

Team Git Rules

Project Name: DonateLife

Repository Type: Public (GitHub Pages deployed)

Branch Strategy: Protected `main` branch with feature branches

1. Purpose of This Document

This document defines **how we use Git and GitHub as a team** so that: - The deployed website is never broken - Everyone can work safely in parallel - Code history stays clean and understandable

These rules are **mandatory for all team members**.

2. Branch Rules (MOST IMPORTANT)

`main` Branch

- `main` is the **production branch**
- GitHub Pages is deployed from `main`
- ~~No one pushes directly to `main` except the **project owner**~~
- `main` must always be **stable and working**

Feature Branches

- Every team member works on **their own branch**
- Branch name format:

```
feature-yourname-task
```

Examples:

- `feature-ishfa-navbar`
 - `feature-partner-footer`
-

3. Daily Workflow (Step-by-Step)

Step 1: Clone the Repository (one time)

```
git clone https://github.com/USERNAME/REPO-NAME.git  
cd REPO-NAME
```

Step 2: Create Your Own Branch

```
git checkout -b feature-yourname-task
```

Step 3: Work on Code

- Modify files
- Test locally
- Do **not** touch deployment settings

Step 4: Commit Changes

```
git add .  
git commit -m "Clear description of what you did"
```

Step 5: Push Your Branch

```
git push origin feature-yourname-task
```

Step 6: Create Pull Request (PR)

- Open GitHub
- Click **Compare & Pull Request**
- Describe what you changed
- Assign the project owner for review

4. Pull Request Rules

Every Pull Request must:

- Have a **clear title**
- Explain **what was changed**
- Mention if UI/logic was affected

The project owner will:

- Review the code
- Request fixes if needed
- Merge into `main`

5. Commit Message Rules

Use simple, meaningful messages:

✓ Good examples:

- `Add footer section`
- `Fix navbar alignment`
- `Update blood chart layout`

✗ Bad examples:

- `update`
- `fix`
- `changes`

6. Syncing With Main Branch

Before starting new work:

```
git checkout main  
git pull origin main  
git checkout feature-yourname-task  
git merge main
```

This prevents conflicts.

7. Conflict Handling Rule

If a merge conflict happens: - Do **not panic** - Inform the project owner - Resolve together if needed

8. Emergency Rule

If the deployed site breaks: - Stop all pushes - Inform the project owner immediately - Fix happens on a **separate branch**, not directly on `main`

9. Team Agreement

By contributing to this repository, each member agrees to follow these Git rules strictly.

Project Owner: Ishfa Mubashira

Date: ____