



Spécifications techniques

| | |
|---------------|----------------------|
| Projet | Menu Maker by Qwenta |
|---------------|----------------------|

| Version | Auteur | Date | Approbation |
|----------------|---------------|-------------|--------------------|
| 1.0 | Webgencia | Mars 2023 | John, Qwenta |

Le but de ce document est de définir et justifier les spécifications techniques de [nom du projet].

I. Choix technologiques

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

| Besoin | Contraintes | Solution | Description de la solution | Justification (2 arguments) |
|---|---|-----------------|--|--|
| Possibilité de se connecter à son propre espace | Via l'ouverture d'une modale qui doit pouvoir être accessible à trois endroits différents de la landing page (boutons « se connecter » dans la barre de | React-modal | Bibliothèque React. React-modal permet d'afficher une fenêtre modale et de la personnaliser grâce à des propriétés CSS. | 1) Bibliothèque React : facile d'utilisation et cohérence avec le choix de développer en React |



| | | | | |
|------------------------------------|--|--|---|---|
| | navigation + bannière et boutons « je me lance »). | | | 2) Populaire 3) Personnalisation du style fenêtre |
| S'authentifier avec adresse e-mail | <p>L'authentification doit se faire avec l'adresse e-mail uniquement. Une fois adresse e-mail de l'utilisateur renseignée et validée, doit recevoir un e-mail dans sa boîte afin de :</p> <ul style="list-style-type: none"> - Se connecter pour la première fois avec informations à renseigner - Confirmer son adresse e-mail si compte déjà existant <p>Avant que le serveur n'envoie un e-mail à l'adresse e-mail renseignée, il faut vérifier que celle-ci soit valide.</p> | <p>JSONWebToken Bcrypt</p> <p>API de Cloudmersive</p> <p>Firebase Auth</p> | <p>JsonWebToken et Bcrypt sont deux outils permettant de gérer l'authentification et l'autorisation des utilisateurs dans les applications web sécurisés. Permet notamment de crypter les données de l'utilisateur dans la base de données. On peut utiliser Bcrypt pour le stockage sécurisé des e-mails.</p> <p>L'API de Cloudmersive permet de vérifier que l'adresse e-mail renseignée par l'utilisateur est valide avant d'exécuter la suite du script qui est d'envoyer un e-mail à cette adresse.</p> <p>Firebase AUTH est une API proposé par Firebase, une plateforme de développement d'applications mobiles et web appartenant à Google. Elle offre un service d'authentification et d'identification des utilisateurs Elle permet de créer une</p> | <p>JsonWebToken et Bcrypt :</p> <ol style="list-style-type: none"> 1) Outils populaires 2) Apportant sécurité à l'authentifications des utilisateurs <p>L'API de Cloudmersive :</p> <ol style="list-style-type: none"> 1) facile à utiliser grâce à une documentation complète 2) nombreuses fonctions <p>Firebase AUTH :</p> <ol style="list-style-type: none"> 1) Facilite la gestion de l'authentification des utilisateurs 2) Répond le mieux au besoin d'une authentification par e-mail |



| | | | | |
|--|---|-------------|--|---|
| | | | authentification uniquement avec l'adresse e-mail de l'utilisateur. Firebase Auth stock les informations d'identification et gère toutes les étapes du processus d'authentification. | |
| Savoir que l'authentification est en cours | Icone d'attente de connexion doit être animé | ReactSping | Bibliothèque React pour l'animation de composants | 1) Bibliothèque React : facile d'utilisation et cohérence avec le choix de développer en React 2) Animations fluides 3) Performances élevées |
| Rediriger l'utilisateur suite interaction avec le site | Au clic sur un élément, l'utilisateur peut être redirigé vers une nouvelle page Exemple, lorsque l'utilisateur a pu s'authentifier, le rediriger à son espace. | ReactRouter | Bibliothèque React qui permet de gérer la navigation dans le site. Redirige l'utilisateur sur une nouvelle route lorsque celui interagit avec des éléments sur site. | 1) Bibliothèque React : facile d'utilisation et cohérence avec le choix de développer en React 2) Performante 3) Flexible (transition entre les routes personnalisables, protection de routes avec autorisation...) 4) Possibilité de prendre en charge l'historique de navigation |



| | | | | |
|----------------------------------|---|---|--|---|
| Création d'une catégorie de menu | <p>L'ajout d'une catégorie doit pouvoir se faire directement sur l'écran de création de menu depuis une modale contenant un formulaire.</p> <p>La modale « vierge » s'ouvre au clic sur « +ajouter... ».</p> <p>La modale se ferme avec x et clic en dehors modale.</p> | <p>React modal</p> <p>React-Hook-Form</p> | <p>React-Hook-Form : bibliothèque de gestion de formulaires dans React (validation des champs de formulaires suivant règles personnalisées, gestions de erreurs, gestion des champs de formulaire de manière dynamique, ...)</p> | <p>React-Hook-Form :</p> <ol style="list-style-type: none"> 1) Bibliothèque React : facile d'utilisation et cohérence avec le choix de développer en React 2) Récente 3) Syntaxe simple et intuitive 4) Performante 5) Légère et flexible 6) Se concentre sur la collecte des données de formulaire |
| Création d'un plat du menu | <p>Idem que « cration d'une catégorie de menu ».</p> <p>Il faut pouvoir renseigner le nom du plat, le prix, une description et ajouter une à deux photos. Il est possible de prévisualiser la photo depuis le formulaire.</p> | <p>React modal</p> <p>React-Hook-Form</p> <p>Module Multer de Node.js</p> | <p>Module Multer :</p> <ul style="list-style-type: none"> - module pour le gestionnaire de packages Node.js qui permet de gérer les fichiers téléchargés dans une application Express. - traite les données de formulaire de manière multipart/form-data, en extrayant les fichiers téléchargés et en les stockant dans un répertoire spécifié. - Permet de définir des | <p>Module Multer :</p> <ol style="list-style-type: none"> 1) Cohérence avec utilisation de Node.js et Express 2) Module populaire 3) Facilite gestion des fichiers téléchargés 4) Facile à installer |



| | | | limites de taille, de types, de noms et autres | |
|--|---|--------------------------------|---|--|
| Modifier une catégorie ou un plat existant et ajouté | Au clic sur le bouton (catégorie) ou l'icône (plat) « modifier » d'une catégorie ou plat existant et ajouté, la modale de création d'une catégorie ou d'un plat s'ouvre avec les informations déjà renseignées sauvegardées. | React modal React-Hook-Form | | idem |
| Modification du style du menu via la typographie et la couleur dominante du menu | Une fois qu'au moins une catégorie et un plat a été renseigné, l'utilisateur doit pouvoir choisir une typographie entre trois différentes. Il peut également choisir une couleur qui va changer la couleur du titre « menu », des catégories et des lignes de séparation. | Google Font React Color | Google Font : bibliothèque en ligne de typographies par Google React Color : librairie React fournissant une gamme de composants de sélecteurs de couleurs (de base, prédéfinies, personnalisés...). | Google Font : propose une large gamme de typographies, simple d'utilisations et populaire React color: 1) Bibliothèque React: facile d'utilisation et cohérence avec le choix de développer en React 2) Répond au besoin 3) Permet aux utilisateurs de sélectionner couleurs pré définies de manière visuelle et interactive |
| Exportation du | Une fois le style choisit, | React-PDF | React PDF : bibliothèque | React PDF : |



| | | | | |
|----------------------------|--|----------------------------|--|--|
| menu finalisé en PDF | l'utilisateur à la possibilité de cliquer sur un bouton permettant d'exporter le menu visualisé en PDF qui se télécharge via son navigateur. On doit donc pouvoir exporter du HTML + CSS en PDF. | FileSaver | <p>JavaScript qui permet de générer des fichiers PDF. Les PDF sont stylisables avec CSS.</p> <p>FileSaver : bibliothèque JavaScript qui fournit une API pour enregistrer des fichiers côté client.</p> | <ol style="list-style-type: none"> 1) Bibliothèque React : facile d'utilisation et cohérence avec le choix de développer en React 2) Possibilité personnalisation 3) Performance 4) Mise à jour très régulière (correction bugs et ajouts nouvelles fonctionnalités) <p>FileSaver :</p> <ol style="list-style-type: none"> 1) Facilité d'utilisation 2) Très populaire, grosse communauté 3) Compatibilité avec beaucoup de navigateur 4) Pas besoin de base de données car se télécharge directement sur le navigateur de l'utilisateur |
| Depuis le Dashboard, avoir | Lorsque le bouton valider dans la création de menu est cliqué, | Node.js Base de données | Node.js : environnement d'exécution côté serveur pour les | Node.js : 1. Rapide (modèle |



| | | | | |
|--|--|--------------------------------------|--|--|
| accès aux menus déjà créés depuis le bouton de nav « mes menus » | le menu en l'état est sauvegardé et apparait dans la page « mes menus ». De là, ils sont modifiables et peuvent être supprimé. | SQL + Express pour créer API backend | <p>applications web. Construit sur le moteur JavaScript.</p> <p>SQL : langage informatique utilisé pour gérer des bases de données relationnelles. API backend pour interagir avec la base de données</p> <p>Express : framework open-source qui facilite le développement d'applications web côté serveur en Node.js. Peut gérer des tâches courantes telles que la gestion des cookies, la compression, la gestion des erreurs, la gestion des sessions, la validation des données, etc.</p> | <p>événementiel et asynchrone pour traiter plusieurs infos en même temps)</p> <ol style="list-style-type: none"> 2. Facile à apprendre (JS comme langage de programmation) 3. Populaire : grosse communauté <p>Express :</p> <ol style="list-style-type: none"> 1) Facilité d'utilisation 2) Populaire <p>SQL :</p> <ol style="list-style-type: none"> 1) Facilité d'utilisation : langage simple et intuitif 2) Populaire et bonne documentation 3) Flexibilité dans la manipulation de données. |
| Depuis le Dashboard, possibilité d'imprimer un menu | L'utilisateur doit pouvoir sélectionner le menu qu'il souhaite imprimer. Ce dernier est généré en PDF et ouvert dans un nouvel onglet. De là l'utilisateur peut l'imprimer | React-PDF | | idem |



| | | | | |
|--|------------------------|--|--|--|
| | depuis son navigateur. | | | |
|--|------------------------|--|--|--|

II. Liens avec le back-end

- Quel langage pour le serveur ?
 - Node.js est un système d'architecture événementielle non bloquante, qui permet aux applications de traiter rapidement les requêtes entrantes et les interactions utilisateurs en temps réel. En utilisant également Javascript, on peut à la fois écrire du code pour le côté client et le côté serveur ce qui simplifie la tâche puisqu'il n'est pas nécessaire d'apprendre un nouvel langage ainsi que faire appel à un développeur spécialisé backend.
 - En utilisant le framework Express.js, on peut mettre en places des routes d'API qui écoutent les requêtes HTTP entrantes et y répondent en conséquence. Ces routes peuvent être utilisées pour gérer les interactions utilisateur, telles que les clics sur des boutons, les soumissions de formulaires et les requêtes AJAX.
- A-t-on besoin d'une API ? Si oui laquelle ?
 - On peut avoir recours à l'API Cloudmersive qui permet de vérifier que l'adresse e-mail renseignée par l'utilisateur est valide.
 - On pourra également utiliser l'API de FileSaver pour enregistrer des fichiers sur le navigateur du client grâce à la méthode `saveAs()` qui est appelée pour télécharger des fichiers binaires tel qu'un fichier PDF
 - On va avoir besoin d'une API de back end afin que les utilisateurs puissent récupérer, modifier ou supprimer des données stockées en base de données grâce à des requêtes http.



- Un des objectifs futurs de cette application est de pouvoir partager le menu créé sur l'application Deliveroo. On devra donc faire appel à leur API de gestion des menus Deliveroo pour télécharger les données du menu créer directement sur la plateforme Deliveroo. Ainsi, on pourra ajouter, modifier et supprimer des plats, des catégories de plats, des photos, mettre à jour les prix...etc. La seule utilisation de l'API deliveroo suffit, il n'est pas nécessaire de créer notre propre API pour partager les données liées aux menus.
- Idem pour l'API de Instagram
- Base de données choisie :

Une base de données SQL est adaptée à notre application car on a besoin d'une structure de données permettant de mettre en lien et de créer une cohérence entre nos données.

On va avoir besoin de stocker des informations sur les catégories de plats, les plats, les images et les prix pour chaque utilisateur. De plus, il faudra qu'on stocke les informations de connexion de ces derniers.

Grâce au modèle relationnel de SQL, on pourra créer des bases de données en tables avec des lignes et des colonnes et en établissant des relations entre les tables quand nécessaire.

III. Préconisations concernant le domaine et l'hébergement

- Nom du domaine
- Nom de l'hébergement.
- Adresses e-mail.



IV. Accessibilité

- Compatibilité navigateur.
- Types d'appareils.

V. Services tiers

- Noms et justification de ceux qui devront être implémentés pour le bon fonctionnement et le suivi du site (par exemple les plugins).

VI. Recommandations en termes de sécurité

- Accès aux comptes, plugins...

VII. Maintenance du site et futures mises à jour

- Grandes lignes du contrat de maintenance.