Webgencia

Spécifications Techniques

Menu Marker By Qwenta

Version	Auteur	Date	Approbation
1.0	Webgencia	00/10/2024	John, Qwenta

Le but de ce document est de définir et justifier les spécifications techniques de **Menu Maker By Qwenta**.

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 non Connectée	Avoir accès aux sections : - Bannière - Personnalisez votre menu - Explications étape par étape	HTML,CSS React	Utilisation de HTML, CSS et React pour créer la landing page. React rend la page interactive. Le CSS sera présent pour les animations comme la transition de la bannière entre les sections.	1. Interactivité avec React : Permet d'ajouter des composants dynamiques et de mettre à jour la page sans rechargement. 2. Animations avec CSS : Assure des transitions fluides et améliore le design visuel, sans affecter la performance.
Page de Login	Ouverture d'une modal, - L'utilisateur peu entrer sont adresse mail - Un mail est envoyé à l'utilisateur pour s'authentifier ou confirmer son adresse lors de sa première connexion à l'application. - Lien « Besoin d'aide » redirigera les utilisateurs vers un formulaire de contact.	HTML,CSS Node,js,Express MongoDB Brcypt JWT	Utilisation de React pour la modale de connexion/inscription. Le tableau de bord sera en HTML/CSS. Node.js/Express gérera les e-mails et la base de données, MongoDB stockera les données et gérera l'authentification sécurisée avec bcrypt pour le hachage des mots de passe. JWT sera utilisé pour les sessions et Nodemailer pour l'envoi d'e-mails de confirmation et de réinitialisation.	1. Sécurité et gestion efficace des utilisateurs : MongoDB, bcrypt et JWT assurent une authentification sécurisée, un stockage fiable des données, et une gestion des sessions sans compromettre la performance. 2. Envoi d'e-mails optimisé : Nodemailer, intégré avec Node.js/Express, permet une gestion fluide des e-mails de confirmation et réinitialisation, garantissant une expérience utilisateur sans interruptions.
Catégorie de plat	Le restaurateur peut créer ou choisir une catégorie de menu via une fenêtre modale depuis l'écran de création.	React Node,js, Express MongoDB	Utilisation de React pour la modale de création de catégorie. Le back-end, via Node.js/Express, interagira avec MongoDB pour gérer la création, validation et sélection des catégories de menu.	1. Modularité et réactivité : React permet de créer une interface interactive et réactive pour gérer facilement la création et la sélection de catégories via une modale. 2. Back-end robuste : Node.js/Express avec MongoDB assure une gestion fiable et sécurisée des catégories, avec des opérations de création et validation rapides grâce à une API dédiée.

Besoin	Contraintes	Solution	Description de la solution	Justification(2 Arguments)
Création de Plat	L'utilisateur doit pouvoir : - Sélectionner ou modifier une catégorie - Entrer les informations du plat dans une modale - Chaque plat inclut : - Une Photo - Un Nom - Un Prix - Une description -Possibilité de créer plusieurs plats	React Node,js, Express MongoDB	Utiliser React pour créer l'interface permettant au restaurateur de saisir les informations des plats. Pour le back-end, nous allons développer un service API avec Node.js et Express, qui interagira avec la base de données MongoDB. Cette API gérera les opérations de création et d'ajout de plats.	1. Interface utilisateur intuitive : React facilite la création d'une interface conviviale pour les restaurateurs, leur permettant de saisir rapidement et facilement les informations des plats. 2. API performante : L'utilisation de Node.js et Express pour le back-end, couplée à MongoDB, garantit une gestion efficace des opérations de création et d'ajout de plats, assurant une performance et une scalabilité optimales.
Style de Menu	Le restaurateur doit pouvoir : - Visualiser le menu crée - Sélectionner une typographie - Choisir une couleur de texte	React Node,js, Express MongoDB React Color Google Fonts	Utiliser React pour l'interface de personnalisation du menu. Node.js et Express créeront une API pour interagir avec MongoDB et récupérer les infos du menu. Intégrer React Color pour les couleurs et Google Fonts pour la typographie. Cette architecture permettra aux restaurateurs de personnaliser et d'enregistrer leurs préférences dans la base de données.	1. Facilité d'utilisation : L'utilisation de React pour l'interface de personnalisation permet aux restaurateurs d'interagir facilement avec leur menu. Les composants réactifs améliorent l'expérience utilisateur, rendant la personnalisation intuitive et accessible. 2. Flexibilité et performance : En combinant Node.js et Express pour l'API avec MongoDB, cette architecture permet une gestion efficace des données. Les restaurateurs peuvent récupérer et enregistrer rapidement leurs préférences de typographie et de couleurs, offrant une grande flexibilité dans la personnalisation de leur menu.

Besoin	Contraintes	Solution	Description de la solution	Justification(2 Arguments)
Exportation PDF	Le restaurant doit pouvoir télécharger son menu crée en PDF.	bibliothèque JavaScript React PDF pour générer des PDF	Pour cette fonctionnalité, nous utiliserons une bibliothèque JavaScript spécialisée dans la génération de fichiers PDF à partir des données de l'application. Une bibliothèque couramment utilisée est React-PDF. Nous ajouterons un bouton "Exporter en .pdf" qui, lorsqu'il sera cliqué, générera le PDF contenant les informations du menu actuel et proposera le téléchargement de ce fichier.	1. Facilité de génération : L'utilisation de jsPDF simplifie la création de fichiers PDF à partir des données de l'application, permettant ainsi un processus fluide pour les restaurateurs. 2. Accessibilité : Le bouton "Exporter en .pdf" offre aux utilisateurs une méthode directe et pratique pour télécharger rapidement leur menu au format PDF.
Commander l'impression d'un menu	Le restaurateur, dois pouvoir commander en un clic l'impression d'un menu - L'encart "Imprimer un menu" doit être visible depuis la page d'accueil - Se Lien ouvre un nouvelle onglet - Qui dois fait vers le back-office de Qwenta	URL de back-office	Pour cette fonctionnalité, nous utiliserons une URL qui mènera directement au back office de Qwenta. Il n'est pas nécessaire de créer une fonction d'impression personnalisée, car les navigateurs modernes ont des options d'impression avancées. Le lien "Imprimer un menu" sur la page d'accueil ouvrira simplement un nouvel onglet avec cette URL.	1. Simplicité d'accès : L'utilisation d'une URL vers le back office facilite la navigation pour les utilisateurs, leur permettant d'accéder rapidement à la fonction d'impression. 2. Optimisation des ressources : En s'appuyant sur les capacités d'impression des navigateurs modernes, nous évitons le besoin de développer une solution complexe, ce qui simplifie le processus de mise en œuvre.
Menus précédents	Lorsqu'un restaurateur clique sur "Mes menus", il accède à la liste de ses menus créés, accompagnée de leur date de création. Il a la possibilité de modifier ou de supprimer un menu existant, ainsi que de créer un nouveau menu directement depuis cette interface	HTML, CSS React Router MongoDB	Mettre en place l'HTML et le CSS de la page "Mes menus" qui affiche les menus précédents. Nous utiliserons React Router pour gérer la navigation entre les pages. Les données des menus précédents seront stockées dans la base de données MongoDB et récupérées via une requête. Les fonctionnalités de modification et de suppression des menus seront gérées par des appels à l'API liée à MongoDB.	1. Navigation fluide : L'utilisation de React Router assure une navigation fluide entre les pages, améliorant ainsi l'expérience utilisateur lors de la consultation et de la gestion des menus. 2. Gestion efficace des données : En stockant les menus dans MongoDB et en utilisant des requêtes API, les restaurateurs peuvent facilement récupérer, modifier et supprimer leurs menus, garantissant ainsi une gestion efficace et rapide des informations.

Besoin	Contraintes	Solution	Description de la solution	Justification(2 Arguments)
Informations légales	En tant qu'internaute, il dois pouvoir accéder au contenu "Mentions légales" dans une modale, et l'information “Tous droits réservés” doit être affichée.	React React Router MongoDb	Pour cette fonctionnalité, nous utilisons React pour gérer l'ouverture de la modale et l'affichage des mentions légales. Le lien "Mentions légales" et la mention "Tous droits réservés" sont des composantes réutilisables ajoutées sur toutes les pages. La navigation et les modales sont gérées avec React Router. Les données des mentions légales, stockées dans MongoDB, sont récupérées à l'ouverture de la modale.	1. Optimisation de la maintenance : En utilisant des composantes réutilisables pour les mentions légales et les droits d'auteur, on garantit une uniformité sur l'ensemble des pages et on simplifie les mises à jour. Toute modification des informations légales se reflète instantanément sur toutes les pages, réduisant le risque d'incohérences. 2. Expérience utilisateur améliorée : La gestion des modales via React et la navigation avec React Router assurent une transition fluide entre les pages et un affichage rapide des mentions légales. Cela permet une expérience utilisateur plus dynamique et interactive, favorisant une meilleure accessibilité aux informations importantes.
Tarifs	L'internaute doit pouvoir cliquer sur l'onglet tarifs qui ouvre un nouvel onglet.	Lien vers une URL Fournie par Qwenta	Nous ajouterons un onglet Tarifs sur le site. Lorsqu'un utilisateur cliquera dessus, le lien s'ouvrira dans un nouvel onglet ou une nouvelle fenêtre grâce à l'attribut HTML target="_blank", affichant les informations tarifaires.	1. L'utilisation de target="_blank" permet à l'utilisateur d'accéder aux informations tarifaires sans quitter la page actuelle, améliorant ainsi son expérience de navigation. 2. Ouvrir les tarifs dans un nouvel onglet offre une navigation fluide, permettant à l'utilisateur de passer facilement entre les informations tarifaires et le site principal.

Besoin	Contraintes	Solution	Description de la solution	Justification(2 Arguments)
Exportation Deliveroo	- L'encart "Diffuser sur Deliveroo" doit s'afficher dans la catégorie "Exportez et diffusez" - Au clic sur l'encart, l'utilisateur doit être redirigé sur l'application Deliveroo	API fournie par Deliveroo MongoDb Node.js Express	Pour cette fonctionnalité, nous intégrons une API Deliveroo. En cliquant sur "Exporter sur Deliveroo", l'application envoie une demande pour créer ou mettre à jour le menu (plats, descriptions, prix) sur la plateforme Deliveroo. Le back-end utilise MongoDB pour gérer les informations du menu, et l'intégration de l'API Deliveroo est réalisée côté serveur avec Node.js et Express	1. Automatisation des mises à jour : En intégrant l'API de Deliveroo, l'export et la synchronisation des menus deviennent automatiques, ce qui réduit le besoin de mises à jour manuelles et garantit une précision constante des informations. 2. Efficacité et évolutivité : L'utilisation de Node.js et Express pour l'intégration côté serveur permet de gérer efficacement les demandes API et de faciliter la communication avec Deliveroo. Cela offre une solution performante et facilement adaptable pour la gestion des menus à grande échelle.
Partage sur Instagram	- L'encart "Partager sur Instagram" doit s'afficher dans la catégorie "Exportez et diffusez" - Au clic sur l'encart, des images du menu au format carré (pour les mettre sur Instagram) sont générées - Le restaurateur est redirigé vers son compte Instagram avec les photos carrées des menus	API d'Instagram html2canvas.	Pour cette fonctionnalité, nous intégrons l'API d'Instagram pour que le restaurateur puisse se connecter à son compte. Une fois connecté, l'application génère des images carrées du menu (plats, descriptions, prix) à l'aide de html2canvas. Ces images sont ensuite prêtes pour le partage sur Instagram via l'API, ouvrant une fenêtre de partage pour publier directement.	1. Facilitation du marketing : En permettant le partage direct sur Instagram, cette fonctionnalité aide le restaurateur à promouvoir facilement ses plats, en augmentant la visibilité de son menu et en touchant une audience plus large. 2. Gain de temps et efficacité : La génération automatique d'images du menu en format Instagram réduit le besoin de créations manuelles, simplifiant ainsi le processus de partage et rendant le marketing plus rapide et cohérent.

Besoin	Contraintes	Solution	Description de la solution	Justification(2 Arguments)
Déconnexion	Le restaurateur doit pouvoir se déconnecter depuis n'importe quelle page connectée.	Passport.js MongoDB Express React Router	Pour mettre en place cette fonctionnalité, nous utiliserons Passport,js pour gérer l'authentification des utilisateurs avec MongoDB. Lorsqu'un utilisateur clique sur le bouton de déconnexion, une requête sera envoyée à une route API dans Express pour terminer sa session active. Après la déconnexion, l'utilisateur sera redirigé vers la page d'accueil de l'application, en utilisant React Router pour gérer la navigation.	1. Sécurité renforcée: L'utilisation de Passport.js pour gérer l'authentification assure une gestion sécurisée des sessions utilisateurs, minimisant les risques d'accès non autorisé et protégeant les données personnelles. 2. Expérience utilisateur fluide: La redirection automatique vers la page d'accueil après la déconnexion, gérée par React Router, offre une navigation intuitive et cohérente, améliorant ainsi l'expérience globale de l'utilisateur au sein de l'application.
Infos utilisateur	En tant que restaurateur, je veux pouvoir modifier mes informations utilisateur. - Lier plusieurs adresses e-mail à son compte. - Modifier son adresse e-mail de base	MongoDB Express Bcrypt	Pour cette fonctionnalité, nous utiliserons MongoDB pour stocker les informations des utilisateurs et créerons une API avec Express.js pour gérer les interactions avec la base de données. - Gestion des adresses e-mail : MongoDB permettra aux utilisateurs d'ajouter ou de supprimer plusieurs adresses e-mail via leur tableau de bord, avec des mises à jour effectuées par l'API. - Mise à jour de l'adresse e-mail principale : Les utilisateurs pourront modifier leur adresse e-mail principale, et les changements seront enregistrés dans MongoDB via l'API. - Sécurité des données : Nous utiliserons bcrypt pour hacher les mots de passe, garantissant que l'authentification et les modifications des informations de compte se font de manière sécurisée.	1. Flexibilité et personnalisation: La possibilité de gérer plusieurs adresses e-mail et de modifier l'adresse principale offre aux utilisateurs une personnalisation de leur compte, répondant mieux à leurs besoins et leur permettant de gérer leurs informations de manière efficace.. 2. Sécurité renforcée : L'utilisation de bcrypt pour le hachage des mots de passe et l'authentification via l'API garantissent une protection solide des données des utilisateurs, réduisant les risques de violations de sécurité et assurant une confiance accrue dans la gestion des informations personnelles.

Besoin	Contraintes	Solution	Description de la solution	Justification(2 Arguments)
Dashboard	En tant que restaurateur, je veux pouvoir avoir accès à un dashboard qui regroupe : - La création de menu - La diffusion de menu - L'impression de menu - La section “Pour aller plus loin” et aux 3 derniers articles de blog qui parlent de Menu Maker	React MongoDB Express API de Deliveroo et Instagram React PDF Bcrypt jsonwebtoken	Mise en place du tableau de bord avec MongoDB et React: Le restaurateur sera redirigé vers un tableau de bord personnalisé à la connexion, comprenant des encarts pour créer, diffuser et imprimer un menu, ainsi qu'une section pour les derniers articles de blog. La création de menus sera gérée via une API Express.js, avec MongoDB pour l'enregistrement. L'encart de diffusion permettra de partager sur Deliveroo et Instagram via leurs API, et les informations peuvent être stockées dans MongoDB. Pour imprimer, un PDF sera généré avec React-PDFà partir des données du menu. L'accès sera sécurisé par authentification, utilisant bcrypt pour le hachage des mots de passe et jsonwebtoken pour les sessions, avec des routes d'authentification dans l'API.	1. Centralisation des fonctionnalités: Le tableau de bord regroupe toutes les fonctions essentielles pour le restaurateur, telles que la création, la diffusion et l'impression des menus, ce qui facilite la gestion de leur activité et améliore l'efficacité opérationnelle. 2. Sécurité et protection des données : L'utilisation de méthodes d'authentification robustes, comme bcrypt et jsonwebtoken, garantit la sécurité des informations sensibles des utilisateurs, renforçant la confiance des restaurateurs dans l'utilisation de la plateforme.
Branding restaurateur	Le restaurateur doit pouvoir ajouter / modifier / supprimer les éléments suivants : - Logo - Couleurs de base	React react-dropzone MongoDB Amazon S3	Pour cette fonctionnalité, nous utiliserons React pour le front-end. Le restaurateur pourra télécharger son logo via React-dropzone, qui sera ensuite stocké sur un service comme Amazon S3, avec l'URL enregistrée dans MongoDB. Concernant les couleurs de base, le restaurateur choisira des couleurs à partir d'une palette prédéfinie, et les valeurs seront également sauvegardées dans MongoDB, associées à son compte.	1. Personnalisation de la marque : La possibilité pour le restaurateur de télécharger son logo et de choisir les couleurs de base permet de créer une identité visuelle unique pour son menu, renforçant ainsi sa marque et son attractivité auprès des clients. 2. Stockage sécurisé et efficace : En utilisant des services de stockage comme Amazon S3 et MongoDB pour enregistrer les logos et les couleurs, nous garantissons un accès rapide et sécurisé aux données, tout en simplifiant la gestion des informations liées à chaque compte restaurateur.

II. Liens avec le back-end

- **Quel langage pour le serveur ?**

  Nous utiliserons Node.js avec le framework Express pour le développement de notre serveur. Node.js permet d'exécuter du JavaScript côté serveur, offrant ainsi une approche rapide et efficace pour gérer les requêtes. Express est un framework léger qui facilite la création d'applications web en simplifiant la gestion des routes et des requêtes HTTP, ce qui rend le développement plus rapide et structuré.

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

  Oui, une API est nécessaire pour permettre la communication entre différentes applications et services. Dans ce cas, nous allons créer une API REST avec Node.js et Express pour gérer les interactions spécifiques à notre application. Cette API facilitera des fonctionnalités telles que la gestion des menus, l'authentification des utilisateurs et le stockage des données, pour faciliter le travail des développeurs, nous utiliserons Swagger pour documenter et tester l'API.

  Nous intégrerons deux API existantes : l'API Deliveroo pour synchroniser les menus avec la plateforme de livraison et l'API Instagram pour permettre aux restaurateurs de partager facilement leurs menus et promotions. Cette approche améliore l'expérience utilisateur et optimise la gestion des informations pour les restaurateurs.

- **Base de données choisie**

  Nous utiliserons MongoDB qui est une base de données NoSQL qui stocke les données sous forme de documents JSON, ce qui la rend flexible et facile à utiliser. Elle peut gérer de grandes quantités de données et s'adapte facilement aux besoins changeants des applications. En plus, son développement est rapide, ce qui est idéal pour les projets en évolution.

III. Préconisations concernant le domaine et l’hébergement

- **Nom de domaine**

  Nous utiliserons pour le projet Menu-Maker.fr comme nom de domaine.

- **Nom de l’hébergement**

  Nous utiliseront Amazon Web Services (AWS) https://aws.amazon.com/fr/websites/?nc2=h_ql_sol_use_web

  AWS offre une infrastructure évolutive et flexible, permettant de s'adapter facilement à la croissance des applications. Avec une large gamme de services (comme EC2 Pour héberger votre application web, S3 Pour stocker des images de menus et d'autres fichiers statiques.), AWS assure une gestion efficace des données, une haute disponibilité et une sécurité renforcée. Son modèle de tarification à la demande permet de réduire les coûts, ce qui en fait un choix idéal pour les projets de toute taille.

- **Adresse e-mail**

  L'adresse e-mail sera déterminée par le nom de domaine ou le sous-domaine sélectionné. Si nous optons pour le nom de domaine mentionné précédemment, l'adresse e-mail sera par exemple : contact@menu-maker.fr.

IV. Accessibilité

● Compatibilité navigateur

Le site doit pouvoir être accessible sur les dernières versions de Chrome, Safari et Mozilla.

● Type d’appareils

Le site doit être compatible uniquement sur ordinateur en premier lieu. Pas de version mobile prévue pour le moment.

V. Recommandation en termes de sécurité

- Utiliser HTTPS : Sécurisez toutes les communications entre le client et le serveur.
- Valider les entrées utilisateur : Vérifiez et assainissez toutes les données envoyées par les utilisateurs pour éviter les attaques.
- Hachage des mots de passe : Utilisez bcrypt pour stocker les mots de passe en toute sécurité.
- Utiliser JWT : Protégez les routes sensibles avec des JSON Web Tokens pour l'authentification.
- Configurer CORS : Limitez les domaines autorisés à accéder à votre API.
- Limiter les tentatives de connexion : Empêchez les attaques par force brute en restreignant les connexions.
- Mettre à jour les bibliothèques : Assurez-vous que toutes les bibliothèques sont à jour pour éviter les vulnérabilités.
- Sécuriser les sessions : Utilisez des cookies sécurisés pour protéger les informations de session.
- Faire des sauvegardes : Réalisez des sauvegardes régulières de votre base de données.
- Configurer les permissions sur AWS : Restreignez l'accès aux ressources AWS selon les besoins.
- Audits de sécurité réguliers : Effectuez des vérifications de sécurité périodiques pour identifier les vulnérabilités.

Webgencia

VI. Maintenance du site et future mises à jour

- **Surveillance des performances :**

  Utilisez des outils comme AWS CloudWatch pour surveiller la performance de votre application et détecter les problèmes en temps réel.
- **Mises à jour régulières :**

  Gardez Node.js, Express, MongoDB, et toutes les bibliothèques (comme bcrypt, jsonwebtoken, etc.) à jour pour bénéficier des dernières fonctionnalités et correctifs de sécurité.
- **Gestion des dépendances :**

  Utilisez des outils comme npm audit pour vérifier et gérer les vulnérabilités des dépendances.
- **Planification des sauvegardes :**

  Mettez en place des sauvegardes automatiques régulières de votre base de données MongoDB pour éviter la perte de données.
- **Documentation continue :**

  Maintenez à jour la documentation de votre code et des API pour faciliter la compréhension et les futures modifications.
- **Évaluation de la sécurité :**

  Effectuez des évaluations de sécurité régulières pour identifier et corriger les vulnérabilités.
- **Optimisation de la base de données :**

  Surveillez et optimisez les performances de votre base de données MongoDB en ajustant les index et les requêtes.
- **Gestion des configurations :**

  Utilisez des fichiers de configuration et des variables d'environnement pour gérer les paramètres de l'application lors des mises à jour.
- **Feedback des utilisateurs :**

  Collectez régulièrement des retours d'utilisateurs pour identifier les domaines d'amélioration et les nouvelles fonctionnalités à développer.