

Spécifications techniques

[Nom du projet + nom du client]

Version	Auteur	Date	Approbation
1.0	Madane AROUL	28/01/2026	QWENTA

I. Choix technologiques	2
II. Liens avec le back-end.....	3
III. Préconisations concernant le domaine et l'hébergement	3
IV. Accessibilité	3
V. Recommandations en termes de sécurité	3
VI. Maintenance du site et futures mises à jour	4

I. Choix technologiques

- État des lieux des besoins fonctionnels et de leurs solutions techniques :

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
Gestion d'une catégorie de menu Modification d'une catégorie de menu Suppression d'une catégorie (CRUD)	L'ajout d'une catégorie doit pouvoir se faire directement sur l'écran de création de menu depuis une modale. Le nom de la catégorie doit être modifiable facilement Une confirmation est nécessaire avant suppression	react-modal Formik react-modal	Cette librairie React permet de créer simplement des modales performantes, accessibles avec un minimum de code. Formik est une librairie React qui facilite la gestion et la validation des formulaires. Cette librairie permet d'afficher une modale de confirmation avant la suppression d'une catégorie.	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. 3) Cela réduit le code nécessaire pour gérer les formulaires 4) Cela est très utilisée dans les projets React 5) Cela limite les erreurs utilisateur 6) Cela s'intègre facilement dans une application existante

Gestion d'un menu Modification d'un menu Suppression d'un menu (CRUD)	Le restaurateur doit pouvoir créer un menu facilement depuis une interface intuitive. Les modifications doivent être visibles immédiatement sans recharge de page. Une confirmation est obligatoire avant toute suppression définitive	React, React Hooks (useState, useEffect) react-modal	React est une librairie JavaScript permettant de créer des interfaces utilisateur dynamiques et réactives avec une gestion efficace des composants. Les hooks React permettent de gérer l'état et les mises à jour de l'interface de manière simple et performante. Cette librairie React permet de créer facilement des modales accessibles et performantes avec un minimum de configuration.	1)Le projet est une application interactive nécessitant des mises à jour en temps réel 2) React est largement utilisé et documenté 3)Les hooks sont intégrés nativement à React 4) Ils permettent une meilleure lisibilité et maintenance du code 5)La librairie est cohérente avec une application React 6) Elle est largement utilisée dans l'écosystème React
--	--	--	--	---

Ajout d'un plat Modification d'un plat Suppression d'un plat (CRUD)	Chaque plat doit contenir un nom, un prix et une description Les informations doivent être modifiables rapidement Une confirmation doit être demandée	Formik + Yup react-modal	Formik gère les formulaires tandis que Yup permet de valider les champs avant soumission Formik permet de préremplir les formulaires pour modifier les données existantes efficacement React Modal Cette librairie permet d'afficher une modale de confirmation avant suppression d'un élément	1) La validation garantit la cohérence des données 2) Ces librairies sont fréquemment utilisées ensemble 3) Elle améliore l'expérience utilisateur 4) Elle simplifie la gestion des formulaires complexes 5) Les actions irréversibles doivent être sécurisées 6) react-modal est une solution fiable et éprouvée
Affichage des menus	Le contenu doit s'adapter à l'écran d'un PC	CSS Flexbox & Media Queries	Flexbox et les media queries permettent de créer des interfaces responsives adaptées	1) Les menus sont majoritairement consultés sur mobile 2) technologies standard
Frontend communique avec Backend (échange de données avec le serveur)	Les données doivent être récupérées et envoyées efficacement	Axios	Axios est une librairie JavaScript permettant de réaliser des requêtes HTTP simplement	1) offre une meilleure syntaxe 2) Très utilisée dans les projets React

Authentification des restaurateurs (sécurisation de l'accès à l'application)	Chaque restaurateur doit accéder uniquement à ses menus	Auth API REST jwt	Le Json Web Token est un standard permettant de sécuriser l'authentification via des tokens	1) utilisé pour sécuriser les API 2) permet une gestion simple des sessions
Stockage et persistance des données	Les données doivent être persistantes, structurées et sécurisées.	MongoDB	MongoDB est une base de données NoSQL orientée documents, adaptée aux applications web modernes et flexibles	1) La structure des menus (menus → catégories → plats) se prête bien à un modèle NoSQL 2) MongoDB est largement utilisée avec des applications JavaScript
Exportation du menu en format PDF	Le menu exporté doit respecter la mise en page définie par le restaurateur et être téléchargeable en un clic.	jsPDF	La librairie jsPDF permet de générer dynamiquement des fichiers PDF directement depuis une application JavaScript. Elle permet de construire un document PDF à partir des données du menu et de l'interface existante	1) Librairie largement utilisée pour la génération de PDF côté client dans des applications web 2) Permet une génération instantanée sans dépendre du serveur

Communication avec des API externes (Instagram & Deliveroo)	Les données doivent être récupérées via des API externes sécurisées et respecter les limitations imposées par chaque service.	Axios	Axios est une librairie JavaScript permettant d'effectuer des requêtes HTTP vers des API externes. Elle facilite la récupération et l'envoi de données depuis des services tiers comme Instagram ou Deliveroo	1) Offre une syntaxe claire et lisible pour gérer les appels aux API externes 2) utilisée et compatible avec les applications React et les API REST
---	---	-------	---	--

II. Liens avec le back-end

- Quel langage pour le serveur ? le langage choisi pour le serveur est javascript
- A-t-on besoin d'une API ? Si oui laquelle ? : Oui, le projet Menu Maker nécessite une API afin de gérer la persistance des données, la communication entre le front-end et la base de données (MONGO DB) , ainsi que la sécurisation des accès.
- L'API sera une API REST développée avec Node.js et Express.JS(pour créer , modifier, supprimer, et mettre à jour)
- Elle permettra de créer, modifier, supprimer et récupérer les menus, catégories et plats, et sera afficher par le

front-end React via des requêtes HTTP effectuées avec Axios

- **La base de données choisie : NoSQL – MongoDB**

permet de stocker des données structurées et imbriquées de manière flexible, ce qui correspond parfaitement à la structure des menus, catégories et plats. Son intégration avec Node.js et Express facilite le développement d'une API REST moderne et évolutive

- Dans le cadre du projet Menu Maker, une base de données NoSQL, et plus précisément MongoDB, est retenue pour le stockage des données.
- MongoDB permet de stocker les informations sous forme de documents JSON, ce qui est particulièrement adapté aux données manipulées par l'application, comme les menus, les catégories, les plats ou les utilisateurs. Ces données peuvent avoir des structures flexibles et évoluer facilement dans le temps.

- **Raisons du choix de MongoDB :**

- **Flexibilité du schéma :**

Les menus et catégories peuvent évoluer (ajout de champs, nouvelles options) sans nécessiter de modifications lourdes de la base de données.

-

- **Facilité d'intégration avec JavaScript :**

MongoDB fonctionne naturellement avec des objets JSON, ce qui s'intègre très bien avec un back-end en Node.js / Express et un front-end en React.

- **Rapidité de développement :**

MongoDB permet de mettre en place rapidement les opérations CRUD (création, lecture, modification, suppression), essentielles pour un outil de gestion de menus.

- **Scalabilité :**

La base de données peut facilement évoluer si le nombre de restaurateurs et de menus augmente.

- **Utilisation dans le projet**

- La base de données MongoDB sera utilisée pour stocker :
- Les comptes utilisateurs (restaurateurs)
- Les menus créés
- Les catégories de menus
- Les plats associés à chaque catégorie
- L'accès à la base de données sera géré par le back-end Express via une API REST sécurisée, assurant la communication entre le front-end et la base de données.

III. Préconisations concernant le domaine et l'hébergement

- **Nom du domaine**

Le nom de domaine doit être clair, simple et représentatif du service proposé. Il doit permettre aux utilisateurs d'identifier facilement la plateforme Menu Maker et son objectif.

Il est recommandé de :

- Choisir un nom court et facile à mémoriser
- Éviter les caractères spéciaux et les tirets multiples
- Utiliser une extension classique et professionnelle comme .fr ou .com, selon la cible du projet

menuMaker.qwenta.com

Le nom de domaine doit être enregistré auprès d'un registrar reconnu afin de garantir sa disponibilité, sa sécurité et sa gestion à long terme.

Nom de l'hébergement

L'hébergement du site doit permettre de répondre aux besoins techniques du projet, notamment l'utilisation d'un front-end en React et d'un back-end basé sur Express.js.

Il est recommandé d'utiliser un hébergeur fiable offrant :

- Une bonne disponibilité (uptime élevé)
- Des performances adaptées à une application web moderne
- Une gestion simplifiée des déploiements

Exemples d'hébergeurs adaptés :

- **Vercel** (pour le front-end React)
- **Heroku** (pour le back-end Express)

Ces solutions permettent une mise en production rapide et une bonne évolutivité du projet.

• Adresses e-mail

La mise en place d'adresses e-mail professionnelles est recommandée afin de renforcer la crédibilité du service et de faciliter la communication avec les utilisateurs.

Exemples d'adresses e-mail :

- contact@menuMaker.qwenta.fr
- support@menuMaker.qwenta.fr

Ces adresses peuvent être utilisées pour :

- Le support utilisateur
- La communication avec les restaurateurs
- Les notifications liées au service

Les adresses e-mail peuvent être hébergées directement par le fournisseur de domaine ou via des services dédiés tels que Google Workspace ou Microsoft 365, garantissant fiabilité et sécurité.

IV. Accessibilité

• Compatibilité navigateur

Le site Menu Maker est conçu pour fonctionner correctement sur l'ensemble des navigateurs web modernes les plus utilisés. L'objectif est de garantir une expérience utilisateur fluide et cohérente, quel que soit le navigateur employé par le restaurateur.

Les navigateurs pris en charge sont notamment :

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Safari
- Le développement repose sur des technologies web standardisées (HTML5, CSS3, JavaScript ES6+) ainsi que sur le framework React, largement supporté par les navigateurs récents. Des tests multi-navigateurs sont prévus afin de vérifier l'affichage, la navigation et le bon fonctionnement des fonctionnalités principales.

• Types d'appareils

L'application Menu Maker est développée pour l'instant sur Ordinateurs de bureau (desktop)

V. Recommandations en termes de sécurité

- Accès aux comptes, API et données
- La sécurité de l'application est un point essentiel afin de protéger les données des utilisateurs et d'assurer le bon fonctionnement du service.
- Les principales mesures de sécurité mises en place ou recommandées sont les suivantes :
- Mise en place d'un système d'authentification sécurisé pour l'accès aux comptes utilisateurs
- Protection des routes sensibles côté back-end grâce au framework Express.js
- Validation et contrôle des données envoyées par le front-end afin d'éviter les erreurs, injections ou comportements malveillants
- Séparation claire entre le front-end et le back-end via une API REST, limitant l'exposition directe des données
- Utilisation de variables d'environnement pour stocker les informations sensibles (clé secrète, accès à la base de données, configuration serveur)
- Sécurisation des échanges de données entre le client et le serveur à l'aide du protocole HTTPS en environnement de production

Ces bonnes pratiques permettent de renforcer la sécurité globale de l'application et de réduire les risques liés aux attaques courantes sur le web.

VI. Maintenance du site et futures mises à jour

• Grandes lignes du contrat de maintenance

La maintenance du site Menu Maker est indispensable pour garantir :
sa stabilité, sa sécurité et son évolution dans le temps.

Elle comprend plusieurs axes principaux :

Correction des bugs détectés après la mise en production

- Mise à jour régulière des dépendances et librairies utilisées (React, Express, outils tiers)
- Surveillance des failles de sécurité et application des correctifs nécessaires
- Vérification continue de la compatibilité avec les navigateurs et appareils récents
- Possibilité d'évolutions fonctionnelles selon les besoins futurs du client (ajout de nouvelles fonctionnalités, amélioration de l'interface, optimisation des performances, responsive smartphone et tablette)
- Un suivi régulier et une maintenance continue permettent d'assurer la pérennité du projet et d'offrir une solution fiable et évolutive aux utilisateurs.
- Sauvegarde du site et de base de données tous les jours une fois à 20h (possibilité d'ajout + de sauvegardes selon le nombre d'