

## nimesha.html

```
1  import java.io.IOException;
2  import java.util.Scanner;
3  import org.apache.http.HttpEntity;
4  import org.apache.http.client.methods.CloseableHttpResponse;
5  import org.apache.http.client.methods.HttpGet;
6  import org.apache.http.impl.client.CloseableHttpClient;
7  import org.apache.http.impl.client.HttpClients;
8  import org.apache.http.util.EntityUtils;
9  import org.json.JSONArray;
10 import org.json.JSONObject;
11
12 public class WeatherApp {
13     private static final String API_KEY = "b6907d289e10d714a6e88b30761fae22";
14     private static final String BASE_URL =
15         "https://samples.openweathermap.org/data/2.5/forecast/hourly?q=London,us";
16
17     public static void main(String[] args) {
18         Scanner scanner = new Scanner(System.in);
19         int choice;
20
21         do {
22             printMenu();
23             choice = scanner.nextInt();
24
25             switch (choice) {
26                 case 1:
27                     getWeather();
28                     break;
29                 case 2:
30                     getWindSpeed();
31                     break;
32                 case 3:
33                     getPressure();
34                     break;
35                 case 0:
36                     System.out.println("Exiting...");
37                     break;
38                 default:
39                     System.out.println("Invalid choice. Please try again.");
40             }
41         } while (choice != 0);
42
43         scanner.close();
44     }
45
46     private static void printMenu() {
47         System.out.println("\nMenu:");
48         System.out.println("1. Get weather");
49         System.out.println("2. Get Wind Speed");
50         System.out.println("3. Get Pressure");
51         System.out.println("0. Exit");
52         System.out.print("Enter your choice: ");
53     }
54
55     private static void getWeather() {
56         Scanner scanner = new Scanner(System.in);
```

```
56     System.out.print("Enter the date (yyyy-MM-dd HH:mm:ss): ");
57     String date = scanner.nextLine();
58
59     String apiUrl = BASE_URL + "&appid=" + API_KEY;
60     String jsonResult = getJsonResponse(apiUrl);
61
62     if (jsonResult != null) {
63         JSONObject jsonObject = new JSONObject(jsonResult);
64         JSONArray forecastArray = jsonObject.getJSONArray("list");
65
66         for (int i = 0; i < forecastArray.length(); i++) {
67             JSONObject forecast = forecastArray.getJSONObject(i);
68             String forecastDate = forecast.getString("dt_txt");
69
70             if (forecastDate.equals(date)) {
71                 JSONObject main = forecast.getJSONObject("main");
72                 double temperature = main.getDouble("temp");
73                 System.out.println("Temperature at " + date + ": " + temperature + "
°C");
74                 return;
75             }
76         }
77         System.out.println("Weather data not found for the specified date.");
78     } else {
79         System.out.println("Failed to fetch weather data.");
80     }
81 }
82
83 private static void getWindSpeed() {
84     Scanner scanner = new Scanner(System.in);
85     System.out.print("Enter the date (yyyy-MM-dd HH:mm:ss): ");
86     String date = scanner.nextLine();
87
88     String apiUrl = BASE_URL + "&appid=" + API_KEY;
89     String jsonResult = getJsonResponse(apiUrl);
90
91     if (jsonResult != null) {
92         JSONObject jsonObject = new JSONObject(jsonResult);
93         JSONArray forecastArray = jsonObject.getJSONArray("list");
94
95         for (int i = 0; i < forecastArray.length(); i++) {
96             JSONObject forecast = forecastArray.getJSONObject(i);
97             String forecastDate = forecast.getString("dt_txt");
98
99             if (forecastDate.equals(date)) {
100                 JSONObject wind = forecast.getJSONObject("wind");
101                 double windSpeed = wind.getDouble("speed");
102                 System.out.println("Wind Speed at " + date + ": " + windSpeed + "
m/s");
103                 return;
104             }
105         }
106         System.out.println("Wind speed data not found for the specified date.");
107     } else {
108         System.out.println("Failed to fetch wind speed data.");
109     }
110 }
111
112 private static void getPressure() {
113     Scanner scanner = new Scanner(System.in);
```

```
114     System.out.print("Enter the date (yyyy-MM-dd HH:mm:ss): ");
115     String date = scanner.nextLine();
116
117     String apiUrl = BASE_URL + "&appid=" + API_KEY;
118     String jsonResponse = getJsonResponse(apiUrl);
119
120     if (jsonResponse != null) {
121         JSONObject jsonObject = new JSONObject(jsonResponse);
122         JSONArray forecastArray = jsonObject.getJSONArray("list");
123
124         for (int i = 0; i < forecastArray.length(); i++) {
125             JSONObject forecast = forecastArray.getJSONObject(i);
126             String forecastDate = forecast.getString("dt_txt");
127
128             if (forecastDate.equals(date)) {
129                 JSONObject main = forecast.getJSONObject("main");
130                 double pressure = main.getDouble("pressure");
131                 System.out.println("Pressure at " + date + ": " + pressure + " hPa");
132                 return;
133             }
134         }
135         System.out.println("Pressure data not found for the specified date.");
136     } else {
137         System.out.println("Failed to fetch pressure data.");
138     }
139 }
140
141 private static String getJsonResponse(String apiUrl) {
142     try (CloseableHttpClient httpClient = HttpClients.createDefault()) {
143         HttpGet httpGet = new HttpGet(apiUrl);
144         try (CloseableHttpResponse response = httpClient.execute(httpGet)) {
145             HttpEntity entity = response.getEntity();
146             if (entity != null) {
147                 return EntityUtils.toString(entity);
148             }
149         }
150     } catch (IOException e) {
151         e.printStackTrace();
152     }
153     return null;
154 }
155 }
156 </body>
157 </html>
```