**Activity**

Data Binding and Event Handling

1.Use interpolation to dynamically display the applicant's name in the form.

Steps:

* Create an Angular component (ApplicantComponent).
* Define a property applicantName in the component with your name.
* In the component's template, use interpolation to display the applicant's name inside an <h1> tag.
* Verify that the applicant's name appears correctly in the rendered form.

2. You are developing a job board application where each job listing has a "Apply Now" button. Use property binding to enable or disable the button based on the availability of job positions.

Steps:

* Create an Angular component (JobListingComponent).
* Define a property jobAvailable in the component and set it to true.
* Use property binding to conditionally enable or disable the "Apply Now" button based on the jobAvailable property.
* Toggle the jobAvailable property and observe the button's state changes.

3.In your job application form, add a "Submit Application" button. Use event binding to display an alert when the button is clicked.

Steps:

* Enhance the ApplicantComponent by adding a "Submit Application" button.
* Use event binding to bind a method (onSubmitApplication()) to the button's click event.
* In the onSubmitApplication() method, display an alert message like "Application Submitted Successfully!".
* Test the application by clicking the "Submit Application" button.

4. You are building a user profile editing page for a job portal. Implement two-way data binding to allow users to edit their profile information in real-time.

Steps:

* Create an Angular component (UserProfileComponent).
* Define properties for user information such as userName, userEmail, etc.
* Use two-way data binding with [(ngModel)] to bind the input fields to the corresponding properties.
* Display the current user information and allow users to edit it in the component template.
* Verify that changes made in the input fields are reflected in the component properties and vice versa.