

Spécifications techniques

Menu Maker, Qwenta

Version	Auteur	Date	Approbation
1.0	Loubna	08-11-2024	Le Product Owner: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

I. Choix technologiques

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

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
Ex. : <i>Création d'une catégorie de menu</i>	Ex. : L'ajout d'une catégorie doit pouvoir se faire directement sur l'écran de création de menu depuis une modale.	Ex. : react-modal	Ex. : Cette librairie React permet de créer simplement des modales performantes, accessibles avec un minimum de code.	Ex. : 1) Nous avons choisi de développer en React, la librairie est cohérente avec ce choix. 2) Il s'agit de la librairie la plus utilisée.
<i>Ajouter un plat ou plusieurs dans le menu</i>	L'utilisateur doit pouvoir ajouter facilement un ou plusieurs plats depuis une interface intuitive et rapide.	react-hook-form	Cette librairie permet de gérer facilement les formulaires de manière performante et avec un minimum de code.	1) react-hook-form est compatible avec React, garantissant une intégration fluide. 2) Elle est légère et rapide, améliorant l'expérience utilisateur.
<i>Personnaliser le style de mon menu</i>	Les utilisateurs doivent pouvoir	SCSS	SCSS est une extension de CSS qui	1) SCSS permet une gestion plus efficace des

<p><i>Pouvoir en un clic télécharger le fichier PDF correspondant à son menu.</i></p>	<p>modifier les couleurs, polices et mises en page facilement via une interface dédiée.</p> <p>Le téléchargement doit être rapide, avec un fichier bien formaté et compatible avec tous les navigateurs.</p>	<p>jsPDF</p>	<p>permet d'utiliser des variables, des mixins et des fonctionnalités avancées pour des styles flexibles et maintenables.</p> <p>jsPDF est une bibliothèque JavaScript qui permet de générer des fichiers PDF dynamiquement depuis le frontend.</p>	<p>styles avec des fonctionnalités avancées comme les variables et les mixins.</p> <p>2) Il s'intègre facilement dans des projets React avec des fichiers modulaires (<code>.module.scss</code>) pour éviter les conflits de styles.</p> <p>1) jsPDF est léger et facile à intégrer dans un projet React.</p> <p>2) Il permet de personnaliser entièrement le contenu du PDF, y compris le style et la mise en page.</p>
<p>Implémentation d'une fonctionnalité de création et de connexion d'utilisateur sur le site</p>	<p>La création et la connexion doivent être sécurisées (gestion des mots de passe, validation des données) et simples à utiliser.</p>	<p>Passport.js</p>	<p>Passport.js est une bibliothèque Node.js qui permet de gérer l'authentification des utilisateurs avec diverses stratégies (locale, OAuth, JWT).</p>	<p>1) Passport.js offre une grande flexibilité et s'intègre parfaitement dans les projets Node.js/Express.</p> <p>2) Il prend en charge des stratégies avancées comme JWT, idéal pour les applications</p>

Pouvoir se déconnecter	La déconnexion doit être simple, sécurisée et invalider la session ou le token en cours.	JWT (JSON Web Token)	Lors de la déconnexion, le token JWT de l'utilisateur est supprimé côté client (par ex., depuis le localStorage) ou ajouté à une liste de révocation côté serveur.	modernes avec un backend sécurisé. 1) JWT offre une gestion sécurisée des sessions grâce à des tokens, faciles à invalider côté client ou serveur. 2) La suppression locale du token simplifie l'expérience utilisateur et maintient une bonne performance.
------------------------	--	----------------------	--	---

L'internaute doit pouvoir accéder aux différentes sections de la landing page sans être connecté	Les sections doivent être accessibles publiquement tout en respectant les bonnes pratiques de sécurité et de performance.	Public Routes (React Router)	Implémenter des routes publiques avec React Router pour afficher les sections de la landing page sans nécessiter d'authentification.	1) React Router permet de gérer facilement des routes publiques et privées, offrant une navigation fluide. 2) Les sections publiques augmentent l'accessibilité et attirent plus d'utilisateurs sans barrière initiale.
Accéder au contenu "Mentions légales" dans une modale, et l'information "Tous droits réservés" doit être affichée <i>Création d'une catégorie de menu</i>	La modale doit être accessible, responsive, et afficher clairement les mentions légales ainsi que "Tous droits réservés".	react-modal	Utiliser la bibliothèque react-modal pour afficher les mentions légales dans une modale facilement personnalisable.	1) react-modal est légère, facile à configurer, et respectueuse des bonnes pratiques d'accessibilité. 2) Elle permet une intégration rapide avec un design personnalisable et responsive.
avoir accès aux tarifs de MenuMaker.	Les tarifs doivent être bien présentés et attractifs pour les utilisateurs.	Grille de cartes avec CSS ou Tailwind	Utiliser une grille de cartes de tarifs stylisées qui affichent les plans, prix et avantages de manière conviviale et engageante.	1) Une grille de cartes rend les informations plus claires et engageantes visuellement.

--	--	--	--	--

II. Liens avec le back-end

- Quel langage pour le serveur ? *Ex: NodeJS.*
- A-t-on besoin d'une API ? Si oui laquelle ?
Oui, une API est nécessaire parce que l'application doit échanger des données avec d'autres services, plateformes, ou bases de données de manière efficace et sécurisée. Par exemple, une application mobile accédant aux données d'un serveur nécessite souvent une API pour gérer les requêtes des utilisateurs. On peut utiliser **API REST** de type API d'authentification parce qu'elle est la plus recommandée pour des services web standard, surtout si l'application nécessite une communication rapide, légère et compatible avec un grand nombre de clients.
- Base de données choisie : *Ex : SQL / NO SQL.*
On choisit une base de données **NoSQL (MongoDB)** pour sa capacité à gérer des données semi-structurées et évolutives, car les menus de restaurants varient beaucoup en structure. NoSQL nous permet de stocker des documents flexibles pour chaque menu, tout en assurant une scalabilité horizontale pour supporter des montées en charge importantes. Cela répond également au besoin de flexibilité de notre projet, où des

éléments peuvent être ajoutés ou supprimés des menus sans contrainte de structure rigide

III. Préconisations concernant le domaine et l'hébergement

- Nom du domaine: Pour ce projet, nous avons choisi un nom de domaine **menu-maker.Qwenta.com** car il est clair, professionnel, et représente bien l'objectif de l'application
- Nom de l'hébergement: Nous recommandons un hébergement avec **AWS (Amazon Web Services)**, car il offre une bonne flexibilité en termes de scalabilité et dispose de solutions de bases de données compatibles avec notre application. AWS garantit aussi des mesures de sécurité avancées, ce qui est essentiel pour les données sensibles des utilisateurs
- Adresses e-mail: Pour les adresses e-mail, nous proposons des adresses liées au domaine principal, comme **support@menu-maker.com** pour le service client et **contact@menu-maker.com** pour des questions générales. Cela assure une communication plus crédible et professionnelle avec les utilisateurs et partenaires.

IV. Accessibilité

- Compatibilité navigateur: Pour assurer une compatibilité optimale, l'application sera testée et optimisée pour les navigateurs **Google Chrome, Firefox, Safari, et Microsoft Edge**. Ces navigateurs couvrent la majorité des

utilisateurs

- Types d'appareils: L'application devra être en version desktop, donc elle doit s'adapter aux écrans d'ordinateurs.

V. Recommandations en termes de sécurité

- Tous les plugins et bibliothèques utilisés seront régulièrement mis à jour pour minimiser les vulnérabilités de sécurité. Nous n'utiliserons que des plugins de **sources fiables** et éviterons les plugins non maintenus.
- Une analyse de sécurité automatisée (comme **npm audit** pour les dépendances Node.js) sera effectuée régulièrement pour identifier et corriger les failles potentielles.
- Ainsi que toutes les communications entre le serveur et les utilisateurs seront chiffrées en **HTTPS**.
- Les données sensibles stockées dans la base de données (comme les mots de passe) seront **hachées** et toute autre donnée confidentielle sera chiffrée pour éviter tout accès non autorisé

VI. Maintenance du site et futures mises à jour

- Le contrat de maintenance inclura un service de suivi et de surveillance du site chaque 6 mois, pour assurer son bon fonctionnement.
- Nous assurerons le contrôle de la performance, la correction des bugs et l'assistance en cas de problème technique. Un rapport de maintenance sera fourni régulièrement pour informer des actions réalisées