

### 1. Initialisation du projet Git

- Initialise un dépôt Git pour ton projet e-commerce avec `git init`.
- Pourquoi est-ce important de créer un dépôt Git dès le début ?

### 2. Suivi des composants React

- Ajoute tous les composants (`Home.js`, `ProductList.js`, `ProductCard.js`, etc.) à l'index Git pour qu'ils soient suivis. (`git add .`)
- Sauvegarde ces changements avec un commit clair. `Git commit` => créer la première version

### 3. Gestion des branches pour tester des fonctionnalités

- Crée une branche `feature-cart` pour développer la page panier (`<Cart />`).
- Passe sur cette branche et ajoute le composant `<Cart />`. `Git add .`
- Commit les changements sur cette branche. `Git commit -u`

### 4. Fusion de fonctionnalités

- Reviens sur la branche `main` et fusionne `feature-cart` pour intégrer la page panier.

(retourner à la branche `main` : `git checkout main`

`Git merge feature-cart`)

### 5. Test et corrections sur les produits

- Crée une branche `feature-product-details` pour tester la page produit (`<ProductDetails />`).
- Modifie un produit dans `ProductDetails.js` et commit tes changements.
- Si une erreur est faite, comment revenir à la dernière version stable sur `main` ?

### 6. Historique et suivi

- Vérifie l'historique des commits avec `git log`.
- Montre comment comparer la version actuelle des fichiers avec l'index ou le dernier commit (`git diff`).

### 7. Gestion de versions et rollback

- Si un composant `<AdminPanel />` supprime par erreur un produit dans le code, quelle commande Git utiliser pour annuler cette modification avant le commit ?

### 8. Pusher vers le dépôt distant (`git push origin main`)