DEVOIR DEVELOPPEMENT WEB: ANGULAR

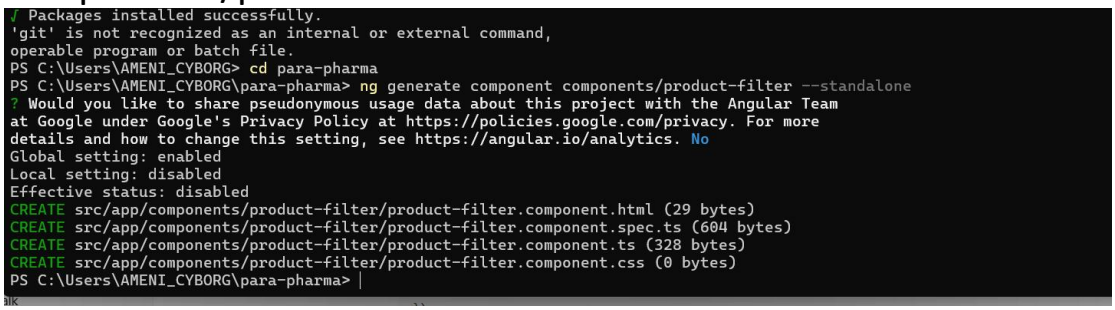
Realisé par Ameni Lahouar.

Partie 1:

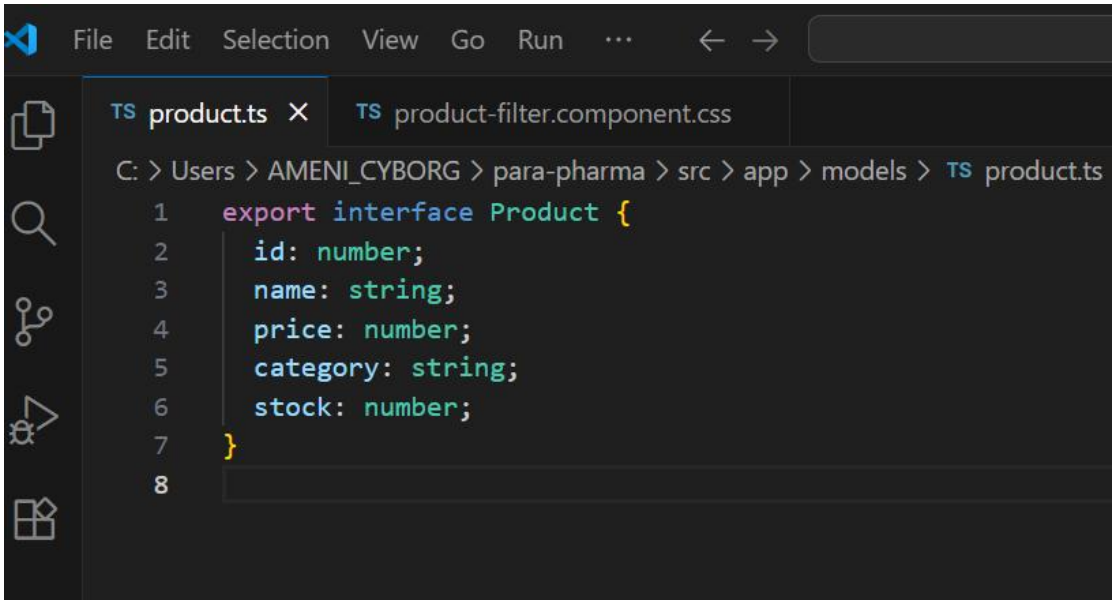
On a fait les commandes: ng new para-pharma --standalone
cd para-pharma

Partie 2:

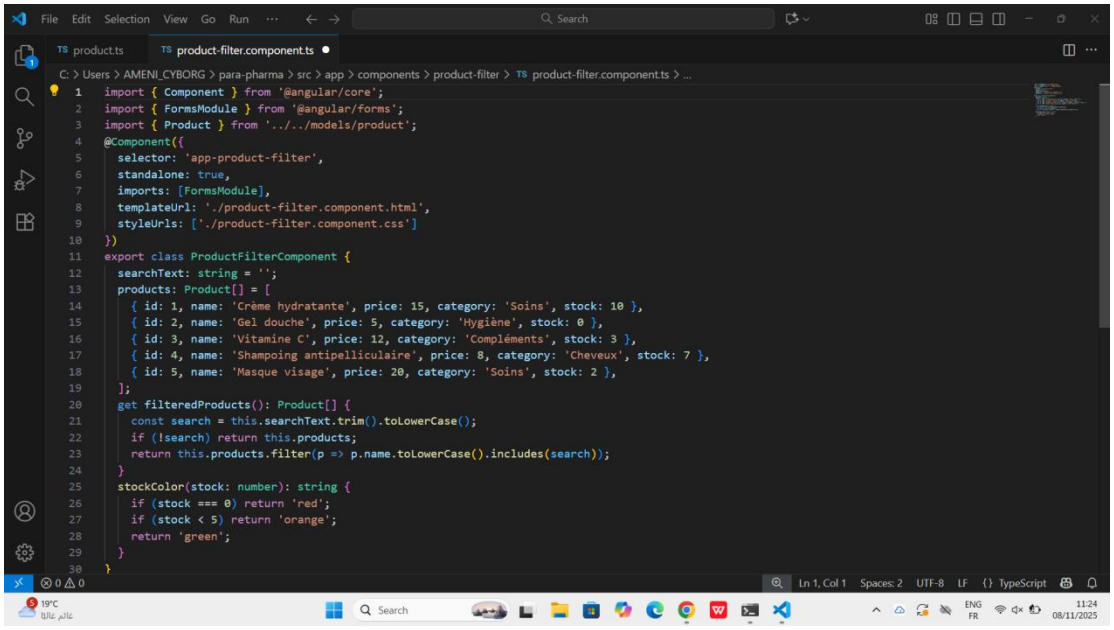
On a fait la commande:ng generate component components/product-filter --standalone



Partie3:

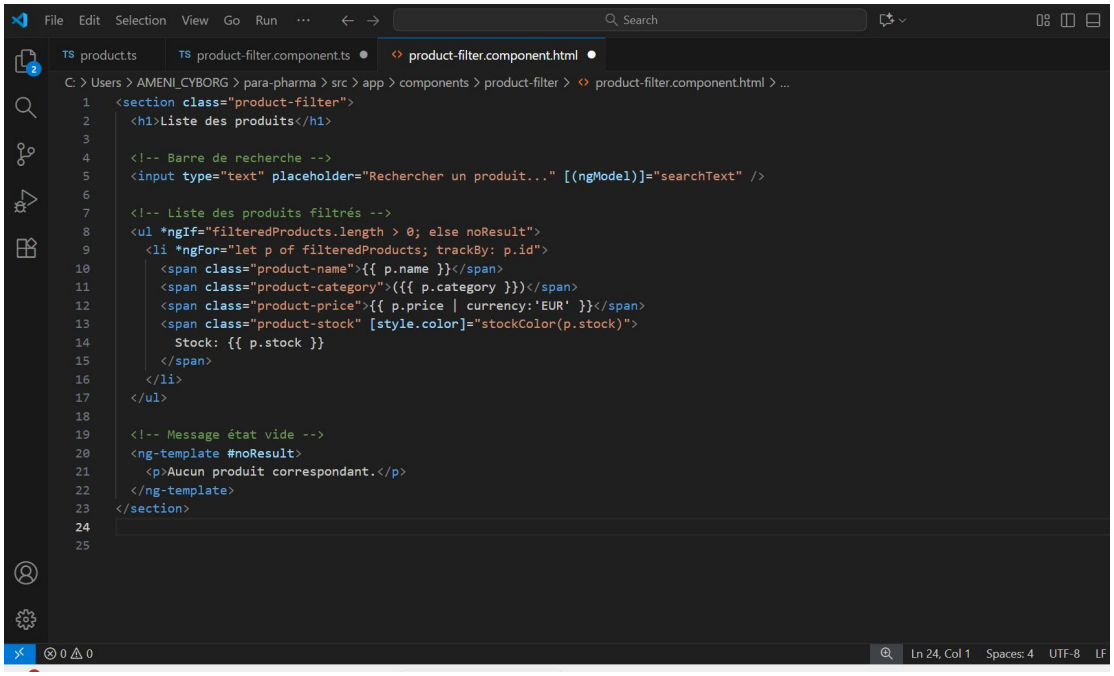


Partie4:



```ts
import { Component } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { Product } from '../../models/product';
@Component({
  selector: 'app-product-filter',
  standalone: true,
  imports: [FormsModule],
  templateUrl: './product-filter.component.html',
  styleUrls: ['./product-filter.component.css']
})
export class ProductFilterComponent {
  searchText: string = '';
  products: Product[] = [
    { id: 1, name: 'Crème hydratante', price: 15, category: 'Soins', stock: 10 },
    { id: 2, name: 'Gel douche', price: 5, category: 'Hygiène', stock: 0 },
    { id: 3, name: 'Vitamine C', price: 12, category: 'Compléments', stock: 3 },
    { id: 4, name: 'Shampoing antipelliculaire', price: 8, category: 'Cheveux', stock: 7 },
    { id: 5, name: 'Masque visage', price: 20, category: 'Soins', stock: 2 },
  ];
  get filteredProducts(): Product[] {
    const search = this.searchText.trim().toLowerCase();
    if (!search) return this.products;
    return this.products.filter(p => p.name.toLowerCase().includes(search));
  }
  stockColor(stock: number): string {
    if (stock === 0) return 'red';
    if (stock < 5) return 'orange';
    return 'green';
  }
}
```

Partie5:



```ts
<section class="product-filter">
  <h1>Liste des produits</h1>

  <!-- Barre de recherche -->
  <input type="text" placeholder="Rechercher un produit..." [(ngModel)]="searchText" />

  <!-- Liste des produits filtrés -->
  <ul *ngIf="filteredProducts.length > 0; else noResult">
    <li *ngFor="let p of filteredProducts; trackBy: p.id">
      <span class="product-name">{{ p.name }}</span>
      <span class="product-category">({{ p.category }})</span>
      <span class="product-price">{{ p.price | currency:'EUR' }}</span>
      <span class="product-stock" [style.color]="stockColor(p.stock)">
        Stock: {{ p.stock }}
      </span>
    </li>
  </ul>

  <!-- Message état vide -->
  <ng-template #noResult>
    <p>Aucun produit correspondant.</p>
  </ng-template>
</section>
```

Partie6:

```ts
  ];
  get filteredProducts(): Product[] {
    const search = this.searchText.trim().toLowerCase();
    if (!search) return this.products;
    return this.products.filter(p => p.name.toLowerCase().includes(search));
  }
  stockColor(stock: number): string {
```

Partie7:

La partie VII concerne l'affichage des produits filtrés avec des **directives conditionnelles et de boucle**.

Utiliser `*ngIf` (ou `@if` en Angular 20) pour afficher soit la liste, soit le message "aucun résultat".

Utiliser `*ngFor` (ou `@for`) pour parcourir `filteredProducts`.

Ajouter `trackBy: p.id` pour optimiser le rendu et éviter de recréer les éléments inutilement.

Colorer dynamiquement le stock avec `[style.color]="stockColor(p.stock)"`.

Résultat attendu :

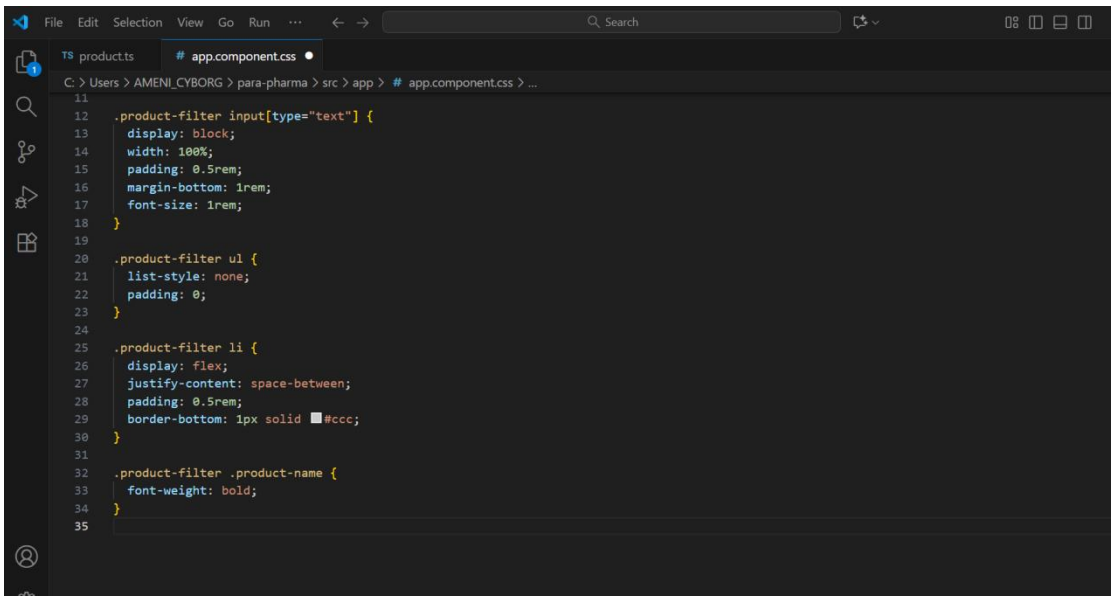
La liste des produits se met à jour en direct lors de la saisie dans le champ de recherche.

Si aucun produit ne correspond → affichage du message "Aucun produit correspondant".

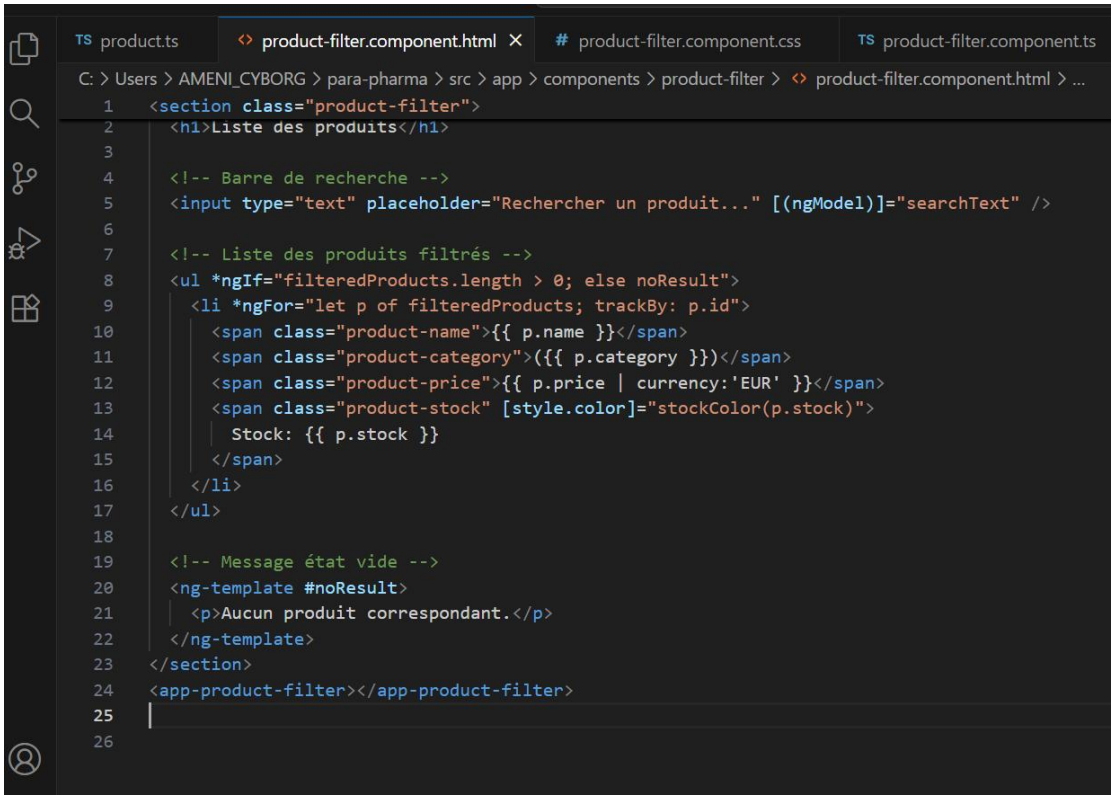
Les couleurs du stock s'appliquent automatiquement selon la quantité.

Il est aussi possible de mettre à jour le code pour utiliser la **nouvelle syntaxe Angular 20** `@if`/`@for`.

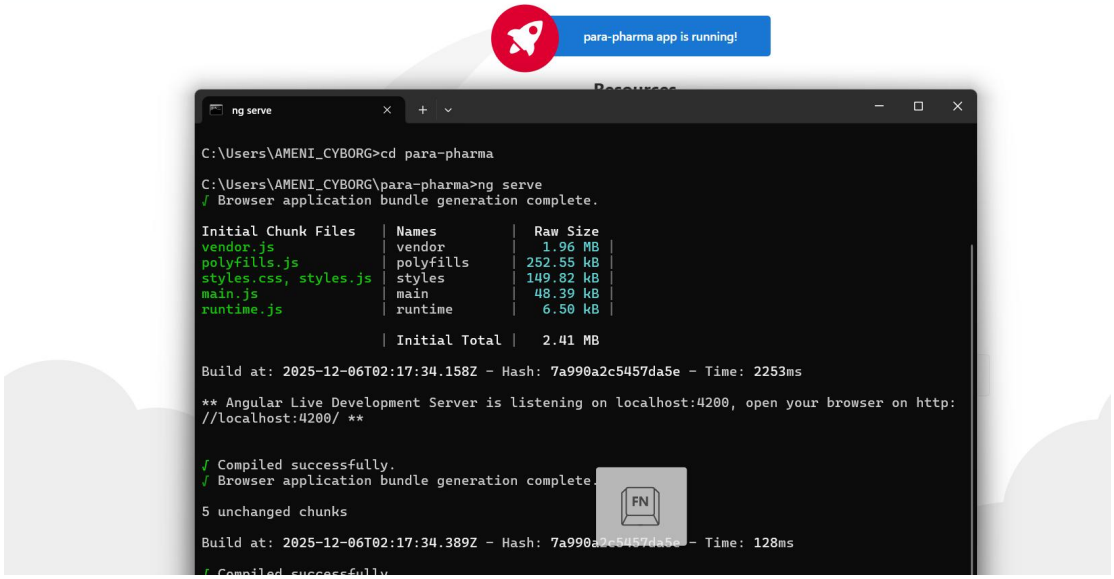
Partie8:



Partie9:



on a ajouté la derniere ligne

Partie10:

On execute maintenant et ouvre le site web: