Spécifications techniques

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Auteur** | **Date** | **Approbation** |
| 1.0 | Arnaud BUHL | 01/2024 | Soufiane |

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

1. **Choix technologiques**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Besoin** | **Contraintes** | **Solution** | **Description de la solution** | **Justification (2 arguments)** |
| Affichage de page dynamiquement | L’affichage doit se faire dynamiquement | React | React permet de créer des interfaces utilisateurs dynamiques | Très utilisé et simple d’utilisation |
| Création d'une catégorie de menu | L'ajout d'une catégorie doit pouvoir se faire directement sur l'écran de création de menu depuis une modale. | react-modal | Cette librairie React permet de créer simplement des modales performantes, accessibles avec un minimum de code. | Nous avons choisi de développer en React, la librairie est cohérente avec ce choix.  Il s'agit de la librairie la plus utilisée. |
| Exportation en format PDF | Génération de fichier PDF correspondant au menu créé par l’utilisateur | react-pdf | Utilisation de react-pdf pour générer le PDF coté client | Ce choix est cohérent avec notre utilisation de React |
| API Backend pour charger dynamiquement des données |  | Express.js | ExpressJS est un Framework pour construire des applications web | Très utilisé, performant et simple à prendre en main |
| Envoyer des mails pour la connexion utilisateur | Utiliser le serveur sans doute déjà en place chez Qwenta | Nodemailer | Nodemailer est un module NodeJS pour envoyer des mails | Nodemailer est simple d’utilisation et pourra s’intégrer facilement au backend en NodeJS |
| Données | | | | |
| Accéder/ajouter des informations utilisateur et menu, branding | Fiabilité | ORM | Utilisation d’un ORM pour faire le lien entre la base de données et le backend  (p.ex sequelize) | Simplifie et rend plus fiable l’accès à la base de données  Evite d’avoir à utiliser des requêtes SQL |
| Stocker les informations utilisateur et menu, branding | Stockage sécurisé des données et performances | SQL | Utilisation d’une base de données SQL comme MariaDB ou MySQL | Une base de données relationnelle pour avoir une structure bien organisée et stable |
| API Externe | | | | |
| Exporter un menu sur l’application Deliveroo | S’adapter à l’api Deliveroo existante | API Deliveroo | Utilisation de l’api Deliveroo pour exporter le menu |  |
| Partager un menu sur Instagram | S’adapter à l’api Instagram existante | API Instagram | Utilisation de l’api Instagram pour partager le menu |  |

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

* Quel langage pour le serveur : ***NodeJS***
* API Externe :
  + **Instagram**
  + **Deliveroo**
* Base de données choisie : ***SQL***

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

* Nom du domaine :
  + **menu-maker.qwenta.fr**
  + **menu.qwenta.fr**
* Nom de l’hébergement : **OVHCloud VPS**
* Adresses e-mail :
  + [**support@qwenta.fr**](mailto:support@qwenta.fr)
  + [**menu-maker@qwenta.fr**](mailto:menu-maker@qwenta.fr)

1. **Accessibilité**

* Compatibilité navigateur : **Chrome**, **Firefox** et **Safari**
* Types d’appareils : **Desktop**

1. **Services tiers**

* Nodemailer : pour envoyer les mails d’authentification en passant par un serveur SMTP
* API Instagram : pour partager les menus
* API Deliveroo : pour exporter les menus

1. **Recommandations en termes de sécurité**

* Lien de connexion :
  + Faire un lien à usage unique avec une expiration courte (p.ex 24h maximum)
  + Le lien doit également être long et aléatoire pour éviter de pouvoir tomber dessus en forçant ou par hasard
* Envoi de mails :
  + L’envoi des mails doit se faire à travers une connexion sécurisée au serveur SMTP
* Dépendances du projet :
  + Les paquets utilisés doivent être connus et éprouvés
  + Ils doivent également ne pas changer de version
* Utiliser une connexion HTTPS au site web

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

* Inclure 3 mois de correction de bug une fois le site terminé et en ligne
* Proposer un contrat de maintenance pour les petites modifications / petites fonctionnalités
* Proposer un contrat de mise à jour préventive
* Proposer également des sauvegardes de la base de données