Webgencia

Spécifications techniques

Menu Maker - Qwenta

Version	Auteur	Date	Approbation
1.0	Noam, Webgencia	27/03/2025	John, Qwenta

I. Choix technologiques

Tableau des spécifications techniques

Besoins	Contraintes	Solutions techniques	Description	Justifications
Gestion des utilisateurs (authentification, connexion, gestion des rôles)	Sécurité des données, compatibilité avec les navigateurs	Authentification via JWT avec Next.js et MongoDB	Sécurisation des comptes, gestion des sessions et accès API	1. JWT permet une gestion fiable des sessions sans stocker d'informations sensibles sur le serveur. 2. Next.js permet une gestion centralisée des API et de l'interface utilisateur, optimisant les performances.
Interface utilisateur dynamique (parcours utilisateur interactif)	UI fluide et rapide	Next.js avec Tailwind CSS	Développement d'une interface moderne et interactive	1. Next.js assure un rendu côté serveur et améliore le SEO. 2. Tailwind CSS permet une conception responsive rapide et efficace.
Base de données (gestion des menus, restaurants, utilisateurs)	Données volumineuses et requêtes rapides	MongoDB hébergé sur un serveur self-managed	Stockage flexible des menus et des informations des utilisateurs	1. NoSQL offre une grande souplesse pour stocker des données non structurées. 2. Scalabilité adaptée à une base de données en croissance.

API REST pour l'interaction avec le front	Sécurisation et rapidité	API interne via Next.js	Centralisation de l'API et du rendu dans Next.js	1. Architecture optimisée : une seule stack pour front et back. 2. Gestion simplifiée des routes et des appels API.
Hébergement (site et API)	Haute disponibilité	VPS sous Linux (AlmaLinux ou Debian) avec Docker	Infrastructure flexible, évolutive et facile à maintenir	1. Scalabilité et contrôle facilités avec Docker. 2. Isolation des services pour éviter les conflits et améliorer la gestion des versions.
Gestion des images (logos, menus, photos de restaurant)	Stockage et temps de chargement	Stockage sur un service cloud type S3 (AWS)	Optimisation des images et rapidité de chargement	1. Stockage performant et accessible rapidement. 2. Outils d'optimisation intégrés pour améliorer les performances.
SEO et accessibilité	Respect des standards Web	Optimisation SEO avec Next.js et rich snippets Schema.org	Amélioration de la visibilité sur les moteurs de recherche	1. Next.js améliore le SEO grâce au rendu côté serveur. 2. Rich snippets Schema.org facilitent l'indexation et l'expérience utilisateur.
Sécurité et conformité **RGPD**	Protection des données clients	Chiffrement des données, consentement cookies	Sécurisation des informations utilisateurs et conformité légale	1. Protection des données via chiffrement et sécurisation des accès. 2. Gestion des cookies conforme aux réglementations RGPD.

II. Liens avec le back-end

Next.js intègre à la fois le front-end et le back-end, permettant ainsi de gérer l'ensemble des traitements au sein d'une seule application. Contrairement aux architectures traditionnelles où le back-end est séparé (comme avec Express.js ou NestJS), Next.js propose une approche hybride avec des **API Routes**, qui permettent d'écrire du code serveur directement dans l'application sans avoir besoin d'un serveur back-end distinct.

Langage utilisé : Next.js comme solution full-stack

Next.js repose sur **Node.js**, ce qui lui permet d'exécuter du code serveur en plus du rendu côté client. Il peut fonctionner en mode **serverless** (où chaque API Route est une fonction indépendante) ou être hébergé sur un **serveur dédié**.

Contrairement à Express.js ou NestJS, qui nécessitent de configurer un serveur séparé et de gérer manuellement les routes et middlewares, Next.js simplifie cette gestion en intégrant directement les API au sein du projet. Cela réduit la complexité et améliore la rapidité de développement, tout en permettant une bonne scalabilité.

Gestion des API avec Next.js

Next.js facilite la mise en place d'API grâce à son système de fichiers : chaque fichier dans `/app/api/` devient une route API accessible depuis le front-end.

Cette approche élimine le besoin d'un serveur API externe comme Express.js, réduisant ainsi la latence et améliorant les performances globales. Contrairement à des solutions comme NestJS, qui imposent une structure plus rigide et une configuration plus lourde, Next.js permet une mise en place rapide et intuitive des endpoints.

Base de données : Pourquoi MongoDB ?

MongoDB est choisi pour sa flexibilité et sa capacité à gérer des données non structurées, ce qui convient bien aux besoins de l'application.

Avec Next.js, l'intégration de MongoDB est simple grâce aux **API Routes**, qui permettent d'établir des connexions directement depuis le serveur sans nécessiter d'application backend supplémentaire. Contrairement à SQL, qui impose une structure rigide, MongoDB permet une gestion plus dynamique des données, ce qui s'adapte mieux aux applications évolutives.

En résumé, Next.js offre une approche plus intégrée et optimisée que les solutions classiques, en réduisant la complexité tout en assurant de bonnes performances et une scalabilité adaptée aux besoins du projet.

II. Préconisations concernant le domaine et l'hébergement

- **Nom du domaine** :
  Le domaine **menu-maker.qwenta.com** est recommandé. Ce nom reflète clairement l'objectif du site tout en renforçant la marque Qwenta, favorisant ainsi la reconnaissance et le SEO.

- **Nom de l'hébergement** :
  L'hébergement se fera sur un **VPS sous AlmaLinux ou Debian**, offrant ainsi un contrôle total sur l'environnement d'exécution du site. Ce choix permet également une scalabilité facile en fonction de l'augmentation du trafic.

- **Adresses e-mail** :
  Les adresses e-mail seront configurées sous **info@menu-maker.qwenta.com**. Ce format professionnel améliore la crédibilité et la confiance des utilisateurs tout en permettant une gestion sécurisée des communications.

III. Accessibilité

- **Compatibilité avec les navigateurs** : Chrome, Firefox, Edge, Safari.
- **Accessibilité pour les personnes en situation de handicap** :
- Contraste élevé et options de personnalisation de l'interface.
- Navigation optimisée pour les lecteurs d'écran (ARIA labels, balises sémantiques).
- Navigation clavier et alternatives pour les contenus visuels.
- Test régulier de conformité avec WCAG 2.1.
- **Adaptabilité** : Responsive design pour une expérience optimale sur tous les appareils.

IV. Recommandations en termes de sécurité

- **Accès aux comptes** :
  L'authentification sera gérée par **JWT (JSON Web Tokens)**. Cette méthode permet de sécuriser les sessions utilisateurs tout en réduisant la charge sur le serveur, puisque les sessions sont gérées côté client.

- **Plugins et sécurité des informations** :
  Tous les plugins seront régulièrement **mis à jour** et un audit de sécurité sera effectué. Cela permet de protéger le site contre les vulnérabilités et de s'assurer que les données utilisateurs sont sécurisées, en conformité avec la réglementation.

V. Maintenance du site et futures mises à jour

Contrat de maintenance :

- Un **contrat de maintenance** couvrira les mises à jour mensuelles de sécurité et l'évolution fonctionnelle du site sur une période de **12 mois**. Ce contrat assurera également un support technique pour résoudre rapidement les problèmes