

Spécifications techniques

[Nom du projet + nom du client]

Version	Auteur	Date	Approbation
1.0	Iloudia Vasseur	05/09/2025	Soufiane, Webgencia

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)
Création de la page d'accueil	Le site devra être en version desktop et en responsive	<ul style="list-style-type: none"> - HTML, CSS, SASS, JSX, - Maquette déjà créée 	<ul style="list-style-type: none"> - HTML et CSS forment la structure de base pour refléter la maquette. - SASS apporte des fonctionnalités supplémentaires et simplifie la maintenance du CSS - JSX permet de créer les composants React pour le DOM, les modales, formulaires et l'interface utilisateur 	<ul style="list-style-type: none"> - Afficher un premier visuel avant la mise en place du backend - Simplifier et organiser le CSS avec SASS - Créer un site interactif et dynamique avec React
Création d'une modale de connexion	La connexion au compte restaurateur doit se faire via une	React-modal	Cette librairie React permet de générer des fenêtres modales	-Permet de créer rapidement des modales fonctionnelles

	<i>modale depuis la page d'accueil</i>		<i>facilement et de manière accessible, avec peu de code nécessaire.</i>	<i>-Outil largement utilisé et reconnu.</i>
<i>Envoi d'un e-mail pour confirmation</i>	<i>Le restaurateur doit recevoir un e-mail de confirmation après avoir rempli la modale.</i>	<i>nodemailer</i>	<i>très populaire et simple pour envoyer des emails via SMTP ou services comme Gmail, Outlook, etc.</i>	<ul style="list-style-type: none"> - <i>Nodemailer s'intègre facilement à Node.js pour envoyer des emails rapidement.</i> - <i>Permet l'envoi via différents SMTP et la personnalisation des emails, assurant une communication sécurisée et professionnelle.</i>
<i>Création d'une modale pour ajouter une catégorie de menu</i>	<i>L'ajout d'une nouvelle catégorie se fera via une modale sur la page « Créer un menu ».</i>	<i>React-modal</i>	<i>Cette bibliothèque React facilite la création de modales efficaces et accessibles avec peu de code.</i>	<ul style="list-style-type: none"> - <i>Librairie React permettant la mise en place de modales.</i> - <i>Très utilisée et reconnue dans la communauté.</i>
<i>Création d'un formulaire dans une modale pour ajouter les plats</i>	<i>Le restaurateur pourra saisir les informations de chaque plat, incluant la photo, le nom, le prix et la description.</i>	<i>React-modal</i> <i>React Hook Form</i>	<i>On utilise React-modal pour afficher le formulaire dans une fenêtre pop-up directement sur la page « Ajouter un plat », et React Hook Form facilement les champs, la validation et la récupération des</i>	<ul style="list-style-type: none"> - <i>Facile à intégrer dans un projet React et adapté aux formulaires dynamiques.</i> - <i>Permettent de gérer la validation et l'état des champs efficacement, offrant</i>

			données. Ces outils permettent de créer un formulaire interactif, clair et réactif pour l'utilisateur.	une expérience utilisateur fluide et sécurisée.
Personnaliser un menu	Le restaurateur pourra ajouter son logo et choisir à la fois la police et la couleur du texte pour personnaliser son menu à son style.	React Colorful	permet d'ajouter facilement un sélecteur de couleur interactif dans une application React, pour que l'utilisateur puisse choisir la couleur du texte ou des éléments de son menu en temps réel.	<ul style="list-style-type: none"> - Fonctionne directement avec React et ne nécessite pas de développement complexe - l'utilisateur peut visualiser immédiatement ses choix de couleur sur le menu.
Exporter le menu en version PDF	Le restaurateur pourra télécharger son menu en PDF d'un simple clic.	jsPdf	jsPDF est une bibliothèque qui permet de créer des fichiers PDF directement depuis le navigateur. Elle s'utilise très bien avec React et rend possible le téléchargement du menu en un seul clic, sans avoir besoin d'un service externe.	<ul style="list-style-type: none"> - Génération rapide et autonome côté navigateur. - Facile à personnaliser pour garder le design du menu.
Partager le menu sur Deliveroo	Le restaurateur pourra publier son menu sur Deliveroo en un clic après sa	API Deliveroo	Cette API permet d'ajouter les menus directement sur la plateforme, avec	<ul style="list-style-type: none"> - Augmente la visibilité et le nombre de commandes.

	création.		images, prix et détails des plats.	<ul style="list-style-type: none"> - Assistance technique disponible en cas de problème d'intégration.
Partager le menu sur Instagram	Le restaurateur pourra publier son menu sur Instagram en un seul clic après sa création.	API Instagram Graph	Cette API permet de publier des images ou médias, ici les menus et photos des plats.	<ul style="list-style-type: none"> - S'intègre facilement à l'application. - Authentification sécurisée pour protéger les données et les interactions avec le compte Instagram.
Gestion du menu et des plats	Le restaurateur doit pouvoir ajouter, modifier ou supprimer un menu et un plat du site et donc de la base de données	MongoDB	stocke les informations sous forme de fichiers JSON flexibles.	<ul style="list-style-type: none"> - Permet de gérer facilement différents types de données pour les plats et menus sans structure fixe. - Optimisée pour les lectures et écritures rapides, idéale pour les applications dynamiques.
Impression du menu	Le restaurateur pourra imprimer le menu directement depuis la page grâce	React-to-print	Cette librairie React permet d'imprimer facilement une section précise de l'interface.	<ul style="list-style-type: none"> - Optimisée pour React. - Permet de

	<i>à la fonction d'impression.</i>			<i>personnaliser les options d'impression.</i>
--	------------------------------------	--	--	--

II. Liens avec le back-end

- **Quel langage pour le serveur ?**

J'ai opté pour node.js surtout parce qu'il fonctionne super bien avec JavaScript et qu'il est très rapide. En plus, son écosystème est énorme grâce à npm, ce qui permet de gagner pas mal de temps pour le développement.

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

Oui, on a besoin de l' API de Deliveroo et d'Instagram Graph pour que les menus puissent être facilement partagés sur ces plateformes.

On a aussi besoin de [node.js](#) pour le backend. C'est grâce à cette API qu'on pourra gérer toute la partie dynamique du site, comme :

- d'ajouter des utilisateurs,
- de se login,
- de récupérer la liste des plats pour le menu,
- et plein d'autres fonctionnalités.

- **Base de données choisie :**

Pour la base de données, J'ai choisi MongoDB. C'est une base NoSQL gratuite, très performante et super flexible. Comme elle est orientée documents, c'est assez simple à prendre en main et pratique à utiliser pour notre projet.

III. Préconisations concernant le domaine et l'hébergement

- **Nom du domaine.**

- Le site sera accessible via <https://www.menu-maker.qwenta.fr>

Pourquoi ?

J'ai choisi ce nom de domaine car il correspond à la logique proposée : "utiliser un sous-domaine de Qwenta" (comme indiqué dans les spécifications fonctionnelles). Ça permet d'intégrer directement le projet dans l'environnement de Qwenta, tout en gardant une adresse claire et facilement identifiable grâce au nom *menu-maker*, qui reflète bien l'objectif du site.

- **Nom de l'hébergement.**

J'ai choisi Hostinger comme hébergeur web, en partie pour son service client rapide et disponible, ce qui est rassurant en cas de besoin. Ses tarifs attractifs et les avis positifs sur TrustPilot ont également influencé notre choix. De plus, Hostinger propose de nombreuses fonctionnalités pratiques, comme le certificat SSL, qui assure une URL sécurisée en « https:// ». Il offre aussi un espace de stockage suffisant pour héberger les photos des menus, ainsi qu'un système de sauvegardes quotidiennes pour garantir la sécurité et la protection des données.

- **Adresses e-mail.**

Il y aura deux adresses mails :

- support@menu-maker.qwenta.fr, pour l'assistance aux utilisateurs, la gestion des questions et des problèmes rencontrés.
- contact@menu-maker.qwenta.fr, pour la communication générale, les partenariats et les demandes d'informations.

IV. Accessibilité

- **Compatibilité navigateur.**
 - Le site fonctionnera sur Safari, Chrome et Firefox.
- **Types d'appareils.**
 - Le site est conçu pour une utilisation sur ordinateur et il n'y aura pas de version mobile à développer ni à prévoir. Il devra être au minimum navigable au clavier et compatible avec un lecteur d'écran, afin de garantir l'accessibilité pour tous les utilisateurs.
- **Navigation et lecture.**
 - L'application devra être accessible au minimum : navigable depuis le clavier et lisible par un lecteur d'écran.

V. Recommandations en termes de sécurité

- **Certificat SSL.**
 - Le site devra utiliser un certificat SSL afin de garantir une connexion sécurisée en « https:// » et de protéger les échanges de données entre les utilisateurs et le serveur.
- **Sauvegardes quotidiennes.**
 - Une sauvegarde automatique des données devra être effectuée chaque jour pour assurer leur récupération en cas d'incident ou de perte. (Hostinger pour l'hébergement et MongoDB pour la base de données)
- **Protection des accès.**
 - Les accès à l'administration et à la base de données devront être protégés par des mots de passe robustes et régulièrement mis à jour.
- **Mises à jour régulières.**
 - Les technologies utilisées (frameworks, bibliothèques et hébergeur) devront être mises à jour afin de corriger les éventuelles failles de sécurité.

VI. Maintenance du site et futures mises à jour

Le contrat de maintenance comprend les points essentiels suivants :

- **Objet du contrat** : Définir les missions de maintenance du site et préciser les responsabilités de chaque partie.
- **Prestations incluses et exclues** : Détail des services couverts par le contrat et ceux qui ne le sont pas.
- **Coût de la prestation** : Montant et modalités de paiement pour les services de maintenance.
- **Durée du contrat** : Période pendant laquelle le contrat est valable.
- **Modalités de rupture** : Conditions permettant de mettre fin au contrat.
- **Clause de résiliation et/ou de sanction** : Mesures prévues en cas de non-respect du contrat.
- **Obligations des parties** : Responsabilités respectives du prestataire et du client.

Le contrat prévoit également différents types de maintenance :

- **Maintenance corrective** : Correction des bogues ou problèmes détectés sur le site.
- **Maintenance préventive** : Contrôles réguliers et mises à jour des fonctionnalités pour assurer la sécurité et la performance du site.
- **Sauvegardes régulières** : Enregistrement périodique des données pour garantir leur récupération en cas d'incident.