Spécifications techniques

|  |  |
| --- | --- |
| **Projet** | Menu Maker pour Qwanta |

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Auteur** | **Date** | **Approbation** |
| 1.0 | Gauthe Alexandre | 27/02/2024 | Soufiane, Webgencia |

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)** |
| *Ensemble du projet* | *Un site d’une seule page dont la navigation se fait essentiellement par le biais de fenêtre modales. Le site à pour but de permettre à des restaurateurs de crée, exporter partager et imprimer leurs menus.* | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* | *React permet ,avec un code relativement réduit, de mettre en place des éléments tels que les modales .*  *Nous utiliserons un serveur Node.js et une API REST afin de traiter les échanges entre le site la base de données du site.*  *La base de données sera écrite en langage SQL* | *React est tout indiqué pour ce projet car il va nous permettre de mener ce projet à bien en limitant le code et donc le temps de développement et par conséquent réduire le budget utilisé. Ilest de plus largement supporté et suivi par une communauté très présente en ligne , sa réputation n’est plus à faire. Il est de plus fortement recommandé pour les projets ne comportant qu’une seule page comme c’est la cas ici.*  *La technologie backend choisie à été retenue car elle nous apportant la souplesse et la puissance nécessaire au projet , qu’il s’agisse du serveur node.js ou de l’API tout deux ont fait leurs preuves et sont très largement utilisés aujourd’hui.*  *Enfin la nature des données qui seront utilisées nous tendent à choisir une base de données SQL.* |
| *Landing Page 1.1 :*  *Version non-connecté*  *Une page d’accueil claire avec une présentation concise de l’outil Menu Maker* | *La page doit être accessible sans connexion préalable.*  *Depuis cette page l’utilisateur doit pouvoir aller sur la page de connexion ou s’inscrire. Il doit pouvoir comprendre le fonctionnement du site grâce à un encart schématique de la solution en bas de page ainsi qu’aux deux CTA présents sur la page.* | *Partie UI : React* |  |  |
| Landing Page 1.2 :  Sur la page d’accueil doit figurer un lien « mentions légales » | Ce lien doit être visible sur la page accueil version connecté et non connecté. Au clic sur le lien une modale statique apparaîtra avec les mentions légales. | *Partie UI : React* |  |  |
| Landing Page 1.3 :  Sur la page d’accueil doit figurer un lien « Tarifs » | Ce lien doit être visible sur la page accueil version connecté et non connecté. Il doit rediriger vers la futur page : <https://URL_DE_QWENTA/tarifs/menumaker> | *Partie UI : React* |  |  |
| Login 1.1 :  Une page de connexion qui contiendra un formulaire sous forme de modale. | Un nouvel utilisateur doit pouvoir se connecter en indiquant simplement son adresse mail. Un restaurateur déjà utilisateur se connecte avec son adresse mail déjà entré. Au clic un mail de confirmation est envoyé a l’utilisateur | *Partie UI : React* |  |  |
| Login 1.2  La modale change d’état pendant le traitement de la requête de connexion | Dans l’attente de la confirmation de l’utilisateur via le mail reçu, le contenu de la modale change et une animation de chargement apparaît. | *Partie UI : React* |  |  |
| Login 1.3  L’utilisateur confirme son inscription sur le mail reçu, Menu Maker Valide la connexion | Quand l’utilsateur valide sa connexion/inscription via le mail reçu, la requête est envoyé et la modale change une dernière fois pour confirmé la connexion puis l’utilisateur sera redirigé vers la page d’accueil version Dashboard . | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Login 1.4 Déconnexion | Au clic sur le lien l’utilisateur doit pouvoir se déconnecter et être rediriger sur la landing page. Ce lien doit être présent sur toutes les pages du site. | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Dashboard 1.1 Aprés connexion un dashboard doit permettre à l’utilisateur d’administrer ses menus. | Après la connexion l’utilisateur arrive son espace personnel présenté sous la forme d’un Dashboard depuis lequel il pourra Créer et gérer ces menus. Le Dashboard doit regroupé les 3 encarts création diffusion et impressions de menu. Une section « Pour aller plus loin » avec les 3 derniers articles qui parlent de menuMaker (Cards = photo+titre+lien)  (A l’avenir , possibilité de création de blog interne) Doit figurer également dans sur cette page le menu de navigation qui permettera à l’utilisateur de passer du Dashboard à ses menus , son restaurant ou encore son compte. | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Dashboard 1.2 Branding restaurateur  L’utilisateur doit pouvoir crée le branding de son restaurant | En cliquant sur le lien « Mon restaurant » le restaurateur pourra au travers d’une modale ajouter modifier ou supprimer un logo et changer les couleurs de base de son branding | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Création menu 1.1  Crée un menu  L’utilisateur doit pouvoir commencer la création de son menu une fois authentifier | Au clic sur l’encart « crée un menu », l’utilisateur bascule sur la partie de création de menu au moyen d’une transition. On retrouve à ce niveau le formulaire de création de menu. Cette modale doit également être accessible depuis la partie « mes menus » | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Création menu 1.2  Ajout catégorie  L’utilisateur doit pouvoir crée des nouvelles catégories de plats | En cliquant sur « ajouter vos plats »  l’utilisateur peut dans un premier temps et au travers d’une modale, crée une catégorie et celle-ci sera conserver à l’avenir. | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Création menu 1.3  Ajout plat  L’utilisateur doit pouvoir ajouter modifier et supprimer des plats après avoir crée une catégorie | Au clic sur « + plat »  l’utilisateur fait apparaître une modale qui contient le formulaire d’ajout de plat. Il est composé d’un champs pour renseigner une image et de trois champs respectivement pour le nom du plat le prix et la description .  Les plats sont affichés les uns de dessus des autres la modale doit être scrollable et il ne doit pas y avoir de limite de plats. | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Création menu 1.4  Personalier le menu  L’utilisateur doit pourvoir personnaliser son menu et prévisualiser celui-ci pendant sa création | L’utilisateur doit pouvoir personnaliser son menu directement dans une modale, ces choix apparaissent dans la fenêtre de prévisualisations qui doit apparaître à droite de la modale. | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Création menu 1.5  Diffusion 1.1  Il doit être possible d’exporter un menu crée en PDF | Après l’étape de personnalisation de menu, l’utilisateur doit avoir la possibilité d’éxporter son menu en pdf en un simple clic . | *Partie UI : React et le module jspdf*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Création menu 1.5  Diffusion 1.2  Il doit être possible de diffuser un menu crée sur Deliveroo | Après l’étape de personnalisation de menu, l’utilisateur doit pouvoir diffuser son menu sur deliveroo en un clic | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Création menu 1.5  Diffusion 1.3  Il doit être possible de diffuser un menu crée sur instagram | Après l’étape de personnalisation de menu, l’utilisateur doit pouvoir diffuser son menu sur instagram en un clic | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Création menu 1.6  Sauvegarde des menus  Un menu crée doit persisté dans l’espace utilisateur | Après avoir crée un menu, celui-ci doit être sauvegarder dans l’espace utilisateur et consultable dans la partie « mes menus ». Depuis cette partie il doit être possible de modifier, de supprimer ou même de crée un menu | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Compte utilisateur  L’utilisateur doit pouvoir modifier son adresse mail ou en lier une nouvelle . (Amélioration future , faire de même pour les moyens de paiements) | Une partie « Mon compte » devra être accessible à l’utilisateur authentifié depuis le menu de navigation. Dans cette partie il pourra modifier son adresse mail renseignée ou en lier une nouvelle. A l’avenir l’utilisateur pourra renseigner et modifier ses moyens paiements | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |
| Backoffice  L’utilisateur doit pouvoir commander l’impression de ses menus. | Depuis le dashboard il doit être possible à l’utilisateur de passer la commande d’impression de son menu. Celle doit ensuite être transmise au backoffice de Qwanta | *Partie UI : React*  *Backend: Node.js et API REST*  *Base de données: SQL* |  |  |

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

* *Le langage utiliser par le serveur sera le Node.Js*
* Nous utiliserons pour ce projet une API REST
* *Nous utiliserons le langage SQL pour notre base de données*

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

* URL du site : Https://www.menumaker/Qwanta.fr
* Le site sera hébergé par IONOS
* Adresse mail : menumaker.qwanta@gmail.com

1. **Accessibilité**

* Menu Maker devra être compatible avec les dernières versions de Chrome, Safari et Firefox.
* Menu Maker sera responsive et donc accessible pour tout les appareils type «Desktop»et Tablette et si le besoin était évoqué plus tard , pour les mobiles également .
* Le site sera conforme aux normes W3C et WCAG et donc accessible à tout les utilisateurs y compris ceux en situation de handicap .

1. **Services tiers**

* Pour répondre à tout les besoin du projet nous devrons utiliser le module jspdf
* Pour la diffusion avec Deliveroo et Instagram, nous aurons besoins de leurs api respectives
* Pour la gestion future des moyens de paiement nous pourrions intégrer Stripe à travers React.

1. **Recommandations en termes de sécurité**

* L’inscription et l’accès aux compte se fera par adresse mail uniquement, un mail de confirmation sera envoyé sur celle-çi avec un lien de vérification avant connexion. Une fois la validation du mail faite et la connexion établie un JWT (jeton d’authentification) sera généré par le serveur à l’utilisateur afin d’assurer une totale sécruité à l’espace personnel de chaque utilisateur de la plateforme.

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

* Le large support dont bénéficies les technologies employés , qu’ils s’agissent de React ou Node.js, nous assurent un fonctionnement optimal du site sur le long termes. En effet ces technologies bénéficies de mise à jour régulières (cela garanti notamment leur sécurité et leur fiabilité), elles sont toutes deux de plus supporter par une très large communauté en ligne. Ces technologies n’ont plus rien à prouver dans le domaine du développement elles sont très largement utilisés sur le web tout entier. React ayant été crée par Facebook cela est un gage de plus si il fallait en rajouté de sa fiabilité et de son support.