**Spécifications techniques**

Menu Maker by Qwenta

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Auteur** | **Date** | **Approbation** |
| 1.0 | Natacha Lefèvre, Webgencia | 01/02/2025 | John, Qwenta |

[I. Choix technologiques](#_409jm82qlgst) 2

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

[III. Préconisations concernant le domaine et l’hébergement](#_a61v3uh5vs19) 3

[IV. Accessibilité](#_n56najuvb5ys) 3

[V. Recommandations en termes de sécurité](#_l80x4g9stizp) 3

[VI. Maintenance du site et futures mises à jour](#_vbbknpfyus5o) 4

## 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 non connectée** | L'internaute doit avoir accès aux sections:  - Bannière  - Personnalisez votre menu  - Explications étape par étape  Il faut respecter la maquette Figma.  Le site nécessite les langages HTML, CSS et Javascript pour fonctionner. | React avec routeur React | React est une bilbiothèque Javascript qui permet de créer des interfaces web, et d’utiliser des modales et composants réutilisables.  Le routeur React permet la navigation entre les pages d’une appli/d’un site à page unique (landing page) sans avoir besoin de recharger la page. | 1. Il s'agit de la librairie la plus utilisée, donc bénéficiant d’un important support de la part des nombreux développeurs qui l’utilisent.  2. Le fonctionnement par modales réutilisables correspond à nos besoins. |
| **Page login** | Cette page s’ouvre sous forme de modale.  La connexion et l’inscription doivent être sécurisées par mail. | React Modal  Passport.js | La librairie React Modal permet de créer facilement des modales et des composants réutilisables.  Passport.js est un middleware pour Node.js | 1. Sur notre projet, la connexion et l’inscription se présentent sous forme de modale.  2. Passport.js permet de créer un compte par email et mot de passe. |
| **Catégorie de plat** | La création de catégorie s'ouvre dans une modale spécifique, et doit pouvoir être validée. | React Modal | La librairie React Modal permet de créer facilement des modales et des composants réutilisables. | 1. Il s'agit de la librairie la plus utilisée et suivie.  2. Le fonctionnement par modales réutilisables correspond à nos besoins. |
| **Création de plat** | Le restaurateur doit pouvoir ajouter des plats directement dans le menu.  Les différents plats s’ouvrent par modale.  Il faut pouvoir ajouter une photo pour chaque plat. | React Modal | Grâce à la modale, le restaurateur peut choisir et modifier une catégorie, et entrer les infos de chaque plat (nom, description, photo, prix). | 1. Il s'agit de la librairie la plus utilisée et suivie.  2. Le fonctionnement par modales réutilisables permet une personnalisation complète. |
| **Style de menu** | Le restaurateur doit pouvoir changer la typo et la couleur des textes, afin de personnaliser son menu. Il doit aussi pouvoir visualiser le menu avant validation. | Material UI | Material UI est une bibliothèque React qui propose des composants pré-faits, et qui intégre le Matériel Design de Google. | 1. Sa compatibilité avec les règles de design de Google le rend fiable pour les années à venir.  2. Facile d’usage, il permet de créer facilement un kit de design, ou d’importer des kits déjà faits. |
| **Exportation PDF** | Le restaurateur doit pouvoir exporter son menu en PDF d’un clic. | jsPDF | JsPDF est une bibliothèque open source qui permet de générer des PDF facilement et en un clic. | 1. C’est une bibliothèque très utilisée, donc avec une large communauté d’utilisateurs pour l’entretenir.  2. Le restaurateur peut télécharger facilement le PDF de son menu. |
| **Commander l’impression d’un menu** | Le restaurateur doit pouvoir commander une impression en un clic. | Gérer l’impression par le back-office de Qwenta (HTML target=blank). | L’impression est dispo via un nouvel onglet qui redirige vers le back-office de Qwenta. | 1. Le processus est facile à utiliser pour le restaurateur.  2. L’impression reste gérée par Qwenta. |
| **Gestion des anciens menus** | Les anciens menus créés par le restaurateur doivent rester accessibles.  Le restaurateur doit pouvoir modifier, supprimer un menu et créer un nouveau menu, sur la même vue. | React Router pour le front-end  MongoDB pour le back-end | Créer une API REST pour communiquer entre le serveur client et la base de données. | 1. React Router permet de créer une vue de tous les menus créés par le restaurateur.  2. La base de données pourra stocker les infos des menus créés par le restaurateur. |
| **Infos légales** | Les mentions légales doivent être accessibles dans une modale.  La mention « Tous droits réservés » doit figurer sur toutes les pages. | React Modal | Quand on clique sur les mentions légales, une modale s’ouvre pour afficher les infos légales. | 1. React Modal permet de créer une seule modale pour les mentions légales qui sera reprise sur toutes les pages.  2. React est très populaire, et dispose d’un suivi accru par ses nombreux utilisateurs. |
| **Tarifs** | Les tarifs vont s’ouvrir dans une page à créer. Cliquer sur le bouton Tarifs depuis la page d’accueil va ouvrir un nouvel onglet. | React | La librairie React Modal permet de créer facilement des modales et des composants réutilisables. | 1. Il s'agit de la librairie la plus utilisée et suivie.  2. Le fonctionnement par modales réutilisables permet une personnalisation complète. |
| **Exportation Deliveroo** | Le restaurateur doit pouvoir exporter son menu en un clic vers l’appli Deliveroo, sur un nouvel onglet. | API de Deliveroo | Deliveroo propose un outil Partner Platform Suite, pour gérer le lien entre un restaurant et Deliveroo. | 1. Deliveroo fournit déjà les outils nécessaires pour gérer ses menus.  2. Cette API propose une documentation complète. |
| **Partage sur Instagram** | Il faut générer des images carrés pour s’adapter au format Instagram.  Le restaurateur est redirigé vers son compte Instagram.  Il faut créer un compte développeur sur Instagram. | React Image File Resizer  API Graph d’Instagram | React Image File Resizer permet de redimensionner les images.  L’API Graph gère les accès et contenus d’Instagram. | 1. React Image File Resizer est simple d’utilisation, et s’intègre à React que l’on utilise pour l’ensemble du site.  2. API Graph est fourni et sécurisé par Instagram (Meta). |
| **Déconnexion** | Le restaurateur doit pouvoir se déconnecter depuis n’importe quelle page. | React | Un bouton de déconnexion est disponible sur chaque page. | 1. Le restaurateur peut se déconnecter à tout moment en un seul clic.  2. React est le choix fait pour le site. |
| **Infos utilisateur** | Le restaurateur doit pouvoir modifier ses infos, son e-mail, et lier plusieurs adresses mail à son compte. | React pour le front-end  MongoDB pour le back-end | Créer une API REST pour communiquer entre le serveur client et la base de données. | 1. React Router est cohérent avec le choix de coder en React.  2. La base de données pourra stocker les infos utilisateurs. |
| **Dashboard** | Le dashboard doit regrouper les infos du menu du restaurateur, aux 3 derniers articles de blog sur MenuMaker. | React | La librairie React Modal permet de créer facilement des modales et des composants réutilisables. | 1. Il s'agit de la librairie la plus utilisée, donc bénéficiant d’un important support de la part des nombreux développeurs qui l’utilisent.  2. Le fonctionnement par modales réutilisables correspond à nos besoins. |
| **Branding restaurateur** | Le restaurateur doit pouvoir créer l’identité visuelle (logo, couleurs) de son restaurant directement sur l’appli. | React pour le front-end  MongoDB pour le back-end | Créer une API REST pour communiquer entre le serveur client et la base de données. | 1. React Router est cohérent avec le choix de coder en React.  2. La base de données pourra stocker les infos utilisateurs. |

## Liens avec le back-end

* **Quel langage pour le serveur ?**

NodeJS, car c’est un langage très populaire et documenté par ses utilisateurs. Il est open source, compatible avec Javascript, et permet d’accéder au front-end et au back-end.

* **A-t-on besoin d’une API ?**

Si oui laquelle ? Oui, une API REST, avec Node.js, Express.js et MongoDB.

* **Base de données choisie :**

Nous utilisons MongoDB, une base de données NoSQL, facile d’utilisation.

## Préconisations concernant le domaine et l’hébergement

* **Nom du domaine :**

Le nom de domaine est encore à valider, probablement un sous-domaine de Qwenta (exemple : [www.menumaker.com](http://www.menumaker.com/) ou [www.menumaker.qwenta.com](http://www.menumaker.qwenta.com/)).

* **Nom de l’hébergement.**

OVHCloud est un hébergeur fiable, ayant fait ses preuves depuis de longues années, et qui dispose d’une version pro (https://www.ovhcloud.com/fr/web-hosting/professional-offer/).

* **Adresses e-mail.**

L’idéal serait des adresses mail qui renvoient directement à MenuMaker.

Exemple : [contact@menumaker.com](mailto:contact@menumaker.com) ou [infos@menumaker.com](mailto:infos@menumaker.com)

## Accessibilité

* Le site doit être compatible avec Firefox, Chrome et Safari.
* Pour ordinateurs (desktop). Pas de version mobile prévue.

## Recommandations en termes de sécurité

* Les données des utilisateurs sont cryptées dans la base de données MongoDB.
* Passport.js permet de crypter et protéger le mot de passe.
* Le protocole HTTPS permet de chiffrer les données.
* Appliquer les règles du RGPD.

## **Maintenance du site et futures mises à jour**

* Conserver le contact entre Qwenta et Webgencia pour repérer et résoudre d’éventuels problèmes et bugs.
* Installer les mises à jour du site régulièrement.
* Assurer la maintenance de la base de données.
* Rester en veille des nouveautés, outils, documentations, pour que le site ne soit pas obsolète ni vulnérable.