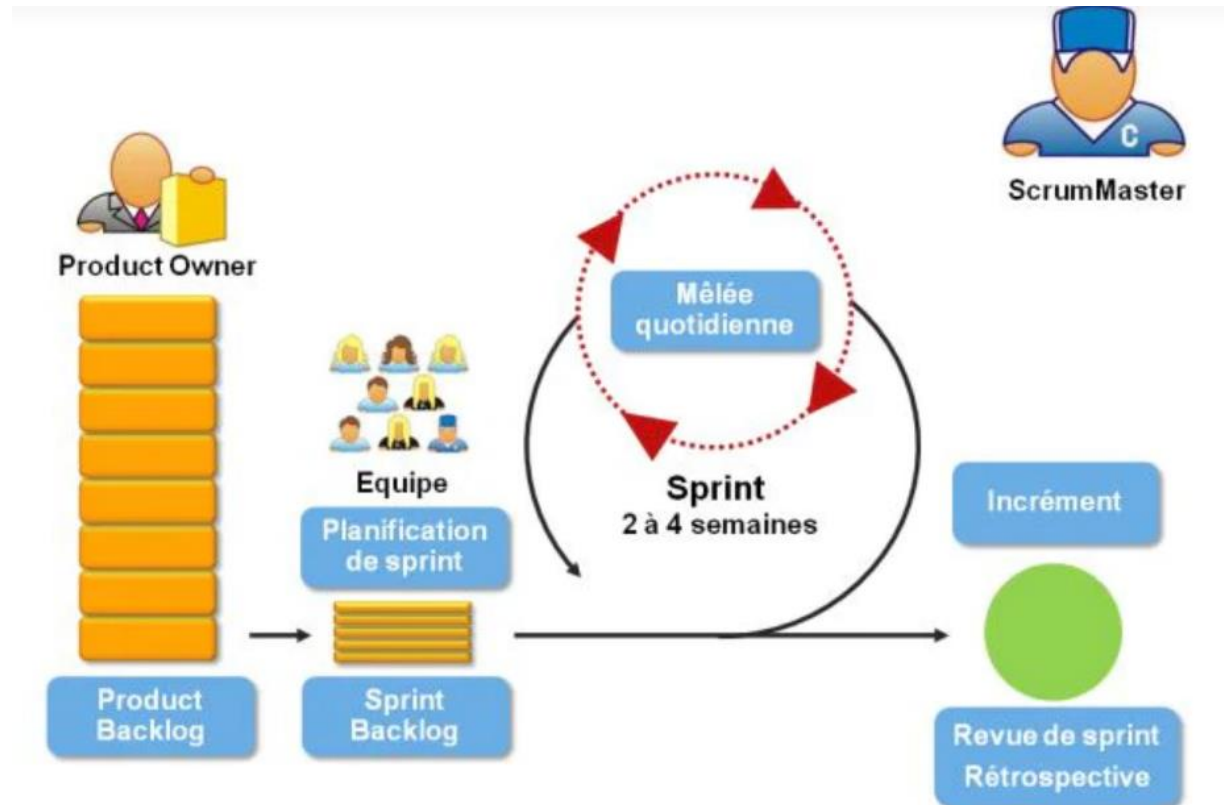


# **SOLUTION TECHNIQUE**

pour le site  
MenuMaker

## DEROULEMENT DU PROJET SELON LA METHODOLOGIE AGILE SCRUM



L'équipe et le client communiquent via SLACK.

Les développeurs utilisent Visual Studio Code pour coder et déposent leur code sur GitHub.

1

### Préparez le Sprint

Identifiez et découpez techniquement les tickets à traiter pendant le Sprint. Estimez la complexité des tickets et engagez-vous collectivement sur des objectifs à atteindre pour le Sprint à venir.

2

### Réalisez le Sprint

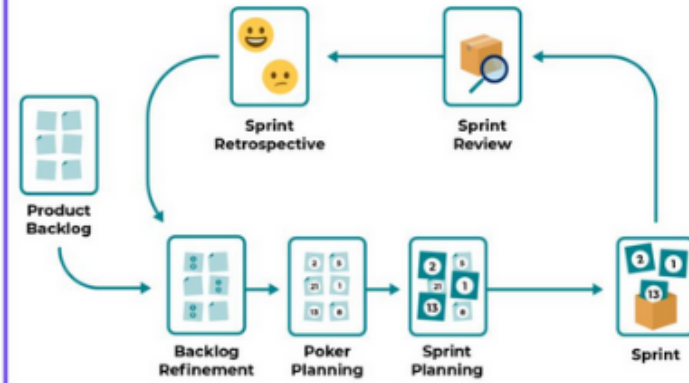
Privilégiez le développement en binôme et les différentes pratiques de développement agile. Communiquez quotidiennement sur vos avancées et validez collectivement le code..

3

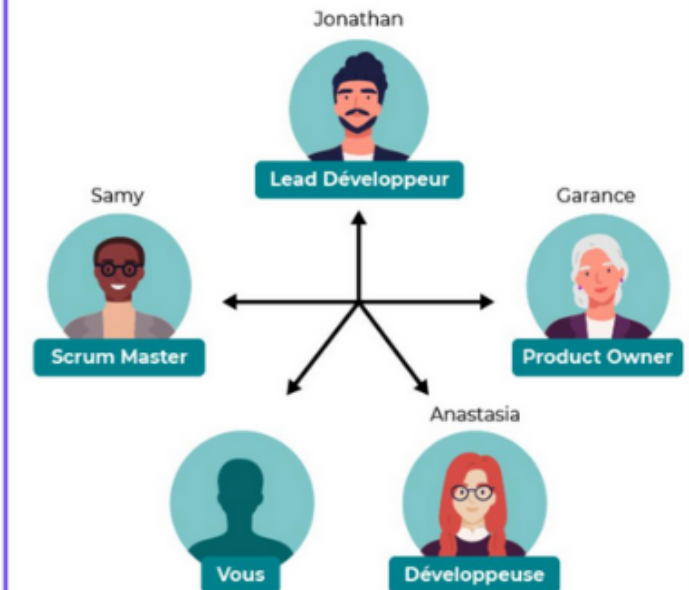
### Faites le bilan du Sprint et itérez

Participez à la Sprint Review et à la Rétrospective. À l'issue de ces deux réunions, itérez en préparant le Sprint suivant.

### Les différentes étapes d'un Sprint



### Les membres de votre équipe agile



### Définitions

#### Les évènements en Scrum

**Sprint** : cycle de développement au cours duquel les développeurs traitent les tickets sur lesquels ils se sont engagés.

**Daily Scrum** : réunion courte et quotidienne où les développeurs font le point sur les avancées et les éventuels points bloquants.

**Backlog Refinement** : réunion d'équipe qui a pour objectif d'identifier et de découper techniquement les tickets à traiter pendant le Sprint.

**Sprint Review** : réunion de bilan du Sprint qui permet à l'équipe Scrum de présenter aux parties prenantes le travail réalisé et de redéfinir la vision du produit en fonction des besoins.

**Sprint Planning** : réunion au cours de laquelle l'équipe s'engage collectivement dans le traitement d'un nombre donné de tickets pendant le Sprint.

**Rétrospective** : réunion de l'équipe Scrum qui consiste à dresser le bilan humain du Sprint, et à identifier des axes d'amélioration.

#### Les outils et méthodes agiles

**Planning Poker** : pratique agile qui permet, sous la forme d'un jeu de cartes, d'estimer collectivement l'effort de développement des tickets à traiter.

**KISS (Keep It Simple and Stupid)** : principe de développement qui consiste à favoriser un code simple et compréhensible par tous, facilement modifiable et maintenable.

**Peer programming (ou Programmation par pair)** : pratique de développement qui consiste à travailler en binôme ou en équipe sur le traitement d'un ticket.

**Refactoring** : opération qui consiste à modifier le code source d'un logiciel, de manière à l'améliorer en continu.

**TDD (Test Driven Development)** : pratique de développement qui consiste à concevoir un logiciel par petites étapes. Les tests sont réalisés avant de développer chaque partie du code correspondante.

L'équipe de développeurs est composée :  
d'un développeur FRONT-END et d'un développeur BACK-END.  
Ils travaillent sur Visual studio code, le code est sauvegardé en ligne au fur et à mesure de l'avancement du projet sur GITHUB.

## Développeur front-end



### Responsabilités :

- Implémenter la logique de fonctionnement de l'interface
- Implémenter la logique de connexion au back end
- Tester l'application sur différents navigateurs

## Développeur back-end



### Responsabilités :

- Traiter les données envoyées par le front end
- Échanger avec la base de données
- Implémenter la logique de fonctionnement côté serveur
- Créer dynamiquement les pages à renvoyer au front end mobile

## Langages et outils utilisés par un développeur FRONT-END

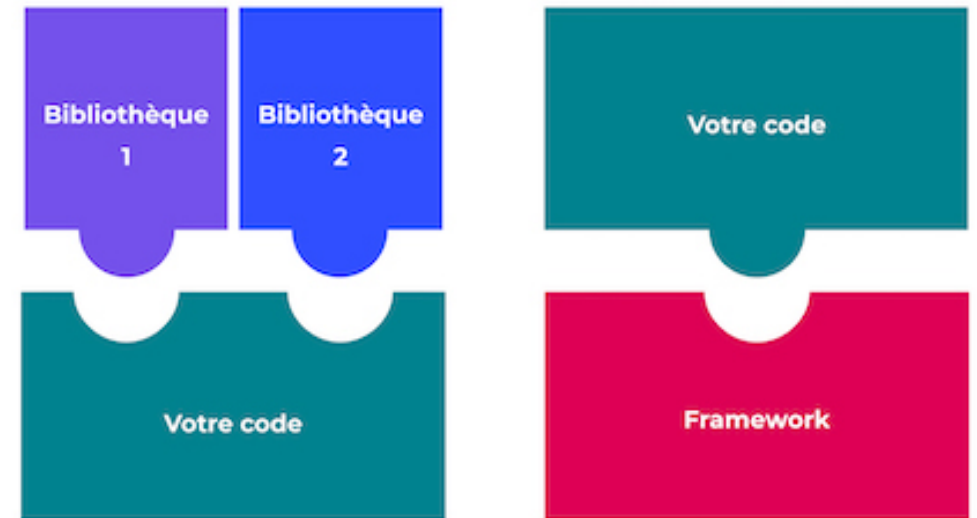
**HTML** : langage de balisage pour le contenu d'une page web

**CSS** : langage qui décrit la présentation des documents HTML et XML

**JAVASCRIPT** : langage de programmation pour dynamiser les pages web. Pour gagner du temps, on peut utiliser des bibliothèques ou des frameworks.

**Bibliothèque ou librairie** : c'est un ensemble d'extensions et de programmes prêts à l'emploi pour faciliter la réalisation de fonctionnalités répétitives et routinières. Elles sont enregistrées dans votre code.

**Framework** : regroupe un ensemble de composants permettant de créer les fondations ainsi que les grandes lignes d'un logiciel/site web/application mobile.



Les frameworks servent de support au code tandis que les bibliothèques viennent s'y intégrer



## OUTILS issus de la VEILLE TECHNIQUE sélectionnés pour notre développeur FRONT-END

**React.js** : est une bibliothèque de composants javascripts pour créer des interfaces utilisateurs dynamiques et réutilisables



**React-transition-group** : Cette bibliothèque permet de créer des effets.



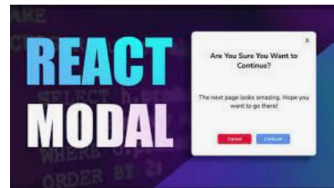
**react-bootstrap** : C' est un ensemble de composants React préconstruits basés sur Bootstrap ( qui est un framework CSS).



**React-router** : C'est une bibliothèque qui permet de transformer une application react en SPA (single page application).



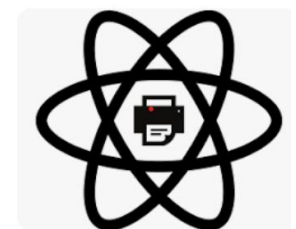
**React-modal** : bibliothèque qui permet de créer et de gérer des modales performantes et accessibles avec un minimum de code.



**React-print** : C'est une bibliothèque de react qui permet d'imprimer des composants react dans le navigateur.



**React-pdf** : C'est une bibliothèque react qui permet de créer des fichiers pdf sur le navigateur et le serveur.



## Langages et outils utilisés par un développeur BACK-END

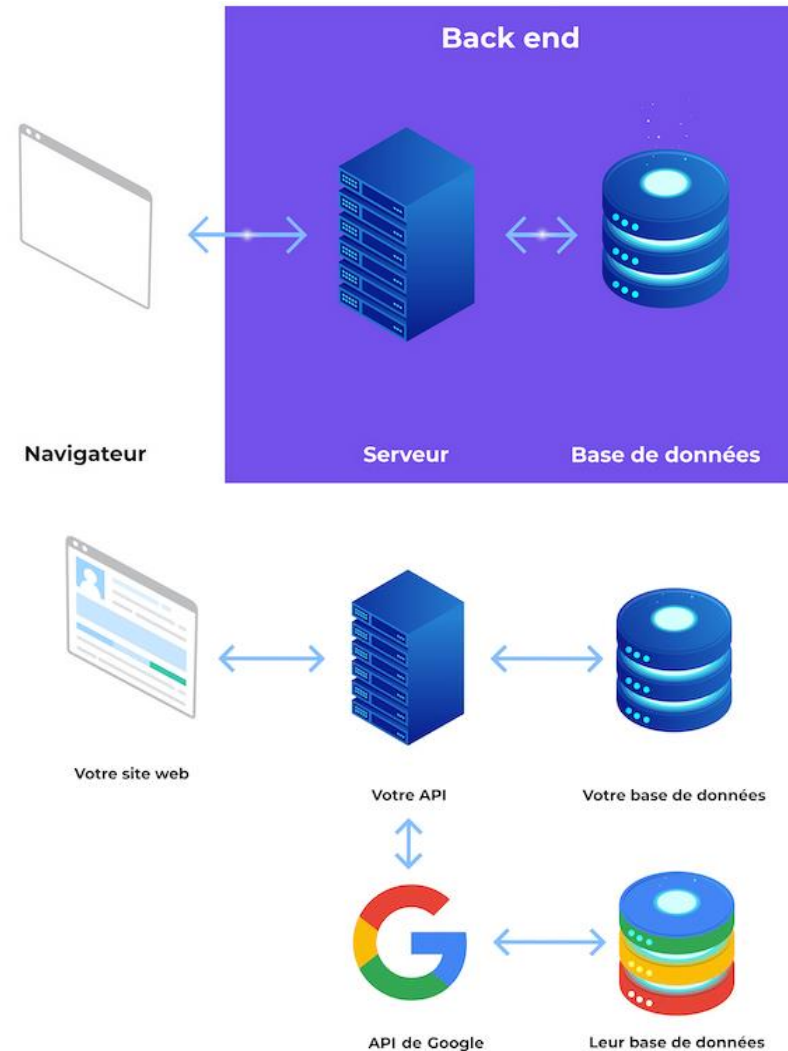
**Langages de programmation back-end** : comme JAVA, PYTHON, PHP et JAVASCRIPT avec NODE.js.

**Serveur** : Un serveur web est soit un logiciel de service de ressources web, soit un serveur informatique qui répond à des requêtes du World Wide Web sur un réseau public ou privé, en utilisant principalement le protocole HTTP

**Base de données** : Une base de données permet de stocker et de retrouver des données structurées, semi-structurées ou des données brutes ou de l'information, souvent en rapport avec un thème ou une activité ; celles-ci peuvent être de natures différentes et plus ou moins reliées entre elles.

**API** : c' est un service web accessible 24 heures sur 24. Son but est de fournir un accès à des données mises à jour régulièrement. Elle accepte également de nouvelles données pour les stocker ou pour déclencher des calculs.

Exemples d'outils pour documenter une API : swagger, postman



## OUTILS issus de la VEILLE TECHNIQUE sélectionnés pour notre développeur BACK-END

**Node.js** : Node.js est un environnement d'exécution qui permet de créer toutes sortes d'applications et d'outils coté serveur en javascript.



**Express.js** : c'est un framework pour node.js. Express.js est utilisé pour la création d'API et d'applications mobiles. Il prend en charge les détails essentiels du back-end tel que les sessions, la gestion des erreurs et le routage.



**MongoDB** : MongoDB est une base de données no sql orientée documents



**openAPI swagger** : c' est un langage de description d'interface permettant de décrire des API exprimées à l'aide de JSON



**Auth0** : C'est une bibliothèque pour authentifier l'utilisateur



**Nodemailer** : c'est une librairie pour Node.js qui facilite l'envoi d'email



**DELIVEROO** : API crée par le site internet DELIVEROO



**API graph d'Instagram** : L'API Basic Display pour Instagram permet aux utilisateurs d'une application d'obtenir des informations générales sur un profil, des vidéos ou des photos d'un compte Instagram





# I PLANIFICATION DU PROJET AVEC LES SPECIFICATIONS TECHNIQUES

Sprint 0 : du 22 au 24/05

**SPRINT 0** : période dédiée à l'**organisation du projet** :  
prise de connaissance avec le produit et  
avec les attentes des utilisateurs,  
recrutement de l'équipe et création du backlog.

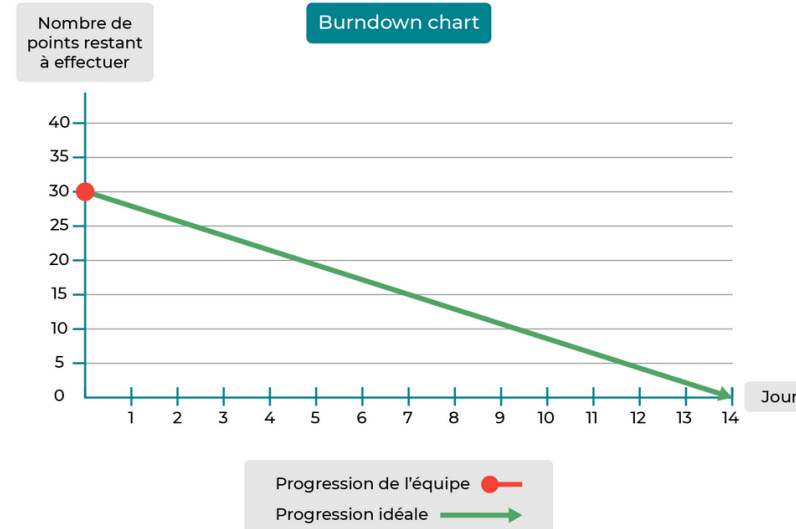
25/05 à 8h réunion : **Backlog refinement** avec l'équipe

26/05 à 8h réunion : **Poker planning** avec l'équipe

26/05 à 10h réunion : **Sprint planning** avec l'équipe

Les développeurs utiliseront pour chaque Sprint un **Burndown Chart** (ou graphique d'avancement)

C'est un support qui permet de visualiser les points restants à faire par l'équipe tout au long du Sprint.



## 1<sup>er</sup> sprint du lundi 29/05 au dimanche 11/06/2023 (2 semaines)

Nb : nombre de page de la maquette P : priorité US : user story spé : spécifications stp : story points						FRONT END			BACK-END		
Nb	P	US	Nom de la US	Nom pages maquette	Travail à faire	Spé techniques	stp	date	Spé techniques	stp	date
1	P1	US1	Landing non connectée	Landing non connectée	FRONT END Implémentation de la landing page	react.js react-bootstrap	5	29-31/05			
2	P1	US2	Page login	-> Log_in_overlay	FRONT-END	react .js react-bootstrap react-modal react-transition-group	3	01-02/06	Node.js express.js openAPI swagger MongoDB  Auth.js nodemailer	7	29/05-02/06
3				->Confirmation_overlay	création de la modale avec le formulaire						
4				->Validation_overlay	BACK-END Création d'un back-end avec une base de données et sa documentation. Création de routes pour la :création de compte utilisateur et la connexion au formulaire						
6	3	US03	Catégorie de plat	Créer un menu_add_category Category_overlay	FRONT-END	react.js react-bootstrap react-modal react-hook form	5	05-08/06			
7					BACK-END crée une route pour l'ajout de catégorie de plat les catégories sont stockées dans la base de données						
									Node.js express.js openAPI swagger MongoDB	2	05-07/06

**Code review :** Jeudi 08/06 à 8h00

**Sprint review :** Vendredi 09/06 à 8h00 à **Réunion** avec toute l'équipe et **le client**

**Rétrospective** Vendredi 09/06 à 9H00 puis on réitère avec le 2<sup>ème</sup> sprint

2<sup>e</sup> sprint

00/00 à 0h00 Backlog refinement

00/00 à 0h00 poker planning

00/00 à 0h00 sprint planning

N	Nom de la US			Nom pages maquette	Travail à faire	Spé techniques	stp	date	Spé techniques	stp	date
						FRONT END			BACK-END		
8	P1	US04	Création de plat	Créer un menu_add_plat	FRONT-END	react-modal react hook form	2		Node.js express.js openAPI swagger MongoDB	2	
9				Plat_overlay							
10				Plat_added_overlay							
11				Créer un menu_plat							
					BACK-END						
					création d'une route pour la création de plat						

						FRONT END			BACK-END		
Nom de la US			Nom pages maquette	Travail à faire		Spé techniques	stp	date	Spé techniques	stp	date
12	P1	US05	Style de menu	Créer un menu_font	FRONT-END	react-bootstrap	5				
13				Créer un menu_font	Implémentation de la section :	plusieurs feuilles					
14				Créer un menu_font_color	Personnalisez votre menu	de style CSS					
15	P2	US06	Exportation pdf	Créer un menu_diffuser 3	FRONT-END	React-pdf	2				
					implémentation de la section: Exportez et diffusez						
	P2	US07	Commander l'impression d'un menu		FRONT-END	React-to-print	2				
					implémenter "imprimer un menu"						
16	P2	US08	Menus précédents	Mes menus	FRONT-END	react-router	1		Node.js express.js openAPI swagger MongoDB	2	
					implémenter l'onglet "mes menus" appel à l'API pour afficher les menus						
					BACK-END						
					créer une route qui réunit tous les menus créés						
	P2	US09	Informations légales		FRONT-END	react-router	2				
					création de la modale						
	P2	US10	Tarifs		FRONT-END	react.js	2				
					création du nouvel onglet "tarifs"	react-router					
	P2	US11	Exportation DELIVEROO		FRONT-END	React+bootstrap	1				
					implémentation du bouton DELIVEROO appel à l'API DELVEROO	Appel à API DELIVEROO					
					BACK-END						
					configuration de l'API DELIVEROO				API DELIVEROO	2	



						FRONT END			BACK-END		
			Nom de la US	Nom pages maquette	Travail à faire	Spé techniques	stp	date	Spé techniques	stp	date
	P2	US12	Partage sur intagram		<div>FRONT-END</div> implémentation du bouton "Partager sur "Partager sur intagram" appelk à l'API GRAPH d'instagram	React+bootstrap Appel à API graph d'instagram	1		API ggraph d'instagram	2	
					<div>BACK-END</div> configuration de l'API GRAPH d'intagram						
	P2	US13	Déconnexion		<div>FRONT-END</div> Script javascript pour se déconnecter	javascript	1				
	P2	US14	Infos utilisateurs		<div>FRONT-END</div> créer un onglet depuis "mon compte" de la sidebar de gauche créer un formulaire pour rajouter plusieurs adresses e-mails reliées à son compte. appel à l'API pour rapatrier les emails d'authentications créer un formulaire pour modifier son adresse e-mail de base	react-bootstrap react hook form	4		Node.js express.js openAPI swagger MongoDB	2	
					<div>BACK-END</div> créer une route pour stocker les adresses e-mail						
5	P2	US15	Dashboard	Dashboard	<div>FRONT-END</div> implémentation du dashboard (la sidebar ayant déjà était crée)	react.js react-router react-bootstrap	4				
	P3	US16	Branding restaurateur		<div>FRONT-END</div> dans l'onglet "mon compte" , créer un formulaire qui pour l'ajout d'un logo qui s'affichera sur le dashboard et sur la carte des menus, et la possibilité de modifier les couleurs de base	React-hook-form	2		Node.js express.js openAPI swagger MongoDB	2	



## Réalisez et diffusez votre menu personnalisé

Grâce à Menu Maker, donnez à votre établissement sa propre identité. Créez votre menu en ligne et partagez-le sur toutes les plateformes !

[Se connecter](#)

## Personnalisez votre menu

Ajoutez vos plats, vos boissons, et organisez votre menu comme le souhaitez. Pour l'impression ou la diffusion en ligne, récupérez votre menu au format que vous souhaitez et diffusez-le auprès de votre clientèle !

[Je me lance](#)

## Comment créer votre menu ?

Réalisez votre menu en seulement trois étapes

1. Listez l'ensemble de vos boissons, entrées, plats, desserts, accompagnements... à afficher sur votre menu.
2. Choisissez le style qui correspond à votre restaurant (logo, couleurs, typog.).
3. Enregistrez votre menu en PDF et diffusez-le sur vos plateformes de vente en ligne.

## II Captures d'écran de la maquette du site

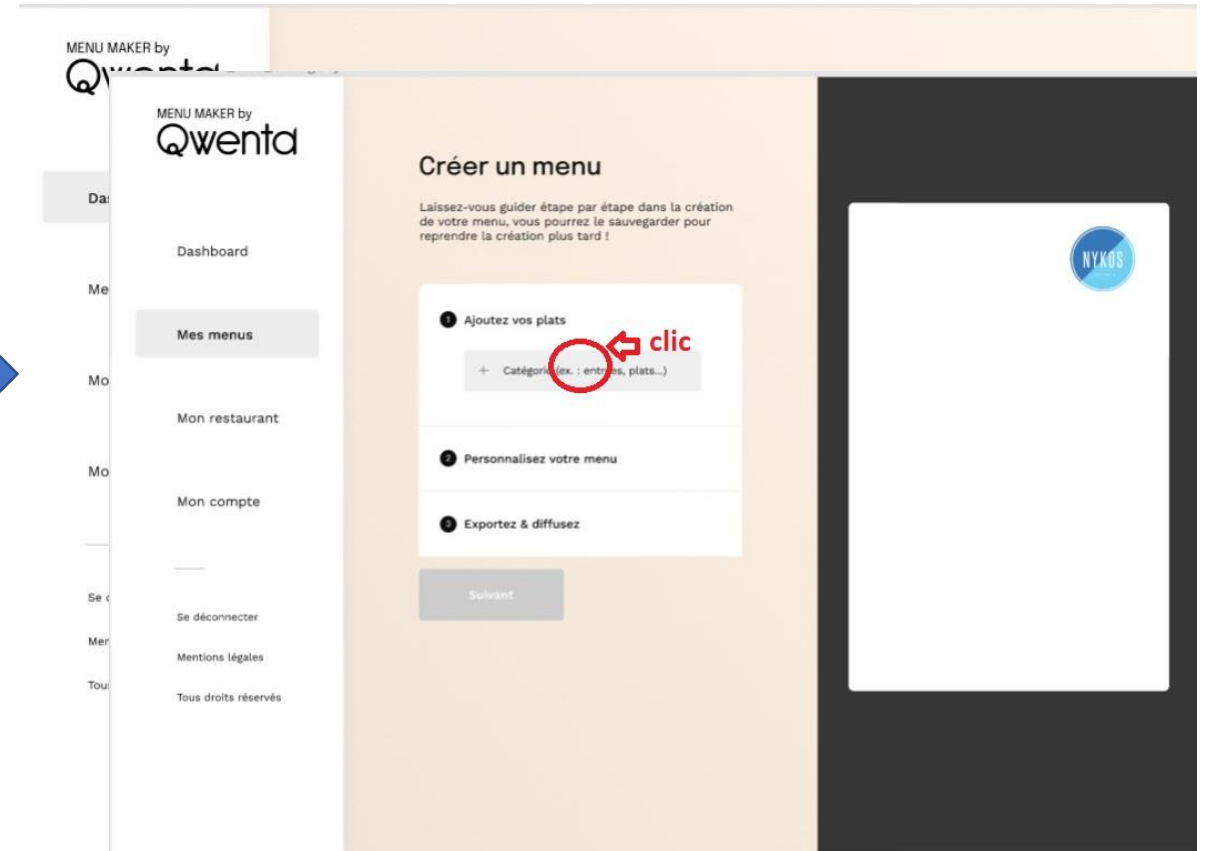
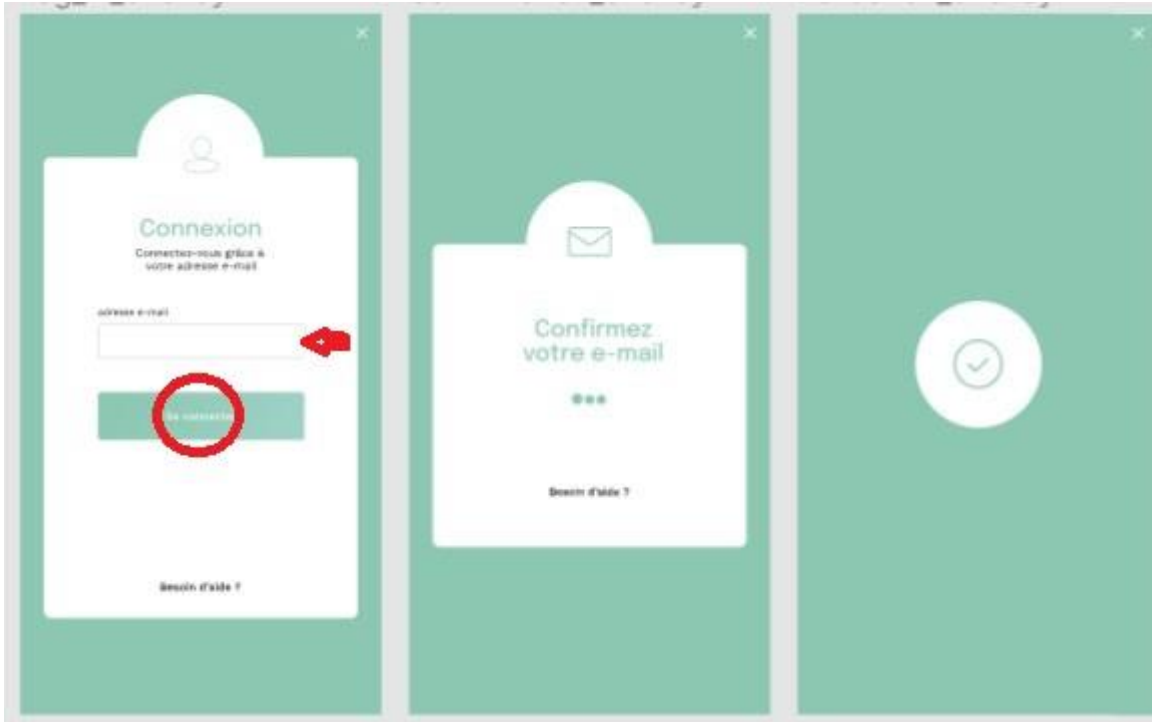
### Spécifications techniques

#### FRONT-END

react.js

react-bootstrap





## Spécifications techniques

### FRONT-END

react .js  
react-bootstrap  
React-modal  
react-transition-group

### BACK-END

Node.js  
express.js  
openAPI  
swagger  
MongoDB

Auth.js  
nodemailer

## Spécifications techniques

### FRONT-END

react.js  
react-bootstrap



### Ajouter une catégorie

Nom de la catégorie

Valider



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

Souvlakis modifier clic


+ Plats (ex. : pâtes, ryros, coca...)

+ Catégorie (ex. : entrées, plats...)

2 Personnalisez votre menu

3 Exportez & diffusez

Étape suivante



Souvlakis ← catégorie ajoutée



## Spécifications techniques

### FRONT-END

react-modal  
react-hook form

### BACK-END

Node.js  
express.js  
openAPI  
swagger  
MongoDB

Ajoutez vos : Souvlakis

Plat 1

  
+Ajouter photo  
max 2mb

Nom du plat

Prix

Description

Plat 2



Valider

### Spécifications techniques

#### FRONT-END

react-modal  
react-hook form

#### BACK-END

Node.js  
express.js  
openAPI  
swagger  
MongoDB

Ajoutez vos : Souvlakis

Plat 1



Nom du plat

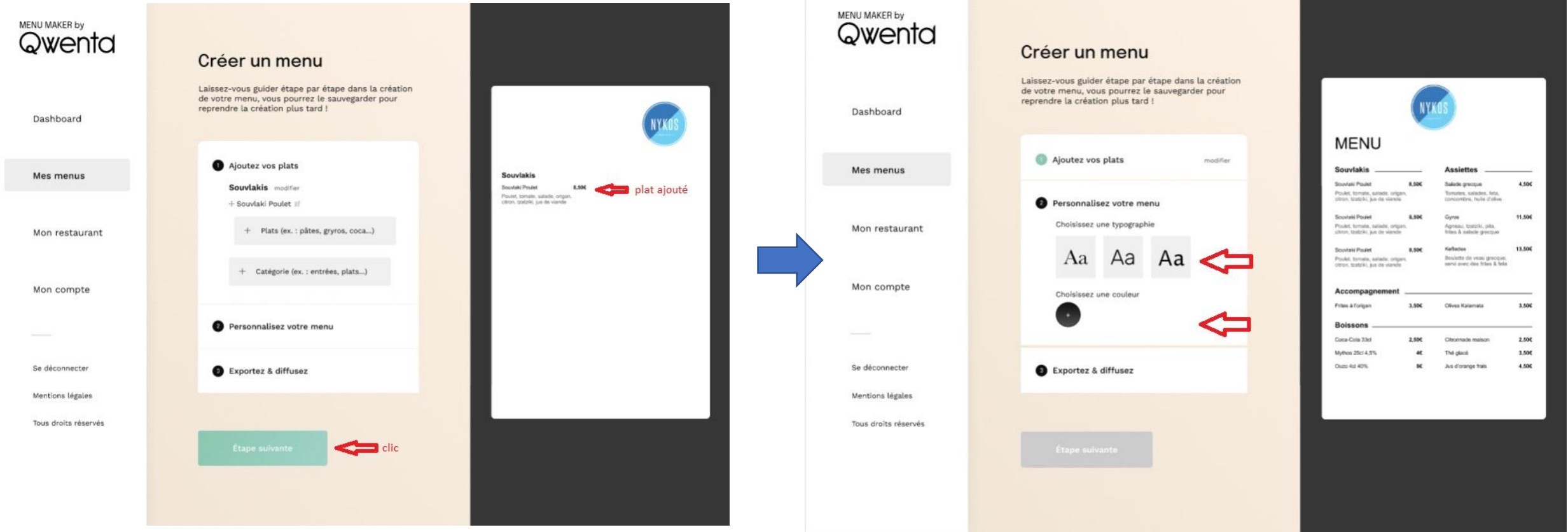
Prix

Description

Plat 2



Valider



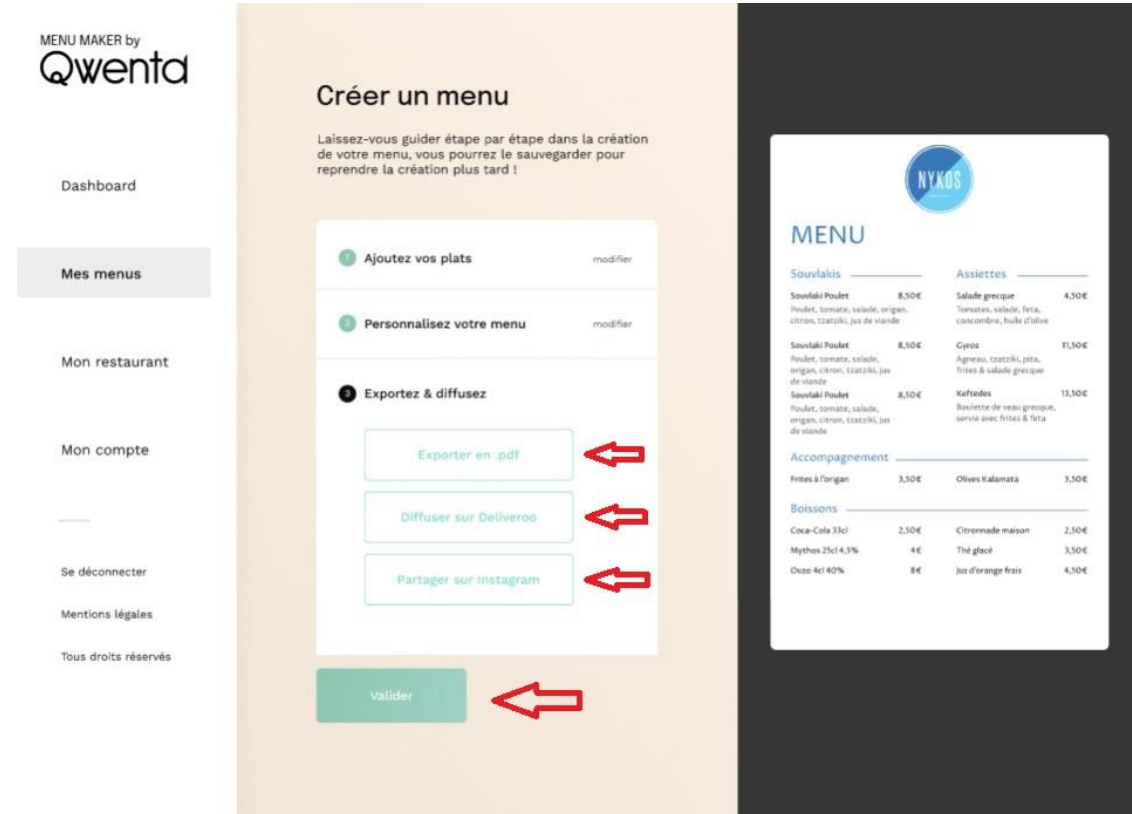
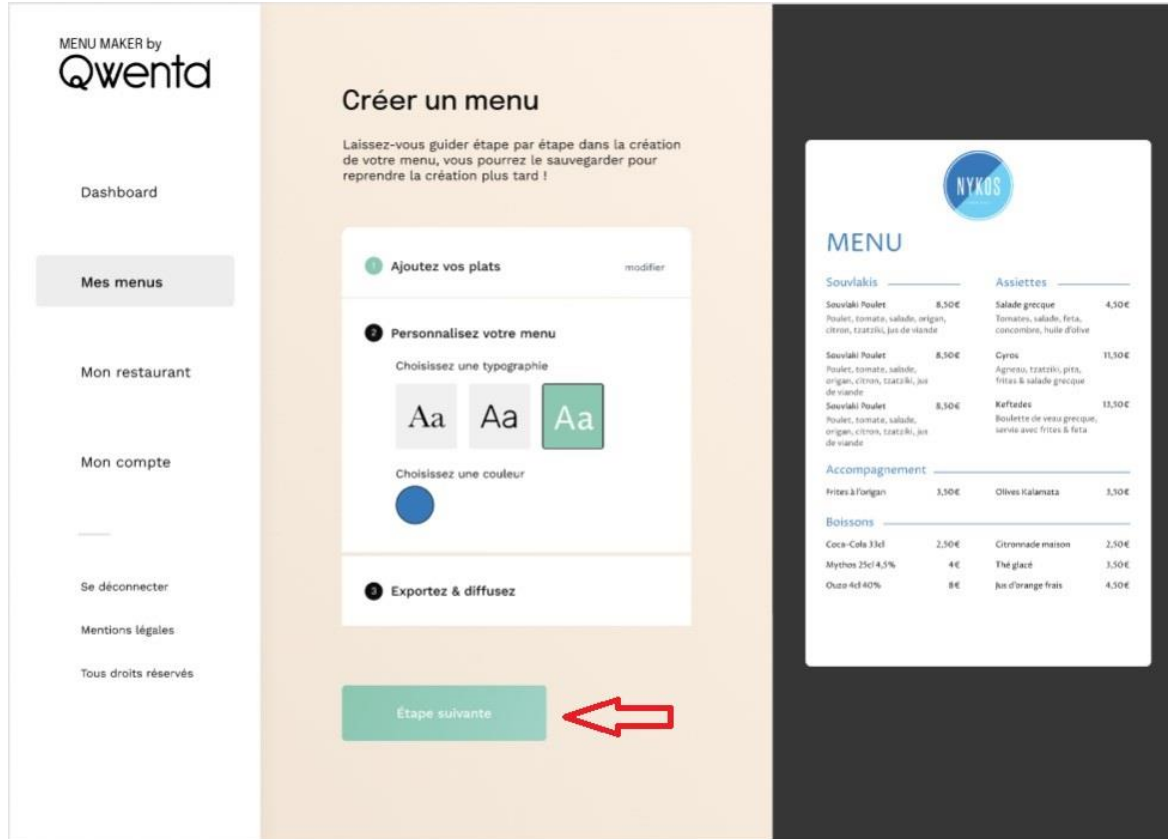
## Spécifications techniques

### FRONT-END

react-bootstrap  
plusieurs feuilles  
de style CSS







## Spécifications techniques

### FRONT-END

react-bootstrap  
plusieurs feuilles  
de style CSS

Appel à API graph  
d'instagram

## Spécifications techniques

### FRONT-END

React-pdf  
React-to-print  
React+bootstrap  
Appel à API  
DELIVEROO

### BACK-END

API DELIVEROO  
API graph  
d'instagram



Dashboard

Mes menus

Mon restaurant

Mon compte

Se déconnecter

Mentions légales

Tous droits réservés

## Mes menus



Le Petit Bistro	
Salade de tomates	€12
Salade de poulet	€15
Salade de boeuf	€18
Salade de porc	€20
Salade de poisson	€22
Salade de viande	€25
Salade de légumes	€10
Salade de fruits	€8
Salade de céréales	€6
Salade de noix	€4

+  
Ajouter  
un menu

Créé le 3 juin

Modifier

[Supprimer](#)

## Spécifications techniques

FRONT-END

react-router

BACK-END

Node.js  
express.js  
openAPI  
swagger  
MongoDB