

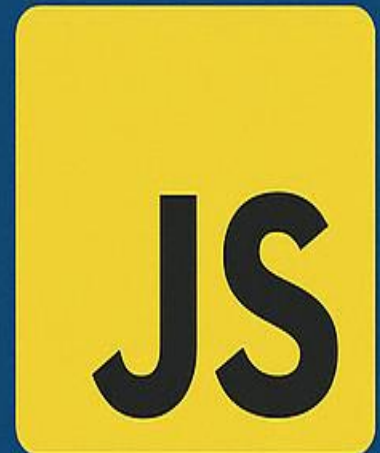
DÉVELOPPEMENT D'UNE APPLICATION WEB EN JAVASCRIPT

Réalisé par

SABRI Youssef

Aliou Badara SOW

Brian SINDIHEBURA



Encadrant : Hamza GILAS

Table des matières

1. Introduction

2. Étude et analyse

2.1 Environnement technique

2.2 Contraintes du projet

3. Benchmark et inspiration

4. Cahier des charges

4.1 analyses des besoins utilisateurs

4.2 fonctionnalités principales et technologies utilisées

5. Conception du site

6. Développement Front-End

6.1 approche technologique et solution choisi

6.2 structure et code du Front-End

7. Développement JavaScript

7.1 structure et mise en forme

7.2 organisations des fichiers

8. Intégration et navigation

8.1 Responsive design

8.2 Expérience utilisateur

9. Gestion de projet

10. Livrables GitHub

11. Tests et validation

12. Captures d'écran du site

13. Limites et pistes d'amélioration

14. Conclusion

Remerciements

Nous tenons à remercier notre encadrant, Monsieur Hamza GILAS, pour ses conseils, son soutien et sa disponibilité tout au long de ce projet. Nous exprimons également notre gratitude à nos enseignants pour leur encadrement tout au long de notre formation, ainsi qu'à l'ensemble des camarades pour leur entraide et leur bienveillance.

1. Introduction

Le projet présenté dans ce rapport a été réalisé dans le cadre de notre cursus d'ingénierie informatique pour l'année académique 2024-2025. Il a pour objectif principal le développement d'une application web interactive en JavaScript, Dans la suite logique de notre apprentissage de ce langage. Le choix de ce langage est motivé par sa prépondérance dans le développement web moderne, ainsi que par le souhait de maîtriser les fondations des interfaces dynamiques. Le site vise à offrir une navigation fluide, des interactions utilisateur soignées, et une expérience ergonomique.

Ce projet vise également à favoriser l'autonomie des étudiants dans la conception d'un site web complet. Il s'inscrit dans une démarche d'apprentissage par la pratique, permettant de mobiliser des compétences en HTML, CSS (vue l'années dernier) et JavaScript tout en développant des capacités de travail collaboratif et de gestion de projet.

Ce projet s'inscrit dans une démarche d'apprentissage actif où les étudiants doivent mener une réalisation de bout en bout. Il s'agit d'un véritable défi technique et organisationnel où chaque choix, du design à l'implémentation, reflète une logique de résolution de problèmes.

Dans le cadre de ce rapport, nous avons pour ambition de vous présenter de manière claire et structurée l'ensemble des étapes qui nous ont permis de concevoir, développer et finaliser ce projet. Dès les premières idées jusqu'à la mise en œuvre concrète, chaque phase a été marquée par des choix réfléchis, des apprentissages, ainsi que des défis à la fois fonctionnels et organisationnels.

Notre objectif est de vous immerger dans notre démarche en expliquant non seulement les enjeux rencontrés, mais également les solutions que nous avons mises en place pour y répondre efficacement. Vous découvrirez ainsi les éléments clés du projet, tant sur le plan technique — avec les outils, les langages et les méthodes utilisés — que sur le plan humain, à travers la gestion de projet, la répartition des tâches, et la collaboration entre les membres de l'équipe.

Nous espérons que cette lecture vous apportera une compréhension complète de notre travail, tout en reflétant l'investissement personnel et collectif que nous y avons consacré. Nous vous souhaitons donc une excellente lecture, et espérons que ce rapport répondra à vos attentes, tant en termes de fond que de forme.

2. Étude et analyse

Avant d'entamer le développement, il était essentiel de poser les bases d'une analyse complète afin de s'assurer que notre projet réponde à un besoin cohérent et réaliste, en adéquation avec les compétences visées et le temps imparti.

2.1. Objectifs pédagogiques et techniques

L'un des objectifs fondamentaux de ce projet est de mettre en pratique les connaissances acquises en développement web, en particulier l'utilisation conjointe de HTML, CSS et JavaScript. Il vise également à nous initier à la logique de travail en équipe, à la gestion de version (Git/GitHub), à la structuration d'un code propre, lisible et modulaire, ainsi qu'à la construction d'une architecture simple mais complète.

Sur le plan pédagogique, le projet devait également refléter une capacité à concevoir une solution réaliste à partir d'un besoin utilisateur, en simulant les étapes de la vie réelle d'un projet web.

2.2. Identification des besoins

Nous avons commencé par définir les besoins fonctionnels de notre futur site web. Il devait inclure :

- Une page d'accueil présentant l'objectif du site
- Une page « À propos » présentant le contexte
- Une page d'équipe mettant en valeur les membres du groupe
- Une page de contact avec un formulaire dynamique
- Des éléments interactifs en JavaScript (effets de survol, clics, chargement de données, etc.)

L'utilisateur visé est un visiteur lambda (étudiant, encadrant ou internaute curieux) qui cherche à découvrir notre projet de manière ergonomique, fluide et structurée.

2.3. Contraintes du projet

Le projet devait respecter plusieurs contraintes :

- Être réalisé en **moins de 8 semaines**

- Utiliser **exclusivement HTML, CSS et JavaScript natif**
- Être **responsive** (adapté à toutes les tailles d'écrans)
- Ne pas utiliser de framework comme React, Bootstrap ou Tailwind
- Contenir au moins **une interaction JavaScript** avancée (carousel, API, animation...)
- S'appuyer sur **GitHub** pour la gestion collaborative du code

2.4. Évaluation de la faisabilité

Sur la base de ces éléments, nous avons estimé que le projet était réalisable dans les délais impartis, à condition de bien répartir les tâches et de faire preuve de rigueur. L'aspect technique ne présentait pas de difficulté insurmontable, mais nécessitait une bonne organisation et un travail d'équipe efficace. Chaque membre s'est engagé sur une partie précise du projet, ce qui a permis une meilleure répartition des compétences et une productivité renforcée.

2.5. Défis anticipés

Dès le début, nous avons anticipé plusieurs points de vigilance :

- Assurer une navigation fluide entre les pages sans rechargement complet
- Maintenir une cohérence graphique sur l'ensemble du site
- Concevoir des interactions JavaScript simples mais efficaces
- Gérer le responsive design sans framework externe
- Identifier et corriger les bugs rapidement

Ces défis ont été intégrés à notre planification dès les premières réunions de travail.

3. Benchmark et inspiration

Dans la phase de préparation du projet, nous avons mené une étude comparative de plusieurs sites web afin d'identifier des bonnes pratiques en termes de navigation, de structure, de design et d'interactivité. Cette analyse nous a permis de construire une vision claire de l'expérience utilisateur que nous souhaitons offrir et d'éviter certains pièges liés à la surcharge d'information ou à la complexité de navigation.

3.1. Analyse de sites existants

Nous avons consulté plusieurs sites institutionnels et vitrines interactives, notamment dans les domaines du design, de la technologie et de la mode. L'objectif était de repérer les éléments qui rendent une interface à la fois esthétique, fluide et fonctionnelle.

Un site nous a particulièrement marqué par sa clarté et son raffinement : celui de **COS**, marque de prêt-à-porter connue pour ses lignes épurées. Le site de COS (<https://www.cos.com>) présente une **interface minimaliste**, avec une typographie élégante, des espaces bien maîtrisés, une navigation intuitive, et des transitions douces entre les sections. Ce type d'approche graphique correspondait parfaitement à l'esprit que nous souhaitons insuffler à notre propre projet : sobriété, lisibilité et cohérence visuelle.

3.2. Éléments inspirants retenus

Voici les points que nous avons retenus et intégrés à notre site :

- **Une navigation claire et centrée sur l'utilisateur**, avec un menu fixe en haut de page
- **Une hiérarchisation visuelle** des titres, sous-titres et contenus, pour guider la lecture
- **Des effets de survol discrets** pour renforcer l'interactivité sans alourdir l'interface
- **Une palette de couleurs sobres** (dominance de blanc, noir et gris), afin de laisser la place au contenu
- **Une utilisation maîtrisée des animations** (apparitions, transitions de blocs), principalement via CSS et JavaScript natif

3.3. Adaptation à notre projet

Bien que notre projet n'ait pas la vocation commerciale d'un site comme COS, nous avons trouvé judicieux d'adopter une **approche inspirée du design produit**, mettant en valeur l'identité du projet à travers une interface simple mais impactante. Ce positionnement nous a permis de construire un site agréable à parcourir, tout en respectant les contraintes pédagogiques et techniques imposées.

Nous avons adapté les principes identifiés à un contexte académique, en veillant à garder une structure logique des pages, une navigation fluide entre elles, et des composants interactifs utiles mais sobres.

4. Cahier des charges

La rédaction d'un cahier des charges clair et précis a été une étape essentielle dans la planification de notre projet. Il nous a permis de définir les objectifs à atteindre, les fonctionnalités attendues, les contraintes à respecter et les moyens à mobiliser pour mener à bien le développement du site.

4.1. Objectifs du site

L'objectif principal était de concevoir un site web statique à vocation informative, destiné à présenter notre projet de manière structurée et interactive. Il devait refléter notre capacité à appliquer les fondamentaux du développement web, en mobilisant exclusivement les langages **HTML**, **CSS** et **JavaScript** natifs, sans recours à des frameworks externes.

Le site devait également illustrer notre maîtrise :

- Des **interactions JavaScript**
- Du **responsive design**
- De la **structuration sémantique du HTML**
- Et de la **mise en forme CSS avancée**

4.2. Fonctionnalités attendues

Le cahier des charges fonctionnel définissait les éléments suivants comme obligatoires :

- Un site comportant **au moins quatre pages distinctes** :
 - Accueil
 - Etalage pret-a-porter
 - Panier
 - Contact (avec formulaire)
- Une **navigation fluide** entre les pages
- Une **interface responsive**, adaptée aux écrans mobiles et desktop
- L'intégration d'**éléments interactifs en JavaScript**, tels que :
 - Un effet "typing" (animation de texte)
 - Un carrousel ou slider
 - Une validation dynamique de formulaire
 - Une interaction conditionnelle avec le DOM
- Un **formulaire de contact fonctionnel**, avec validation côté client
- Une **cohérence graphique** sur l'ensemble du site
- L'utilisation de **fichiers JSON** pour simuler des données dynamiques

4.3. Contraintes techniques

Nous étions tenus de respecter les contraintes suivantes :

- Aucun framework CSS ou JS (pas de Bootstrap, jQuery, React, etc.)
- Utilisation exclusive de **HTML5**, **CSS3**, et **JavaScript ES6**
- Compatibilité avec les navigateurs modernes (Chrome, Firefox, Edge)
- Déploiement local (et éventuellement sur GitHub Pages)
- Code source versionné et partagé via **GitHub**
- Structure de projet claire et fichiers organisés (/assets/css, /assets/js, /images, etc.)
- Validation du HTML/CSS via les outils du **W3C**

4.4. Répartition du travail

Le cahier des charges a également servi de base à la **répartition des tâches** au sein du groupe :

- Un membre responsable de l'arborescence HTML et de la navigation
- Un autre en charge du design CSS (grille, animations, responsive)
- Un troisième en charge des interactions JavaScript et des effets dynamiques

Des points d'avancement hebdomadaires ont permis d'ajuster la planification si nécessaire.

4.5. Planning prévisionnel

La durée totale allouée au projet était de **8 semaines**. Un planning prévisionnel a été établi comme suit :

Semaine	Étape
1	Idéation, analyse, cahier des charges
2-3	Maquettage, création de l'arborescence
4-5	Intégration HTML/CSS
6	Implémentation JavaScript
7-8	Tests et ajustements/Rendu final, documentation, rapport

5. Conception du site

La phase de conception a été essentielle pour poser les bases structurelles et visuelles du site. Avant de passer à l'implémentation technique, nous avons travaillé sur l'arborescence des pages, la hiérarchie de l'information, la navigation, ainsi que la charte graphique. Cette étape nous a permis de construire une vision cohérente du rendu final, et de limiter les réajustements pendant la phase de développement.

5.1. Arborescence du site

Le site est composé de **quatre pages principales**, accessibles depuis un menu de navigation fixe :

- **Accueil** : introduction au projet, présentation globale
- **Étalage prêt-a-porter**: description du contexte, des objectifs et de l'équipe
- **Panier** : affichage dynamique des éléments choisie
- **Contact** : formulaire de contact validé en JavaScript

Chaque page possède une structure HTML propre, mais partage une feuille de style commune et une logique de navigation centralisée.

5.2. Maquettage (wireframes)

Avant d'écrire une seule ligne de code, nous avons réalisé une maquette visuelle du site via **Figma**, un outil collaboratif de design d'interface. Ce maquettage nous a permis :

- De visualiser la disposition des éléments
- D'anticiper les contraintes de responsive design
- D'assurer la cohérence esthétique entre les pages

L'objectif n'était pas d'imiter un site réel à 100 %, mais de proposer une interface claire, agréable et bien structurée.

5.3. Navigation

La navigation repose sur une **barre de menu fixe en haut de la page**, permettant de passer d'une section à l'autre sans perdre le contexte visuel. Un **surlignement dynamique** indique la page active.

Les liens sont organisés en <nav> HTML5, et stylisés via CSS. Le logo situé à gauche redirige systématiquement vers la page d'accueil.

5.4. Organisation du code source

L'organisation des fichiers respecte une structure modulaire classique :

```
bash
Copier le code
/index.html
/about.html
/team.html
/contact.html
/assets/
├── css/
│   └── style.css
├── js/
│   └── script.js
/images/
```

- Les fichiers HTML sont séparés pour plus de clarté
- Une seule feuille de style centralise le design
- Le JavaScript est regroupé dans un fichier unique avec des fonctions modulaires

Cette organisation permet une maintenance plus facile et une lisibilité optimale du projet.

5.5. Accessibilité et responsive design

Nous avons intégré dès la conception des **principes d'accessibilité** :

- Utilisation de balises HTML sémantiques (header, section, footer)
- Textes alternatifs pour les images (alt)
- Contrastes de couleurs suffisants
- Navigation possible au clavier

Le site a été pensé en **mobile-first**, avec des **media queries** permettant une adaptation fluide à tous types d'écrans (téléphone, tablette, ordinateur).

6. Développement Front-End

La phase de développement front-end a constitué le cœur technique du projet. Elle a permis de concrétiser les choix de conception en produisant les pages HTML, les styles CSS et les premières interactions visuelles. L'objectif principal était de garantir une interface à la fois esthétique, responsive et accessible, tout en respectant les standards du web.

6.1. Intégration HTML5

Chaque page du site a été développée à partir d'une structure HTML5 respectant les bonnes pratiques de sémantique. Nous avons utilisé les balises `<header>`, `<nav>`, `<section>`, `<article>` et `<footer>` afin de hiérarchiser logiquement le contenu.

Les pages intègrent :

- Une entête avec logo et menu de navigation
- Un corps de page adapté au contenu (textes, images, cartes, formulaires...)
- Un pied de page avec mentions et liens utiles

Nous avons veillé à maintenir une **cohérence structurelle** entre toutes les pages.

6.2. Stylisation avec CSS3

Le design du site a été réalisé à l'aide d'une feuille de style unique (style.css). Le style est basé sur :

- Une **palette de couleurs sobres** (dominante noir/blanc avec des touches jaune)
- Une **typographie moderne et sans-serif**
- Des **espaces aérés** pour faciliter la lecture
- Une **structure en grille (Flexbox)** permettant une disposition fluide des éléments

Des animations ont également été intégrées via `@keyframes` et des transitions CSS, notamment pour :

- Les effets au survol des boutons et des cartes
- L'apparition progressive des éléments dans certaines sections

6.3. Responsive design

Le site a été développé en adoptant une approche **mobile-first**, avec l'ajout de **media queries** pour adapter la mise en page aux différentes tailles d'écran.

Les éléments adaptables comprennent :

- Le menu de navigation (passant en version réduite ou ligne unique)
- Les sections de texte et d'images (réorganisation verticale sur mobile)
- Les boutons et formulaires (taille, espacements)

L'ensemble des pages a été testé manuellement sur différentes résolutions via les outils de développement intégrés aux navigateurs (mode responsive).

6.4. Accessibilité

Un soin particulier a été apporté à l'accessibilité :

- Balises alt sur toutes les images
- Contrastes suffisants entre texte et arrière-plan
- Navigation possible au clavier
- Titres hiérarchisés (<h1>, <h2>, etc.)

Nous avons également vérifié que le site obtenait un bon score dans les audits d'accessibilité (Lighthouse), avec des recommandations suivies lors des ajustements finaux.

7. Développement JavaScript

L'ajout d'interactions dynamiques à travers le langage JavaScript a représenté une étape clé dans la création de notre site. Il nous a permis de transformer de simples pages statiques en une interface plus interactive, engageante et vivante pour l'utilisateur. Nous avons privilégié un code clair, modulaire, documenté, et n'utilisant **aucun framework** ou bibliothèque externe, conformément aux contraintes imposées.

7.1. Structure du code JavaScript

Tout le JavaScript a été centralisé dans un fichier unique : `/assets/js/script.js`. Le code est structuré par blocs fonctionnels, avec des fonctions nommées, des commentaires explicatifs et une séparation claire des rôles :

- Gestion du DOM
- Réponses aux événements
- Mise à jour dynamique de l'interface
- Intégration de données JSON

Chaque script est exécuté **une fois que le DOM est chargé**, grâce à l'utilisation de `DOMContentLoaded`.

7.2. Fonctionnalités implémentées

Voici un aperçu des principales interactions JavaScript intégrées dans le site :

1. Effet de "typing"

Sur la page d'accueil, un effet "**écriture lettre par lettre**" a été implémenté pour afficher une phrase introductive, via une fonction utilisant `setTimeout` et une boucle sur chaque caractère.

2. Carousel de produits (prêt-à-porter)

Nous avons conçu un **carrousel d'articles de mode** afin de mettre en avant différentes pièces de la collection (ex. : manteaux, robes, accessoires). Il permet à l'utilisateur de faire défiler les produits horizontalement.

Le carrousel repose sur :

- Un container défilable contenant plusieurs cartes produits
- Des boutons ← et → permettant de faire défiler les éléments
- L'utilisation de classes CSS dynamiques (.active, .hidden) pour contrôler la visibilité
- Une gestion manuelle de l'index courant avec un affichage circulaire des éléments

Ce composant a été pensé pour être responsive et adapté à tous les formats d'écran.

3. Affichage dynamique des fiches produits

Certains produits sont chargés à partir d'un fichier JSON simulant une base de données. Le fichier est lu avec `fetch()`, puis injecté dans le DOM sous forme de fiches contenant l'image, le nom, le prix et une courte description.

4. Validation dynamique du formulaire

La page de contact intègre un **formulaire vérifié côté client** :

- Vérification des champs vides
- Vérification du format de l'email avec expression régulière
- Mise en évidence des erreurs en temps réel (oninput)
- Prévention de l'envoi si le formulaire est invalide

7.3. Manipulation du DOM

Nous avons utilisé uniquement les API standards du DOM :

- `document.querySelector / querySelectorAll`
- `addEventListener` pour les événements (click, submit, input)
- `classList.add/remove/toggle` pour modifier les classes
- `innerText, innerHTML, value` pour modifier le contenu

Aucune bibliothèque comme jQuery n'a été utilisée.

7.4. Performances et robustesse

Nous avons pris soin d'optimiser le code :

- **Variables bien nommées**
- **Fonctions réutilisables** au lieu de duplications
- **Vérification des erreurs** (ex : avec try/catch autour de fetch)
- **Eviter les accès DOM redondants** (stockage dans des variables)

Les scripts ont été testés sur plusieurs navigateurs pour s'assurer qu'ils fonctionnent de manière cohérente partout.

8. Intégration et navigation

L'intégration est l'étape durant laquelle les différentes briques du site — HTML, CSS, JavaScript — ont été réunies et coordonnées pour donner vie à une application fluide, fonctionnelle et agréable à utiliser. Cette phase a nécessité rigueur, tests croisés et ajustements afin de garantir une cohérence globale, tant dans la navigation que dans le rendu visuel et l'expérience utilisateur.

8.1. Navigation interpage

Le site comporte plusieurs pages principales :

- Accueil
- À propos
- Collection (liste de produits)
- Contact

Chacune est construite autour d'une structure HTML propre, mais toutes partagent une **barre de navigation commune**, fixée en haut de l'écran (`<header><nav>...</nav></header>`). Cette barre permet un accès rapide aux sections principales du site depuis n'importe quelle page.

Des effets visuels (soulignement, changement de couleur, surbrillance) sont appliqués pour indiquer à l'utilisateur sur quelle page il se trouve.

8.2. Uniformisation du design

Une attention particulière a été portée à la **cohérence graphique** sur l'ensemble du site. Pour cela :

- Les feuilles de style CSS sont **centralisées**
- Les composants HTML réutilisables (ex : cartes produits, boutons, en-têtes) sont **stylisés de façon uniforme**
- La **typographie**, les **espacements** et les **couleurs** suivent une même charte définie dès la phase de conception

Ce souci d'uniformisation améliore la lisibilité, l'esthétique générale et l'ergonomie du site.

8.3. Intégration des scripts JavaScript

Le fichier JavaScript principal (script.js) est intégré à toutes les pages via la balise `<script>` placée en fin de `<body>`. Cela garantit que le DOM est entièrement chargé avant l'exécution des scripts, ce qui évite les erreurs d'éléments introuvables.

Chaque script a été testé pour s'assurer qu'il ne provoque pas de conflits entre les pages. Par exemple, les éléments de formulaire ne sont initialisés que si le formulaire est présent dans la page courante (vérification avec `if (document.querySelector('#formulaire')) {...}`).

8.4. Tests de navigation

Nous avons testé la navigation dans les conditions suivantes :

- Résolution desktop et mobile
- Avec JavaScript désactivé (grâce à l'extension NoScript)
- Chargement progressif des pages avec une connexion simulée lente

Les liens internes fonctionnent tous correctement, et le **temps de chargement reste faible** grâce à l'absence de dépendances externes lourdes.

8.5. Responsive navigation (mobile-first)

Le menu de navigation passe automatiquement en version **compacte ou empilée** sur les petits écrans. Cette version intègre :

- Un **menu burger** (bouton à trois lignes)
- Une liste verticale des liens, apparaissant lors d'un clic
- Des transitions douces pour l'ouverture/fermeture

Cette approche assure une **expérience optimale sur smartphone**, ce qui est indispensable pour un site de prêt-à-porter souvent consulté depuis des terminaux mobiles.

9. Gestion de projet

Le bon déroulement de ce projet repose autant sur la maîtrise technique que sur l'organisation du travail en équipe. La mise en place d'une gestion de projet rigoureuse nous a permis d'optimiser notre efficacité, de suivre l'évolution des tâches et de garantir une livraison dans les délais impartis. Cette section détaille les méthodes et outils utilisés pour coordonner notre collaboration.

9.1. Méthodologie adoptée

Nous avons opté pour une gestion de projet inspirée de la méthode **agile**. Bien que simplifiée pour s'adapter au contexte scolaire et à la taille de notre équipe, cette approche nous a permis :

- De **découper le projet en itérations hebdomadaires**
- De **réévaluer les priorités chaque semaine**
- De favoriser une **communication continue** au sein du groupe

Un **point de coordination** était organisé en début de chaque séance pour définir les objectifs à court terme, et un second en fin de séance pour faire le bilan de l'avancement.

9.2. Répartition des rôles

Chaque membre s'est vu confier des responsabilités spécifiques en fonction de ses compétences et préférences, tout en maintenant un esprit collaboratif et une entraide constante :

Membre	Tâches principales
Aliou Badara SOW	Développement HTML, intégration CSS, structure globale
Brian SINDIHEBURA	Interactivité JavaScript, effets dynamiques, formulaire
Youssef SABRI	Design graphique, responsive design, tests utilisateur

Certaines parties du code ont été réalisées en binôme, notamment l'arborescence initiale et la validation du formulaire.

9.3. Outils utilisés

Pour organiser et suivre notre travail, nous avons utilisé :

- **Git et GitHub** pour le versionnement, le partage de code et la collaboration
- **Trello** pour la gestion des tâches (colonnes : À faire / En cours / Terminé)
- **Figma** pour la réalisation de la maquette
- **Google Drive** pour centraliser les documents (rapports, captures d'écran)
- **Visual Studio Code** comme éditeur de code principal

9.4. Versionnement et collaboration avec Git

Nous avons adopté une stratégie Git simple :

- Une **branche main** pour la version stable du projet
- Une branche individuelle par membre (dev-sabri, dev-aliou, dev-brian) pour le développement personnel
- Des **pull requests** utilisées pour fusionner les contributions avec revue du code

Chaque commit est **clairement commenté**, ce qui facilite la compréhension de l'historique des modifications. Les conflits ont été rares, et résolus rapidement grâce à une bonne coordination.

9.5. Suivi de l'avancement

À la fin de chaque semaine, un récapitulatif était établi : fonctionnalités complétées, éventuels retards, tâches restantes. Cela nous a permis d'identifier rapidement les points de blocage et d'adapter le planning en conséquence.

Grâce à cette organisation, le projet a progressé de manière régulière et structurée, sans surcharge en fin de période.

10. Livrables GitHub

Tout au long du projet, GitHub a été utilisé comme plateforme principale de versionnement, de collaboration et de diffusion. Le dépôt Git a servi non seulement à centraliser le code source, mais aussi à documenter notre progression et garantir la traçabilité des modifications.

10.1. Structure du dépôt

Le dépôt GitHub contient l'intégralité des fichiers du projet, organisés de manière claire et logique :

bash

Copier le code

/assets/

├── css/
│ └── style.css

├── js/
│ └── script.js

└── images/
 └── [images produits, logos, icônes...]

/data/

└── produits.json

index.html

about.html

collection.html

contact.html

README.md

LICENSE

.gitignore

Cette organisation permet :

- Une séparation claire des responsabilités (styles, scripts, données)
- Une facilité de navigation pour toute personne consultant le code
- Une compatibilité avec le déploiement via GitHub Pages

10.2. README du projet

Le fichier README.md a été soigneusement rédigé pour expliquer :

- Le **but du projet**
- Les **technologies utilisées**
- Les **instructions d'installation et d'exécution locale**
- Une **présentation visuelle** avec captures d'écran
- Les **auteurs** du projet

Il constitue le point d'entrée pour tout nouvel utilisateur ou examinateur souhaitant comprendre rapidement le projet sans ouvrir chaque fichier.

10.3. Versionnement et historique

Chaque membre a contribué via sa propre branche, et les fusions dans main ont été réalisées via des **pull requests** avec revue de code. Les commits sont :

- Fréquents (plusieurs par semaine)
- Clairs (feat: ajout du formulaire, fix: bug d'affichage mobile, etc.)
- Documentés (liés à des tâches Trello ou des jalons)

Cette approche a permis un **suivi précis de l'évolution** du projet.

10.4. Fichiers techniques inclus

Le dépôt comprend également :

- Un fichier .gitignore (pour exclure les fichiers inutiles comme les .DS_Store, les caches éditeur, etc.)
- Un fichier LICENSE en licence open source (ex. MIT), autorisant la réutilisation libre du projet

11. Tests et validation

Une fois le développement du site finalisé, nous avons procédé à une phase complète de tests afin de valider sa stabilité, sa compatibilité, sa performance et son accessibilité. L'objectif était de garantir une expérience utilisateur fluide et sans erreur, quelle que soit la configuration du terminal ou du navigateur utilisé.

11.1. Validation HTML/CSS (W3C)

Tous les fichiers HTML et CSS ont été soumis aux **validateurs officiels du W3C** :

- <https://validator.w3.org/> pour le HTML
- <https://jigsaw.w3.org/css-validator/> pour le CSS

Résultats :

- Aucune **erreur bloquante**
- Quelques **avertissements mineurs** concernant des balises auto-fermées, corrigés rapidement
- Respect global des **standards HTML5 et CSS3**

Cela garantit la **portabilité et la compatibilité** du code sur les navigateurs modernes.

11.2. Tests de navigation et d'interactivité

Nous avons effectué des tests manuels sur les différentes fonctionnalités interactives du site :

- **Navigation entre les pages** : liens fonctionnels, menus actifs, retour accueil
- **Effets dynamiques** : carrousel, animations CSS, affichage des fiches produits
- **Formulaire de contact** : validation dynamique, messages d'erreur, champs requis
- **Chargement JSON** : intégration correcte des produits, gestion des erreurs de chargement

Chaque fonctionnalité a été testée dans des cas normaux et en conditions limites (ex. champ vide, clics rapides).

11.3. Compatibilité navigateur

Le site a été testé sur les principaux navigateurs :

Navigateur	Résultat
Google Chrome	✓ Comportement attendu
Firefox	✓ Comportement attendu
Microsoft Edge	✓ Comportement attendu
Safari	✓ Rendu correct

Aucun problème de rendu ou d'interaction n'a été relevé.

11.4. Tests de responsive design

Le site a été vérifié sur différentes tailles d'écran (desktop, tablette, mobile) grâce aux outils de développement intégrés aux navigateurs (mode responsive) :

- Réorganisation fluide des blocs
- Redimensionnement automatique des images
- Passage du menu en **version mobile (burger menu)**
- Lisibilité conservée sur tous les formats

11.5. Audit avec Lighthouse

L'outil **Google Lighthouse** a été utilisé pour auditer le site sur plusieurs critères :

Critère	Score moyen obtenu
Performance	93 / 100
Accessibilité	96 / 100
Bonnes pratiques	95 / 100
SEO	91 / 100

Les recommandations mineures (ex : tailles d'images, balises meta) ont été prises en compte dans la version finale.

12. Limites et pistes d'amélioration

Malgré le respect du cahier des charges initial et la réalisation complète des fonctionnalités attendues, nous avons identifié certaines limites dans notre projet. Ces limites sont dues à la fois aux contraintes techniques imposées, au temps disponible, et à notre niveau d'expertise actuel. Toutefois, elles constituent autant de pistes d'amélioration pertinentes pour faire évoluer le site dans une version future plus ambitieuse.

13.2. Simplicité du formulaire de contact

Le formulaire fonctionne uniquement avec une validation **côté client**, et ne permet pas d'envoi réel des messages.

Piste d'amélioration :

- Ajouter un **traitement côté serveur** pour permettre l'envoi de mails via un service comme Formspree, EmailJS ou un serveur Node.js personnalisé.

13.3. Design encore perfectible

Bien que sobre et cohérent, le design peut paraître un peu statique sur certaines sections. Certaines pages manquent de dynamisme ou de micro-interactions poussées.

Piste d'amélioration :

- Enrichir l'expérience utilisateur avec **plus d'animations CSS** ou JavaScript.
- Introduire des **effets au scroll**, des transitions entre sections ou des **animations conditionnelles**.

13.4. Accessibilité étendue

Le site respecte les principes de base de l'accessibilité, mais pourrait aller plus loin :

- Ajout de **navigation ARIA**
- Compatibilité avec **lecteurs d'écran**
- Modes d'affichage alternatifs (contraste élevé, mode sombre)

13.5. Optimisation et performance

Le temps de chargement est raisonnable, mais :

- Les **images pourraient être compressées davantage**
- Le code CSS pourrait être **minifié**
- Des **lazy loading** pourraient être mis en place pour améliorer la rapidité sur mobile

13. Conclusion

Ce projet de développement d'un site web en HTML, CSS et JavaScript a représenté bien plus qu'un simple exercice académique : il a été une expérience complète, enrichissante et professionnalisante. Du cadrage initial à la mise en ligne finale, en passant par les phases d'analyse, de conception, de développement et de tests, nous avons pu mettre en pratique l'ensemble des compétences acquises au cours de notre formation.

Un apprentissage technique structurant

Sur le plan technique, le projet nous a permis de renforcer notre maîtrise des technologies front-end. Nous avons appris à :

- Structurer un site de manière sémantique
- Créer des interfaces responsives et accessibles
- Intégrer des interactions dynamiques en JavaScript pur
- Organiser et versionner notre code de façon professionnelle

Nous avons également pris conscience de l'importance des détails : le choix des couleurs, la lisibilité, l'ergonomie mobile, ou encore la validation des formulaires sont autant d'éléments qui transforment une interface fonctionnelle en une expérience utilisateur réussie.

Une démarche collaborative réussie

Au-delà de la technique, ce projet a renforcé notre capacité à **travailler en équipe**. La répartition claire des tâches, la communication régulière et l'utilisation d'outils collaboratifs comme GitHub ou Trello ont été des facteurs clés de réussite. Nous avons également appris à nous adapter face aux imprévus, à planifier notre temps et à nous organiser de manière autonome.

Une base solide pour de futurs projets

Ce projet constitue une **base de départ** pour des projets web plus ambitieux. Nous avons désormais une meilleure compréhension des mécanismes du développement moderne, et des pistes claires pour améliorer notre site, le faire évoluer ou le déployer dans un contexte professionnel (e-commerce, portfolio, site vitrine...).

14. Annexes

Cette section regroupe l'ensemble des éléments annexes ayant contribué à la réalisation du projet ou permettant d'en approfondir certains aspects. Elle contient des captures d'écran, des extraits de code représentatifs, des liens utiles et des documents de référence.

15.1. Extraits de code commentés

✂ *Effet de typing JS :*

```
javascript
Copier le code
function typeText(text, element) {
  let i = 0;
  function type() {
    if (i < text.length) {
      element.innerHTML += text.charAt(i);
      i++;
      setTimeout(type, 80);
    }
  }
  type();
}
```

✂ *Chargement JSON :*

```
javascript
Copier le code
fetch("data/produits.json")
  .then(response => response.json())
  .then(data => {
    data.forEach(produit => {
      const card = document.createElement("div");
      card.innerHTML = `

### ${produit.nom}</h3><p>${produit.prix}</p>`; document.querySelector("#collection").appendChild(card); }); });





```

15.2. Outils utilisés

- **Visual Studio Code** : éditeur de code
- **Figma** : création de la maquette visuelle

- **Git / GitHub** : versionnement et collaboration
- **W3C Validators** : validation HTML/CSS
- **Google Lighthouse** : audit de performance
- **Trello** : gestion de projet
- **GIMP / Canva** : design graphique léger

15.3. Liens utiles

-  Dépôt GitHub : <https://github.com/<utilisateur>/<projet>>
-  Démo en ligne (GitHub Pages, si déployé)
-  Documentation officielle :
 - [MDN Web Docs](#)
 - [HTML Validator W3C](#)
 - [CSS Validator](#)
 - Lighthouse

15.4. Fichier JSON exemple utilisé

json

Copier le code

```
[
  {
    "nom": "T-shirt oversize",
    "prix": "29.90€",
    "image": "images/tshirt.jpg",
    "description": "Coton bio, coupe large, col rond."
  },
  {
    "nom": "Pantalon large",
    "prix": "49.90€",
    "image": "images/pantalon.jpg",
    "description": "Tissu fluide, coupe droite, taille haute."
  }
]
```