

---

## Farm-to-Market Git Workflow Sheet

### 1 Branch Overview

#### Branch Purpose

main Production-ready, never push directly here

dev Central integration branch; all features merge here first

feature/\* Individual member branches (one per module)

#### Feature branches for our team:

- feature/auth-profiles → Thanushi
- feature/farmer-module → Kavya
- feature/buyer-module → Methsani
- feature/marketplace-transactions → Dilutha
- feature/admin-analytics → Hasinika

---

### 2 Daily Workflow (when starting work)

# 1. Switch to dev and pull latest changes

git checkout dev

git pull origin dev

# 2. Switch to your feature branch

git checkout feature/your-branch

# 3. Merge latest dev into your branch

git merge dev

✅ This ensures you're up-to-date before coding.

---

### 3 Save & Commit Work

`git add .`

`git commit -m "feat(module): short descriptive message"`

`git push origin feature/your-branch`

#### Commit message examples:

- `feat(auth): add login page for buyer`
- `fix(farmer): correct crop image validation`
- `docs(readme): add setup instructions`

---

### 4 Pull Request (PR) Workflow

1. Open PR on GitHub: **feature branch** → **dev**
2. Title: short + descriptive
3. Description: what you did, why, screenshots, migration steps
4. Assign reviewer (optional)
5. Merge only after approval

**Tip:** Use Squash & merge to keep history clean.

---

### 5 Keeping Your Branch Updated

Before starting next work:

`git checkout dev`

`git pull origin dev`

`git checkout feature/your-branch`

`git merge dev`

---

### 6 Dev → Main (Release Workflow)

1. Only when dev is stable + tested
2. Open PR: dev → main
3. Merge with review + optional tag:

git checkout main

git pull origin main

git tag -a v1.0.0 -m "Release v1.0.0"

git push origin v1.0.0

---

## Do's & Don'ts

### Do:

- Pull dev before starting work
- Keep commits small + descriptive
- Use PRs for all merges into dev
- Keep .env out of Git (.gitignore)

### Don't:

- Push directly to main
- Make huge PRs with multiple unrelated changes
- Commit secrets, API keys, or passwords

---

This cheat sheet is all your team needs to **stay organized** before you start coding.

---

If you want, I can also **make a super compact version** (like 1 paragraph) that you can **paste in Slack/WhatsApp or pinned in VS Code**, so your team can follow it quickly.

Do you want me to do that?