

Spécifications techniques

Projet	Menu Maker by Qwenta
---------------	----------------------

Version	Auteur	Date	Approbation
1.0	Quentin Duchesne	02.2025	Soufiane Product Owner

1. Choix technologiques 2 - 7
2. Liens avec le back-end 8
3. Préconisations concernant le domaine et l'hébergement 8
4. Accessibilité 9
5. Recommandations en termes de sécurité 10
6. Maintenance du site et futures mises à jour 11 - 12

I. Choix technologiques

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

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
Landing non connectée	L'utilisateur doit pouvoir avoir accès aux différentes sections de la landing page non connectée : <ul style="list-style-type: none"> Bannière Personnalisez votre menu Explications étape par étape 	HTML, SASS	Structuration et mise en forme, ainsi qu'animations simples comme le hover sur les éléments clés interactifs de la page web (liens, boutons)	<ol style="list-style-type: none"> HTML et SASS offrent une structuration claire et des possibilités de style avancées pour un code maintenable et performant. Les animations simples améliorent l'engagement utilisateur tout en garantissant un chargement fluide et rapide.
Page login	Mettre en place une fonctionnalité d'authentification complète permettant aux utilisateurs de : <ul style="list-style-type: none"> Se connecter via leur adresse email (avec détection automatique d'inscription ou 	React.js, React Email + Resend, Node.js avec Express	Une modale de connexion intuitive, avec un React.js, React Email + Resend, Node.js avec Express.processus simplifié qui détecte automatiquement si un utilisateur est nouveau ou déjà inscrit. Gestion fluide des emails via Resend.	<ol style="list-style-type: none"> Technologies modernes et performantes facilitant la mise en production. Performances élevées assurées par React et Node.js pour une expérience utilisateur fluide. Adaptation facile aux

	création de compte).			besoins futurs grâce à l'architecture modulaire.
Catégorie de plat	L'ajout d'une catégorie doit se faire via une modale dédiée.	React.js, React-modal	Création et sélection d'une catégorie via une interface intuitive. Sauvegarde et récupération des catégories existantes.	<ol style="list-style-type: none"> 1. Expérience fluide : Ajout et sélection rapides des catégories. 2. Évolutif : Facilité d'ajout de nouvelles fonctionnalités (édition, suppression).
Création de plat	L'ajout et la modification des plats doivent se faire via une modale dédiée avec scroll pour afficher toutes les informations.	React.js, React-modal	Sélection et modification d'une catégorie pour le plat. Formulaire complet dans une modale permettant de renseigner : Photo, nom, prix, description du plat. Possibilité d'ajouter un nombre illimité de plats.	<ol style="list-style-type: none"> 1. Ergonomie optimisée : Interface fluide avec scroll pour afficher toutes les infos. 2. Adaptation : Possibilité d'ajouter un nombre illimité de plats.
Style de menu	Le restaurateur doit pouvoir personnaliser le style de son menu.	React.js, SASS	Sélection de la typographie et choix de la couleur du texte via une interface intuitive. Aperçu en direct du menu avec les modifications appliquées.	<ol style="list-style-type: none"> 1. Expérience fluide : Modification et visualisation immédiate du style du menu. 2. Personnalisation complète : Les restaurateurs peuvent

				personnaliser facilement l'apparence de leur menu.
Exportation PDF	Le restaurateur doit pouvoir en un clic télécharger le fichier PDF correspondant à son menu.	React-pdf	Utilisation de React-pdf pour générer le PDF du menu. Le restaurateur peut cliquer sur un bouton pour télécharger le PDF du menu.	<ol style="list-style-type: none"> 1. Simplicité : Un seul clic pour télécharger le menu au format PDF. 2. Génération rapide et envoi immédiat.
Commander l'impression d'un menu	<p>Le restaurateur doit pouvoir accéder à la fonction d'impression de menu depuis un encart visible sur la page d'accueil.</p> <p>Le lien redirige l'utilisateur vers le back-office de Qwenta pour finaliser la commande.</p>	React.js, SASS	Ajouter un lien bien visible qui redirige vers le back-office de Qwenta.	<ol style="list-style-type: none"> 1. Intégration fluide : Le back-office de Qwenta est déjà en place, ce qui permet une intégration sans complexité majeure. 2. Simplification du processus : Offrir une solution simple et rapide pour que le restaurateur puisse commander l'impression de son menu.
Menus précédents	<p>Le restaurateur doit pouvoir accéder facilement à une liste de menus créés précédemment depuis une vue centralisée.</p> <p>Il doit pouvoir modifier ou</p>	React.js, Axios ou Fetch API	<p>Une vue dédiée affichant les menus précédents avec les informations essentielles :</p> <ul style="list-style-type: none"> • Date de création de chaque menu. • Options pour modifier 	<ol style="list-style-type: none"> 1. Gestion simple : Permet au restaurateur de voir, modifier, et supprimer ses menus de manière centralisée. 2. Scalabilité : Facilité d'ajout

	supprimer un menu existant directement depuis cette vue.		ou supprimer un menu. <ul style="list-style-type: none"> Un bouton pour créer un nouveau menu. 	de fonctionnalités comme la recherche ou le filtrage des menus dans le futur.
Informations légales	La modale doit être accessible sur toutes les pages (connectées et déconnectées). Le texte statique doit être affiché dans la modale, sans besoin de mise à jour dynamique. La mention "Tous droits réservés" doit être présente en bas de toutes les pages.	React.js, SASS, React-modal	Modale React pour afficher le contenu des mentions légales avec un bouton pour l'ouvrir. Composant statique contenant les mentions légales, pouvant être réutilisé dans la modale. La mention "Tous droits réservés" est intégrée en bas de chaque page à travers un composant global, comme un footer.	<ol style="list-style-type: none"> 1. Accessibilité : Facilité d'accès aux mentions légales de manière simple et rapide, tout en gardant une bonne expérience utilisateur. 2. Simplicité : Solution légère avec un contenu statique dans la modale, qui ne nécessite pas de gestion complexe de l'état.
Tarifs	Le lien vers les tarifs doit ouvrir dans un nouvel onglet pour ne pas quitter la page actuelle de MenuMaker. L'URL des tarifs est à définir, mais elle suivra la logique https://URL_DE_QWENTA/tarifs/menumaker .	HTML, React.js	Créer un lien dans l'interface qui redirige l'internaute vers la page des tarifs. Utiliser un attribut pour ouvrir l'URL dans un nouvel onglet.	<ol style="list-style-type: none"> 1. Accessibilité facile : Les internautes peuvent accéder aux tarifs sans quitter la page MenuMaker. 2. Simplicité : Utilisation d'un simple lien avec redirection vers la page externe des tarifs.
Diffusion réseaux	Le restaurateur doit pouvoir	Deliveroo API,	Utiliser les API pour envoyer	1. Accès aux données en

Exportation Deliveroo Partage sur Instagram	en un clic partager son/ses menus vers l'application Deliveroo et Instagram.	Instagram Graph API	les données des menus et les synchroniser avec les comptes Deliveroo et Instagram.	temps réel et automatisation des services. 2. Gestion améliorée de la présence en ligne.
Déconnexion	Le restaurateur doit pouvoir se déconnecter facilement depuis n'importe quelle page lorsqu'il est connecté.	React.js, React Router	Ajouter un bouton de déconnexion accessible. Lors du clic sur ce bouton, une action est déclenchée pour déconnecter l'utilisateur et il peut-être redirigé vers la page d'accueil ou autre.	1. Simplicité d'utilisation : Un bouton de déconnexion accessible partout pour permettre à l'utilisateur de se déconnecter facilement. 2. Maintien de la sécurité : L'utilisateur sera correctement déconnecté et redirigé vers une page d'accueil sécurisée.
Infos utilisateur	Le restaurateur doit pouvoir lier plusieurs adresses e-mail à son compte. Il doit également pouvoir modifier son adresse e-mail principale.	React.js, Node.js avec Express, MySQL	Ajouter un formulaire de gestion des informations utilisateur. Gérer les mises à jour des adresses e-mail dans la base de données.	1. Flexibilité : Permet de gérer plusieurs adresses e-mail pour un même compte, ce qui peut être utile pour la gestion d'équipe. 2. Sécurité : La validation de l'e-mail principal via un e-mail de confirmation garantit la sécurité des changements.

Dashboard	<p>Le restaurateur doit avoir accès à un Dashboard qui regroupe :</p> <ul style="list-style-type: none"> • La création de menu • La diffusion de menu • L'impression de menu • Les 3 derniers articles du blog de Qwenta qui parlent de MenuMaker 	React.js, Node.js avec Express, MySQL	Organiser le Dashboard en plusieurs sections avec des encarts cliquables pour la création, diffusion et impression de menus.	<ol style="list-style-type: none"> 1. Optimisation de l'expérience utilisateur : Le Dashboard centralise toutes les fonctionnalités importantes du restaurateur en un seul endroit, simplifiant la navigation et l'accès aux actions essentielles. 2. Gestion simplifiée des tâches : En permettant un accès direct à la création, la diffusion et l'impression des menus, le restaurateur peut accomplir ses tâches de manière plus fluide.
Branding restaurateur	Le restaurateur doit pouvoir personnaliser son branding depuis une interface.	React.js, SASS, Node.js avec Express, MySQL	Interface utilisateur pour permettre l'ajout, la modification et la suppression des éléments de branding avec un aperçu instantané des modifications. Gérer la sauvegarde des informations du branding dans la base de données.	<ol style="list-style-type: none"> 1. Simplicité d'utilisation : Le processus de création du branding doit être facile, rapide et visuellement interactif. 2. Personnalisation complète : Cela permet au restaurateur de mieux personnaliser son espace

				et rendre son interface plus cohérente avec son image de marque.
--	--	--	--	--

II. Liens avec le back-end

- **Langage pour le serveur**

L'application repose sur React.js pour le front et Node.js avec Express pour le back-end, un choix cohérent pour une application dynamique et performante.

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

L'application nécessite plusieurs API pour assurer ses fonctionnalités principales :

- API REST interne : pour gérer l'authentification, la gestion des menus et les préférences des utilisateurs.
- API Deliveroo : pour permettre aux restaurateurs de publier leurs menus directement sur Deliveroo.
- API Instagram Graph : pour permettre le partage des menus sur Instagram.

- **Base de données choisie**

MySQL permet une bonne structuration des données et offre une scalabilité adaptée aux besoins d'un outil de gestion de menus.

III. Préconisations concernant le domaine et l'hébergement

- **Nom du domaine**

Étant donné que Menu Maker est un service de Qwenta, il est recommandé d'utiliser un sous-domaine du site principal de Qwenta. Cela garantit la cohérence de la marque et facilite la gestion des accès.

Exemple de sous-domaine : `menumaker.qwenta.fr`

- **Nom de l'hébergement**

Le choix de Vercel est adapté pour le frontend en React.js, car il permet un déploiement rapide, un bon CDN et une gestion des performances optimisée.

Le backend en Node.js avec Express sera hébergé sur Railway, qui offre une solution flexible et évolutive pour gérer l'API REST et la base de données MySQL.

- **Adresses e-mail**

Pour assurer une **communication fluide** avec les restaurateurs et une bonne gestion des notifications, voici les recommandations :

- `contact@qwenta.fr` → Communication générale de l'entreprise.
- `support@menumaker.qwenta.fr` → Assistance spécifique à Menu Maker.
- `no-reply@menumaker.qwenta.fr` → Notifications automatiques (confirmation de compte, factures, etc.).

IV. Accessibilité

- **Compatibilité navigateur**

L'application devra être compatible avec les dernières versions des principaux navigateurs modernes :

- Google Chrome
- Mozilla Firefox

- Apple Safari

L'objectif est d'assurer une expérience utilisateur fluide et sans bugs sur ces navigateurs qui couvrent la majorité des usages professionnels.

- **Types d'appareils**

Desktop uniquement : L'application est conçue pour une utilisation sur ordinateur. Il n'est pas prévu de version mobile ou tablette à ce stade du projet.

Navigation clavier et lecteurs d'écran : L'interface devra être navigable au clavier et compatible avec les lecteurs d'écran pour garantir un minimum d'accessibilité.

Bonnes pratiques supplémentaires : Vérification du contraste des couleurs pour assurer une lisibilité optimale. Utilisation de balises ARIA pour améliorer l'accessibilité des composants interactifs.

V. Recommandations en termes de sécurité

Authentification sécurisée :

- Utilisation de JWT (JSON Web Token) pour la gestion des sessions.
- Expiration automatique des tokens et mécanisme de rafraîchissement.
- Hachage des mots de passe avec « bcrypt », algorithme sécurisé couramment utilisé en Node.js.
- Politique de mots de passe sécurisée en imposant une longueur minimale de 8 caractères avec au moins une majuscule, un chiffre et un caractère spécial.

Gestion des accès :

- Séparation des rôles utilisateur (ex : restaurateur, administrateur).

- Vérification des permissions avant chaque action critique.

Plugins et dépendances

- Mise à jour régulière des dépendances pour éviter les vulnérabilités connues.
- Limitation des dépendances externes aux librairies essentielles pour réduire les risques et optimiser les performances.
- Audit de sécurité pour identifier et corriger les failles potentielles.

VI. Maintenance du site et futures mises à jour

- **Grandes lignes du contrat de maintenance :**

Surveillance et correctifs :

- Correction des bugs critiques sous 48h.
- Mise à jour régulière des packages et des dépendances.

Support et assistance :

- Support technique via e-mail pour les restaurateurs en cas de problème (ex: support@menu maker.qwenta.com).
- Assistance pour l'utilisation des fonctionnalités principales.

Évolutions futures :

- Ajout de nouvelles intégrations (ex: créer un blog interne à MenuMaker, Uber Eats en complément de Deliveroo).
- Optimisation des performances et amélioration de l'expérience utilisateur.
- Éventuelle adaptation à une version mobile si nécessaire.