**Rapport de projet - Portfolio**

**Valentin Martine  
Technologies du Web**2024–2025 **Réalisation d’un portfolio web personnel avec intégration d’un projet interactif**

**Lien du site :** [**https://valentinmartine.github.io/portfolio**](https://valentinmartine.github.io/portfolio)

**1. Objectif du projet**

L’objectif était de créer un portfolio web personnel permettant de présenter mon parcours académique, mes compétences, mes intérêts, mes expériences et un projet interactif : un quiz sur les médicaments. Ce portfolio devait être conçu en HTML/CSS/JS en respectant les pratiques vues en cours, tout en expliquant les ajouts non abordés.

**2. Maquette du site et arborescence**

Le site repose sur une navigation unique, avec des sections verticales accessibles via une table des matières horizontale fixe :

portfolio/

── index.html

── css/

└── style.css

── js/

└── script.js

── projets/

└── quiz\_medicaments/

├── index2.html

├── questions.js

├── quiz.js

└── style2.css

── rapport\_projets.docx

**3. Cours appliqués**

**HTML (CM1)**

* Structure avec <head>, <body>, <section>, <nav>
* Titres hiérarchiques (<h1>, <h2>, <h3>)
* Paragraphes <p>, listes <ul>, <li>
* Navigation interne avec id et href
* Formulaires avec <input>, <button>
* Sémantique et indentation respectées

**CSS (CM2)**

* Feuille de style externe liée via <link>
* Sélecteurs par balise, classe et ID
* Mise en forme (polices, marges, couleurs, responsive)
* Media queries pour adaptation mobile
* style2.css séparé pour le quiz

**JavaScript (CM3 et CM4)**

* DOM manipulé avec getElementById, querySelector
* Événements utilisateur (onclick, submit)
* Logique quiz : variables, tableaux, boucles, fonctions
* script.js utilisé pour le scroll fluide (effet UX)
* localStorage utilisé pour connexion locale
* Code clair, modulaire, bien commenté

**4. Projet interactif : Quiz sur les Médicaments**

Accessible depuis la section “Portfolio”, le quiz propose :

* 3 questions tirées aléatoirement parmi 10
* Affichage immédiat des bonnes/mauvaises réponses
* Score final affiché
* Possibilité de recommencer le quiz

Structure technique :

* index2.html → structure HTML du quiz
* questions.js → base de données des questions
* quiz.js → logique dynamique (DOM, random, score)
* style2.css → style dédié
* script.js → scroll fluide et UX (extérieur au quiz)

**4.1 Connexion requise pour accéder au quiz**

Afin de respecter la consigne “les utilisateurs doivent pouvoir se connecter pour accéder au projet”, j’ai intégré une authentification locale.

Fonctionnement :

* Création de compte via formulaire
* Données stockées dans localStorage
* Connexion avec identifiants → accès au quiz
* Message de bienvenue + bouton “Se déconnecter”

Justification pédagogique :  
Bien que non vu en cours, cette méthode utilise des fonctions simples (localStorage.setItem, getItem, removeItem) pour créer une logique d’accès restreint sans base de données ni serveur.

Limites :

* Mots de passe non sécurisés
* Fonctionne uniquement dans le navigateur local
* Pas multi-utilisateur réel

Mais elle est pertinente dans un cadre pédagogique local et respecte les exigences du projet.

**5. Déploiement du site**

* Plateforme utilisée : GitHub Pages
* Dépôt GitHub : <https://github.com/ValentinMartine/portfolio>
* Branche : main, racine /
* URL publique : <https://valentinmartine.github.io/portfolio>

**6. Difficultés rencontrées**

* Séparer proprement les styles (portfolio vs quiz)
* Organiser la logique JS pour éviter les conflits
* Gérer le responsive sur petits écrans
* Remplacer alert() par des affichages DOM plus élégants

**7. Conclusion et perspectives**

Ce projet m’a permis de :

* Mettre en œuvre les compétences acquises en HTML, CSS et JavaScript
* Construire une interface claire, dynamique et responsive
* Ajouter des interactions logiques via l’authentification locale
* Publier un site web personnel fonctionnel et propre

Évolutions possibles :

* Ajouter des thématiques (neuro, antibiotiques, etc.)
* Stocker les scores dans localStorage ou en ligne
* Créer un classement des utilisateurs
* Intégrer une base de données réelle via Firebase ou Supabase