

Professional GitHub Workflow for a MERN Stack Project

1. Initial Project Setup

Organize your GitHub repository like this:

my-app/

■■■ backend/ (Node.js/Express backend)

■■■ frontend/ (React frontend)

■■■ docs/ (Design, requirements, architecture)

■■■ .github/ (GitHub workflows)

■■■ .gitignore

■■■ README.md

■■■ LICENSE

2. Documentation & Planning

README.md:

- App overview
- Tech stack
- How to install/run
- Links to docs

docs/ folder:

- requirements.md: Functional + non-functional requirements
- features.md: Feature list (MVP, v1.0, etc.)
- architecture.md: Diagrams & tech decisions
- api-spec.md: REST API design (Swagger if desired)

3. Project Management

GitHub Issues:

- Epics: Big features like 'User Authentication'
- Issues: Smaller tasks

GitHub Projects:

- Backlog, In Progress, Done

Milestones:

- MVP, Alpha, Production release

4. Version Control Best Practices

Branching Strategy:

- main: production-ready
- dev: development
- feature/xyz: features
- fix/xyz: bug fixes

Commit Messages:

Use conventional commits:

feat: add login form

fix: correct input validation

docs: update API docs

5. Pull Requests & Code Reviews

- Open PRs for each task
- Use PR templates
- Link issues with 'Fixes #12'
- Self-review before assigning

6. Testing & CI/CD

Testing:

- Jest for backend/frontend
- React Testing Library for UI

CI with GitHub Actions:

- Run tests, lint code, build app

7. Deployment

- Use Heroku, Vercel, or Netlify
- Automate deploy on push to main

8. Security & Secrets

- Use GitHub Secrets
- Add .env.example to repo, never commit .env

Bonus: Communication & Planning

- Write issues, update docs, do retros
- Helps onboard collaborators later