

Spécifications techniques

Projet	Qwenta Menu Maker
--------	-------------------

Version	Auteur	Date	Approbation
1.0	Maud Curti	Fevrier 2024	Soufiane

Le but de ce document est de définir et justifier les spécifications techniques du Menu Maker de QWENTA .

I. Choix technologiques

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

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
Couleurs	Code de Couleurs deja définies dans les specifications fonctionnelles	Appliques les couleurs voulues par le client	Utiliser les couleurs choisies par le client {disponible dans les spécifications fonctionnelles page3} <u>Choix étudié en amont par le client ne peux pas être changer</u>	
Design	Le client a deja un design définie	Se rapprocher au plus du design voulu en utilisant CSS et Bootstrap	Faire en sorte que notre design ressemble au maximum {projet figma} <u>Choix étudié en amont par le client ne peux pas être changer</u>	



Webgencia

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
<i>Language</i>	<i>Doit fonctionner et être accessible a tout le monde</i>	<i>Javascript</i>	<i>Utilisation de JS et sa bibliothèque REACT pour la creation d'interfaces utilisateur interactive Node.js et son framework Express pour le développement</i>	<i>- Utilisation du meme langage pour le front et back end - Tres populaire</i>
<i>Connexion</i>	<i>Il faut un système sécurisé</i>	<i>Firebase Authentication.</i>	<i>Evolutif et fiable , personnalisable , gestion des utilisateurs , sécurité et facilite d'intégration</i>	<i>- rapidite de la mise en place du système de connexion - bien documente</i>
<i>Base de données</i>	<i>Coherent et adapte au besoins spécifique du site</i>	<i>MongoDB</i>	<i>base de données NoSQL, open-source, orientée document, conçue pour stocker, gérer et interroger des données de manière flexible et scalable. Elle utilise un modèle de données basé sur des documents JSON.</i>	<i>- Simple , modern et rapide - Capable de s'adapter aux besoins évolutifs des entreprises.</i>

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
<i>Creation des menus et catégorie (user Story)</i>	<i>Creation de formulaire qui permet a l'utilisateur de créer iun menu ou une catégorie</i>	<i>React formik</i>	<i>React Formik simplifie le processus de création et de gestion des formulaires, tout en s'intégrant facilement avec React et en étant hautement personnalisable.</i>	<i>- Gere et enregistre les preferences des formulaires des utilisateurs</i>
<i>Creation catégorie de menu , ajout des plats et page de connexion</i>	<i>Apparait en modal</i>	<i>react-modal</i>	<i>Cette librairie React permet de créer simplement des modales performantes, accessibles avec un minimum de code.</i>	<i>- Populaire pour la creation de modal - Integration facile et tres bien documenter pour simplifier son utilisation</i>
<i>Librairie CSS</i>	<i>Selection de police , couleur ,style ...</i>	<i>Bootstrap</i>	<i>Open Source, permet une personnalisation optimal du projet</i>	<i>- Classes pré-définie et hautement personnalisable - Populaire et bien documente</i>
<i>Exportation sur instagram</i>	<i>Au clique import du menu sur instagram</i>	<i>Api publique du site react-instagram-embed</i>	https://developers.facebook.com/docs/instagram-basic-display-api/overview?locale=fr_FR <i>- deja developper par meta - Gain de temps</i>	
<i>Exportation sur délivreroo</i>	<i>Au clique import du menu sur délivrer</i>	<i>Api publique du site</i>	https://api-docs.deliveroo.com/docs/introduction <i>- deja développe par deliveroo - gain de temps</i>	

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
<i>Systeme de mail</i>	<i>Pouvoir recevoir des mails pour le support technique , envoie automatique de mail pour la creation de compte</i>	<i>SendGrid</i>	<i>Permet de :</i> <ul style="list-style-type: none"> - reception des e-mail - envoi automatique des emails - Integration avec REACT 	<ul style="list-style-type: none"> - Personnalisable et évolutif - Bien documente
<i>Exportation en pdf</i>	<i>Au click exporte et telecharge le pdf</i>	<i>react PDF</i>	<i>Il est construit sur deux autres bibliothèques, Yoga et PDF Kit. Yoga fournit des capacités de mise en page et de style tandis que PDF Kit fournit des capacités de rendu PDF.</i>	<ul style="list-style-type: none"> - Facile à transmettre dans les styles pour créer des styles dynamiques - Intuitif pour ceux qui connaissent React DOM (et surtout React Native)
<i>Imprimer</i>	<i>Au clique imprime la selection</i>	<i>react-to-print</i>	<i>outil puissant pour simplifier le processus d'impression dans les applications web React</i>	<ul style="list-style-type: none"> - Facile d'utilisation - Hautement personnalisable



II. Liens avec le back-end

- Quel langage pour le serveur ? *Javascript avec son serveur nodeJs (et son framework express)*
- A-t-on besoin d'une API ? Si oui laquelle ? API REST en utilisant le framework express , nous utiliserons aussi les API des sites instagram et deliveroo .
- Base de données choisie :mongodb

III. Préconisations concernant le domaine et l'hébergement

- Nom du domaine : menu-maker.qwenta.com
- Nom de l'hébergement. VERCEL
- Adresses e-mail ➡ en cours de validation. :
 - menu-maker@qwenta.com : adresse lambda
 - help-menu-maker@qwenta.com : adresse pour le centre d'aide

IV. Accessibilité

- Compatibilité navigateur : Safari , Chrome , Firefox
- Appareils types Desktop : ordinateur de bureau , ordinateur portable , uniquement pour le moment.
- Navigable par clavier et lecture possible depuis un lecteur d'écran.



V. Services tiers

- Les API : sont trouvées sur les sites officiels , elles sont gratuite et libre d'accès
 1. instagram : <https://developers.facebook.com/products/instagram/apis/>
 2. deliveroo : <https://api-docs.deliveroo.com/reference/get-auth-verify#>
- GitHub -> gestion + codeReview + sauvegarde collaborative du code
- Notion que nous allons utiliser pour créer notre Kanban

VI. Recommandations en termes de sécurité

- Site HTTPS a demander le certificat SSL , certainement dispo via l'hébergement VERCEL
- Authentification sécurisée : mettre en place un mot de passe robuste , ou/et une double verification .
- Mises a jour régulières :pour intégrer de nouvelles fonctions et corriger les bugs
- Règlement général sur la protection des données (RGPD)

VII. Maintenance du site et futures mises à jour

- Grandes lignes du contrat de maintenance :
 1. **Définition des parties impliquées** : le client et le prestataire de services de maintenance(nom adresse numero de siret ...)
 2. **Objectifs du contrat** :le maintien de la fonctionnalité, la mise à jour du contenu, la résolution des problèmes techniques, etc.
 3. **Durée du contrat** : la période pendant laquelle le contrat de maintenance est en vigueur, y compris la date de début et la date de fin (si applicable).
 4. **Services inclus** : les services qui seront fournis dans le cadre du contrat de maintenance. Cela peut inclure des mises à jour régulières du logiciel, des correctifs de sécurité, des sauvegardes régulières, la surveillance du site web pour les pannes, etc.
 5. **Fréquence des services** : la fréquence à laquelle les différents services seront fournis. Par exemple, les mises à jour logicielles peuvent être effectuées mensuellement, les sauvegardes peuvent être effectuées quotidiennement, etc.
 6. **Responsabilités des parties** : Clarifiez les responsabilités de chaque partie impliquée dans le contrat. Par exemple, le client peut être responsable de fournir un accès sécurisé au site web, tandis que le prestataire de services est responsable de garantir que le site web reste opérationnel et sécurisé.
 7. **Procédures de communication** : Décrivez les canaux de communication à utiliser en cas de problèmes ou de questions liés à la maintenance du site web. Cela peut inclure des numéros de téléphone d'urgence, des adresses e-mail spécifiques, etc.
 8. **Frais et paiements** : Il s'agit des frais associés au contrat de maintenance, y compris le montant total, les modalités de paiement, les pénalités pour les paiements en retard, etc.
 9. **Termes de résiliation** : Les conditions dans lesquelles le contrat de maintenance peut être résilié par l'une ou l'autre des parties, y compris les préavis requis, les frais de résiliation, etc.
 10. **Clause de non-responsabilité** : la clause de non-responsabilité pour protéger les parties contre les pertes ou les dommages résultant de circonstances indépendantes de leur volonté, telles que des pannes de réseau, des attaques de sécurité, etc.
 11. **Date et signature**

