

Spécifications techniques

[*Menu Maker by Qwenta /Qwenta*]

Version	Auteur	Date	Approbation
1.0	[Chawki]	[09/05/2024]	[Soufiane,Qwenta]

Sommaire

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)
Création de la page d'accueil	Le site devra être en version desktop. Il comprendra: - des modales - des formulaires - connexion/déconnexion - des redirections de page - des requêtes couleurs principales: - Beige: #FFF4E8 - Green: #8BC7B1 - Black: #000 - White: #FFF - Brown: #C5A073	1) HTML, CSS, SASS, JSX. 2) Maquette déjà créée.	HTML et CSS sont la structure de base du site web permettant d'afficher un visuel identique à la maquette proposée. SASS ajoute des fonctionnalités au CSS. JSX: composants React pour la structure du DOM, des modales, des formulaires et de l'interface utilisateur.	1) Intégrer un premier visuel du site en attendant la configuration du Back-end. 2) SASS permet de simplifier le code CSS et de faciliter sa mise à jour. 3) React pour créer le site dynamiquement et avoir une interface utilisateur interactive.
Création des modales: - modale: S'inscrire/ Se connecter. - modale: Création d'une Catégorie de menu.	La connexion au (futur) compte de l'utilisateur doit pouvoir se faire directement sur la page d'accueil depuis une modale. L'ajout d'une catégorie doit pouvoir se faire	React modal	Cette librairie React permet de créer simplement des modales performantes accessibles avec un minimum de code .	1) Librairie React permettant de créer des modales. 2) Répond aux exigences d'accessibilité du web moderne. 3) Librairie la plus utilisée.

	directement sur l'écran de création de menu depuis une modale.			
-modale: Informations légales	Doit reprendre le même style de site.	React modal	L'utilisateur peut accéder aux informations quand il est connecté ou déconnecté, en cliquant dessus une modale s'ouvre.	1) Librairie React permettant de créer des modales. 3) Librairie la plus utilisée.
<u>Menu :</u>	<u>Contraintes</u>	<u>Solution</u>	<u>Description de la solution</u>	<u>Justification (2 arguments)</u>
Créer un menu	Créer un formulaire pour récupérer les données saisies, les images par le restaurateur	React-formik	Cette librairie React facilite la gestion des formulaires .	1) Gère l'utilisation des formulaires 2) Gère la validation et la soumission des formulaires 3) Librairie la plus utilisée .
Personnaliser un menu	Les restaurateurs pourront : – enregistrer leurs préférences de branding. – logo du restaurant – choisir de façon dynamique la police et sa couleur .	React-formik API Développer de Google	Cette librairie permet de créer un formulaire pour enregistrer leurs préférences et leur logo L'API de Google pour avoir accès à toutes les polices d'écriture .	1) Gère l'utilisation des formulaires 2) Enregistre les données des utilisateurs 3) API Google pour gérer les polices .
Prévisualiser un menu	Lors de la création du menu, les restaurateurs	React-doc-viewer	Cette librairie React permet de visualiser le menu fait	1) Facile à utiliser 2) Peut lire les PDF, images, et

	pourront voir leur menu prendre forme .		par le restaurateur .	les documents aux formats Microsoft Office .
<i>Diffuser un menu</i>	Une fois le menu prêt, il est possible de : – l'exporter en PDF le diffuser sur Deliveroo – le partager sur Instagram .	API Instagram	connexion avec le compte du restaurateur .	Instagram permettant le partage sur ces plateformes .
<i>Imprimer un menu</i>	Une fois le menu prêt, il est possible de l'imprimer .	React-to-print	React-to-print est une librairie React qui permet d'imprimer les menus de manière simple et efficace .	1) Facile d'utilisation 2) Permet de personnaliser l'apparence d'impression .
<u>Arborescence du site :</u>	<u>Contraintes</u>	<u>Solution</u>	<u>Description de la solution</u>	<u>Justification (2 arguments)</u>
<i>Redirections des pages</i>	Créer des routes en utilisant le schéma d'arborescence des spécificités fonctionnelles .	React-router	L'utilisateur peut accéder aux informations légales quand il est connecté ou déconnecté en cliquant dessus, une modale s'ouvre .	1) Librairie React permettant de créer les modales 2) Librairie la plus utilisée
<i>Gestion des requêtes API</i>	Communiquer avec l'API .	Axios	Facilite la communication (requêtes) avec l'API .	1) Simplifie les requêtes HTTP entre le client et le serveur 2) Sécurise contre les attaques XRSF (requête falsifiée) .
<u>Back-end :</u>	<u>Contraintes</u>	<u>Solution</u>	<u>Description de la solution</u>	<u>Justification (2 arguments)</u>

Création de l'architecture de l'API	Création des routes Create = POST Read = GET Update = PUT Delete = DELETE	Express.js	La création de l'API facilitera la relation client / serveur	1) API claire et facile à comprendre. 2) Facilite la gestion des routes .
Sécuriser l'API	Protocoles de sécurité : – Validation par token – Cryptage des mots de passe – Surveillance des modifications de fichiers .	Express-jwt Bcryptjs Nodemon	– JWT sécurise avec la création de token d'authentification unique – Bcryptjs permet le cryptage des mots de passe – Nodemon redémarre automatiquement le serveur en cas de modification de fichiers .	1) Se prémunir des attaques 2) Éviter la fuite de données des utilisateurs .
Création de la base de données	Concevoir la structure de la base de données et l'optimiser .	MongoDB / Mongoose	Système de gestion de base de données NoSQL stockées sous la forme de documents JSON .	1) Permet l'ajout / la suppression / la mise à jour de données 2) Permet de stocker différents types de données dans un même document .
Mise en place de la gestion des données	– Gestion utilisateur (création des comptes, connexion) – Gestion de création des menus (création, modification, suppression)	Express.js	Express est un framework minimaliste : il permet la simplification de la logique de développement .	1) Simple et efficace dans la création des routes 2) Grande communauté pour aider en cas de problèmes .

	– Gestion des commandes d'impression .			
Aide au développement et documentation de l'API	Documenter les exemples de requêtes et de réponses .	Swagger UI	Permet de créer une interface documentée claire et détaillée de l'API.	1) Facile à comprendre 2) Permet aux développeurs de détecter les erreurs et les bugs dans le code de l'API .

II. Liens avec le back-end

- Quel langage pour le serveur ?

→ *Node.js et Express.js*

Node.js et Express.js son framework très bien documenté, permet de gérer les requêtes HTTP, de gagner du temps et de l'efficacité pour la création de serveur web.

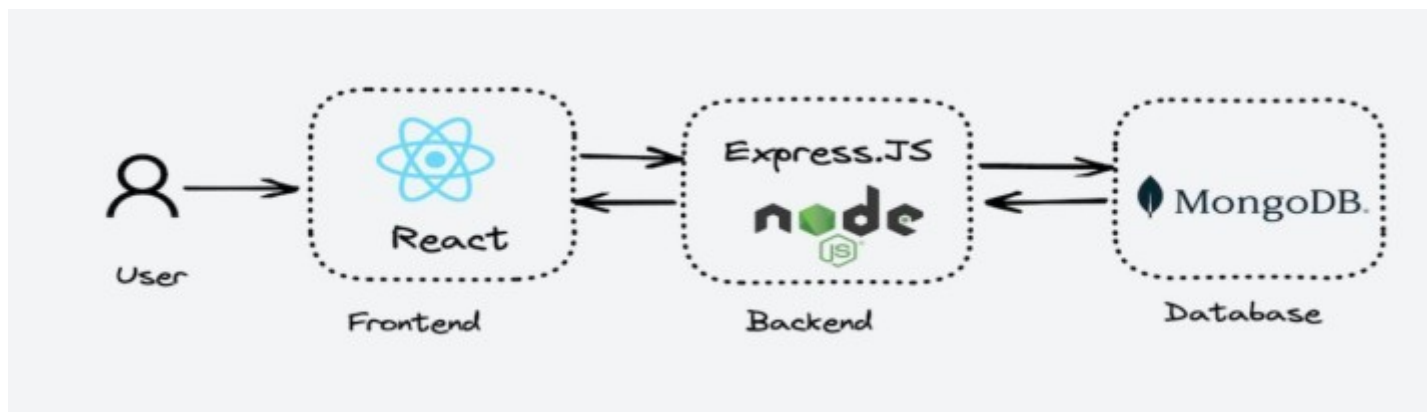
- A-t-on besoin d'une API ? Si oui laquelle ?

→ *API : oui*

→ *API REST* crée avec Node.js et Express.js. Son traitement non bloquant pour optimiser très rapidement les requêtes entrantes, sa performance et sa scalabilité. Son écosystème riche pour un développement plus rapide.

- Base de données choisie :

→ *MongoDB* : Base de données NoSQL s'adaptant au langage JavaScript. Pas de requêtes SQL pour récupérer les données.



Offre des performances élevées, une grande flexibilité de stockage de données et une évolutivité pour les futurs besoin du site web.

III. Préconisations concernant le domaine et l'hébergement

- **Nom du domaine.**

→ <https://www.menu-maker.qwenta.fr>

Ce choix est fait en fonction du nom du projet de Qwenta.

- **Nom de l'hébergement.**

→ **LWS**

Ce choix s'est fait par rapport aux derniers avis de différents sites spécialisés dans la technologie. LWS ressort en tête des hébergeurs grâce notamment à ses prix très attractifs.

- **Adresses e-mail.**

→ help.menu-maker@qwenta.fr

Contact lorsque l'utilisateur a besoin d'aide pour s'inscrire/se connecter.

→ contact.menu-maker@qwenta.fr

Contact lorsqu'il y a un problème sur le site.

IV. Accessibilité

- **Compatibilité navigateur:**

→ Firefox

→ Chrome

→ Safari

- **Types d'appareils :**

→ Ordinateurs

- **Autres types d'accessibilité :**

→ Navigable au clavier

→ Lisible par un lecteur d'écran

V. *Recommandations en termes de sécurité*

- **Sécurité du site :**

- Utilisation de certificats SSL

- Permet une connexion chiffrée entre l'utilisateur et le site.

- L'URL commencera donc par HTTPS : « S » signifie Secure. Permet au visiteur de vérifier l'identité du site auquel il accède, grâce à un certificat d'authentification. Garantit la confidentialité et l'intégrité des données envoyées par l'utilisateur et reçues par le serveur.

- Sauvegardes régulières des données

- **Accès aux comptes :**

- Mot de passe doit être fort et unique pour chaque utilisateur.

- Double authentification permet de vérifier l'identité après saisi du mot de passe, par une application tel que Google Authenticator par exemple, ou par SMS.

- Plugin mongoose-unique-validator Permet d'empêcher la création d'un autre compte en utilisant la même adresse mail

- **Sécurité API:**

- Le token d'authentification

- Création d'un jeton d'authentification unique lors de la connexion de l'utilisateur. Le jeton est supprimé lorsque l'utilisateur se déconnecte et possède une durée limitée.

- Plugin Bcryptjs

- Permet de crypter les mots de passe.

- Plugin Nodemon

- Permet un redémarrage automatique du serveur et la surveillance des fichiers.
 - RGPD (Règlement général sur la protection des données)

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

- ***Grandes lignes du contrat de maintenance :***

- 1. Entre les soussignés :**

- Informations légales concernant le client Qwenta.

- Informations légales concernant le prestataire Webgencia.

2. Objet de la prestation :

→ Maintenance du site, du serveur associé et de la base de données Menu Maker par Qwenta :

- Assurer les mises à jour nécessaires.
- Remédier à toute anomalie de fonctionnement du site, du serveur associé et de la base de données.
- Sécuriser le site contre le piratage et limiter les bugs.
- Sauvegarder régulièrement les données.
- Apporter son assistance au client en cas d'incident.
- Signaler au client toute modification majeure apportée au site, au serveur associé, ou à la base de donnée.

3. Prestations non incluses :

- Reconstitution de tout fichier détruit accidentellement.
- Entretien d'un site ou d'un serveur autres que ceux visés par le présent contrat.
- Sauvegarde d'un fichier quelconque .

4. Obligations du prestataire :

Webgencia reconnaît que **Qwenta** lui a donné les informations complètes sur ses besoins et sur les impératifs à respecter. Il s'engage à observer la confidentialité la plus totale en ce qui concerne le contenu de la mission et tous les documents/informations fournis que le client lui aura communiqués. Sa responsabilité pourra être engagée s'il est établi qu'il a manqué à son obligation contractuelle. En revanche, elle ne pourra être engagée en cas de retard résultant d'une cause indépendante ou si le client omet de lui transmettre une information nécessaire pour la mission.

5. Obligations du client :

Afin de permettre à **Webgencia** de réaliser la mission dans de bonnes conditions, **Qwenta** s'engage à lui remettre tous les documents nécessaires dans les meilleurs délais, et est tenu de laisser les collaborateurs de **Webgencia** à accéder à la zone d'administration du site ou serveur, le cas échéant.

6. Droit applicable et juridiction compétente :

Le présent contrat est assujéti au droit français. Tout litige qui résulterait de son exécution sera soumis aux tribunaux dont dépend le siège social du prestataire.

7. Évolutivité du site :

Pour toutes demandes concernant l'ajout de nouvelles fonctionnalités à l'application, des frais supplémentaires seront appliqués.

8 . Date et signatures des parties :