Spécifications techniques

|  |  |
| --- | --- |
| **Projet** | Menu Maker By Qwenta |

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Auteur** | **Date** | **Approbation** |
| 1.0 | Glanowski Sally | 10/07/2023 | Soufiane |

Le but de ce document est de définir et justifier les spécifications techniques de Menu Maker By 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)** |
| **Connexion** | Le restaurateur doit pouvoir avoir un compte client et s’y connecter de façon sécurisée | Passeport JS  Certificat SSL | Permettre une stratégie d’authentification en utilisant Passport.JS qui est un middleware d’authentification pour Node.js  Le certificat SSL permettra de crypter les données en activant un protocole HTTPS | - Passport.js s’occupe de la logique d’authentification, ce qui facilite l’écriture du code en assure le bon fonctionnement car flexible et modulaire.  -le certificat SSL s’occupe de la sécurité de la communication entre le serveur web et le navigateur   * protègent à la fois l’accès à l’application et la confidentialité des informations transmises. |
| **Mail de confirmation** | Doit pouvoir réceptionner un mail afin de pouvoir confirmer la connexion | Node mailer (selon hébergeur), si beaucoup de mails, envisager de passer par un prestataire | [permet d’envoyer des emails depuis un serveur](https://nodemailer.com/). [Il expose une API qui offre des méthodes et des options pour créer et envoyer des messages à l’aide de différents transports comme : AWS SES d’Amazon (solutions d’envois de mail à grande échelle)](https://www.npmjs.com/package/nodemailer)  SMTP :protocole standard | -[simplifie le processus d’envoi d’emails depuis un serveur Node.js, sans avoir à se soucier des détails techniques ou des protocoles](https://pacomail.io/blog/comment-envoyer-un-email-avec-node-js/).  [-offre une grande flexibilité et une large gamme de stratégies d’envoi, ce qui permet de s’adapter à différents besoins et scénarios](https://ichi.pro/fr/comment-envoyer-des-e-mails-avec-node-js-a-l-aide-de-nodemailer-et-twilio-sendgrid-52954753888192) |
| **Créer une catégorie de menu** | Doit pouvoir ajouter des catégories et des menus par le biais d’une modale | React-modal | Module qui permet de créer des fenêtres modales pour les applications React.js. | -respecte les normes d’accessibilité en annonçant le contenu de la fenêtre et en gérant le focus  - [Offre une grande personnalisation du style et du comportement de la fenêtre modale, en utilisant des props, des classes CSS ou des styles en ligne](https://www.npmjs.com/package/react-modal) |
| **Créer un plat** | Le restaurateur doit pouvoir avoir un aperçu des modifications qu’il effectue sur ses plats et/ou menus lorsqu’il renseigne le formulaire d’ajout de catégories | Réact / file Reader | FileReader est une classe JavaScript fournie par l'API File de HTML5 pour lire des fichiers côté client. | -permet d’effectuer des « glisser-déposer » pour ajouter des photos stockées sur l’ordinateur du restaurateur  -permet de voir les changements en temps réel sur sa page |
| **Personnaliser un menu (logo, polices)** | Le client doit pouvoir changer la police et sa couleur et importer son logo. | Bootstrap | Bibliothèque qui permet de personnaliser le style et applicable dans l’environnement React | -des composants prêts à l’emploi et des plugins Javascript  -une large communauté d’utilisateurs assurant une maintenance optimale et l’amélioration des styles proposés par ses créateurs |
| **Diffuser le menu notamment l’export PDF** | Le restaurateur doit pouvoir exporter son menu au format PDF. | React-pdf. | Bibliothèque qui permet de créer des PDF à partir de composant react. | -génère des fichiers PDF  -permet de personnaliser les PDF |
| **Imprimer un menu** | Besoin pour le restaurateur d’imprimer son menu. | React-to-print | Bibliothèque qui permet de générer un PDF imprimable | -permet de lancer une fenêtre d’impression  -récupère les styles CSS du PDF et les injecte dans le portail |
| **Définir les données** | Besoin pour le restaurateur d’ajuster ses données (prix, dates, devises, description menus) | React Intl | Bibliothèque qui fournit des composants et des fonctions pour formater les dates, les nombres, les devises et les messages selon la langue et la région de l’utilisateur. | Permet de faciliter l’écriture des données en définissant leur structure en fonction du pays et de la langue (alphabet, type de devise etc)  React intel offre une API pour formater les dates, nombres et les chaînes de caractères |
| **Définition d’un menu** | Besoin pour le restaurateur d’élaborer une carte par catégories | Données NoSQL  -entrée  -plat  -dessert  -boissons | Utilisation de données type documents | Bien adapté pour enregistrer des données semi-structurées  Permet de modifier régulièrement les données et changer les menus. |
| **Insertion des prix** | Besoin pour le restaurateur d’insérer les prix | NoSQL  Création de données numériques type décimales (ex 25.99) | Avant de démarrer l’application, nous aurons définit avec l’agence combien de chiffres après la virgule seront nécessaires pour insérer les prix. Les prix seront en relation avec le nom du plat | Le nom du plat permet de retrouver le prix facilement  Le prix peut être facilement changé sans devoir parcourir plusieurs tables |
| **Liste des allergènes** | Besoin pour le restaurateur d’indiquer quels sont les aliments allergènes dans ses plats | NoSQL  Création d’une liste d’allergènes | Permet d’inclure les allergènes directement dans l’objet « plat » | Permet d’ajouter ou de supprimer des allergènes dans chaque plat  Toutes les informations concernant les allergènes sont rassemblées dans le même plat |
| **Description du plat** | Le restaurateur aura besoin d’effectuer une courte description de chaque plat | NoSQL  Création d’un texte (nombre de lettres à définir) | La description est directement intégrée dans l’objet « plat » | Ajout simple d’une description associée au nom du plat dans l’attribut « description »  Permet à chaque plat d’avoir une description qui lui est unique |
| **Insertion des photos des plats** | Le restaurateur aura besoin d’associer une photo à son plat | Utilisation de React file reader pour gérer les input de photos | La photo en format JPEG n’excédant pas 200 Ko  Pour éviter les ralentissements de chargement de la page.  Néanmoins MongoDB offre une grande performance en terme de stockage de données | Réact file reader est déjà un composant de l’environnement React  Offre des méthodes pour contrôler le format, la taille, le nom et le contenu des fichiers. |
| **Type de plats** | Le restaurateur aura besoin d’indiquer si ses plats sont végan, casher ou hallal etc | NoSQL | Création d’un attribut qui indiquera si le plat est végé, casher ou hallal | La spécificité est directement inclue dans le plat  Le restaurateur peut modifier cette indication s’il rajoute un nouvel aliment dans le plat |

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

* Langage serveur : **NodeJS**

🡪 Environnement d'exécution JavaScript côté serveur, permet aux développeurs d’exécuter du code Javascript côté serveur et pas seulement côté client (navigateur).

🡪 Il utilise le moteur JS V8 de Chrome qui est très rapide et performant pour interpréter et exécuter du code JS.

🡪 Il permet de créer des applications modernes en utilisant des frameworks comme Express, React etc.

* l’API : **ExpressJs**.
* Ce framework permet de créer des applications web dans l’environnement Node.js en utilisant un ensemble de méthodes et de fonctions. Avec l’API Express.js, il est possible de définir des routes qui gèrent les requêtes et les réponses HTTP, en échangeant des données de différents formats (JSON, XML, HTML, etc.).
* Base de données choisie :
* **NoSQL avec MongoDB** : Stockage de données flexible : MongoDB peut stocker des données de n'importe quelle structure (chiffre, caractère, document, images etc) et n'impose pas de schéma rigide, il permet donc une grande flexibilité pour gérer des données de différentes nature).

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

* Nom du domaine : **qwenta.menu**
* Nom de l’hébergement : **qwenta.menu.host** (car hébergement chez PlanetHoster pour ressources illimitées et sécurité optimale)
* Adresses e-mail : **contact@qwenta.menu**

1. **Accessibilité**

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

1. **Services tiers**

* **Sharethis** : permet d’ajouter des boutons de partage sur le site web pour que les visiteurs puissent également partager le contenu sur leurs propres réseaux sociaux
* **Bluffer**:permet de créer et de gérer les comptes de réseaux sociaux, comme Facebook, Twitter, Instagram, etc. Permet de publier les photos etc. Peut aussi interagir avec les abonnés et les clients. Il est aussi possible d’analyser les performances des posts, comme le nombre de vues, de likes, de commentaires etc
* **Axio** : bibliothèque qui gère les données JSON et qui utilise les méthodes http (get, post etc).

Il a une meilleure gestion des erreurs en générant des erreurs de code status (400 à 500). Il diminue le code en utilisant des « callback .then() » pour obtenir, récupérer et atteindre les données JSON.

1. **Recommandations en termes de sécurité**

* Une en-tête X-XSS-Protection qui active le filtre XSS du navigateur, bloque ou supprime les scripts malveillants détectés dans le site web.
* Une en-tête X-Frame-Options qui empêche le site web d’être affiché dans une iframe (élément HTML qui permet d’insérer une page web dans une autre. Cela évite que le site web soit utilisé dans une attaque par clickjacking).
* Utiliser le protocole HTTPS (et installer un certificat SSL) pour garantir l’identité du site et la confidentialité en cryptant les données transmises
* Utiliser des mots de passe suffisamment longs, complexes et différents pour chaque service ou compte du site web (utilisation de majuscules, chiffre et caractère spécial) et vérifier l’adresse mail.

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

* Sauvegarde régulière du site (sur un serveur distant par exemple, ici PlanetHoster)
* Mise à jour régulière du site qui peuvent demander des correctifs de sécurité, des améliorations de performances et de nouvelles fonctionnalités. Le travail de veille sera important pour s’assurer du bon fonctionnement du site (temps de chargement, bugs ou problème d’affichage).