Equipe 13 : FATHI, DUEZ, PIRES, NEEL, PAIVA ASRBD L3

Intégration Continue

Table des matières

Réflexion sur la gestion de projet :

En premier lieu nous avons défini ce que nous voulions développer, ce fut un site marchand qui l'a emporté. Nous nous sommes attribués les tâches d'un accord commun avec les préférences de chacun sur ce qu'il y avait à faire. Si une personne finissait plus tôt sa partie que les autres, celle-ci proposait son aide aux autres que ce soit par la recherche d'informations ou simplement par la réponse à des questions dont la réponse était connue

Nous avons donc commencé par la création du repository sur GitHub avec les différentes branches.
Au début du projet nous étions tout d'abord un peu hésitant quant aux commandes qu'il fallait exécuter et comment initialiser des repository en local. Le début a donc été une phase initiatique entre nous où on s'entraidait afin d'avoir tout de prêt pour le début du développement.

Lorsque tout était préparé, nous nous sommes donnés tout d'abord des tâches simples comme tout d'abord créer un template vide pour le site web avec un index.html où ça a été une tâche commune.
Puis nous nous sommes attribués des tâches de création de pages pour le site web (Chaque personne avait une page à faire comme : la page de connexion, d'achats, faq, etc…) sous un format générique d'abord puis nous nous sommes déplacés dans les différentes branches en fonction de ce que nous essayions de développer.

Pour la partie CSS, nous sommes restés sur la branche « develop » ainsi que pour développer les différents boutons du site.
Pour les features comme une barre de recherche dynamique affichant les produits en fonction de ce que l'on écrit, nous avons estimé que c'était une feature et nous nous sommes donc déplacés vers cette branche.

Lorsque nos tests manuels étaient concluant nous mergions nos branches develop vers le « main » afin d'avoir un code propre sur celle-ci. Pour les features, nous les mergions d'abord sur la branche « develop » afin de tester avec le code déjà existant et non isoler avant de le merge avec « main ».

Lorsque la création du site a été concluante pour le groupe nous avons commencé à chercher à mettre en place différents tests unitaires et la rédaction de ce document.

Difficultés rencontrées :

Le projet n'a pas été que couronné de succès et sans accrocs, nous avons eu plusieurs erreurs avec git que nous avons dû régler.

- L'erreur qui revenait le plus souvent est l'oublie d'ajouter le(s) fichier(s) dans l'indexation des commit :

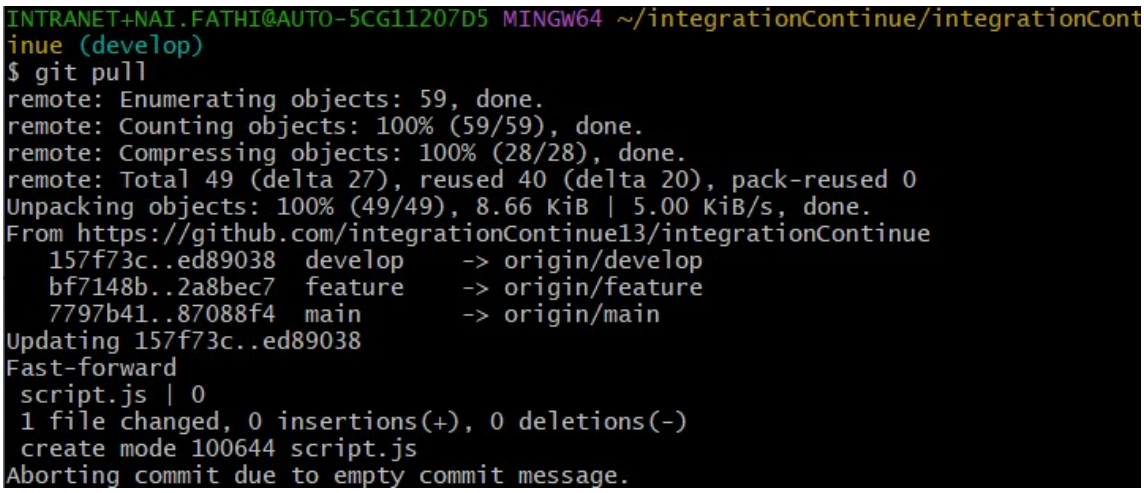
```
Julien@MSI MINGW64 ~/repo/integration_continue2023-2024/cours1/integrationContinue (develop)
$ git commit -m "modify achat"
On branch develop
Your branch is up to date with 'origin/develop'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   achat.html

no changes added to commit (use "git add" and/or "git commit -a")
```

Pour régler ce souci, on faisait simplement un « git add <nom_fichier> » puis on relançait le commit.

```
Julien@MSI MINGW64 ~/repo/integration_continue2023-2024/cours1/integrationContinue (develop)
$ git status
On branch develop
Your branch and 'origin/develop' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" if you want to integrate the remote branch with yours)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    achat.html

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   produit.html


Julien@MSI MINGW64 ~/repo/integration_continue2023-2024/cours1/integrationContinue (develop)
$ git add produit.html

Julien@MSI MINGW64 ~/repo/integration_continue2023-2024/cours1/integrationContinue (develop)
$ git status
On branch develop
Your branch and 'origin/develop' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" if you want to integrate the remote branch with yours)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    achat.html
        modified:   produit.html
```

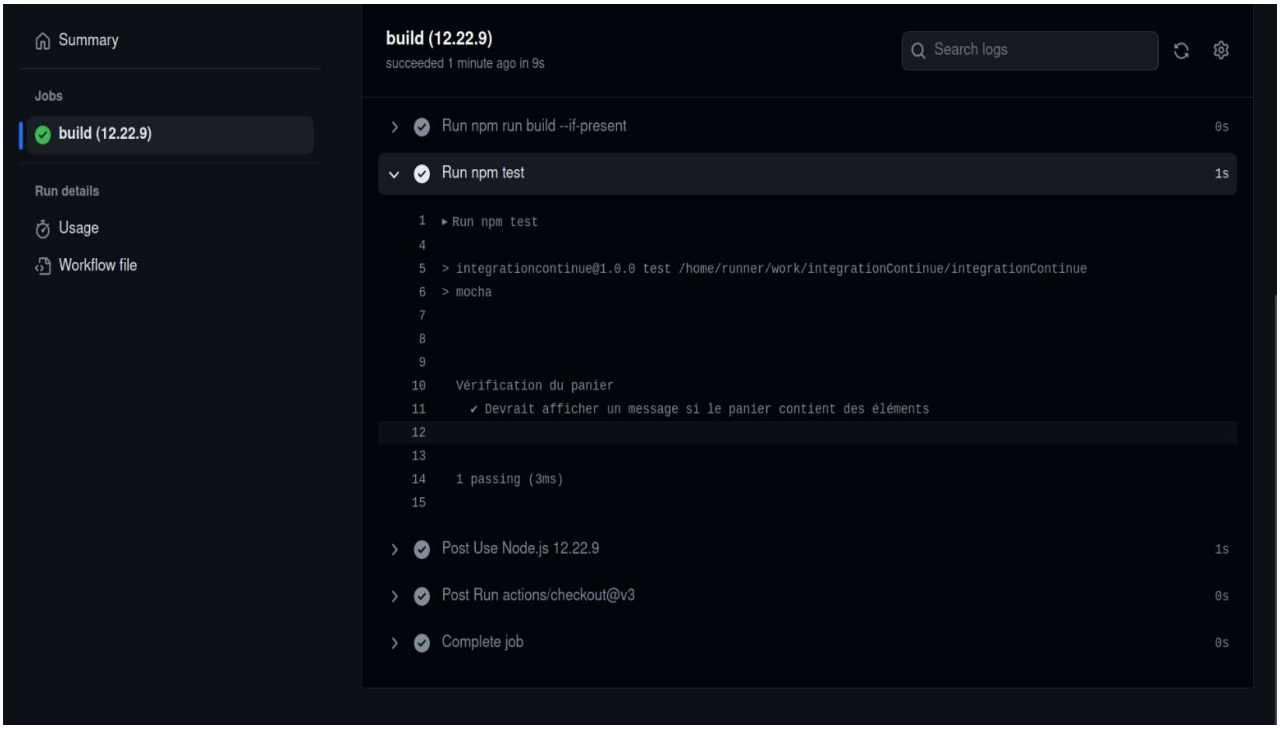
- Une autre erreur qui revenait assez souvent durant notre projet est l'erreur suivante :

Cette erreur est apparue plusieurs fois car nous n'avions pas les bonnes pratiques d'encore intégrées dans nos process où il fallait prendre l'habitude de faire des « git pull » pour mettre à jour nos branches en local avant de faire des « git push » :



Le Code et le Testing :

Voici le code pour le test unitaire :



Ce test unitaire exécute la commande npm test, pour tout action sur le git, ce test va dans notre cas tester si le panier est vide ou non.

Ce code est une simulation de test automatisé qui vérifie si le panier de courses est vide ou contient des articles. Il utilise un délai de 500 millisecondes pour simuler une vérification, puis vérifie le contenu du panier.

Le code ci-dessous est une simulation de test automatisé qui vérifie si le panier de courses est vide ou contient des articles. Il utilise un délai de 500 millisecondes pour simuler une vérification, puis vérifie le contenu du panier.
Nous avons laisser le code tel quel avec les commentaires afin d'expliquer ce que fait ce code.

```
const assert = require('assert');

// Mocking localStorage
global.localStorage = {
    getItem: function() {},
    setItem: function() {},
    removeItem: function() {}
};

describe('Vérification du panier', function() {
    it('Devrait afficher un message si le panier contient des éléments', function() {
        // Attendez un peu pour laisser le temps au panier d'être potentiellement mis à jour
        setTimeout(function() {
            // Récupérez le contenu du panier depuis le localStorage
            var panierActuel = localStorage.getItem("panier");

            // Vérifiez si le panier contient quelque chose
            if (panierActuel && JSON.parse(panierActuel).length > 0) {
                // Utilisez assert pour vérifier que le panier contient des éléments
                assert.ok(true, 'Le panier contient des éléments');
            } else {
                // Utilisez assert pour vérifier que le panier est vide
                assert.ok(true, 'Le panier est vide');
            }
        }, 500); // Attendez 500 millisecondes (ajustez selon vos besoins)
    });
});
```