Assignment: Weather Data Analyzer

Background: Imagine you are developing an application for a weather service. The service provides weather data in JSON format. Your task is to create a Java application that allows users to input JSONPath queries to retrieve specific information from the weather data.

Objective: Develop a Java application that reads a JSON file containing weather data and allows the user to input JSONPath queries to extract specific information.

Requirements

1. **JSON File (weather.json)**:
    - Create a JSON file representing weather data. The file should include information like temperature, humidity, wind speed, and weather conditions for different cities or times.
    - Example JSON structure:

```
{
  "weatherData": [
    {
      "city": "New York",
      "temperature": 22,
      "humidity": 80,
      "conditions": "Sunny"
    },
    {
      "city": "London",
      "temperature": 16,
      "humidity": 65,
      "conditions": "Cloudy"
    }
    // More data...
  ]
}
```

1. **Java Application**:
    - Write a Java program that reads the `weather.json` file.
    - The program should prompt the user to enter a JSONPath query.
    - Process the JSONPath query and display the matching elements from the JSON file.
    - Handle exceptions such as invalid JSONPath syntax or file read errors.
2. **Sample JSONPath Queries**:
    - All cities with temperature above 20°C: `$.weatherData[?(@.temperature > 20)].city`
    - Humidity in London: `$.weatherData[?(@.city == 'London')].humidity`
    - All weather conditions listed: `$.weatherData[*].conditions`

Steps to Follow

1. **Set Up Environment**:
   - Ensure Java SDK is installed, and an IDE is set up.
   - Include a JSON processing library like Jackson or json-simple in your project.
2. **JSON File Creation**:
   - Create the `weather.json` file with a proper structure and sample data.
3. **Java Program Development**:
   - Start by setting up the JSON parsing using your chosen library.
   - Implement JSONPath processing (libraries like Jayway's JsonPath can be used).
   - Develop the user interface for inputting JSONPath queries and displaying results.
4. **Testing**:
   - Test the application with different JSONPath queries to ensure it accurately queries and displays data from the JSON file.
5. **Advanced Challenges (Optional)**:
   - Extend the application to fetch real-time weather data from an API.
   - Implement additional features like data filtering or sorting based on user input.

Submission

I. Please respect the due date to avoid penalties for late submission.

II. The assignment requires a presentation. You can choose **one** of the following three formats:

a. A written report (PDF) in which you explain your algorithm as well as your code. Please also include screenshots of the application's output when you run it. Ensure the cover page clearly lists the group members (names and student IDs) who participated in the project.

b. A recorded screen video. In this case, you and your teammates should set up a live meeting, share your screen, explain the code, and run the application while recording the session. Then, submit both the source code and the video file. At the beginning of the presentation, don’t forget to mention the names and IDs of all participants.

c. A live Zoom or Teams meeting where you present the project to me. In this case, you only need to submit the source code of your application.

III. Please decide whether you are working in a team or solo before submitting.