

## Technical Interview Questions - Student Answers

### 1. Briefly explain the two types of Angular Forms.

As I've learned in this course, Angular offers two main approaches to building forms:

**Template-Driven Forms:** These are the forms I started with because they felt more intuitive. You basically add Angular directives like `ngModel` directly in your HTML template. I found them easier to set up for simple forms like our login form. The form controls are created automatically behind the scenes. The trade-off is that they're harder to unit test and less flexible for complex scenarios.

**Reactive Forms:** These took me longer to understand but are more powerful. With reactive forms, I explicitly create form controls in the TypeScript component class. I've learned they're better for complex forms because you have more control over validation and form state. In our project, if we needed complex validation rules for user registration, reactive forms would be the better choice.

### 2. What does data-binding in Angular allow you to do?

From my experience working on the J-Flix project, data-binding is one of Angular's most helpful features. It lets me connect my component's data (in TypeScript) with what the user sees in the HTML.

Before understanding data binding, I was manually updating the DOM or using event listeners like in vanilla JavaScript. But with Angular's data binding, I can:

- Display component data in the template (like showing the username in our welcome message)
- Listen for user events (like our login button clicks)
- Update component data when form inputs change (like collecting login credentials)

I especially appreciate two-way binding with `[(ngModel)]` in our forms because it automatically updates my component variables as the user types, saving me from writing extra code.

### 3. What is interpolation?

Interpolation is the simplest form of data binding I've used in Angular. It's those double curly braces `{{ }}` that let me display text from my component directly in the HTML.

In our J-Flix app, I used interpolation to display the welcome message with `<h1>Welcome to {{title}}!</h1>`. Angular replaces the expression inside the braces with the actual value from my component.

What I like about interpolation is that it's not just limited to property names because I can put any valid JavaScript expression inside the braces. For example, I could do something like `{{ username || 'Guest' }}` to show either the username or "Guest" if username is empty. I found this really helpful when creating conditional text in my templates.