Compte rendu : *Projet SushiApp*

Tables des matières

Projet réalisé par :
SONG Steeven

Du 12/02/2024 au 30/04/2024

Contexte :

```
Dans le cadre de notre projet SushiApp, l'objectif était de nous permettre d'apprendre
Javascript et d'utiliser une API pour réaliser un site de commande à Sushi.
```

Introduction :

```
Dans un premier temps, nous avons realisé une maquette avec des consignes précis pour
nous donner un apercu du site dans lequel nous allons réaliser et après avoir réaliser
la maquette, nous avons mis en place le travail sur le site.
```

IMPORTANT | Lien Github :

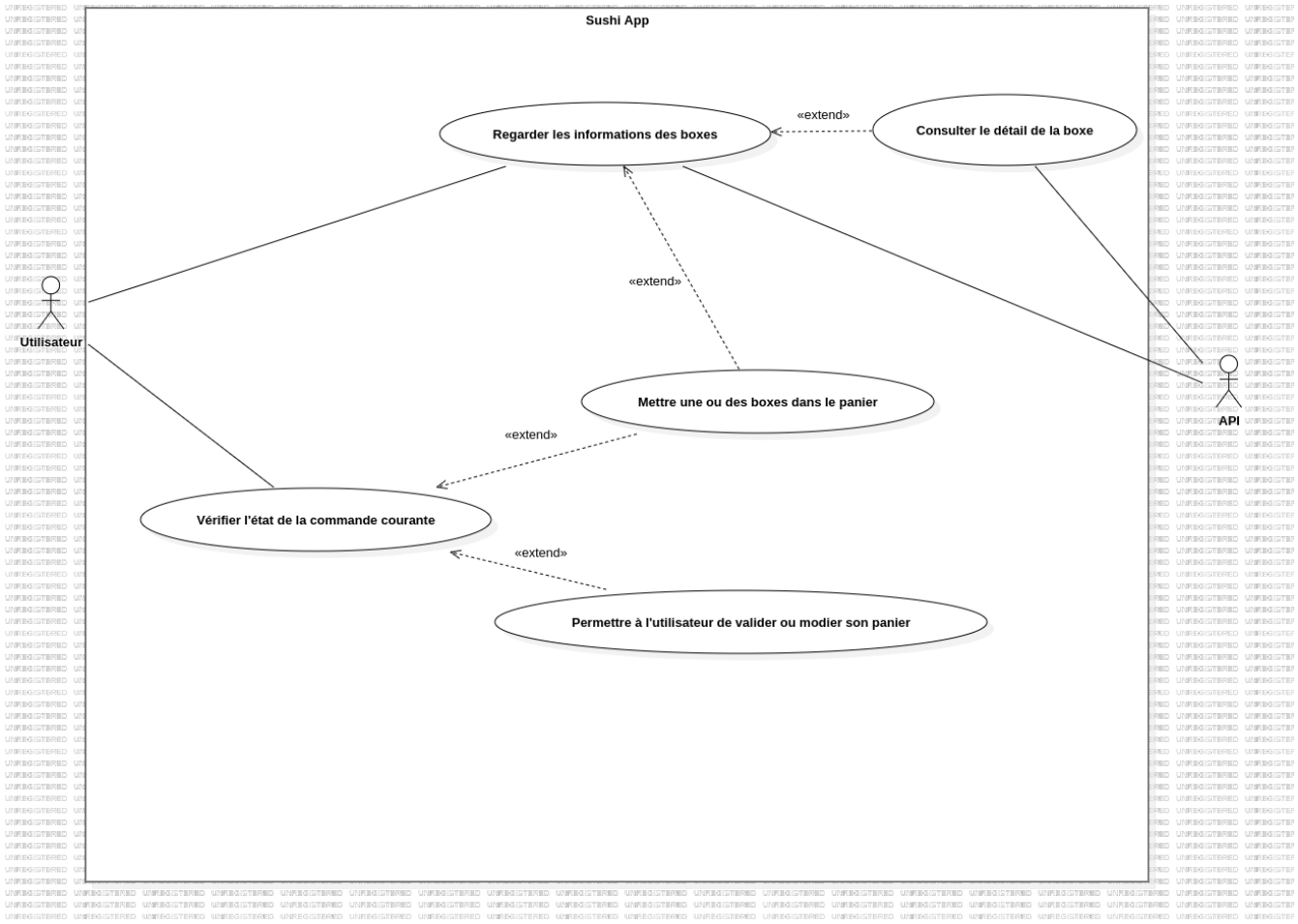
I. Réalisation des premieres tâches

a) Maquette

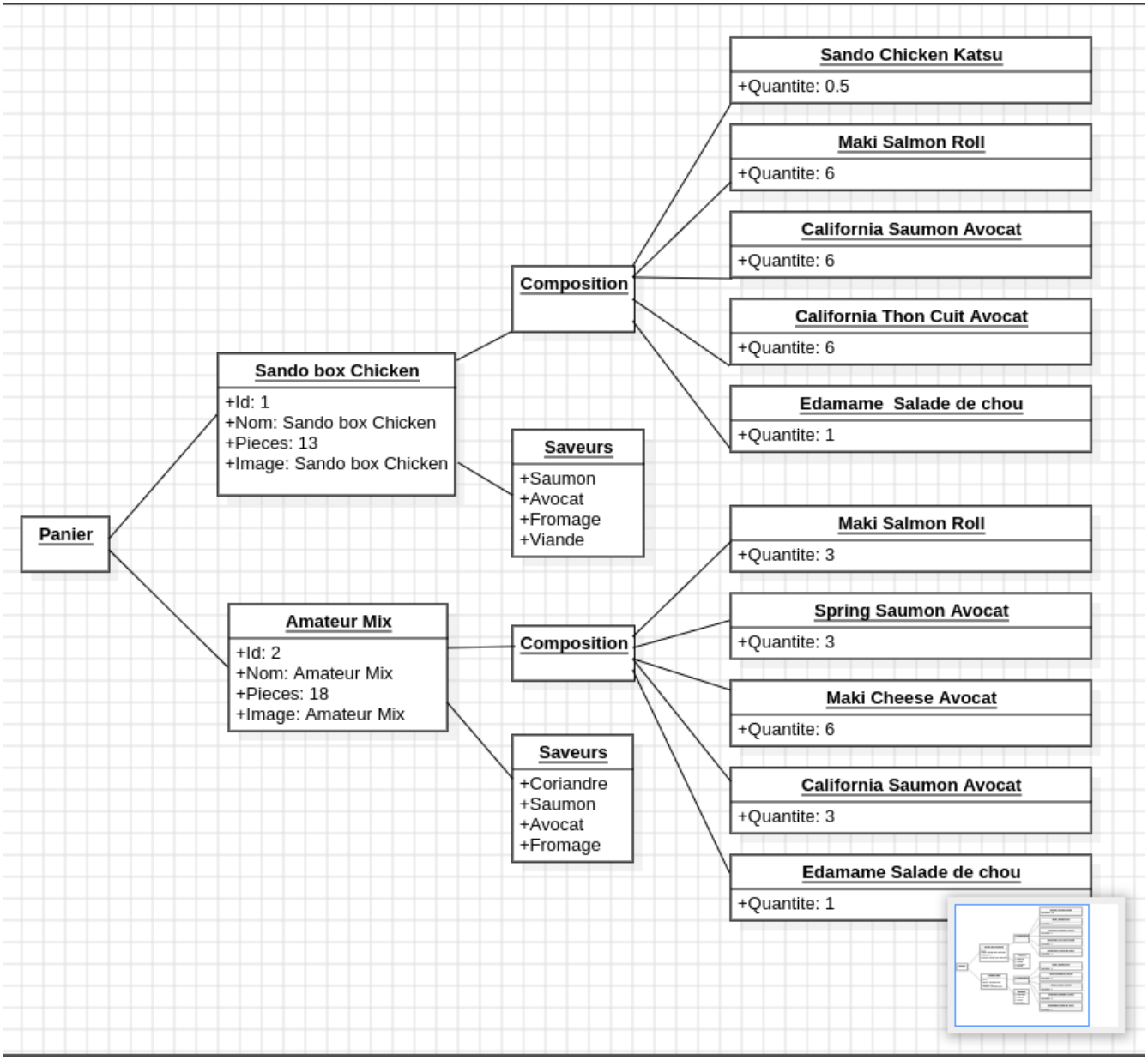
Dans un premier temps, nous avons réalisé une maquette sur Figma, cette maquette nous sert de visualisation pour voir comment nous allons procéder pour réaliser le site.

IMPORTANT	Lien Maquette sur Figma : https://www.figma.com/file/eApxfuprNF8OU5HlIO43KR/Untitled?type=design&node-id=6%3A2&mode=design&t=5zl2LdaHbOhCKr6Y-1

b) Cas utilisateur



c) Diagramme Json



I. Réalisation du site

I.b : Installation du service

Avant de commencer à faire notre premiere page de code, nous avons tout d'abord crée un service, ce service va nous permettre d'instancier nos box et de les utiliser par la suite dans notre site.

```
export class SushiBoxService {
constructor(private http:HttpClient) { }
public Box(): Observable<any> {
let resultat =this.http.get(environment.apiBaseUrl);
console.log(resultat)
return resultat;
}
```

a) La carte

Dans un premier temps, j'ai réalisé la première page du site. C'est la page principale ou nous allons trouver toutes les boxes à Sushi et c'est aussi la page ou nous allons pouvoir séléctionner nos boxes pour l'ajouter dans notre panier.

```
   <!-- Modal -->
            <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby=
"exampleModalLabel"
                 aria-hidden="true">
                 <div class="modal-dialog">
                     <div class="modal-content">
                         <div class="modal-header">
                             <h1 class="modal-title fs-5" id="exampleModalLabel"
>Details de {{detailBox?.nom}} </h1>
                             <button type="button" class="btn-close" data-bs-dismiss=
"modal" aria-label="Close"></button>
                         </div>
                         <div class="modal-body">
                             <img [src]="'http://localhost:8080/api/image/' +
detailBox?.image" class="card-img-top" alt="">
                             <h1 class="modal-title fs-5" id="exampleModalLabel"
>Aliments</h1>
                             <p *ngFor="let aliment of detailBox?.aliments ?? [] "
>{{aliment.nom }}</p>
                             <h1 class="modal-title fs-5" id="exampleModalLabel">
Saveur</h1>
                             <p>{{detailBox?.saveurs}}</p>
                         </div>
                     </div>
                 </div>
             </div>
```

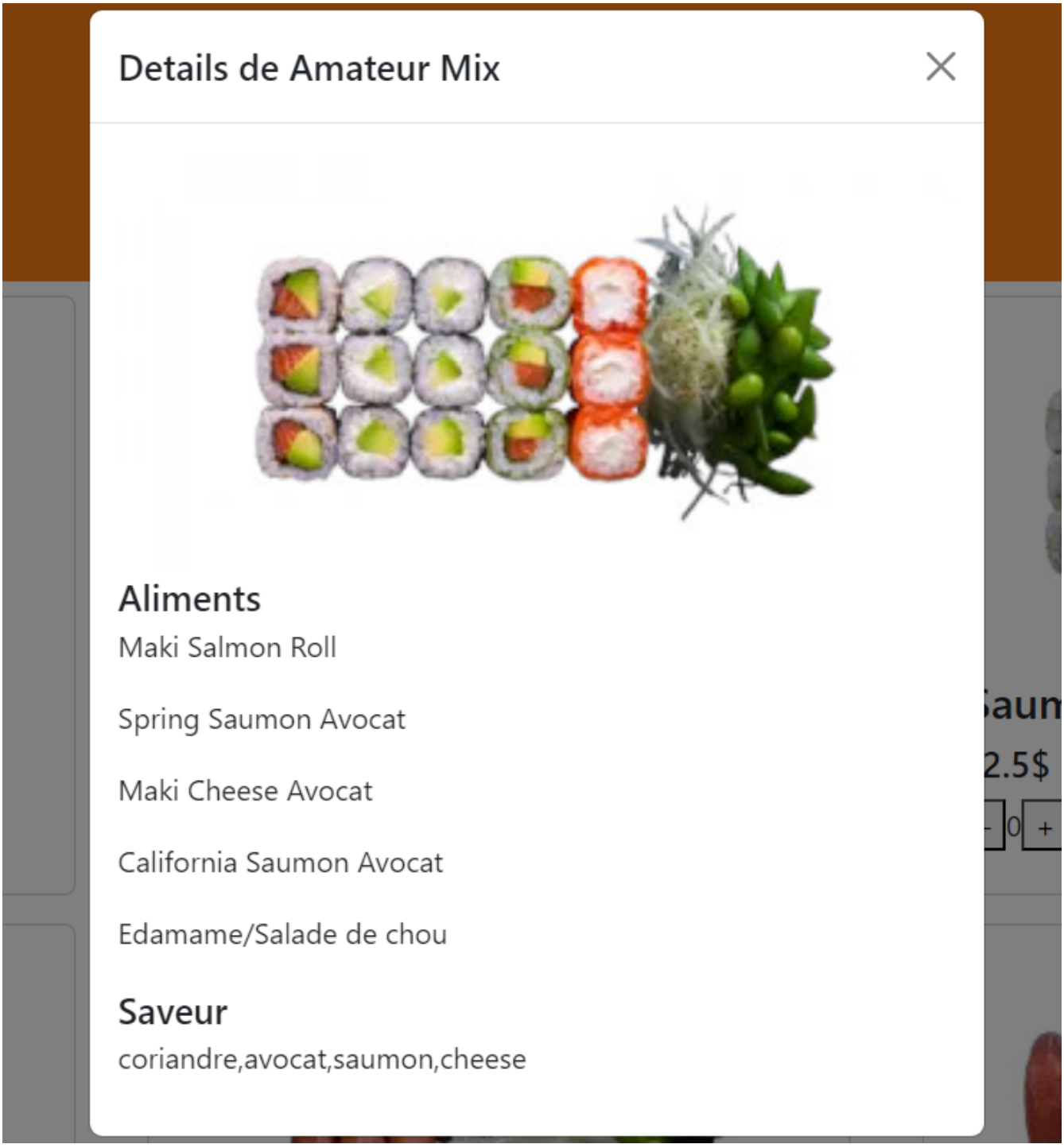
Dans ce code j'ai utilisé un modal issu de Bootstrap dans laquelle j'ai implémenter mes méthodes que j'ai crée pour importer les détails de mes boxes.

```
export class CarteComponent {
  boxes:Array<Box>
  detailBox:Box | null
  numero: number = 0;
  constructor(private sushiBoxService: SushiBoxService, private panierService:
ManagerPanierService){
    this.detailBox=null
    this.getBoxes();
    this.boxes=[]
  }
getBoxes(): void {
this.sushiBoxService.Box().subscribe((res: any)=>{
```

```
    this.boxes=res
    console.log(this.boxes);
  })
}
onDetails(laBox:Box){
this.detailBox=laBox
console.log(this.detailBox)
}
```

Dans ce code j'ai crée 2 méthodes, getboxes() et onDetails() pour pouvoir comme je l'ai dis précédent importer les détails des box.

Resulat



II. Le panier

II.b : Installation du service pour le panier & page confidentialité

Idem avec la carte. Nous allons crée un service pour importer ce dont nous avons besoin dans notre page panier.

```
export class ManagerPanierService {
  lignes: Array<LignePanier>
  constructor() {
    this.lignes = JSON.parse(localStorage.getItem("panier") ?? "[]")
  }
  getPanier() {
    return this.lignes
  }
  add(uneBox: Box, quantite: number) {
    let ligne = new LignePanier(quantite,uneBox)
    let boxExistante = false;
    for (let boxe of this.lignes){
    if (boxe.uneBox.id == ligne.uneBox.id){
      boxe.quantite+=quantite
      boxExistante = true
    }
  }
  if (boxExistante == false){
   this.lignes.push(ligne)
  }
  localStorage.setItem("panier",JSON.stringify(this.lignes))
}
  remove(uneBox: Box, quantite: number){
  for (let i = 0 ; i <  this.lignes.length; i++){
  if (this.lignes[i].uneBox.id === uneBox.id){
    if (this.lignes[i].quantite > quantite){
      this.lignes[i].quantite -= quantite;
    }else{
      this.lignes.splice(i,1);
    }
    localStorage.setItem("panier",JSON.stringify(this.lignes))
    return
  }
}
localStorage.setItem("panier",JSON.stringify(this.lignes))
}
clearPanier() {
localStorage.clear();
this.lignes = [];
```

```
}
}
```

Dans ce code, j'ai ajouté des méthodes pour ajouter , retirer et récuperer la box.

a) Le panier

A la différence de la page carte, nous avons utilisé des codes HTML/CSS pour faire notre page panier et nous avons importé les boxes qui auront été selectionner dans notre page panier.

```
<body>
    <div class="container">
        <h1>Panier</h1>
        <div *ngFor="let ligne of lignesPanier" class="panier-ligne">
            <div class="nom-box">{{ ligne.uneBox.nom }}</div>
            <div class="quantite">
                Quantité:
                <a><button (click)="onDelete(ligne.uneBox)">-</button></a>
                <span>{{ ligne.quantite }}</span>
                <a><button (click)="onAjout(ligne.uneBox)">+</button></a>
            </div>
            <div class="prix">Prix : {{ ligne.uneBox.prix * ligne.quantite }}$</div>
        </div>
        <div class="total">Total: {{ calculerPrixTotal() }}$</div>
         <button class="valider-panier" (click)="validerPanier()">Valider
Panier</button>
        <button class="vider-panier" (click)="clearPanier()">Vider le panier</button>
    </div>
</body>
```

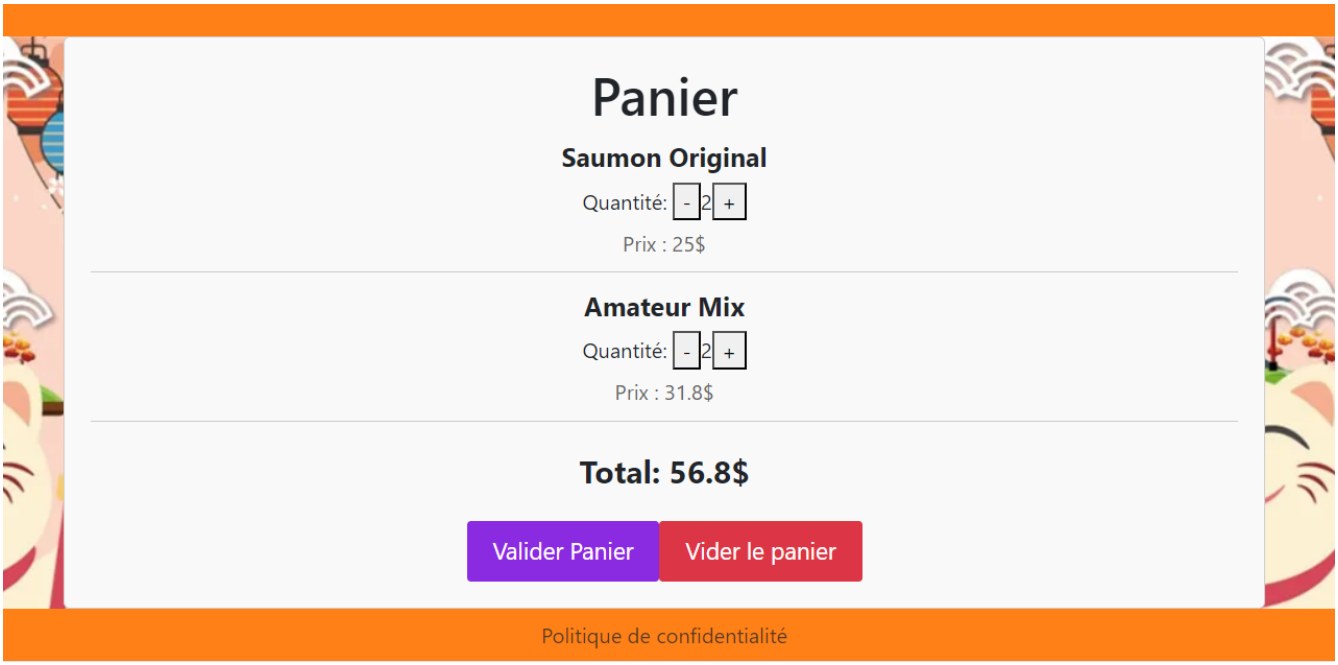
```
export class PanierComponent{
  @Input()
  box:Box | undefined
  lignesPanier:LignePanier[]=[];
  isModalVisible: boolean = false;

  constructor(private panierService: ManagerPanierService){}
    ngOnInit(): void {
      this.lignesPanier = this.panierService.getPanier();
    }
    clearPanier() {
      this.panierService.clearPanier();
      this.lignesPanier = [];
    }
    onAjout(uneBox: Box) {
      this.panierService.add(uneBox, 1);
    }
    onDelete(uneBox: Box) {
```

```
  this.panierService.remove(uneBox,1)
    }
    calculerPrixTotal(): number {
      let prixTotal = 0;
      for (const ligne of this.lignesPanier) {
        prixTotal += ligne.uneBox.prix * ligne.quantite;
      }
      return prixTotal;
    }
```

Dans ce code, nous avons ajouté plusieurs méthodes dont des méthodes pour clear le panier, ajouter des boxes directement dans le panier et calculer le prix total du panier.

Résultat



b) Page de confidentialité

Résultat

Nous avons fais une page de confidentialité simple avec HTML & CSS. Cette page de confidentialité est relié a tous les autres page.

III. Conclusion

Pour en conclure, le projet nous a permis d'expérimenté et acquérir des connaissance sur javascript et de pouvoir utiliser une API pour réaliser de nouvelles choses.