VU21CSEN0101734

G.SAI SRI ANJALI

**2) THEMES** **:** AGRICULTURE

**PROBLEM STATEMENT :** CROP AND SOIL MANAGEMENT SYSTEM:

FARMERS FACE SEVERAL CHALLENGES RELATED TO CROP SELECTION,SOIL MANAGEMENT,DISEASE IDENTIFICATIONAND OTHER FACTORS, WHICH CAN IMPACT AGRICULTURAL PRODUCTIVITY AND SUSTAINABILTY. TO ADDRESS THESE VHALLENGES, WE NEED AN APPLICATION TO HELP FARMERS FOR FULL-FLEDGED FARMING.

**RESULT :**

ZERO HUNGER

**ABOUT CROP AND MANAGEMENT SYSYTEM:**

Crop production and management ensure the availability of food to everyone in the world. Crops are a primary source of food for humans and crop production and management food availability can be ensured to the growing population of the world. It also makes for a source of income for many.

**CODE:**

# Mock data for crops, soil types, and diseases

crop\_data = {

"wheat": {"soil": "loam", "season": "winter", "pH\_range": (6.0, 7.5)},

"rice": {"soil": "clay", "season": "rainy", "pH\_range": (5.0, 6.5)},

"maize": {"soil": "sandy\_loam", "season": "summer", "pH\_range": (5.5, 7.0)},

}

soil\_data = {

"loam": {"pH": 6.8, "water\_retention": "medium"},

"clay": {"pH": 5.8, "water\_retention": "high"},

"sandy\_loam": {"pH": 6.2, "water\_retention": "low"},

}

# Sample function to suggest crops based on soil type and pH level

def suggest\_crops(soil\_type, soil\_pH):

suggestions = []

for crop, data in crop\_data.items():

if data["soil"] == soil\_type and data["pH\_range"][0] <= soil\_pH <= data["pH\_range"][1]:

suggestions.append(crop)

return suggestions if suggestions else "No suitable crops found."

# Example function to manage soil, suggesting improvements if pH is out of range

def manage\_soil(soil\_type, soil\_pH):

if soil\_type not in soil\_data:

return "Unknown soil type. Please enter a valid soil type."

soil\_info = soil\_data[soil\_type]

if soil\_pH < soil\_info["pH"]:

return f"Increase soil pH by adding lime."

elif soil\_pH > soil\_info["pH"]:

return f"Decrease soil pH by adding sulfur."

else:

return "Soil pH is optimal."

# Sample function for disease identification based on symptoms (mock data)

disease\_data = {

"yellowing\_leaves": "Nitrogen deficiency",

"brown\_spots": "Fungal infection",

}

def identify\_disease(symptoms):

return disease\_data.get(symptoms, "No disease found for the given symptoms.")

# Main function to simulate farmer interaction

def farmer\_assistant(soil\_type, soil\_pH, symptoms=None):

print("Crop and Soil Management System")

print("------------------------------")

print(f"\nSoil Type: {soil\_type}")

print(f"Soil pH: {soil\_pH}")

# Crop suggestions

print("\nSuggested Crops:")

crops = suggest\_crops(soil\_type, soil\_pH)

print(crops)

# Soil management advice

print("\nSoil Management Advice:")

advice = manage\_soil(soil\_type, soil\_pH)

print(advice)

# Disease identification (if symptoms are provided)

if symptoms:

print("\nDisease Identification:")

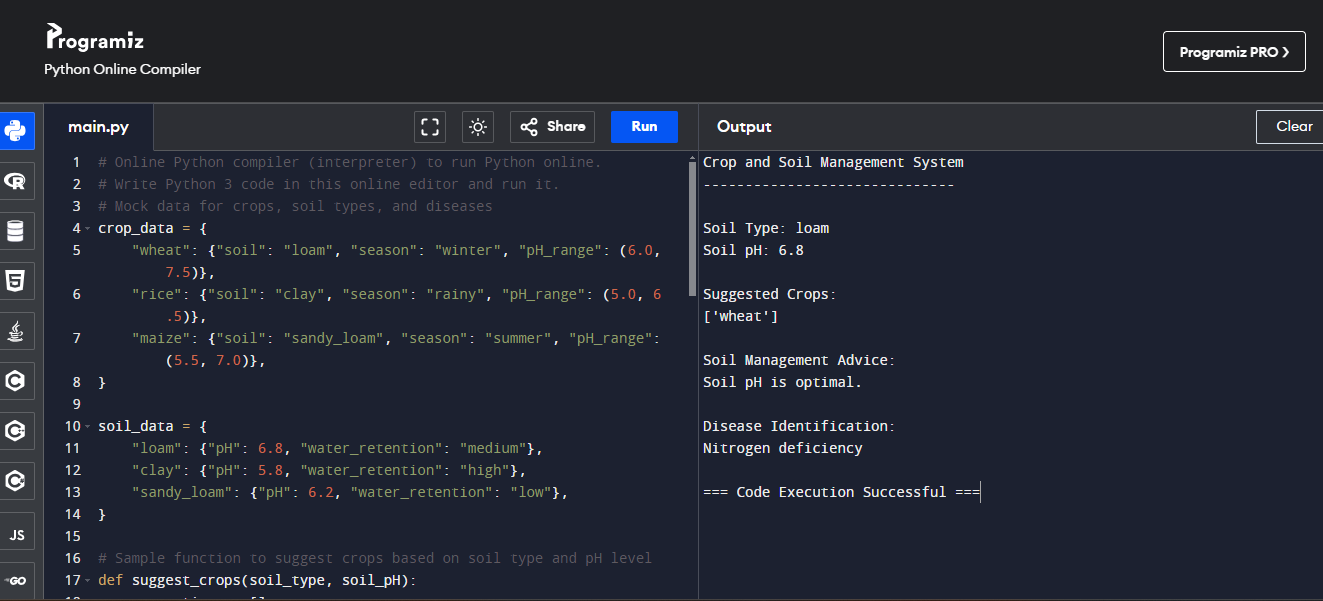
disease = identify\_disease(symptoms)

print(disease)

# Example usage of the system

farmer\_assistant("loam", 6.8, symptoms="yellowing\_leaves")

**OUTPUT :**

****

**THANK YOU**