Webgencia

Spécifications techniques

[Menu Maker + Qwenta]

Version	Auteur	Date	Approbation
1.0	Mezouar Mohamed	03/08/85	Soufiane

I. Choix technologiques

1) État des lieux des besoins fonctionnels et de leurs solutions techniques :

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
Création d'une catégorie de menu	L'ajout d'une catégorie doit pouvoir se faire directement sur l'écran de création de menu depuis une modale.	react-modal	Cette librairie React permet de créer simplement des modales performantes, accessibles avec un minimum de code.	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.
Exportation de menu en PDF	Le menu doit être exportable en PDF sans altérer la mise en page et le design.	html-pdf	**html-pdf est un package Node.js qui permet de convertir des pages HTML/CSS en PDF de manière efficace.**	1) Solution rapide et simple à mettre en place pour l'exportation des menus. 2) Compatible avec les technologies backend utilisées (Node.js).
Diffusion de menu sur Deliveroo	Le menu doit pouvoir être diffusé directement sur Deliveroo via une API	Deliveroo API	**Deliveroo API permet de connecter les menus du site à la plateforme Deliveroo pour une**	1) Utilisation d'une API officielle garantit la compatibilité et la

	dédiée.		**diffusion rapide et automatisée.**	maintenance. 2) Intégration fluide avec le back-end en Node.js.
Authentification des restaurateurs	Les restaurateurs doivent pouvoir se connecter de manière sécurisée.	JWT (JSON Web Token) + SendGrid	**JWT permet une authentification sécurisée et la gestion des sessions utilisateurs côté serveur. en complément avec un système de vérification d'adresse e-mail géré par SendGrid pour renforcer la sécurité et éviter l'usage de mots de passe.**	1) JWT est léger et compatible avec Node.js et Express. 2) SendGrid est une solution robuste et bien adaptée à l'envoi sécurisé de courriers électroniques de vérification, facilitant ainsi l'intégration avec Node.js et Express.
Personnalisation de l'interface de menu	Les restaurateurs doivent pouvoir personnaliser l'apparence de leurs menus avec leurs couleurs et logos.	Scss	**permettra de profiter d'une syntaxe CSS native tout en gardant la structure et les fonctionnalités avancées du préprocesseur gère les styles de manière modulaire, en offrant des fonctionnalités avancées pour la personnalisation**	SCSS est largement utilisé aujourd'hui et offre une syntaxe plus intuitive et proche du CSS, facilitant la gestion des styles. La modularité de SCSS simplifie l'organisation des thèmes et la personnalisation visuelle des interfaces, essentiel pour un projet de menus personnalisables.
Base de données	Les restaurateurs doivent pouvoir stocker les données de leurs menus sur	MongoDB (NoSQL)	MongoDB permet une gestion flexible des données des menus, avec Mongoose pour	Simple a mettre en place, efficace et largement utilisé

	un serveur sécuriser		simplifier les opérations de manipulation de données.	

II. Liens avec le back-end

1) Quel langage pour le serveur ?

Node.js est un choix naturel pour ce projet en raison de ses performances élevées, de sa compatibilité avec le front-end (React), et de la facilité de gestion des requêtes asynchrones qui seront géré avec Async/Await , essentielles dans un environnement de développement web moderne. En outre, Node.js est largement adopté dans les applications web où la rapidité et la scalabilité sont des priorités, notamment pour des fonctionnalités telles que l'authentification, la gestion des menus en temps réel et les intégrations API.

- 2) A-t-on besoin d'une API ? Oui

  - Pour permettre la diffusion des menus directement sur Deliveroo, il est nécessaire d'intégrer l'API officielle de Deliveroo. Cette API permettra de publier les menus créés sur la plateforme Deliveroo de manière fluide et automatisée.
  - Pour permettre aux restaurateurs de partager leurs menus sur Instagram, l'API d'Instagram sera utile pour faciliter la diffusion des menus via ce réseau social.

- 3) PDF génération API :

  - Pour l'exportation des menus en PDF, tu utiliseras un package comme html-pdf dans Node.js qui permet de convertir des pages HTML/CSS en fichiers PDF.

4) Base de données choisie :

MongoDB est un excellent choix pour des projets dynamiques comme Menu Maker où les structures de données peuvent varier (par exemple, les menus des restaurateurs peuvent être très différents en termes de composition et de structure). MongoDB est flexible, performant et s'intègre facilement avec Node.js grâce à Mongoose, un ORM (Object Relational Mapping) qui facilite la manipulation des données.

III. Préconisations concernant le domaine et l'hébergement

- Nom du domaine. menumaker.qwenta.com
- Nom de l'hébergement. Heroku
- Adresses e-mail. admin@qwenta.com

IV. Accessibilité

1) Compatibilité :

- Compatibilité navigateur. Chrome, firefox, safari
- Types d'appareils. Descktop

2) Accessibilité visuelle :

Contraste élevé et tailles de texte ajustables Utilisé des variables CSS pour ajuster les couleurs et tailles de texte

Utilisé des outils comme Sass pour gérer les thèmes en facilitant l'ajustement des couleurs via des variables dynamiques.

3) Balises alt et compatibilité lecteur d'écran :

- utiliser des balises alt détaillées
- Teste la compatibilité avec des lecteurs d'écran tels que NVDA (Windows) ou VoiceOver (Mac)

4) Navigation au clavier :

- Implémente des fonctionnalités de tabindex pour que tous les éléments interactifs soient accessibles avec le clavier.
- Utilise des bibliothèques comme Axe Accessibility pour vérifier l'ordre de tabulation et l'accessibilité au clavier

5) Accessibilité motrice :

- S'assurer que tout les éléments interactifs (formulaires, boutons) sont accessibles au clavier via les balises tabindex et aria

6) Outils de tests d'accessibilité :

- Wave : Extension pour Chrome/Firefox qui permet de tester l'accessibilité d'une page Web et de vérifier si elle est conforme aux normes WCAG.
- Axe Accessibility : Un autre outil de test automatisé qui aide à détecter les problèmes d'accessibilité dans le code.
- Lighthouse : Intégré dans Chrome DevTools, il permet de vérifier les performances, l'accessibilité, et les meilleures pratiques d'un site.

7) Outils pour s'assurer de la conformité aux normes WCAG :

- AccessLint : Un outil qui s'intègre à GitHub pour tester l'accessibilité à chaque commit.
- Tenon.io : Un service qui aide à évaluer la conformité d'un site avec les directives WCAG 2.1 et ARIA

V. Recommandations en termes de sécurité

1. Sécurité de l'accès aux comptes utilisateurs

- Utiliser JWT pour la gestion des sessions utilisateurs, les tokens doivent être chiffrés (Pour cela, utiliser des algorithmes comme AES) et avoir une durée de vie limitée
- Utilisation de HTTPS **Sécuriser toutes les communications entre les clients et le serveur avec un** certificat TLS

2. Sécurisation des données sensibles :

- Hashing des mots de passe avec bcrypt

- Chiffrement des données sensibles avec crypto en utilisant AES-256

3. Plugins et dépendances de sécurité

- Vérification des dépendances (npm audit) pour vérifier que toutes les dépendances sont à jour et ne présentent pas de vulnérabilités connues.
- Helmet.js pour sécuriser les en-têtes HTTP. Installer le plugin Helmet.js dans l'application Express pour configurer des en-têtes HTTP sécurisés.
- Rate Limiting Mettre en place un système de limitation des requêtes avec express-rate-limit
- CORS (Cross-Origin Resource Sharing) Configurer correctement les règles CORS pour limiter les sites autorisés à accéder aux API.

4. Protection contre les attaques courantes

- Prévention des attaques CSRF (Cross-Site Request Forgery). **Utiliser un middleware comme csurf pour Express, qui génère des tokens CSRF afin d'empêcher des requêtes malveillantes non authentifiées de manipuler les actions des utilisateurs.**
- Protection contre les injections NoSQL Utiliser Mongoose pour éviter les injections NoSQL et valider soigneusement les entrées utilisateurs.
- Validation des entrées utilisateurs Utiliser des bibliothèques comme Joi pour valider et assainir toutes les entrées utilisateurs,

5. Sauvegardes et restauration des données

- Mise en place d'un système de sauvegarde régulier. **Planifier des** sauvegardes automatiques **de la base de données MongoDB pour éviter la perte de données en cas de panne ou d'attaque.**

- Veiller à ce que les sauvegardes soient stockées de manière sécurisée, avec un chiffrement fort, et que les procédures de restauration soient régulièrement testées.

VI. Maintenance du site et futures mises à jour

. **Durée du contrat : 12 ou 24 mois, renouvelable.**
- **Types de maintenance : Corrective, préventive, évolutive, et adaptative.**
- **Modalités d'intervention : Délai de réponse pour les bugs critiques et mineurs.**
- **Mises à jour de sécurité : Suivi des vulnérabilités, application de patchs, renouvellement SSL.**
- **Sauvegardes et restauration : Sauvegardes régulières et plan de récupération.**
- **Garantie de performance : Monitoring, rapports de performance, et optimisation continue.**
- **Coûts : Forfait mensuel ou annuel avec options pour les coûts additionnels.**
- **Fin du contrat : Modalités de résiliation et transmission des données.**