ANGULAR ASSIGNMENTS

Code

<img [src]="imageUrl" alt="Flower" class="small-image">

<br />

<h1>Name: {{name}}</h1>

<h2>Price: {{price}}</h2>

export class Assignment1Component {

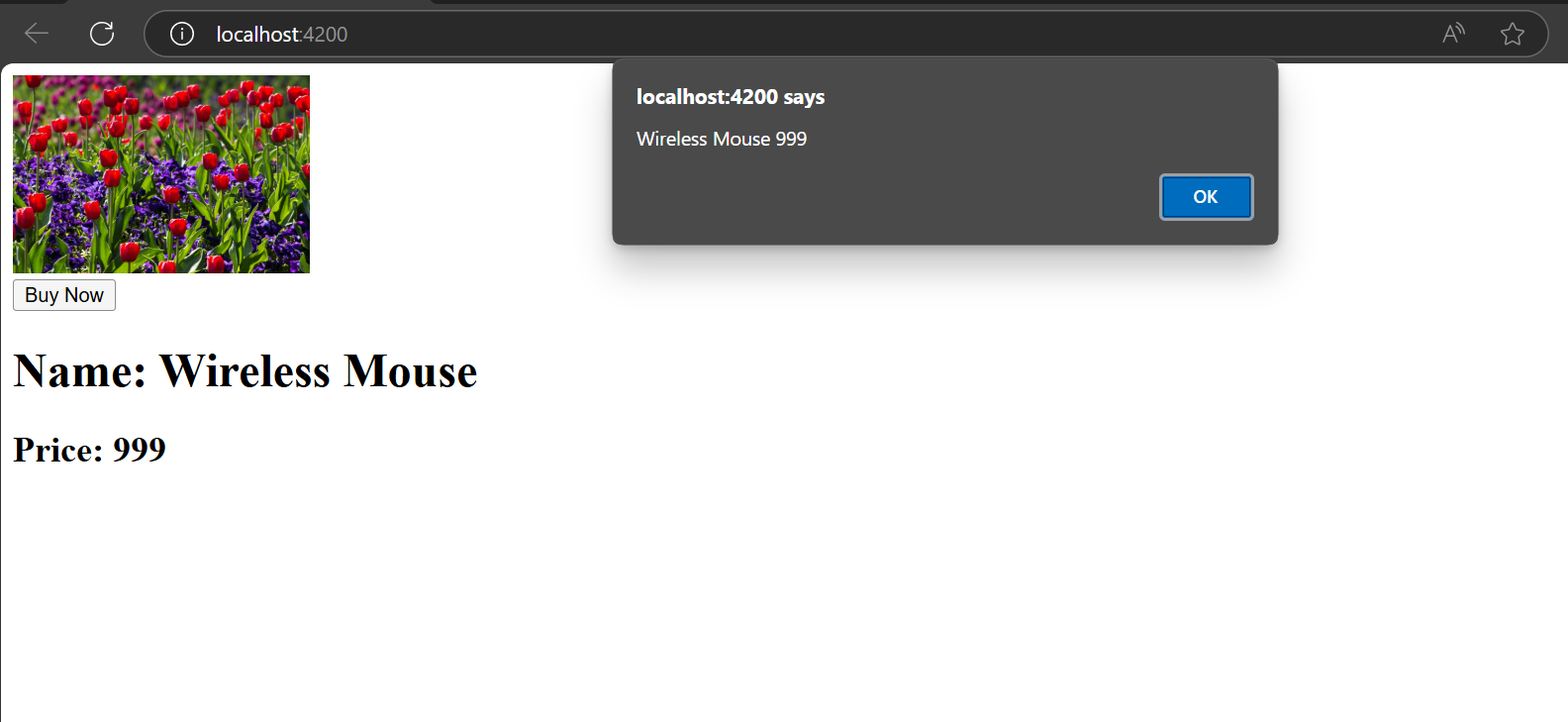
  name:string= 'Wireless Mouse';

  imageUrl= './assets/flowers.jpg';

  price:number= 999;

}

Output:-



2. Code:-

<div \*ngIf="userRole==='admin'">

    Admin LoggedIn

</div>

  <div [ngSwitch]="userRole">

    <p \*ngSwitchCase="'admin'">Admin</p>

    <p \*ngSwitchCase="'guest'">Guest</p>

    <p \*ngSwitchCase="'user'">User</p>

    <p \*ngSwitchDefault>Unknown Role</p>

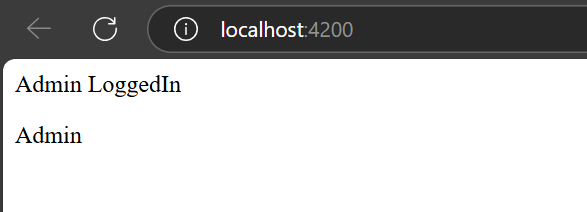
  </div>

export class Assignment2Component {

  userRole: 'admin' | 'guest' | 'user' = 'admin';

}

Output:-



3. Code:-

<input placeholder="Enter Your feedback" [(ngModel)]="feedback" />

<div>{{feedback}}</div>

import { Component } from '@angular/core';

@Component({

  selector: 'app-assignment-3',

  templateUrl: './assignment-3.component.html',

  styleUrls: ['./assignment-3.component.css']

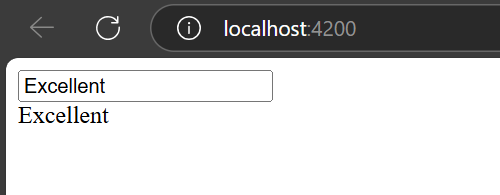
})

export class Assignment3Component {

  feedback:string='';

}

Output:-



4. Code

<div \*ngFor="let product of products">

    {{product.name}}

    {{product.price}}

</div>

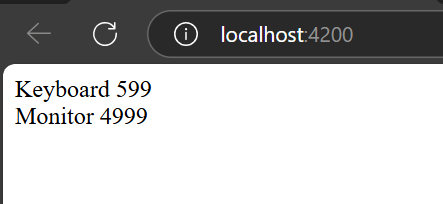
 products = [

    { name: 'Keyboard', price: 599 },

    { name: 'Monitor', price: 4999 }

    ];

Output:-



5.

import { Pipe, PipeTransform } from '@angular/core';

@Pipe({

  name: 'priceFormat'

})

export class PriceFormatPipe implements PipeTransform {

  transform(value: number, ...args: unknown[]): unknown {

    if(args.length > 0 && args[0] === 'symbol'){

      return  "₹"+ value.toFixed(2) + " " + "INR";

    }

    else{

      return  value.toFixed(2) + " " + "INR";

    }

  }

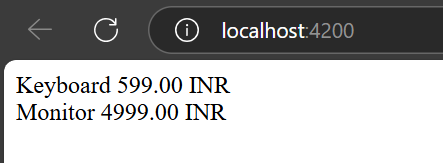
}

<div \*ngFor="let product of products">

    {{product.name}}

    {{product.price | priceFormat}}

</div>



6.

<div class="product-card">

    <h3>{{ name }}</h3>

    <p>Price: ${{ price }}</p>

  </div>

import { Component, Input } from '@angular/core';

import { CommonModule } from '@angular/common';

import { FormsModule } from '@angular/forms';

@Component({

  selector: 'app-product-item',

  standalone: true,

  imports:[FormsModule,CommonModule],

  templateUrl: './product-item.component.html',

  styleUrls: ['./product-item.component.css']

})

export class ProductItemComponent {

  @Input() name:string="";

  @Input() price:number=0;

}

import { Component } from '@angular/core';

import { CommonModule } from '@angular/common'; // ✅ Needed for \*ngFor

import { ProductItemComponent } from '../product-item/product-item.component';

@Component({

  selector: 'app-product-list',

  standalone: true,

  imports: [CommonModule, ProductItemComponent],

  templateUrl: './product-list.component.html',

  styleUrls: ['./product-list.component.css']

})

export class ProductListComponent {

  name:string="Abdul Raashid";

  price:number=999;

}

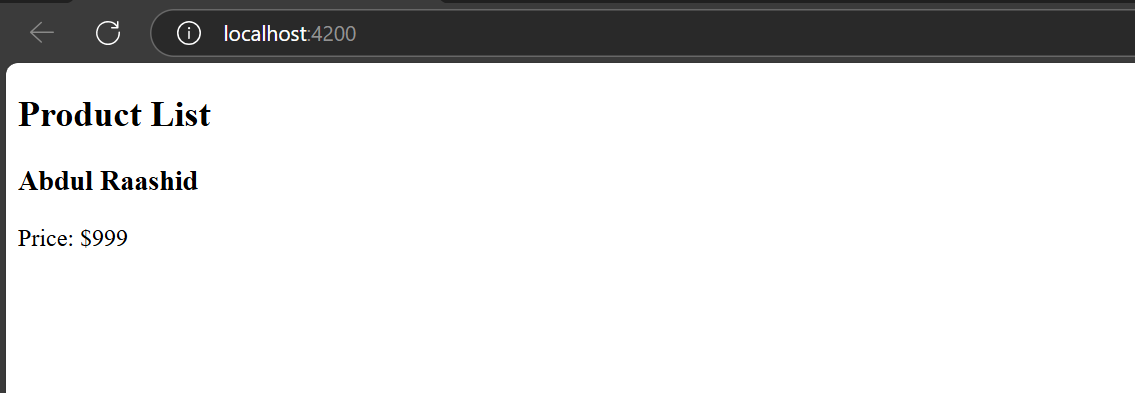
<h2>Product List</h2>

<app-product-item

  [name]="name"

  [price]="price">

</app-product-item>



7.

import { Component } from '@angular/core';

import { FormGroup, FormControl, Validators } from '@angular/forms';

@Component({

  selector: 'app-shopping-cart',

  templateUrl: './shopping-cart.component.html',

  styleUrls: ['./shopping-cart.component.css']

})

export class ShoppingCartComponent {

  cartForm = new FormGroup({

    name: new FormControl('', Validators.required),

    quantity: new FormControl(1, [Validators.required, Validators.min(1)])

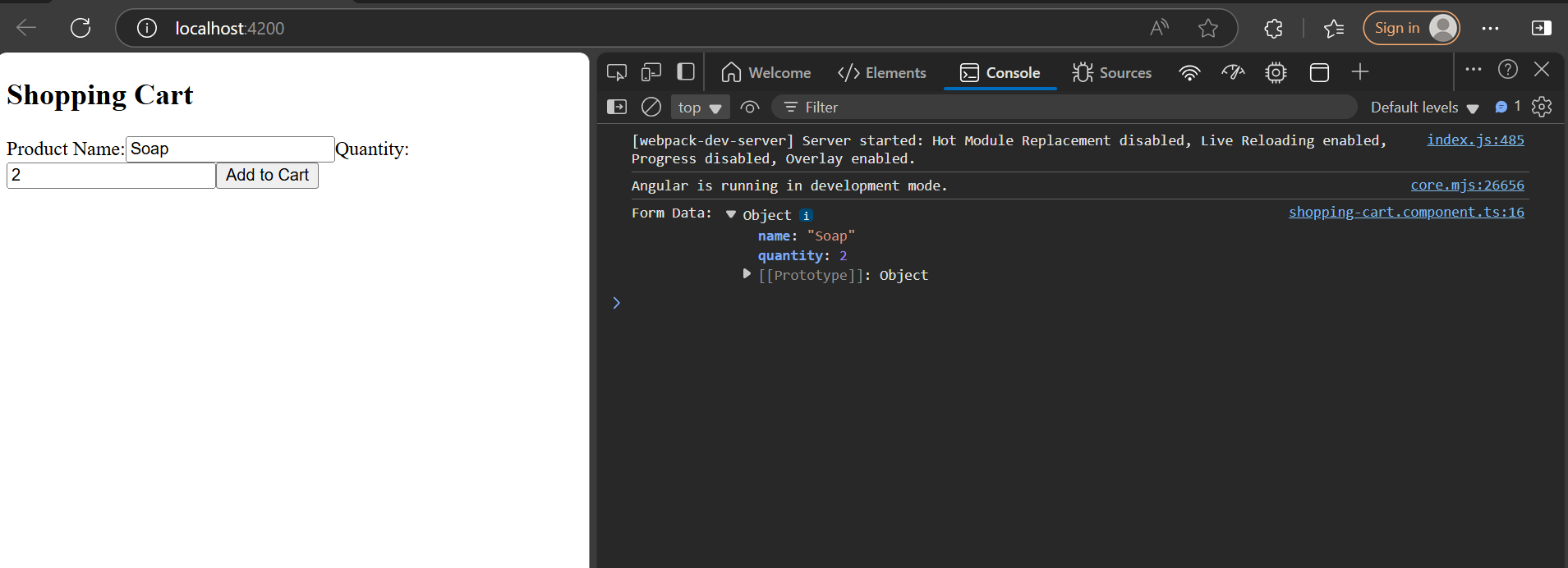
  });

  submitForm() {

    console.log('Form Data:', this.cartForm.value);

  }

}



8.

import { Component } from '@angular/core';

import { ReactiveFormsModule,FormGroup,FormControl, FormBuilder, Validators } from '@angular/forms';

@Component({

selector: 'app-eight',

standalone: false,

templateUrl: './eight.component.html',

styleUrl: './eight.component.scss'

})

export class EightComponent {

shoppingForm!:FormGroup;

constructor(private formBuilder:FormBuilder){

this.shoppingForm= this.formBuilder.group({

Name:['',Validators.required],

Quantity:['',Validators.required]

})

}

onSubmit(){

if(this.shoppingForm.valid){

console.log(this.shoppingForm.value)

}

}

}  
  
  
<p>eight works!</p>

<form [formGroup]="shoppingForm" (ngSubmit)="onSubmit()">

<input placeholder="product name" formControlName="Name">

<input placeholder="product Quantity" formControlName="Quantity">

<button type="submit">Submit</button>

</form>

9.

import { ProductService } from './../../Services/product.service';

import { Component, OnInit } from '@angular/core';

import { Product } from '../../Services/product.service';

@Component({

selector: 'app-nine',

standalone: false,

templateUrl: './nine.component.html',

styleUrl: './nine.component.scss'

})

export class NineComponent implements OnInit {

products:Product[]=[];

loading=true;

error='';

constructor(private productService:ProductService){}

ngOnInit() {

this.productService.getProducts().subscribe({

next: (data) => {

this.products = data;

this.loading = false;

},

error: (err) => {

this.error = 'Failed to load products';

this.loading = false;

console.error(err);

},

});

}

}

<h2>Product List</h2>

<div \*ngIf="loading">Loading products...</div>

<div \*ngIf="error">{{ error }}</div>

<div \*ngIf="!loading && !error">

<div \*ngFor="let product of products">

<h3>{{ product.title }}</h3>

<img [src]="product.image" alt="{{ product.title }}" width="100" />

<p>Price: ₹{{ product.price.toFixed(2) }}</p>

</div>

</div>

10.

import { Component } from '@angular/core';

@Component({

selector: 'app-ten',

standalone: false,

templateUrl: './ten.component.html',

styleUrl: './ten.component.scss'

})

export class TenComponent {

blogs = [

'Post 1', 'Post 2', 'Post 3', 'Post 4', 'Post 5',

'Post 6', 'Post 7', 'Post 8', 'Post 9', 'Post 10'

];

pageSize = 5;

currentPage = 0;

get startIndex(): number {

return this.currentPage \* this.pageSize;

}

get endIndex(): number {

return this.startIndex + this.pageSize;

}

nextPage() {

if (this.endIndex < this.blogs.length) {

this.currentPage++;

}

}

prevPage() {

if (this.currentPage > 0) {

this.currentPage--;

}

}

}

<p>ten works!</p>

<h2>Blog Posts (Page {{ currentPage + 1 }})</h2>

<ul>

<li \*ngFor="let blog of blogs | slice:startIndex:endIndex">

{{ blog }}

</li>

</ul>

<button (click)="prevPage()" [disabled]="currentPage === 0">Previous</button>

<button (click)="nextPage()" [disabled]="endIndex >= blogs.length">Next</button>

11.

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute } from '@angular/router';

interface Product {

name: string;

category: string;

}

@Component({

selector: 'app-eleven',

standalone: false,

templateUrl: './eleven.component.html',

styleUrl: './eleven.component.scss'

})

export class ElevenComponent implements OnInit {

allProducts: Product[] = [

{ name: 'Laptop', category: 'Electronics' },

{ name: 'Book A', category: 'Books' },

{ name: 'Smartphone', category: 'Electronics' },

{ name: 'Book B', category: 'Books' },

{ name: 'Shirt', category: 'Clothing' }

];

filteredProducts: Product[] = [];

constructor(private route: ActivatedRoute) {}

ngOnInit(): void {

this.route.queryParamMap.subscribe(params => {

const category = params.get('category');

if (category) {

this.filteredProducts = this.allProducts.filter(

p => p.category.toLowerCase() === category.toLowerCase()

);

} else {

this.filteredProducts = this.allProducts;

}

});

}

}

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute } from '@angular/router';

interface Product {

name: string;

category: string;

}

@Component({

selector: 'app-eleven',

standalone: false,

templateUrl: './eleven.component.html',

styleUrl: './eleven.component.scss'

})

export class ElevenComponent implements OnInit {

allProducts: Product[] = [

{ name: 'Laptop', category: 'Electronics' },

{ name: 'Book A', category: 'Books' },

{ name: 'Smartphone', category: 'Electronics' },

{ name: 'Book B', category: 'Books' },

{ name: 'Shirt', category: 'Clothing' }

];

filteredProducts: Product[] = [];

constructor(private route: ActivatedRoute) {}

ngOnInit(): void {

this.route.queryParamMap.subscribe(params => {

const category = params.get('category');

if (category) {

this.filteredProducts = this.allProducts.filter(

p => p.category.toLowerCase() === category.toLowerCase()

);

} else {

this.filteredProducts = this.allProducts;

}

});

}

}