

Spécifications techniques

Menu Maker - Qwenta

Version	Auteur	Date	Approbation
1.0	Clouet Léo	09 11 24	Soufiane,Product Owner

I. Choix technologiques

II. Liens avec le back-end

III. Préconisations concernant le domaine et l'hébergement

IV. Accessibilité

V. Recommandations en termes de sécurité

VI. Maintenance du site et futures mises à jour

I. Choix technologiques

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

Besoin	Contraintes	Solutions	Description de la solution	Justifications (2 arguments)
Création de la page d'accueil	Site en version desktop	HTLM, CSS, SASS, JSX	Utiliser HTML pour structurer la page, CSS/SASS pour les styles et la personnalisation avancée, et JSX pour intégrer la logique React dans le composant d'accueil.	<p>1. Modularité : Avec JSX, chaque composant de l'interface peut être structuré séparément, facilitant la maintenance.</p> <p>2. Efficacité de développement : SASS permet d'organiser le CSS, d'utiliser des variables et des mixins, ce qui accélère la création et la gestion des styles de manière flexible.</p>
Connexion	L'utilisateur doit pouvoir se connecter de manière sécurisé	node.js/SuperTokens	Utiliser Node.js pour gérer le serveur et les opérations back-end, et SuperTokens pour fournir une authentification sécurisée, notamment par mot de passe et réseaux sociaux.	<p>1. Sécurité avancée : SuperTokens offre des protections contre les attaques (comme les attaques par force brute) et gère la gestion sécurisée des sessions.</p> <p>2. Simplicité d'intégration : SuperTokens s'intègre facilement avec Node.js, offrant une API simplifiée et rapide pour mettre en place l'authentification.</p>

Besoin	Contraintes	Solutions	Description de la solution	Justifications (2 arguments)
Création de plat	L'utilisateur doit pouvoir avoir un aperçu de son plat lorsqu'il remplit le formulaire	React File Reader	Utiliser React File Reader pour permettre aux utilisateurs d'importer facilement des images de leurs plats dans l'interface de création. Cette solution gère le chargement de fichiers via le front-end et permet de lire les images localement avant l'envoi au serveur.	<p>1. Simplicité d'utilisation : React File Reader permet une intégration facile des fonctions de chargement de fichiers, offrant une expérience utilisateur fluide.</p> <p>2. Prévisualisation immédiate : Les images peuvent être lues et affichées avant envoi, garantissant une bonne qualité visuelle et permettant aux utilisateurs de voir le rendu directement.</p>
Création d'une catégorie de menu	L'utilisateur doit pouvoir créer une nouvelle catégorie de menu directement à partir de l'écran de création de menu	React-modal	Utiliser React-Modal pour afficher une fenêtre modale lors de la création d'une nouvelle catégorie de menu. Cette fenêtre modale s'ouvre de manière dynamique pour permettre aux utilisateurs d'ajouter les informations de la catégorie sans quitter la page actuelle.	<p>1. Expérience utilisateur : Avec React-modal, l'ajout de catégories se fait sans redirection, garantissant une navigation fluide et intuitive.</p> <p>2. Personnalisation et accessibilité : React-modal est hautement personnalisable et prend en charge l'accessibilité, répondant aux standards pour tous les utilisateurs.</p>

Besoin	Contraintes	Solutions	Description de la solution	Justifications (2 arguments)
Impression du menu	Une fois le menu terminé, il est possible à l'utilisateur de l'imprimer	React-to-print	Utiliser React-to-print pour permettre aux utilisateurs d'imprimer facilement leur menu directement depuis l'application. Cette solution offre une interface rapide pour générer un aperçu imprimable et sélectionner les éléments nécessaires.	1. Simplicité d'intégration : React-to-print s'intègre facilement dans un composant React, permettant de déclencher l'impression avec un simple bouton. 2. Contrôle du style d'impression : Cette solution permet de personnaliser les styles appliqués lors de l'impression, garantissant un rendu professionnel et adapté pour le menu.
Exportation du menu	L'utilisateur doit pouvoir télécharger son menu	JsPDF	Utiliser jsPDF pour générer un fichier PDF du menu directement depuis l'application. Cette solution permet aux utilisateurs d'exporter leur menu en format PDF, avec des styles et une mise en page personnalisée.	1. Portabilité : Le PDF est un format universel qui peut être partagé et imprimé facilement par les restaurateurs. 2. Personnalisation des styles : jsPDF permet de contrôler la mise en page, offrant un rendu professionnel et adapté au branding du restaurant.

Besoin	Contraintes	Solutions	Description de la solution	Justifications (2 arguments)
Diffuser un menu	L'utilisateur doit pouvoir diffuser/partager son menu sur Deliveroo et Instagram	API Instagram API Deliveroo	Utiliser les API de Deliveroo pour partager les menus directement sur leur plateforme, et l'API d'Instagram pour promouvoir les plats via des publications automatisées ou des stories. Cette solution permet une diffusion rapide sur des canaux populaires, augmentant ainsi la visibilité du restaurant.	1. Visibilité accrue : Ces plateformes sont très fréquentées, offrant une large audience potentielle. 2. Automatisation : Les API permettent de mettre à jour le menu en temps réel sur Deliveroo et de synchroniser les promotions Instagram, sans interventions manuelles fréquentes.
Personnalisation du menu	L'utilisateur doit pouvoir personnaliser son menu	React-formik API Google Fonts	Utiliser React-Formik pour gérer la saisie des informations personnalisées du menu (comme les noms des plats, les descriptions, et les prix), tout en simplifiant la validation des données. L'API Google Fonts permet d'intégrer facilement des polices d'écriture variées et adaptées pour personnaliser l'apparence du menu.	1. Gestion des formulaires : React-Formik offre une solution robuste et flexible pour la gestion des formulaires avec une validation en temps réel. 2. Large choix de polices : L'API Google Fonts offre un large éventail de polices gratuites et optimisées pour le web, permettant de personnaliser facilement l'apparence du menu tout en améliorant son lisibilité.

Besoin	Contraintes	Solutions	Description de la solution	Justifications (2 arguments)
Gestion des emails	Un email de confirmation doit être envoyé à l'utilisateur/Il doit pouvoir contacter l'équipe technique en cas de besoin	NodeMailer	Utiliser NodeMailer pour envoyer des emails automatiquement depuis l'application. NodeMailer permet d'intégrer facilement un service d'envoi d'emails pour la confirmation d'inscription, la gestion des notifications et autres communications avec les utilisateurs.	<p>1. Fiabilité : NodeMailer est une solution bien établie et largement utilisée pour l'envoi d'emails en Node.js, garantissant une bonne fiabilité.</p> <p>2. Flexibilité : Il permet de configurer plusieurs services d'email (comme Gmail, SendGrid, ou SMTP), offrant ainsi de nombreuses options pour répondre aux besoins spécifiques du projet.</p>
Authentification	Assurer la protection des sessions contre des attaques	JSONWebTokens	Utiliser JSON Web Tokens (JWT) pour sécuriser l'authentification des utilisateurs. JWT permet de générer des tokens qui seront utilisés pour vérifier l'identité des utilisateurs sur chaque requête, sans avoir besoin de stocker des informations de session côté serveur.	<p>1. Sécurité : Les tokens JWT sont signés et chiffrés, garantissant la sécurité des échanges entre le client et le serveur.</p> <p>2. Scalabilité : Comme JWT ne nécessite pas de stockage côté serveur pour les sessions, il permet une gestion plus légère et scalable des utilisateurs, idéale pour les applications distribuées.</p>

II. Liens avec le back-end

Quel langage pour le serveur ?

Node.js a été choisi pour le serveur en raison de sa performance élevée et de son modèle non-bloquant, permettant de gérer efficacement les requêtes en temps réel et de créer des applications web scalables et réactives avec une architecture légère.

A-t-on besoin d'une API ?

API Instagram

API Deliveroo

Base de données choisie

MongoDB a été choisi pour sa scalabilité et flexibilité, permettant de gérer des données non structurées, facilement évolutives, et adaptées à des modèles dynamiques comme ceux nécessaires pour un système de gestion de menus et de catégories.

III. Préconisations concernant le domaine et l'hébergement

Nom du domaine :

<http://www.menu-maker.qwenta.fr>

Nom de l'hébergement :

Hostinger a été choisi pour l'hébergement en raison de son excellent rapport qualité-prix et de ses performances fiables, offrant des serveurs rapides et une infrastructure optimisée pour les applications web à fort trafic, tout en étant facile à configurer et à gérer. De plus, Hostinger propose une assistance client réactive et des options de scalabilité, ce qui permet de s'adapter facilement à la croissance du projet.

Adresses e-mail :

help.menu-maker@qwenta.fr => Contact pour les demandes d'aide

contact.menu-maker@qwenta.fr => Contact lorsqu'il y'a un problème sur le site

IV. Accessibilité

Compatibilité navigateur

La solution développée doit être compatible avec les principaux navigateurs web modernes, tels que **Google Chrome, Mozilla Firefox, Safari**, et **Microsoft Edge**. Il est également recommandé de vérifier la compatibilité avec les anciennes versions de ces navigateurs, en particulier **Chrome** et **Firefox**, pour garantir une expérience optimale pour un large éventail d'utilisateurs.

Types d'appareils

Ordinateurs de bureau : L'application doit être parfaitement fonctionnelle sur les principales résolutions d'écran des ordinateurs de bureau et portables, en garantissant une interface fluide et bien structurée.

Autres types d'accessibilité :

Accessibilité pour les personnes malvoyantes : Assurer que le contenu de l'application est accessible via des lecteurs d'écran (par exemple, NVDA, JAWS ou VoiceOver).

Accessibilité cognitive : Utiliser un langage simple et clair pour éviter la confusion, particulièrement pour les utilisateurs ayant des troubles cognitifs. Cela inclut des menus faciles à comprendre et des instructions claires pour la navigation.

Accessibilité motrice : S'assurer que l'application peut être utilisée efficacement via le clavier.

V. Recommandations en termes de sécurité

Protection des données sensibles

Utiliser des algorithmes de hachage sécurisés comme bcrypt ou Argon2 pour stocker les mots de passe de manière sécurisée. Ne jamais stocker les mots de passe en clair.

Assurez-vous que toutes les communications entre le client et le serveur soient chiffrées via HTTPS pour protéger les données en transit.

Authentification et gestion des sessions

Implémenter JWT pour l'authentification et les sessions utilisateur. Assurez-vous que les tokens sont sécurisés (avec une expiration adéquate et un mécanisme de rafraîchissement de tokens).

Sécurisation des entrées utilisateurs

Intégrer des bibliothèques de sécurité comme Helmet.js (pour sécuriser les en-têtes HTTP) et cors (pour gérer les politiques de partage de ressources entre origines).

Surveillance et détection des incidents de sécurité

Implémenter une solution de gestion des logs pour suivre les événements de sécurité comme les tentatives de connexion suspectes et les erreurs systèmes.

Utiliser des outils de détection des intrusions pour identifier toute activité anormale dans l'application ou le réseau, comme Fail2Ban pour limiter les tentatives de connexion échouées.

VI. Maintenance du site et futures mises à jour

Maintenance régulière

Il est essentiel de maintenir les serveurs, les bibliothèques et les frameworks à jour pour éviter les vulnérabilités de sécurité. Des mises à jour régulières des dépendances du projet (comme les packages Node.js ou les plugins React) sont nécessaires pour garantir la compatibilité avec les dernières versions et fonctionnalités.

Support et assistance utilisateur

Mettre en place un système de support pour répondre aux questions et résoudre les problèmes rencontrés par les utilisateurs. Cela pourrait inclure un système de tickets, une FAQ détaillée, ou un chat en direct.

Support et assistance utilisateur

Prévoir des sauvegardes régulières des données utilisateurs et de l'application afin d'éviter la perte de données importantes en cas de panne ou de faille de sécurité.

Planification des mises à jour futures

Le projet doit être conçu pour s'adapter facilement à de futures fonctionnalités ou à une augmentation du nombre d'utilisateurs. Cela inclut la possibilité d'intégrer des outils externes, d'étendre l'architecture et d'ajouter des capacités supplémentaires (comme la gestion de menus plus complexes ou l'intégration avec d'autres services).

Documentation continue

La documentation technique et utilisateur doit être régulièrement mise à jour en fonction des nouvelles fonctionnalités, des modifications apportées au système et des processus internes.