Group Assignment: Introduction to Scrum & JSON Data Manipulation

Objective:
This assignment is designed to introduce you to both **Scrum** (an agile project management framework) and basic data manipulation with a JSON file. You'll work in groups of three to complete a small data management project while using Scrum to manage your teamwork and workflow.

Introduction

For this assignment, you'll create a small program that interacts with a JSON file named **`users.json`**. The program will present a menu with three options for managing user data. Each option will perform a specific action on the `users.json` file, allowing you to practice working with JSON data and basic file I/O.

To get started, you'll need to gather user input from the console. Below is a code snippet that shows how to read user input in Node.js using the `readline` module:

```
const readline = require("readline").createInterface({
    input: process.stdin,
    output: process.stdout,
});

readline.question("Who are you?", (name) => {
    console.log(`Hey there ${name}!`);
    readline.close();
});
```

In this example:

- readline.question() prompts the user with a question ('Who are you?'), and stores the input in a variable called name.
- console.log() then outputs a message using the name variable.
- Finally, readline.close() closes the input interface.

For this project, you'll use readline to gather information for each menu choice.

Project Overview

You will create a program that allows users to:

1. Add a new user with specific details.
2. Display a list of all users.
3. Delete a user by ID.
4. Exit the program

This project will help you learn about working with JSON data, file I/O in Node.js, and give you experience with Scrum practices.

Part 1: Scrum Setup and Planning

1. **Roles:**

   - Assign each team member one of these Scrum roles:
     - **Scrum Master:** Ensures Scrum practices are followed, facilitates meetings, and supports team collaboration.
     - **Product Owner:** Responsible for defining what features are needed and prioritizing tasks.
     - **Developer:** Focuses on implementing the features as defined by the Product Owner. (In this small project, all members will help with development, but this role ensures that one person keeps the team on track technically).

2. **Backlog Creation:**

   - As a team, break down the project into small, manageable tasks (User Stories). Examples of ser stories:
     - As a user, I want to add a new user entry to the JSON file so that I can store user information.
     - As a user, I want to display a list of all users so that I can view all stored data.
     - As a user, I want to delete a user by ID so that I can remove outdated or incorrect entries.

3. **Sprint Planning:**

   - Plan a short, two-day sprint. Decide as a team what can be realistically achieved in this sprint.
   - Prioritize your tasks and select a few user stories that you aim to complete within this sprint.

4. **Sprint Board:**

   - Set up a simple Scrum board (using tools like Trello, Jira, or even a whiteboard) with columns: To-Do, In Progress, and Done.
   - Move tasks through the board as you work through them.

Part 2: Building the Program

1. **Setup:**

   - Create a JSON file named users.json to store user data. Initialize it as an empty array

2. **Program Structure:**

   - Write a program that presents a menu with four options:
     1. Add User:
        - Prompt the user for the following details:
          - First Name
          - Last Name Date of Birth(e.g., "YYYY-MM-DD")
          - Whether they have a driver's license (yes/no)
        - Add the new user to users.json, assingning each user a unique ID (you can use a simple counter).

    2. Display All Users:
        - Read the contents of users.json and print each user's details to the console in a readable format.
    3. Delete a User by ID:
        - Prompt the user to enter an ID, then delete the corresponding user from users.json.

3. **File I/O:**

    - Use Node.js’s file system module (fs) to read from and write to the JSON file.
    - Ensure to handle potential errors (e.g., file not found, invalid JSON format).

4. **Testing:**

    - Test each menu option to ensure that the program behaves as expected. Verify that users can be added, displayed, and deleted correctly.

Part 3: Scrum Events and Reflection

1. **Daily Stand-Up:**
    - Each day, have a 5-10 minute meeting where each team member answers:
        - What did you accomplish yesterday?
        - What will you work on today?
        - Are there any blockers preventing progress?
2. **Sprint Review and Retrospective:**
    - At the end of the sprint, hold a Sprint Review to discuss what was completed.
    - Conduct a **Retrospective** to reflect on:
        - What went will?
        - What could be improved in your next sprint?
        - How did the Scrum help in orginizing your teamwork?
3. **Documentation:**
    - Document how to use the program and each menu option.
    - Write a brief summary of your experience with Scrum. What did you learn? How did it help with teamwork?