

Spécifications techniques

[Menu Maker by Qwenta]

Version	Auteur	Date	Approbation
1.0	Duvivier Sébastien	14/02/2025	Soufiane

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

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

Besoin	Contraintes	Solution	Description de la solution	Justification
Landing non connectée	<i>L'utilisateur doit comprendre l'utilité de Menu Maker sans être connecté et accéder aux différentes section</i>	<i>React SCSS</i>	<i>Création d'une page d'accueil dynamique avec des composants React réutilisable</i> <i>Utilisation de SCSS pour le style de la page</i>	<ol style="list-style-type: none">1. React permet de créer des interfaces interactives et réactives grâce à ses composants réutilisable, améliorant la performance et la maintenance2. SCSS permet d'écrire des styles plus efficace et de les maintenir grâce à des fonctionnalités comme les variables et les imbrications
<i>Connexion / création de compte / Déconnexion</i>	<i>L'utilisateur doit pouvoir se connecter ou créer un compte via une modale, L'utilisateur doit avoir une connexion sécurisée et doit être déconnecté sans rafraîchissement</i>	<i>React-Modal Node.j (Express) JWT</i>	<i>Utilisation de React-Modal pour l'affichage de la modale</i> <i>Utilisation de Node.js(Express.js) pour gérer les requêtes HTTP</i> <i>Utilisation de JWT pour générer un Token d'authentification lors de la connection et la suppression du token lors de la déconnection</i>	<ol style="list-style-type: none">1. Node.js avec Express.js permet une performance et une scalabilité pour un développement rapide et flexible2. JWT permet de gérer une authentification sécurisée et flexible, sans nécessiter le stockage des sessions sur le serveur

<i>Dashboard utilisateur</i>	<i>L'utilisateur doit avoir une vue d'ensemble claire de ses activités (menus, diffusions, impressions, artc de blog</i>	<i>React API Node.js (GET) + MongoDB</i>	<i>Interface de tableau de bord affichant les options de gestion de menus et informations contextuelles L'appel API pour récupérer les données liées au restaurateur et mongoDB pour les stocker</i>	<ol style="list-style-type: none"> 1. L'API node.js permet de gérer les requêtes assurant une interaction optimale avec le front-end tout en facilitant les mise à jour en temps réel 2. MongoDB permet de centraliser les données
<i>Création/modification de menu</i>	<i>L'utilisateur doit pouvoir créer ou modifier son menu de manière intuitive avec des modales qui permettent l'ajout de catégories ou de plats L'utilisateur peut également y personnaliser le style de son menu.</i>	<i>React avec gestion des états et modales API Node.js + MongoDB</i>	<i>Utilisation de composants dynamiques pour gérer les modales et les formulaires d'ajout/édition de menus. L'API gère les requêtes avec des opérations CRUD pour récupérer les données sur MongoDB</i>	<ol style="list-style-type: none"> 1. React est choisi pour sa gestion efficace des états des modales, permettant des interfaces utilisateur dynamiques et interactives 2. MongoDB stocke les données de manière flexible, facilitant les opérations CRUD pour une gestion optimale des informations

<i>Stockage d'images</i>	<i>Le restaurateur doit pouvoir ajouter des images a chaque plat et garantir leur affichage rapide sans ralentir le site</i>	<i>Cloudinary (API de stockage) + MongoDB</i>	<i>Utilisation de l'API Cloudinary pour stocker et optimiser les images, les télécharger et pour générer un URL</i> <i>Utilisation de MongoDB pour faciliter l'association des images</i>	<ol style="list-style-type: none"> 1. Cloudinary optimise automatiquement les images pour une meilleure performance web 2. MongoDB permet d'associer les images au plat avec l'URL
<i>Exportation PDF du menu</i>	<i>Le restaurateur doit pouvoir générer un fichier PDF et l'exporter</i>	<i>Librairie HTML2PDF</i>	<i>Utilisation de HTML2PDF pour convertir dynamiquement les pages HTML en PDF pour télécharger son menu directement depuis l'interface</i>	<ol style="list-style-type: none"> 1. HTML2PDF est une solution simple à implémenter et compatible avec tous les navigateurs modernes 2. L'export est immédiat et fidèle à la mise en page
<i>Commander l'impression de menu</i>	<i>Le restaurateur doit pouvoir finaliser la commande d'impression facilement, en suivant un processus sécurisé et rapide</i>	<i>Lien de redirection externe</i>	<i>Mise en place d'une redirection vers le Back-Office de Qwenta permettant au restaurateur de finaliser sa commande</i>	<ol style="list-style-type: none"> 1. Permet une gestion simplifiée des commandes d'impressions 2. Externalise le traitement à un service spécialisé
<i>Partage sur Instagram</i>	<i>Le restaurateur doit pouvoir partager sur instagram</i>	<i>API Instagram</i>	<i>Utilisation de l'API instagram et génération d'image carrée côté Front-end</i>	<ol style="list-style-type: none"> 1. Facilite le partage sur Instagram 2. Respect des contraintes visuelles d'Instagram

<i>Export vers Deliveroo</i>	<i>Le restaurateur pouvoir exporter sur Deliveroo et être conforme à leurs exigences</i>	<i>API Deliveroo</i>	<i>Connexion directe via l'API Deliveroo</i>	<ol style="list-style-type: none"> 1. Intégration simple avec la plateforme Deliveroo 2. Automatisation du transfert des menus vers Deliveroo
<i>Page des menus</i>	<i>Le restaurateur doit pouvoir voir, modifier ou supprimer des menus précédemment créés</i>	<i>React avec gestion des états</i> <i>API Node.js + MongoDB (CRUD)</i>	<i>Utilisation de composants React pour afficher les menus précédemment créés avec des options de modification ou suppression</i> <i>Les menus créés sont récupérés via des requêtes GET, PUT et DELETE dans MongoDB</i>	<ol style="list-style-type: none"> 1. Mise à jour dynamique sans rechargement complet grâce à la gestion d'état de REACT et aux méthodes CRUD 2. Optimisation du stockage des menus via MongoDB
<i>Infos utilisateurs</i>	<i>Le restaurateur doit pouvoir consulter et modifier ses informations personnelles</i>	<i>React avec gestion des états</i> <i>API Node.js + MongoDB pour les données utilisateur</i>	<i>Page dédiée sur laquelle l'API permet la récupération et la modification des informations utilisateurs via MongoDB</i>	<ol style="list-style-type: none"> 1. React assure une mise à jour dynamique des informations 2. MongoDB garantit un stockage efficace et sécurisé

<i>Pages statiques (Mentions légales, tarifs)</i>	<i>Les utilisateurs doivent avoir accès aux pages</i>	<i>React</i>	<i>Utilisation de React et ses composants pour afficher "les mentions légales" et les "tarifs"</i>	<ol style="list-style-type: none"> <i>1. React et ses composants permettent la maintenance et la mise à jour rapide</i> <i>2. Réduction de la complexité</i>
---	---	--------------	--	--

II. Liens avec le back-end

- Langage pour le serveur : Node.js (JavaScript)
- API de type RESTful, est gérée par Express.js pour faciliter les échanges entre le front-end et le back-end.
- Base de données choisie : Nous avons opté pour MongoDB (NoSQL) comme base de données, qui stocke les informations des utilisateurs, les menus, ainsi que les plats, catégories et autres données associées.

III. Préconisations concernant le domaine et l'hébergement

- Nom du domaine :
 - Possibilité 1 : **Créer un sous-domaine à partir du domaine principal de Qwenta (c'est l'option envisagée)**
 - Possibilité 2 : Créer un nouveau domaine, par exemple : menu-maker-byqwenta.com
- Hébergement : OVHCloud est le choix idéal grâce à sa fiabilité éprouvée, sa sécurité renforcée, sa scalabilité flexible et garantissant une performance optimale pour notre projet
- Adresses e-mail :
 - Pour possibilité 1 : contact-menu-maker@qwenta.com
 - Pour possibilité 2 : contact@ menu-maker-byqwenta.com

IV. Accessibilité

- Compatibilité navigateur :
 - Le site doit être compatible avec les navigateurs récents tels que Chrome, Firefox, Safari.
- Types d'appareils :
 - L'application doit être accessible en version desktop pour l'instant pas de version responsive ou tablette

V. Recommandations en termes de sécurité

Accès aux comptes :

- **Authentification sécurisée avec JWT** : Utilisation de JWT pour la gestion des sessions et de l'authentification. Les tokens sont signés avec une clé secrète robuste et expirent après une courte durée pour renforcer la sécurité
- **Authentification par e-mail** : Validation des utilisateurs via leur e-mail pour éviter l'utilisation de mots de passe faibles, en s'assurant que seules les personnes autorisées accèdent à leur compte

Sécurité des plugins :

- **Librairies et plugins sécurisés** : Utilisation de librairies et plugins régulièrement mis à jour pour éviter les vulnérabilités. Les mise à jour de

sécurité sont appliqués immédiatement dès leur publications

Gestion des données sensibles :

- **Chiffrement des données sensibles** : Toutes les données sensibles, comme les informations personnelles des utilisateurs et des restaurateurs, sont chiffrées pour garantir leur confidentialité
- **Protocole HTTPS** : Toutes les communications entre le front-end et le back-end sont sécurisées via HTTPS, protégeant ainsi les données pendant leur transmission

VI. Maintenance du site et futures mises à jour

- Mettre à jour régulièrement les dépendances via `npm update`
- Surveiller les performances du site avec des outils comme LightHouse
- Effectuer des sauvegardes régulières des données de la base de données
- Nettoyer périodiquement les données pour éviter la surcharge du serveur
- Assurer la mise à jour constante du site en cas de besoin, effectuer les mises à jour nécessaires
- Garantir la compatibilité du site avec tous les navigateurs, en suivant leurs mises à jour et en ajustant le site en conséquence, Négliger cette tâche risquerait de limiter l'accès au site et d'affecter sa visibilité.
- Fournir un canal de communication pour que les utilisateurs signalent les problèmes rencontrés
- Mettre en place des tests automatisés pour garantir la qualité et la stabilité du site