**Assignment: Python Programming for GUI Development**

Name : Rakina.S

Register Number :192324288

Department :AIDS

Date of Submission :26-08-2024

**Problem 1: Real-Time Weather Monitoring System**

**Scenario:**

You are developing a real-time weather monitoring system for a weather forecasting company. The system needs to fetch and display weather data for a specified location.

**Tasks:**

1. **Model the data flow for fetching weather information from an external API and displaying it to the user.**
2. **Implement a Python application that integrates with a weather API (e.g., OpenWeatherMap) to fetch real-time weather data.**
3. **Display the current weather information, including temperature, weather conditions, humidity, and wind speed.**
4. **Allow users to input the location (city name or coordinates) and display the corresponding weather data.**

**Deliverables:**

* Data flow diagram illustrating the interaction between the application and the API.
* Pseudocode and implementation of the weather monitoring system.
* Documentation of the API integration and the methods used to fetch and display weather data.
* Explanation of any assumptions made and potential improvements.

Introduction:

. Describe the purpose of the real-time monitoring system.

. Explain the problems it aims to solve or the improvements it provides over existing solutions.

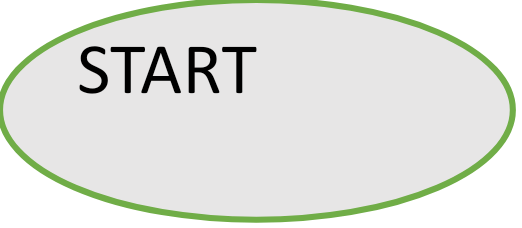
Components:

. data sources ,data ingestion ,processing Engine ,storage ,User Interface

Maintenance:

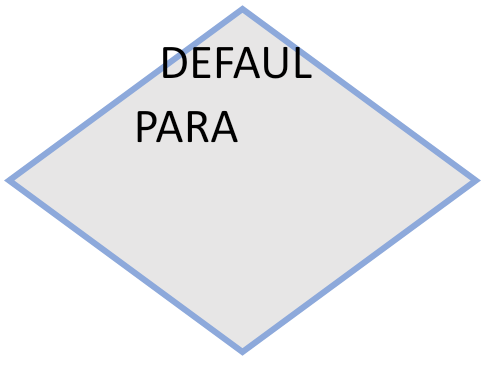
. Outline regular maintenances tasks and schdules.

. Provide guidelines for updating and upgrading the system

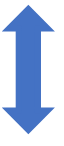
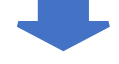
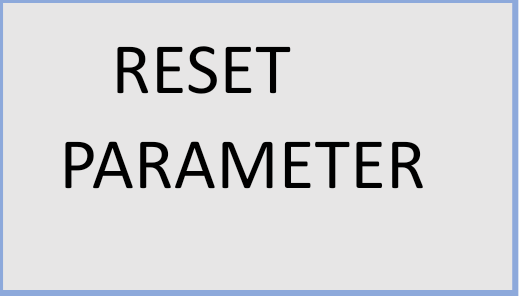
Real-Time Weather Monitoring System

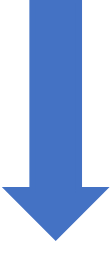
# 1.Data Flow Diagram





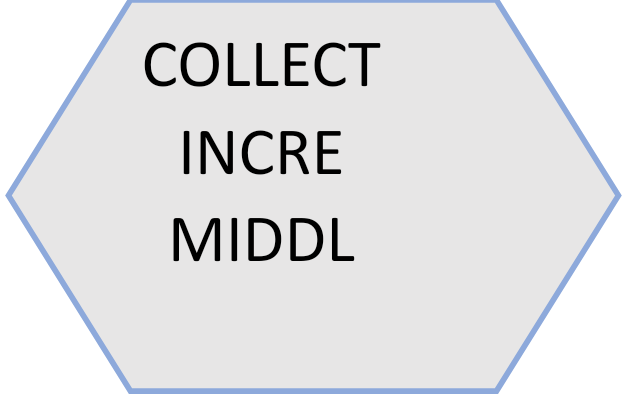
F



T

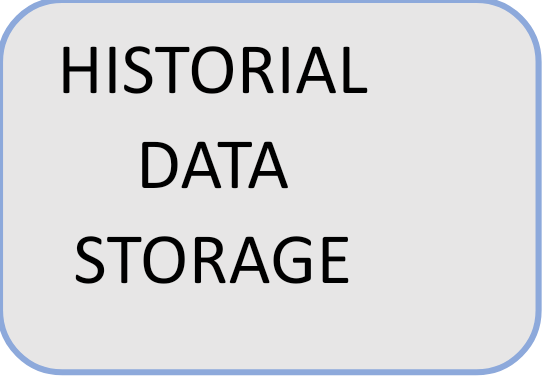
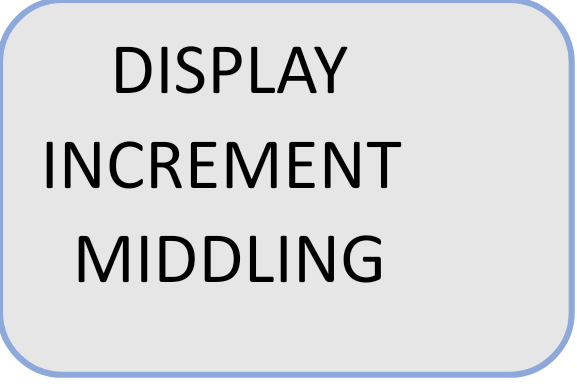


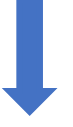
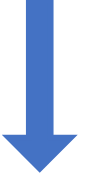




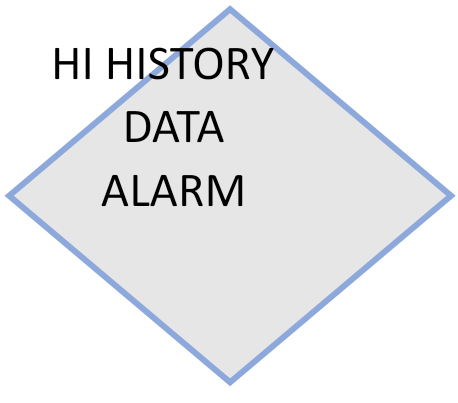
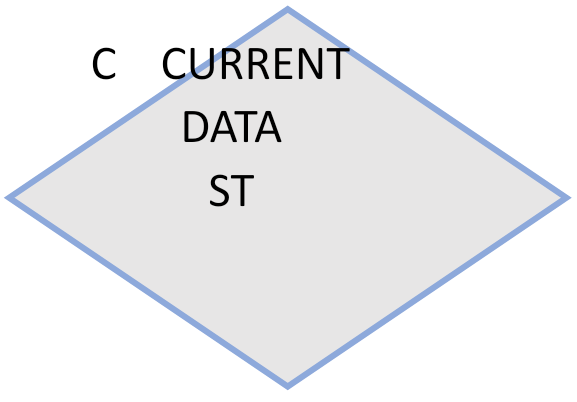
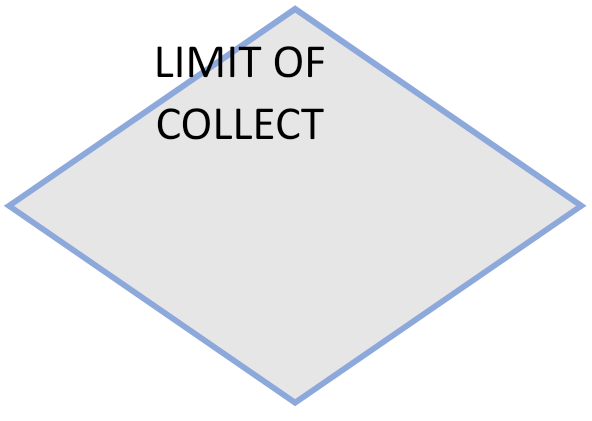




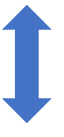




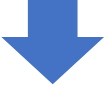


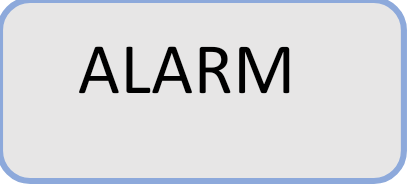


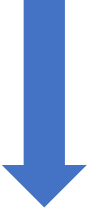
F F

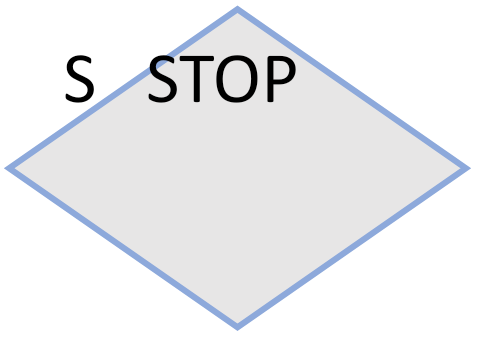


T T T

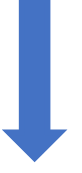


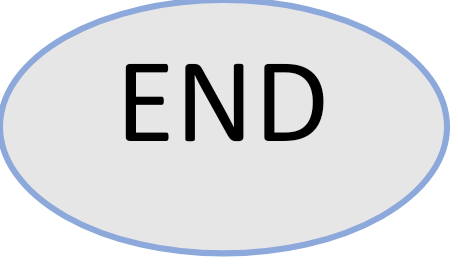






F

T



# 2. Implementation

import requests

api\_key="0b563bad1d7970f0a7d1a2d766f968f2"

user\_input=input("enter the city name:")

weather\_data=requests.get(

   f"https://api.openweathermap.org/data/2.5/weather?q={user\_input}&units=imperial&APPID={api\_key}")

if weather\_data.json()["cod"]=="404":

    print("city not found")

else:

    #print(weather\_data.json())

    weather=weather\_data.json()["weather"][0]["main"]

    temp=round(weather\_data.json()["main"]["temp"])

    humidity=weather\_data.json()["main"]["humidity"]

    windspeed=weather\_data.json()["wind"]["speed"]

    country=weather\_data.json()["sys"]["country"]

    print(f"the weather in {user\_input} is {weather}")

    print(f"the temperture in {user\_input} is {temp}F")

    print(f"the humidity in {user\_input} is {humidity}")

    print(f"the windspeed in {user\_input} is {windspeed}kmph")

    print(f"the {user\_input} is in {country}")

# 3.Display the Current weather information

enter the city: Kurnool

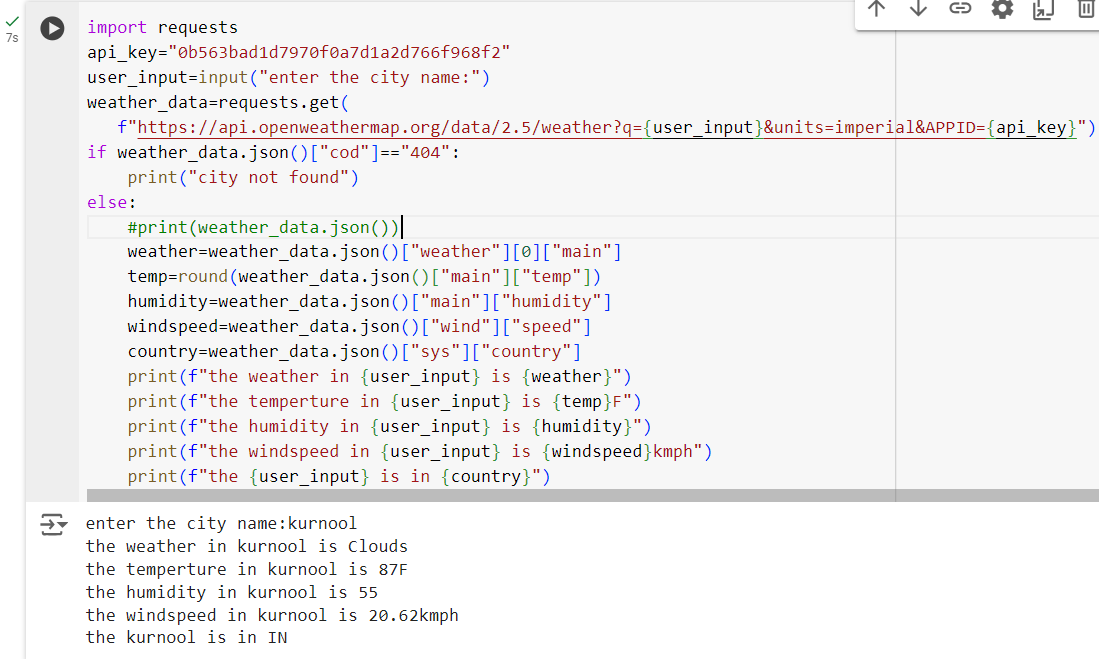
 Temperature (in kelvin unit) = 304.42

 atmospheric pressure (in hPa unit) = 1002

 humidity (in percentage) = 53

 description = overcast clouds

# 4.User Input



**5.Documentation**

**API Overview**

* **Describe the purpose of the API and its main functionalities.**

**Endpoints**

* **List and describe the available API endpoints.**
* **Include request and response formats, parameters, and example requests.**

**Authentication**

* **Explain the authentication methods used for accessing the API.**

**Appendices**

**Glossary**

* **Define key terms and acronyms used in the documentation.**

**References**

* **List any references or additional resources related to the system.**

**Change Log**

* **Maintain a record of changes and updates to the documentation.**

**Contact Information**

* **Provide contact details for support or further inquiries.**

**This structured documentation ensures that users, developers, and stakeholders have a clear understanding of the real-time monitoring system and how to interact with it effectively.**

# 