Spécifications techniques

|  |  |
| --- | --- |
| **Projet** | Qwenta |

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Auteur** | **Date** | **Approbation** |
| 1.0 | Dos Santos Vincent | 01/03/2023 | John |

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 de menu | L’utilisateur doit pouvoir créer un menu une fois connecté. Il doit pouvoir ajouter à ce menu : les plats, les prix ainsi que des descriptions. | Utilisation de python afin de pouvoir créer un menu et l’envoyer à l’API | Avec un formulaire HTML et le langage python on va pouvoir faire une option de création de menu | 1. La possibilité de créer un menu est la base du site 2. HTML et python nous permettrons de mettre en place cette option de création de menu |
| Personnalisation de menu | Le menu doit être très personnalisable (couleur, typographie, images). L’utilisateur doit également pouvoir enregistrer ses préférences de branding. | Langage python | Ce langage peut nous permettre d’offrir des options de personnalisation à l’utilisateur | 1. Python nous permettra de personnaliser les menus 2. La personnalisation de menu est indispensable afin que l’utilisateur crée des menus sur notre site |
| Stockage des menus envoyés | Ces menus doivent pouvoir être stockés et utilisés | Utilisation d’une API et du langage SQL | Avec une API et SQL nous pourrons stocker les menus envoyés et les afficher | 1. Nous utiliserons Python pour l’API car c’est le langage choisi pour le côté serveur 2. Nous utilisons SQL car nous nous servons de données fixe. |
| Système de connexion internaute | L’utilisateur lambda doit pouvoir se créer un compte et se connecter afin de stocker ses menus préférés dans ses ‘favoris’. | Utilisation d’une API et du langage SQL. Sécurisation avec différentes pratiques de sécurités | Avec une API et SQL nous pourrons stocker les identifiants et les mots de passe pour créer un système de connexion | 1. Nous utiliserons une API crée avec Python 2. Nous utilisons SQL car nous nous servons de données fixe. |
| Système de connexion restaurateur | L’utilisateur qui compte créer des menus et non pas seulement les visionner doit pouvoir se créer un compte et se connecter. | Modification du formulaire de connexion. | Une case ‘Professionnel’ sera à cocher sur le formulaire de connexion afin d’accéder aux permissions réservées aux restaurateurs comme la création de menu. | 1. Nous utiliserons une API crée avec Python pour le système de connexion 2. Séparer les utilisateurs lambdas des restaurateurs va nous permettre de faire de la publicité ciblé |
| Diffuser un menu | L’utilisateur doit pouvoir diffuser ses menus afin de les partager au public | Un lien permettant de partager le menu choisi via différents réseaux sociaux comme Instagram ou twitter ainsi que Deliveroo. L’utilisateur pourra également exporter le menu en PDF. | En cliquant sur le logo du réseau choisi l’utilisateur pourra partager facilement son menu depuis le site. Pour cela nous aurons besoin de l’API d’Instagram et de Deliveroo. | 1. Partager les menus au public permettra de faire de la publicité au site et donc d’attirer d’autres utilisateurs 2. Partager son travail sera une source de satisfaction pour le restaurateur |
| Galerie photos | Le site doit avoir une page avec une galerie représentant un menu choisi sous différents formats. | Utilisation d’une galerie React | React dispose d’une galerie en carrousel déjà faite dans laquelle on pourra mettre les menus sous différents formats : https://www.npmjs.com/package/react-image-gallery | 1. Nous avons choisi React comme Framework pour le front-end, utiliser ses fonctionnalités est donc logique 2. La galerie aura différents formats d’image représentant les formats dans lesquels il est possible d’imprimer un menu |
| Imprimer un menu | Dans la galerie, le restaurateur doit pouvoir imprimer son menu sous en PDF dans le format choisi. | Python | Avec Python on convertira l’image du menu en format PDF afin que l’utilisateur puisse l’exporter. | 1. En convertissant le menu en PDF l’utilisateur pourra faire ce qu’il souhaite du menu 2. Après la création d’un menu personnalisé, il est logique de vouloir le conserver |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fonctionnalités envisageables à l’avenir** | | | | |
| **Besoin** | **Contraintes** | **Solution** | **Description de la solution** | **Justification (2 arguments)** |
| Rappel par mail | Si le restaurateur quitte le site au milieu du processus de commande d’impression alors un mail lui demandant s’il souhaite reprendre sa commande lui sera envoyé. | Javascript (EmailJS) | A l’aide d’EmailJS il est possible d’envoyer facilement un mail à l’utilisateur sans code serveur : https://www.emailjs.com/ | 1. Le rappel par mail permettra surement à certains clients de reprendre leurs commandes stoppées 2. Les assurances et site d’achats en ligne utilisent souvent ce système afin de générer un maximum de profit |
| Commander un menu imprimé | Dans la galerie, le restaurateur doit pouvoir commander le menu choisi dans un des différents formats choisis (menu, kakemono, flyers …). | Python | Avec Python on créera un script permettant de passer une commande du menu au format choisi | 1. Le choix du format est très important pour que l’utilisateur puisse donner la forme physique de son choix à son menu crée 2. La commande de menu imprimer permettra au site de générer des revenus à l’entreprise |

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

* *Mon choix pour le langage côté serveur se tourne vers Python car c’est un langage à la fois populaire et simple. Je le choisis également car je ne l’ai jamais utilisé mais que j’en ai beaucoup entendu parler ce qui me donne envie de l’essayer.*
* Nous allons avoir besoin d’une API car il y aura un système de connexion sur le site ainsi qu’un besoin de stocker des données. Nous utiliserons donc Python pour créer notre API.
* Base de données choisie : *Pour la base de données nous partons sur du SQL. Nous avons choisi ce langage car nous utiliserons des données fixe et non dynamique contrairement à ce que propose NO SQL par exemple. Ce langage parait être le plus adapté pour ce projet et j’ai d’ailleurs quelques bases dessus l’ayant déjà étudié.*

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

* Nous utiliserons le nom de domaine : ‘Qwenta.fr’. Ce nom est concis, unique et mémorable. Il dispose également d’un top-Domain level avec la géolocalisation en France.
* Comme hébergement nous utiliserons LVS qui propose des tarifs adaptés à notre demande ainsi que plusieurs services offerts. Leur datacenter est d’ailleurs en France ce qui est aussi un plus.
* Nous utiliserons le nom de domaine : ‘contact@qwenta.fr’. Ce mail est court et facilement mémorable. Il dispose également d’une extension de domaine rappelant le pays du site.

1. **Accessibilité**

* Le site doit être compatible avec les dernières versions des navigateurs comme google chrome (Version 111.0.5563.65), Firefox (Version 110.0.1), Microsoft Edge (Version 111.0.1661.41) etc…
* Pour l’instant nous travaillerons seulement sur l’accessibilité du site sur ordinateur. Les versions tablettes et mobiles seront envisagés plus tard en cas de gain de popularité du site.
* Le site doit être accessible aux malvoyants avec un système de lecture d’écran et de navigation au clavier.

1. **Services tiers**

* Plusieurs outils seront utilisés pour le bon fonctionnement du site :

- Utilisation de l’API d’Instagram et Deliveroo pour les fonctions de partage

* Outils et plugins envisageable à l’avenir :

- Google search console nous permettra de savoir comment les utilisateurs sont arrivés sur notre site

- Un plugin nous informant dès qu’une erreur survient lors de la création d’un menu ou d’autres actions de l’utilisateur

- Un plugin qui enverrait un mail à un utilisateur s’il n’a pas fini la création de sa menu ou sa commande d’impression pour lui rappeler de le faire

1. **Recommandations en termes de sécurité**

* La sécurité du site est l’un des aspects les plus importants, voici donc plusieurs mesures prises :
* Nous utiliserons un lien ‘https’ et non ‘http’ malgré le coût supplémentaire afin de chiffrer les données
* Lors de l’inscription le mot de passe devra être conforme à plusieurs critères de sécurité

(Majuscule, chiffres, caractères spéciaux, longueur minimale...)

* Nous chiffrerons les mails ainsi que les mots de passe dans la base de données afin d’avoir une sécurité supplémentaire
* En cas de plusieurs échecs à l’entrée du mot de passe, un plugin enverra un message au mail utilisé pour le prévenir
* Des mises à jour régulières des langages et des librairies auront lieu afin de s’assurer d’avoir les dernières versions les plus sécuritaires possible

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

* Grandes lignes du contrat de maintenance :

- Le site ne sera pour l’instant disponible seulement de 6h à 23h afin de réaliser des économies

- En cas de besoin de maintenance logiciel du site, nous interviendrons dans les 48h

- En cas d’autre problèmes de maintenance nous contacterons l’hébergeur afin de régler le problème au plus vite.

- Pour une demande d’ajout il faudra compter une semaine avant que l’on commence à s’en charger

- Des mises à jour hebdomadaires auront lieu tous les lundis

- En fonction du gain de popularité du site, l’hébergement sera ré-évalué