



# Spécifications techniques

<b>Projet</b>	Menu maker by Qwenta
---------------	----------------------

<b>Version</b>	<b>Auteur</b>	<b>Date</b>	<b>Approbation</b>
1.0	Clément RENAUD	26/07/2023	John (Qwenta)

Le but de ce document est de définir et justifier les spécifications techniques du projet Menu Maker by Qwenta.

## I. Choix technologiques

---

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

**Pour l'ensemble du projet le respect de la charte graphique et du parcours client est primordial.**

Lien de la maquette Figma : <https://www.figma.com/file/Q6NEUPqwz1U3HF7N/Maquette-desktop---Menu-Maker-by-Qwenta?node-id=54%3A145&mode=dev>



Besoin	Contrainte	Solution	Description de la Solution	Justification (2 arguments)
<p>Landing non connectée :</p> <p>Les internautes doivent pouvoir accéder aux différentes sections de la landing page non connectée.</p>	N/A	HTML, CSS et JavaScript avec React	Création d'une landing page avec différentes sections (Bannière, Personnaliser votre menu, explication étape par étape) pour attirer les clients restaurateurs.	<p>- Utilisation d'HTML, CSS et JavaScript avec React permettra de créer une interface dynamique et interactive pour les internautes.</p> <p>- React facilite la construction de composants réutilisables, ce qui peut accélérer le développement de la landing page.</p>
<p>Page login :</p> <p>Les restaurateurs doivent pouvoir se connecter ou créer un compte pour accéder à l'application.</p>	Utilisation de modal pour la création de compte, saisie d'e-mail et envoi de mails de confirmation.	<p>Node.js pour le backend</p> <p>React pour le frontend</p>	Conception d'une page de connexion avec une modal pour la création de compte, saisie d'e-mail et envoi de mails de confirmation pour l'authentification.	<p>- Node.js offre une gestion facile des opérations côté serveur, y compris l'envoi de mails de confirmation.</p> <p>- React facilite la construction d'interfaces interactives pour les restaurateurs.</p>



Besoin	Contrainte	Solution	Description de la Solution	Justification (2 arguments)
<p>Catégorie de plat :</p> <p>Les restaurateurs doivent pouvoir renseigner une catégorie de plat pour organiser leur menu.</p>	<p>La création de catégorie doit se faire dans une modale spécifique et doit être validée.</p>	<p>Node.js pour le backend</p> <p>React pour le frontend</p>	<p>Mise en place d'une interface pour permettre la création et la sélection de catégorie de plat, avec une modale spécifique pour créer une nouvelle catégorie.</p>	<ul style="list-style-type: none"> <li>- Node.js permet de gérer les requêtes backend pour créer et valider les nouvelles catégories.</li> <li>- React facilite la construction d'interfaces conviviales pour les restaurateurs.</li> </ul>
<p>Création de plat :</p> <p>Les restaurateurs doivent pouvoir ajouter des plats dans leur menu avec des détails tels que nom, prix, description, etc.</p>	<p>Possibilité de créer autant de plats que souhaité dans chaque catégorie.</p>	<p>Node.js pour le backend</p> <p>React pour le frontend</p>	<p>Conception d'une interface permettant d'ajouter des plats dans le menu, de les associer à une catégorie et de fournir les informations nécessaires telles que nom, prix, description, etc.</p>	<ul style="list-style-type: none"> <li>- Node.js permet de gérer les requêtes backend pour enregistrer les informations des plats dans la base de données.</li> <li>- React facilite la création d'une interface réactive pour la saisie des informations des plats.</li> </ul>
<p>Style de menu :</p> <p>Les restaurateurs connectés doivent pouvoir personnaliser le style de leur menu avec une typographie et une couleur de texte choisies.</p>	<p>N/A</p>	<p>CSS et React</p>	<p>Conception d'une interface pour permettre aux restaurateurs de sélectionner une typographie et une couleur de texte pour personnaliser le style de leur menu.</p>	<ul style="list-style-type: none"> <li>- CSS permet de personnaliser l'apparence du menu en fonction des choix du restaurateur.</li> <li>- React facilite la gestion interactive des choix de style.</li> </ul>
<p>Exportation du menu en pdf :</p> <p>Les restaurateurs doivent pouvoir télécharger leur menu en format PDF en un clic.</p>	<p>N/A</p>	<p>Node.js pour le backend</p> <p>React pour le frontend</p>	<p>Mise en place d'un bouton d'exportation PDF sur la page du menu, qui permettra de générer le fichier PDF et de le rendre disponible au téléchargement.</p>	<ul style="list-style-type: none"> <li>- Node.js permettra de gérer la génération du PDF à partir des données du menu (PDFKit).</li> <li>- React facilitera l'interaction avec le bouton d'exportation PDF.</li> </ul>



Besoin	Contrainte	Solution	Description de la Solution	Justification (2 arguments)
<p>Commande d'impression du menu :</p> <p>Les restaurateurs doivent pouvoir commander l'impression de leur menu en un clic.</p>	<p>Le lien doit s'ouvrir dans un nouvel onglet vers le back office de l'agence d'impression.</p>	N/A	<p>Ajout d'un lien "Imprimer menu" sur la page d'accueil qui ouvrira un nouvel onglet vers le back office de l'agence d'impression pour passer la commande d'impression du menu.</p>	<p>- Le lien s'ouvrant dans un nouvel onglet évite que l'utilisateur quitte la page principale.</p> <p>- L'utilisation du back office de l'agence d'impression facilite le processus de commande d'impression.</p>
<p>Menus précédents :</p> <p>Les restaurateurs doivent pouvoir accéder à une vue regroupant tous les menus qu'ils ont créés précédemment, les modifier, les supprimer ou en créer de nouveaux.</p>	N/A	<p>Node.js pour le backend et React pour le frontend</p>	<p>Mise en place d'une vue regroupant les menus créés précédemment, avec des options pour modifier, supprimer et créer de nouveaux menus. Affichage des informations de date de création ou de modification.</p>	<p>- Node.js permettra de récupérer les données des menus depuis la base de données.</p> <p>- React facilitera l'affichage des menus et des options de gestion.</p>
<p>Mentions légales :</p> <p>Les internautes doivent pouvoir accéder au contenu "Mentions légales" dans une modale.</p>	N/A	<p>HTML, CSS et JavaScript avec React</p>	<p>Conception d'une modale pour afficher le contenu "Mentions légales", avec l'information "Tous droits réservés".</p>	<p>- Utilisation d'HTML, CSS et JavaScript avec React pour créer une modale interactive pour les internautes.</p> <p>- Révéler l'information "Tous droits réservés" dans la modale pour informer les utilisateurs sur les droits d'auteur.</p>



Besoin	Contrainte	Solution	Description de la Solution	Justification (2 arguments)
<p>Onglet tarif :</p> <p>Les internautes doivent pouvoir ouvrir un nouvel onglet avec les tarifs de Qwenta.</p>	<p>L'URL de la page de tarifs de Qwenta n'est pas encore disponible.</p>	<p>N/A</p>	<p>Ajout d'un onglet "Tarifs" sur la page du menu qui ouvrira un nouvel onglet avec l'URL de Qwenta pour la page de tarifs (lorsqu'elle sera disponible).</p>	<ul style="list-style-type: none"> <li>- L'ajout de l'onglet "Tarifs" permettra aux internautes d'accéder facilement aux informations de tarification.</li> <li>- En attendant l'URL de Qwenta, le lien pourra être ajouté ultérieurement.</li> </ul>
<p>Export Deliveroo :</p> <p>Les restaurateurs doivent pouvoir exporter leur menu vers l'application Deliveroo en un clic.</p>	<p>N/A</p>	<p>Node.js pour le backend</p> <p>React pour le frontend</p>	<p>Ajout d'un bouton d'exportation vers Deliveroo sur la page du menu, qui permettra aux restaurateurs d'exporter leur menu vers l'application Deliveroo.</p>	<ul style="list-style-type: none"> <li>- Node.js permettra de gérer les requêtes backend vers l'API publique de Deliveroo pour l'exportation du menu.</li> <li>- React facilitera l'interaction avec le bouton d'exportation vers Deliveroo.</li> </ul>
<p>Partage Instagram :</p> <p>Les restaurateurs doivent pouvoir partager leur menu sur Instagram facilement.</p>	<p>N/A</p>	<p>CSS et React</p>	<p>Conception d'une interface pour permettre aux restaurateurs de partager leur menu sur Instagram avec leur logo et leur couleur de base.</p>	<ul style="list-style-type: none"> <li>- CSS permettra de personnaliser le branding du menu en fonction des choix du restaurateur.</li> <li>- React facilitera la gestion interactive du partage du menu sur Instagram.</li> </ul>
<p>Logout :</p> <p>Les restaurateurs doivent pouvoir se déconnecter de l'application.</p>	<p>N/A</p>	<p>Node.js pour le backend</p> <p>React pour le frontend</p>	<p>Mise en place d'une fonction de déconnexion pour permettre aux restaurateurs de se déconnecter de l'application en un clic.</p>	<ul style="list-style-type: none"> <li>- Node.js permettra de gérer les requêtes backend pour déconnecter l'utilisateur et vider la session.</li> <li>- React facilitera la gestion de la déconnexion côté frontend.</li> </ul>



Besoin	Contrainte	Solution	Description de la Solution	Justification (2 arguments)
<p>Modification des infos user :</p> <p>Les restaurateurs doivent pouvoir modifier leurs informations utilisateur.</p>	N/A	<p>Node.js pour le backend</p> <p>React pour le frontend</p>	<p>Conception d'une interface pour permettre aux restaurateurs de modifier leurs informations utilisateur, telles que leur nom, leur e-mail, etc.</p>	<p>- Node.js permettra de gérer les requêtes backend pour mettre à jour les informations utilisateur dans la base de données.</p> <p>- React facilitera l'affichage des informations actuelles et la gestion des modifications côté frontend.</p>
<p>Dashboard :</p> <p>Les restaurateurs doivent pouvoir accéder à un dashboard regroupant les fonctionnalités de création, diffusion et impression de menus, ainsi que les trois derniers articles de Qwenta parlant de MenuMaker.</p>	N/A	<p>Node.js pour le backend</p> <p>React pour le frontend</p>	<p>Mise en place d'une vue dashboard pour les restaurateurs, qui regroupera les fonctionnalités de création de menu, diffusion de menu, impression de menu et affichera les trois derniers articles de Qwenta concernant MenuMaker.</p>	<p>- Node.js permettra de récupérer les informations nécessaires depuis la base de données et les API de Qwenta.</p> <p>- React facilitera l'affichage des fonctionnalités du dashboard et des articles de Qwenta.</p>
<p>Branding :</p> <p>Les restaurateurs doivent pouvoir créer leur logo et définir la couleur de base de leur menu.</p>	N/A	CSS et React	<p>Conception d'une interface pour permettre aux restaurateurs de télécharger leur logo et de choisir la couleur de base de leur menu.</p>	<p>- CSS permettra de personnaliser le branding du menu en fonction des choix du restaurateur.</p> <p>- React facilitera l'interaction avec les options de branding.</p>



## II. Liens avec le back-end

---

- Quel langage pour le serveur ?

Pour le serveur, nous avons choisi d'utiliser Node.js. Node.js est un environnement d'exécution JavaScript côté serveur qui nous permet de construire des applications web évolutives et performantes. Il est basé sur le moteur JavaScript V8 de Chrome, ce qui lui confère une excellente vitesse d'exécution. De plus, Node.js est largement utilisé dans l'industrie et bénéficie d'une vaste communauté de développeurs, ce qui facilite l'accès à des ressources et des modules supplémentaires.

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

Oui, nous avons besoin d'une API pour permettre la communication entre le frontend et le backend de notre application. L'API servira de pont pour transférer les données entre les deux parties. Nous utiliserons une API RESTful pour cette tâche. Une API RESTful est un style d'architecture web qui utilise les méthodes HTTP standard pour effectuer des opérations CRUD (Create, Read, Update, Delete) sur les ressources. Cela nous permettra de rendre notre application plus modulaire et flexible, en séparant les responsabilités entre le frontend et le backend.

- + API Deliveroo
- + API Instagram



- Base de données choisie :

Pour la base de données, nous avons choisi PostgreSQL comme système de gestion de base de données. PostgreSQL est un système de base de données relationnel open source réputé pour sa stabilité, ses performances et sa compatibilité avec les normes SQL. Il offre des fonctionnalités avancées telles que les transactions ACID (Atomicity, Consistency, Isolation, Durability), les index, les vues, les procédures stockées, etc. PostgreSQL est également bien adapté aux applications web à fort trafic et est largement utilisé dans l'industrie pour des projets de grande envergure. En choisissant PostgreSQL, nous pourrions stocker de manière fiable les données des utilisateurs, des menus et autres informations essentielles pour le fonctionnement de notre application de menu maker.

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

---

- Nom du domaine :

Le nom de domaine que nous avons choisi pour notre site de menu maker est "qwenta-menumaker.com". Ce nom de domaine est représentatif de l'objectif de notre application, qui est de permettre aux restaurateurs de créer et personnaliser facilement leur menu.

La disponibilité du nom de domaine a été vérifiée.

- Nom de l'hébergement :





Pour l'hébergement de notre application, nous avons opté pour le service d'hébergement cloud "Amazon Web Services" (AWS). AWS offre une infrastructure hautement évolutive et fiable, adaptée aux applications web à forte demande de ressources. En utilisant AWS, nous pouvons facilement adapter les ressources de notre serveur en fonction de la charge du trafic, assurant ainsi des performances optimales pour nos utilisateurs.

- Adresses e-mail :

Nous utiliserons des adresses e-mail professionnelles pour notre équipe de développement et pour les échanges avec les clients et les utilisateurs. Exemples d' adresses e-mail que nous avons configurées pour notre projet :

- [scrum.master@qwenta-menumaker.com](mailto:scrum.master@qwenta-menumaker.com) : L'adresse e-mail du Scrum Master, responsable de la coordination de l'équipe et de la gestion du projet.
- [dev.front1@qwenta-menumaker.com](mailto:dev.front1@qwenta-menumaker.com) : L'adresse e-mail du développeur frontend 1.
- [dev.front2@qwenta-menumaker.com](mailto:dev.front2@qwenta-menumaker.com) : L'adresse e-mail du développeur frontend 2.
- [dev.back1@qwenta-menumaker.com](mailto:dev.back1@qwenta-menumaker.com) : L'adresse e-mail du développeur backend 1.
- [dev.back2@qwenta-menumaker.com](mailto:dev.back2@qwenta-menumaker.com) : L'adresse e-mail du développeur backend 2.

Ces adresses e-mail seront utilisées pour les communications internes de l'équipe et pour les échanges avec les clients et les utilisateurs concernant les problèmes techniques, les mises à jour du projet et tout autre sujet lié au développement de l'application



## IV. Accessibilité

---

- Compatibilité navigateur :

Nous nous assurerons que notre application de menu maker est compatible avec les principaux navigateurs modernes. Cela inclut, mais n'est pas limité à :

- Google Chrome
- Mozilla Firefox
- Apple Safari
- Microsoft Edge

Nous veillerons à tester et à optimiser l'application pour que les fonctionnalités et l'expérience utilisateur soient cohérentes et sans problème sur tous ces navigateurs. Nous prendrons en compte les différences de rendu et de comportement entre les navigateurs pour garantir une expérience utilisateur uniforme.

- Types d'appareils :

Nous concevrons notre application pour qu'elle soit réactive et adaptable à différents types d'appareils, afin d'offrir une expérience utilisateur optimale sur une large gamme de dispositifs. Cela comprendra :

- Ordinateurs de bureau et ordinateurs portables
- Tablettes
- Smartphones



Nous utiliserons des techniques de conception et de développement adaptatif (responsive design) pour que l'interface de l'application s'ajuste automatiquement en fonction de la taille de l'écran de l'appareil utilisé. Cela garantira que les utilisateurs peuvent accéder à l'application et utiliser toutes ses fonctionnalités, quelle que soit la taille de leur écran.

En assurant la compatibilité avec différents navigateurs et appareils, nous nous assurons que notre application de menu maker est accessible à un large public d'utilisateurs, améliorant ainsi l'expérience client et l'adoption de notre solution par les restaurateurs

## V. Services tiers

---

- Noms et justification de ceux qui devront être implémentés pour le bon fonctionnement et le suivi du site (par exemple les plugins):

*Outils de suivi et d'analyse des performances :*

- Google Analytics : Nous mettrons en place Google Analytics pour suivre les performances de notre site, analyser le trafic, les comportements des utilisateurs, et identifier les pages les plus visitées et les taux de conversion. Cela nous permettra de prendre des décisions basées sur des données concrètes et d'optimiser continuellement notre site pour améliorer l'expérience utilisateur.



*Outil de sauvegarde automatique :*

Cron Jobs : Pour assurer la sécurité et la disponibilité de notre site, nous mettrons en place des tâches cron (Cron Jobs) pour effectuer des sauvegardes automatiques régulières de notre base de données et de notre code source. Ces sauvegardes seront stockées en toute sécurité pour permettre une restauration en cas de besoin.

*Outil de gestion de versions (version control) :*

Git : Pour assurer une gestion efficace du code source de notre projet, nous utiliserons Git, un système de contrôle de versions décentralisé. Git nous permettra de suivre les modifications du code, de travailler en collaboration en toute sécurité et de revenir à des versions précédentes en cas de besoin.

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

---

- Accès aux comptes, plugins :

Outil de sécurité :

Fail2Ban : Nous utiliserons l'outil Fail2Ban pour renforcer la sécurité de notre serveur. Il surveillera les journaux d'accès du serveur et bloquera automatiquement les adresses IP suspectes après un certain nombre de tentatives de connexion échouées, protégeant ainsi notre site contre les attaques par force brute.

## **VII. Maintenance du site et futures mises à jour**

---



- Grandes lignes du contrat de maintenance :

*Définition des services de maintenance :*

-Description des services de maintenance offerts, tels que la surveillance continue du site, la gestion des sauvegardes régulières, la résolution des problèmes de performance, les mises à jour de sécurité, etc.

*Durée du contrat :*

-Indication de la durée du contrat de maintenance, par exemple, une année renouvelable automatiquement, ou une période spécifique selon les besoins du client.

*Fréquence des interventions :*

-Précision de la fréquence à laquelle les différentes tâches de maintenance seront effectuées, par exemple, les sauvegardes hebdomadaires, les mises à jour mensuelles, etc.

*Responsabilités du prestataire de maintenance :*

-Détail des responsabilités du prestataire de maintenance, telles que la surveillance du site pour détecter les problèmes, la correction des bugs, la mise en place de correctifs de sécurité, etc.

*Responsabilités du client :*



- Précision des responsabilités du client, par exemple, la fourniture en temps voulu des informations d'identification nécessaires, la communication des problèmes rencontrés, etc.

*Support technique :*

- Indication des canaux de communication pour le support technique, par exemple, par e-mail, par téléphone, etc. ainsi que les horaires de disponibilité du support.

*Clause de résiliation :*

- Conditions dans lesquelles le contrat de maintenance peut être résilié, par exemple, en cas de non-paiement, d'insatisfaction du client ou de modification des besoins du projet.

*Tarification et modalités de paiement :*

- Tarifs pour les services de maintenance et modalités de paiement, par exemple, paiement mensuel, trimestriel ou annuel.

*Engagements de confidentialité et de sécurité :*

- Engagement du prestataire de maintenance à respecter la confidentialité des données du client et à mettre en place des mesures de sécurité pour protéger le site contre les intrusions.

*Clause de non-responsabilité :*

- Décharge de responsabilité du prestataire de maintenance pour tout dommage résultant de facteurs indépendants de sa volonté, tels que des pannes de serveur, des attaques de pirates informatiques, etc.