How to Review Code Locally & Clean Up Afterwards

🔹 Overview

After reviewing a teammate's feature branch locally, it's important to clean up your local repository so you don't clutter it with old branches. This guide will walk you through the steps to review code, delete the local branch, and ensure your `**master**` branch is up to date.

🔹 Steps to Review & Clean Up Locally

1️⃣Pull the Feature Branch to Review

When assigned a merge request to review, pull the branch to your local machine:

`**git fetch origin feature-branch-name**` (whatever the relevant branch name is)
`**git checkout feature-branch-name**` (whatever the relevant branch name is)

2️⃣Open & Test the Code in the Browser

- Open the `**index.html**` (or relevant file) in a web browser.
- Check that everything behaves as expected.
- If reviewing CSS, make sure styles apply correctly.

3️⃣Complete the Review & Approve the Merge Request

- If the code looks good, let the developer know with a comment and approve the merge request in GitLab.
- If there are issues, leave comments and request changes. Then communicate with the developer to let her know you've completed the review and there are some comments for her to view.

4️⃣Switch Back to Master

Once your review is complete, switch back to the `master` branch:

`**git checkout master**`

5️⃣Delete the Local Feature Branch

Remove the reviewed branch from your local machine:

`**git branch -d feature-branch-name**` (whatever the relevant branch name is)

💡 If Git warns that the branch is not fully merged, force delete it with:

`**git branch -D feature-branch-name**` (be careful with this – make sure you're deleting the correct feature branch).

6️⃣Pull the Latest Updates to Stay Synced

Since you're back on `master`, update your local copy:

`git pull`

(This ensures your `master` branch is fully up-to-date before starting a new task.)

🔹 Why This Matters

✅ Prevents clutter — Keeps your local repository clean.
✅ Ensures fresh code — Always start new work with the latest updates.
✅ Avoids confusion — Prevents accidental work on outdated branches.

🚀 Follow this guide to keep your workflow efficient and your repository clean!