Activity 3: Part 1

Ryan Coon

CST-391

Professor Bobby Estey

July 16, 2023

**Screenshots:**

**A screenshot of a computer

Description automatically generated**

**Here is the grid shown on a normal screen intended for a computer monitor resolution.**

**A screenshot of a computer

Description automatically generated**

**Here is the same grid that has responsively changed its resolution to fit a mobile phone.**

**A white background with a black line

Description automatically generated**

**Here is the new form we created in the Shop component**

**A white background with black lines

Description automatically generated**

**Here we see that when a name is entered, it changes with the onSubmit function.**

**A white text box with black lines

Description automatically generated**

**Here is the new infoComponent with the form in its view. Here we can select from a list, submit, or reset the form.A screenshot of a computer

Description automatically generated**

**Here is the console view that we can see in the info form when a button is pressed, it logs it into the console.**

**Research:**

**Describe @Input decorator used in info.component.ts**

The @Input decorator in Angular is used to define an input property in a component. It allows data to be passed into the component from its parent component or another component. The @Input decorator is imported from the @angular/core package and is used as a decorator function that can be applied to a class property.

**Describe [value] used in info.component.html**

The [value] attribute binding in Angular is used to bind a value to an input, select, or textarea element. It allows you to set the initial value of the form control or dynamically update the value based on the component's data. By using [value] binding, you can easily synchronize the values of form controls with the component's data, allowing for two-way data binding and a seamless user experience.

**Describe [(ngModel)] also used in info.component.html**

The [(ngModel)] directive in Angular is a two-way data binding syntax that allows you to bind a form control element to a component property. It combines the property binding [ngModel] and the event binding (ngModelChange) into a single shorthand syntax for two-way data binding. By using [(ngModel)], you can easily bind form controls to component properties and keep them in sync. It simplifies the process of capturing user input and updating the component's data model, enabling a smooth and interactive user experience.