

PRÉSENTATION

Menu Maker by Qwenta

Sommaire

1. Contexte du projet
2. Aperçu de la maquette
3. Méthodologie utilisée
4. Tableau Kanban
5. Spécifications techniques
6. Veille technologique
7. Conclusion
8. Questions

Contexte du projet

Menu Maker est une application web permettant aux restaurants de créer et gérer leurs menus de manière simple et interactive.

Objectifs principaux :

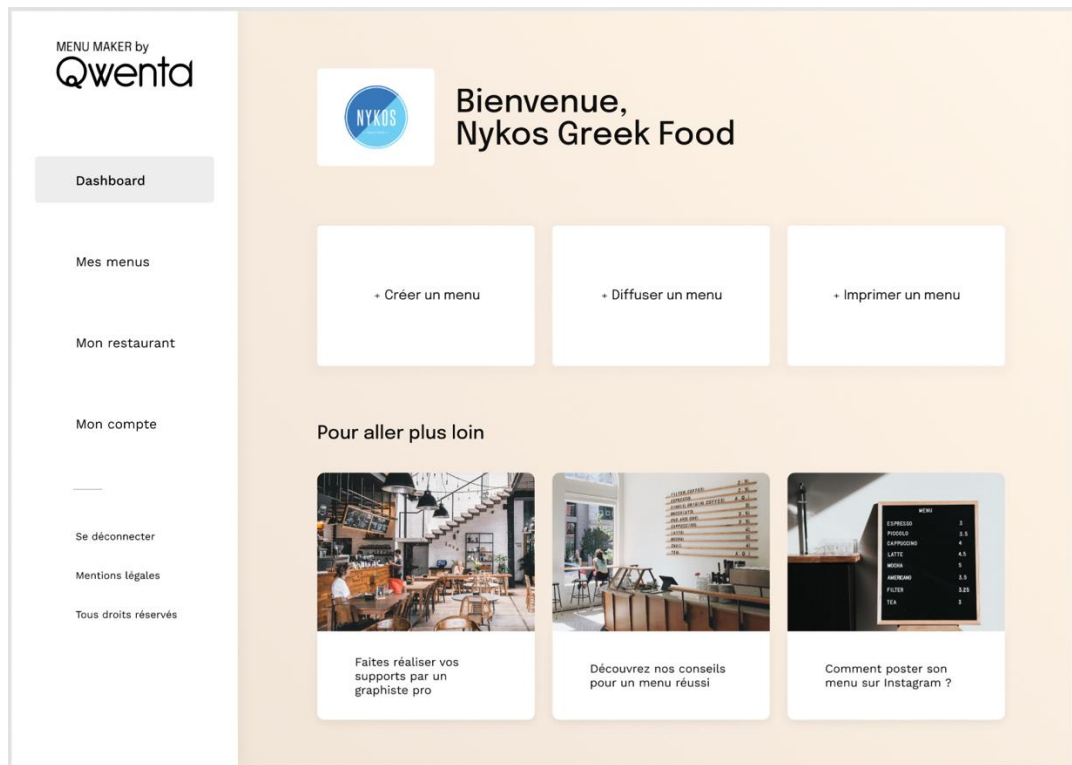
Faciliter la création et la modification des menus.

Structurer les plats par catégories.

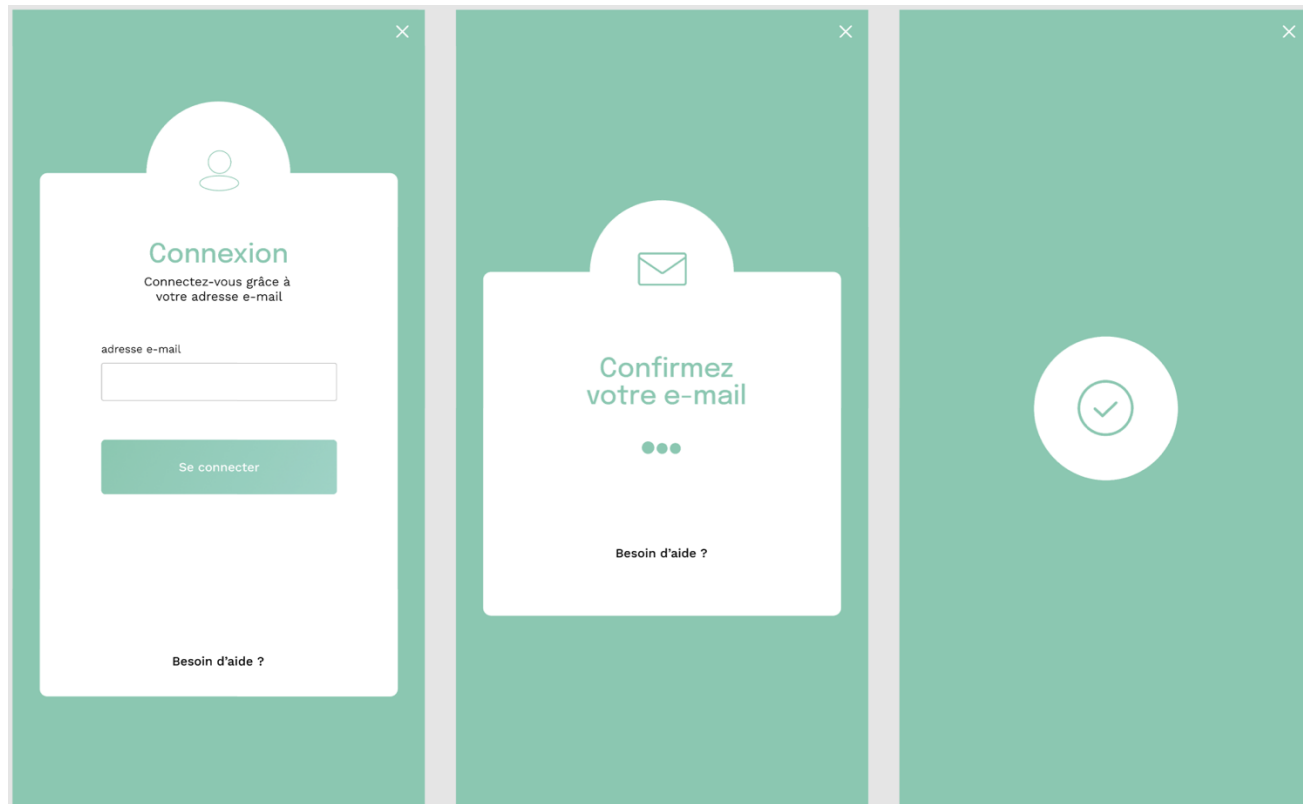
Proposer une interface intuitive pour les utilisateurs et les administrateurs.

Aperçu de la maquette

Dashboard



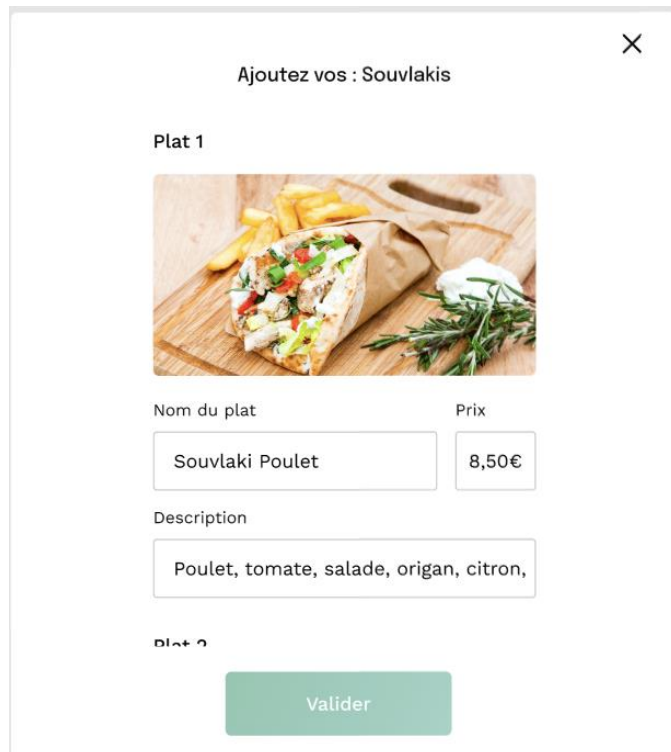
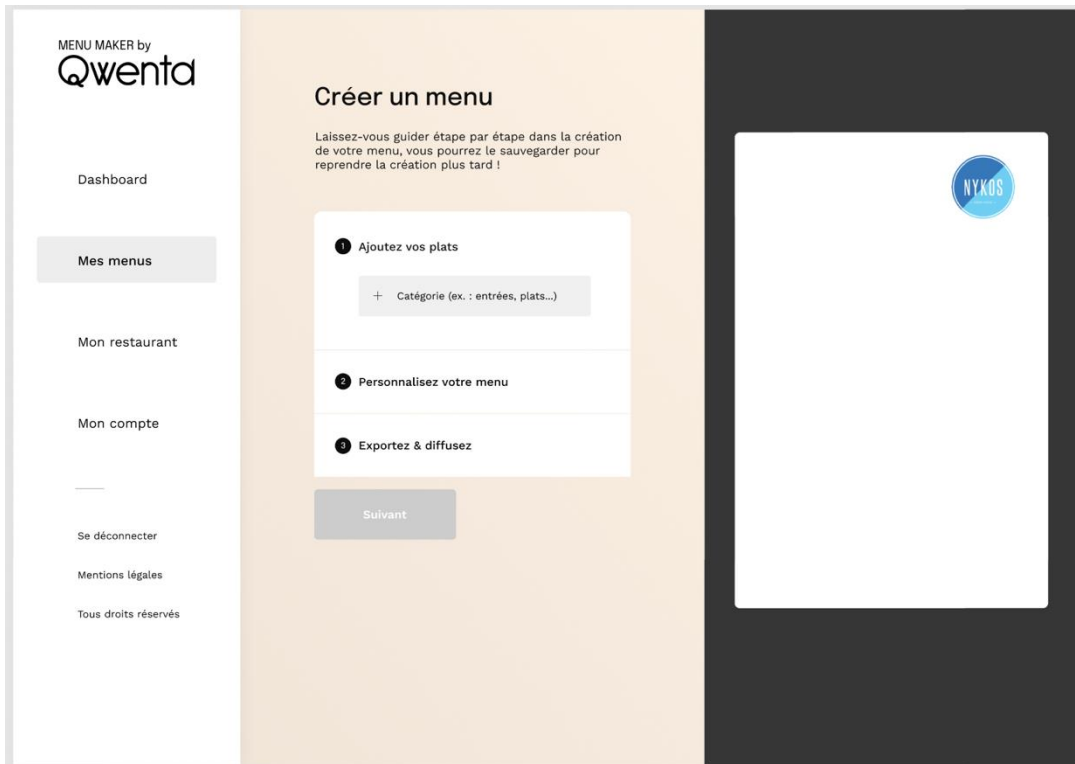
Aperçu de la maquette



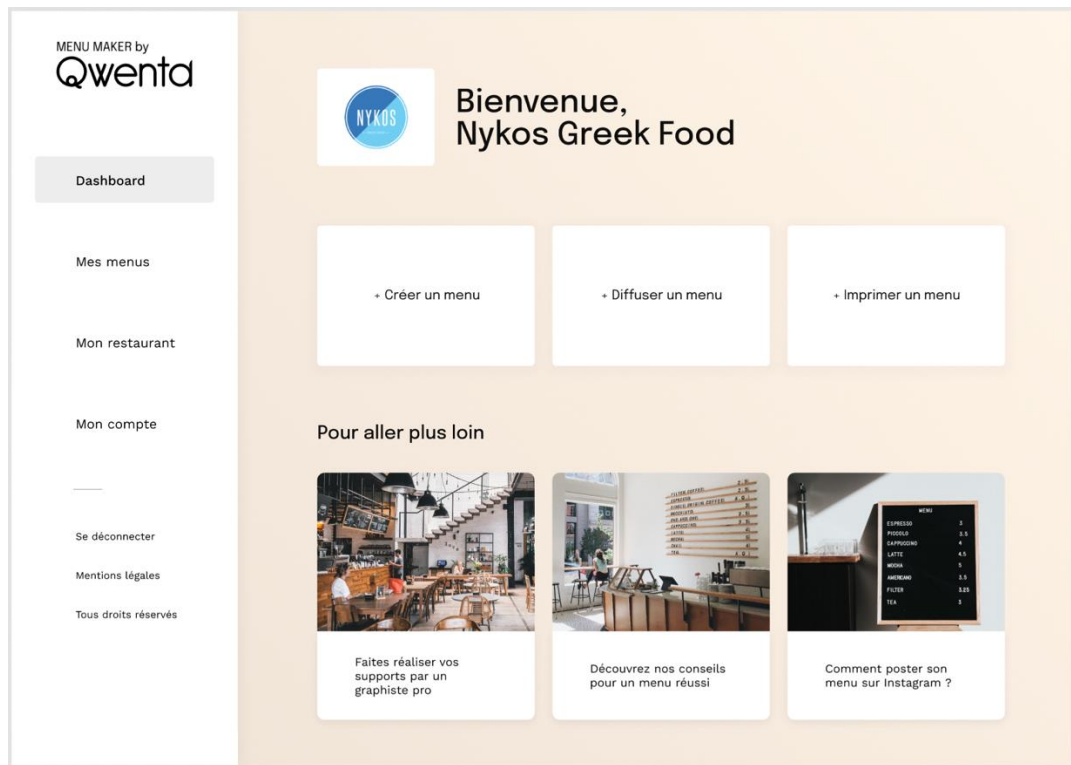
Gestions des utilisateurs

Aperçu de la maquette

Gestions des menus



Aperçu de la maquette



Aperçu de la maquette

Filtrage par catégories.

MENU MAKER by
Qwenta

Dashboard

Mes menus

Mon restaurant

Mon compte

Se déconnecter

Mentions légales

Tous droits réservés

Créer un menu

Laissez-vous guider étape par étape dans la création de votre menu, vous pourrez le sauvegarder pour reprendre la création plus tard !

1 Ajoutez vos plats

+ Catégorie (ex. : entrées, plats...)

2 Personnalisez votre menu

3 Exportez & diffusez

Suivant



Méthodologie utilisée

Agile et Scrum :

- **Agile** est une méthode de gestion de projet qui privilégie la flexibilité, l'adaptation rapide aux changements et la collaboration entre les membres de l'équipe.
- **Scrum** est un cadre de travail Agile qui organise le projet en **sprints**, c'est-à-dire des périodes courtes et répétitives (souvent 1 à 2 semaines) avec des objectifs précis.

Chaque sprint permet de **planifier, développer, tester et livrer** un ensemble de fonctionnalités.

Des réunions quotidiennes (stand-ups) et des revues de sprint permettent de **suivre l'avancement**, de résoudre les problèmes rapidement et d'ajuster le projet en continu.

Avantages pour le projet Menu Maker :

- Flexibilité face aux changements de besoins ou fonctionnalités.
- Priorisation claire des tâches et objectifs par sprint.
- Suivi régulier et coordination efficace de l'équipe.

Suivi du projet avec le Kanban

Kanban Menu Maker

Tableau kanban

+

Pas commencé 8

Landing non connectée

Page login

Catégorie de plat

Menus précédents

Partage sur Instagram

Branding restaurateur

Personnaliser un menu

Diffuser un menu

+ Nouvelle page

En cours 6

Style de menu

Création de plat

Exportation Deliveroo

Infos utilisateurs

Dashboard

Créer un menu

+ Nouvelle page

A tester 2

Imprimer un menu

Exportation PDF

+ Nouvelle page

Terminé 3

Informations légales

Tarifs

Déconnexion

+ Nouvelle page

+ Nouveau groupe

Nouveau

Suivi du projet avec le Kanban

Explication des User Stories (US) et tâches :

Les **User Stories (US)** représentent les besoins fonctionnels du point de vue de l'utilisateur. Chaque US est décomposée en **tâches concrètes** à réaliser par l'équipe.

Exemples :

US : Ajouter un plat → Tâches : création du formulaire, validation des champs, stockage en base, affichage dans la liste des plats.

US : Filtrer les menus par catégorie → Tâches : développement du filtre côté front, test de l'affichage dynamique.

US : Gestion des comptes utilisateurs → Tâches : création du système de connexion, assignation des rôles (admin/restauranteur), sécurisation JWT.

Organisation du tableau :

Le tableau Kanban est divisé en colonnes : **À faire / En cours / À tester / Terminé.**

Chaque tâche passe d'une colonne à l'autre au fur et à mesure de son avancement.

Suivi du projet avec le Kanban

Comment le tableau facilite le suivi et la coordination de l'équipe :

Visualisation rapide de l'avancement : Les membres de l'équipe savent immédiatement quelles tâches sont en cours, à tester ou terminées.

Priorisation claire : Les tâches les plus importantes peuvent être déplacées vers le haut de la colonne « À faire ».

Communication simplifiée : Les réunions quotidiennes peuvent se concentrer sur les tâches en cours ou bloquées.

Responsabilisation : Chaque tâche est attribuée à un membre, ce qui permet de suivre qui fait quoi.

Lien:https://www.notion.so/21d92e36f97f8086b189d187f82b9f3d?v=21d92e36f97f80c584ab000cd0e72dcb&source=copy_link

Spécifications techniques

Sauvegarde des menus (plats + catégories)	Stockage flexible avec relations simples	PostgreSQL	Base de données relationnelle adaptée aux relations entre menus, plats et catégories.	<ol style="list-style-type: none"> 1. Forte cohérence des données. 2. Large communauté et support cloud.
API pour communication front <-> back	Architecture découplée	Express.js + REST API	Création d'une API RESTful côté serveur avec Express.js pour gérer les routes.	<ol style="list-style-type: none"> 1. Framework Node simple à utiliser. 2. Compatible avec JWT et PostgreSQL.
Partage d'un menu via un lien	Lien public	<u>React Router</u>	Génération d'un lien unique affichable sans authentification.	<ol style="list-style-type: none"> 1. Simple à mettre en œuvre. 2. Permet la consultation sans modifier le menu.
Responsive design	Consultation sur smartphone et tablette	<u>Flexbox</u> + <u>Media Queries</u>	Mise en page adaptable aux écrans.	<ol style="list-style-type: none"> 1. Standards CSS modernes. 2. Compatible avec tous navigateurs.
Accessibilité	Menu consultable par tous, même avec handicap	WCAG + ARIA	Respect des bonnes pratiques (contrastes, navigation clavier, ARIA)	<ol style="list-style-type: none"> 1. Compatible lecteurs d'écran. 2. Meilleure note <u>Lighthouse</u>.
Sécurité	Protection des données utilisateur	Helmet.js + <u>Bcrypt</u> + CORS	<u>Helmet</u> sécurise les headers HTTP, <u>Bcrypt</u> hash les mots de passe, CORS limite les origines.	<ol style="list-style-type: none"> 1. Standards de sécurité Web. 2. Conformes RGPD.

Spécifications techniques

Explication et justification :

PostgreSQL a été choisi pour sa capacité à gérer efficacement des relations complexes entre tables et pour sa robustesse.

JWT + middleware sécurise l'accès aux API et permet de contrôler les rôles utilisateurs sans complexité côté front.

Node.js/Express offre un backend simple à maintenir, performant et modulable pour les API REST.

Le **front-end dynamique** et les filtres JavaScript améliorent l'expérience utilisateur en affichant instantanément les informations sans rechargement complet de la page.

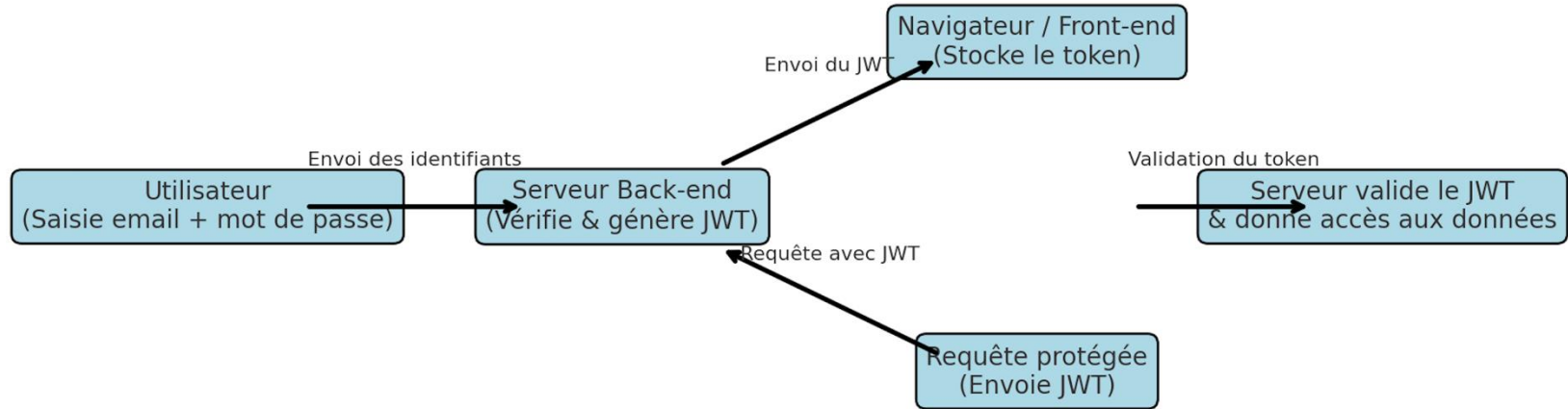
Spécifications techniques

Liste des principales spécifications techniques.

- Front-end en HTML, CSS, JavaScript
- Back-end en Node.js avec Express
- Base de données MongoDB
- API REST pour la communication entre front et back
- Authentification avec JWT (tokens)
- Déploiement sur Heroku
- Gestion du code avec GitHub

Spécification clé choisie : API REST (car c'est le cœur de la communication entre le site et la base de données).

Schéma simplifié de l'authentification avec JWT



Veille Technologique

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
Création d'un compte utilisateur	Connexion sécurisée, création de compte	JWT (JSON Web Token)	Utilisation d'un <u>token</u> généré côté serveur (Express.js) pour sécuriser les échanges et authentifier les utilisateurs.	1. Sécurité sans stocker les identifiants côté client. 2. Fonctionne parfaitement avec une SPA <u>React</u> .
Interface dynamique pour créer un menu	Interface fluide et modifiable sans rechargement	React.js	Framework <u>front-end</u> basé sur des composants réactifs, parfait pour créer des interfaces dynamiques.	1. Composants réutilisables pour les plats, les sections, etc. 2. Très bien documenté et rapide à implémenter.
Organisation et réorganisation des plats (drag & drop)	Glisser-déposer dans l'éditeur	<u>React Beautiful DnD</u>	Librairie <u>React</u> pour gérer simplement le drag & drop avec accessibilité intégrée.	1. Maintenue par <u>Atlassian</u> , utilisée dans Trello. 2. Intégration simple avec <u>React</u> .
Dashboard	Interface de gestion regroupant toutes les fonctionnalités (menus, compte, restaurant).	React.js + <u>React Router</u>	Le Dashboard est une Single Page Application (SPA) organisée en sections accessibles via le routage.	1. Centralise toutes les fonctionnalités pour l'utilisateur. 2. Simplifie la navigation et améliore l'expérience.

Veille Technologique

Méthode de classification des sources :

Les sources ont été classées selon deux axes principaux :

Technologies front-end (ex. frameworks JavaScript, CSS, UX/UI)

Technologies back-end et bases de données (ex. Node.js, Express, PostgreSQL, sécurité API)

Chaque source est évaluée selon sa fiabilité, sa pertinence pour le projet, et sa contribution à l'amélioration technique.

Veille Technologique

Exemples et justification des choix

Axe de veille	Source	Choix et justification
Front-end	Documentation officielle React	A permis de décider de la structure dynamique des composants pour afficher les menus et filtres en temps réel, facilitant la mise à jour sans rechargement complet.
Back-end / Base de données	Article sur PostgreSQL et relations complexes	A confirmé que PostgreSQL est la solution la plus adaptée pour gérer les liens entre Menu, Plat et Catégorie de manière sécurisée et fiable.
Sécurité API	Tutoriel JWT + Express	A guidé la mise en place d'un système d'authentification sécurisé, avec tokens stockés côté client dans un cookie HttpOnly et middleware de contrôle des rôles.
UX/UI	Blog sur la gestion des formulaires interactifs	A inspiré le design des formulaires d'ajout et modification de plats, avec validation et retour utilisateur immédiat.

Veille Technologique

Contribution de la veille à l'élaboration des spécifications techniques :

Les **choix technologiques** (PostgreSQL, Node.js, JWT) sont directement issus des informations récoltées pendant la veille.

La veille a permis d'identifier les bonnes pratiques de sécurité et de structuration des données.

Les solutions front-end dynamiques et les filtres interactifs ont été optimisées grâce aux informations trouvées sur les frameworks et tutoriels UX/UI.

Au final, la veille a servi à valider et améliorer les spécifications techniques, rendant le projet plus robuste, sécurisé et agréable à utiliser.

Conclusion

Le projet **Menu Maker** a été développé de manière structurée et efficace grâce à l'utilisation de la méthodologie Agile/Scrum, avec une organisation en sprints qui a permis un suivi régulier et une livraison progressive des fonctionnalités.

Les principaux points à retenir :

Technologies choisies : PostgreSQL pour la base de données, Node.js/Express pour le backend, HTML/CSS/JavaScript pour le front-end.

Sécurité : authentification via JWT, stockage sécurisé des tokens, contrôle des rôles et validation des données.

Organisation : le tableau Kanban a permis de visualiser l'avancement des tâches, de coordonner l'équipe et de prioriser les besoins efficacement.

Veille technologique : a contribué à orienter les choix techniques, optimiser l'expérience utilisateur et sécuriser l'application.

Ainsi, Menu Maker répond aux besoins fonctionnels et aux contraintes techniques du projet tout en offrant une solution sécurisée, dynamique et facile à maintenir.

QUESTIONS ?