

Cahier des Charges - Projet Tech Web 2

Objectif du Projet

Vous devrez réaliser un site web basé sur un sujet de votre choix. Ce site devra respecter les principes suivants :

- Utilisation du **Modèle-Vue-Contrôleur (MVC)** pour l'architecture.
- Stockage des données en format **JSON**.
- Programmation en **TypeScript** pour la logique et en **HTML/CSS** pour l'affichage.
- Mise en place d'une méthodologie de projet structurée incluant :
 - Un dépôt Git avec gestion des branches.
 - Des tests unitaires avec **JEST**.

1. Livrables attendus

1. **Code source** organisé et structuré :
 - Répertoire bien structuré (ex. `src`, `tests`, `dist`, `assets`, etc.).
 - Fichiers séparés pour les modèles, contrôleurs et vues.
2. **Dépôt Git** hébergé sur GitHub :
 - Branches bien définies : `main/master`, `dev`, `test`.
 - Historique des commits propre et descriptif.
3. **Documentation** :
 - Fichier `README.md` détaillant :
 - Le sujet du site web.
 - La configuration du projet.
 - Les instructions pour exécuter le projet et les tests.
4. **Tests unitaires** avec JEST :
 - Tests pour valider le fonctionnement des classes et des principales fonctionnalités.

2. Étapes du Projet

Étape 1 : Initialisation

- Choisir un sujet et rédiger une brève description du site.
- Créer le dépôt GitHub et initialiser les branches :
 - **main/master** : branche stable.
 - **dev** : branche pour le développement.
 - **test** : branche pour les tests unitaires.

Étape 2 : Modélisation des Données

- Identifier les entités principales du site (exemple : Utilisateurs, Produits, Articles).
- Créer un fichier **JSON** contenant des données fictives pour ces entités.

Étape 3 : Développement de l'Architecture MVC

1. Modèle

- Créer des classes TypeScript représentant les données.
- Ajouter des méthodes pour charger les données depuis le fichier JSON.

2. Vue

- Développer les pages web en HTML/CSS pour afficher les données.
- Utiliser des classes CSS pour une présentation propre et responsive.

3. Contrôleur

- Créer des classes ou des modules TypeScript pour gérer les interactions entre le modèle et la vue.
 - Exemple : chargement des données, gestion des clics utilisateur, navigation.

Étape 4 : Tests Unitaires avec JEST

- Configurer JEST pour Typescript
- Écrire des tests pour valider le comportement des classes et méthodes.

Étape 5 : Gestion de Projet avec Git

- Utiliser Git pour versionner le projet :
 - Faire des commits réguliers et descriptifs.
 - Utiliser les branches pour isoler les développements.
- Fusionner les branches en respectant une stratégie (ex. pull requests).

3. Critères d'Évaluation

Critère	Détails
Respect du sujet	Sujet clairement défini et respecté.
Structure MVC	Architecture claire et bien implémentée (modèle, vue, contrôleur).
Utilisation de JSON	Données bien stockées et utilisées depuis un fichier JSON.

Critère	Détails
Code TypeScript	Code propre, structuré et respectant les bonnes pratiques TypeScript.
HTML/CSS	Pages esthétiques et responsive.
Tests unitaires (JEST)	Tests présents, fonctionnels et pertinents.
Gestion Git	Utilisation des branches, commits descriptifs, dépôt bien structuré.

4. Conseils et Ressources

Ressources suggérées

- [Documentation TypeScript](#)
- [Documentation JEST](#)
- [Tutoriel sur MVC](#)
- [Tutoriel Git](#)

5. Planification

Ordre (Date si nécessaire)	Tâche
1 (02/12)	Choix du sujet, initialisation du dépôt Git.
2	Modélisation des données (JSON).
3	Développement des modèles et des vues.
4	Développement du contrôleur.
5	Ajout des tests unitaires, finalisation et documentation.
6 (06/01)	Dépôt eCampus.

Bonne chance !