

# **MISSION 1:**

**Objectif : créer, structurer et versionner un projet complet avec Git.**



# Étape 1 — Mise en place du projet

## Création du dossier

Dossier créé : **projet-fil-rouge-groupeX**

## Structure du projet

- ☐ Création du dossier /src
- ☐ Création du fichier src/index.html
- ☐ Création du fichier src/style.css
- ☐ Création du dossier /docs
- ☐ Création du fichier docs/journal.md
- ☐ Création du fichier README.md

## Initialisation Git

- ☐ Ouvrir le dossier dans VS Code
- ☐ Terminal ouvert dans la racine du projet
- ☐ git init effectué
- ☐ git status vérifié

## Premier commit

- ☐ git add .
- ☐ git commit -m "Initialisation du projet"
- ☐ Log visible dans git log --oneline



# Étape 2 — Produire plusieurs commits (minimum 6)

Pour chaque modification, cocher :

## Modifications 1-3

- ☐ Modification 1 effectuée
- ☐ git add fichier
- ☐ git commit
- ☐ Journal mis à jour
- ☐ Modification 2 effectuée
- ☐ git add fichier
- ☐ git commit
- ☐ Journal mis à jour
- ☐ Modification 3 effectuée
- ☐ git add fichier
- ☐ git commit
- ☐ Journal mis à jour

## Modifications 4-6

- ☐ Modification 4 effectuée
- ☐ git add fichier
- ☐ git commit
- ☐ Journal mis à jour
- ☐ Modification 5 effectuée
- ☐ git add fichier
- ☐ git commit
- ☐ Journal mis à jour
- ☐ Modification 6 effectuée
- ☐ git add fichier
- ☐ git commit
- ☐ Journal mis à jour

(Vous pouvez continuer au-delà si nécessaire)



# Étape 3 — Mise à jour du journal de bord

## Documentation des contributions

- ☐ Chaque membre a noté ce qu'il a fait
- ☐ SHA des commits notés

## Suivi du projet

- ☐ Difficultés notées
- ☐ Répartition du travail expliquée



# Étape 4 — Supprimer et renommer un fichier

## Suppression

- ☐ Un fichier inutile choisi
- ☐ `git rm fichier`
- ☐ `git commit -m "Suppression"`
- ☐ Journal mis à jour

## Renommage

- ☐ Un fichier choisi à renommer
- ☐ `git mv ancien nouveau`
- ☐ `git commit -m "Renommage"`
- ☐ Journal mis à jour



# Étape 5 — Faire une erreur volontaire + revert

01

---

## Créer l'erreur

☐ Une erreur volontaire a été ajoutée

02

---

## Committer l'erreur

☐ L'erreur a été committée

03

---

## Annuler avec revert

☐ Le commit problématique a été revert : `git revert SHA`

04

---

## Documenter

☐ Journal mis à jour



# Étape 6 — Défi final de l'équipe

## Gestion du staging

- ☐ Modifier deux fichiers
- ☐ Ajouter **un seul** fichier dans le staging
- ☐ Faire un commit

## Annulation et création













- ☐ Modifier un fichier puis annuler la modification
- ☐ Créer un nouveau fichier dans /docs
- ☐ Commit du nouveau fichier

## Vérifications finales

- ☐ Afficher les 4 derniers commits
- ☐ Afficher les différences d'un fichier
- ☐ Vérifier que chaque membre a fait au moins un commit
- ☐ Définir ce qu'est un SHA dans journal.md



# MISSION VALIDÉE SI :

-   **6 commits minimum**
-   **Structure respectée**
-   **Journal complet**
-   **Suppression + renommage faits**
-   **Un revert effectué**
-   **Défi final complet**