains icons for file operations, search, and other development tools. The main editor window displays a Python script named "Medical_Diagnosis_System.py". The code implements a fuzzy logic system for heart disease and diabetes risk.

```
Medical_Diagnosis_System.py 5 ×
40     resting_bp.automf(3)
41
42     heart_risk.automf(3)
43
44
45
46
47
48
49
50
51     rules = [
52         ctrl.Rule(age['poor'] & cholesterol['poor'], heart_risk['poor']),
53         ctrl.Rule(chest_pain['poor'] & resting_bp['poor'], heart_risk['poor']),
54         ctrl.Rule(age['average'] & cholesterol['average'], heart_risk['average']),
55         ctrl.Rule(thalach['good'] & cholesterol['good'], heart_risk['good']),
56         ctrl.Rule(chest_pain['good'] & thalach['good'] & age['good'], heart_risk['good']),
57     ]
58
59     heart_ctrl = ctrl.ControlSystem(rules)
60     return ctrl.ControlSystemSimulation(heart_ctrl)
61
62 # --- Fuzzy Logic for Diabetes ---
63 Tabnine | Edit | Test | Explain | Document
64 def create_fuzzy_diabetes(df):
65     glucose = ctrl.Antecedent(np.arange(df['Glucose'].min(), df['Glucose'].max() + 1, 1), 'glucose')
66     bmi = ctrl.Antecedent(np.arange(df['BMI'].min(), df['BMI'].max() + 1, 1), 'bmi')
67     age = ctrl.Antecedent(np.arange(df['Age'].min(), df['Age'].max() + 1, 1), 'age')
68     blood_pressure = ctrl.Antecedent(np.arange(df['BloodPressure'].min(), df['BloodPressure'].max() + 1, 1), 'blood_pressure')
69     diabetes_risk = ctrl.Consequent(np.arange(0, 101, 1), 'diabetes_risk')
70
71     glucose.automf(3)
72     bmi.automf(3)
73     age.automf(3)
74     blood_pressure.automf(3)
75     diabetes_risk.automf(3)
76
76     rules = [
77         ctrl.Rule(glucose['poor'] & bmi['poor'] & age['poor'], diabetes_risk['poor']),
78         ctrl.Rule(glucose['average'] & bmi['average'] & age['average'], diabetes_risk['average']),
79         ctrl.Rule(glucose['good'] & bmi['good'] & age['good'], diabetes_risk['good']),
80         ctrl.Rule(blood_pressure['poor'] & glucose['poor'], diabetes_risk['poor']),
81         ctrl.Rule(blood_pressure['average'] & bmi['average'] & age['average'], diabetes_risk['average']),
82         ctrl.Rule(blood_pressure['good'] & bmi['good'] & age['good'], diabetes_risk['good']),
83     ]
84
85     diabetes_ctrl = ctrl.ControlSystem(rules)
86     return ctrl.ControlSystemSimulation(diabetes_ctrl)
```

The image shows a screenshot of the PyCharm IDE. The title bar at the top displays "Systemwith_3Disese". The left sidebar contains various icons for navigation and project management. The main editor area shows a Python script named "Medical_Diagnosis_System.py". The code implements two fuzzy logic systems for diabetes and thyroid risk prediction based on input features like glucose, blood pressure, BMI, and age.

```
75     rules = [
76         ctrl.Rule(glucose['poor'] & bmi['poor'] & age['poor'], diabetes_risk['poor']),
77         ctrl.Rule(glucose['average'] & bmi['average'] & age['average'], diabetes_risk['average']),
78         ctrl.Rule(glucose['good'] & bmi['good'] & age['good'], diabetes_risk['good']),
79         ctrl.Rule(blood_pressure['poor'] & glucose['poor'], diabetes_risk['poor']),
80         ctrl.Rule(blood_pressure['average'] & bmi['average'] & age['average'], diabetes_risk['average']),
81         ctrl.Rule(glucose['average'] & blood_pressure['good'], diabetes_risk['average'])
82     ]
83
84
85     diabetes_ctrl = ctrl.ControlSystem(rules)
86     return ctrl.ControlSystemSimulation(diabetes_ctrl)
87
88 # --- Fuzzy Logic for Thyroid ---
89 Tabnine | Edit | Test | Explain | Document
90 def create_fuzzy_thyroid(df):
91     tsh = ctrl.Antecedent(np.arange(df['TSH'].min(), df['TSH'].max() + 1, 0.1), 'tsh')
92     t3 = ctrl.Antecedent(np.arange(df['T3'].min(), df['T3'].max() + 1, 1), 't3')
93     t4 = ctrl.Antecedent(np.arange(df['T4'].min(), df['T4'].max() + 1, 0.1), 't4')
94     thyroid_risk = ctrl.Consequent(np.arange(0, 101, 1), 'thyroid_risk')
95
96     tsh.automf(3)
97     t3.automf(3)
98     t4.automf(3)
99     thyroid_risk.automf(3)
100
101    rules = [
102        ctrl.Rule(tsh['good'] & t3['good'] & t4['good'], thyroid_risk['poor']),
103        ctrl.Rule(tsh['average'] | t3['average'] | t4['average'], thyroid_risk['average']),
104        ctrl.Rule(tsh['poor'] | t3['poor'] | t4['poor'], thyroid_risk['good']),
105    ]
106
107    system = ctrl.ControlSystem(rules)
108    return ctrl.ControlSystemSimulation(system)
```

Medical_Diagnosis_System.py 5 ×

```
108
109
110 # --- Fuzzy Logic for PCOD ---
Tabnine | Edit | Test | Explain | Document
111 def create_fuzzy_pcod(df):
112     bmi = ctrl.Antecedent(np.arange(df['BMI'].min(), df['BMI'].max() + 1, 1), 'bmi')
113     insulin = ctrl.Antecedent(np.arange(df['Insulin_Level'].min(), df['Insulin_Level'].max() + 1, 1), 'insulin')
114     lh = ctrl.Antecedent(np.arange(df['LH'].min(), df['LH'].max() + 1, 1), 'lh')
115     pcod_risk = ctrl.Consequent(np.arange(0, 101, 1), 'pcod_risk')

116     bmi.automf(3)
117     insulin.automf(3)
118     lh.automf(3)
119     pcod_risk.automf(3)

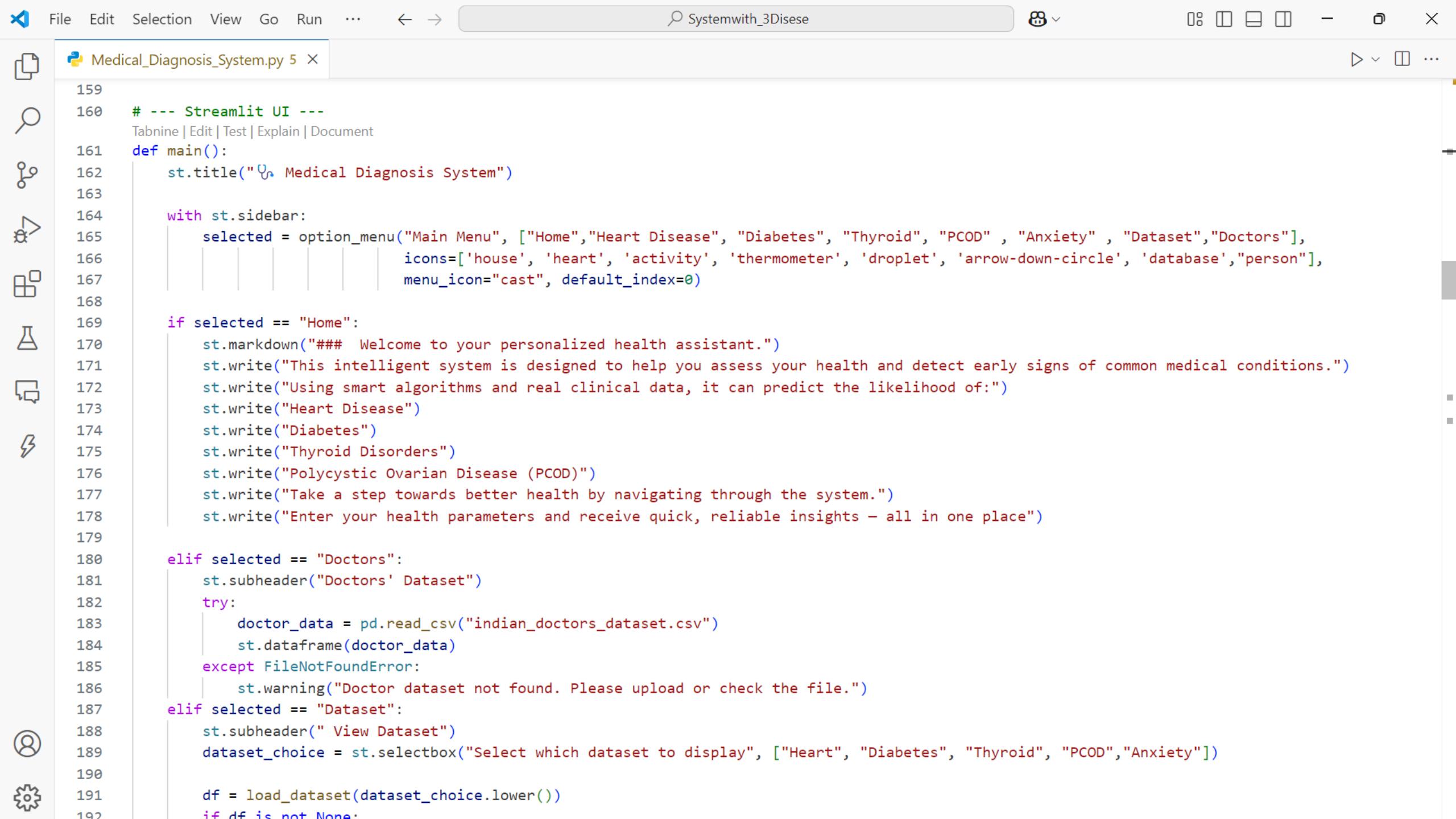
120     rules = [
121         ctrl.Rule(bmi['good'] & insulin['good'] & lh['good'], pcod_risk['poor']),
122         ctrl.Rule(bmi['average'] | insulin['average'] | lh['average'], pcod_risk['average']),
123         ctrl.Rule(bmi['poor'] | insulin['poor'] | lh['poor'], pcod_risk['good'])
124     ]
125
126     system = ctrl.ControlSystem(rules)
127     return ctrl.ControlSystemSimulation(system)

128
129
130 Tabnine | Edit | Test | Explain | Document
131 def create_fuzzy_anxiety(df):
132     sleep = ctrl.Antecedent(np.arange(df['SleepHours'].min(), df['SleepHours'].max()+0.1, 0.1), 'sleep')
133     heart_rate = ctrl.Antecedent(np.arange(df['HeartRate'].min(), df['HeartRate'].max()+1, 1), 'heart_rate')
134     fatigue = ctrl.Antecedent(np.arange(0, 2, 1), 'fatigue')
135     irritability = ctrl.Antecedent(np.arange(0, 2, 1), 'irritability')
136     restlessness = ctrl.Antecedent(np.arange(0, 2, 1), 'restlessness')
137     score = ctrl.Antecedent(np.arange(df['ScoreGAD7'].min(), df['ScoreGAD7'].max()+1, 1), 'score')
138     anxiety_risk = ctrl.Consequent(np.arange(0, 101, 1), 'anxiety_risk')

139     sleep.automf(3)
```

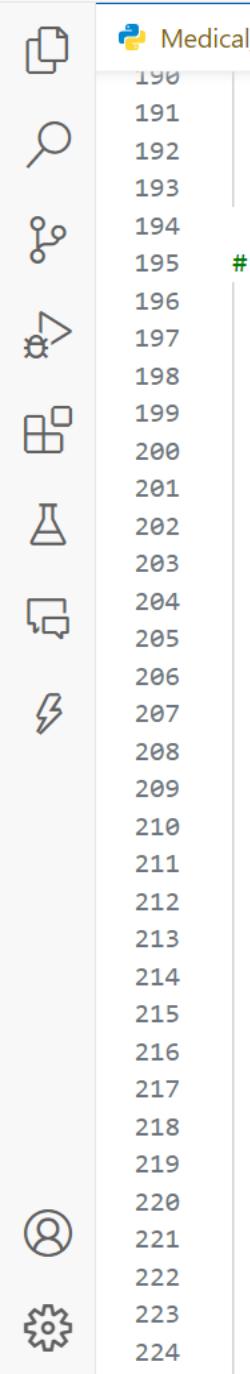
Medical_Diagnosis_System.py 5 ×

```
Tabnine | Edit | Test | Explain | Document
131 def create_fuzzy_anxiety(df):
132     sleep = ctrl.Antecedent(np.arange(df['SleepHours'].min(), df['SleepHours'].max()+0.1, 0.1), 'sleep')
133     heart_rate = ctrl.Antecedent(np.arange(df['HeartRate'].min(), df['HeartRate'].max()+1, 1), 'heart_rate')
134     fatigue = ctrl.Antecedent(np.arange(0, 2, 1), 'fatigue')
135     irritability = ctrl.Antecedent(np.arange(0, 2, 1), 'irritability')
136     restlessness = ctrl.Antecedent(np.arange(0, 2, 1), 'restlessness')
137     score = ctrl.Antecedent(np.arange(df['ScoreGAD7'].min(), df['ScoreGAD7'].max()+1, 1), 'score')
138     anxiety_risk = ctrl.Consequent(np.arange(0, 101, 1), 'anxiety_risk')
139
140     sleep.automf(3)
141     heart_rate.automf(3)
142     fatigue.automf(3)
143     irritability.automf(3)
144     restlessness.automf(3)
145     score.automf(3)
146     anxiety_risk.automf(3)
147
148     rules = [
149         ctrl.Rule(sleep['poor'] | score['good'] | heart_rate['good'], anxiety_risk['good']),
150         ctrl.Rule(score['average'] | irritability['average'] | fatigue['average'], anxiety_risk['average']),
151         ctrl.Rule(sleep['good'] & fatigue['poor'] & irritability['poor'], anxiety_risk['poor']),
152         ctrl.Rule(restlessness['poor'] | fatigue['good'], anxiety_risk['good']), # ✓ Added rule with restlessness
153     ]
154
155
156     system = ctrl.ControlSystem(rules)
157     return ctrl.ControlSystemSimulation(system)
158
159
160 # --- Streamlit UI ---
Tabnine | Edit | Test | Explain | Document
161 def main():
162     st.title("Medical Diagnosis System")
163
```



Medical_Diagnosis_System.py 5 X

```
159
160 # --- Streamlit UI ---
161 Tabnine | Edit | Test | Explain | Document
162 def main():
163     st.title("⚕️ Medical Diagnosis System")
164
165     with st.sidebar:
166         selected = option_menu("Main Menu", ["Home", "Heart Disease", "Diabetes", "Thyroid", "PCOD", "Anxiety", "Dataset", "Doctors"], icons=['house', 'heart', 'activity', 'thermometer', 'droplet', 'arrow-down-circle', 'database', 'person'], menu_icon="cast", default_index=0)
167
168     if selected == "Home":
169         st.markdown("### Welcome to your personalized health assistant.")
170         st.write("This intelligent system is designed to help you assess your health and detect early signs of common medical conditions.")
171         st.write("Using smart algorithms and real clinical data, it can predict the likelihood of:")
172         st.write("Heart Disease")
173         st.write("Diabetes")
174         st.write("Thyroid Disorders")
175         st.write("Polycystic Ovarian Disease (PCOD)")
176         st.write("Take a step towards better health by navigating through the system.")
177         st.write("Enter your health parameters and receive quick, reliable insights – all in one place")
178
179
180     elif selected == "Doctors":
181         st.subheader("Doctors' Dataset")
182         try:
183             doctor_data = pd.read_csv("indian_doctors_dataset.csv")
184             st.dataframe(doctor_data)
185         except FileNotFoundError:
186             st.warning("Doctor dataset not found. Please upload or check the file.")
187     elif selected == "Dataset":
188         st.subheader("View Dataset")
189         dataset_choice = st.selectbox("Select which dataset to display", ["Heart", "Diabetes", "Thyroid", "PCOD", "Anxiety"])
190
191         df = load_dataset(dataset_choice.lower())
192         if df is not None:
```



Medical_Diagnosis_System.py 5 X

```
190
191     df = load_dataset(dataset_choice.lower())
192     if df is not None:
193         st.dataframe(df)
194
195 # Heart disease logic
196
197     elif selected == "Heart Disease":
198         df = load_dataset("heart")
199         if df is None:
200             return
201         sim = create_fuzzy_heart(df)
202
203         st.subheader("Heart Disease")
204         age = st.number_input("Age", int(df['age'].min()), int(df['age'].max()), int(df['age'].mean()))
205         cholesterol = st.number_input("Cholesterol", int(df['Cholesterol'].min()), int(df['Cholesterol'].max()), int(df['Cholesterol'].mean()))
206         thalach = st.number_input("Max Heart Rate", int(df['thalach'].min()), int(df['thalach'].max()), int(df['thalach'].mean()))
207         chest_pain = st.slider("Chest Pain Type (0: Typical, 1: Atypical, 2: Non-anginal, 3: Asymptomatic)", 0, 3, 1)
208         resting_bp = st.number_input("Resting Blood Pressure", int(df['trestbps'].min()), int(df['trestbps'].max()), int(df['trestbps'].mean()))
209
210         col1, col2, col3 = st.columns([1, 1, 1])
211         with col1:
212             predict = st.button("Predict Heart Status", key="predict_heart")
213         with col3:
214             show_doctors = st.button("Show Recommended Doctors", key="show_heart_doctors")
215
216         if predict:
217             sim.input['age'] = age
218             sim.input['cholesterol'] = cholesterol
219             sim.input['thalach'] = thalach #Max Heart Rate
220             sim.input['chest_pain'] = chest_pain
221             sim.input['resting_bp'] = resting_bp
222
223             sim.compute()
224             risk = sim.output['heart_risk']
```

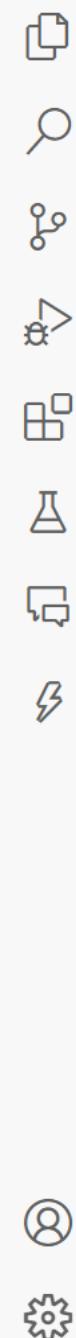
The image shows a screenshot of the PyCharm IDE. The title bar at the top reads "Systemwith_3Disese". The left sidebar contains various icons for navigation and project management. The main editor window displays a Python script named "Medical_Diagnosis_System.py". The code implements a medical diagnosis system using a trained model (sim) and performs several checks on input variables (cholesterol, resting_bp, thalach, age, chest_pain) to provide health warnings. It also handles a "show_doctors" case by reading a CSV file ("indian_doctors_dataset.csv") and filtering for heart specialists.

```
223     sim.compute()
224     risk = sim.output['heart_risk']
225     diagnosis = "Yes" if risk >= 50 else "No"
226     st.success(f"Predicted Heart Disease Risk: {diagnosis}")
227
228     warnings = []
229     if cholesterol > 240:
230         warnings.append("⚠️ High Cholesterol - Increased risk of heart disease.")
231     if cholesterol < 125:
232         warnings.append("⚠️ Low Cholesterol - May indicate underlying health issues or malnutrition.")
233
234     if resting_bp > 140:
235         warnings.append("⚠️ High Resting Blood Pressure - May indicate hypertension.")
236     if resting_bp < 100:
237         warnings.append("⚠️ Low Resting Blood Pressure - May lead to dizziness or fainting.")
238
239     if thalach < 100:
240         warnings.append("⚠️ Low Max Heart Rate - May be a sign of poor cardiovascular fitness.")
241     if age > 60:
242         warnings.append("⚠️ Age over 60 - Age is a major risk factor for heart disease.")
243     if chest_pain == 3:
244         warnings.append("⚠️ Asymptomatic Chest Pain Type - Often linked with higher heart disease risk.")
245
246     if warnings:
247         warning_html = '<div style="background-color:#fbeaea;padding:10px;border-radius:8px;">' + \
248             '<br>'.join(f'<span style="color:#d00000;">{w}</span>' for w in warnings) + \
249             '</div>'
250         st.markdown(warning_html, unsafe_allow_html=True)
251     elif show_doctors:
252         st.subheader("Recommended Doctors for Heart Disease")
253         try:
254             doctor_data = pd.read_csv("indian_doctors_dataset.csv")
255             h_doctors = doctor_data[
256                 doctor_data['Specialist'].str.lower().str.contains("heart", na=False)
```

```
Medical_Diagnosis_System.py 5 X
252     st.subheader("Recommended Doctors for Heart Disease")
253     try:
254         doctor_data = pd.read_csv("indian_doctors_dataset.csv")
255         h_doctors = doctor_data[
256             doctor_data['Specialist'].str.lower().str.contains("heart", na=False)
257         ]
258         if not h_doctors.empty:
259             grouped = h_doctors.groupby('Clinic Address')
260             count = 0
261             for name, group in grouped:
262                 st.markdown(f"##### 🏥 Location: {name}")
263                 top5 = group.head(3)[['Doctor Name', 'Specialist', 'Phone Number', 'Email']]
264                 st.dataframe(top5, use_container_width=True)
265                 count += 1
266                 if count >= 3:
267                     break
268             else:
269                 st.info("No doctors found for Heart Disease.")
270     except Exception as e:
271         st.error(f"Error loading doctors data: {e}")
272
273 # Diabetes logic
274
275     elif selected == "Diabetes":
276         df = load_dataset("diabetes")
277         if df is None:
278             return
279         sim = create_fuzzy_diabetes(df)
280         st.subheader("Diabetes ")
281         #glucose = st.number_input("Glucose", int(df['Glucose'].min()), int(df['Glucose'].max()), int(df['Glucose'].mean()))
282         glucose = st.number_input("Glucose", int(df['Glucose'].min()), int(df['Glucose'].max()), int(df['Glucose'].mean()), key="glucose_input")
283
284         bmi = st.number_input("BMI", float(df['BMI'].min()), float(df['BMI'].max()), float(df['BMI'].mean()))
285         age = st.number_input("Age", int(df['Age'].min()), int(df['Age'].max()), int(df['Age'].mean()))
286         bp = st.number_input("Blood Pressure", int(df['BloodPressure'].min()), int(df['BloodPressure'].max()), int(df['BloodPressure'].mean()))
```

Medical_Diagnosis_System.py 5 X

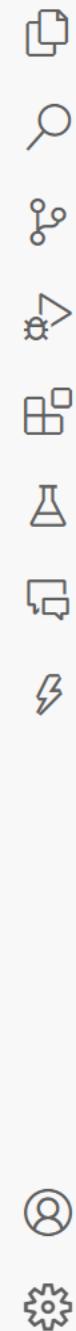
```
286     bp = st.number_input("Blood Pressure", int(df['BloodPressure'].min()), int(df['BloodPressure'].max()), int(df['BloodPressure'].mean()))
287
288     st.markdown("### 📈 Upload Blood Test Report (CSV or Excel)")
289     uploaded_file = st.file_uploader("Choose a file", type=["csv", "xlsx"])
290
291     if uploaded_file:
292         try:
293             if uploaded_file.name.endswith('.csv'):
294                 user_data = pd.read_csv(uploaded_file)
295             else:
296                 user_data = pd.read_excel(uploaded_file)
297
298             st.success("✅ Report uploaded successfully!")
299             st.dataframe(user_data)
300
301             # Assuming the report has correct column names
302             glucose = user_data['Glucose'][0]
303             bmi = user_data['BMI'][0]
304             age = user_data['Age'][0]
305             bp = user_data['BloodPressure'][0]
306
307             except Exception as e:
308                 st.error(f"⚠️ Error reading file: {e}")
309                 return
310             col1, col2, col3 = st.columns([1, 1, 1])
311             with col1:
312                 predict = st.button("Predict Diabetes Status", key="predict_diabetes")
313             with col3:
314                 show_doctors = st.button("Show Recommended Doctors", key="show_diabetes_doctors")
315
316             if predict:
317                 sim.input['glucose'] = glucose
318                 sim.input['bmi'] = bmi
319                 sim.input['age'] = age
320                 sim.input['blood pressure'] = bp
```



Medical_Diagnosis_System.py 5 X

```
316     if predict:
317         sim.input['glucose'] = glucose
318         sim.input['bmi'] = bmi
319         sim.input['age'] = age
320         sim.input['blood_pressure'] = bp
321         sim.compute()
322         risk = sim.output['diabetes_risk']
323         diagnosis = "Yes" if risk >= 50 else "No"
324         st.success(f"Predicted Diabetes Status: {diagnosis}")
325
326         # Collect warnings
327         warnings = []
328         if glucose > 140:
329             warnings.append("⚠️ High glucose")
330         if bmi > 30:
331             warnings.append("⚠️ High BMI")
332         if bp > 80:
333             warnings.append("⚠️ High Blood pressure")
334         if glucose < 70:
335             warnings.append("⚠️ Low glucose - Your sugar level is too low. You might feel weak, shaky, or tired.")
336         if bmi < 18.5:
337             warnings.append("⚠️ Low BMI - Your body weight is quite low. You may need to eat more to stay healthy.")
338         if bp < 60:
339             warnings.append("⚠️ Low blood pressure - Your blood pressure is low. This can make you feel dizzy or faint.")
340         # Display warnings in one line using markdown
341         if warnings:
342             warning_html = '<div style="background-color:#fbeaea;padding:10px;border-radius:8px;">' + \
343                         ' '.join(f'<br><span style="color:#d00000;">{w}</span>' for w in warnings) + \
344                         '</div>'
345             st.markdown(warning_html, unsafe_allow_html=True)
346
347         elif show_doctors:
348             st.subheader("Recommended Doctors for Diabetes")
349             try:
350                 doctors = pd.read_csv("https://tinyurl.com/3yj2nq4c")
```

```
Medical_Diagnosis_System.py 5 X
347     elif show_doctors:
348         st.subheader("Recommended Doctors for Diabetes")
349         try:
350             doctor_data = pd.read_csv("indian_doctors_dataset.csv")
351             diabetes_doctors = doctor_data[
352                 doctor_data['Specialist'].str.lower().str.contains("diabet", na=False)
353             ]
354             if not diabetes_doctors.empty:
355                 grouped = diabetes_doctors.groupby('Clinic Address')
356                 count = 0
357                 for name, group in grouped:
358                     st.markdown(f"##### * Location: {name}")
359                     top5 = group.head(3)[['Doctor Name', 'Specialist', 'Phone Number', 'Email']]
360                     st.dataframe(top5, use_container_width=True)
361                     count += 1
362                     if count >= 3:
363                         break
364                 else:
365                     st.info("No doctors found for Diabetes.")
366             except Exception as e:
367                 st.error(f"Error loading doctors data: {e}")
368
369 # Thyroid logic
370
371     elif selected == "Thyroid":
372         df = load_dataset("thyroid")
373         if df is None:
374             return
375         sim = create_fuzzy_thyroid(df)
376         st.subheader("Thyroid")
377         tsh = st.number_input("TSH (mIU/L)", float(df['TSH'].min()), float(df['TSH'].max()), float(df['TSH'].mean()))
378         t3 = st.number_input("T3 (ng/dL)", float(df['T3'].min()), float(df['T3'].max()), float(df['T3'].mean()))
379         t4 = st.number_input("T4 (μg/dL)", float(df['T4'].min()), float(df['T4'].max()), float(df['T4'].mean()))
```

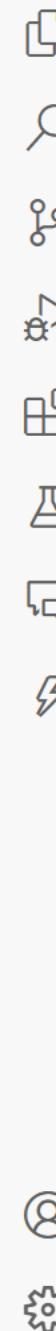


Medical_Diagnosis_System.py 5 X

```
374         return
375     sim = create_fuzzy_thyroid(df)
376     st.subheader("Thyroid")
377     tsh = st.number_input("TSH (mIU/L)", float(df['TSH'].min()), float(df['TSH'].max()), float(df['TSH'].mean()))
378     t3 = st.number_input("T3 (ng/dL)", float(df['T3'].min()), float(df['T3'].max()), float(df['T3'].mean()))
379     t4 = st.number_input("T4 (μg/dL)", float(df['T4'].min()), float(df['T4'].max()), float(df['T4'].mean()))
380
381     st.markdown("### 📈 Upload blood Test Report (CSV or Excel)")
382     uploaded_file = st.file_uploader("Choose a file", type=["csv", "xlsx"])
383
384
385     if uploaded_file:
386         try:
387             if uploaded_file.name.endswith('.csv'):
388                 user_data = pd.read_csv(uploaded_file)
389             else:
390                 user_data = pd.read_excel(uploaded_file)
391
392             st.success("✅ Report uploaded successfully!")
393             st.dataframe(user_data)
394
395             # Assuming the report has correct column names
396             tsh = user_data['TSH (mIU/L)'][0]
397             t3 = user_data['T3 (ng/dL)'][0]
398             t4 = user_data['T4 (μg/dL)'][0]
399
400         except Exception as e:
401             st.error(f⚠️ Error reading file: {e}")
402             return
403         col1, col2, col3 = st.columns([1, 1, 1])
404         with col1:
405             predict = st.button("Predict Thyroid Status", key="predict_thyroid")
406         with col3:
407             show_doctors = st.button("Show Recommended Doctors", key="show_thyroid_doctors")
```

```
Medical_Diagnosis_System.py 5 ×

406     with col3:
407         show_doctors = st.button("Show Recommended Doctors", key="show_thyroid_doctors")
408
409     if predict:
410         sim.input['tsh'] = tsh
411         sim.input['t3'] = t3
412         sim.input['t4'] = t4
413         sim.compute()
414         risk = sim.output['thyroid_risk']
415         diagnosis = "Yes" if risk >= 50 else "No"
416         st.success(f"Predicted Thyroid Status: {diagnosis}")
417
418         warnings = []
419         if tsh > 4.0:
420             warnings.append("⚠️ High TSH 📈 Could indicate hypothyroidism.")
421         if tsh < 0.4 :
422             warnings.append("⚠️ Low TSH 📈 Could be a sign of hyperthyroidism.")
423         if t3 < 70:
424             warnings.append("⚠️ Low T3 📈 Often seen in hypothyroidism.")
425         if t3 > 200:
426             warnings.append("⚠️ High T3 📈 Might indicate an overactive thyroid.")
427         if t4 < 5.0:
428             warnings.append("⚠️ Low T4 📈 May signal thyroid hormone deficiency.")
429
430         if warnings:
431             warning_html = '<div style="background-color:#fbeaea;padding:10px;border-radius:8px;">' + \
432             ' '.join(f'<span style="color:#d00000;">{w}</span>' for w in warnings) + \
433             '</div>'
434             st.markdown(warning_html, unsafe_allow_html=True)
435     elif show_doctors:
436         st.subheader("Recommended Doctors for Thyroid")
437         try:
438             doctor_data = pd.read_csv("indian_doctors_dataset.csv")
439             t_doctors = doctor_data[
440                 doctor_data['Specialist'].str.lower().str.contains("thyroid", na=False)
```



Medical_Diagnosis_System.py 5 X

```
435     elif show_doctors:
436         st.subheader("Recommended Doctors for Thyroid")
437         try:
438             doctor_data = pd.read_csv("indian_doctors_dataset.csv")
439             t_doctors = doctor_data[
440                 doctor_data['Specialist'].str.lower().str.contains("thyroid", na=False)
441             ]
442             if not t_doctors.empty:
443                 grouped = t_doctors.groupby('Clinic Address')
444                 count = 0
445                 for name, group in grouped:
446                     st.markdown(f"##### * Location: {name}*")
447                     top5 = group.head(3)[['Doctor Name', 'Specialist', 'Phone Number', 'Email']]
448                     st.dataframe(top5, use_container_width=True)
449                     count += 1
450                     if count >= 3:
451                         break
452                 else:
453                     st.info("No doctors found for Thyroid.")
454             except Exception as e:
455                 st.error(f"Error loading doctors data: {e}")
456
457 # PCOD Logic
458
459     elif selected == "PCOD":
460         df = load_dataset("pcod")
461         if df is None:
462             return
463         sim = create_fuzzy_pcod(df)
464         st.subheader("PCOD")
465         bmi = st.number_input("BMI", float(df['BMI'].min()), float(df['BMI'].max()), float(df['BMI'].mean()))
466         insulin = st.number_input("Insulin Level", float(df['Insulin_Level'].min()), float(df['Insulin_Level'].max()), float(df['Insulin_Level'].mean()))
467         lh = st.number_input("LH", float(df['LH'].min()), float(df['LH'].max()), float(df['LH'].mean()))
468
469         st.markdown("### 📈 Upload blood Test Report (CSV or Excel)")
```

The image shows a screenshot of the PyCharm Python IDE. The main window displays a code editor with a Python script titled "Medical_Diagnosis_System.py". The script contains 497 numbered lines of code, primarily using the Streamlit library to build a web application for medical diagnosis. The code includes importing libraries, defining functions for fuzzy logic, handling file uploads, and creating Streamlit components like buttons and dataframes. The PyCharm interface features a toolbar with various icons for file operations, a search bar, and a status bar at the bottom.

```
463     sim = create_fuzzy_pcod(df)
464     st.subheader("PCOD")
465     bmi = st.number_input("BMI", float(df['BMI'].min()), float(df['BMI'].max()), float(df['BMI'].mean()))
466     insulin = st.number_input("Insulin Level", float(df['Insulin_Level'].min()), float(df['Insulin_Level'].max()), float(df['Insulin_Level'].mean()))
467     lh = st.number_input("LH", float(df['LH'].min()), float(df['LH'].max()), float(df['LH'].mean()))
468
469     st.markdown("### 📈 Upload blood Test Report (CSV or Excel)")
470     uploaded_file = st.file_uploader("Choose a file", type=["csv", "xlsx"])
471
472     if uploaded_file:
473         try:
474             if uploaded_file.name.endswith('.csv'):
475                 user_data = pd.read_csv(uploaded_file)
476             else:
477                 user_data = pd.read_excel(uploaded_file)
478
479             st.success("✅ Report uploaded successfully!")
480             st.dataframe(user_data)
481
482             # Assuming the report has correct column names
483             bmi = user_data['BMI'][0]
484             insulin = user_data["Insulin_Level"][0]
485             lh = user_data['LH'][0]
486
487         except Exception as e:
488             st.error(f"⚠️ Error reading file: {e}")
489             return
490
491     col1, col2, col3 = st.columns([1, 1, 1])
492     with col1:
493         predict = st.button("Predict PCOD Status", key="predict_pcod")
494     with col3:
495         show_doctors = st.button("Show Recommended Doctors", key="show_pcod_doctors")
496
497     if predict:
```

The image shows a screenshot of the PyCharm IDE. The title bar at the top reads "Systemwith_3Disease". The left sidebar contains various icons for file operations like Open, Save, Find, and Settings. The main editor window displays a Python script titled "Medical_Diagnosis_System.py". The code implements a medical diagnosis system for PCOD based on user input for BMI, insulin levels, and LH levels. It includes logic to predict PCOD risk and provide health warnings for users. The script uses the Streamlit library for presentation.

```
Medical_Diagnosis_System.py 5 X
497     if predict:
498         sim.input['bmi'] = bmi
499         sim.input['insulin'] = insulin
500         sim.input['lh'] = lh
501         sim.compute()
502         risk = sim.output['pcod_risk']
503         diagnosis = "Yes" if risk >= 50 else "No"
504         st.success(f"Predicted PCOD Status: {diagnosis}")
505
506         warnings = []
507         if bmi > 25:
508             warnings.append("⚠️ High BMI ⚠️ Obesity is a major risk factor for PCOD.")
509         if bmi < 18.5:
510             warnings.append("⚠️ Low BMI ⚠️ Consider monitoring nutritional health.")
511         if insulin > 15:
512             warnings.append("⚠️ Elevated Insulin ⚠️ May indicate insulin resistance.")
513         if insulin < 10:
514             warnings.append("⚠️ Very Low Insulin ⚠️ May need medical attention.")
515         if lh > 9:
516             warnings.append("⚠️ High LH ⚠️ Can contribute to irregular ovulation.")
517         if warnings:
518             warning_html = '<div style="background-color:#fbeaea;padding:10px;border-radius:8px;">' + \
519                         '<br>'.join(f'<span style="color:#d00000;>{w}</span>' for w in warnings) + \
520             '</div>'
521             st.markdown(warning_html, unsafe_allow_html=True)
522 elif show_doctors:
523     st.subheader("Recommended Doctors for PCOD")
524     try:
525         doctor_data = pd.read_csv("indian_doctors_dataset.csv")
526         p_doctors = doctor_data[
527             doctor_data['Specialist'].str.lower().str.contains("pcod", na=False)
528         ]
529         if not p_doctors.empty:
530             grouped = p_doctors.groupby('Clinic Address')
```

```
# Medical_Diagnosis_System.py 5 ×

529         if not p_doctors.empty:
530             grouped = p_doctors.groupby('clinic Address')
531             count = 0
532             for name, group in grouped:
533                 st.markdown(f"##### • Location: {name}")
534                 top5 = group.head(3)[['Doctor Name', 'Specialist', 'Phone Number', 'Email']]
535                 st.dataframe(top5, use_container_width=True)
536                 count += 1
537                 if count >= 3:
538                     break
539             else:
540                 st.info("No doctors found for PCOD.")
541         except Exception as e:
542             st.error(f"Error loading doctors data: {e}")
543
544 # Anxiety logic
545
546 elif selected == "Anxiety":
547     df = load_dataset("anxiety")
548     if df is None:
549         return
550     sim = create_fuzzy_anxiety(df)
551     st.subheader("Anxiety")
552     sleep = st.number_input("Sleep Hours", float(df['SleepHours'].min()), float(df['SleepHours'].max()), float(df['SleepHours'].mean()))
553     heart_rate = st.number_input("Heart Rate", int(df['HeartRate'].min()), int(df['HeartRate'].max()), int(df['HeartRate'].mean()))
554
555     fatigue_str = st.selectbox("Fatigue", ["No", "Yes"])
556     fatigue = 1 if fatigue_str == "Yes" else 0
557
558     irritability_str = st.selectbox("Irritability", ["No", "Yes"])
559     irritability = 1 if irritability_str == "Yes" else 0
560
561     restlessness_str = st.selectbox("Restlessness", ["No", "Yes"])
562     restlessness = 1 if restlessness_str == "Yes" else 0
563
```

Medical_Diagnosis_System.py 5 X

```
561     restlessness_str = st.selectbox("Restlessness", ["No", "Yes"])
562     restlessness = 1 if restlessness_str == "Yes" else 0
563
564     score = st.slider("GAD-7 Score", int(df['ScoreGAD7'].min()), int(df['ScoreGAD7'].max()), int(df['ScoreGAD7'].mean()))
565
566     col1, col2, col3 = st.columns([1, 1, 1])
567     with col1:
568         predict = st.button("Predict Anxiety Status", key="predict_anxiety")
569     with col3:
570         show_doctors = st.button("Show Recommended Doctors", key="show_anxiety_doctors")
571
572     if predict:
573         sim.input['sleep'] = sleep
574         sim.input['heart_rate'] = heart_rate
575         sim.input['fatigue'] = fatigue
576         sim.input['irritability'] = irritability
577         sim.input['restlessness'] = restlessness
578         sim.input['score'] = score
579         sim.compute()
580
581         risk = sim.output['anxiety_risk']
582         diagnosis = "Yes" if risk >= 50 else "No"
583         st.success(f"Predicted Anxiety Status: {diagnosis}")
584
585         warnings = []
586
587         if restlessness == 1: # Assuming 1 = Yes, 0 = No
588             warnings.append("⚠️ High Restlessness ⚠️ Major contributor to anxiety.")
589         if sleep < 6:
590             warnings.append("⚠️ Poor Sleep ⚠️ Can worsen anxiety.")
591         if fatigue == 1:
592             warnings.append("⚠️ Fatigue ⚠️ Often linked with anxiety symptoms.")
593         if irritability == 1:
594             warnings.append("⚠️ Irritability ⚠️ A strong emotional indicator.")
595         if sleep > 8:
```

Medical_Diagnosis_System.py 5 X

```
594     warnings.append("⚠️ Irritability - A strong emotional indicator.")
595     if sleep > 9:
596         warnings.append("⚠️ Oversleeping - May signal underlying issues.")
597     if risk < 20:
598         warnings.append("⚠️ Low Stress - That's good, keep it balanced!")
599
600     if warnings:
601         warning_html = '<div style="background-color:#fbeaea;padding:10px;border-radius:8px;">' + \
602             '\n'.join(f'<span style="color:#d00000;">{w}</span>' for w in warnings) + \
603             '</div>'
604         st.markdown(warning_html, unsafe_allow_html=True)
605 elif show_doctors:
606     st.subheader("Recommended Doctors for Anxiety")
607     try:
608         doctor_data = pd.read_csv("indian_doctors_dataset.csv")
609         a_doctors = doctor_data[
610             doctor_data['Specialist'].str.lower().str.contains("anxiety", na=False)
611         ]
612         if not a_doctors.empty:
613             grouped = a_doctors.groupby('Clinic Address')
614             count = 0
615             for name, group in grouped:
616                 st.markdown(f"##### 🌍 Location: {name}")
617                 top5 = group.head(3)[['Doctor Name', 'Specialist', 'Phone Number', 'Email']]
618                 st.dataframe(top5, use_container_width=True)
619                 count += 1
620                 if count >= 3:
621                     break
622             else:
623                 st.info("No doctors found for Anxiety.")
624     except Exception as e:
625         st.error(f"Error loading doctors data: {e}")
626 if __name__ == "__main__":
627     main()
628
```



HEART DISEASE

Main Menu

 Home

 Heart Disease

 Diabetes

 Thyroid

 PCOD

 Anxiety

 Dataset

 Doctors

Medical Diagnosis System

Welcome to your personalized health assistant.

This intelligent system is designed to help you assess your health and detect early signs of common medical conditions.

Using smart algorithms and real clinical data, it can predict the likelihood of:

Heart Disease

Diabetes

Thyroid Disorders

Polycystic Ovarian Disease (PCOD)

Take a step towards better health by navigating through the system.

Enter your health parameters and receive quick, reliable insights — all in one place

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

Medical Diagnosis System

Heart Disease

Age

54

- +

Cholesterol

246

- +

Max Heart Rate

149

- +

Chest Pain Type (0: Typical, 1: Atypical, 2: Non-anginal, 3: Asymptomatic)

1

0

3

Resting Blood Pressure

131

- +

Predict Heart Status

Show Recommended Doctors

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

54

- +

Cholesterol

246

- +

Max Heart Rate

149

- +

Chest Pain Type (0: Typical, 1: Atypical, 2: Non-anginal, 3: Asymptomatic)

1

0

3

Resting Blood Pressure

131

- +

Predict Heart Status

Show Recommended Doctors

Predicted Heart Disease Risk: No

⚠️ High Cholesterol – Increased risk of heart disease.

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

Resting Blood Pressure

131

3

Predict Heart Status

Show Recommended Doctors

Recommended Doctors for Heart Disease

📍 Location: AXZ Hospital, Hyderabad

	Doctor Name	Specialist	Phone Number	Email
24	Dr. Vikram Sinha	Heart	+91-8030327890	dr.vikram.sinha@clinicindia.com

📍 Location: CSR Hospital, Bengaluru

	Doctor Name	Specialist	Phone Number	Email
25	Dr. Nitin Aggarwal	Heart	+91-9298044361	dr..nitin.aggarwal@clinicindia.com

📍 Location: EGN Hospital, Mumbai

	Doctor Name	Specialist	Phone Number	Email
28	Dr. Rita Malhotra	Heart	+91-9187027306	dr.rita.malhotra@clinicindia.com



DIABETES DISEASE

 Medical Diagnosis System Main Menu Home Heart Disease Diabetes Thyroid PCOD Anxiety Dataset Doctors

Diabetes

Glucose

120

-

+

BMI

31.99

-

+

Age

33

-

+

Blood Pressure

69

-

+

 Upload Blood Test Report (CSV or Excel)

Choose a file



Drag and drop file here

Limit 200MB per file • CSV, XLSX

Browse files

Predict Diabetes Status

Show Recommended Doctors

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

31.99

- +

Age

33

- +

Blood Pressure

69

- +



Upload Blood Test Report (CSV or Excel)

Choose a file



Drag and drop file here

Limit 200MB per file • CSV, XLSX

[Browse files](#)

[Predict Diabetes Status](#)

[Show Recommended Doctors](#)

Predicted Diabetes Status: Yes

⚠️ High BMI

Main Menu

- Home
- Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

69

-

+

Upload Blood Test Report (CSV or Excel)

Choose a file



Drag and drop file here

Limit 200MB per file • CSV, XLSX

Browse files



Nirja_upadhyay.csv 125.0B



✓ Report uploaded successfully!

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	111	120	6	151	90	4	

Predict Diabetes Status

Show Recommended Doctors

Predicted Diabetes Status: Yes

⚠ High BMI

⚠ High Blood pressure

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

Predict Diabetes Status

Browse files

Show Recommended Doctors

Recommended Doctors for Diabetes

📍 Location: AYC Hospital, Chennai

	Doctor Name	Specialist	Phone Number	Email
10	Dr. Sunita Rao	Diabetes	+91-7727249681	dr..sunita.rao@clinicindia.com

📍 Location: BXJ Hospital, Ahmedabad

	Doctor Name	Specialist	Phone Number	Email
4	Dr. Shalini Sinha	Diabetes	+91-7683340562	dr..shalini.sinha@clinicindia.com

📍 Location: EEG Hospital, Hyderabad

	Doctor Name	Specialist	Phone Number	Email
7	Dr. Amit Bhalla	Diabetes	+91-8096450753	dr..amit.bhalla@clinicindia.com



THYROID DISEASE

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

Medical Diagnosis System

Thyroid

TSH (mIU/L)

2.48

- +

T3 (ng/dL)

119.36

- +

T4 (μ g/dL)

8.16

- +

Upload blood Test Report (CSV or Excel)

Choose a file



Drag and drop file here

Limit 200MB per file • CSV, XLSX

Browse files

Predict Thyroid Status

Show Recommended Doctors

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

TSH (mIU/L)

5.00

- +

T3 (ng/dL)

119.36

- +

T4 (μ g/dL)

8.16

- +

Upload blood Test Report (CSV or Excel)

Choose a file



Drag and drop file here

Limit 200MB per file • CSV, XLSX

Browse files

Predict Thyroid Status

Show Recommended Doctors

Predicted Thyroid Status: Yes

⚠️ High TSH – Could indicate hypothyroidism.

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

Upload blood Test Report (CSV or Excel)

Choose a file



Drag and drop file here

Limit 200MB per file • CSV, XLSX

Browse files



Sandeep_parmaar.csv 85.0B



Report uploaded successfully!

	TSH (mIU/L)	T3 (ng/dL)	T4 (μg/dL)	Diagnosis
0	4.26	94.5	7.3	Subclinical Hypothyroid

Predict Thyroid Status

Show Recommended Doctors

Predicted Thyroid Status: Yes

High TSH - Could indicate hypothyroidism.

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

Limit 200MB per file • CSV, XLSX

Predict Thyroid Status

Show Recommended Doctors

Recommended Doctors for Thyroid

📍 Location: EFM Hospital, Bengaluru

	Doctor Name	Specialist	Phone Number	Email
31	Dr. Suraj Mehta	Thyroid	+91-8048523854	dr.suraj.mehta@clinicindia.com

📍 Location: ILF Hospital, Bengaluru

	Doctor Name	Specialist	Phone Number	Email
38	Dr. Lata Agarwal	Thyroid	+91-9373834078	dr.lata.agarwal@clinicindia.com

📍 Location: MVW Hospital, Delhi

	Doctor Name	Specialist	Phone Number	Email
40	Dr. Sneha Iyer	Thyroid	+91-8555630036	dr.sneha.iyer@clinicindia.com



PCOD

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

Medical Diagnosis System

PCOD

BMI

26.45

-

+

Insulin Level

19.28

-

+

LH

9.14

-

+

Upload blood Test Report (CSV or Excel)

Choose a file



Drag and drop file here

Limit 200MB per file • CSV, XLSX

[Browse files](#)

[Predict PCOD Status](#)

[Show Recommended Doctors](#)

Main Menu

- Home
- Heart Disease
- Diabetes
- Thyroid
- PCOD**
- Anxiety
- Dataset
- Doctors

PCOD

BMI
26.45 - +

Insulin Level
19.28 - +

LH
9.14 - +

Upload blood Test Report (CSV or Excel)

Choose a file

Drag and drop file here
Limit 200MB per file • CSV, XLSX Browse files

Predict PCOD Status Show Recommended Doctors

Predicted PCOD Status: Yes

⚠ High BMI – Obesity is a major risk factor for PCOD.
⚠ Elevated Insulin – May indicate insulin resistance.
⚠ High LH – Can contribute to irregular ovulation.

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

Choose a file



Drag and drop file here

Limit 200MB per file • CSV, XLSX

Browse files



Kavya_Kulkarni.csv 174.0B



Report uploaded successfully!

	Age	BMI	Irregular_Menstruation	Hair_Growth	Skin_Darkening	Acne	Weight_Gain	Hair_Los
0	20	35.7		1	1	1	1	1

Predict PCOD Status

Show Recommended Doctors

Predicted PCOD Status: No

⚠️ High BMI – Obesity is a major risk factor for PCOD.

⚠️ Elevated Insulin – May indicate insulin resistance.

⚠️ High LH – Can contribute to irregular ovulation.

U	ZU	55.7		1	1	1	1	1
---	----	------	--	---	---	---	---	---

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

Predict PCOD Status

Show Recommended Doctors

Recommended Doctors for PCOD

📍 Location: CQA Hospital, Delhi

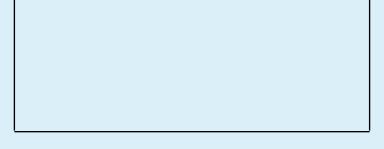
	Doctor Name	Specialist	Phone Number	Email
53	Dr. Savita Bansal	PCOD	+91-7041133410	dr..savita.bansal@clinicindia.com

📍 Location: CTC Hospital, Mumbai

	Doctor Name	Specialist	Phone Number	Email
46	Dr. Neelam Joshi	PCOD	+91-8160672786	dr..neelam.joshi@clinicindia.com

📍 Location: JDD Hospital, Chennai

	Doctor Name	Specialist	Phone Number	Email
58	Dr. Meera Reddy	PCOD	+91-9075898580	dr..meera.reddy@clinicindia.com



ANXIETY DISEASE

 Medical Diagnosis System Main Menu Home Heart Disease Diabetes Thyroid PCOD Anxiety Dataset Doctors

Anxiety

Sleep Hours

6.11

-

+

Heart Rate

89

-

+

Fatigue

No

▼

Irritability

No

▼

Restlessness

No

▼

GAD-7 Score

 Predict Anxiety Status Show Recommended Doctors

Main Menu

- Home
- Heart Disease
- Diabetes
- Thyroid
- PCOD
- Anxiety
- Dataset
- Doctors

Anxiety

Sleep Hours

5.00

- +

Heart Rate

89

- +

Fatigue

No

▼

Irritability

No

▼

Restlessness

No

▼

GAD-7 Score

0

10

20

Predict Anxiety Status

Show Recommended Doctors

Predicted Anxiety Status: Yes

⚠ Poor Sleep – Can worsen anxiety.

Main Menu

Home

Heart Disease

Diabetes

Thyroid

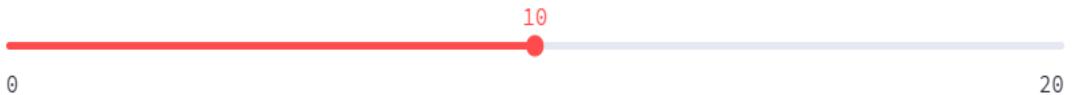
PCOD

Anxiety

Dataset

Doctors

GAD-7 Score



Predict Anxiety Status

Show Recommended Doctors

Recommended Doctors for Anxiety

📍 Location: BTM Hospital, Mumbai

	Doctor Name	Specialist	Phone Number	Email
62	Dr. Harish Nair	Anxiety	+91-9911600257	dr.harish.nair@clinicindia.com

📍 Location: EBX Hospital, Kolkata

	Doctor Name	Specialist	Phone Number	Email
71	Dr. Tanuj Shah	Anxiety	+91-9947188880	dr.tanuj.shah@clinicindia.com

📍 Location: EHD Hospital, Hyderabad

	Doctor Name	Specialist	Phone Number	Email
67	Dr. Laxmi Kapoor	Anxiety	+91-8099877219	dr.laxmi.kapoor@clinicindia.com



HEART DATASET

Medical Diagnosis System

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

View Dataset

Select which dataset to display

Heart

	age	sex	cp	trestbps	Cholesterol	fbst	restecg	thalach	exang	oldpeak	slope	ca
0	63	1	3	145	233	1	0	150	0	2.3	0	
1	37	1	2	130	250	0	1	187	0	3.5	0	
2	41	0	1	130	204	0	0	172	0	1.4	2	
3	56	1	1	120	236	0	1	178	0	0.8	2	
4	57	0	0	120	354	0	1	163	1	0.6	2	
5	57	1	0	140	192	0	1	148	0	0.4	1	
6	56	0	1	140	294	0	0	153	0	1.3	1	
7	44	1	1	120	263	0	1	173	0	0	2	
8	52	1	2	172	199	1	1	162	0	0.5	2	
9	57	1	2	150	168	0	1	174	0	1.6	2	



DIABETES DATASET

 Medical Diagnosis System Main Menu Home Heart Disease Diabetes Thyroid PCOD Anxiety Dataset Doctors

View Dataset

Select which dataset to display

Diabetes



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	
1	1	85	66	29	0	26.6	0.351	
2	8	183	64	0	0	23.3	0.672	
3	1	89	66	23	94	28.1	0.167	
4	0	137	40	35	168	43.1	2.288	
5	5	116	74	0	0	25.6	0.201	
6	3	78	50	32	88	31	0.248	
7	10	115	0	0	0	35.3	0.134	
8	2	197	70	45	543	30.5	0.158	
9	8	125	96	0	0	0	0.232	



THYROID DISEASE

 Medical Diagnosis System Main Menu Home Heart Disease Diabetes Thyroid PCOD Anxiety Dataset DoctorsView Dataset 

Select which dataset to display

Thyroid

	TSH	T3	T4	Diagnosis	Thyroid (1=Yes, 0=No)
0	3.99	95.1	9.5	Normal	0
1	2.09	103.2	6.2	Normal	0
2	4.44	142.4	9.7	Subclinical Hypothyroid	1
3	7.07	138.3	10.7	Subclinical Hypothyroid	1
4	1.8	119.4	8.8	Normal	0
5	1.8	123.5	11.8	Normal	0
6	7.24	158.3	6.5	Subclinical Hypothyroid	1
7	4.8	102.3	5.5	Subclinical Hypothyroid	1
8	1.09	136.4	4.4	Unclear	1
9	4.13	113.9	11	Subclinical Hypothyroid	1



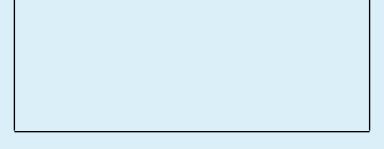
PCOD DISEASE

 Medical Diagnosis System Main Menu Home Heart Disease Diabetes Thyroid PCOD Anxiety Dataset Doctors

Select which dataset to display

PCOD

	Age	BMI	Irregular_Menstruation	Hair_Growth	Skin_Darkening	Acne	Weight_Gain	Hair_Los
0	38	23.5	0	0	1	0	0	
1	18	24.5	0	0	1	1	0	
2	33	22.1	0	0	0	0	0	
3	30	28.9	1	1	0	0	1	
4	31	22.9	0	0	0	0	0	
5	38	24.6	0	0	0	1	0	
6	29	25.7	0	0	0	0	0	
7	18	19.4	0	0	0	0	0	
8	32	30.6	1	0	1	0	1	
9	38	23.9	0	0	0	1	0	



ANXIETY DISEASE

 Medical Diagnosis System Main Menu Home Heart Disease Diabetes Thyroid PCOD Anxiety Dataset DoctorsView Dataset 

Select which dataset to display

Anxiety

	Age	Gender	SleepHours	HeartRate	Fatigue	Irritability	Restlessness	Concentration	PanicAttack
0	38	Female	3.4	76	Yes	No	No	No	Yes
1	39	Other	4.7	60	No	Yes	No	Yes	Yes
2	22	Male	7.2	113	No	No	No	Yes	No
3	36	Male	3.2	106	Yes	No	No	No	No
4	37	Male	6	87	No	No	No	Yes	Yes
5	32	Other	8.6	76	No	Yes	Yes	No	No
6	40	Other	3.3	74	Yes	No	No	No	Yes
7	35	Other	7.2	88	Yes	No	No	No	No
8	40	Female	6.4	65	No	No	No	Yes	Yes
9	34	Other	7.1	83	Yes	Yes	Yes	No	No



DOCTOR DATASET

Main Menu

Home

Heart Disease

Diabetes

Thyroid

PCOD

Anxiety

Dataset

Doctors

Medical Diagnosis System

Doctors' Dataset

	Doctor Name	Specialist	Phone Number	Email	Clinic Address
0	Dr. Aarti Sharma	Diabetes	+91-9254540279	dr..aarti.sharma@clinicindia.com	ZKB Hospital, Hydera
1	Dr. Rajeev Kapoor	Diabetes	+91-7518811307	dr..rajeev.kapoor@clinicindia.com	SYL Hospital, Pune
2	Dr. Nidhi Verma	Diabetes	+91-7826209413	dr..nidhi.verma@clinicindia.com	PPE Hospital, Pune
3	Dr. Manoj Reddy	Diabetes	+91-8485501366	dr..manoj.reddy@clinicindia.com	NEB Hospital, Chenn
4	Dr. Shalini Sinha	Diabetes	+91-7683340562	dr..shalini.sinha@clinicindia.com	BXJ Hospital, Ahmed
5	Dr. Prakash Iyer	Diabetes	+91-9841525981	dr..prakash.iyer@clinicindia.com	IOZ Hospital, Hyderal
6	Dr. Kavita Joshi	Diabetes	+91-8468099338	dr..kavita.joshi@clinicindia.com	NNJ Hospital, Delhi
7	Dr. Amit Bhalla	Diabetes	+91-8096450753	dr..amit.bhalla@clinicindia.com	EEG Hospital, Hydera
8	Dr. Meena Gupta	Diabetes	+91-7120444659	dr..meena.gupta@clinicindia.com	YUH Hospital, Delhi
9	Dr. Rohit Khurana	Diabetes	+91-8141160083	dr..rohit.khurana@clinicindia.com	UKM Hospital, Chenn