

# Spécifications techniques

[Menu Maker by Qwenta]

Version	Auteur	Date	Approbation
1.0	Aurélien LE HYARIC	11/06/2024	Soufiane

## Sommaire

I.	Choix technologiques .....	2
II.	Liens avec le back-end .....	7
III.	Préconisations concernant le domaine et l'hébergement .....	8
IV.	Accessibilité.....	8
V.	Recommandations en termes de sécurité .....	9
VI.	Maintenance du site et futures mises à jour.....	10

## I. Choix technologiques

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

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
<b>Accès visiteur au site (landing page)</b>	Navigation fluide dans le menu	ReactJS pour le développement de la landing page.	React est une bibliothèque JavaScript open-source pour construire des interfaces utilisateur, particulièrement pour les applications web à page unique où l'interaction rapide et réactive est essentielle.	1)React utilise un DOM virtuel qui permet de minimiser les manipulations du DOM en n'appliquant que les changements nécessaires. Cela améliore les performances et la réactivité de la page, offrant une navigation fluide et rapide. 2)React permet de créer des composants modulaires et réutilisables. Les composants peuvent être testés individuellement, réutilisés dans différentes parties de l'application et rapidement mis à jour sans affecter l'ensemble

				du site.
<b>Connexion utilisateur</b>	<p>S'effectue dans une modale</p> <p>L'utilisateur peut entrer son adresse mail</p> <p>Mail d'authentification ou confirmation d'email</p> <p>Un lien "Besoin d'aide" permet d'envoyer directement un e-mail à nos équipes</p>	JSON Web Token (JWT) et Nodemailer	Utilisation de JSON Web Token pour gérer l'authentification des utilisateurs et Nodemailer pour envoyer des emails d'authentification sans mot de passe.	<p>1) Les JWT sont signés et peuvent être chiffrés, garantissant l'intégrité et la confidentialité des données transmises.</p> <p>2) Nodemailer permet d'envoyer des emails en utilisant divers services de messagerie comme Gmail, SMTP, SendGrid, etc</p>
<b>Dashboard utilisateur</b>	<p>Accès à la création, diffusion et impression de menu.</p> <p>Accès à la section « pour aller plus loin ».</p> <p>Accès aux 3 derniers articles de blogs.</p>	MongoDB Atlas	<p>Utilisation de MongoDB Atlas pour stocker et récupérer les données utilisateur.</p> <p>API RESTful avec ExpressJS pour gérer les interactions entre le front-end et la base de données.</p>	<p>1) MongoDB est une base de données NoSQL évolutive et performante.</p> <p>2) Offre une intégration facile avec Node.js et Express.</p>
<b>Créer une catégorie de plats</b>	La création ou la sélection d'une catégorie s'effectue dans une modale.	React-modal	Utilisation de react-modal pour créer des modales permettant l'ajout ou la sélection de catégories de plats.	1) S'intègre facilement avec React et permet de nombreuses options de personnalisation

				2) Permet de créer des modales accessibles (focus du clavier, pour les lecteurs d'écrans par exemple)
<b>Créer un plat</b>	<p>La création ou la sélection d'un plat s'effectue dans une modale.</p> <p>La photo du plat doit apparaître en aperçu avant validation du formulaire.</p>	React-modal	Utilisation de react-modal pour la création de plats, avec un aperçu du menu avant validation.	<p>1) S'intègre facilement avec React et permet de nombreuses options de personnalisation</p> <p>2) Permet de gérer l'affichage d'un aperçu du menu en temps réel</p>
<b>Styliser un menu (police couleur)</b>	<p>Choix d'une police parmi une sélection. Choix d'une couleur.</p> <p>Génération d'un aperçu du menu pour visualiser le style appliqué.</p>	Styled-components	<p>Utilisation de styled-components pour :</p> <ul style="list-style-type: none"> <li>-gérer les styles dynamiques des menus.</li> <li>-permettre de générer et de mettre à jour l'aperçu du menu en temps réel.</li> </ul>	<p>1) Permet d'appliquer des styles dynamiquement (police, couleur)</p> <p>2) Facilite la gestion des styles dynamiques directement dans les composants.</p>
<b>Exporter le menu en PDF</b>	Générer le PDF pour pouvoir le télécharger.	jsPDF	Utilisation de jsPDF pour générer des fichiers PDF téléchargeables depuis le site.	<p>1)jsPDF permet de créer des PDF dynamiquement à partir de contenu HTML.</p> <p>2) Facilement intégrable avec une application React.</p>

<b>Partager le menu sur Instagram</b>	Prévisualisation de photos carrées puis redirection auto vers Instagram au clic sur l'encart correspondant.	Sharp	Utilisation de Sharp pour redimensionner les images au format carrés en vue de les partager sur Instagram	1) Sharp permet de redimensionner, convertir et optimiser les images rapidement 2) Sharp utilise des buffers pour un traitement d'image rapide en mémoire, améliorant ainsi les performances globales.
<b>Imprimer le menu</b>	Depuis l'encart correspondant sur la page d'accueil redirection via un nouvel onglet vers le back-office de Qwenta.	Fonction d'impression du navigateur	Utilisation de la fonction d'impression du navigateur pour permettre l'impression directe des menus.	1) Pas besoin de librairie supplémentaire, utilisation de fonctionnalités natives du navigateur. 2) Simplifie le processus d'impression pour les utilisateurs.
<b>Menus précédents</b>	Affichage des menus précédemment créés avec photo+date.  Modifier un menu. Supprimer un menu. Ajouter un menu.	MongoDB Atlas ExpressJS API RESTful	Utilisation de MongoDB Atlas pour stocker et récupérer les données des menus précédents.  API RESTful avec ExpressJS pour gérer les interactions entre le front-end et la base de données.	1) MongoDB est une base de données NoSQL évolutive et performante. 2) Facilite la gestion des historiques de menus.

<b>Informations légales</b>	S'ouvre dans une modale.	React-modal	Utilisation de react-modal pour afficher les informations légales dans une modale.	1) S'intègre facilement avec React 2) Permet d'afficher des informations sans quitter la page actuelle.
<b>Tarifs</b>	S'ouvre dans un nouvel onglet.	Lien HTML	Utilisation d'un lien HTML pour ouvrir les tarifs dans un nouvel onglet.	1) Simple pour ouvrir des liens externes. 2) Permet de garder la page principale intacte.
<b>Déconnexion</b>	Accessible depuis n'importe quelle page.	JSON Web Token (JWT)	Utilisation de JSON Web Token pour gérer la déconnexion des utilisateurs depuis n'importe quelle page.	1) Les JWT peuvent être invalidés côté serveur pour une déconnexion immédiate. 2) Permet de gérer les sessions utilisateur de manière efficace.
<b>Infos utilisateur</b>	Lier plusieurs adresses e-mails à son compte. Modifier son adresse email de base.	Nodemailer	Utilisation de Nodemailer pour gérer les modifications des adresses email et envoyer des emails de confirmation.	1) Nodemailer permet d'envoyer des emails via divers services de messagerie comme Gmail, SMTP, etc 2) Facile à configurer et à utiliser.
<b>Branding restaurateur</b>	Accès aux ajout/modifications/suppression du logo et des couleurs de base.	MongoDB Atlas	Utilisation de MongoDB Atlas pour stocker et récupérer les données des utilisateurs	1) MongoDB est une base de données NoSQL évolutive et performante.

				2) Facilite la gestion des utilisateurs
--	--	--	--	---

## II. Liens avec le back-end

- Quel langage pour le serveur ?

Pour le serveur, le choix s'est porté sur l'utilisation de **Node.js** avec **Express.js** comme framework. Node.js est idéal pour le développement d'applications en temps réel et Express.js offre une structure légère et flexible pour créer des API et gérer les requêtes HTTP. L'utilisation de Node.js permet de maintenir un environnement JavaScript cohérent côté client (React) et côté serveur, facilitant ainsi le flux de développement et le partage de certaines bibliothèques entre les deux aspects de l'application.

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

II.

Oui, nous avons besoin d'une API (avec ExpressJS) pour différentes fonctionnalités de l'application telles que la gestion des utilisateurs, la création et la modification des menus, et l'authentification.

- Base de données choisie :

Pour la base de données, nous avons choisi **MongoDB Atlas**. MongoDB est une base de données NoSQL qui offre une grande flexibilité et scalabilité, ce qui est idéal pour gérer les données de Menu Maker. MongoDB Atlas fournit un service de base de données entièrement géré et sécurisé, ce qui simplifie la gestion des données et permet une intégration fluide avec les applications Node.js et Express.

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

- Nom du domaine: [www.menu-maker.qwenta.com](http://www.menu-maker.qwenta.com)  
Le nom de domaine sera très probablement un sous-domaine de Qwenta, en cours de validation
- Nom de l'hébergement: Hostinger
- Adresses e-mail:  
[contact@menu-maker.com](mailto:contact@menu-maker.com) pour les communications générales  
[support@menu-maker.com](mailto:support@menu-maker.com) pour l'assistance et les demandes utilisateurs  
[feedback@menu-maker.com](mailto:feedback@menu-maker.com) pour recueillir les avis et retours d'utilisateurs

### IV. Accessibilité

- Compatibilité avec les navigateurs :  
Chrome, Safari et Firefox pour le moment
- Types d'appareils :  
Le site devra être en version Desktop. Pas de version mobile à développer ni à prévoir
- Accessibilité :  
L'application devra être accessible au minimum : navigable depuis le clavier, et lisible par un lecteur d'écran.



## V. Recommandations en termes de sécurité

Pour garantir la sécurité des accès et des plug-ins, nous mettrons en place les mesures suivantes :

### 1) Authentification et autorisation

- **JSON Web Tokens (JWT)** : Sécuriser les échanges entre le client et le serveur.
- **Nodemailer** : Validation des connexions et inscriptions via des emails sans mot de passe.

### 2) Validation des données

- **Validation stricte des données** : Prévenir les attaques SQL Injection et XSS.
- **Tokens CSRF** : Protéger les formulaires et requêtes sensibles.

### 3) Sécurité des communications

- **HTTPS** : Utilisation pour toutes les communications, garantissant la confidentialité et l'intégrité des données.
- **HSTS** : Implémentation pour renforcer la sécurité des communications HTTPS.

### 4) Surveillance et audit

- **Outils de surveillance** : Détecter les activités suspectes.
- **Journaux de sécurité** : Auditer les accès et actions des utilisateurs.

### 5) Gestion des plug-ins et bibliothèques

- **Mises à jour régulières** : Utilisation de plug-ins et bibliothèques régulièrement mis à jour et vérifiés.
- **Outils d'analyse de sécurité** : Utilisation de Dependabot pour détecter les vulnérabilités.

### 6) Conformité RGPD

- **Gestion des données personnelles** : Conformité au RGPD, options pour accéder, modifier ou supprimer les données des utilisateurs.

### 7) Autres mesures de sécurité

- **Helmet** : Configuration des en-têtes HTTP pour prévenir les attaques comme XSS et clickjacking.
- **too-busy-js** : Surveillance et limitation des demandes en cas de surcharge pour prévenir des attaques DOS.
- **Limitation de la taille des requêtes**: Limitation de la taille des charges utiles des requêtes.
- **Limitation des taux de requête**: Utilisation de express-rate-limit pour limiter les requêtes malveillantes.

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

Pour assurer la pérennité et la performance de l'application Menu Maker, nous établirons un contrat de maintenance couvrant les aspects clés suivants :

### 1) Surveillance continue :

- Mise en place d'outils de surveillance pour suivre les performances de l'application en temps réel, incluant la surveillance des temps de réponse, des taux d'erreur et de la disponibilité du service.

### 2) Mises à jour de sécurité :

- Application des correctifs de sécurité dès qu'ils sont disponibles pour prévenir les vulnérabilités.

### 3) Maintenance préventive :

- Planification de maintenances préventives régulières pour vérifier la stabilité, la sécurité et la performance de l'application.
- Tests de sécurité périodiques et des tests de pénétration pour identifier et corriger les failles de sécurité potentielles.
- Mise à jour régulière des dépendances et des bibliothèques utilisées par l'application.

### 4) Compatibilité des navigateurs et des appareils :

- Se tenir informé des mises à jour des navigateurs et s'assurer que l'application reste compatible avec les versions les plus récentes.

- Tester l'application sur différents appareils et navigateurs pour garantir une expérience utilisateur cohérente.

**5) Suivi des erreurs :**

- Utilisation d'outils de suivi des erreurs pour collecter des informations sur les erreurs et les bogues rencontrés par les utilisateurs, permettant d'identifier les problèmes récurrents et de les résoudre rapidement.

**6) Optimisation des performances :**

- Surveillance des performances de l'application et optimisations si nécessaire pour maintenir des temps de chargement rapides et une expérience utilisateur fluide.

**7) Mises à Jour fonctionnelles :**

- Prévoir des mises à jour fonctionnelles pour ajouter de nouvelles fonctionnalités, améliorer l'expérience utilisateur et répondre aux besoins changeants en fonction des retours des utilisateurs et des évolutions du marché.

**8) Sauvegardes régulières :**

- Mise en place de sauvegardes régulières des données et du code pour prévenir la perte de données en cas de problème majeur.

**9) Documentation à jour :**

- S'assurer que la documentation de l'application est à jour (instructions d'installation, de configuration et d'utilisation), facilitant ainsi la maintenance et les futures évolutions du projet.