

Compte-rendu SushiApp

Table of Contents

Contexte	1
Introduction.....	1
Maquettes.....	1
Récupération Api	5
Header.....	6
Footer.....	7
La page principale	8
Panier	9
Historique.....	11
RGPD	13
Problèmes rencontrés	14
Conclusion	14
Lien	14

Réalisé par :

- CHONG TOUA Joshua

Date début : 03/2024

Contexte

Projet de seconde année de BTS SIO, le but est de créer SushiApp qui est une application pour commander des sushis.

Introduction

Pour démarrer le projet, on a créé des maquettes pour pouvoir être guidé tout au long du projet.

Maquettes

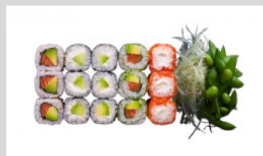


Boxes sushi :



Tasty blend

12.50 € +



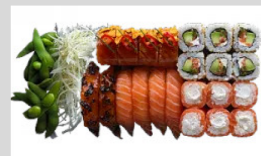
Amateur mix

15.90 € +



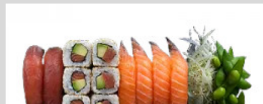
California Dream

19.90 € +



Fresh Mix

24.50 € +

Annuler la
commande

Panier

Validé : 37.50€



Tasty Blend

Détails



- x3 California Saumon avocat
- x3 Sushi Saumon
- x3 Spring Avocat Cheese
- x3 California Pacific
- x1 Edamame / Salade de chou

Tasty blend 12.50 € +

Retourner à la
commandeAjouter au
panier



Historique des commandes

Numéro commande	Produit	Date	Prix
567653211	tacos 3 sushi x2 yaourt sushi x1	59/24/-1500	37.50 cailloux
155345514	Salmon Classic x22222 Saumon Original x4	1/1/1939	48.50 franc
878796543	pizza sushi x2 tiramisu sushi x1	42/14/1330	37.50 écus
657564231	Kebab sushi x2 glace sushi x1	37/12/1945	27.50 €

[Quitter](#)Panier

Produit	Quantité	Prix
Tasty blend	— 2 +	25.00 €
Amateur mix	— 1 +	15.90 €

[Retour](#)

Total : 40.90 €

[Commander](#)



Numéro de la Commande :
78548525

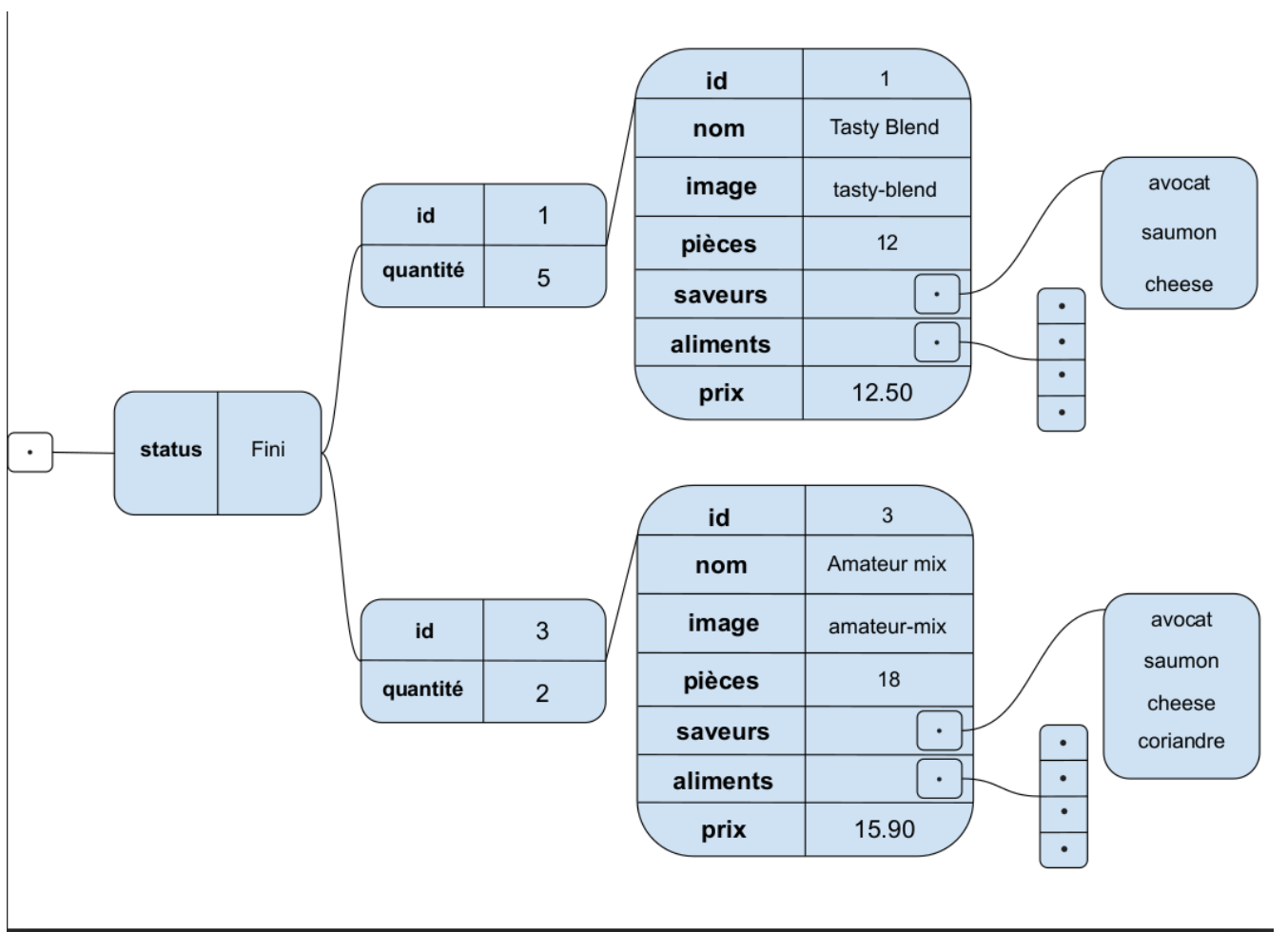
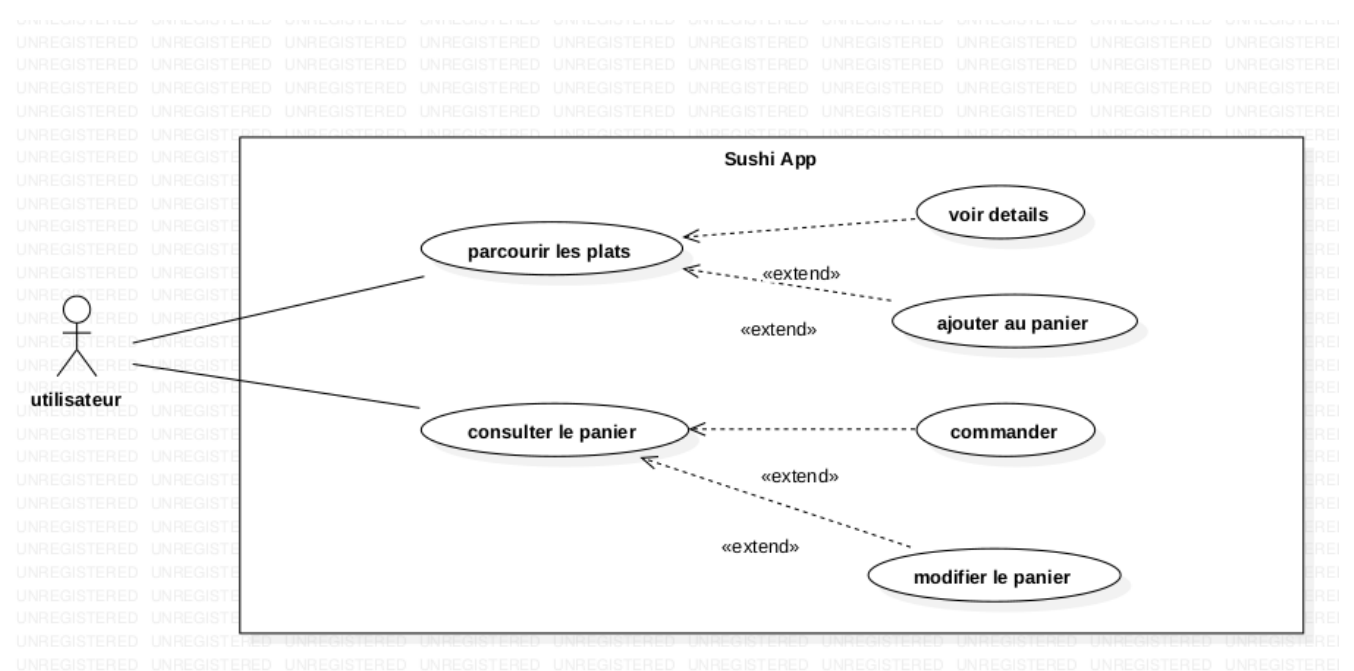


注文が検証されました
支払われた

(Merci de votre commande)



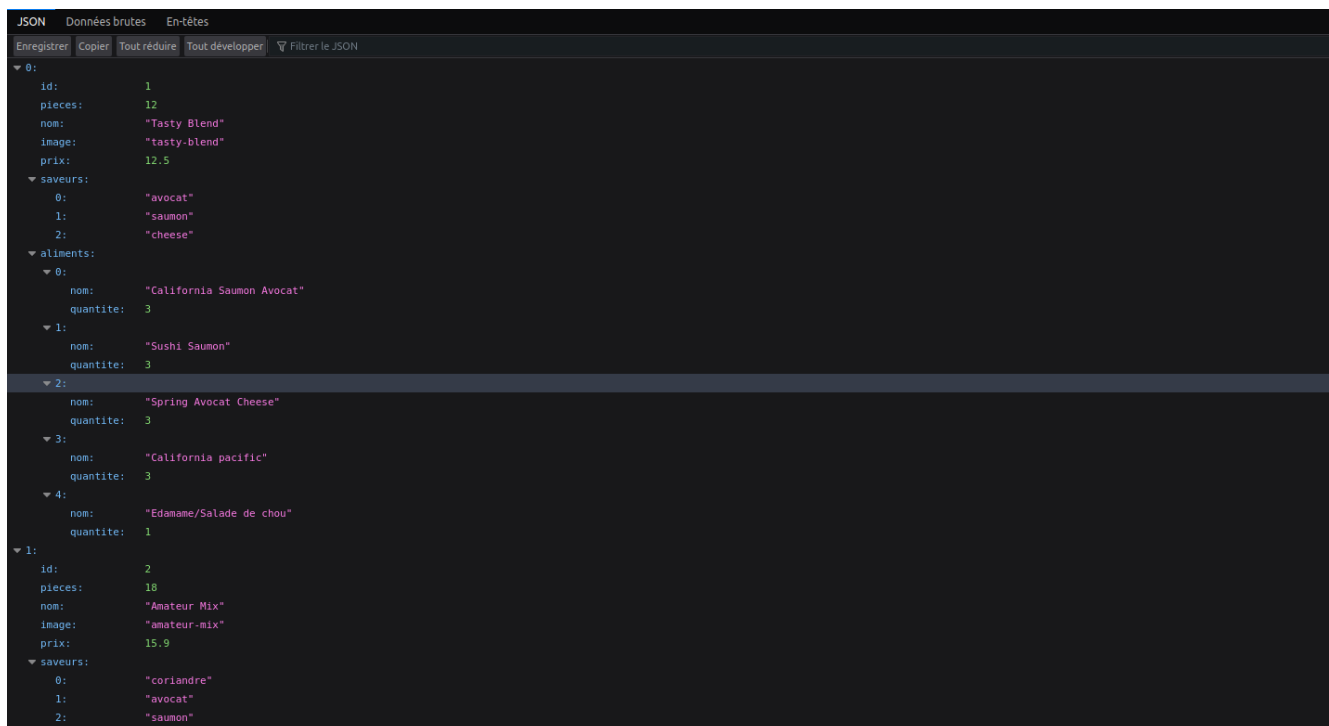
Règlement (UE) 2016/679 du Parlement européen
et du Conseil du 27 avril 2016, relatif à la protection
des personnes physiques à l'égard du traitement
des données à caractère personnel et à la libre
circulation de ces données, et abrogeant la
directive 95/46/CE (règlement général sur la
protection des données).



Récupération Api

Pour obtenir les données des boîtes de sushis, on a récupéré le dossier contenant l'api sushis.

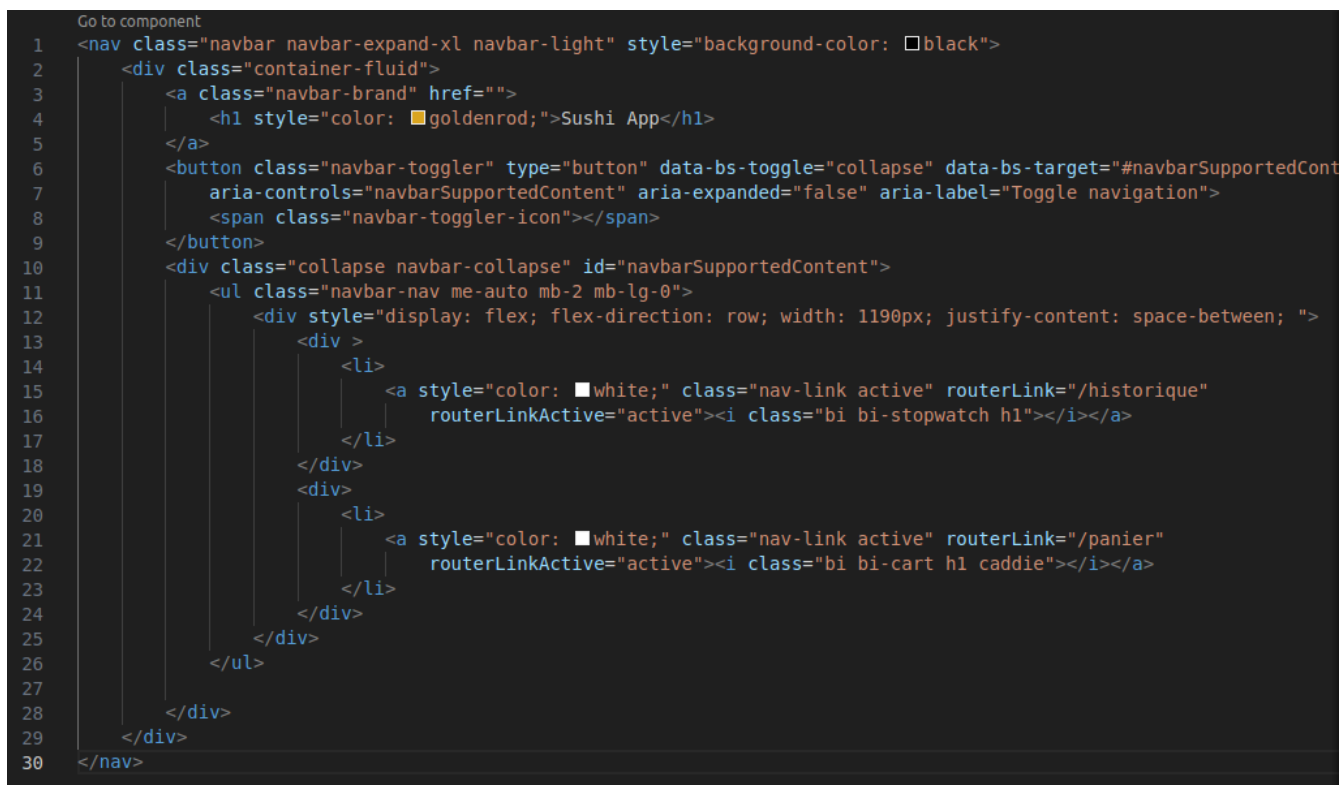
On obtient:



Header

Pour démarrer le projet, la première chose que j'ai créé était le header. Il contient la page qui nous amène à la sélection de sushi , le panier et l'historique des commandes.

Code HTML :



Rendu :

Footer

Le footer contient un bouton (rouge) pour réinitialiser le panier, un bouton rgpd qui nous affiche le texte rgpd et un bouton (vert) contenant le prix total de la commande et lors d'un click il nous mène sur la page panier.

```
src > app > component > Footer > Footer.component.html > Footer > a.nav-link.active
Go to component
1 <footer style="background-color: black">
2
3   <button class="btn btn-danger" style="width: 10%;" data-bs-toggle="modal" data-bs-target="#exampleModal">Annuler
4   | commande
5   </button>
6
7   <a style="color: white;" class="nav-link active" routerLink="/rgpd" routerLinkActive="active">
8   | <p class="rgpd">rgpd</p>
9   </a>
10
11   <button routerLink="/panier" routerLinkActive="active" class="btn btn-success" style="width: 10%;">Total:
12   | {{getTotal()| number: '1.2-2'}}</button>
13
14 </footer>
15
16
17 <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
18   <div class="modal-dialog">
19     <div class="modal-content">
20       <div class="modal-header">
21         <h1 class="modal-title fs-5" id="exampleModalLabel">Etes-vous sur d'annuler la commande?</h1>
22         <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
23       </div>
24       <div class="modal-footer">
25         <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Non</button>
26         <button type="button" class="btn btn-primary" data-bs-dismiss="modal" (click)="cancel()">Oui</button>
27       </div>
28     </div>
29   </div>
30 </div>
```

Le total de la commande est obtenu grâce à l'appel d'une fonction dans le fichier ts de footer qui va utiliser la méthode du service panier.

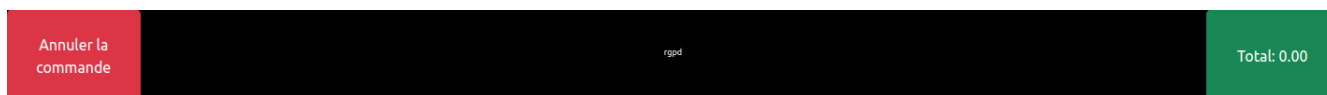
Fichier TypeScript :

```
// Méthode pour obtenir le total du panier
getTotal(): number {
  return this.getPanier.getTotal();
}
```

Fichier Service :

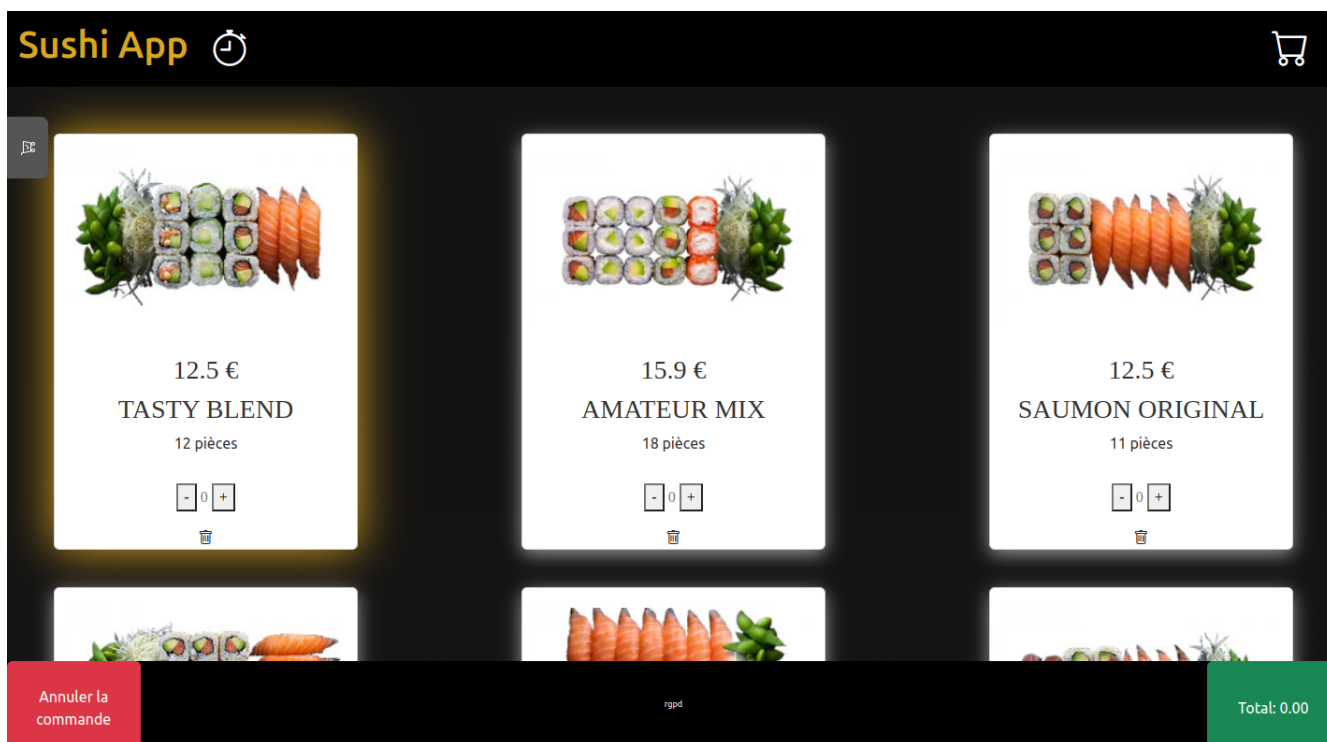
```
// Méthode pour obtenir le total de tous les éléments dans le panier
getTotal(): number {
  let total = 0;
  for (let elt of this.panier) {
    total += this.getTotalOnlyBox(elt.box.id);
  }
  return total;
}
```

Rendu :



La page principale

Sur la page principale, on y trouve toutes les boîtes de sushis présentent dans l'api.



Dans les boxes on peut y trouver:

- le prix de la boîte
- le nom de la boîte
- le nombre de pièces dans la boîte
- la quantité de boîte qu'on a ajouté dans le panier
- un bouton plus (+) pour ajouter dans le panier
- un bouton moins (-) pour retirer du panier qui se limite à la valeur 1
- un bouton poubelle pour supprimer la/les boîtes du panier



12.5 €

TASTY BLEND

12 pièces

- 3 +



Panier

On peut retrouver un panier sur la page principale en cliquant sur le bouton gris à gauche de la page, ou sur le chariot en haut à droite de la page ou sur le bouton total en bas à droite de la page.

Sushi App

Panier

Produits	Quantite	Prix	Total
Tasty Blend	- 3 +	12.5 €	37.50 €

Total: 37.50

[Valider Commande](#)

12.5 €

TASTY BLEND

12 pièces

- 3 +

15.9 €

AMATEUR MIX

18 pièces

- 0 +

12.5 €

SAUMON ORIGINAL

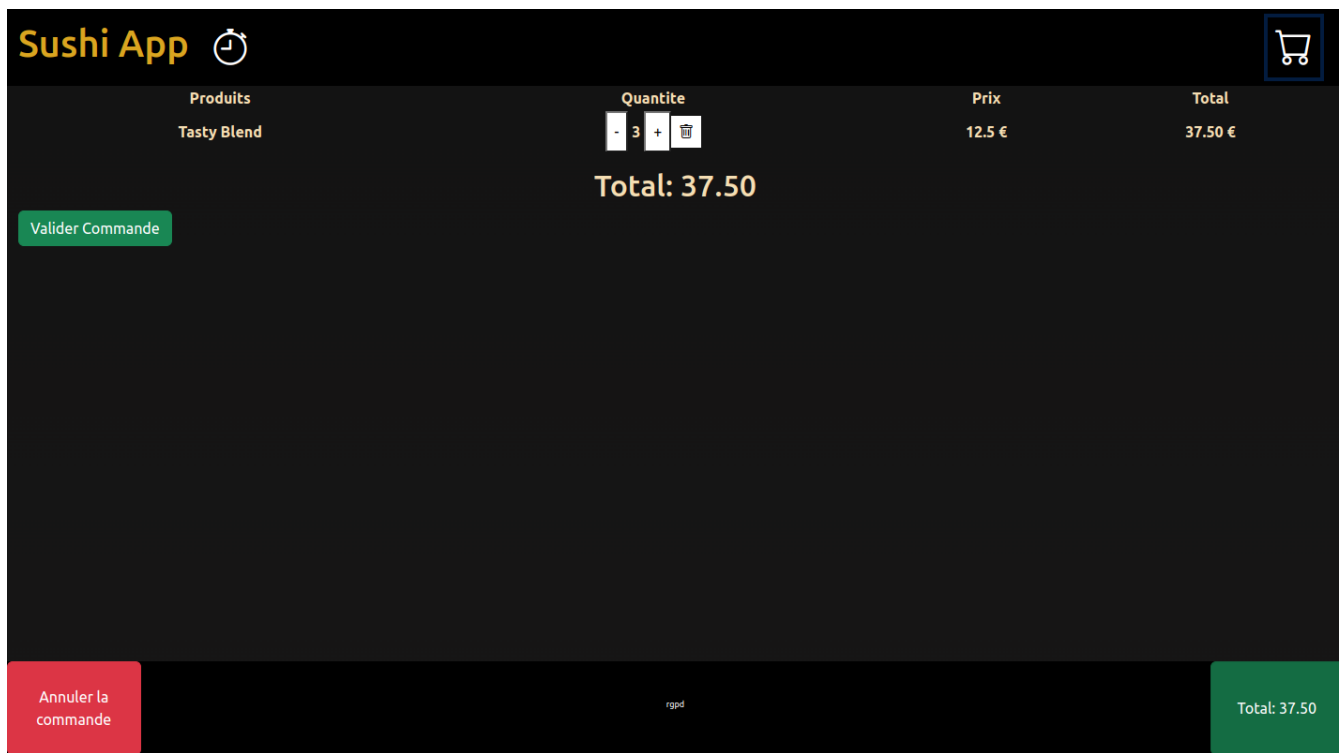
11 pièces

- 0 +

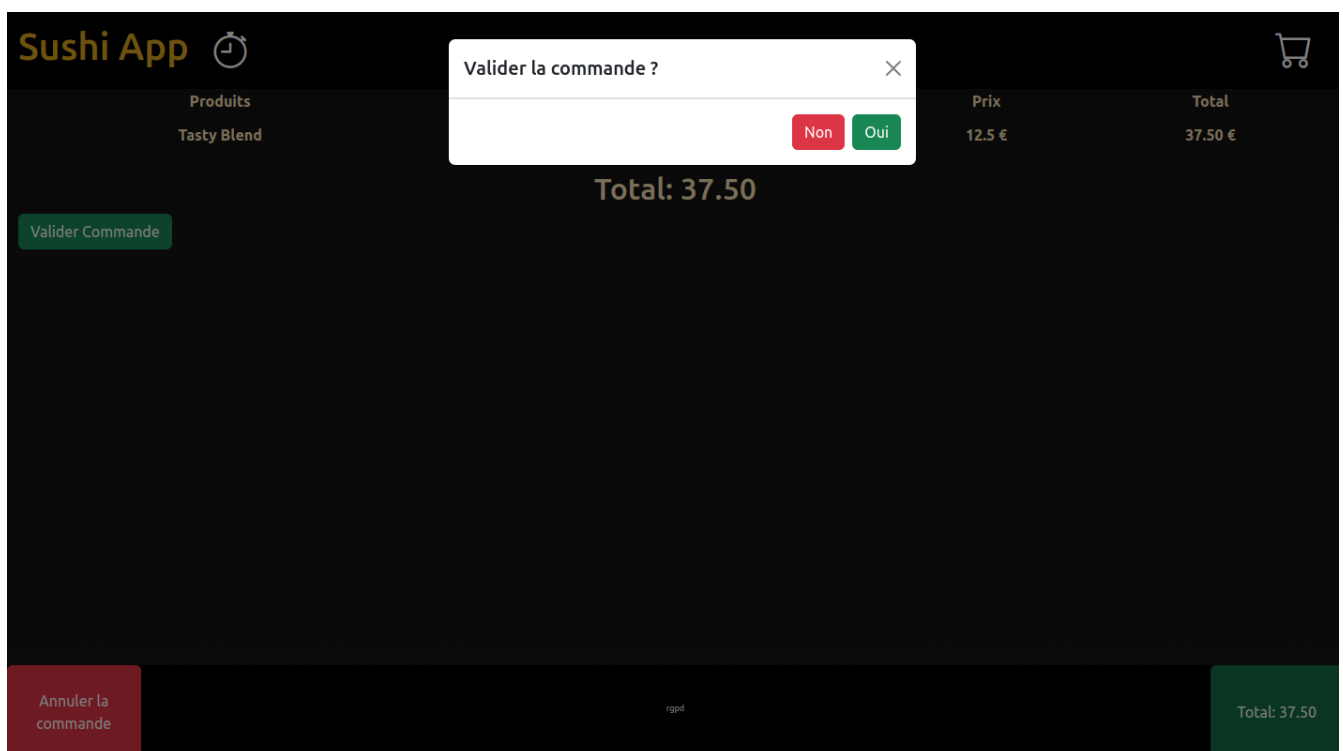
[Annuler la commande](#)

rgpd

Total: 37.50



En cliquant sur le bouton "Valider Commande", un modal nous demande si on est sur de valider la commande puis le numéro de commande est créé.



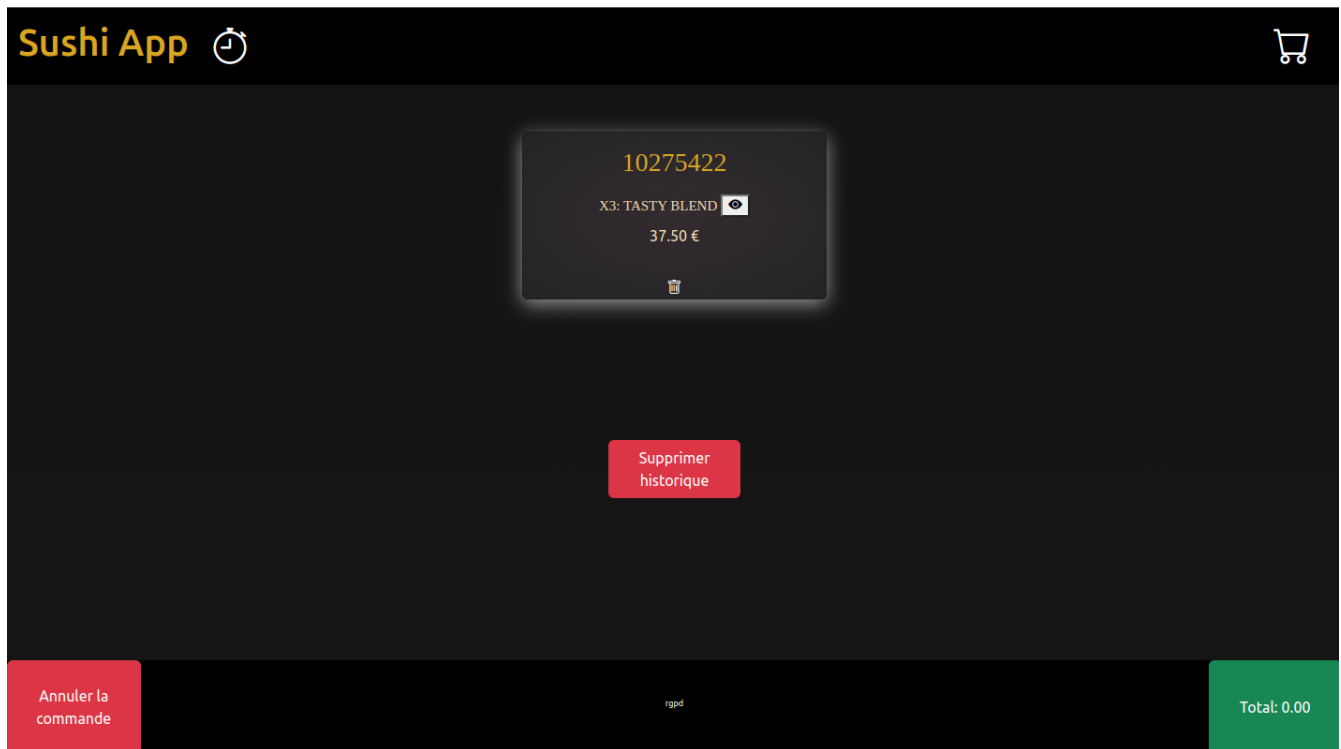
Code pour générer le numéro de commande :

```
// Méthode pour générer un nouvel identifiant de commande
nCommande() {
  let num: string = "";
  for (let i = 0; i < 7; i++) {
    // Génère un chiffre aléatoire entre 0 et 9
    let random = Math.floor(Math.random() * (9 - 0 + 1)) + 1;
    // Convertit le chiffre en chaîne de caractères et l'ajoute à l'identifiant de commande
    num += random.toString();
  }
  console.log(num); // Affiche l'identifiant de commande dans la console (facultatif)
  return num;
}
```

Historique

Pour accéder à l'historique, il suffit de cliquer sur le chronomètre dans le header.

A l'intérieur, on peut y trouver toute les commandes qui ont été passé.



Code HTML :

```

src > app > component > historique > historique.component.html > div.container-fluid > div.card-container-fluid > div > div.col-6.card > div.card-body >
Go to component
1 <div class="container-fluid">
2   <div *ngIf="historique" class="card-container-fluid">
3     <div *ngFor="let commande of historique">
4       <div class="col-6 card">
5         <div class="card-body">
6           <div class="card-price">
7             <h3 style="color: #goldenrod;">{{commande.idCommande}}</h3>
8           </div>
9           <div class="card-title">
10            <h3>
11              <span class="fond" style="color: #wheat; font-size: 15px;" *ngFor="let elt of commande">
12                <button data-bs-toggle="modal" data-bs-target="#plat">
13                  <i class="bi bi-eye-fill"></i>
14                </button>
15              <br>
16              <div class="modal fade" id="plat" tabindex="-1" aria-labelledby="platLabel" aria-hi
17                <div class="modal-dialog">
18                  <div class="modal-content">
19                    <div class="modal-body">
20                      <h1 class="modal-title fs-5" id="platLabel">{{elt.box.name}}</h1>
21                      
24                  </div>
25                </div>
26              </div>
27            </span>
28          </h3>
29        </div>
30      </div>
31    </div>
32  </div>

```

Code TypeScript :

```

constructor(private lookupBoxesService: LookupBoxesService, private getPanier: AddPanierService, private commande
// Initialise l'ID de la commande en utilisant le service CommandeService
this.idCommande = this.commandeService.getId();
}

ngOnInit(): void {
// Initialise l'historique des commandes en récupérant les données du service CommandeService
this.historique = this.commandeService.getHistorique();

// Souscrit aux changements de l'historique des commandes pour mettre à jour les données
this.commandeService.majHistorique.subscribe((result: undefined) => {
  this.historique = this.commandeService.getHistorique();
});
}

// Ajoute une commande à l'historique
addHistorique() {
  this.commandeService.addHistorique(this.etat, this.support, this.total);
}

// Supprime une ligne de commande de l'historique
deleteLigne(id: string) {
  this.commandeService.deleteLigne(id);
  this.historique = this.commandeService.getHistorique();
}

// Supprime l'historique complet des commandes
deleteHistorique() {
  this.commandeService.cancel();
}

```

Code du Service :

```

// Méthode pour ajouter une commande à l'historique
addHistorique(etat: boolean, support: Array<Support>, total: number) {
  // Crée une nouvelle commande
  let commande = new Commande(this.idCommande, etat, support, total);
  // Ajoute la commande à l'historique
  this.historique.push(commande);
  // Met à jour les données dans le stockage local
  localStorage.setItem("commande", JSON.stringify(this.historique));
  // Génère un nouvel identifiant de commande pour la prochaine commande
  this.idCommande = this.nCommande();
}

// Méthode pour obtenir l'historique des commandes
getHistorique() {
  return this.historique;
}

// Méthode pour obtenir l'identifiant de la commande actuelle
getId() {
  return this.idCommande;
}

// Méthode pour annuler l'historique des commandes
cancel() {
  // Réinitialise l'historique des commandes
  this.historique = [];
  // Supprime les données de l'historique des commandes du stockage local
  localStorage.removeItem("commande");
  // Émet un événement pour notifier les composants de la modification de l'historique des commandes
  this.majHistorique.emit();
}

// Méthode pour supprimer une ligne de commande de l'historique
deleteLigne(id: string) {
  let nouveauHistorique: Array<Support> = [];
  for (let elt of this.historique) {
    if (elt.idCommande !== id) {

```

RGPD

Pour accéder à la page RGPD, il suffit de cliquer sur le "rgpd" dans le footer.

Sur la page, on y trouve le texte RGPD.

Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016, relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE (règlement général sur la protection des données).

Annuler la commande

rgpd

Total: 0.00

Problèmes rencontrés

Tout au long de ce projet, j'ai pu rencontrer des problèmes :

- La compréhension du fonctionnement d'Angular
- Le fonctionnement des Services avec les fichiers TypeScript des composants
- L'utilisation du localStorage.

Conclusion

Pour conclure, ce projet a apporté des points positifs à ma formation car il m'a permis de découvrir un nouveau langage (TypeScript, Angular) et de m'améliorer en front-end.

Lien

Lien github: <https://github.com/JoshuaChongToua/SushiApp>