

# 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 by Qwenta

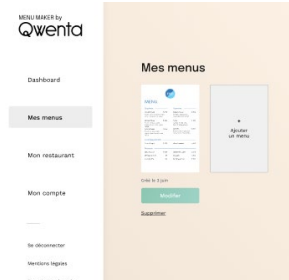
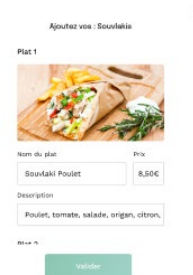
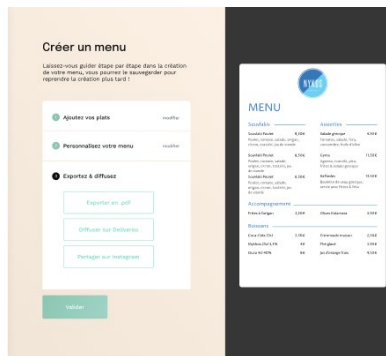
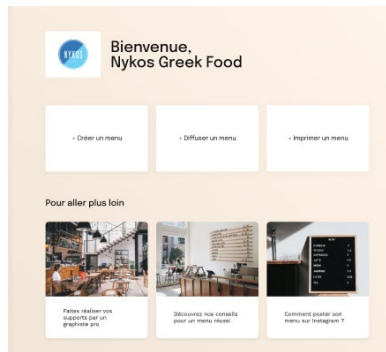
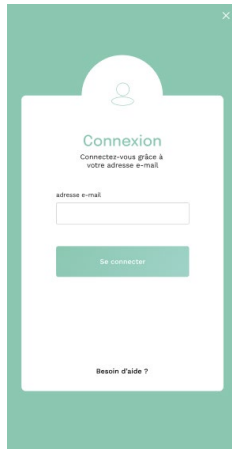
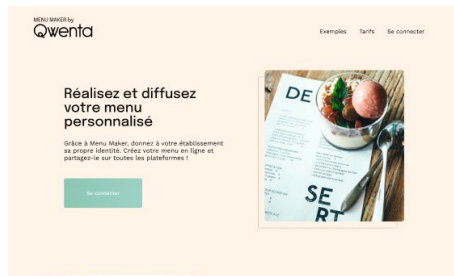
- *Brève introduction du site de Menu Maker :*



**Menu Maker by Qwenta** est une interface en ligne permettant aux restaurateurs de créer, personnaliser et diffuser leurs menus de manière dynamique. Les utilisateurs peuvent ajouter des plats avec descriptions et prix, personnaliser le design en intégrant leur logo et en choisissant les polices et couleurs, puis exporter le menu en PDF, le partager sur des plateformes comme Deliveroo et Instagram, ou l'imprimer. Ce projet s'inscrit dans la volonté de Qwenta, leader historique de l'impression de supports professionnels, de diversifier ses activités en proposant des outils numériques innovants à ses clients.

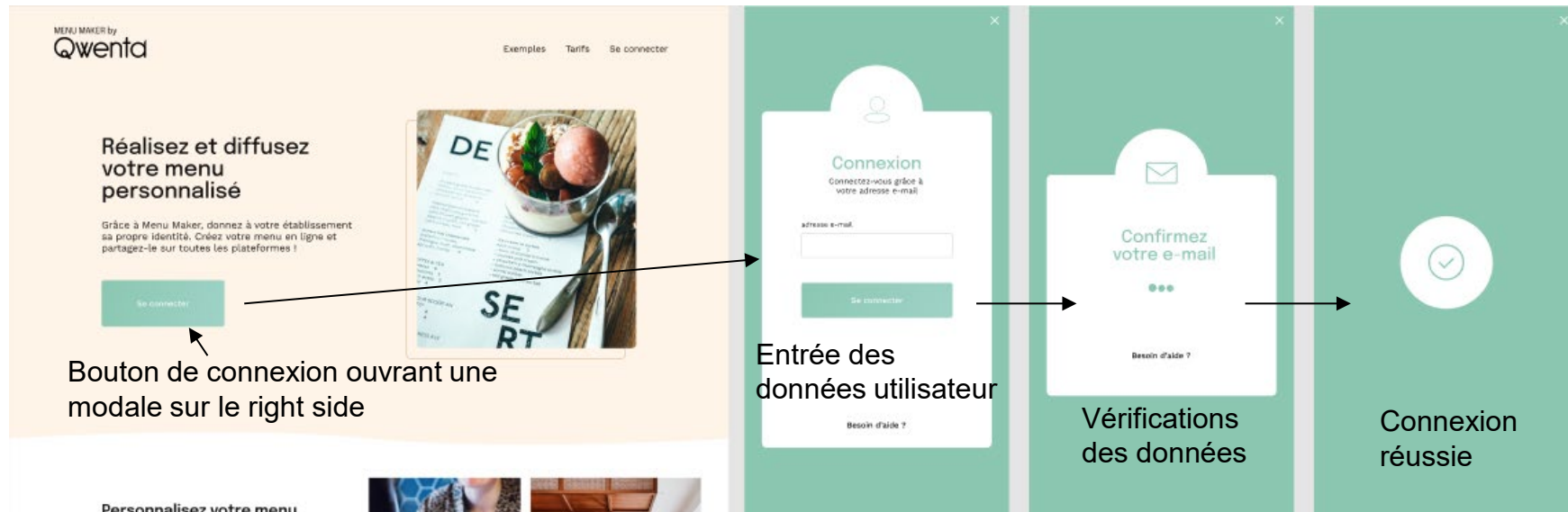
# Aperçu de la maquette

- Capture(s) d'écran de la maquette.



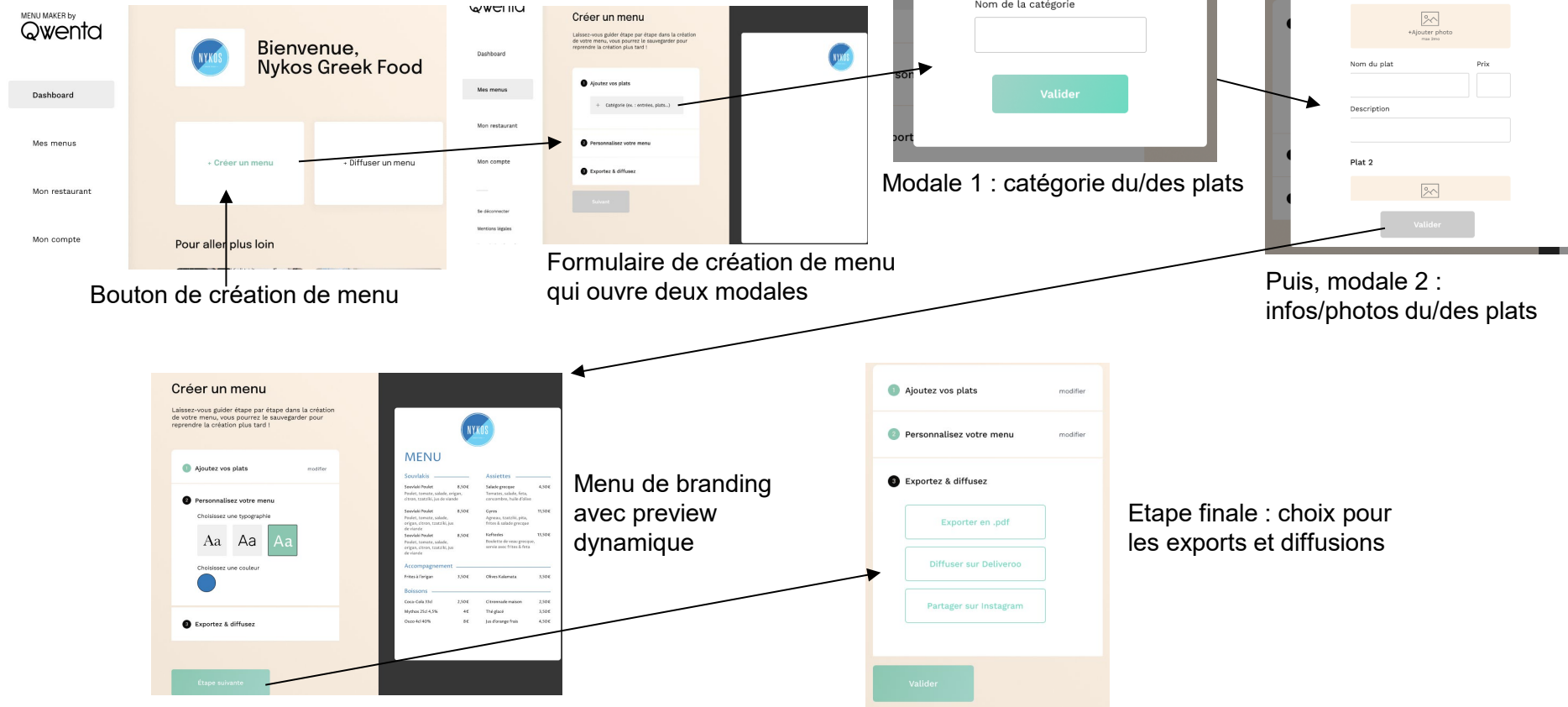
# Aperçu de la maquette

- *Démonstration des fonctionnalités clés*



# Aperçu de la maquette

## • Démonstration des fonctionnalités clés



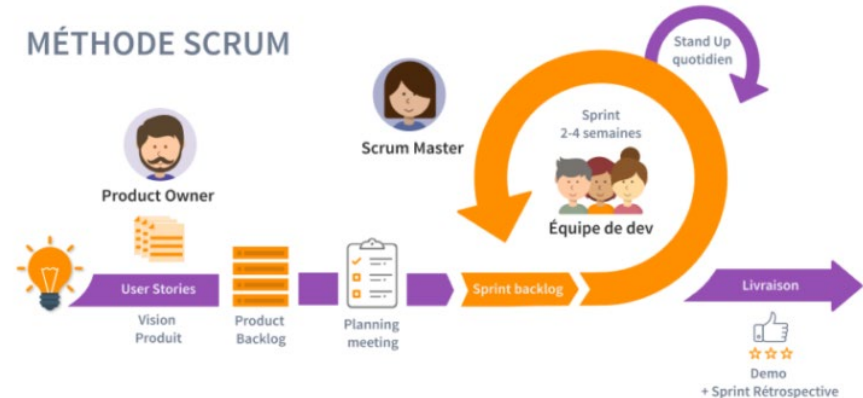
# Méthodologie utilisée

## 1. Explication brève de la méthodologie Agile et Scrum

La **méthodologie Agile** est une approche de gestion de projet qui privilégie l'adaptabilité, la collaboration et l'amélioration continue. Elle repose sur des itérations courtes appelées **sprints**, permettant d'avancer progressivement en livrant des parties fonctionnelles du projet.

Le **Scrum** est un cadre Agile qui définit des rôles, des événements et des artefacts pour structurer le développement d'un projet :

- **Rôles :**
  - Product Owner** : responsable du backlog produit (liste des fonctionnalités à développer).
  - Scrum Master** : veille à l'application des principes Agile et Scrum.
  - Équipe de développement** : réalise les tâches planifiées.
- **Événements principaux :**
  - Sprint Planning** : planification du sprint.
  - Daily Scrum** : réunion quotidienne pour suivre l'avancement.
  - Sprint Review** : présentation des fonctionnalités développées.
  - Sprint Retrospective** : amélioration continue du processus.



# Méthodologie utilisée

## 2. Description de la structuration en sprints, leurs durées et les objectifs

Un **sprint** est un cycle court (généralement 1 à 4 semaines) durant lequel l'équipe travaille sur un ensemble de tâches définies dans le backlog.

### Objectifs des sprints :

- Livrer une **version fonctionnelle** du produit à chaque sprint.
- Faciliter les ajustements en fonction des retours utilisateurs.
- Assurer une amélioration continue grâce aux rétrospectives.

### Exemple pour le projet Menu Maker :

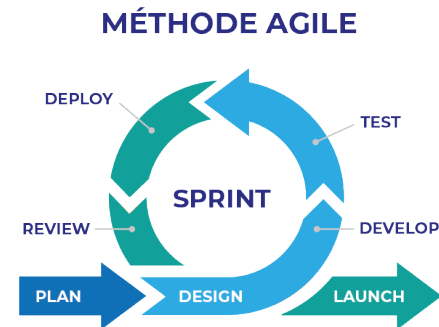
- Sprint 0 : préparation de l'environnement, du backlog et des maquettes pour lancer le développement
- Sprint 1 : Mise en place de l'environnement et création du design initial.
- Sprint 2 : Développement des fonctionnalités principales (gestion des menus).

## 3. Avantages de cette approche pour le projet Menu Maker

- Flexibilité** : adaptation rapide aux besoins des utilisateurs.
- Livraison progressive** : chaque sprint apporte une valeur ajoutée.
- Collaboration accrue** : échanges fréquents entre les parties prenantes.
- Amélioration continue** : identification des axes d'amélioration après chaque sprint.
- Réduction des risques** : détection et correction des problèmes au fur et à mesure.

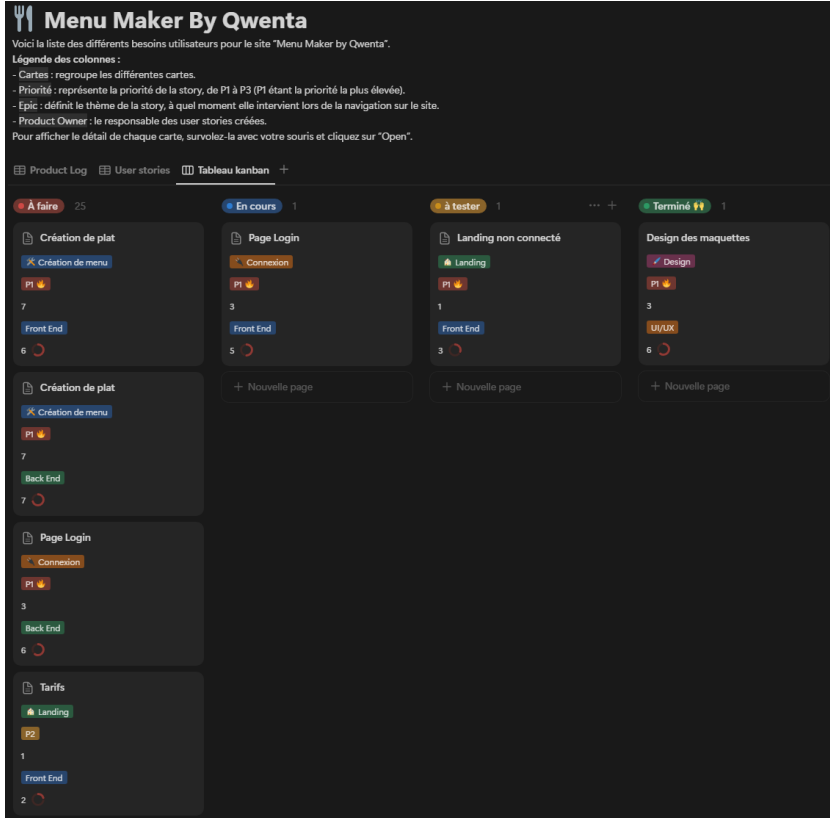
## 4. Méthodologie utilisée

- Pour le projet **Menu Maker**, l'approche Agile et le cadre Scrum sont utilisés afin de :  
Décomposer le projet en **sprints clairs et définis**.
- Favoriser une **communication fluide** entre les membres de l'équipe.
- Assurer un **développement itératif** permettant d'intégrer rapidement les retours clients.





# Suivi du projet avec le Kanban



## Tableau Kanban via notion

Un **Kanban** est un tableau visuel utilisé pour organiser et suivre les tâches d'un projet. Dans **Notion**, il est structuré en colonnes comme "**À faire**", "**En cours**", "**À tester**", et "**Terminé**".

Chaque tâche est représentée par une **carte**, qui contient des détails comme la priorité, les membres responsables, et l'état d'avancement. En déplaçant les cartes entre les colonnes, on suit facilement la progression des tâches.

💡 Avec Notion, le Kanban centralise toutes les informations du projet, facilite la communication et garantit une vue claire pour toute l'équipe.

Lien du Kanban : [ICI](#)

# Suivi du projet avec le Kanban

## Informations légales

- Epic: **Landing**
- Priorité: **P2**
- État: **À faire**
- Description: En tant qu'internaute, je dois pouvoir accéder au contenu "Mentions légales" dans une modale, et l'information "Tous droits réservés" doit être affichée.
- # Temps estimé: 1
- Product Owner: **Soufiane**
- Solutions technique...: **FE → Modale React contenant le texte statique.**
- # Story points: 2
- équipe: **Front End**

## User story

En tant qu'internaute, je dois pouvoir accéder au contenu "Mentions légales" dans une modale, et l'information "Tous droits réservés" doit être affichée.

## Précisions

Les éléments texte contenus dans la modale sont statiques et ne sont pas amenés à bouger pour le moment ; elle peut donc être statique.

## Succès

- ☐ L'internaute doit pouvoir cliquer sur "Mentions légales" depuis les pages connectées et déconnectées
- ☐ La modale s'ouvre alors
- ☐ La mention "Tous droits réservés" doit figurer sur la page d'accueil et sur toutes les autres pages

## Design

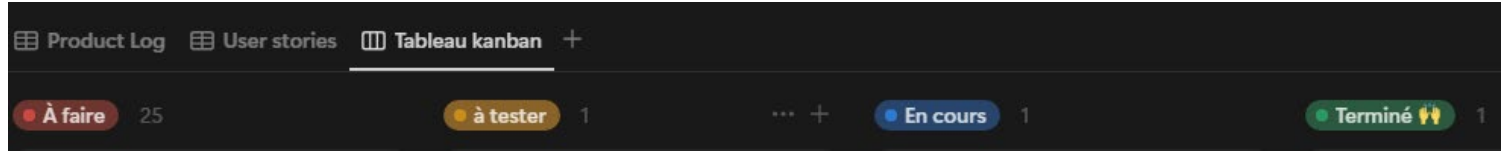
[Lien Maquette Figma](#)

## Les User Stories (US), tâches attribuées, etc.

Les User Stories décrivent les besoins utilisateurs en langage simple (ex. : "En tant qu'utilisateur, je veux pouvoir me connecter").

- Chaque User Story est décomposée en tâches spécifiques attribuées à des développeurs (Front-End, Back-End).
- Les priorités (P1, P2, P3) et deadlines permettent de structurer le travail efficacement.

# Suivi du projet avec le Kanban



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

Le tableau Kanban centralise les tâches et leur progression :

- **Suivi** : Tout le monde peut voir quelles tâches sont en cours ou terminées en un coup d'œil. (avec les états (à faire, en cours, à tester, terminée..))
- **Coordination** : Les développeurs savent exactement quoi faire et peuvent collaborer facilement en fonction des priorités et des dépendances.

# Spécifications techniques

- *Liste des principales spécifications techniques.*

## 1. Technologies utilisées

- **Front-end** : React.js pour une interface dynamique et réactive.
- **Back-end** : Node.js avec Express.js pour gérer les requêtes et les API.
- **Base de données** : MongoDB hébergé sur MongoDB Atlas pour stocker les menus et les utilisateurs.

## 2. Fonctionnalités techniques principales

- Création et gestion de menus avec ajout de plats via un formulaire dynamique à l'intérieur d'une modale.
- Personnalisation des menus avec un éditeur visuel dynamique permettant d'ajouter un logo, des couleurs, des polices et une mise en page.
- Génération dynamique des menus en PDF grâce à la bibliothèque **jsPDF**.
- Intégration des API **Deliveroo** et **Instagram** pour partager les menus directement depuis la plateforme.
- Impression des menus via un lien avec le back-office de Qwenta.

## 3. Sécurité et authentification

- Utilisation de **JWT (JSON Web Token)** pour sécuriser l'authentification et protéger les accès aux fonctionnalités sensibles.
- Stockage des mots de passe avec **bcrypt** pour garantir leur confidentialité.
- Protection contre les attaques **XSS** et **CSRF** pour empêcher les injections malveillantes.
- Sécurisation des API avec **OAuth 2.0** et **rate-limiting** pour éviter les abus.

## 4. Hébergement et compatibilité

- **Hébergement** sur **Vercel**, idéal pour une application front-end scalable.
- **Compatibilité** assurée uniquement avec les navigateurs récents (Chrome, Firefox, Safari).

## Spécifications techniques

- *Choix d'une spécification technique clé à vulgariser.*

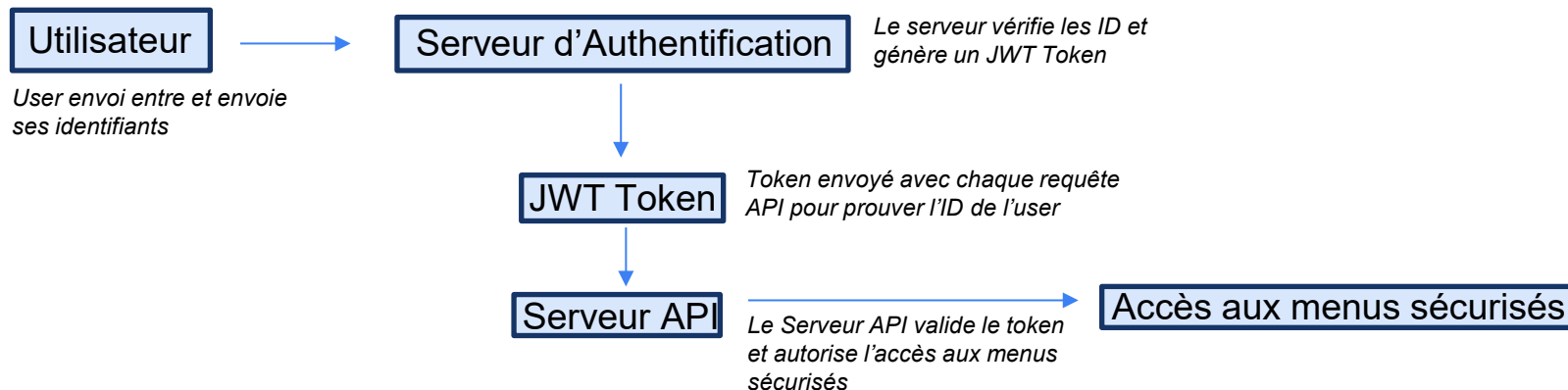
### **Pourquoi JWT ?**

JWT permet aux utilisateurs de se connecter une fois et de rester authentifiés sans avoir à ressaisir leurs identifiants à chaque action.

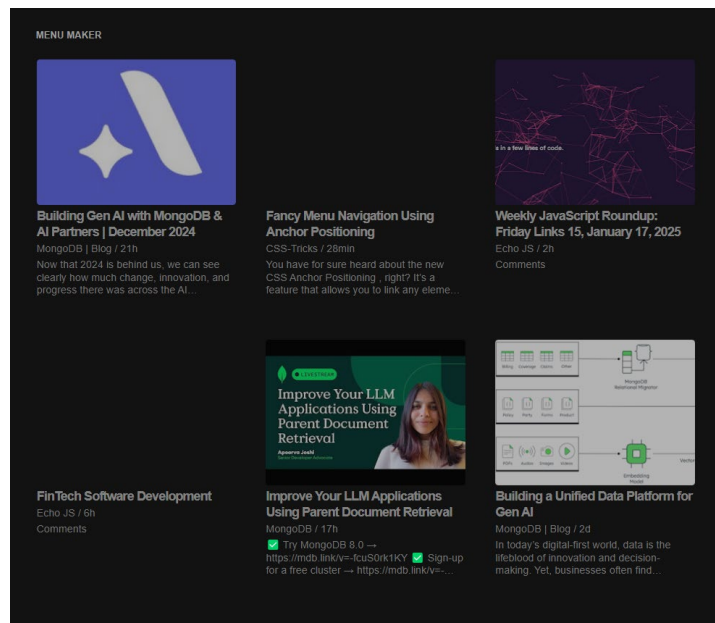
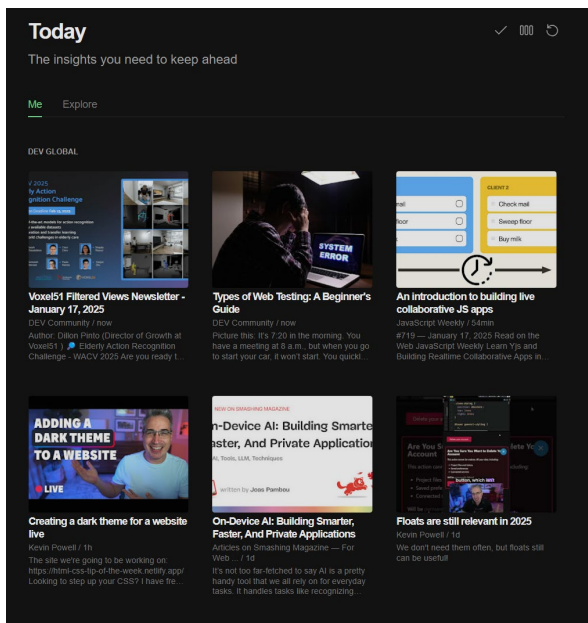
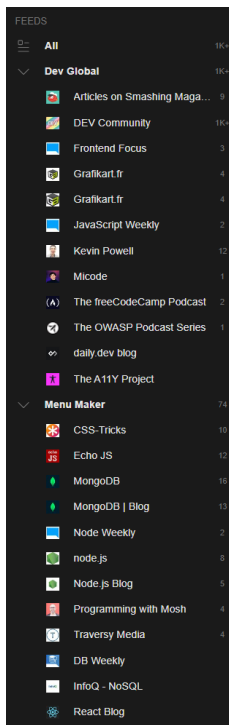
### **Principe simplifié :**

1. L'utilisateur entre son **email et mot de passe**.
2. Le serveur vérifie ses informations et génère un **token JWT**.
3. Ce token est stocké dans le navigateur et envoyé avec chaque requête API.
4. Le serveur vérifie le token pour valider l'accès aux menus personnalisés

### **Schema explicatif :**



## Veille Technologique avec :



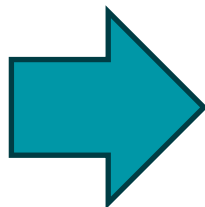
# Veille Technologique

## 2. Méthode de classification des sources d'information



Dans Feedly, les sources d'informations ont été organisées en **deux catégories principales** pour répondre aux besoins spécifiques du projet **Menu Maker** :

Menu Maker	75
CSS-Tricks	10
Echo JS	12
MongoDB	16
MongoDB   Blog	13
Node Weekly	2
node.js	9
Node.js Blog	5
Programming with Mosh	4
Traversy Media	4
DB Weekly	
InfoQ - NoSQL	
React Blog	



### 1. Axe de curation « Menu Maker »

#### Objectif :

- Se concentrer sur les technologies directement liées au projet **Menu Maker**, comme MongoDB, Node.js, et React.

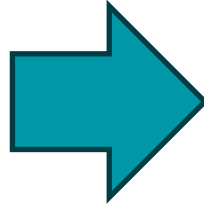
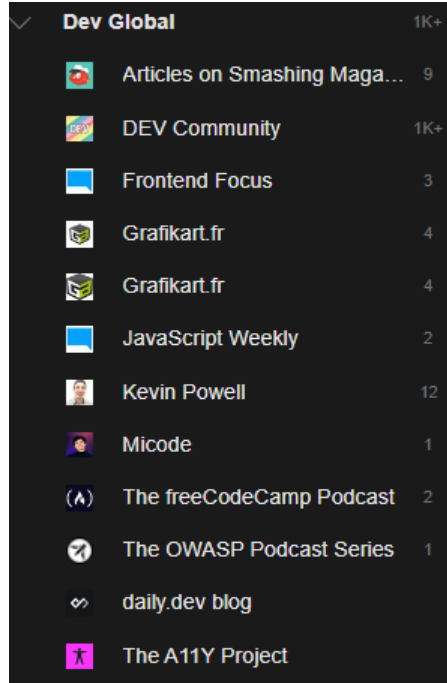
#### Exemples de sources :

- **MongoDB Blog** : Meilleures pratiques pour utiliser MongoDB.
- **Node Weekly** : Contenu dédié à l'écosystème Node.js.
- **CSS-Tricks** : Conseils avancés pour styliser l'interface.
- **React Blog** : Mises à jour et bonnes pratiques pour React.js.
- **Programming with Mosh** : chaîne YouTube incontournable pour apprendre et maîtriser les bases et les outils avancés du développement web et logiciel (React, Node.js, Express, BDD, Javascript).

#### Critère de sélection :

- Sources orientées vers les technologies et outils utilisés dans le projet

# Veille Technologique



## 2. Axe de curation « Dev Global »

### Objectif :

- Suivre les **tendances générales du développement web** pour rester informé des évolutions dans les frameworks, bibliothèques, et meilleures pratiques.

### Exemples de sources :

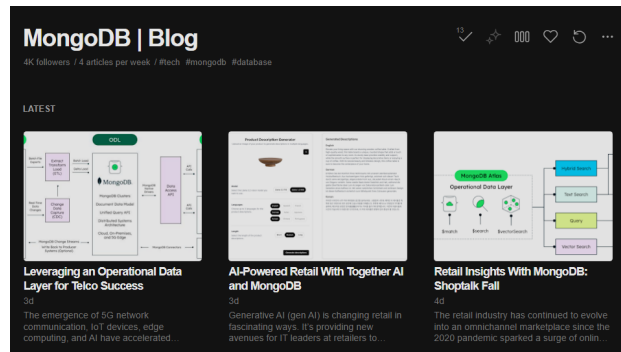
- **DEV Community** : Articles techniques sur tous les aspects du développement.
- **Frontend Focus** : Focus sur les technologies front-end.
- **The A11Y Project** : Ressources pour l'accessibilité web.
- **JavaScript Weekly** : Mise à jour sur l'écosystème JavaScript.
- **Micode** :

### Critère de sélection :

- Blogs, newsletters, et podcasts abordant des sujets globaux liés au développement web (HTML, CSS, JavaScript, accessibilité).



# Veille Technologique



## Axe 1 : "Menu Maker" (Focus projet spécifique)

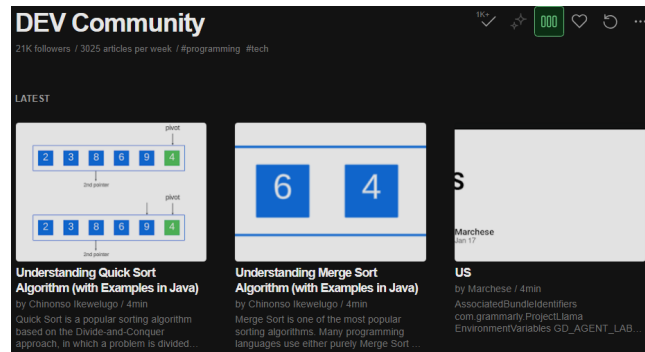
Source choisie : [MongoDB Blog](#)

### Pourquoi MongoDB Blog ?

- Ce blog offre des articles et des guides officiels sur l'utilisation et l'optimisation de **MongoDB**, qui est la base de données principale du projet Menu Maker.
- Les articles expliquent des sujets pratiques comme le **design des schémas NoSQL**, les meilleures pratiques pour les requêtes, et l'utilisation de **MongoDB Atlas**, une solution cloud adoptée pour ce projet.

### Impact sur le projet :

Cette source garantit que l'équipe de développement reste informée sur les bonnes pratiques et les nouveautés liées à MongoDB, ce qui améliore la fiabilité et la scalabilité de la base de données.



## Axe 2 : "Dev Global" (Tendances générales du développement)

Source choisie : [DEV Community](#)

### Pourquoi DEV Community ?

- C'est une plateforme collaborative regroupant des articles, des tutoriels, et des discussions sur **les dernières tendances du développement web** : React.js, Node.js, CSS3, JavaScript ES6, et plus encore.
- Cette diversité de contenu aide à résoudre des problématiques techniques variées et à découvrir de nouvelles méthodes ou outils.

### Impact sur le projet :

DEV Community est utile pour se tenir à jour sur les frameworks utilisés dans le projet et pour appliquer des pratiques modernes dans le développement front-end et back-end.

# Veille Technologique

## Contribution de la veille aux spécifications techniques :

*La veille contribue directement à l'élaboration des spécifications techniques en fournissant des informations pertinentes et à jour*

- ➡ **Choix des technologies adaptées :**
  - Grâce aux sources comme MongoDB Blog et DEV Community, la veille aide à sélectionner les outils idéaux pour le projet, comme MongoDB Atlas pour la base de données ou React.js pour le front-end.
- ➡ **Mise en œuvre des bonnes pratiques :**
  - Des ressources comme The A11Y Project ou OWASP Podcast permettent d'intégrer des normes modernes en sécurité (JWT, CSRF) ou accessibilité (attributs ARIA).
- ➡ **Anticipation des contraintes techniques :**
  - Les guides sur les API (Deliveroo, Instagram) ou la gestion de bases de données aident à identifier des solutions concrètes face aux défis techniques du projet.
- ➡ **Justification des choix techniques :**
  - La veille facilite la comparaison entre différentes solutions (ex. Puppeteer vs jsPDF) pour sélectionner et justifier les approches les plus pertinentes dans les spécifications.

## Conclusion

*La veille enrichit les spécifications techniques en garantissant des choix éclairés, des pratiques modernes, et des solutions adaptées aux besoins du projet.*

# Conclusion

## Résumé des points clés de la présentation :

### 1.Contexte du projet :

- **Menu Maker by Qwenta** est une plateforme destinée aux restaurateurs pour créer, personnaliser et diffuser leurs menus de manière dynamique.

### 2.Méthodologie utilisée :

- Adoption de la **méthodologie Agile et Scrum** pour structurer le projet en **sprints courts**, favorisant la collaboration et l'amélioration continue.

### 3.Tableau Kanban :

- Utilisation de **Notion** pour organiser et suivre les tâches via un tableau Kanban. Cela a permis de centraliser les informations, de coordonner l'équipe, et de visualiser la progression en temps réel.

### 4.Spécifications techniques :

- Technologies principales :
- **Front-End** : React.js.
- **Back-End** : Node.js et Express.js.
- **Base de données** : MongoDB sur Atlas.
- Fonctionnalités phares : création de menus, export en PDF (jsPDF), intégration API (Deliveroo, Instagram), et sécurité renforcée (JWT, CSRF).

### 5.Veille technologique :

- Organisation en deux axes :
- **"Menu Maker"** : Focus sur les outils clés du projet (ex. MongoDB Blog, React Blog).
- **"Dev Global"** : Tendances générales du développement (ex. DEV Community, JavaScript Weekly).
- La veille a permis de choisir des technologies adaptées, de suivre les bonnes pratiques, et d'anticiper les contraintes techniques.

## *Le mot de la fin*

La présentation a démontré une approche structurée et moderne pour le développement du projet Menu Maker, en mettant en avant les outils technologiques, une méthodologie agile, et une veille technologique efficace pour garantir le succès du projet.

QUESTIONS ?