

Spécifications techniques – Menu Maker by Qwenta

Version : 1.2

Auteur : Mr Laurent

Date : 25/09/2025

Approbation : Soufiane / Qwenta

I. Choix technologiques

Élément	Détails
Création d'un menu avec catégories	<ul style="list-style-type: none">- Ajouter plats par catégories : entrée, plat, dessert- Solution : React + State local + modales (react-modal)- Gestion dynamique via hooks React- Justification : React facilite le rendu dynamique et les modales sont accessibles
Saisie nom, prix et description	<ul style="list-style-type: none">- Interface claire pour restaurateur- Solution : Composants React Form + validation front-end- Champs contrôlés avec messages d'erreur- Justification : UX claire et facilement extensible
Personnalisation branding	<ul style="list-style-type: none">- Sauvegarde préférences pour chaque restaurateur- Solution : React + API back-end /preferences- Justification : données liées à l'utilisateur, aperçu temps réel

Export PDF	<ul style="list-style-type: none"> - Générer un PDF fidèle au design - Solution : jspdf ou pdfkit - Justification : Compatible React/Node, téléchargement ou impression possible
Diffusion Deliveroo / Instagram	<ul style="list-style-type: none"> - Partage via lien ou API - Solution : API REST back-end / Webhooks - Justification : automatisation sécurisée et standardisée
Impression	<ul style="list-style-type: none"> - Impression depuis navigateur - Solution : CSS Print + React, bouton window.print() - Justification : simple, compatible navigateurs
Gestion erreurs utilisateurs	<ul style="list-style-type: none"> - Pages 404 / 500 personnalisées, messages d'erreur front et back - Justification : UX améliorée et informations claires en cas de problème
Performance front-end	<ul style="list-style-type: none"> - Lazy loading images, code splitting, compression gzip/brotli - Justification : accélère le rendu et diminue la latence initiale

II. Liens avec le back-end

Élément	Détails

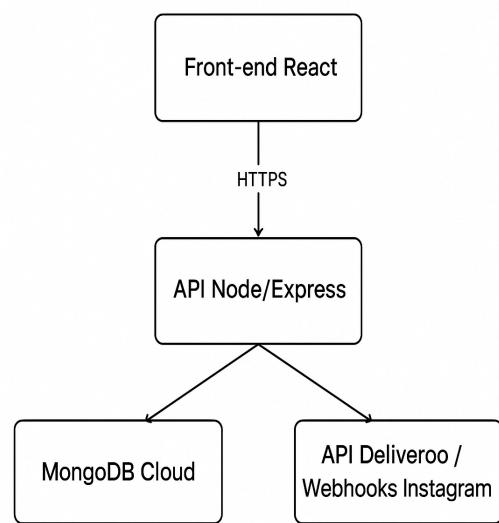
Langage serveur	Node.js avec Express
API	<p>1. Menus</p> <ul style="list-style-type: none"> • <code>GET /menus</code> → lister tous les menus d'un restaurateur (ou filtrer par catégorie, date...) • <code>GET /menus/:id</code> → récupérer un menu précis • <code>POST /menus</code> → créer un menu • <code>PUT /menus/:id</code> → modifier un menu • <code>DELETE /menus/:id</code> → supprimer un menu <p>2. Dishes (plats individuels)</p> <ul style="list-style-type: none"> • <code>POST /menus/:menuId/dishes</code> → ajouter un plat à un menu • <code>PUT /menus/:menuId/dishes/:dishId</code> → modifier un plat • <code>DELETE /menus/:menuId/dishes/:dishId</code> → supprimer un plat <p>3. Preferences / Branding</p> <ul style="list-style-type: none"> • <code>GET /preferences</code> → récupérer préférences de l'utilisateur • <code>PUT /preferences</code> → mettre à jour le logo, couleurs, police <p>4. Export / PDF</p> <ul style="list-style-type: none"> • <code>GET /export/pdf/:menuId</code> → générer et télécharger le PDF <p>5. Diffusion / Webhooks</p> <ul style="list-style-type: none"> • <code>POST /deliveroo/:menuId</code> → envoyer un menu à Deliveroo • <code>POST /instagram/:menuId</code> → envoyer un menu sur Instagram via webhook

	<p>6. Sécurité / Auth</p> <ul style="list-style-type: none"> • POST <code>/auth/refresh-token</code> → renouveler JWT si expiration • Middleware pour vérifier rôle utilisateur sur tous les endpoints sensibles <p>7. Monitoring / Logs</p> <ul style="list-style-type: none"> • GET <code>/logs/front</code> → logs front (Sentry / LogRocket) • GET <code>/logs/back</code> → logs back (Winston / Morgan)
Version API	<code>/api/v1/...</code>
Documentation	Swagger / Postman prévue
Base de données	MongoDB (NoSQL)
Collections principales	<ul style="list-style-type: none"> - users : <code>{_id, email, passwordHash, role, createdAt}</code> - menus : <code>{_id, userId, title, sections:[{name, dishes:[{name, price, description}]}]}, createdAt</code> - preferences : <code>{userId, branding:{logo, colors, font}, lastUpdated}</code>
Indexation	<code>_id, userId</code> pour requêtes rapides
Sécurité back-end	<ul style="list-style-type: none"> - JWT 24h + refresh token - bcrypt pour mots de passe - Rate limiting - Validation côté serveur

	<ul style="list-style-type: none"> - HTTPS obligatoire, CORS configuré
Tests	<ul style="list-style-type: none"> - Unitaires : Jest / Supertest - Intégration : API / Base de données - End-to-end : Cypress

III. Architecture et hébergement

Élément	Détails
Front-end	React (hébergement : Vercel ou Netlify)
Back-end	Node.js / Express (Heroku, Render ou AWS)
Base de données	MongoDB cloud (Mongo Atlas)
Communication	API REST sécurisée front ↔ back ↔ plateformes externes
Schéma technique	Schéma technique – Architecture Menu Maker



Légende :

- **Front-end React** : Interface utilisateur sur navigateur (desktop).
- **API Node/Express** : Back-end sécurisé, gestion des menus et préférences.
- **MongoDB Cloud** : Base de données NoSQL pour utilisateurs, menus et préférences.
- **API Deliveroo / Webhooks Instagram** : Intégration pour diffusion externe des menus.
- **HTTPS** : Communication sécurisée entre front et back-end.

Nom de domaine	sous-domaine Qwenta (ex. menu-maker.qwenta.com)
----------------	---

Emails	contact@menu-maker-qwenta.com, support@menu-maker-qwenta.com
--------	---

IV. Accessibilité

Élément	Détails
Navigateurs compatibles	Chrome, Safari, Firefox (dernières versions)
Types d'appareils	Desktop uniquement (mobile non prioritaire)
Préconisations	<ul style="list-style-type: none">- Navigation clavier- Support lecteur d'écran- Contraste suffisant (WCAG 2.1)- Aria-labels sur boutons et champs

V. Sécurité front-end

Élément	Détails
Validation inputs	Éviter XSS / injections
Librairies utilisées	react-modal, jspdf (librairies maintenues)
Protection routes	Composants sensibles sécurisés et protégés

VI. Maintenance et monitoring

Élément	Détails
Vérification dépendances	npm audit régulier
Sauvegarde	Automatique de la base de données
Mise à jour	Composants React et librairies régulièrement
Futur	Nouveaux templates, options de partage
Monitoring	<ul style="list-style-type: none">- Logs serveur : winston, morgan- Erreurs front : Sentry / LogRocket- Alertes en cas de panne ou erreur critique
RGPD / Données personnelles	Gestion des droits utilisateurs : accès, suppression, conservation des données

VII. CI/CD et déploiement

Élément	Détails
Déploiement front	Automatisé sur Vercel / Netlify

Déploiement back	Automatisé sur Heroku / Render / AWS
Environnements	Gestion dev / staging / prod via variables d'environnement
Pipeline CI/CD	Lint → Tests → Build → Deploy