



Webgencia

Spécifications techniques

Projet	MENU MAKER by Qwenta
--------	----------------------

Version	Auteur	Date	Approbation
1.0	Bernard , Webgencia	17/02/2024	John, chef de projet chez Qwenta

Le but de ce document est de définir et justifier les spécifications techniques du site “Menu Maker” by Qwenta.

Il s’agit d’un outil en ligne permettant aux restaurateurs d’afficher, de personnaliser et d’imprimer leurs menus facilement, en quelques clics.

Qwenta a fait appel à Webgencia pour organiser la gestion de ce projet. La solution retenue dans ce document semble être le meilleur compromis possible. Cette solution sera soumise à l'approbation de John, chef de projet chez Qwenta.

Ce document comporte 7 rubriques

- 1) Les choix technologiques.
- 2) Les liens avec le back-end.
- 3) Les préconisations concernant le domaine et l'hébergement.
- 4) L'accessibilité.
- 5) Les services tiers.
- 6) Les recommandations en termes de sécurité.
- 7) La maintenance du site et les futures mises à jour.



I. Choix technologiques

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

L'utilisation de HTML5, CSS3, et JAVASCRIPT sera la base utilisée pendant toute la phase de développement. Les développeurs seront aussi tenus d'utiliser Visual Studio Code comme éditeur de code, Git comme système de contrôle de version, et GitHub comme service d'hébergement et de gestion des dépôts distants. Les tests de validation W3C seront systématiques pour les fichiers HTML et CSS. Les fonctionnalités seront testées par les développeurs eux-mêmes. Le Scrum Master devra les valider avant mise en production.

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
Toutes les pages du site. Landing page	<i>Respect des critères de la maquette Figma. Utilisation du HTML 5, du CSS 3, de Javascript. Validation du code par des tests W3C. Le code devra être suffisamment commenté.</i>	<i>React</i>	<i>React est une bibliothèque Javascript qui permet de créer des interfaces utilisateurs interactives et dynamiques. Il est possible de rajouter des animations plus tard.</i>	<i>1) React permet d'assembler de multiples composants pour former une interface utilisateur rapide. 2) Très populaire, React bénéficie d'un support continu, du fait d'une importante communauté de développeurs.</i>
Mentions légales	<i>Les mentions légales s'ouvrent dans une fenêtre modale. L'information "Tous droits réservés" doit être affichée.</i>	<i>React-modal</i>	<i>Cette librairie React permet de créer simplement des modales performantes, accessibles avec un minimum de code.</i>	<i>1) Nous avons choisi de développer en React, cette librairie est donc cohérente avec ce choix. 2) Il s'agit de la librairie la plus utilisée.</i>



Tarifs	<i>Un clic sur le lien "Tarifs" ouvre un nouvel onglet contenant les tarifs.</i>	<i>React</i>	<i>React permet de créer des interfaces utilisateurs. Cette nouvelle page sera créée sur une URL de Qwenta, à définir.</i>	<p>1) On utilise déjà React.</p> <p>2) Les tarifs seront facilement modifiables dans le temps.</p>
Identification de l'utilisateur <i>(les adresses mails seront stockées en base de données, voir plus bas)</i>	<i>L'identification doit se faire dans une fenêtre modale</i> <i>Création de compte et identification de manière sécurisée, par e-mail</i>	<i>React-modal</i> <i>Passport.js</i>	<i>Cette librairie React permet de créer simplement des modales performantes, accessibles avec un minimum de code.</i> <i>Passport.js est un middleware qui facilite la mise en œuvre de l'authentification utilisant E-mail Magic Link (Connexion uniquement par une adresse mail, sans mot de passe)</i>	<p>1) Nous avons choisi de développer en React, la librairie est cohérente avec ce choix.</p> <p>2) Il s'agit de la librairie la plus utilisée.</p> <p>1) Passport.js est compatible avec notre projet.</p> <p>2) Son utilisation est simple et de nombreux tutoriels sont disponibles</p>
Envoi et réception automatisée de mail	<i>Après avoir renseigné son adresse e-mail, l'utilisateur doit recevoir un mail contenant le lien de connexion valide. Possibilité d'avoir une assistance utilisateur.</i>	<i>Nodemailer</i>	<i>Nodemailer est un module permettant l'envoi automatisé de mails, en toute sécurité. Sa mise en place est assez simple et rapide.</i>	<p>1) Nodemailer est compatible avec notre projet, pour envoyer des mails de façon automatisée et sécurisée.</p> <p>2) Nodemailer est couramment utilisé, depuis de nombreuses années, et de façon sécurisée. Son support est assuré.</p>



Création d'une catégorie de plats <i>(les catégories seront stockées en base de données, voir plus bas)</i>	<i>L'ajout d'une catégorie se fait dans une modale spécifique. Le résultat doit apparaître dans le menu.</i>	<i>React-modal</i>	<i>Cette librairie React permet de créer simplement des modales performantes, accessibles avec un minimum de code. On peut réutiliser des composants déjà créés.</i>	<i>1) Déjà utilisée sur ce projet, cette librairie est cohérente avec nos choix.</i> <i>2) Il s'agit de la librairie la plus utilisée.</i>
Création de plats <i>(les plats seront stockés en base de données, voir plus bas)</i>	<i>L'ajout de plats se fait dans une modale. Après avoir choisi la catégorie, l'utilisateur peut composer son plat avec : une photo, un nom, un prix, une description. L'utilisateur peut scroller pour voir les informations affichées dans chaque plat.</i>	<i>React-modal</i> <i>React Hook Form</i>	<i>La librairie React-modal permet de créer simplement des modales performantes, accessibles avec un minimum de code.</i> <i>React Hook Form est une bibliothèque qui permet de créer facilement un formulaire avec importation d'image et comportant plusieurs champs.</i>	<i>1) Déjà utilisée sur ce projet, cette librairie est cohérente avec nos choix.</i> <i>2) Il s'agit de la librairie la plus utilisée.</i> <i>1) Cette bibliothèque React est d'une utilisation simple.</i> <i>2) Des règles de validation sont intégrées dans les formulaires.</i>
Personnalisation du style de menu <i>(les polices et couleurs seront stockées en base de données, voir plus bas)</i>	<i>L'utilisateur doit pouvoir sélectionner une typographie et choisir une couleur de texte, tout en visualisant le menu choisi.</i>	<i>Material UI</i>	<i>Material UI est une bibliothèque qui implémente le Material Design de Google. Cela permet de gérer les polices et les couleurs de manière simple et conviviale.</i>	<i>1) Cette bibliothèque de composants est compatible avec le projet, et profite d'une grande communauté d'utilisateurs.</i> <i>2) On peut créer son propre kit de composants, ou personnaliser un des nombreux kits existants.</i>



Exportation d'un menu en PDF	<i>Une fois le menu créé, le restaurateur doit pouvoir le transformer en fichier PDF et le télécharger, en un clic.</i>	<i>JsPDF</i>	<i>JsPDF est une bibliothèque open-source pour générer des fichiers PDF (en JavaScript côté client). Elle permet à l'utilisateur de télécharger facilement le fichier PDF créé.</i>	<i>1) Compatible avec notre projet, JsPDF est largement utilisé, depuis de nombreuses années, et possède une grande communauté d'utilisateurs. 2) L'exécution "côté client" assure une rapidité d'affichage et de téléchargement. Les fichiers créés respectent les normes PDF en vigueur.</i>
Partage sur Instagram	<i>L'utilisateur doit pouvoir partager facilement son menu sur Instagram. Il peut se connecter en indiquant ses identifiants personnels d'Instagram ou de Facebook. Ouvrir Instagram dans un nouvel onglet. Il faudra redimensionner les images au format "carré" imposé par Instagram.</i>	<i>React Image File Resizer API Graph d'Instagram</i>	<i>React-image-file-resizer est un module React qui permet de redimensionner les images locales. L' API Graph d'Instagram permet d'obtenir l'autorisation d'accès à Instagram, de diffuser des contenus, et répondre aux commentaires.</i>	<i>1) Gratuit et entièrement compatible avec notre projet. 2) Son utilisation est simple. De nombreux tutoriels existent 1) Fournie par Meta. Sécurité et support assurés. 2) Mise en place compliquée, mais obligatoire pour avoir accès à Instagram. Il faudra créer un compte développeur d'Instagram.</i>



Exportation Deliveroo	<i>L'utilisateur doit pouvoir exporter son menu vers l'application Deliveroo, en un clic. Ouvrir Deliveroo dans un nouvel onglet.</i>	<i>API de Deliveroo : Partner Platform Suite</i>	<i>Partner Platform Suite, de Deliveroo. C'est un ensemble d'API essentielles pour gérer un restaurant sur Deliveroo.</i>	<p><i>1) Deliveroo fournit de quoi gérer ses menus, commandes, livraisons et facturations, de manière sécurisée.</i></p> <p><i>2) Cette API dispose d'une documentation fournie, d'un support, et de mises à jour régulières</i></p>
Commander des impressions du menu	<i>Une fois le menu créé, l'utilisateur doit pouvoir commander des impressions professionnelles du menu.</i>	<i>Nouvel onglet, qui renvoie vers la partie Back Office de Qwenta.</i>	<i>Modalités et tarifs des impressions de Qwenta.</i>	<p><i>1) A étudier avec l'équipe commerciale de Qwenta.</i></p> <p><i>2) Prévoir de nouvelles spécifications techniques.</i></p>
Déconnexion	<i>L'utilisateur peut se déconnecter sur n'importe quelle page du site.</i>	<i>React</i>	<i>Déconnexion et retour sur la page d'accueil.</i>	<p><i>1) Déconnexion simple.</i></p> <p><i>2) Rester sur le site.</i></p>
Dashboard	<i>L'utilisateur souhaite avoir accès à un dashboard qui regroupe la création, la diffusion et l'impression des menus, ainsi que les 3 derniers articles du blog de Qwenta qui parlent de MenuMaker</i>	<i>React</i>	<i>React est une bibliothèque Javascript qui permet de créer des interfaces utilisateurs interactives et dynamiques. Il est possible de rajouter des animations plus tard.</i>	<p><i>1) React permet d'assembler de multiples composants pour former une interface utilisateur rapide, sans rafraichissement de la page</i></p> <p><i>2) React bénéficie d'un support continu, du fait d'une importante communauté de développeurs.</i></p>



Infos utilisateur, Branding et accès aux menus créés précédemment.	<i>L'utilisateur doit pouvoir gérer ses informations, adresses e-mail et le Branding de son restaurant (Logo, couleur de base). Il doit avoir accès aux menus créés précédemment.</i>	<i>React pour la partie Front-end ExpressJS, NodeJS, et MongoDB pour la partie Back-end</i>	<i>Il faudra créer une API REST, qui permettra à l'interface utilisateur de communiquer facilement avec la base de données. Ainsi toutes les informations pourront être stockées, et récupérées facilement et rapidement.</i>	<i>1) Cette combinaison de fonctionnalités est complètement compatible avec le projet. 2) Toutes ces technologies ont la particularité d'utiliser Javascript.</i>
Base de données	<i>Stockage de données : - des adresses mail, - des restaurateurs, - des menus, - des catégories, - des plats, - des logos, - des polices, - des couleurs de base.</i>	<i>MongoDB</i>	<i>Stockage des données avec MongoDB : une base de données NoSQL, orientée documents et collections, qui peut utiliser des données structurées ou non, au format BSON (Similaire au JSON, couramment utilisé.)</i>	<i>1) Mongo DB est disponible en version gratuite, est adaptable, flexible, rapide et simple à utiliser. 2) Complètement compatible avec le projet , cette base de données permet de conserver et restituer les données.</i>
Liaison entre Front-end et Back-end	<i>Pour assurer la liaison entre l'interface et la base de données, on utilisera la stack MERN. (C'est un acronyme de MongoDB, ExpressJS, React et NodeJS).</i>	<i>MongoDB Express.JS React Node.js</i>	<i>React peut récupérer et stocker les données dans la base de données de MongoDB. Pour cela il passe par un serveur géré par Node. Express gère les requêtes http. La création d'une Api facilite cette liaison.</i>	<i>1) Ces 4 éléments forment la Stack MERN, qui est très connue par la communauté des développeurs 2) Cet ensemble permet d'utiliser Javascript à la fois pour le Front-end et pour le Back-end.</i>



II. Liens avec le back-end

- Quel langage pour le serveur ? *Le langage choisi est NodeJS. C'est l'environnement le plus connu et le plus populaire . Il utilise Javascript, est Open source, permet d'accéder à la fois au Front-end et au Back-end, est documenté par une grande communauté.*
- A-t-on besoin d'une API ? *Oui. Il faudra créer une API REST avec l'aide de Node.js, Express.js et MongoDB.*
- Base de données choisie : *MongoDb, une base de données NoSQL, orientée documents, qui utilise le format de données BSON.*

III. Préconisations concernant le domaine et l'hébergement

- Nom du domaine: *Il faut prévoir un sous-domaine de Qwenta, en cours de validation actuellement. (MenuMaker.Qwenta.com)*
- Nom de l'hébergement: *<https://hostinger.com> C'est un hébergeur ayant fait ses preuves, assurant une rapidité et une sécurité maximale. Les tarifs sont corrects (Entre 12 et 20 euros par mois, selon l'option choisie). De nombreux services sont disponibles.*
- Adresses e-mail: *On pourrait envisager : supportMenuMaker@qwenta.com*

IV. Accessibilité

- Compatibilité navigateur: *Dernières versions de Chrome, Safari et Firefox.*
- Types d'appareils: *Ordinateurs (Desktop) uniquement.*
- *Pas de version mobile prévue.*



V. Services tiers

- *Nous aurons besoin de l' API Partner Platform Suite, de Deliveroo, pour pouvoir diffuser un menu créé, sur le site Deliveroo.*
- *Nous aurons besoin de l' API Graph d'Instagram, pour pouvoir partager un menu sur Instagram.*
- *Dans l'avenir, on pourra envisager d'utiliser des services de paiement en ligne.*

VI. Recommandations en termes de sécurité

A notre époque, la sécurité d'une application, ou d'un site web est primordiale.

Menu Maker bénéficie de toutes les options de sécurité connues actuellement :

- *Utilisation du protocole HTTPS pour chiffrer les données en transit, et éviter une interception des données.*
- *Cryptage des données dans la base de données MongoDB, pour éviter un piratage des données des utilisateurs.*
- *Système d'authentification renforcé avec Passport.js et E-mail Magic Link, pour éviter un piratage de compte utilisateur.*
- *Sécurisation des validations de formulaires, pour éviter l'injection de code malveillant.*
- *Sécurité supplémentaire fournie par l'hébergeur Hostinger (Certificat SSL, protection DDoS)*



VII. Maintenance du site et futures mises à jour

Afin de maintenir le bon fonctionnement du site, il est nécessaire de surveiller régulièrement le site:

- Prise en charge des problèmes lors du lancement du site,
 - Maintenance de la base de données,
 - Sauvegarde régulière du site,
 - Mises à jour de sécurité régulières,
 - Modification des fonctionnalités.
-
- *Un contact mensuel entre Qwenta et Webgencia est souhaitable.*
 - *Qwenta pourra évidemment signaler les bugs éventuels lorsqu'ils apparaissent.*
 - *Webgencia contactera Qwenta lorsque des mises à jour sont nécessaires.*
 - *L' utilisation de méthodes de développement AGILE permet justement de livrer, de manière continue et itérative, de nouvelles fonctionnalités et mises à jour.*
 - *Chaque changement devra être testé préalablement, pour assurer le bon fonctionnement et la sécurité du site.*
