**Workflow Steps**

**1. Create a New Feature Branch**

Start by creating a new branch off the main branch:

git checkout main

git pull origin main # Ensure the main branch is up-to-date

git checkout -b feature/your-feature-name

Use a clear and descriptive name for the branch, such as:

feature/login-page

feature/user-authentication

hotfix/ui-redesign

bugfix/ui-redesign

**2. Develop the Feature**

Make changes and commit them to the feature branch.

Commit messages should be descriptive and concise:

git add .

git commit -m "Add login page with form validation"

Regularly commit small, logical chunks of work.

**3. Sync with the Main Branch**

Periodically pull the latest changes from the main branch to keep the feature branch updated:

git checkout main

git pull origin main

git checkout feature/your-feature-name

git merge main

**4. Push the Feature Branch**

Push the branch to the remote repository for sharing and collaboration:

git push origin feature/your-feature-name

**5. Open a Pull Request (PR)**

Create a pull request in your Git platform (e.g., GitHub, GitLab, Bitbucket).

Add relevant reviewers and provide a clear description of the feature.

**6. Code Review and Feedback**

Reviewers inspect the code for quality, adherence to guidelines, and functionality.

Address feedback by making changes in the feature branch and updating the PR:

git add .

git commit -m "Fix feedback on form validation"

git push origin feature/your-feature-name

**7. Merge into Main Branch**

Once approved and tested, merge the feature branch into the main branch

Preferably use the "Squash and Merge" or "Rebase and Merge" strategy to maintain a clean history.

git checkout main

git merge feature/your-feature-name

git push origin main

**Best Practices**

**Branch Naming Conventions**:

Use prefixes like feature/, bugfix/, or hotfix/ for clarity.

Keep branch names descriptive but concise.

**Keep Commits Atomic**:

Each commit should represent a single, logical change.

**Test Locally**:

Ensure your changes are functional and pass tests before pushing.

**Review Changes Before Commit**:

Use git diff to inspect changes before committing.

**Write Clear PR Descriptions**:

Include details about what the feature does, how it was implemented, and any relevant screenshots or references.

IF Require help Devops team ready to assist you