Spéciﬁcations techniques

Menu Maker by Qwenta

**Projet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Auteur** | **Date** | **Approbation** |
| 1.0 | Mathieu | 12/09/2024 | Soufiane |

[I. Choix technologiques](#_409jm82qlgst) 2

[II. Liens avec le back-end](#_ex0md8td8283) 3

[III. Préconisations concernant le domaine et l’hébergement](#_a61v3uh5vs19) 3

[IV. Accessibilité](#_n56najuvb5ys) 3

[V. Recommandations en termes de sécurité](#_l80x4g9stizp) 3

[VI. Maintenance du site et futures mises à jour](#_vbbknpfyus5o) 4

1. **Choix technologiques**

* État des lieux des besoins fonctionnels et de leurs solutions techniques :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Besoin** | **Contraintes** | **Solution** | **Description de la solution** | **Justification (2 arguments)** |
| Landing non connecté | Ajout d’animation sur la photo de bannière et des sections | CSS en utilisant useState et useEffect | Ajout d’animation simple et fluide | Les animations seront fluides se déclencherons après le montage des composants  Animations simple à implémenter |
| Sur la landing page (avant que l’utilisateur soit connecté), accès au Dashboard, aux « tarifs », « Mentions légales » et « Tous droits réservés ». | Modale « Tous droits réservés », doit être accessible sur toutes les pages. | React-modal | Création d’un composant « modal » | Implémentation simple dans les différentes pages  Accessibilité simple pour l’utilisateur |
| Connexion/déconnexion de l’utilisateur | L’utilisateur doit pouvoir se connecter grâce à une adresse mail, via une plateforme sécurisée | Auth0 | Auth0 est une plateforme d'authentification et d'autorisation qui permet en toute sécurité les connexions utilisateur | Utilisation d’un JWT (Json web token), qui contient toutes les informations et les permissions  Utilisation d’un token afin de pouvoir effectuer des requêtes API plus tard. |
| Création des catégories | L’utilisateur doit pouvoir créer une catégorie ou bien choisir une catégorie créer en amont | Framework React via Fetch | Stockage des catégories crées précédemment de façon permanente entre sessions. | Possibilité d’utiliser des catégories identiques sur 2 sessions différentes.  Accessibilité simple pour l’utilisateur |
| Création du Menu | Pouvoir entrer le nom des plats, le prix, la description, et l’image. Et de tout personnaliser | Créer un formulaire avec des imputs | Le formulaire est une manière simple et efficace pour envoyer des données. | Possibilité de récupérer des données via le backend  Affichage clair et lisible pour le remplissage du formulaire via des labels |
| Exportation de menus en PDF | Les utilisateurs doivent être capables d'exporter leur menu en PDF directement depuis l'application. | jsPDF | Permettre aux utilisateurs de générer et de télécharger des menus en format PDF à partir de l'application. | Intégration facile via React  Parfaitement adaptée pour l’exportation de fichier PDF |
| Impression du menu | L’utilisateur doit pouvoir imprimer son menu une fois totalement crée | React-to-print | Permettre aux utilisateurs d'imprimer leur menu directement à partir de l'application. | Permet d'ajouter rapidement une fonctionnalité d'impression à n'importe quel composant  Possibilité de styliser et de contrôler précisément ce qui est imprimé |
| Intégration avec les plateformes de réseaux sociaux et de livraison | Les utilisateurs doivent pouvoir partager leur menu sur Instagram et exporter facilement leurs menus sur  Deliveroo | API | Intégration des API Instagram et Deliveroo pour permettre le partage et l'exportation du menu sur ces plateformes | Améliore la visibilité du restaurant via les réseaux sociaux  Possibilité de meilleures ventes grâce à d’autres application |
| Diffuser ou imprimer un menu via le Dashboard | L’utilisateur doit-être rediriger vers le back-office | React-router | Redirection simple de l’utilisateur via react-router | Redirection d’une page vers une autre  Faciliter de navigation pour l’utilisateur |
| Gestion des menus | Modification / suppression de ses menus | Requête fetch avec GET POST et DELETE | Utilisation des requêtes pour récupérer des données du backend | En utilisant comme précédemment JWT, aucun problème d’affichage de menu qui ne sont pas en lien avec le restaurant  Affichage dynamique des menus |
| Gestion du compte | Le restaurateur doit pouvoir modifier son adresse ou en rajouter plusieurs. | Firebase Authentification | Pouvoir gérer en toute sécurité la gestion de ses adresse mails | Possibilité de lier plusieurs adresse mails  Possibilité de modifier les données d’un compte avec une sécurité amélioré |
| Personnalisation du branding du restaurant | Les restaurateurs doivent pouvoir ajouter, modifier et supprimer leur logo et les couleurs de base | React-color et Firebase Storage | Choix facile d’un panel de couleur et du choix d’un logo | Facile à configurer, stockage sécurisé, gestion des fichiers en temps réel  Fonctionne très bien avec react |

1. **Liens avec le backend**

Langage côté serveur :Node.js avec Express.js qui est idéal pour des applications web en temps réel et une gestion simultanée de plusieurs requêtes

API : Il est primordial d’utilisé ici une api car nous avons une partie frontend qui communique avec le backend.  
Ici nous avons les différents menus qui serons stocké dans notre backend, et qui serons rappelés grâce à JWT du restaurant.

De même pour faire appel à des API externe comme Deliveroo pour diffuser le menu.

Base de données : Ici j’utiliserais MongoDB qui une base de données NoSQL qui stocke des données sous forme de documents JSON.

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

Nom du domaine : menumaker.qwenta.com

Cohérence avec la marque Qwenta et sous-domaine lié

Nom de l’hébergement : Amazon Web Services (AWS)

Plateformes permettant de gérer des pics de trafic sans problème

Adresses e-mail : support@menumaker.qwenta.com

1. **Accessibilité**

Compatibilité navigateur : Chrome, Safari et Firefox

Types d’appareils : Desktop uniquement

L'application devra être accessible au minimum : navigable depuis le clavier, et lisible par un lecteur d'écran.

1. **Recommandations en termes de sécurité**

Auth0 gère la sécurité des connexions utilisateurs et la gestion des identités.

JWT permet de sécuriser les échanges entre le client et le serveur en encodant les informations d'authentification dans un token.

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

* Mise à jour régulière des dépendances des bibliothèques utilisés afin d’éviter tout problème de fonctionnement de l’application.
* Mise à jour régulière pour la compatibilité des navigateurs choisit.
* Mise en place d’une solution de monitoring du site.
* Mise en place de sauvegarde automatique de donnée afin de palier a toute éventuelle perte de données.
* Mise en place d’un **Certificat SSL et protection des données** : Le site devra être continuellement surveillé pour assurer que le certificat SSL est valide et quelles communications entre le site et ses utilisateurs sont sécurisées.