

# ***Cahier des charges***

[enzo-dev.com](https://enzo-dev.com)

## *Base du projet*

**Next.js** (framework React) : utilisé pour le SEO optimisé notamment via le SSR (Server-Side Rendering) ainsi que la simplicité d'utilisation des Server Actions

---

## *Hébergement du site et des données*

**Vercel** (plateforme d'hébergement) : sera utilisé pour l'hébergement grâce à ses nombreux avantages gratuits, maintenu par l'équipe de Next.js garantissant une intégration parfaite, déploiement automatique à chaque commit.

**Neon** (base de données en ligne) : sera utilisée pour stocker les données de l'application. Vercel étant une plateforme serverless, nécessite une base de données en ligne pour la persistance des données.

**Prisma** (ORM) : facilitera les interactions avec la base de données.

---

## *Ecosystème*

**ShadCN** (ensemble de composants) : utilisé pour un design cohérent et rapide. Contrairement aux bibliothèques classiques, ShadCN ne fournit pas de composants précompilés mais des composants copiables et modifiables directement dans le code, offrant une grande flexibilité.

**Tailwind CSS** (framework CSS) : utilisé pour le design frontend, offrant rapidité et intégration directe avec le JSX. Des classes spécifiques pourront être ajoutées pour des besoins plus techniques.

**TypeScript** (surcouche de JavaScript) : ajoute un système de typage statique permettant de détecter les erreurs avant l'exécution et d'améliorer la maintenabilité du code.

**Zod** (bibliothèque TypeScript) : utilisé pour valider dynamiquement les données au runtime. Permet de sécuriser les formulaires et de garantir que les données récupérées depuis la base de données respectent la bonne structure.

**TanStack Query** (bibliothèque JavaScript) : simplifie le data fetching, évite la gestion manuelle des états (loading, error) et optimise la mise en cache des données.

**Sonner** (bibliothèque React) : bibliothèque légère permettant l'affichage de notifications toast dans React et Next.js.

**next-safe-action** (bibliothèque Next.js) : permet de sécuriser les paramètres des Server Actions Next.js (via les schema Zod) et une gestion optimisée des erreurs.

**Date-fns** (bibliothèque JavaScript) : utilisé pour manipuler, formater, et calculer des dates simplement.

**Better Auth** (bibliothèque Next.js) : facilite la mise en place d'une authentification sécurisée et une intégration directe avec Next.js.

**Vitest** (framework JavaScript) : utilisé pour les tests unitaires et d'intégration, offre une syntaxe simple et rapide idéale pour un Portfolio.

**Resend** (service) : est un service conçu pour l'envoi d'emails depuis des applications web (réinitialisation de mot de passe, formulaire de contact, vérification d'email...).

---

## Composition du site

Une première page dédiée à l'authentification permet aux utilisateurs de se connecter, de s'inscrire et de récupérer leur mot de passe en cas d'oubli. Ce système intègre également une gestion des sessions qui évite aux utilisateurs de devoir se reconnecter en permanence. À chaque arrivée sur le site, la validité de la session est vérifiée et, si elle a expiré, l'utilisateur est automatiquement redirigé vers la page de connexion. Cette authentification témoigne à la fois de ma capacité à mettre en place un système sécurisé complet et de récupérer des informations sur l'utilisateur, utilisées dans les parties suivantes du site.

Une barre de navigation permet de changer rapidement de section et de savoir où l'on se situe. C'est également ici que se trouve le bouton de déconnexion. Sur les grands écrans, la barre de navigation s'affiche horizontalement de façon classique, tandis que sur les petits écrans, elle se transforme en un header épuré intégrant un bouton. Ce bouton ouvre alors un menu latéral vertical (sidebar) qui glisse depuis le côté, offrant une navigation adaptée pour les smartphones par exemple.

Une section "Home" destinée à rapidement avoir les informations clés, à savoir : mon identité, de quel type de site il s'agit ainsi que mes réseaux sociaux.

Puis la section "Projets" qui permet de créer de l'interaction avec l'utilisateur via un système de défilement de mes projets qui dévoile dans le même temps les technologies qui y sont associées.

Suivie de "Timeline" qui, dans la continuité de la partie précédente, dévoile petit à petit mon parcours sur les dernières années via une frise chronologique, se basant également sur l'interaction de l'utilisateur afin de conserver son attention.

Ces deux systèmes "Project" et "Timeline" permettent entre autres de mettre en avant ma maîtrise des mises à jour dynamiques du DOM.

En bas de page, on retrouvera les sections "Contact" et "Commentaires", qui permettront respectivement de me contacter et de laisser un avis sur mon travail. Au-delà de leur fonction première, ces fonctionnalités démontrent mes compétences en développement back-end.

En effet, elles illustrent ma capacité à gérer l'envoi, le traitement et la sauvegarde sécurisée des données, tout en assurant une gestion efficace des erreurs potentielles. De plus, en sens inverse, elles prouvent que je maîtrise la récupération asynchrone des données tout en garantissant une expérience fluide et performante pour les utilisateurs.

---

## Back-end

L'envoi d'e-mails ainsi que l'envoi et la récupération des commentaires sont gérés côté serveur grâce aux Server Actions de Next.js, garantissant un traitement sécurisé et performant grâce aux technologies mentionnées dans la section "Écosystème". Chaque action côté serveur inclut également une vérification de la validité de la session, renforçant ainsi la sécurité et la confidentialité des utilisateurs.

Pour ce qui est des commentaires, chacun d'eux est directement relié à l'utilisateur qui l'a rédigé dans la base de données. Cela permet d'identifier clairement l'auteur de chaque commentaire et de garder une traçabilité des retours.

Concernant l'envoi d'e-mails, celui-ci se base sur l'adresse email choisie à l'inscription. Cependant, pour garantir la sécurité, j'utilise Resend, un service qui nécessite l'utilisation d'une adresse mail vérifiée. Ainsi, l'email est envoyé depuis ma boîte mail [contact@enzo-dev.com](mailto:contact@enzo-dev.com) vers cette même adresse, mais je précise l'adresse mail de l'utilisateur dans le champ "reply-to" ainsi que dans le corps du message. Cela permet, au moment où je souhaite répondre, que l'email de l'utilisateur soit automatiquement sélectionné comme destinataire, tout en assurant que l'envoi initial respecte les contraintes de sécurité et de vérification imposées par Resend.

Ainsi, ce portfolio intègre pleinement un front-end et un back-end, avec une architecture sécurisée et optimisée, démontrant ma maîtrise complète du développement d'applications web, aussi bien dans la gestion des données et de la sécurité que dans l'expérience utilisateur.

---

## Design

Le design du portfolio est moderne, avec une esthétique cyberpunk et futuriste. Voici quelques éléments qui contribuent à cette atmosphère :

Palette de couleurs sombres et néon : L'arrière-plan grisé associé aux textes en néon bleu donne un effet high-tech, rappelant les interfaces de science-fiction, accentuant le côté cyber et tech.

Typographie digitale : la police de caractères utilisée, JetBrains Mono, évoque un terminal informatique.

Disposition épurée et intuitive : L'agencement des éléments est minimaliste, ce qui permet une navigation claire et agréable.

---

## Accessibilité

Les formulaires exploitent Radix UI, ce qui implique une gestion automatique des attributs ARIA, renforçant ainsi l'accessibilité et permettant aux utilisateurs utilisant des technologies d'assistance de naviguer et interagir aisément avec l'interface.

Les notifications toast, intégrées avec Sonner, offrent un retour visuel discret mais efficace, améliorant la perception générale de réactivité de l'application.

Le système de navigation dynamique, met automatiquement en évidence la section active à mesure que l'utilisateur défile la page enrichit considérablement la clarté et l'orientation utilisateur.

Le design responsive garantit une expérience utilisateur homogène et efficace sur tous les supports, qu'ils soient mobiles, tablettes ou ordinateurs.

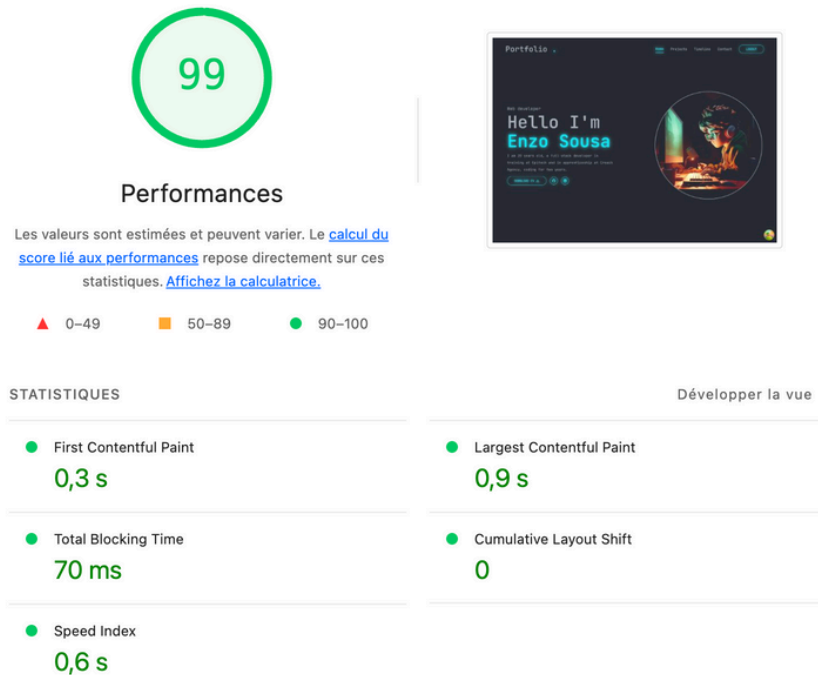
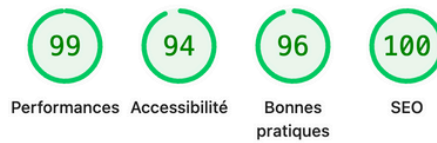
L'esthétique visuelle du site se distingue d'abord avec des contrastes de couleurs mettant aisément en avant les éléments importants. Ensuite, l'espacement et l'organisation épurée des éléments renforcent cette lisibilité, offrant une hiérarchie visuelle linéaire intuitive qui guide naturellement l'utilisateur. Et enfin par une linéarité offerte par le format "one page" qui garantit que l'utilisateur ne manquera aucune informations.

---

## SEO

Le code révèle une bonne structuration sémantique grâce à l'utilisation pertinente des balises, une hiérarchie claire des titres et des métadonnées bien choisies.

De plus, l'utilisation de Next.js apporte un avantage significatif en matière de SEO (comme indiqué dans "Base du projet") grâce à son rendu côté serveur (SSR), permettant un meilleur référencement par les moteurs de recherche en facilitant l'indexation du contenu.



## Monitoring

Pour le monitoring technique de mon portfolio, j'utilise plusieurs outils complémentaires qui me permettent de suivre l'état, la performance et l'utilisation de mon site.

Tout d'abord, j'ai accès aux rapports fournis par Neon. La plateforme propose des tableaux de bord détaillés sur l'utilisation de la base de données, comme le nombre de requêtes effectuées, la consommation de ressources... Ces informations sont utiles pour identifier rapidement un problème de requête mal optimisée ou un pic d'utilisation.

Ensuite, j'utilise les logs serveur de Vercel. Ces logs permettent de consulter en temps réel toutes les requêtes entrantes et leurs réponses, d'identifier d'éventuelles erreurs côté serveur ou encore de mesurer les temps de réponse.

Je complète cela avec les analytics Vercel, qui fournissent des données sur le trafic du site. Grâce à ces analytics, je peux connaître le nombre de visiteurs ainsi que leurs caractéristiques (zone géographique, type d'appareil, etc.).

En ce qui concerne le monitoring des avis utilisateurs, il passe directement par mon portfolio avec la possibilité de m'envoyer un email ou de laisser un commentaire.

---

## Améliorations prévues

Certaines améliorations sont envisageables pour mon portfolio à l'avenir. Par exemple, actuellement, les tests sont exécutés sur la même base de données que celle utilisée par les utilisateurs, dans laquelle je crée des données factices pour ces tests que je supprime ensuite automatiquement. Toutefois, il serait préférable de disposer d'une base de données dédiée. Cela permettrait d'isoler complètement les données de tests des données réelles, évitant ainsi tout risque d'altération ou de perte accidentelle d'informations utilisateur. De plus, un environnement séparé garantit des tests plus fiables et reproductibles, sans impact sur la production ni sur l'expérience des utilisateurs.

Côté performance cette fois-ci, une amélioration possible serait d'utiliser le partial pre-rendering (PPR) de Next.js, qui permet de charger certaines parties de l'application de manière asynchrone tout en affichant immédiatement les autres sections ne nécessitant pas de chargement. Par exemple, dans mon cas, au lieu d'attendre la récupération des commentaires des utilisateurs avant d'afficher le portfolio, je pourrais afficher directement le contenu principal du portfolio pendant que les commentaires se chargent en arrière-plan et s'affichent dès qu'ils sont disponibles. Cela améliore la rapidité perçue et l'expérience utilisateur en réduisant le temps d'attente initial.