

# Spécifications techniques

[ Projet Menu Maker by Qwenta ]

Version	Auteur	Date	Approbation
1.0	Emma Fortat (dév Front-End)	06/2025	Soufiane (Product Owner)

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)
Landing page Afficher l'utilité de Menu Maker à un nouvel utilisateur	Doit être visible sans connexion, contenu informatif, clair à quoi va servir l'outil pour l'utilisateur.	React + HTML5 + SCSS	Utilisation de composants React pour structurer la page. CSS via SCSS pour une gestion modulaire du style. Page construite en composants réutilisables : bannière, sections explicatives, personnalisation.	1) React permet un rendu rapide sans rechargement. 2) Composants réutilisables
Authentification par e-mail	Authentification sans mot de passe. Identification de l'utilisateur. Lien sécurisé, modale dédiée, lien support.	Auth via Node.js + Express + JWT + Nodemailer	Création d'une route /auth qui envoie un token JWT signé via e-mail grâce à Nodemailer. Authentification à usage unique via lien.	1) JWT garantit une authentification sans stockage serveur 2) Nodemailer est compatible avec de nombreuses boîtes mail

Création de catégories de plats	Utilisation en modale, modification dynamique	React + useState / react-modal	Utilisation de react-modal pour afficher une fenêtre de création de catégorie. Données gérées avec useState et envoyées via formulaire.	1) UX fluide sans rechargement 2) Intégration simple avec React
Ajout de plats dans un menu	Interface riche, champs multiples nom du plat, prix, description, photos	React + Cloud Storage + formulaire dynamique	Formulaire embarqué dans une modale. Les images sont uploadées vers un service cloud (via API).	1) Fluide et dynamique. 2) Cloud Storage facilite la gestion de fichiers.
Personnalisation du style du menu	Changement de polices et couleurs	Sélecteurs React + styled-components	Utilisation de composants contrôlés (ex: ThemeProvider) combinés à styled-components pour modifier dynamiquement le thème du menu.	1) styled-components permet de gérer dynamiquement le style 2) Le composant ThemeProvider permet de centraliser la gestion des couleurs et typographies de manière fluide.

Export PDF	Export en un clic	React-PDF	Utilisation de la librairie React-PDF pour générer un fichier PDF depuis le contenu HTML du menu.	1) Outil répandu qui permet un export simple HTML → PDF. 2) Permet d'imprimer ou envoyer
Commande d'impression	Lien vers service externe	Lien vers back-office de Qwenta	Bouton redirigeant vers la page d'impression gérée par Qwenta dans un nouvel onglet.	1) Externalisation de l'impression 2) Intégration simple
Gestion des anciens menus	Visualisation, modification, suppression	API REST Express + MongoDB	Stockage des menus dans MongoDB, accessible via des routes API REST sécurisées.	1) MongoDB est idéal pour structure flexible (menus variés) 2) Express facilite la mise en place d'API CRUD
Mentions légales	Accessible, affichage constant	Modale statique	Modale React contenant les mentions, accessible sur chaque page.	1) Implémentation simple 2) Conformité légale
Tarifs	Lien externe	Lien vers Qwenta	Nouvelle page ouverte avec target="_blank"	1) Facilité d'accès 2) Pas de maintenance côté Menu Maker

Export Deliveroo	Redirection	Lien vers Deliveroo	Encart dans l'interface qui redirige vers la plateforme Deliveroo.	1) Gain de temps pour le restaurateur 2) Pas de configuration technique
Partage du menu sur Instagram	L'encart "Partager sur Instagram" doit s'afficher dans la catégorie "Exportez et diffusez" Au clic sur l'encart, des images du menu au format carré (pour les mettre sur Instagram) sont générées Le restaurateur est redirigé vers son compte Instagram avec les photos carrées des menus	Convertir le PDF en image avec une librairie trouvée sur stackoverflow pdfjs-dist + fonction convertPdfToImages Lien de redirection vers Instagram	Création de visuels exportables optimisés pour Instagram avec lien direct vers la plateforme en cliquant sur l'icône instagram.	1) Automatisation du contenu 2) Outil marketing intégré
Déconnexion de l'utilisateur	La déconnexion doit être immédiate et accessible à partir de toutes les pages	Suppression du token JWT côté client	Mise en place d'un bouton de déconnexion qui supprime le JWT du localStorage ou des	1) Sécurisé 2) Déconnexion immédiate sans appel API

	de l'application.		cookies, ce qui invalide immédiatement la session utilisateur	
Modification profil utilisateur	Ajout e-mail secondaire ou modification profil utilisateur	API Express + MongoDB	Mise à jour des informations via formulaires React reliés à une API REST	1) Flexibilité des comptes, 2) Gestion centralisée avec base NoSQL
Dashboard	Regroupe les fonctions clés et les articles	Page React et API backend	Composants "Créer", "Diffuser", "Imprimer", et un appel API vers le blog Qwenta.	1) Vue synthétique pour l'utilisateur, centralisation des besoins 2) L'appel à l'API permet la récupération des derniers articles automatiquement ce qui permet de le garder continuellement à jour.
Branding restaurateur	Le restaurateur doit pouvoir ajouter / modifier / supprimer le logo et les couleurs de base	Formulaire React et API CRUD	Formulaire React avec champ image et sélecteurs de couleurs. Données stockées via l'API.	1) Image de marque personnalisée 2) La solution permet d'ajouter facilement d'autres éléments visuels à l'avenir avec une cohérence visuelle

## II. Liens avec le back-end

- Quel langage pour le serveur ?
  - ❖ Nous utiliserons [Node.js](#) avec Express.

*Node.js avec Express permet de créer un serveur rapide, évolutif et facile à maintenir, idéal pour des APIs REST modernes.*

*Il offre une parfaite intégration avec JWT, MongoDB et les outils JavaScript.*

- A-t-on besoin d'une API ? Si oui laquelle ?
  - ❖ REST API pour gérer les menus CRUD (GET, POST, PUT, DELETE)
  - ❖ Base de données choisie : NoSQL : *MongoDB Atlas*

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

- Nom du domaine : `menumaker.qwenta.fr` (sous-domaine du site client)
- Nom de l'hébergement :
  - ❖ Front-end : Render déploiement de l'application React (Déploiement automatisé depuis GitHub, HTTPS intégré, hébergement optimisé pour SPAs)
  - ❖ Back-end : Render serveur Node.js avec Express (Gestion simple des services back-end, support des APIs REST, logs, variables d'environnement)
  - ❖ Base de données : MongoDB Atlas - base NoSQL (Stockage flexible et scalable, hébergement sécurisé, connexion facile avec le back-end)

- Adresses e-mail recommandées :
  - ❖ support.menumaker@qwenta.fr (assistance technique)
  - ❖ contact.menumaker@qwenta.fr (questions générales ou commerciales)

## **IV. Accessibilité**

- Compatibilité navigateur : Chrome, Firefox, Safari, Edge
- Types d'appareils : desktop et plus tard développer les versions tablette / mobile.

Points de vigilance :

- ❖ Audit à réaliser avec Wave ou Lighthouse
- ❖ Contraste minimum : WCAG AA
- ❖ Ajouter aria-label, alt sur les images et navigation clavier

## **V. Recommandations en termes de sécurité**

- Authentification avec JWT (expiration)
- Utilisation de Bcrypt avec hachage de mot de passe
- Utiliser HTTPS sur l'ensemble du site et des appels API
- Maintenir à jour toutes les bibliothèques, frameworks et dépendances
- Ne pas stocker d'informations sensibles dans le code source.
- Restriction IP API Mongo Atlas
- CORS mécanisme de sécurité du navigateur pour contrôler les requêtes HTTP



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

- Grandes lignes du contrat de maintenance :
  - ❖ le périmètre de la prestation
  - ❖ les modalités de livraison
  - ❖ le prix
  - ❖ résiliation
  - ❖ droits et propriété
  - ❖ annexes (cahier des charges, maquettes Figma ...)

Les points spécifiques de maintenance / mise à jour qui pourraient être :

- ❖ Effectuer des mises à jour régulières du système et dépendances (REACT / Node / Nginx..)
  - ❖ Surveiller en continu les performances du site
  - ❖ Appliquer des correctifs de bugs en priorité selon leur criticité (par exemple sécurité en 1er)
  - ❖ Maintenir un support utilisateur de base (par e-mail)
  - ❖ Journal des versions (changelog)
  - ❖ Prévoir des revues techniques mensuelles pour anticiper les évolutions nécessaires.
- Mises à jour futures :
  - ❖ Adapter la plateforme aux formats mobile et tablette avec des maquettes dédiées (responsive)
  - ❖ Intégrer des nouveaux modèles de menus personnalisables
  - ❖ Ajouter des animations sur la photo de la bannière et sur les formes géométriques des sections.
  - ❖ Suggestion d'idée de personnalisation de modèles de menu avec l'aide de l'IA
  - ❖ Envisager la mise en place de Dependency-track (rester à jour sur les vulnérabilités et les montées de version)