GitHub Team Development Using Branching Strategy

1. Initial Project Setup

A. Create the Repository:

- One person creates the repo on GitHub.
- Choose Private or Public.
- Add a README, .gitignore, and license (if needed).

B. Clone the Repository:

git clone https://github.com/your-org/your-repo.git

cd your-repo

2. Decide on a Branching Strategy

Use a strategy like Git Flow, GitHub Flow, or Trunk-Based Development.

Recommended: GitHub Flow.

Branch Types:

- main: production-ready code
- dev: integration branch (optional)
- feature/xyz: per-feature branches
- bugfix/xyz: bug fixes
- hotfix/xyz: for production hotfixes

3. Create and Use Branches

Create a new feature branch:

git checkout -b feature/login-api

Make changes and commit:

git add .

git commit -m "Add login endpoint"

GitHub Team Development Using Branching Strategy

4. Push Branch to GitHub

git push -u origin feature/login-api

5. Open Pull Request (PR)

- Go to GitHub and click 'Compare & pull request'
- Base branch: main or dev
- Add title, description, and reviewers

6. Code Review & Collaboration

- Reviewers provide feedback or approve
- Push more commits if needed

7. Merge the PR

- Click 'Merge Pull Request'
- Delete the feature branch

Tip: Enable branch protection rules on main.

8. Pull Latest Changes

git checkout main

git pull origin main

9. Handle Conflicts

git fetch origin

git checkout feature/login-api

git rebase origin/main # or merge

10. CI/CD Integration

- Use GitHub Actions or other CI tools
- Create .github/workflows to run tests or deploy

Summary Flow Diagram

main ←— dev ←— feature/login-api

↑

PR merged after review

Extra Tips

- Squash commits when merging PRs
- Use GitHub Projects or Issues
- Add CODEOWNERS to auto-assign reviewers