



Spécifications techniques

[Menu Maker, Qwenta]

Version	Auteur	Date	Approbation
1.0	Maria, Webgencia	14/11/2024	Sofiane

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)
<i>Landing page non connectée</i>	<i>L'utilisateur doit pouvoir comprendre ce qu'il peut faire avec cette application, avant de s'y lancer. La page doit contenir sections suivantes :</i> <ul style="list-style-type: none"> – Bannière – Personnalisez votre menu – Explication étape par étape 	<ul style="list-style-type: none"> • React 	<i>Utilisation de composants React</i>	<i>1) L'utilisation de composants modulaires et réutilisables engendre la facilité de maintenance, les performances optimisées et un écosystème riche.</i> <i>2) Il s'agit de la librairie populaire chez les développeurs..</i>
<i>Connexion / Création d'un compte</i>	<i>Fenêtre qui glisse sous forme de modale</i> <i>Connexion / création</i>	<ul style="list-style-type: none"> • Firebase Authentication 	<i>Fenêtre mises en œuvre par react-modal Firebase Authentication permet la connexion /</i>	<i>1) Firebase offre des fonctionnalités d'authentification et de création de comptes</i>

	<p><i>par e-mail</i></p> <p><i>E-mail de confirmation nécessaire pour connexion / création d</i></p> <p><i>Contact avec l'équipe de Menu maker au click du lien 'Besoin d'aide'</i></p>	<ul style="list-style-type: none"> <i>React-modal</i> 	<p><i>création du compte via l'adresse e-mail</i></p>	<p><i>sécurisées, notamment l'authentification avec des méthodes personnalisées, pour notre cas via la confirmation d'e-mail ou encore par réseaux sociaux (Google, Facebook, Twitter, etc.).</i></p> <p><i>2) React-modal permet une intégration facile respectant la maquette</i></p>
Créer un menu	<p><i>L'utilisateur doit pouvoir créer son menu avec l'ajout de plats pour lesquels il pourra configurer : la catégorie, le prix, son titre, description, ainsi qu'une image de chacun.</i></p>	<ul style="list-style-type: none"> <i>React-hook-form</i> 	<p><i>React Hook Form est une bibliothèque de gestion de formulaires basée sur les hooks de React. Elle permet de simplifier le processus de validation des formulaires et de gérer leur état de manière efficace.</i></p>	<p><i>1) React Hook Form encourage la réutilisation des composants en fournissant des hooks et des helpers qui facilitent la création de champs de formulaire réutilisables.</i></p> <p><i>2) L'utilisation des hooks permet de gérer l'état des éléments dans différents contextes</i></p>
Création d'une catégorie de menu	<p><i>L'ajout d'une catégorie doit pouvoir se faire directement sur l'écran de création de menu depuis une modale.</i></p>	<ul style="list-style-type: none"> <i>react-modal</i> <i>Express (Node.js)</i> 	<p><i>Cette librairie React permet de créer simplement des modales performantes, accessibles avec un minimum de code. On peut créer des routes et les outils nécessaires pour gérer la création de nouvelles catégories</i></p>	<p><i>1) Nous avons choisi de développer en React, la librairie est cohérente avec ce choix.</i></p> <p><i>2) La création de catégories en backend permettra au côté frontend de collecter et de transmettre les informations nécessaires.</i></p>

Personnaliser le menu	<p>L'utilisateur pourra :</p> <ul style="list-style-type: none"> _ Choisir une typographie _ En choisir la couleur _ Visualiser le menu dans le côté droit de l'écran pendant sa personnalisation 	<ul style="list-style-type: none"> • Styled-components 	Elle permet de créer des composants réutilisables avec des styles CSS encapsulés de manière dynamique.	<p>1) Évite les conflits de noms de classes CSS et permet une isolation des styles par rapport aux autres éléments de la page.</p> <p>2) Permet d'interagir avec les propriétés et les états du composant pour générer des styles dynamiques. Et offre donc une grande flexibilité dans la personnalisation et l'adaptation des styles en fonction des conditions et des interactions utilisateur.</p>
Exportation PDF et diffusion du menu	Pour finaliser la création du menu l'utilisateur doit pouvoir visualiser le résultat final et l'exporter en PDF, ou diffuser sur Deliveroo, ou le partager sur Instagram	<ul style="list-style-type: none"> • react-pdf • Deliveroo API • Instagram API 	<p>1) React-pdf permet de générer des fichiers PDF à partir de composants React. Elle facilite la création de documents PDF dynamiques et réactifs.</p> <p>2) Deliveroo API: offre des fonctionnalités pour intégrer directement les menus dans l'application.</p> <p>3) Instagram API: permet de partager automatiquement les menus sur l'application.</p>	<p>1) Assure une expérience utilisateur cohérente en proposant plusieurs solutions de partage.</p> <p>2) Réactivité en temps réel, toute modification sera prise en compte immédiatement et affichée sur la prévisualisation du menu.</p>
Impression de menus par QWENTA	L'utilisateur doit pouvoir commander	Redirection via un lien externe	Un lien fourni vers le back-office de QWENTA pour	1) Simplifie la commande d'impression en séparant les

	<i>l'impression du menu finalisé.</i>	<i>permettant d'avoir accès aux prix d'impression.</i>	<i>passer la commande</i>	<i>deux actions (création et impression).</i> 2) Externalise le processus complexe d'impression à la 'maison mère' spécialisée.
<i>Gestion des informations de l'utilisateur</i>	<i>L'utilisateur doit pouvoir consulter et modifier ses informations personnelles</i>	<ul style="list-style-type: none"> • React avec gestion des états • API Node.js et MongoDB pour les données 	<i>L'API permettra de récupérer et de mettre à jour les informations de l'utilisateur via MongoDB (e-mail(s), ou autre)</i>	1) React permet la gestion des informations de manière dynamique. 2) MongoDB stocke efficacement ces données et l'API de les mettre à jour.
<i>Menus précédents</i>	<i>L'utilisateur doit pouvoir visualiser, supprimer ou mettre à jour les menus qu'il aurait créé précédemment</i>	<ul style="list-style-type: none"> • React avec gestion des états • API Node.js et MongoDB pour les données 	<i>Utilisation de composants React pour afficher ces menus avec des outils de modification ou de suppression.</i> <i>Ces menus récupérés dynamiquement via des requêtes GET, PUT et DELETE dans MongoDB</i>	1) Gestion des menus facilitée avec React. 2) Mise à jour dynamique des menus sans rechargement de la page et flexibilité avec MongoDB. 3) Interaction fluide avec le Frontend via l'API Node.js.
<i>Déconnexion</i>	<i>L'utilisateur doit pouvoir se déconnecter à tout moment</i>	<ul style="list-style-type: none"> • Firebase Authentication 	<i>Met fin à la session de l'utilisateur et supprime le token d'authentification.</i>	1) Supprime l'accès aux fonctionnalités nécessitant une connexion. 2) Permet d'éviter tout accès non autorisé à ces fonctionnalités.

II. Liens avec le back-end

- Quel langage pour le serveur : *NodeJS.*(*Node.js est un environnement d'exécution JavaScript côté serveur, qui permet aux développeurs de créer des applications web rapides et évolutives.*)
- API : API RESTful avec Express.js pour gérer les requêtes du front vers back-end
- Base de données choisie : *NO SQL avec MongoDB.*(*Stockage de données flexible : MongoDB peut stocker des données de n'importe quelle structure et n'impose pas de schéma rigide, offrant ainsi une grande flexibilité pour gérer des données de nature diverse.*).

III. Préconisations concernant le domaine et l'hébergement

- Nom du domaine : www.menu-maker.com
Le nom sera probablement sous un domaine de QWENTA, en cours de validation
- Nom de l'hébergement : Firebase Hosting
Pour une intégration harmonieuse avec Firebase Authentication, Firebase Hosting serait une option naturelle.
- Adresses e-mail : contact@menu-maker.com

IV. Accessibilité

- Compatibilité navigateur : Chrome, Safari et Firefox pour le moment.
- Types d'appareils : *Le site devra être en version desktop. Pas de version mobile à développer ni à prévoir.*

- Accessibilité : Richesse en attributs d'accessibilité. L'application doit l'être au minimum depuis le clavier et lisible par un lecteur d'écran.

V. Recommandations en termes de sécurité

- *Méthode d'authentification à deux facteurs, contrainte au mot de passe fort et vérification adresse e-mail obligatoire*
- *Limitez les autorisations : Accordez uniquement les autorisations nécessaires aux utilisateurs pour éviter les abus ou l'accès non autorisé. Assurez-vous de respecter le principe du moindre privilège.*
- *S'assurer que le site utilise HTTPS pour protéger les données en transit et utilisez des certificats SSL valides pour garantir l'authenticité du site web*
- *Utilisez des plugins réputés et vérifiés : Choisissez des plugins provenant de sources fiables et faites des recherches sur leur réputation. Évitez les plugins obsolètes ou qui ne sont plus pris en charge, car ils pourraient présenter des failles de sécurité.*

VI. Maintenance du site et futures mises à jour

- *Mise à jour régulière du site pouvant inclure des correctifs de sécurité, des améliorations de performances et de nouvelles fonctionnalités (maintenance de veille technologique).*
- *Surveillance des performances du site afin de s'assurer qu'il fonctionne correctement (débugage) et que les temps de chargement sont acceptables.*
- *Sauvegardez régulièrement vos données : Effectuez des sauvegardes régulières de votre site web, de vos bases de données et de tous les fichiers importants. Ainsi, vous pourrez restaurer votre site en cas de problème ou de piratage.*