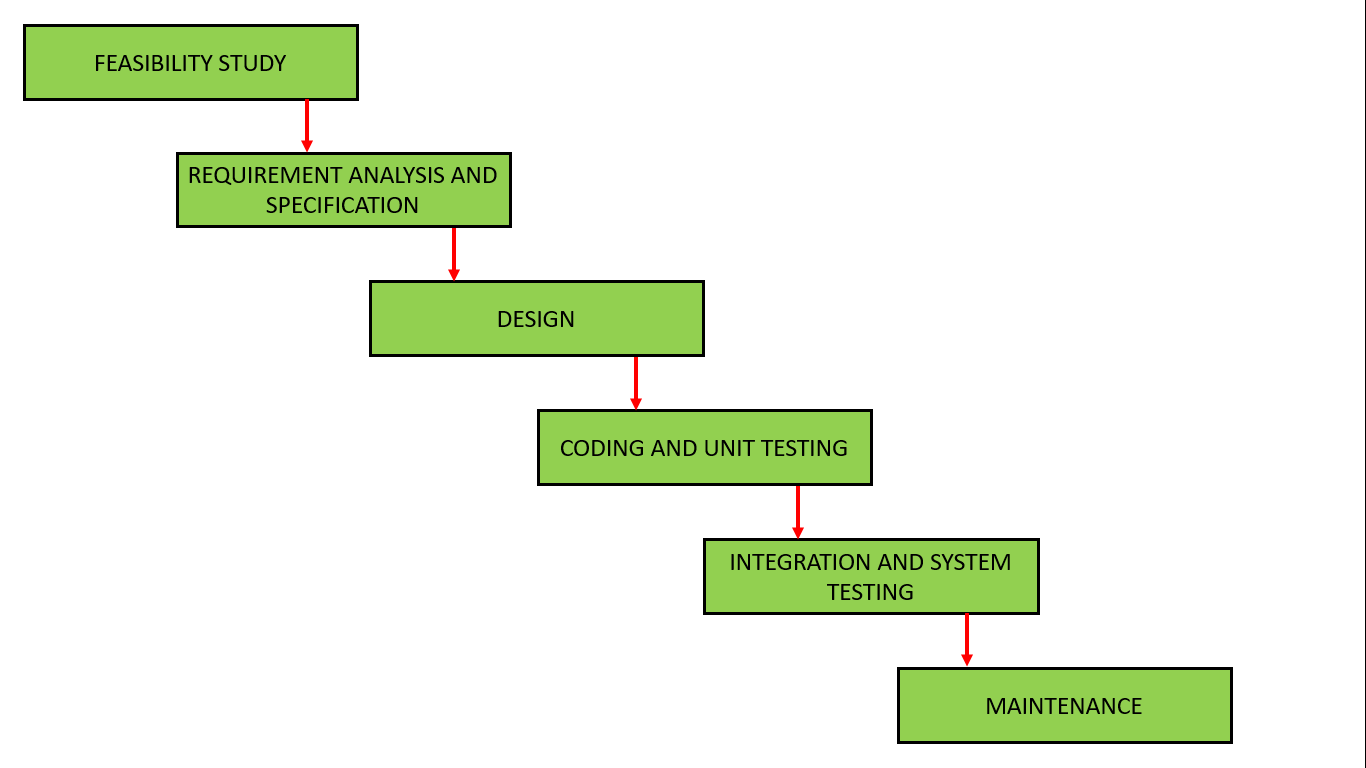
**Develop weather modeling using the quadratic model using Waterfall model**



A weather modeling system using a quadratic model can be developed using the Waterfall model by following a structured, sequential approach. This involves defining requirements, designing the system, implementing the quadratic model, testing, and finally deploying and maintaining the system.

**scope:**

The objective of this project is to develop a simple weather prediction model that estimates **rainfall (in mm)** and **humidity (in %)** over a 24-hour period using a **quadratic regression approach**. By using historical time-based data points, the model fits a quadratic curve that can predict values at every hour of the day

**inputs:**

 **Time (in hours)**

time = [0, 4, 8, 12, 16, 20]

 **Rainfall (in millimeters)**

rainfall = [0, 2, 8, 15, 10, 3]

 **Humidity (in percentage)**

humidity = [95, 85, 70, 60, 75, 90]

**Determine Accuracy Requirements :**

The predictions should be **close to the real values**.

Rainfall can be off by **±2 mm**, and humidity by **±5%**.

We want the model to follow the **general weather pattern** during the day.

Use simple checks or tools like **MAE or RMSE** to see how accurate the model is.

**System Architecture:**

**Input**: Add time, rainfall, and humidity data.

**Processing**: The computer uses a math formula to predict future values.

**Output**: It shows the results in a list and a graph for easy understanding.

**Quadratic Model Design:**

We use the formula **y = a·t² + b·t + c** to predict rainfall or humidity based on time.

**Data Handling:**

We use clean time, rainfall, and humidity data directly, then store the predicted results in lists and show them using print and graphs.

**Implementation:**

**Coding**: The weather model is written using **Python**, using NumPy for math and Matplotlib for graphs.

**Unit Testing**: Each part (like input arrays, formula, and graph code) is tested separately to ensure it works correctly.

**System Testing**: The full program is run to check if it meets the weather prediction goal.

**Verification**: Predicted values are compared with real-world weather data to check if the model is reliable.

**User Acceptance Testing (UAT)**: Users run the program, view outputs, and give feedback on the predictions and graphs.

**Deployment**:  
Run the weather model on a computer or share it online using platforms like **Google Colab**, **GitHub**, or a simple website.

**Maintenance**:  
Keep checking if the program is working well, and **update the data or fix errors** when needed.

# Step 1: Input Data

time = np.array([0, 4, 8, 12, 16, 20])  # Time in hours

rainfall = np.array([0, 2, 8, 15, 10, 3])     # Rainfall in mm

humidity = np.array([95, 85, 70, 60, 75, 90]) # Humidity in %

# Step 2: Fit quadratic models

rain\_coeff = np.polyfit(time, rainfall, 2)

hum\_coeff = np.polyfit(time, humidity, 2)

a\_r, b\_r, c\_r = rain\_coeff

a\_h, b\_h, c\_h = hum\_coeff

print(f"\nDeveloped Quadratic Model for Rainfall:")

print(f"R(t) = {a\_r:.4f}t² + {b\_r:.4f}t + {c\_r:.4f}\n")

print(f"Developed Quadratic Model for Humidity:")

print(f"H(t) = {a\_h:.4f}t² + {b\_h:.4f}t + {c\_h:.4f}\n")

# Step 3: Predict for every hour from 0 to 24

t\_values = np.arange(0, 25, 1)

predicted\_rainfall = a\_r \* t\_values\*\*2 + b\_r \* t\_values + c\_r

predicted\_humidity = a\_h \* t\_values\*\*2 + b\_h \* t\_values + c\_h

print("Predicted Rainfall (mm) and Humidity (%) for 24 Hours:\n")

for t, rain, hum in zip(t\_values, predicted\_rainfall, predicted\_humidity):

    print(f"At {t:02d}:00 hrs -> Rainfall: {rain:.2f} mm, Humidity: {hum:.2f} %")

# Step 4: Plot (if matplotlib is available)

try:

    import matplotlib.pyplot as plt

    plt.figure(figsize=(12, 6))

    # Rainfall Plot

    plt.subplot(1, 2, 1)

    plt.scatter(time, rainfall, color='blue', label='Original Rainfall Data', zorder=5)

    plt.plot(t\_values, predicted\_rainfall, color='red', linestyle='--', label='Rainfall Prediction')

    plt.title('Rainfall Prediction using Quadratic Model')

    plt.xlabel('Time (Hours)')

    plt.ylabel('Rainfall (mm)')

    plt.xticks(np.arange(0, 25, 2))

    plt.grid(True)

    plt.legend()

    # Humidity Plot

    plt.subplot(1, 2, 2)

    plt.scatter(time, humidity, color='green', label='Original Humidity Data', zorder=5)

    plt.plot(t\_values, predicted\_humidity, color='orange', linestyle='--', label='Humidity Prediction')

    plt.title('Humidity Prediction using Quadratic Model')

    plt.xlabel('Time (Hours)')

    plt.ylabel('Humidity (%)')

    plt.xticks(np.arange(0, 25, 2))

    plt.grid(True)

    plt.legend()

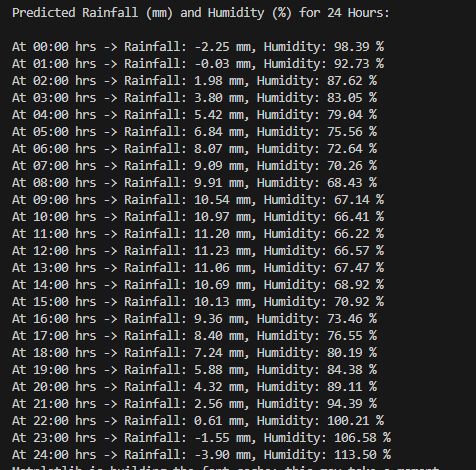
    plt.tight\_layout()

    plt.show()

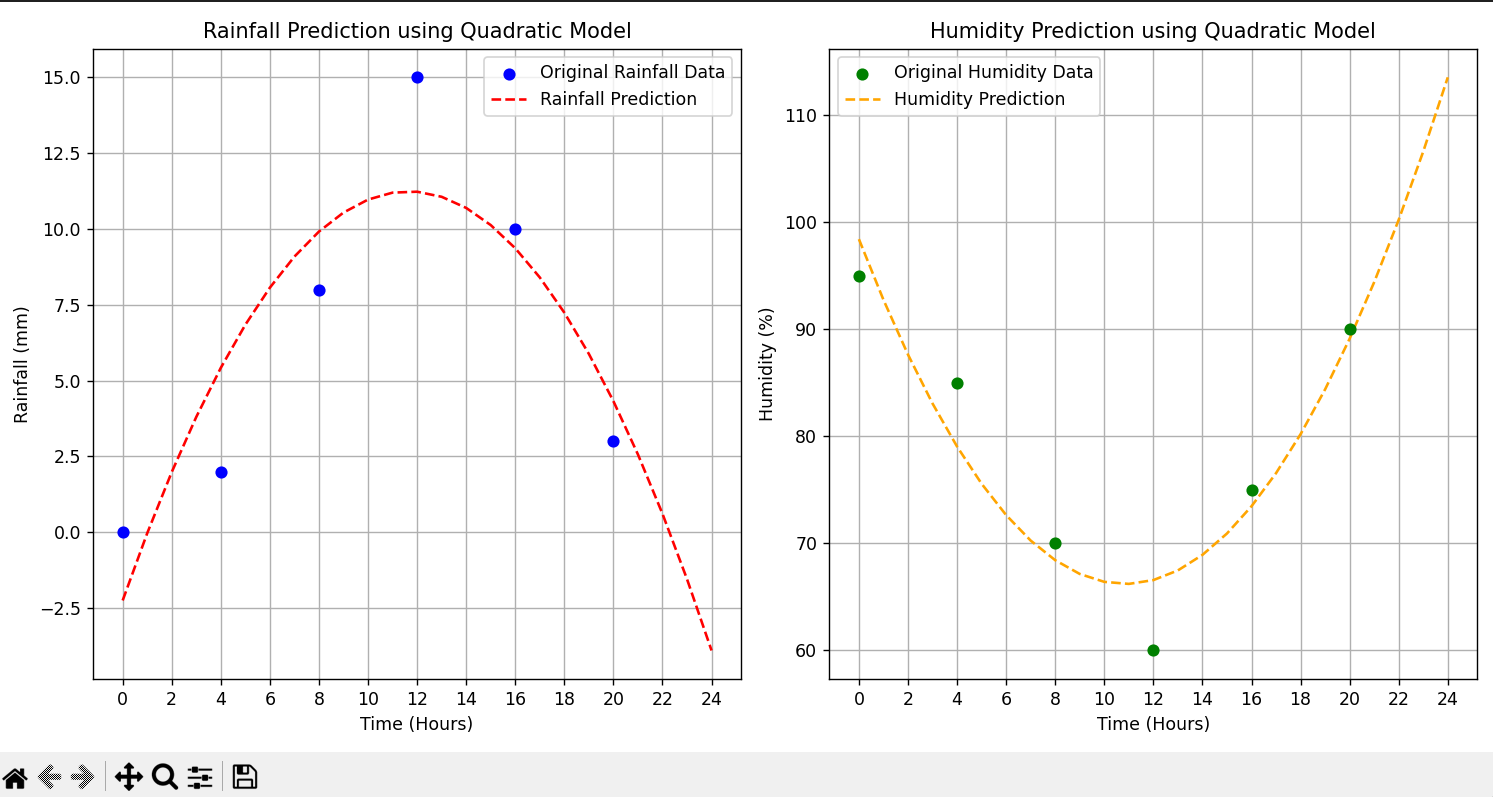
except ImportError:

    print("\nNOTE: 'matplotlib' is not installed. Skipping graph display.")

    print("To install it, run: pip install matplotlib")



**OUTPUT**

****

**Name- PV Hemanth Kumar (2023001468)**