Assignment on Angular forms
Title: Product Entry Form using Angular Reactive Forms

Objective
The objective of this assignment is to build an Angular application that uses Reactive Forms to create a product entry form. The form should include fields for selecting a product category, brand, and inputs for product name, price, and description. The form should provide real-time validation feedback and prevent submission until all input fields are valid.
Tools needed

- Angular CLI
- A text editor (Visual Studio Code, Atom, Sublime Text, etc.)
- A web browser for testing

Tasks
Task 1: Setup Angular Environment
If you haven't already, install the Angular CLI. Create a new Angular project using the **ng new** command in your terminal or command prompt.
Task 2: Create a Category-Brand Service
Create a service that will provide data for categories and brands.
Use the **ng generate service** command to generate a new service. This service should contain two methods: **getCategories()** and **getBrands(category: string)**. You can start by returning hardcoded arrays of categories and brands from these methods.
Task 3: Create a Reactive Form
Generate a new component using **ng generate component**. In this component, create a new FormGroup instance **productForm** which includes FormControl instances for the following:

- **categoryControl** - For selecting the product category.
- **brandControl** - For selecting the product brand.
- **productNameControl** - For entering the product name.
- **priceControl** - For entering the product price.
- **descriptionControl** - For entering the product description.

The FormControl for brand should initially be disabled, because a brand can't be selected until a category is selected.
The FormControl instances should be initialized with appropriate Validators for required fields, minimum lengths, maximum lengths, and minimum values where applicable.
Task 4: Populate the Category and Brand Form Controls
Use the **getCategories()** and **getBrands(category: string)** methods from your service to populate the options for **categoryControl** and **brandControl**. The brand control should be populated based on the selected category.
Task 5: Enable the Brand Control Upon Category Selection
Subscribe to changes on **categoryControl**. When the value changes, enable **brandControl** and populate it with brands that belong to the selected category.
Task 6: Display the Form
In the component's template, create a form using formGroup to bind **productForm**. Create form fields using formControlName to bind the form controls. Use ***ngFor** to loop through and display the options for category and brand.
Task 7: Validate the Form

Implement a validation method for **productForm** that checks if all form controls are valid. If the form controls are invalid, display error messages in the form fields.
Task 8: Submit the Form
Create a submit button for the form. When the button is clicked, log the current value of **productForm** if it's valid. If it's not valid, display an error message.

Expected Outcome
At the end of this assignment, you should have an Angular application that includes a form where users can select a category and a brand, enter a product name, price, and description, and submit the form. The form should have real-time validation for all fields and display error messages for invalid fields. The form should prevent submission until all fields are valid.

Bonus Tasks
Bonus Task 1: Complex Validation
Add complex validation such as checking if the entered price is within a reasonable range for the selected category and brand, or checking if the entered product name matches a certain pattern.
Bonus Task 2: Reset Form
Implement a "reset form"

Use following sample json for the reference
categorories.json

```
{
  "categories": [
    { "id": 1, "name": "Electronics"},
    { "id": 2, "name": "Automobiles" },
    { "id": 3, "name": "Clothing" }
  ]
}
```

brands.json

```
{
  "brands": [
    { "categoryId": 1, "name": "Apple" },
    { "categoryId": 1, "name": "Samsung" },
    { "categoryId": 1, "name": "Sony" },
    { "categoryId": 2, "name": "Tesla" },
    { "categoryId": 2, "name": "Ford" },
    { "categoryId": 2, "name": "Toyota" },
    { "categoryId": 3, "name": "Levis" },
    { "categoryId": 3, "name": "H&M" },
    { "categoryId": 3, "name": "Zara" }
  ]
}
```

Add brand id to brands.json if you need.