Webgencia

Spécifications techniques

Projet : Menu maker de Qwenta

Version	Auteur	Date	Approbation
1.0	Hosna	07/09/2024	John, Qwenta

Webgencia

I. 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	L'utilisateur doit comprendre l'utilité de l'outil pour créer un menu.	React, CSS, SASS	React gère les composants dynamiques pour une UX fluide et modulaire. CSS/SASS pour un design cohérent.	1) React permet une gestion fluide des états et des composants dynamiques. 2) Code modulaire, facilitant l'évolution du projet.
Connexion/ Création de compte	Authentification sécurisée avec un stockage sécurisé des informations utilisateurs	Firebase Authentication.	Utilisation de Firebase Authentication pour gérer l'inscription, la connexion et la déconnexion des utilisateurs.	1) Firebase Authentication est facile à intégrer avec un front-end React grâce à son SDK. 2) Il offre une gestion sécurisée des sessions utilisateur avec des outils intégrés comme OAuth.
Déconnexion	L'utilisateur doit pouvoir se déconnecter facilement et rapidement, supprimant sa session active.	API & JWT (JSON Web Token)	L'API permet de supprimer le jeton JWT du localStorage ou des cookies, révoquant ainsi la session de l'utilisateur.	1) L'API permet de gérer l'invalidation des sessions de manière centralisée et sécurisée. 2) L'utilisation de JWT permet un contrôle d'accès plus précis et sécurisé pour la déconnexion.
Création de menu	L'utilisateur doit pouvoir créer, ajouter et organiser un menu facilement.	React, Modale	Utilisation de React pour gérer l'interface dynamique, avec des modales pour ajouter des éléments au menu.	React gère fluidement les états et événements des modales. Les modales offrent une interface simple et centrée, sans rechargement de page.

Personnaliser un menu	Options de personnalisation intuitives et visuelles (image , titre, ingrédients)	Firebase Firestore et Firebase Cloud Storage.	Les données des menus personnalisés sont stockées dans Firestore, et les images sont hébergées dans Firebase Cloud Storage.	1) Firestore est une base NoSQL rapide et scalable qui s'intègre bien avec Firebase Authentication. 2)Firebase Cloud Storage offre un stockage rapide et sécurisé pour les fichiers médias.

Stocker les images	Besoin d'un stockage sécurisé et rapide pour les images téléchargées	Firebase Cloud Storage.	Les images téléchargées sont stockées sur Firebase Cloud Storage, avec des règles de sécurité basées sur Firebase Authentication.	1)Intégration directe avec Firebase Authentication pour des permissions utilisateur. 2)Haute disponibilité et gestion des fichiers simplifiée
Exportation du menu en PDF	Conversion des menus créés en un fichier PDF imprimable et partageable	jsPDF ou html2pdf.js	Ces bibliothèques permettent de convertir dynamiquement les éléments HTML en fichiers PDF, tout en offrant un contrôle précis sur la mise en page.	1) Facile d'utilisation avec React. 2) Contrôle total sur la mise en page du fichier PDF généré.
Diffuser le menu sur instagram	Partager les images du menu optimisées pour Instagram	Sharp & API Instagram	Utilisation de **Sharp** pour redimensionner et optimiser les images avant publication, puis utilisation d'un outil tiers ou de l'**API Instagram** pour publier les images.	1) **Sharp** permet d'optimiser les images pour Instagram. 2) L'API Instagram ou des outils comme Buffer permettent de publier automatiquement ou manuellement les images.
Imprimer un menu	Permettre à l'utilisateur d'imprimer le menu directement depuis le site	`window.print()` + CSS `@media print`	Utilisation de `window.print()` pour ouvrir le panneau d'impression, avec des styles CSS spécifiques pour l'impression.	1) Permet une impression sans redirection externe. 2) Personnalisation complète de la mise en page pour l'impression via CSS.

II. Liens avec le back-end

- **Quel langage pour le serveur ?**

J'ai choisi Node.js pour le serveur, car c'est un langage très rapide et bien adapté pour gérer plusieurs utilisateurs en même temps. Il utilise JavaScript, ce qui permet d'avoir un code plus cohérent entre le front-end et le back-end, puisque les deux utilisent le même langage. Node.js est aussi connu pour sa performance et sa capacité à gérer des demandes de manière asynchrone, ce qui est super pour les applications web modernes.

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

Oui, une **API RESTful** sera mise en place pour assurer la communication entre le front-end et le back-end. Cette API structurera les échanges de données (menus, utilisateurs, commandes, etc.) de manière efficace. Avec l'intégration de Firebase, certaines fonctionnalités comme l'authentification, la base de données et le stockage peuvent être directement gérées via son SDK, ce qui simplifie considérablement le développement.

- **Base de données choisie :**

Nous avons choisi Firebase pour gérer la base de données. Firebase propose Cloud Firestore, une base de données NoSQL évolutive et flexible. Elle permet de stocker les données sous forme de documents et collections, tout en offrant une synchronisation en temps réel entre les utilisateurs. Firebase simplifie également l'intégration avec le back-end grâce à ses outils natifs pour la gestion des données, l'authentification, et le stockage.

III. Préconisations concernant le domaine et l'hébergement

Nom du domaine.

- Le nom de domaine sera probablement un sous-domaine d'un domaine principal comme `qwenta.com`.
- Si ce n'est pas le cas, nous pourrions nous tourner vers : `monmenu.com` ou `menuexpress.fr`, selon le public cible et la disponibilité du domaine.

Nom de l'hébergement:

- Nous avons choisi OVH pour l'hébergement de notre site, tout en utilisant Firebase pour gérer les services backend tels que la base de données, l'authentification, et le stockage, avec une communication fluide entre les deux via le SDK Firebase.

Adresse e-mail.

- Si nous utilisons le nom de domaine qwenta.com, l'adresse e-mail professionnelle pourrait être : `contact@qwenta.com` ou `support@qwenta.com`.

IV. Accessibilité

- **Compatibilité navigateur:**

L'application sera compatible avec les navigateurs principaux comme Chrome, Firefox, Edge, et Safari, en garantissant une expérience fluide et sans problèmes sur les versions modernes de ces navigateurs.

- **Accessibilité pour tous les utilisateurs :**

L'application doit être accessible à un large public, y compris aux utilisateurs ayant des besoins spécifiques en matière de technologie d'assistance. Pour cela, nous avons pris en compte les recommandations des **Web Content Accessibility Guidelines (WCAG).**Par exemple, elle prendra en charge la navigation par clavier et sera compatible avec les lecteurs d'écran, facilitant l'accès aux utilisateurs malvoyants ou non-voyants.

- **Types d'appareils:**

L'application sera principalement optimisée en mode desktop.

V. Recommandations en termes de sécurité

- **Authentification** : Utiliser des JSON Web Tokens (JWT) pour sécuriser les comptes.
- **Chiffrement des mots de passe** : Protéger les mots de passe avec des algorithmes de hachage forts comme **bcrypt**.
- **Mises à jour et audit** : Maintenir les plugins et dépendances à jour, et utiliser des outils comme **Dependabot** pour surveiller les vulnérabilités.
- **Accès limité** : Configurer des permissions selon les rôles utilisateurs, appliquant le **principe du moindre privilège**.

VI. Maintenance du site et futures mises à jour

- Grandes lignes du contrat de maintenance

Surveillance des erreurs : Faire un suivi régulier pour identifier et corriger rapidement les bugs qui pourraient survenir.

Mises à jour de sécurité : Appliquer les mises à jour de sécurité dès qu'elles sont disponibles pour garantir la sécurité des données.
Améliorations continues : Ajouter de nouvelles fonctionnalités et améliorer celles qui existent déjà en fonction des retours des utilisateurs.
Optimisation : Maintenir les performances du site pour offrir une navigation fluide et rapide.
Support technique : Fournir un support aux utilisateurs pour résoudre les éventuels problèmes rencontrés.