

# Spécifications techniques

Menu Maker by Qwenta

<b>Projet</b>	Menu Maker by Qwenta
---------------	----------------------

<b>Version</b>	<b>Auteur</b>	<b>Date</b>	<b>Approbation</b>
1.0	Youssef, Webgencia	31 /10/2024	John, Qwenta

I. Choix technologiques .....	2
II. Liens avec le back-end.....	3
III. Préconisations concernant le domaine et l'hébergement .....	3
IV. Accessibilité .....	3
V. Recommandations en termes de sécurité .....	3
VI. Maintenance du site et futures mises à jour .....	4

## 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>Landing page</b>	La landing page doit être attrayante, responsive et optimisée pour la conversion.	HTML, JavaScript (React.js) et SASS	Ces langages de développement web sont utilisés pour la création de page statique.	-Ces langages sont obligatoires pour la création d'une structure web. -Permet une maintenance facile du code.
<b>Obtention des informations (fonctionnement, tarifs, mentions légales)</b>	L'internaute doit pouvoir cliquer sur l'onglet "Tarifs" pour ouvrir un nouvel onglet, tandis que les autres informations sont affichées dans une modale.	React Modal avec lien externe pour "Tarifs"	React Modal permet d'afficher les sections "Fonctionnement" et "Mentions légales" dans une modale dédiée, tandis que le clic sur "Tarifs" ouvre un nouvel onglet.	- React Modal rend les informations accessibles sans encombrer la page principale. - Ouvrir les "Tarifs" dans un nouvel onglet offre une navigation fluide et directe vers une page dédiée pour des détails supplémentaires.
<b>Authentification (Créer un compte/se connecter)</b>	Le système doit garantir la sécurité des données des utilisateurs et permettre une gestion facile des sessions.	JWT et bcrypt	JWT gère les sessions par jetons côté client, et bcrypt sécurise les mots de passe par hachage.	-Sécurité : Authentification sécurisée et simplifiée avec JWT et bcrypt. -Scalabilité : JWT offre une gestion sans état, idéale pour les applications performantes et évolutives.

<b>Création du Menu</b>	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.	<ul style="list-style-type: none"> <li>- En utilisant React, cette librairie s'intègre parfaitement avec notre stack technologique.</li> <li>- C'est l'une des solutions de modales les plus populaires, assurant une large communauté et une bonne documentation.</li> </ul>
<b>Upload et gestions des images</b>	Le système doit permettre l'upload d'images, leur traitement et garantir la gestion des fichiers de manière sécurisée.	Node.js = les middlewares Multer et sharp	Multer facilite la gestion des uploads de fichiers, tandis que Sharp permet de traiter et de redimensionner les images de manière rapide et efficace.	<ul style="list-style-type: none"> <li>- Multer simplifie le processus d'upload, gérant les différents types de fichiers tout en assurant la sécurité des données.</li> <li>- Sharp offre des performances optimisées pour le traitement des images, réduisant le temps de chargement et améliorant l'expérience utilisateur.</li> </ul>
<b>Personnalisation du menu</b>	L'utilisateur doit pouvoir choisir la couleur du texte et la police du menu à partir d'options prédéfinies.	Styled-Components	Styled-Components permet d'appliquer des styles dynamiques (couleur, police) en fonction des choix de l'utilisateur, directement dans le code React.	<ul style="list-style-type: none"> <li>- Styled-Components permet de conditionner les styles en fonction des préférences, comme la couleur et la police sélectionnées, tout en gardant le code propre et modulaire.</li> <li>- Cette solution est compatible avec React et assure une gestion efficace des thèmes et styles prédéfinis, offrant une expérience de personnalisation fluide.</li> </ul>

<b>Accès aux menus précédents</b>	Le restaurateur doit pouvoir visualiser la liste des menus, y compris la date de création, et avoir la possibilité de modifier, supprimer ou créer un nouveau menu.	Base de données MongoDB	MongoDB est une base de données NoSQL qui stocke les menus et leurs informations (date de création) de manière flexible, facilitant leur récupération.	<ul style="list-style-type: none"> <li>- MongoDB offre une structure de données flexible et scalable, idéale pour gérer les informations des menus d'un restaurateur.</li> <li>- Les requêtes peuvent être exécutées rapidement pour récupérer, modifier ou supprimer les données des menus, facilitant ainsi une interaction efficace dans l'application.</li> </ul>
<b>Exportation du Menu en PDF</b>	L'exportation doit être facile, rapide et conserver la mise en page du menu pour une impression de qualité.	Bibliothèque jsPDF	jsPDF est une bibliothèque JavaScript qui génère des fichiers PDF directement dans le navigateur, permettant d'exporter le menu en un document de qualité.	<ul style="list-style-type: none"> <li>- jsPDF s'intègre facilement dans des applications React, facilitant la création de PDF dynamiques.</li> <li>- Elle offre un contrôle précis sur la mise en page et le style, assurant un export professionnel du menu.</li> </ul>
<b>Exportation Deliveroo</b>	L'exportation doit être simple, rapide et permettre une intégration fluide avec l'application Deliveroo.	API Deliveroo	Utiliser l'API de Deliveroo pour permettre l'exportation du menu en un clic, nécessitant une intégration technique pour transmettre les données.	<ul style="list-style-type: none"> <li>- L'API de Deliveroo automatise le processus d'exportation, le rendant rapide et efficace.</li> <li>- Cela assure que les informations du menu restent à jour sur la plateforme de livraison, optimisant la visibilité et les ventes.</li> </ul>

<b>Partage sur Instagram</b>	<p>Le partage doit être simple, attractif et permettre de présenter visuellement le menu de manière engageante.</p>	html2canvas	<p><b>html2canvas</b> est une bibliothèque JavaScript qui permet de capturer un élément DOM (tel qu'un menu) sous forme d'image. Une fois l'image générée, elle peut être téléchargée et partagée facilement sur des plateformes sociales comme Instagram.</p>	<p>- <b>Simplicité d'utilisation</b> : html2canvas est simple à intégrer dans une application React et permet de générer une image du menu en un clic, ce qui est essentiel pour l'expérience utilisateur.</p> <p>- <b>Fidélité et flexibilité</b> : html2canvas conserve tous les éléments visuels du menu (texte, images, styles) dans l'image capturée, garantissant une reproduction fidèle du design.</p>
<b>Commander l'impression d'un menu</b>	<p>L'encart "Imprimer un menu" doit être facilement accessible depuis la page d'accueil et le lien doit ouvrir le back-office de Qwenta dans un nouvel onglet.</p>	<p>Cette spécification nécessite plus d'informations afin d'être étudiée. Selon ce qui est déjà mis en place au niveau du back-office de Qwenta, il peut y avoir plusieurs choses à mettre en place.</p>		

## II. Liens avec le back-end

- ❖ Quel langage pour le serveur ?
  - Nous allons utiliser Node.js avec Express :
    - Rapidité de développement : Avec Express, Node.js permet un développement rapide grâce à sa nature asynchrone, à son écosystème de modules npm et à la simplicité de son framework Express, ce qui facilite la création rapide de serveurs web robustes.
    - Évolutivité et performances : Node.js est conçu pour être évolutif et capable de gérer un grand nombre de connexions simultanées grâce à son modèle non bloquant, telles que les applications en temps réel ou les API. Express quant à lui offre une structure légère et flexible pour développer des applications web efficaces.
- ❖ A-t-on besoin d'une API ? Si oui laquelle ?
  - Oui, il sera nécessaire de mettre en place plusieurs API :
    - Menu API de Deliveroo
    - L'API graph de Meta pour Instagram
    - API RESTful pour gérer les interactions entre le front-end et le back-end, notamment pour l'authentification, la gestion des menus, et le traitement des images.
- ❖ Base de données choisie :
  - **NoSQL** avec **MongoDB** est la base de données choisie, car elle permet une structure flexible et évolutive pour le stockage des menus et des informations associées.

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

#### ❖ Nom du domaine :

- Le nom de domaine sera très probablement un sous-domaine de Qwenta, par exemple : **menu.qwenta.com**.

#### ❖ Nom de l'hébergement :

Si on parle de l'hébergement de la base de données MongoDB, alors **MongoDB Atlas** est une très bonne solution à envisager. Pour l'hébergement de l'application elle-même (c'est-à-dire le serveur qui exécute le code back-end), on pourra envisager des options comme **Heroku**, **DigitalOcean**, ou **AWS**.

#### ❖ Adresse e-mail :

Les adresses e-mail peuvent être configurées sous le domaine principal, par exemple : [contact@qwenta.com](mailto:contact@qwenta.com), [support@qwenta.com](mailto:support@qwenta.com), et [info@qwenta.com](mailto:info@qwenta.com).

### IV. Accessibilité

#### ❖ Compatibilité navigateur :

Le site doit être compatible avec les dernières versions de Chrome, Safari, et Firefox.

#### ❖ Types d'appareils :

Bien qu'il n'y ait pas de version mobile à développer, le site doit être responsive pour s'adapter aux différentes tailles d'écran des ordinateurs de bureau.

## V. Recommandations en termes de sécurité

### ❖ Accès aux comptes :

- Utilisation de **JWT (JSON Web Tokens)** pour gérer l'authentification et sécuriser les sessions utilisateur.
- Les mots de passe doivent être hachés à l'aide de **bcrypt** pour garantir la sécurité des données.
- Implémentation de **SSL/TLS** pour chiffrer les données transmises entre le client et le serveur.
- Veiller à donner des durées de vie raisonnables pour les liens d'authentification et les token de connexion

### ❖ Plugins et dépendances :

- Tous les plugins et dépendances doivent être régulièrement mis à jour pour éviter les vulnérabilités.
- Éviter d'utiliser des bibliothèques ou des frameworks non maintenus.

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

### ❖ Grandes lignes du contrat de maintenance :

- *Surveillance* : Surveillance continue de la performance et de la sécurité du site.
- *Mises à jour régulières* : Application des mises à jour pour le système, les plugins, et les dépendances au moins une fois par mois.
- *Support technique* : Fournir un support technique aux utilisateurs, avec une réponse dans un délai de 24 à 48 heures.
- *Sauvegardes* : Réaliser des sauvegardes régulières de la base de données et des fichiers du site pour garantir la récupération en cas de problème.



❖ **Mises à jour futures :**

Certaines fonctionnalités pourront être ajoutées par la suite, en voici quelques-unes :

- *Développer plusieurs modèles de menu.*
- *Concevoir et intégrer des maquettes pour les tablettes et smartphones.*
- *Intégrer un blog interne dédié au site Menu Maker.*
- *Créer de multiples profils pour les clients ayant plusieurs restaurants.*
- *Implémenter des animations pour dynamiser le site.*
- *Rester informé des dernières tendances pour une évolution continue.*
- *Support Multilingue : Permet de créer des menus dans plusieurs langues pour attirer une clientèle internationale ou multilingue.*