Spécifications techniques

|  |  |
| --- | --- |
| **Projet** | Menu Maker by Qwenta |

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Auteur** | **Date** | **Approbation** |
| 1.0 | Kellian, Webgencia | 13/02/2024 | John, Qwenta |

Le but de ce document est de définir et justifier les spécifications techniques de Qwenta.

1. **Choix technologiques**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Besoin** | **Contraintes** | **Solution** | **Description de la solution** | **Justification (2 arguments)** |
| Obtenir des informations (Fonctionnement, tarifs, mentions légales) | Les mentions légales s’ouvrent dans une modale | React-modal | React-modal est un composant de la librairie React, permettant de mettre en place des fenêtres modales accessibles | - React est une librairie très populaire et donc bénéficiant d’un large support.    - Ce composant permet de garder facilement l’accessibilité sur les modales. |
| Authentification | Identification via E-mail Assistance | Firebase authentification | Firebase authentification est un service de Firebase (google), gérant les identités des utilisateurs | - En externalisant l’authentification, on bénéficie d’un meilleur soutien en termes de sécurité et de maintenance.  - Le coût de cette solution est inexistant jusqu’à un certain volume. Au-delà de ça la facturation s’effectue au fur et à mesure, ce qui permet de gérer la croissance de l’activité sans stress financier. |
| Création de menus | Visualisation dynamique des modifications | React | La librairie React permet de créer des composants réutilisables et de gérer efficacement l’état de l’application. | - Il permet une mise à jour rapide et sans rafraîchissement de la page, ce qui rend efficacement l’état de l’application. l'expérience utilisateur plus fluide et agréable.  - La popularité de cette librairie lui fait bénéficier d’un support continu. |
| Upload d’images | Gestion du téléchargement et stockage des fichiers | Cloud storage for Firebase | Il s’agit du stockage de fichiers de l’architecture Firebase. | - Sécurité liée à l’authentification Firebase.  - La plateforme est soutenue par Google. |
| Accéder aux menus précédents, enregistrer un branding | Persistance des données | Firebase realtime database | Realtime database fait partie de la boite à outils firebase. Il s’agit d’une base de données non relationnelle en JSON. | - L’utilisation de cette base de données et des SDK Firebase, ne nécessite pas de serveur, ni de développer une API.  - Le stockage des données se fait en temps réel et est accessible rapidement. |
| Exporter en PDF | En un click | PDFKit (et Google Cloud Functions) | PDFKit est une bibliothèque open-source pour Node.js qui permet de créer des documents PDF. | - C’est une bibliothèque robuste et largement utilisée pour la génération de PDF en Node.js  - Elle offre de nombreuses fonctionnalités, notamment la gestion des polices, des images et des couleurs |
| Commander des impression | Nouvel onglet vers le back-office de Qwenta |  |  |  |
| Diffuser le menu sur instagram | Un découpage du menu en carrés doit pouvoir s’effectuer avant l’export vers instagram. | React-to-image | React-to-image est un composant wrapper construit sur la librairie html-to-image. | - Il est compatible avec React et facile d’utilisation.  - Il permet de créer des images directement à partir du HTML. |

Pourquoi choisir REACT?

ReactJS est très rapide, La communauté est grande, ReactJS est simple à lire, Le Javascript plus simple à écrire.

1. **Liens avec le back-end**

* Quel langage pour le serveur ? NodeJS

Qu’est-ce que Node.js et pourquoi l’utiliser ?

Node.js est un environnement d’exécution single-thread, open-source et multi-plateforme permettant de créer des applications rapides et évolutives côté serveur et en réseau. Il fonctionne avec le moteur d’exécution JavaScript V8 et utilise une architecture d’E / S non bloquante et pilotée par les événements, ce qui le rend efficace et adapté aux applications en temps réel.

* A-t-on besoin d’une API ?

Oui Une API Node JS Pourquoi utiliser Node JS pour construire une API REST ?

1.Pour la construction d’une API Node JS est un choix qui est souvent pertinent pour les raisons suivantes:

2.Son traitement non bloquant des requêtes

3.Sa performance et sa scalabilité

4.L’écosystème JavaScript et les packages open source disponibles

1. **Préconisations concernant le domaine et l’hébergement**

* Nom du domaine. (Le nom de domaine sera très probablement un sous-domaine de Qwenta)
* Nom de l’hébergement. (en cours de validation)
* Adresses e-mail. (en cours de validation)

1. **Accessibilité**

* Compatibilité navigateur.
* Types d’appareils. (Le site devra être en version desktop. Pas de version mobile à développer ni à prévoir.)

1. **Services tiers**

* Noms et justification de ceux qui devront être implémentés pour le bon fonctionnement et le suivi du site. (Il n'y aura pas d'outil à intégrer pour capter le comportement des utilisateurs pour le moment.).

1. **Recommandations en termes de sécurité**

* Accès aux comptes, plugins…

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

* Grandes lignes du contrat de maintenance.