

Team Git Workflow Guide

This guide explains the full workflow your team should follow when pulling your repository, making changes, and contributing safely using branches and pull requests.

1. Clone the Repository (First Time Only)

Use HTTPS or SSH:

```
git clone https://github.com/your-username/your-repo.git  
# OR  
git clone git@github.com:your-username/your-repo.git
```

Move into the project:

```
cd your-repo
```

2. Set Up the Development Environment

Create & activate a virtual environment

macOS / Linux:

```
python -m venv .venv  
source .venv/bin/activate  
pip install -r requirements.txt
```

Windows (PowerShell):

```
python -m venv .venv  
.venv\Scripts\Activate.ps1  
pip install -r requirements.txt
```

Create your `.env` file

```
cp .env.example .env
```

Edit `.env` and fill in environment variables.

Run initial migrations

```
python manage.py migrate  
python manage.py runserver
```

3. Everyday Workflow for Making Changes

1. Start with the latest `main`

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

2. Create a feature branch

```
git checkout -b feature/your-description
```

3. Make changes locally

Run the server, tests, and modify files.

4. Stage and commit changes

```
git add .  
git commit -m "feat: describe your change here"
```

5. Push your branch

```
git push -u origin feature/your-description
```

6. Open a Pull Request on GitHub

- Click **Compare & pull request**
- Add a clear title and description
- Mention how to test the change

- Mark as **Draft** if work is incomplete
-

4. Keeping Your Branch Updated

If `main` has new commits, merge them into your branch:

```
git checkout main
git pull origin main

git checkout feature/your-description
git merge main
```

Resolve any conflicts, then:

```
git add .
git commit
git push
```

5. Review Workflow for Teammates

Pull latest changes

```
git fetch origin
git checkout feature/your-description
git pull origin feature/your-description
```

Run code locally

- Run migrations
- Run the server
- Check functionality

Add review comments on the PR

6. After Merge

Once your PR is merged:

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

Delete your merged feature branch (optional):

```
git branch -d feature/your-description
```

Remote delete (optional):

```
git push origin --delete feature/your-description
```

7. What to Commit vs Not Commit

Commit these:

- Python files
- Templates
- Static source files
- `requirements.txt`
- Migrations
- `.env.example`

Do NOT commit:

- `.env` (real secrets)
- `db.sqlite3`
- `media/`
- `staticfiles/`
- `.vscode/`, `.idea/`

Already handled by `.gitignore`.

8. Helpful Git Commands Cheat Sheet

```
# update main  
git checkout main  
git pull origin main  
  
# create branch
```

```
git checkout -b feature/name

# stage & commit
git add .
git commit -m "message"

# push branch
git push -u origin feature/name

# merge main into branch
git checkout feature/name
git merge main

# delete branch after merge
git branch -d feature/name
git push origin --delete feature/name
```

9. Pull Request Review Checklist

- Does the code run locally?
 - Are migrations included?
 - No secrets added accidentally?
 - Code follows project style & structure?
 - Tests updated if needed?
 - Small, focused changes?
-

This document can now be shared directly with your teammates to guide your entire Git workflow.