

Spécifications Techniques

Portfolio Alix Hennion

Projet 12	Nom du Projet: Portfolio Alix Hennion
-----------	---------------------------------------

Version:	Auteur:	Date:	Nombre collaborateur
1.0	Hennion Alix	18/12/2023	0

Sommaire

I. Choix Technologiques (page 1)

II. Liens avec le Back-end (page 2)

III. Préconisations Domaine (page 2)

IV. Accessibilité (page 2)

V. Services Tiers (page 3)

VI. Maintenance et Mises à Jour (page 4)

VII. Style esthétique (page 5)

I. Choix technologiques

- Choix de React.js pour le développement Frontend :

1. Facilité de développement:

- React.js offre une conception modulaire basée sur des composants réutilisables, simplifiant la création d'interfaces interactives.
- La modularité favorise la collaboration et accélère le développement de fonctionnalités.

2. Performances optimales:

- La stratégie de rendu virtuel et de mise à jour sélective minimise les mises à jour coûteuses du DOM, assurant une expérience utilisateur fluide.

3. Maintenance simplifiée:

- La conception basée sur des composants autonomes facilite la maintenance, les tests unitaires, et les modifications futures.

4. Communauté étendue:

- React.js bénéficie d'une communauté active, offrant des ressources, didacticiels, et composants open source.

5. Modernité et succès avéré:

- Utilisé par des géants comme Facebook, Instagram, et WhatsApp, témoignant de sa pertinence et de sa modernité.

6. Avantage compétitif:

- Renforce la crédibilité du projet en tant que solution moderne, innovante, et adoptée par les leaders du secteur.

II. Liens avec le Back-end

- Actuellement, aucun lien avec un backend n'est nécessaire, car le site est statique. Toutefois, des fonctionnalités futures peuvent nécessiter des intégrations back-end.

III. Préconisations concernant le domaine

- Nom du domaine:Portfolio-Alix-Hennion
- Adresses e-mail: contact@portfolio-alix-hennion.com

IV. Accessibilité

- Compatibilité navigateur: Chrome, Safari, Firefox
- Types d'appareils: Principalement pour les appareils de bureau
- Taille adapté en responsive design pour le desktop first vers le mobile en passant par la tablette garantissant une expérience optimale sur des écrans de grande taille.

V. Services tiers

- Principaux services tiers:

1. Weather APP:

Intégration d'une application météo pour fournir des informations en temps réel sur la localisation de l'utilisateur.

2. Font Awesome:

Intégration d'icônes modernes pour améliorer l'interface utilisateur.

3. Formik:

Intégration d'un formulaire facilitée grâce à Formik.

4.Axios:

Intégration de la bibliothèque Axios pour faciliter les requêtes HTTP et les interactions avec le backend. Cela permettra une gestion plus efficace des appels API et des données.

5. i18next:

Mise en œuvre de i18next pour la gestion de la localisation et de la traduction. Cela permettra de rendre le site accessible à un public plus large en prenant en charge plusieurs langues.

6. React-calendar-timeline:

Utilisation de la bibliothèque React-calendar-timeline pour créer et afficher des timelines interactives sur le site. Cela pourrait être particulièrement utile pour afficher des événements ou des projets dans un format chronologique.

7. Redux:

Intégration du gestionnaire d'état Redux pour faciliter la gestion de l'état global de l'application. Cela améliorera la gestion des données et des interactions entre les composants, garantissant une expérience utilisateur cohérente.

8. Yup:

Utilisation de la bibliothèque Yup pour la validation des formulaires. Cette intégration simplifiera la validation des données côté client, garantissant que les informations saisies sont conformes aux exigences spécifiées.

VI. Maintenance du site et futures mises à jour

● Maintenance :

Pour assurer la qualité et la stabilité du code, la plupart des composants et fonctionnalités de ce projet sont accompagnés de fichiers de test. Cette approche systématique de l'utilisation de tests unitaires offre un ensemble d'avantages essentiels. En premier lieu, elle joue un rôle crucial dans la prévention et la détection précoce des erreurs, contribuant ainsi à la création d'un code robuste et fiable. De plus, l'intégration de tests facilite la maintenance du système en permettant une identification rapide des problèmes potentiels lors de modifications futures. En réduisant les risques d'anomalies, cette pratique renforce la qualité globale du projet et garantit une expérience utilisateur optimale.

● Sécurité renforcée:

- Mises à jour régulières de tous les composants logiciels, y compris les dépendances tierces, pour remédier aux vulnérabilités de sécurité connues. Cela garantira un environnement sécurisé et à jour.

● Optimisation des performances:

- Surveillance constante des performances du site à l'aide d'outils spécialisés. Ajustement des éléments identifiés comme des points de friction pour maintenir une expérience utilisateur rapide et réactive.

Planification des mises à jour futures:

- Ajout de nouvelles fonctionnalités en fonction des retours utilisateurs et des tendances du secteur. Cela peut inclure l'intégration de nouvelles technologies ou l'amélioration des fonctionnalités existantes.

- Évolution continue du design pour suivre les tendances esthétiques actuelles et maintenir un aspect visuel attrayant.

- Assurer la compatibilité avec les dernières technologies, navigateurs et appareils pour garantir une expérience utilisateur sans heurts.

- Utilisation des retours utilisateurs pour des améliorations continues et une adaptation aux besoins changeants de l'utilisateur.

● Gestion des mises à jour:

- Via une planification structurée, des sauvegardes régulières, une communication transparente et un plan de retour en arrière en cas de problèmes majeurs. Le site sera versionné sur GitHub avec des commits réguliers pour suivre les modifications apportées.

VII Style esthétique

- **Choix de Sass pour le style**

1. Styles et Esthétique :

Sass est utilisé pour garantir une esthétique cohérente et attrayante dans l'ensemble du projet.

2. Avantages de Sass :

Modularité :

Sass permet une écriture de feuilles de style modulaire, facilitant la gestion de composants distincts du projet.

Variables :

L'utilisation de variables simplifie la maintenance en permettant la définition et la modification facile de valeurs réutilisables.

Imports :

La fonction d'importation de Sass améliore la structuration des fichiers de style et facilite la gestion de la complexité.

3. Productivité du Développement :

Sass contribue à une expérience de développement plus efficace en permettant des modifications rapides et systématiques de l'apparence du site.

4. Maintenance Simplifiée :

La syntaxe avancée de Sass facilite la gestion de la complexité des feuilles de style, offrant une organisation claire et structurée.