.

📌 Project Title:

Agile Team Sprint Tracker with GitHub Integration (Multi-Sprint Support)

🎯 Objective:

Build a Java-based system that allows Agile teams to manage multiple sprints, track user stories, assign tasks to team members, and automatically fetch updates/issues from GitHub repositories to maintain sprint visibility.

🧩 Key Features:

- Create and manage **multiple Agile sprints**.
- Add, edit, and remove **user stories** and **tasks** per sprint.
- Assign tasks to team members.
- Track progress and status of tasks (To Do, In Progress, Done).
- Integrate with **GitHub Issues API** to pull live GitHub issues into a sprint.
- Visual display (CLI or basic GUI) of sprint overview and progress.

🏗 Project Structure — File & Class Guide:

Here's a **step-by-step breakdown** of what files/classes to create and their responsibility **(no code, just descriptions):

---

1. Main.java

Purpose:
Acts as the **entry point** of the application. Initializes the UI/CLI and connects major components together.

---

2. Sprint.java

Purpose:
Represents a **single sprint**. Holds:

- Sprint name
- Start and end dates
- List of user stories

---

3. UserStory.java

Purpose:
Represents an **Agile user story**. Includes:

- Title
- Description
- List of tasks
- Priority level
- Story points

---

4. Task.java

Purpose:
Represents a **task** within a user story. Includes:

- Task title
- Assigned team member
- Status (To Do, In Progress, Done)

---

5. TeamMember.java

Purpose:
Stores **team member** details:

- Name
- Email or GitHub username
- Assigned tasks

---

6. SprintManager.java

Purpose:
Controls the **creation, deletion, and management of multiple sprints**.
Handles switching between sprints and updating their data.

---

7. GitHubIntegration.java

Purpose:
Handles connection to the **GitHub Issues API**.
Fetches issues from a GitHub repository and converts them into tasks or user stories.

8. DataStorage.java

Purpose:
Responsible for saving and loading project data (sprints, stories, tasks) to a local file or JSON format for persistence.

9. UIManager.java

Purpose:
Manages user interface (console-based or GUI).
Presents menus, handles inputs, and displays sprint dashboards and updates.

🔧 Technologies to Use:

- **Java (Core, OOP principles)**
- **REST API (for GitHub integration)**
- **JSON** (for data storage)
- Optional: **Swing or JavaFX** (if GUI is needed)

🔁 Flow of Application:

1. User runs the app via Main.java.
2. SprintManager loads existing sprints (if any) using DataStorage.
3. User can create a new sprint or manage existing ones.
4. Within a sprint, the user can:
    - Add user stories
    - Create and assign tasks
    - Fetch GitHub issues (via GitHubIntegration)
5. Tasks get assigned and tracked by status.
6. Progress and reports can be shown via UIManager.

✅ Expected Outcome:

A fully functional **Agile Sprint Tracker** that helps a team organize sprints and tasks, and integrate issues from GitHub for real-time synchronization.