HEALTHCARE DIAGNOSIS AND TREATMENT

**PYTHON FILE :**

# healthcare\_diagnosis.py

def get\_possible\_conditions(symptoms):

condition\_db = {

"Common Cold": {"cough", "sore throat", "runny nose", "congestion", "sneezing"},

"Flu": {"fever", "chills", "muscle aches", "fatigue", "cough"},

"COVID-19": {"fever", "dry cough", "loss of taste", "loss of smell", "fatigue"},

"Allergy": {"sneezing", "runny nose", "itchy eyes", "congestion"},

"Migraine": {"headache", "nausea", "sensitivity to light", "sensitivity to sound"},

}

matched\_conditions = []

for condition, condition\_symptoms in condition\_db.items():

match\_count = len(symptoms & condition\_symptoms)

if match\_count >= 2: # Basic threshold for match

matched\_conditions.append((condition, match\_count))

matched\_conditions.sort(key=lambda x: -x[1]) # Sort by best match

return matched\_conditions

def suggest\_treatment(condition):

treatment\_db = {

"Common Cold": "Rest, drink fluids, and use over-the-counter cold remedies.",

"Flu": "Rest, antiviral medications if early, and stay hydrated.",

"COVID-19": "Isolate, rest, monitor oxygen levels, and consult a healthcare provider.",

"Allergy": "Antihistamines, avoid allergens, and nasal sprays.",

"Migraine": "Pain relievers, rest in a dark room, and avoid triggers.",

}

return treatment\_db.get(condition, "Consult a healthcare provider for diagnosis and treatment.")

def main():

print("Welcome to the Healthcare Diagnosis System.")

user\_input = input("Enter symptoms separated by commas: ")

symptoms = {s.strip().lower() for s in user\_input.split(",")}

possible\_conditions = get\_possible\_conditions(symptoms)

if not possible\_conditions:

print("No conditions matched your symptoms. Please consult a doctor.")

return

print("\nPossible Diagnoses:")

for condition, score in possible\_conditions:

print(f"- {condition} (matched {score} symptoms)")

print("\nSuggested Treatment for top condition:")

top\_condition = possible\_conditions[0][0]

print(suggest\_treatment(top\_condition))

if \_\_name\_\_ == "\_\_main\_\_":

main()

**\*\*\***