

TS Capstone project

September 18, 2024

4NM21CS071

Code:

```
// Interface
interface IProduct {
    id: string;
    name: string;
    price: number;
}

interface ClothingProduct extends IProduct {
    fit: 'regular' | 'plus';
    size: string;
}

interface FoodProduct extends IProduct {
    expiry: Date;
}
```

```
// Decorators
function ValidateNonEmpty(target: any, propertyKey: string | symbol) {
    const originalMethod = target[propertyKey as string];

    target[propertyKey as string] = function (...args: any[]) {
        const [product] = args;

        if (!product.name || product.price === undefined) {
            throw new Error(
                `Validation failed: ${String(
                    propertyKey
                )} - Name and Price are required.`
            );
        }

        return originalMethod.apply(this, args);
    };
}
```

```
class ProductService<T extends IProduct> {
    private products: Map<string, T> = new Map();
```

```
    // CRUD operations
    // create
    @ValidateNonEmpty
    create(product: T): void {
```

```

    if (this.products.has(product.id)) {
        throw new Error("Product already exists.");
    }
    this.products.set(product.id, product);
}

// read
read(id: string): T | undefined {
    return this.products.get(id);
}

// update
@ValidateNonEmpty
update(id: string, updatedProduct: Partial<T>): void {
    const product = this.products.get(id);

    if (!product) {
        throw new Error("Product not found.");
    }

    this.products.set(id, { ...product, ...updatedProduct } as T);
}

// delete
delete(id: string): void {
    if (!this.products.has(id)) {
        throw new Error("Product not found.");
    }
    this.products.delete(id);
}

// get all
getAll(): T[] {
    return Array.from(this.products.values());
}
}

const clothingProductService = new ProductService<ClothingProduct>();
const foodProductService = new ProductService<FoodProduct>();

// input outputs
const shirt: ClothingProduct = {
    id: "1",
    name: "Shirt",

```

```
    price: 1500,  
    fit: "regular",  
    size: "L",  
  };  
  
  const chips: FoodProduct = {  
    id: "2",  
    name: "Chips",  
    price: 50,  
    expiry: new Date("2024-11-15"),  
  };  
};
```

```
clothingProductService.create(shirt);  
foodProductService.create(chips);
```

```
console.log("All clothing products:", clothingProductService.getAll());  
console.log("All food products:", foodProductService.getAll());
```

```
try {  
  clothingProductService.update("1", { price: 1200 });  
  console.log(  
    "Updated clothing product:",  
    clothingProductService.read("1")  
  );  
} catch (error) {  
  console.error("Error updating product:", error);  
}
```

```
console.log("Updated clothing product:", clothingProductService.read("1"));
```

```
try {  
  foodProductService.delete("2");  
  console.log(  
    "Food products after deletion:",  
    foodProductService.getAll()  
  );  
} catch (error) {  
  console.error("Error deleting product:", error);  
}
```

```

console.log("\n\n");

const jacket: ClothingProduct = {
  id: "3",
  name: "Jacket",
  price: 3500,
  fit: "plus",
  size: "XL",
};
clothingProductService.create(jacket);
console.log("All clothing products:", clothingProductService.getAll());

console.log("\n\n");

const milk: FoodProduct = {
  id: "4",
  name: "Milk",
  price: 50,
  expiry: new Date("2024-10-10"),
};
foodProductService.create(milk);
console.log("All food products:", foodProductService.getAll());

console.log("\n\n");
try {
  clothingProductService.update("1", { size: "XL" });
  console.log("Updated clothing product:", clothingProductService.read("1"));
} catch (error) {
  console.error("Error updating clothing product:", error);
}

console.log("\n\n");
try {
  foodProductService.update("4", { price: 55 });
  console.log("Updated food product:", foodProductService.read("4"));
} catch (error) {
  console.error("Error updating food product:", error);
}

console.log("\n\n");

try {
  const duplicate: FoodProduct = {
    id: "4",

```

```
        name: "Nandini",
        price: 55,
        expiry: new Date("2024-11-01"),
    };
    foodProductService.create(duplicate);
} catch (error) {
    console.error("Error creating duplicate food product:", error);
}

console.log("\n\n");
try {
    clothingProductService.update("71", { price: 2000 });
} catch (error) {
    console.error("Error updating non-existing product:", error);
}

console.log("\n\n");
try {
    foodProductService.delete("45");
} catch (error) {
    console.error("Error deleting non-existing product:", error);
}

console.log("\n\n");
```

Screenshots:

```
PS D:\Finacle Training\TypeScript> node .\src\main.js
All clothing products: [ { id: '1', name: 'Shirt', price: 1500, fit: 'regular', size: 'L' } ]
All food products: [
  {
    id: '2',
    name: 'Chips',
    price: 50,
    expiry: 2024-11-15T00:00:00.000Z
  }
]
```

```
Updated clothing product: { id: '1', name: 'Shirt', price: 1200, fit: 'regular', size: 'L' }
```

```
Food products after deletion: []
```

```
All clothing products: [
  { id: '1', name: 'Shirt', price: 1200, fit: 'regular', size: 'L' },
  { id: '3', name: 'Jacket', price: 3500, fit: 'plus', size: 'XL' }
]
```

```
All food products: [
  {
    id: '4',
    name: 'Milk',
    price: 50,
    expiry: 2024-10-10T00:00:00.000Z
  }
]
```

```
Updated clothing product: { id: '1', name: 'Shirt', price: 1200, fit: 'regular', size: 'XL' }
```

```
Updated food product: { id: '4', name: 'Milk', price: 55, expiry: 2024-10-10T00:00:00.000Z }
```

//ERROR

```
Error creating duplicate food product: Error: Product already exists.  
  at ProductService.create (D:\Finacle Training\TypeScript\src\main.js:74:27)  
  at Object.<anonymous> (D:\Finacle Training\TypeScript\src\main.js:194:24)  
  at Module._compile (node:internal/modules/cjs/loader:1241:14)  
  at Module._extensions..js (node:internal/modules/cjs/loader:1295:10)  
  at Module.load (node:internal/modules/cjs/loader:1091:32)  
  at Module._load (node:internal/modules/cjs/loader:938:12)  
  at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:83:12)  
  at node:internal/main/run_main_module:23:47
```

```
Error updating non-existing product: Error: Product not found.  
  at ProductService.update (D:\Finacle Training\TypeScript\src\main.js:86:27)  
  at Object.<anonymous> (D:\Finacle Training\TypeScript\src\main.js:201:28)  
  at Module._compile (node:internal/modules/cjs/loader:1241:14)  
  at Module._extensions..js (node:internal/modules/cjs/loader:1295:10)  
  at Module.load (node:internal/modules/cjs/loader:1091:32)  
  at Module._load (node:internal/modules/cjs/loader:938:12)  
  at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:83:12)  
  at node:internal/main/run_main_module:23:47
```

```
Error deleting non-existing product: Error: Product not found.  
  at ProductService.delete (D:\Finacle Training\TypeScript\src\main.js:93:27)  
  at Object.<anonymous> (D:\Finacle Training\TypeScript\src\main.js:208:30)  
  at Module._compile (node:internal/modules/cjs/loader:1241:14)  
  at Module._extensions..js (node:internal/modules/cjs/loader:1295:10)  
  at Module.load (node:internal/modules/cjs/loader:1091:32)  
  at Module._load (node:internal/modules/cjs/loader:938:12)  
  at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:83:12)  
  at node:internal/main/run_main_module:23:47
```