

📌 GitHub Team Collaboration Guide (For Beginners)

Use these steps to safely **collaborate, edit, and push your code** on GitHub without breaking each other's work.

📌 1. Clone the Project (Only Do Once)

```
git clone https://github.com/<your-team-or-username>/<your-repo-name>.git
cd <your-repo-name>
```

Replace with the correct GitHub repo URL. You now have a local copy of the project.

📌 2. Pull Latest Changes (Do This Every Day Before Starting Work)

```
git checkout main
git pull origin main
```

This updates your local copy with your teammates' changes from the `main` branch. Use this to keep your `main` up to date before starting a new branch.

📌 3. Create Your Own Branch (Do This Before You Start Editing)

```
git checkout -b <your-name-or-feature-branch>
```

Examples:

```
git checkout -b min-eda-fixes
git checkout -b habib-model-train
```

You are now working safely on your own branch.

📌 4. Do Your Work and Save Your Changes

- Edit code, notebooks, etc.
- Save your work regularly.

Check your changes:

```
git status
```

📌 5. Add and Commit Your Changes

```
git add .
git commit -m "Add EDA summary for review dataset"
```

- `git add .` adds all your changed files.
 - Write a clear commit message in `git commit -m "..."`.
-

📌 6. Push Your Branch to GitHub

```
git push origin <your-branch-name>
```

Example:

```
git push origin min-eda-fixes
```

7. Create a Pull Request (PR)

Go to GitHub in browser → You'll see **"Compare & Pull Request"** → Click it → Add a description → Submit.

This is how others can review and merge your work into `main`.

8. Keeping Your Branch Up to Date (Optional But Important)

Do this when you have been working on your branch for too long.

```
git pull origin main --rebase
```

Use this to bring the latest changes from `main` into your own branch. This helps prevent conflicts when you create a pull request.

9. Switch Between Branches

```
git checkout main
```

10. (Optional) After Your PR is Merged, Clean Up

```
git branch -d <your-old-branch>
```

Troubleshooting Tips

- **Accidentally edited on `main` ?**
 - Copy your changes
 - `git stash`
 - Create a branch → paste the changes → commit
- **Merge conflicts?**
 - Ask help! But usually Git will show where conflicts happen and you just need to fix the lines manually.