Spécifications techniques

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

| **Version** | **Auteur** | **Date** | **Approbation** |
| --- | --- | --- | --- |
| 1.0 | Téri, Webgencia | 11/05/2024 | John, Qwenta |

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)** |
| --- | --- | --- | --- | --- |
| **Landing page** | L’utilisateur doit pouvoir comprendre à quoi va lui servir cet outil pour ensuite créer son menu. | HTML  SASS | Ces langages de développement web sont utilisés pour la création de page statique. | -Ces langages sont obligatoires pour la création d’une structure web.  -Permet une maintenance facile du code. |
| **Créer son propre menu (Ajout de plats)** | L’utilisateur doit pouvoir créer son propre menu :  -Catégories (entrée, plat, dessert, …)  -Ajouter une image, un nom, un prix et une description | React | React est un framework JavaScript qui permet de construire des interfaces utilisateur interactives et dynamiques. | -Création de composants réutilisables, ce qui simplifie le développement en réduisant la duplication de code.  -Permet une mise à jour rapide et sans rafraîchissement de la page, rendant l’expérience utilisateur plus fluide et agréable. |
| **Connexion utilisateur** | L’utilisateur doit pouvoir se connecter à son propre espace pour accéder à l’outil. | Firebase Authentification | Firebase Authentification permet de gérer l’authentification des utilisateurs via différents moyens : Email, Google, etc. | -En externalisant l’authentification, on bénéficie d’un meilleur soutien en termes de sécurité et maintenance.  -Sécurité avancée et robuste (avec notamment la prise en charge de l’authentification à deux facteurs). |
| **Ouverture des modales** | Avec Menu Maker, beaucoup de modales vont devoir s’ouvrir lors des créations de menus ou encore au moment de la connexion. | -React-modal | React-modal est une bibliothèque React conçue pour créer des fenêtres modales interactives. | -Création facile de modales.  -Amélioration de l’expérience utilisateur avec des interactions contextuelles. |
| **Dashboard** | L’utilisateur va pouvoir se connecter à son Dashboard pour accéder aux menus déjà créés, en modifier ou en créer de nouveaux. | Firebase Firestore | Firebase Firestore est une base de données NoSQL évolutive proposant une synchronisation en temps réel. | -Synchronisation des données en temps réel entre les clients et le cloud ce qui permet des mises à jour instantanées.  -Évolutivité horizontale automatique. |
| **Personnalisation du menu** | L’utilisateur pourra changer (avec un choix prédéfini) la couleur du texte et la police de son menu. | Composant React Styled | Les composants React Styles intègrent les styles dans leur définition, simplifiant la gestion des styles pour les interfaces utilisateurs. | -Avec une performance optimisée et une utilisation très répandue dans la communauté React, ces composants sont un choix populaire pour la gestion des styles.  -Facilite la personnalisation dynamique de la police d’écriture et de la couleur du texte grâce à sa syntaxe orientée composant et à son support intégré pour les thèmes/ |
| **Exportation et aperçu du document** | Lors de la création du menu, l’utilisateur devra disposer d’un aperçu de sa carte et de son menu en direct. | React PDF | React PDF permet de créer/exporter des PDF de manière dynamique en utilisant les composants React, offrant une personnalisation aisée des contenus PDF. | -Intégration native avec l’écosystème React, simplifiant le processus de développement.  -Personnalisation avancée du PDF en temps réel permettant à l’utilisateur d’avoir un aperçu avant validation du document. |
| **Enregistrer les images des menus** | Les images du menu vont devoir être stockées. | Firebase Cloud Storage | Firebase Cloud Storage est un service de stockage dans le cloud pour les fichiers statiques. | -Stockage sécurisé et évolutif des fichiers dans le cloud.  -Intégration facile avec d’autres services Firebase, transferts sécurisés.  -Possibilité d’utiliser Firebase Cloud Functions si l’on souhaite ajouter des fonctions de redimensionnement d’image par exemple. |
| **Impression des menus bu Qwenta** | L’utilisateur va pouvoir demander l’impression de son menu directement par Qwenta. | Cette spécification nécessite plus d’informations afin d’être étudiée.  Selon ce qui est déjà mis en place au niveau du back-office de Qwenta, il peut y avoir plusieurs choses à mettre en place. | | |
| **Diffuser le menu sur Instagram** | Un découpage du menu en format carré doit pouvoir s’effectuer | React-to-image | React-to-image est un composant wrapper construit sur la librairie html-to-image. | -Compatible avec React et facile d’utilisation.  -Permet de créer des images directement à partir du HTML. |

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

* Quel langage pour le serveur ?
  + Nous allons utiliser Node.js avec Express :
    - *Rapidité de développement* : Avec Express, Node.js permet un développement rapide grâce à sa nature asynchrone, à son écosystème de modules npm et à la simplicité de son framework Express, ce qui facilite la création rapide de serveurs web robustes.
    - *Évolutivité et performances* : Node.js est conçu pour être évolutif et capable de gérer un grand nombre de connexions simultanées grâce à son modèle non bloquant, telles que les applications en temps réel ou les API. Express quant à lui offre une structure légère et flexible pour développer des applications web efficaces.
* A-t-on besoin d’une API ? Si oui laquelle ?
  + Nous aurons besoin de 2 API externes :
    - Menu API de Deliveroo
    - L’API graph de Meta pour Instagram
  + Il faudra également construire une nouvelle API pour communiquer avec Firebase (pour la gestion de l’authentification, des menus, etc.) :
    - API Rest
* Base de données choisie :
  + Nous avons choisi la base de données Firebase Firestore.

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

* Nom du domaine
  + Le nom de domaine sera probablement un sous-domaine de Qwenta.
  + Si ce n’était pas le cas, nous pourrions nous tourner vers : **menu-maker.fr**
* Nom de l’hébergement.
  + Ayant choisi d’utiliser Firebase Cloud Storage, Firebase Firestore, etc. pour notre projet, il convient alors d’utiliser Firebase Hosting pour l’hébergement.
* Adresse e-mail.
  + Si nous utilisons le nom de domaine spécifié plus haut, l’adresse mail pourrait être **contact@menu-maker.fr**.
  + Néanmoins, la solution la plus efficace et pratique pourrait être **contact@qwenta.fr**.

1. **Accessibilité**

* Compatibilité navigateur.
  + Le site doit être compatible avec les dernières versions de Chrome, Safari et Firefox.
* Types d’appareils.
  + Pour l’instant aucune version sur mobile n’est prévue. Le site a donc seulement besoin d’être compatible sur desktop.

1. **Services tiers**

* **API Deliveroo** : Pour faciliter les commandes en ligne, en fournissant des informations actualisées sur les menus et en offrant un service de livraison intégré.
* **API Instagram** : Pour enrichir l’expérience utilisateur en lui offrant une variété de contenu visuel, et en favorisant l’engagement des utilisateurs (notamment au niveau de la visibilité et de la promotion du projet).
* **SendGrid** : SendGrid facilite la communication avec les utilisateurs en permettant l’envoi d’e-mails, notamment pour les mises à jour de menus, notifications de commandes, etc.
* **Stripe** : Permet à notre projet de disposer d’un service de paiement innovant, proposant plusieurs moyens de paiement sécurisés.
* **TypeForm** : Si toutefois notre projet avait besoin de retour d’expériences utilisateurs, TypeForm permet de concevoir des enquêtes, questionnaires ou encore sondages visuellement attrayants.

1. **Recommandations en termes de sécurité**

* Activer l’audit et la surveillance dans Firebase pour détecter toute activité suspecte et prévenir les incidents.
* Configurer correctement les règles d’accès et d’autorisation pour les Firebase Cloud Functions afin de bloquer tout accès non autorisé.
* S’assurer que toutes les bibliothèques, frameworks et dépendances sont régulièrement mises à jour pour intégrer les correctifs de sécurité.
* Utiliser les règles de sécurité Firebase Cloud Storage pour contrôler l’accès aux fichiers multimédias, garantissant ainsi que seuls les utilisateurs autorisés y accèdent.
* Éviter le stockage direct d’informations sensibles dans le code source.
* Implémenter des règles de sécurité Firebase Firestore pour restreindre l’accès aux données sensibles uniquement aux utilisateurs authentifiés.
* Ne pas oublier d’installer un certificat SSL pour activer le protocole HTTPS.
* Veiller à donner des durées de vie raisonnables pour les liens d’authentification et les token de connexion (ce point pourra être discuté par la suite).
* Valider les données saisies par les utilisateurs pour éviter les attaques d’injection de code.

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

* **Maintenance** :
  + Effectuer des mises à jour régulières du système.
  + Surveiller attentivement les performances de notre site.
  + Vérifier la sécurité globale du site de manière constante.
  + Établir et maintenir une documentation à jour.
* **Mises à jour futures** :
  + Développer plusieurs modèles de menu.
  + Intégrer de nouvelles fonctionnalités pour la création et la modification de menus.
  + Concevoir et intégrer des maquettes pour les tablettes et smartphones.
  + Intégrer un blog interne dédié au site Menu Maker.
  + Créer de multiples profils pour les clients ayant plusieurs restaurants.
  + Implémenter des animations pour dynamiser le site.
  + Rester informé des dernières tendances pour une évolution continue.