Spécifications techniques

|  |  |
| --- | --- |
| **Projet** | Menu Maker by QWENTA |

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Auteur** | **Date** | **Approbation** |
| 1.0 | Prunelle Stoëssel | 13/11/2023 | Guillaume Kpomda |

Le but de ce document est de définir et justifier les spécifications techniques de [nom du projet].

1. **Choix technologiques**

* État des lieux des besoins fonctionnels et de leurs solutions techniques :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Besoin** | **Contraintes** | **Solution** | **Description de la solution** | **Justification (2 arguments)** |
| Création d'une catégorie de menu | L'ajout d'une catégorie doit pouvoir se faire directement sur l'écran de création de menu depuis une modale. | react-modal | Cette librairie React permet de créer simplement des modales performantes, accessibles avec un minimum de code. | 1) Nous avons choisi de développer en React, la librairie est cohérente avec ce choix.  2) Il s'agit de la librairie la plus utilisée. |
| Choisir la police du menu | La modification de la police doit se répercuter dans l’aperçu du menu | Boutons via interface | Proposer 3 choix de police sous formes de boutons à l’utilisateur | 1/ Un contrôle sur les polices utilisée  2/ Facilité d’utilisation |
| Choisir la couleur du menu | La sélection se fait sur l’écran et la couleur doit changer dans l’aperçu du menu | Utilisation d’un color picker (input type color) | Mettre en place un color picker dans un fichier Javascript. | 1/ Solution native de JS.  2/ Pas besoin de librairie |
| Enregistrer ses préférences | L’enregistrement doit être dynamique | Lien vers l’API de Manu Maker. | Envoyer au serveur, le texte, la police et la couleur du menu au Format JSON.  Enregistrement au clic du bouton « Valider », | 1/ l’affichage de l’aperçu sera dynamique.  2/ Pas besoin de rafraichir la page. |
| Diffuser un menu | La diffusion du menu doit pouvoir se faire grâce à l’appli MenuMaker | Utilisation des API insta et deliveroo pour diffuser le menu. | Les API permettent de créer un lien entre l’application et les réseaux | 1/ Les API sont directement fournis par Instagram et Deliveroo.  2/ Mode d’emploi fournis |
| Exporter un menu | Pouvoir récupérer son menu créé en PDF | React pdf | React permettra d’exporter simplement un fichier en pdf avec peu de code. | 1/ On continue d’utiliser react pour le front end  2/ Librairie la plus utilisée |
| Pouvoir commander l’impression d’un menu | Option accessible sur la page d’accueil.  Un nouvel onglet s’ouvre vers le back-end de Qwenta | Lien href | Créer un lien envoyant l’utilisateur vers une page du site qwenta pour faire commande d’impression | 1/ Qwenta pourra s’occuper de conseiller le client pour l’impression.  2/ Distinction claire entre le site de menu maker et celui de Qwenta |
| Création d’un serveur pour les besoins de l’application | Doit pouvoir stocker les menus des clients mais aussi leurs informations personnels | Express.js | Framework de développement back end simple et minimaliste | 1/ Garder le même langage que le front end  2/ Pas besoin de multiplier les expertises |

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

* Quel langage pour le serveur ?
* *JavaScript en utilisant Node.js et Express.js*
* A-t-on besoin d’une API ? Si oui laquelle ?
* API Instagram
* API Deliveroo
* Api MenuMaker
* Base de données choisie : *SQL avec MySQL*

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

* Nom du domaine : www.qwenta-menumaker.fr
* Nom de l’hébergement : Microsoft Azure
* Adresses e-mail : qwenta.menumaker@contact.fr

1. **Accessibilité**

* Compatibilité navigateur.
* Chrome
* Firefox
* Safari
* Types d’appareils.
* Desktop uniquement
* Respecter le standard ARIA

1. **Services tiers**

* Noms et justification de ceux qui devront être implémentés pour le bon fonctionnement et le suivi du site (par exemple les plugins).
* Express.js : librairie javascript pour aider au développement backend de l’application.
* API Instagram : nécessaire pour la publication des photos directement sur le compte du client.
* API Deliveroo : nécessaire pour la diffusion du menu directement sur le compte du client.
* Microsoft Azure : serveur sur lequel seront hébergés les différentes parties de l’application.

1. **Recommandations en termes de sécurité**

* Ne pas installer de plug-in sans consulter Webgencia.
* Ne pas stocker les données utilisateurs en clair.
* Héberger frontend, backend et bdd sur 3 serveurs différents.
* Réaliser un pentest avant le lancement de l’appli

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

* Grandes lignes du contrat de maintenance.

3 itérations possibles dans une limite de 15 jours de travail ensuite facturation des mises à jour.