

# Spécifications techniques

Menu maker by Qwenta

Version	Auteur	Date	Approbation
1.0	Audrey	04/12/2024	Soufiane

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

**“Menu Maker by Qwenta** est une interface sur laquelle les restaurateurs peuvent se connecter pour créer leurs menus à diffuser en ligne ou à imprimer.”

## I. Choix technologiques

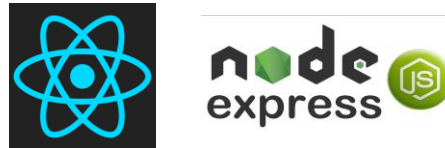
- 1-1 : État des lieux des besoins fonctionnels et de leurs solutions techniques

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
<b>Landing non connectée</b>	L'internaute doit pouvoir comprendre l'utilité de cette application	<i>Html, Scss</i>	La page landing non connectée sera conçue avec <u>HTML</u> pour une structure claire et <u>SCSS</u> pour une mise en page responsive et organisée, offrant une expérience utilisateur optimale et des appels à l'action efficaces	1-L'utilisation de <i>HTML</i> et <i>SCSS</i> permet de créer une page bien structurée et esthétiquement plaisante, avec une organisation claire du contenu 2- La séparation du contenu <i>HTML</i> et du style via <i>SCSS</i> permet une optimisation de la page, ce qui améliore le temps de chargement et l'expérience utilisateur
<b>Page login/Déconnexion</b>	Le restaurateur doit pouvoir créer un compte automatiquement avec mon adresse mail et ayant un compte, je veux pouvoir me connecter	<i>Auth0</i> + <i>MongoDB</i>	L'authentification sera gérée via <u>Auth0</u> , permettant au restaurateur de créer un compte en utilisant son adresse mail et de se connecter facilement avec un identifiant sécurisé. <u>MongoDB</u> stockera les informations utilisateur, permettant la gestion des comptes et de leurs données associées de manière sécurisée	1- <i>Auth0</i> offre une authentification sécurisée avec des options comme la validation en deux étapes, réduisant ainsi les risques de piratage 2- <i>Auth0</i> s'intègre facilement avec <i>MongoDB</i> , permettant une gestion des utilisateurs et une évolutivité sans contrainte
<b>Dashboard</b>	Le restaurateur doit pouvoir accéder à : la création de menu, la diffusion de menu, l'impression de menu Une fois le menu prêt, il est possible de : -l'exporter en	<i>React</i> + <i>Api Node.js</i> + <i>MongoDB</i>	La solution utilise <u>React</u> pour créer une interface utilisateur dynamique et interactive, permettant la gestion des menus et la consultation des articles du blog. Le backend est assuré par une <u>API Node.js</u> qui interagit avec	1- La séparation entre le frontend <i>React</i> et l' <i>API Node.js</i> permet de développer et d'étendre chaque partie de manière indépendante 2- <i>MongoDB</i> permet une grande flexibilité dans le stockage des données grâce à sa structure orientée

	PDF ; -le diffuser sur Deliveroo ; - le partager sur Instagram et il est également possible de l'imprimer		<u>MongoDB</u> pour stocker et récupérer les données liées aux menus et aux articles	documents, facilitant l'ajout et la mise à jour des informations liées aux menus sans contrainte de schéma fixe
<b>Catégorie de plat</b>	Le restaurateur inscrit doit pouvoir renseigner une catégorie de plat (par exemple, "Entrées", "Plats", etc.). Le restaurateur doit pouvoir ajouter un plat / des plats dans son menu	<i>React + Api Node.js + MongoDB</i>	La gestion des catégories de plat sera développée en <u>React</u> avec un formulaire interactif permettant aux restaurateurs d'ajouter et gérer les différentes catégories de manière dynamique. L' <u>API Node.js</u> interagira avec <u>MongoDB</u> pour effectuer des opérations <i>CRUD</i> (Créer, Lire, Mettre à jour, Supprimer) sur les catégories et menus, assurant ainsi une gestion flexible et efficace des données	1- <i>React</i> permet de créer des interfaces interactives et réactives, offrant aux restaurateurs une expérience fluide et intuitive pour gérer les catégories de plats 2- <i>MongoDB</i> est une base de données <i>NoSQL</i> idéale pour les structures de données variées comme les catégories de plats, permettant des mises à jour rapides et efficaces
<b>Création de plats/Tarifs</b>	Pour créer son menu, le restaurateur pourra ajouter : -le nom des plats qui le composent par catégorie (entrée, plat ou dessert) ; -le prix du plat ; -la description du plat	<i>React + Api Node.js + MongoDB</i>	Le restaurateur pourra utiliser une interface en <u>React</u> pour ajouter des plats par catégorie (entrée, plat, dessert) avec nom, prix et description. Les données seront gérées par une <u>API Node.js</u> qui interagira avec <i>MongoDB</i> pour stocker et récupérer les informations	1- <i>React</i> permet une gestion dynamique des plats et une interface fluide pour les utilisateurs 2- <i>MongoDB</i> offre une solution de stockage flexible et adaptée à la gestion des données en temps réel
<b>Ouverture des modales</b>	La création de catégorie s'ouvre dans une modale spécifique, et doit pouvoir être validée. Une modale s'ouvre pour saisir les différents plats de la catégorie	<i>React modal</i>	<u>React</u> modal est une bibliothèque ou un composant permettant la création de fenêtres modales (popups) dans une application <u>React</u>	1- Les fenêtres modales permettent d'afficher des informations ou des formulaires sans quitter la page, offrant ainsi une navigation sans interruptions 2- <i>React Modal</i> facilite la gestion des états d'ouverture et de fermeture des modales à travers des composants, rendant le code plus propre et facile à maintenir
<b>Stockages des images</b>	Le restaurateur doit pouvoir associer une image à un plat	<i>Cloudinary + MongoDB</i>	Une <u>API</u> comme <u>Cloudinary</u> sera intégrée pour gérer le stockage, l'optimisation et la transformation des images. Les fichiers seront	1- Optimisation automatique des images : Des <u>API</u> comme <i>Cloudinary</i> permettent de redimensionner, compresser et optimiser les images

			envoyés à l'API, qui générera des liens optimisés à stocker dans <u>MongoDB</u> pour un affichage performant sur le web	automatiquement pour différents appareils, assurant une performance optimale sur le web 2- Gestion avancée et flexible (recadrage, ajout de filtres) et le support des formats modernes
<b>Style de menu</b>	Le restaurateur connecté doit pouvoir personnaliser le style de son menu Les restaurateurs pourront : -enregistrer leurs préférences de branding, notamment pour enregistrer le logo du restaurant ; -choisir de façon dynamique la police et sa couleur	<i>React styled-components</i>	<u>React styled-components</u> permet de créer des composants <u>React</u> avec des styles <i>CSS dynamiques</i> .	1- <u>React styled-components</u> permet de gérer des styles dynamiques et réactifs, et de changer les styles du menu en fonction des préférences de l'utilisateur sans recharger la page 2- Le code reste propre et organisé, chaque style étant encapsulé dans un composant
<b>Branding restaurateur</b>	Le restaurateur doit pouvoir créer le branding de mon restaurant. Il doit pouvoir ajouter / modifier / supprimer le logo, couleurs de base	<i>React +Nodje.Js +MongoDB</i>	L'interface en <u>React</u> permettra au restaurateur de créer et personnaliser le branding de son restaurant en ajoutant, modifiant ou supprimant le logo et les couleurs de base. Une API en <u>Node.js</u> gèrera les requêtes pour stocker ces éléments dans la base de données <u>MongoDB</u>	1- La gestion en <i>React</i> permet une personnalisation en temps réel de l'identité visuelle du restaurant, offrant une expérience fluide. 2- <u>MongoDB</u> facilite la gestion des éléments visuels grâce à sa capacité à stocker différents types de données (textes, images).
<b>Exportation Pdf</b>	Le restaurateur doit pouvoir exporter son menu en PDF	<i>React-pdf</i>	<u>React-pdf</u> permet de générer des fichiers PDF en utilisant une syntaxe JSX, de manière similaire à la manière dont on écrit des composants <i>React</i>	1-L'utilisateur peut personnaliser le format du PDF en fonction de ses besoins (par exemple, ajouter des styles, des images, etc.) 2-La syntaxe JSX est intuitive pour les développeurs <i>React</i> , permettant de générer des documents directement à partir des composants <i>React</i>
<b>Commander l'impression d'un menu</b>	Le restaurateur doit pouvoir commander en un clic l'impression d'un menu	<i>Redirection via un lien</i>	Un lien est fourni vers le back-office de Qwenta pour l'impression du menu	1- Simplification du processus d'impression 2- L'utilisation du back-office permet à Qwenta de gérer efficacement les commandes d'impression

<b>Menus précédents</b>	Le restaurateur doit pouvoir avoir accès à une vue regroupant les menus créés précédemment	<i>React</i> + <i>Api Node.js</i> + <i>MongoDB</i>	L'interface en <u>React</u> permettra au restaurateur de visualiser ses menus précédemment créés. <u>Node.js</u> gèrera les requêtes <u>API</u> pour effectuer les opérations <u>CRUD</u> sur les menus stockés dans <u>MongoDB</u> , garantissant une gestion dynamique, avec modification et suppression	1- <i>React</i> offre une interface dynamique pour une gestion fluide des menus 2- <i>Node.js</i> et <i>MongoDB</i> assurent une manipulation rapide et flexible des menus
<b>Informations légales</b>	L'internaute doit pouvoir accéder au contenu "Mentions légales" dans une modale, et l'information "Tous droits réservés" doit être affichée.	<i>React</i>	En <u>React</u> , les mentions légales seront affichées dans une modale accessible au clic, et l'information "Tous droits réservés" sera affichée en bas de la page. Les tarifs de MenuMaker seront présentés dans une section dédiée	1- <i>React</i> permet d'afficher les mentions légales et les tarifs de manière dynamique sans rechargement de page 2- La modale et la section tarifaire permettent une consultation facile et rapide des informations importantes
<b>Exportation vers Deliveroo</b>	Le restaurateur doit pouvoir exporter son menu en un clic vers l'application Deliveroo	<i>Api Deliveroo</i>	Une intégration avec l' <u>API Deliveroo</u> permettra au restaurateur d'exporter son menu en un clic via l'interface en <u>React</u> , en utilisant <u>Node.js</u> pour transmettre les données au format attendu	1- Un export en un clic facilite la gestion des menus sur plusieurs plateformes 2- L'utilisation de l' <i>API Deliveroo</i> garantit une transmission fluide et conforme
<b>Partage sur Instagram</b>	Le restaurateur doit pouvoir partager facilement son menu sur Instagram	<i>Api Instagram</i> + <i>html-to-image</i>	Utiliser l' <u>API</u> de partage sur <i>Instagram</i> pour faciliter la publication des menus sous forme d'images ou de liens. <u>html-to-image</u> est une bibliothèque pour convertir la vue du menu en une image que le restaurateur pourra partager sur Instagram	1- L' <i>API Instagram</i> permet un partage direct depuis l'application, simplifiant la diffusion sur les réseaux sociaux. 2- Les bibliothèques comme <i>html-to-image</i> facilitent la création d'une image du menu, parfaitement adaptée pour le partage visuel sur Instagram
<b>Infos utilisateur</b>	Le restaurateur doit pouvoir modifier ses informations utilisateur	<i>React</i> + <i>Api Node.js</i> + <i>MongoDB</i>	Le restaurateur pourra modifier ses informations utilisateur directement depuis l'interface en <u>React</u> , à travers un formulaire sécurisé. L' <u>API Node.js</u> gèrera les modifications en effectuant les mises à jour dans la base de données <u>MongoDB</u>	1- L'interface en <i>React</i> permet une gestion simple et fluide des informations utilisateur 2- <i>Node.js</i> assure que les informations modifiées sont bien mises à jour dans la base de données de manière sécurisée et efficace



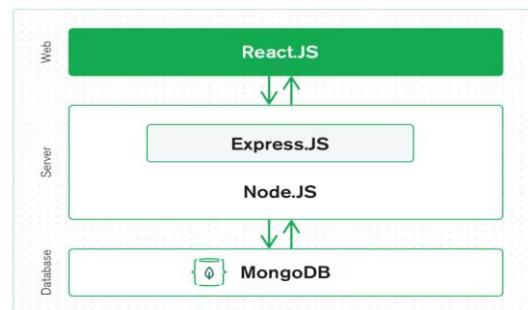
- 1-2 : Architecture du site

- **Front-end** : Le choix de React.js pour le front-end permet de construire des interfaces utilisateurs dynamiques et réactives, en facilitant la gestion des composants et en optimisant les performances grâce à un rendu rapide et efficace.

- **Back-end** : Le choix de Node.js avec Express pour le backend permet de créer des API performantes et scalables en utilisant JavaScript côté serveur, tout en bénéficiant d'un cadre léger et flexible pour gérer les requêtes HTTP, la gestion des routes et la communication avec la base de données.

- **Base de données** : MongoDB est une base de données NoSQL flexible qui stocke les données sous forme de documents JSON, idéale pour les applications nécessitant des mises à jour fréquentes et un stockage évolutif. Elle s'intègre facilement avec Node.js, offrant une gestion efficace des données.

- **Herbergement** : OVH a été choisi pour l'hébergement du site car il offre des solutions fiables, évolutives et adaptées aux besoins professionnels, avec un bon rapport qualité-prix et un support technique efficace.



## II. Liens avec le back-end

- Le langage choisi pour le serveur est Node.js grâce à sa rapidité, son modèle non-bloquant, et sa capacité à gérer des applications à fort trafic en utilisant JavaScript à la fois côté client et serveur.
- L'**API RESTful** avec Express.js permet de créer des services web rapides et efficaces en utilisant le framework Express sur Node.js, facilitant la gestion des requêtes HTTP, les routes et l'intégration avec une base de données.

- **MongoDB est une base de données NoSQL** qui stocke les données sous forme de documents JSON. Flexible, évolutive et bien adaptée pour des applications où la structure des données peut changer fréquemment, permettant un stockage rapide et l'accès aux informations sans nécessiter de schémas rigides.

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

- Le nom de domaine sera un sous-domaine de Qwenta.com : menu-maker.qwenta.com.
- OVH est un fournisseur européen d'hébergement fiable qui offre une gamme complète de services cloud.
- Il peut y avoir plusieurs possibilités pour le choix des adresses emails : [menumaker@qwenta.com](mailto:menumaker@qwenta.com), [info@qwenta.com](mailto:info@qwenta.com)...

### IV. Accessibilité

- Le site devra être en version desktop. Pas de version mobile à développer ni à prévoir.
- Le site doit être compatible avec les navigateurs récents tels que Chrome, Firefox et Safari.
- L'application devra être conçue de manière à être pleinement navigable à l'aide du clavier et à être compatible avec les lecteurs d'écran pour garantir son accessibilité.

### V. Recommandations en termes de sécurité

- L'Accès aux comptes avec l'authentification multi-facteurs (MFA) d'Auth0 pour garantir que seul l'utilisateur autorisé puisse accéder à son compte, même en cas de compromission du mot de passe. Auth0 propose également la gestion de sessions sécurisées, permettant de limiter les tentatives de connexion suspectes via des mécanismes comme la détection des connexions depuis des emplacements inhabituels.
- Pour la sécurité des plugins, tous les plugins utilisés dans l'application sont régulièrement mis à jour et proviennent de sources fiables pour éviter les vulnérabilités. Npm audit scanne et identifie les vulnérabilités dans les plugins npm d'une application Node.js, garantissant que les dépendances sont sûres et à jour.
- Pour la gestion des données sensibles, toutes les communications entre le client et le serveur utilisent HTTPS pour garantir la sécurité. Le chiffrement des informations sensibles des restaurateurs, telles que les données personnelles et financières, sera effectué à l'aide de AES pour garantir leur sécurité.

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

- Surveillance continue pour garantir la disponibilité et les performances du site. Suivi des temps de chargement et des erreurs.

- Résolution des bugs et des failles de sécurité. Application régulière des correctifs de sécurité pour prévenir les vulnérabilités.
- L'ajout et amélioration des fonctionnalités selon les besoins du client.
- Garantir l'adaptation du site aux différents types d'écrans (mobile, tablette, ordinateur) lors des futures mises à jour. Cette évolution vers un design responsive est prioritaire pour améliorer l'accessibilité et l'expérience utilisateur.
- Assistance pour résoudre les problèmes techniques et répondre aux questions des utilisateurs, avec un temps de réponse défini.
- Mise en place de sauvegardes automatiques pour sécuriser les données, avec la possibilité de restaurer rapidement le site en cas de besoin.
- Calendrier de mises à jour pour intégrer de nouvelles fonctionnalités et assurer la compatibilité avec les dernières technologies, notamment en rendant le site plus accessible sur les appareils mobiles (responsive).
- Il n'y aura pas d'outil à intégrer pour capter le comportement des utilisateurs pour le moment.