**Spécifications techniques**

Menu Maker By Qwenta

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Auteur** | **Date** | **Approbation** |
| 1.0 | Tony, Webgancia | 21/10/2024 | Soufiane, Webgancia |

**SOMMAIRE**

I. Choix technologiques

II. Liens avec le back-end

III. Préconisations concernant le domaine et l’hébergement

IV. Accessibilité

V. Autres spécifications

VI. Recommandations en termes de sécurité

VII. Maintenance du site et futures mises à jour

1. **Choix technologiques**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Besoin** | **Contraintes** | **Solution** | **Description de la solution** | **Justification (2 arguments)** |
| **Landing page non connectée** | L’utilisateur doit pouvoir comprendre le principe de cette application dès le premier abord. | -SASS  -HTML | Deux langages simple d’utilisation pour une single page. | 1) Ce sont deux langages primordial pour la conception d’un site web.  2) Simplicité et réactivité lors de l’ouverte de la page. |
| **Page login** | La connexion doit être rapide et sécurisée. | -React modal  -Express.js, Passport.js | Cette bibliothèque React permet de créer simplement des modales. Passport.js est un middleware d'authentification pour Node.js. | 1) Il s'agit de la librairie la plus utilisée.  2) Passport.js supporte de nombreux types d'authentification (email/password, OAuth, etc.) . |
| **Dashboard** | Une fois connecté l’utilisateur doit pouvoir se diriger vers la section qu’il souhaite. | -React  -React Router | Grâce à cette librairie nous pourrons naviguer entre chaque section de la page et ensuite créer des composants réutilisables. | 1) Grâce au router nous allons pouvoir naviguer entre chaque section sans rechargement de la page.  2) Chaque section peut être encapsulé dans un composant pour que le code soit propre et maintenable. |
| **Création d'une catégorie de menu** | L'ajout d'une catégorie doit pouvoir se faire directement sur l'écran de création de menu depuis une modale. | -React modal | Cette librairie React permet de créer simplement des modales performantes, accessibles avec un minimum de code. | 1) Nous avons choisi de développer en React, la librairie est cohérente avec ce choix.  2) Il s'agit de la librairie la plus utilisée. |
| **Création du menu** | L’ajout d’élément nécessaire à la création d’un menu doit pouvoir se faire sur l'écran de création de menu depuis une autre modale que celle de la catégorie. | -React, Redux -React modal | Cette bibliothèque va nous permettre de mieux gérer l’état de nos composants afin de les réactualiser. | 1)Redux est l’une des bibliothèques les plus utilisés avec React.  2) Redux facilite la collaboration, le débogage, et l'évolutivité des fonctionnalités. |
| **Personnaliser un menu** | Le changement de typographie, de couleur et l’importation d’un logo doit pouvoir se faire directement depuis l’écran de création de menu. | -Styled Components  -React Dropzone | Styled Components va nous permettre de styliser les composants React.  React Dropzone permet d’implémenter facilement une fonctionnalité de drag-and-drop pour le téléchargement de fichiers. | 1) Grâce à Styled Components, nous allons facilement styliser nos composants de façon isolée.  2) L’atout principale de cette librairie est sa simplicité d’intégration. Grâce à son API intuitive, nous pouvons rapidement ajouter des fonctionnalités de téléchargement de fichiers à leur application. |
| **Visualiser le menu actuellement créé** | L’utilisateur doit pouvoir visualiser les changements de son menu dynamiquement. | -MongoDB | MongoDB permet de stocker des données flexibles, que nous pouvons ensuite utiliser pour afficher le menu avec les informations mises à jour. | 1) MongoDB est idéale pour des applications évolutives grâce à ses données flexibles sous forme de documents JSON.  2) Cette base de données est connue pour ses performances rapides grâce à ses mises à jour fréquentes. |
| **Exporter le menu en PDF** | L’utilisateur doit pouvoir en un clic télécharger son menu en PDF. | -React-PDF | React-PDF est une librairie qui permet de créer, gérer et télécharger des documents PDF. | 1) Cette librairie s’intègre naturellement avec React, ce qui simplifie la création de documents PDF interactifs en utilisant des composants React familiers.  2) Elle permet de personnaliser l'affichage des PDF, ce qui permet de mieux gérer la présentation et la fonctionnalité du document. |
| **Exporter le menu vers Deliveroo** | L’utilisateur doit pouvoir en un clic exporter son menu vers Deliveroo. | -API Menu de Deliveroo | L’API permet la connexion des deux plateformes, et la transmission des données du menu à Deliveroo. | 1) L’API Deliveroo permet d'automatiser la gestion des menus, réduisant ainsi le temps et les efforts nécessaires pour mettre à jour les informations sur les plats.  2) Elle offre une visibilité maximale en exportant vos  menus vers une plateforme avec une large  audience. |
| **Partager le menu vers Instagram** | L’utilisateur doit pouvoir partager facilement son menu sur instagram. | -Instagram Graph API | L’API permet la connexion des deux plateformes, et la transmission des données du menu à Instagram. | 1) L’API Instagram permet d'automatiser la gestion des menus, réduisant ainsi le temps et les efforts nécessaires pour mettre à jour les informations sur les plats.  2) Elle offre une visibilité maximale en exportant vos  menus vers une plateforme avec une large  audience. |
| **Imprimer un menu** | L’utilisateur doit pouvoir commander en un clic l’impression d’un menu dans un nouvel onglet en ayant un visuel de l’encart depuis la page d’accueil. | -Lien vers le back-office de Qwenta | Ce lien intitulé « Imprimer un menu » sera présent sur la page d’accueil qui redirigera l’utilisateur vers le back-office de Qwenta. | 1) En ajoutant le lien sur la page d’accueil, le processus d’impression du menu sera plus intuitif et facile d’accès pour l’utilisateur.  2) L’ouverture du lien dans un nouvel onglet permet à l’utilisateur de conserver l’interface de gestion de son menu intacte. |
| **Avoir accès à une vue regroupant tous les menus créés précédemment** | L’utilisateur doit, en cliquant sur « Mes menus » avoir accès à une vue regroupant mes menus créés précédemment.  Sur la même vue, il doit pouvoir créer un nouveau menu. | -Mongoose | Mongoose rendra l’application plus robuste et fonctionnelle. Cela permet de stocker, de récupérer et de gérer facilement les données de manière dynamique. | 1) Cette librairie fournit une interface simple et intuitive pour interagir avec MongoDB. Les méthodes de Mongoose sont faciles à utiliser, ce qui simplifie les opérations CRUD.  2) Mongoose est bien intégré avec d'autres bibliothèques et frameworks Node.js, ce qui facilite son utilisation dans des projets variés. |
| **Avoir accès à une modale où sont indiquées les informations légales** | En cliquant sur « Mentions légales », une modale doit pouvoir s’ouvrir et afficher la mention « Tous droits réservés ». | -React-modal | Un lien « Mentions légales » sera présent sur les pages connectées et déconnectées. Au clic sur ce lien, une modale mise en place avec React-Modal s’ouvrira pour afficher les informations légales.  La mention « Tous droits réservés » sera présente en bas de toutes les pages du site. | 1) L'utilisation de React-Modal pour afficher les mentions légales offres une expérience utilisateur sans interruption, sans rediriger l'utilisateur vers une nouvelle page.  2) En affichant la mention « Tous droits réservés » en bas de toutes les pages, la conformité légale du site est assurée, ce qui est essentiel pour protéger les droits de propriété intellectuelle. |

1. **Liens avec le back-end**

* Quel langage pour le serveur ?

JavaScript sera utilisé comme langage principal, à la fois pour le Frontend avec React et pour le Backend avec Node.js, accompagné du framework Express.js, très populaire pour le développement d'applications serveur.

* A-t-on besoin d’une API ? Si oui laquelle ?

Nous utiliserons une API RESTqui possède un cadre moins strict que les autres API existantes, et peut être développée à l'aide de pratiquement n'importe quel langage de programmation et prend en charge divers formats de données.

* Base de données choisie :

Nous avons choisi MongoDB en raison de sa flexibilité pour gérer des données variées sous forme de documents JSON, ce qui permet d’adapter facilement la structure des menus et des plats. De plus, elle peut évoluer facilement dans le futur.

1. **Préconisations concernant le domaine et l’hébergement**

* Nom du domaine :

Nous avons pensé à utiliser le nom de domaine « menumaker.qwenta.com ». Ce choix est pertinent car il reflète clairement l'objectif de l'application tout en intégrant le nom de l’entreprise Qwenta.

* Nom de l’hébergement :

Nous avons identifié quatre hébergeurs disposant de centres de données en France : OVH. Le choix de ces hébergeurs garantit la conformité aux réglementations locales tout en améliorant la performance de l'application grâce à leur proximité géographique.

* Adresses e-mails :

Pour les adresses e-mails, nous avons envisagé les options suivantes : contact@menumaker.qwenta.com ou [support@qwenta.com](mailto:support@qwenta.com).

1. **Accessibilité**

* Compatibilité navigateur:

Le site sera compatible sur les dernières versions de Google Chrome, Mozilla et Safari.

* Types d’appareils :

Le site sera uniquement en version Desktop pour le moment.

1. **Autres spécifications**

* Couleurs du site :

Les couleurs principales du site seront :

- Beige : #FFF4E8;

- Green : #8BC7B1

- Black : #000

- White : #FFF

- Brown : #C5A073

* Accessibilité :

L'application sera conçue pour être accessible en étant navigable au clavier et par lecteur d'écran. Pour cela, des attributs ARIA seront utilisés, ainsi que des balises HTML sémantiques, et des indicateurs visuels de focus, tout en intégrant des outils comme Axe DevTools et Lighthouse pour auditer et améliorer continuellement l'accessibilité.

1. **Recommandations en termes de sécurité**

* Gestion des accès aux comptes :

Utiliser une authentification forte, comme l'authentification à deux facteurs (2FA), pour renforcer la sécurité des comptes utilisateurs.

* Sécurisation des plugins :

Effectuer des audits de sécurité réguliers pour identifier et corriger les failles potentielles dans les plugins installés.

* Protection des données :

Mettre en œuvre des sauvegardes régulières pour assurer la récupération des données en cas de perte ou de corruption.

* Mises à jour régulières :

Assurer que toutes les dépendances, y compris le système d'exploitation, les bibliothèques et les frameworks, sont régulièrement mis à jour pour bénéficier des derniers correctifs de sécurité.

* Surveillance et journalisation :

Mettre en place une surveillance des journaux d'accès et des événements pour détecter les activités suspectes et répondre rapidement aux incidents de sécurité.

1. **Maintenance du site et futures mises à jour**

La maintenance du site est essentielle pour garantir son bon fonctionnement, sa sécurité et sa capacité à évoluer avec les besoins des utilisateurs.

* Durée du Contrat :

Le contrat de maintenance initial couvrira une période spécifique, et les modalités de renouvellement seront précisées dans le contrat.

* Fréquence des mises à jour :

Mises à jour régulières du site, incluant les mises à jour de sécurité, les améliorations de performance et les nouvelles fonctionnalités tous les trimestres.

* Surveillance de la performance :

Mise en place des outils de surveillance pour évaluer la performance du site et détecter les problèmes techniques.

* Support technique :

Un support technique réactif sera fourni pour résoudre rapidement les problèmes signalés par les utilisateurs, avec des temps de réponse garantis.

* Sauvegardes :

Sauvegardes régulières des données et du site pour éviter la perte d'informations en cas de problème technique.

* Rapports de maintenance :

Des rapports périodiques seront fournis sur les activités de maintenance, les mises à jour effectuées et les statistiques de performance du site.

La mise en place d'un contrat de maintenance robuste et de mises à jour régulières est essentielle pour garantir la performance, la sécurité et l'évolutivité du site à long terme.