

Projet de site web dynamique d'E-commerce avec HTML5, CSS3 et JavaScript

Objectif du projet : Développer un site web d'e-commerce dynamique en utilisant uniquement les technologies HTML5, CSS3 et JavaScript, avec une gestion complète du code source via Git/GitHub, et l'intégration continue/déploiement continu (CI/CD) avec GitHub Actions.

Technologies utilisées

- **Frontend :** HTML5, CSS3, Javascript.
- **Outil de contrôle de version :** Git/GitHub.
- **CI/CD :** GitHub Actions.

Livrables attendus

- Code source hébergé sur GitHub avec des commits réguliers.
- Rapport détaillé sur les choix technologiques, la structure du site web, et les étapes de déploiement continu.
- Site web d'e-commerce déployé sur un serveur de test.

Répartition des tâches (par Étudiant)

- **Designer et Intégrateur Frontend :** Création de la maquette et design du site. Développement du frontend avec HTML5, CSS3 et JavaScript.
- **Développeur JavaScript :** Implémentation des fonctionnalités dynamiques avec JavaScript.
- **Responsable Git/GitHub :** Gestion du code source avec Git/GitHub. Vérification des bonnes pratiques de gestion de version.
- **Responsable CI/CD :** Configuration de GitHub Actions pour l'intégration continue et le déploiement continu. Automatisation des tests unitaires et du déploiement sur un serveur de test.

Étapes du projet

1. Conception du site web

Définir la structure du site web d'e-commerce. Identifier les pages nécessaires (accueil, catalogue de produits, panier, etc.).

2. Maquette et design

Créer une maquette de l'interface utilisateur du site. Mettre en œuvre le design en utilisant HTML5 et CSS3 pour créer des pages attrayantes et réactives.

3. Frontend avec Javascript

Ajouter des fonctionnalités dynamiques en utilisant JavaScript :

- Ajout de produits au panier en temps réel.
- Calcul du total du panier.
- Formulaire de commande avec validation.

4. Gestion du code source avec Git/GitHub

Créer un repository GitHub pour le projet. Inviter tous les membres de l'équipe en tant que collaborateurs. Définir des branches pour les fonctionnalités, les correctifs, etc. Mettre en œuvre des bonnes pratiques de gestion de version avec des commits descriptifs.

La gestion efficace du code source est essentielle pour assurer la collaboration harmonieuse au sein de l'équipe de développement. Dans cette étape, nous mettrons en place un système de contrôle de version avec Git/GitHub.

- **Création du repository GitHub :** Commencez par créer un repository GitHub pour le projet. Choisissez un nom significatif et ajoutez une description détaillée du projet. Assurez-vous que le repository est initialisé avec un fichier README.
- **Inviter les collaborateurs :** Invitez tous les membres de l'équipe en tant que collaborateurs du repository. Cela permettra à chaque membre de contribuer au code et d'accéder aux fonctionnalités de gestion du code source. **Nb :** ajouter aussi votre enseignant (**username : YvanGuifo**) comme collaborateur de votre repository.
- **Définir les branches :** Établissez une stratégie de gestion des branches. Créez des branches pour les fonctionnalités, les correctifs de bugs, et éventuellement une branche principale (main) qui représente la version stable du code.
- **Bonnes pratiques de gestion de version :** Enforcez des bonnes pratiques de gestion de version. Encouragez des messages de commit descriptifs et informatifs. Utilisez des branches spécifiques pour chaque fonctionnalité ou correctif.
- **Gestion des issues :** Utilisez la fonctionnalité d'Issues de GitHub pour suivre les tâches, les bogues, et les améliorations prévues. Chaque issue peut être liée à une branche spécifique.
- **Discussions, commentaire et revue de code :**
 - Encouragez les discussions et les commentaires au sein des pull requests et des issues. Cela facilitera la communication au sein de l'équipe et permettra de résoudre rapidement les problèmes.
 - Instaurez une pratique de revue de code avant la fusion des branches. Cela garantira la qualité du code et permettra aux membres de l'équipe de partager leurs connaissances.
- **Documentation du processus :** Documentez le processus de gestion du code source. Expliquez les étapes à suivre pour créer une nouvelle fonctionnalité, soumettre un correctif, et effectuer une revue de code. La mise en place d'une gestion efficace du code source garantira un flux de travail fluide tout au long du projet, facilitant la collaboration et minimisant les conflits potentiels.

5. Gestion du code source avec Git/GitHub

- **Intégration avec GitHub Actions :**
 - Explorez les fonctionnalités de CI/CD offertes par GitHub Actions. Configurez des workflows pour automatiser les tests unitaires et le déploiement continu sur un serveur de test.
 - Configurer GitHub Actions pour automatiser les tests unitaires.
 - Mettre en place un pipeline de déploiement continu pour déployer automatiquement les modifications validées sur un serveur de test.
- **Tests et validation :**
 - Mettre en place des tests unitaires en utilisant JavaScript. Effectuer des tests manuels pour s'assurer de la qualité de l'interface utilisateur.

6. Documentation

Documenter le code (y compris les conventions de codage). Créer une documentation utilisateur pour expliquer le fonctionnement du site d'e-commerce.