Spécifications techniques

|  |  |
| --- | --- |
| **Projet** | [Menu Maker by Qwenta + Qwenta] |

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Auteur** | **Date** | **Approbation** |
| 1.0 | [Soufiane, Webgencia] | [26 Septembre 2023] | [John, Qwenta] |

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)** |
| *Ex. :*  *Création d'une catégorie de menu* | *Ex. :*  *L'ajout d'une catégorie doit pouvoir se faire directement sur l'écran de création de menu depuis une modale.* | *Ex. :*  *react-modal* | *Ex. :*  *Cette librairie React permet de créer simplement des modales performantes, accessibles avec un minimum de code.* | *Ex. :*  *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.* |
| *Création d’un outil en ligne dynamique permettant aux restaurateurs de publier, partager, mettre en forme et imprimer leurs menus.* | *Les restaurateurs doivent se connecter à un espace privé et sécurisé pour créer et modifier les menus.* | *Framework Front- end :* ***REACT***  *Framework Backend:* ***EXPRESS JS*** *(Language NodeJS)* | *Elle est utilisée pour créer des applications web à petite ou grande échelle et est particulièrement utile pour créer des sites web interactifs.* | *1) Nous avons choisi de développer en React et Express JS 2) Il s'agit de la librairie très utilisée. (Facilite la compréhension)*  *3) Utilise les fonctionnalités de Full-stack JS ( économiser du temps et de l’argent)* |

**Pourquoi choisir REACT ET EXPRESS ?**

ReactJS est très **rapide**, **La communauté** est grande, ReactJS est simple à lire, Le Javascript plus **simple** à écrire.

Avec EXPRESS Grâce à l’utilisation d’un serveur NodeJS, le code va pouvoir être généré **côté client ET côté serveur en Javascript** et donc faire appel à un seul Développeur **Full-stack** JS.

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

* Quel langage pour le serveur ? *Ex. :* *NodeJS / PHP / Python, etc.*

***Qu’est-ce que Node.js et pourquoi l’utiliser ?***

Node.js est un environnement d’exécution single-thread, open-source et multi-plateforme permettant de créer des applications rapides et évolutives côté serveur et en réseau. Il fonctionne avec le moteur d’exécution JavaScript V8 et utilise une architecture d’E / S non bloquante et pilotée par les événements, ce qui le rend efficace et adapté aux applications en temps réel.

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

***Pourquoi utiliser Node JS pour construire une API REST ?***

1.Pour la construction d’une API Node JS est un choix qui est souvent pertinent pour les raisons suivantes:

2.Son traitement non bloquant des requêtes

3.Sa performance et sa scalabilité 4.L’écosystème JavaScript et les

packages open source disponibles

* Base de données choisie : *Ex : SQL / NO SQL*.

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

* Nom du domaine.

Nom du domaine. (*Le nom de domaine sera très probablement un sous-domaine de Qwenta*)

* Nom de l’hébergement.
* Adresses e-mail.

1. **Accessibilité**

* Compatibilité navigateur.

Il est d'une importance primordiale de vérifier si votre site ou application s'affiche correctement sur les différents supports (ordinateurs, tablettes, smartphones) et en fonction des différents navigateurs. **Cela permet, en plus de répondre aux attentes des internautes, d'être mieux référencé dans les résultats de recherche**.

* Types d’appareils.

Le site devra être en version desktop. Pas de version mobile à développer ni à prévoir.

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).

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.