Abstract: Checking Weather Application

Objective: To create a user-friendly application that allows users to check the weather status of any city.

Technologies Used:

Backend: Python (for data retrieval and storage)

- Utilizes an API key to fetch current weather data.
- Stores weather information in a MongoDB database.

Frontend: Java Swing (using the tkinter GUI module)

- Provides an intuitive interface for user interaction.
- Allows users to input the name of a city.

Key Features:

1. **City Input and Weather Retrieval:**
    - Users enter the name of a city in the input field.
    - The application fetches real-time weather data for that city using an API key.
    - Weather parameters include temperature, humidity, wind speed, and conditions (e.g., sunny, cloudy, rainy).

2. **Data Storage in MongoDB:**
    - The retrieved weather information is stored in a MongoDB database.
    - Each city's data is associated with a timestamp.

3. **Displaying Current Weather:**

- Users can view the current weather status for the specified city.
- The application presents relevant details on the frontend.

4. **Historical Weather Data:**

- Users can access historical weather data for the same city.
- The application displays a limited set of historical records (up to 11 documents) based on timestamps.
- This feature enables users to track weather trends over time.

Implementation Details:

Backend Logic:

- Python scripts handle API requests and data retrieval.
- MongoDB integration ensures efficient data storage.

Frontend Interface:

- Java Swing (with tkinter) provides a responsive and visually appealing GUI.
- Users input the city name and receive weather updates.

Conclusion:

The **Checking Weather Application** aims to enhance user experience by providing accurate weather information, both in real-time and historically.