Here's an **exercise** designed for freshers to practice Git and GitHub collaboration in a team environment. This activity simulates a real-world project scenario and helps them understand teamwork using GitHub.

Git and GitHub Team Collaboration Exercise

Scenario:

Your team is working on creating a collaborative **"Personal Portfolio Website"** project. Each team member will contribute to different sections of the website. You will practice version control, branching, merging, and resolving conflicts while working on this project.

Steps for the Exercise:

1. Create the GitHub Repository (Team Lead Task):

- One team member (Team Lead) creates a new repository on GitHub, e.g., `team-portfolio`.
- Initialize the repository with a `README.md` file that describes the project.
- Add other team members as **collaborators** in the repository.

---

2. Clone the Repository (All Members):

- Each team member clones the repository locally using:

```
git clone <repository_url>
cd team-portfolio
```

---

3. Project Setup:

The repository should contain:

1. A `README.md` file for project description.

2. A `project` folder with the following structure:

```
project/
  ├── index.html
  ├── about.html
  ├── contact.html
  └── styles.css
```

3. The Team Lead creates this structure and pushes it to the main branch:

```
git add .
git commit -m "Setup project structure"
git push origin main
```

---

4. Assign Tasks to Team Members:

Each team member is responsible for one file:

- **Member 1:** Edit `index.html` (Home page).
- **Member 2:** Edit `about.html` (About Us section).
- **Member 3:** Edit `contact.html` (Contact Us form).
- **Member 4:** Edit `styles.css` (Styling for the website).

Each member will work on their assigned tasks in separate **feature branches**.

---

5. Branching and Development:

Each team member follows these steps:

1. Create a new branch for their task:

```
git checkout -b feature/<task-name>
# Example: git checkout -b feature/index-page
```

2. Make changes to the assigned file.
3. Stage and commit changes:

```
git add <file_name>
git commit -m "Add content for <task_name>"
```

4. Push the branch to GitHub:

```
git push origin feature/<task-name>
```

---

6. Review and Merge:

- Team members review each other's PRs.
- Approve and merge the PRs one by one to avoid conflicts.
- The Team Lead ensures all changes are integrated into the `main` branch.

---

7. Handle Merge Conflicts (If Any):

- If multiple members modify the same file (e.g., `styles.css`), Git may flag a **merge conflict**.
- Resolve the conflict locally:

```
git pull origin main
git merge feature/<task-name>
```

- Fix the conflicts manually in the code, then:

```
git add .
git commit -m "Resolve merge conflict"
git push origin feature/<task-name>
```

Deliverables:

1. A fully functional **Personal Portfolio Website** pushed to the `main/master` branch.
2. A `README.md` with:
    - Team members' contributions.
    - A brief description of the project.

Key Learning Outcomes:

1. Using GitHub to collaborate as a team.
2. Creating and working with branches.
3. Understanding and resolving merge conflicts.
4. Pushing and pulling changes from GitHub.