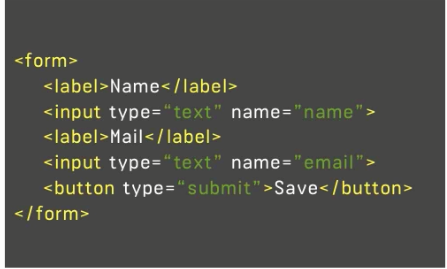
**Section 15: Handling forms in Angular Apps**

**Section 15: Lecture 168//Module Introduction**

1. This course module is about forms i.e. forms in our HTML page and how angular can help us with those forms.
2. We know that a form is something that we can submit to the server and then we must keep in mind that we are creating single page application – So there is no submitting to the server.
3. Instead we would need to handle the form through angular and then if we must submit something to server then we would need to submit is through angular HTTP Service – which we will cover later.
4. So, we must handle our forms through angular and actually angular will help us handling it. It will not only allow us to check the values that user entered but it will also allow us to check if the form is valid.
5. It will allow us to change the way in which form is displayed, put some red borders around invalid – so angular really ships with some powerful tools when it comes to working with forms.

**Section 15: Lecture 169//Why do we need Angular’ s help**

1. Let’s have a closer look at how angular works together with forms. Consider below given form:



1. This is the HTML code of the form which we might use and there is nothing angular specific in this form. This is normal HTML code which you place in your HTML documents.
2. Angular’ s job here is to allow you to retrieve the values of the fields entered by you here and also check some other things like - is the form valid?; did the user entered the valid information and all that will happen in JavaScript i.e. in Typescript so on the angular 2 side of the App.
3. So, we somehow need to be able to parse the values, the user enters and you somehow need some JavaScript object representation of your form in your typescript code to work with.
4. So, we need to fetch the value of the form in JavaScript code and there we could have the key value pairs i.e. name value which is referred to as key and value as the email here. These all are mentioned as name in the html input type; so we can fetch the value of these in JavaScript by using these names.
5. There can be other properties which would change on the scenarios according to which the values of the input type changes
6. Javascript object representation of our form makes it simple for us to retrieve the values and to see the state of the form and to work with it.
7. In the next lectures we are going to have the detailed look.