Lab Exercise: Angular Data Binding

Objective

To understand and practice the various forms of data binding in Angular, including interpolation, property binding, event binding, and two-way binding.

Requirements

- Angular CLI installed on your machine.
- Basic knowledge of Angular project structure.
- Basic knowledge of TypeScript.

Setup

1. Create a new Angular project by running `ng new data-binding-practice`.
2. Navigate into your project directory (`cd data-binding-practice`).
3. Serve your application to verify its working: `ng serve`.
4. Open your project in your favorite code editor.

Exercises

Exercise 1: Interpolation

1. In your app component (`app.component.ts`), define a new property `title` with the value "Angular Data Binding Practice".
2. In the app component template (`app.component.html`), use interpolation to display the `title` property within an `<h1>` tag.
3. Serve your application to see the changes.

Exercise 2: Property Binding

1. Add a new property `imageSrc` in your app component with the value of any image URL.
2. In the app component template, use property binding to set the `src` attribute of an `<img>` tag to the `imageSrc` property.
3. Serve your application to ensure the image is displayed.

Exercise 3: Event Binding

1. Add a button to your app component template with the text "Click me".
2. In the app component, define a method `onButtonClick()` that simply logs "Button was clicked!" to the console.
3. Use event binding to call `onButtonClick()` when the button is clicked.
4. Serve your application and open the developer console to see the log message when you click the button.

Exercise 4: Two-Way Binding

1. First, ensure `FormsModule` is imported in your `app.module.ts` file to use two-way data binding.

2. In your app component, define a new property `userInput` with an initial value of an empty string.
3. Add an `<input>` element to your app component template.
4. Use two-way binding to bind the `userInput` property to the `<input>` element's value.
5. Below the `<input>`, use interpolation to dynamically display the `userInput` value as the user types.
6. Serve your application to see two-way data binding in action.

Conclusion

By completing these exercises, you should have a solid understanding of the different types of data binding in Angular and how they can be used to create dynamic, interactive web applications.