

Spécification techniques

[Menu Maker - QWENTA]

Version	Auteur	Date	Approbation
1.0	Martial, Webgencia	6 octobre 2024	Responsable, Qwenta

I. Choix technologiques 1

II. Liens avec le back-end 3

III. Préconisations concernant le domaine et l'hébergement.....4

IV. Accessibilité4

V. Recommandations en termes de sécurité5

VI. Maintenance du site et futures mises à jour..... 5

I. Choix technologiques

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

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
Obtenir des informations (Landing page, tarifs, exemples, mentions légales)	Accessibles à tous les utilisateurs. Les mentions légales s'ouvrent dans une modale	- React.js - React-modal	React-modal permet de mettre en place des fenêtre sous forme de modale.	1) React est une librairie très populaire qui permet de créer des interfaces utilisateurs rapidement à l'aide de composants réutilisable. 2) Le composant react-modal permet de garder facilement l'accessibilité sur les modales.
Inscription et authentification des utilisateurs	Système de sécurité. Vérification de l'email.	- MongoDB - Mongoose - JSON Web Token (JWT) - Regex - Nodemailer	L'authentification sera réalisée via des tokens JWT envoyés par email. Lorsque l'utilisateur clique sur le lien de confirmation, le token sera vérifié par le backend pour valider la connexion. Les tokens JWT géreront également la session de l'utilisateur après connexion. L'envoi d'emails se fera via Nodemailer .	1) JWT permet de créer un Token (clé secrète) qui peut être actif pendant une durée prédéfinie. 2) Nodemailer est le module le plus utilisé pour les applications Node.js qui permet la validation et la vérification d'email utilisateur.
Créer un menu avec des catégories et des plats (nom, prix, description, photo).	Les menus peuvent varier selon les restaurants (différentes catégories et plats).- Le système doit être simple et rapide pour les restaurateurs.	- MongoDB - React-modale	Utiliser React.js pour créer des formulaires interactifs permettant aux restaurateurs de configurer leurs menus (ajout de catégories et de plats) en temps réel. Node.js servira d'API pour gérer les opérations de création, modification et suppression, avec MongoDB pour stocker les menus et plats sous forme de documents JSON flexibles.	1) Flexibilité : MongoDB permet d'adapter la structure des menus en fonction des besoins spécifiques des utilisateurs sans contraintes de schéma rigide. 2) Simplicité : React.js facilite la création d'une interface utilisateur fluide et interactive pour ajouter ou modifier des catégories et plats dans un menu.

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
Visualisation du menu	Après chaque validation d'étape, le contenu doit s'afficher dans la preview qui se trouve à droite.	- React + useState	Le fait d'utiliser React et son gestionnaire d'état local avec useState permet de gérer l'affichage en temps réel. Cela offre une visualisation et permet d'afficher dynamiquement la conception du menu sans recharger la page.	<p>1) Mise à jour en temps réel en utilisant l'état local de React avec useState, toutes modifications sont instantanément reflétées dans le navigateur. Cela offre une expérience utilisateur fluide et sans chargement de page.</p> <p>2) Simplicité et cohérence en restant dans l'écosystème React sans avoir besoin d'ajouter une bibliothèque ou plugins supplémentaires. Cela permet de maintenir un code plus léger, cohérent, et facile à gérer.</p>
Les utilisateurs doivent pouvoir télécharger des photos pour les plats et un logo pour leur restaurant.	Les fichiers doivent être optimisés pour garantir un chargement rapide et rester compatibles avec tous les navigateurs.	- MongoDB - Multer - Sharp	Utiliser Multer pour gérer l'upload des fichiers sur le serveur et Sharp pour optimiser les images (compression et redimensionnement).	<p>1) Optimisation : Sharp permet de compresser les fichiers d'image sans perte de qualité, garantissant une performance optimale lors d'affichage des photos de plats et du logo.</p> <p>2) Compatibilité : Multer garantit que les images sont converties dans des formats compatibles avec tous les navigateurs (JPEG, PNG).</p>
Exporter en PDF	Lancement instantané du téléchargement.	- React-PDF	React-PDF permet de générer des fichiers PDF depuis plusieurs formats.	<p>1) Compatibilité avec React et facile d'utilisation.</p> <p>2) Offre de nombreuses fonctionnalités.</p>
Imprimer un menu	Lien qui s'ouvre dans un nouvel onglet	- React-to-print	React-to-print est une solution rapide et efficace pour ajouter une fonctionnalité d'impression dans une application React.	<p>1) Flexibilité : Génère du contenu imprimable tout en conservant les styles.</p> <p>2) Ciblage : Permet d'imprimer un composant spécifique sans imprimer toute la page.</p>
Menus précédents	Gestion des données par utilisateur	- MongoDB - Mongoose	MongoDB stocke les menus précédemment créés par l'utilisateur et lui renvoie sous forme de tableau avec la possibilité de le modifier ou le supprimer.	<p>1) MongoDB permet de stocker les informations dans sa base de données, qui est flexible et facile à mettre en place.</p> <p>2) MongoDB peut stocker différents types de données.</p>
Diffuser sur Deliveroo	Gestion des données et connexion à l'API.	- API de Deliveroo	L'API Deliveroo permet de se connecter puis de lier un menu et ses plats.	<p>1) Accès direct à une clientèle ciblée, le menu est accessible à une clientèle active et prête à commander, augmentant ainsi les chances de ventes.</p> <p>2) Intégration fluide des commandes grâce à l'API, permet de gérer les commandes en temps réel, offrant une expérience fluide pour les utilisateurs, tout en optimisant la gestion des livraisons.</p>

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
Diffuser et partager le menu sur Instagram	Mise en forme du partage et de l'image, et connexion à l'API.	<ul style="list-style-type: none"> - API d'Instagram - React Image 	<p>L'API d'Instagram permet de se connecter puis de publier le menu.</p> <p>Un composant React qui affiche différents type d'image, les images statiques, réseaux, temporaires etc...</p>	<p>1) Visibilité étendue : Utiliser l'API Instagram permet de partager les menu à un large public, augmentant instantanément la portée ainsi que la notoriété.</p> <p>2) Engagement visuel : Instagram, centré sur le visuel, est idéal pour attirer l'attention et encourager l'interaction avec des menus attrayants via posts et stories.</p>

II. Liens avec le back-end

• Quel langage pour le serveur ?

Language : Javascript

Framework :

- Node.js
- Express.js

L'utilisation de **JavaScript** pour le serveur permet de bénéficier d'une cohérence technologique entre le frontend et le backend, ce qui simplifie le développement fullstack en utilisant un seul langage sur l'ensemble de l'application, et permet une meilleure synergie avec **React.js** pour des échanges rapides et efficaces entre les deux couches. En combinant **Node.js** et **Express**, on obtiens un environnement léger, performant et non-bloquant qui permet de gérer efficacement les requêtes API, tout en profitant de la simplicité et de la flexibilité d'Express pour la création d'API RESTful.

• A-t-on besoin d'une API ?

1. API interne (backend avec Node.js et Express) :

Rôle : Cette API servira de pont entre le frontend (React.js) et la base de données (MongoDB). Elle permettra de gérer les utilisateurs, les menus, les authentifications, et toutes les actions CRUD (Créer, Lire, Mettre à jour, Supprimer) pour les menus et les utilisateurs.

Pourquoi ? : Une API est nécessaire pour permettre à l'application d'échanger des données entre le serveur et le frontend. Par exemple, pour créer un menu ou récupérer des informations sur les utilisateurs.

2. API d'envoi d'e-mails Nodemailer :

Rôle : Envoyer des e-mails de confirmation lors de l'inscription des utilisateurs, ainsi que des e-mails de validation lors de la connexion.

Pourquoi ? : Le système nécessite une confirmation par e-mail après inscription et pour valider la connexion. Une API d'envoi d'e-mails est donc essentielle pour cette fonctionnalité.

3. API Deliveroo :

Rôle : Connecter directement l'application avec Deliveroo pour permettre la gestion des commandes de repas via Deliveroo.

Pourquoi ? : L'intégration avec Deliveroo permet aux utilisateurs de commander les plats du menu directement depuis l'interface, ce qui est un avantage concurrentiel pour attirer des clients.

4. API Instagram :

Rôle : Permet de partager et diffuser un menu directement sur Instagram en utilisant leur API pour publier du contenu visuel sur le compte professionnel.

Pourquoi ? : Instagram est une excellente plateforme pour promouvoir des menus et attirer des clients grâce à la diffusion de photos attractives des plats.

- **Base de données choisie : MongoDB (NoSQL)**

MongoDB (NoSQL)

1. **Flexibilité des données** : MongoDB permet de stocker des données non structurées (catégories, plats, options) sans schéma rigide, ce qui est idéal pour des menus variés et évolutifs.
2. **Scalabilité et performance** : Conçu pour une grande scalabilité, MongoDB gère efficacement les volumes de données croissants, assurant des performances optimales à mesure que l'application se développe.
3. **Intégration avec Node.js** : MongoDB utilise le format JSON, parfaitement compatible avec l'écosystème JavaScript/Node.js, facilitant les échanges entre frontend (React) et backend.
4. **Simplicité et rapidité** : MongoDB simplifie la gestion des opérations complexes (ajout de catégories, plats) sans nécessiter de jointures lourdes, ce qui accélère le développement et l'itération.

III. Préconisations concernant le domaine et l'hébergement

- **Nom du domaine** :
Exemple : www.menu-maker.fr

Le nom de domaine sera probablement un sous domaine de Qwenta, cette question est en cours de validation.

- **Nom de l'hébergement** :

AWS est idéal pour ce projet car il offre une **scalabilité flexible** qui permet de répondre aux besoins changeants de l'application à mesure qu'elle grandit, tout en garantissant une **haute disponibilité** pour assurer un accès constant aux menus et aux utilisateurs. Avec **MongoDB** comme base de données, AWS facilite la gestion de gros volumes de données non structurées et permet une intégration transparente avec des services comme **EC2** pour l'hébergement des serveurs et **S3** pour le stockage d'images, offrant ainsi une solution complète et évolutive adaptée aux exigences de mon projet de gestion de menus en ligne.

- **Adresse e-mail** :
contact@menu-maker.fr
ou infos@menu-maker.fr

Ces adresses sont simple, claires, et directement associées au nom de votre site, ce qui facilite la communication avec vos utilisateurs et partenaires. Il faut s'assurer que le domaine est enregistré et que les paramètres de messagerie soient configurés pour recevoir les messages envoyés à cette adresse.

IV. Accessibilité

L'accessibilité est cruciale pour garantir que tous les utilisateurs puissent naviguer et interagir avec le site de manière fluide, quel que soit leur navigateur ou appareil. En assurant la compatibilité avec les navigateurs principaux comme **Chrome**, **Safari** et **Firefox**, on maximise l'accès à une grande diversité d'utilisateurs, tout en s'assurant que les fonctionnalités et l'affichage sont cohérents. De plus, même si le projet est orienté **Desktop**, l'optimisation pour des écrans de bureau garantit que les utilisateurs bénéficieront d'une expérience ergonomique, efficace et performante, en éliminant les obstacles potentiels qui pourraient nuire à l'utilisation du site. Une bonne accessibilité favorise une adoption plus large et une satisfaction accrue des utilisateurs.

- **Compatibilité avec les navigateurs** :

Le site sera optimisé pour une utilisation sur les principaux navigateurs Desktop : **Google Chrome**, **Safari**, et **Mozilla Firefox**. Ces navigateurs étant les plus utilisés, cela garantit une large couverture des utilisateurs potentiels. Des tests approfondis seront réalisés pour assurer une expérience utilisateur fluide et homogène sur ces plateformes.

- **Types d'appareils** :

Le site est conçu exclusivement pour une utilisation **Desktop**. L'interface, la mise en page et les fonctionnalités seront spécifiquement adaptées aux grands écrans d'ordinateurs de bureau et de portables, sans besoin de version mobile. Cette approche permet de concentrer les efforts de conception et de développement sur une expérience optimisée pour les utilisateurs de bureau.

- **SEO (Search Engine Optimization)** :

La stratégie d'acquisition ne prévoit pas d'exigences particulières en termes de SEO.

- **Analyse du comportement utilisateur :**

Il n'y aura pas d'outils à intégrer pour capter le comportement des utilisateurs pour le moment.

- **Couleurs du site :**

- **Beige** : #FFF4E8
- **Green** : #8BC7B1
- **Black** : #000
- **White** : #FFF

V. Recommandations en termes de sécurité

La sécurité est primordiale pour garantir la confidentialité des données sensibles des utilisateurs et protéger l'application contre les cyberattaques et les violations de données. En mettant en place des mesures de sécurité adaptées, on peut prévenir les pertes financières et la dégradation de la réputation de l'entreprise. La sécurité est cruciale pour maintenir la confiance des utilisateurs et assurer la pérennité de l'application.

Sécurité Frontend (React) :

- **Validation des entrées** : Nettoyer et valider les données d'entrée des utilisateurs pour prévenir les attaques XSS (Cross-Site Scripting).
- **Utilisation de HTTPS** : Assurez-vous que toutes les communications entre le client et le serveur sont chiffrées en utilisant HTTPS.
- **Mise à jour des bibliothèques** : Garder **React** et toutes les bibliothèques tierces à jour pour bénéficier des dernières corrections de sécurité.
- **Gestion des secrets** : Stocker les secrets (clés API, jetons) dans des variables d'environnement et non dans le code source pour éviter les fuites.

Configuration de la Sécurité MongoDB :

- **Authentification activée** : Assurez-vous que l'authentification est activée et que chaque utilisateur a des rôles avec les permissions minimales nécessaires pour leurs actions.
- **Chiffrement des données** : Utiliser le chiffrement des données pour protéger les informations sensibles dans la base de données. Cela inclut les données au repos et en transit.

En appliquant ces bonnes pratiques, la sécurité de votre application sera renforcé et permettra de minimiser les risques de vulnérabilité et d'exploitation.

VI. Maintenance du site et futures mises à jour

La maintenance du site comprend plusieurs aspect essentiel lié à l'entretien, la mise à jour, la surveillance continue du site web pour assurer un bon fonctionnement, sa disponibilité, sa sécurité ainsi que son évolution avec les dernières nouveautés du web.

1. **Mises à jour régulières** : Mettre à jour les dépendances (React, Node.js, MongoDB) et les bibliothèques tierces pour assurer la sécurité et la compatibilité.
2. **Surveillance des performances** : Utiliser des outils de monitoring comme **Google Analytics** pour surveiller les performances et détecter les erreurs rapidement.
3. **Sauvegardes régulières** : Sauvegarder automatiquement la base de données et les fichiers critiques, avec des solutions de restauration en cas de problème.
4. **Gestion des logs** : Surveiller et analyser les logs pour identifier les anomalies et prévenir les incidents.
5. **Optimisation continue** : Auditer régulièrement les performances du site (temps de chargement, gestion des caches) et améliorer l'expérience utilisateur.
6. **Sécurité** : Effectuer des contrôles de sécurité réguliers (mises à jour, patches) et des tests de pénétration pour prévenir les failles.
7. **Support et gestion des incidents** : Mettre en place un processus de gestion des incidents et offrir un support pour répondre aux problèmes des utilisateurs.