

Spécifications techniques

Menu Maker - Qwenta

Version	Auteur	Date	Approbation
1.0	Pautonnier Sandra	17/01/2025	Soufiane

I. Choix technologiques	2
II. Liens avec le back-end	6
III. Préconisations concernant le domaine et l'hébergement	7
IV. Accessibilité	8
V. Recommandations en termes de sécurité	8
VI. Maintenance du site et futures mises à jour	9
VII. Conclusion	10

I. Choix technologiques

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

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
Landing non connectée	L'internaute doit pouvoir avoir accès aux différentes sections de la landing page non connectée.	React	Cette librairie React permet de créer simplement des composants réutilisables.	1) A l'avenir nous pourrions utiliser des fonctionnalités de React afin de rendre cette page d'accueil plus dynamique.2) Accessible avec un minimum de code.
Page login	L'ouverture de cette fenêtre se fait sous forme de modale.	React-modal	Création de modales performantes, accessibles avec un minimum de code.	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.
	La connexion doit être sécurisée.	JWT	JWT offre une authentification sécurisée.	1)Il protège l'intégrité des données et fonctionne avec des frameworks modernes comme React ou Node.js.2) Le refresh token renforcent la sécurité
Création d'une catégorie menu	L'ajout d'une catégorie doit pouvoir se faire directement sur l'écran de création de menu	React modal + API	Avec React nous pouvons créer une modale. Et via l'API pour créer ou afficher une catégorie.	1) Cohérente avec le reste de l'application 2) Simple à mettre en place

	depuis une modale.			
Création d'un menu	La création d'un menu se fait via la sélection d'une catégorie créé précédemment ou en créant la catégorie directement	React, formulaire + Redux Toolkit API	Utiliser Redux Toolkit pour gérer efficacement les états complexes liés aux catégories. API pour créer, modifier ou supprimer un menu.	1) Facilite la gestion des états globaux. 2) Réduit les bugs grâce à des outils intégrés comme "createSlice".
Style de menu	L'utilisateur doit pouvoir personnaliser le style de son menu en changeant la police et la couleur.	Redux Toolkit	Nous allons créer un reducer pour pouvoir récupérer la police choisie et la couleur.	1) Permet une personnalisation rapide et réactive. 2) Code rapide et simple.
Exportation PDF	Le PDF généré doit respecter la mise en page du menu personnalisé.	react-to-pdf	Cette bibliothèque facilite la création de documents PDF à partir de composants React en capturant le contenu du composant et en le convertissant en PDF.	1) react-to-pdf est une bibliothèque conçue spécifiquement pour les applications React. 2) Elle facilite la conversion de composants React en fichiers PDF en capturant le contenu du composant et en le transformant en PDF.
Commander l'impression d'un menu	L'utilisateur doit pouvoir imprimer son menu rapidement	window.print()	Permet d'imprimer directement une page web ou un PDF.	1) Permet de laisser l'utilisateur gérer l'impression via son application de lecteur PDF 2) Simple à implémenter.
Menus précédents	L'utilisateur doit pouvoir revenir sur son menu créé et son fichier pdf	Amazon S3 (AWS) MongoDB	Stockage des fichiers pdf sur un cloud et stockage du lien d'accès dans la base de données	1) Stockage léger des pdf 2) Possibilité de récupérer pour afficher ou télécharger les fichiers (avec les liens) via une requête API

Informations légales	L'internaute doit pouvoir cliquer sur "Mentions Légales" depuis les pages connectées et déconnectées ("Tous droits réservés" doit être affichée).	Static Content Footer	Ces informations doivent être affichées dans le footer comme sur la maquette de la page de création de menu lorsqu'on est connecté. Mais aussi en bas de page dans un footer lorsque l'on est pas connecté.	1) Facilité de gestion du contenu légal. 2) Contenu statique optimisé pour le SEO.
Tarifs	Sur le landing non connecté, l'utilisateur doit pouvoir cliquer sur "Tarifs" et ouvrir cet onglet	React components	Il faudra afficher dans cet onglet les plans des tarifs afin d'avoir pour l'utilisateur cette information claire et précise.	1) Simplifie la gestion des informations tarifaires. 2) Interface utilisateur claire et réactive.
Exportation Deliveroo	L'utilisateur doit pouvoir exporter son menu en un clic vers l'application Deliveroo.	Deliveroo API + OAuth	Utilisation de l'API Deliveroo pour automatiser l'envoi des pdf dans le format requis. Et utilisation de OAuth car l'utilisateur doit donner explicitement son autorisation pour importer son pdf.	1) OAuth est basé sur des jetons temporaires qui expirent, ce qui limite les risques en cas de fuite. 2) OAuth est un protocole standardisé conçu pour autoriser l'accès à des ressources sans partager de mots de passe.
Partage sur Instagram	Publication directe des menus sur Instagram avec une authentification sécurisée.	Instagram API + OAuth	OAuth 2.0 pour gérer l'authentification sécurisée et publier via l'API officielle d'Instagram.	1) Utilisation de normes de sécurité standard (OAuth 2.0). 2) Respect des guidelines d'Instagram pour éviter des restrictions.
Déconnexion	L'utilisateur doit pouvoir se déconnecter depuis n'importe quelle page	JWT	Le bouton "Se déconnecter" apparaît dans le footer pour pouvoir le faire. Invalidation	1) Assure une gestion sécurisée des sessions utilisateur. 2) Expérience

	connectée		des tokens JWT côté client et serveur.	utilisateur fluide lors de la déconnexion.
Infos utilisateur	Le restaurateur doit pouvoir modifier ses informations utilisateurs et lier plusieurs adresses e-mail à son compte.	API REST	Nous utiliserons simplement une API avec GET, POST et PUT . Pour le moment, nous allons faire une authentification par n'importe quel e-mail enregistré.	1) Cette solution est simple et est bien documentée ainsi l'utilisateur dispose d'une gestion simplifiée et d'une connexion flexible. 2) Peut utiliser l'e-mail qu'il préfère sans être contraint à un seul.
Dashboard	L'utilisateur doit pouvoir avoir accès à plusieurs éléments pour le Dashboard (La création de menu, La diffusion de menu, L'impression de menu, Les 3 derniers articles du blog de Qwenta qui parlent de MenuMaker). Mais aussi pour voir ou modifier ses infos utilisateurs ou bien se déconnecter)	React components	Affichage des différents éléments avec des composants réutilisables en y accédant via des boutons.	1) Visualisation claire et précise et expérience pertinente 2) Intégration rapide avec React pour des visualisations claires.
Branding restaurateur	Doit permettre la personnalisation du branding avec logo, couleurs, etc.	React avec userState et Redux Toolkit	Nous allons utiliser userState pour charger une image et choisir une couleur. Pour garder ces states, nous utiliserons Redux	1) Facilite la gestion des états globaux. 2) Permet une gestion simplifiée des ressources média.

II. Liens avec le back-end

- **Langage pour le serveur : Node.js**

- Il est performant, supporte de nombreux utilisateurs simultanément et dispose d'une large communauté.
- Compatible avec la stack JavaScript utilisée en front-end.

- **API :**

- **Express.js** - Framework léger et rapide pour Node.js, idéal pour construire l'API REST du projet :
 - Module authentification appel API :
 - POST : Création d'un compte via l'email
 - POST : Connexion de l'utilisateur
 - PUT : Modification de l'adresse e-mail
 - DELETE : Suppression d'un compte
 - Module restaurateur appel API :
 - POST : Ajout du nom du restaurant, d'un logo et choix d'une couleur
 - PUT : Modification du nom du restaurant, d'un logo et choix d'une couleur
 - GET : Récupération du nom du restaurant, d'un logo et choix d'une couleur
 - Module menu appel API :
 - POST : Création d'un nouveau menu
 - PUT : Modification d'un menu existant
 - GET : Récupération un menu existant
 - DELETE : Suppression d'un menu existant
- Instagram
- Deliveroo

- **Base de données choisie :**

- **MongoDB (NoSQL) :**

- Permet de stocker les données des menus et des utilisateurs dans un format JSON-like, ce qui simplifie les interactions avec le front-end.
 - Flexible pour gérer des données non structurées, comme des informations de menu personnalisées.

III. Préconisations concernant le domaine et l'hébergement

- **Nom du domaine :**

`menumaker/qwenta.com`

- **Hébergement :**

Vercel : Parfait pour les projets React avec Next.js avec déploiement rapide et optimisations intégrées.

- **Adresses e-mail :**

Nous utiliserons **Google Workspace** pour gérer les adresses e-mail professionnelles (ex. `support@qwenta.com`).

IV. Accessibilité

- **Compatibilité navigateur :**
 - Chrome, Safari, Firefox, Edge (versions récentes).
 - Nous pouvons réaliser des tests avec des outils comme **BrowserStack** pour valider la compatibilité.
- **Types d'appareils :**
 - **PC** : Nous allons nous concentrer dans un premier temps sur la version desktop.

V. Recommandations en termes de sécurité

- **Sécurisation des comptes :**
 - Utilisation de **JWT** pour les sessions : offre une authentification sécurisée et **stateless**, idéale pour les applications scalables. Grâce à sa signature, il protège l'intégrité des données et fonctionne avec des frameworks modernes comme React ou Node.js. Son expiration configurable et le **refresh token** renforcent la sécurité.
- **Protection des données :**
 - Utilisation de **HTTPS** via un certificat SSL pour sécuriser les communications : chiffre les échanges entre le client et le serveur, empêchant l'interception des données. Il garantit l'authenticité du site, renforce la confiance des utilisateurs et améliore le référencement SEO. Essentiel pour les **logins, paiements et API**.
- **Prévention des attaques :**
 - Suivi des vulnérabilités via des outils comme **Snyk** : permet de détecter et corriger les failles de sécurité dans le code et les dépendances. Il analyse en temps réel les bibliothèques utilisées, propose des correctifs et aide à prévenir les exploits. Essentiel pour sécuriser les applications web et API, il s'intègre aux workflows DevOps.

VI. Maintenance du site et futures mises à jour

- **Contrat de maintenance :**

- Suivi des mises à jour de sécurité tous les trimestres.
- Mise à jour régulière des dépendances via un outil comme **Dependabot**.
- Support utilisateur disponible par e-mail (ex. support@qwenta.com).

- **Futures évolutions :**

- Ajout de nouvelles fonctionnalités comme la personnalisation avancée des menus.
- Extension vers d'autres plateformes de livraison (ex. Uber Eats, Just Eat).
- Amélioration de l'accessibilité avec des tests utilisateurs réguliers.
- Créer un blog interne à Menu Maker
- Créer la possibilité de changer de moyen de paiement
- Sur la page d'accueil, ajouter des animations sur la photo de la bannière ainsi que sur les formes géométriques des sections.
- Proposer un compte maître pouvant accéder à toutes les modifications et au compte en lui-même, et plusieurs comptes où le nom du restaurant, les adresses emails ne pouvant pas être modifiées et où il peut seulement créer des menus.
- Orienter le restaurateur dès l'arrivée sur le site après la création du compte à choisir un logo et une couleur
- Dans l'état actuel l'utilisateur ne peut pas supprimer de catégorie lui-même.
- Actuellement, sur le "Dashboard", lorsqu'on clique sur "Diffuser un menu" ou "Imprimer un menu" cela ouvre un nouvel onglet, nous avons une autre possibilité d'ajouter ces fonctionnalités via des boutons dans la page "Mes Menus". Dans ce cas les encart Diffuser et Imprimer renverront sur "Mes Menus". Car l'utilisateur ne sait pas quel menu il peut sélectionner.

VII. Conclusion

- **Front-end :**
 - ↳ React : permet de créer des **composants réutilisables**, facilitant la maintenance, la modularité et la cohérence de l'interface tout en optimisant les performances.
 - ↳ Redux Toolkit : **centralise et synchronise les états globaux**, facilitant la gestion des menus, des utilisateurs et des préférences tout en assurant une mise à jour fluide de l'interface.
- **Back-end :**
 - ↳ Node.js avec Express.js : simplifie la gestion des requêtes et des routes.
- **Base de données :**
 - ↳ MongoDB (NoSQL) : gère efficacement des données variées et évolutives (menus, utilisateurs)
- **Hébergement :**
 - ↳ Vercel : Parfait pour les projets React avec déploiement rapide et optimisations intégrées.
- **Authentication :**
 - ↳ JWT : est idéal pour **sécuriser l'authentification**, protéger les routes API et gérer le Single Sign-On (SSO).
 - ↳ OAuth 2.0 (Instagram, Deliveroo) : est basé sur des jetons temporaires qui expirent, ce qui limite les risques en cas de fuite.
- **Stockage fichiers :**
 - ↳ Amazon S3 (AWS) : permet de stocker et gérer facilement les fichiers de menu et images de manière sécurisée et scalable.
- **Nom du domaine :**
 - ↳ menumaker/qwenta.com
- **Adresse e-mail :**
 - ↳ Google Workspace
- **Accessibilité :**
 - ↳ Types d'appareils : PC