Spécifications techniques

[Menu Maker, Qwenta]

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Auteur | Date | Approbation |
| 1.0 | [Raux Clément] | [Janvier 2025] | [Soufiane, Webgencia] |

[I. Choix technologiques](#_409jm82qlgst)2

[II. Liens avec le back-end](#_ex0md8td8283)3

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’User même s’il ne connait pas Menu Maker doit comprendre et voir ce qu’il peut faire avec Menu Maker (insérer les tarifs et informations légales). | *Html et Sass* | *Pour la landing Page qui est statique ces deux technologies Html et Sass sont parfaites.* | 1. *Le Html est la base d’une application Web.*   *2) Sass est utile pour les variables et les mixins, mais peut rendre plus facile la gestion des animations à venir sur la Landing Page.* |
| Login Page | Le nouvel utilisateur doit pouvoir créer un compte facilement avec son email et pouvoir se connecter s’il a déjà un compte | Auth0 (Okta) | Auth0 est une plateforme d’authentification qui prend en charge OAuth mais aussi les authentifications email/mdp. | 1. Possibilité intégrer OAuth pour favoriser la sécurité via Google ou autres plateformes.   2) Outil fournis pour facilement gérer les utilisateurs à l’aide d’un tableau de bord. (Ajouter des règles ou non) |
| Création de menus | Possibilité d’ajouter des plats dans le menu avec des catégories pour ceux-ci (description, images et prix possible). Pas de limite de plats. | React | React permet de mettre en place des Applications Web modernes qui sont dynamiques. De plus React a une forte communauté très active. | 1. Les changements qu’apportent React sont quasi instantané et donc permettent une aisance dans l’expérience utilisateur.   2) Code facilement Modulable à l’aide de composants réutilisables. |
| Personnalisation Menu | L’utilisateur connecté va pouvoir personnaliser son menu que ce soit sur la couleur ou la typographie. |  |  |  |
| Historique des Menus | L’utilisateur connecté doit avoir accès à un historique qui présente les menus qu’il a déjà composés. (possibilité de PUT/DELETE,) | MongoDB | Base de donnée NoSQL qui va permettre d’enregistrer un historique des menus. | 1. Chaque menu peut avoir une structure différente car MongoDB se base sur les documents permettant ainsi une flexibilité. 2. MongoDb est hautement scalable donc nous n’aurons pas de soucis de performance au lancement de l’application même si plusieurs restaurateurs mettent à jours le menus régulièrement. |
| Exportation PDF | L’utilisateur doit pouvoir en un clic télécharger son menu au format PDF après avoir eu un aperçu de celui-ci. | React PDF | React Pdf va permettre de rendre un composant sous forme de PDF. | 1. Cette bibliothèque utilise les composant React elle est donc intégré naturellement à l’application. 2. Possibilité de mettre en place une mise en page avancée et sophistiquée. |
| Imprimer les menus | En plus de pouvoir créer et diffuser un menu, l’utilisateur connecté pourra directement commander l’impression d’un menu. | Nodes.js avec Express | Créer un tableau de bord interne avec un suivis des commandes Via une API | 1. Suivis et historique des commandes d’impression. 2. Automatisation possible une fois que la commande est validée du coté du restaurateur alors l’impression est lancé. |
| Exportation Deliveroo | Si l’utilisateur est connecté il doit pouvoir avoir le choix donc entre exporter par Pdf, Deliveroo, et Instagram ci-dessous. | Nodes.js avec Express | Backend de l’application, d’ici seront effectuées les appels à l’API de Deliveroo. | 1. L’API va gérer la communication avec Deliveroo directement.   2) En utilisant l’API externe de Deliveroo le processus d’exportation est automatisé ce qui evite l’entrée des données à la main sur une autre plateforme. |
| Exportation sur Instagram | Exportation du menu directement sur instagram si l’utilisateur est connecté. | Nodes.js avec Express | Générer une image à partir du pdf dans le Back et utiliser l’API pour publier l’image sur le compte pro. | 1. Automatisation pour le marketing encore une fois. 2. Être présent de manière professionnel sur un réseau comme Instagram permet d’accéder à des outils et des statistiques sur la clientèle potentielle. |
| Modifications des informations utilisateurs? Branding? |  |  |  |  |

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

* ***Quel langage pour le serveur ?***

*Nodes.js avec Express qui est un framework facilitant la création d’API REST, avec MongoDB pour la gestion des données (utilisateurs, menus).*

* ***A-t-on besoin d’une API ? Si oui laquelle ?***

***Oui il va falloir une API pour communiquer entre le Front (React) et le Back (Nodes.js) en permettant à React d’accéder à la base de données par différentes opérations de CRUD, il faudra donc créer une API REST (Requête HTTP, représentation des données en JSON, réponse typé selon code..) cela sera donc l’API créée par notre dev Backend.***

***Il faut aussi se pencher sur les besoins des clients et donc intégrer 2 autres API au projet, la première sera l’API de Deliveroo pour permettre aux restaurateurs d’avoir la possibilité d’exporter directement leurs menu sur l’application Deliveroo. De plus il faudra intégrer l’API GRAPH d’Instagram qui permettra aux utilisateurs d’un compte pro (les restaurateurs) de partager une image de leurs menus directement sur le réseau social.***

* ***Base de données choisie :***

***Pour cette application web il serait judicieux de porter notre choix sur une base de données NoSQL, en effet il n’y a pas vraiment d’organisation fixe et limitée puisque la création du menu est entièrement personnalisable, il sera rare que deux menus soit totalement similaires entre différents restaurants. De ce fait, les utilisateurs qui vont eux mêmes créer keurs menus vont engendrer une quantité d’informations diverses et variées et donc des données tout aussi variées. Le fait de choisir ici une base de données NoSQL rajoute une flexibilité dans la gestion des informations et permet aussi une maintenabilité des performances même si l’application vient à évoluer. Nous allons donc utiliser une base de données NoSQL de type document (MongoDB avec des JSON), ainsi chaque restaurants pourra avoir un menu structuré à sa manière, c***ette flexibilité permet d'adapter facilement les données au fur et à mesure que l'application évolue ou que de nouveaux types de menus ou d’éléments sont ajoutés sans pour autant mettre de coté les performances au vus des données à gérer.

*.*

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

***Nom du domaine :***

*Le nom de domaine serra un sous domaine de Qwenta -> menu-maker.Qwenta.fr*

***Nom de l’hébergement :***

***Adresses e-mail : Adresse email pour le support : [contact@menu-maker.Qwenta.fr](mailto:contact@menu-maker.Qwenta.fr)***

1. ***Accessibilité***

***Compatibilité navigateur : L’application Menu Maker sera compatible avec les dernières mise à jour des navigateurs que sont Firefox, Chrome ainsi que Safari.***

***Types d’appareils : Il est est prévu la version Desktop, cependant il n’y a pas de Responsive à réaliser car pas de version mobile disponible pour le moment pour Menu Maker.***

***Outils :***

***De plus la navigation au clavier et l’utilisation de liseuse doit être effective, c’est pourquoi nous utiliseront un Tool qui est Wave, en effet c’est un outil très pertinent qui permet d’évaluer l’accessibilité sur un site Web et suggère des améliorations pour les utilisateurs qui en ont besoins.***

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.***