nimesha.html

```
import java.io.IOException;
import java.util.Scanner;
import org.apache.http.HttpEntity;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;
import org.json.JSONArray;
import org.json.JSONObject;

public class WeatherApp {
    private static final String API_KEY = "b6907d289e10d714a6e88b30761fae22";
    private static final String BASE_URL =
"https://samples.openweathermap.org/data/2.5/forecast/hourly?q=London,us";

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choice;

        do {
            printMenu();
            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    getWeather();
                    break;
                case 2:
                    getWindSpeed();
                    break;
                case 3:
                    getPressure();
                    break;
                case 0:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 0);

        scanner.close();
    }

    private static void printMenu() {
        System.out.println("\nMenu:");
        System.out.println("1. Get weather");
        System.out.println("2. Get Wind Speed");
        System.out.println("3. Get Pressure");
        System.out.println("0. Exit");
        System.out.print("Enter your choice: ");
    }

    private static void getWeather() {
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter the date (yyyy-MM-dd HH:mm:ss): ");
        String date = scanner.nextLine();

        String apiUrl = BASE_URL + "&appid=" + API_KEY;
        String jsonResult = getJsonResponse(apiUrl);

        if (jsonResult != null) {
            JSONObject jsonObject = new JSONObject(jsonResult);
            JSONArray forecastArray = jsonObject.getJSONArray("list");

            for (int i = 0; i < forecastArray.length(); i++) {
                JSONObject forecast = forecastArray.getJSONObject(i);
                String forecastDate = forecast.getString("dt_txt");

                if (forecastDate.equals(date)) {
                    JSONObject main = forecast.getJSONObject("main");
                    double temperature = main.getDouble("temp");
                    System.out.println("Temperature at " + date + ": " + temperature + "
°C");
                    return;
                }
            }
            System.out.println("Weather data not found for the specified date.");
        } else {
            System.out.println("Failed to fetch weather data.");
        }
    }

    private static void getWindSpeed() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the date (yyyy-MM-dd HH:mm:ss): ");
        String date = scanner.nextLine();

        String apiUrl = BASE_URL + "&appid=" + API_KEY;
        String jsonResult = getJsonResponse(apiUrl);

        if (jsonResult != null) {
            JSONObject jsonObject = new JSONObject(jsonResult);
            JSONArray forecastArray = jsonObject.getJSONArray("list");

            for (int i = 0; i < forecastArray.length(); i++) {
                JSONObject forecast = forecastArray.getJSONObject(i);
                String forecastDate = forecast.getString("dt_txt");

                if (forecastDate.equals(date)) {
                    JSONObject wind = forecast.getJSONObject("wind");
                    double windSpeed = wind.getDouble("speed");
                    System.out.println("Wind Speed at " + date + ": " + windSpeed + "
m/s");
                    return;
                }
            }
            System.out.println("Wind speed data not found for the specified date.");
        } else {
            System.out.println("Failed to fetch wind speed data.");
        }
    }

    private static void getPressure() {
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter the date (yyyy-MM-dd HH:mm:ss): ");
        String date = scanner.nextLine();

        String apiUrl = BASE_URL + "&appid=" + API_KEY;
        String jsonResult = getJsonResponse(apiUrl);

        if (jsonResult != null) {
            JSONObject jsonObject = new JSONObject(jsonResult);
            JSONArray forecastArray = jsonObject.getJSONArray("list");

            for (int i = 0; i < forecastArray.length(); i++) {
                JSONObject forecast = forecastArray.getJSONObject(i);
                String forecastDate = forecast.getString("dt_txt");

                if (forecastDate.equals(date)) {
                    JSONObject main = forecast.getJSONObject("main");
                    double pressure = main.getDouble("pressure");
                    System.out.println("Pressure at " + date + ": " + pressure + " hPa");
                    return;
                }
            }
            System.out.println("Pressure data not found for the specified date.");
        } else {
            System.out.println("Failed to fetch pressure data.");
        }
    }

    private static String getJsonResponse(String apiUrl) {
        try (CloseableHttpClient httpClient = HttpClients.createDefault()) {
            HttpGet httpGet = new HttpGet(apiUrl);
            try (CloseableHttpResponse response = httpClient.execute(httpGet)) {
                HttpEntity entity = response.getEntity();
                if (entity != null) {
                    return EntityUtils.toString(entity);
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }
}
</body>
</html>
```