Sreyas Institute of Engineering and Technology

An Autonomous Institution

Approved by AICTE, Affiliated to JNTUH

Accredited by NAAC-A Grade, NBA (CSE, ECE & ME) & ISO 9001:2015 Certified

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

Object Oriented Programming Through Java Seminar

TOPIC : Weather Forecasting Application (Using API)

Class/Section : B.Tech , III Year - II semester , ECE – 'A'

Academic Year : 2024-25

Presented by Batch : A-19

Nama Rahul - 22VE1A0439

Kanduri Srujan - 23VE5A0402

A.Nithin - 22VE1A0403

Abstract

The Weather Forecasting Application utilizing APIs is designed to provide users with accurate and real-time weather information by integrating data from reliable sources like OpenWeatherMap. This application leverages modern web technologies to fetch and display current weather conditions, forecasts, and other meteorological parameters for any specified location. By harnessing the power of APIs, the system ensures timely updates and broad accessibility, making it a valuable tool for individuals and organizations alike. The application's user-friendly interface allows for seamless interaction, enabling users to make informed decisions based on the latest weather data. This project not only demonstrates the practical implementation of API integration but also highlights the significance of real-time data in enhancing daily life and operational efficiency across various sectors.

Equipment

- PC,
- JAVA Software
- OpenWeatherMap API key

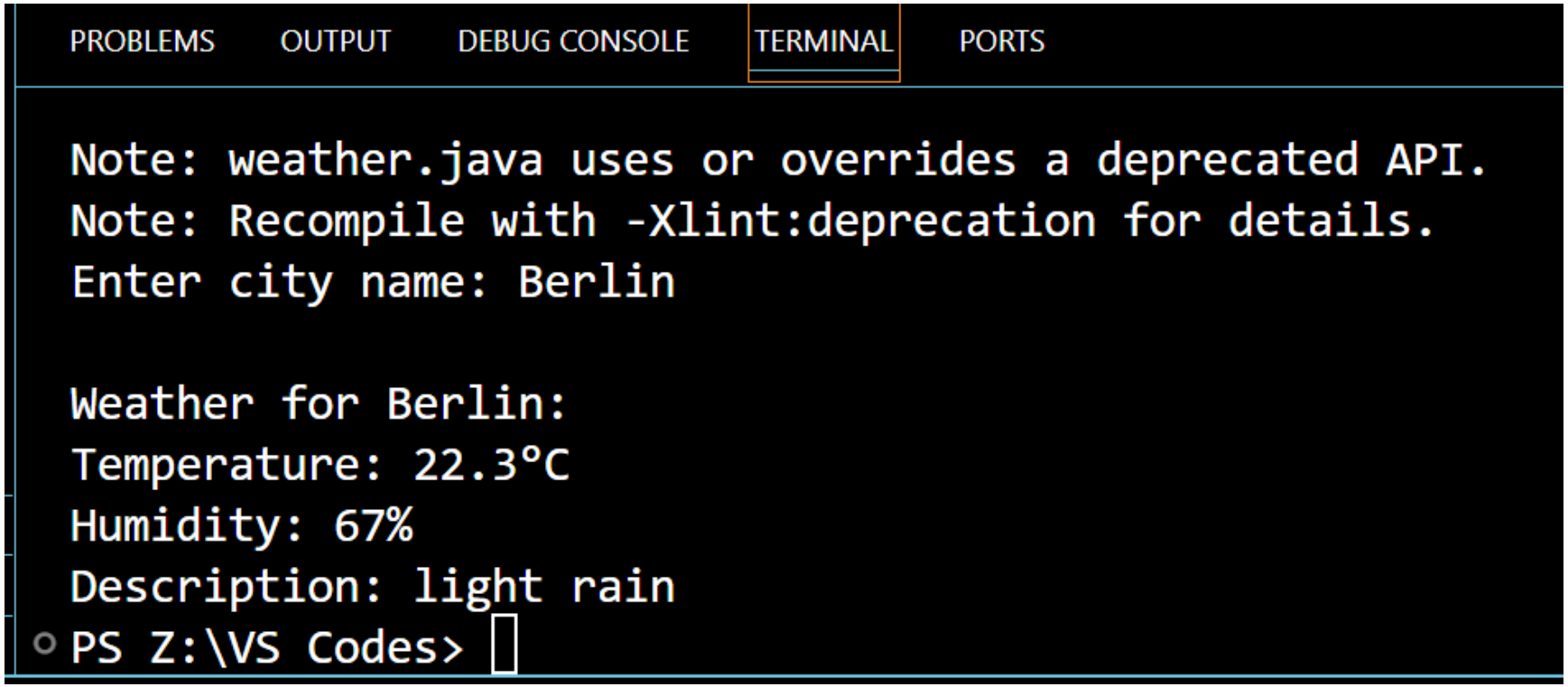
Program

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.Scanner;
public class weather {
    private static final String API_KEY = "afe69f0c4c055e3c55ee8415b7bf2494";
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter city name: ");
        String city = scanner.nextLine().trim();
        scanner.close();
        if (city.isEmpty()) {
            System.out.println("Error: City name cannot be empty");
            return;
        }
        try {
            String encodedCity = URLEncoder.encode(city, "UTF-8");
            String urlString = "https://api.openweathermap.org/data/2.5/weather?q="
                + encodedCity + "&appid=" + API_KEY + "&units=metric";
            URL url = new URL(urlString);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("GET");
            int responseCode = conn.getResponseCode();
            if (responseCode == 200) {
                String jsonResponse = readResponse(conn);
                parseAndDisplayWeather(jsonResponse);
            } else {
                handleErrorResponse(responseCode, conn);
            }
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
    private static String readResponse(HttpURLConnection conn) throws Exception {
        BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
        StringBuilder result = new StringBuilder();

        String line;
        while ((line = reader.readLine()) != null) {
            result.append(line);
        }
        reader.close();
        return result.toString();
    }
    private static void parseAndDisplayWeather(String json) {
        try {
```

```
        String temp = extractValue(json, "\"temp\":", ",", "}");
        String humidity = extractValue(json, "\"humidity\":", ",", "}");
        String description = extractDescription(json);
        String cityName = extractValue(json, "\"name\":\"", "\"", "\"");
        if (!cityName.equals("N/A")) {
            System.out.println("\nWeather for " + cityName + ":");
        } else {
            System.out.println("\nWeather Information:");
        }
        System.out.println("Temperature: " + temp + "°C");
        System.out.println("Humidity: " + humidity + "%");
        System.out.println("Description: " + description);
    } catch (Exception e) {
        System.out.println("Error parsing weather data");
    }
}
private static String extractValue(String json, String startKey, String... endChars) {
    int start = json.indexOf(startKey);
    if (start == -1) return "N/A";
    start += startKey.length();
    int end = json.length();
    for (String endChar : endChars) {
        int tempEnd = json.indexOf(endChar, start);
        if (tempEnd != -1 && tempEnd < end) {
            end = tempEnd;
        }
    }
    if (end > start) {
        return json.substring(start, end).replace("\"", "").trim();
    }
    return "N/A";
}
private static String extractDescription(String json) {
    int weatherStart = json.indexOf("\"weather\":[");
    if (weatherStart == -1) return "N/A";
    int descStart = json.indexOf("\"description\":\"", weatherStart);
    if (descStart == -1) return "N/A";
    descStart += "\"description\":\"".length();
    int descEnd = json.indexOf("\"", descStart);
    if (descEnd != -1) {
        return json.substring(descStart, descEnd);
    }
    return "N/A";
}
private static void handleErrorResponse(int responseCode, HttpURLConnection conn) {
    String errorMessage;
    switch (responseCode) {
        case 401:
            errorMessage = "Invalid API key";
            break;
        case 404:
            errorMessage = "City not found";
            break;
        case 429:
            errorMessage = "Too many requests. Please try again later";
            break;
        default:
            errorMessage = "HTTP Error: " + responseCode;
    }
    System.out.println("Error: " + errorMessage);
}
}
```

Simulation Results



Parameters Evaluated

Primary Weather Parameters:

- Temperature (current, minimum, maximum)
- Humidity levels
- Atmospheric pressure
- Wind speed and direction
- Precipitation levels
- Cloud cover percentage

Derived Metrics:

- Feels-like temperature
- Weather condition descriptions (e.g., clear, cloudy, rain)
- Sunrise and sunset times

Advantages

- **Real-Time Data Access**: Immediate retrieval of current weather conditions.
- **Accuracy**: Enhanced forecasting precision through reliable API sources.
- **User Convenience**: Accessible via web browsers without the need for installations.
- **Scalability**: Easily extendable to include additional features or support more locations.
- **Cost-Effective**: Utilizes free or affordable API services, reducing development costs.

Applications

- **Agriculture**: Assisting farmers in planning irrigation and harvesting schedules.
- **Travel and Transportation**: Informing travelers and logistics companies about weather-related disruptions.
- **Event Planning**: Helping organizers choose suitable dates and venues based on weather forecasts.
- **Disaster Management**: Providing early warnings for severe weather conditions.
- **Healthcare**: Advising vulnerable populations during extreme weather events.

Conclusion

The application enhances decision-making by providing users with timely weather updates, enabling informed choices in activities like travel planning, agriculture, and event management. Its modular architecture ensures scalability and flexibility, allowing easy integration of additional features such as severe weather alerts, air quality indices, or IoT device connectivity. As a web-based solution, it offers cross-platform accessibility, making it convenient for users across various devices. Additionally, this project holds educational value by demonstrating the practical use of APIs and web development skills to address real-world challenges.

Future Scope

Enhanced Features:

•Incorporation of machine learning algorithms for predictive analytics.

•Development of mobile application versions for broader accessibility.

•Integration with IoT devices for localized weather monitoring.

Expanded Data Sources:

•Utilizing multiple APIs to cross-verify and enrich weather data.

•Incorporating satellite imagery for visual weather representations.

References

•Suraj Singh et al., "Weather Forecasting Using API: A Comparative Study," IRJMETS, May 2023.

•Sathyabama Institute of Science and Technology, "Weather Forecasting System," 2022.

•GeeksforGeeks, "Forecast Weather Project," 2023.

•OpenWeatherMap API Documentation.

•Wikipedia, "Weather Forecasting."