

::GUIDE TECHNIQUE::

Portafolio Project

Sommaire::

- 1:: Introduction
- 2:: Objectifs du projet
- 3:: Architecture de la solution
- 4:: Fonctionnalités principales
- 5:: Technologies utilisées
- 6:: Design et ergonomie
- 7:: Contraintes et obligations
- 8:: Tests et validation
- 9:: Plan de déploiement
- 10:: Évolution future
- 11:: Conclusion

1. Introduction

Cette documentation technique et fonctionnelle a pour objectif de fournir une vue d'ensemble complète du projet de développement du portfolio . Elle s'adresse aux équipes parties prenantes (développeurs, responsables de projet, administrateurs système, etc.) afin de garantir la pérennité et l'évolution future de la solution web.

2. Objectifs du projet

Objectif principal : Créer un portfolio en ligne pour mettre en valeur les compétences techniques et les réalisations en développement web.

Public cible : Recruteurs et clients potentiels.

3. Architecture de la solution

3.1. Description générale de l'architecture

Le projet repose sur une architecture client-serveur avec un front-end en React et des fichiers statiques (HTML, SCSS) pour le style. Il est relié à un back-end simple permettant l'envoi de mail via le formulaire de contact.

3.2. Structure des fichiers

Le projet est organisé comme suit :

/src

 /components # Composants React

 /assets # Images, icônes, fichiers de style (SCSS)

 /styles # Fichiers SCSS pour le design

 /pages # Pages principales : Home, Projects, About Me

/public

 index.html # Point d'entrée de l'application

 assets # Ressources statiques (images, fonts, etc.)

/build # Dossier de build pour le déploiement

4. Fonctionnalités principales

4.1. Home

Affichage de la phrase d'accroche avec une photo professionnelle.

Formulaire de contact pour faciliter l'échange avec les recruteurs/clients.

4.2. Projects

Affichage des projets réalisés avec des détails sur les technologies utilisées.

Option de téléchargement du CV au format PDF.

4.3. About Me

Présentation de l'historique professionnel et personnel.

Timeline interactive pour visualiser l'évolution de la carrière.

4.4. Responsivité

Le site doit être responsive, avec une navigation fluide sur tous les types d'appareils : mobile, tablette, et desktop.

5. Technologies utilisées

5.1. Frontend

React.js : Framework JavaScript utilisé pour la création de composants et la gestion de l'état de l'application.

SCSS : Préprocesseur CSS pour un style modulable et réutilisable.

HTML : Structure du contenu web.

5.2. Outils de développement

Visual Studio Code : Environnement de développement intégré (IDE) pour le développement du projet.

Git/GitHub : Gestion de version et collaboration pour le code source.

5.3. Hébergement

Netlify ou Vercel : Plateformes d'hébergement pour déployer l'application en production.

Domaine personnalisé : Option de connecter un domaine personnalisé pour le site une fois déployé.

6. Design et ergonomie

6.1. Charte graphique

Couleurs : Fond sombre avec des accents de couleurs vives pour les éléments interactifs.

Fond : #242424

Texte : rgba(211, 211, 211, 0.786)

Boutons : #646cff (violet), #f8b300 (jaune)

Typographie : Utilisation de la police 'Fira Code' pour un rendu moderne et lisible.

6.2. Navigation

Menu de navigation simplifié avec des liens vers les sections principales : Home, Projects, About Me.

Structure fluide et facile à comprendre pour une expérience utilisateur optimale.

7. Contraintes et obligations

7.1. Contraintes légales

Aucun traitement de données personnelles, donc aucune exigence RGPD spécifique.

7.2. Contraintes techniques

Compatibilité navigateurs : Le site doit être compatible avec les navigateurs modernes : Chrome, Firefox, Safari, Edge.

Performance : Optimisation des images et des ressources pour un chargement rapide.

8. Tests et validation

8.1. Tests unitaires

Des tests unitaires seront effectués pour vérifier le bon fonctionnement des composants React.

8.2. Tests de performance

Utilisation de Google Lighthouse pour tester les performances et l'accessibilité du site.

8.3. Tests de compatibilité

Vérification de l'affichage et de la fonctionnalité du site sur différents navigateurs et dispositifs (mobile, tablette, desktop).

9. Plan de déploiement

9.1. Déploiement initial

Déploiement sur Vercel pour la mise en ligne rapide et gratuite.

9.2. Maintenance et mise à jour

Le site étant statique, les mises à jour seront effectuées manuellement en modifiant les fichiers sources.

Les changements seront gérés via Git et déployés automatiquement sur les plateformes d'hébergement.

10. Évolution future

Ajouter des fonctionnalités comme un blog ou un formulaire d'inscription à une newsletter.

Ajouter des sections supplémentaires pour détailler davantage les compétences et projets réalisés.

Intégrer une base de données pour la gestion d'un blog ou d'un portfolio dynamique, si nécessaire à l'avenir.

11. Conclusion

Cette documentation fournit une vue complète du projet de portfolio personnel, des technologies utilisées, de la structure du site et des processus de maintenance. Elle permet aux parties prenantes de comprendre comment le projet est conçu, comment il fonctionne et comment il peut être évolutif à l'avenir.