Python Project: Smart Plant Monitoring System

Description:

A system where users can monitor the health of their plants. The program will ask users for data like soil moisture levels, temperature, and light exposure, and then give feedback on the plant's health and recommendations for care.

This project includes the following features:

Plant Profile Management:

- *Load existing plant data from a saved file.*
- *Create a new plant profile by inputting details like name, soil moisture, temperature, and light exposure.*

Plant Health Evaluation:

- *Automatically evaluate the health of the plant based on input parameters.*
- *Display health status as "Healthy" or "Unhealthy" based on predefined conditions.*

Care Recommendations:

- *Provide care recommendations based on the plant's current conditions.*
- *Offer suggestions on watering, relocating for optimal temperature, and ensuring adequate sunlight.*

Data Persistence:

- *Save the current plant's data to a file for future reference.*
- *Automatically load saved plant data when starting the program, avoiding the need to re-enter details each time.*

```
class Plant:
    def __init__(self, name, soil_moisture, temperature, light_exposure):
        self.name = name
        self.soil_moisture = soil_moisture
        self.temperature = temperature
        self.light_exposure = light_exposure

    def display_health(self):
```

```
        # Simplified logic to evaluate plant health
        if self.soil_moisture > 70 and self.temperature > 20 and self.light_exposure > 6:
            return "Healthy"
        else:
            return "Unhealthy"

    def get_care_recommendation(self):
        recommendations = []
        if self.soil_moisture < 30:
            recommendations.append("Water the plant.")
        if self.temperature < 18:
            recommendations.append("Move the plant to a warmer location.")
        if self.light_exposure < 4:
            recommendations.append("Ensure the plant gets more sunlight.")
        return recommendations

def save_plant_data(plant):
    with open("plant_data.txt", "w") as file:
        file.write(f"{plant.name}\n")
        file.write(f"{plant.soil_moisture}\n")
        file.write(f"{plant.temperature}\n")
        file.write(f"{plant.light_exposure}\n")

def load_plant_data():
    try:
        with open("plant_data.txt", "r") as file:
            name = file.readline().strip()
            soil_moisture = int(file.readline().strip())
```

```
            temperature = int(file.readline().strip())
            light_exposure = int(file.readline().strip())
        return Plant(name, soil_moisture, temperature, light_exposure)
    except FileNotFoundError:
        return None

def user_interface():
    loaded_plant = load_plant_data()

    if loaded_plant:
        print(f"Loaded data for {loaded_plant.name}.")
        plant = loaded_plant
    else:
        name = input("Enter the plant name: ")
        soil_moisture = int(input("Enter soil moisture (0-100): "))
        temperature = int(input("Enter temperature (in Celsius): "))
        light_exposure = int(input("Enter hours of light exposure: "))
        plant = Plant(name, soil_moisture, temperature, light_exposure)

    print(f"Plant Health: {plant.display_health()}")
    recommendations = plant.get_care_recommendation()
    if recommendations:
        print("Care Recommendations:")
        for r in recommendations:
            print(f"- {r}")
    else:
        print("Your plant is in great shape!")
```

```
    save_choice = input("Do you want to save this data? (yes/no): ").strip().lower()
    if save_choice == "yes":
        save_plant_data(plant)
        print("Data saved!")

if __name__ == "__main__":
    user_interface()
```