**Solutions techniques**

**Projet Menu Maker by Qwenta.**

**I.Choix Technologiques:**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Besoin** | **Contraintes** | **Solution** | Description  de la solution | **Justification (2 arguments)** |
| Authentification (Connexion) | Les utilisateurs doivent pouvoir se connecter avec leur adresse e-mail | Firebase Auth | Utilisation de Firebase Authentication pour gérer l'authentification des utilisateurs | 1) Firebase offre une solution rapide et sécurisée.  2) Intégration facile avec React. |
| Authentification (Déconnexion) | Les utilisateurs doivent pouvoir se déconnecter facilement | Firebase Auth | Utilisation de Firebase Authentication pour gérer la déconnexion des utilisateurs | 1) Cohérence avec le choix de Firebase pour l'authentification. 2) Gestion simplifiée des sessions. |
| Tableau de bord (Dashboard) | Interface claire et intuitive avec accès rapide aux sections clés | React Router | Utilisation de React Router pour gérer la navigation entre les sections | 1) React Router est une solution standard pour la navigation dans les applications React.  Flexibilité et facilité d'utilisation. |
| Création de menu (Ajout de catégories) | L'ajout d'une catégorie doit pouvoir se faire directement sur l'écran de création de menu depuis une modale | react-modal | Utilisation de la librairie react-modal pour créer des modales performantes et accessibles | 1) Cohérence avec l'utilisation de React.  2) Simplicité d'implémentation et large adoption. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Besoin** | **Contraintes** | **Solution** | Description  de la solution | **Justification (2 arguments)** |
| Personnalisation (Branding) | Les utilisateurs doivent pouvoir enregistrer leurs préférences de branding (logo, etc.) | Cloudinary | Utilisation de Cloudinary pour gérer le téléchargement et le stockage des images | 1) Cloudinary est une solution robuste pour la gestion des images.  2) Facilite le traitement et l'optimisation des images. |
| Personnalisation (Style de texte) | Choix dynamique de la police et de la couleur | Google Fonts API | Utilisation de l'API Google Fonts pour intégrer et sélectionner différentes polices | 1) Large choix de polices.  2) Facilité d'intégration avec React |
| Sauvegarde des menus | Les utilisateurs doivent pouvoir sauvegarder un menu en cours de création pour y revenir plus tard | Firebase Firestore | Utilisation de Firebase Firestore pour le stockage des menus | 1) Intégration facile avec Firebase Auth.  2) Solution de base de données NoSQL performante. |
| Exportation PDF | Exporter les menus créés au format PDF | jsPDF | Utilisation de la librairie jsPDF pour générer des PDF directement dans le navigateur | 1) jsPDF est une solution légère et efficace.  2) Permet de générer des PDF sans serveur. |
| Diffusion en ligne (Deliveroo) | Intégration pour diffuser le menu sur Deliveroo | API Deliveroo | Utilisation de l'API Deliveroo pour partager les menus | 1) Permet une intégration directe avec Deliveroo.  2) Simplifie la gestion des menus sur la plateforme. |
| Diffusion en ligne (Instagram) | Partage direct du menu sur Instagram | API Instagram | Utilisation de l'API Instagram pour partager les menus | 1) Permet une intégration directe avec Instagram.  2) Augmente la visibilité des menus sur les réseaux sociaux. |
| Impression des menus | Les utilisateurs peuvent imprimer leurs menus directement | print-js | Utilisation de print-js pour gérer l'impression des documents | ) print-js est une solution simple pour l'impression.  2) Compatible avec tous les navigateurs modernes. |

**II. Liens avec le back-end:**

**Langage pour le serveur:**

**Recommandation :** Node.js

**Justification :**

1. JavaScript partout : Utiliser Node.js permet d'avoir une cohérence en utilisant

2. JavaScript à la fois pour le front-end et le back-end.

3. Écosystème riche : Node.js possède un écosystème de modules et de bibliothèques très vaste grâce à npm (Node Package Manager), facilitant l'ajout de fonctionnalités et l'accélération du développement.

4. Performance et scalabilité : Node.js est connu pour sa performance en temps réel et sa capacité à gérer un grand nombre de connexions simultanées, ce qui est avantageux pour une application dynamique comme "Menu Maker".

-A-t-on besoin d’une API ?

Réponse : Oui

Type d'API recommandée : RESTful API avec Express.js (pour Node.js)

Justification :

1. Modularité : Une API RESTful permet de structurer l'application en différentes parties modulaires, facilitant la maintenance et l'évolution du projet.

2. Simplicité et popularité : Express.js est un framework minimaliste pour Node.js qui permet de créer des API RESTful de manière simple et efficace. C'est une solution très populaire et bien documentée.

3. Interopérabilité : Les API RESTful sont indépendantes des langages de programmation et des technologies de front-end, permettant une intégration facile avec diverses plateformes et clients.

Intégration avec des services externes : Utilisation d'API spécifiques pour l'intégration avec des services externes comme Instagram et Deliveroo.

**API Instagram :**

But : Permettre le partage direct des menus sur Instagram.

Utilisation : Utilisation de l'API Instagram pour poster des images et des descriptions des menus sur le profil Instagram des restaurants.

Justification : Augmente la visibilité des menus et permet aux restaurants de partager facilement leurs menus avec leurs followers.

**API Deliveroo :**

But : Diffuser les menus directement sur Deliveroo.

Utilisation : Utilisation de l'API Deliveroo pour publier et mettre à jour les menus des restaurants sur la plateforme Deliveroo.

Justification : Facilite la gestion des menus sur une plateforme de livraison populaire, augmentant ainsi la portée et les ventes potentielles des restaurants.

Base de données choisie

Recommandation : NoSQL (Firebase Firestore)

Justification :

Scalabilité : Les bases de données NoSQL comme Firebase Firestore sont conçues pour être facilement évolutives, ce qui est important pour une application potentiellement utilisée par de nombreux restaurants.

Simplicité d'intégration : Firebase Firestore s'intègre facilement avec Firebase Authentication, facilitant la gestion des utilisateurs et de leurs données.

Temps réel : Firestore offre des mises à jour en temps réel, ce qui est utile pour une application dynamique où les menus peuvent être modifiés et affichés instantanément.

Gestion automatique : Firebase gère automatiquement la synchronisation des données et les mises à jour, réduisant la charge de gestion sur les développeurs.

**Résumé:**

**Langage pour le serveur** : **Node.js**

API nécessaire : Oui, RESTful API avec Express.js

API externes :

API Instagram pour le partage des menus sur Instagram.

API Deliveroo pour la diffusion des menus sur Deliveroo.

Base de données choisie : NoSQL, Firebase Firestore

Ces choix offrent une architecture moderne, évolutive et efficace, facilitant le développement, la maintenance et l'intégration avec des services externes pour l'application "Menu Maker by Qwenta".

**III. Préconisations concernant le domaine et l’hébergement:**

**Nom du domaine : menumaker.qwenta.com**

**Justification :**

Cohérence de marque : Utiliser un sous-domaine de qwenta.com renforce l'identité de marque de Qwenta et montre que Menu Maker fait partie de leur offre de services.

Simplicité et mémorabilité : Un nom de domaine clair et concis est plus facile à retenir pour les utilisateurs.

SEO : Avoir un domaine lié à la marque principale peut améliorer le référencement sur les moteurs de recherche.

**Nom de l’hébergement:**

**Recommandation :**

**Nom de l'hébergement : Firebase Hosting**

**Justification :**

Intégration facile : Firebase Hosting s'intègre parfaitement avec Firebase Firestore et Firebase Authentication, qui sont déjà recommandés pour ce projet.

Performance : Firebase Hosting offre un CDN (Content Delivery Network) mondial, garantissant des temps de chargement rapides et une haute disponibilité.

Simplicité de déploiement : Firebase Hosting permet de déployer des applications web en quelques étapes simples, ce qui accélère le processus de mise en ligne.

Sécurité : Firebase Hosting gère automatiquement les certificats SSL, assurant que les connexions sont sécurisées.

Évolutivité : Firebase Hosting peut gérer une grande quantité de trafic, s'adaptant facilement aux besoins croissants de l'application sans nécessiter une gestion complexe de l'infrastructure.

**Adresse e-mail : contact@menumaker.qwenta.com**

**Résumé:**

Nom du domaine : menumaker.qwenta.com

Nom de l'hébergement : Firebase Hosting

Ces préconisations assurent une solution de domaine et d'hébergement qui est à la fois cohérente avec l'image de marque de Qwenta, facile à gérer et capable de fournir une performance et une sécurité de haut niveau pour l'application "Menu Maker".

**IV. Accessibilité:**

**Compatibilité navigateur:**

**Recommandation :** Assurer la compatibilité avec les navigateurs suivants :

Google Chrome

Mozilla Firefox

Safari

**Microsoft Edge**

**Justification :**

Part de marché : Ces navigateurs couvrent la majorité des utilisateurs d'Internet.

Standards web : Tous ces navigateurs respectent les standards web modernes, garantissant une expérience utilisateur cohérente et de haute qualité.

Performance et sécurité : Ces navigateurs sont régulièrement mis à jour pour améliorer les performances et corriger les vulnérabilités de sécurité.

**Types d’appareils:**

**Recommandation :**

Assurer la compatibilité avec les types d'appareils suivants :

Ordinateurs de bureau (Windows, macOS, Linux)

Tablettes (iOS, Android)

Smartphones (iOS, Android)

**Justification :**

Accessibilité : Les restaurateurs et les clients doivent pouvoir accéder à "Menu Maker" depuis divers appareils.

Utilisation mobile : Une part croissante des utilisateurs accède aux services en ligne via des appareils mobiles.

Expérience utilisateur : Une conception responsive assure une bonne expérience utilisateur quel que soit l'appareil utilisé.

**V. Recommandations en termes de sécurité:**

**Accès aux comptes:**

**Recommandation :** Utiliser Firebase Authentication pour la gestion des comptes utilisateurs.

Authentification multi-facteurs (MFA) : Ajouter une couche supplémentaire de sécurité en activant l'authentification à deux facteurs (2FA).

Hachage des mots de passe : Assurer que les mots de passe sont hachés et sécurisés avant d'être stockés.

**Justification :**

Simplicité d'intégration : Firebase Authentication s'intègre facilement avec Firebase Firestore.

Sécurité renforcée : La 2FA réduit le risque de compromission des comptes.

**Plugins de sécurité:**

**Recommandation :**

Utiliser des plugins de sécurité pour surveiller et protéger l'application, comme Helmet.js pour Express.js.

**Justification :**

Protection contre les attaques courantes : Helmet.js aide à protéger l'application contre les attaques XSS, de clickjacking, etc.

Meilleures pratiques : Utiliser des plugins de sécurité suit les meilleures pratiques en matière de développement sécurisé.

**VI. Maintenance du site et futures mises à jour:**

**Grandes lignes du contrat de maintenance:**

**Contrat de maintenance :**

1. **Support technique :**
   * Disponibilité : Assurer une disponibilité du support technique pendant les heures ouvrables.
   * Canaux de communication : Fournir un support via e-mail, téléphone et chat.
2. **Mises à jour régulières :**
   * Mises à jour de sécurité : Appliquer les patchs de sécurité dès qu'ils sont disponibles pour protéger contre les vulnérabilités.
   * Mises à jour fonctionnelles : Ajouter de nouvelles fonctionnalités et améliorer les fonctionnalités existantes en fonction des retours des utilisateurs.
3. **Surveillance et monitoring :**
   * Surveillance continue : Utiliser des outils de monitoring pour surveiller la performance et la disponibilité du site.
   * Rapports mensuels : Fournir des rapports réguliers sur l'état du site, incluant les incidents et les mesures correctives prises.
4. **Sauvegardes régulières :**
   * Fréquence : Effectuer des sauvegardes quotidiennes de la base de données et des fichiers importants.
   * Restauration : Assurer un processus de restauration rapide en cas de perte de données.
5. **Gestion des incidents :**
   * Plan de réponse aux incidents : Mettre en place un plan de réponse rapide en cas de défaillance ou de cyberattaque.
   * Délai de réponse : Fixer des délais de réponse et de résolution en fonction de la gravité des incidents.

**Justification :**

Fiabilité : Un contrat de maintenance assure que le site reste fiable et sécurisé.

Continuité : Assurer la continuité des services et des améliorations en fonction des besoins des utilisateurs.

Sécurité des données : Protéger les données des utilisateurs et maintenir la confiance des clients.

**Résumé:**

* Compatibilité navigateur : Chrome, Firefox, Safari, Edge
* Types d’appareils : Ordinateurs de bureau, tablettes, smartphones
* Sécurité : Accès aux comptes via Firebase Authentication avec MFA, plugins de sécurité comme Helmet.js
* Contrat de maintenance :
  1. Support technique
  2. Mises à jour régulières
  3. Surveillance et monitoring
  4. Sauvegardes régulières
  5. Gestion des incidents

Ces préconisations assurent une application accessible, sécurisée et maintenable, offrant ainsi une expérience utilisateur optimale et une gestion efficace du projet "Menu Maker by Qwenta".