

PRÉSENTATION

Menu Maker by Qwenta

Sommaire

- Contexte du projet
- Aperçu de la maquette
- Méthodologie utilisée
- Tableau Kanban
- Spécifications techniques
- Veille technologique
- Conclusion
- Questions

Contexte du projet

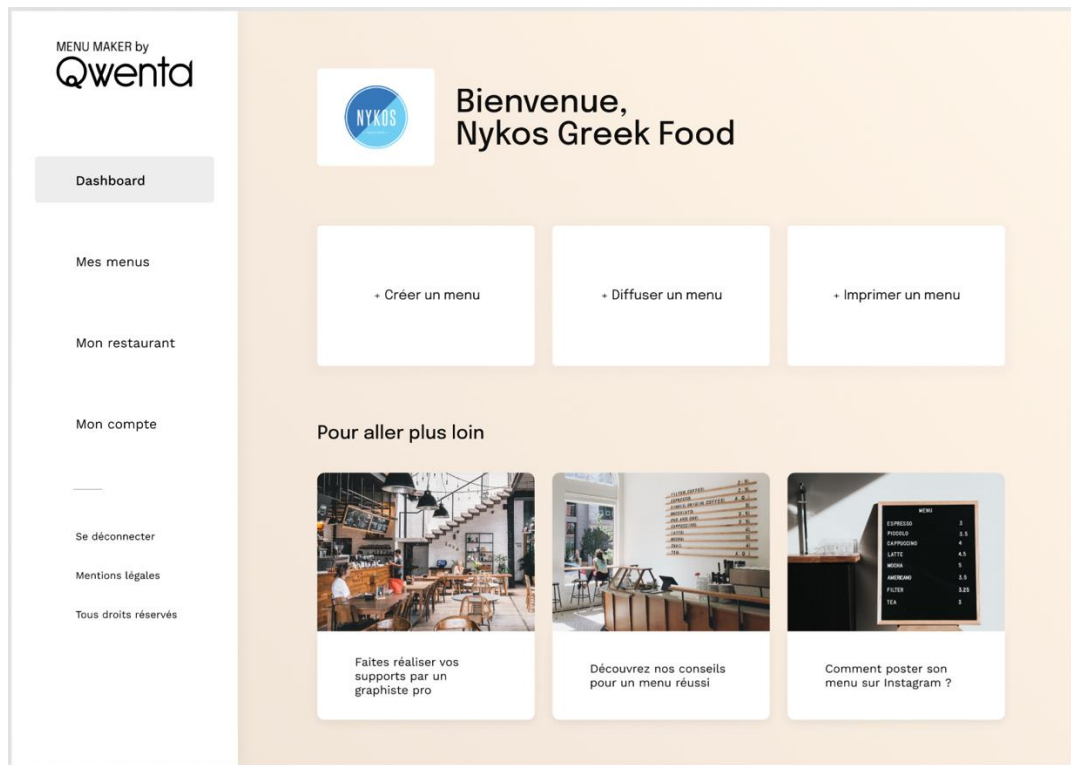
Application *Menu Maker* pour restaurateurs

Objectif : créer, personnaliser et diffuser des menus

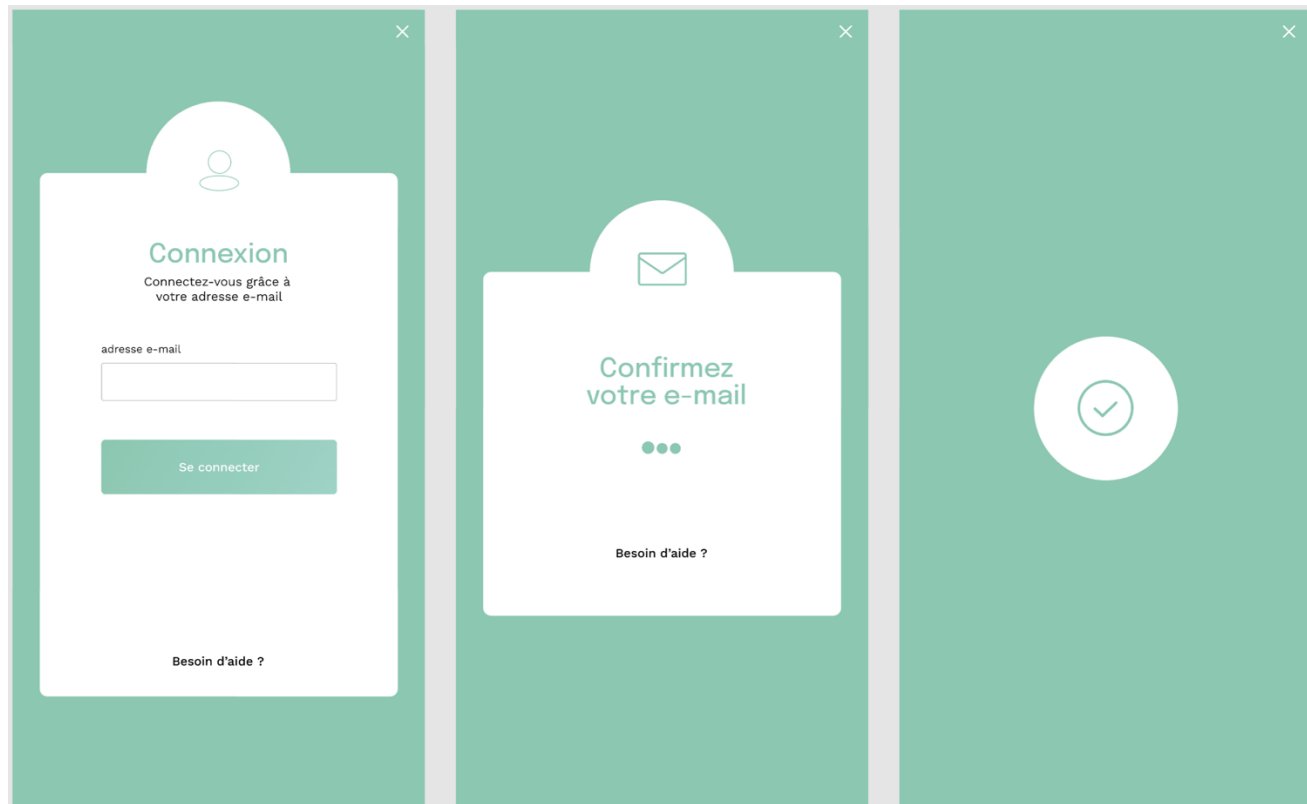
Public cible : restaurateurs indépendants

Aperçu de la maquette

Dashboard



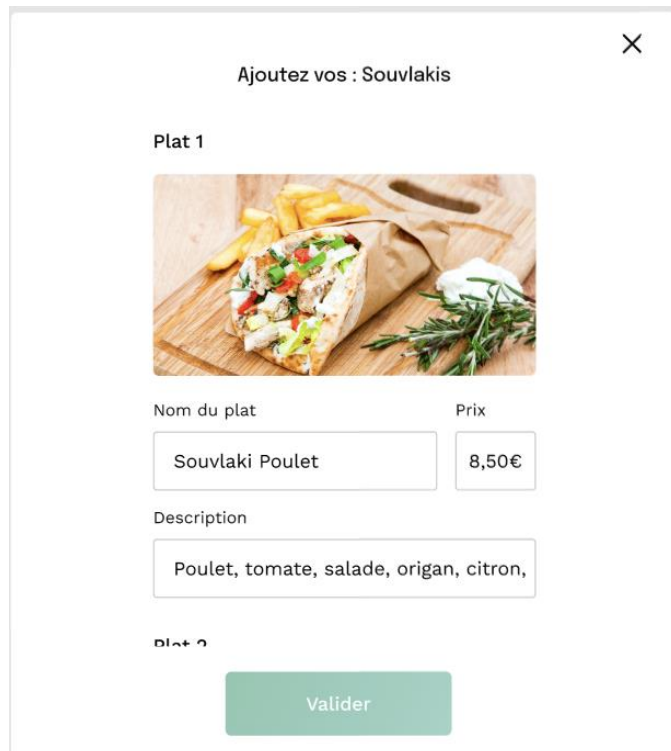
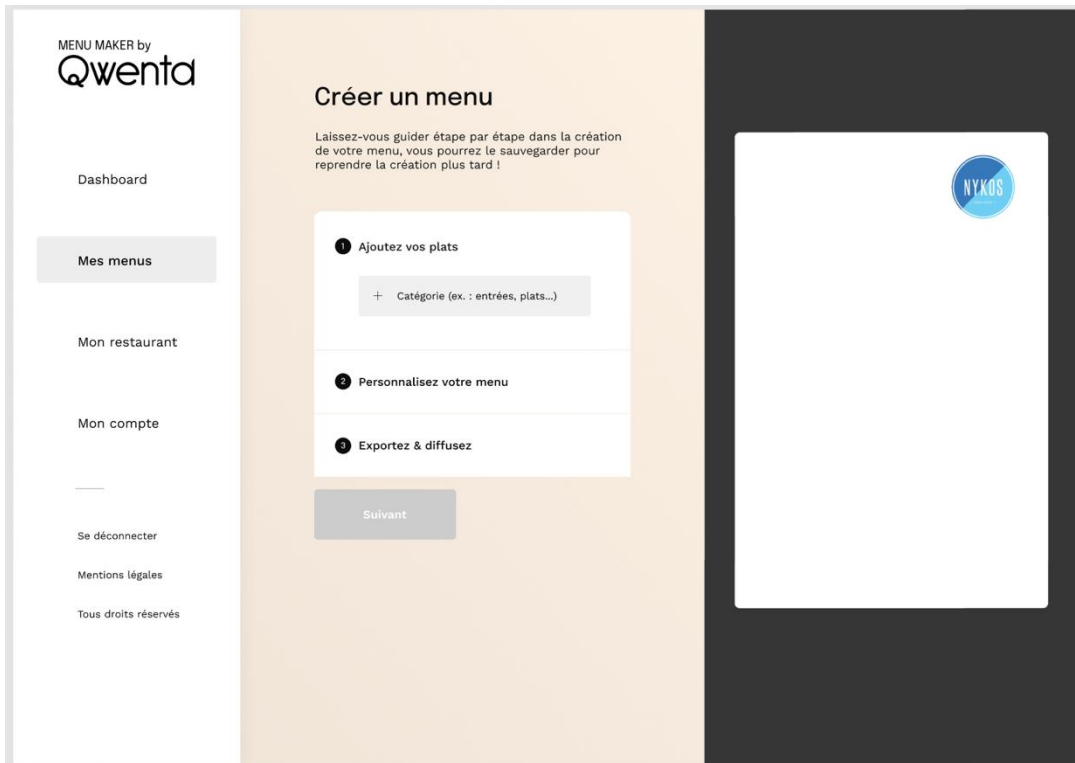
Aperçu de la maquette



Gestions des utilisateurs

Aperçu de la maquette

Gestions des menus



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 = flexibilité, collaboration
- Scrum = sprints courts (1–2 semaines)
- Suivi : stand-ups et revues
- Bénéfices : flexibilité, priorisation, suivi régulier

Suivi du projet avec le Kanban

Tableau Kaban (notion)

Kanban Menu Maker

Tableau kanban

+

☰

↑↓

⚡

🔍

↔

Nouveau

▼

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

Explication et avantages du tableau kanban

👉 Structure :

User Stories décomposées en tâches techniques

Exemple : *Ajouter un plat* → *créer formulaire*, *valider champs*, *stocker en base*, *afficher*

Progression : **À faire** → **En cours** → **À tester** → **Terminé**

👉 Avantages :

Visualisation claire de l'avancement

Priorisation simple des tâches

Réunions centrées sur les blocages

Attribution claire aux membres → responsabilisation

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

Spécifications techniques

Front-end : React, HTML/CSS/JavaScript (UI dynamique et responsive)

Back-end : Node.js + Express (API REST)

Base de données : PostgreSQL (relations complexes, robustesse)

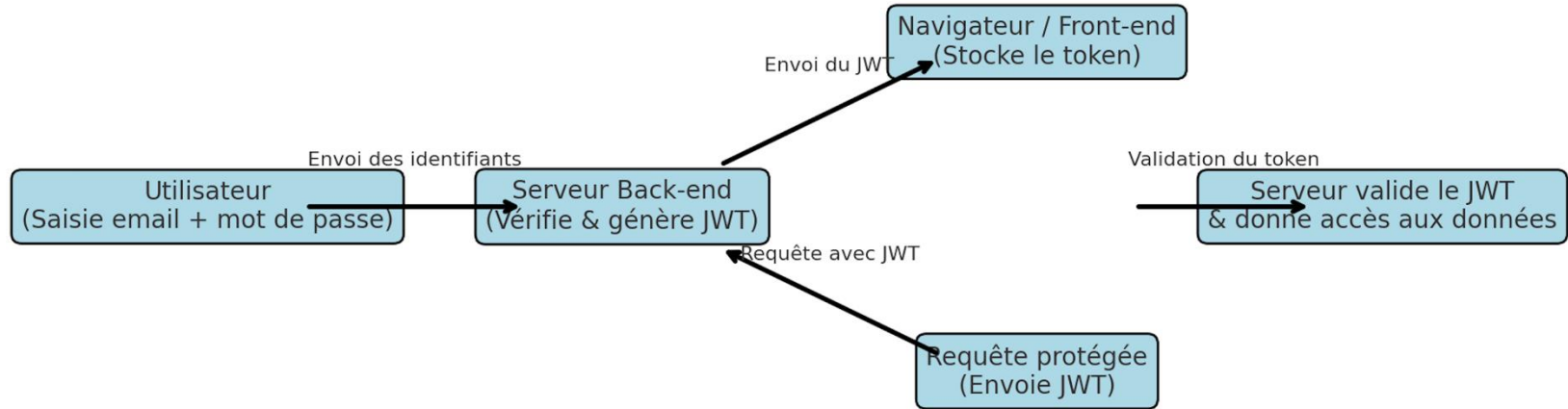
Authentication : JWT + Auth0 (sécurité et gestion des rôles)

Organisation : Kanban + Agile/Scrum (suivi et priorisation)

Focus : Authentication avec JWT

→ Comme un **badge d'accès** : une fois validé, il permet d'accéder aux ressources sans redonner son mot de passe à chaque action.

Schéma simplifié de l'authentification avec JWT



Veille Technologique avec Feedly

2 axes → Front-end / Back-end.

▽	Veille – Développement Web	33
	Articles on Smashing Magazine — For Web ...	11
	CSS-Tricks	16
	MDN Blog	1
	OWASP	2
	web.dev: Blog	3

▽	Veille – Technique	49
	Auth0 Blog	18
	Cloud Blog	22
	Node.js Blog	4
	Planète PostgreSQL	5
	React Blog – React	

Veille Technologique

Méthode de classification

Sources classées en 2 axes : front-end / back-end

Évaluation selon : fiabilité, pertinence, contribution technique

Exemple choix de sources

Front-end → React Blog : (réutilisation composants), Smashing Magazine (UI/UX).

Back-end → Node.js Blog, Planet PostgreSQL, Auth0 Blog : bonnes pratiques UI/UX

Contribution aux spécifications

PostgreSQL validé pour la robustesse

JWT retenu pour sécuriser l'authentification

Inspiration UX pour améliorer la navigation

Veille Technologique

Choix technos (PostgreSQL, Node.js, JWT) issus de la veille

Bonnes pratiques en sécurité & structuration des données

Optimisation UX/UI grâce aux sources front-end

Veille = validation + amélioration des spécifications

Conclusion

Agile/Scrum : suivi régulier, livraison progressive

Technos : PostgreSQL, Node.js/Express, Solution robuste

Sécurité : JWT, rôles, validation des données et sécurité

Organisation : Kanban = avancement clair et priorisation

Veille : a guidé les choix, optimisé l'UX et la sécurité

Amélioration future: connexion à des services d'impression professionnelle (ex : VistaPrint)

QUESTIONS ?