GreenHomeAI — Git Workflow + Troubleshooting (Balpreet & Akshay)

Golden Routine for Any New Feature/Task

1) Start fresh:

```
git switch main
git pull
```

2) Create a feature branch:

```
git switch -c feat/<short-slug>
# e.g., feat/eda, feat/data-loader
```

3) Work and save progress:

```
git add -A
git commit -m "feat: short description"
```

4) Push branch to remote:

```
git push -u origin feat/<short-slug>
```

5) Keep your branch updated with main:

```
git fetch origin
git rebase origin/main
# fix conflicts → git add <file>
# continue → git rebase --continue
# safe force push after rebase
git push --force-with-lease
```

6) Open PR on GitHub, request review, Squash & Merge.

7) After merge cleanup:

```
git switch main
git pull
git branch -d feat/<short-slug>   # delete local branch
# delete remote branch in GitHub UI
```

Common Mistakes & How to Fix Them

A) I committed directly to main by accident (not pushed yet)

Move the commit onto a new feature branch:

```
git switch -c feat/move-accidental-commit
```

B) I pushed to main by accident

Create a revert commit (public, safe):

```
git revert <bad-commit-sha>
git push
```

If multiple commits, revert a range (oldest..newest):

```
git revert <oldest>^..<newest>
# then push
```

C) I staged/committed the wrong files

Unstage, keep changes in working tree:

```
git restore --staged <file>
```

OR discard local changes to a file (careful):

```
git checkout -- <file>
```

Amend the last commit before pushing:

```
git add <file>
```

```
git commit --amend
```

D) I committed a huge file / GitHub rejected 100MB+

Remove the file from history and keep it locally ignored:

```
git rm --cached path/to/huge.file
# add to .gitignore if not already
git commit -m "chore: remove large file from repo"
git push
```

If it already exists in remote history, use filter-repo (advanced):

```
pipx install git-filter-repo  # or brew install git-filter-repo
git filter-repo --path path/to/huge.file --invert-paths
git push --force-with-lease
```

E) Merge conflicts during rebase

Open each conflicted file, resolve the <<<<<<< >>>>>>> markers:

```
git add <fixed-file>
git rebase --continue
```

Abort rebase if necessary:

```
git rebase --abort
```

F) I pulled and got a messy merge commit

Prefer rebasing pulls (config once):

```
git config pull.rebase true
git config branch.main.rebase true
```

Fix current branch now:

```
git fetch origin
git rebase origin/main
```

G) I think I lost work after rebase/force-push

Use the reflog to find and restore the commit:

```
git reflog
# copy the commit id where work existed
git switch -c rescue/<name> <commit-id>
```

H) I need to stash temporary changes

Stash → switch → come back:

```
git stash push -m "wip: <desc>"
git switch main
# later
git switch <your-branch>
git stash list
git stash pop
```

I) Change the last commit message (not pushed)

```
git commit --amend -m "new message"
```

J) Change an older commit message (advanced)

Interactive rebase, edit message, then force-with-lease:

```
git rebase -i HEAD~3
# change 'pick' → 'reword' for the commit
```

```
# save, edit message
git push --force-with-lease
```

K) Accidentally added secrets (.env)

1) Rotate/revoke the secret immediately.
2) Purge from repo and add to .gitignore:

```
git rm --cached .env
git commit -m "chore(security): remove .env and ignore"
git push
# For history removal use filter-repo as in D)
```

L) Need to delete a remote branch

```
git push origin --delete feat/<slug>
```

M) Partner and I both edited same file

Communicate; then rebase and resolve conflicts carefully:

```
git fetch origin
git rebase origin/main
# fix conflicts → git add → git rebase --continue
```

Data & LFS Policy (to avoid future pain)

- Raw data → data/raw/ (ignored). Do NOT commit.
- Processed tiny samples only → data/processed/ (tracked).
- Medium binaries match .gitattributes → LFS will handle.
- Never commit files ≥100MB (GitHub blocks them).

Useful Inspect Commands

```
git status
git log --oneline --graph --decorate --all
git diff
git blame <file>
```

Contact Points

If stuck: write the exact command + error. One of us reproduces, then we fix together.