

# Spécifications techniques

[Nom du projet + nom du client]

Version	Auteur	Date	Approbation
1.0	[nom de l'auteur]	[date de réalisation du document]	[nom de la personne qui doit valider le document]

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 d'une catégorie de menu	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.	1. Nous avons choisi de développer en React, la librairie est cohérente avec ce choix. 2. Il s'agit de la librairie la plus utilisée.
Authentification sécurisée des utilisateurs (restaurateurs)	Doit être intuitif et adaptable	<b>Mongoose (MongoDB)</b>	Utilisation de Mongoose pour gérer les catégories en base de données	1. MongoDB est flexible pour des structures dynamiques 2. Mongoose simplifie les interactions avec la base

Authentification utilisateur Connexion	Sécurité et scalabilité, simplicité d'intégration	Auth0	Auth0 propose une authentification simple à intégrer avec gestion des rôles et permissions	1. Auth0 est sécurisé et offre une API bien documentée 2. Solution clé en main sans gestion de l'infrastructure
	Gestion de la session et des tokens	<b>JWT (JSON Web Tokens)</b>	JWT permet de gérer les sessions de manière sécurisée	1. Solution standard pour les API REST 2. Facile à intégrer avec des frameworks modernes
Interface pour créer des menus (ajout de plats, gestion de l'ordre, etc.)	Doit être dynamique et responsive	et Draft.js	Draft.js permet de créer un éditeur de texte riche, idéal pour la gestion des sections de menu	1. Permet des fonctionnalités avancées d'édition 2. Facile à personnaliser pour gérer différents types de contenu

	Facilité d'intégration dans l'application	PostgreSQL	PostgreSQL offre une gestion efficace des relations entre les éléments du menu	1. Relationnel et performant pour la gestion des menus complexes 2. Bonne gestion des transactions et requêtes complexes
Customiser les styles du menu (couleurs, polices, images)	L'interface doit être fluide et simple à utiliser	<b>Emotion.js</b>	Emotion permet de gérer le CSS en JS tout en restant performant	1. Plus rapide et léger que Styled Components 2. Hautement personnalisable
	Gestion des thèmes de manière dynamique	Tailwind CSS	Tailwind permet de rapidement appliquer des styles CSS modulaires	1. Solution responsive out-of-the-box 2. Offre une personnalisation rapide et claire pour le design

Partager le menu sur les réseaux sociaux et plateformes partenaires	API disponibles pour l'intégration (Instagram, Deliveroo)	Zapier	Zapier permet d'automatiser les flux de diffusion vers des plateformes comme Instagram ou Deliveroo	1. Automatisation sans besoin de développement complexe 2. Facilité d'intégration avec plusieurs API
	Maintenance et gestion des connexions API	Node.js + Instagram API	Utilisation de l'API Instagram via Node.js pour partager les menus	1. Node.js permet de gérer facilement les requêtes HTTP 2. Bonne documentation de l'API Instagram

Générer un menu prêt à imprimer	Doit supporter différents formats d'impression	Puppeteer (Node.js)	Puppeteer permet de générer des PDF directement à partir d'une page web en prenant en compte le CSS	<ol style="list-style-type: none"><li>1. Simple à utiliser et intègre bien les éléments visuels</li><li>2. Permet de générer des fichiers PDF prêts pour l'impression</li></ol>
	Gestion des fichiers imprimés et sauvegarde	AWS S3	AWS S3 permet de stocker et gérer les fichiers générés	<ol style="list-style-type: none"><li>1. S3 est scalable et sécurisé</li><li>2. Facilité de gestion des fichiers de grande taille</li></ol>

## II. Liens avec le back-end

Langage pour le serveur :

- **Node.js**
  - Justification : Node.js est idéal pour des applications réactives à haute performance. Il permet la gestion simultanée de nombreuses connexions et favorise une communication fluide entre le front-end et le back-end.

A-t-on besoin d'une API ?

- **Oui, une API RESTful via Express.js**
  - Justification : Une API RESTful permet une architecture flexible pour gérer les fonctionnalités comme la création de menus, l'authentification, la personnalisation et la diffusion des menus. Express.js, en tant que framework léger, facilite la gestion des requêtes et des ressources.

Base de données choisie :

- **MongoDB (NoSQL)**
  - Justification : MongoDB est un choix flexible et évolutif pour une gestion rapide des données non structurées, telles que les menus personnalisés, les logos, les couleurs, et les utilisateurs. Son approche document-centric s'adapte bien à la gestion de contenus variés et changeants.

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

Nom du domaine :

- **qwenta-menumaker.com**

- Justification : Un nom de domaine spécifique et simple aide à renforcer l'identité de la plateforme. Il reflète directement la fonctionnalité principale du site : un créateur de menus (MenuMaker) sous la marque Qwenta.

Nom de l'hébergement :

- **Heroku ou Vercel**

- Justification : Ces plateformes permettent un déploiement rapide et facile des applications web Node.js, tout en offrant des options de scaling automatique pour gérer la croissance des utilisateurs.

Adresses e-mail :

- **Exemple : support@qwenta-menumaker.com**

- Justification : Un système d'adresses e-mail personnalisé renforce la crédibilité de la plateforme et facilite la gestion du support utilisateur.

## **IV. Accessibilité**

Compatibilité navigateur :

- **Chrome, Firefox, Safari, Edge**

- Justification : Garantir la compatibilité avec les principaux navigateurs est essentiel pour toucher un large éventail d'utilisateurs et garantir une expérience fluide sur toutes les plateformes.

Types d'appareils :

- **Ordinateurs de bureau, tablettes et smartphones (responsivité)**

- Justification : Le site doit être responsive pour permettre aux restaurateurs d'utiliser la plateforme aussi bien sur ordinateur que sur des appareils mobiles, en assurant une continuité de l'expérience utilisateur sur différents écrans.



## V. Recommandations en termes de sécurité

Accès aux comptes, plugins :

- **Utilisation de JWT pour les sessions utilisateur et OAuth2 pour l'intégration avec les API tierces (Instagram, Deliveroo)**
  - Justification : JWT permet une gestion sécurisée des sessions tout en garantissant une communication cryptée entre le front et le back-end. OAuth2 est la norme pour les authentifications sécurisées avec des services externes, assurant une connexion sûre et fluide.
- **Plugin de sécurité : Helmet pour Express.js**
  - Justification : Helmet ajoute une protection par défaut contre des vulnérabilités courantes, comme le Cross-Site Scripting (XSS) et les attaques par injection de contenu.

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

Grandes lignes du contrat de maintenance :

1. **Veille technologique :**
  - La maintenance inclura des mises à jour régulières des dépendances Node.js, Express.js, et de la base de données MongoDB pour s'assurer de la compatibilité avec les dernières normes de sécurité et de performance.
2. **Contrôles et mises à jour régulières :**
  - Une vérification mensuelle des performances du site sera effectuée pour identifier les éventuels goulots d'étranglement et prévenir les pannes ou interruptions de service.
3. **Surveillance continue :**
  - Utilisation d'outils de monitoring comme New Relic ou Datadog pour surveiller les performances en temps réel et prévenir tout problème lié aux surcharges serveur, fuites de mémoire ou vulnérabilités de sécurité.