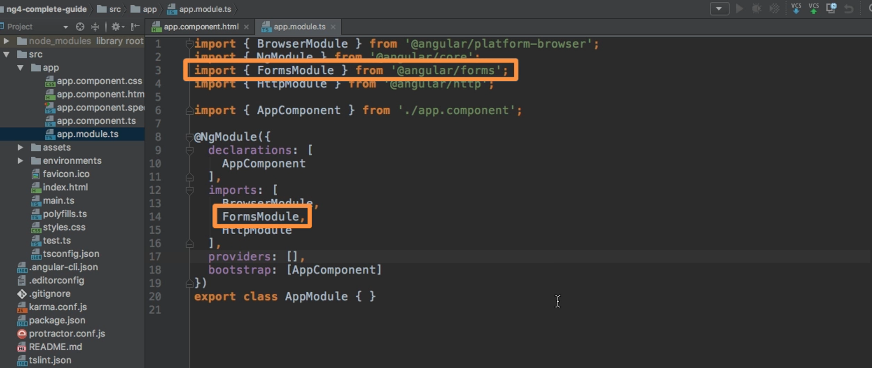
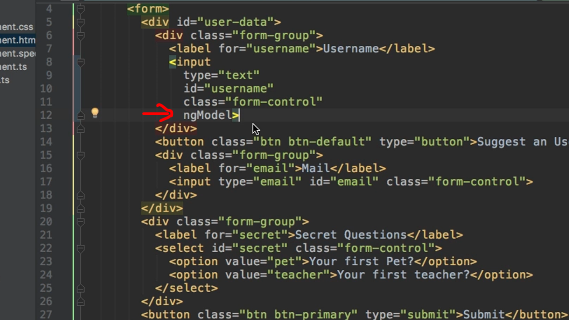
**186. TD Creating the Form and Registering the Controls**

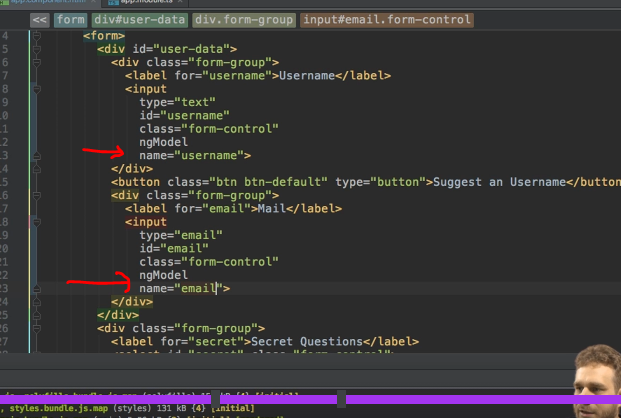
* Instructor: In the last video we had a look at the basic setup of our form .
* Now let's understand how Angular creates such a JavaScript object representing our form in the template driven approach .
* The great thing is you don't have to do anything while almost anything .
* **1) Imports:**
* Make sure that in your app module you actually import the forms module added here to your imports array and have the import at the top of this file .



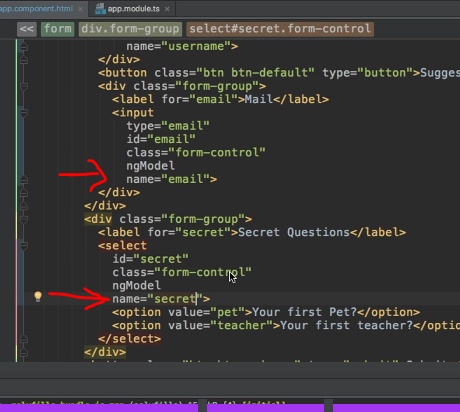
* As the name implies, this built-in module shipping with Angular includes a lot of forms related functionalities and is actually needed to get this template driven approach to work to get this form creation by Angular to work .
* Now by default in a CLI project this should already be imported .
* So it show to work fine, but again, I'm just highlighting it because it's super important that you have this import .
* Otherwise, the following steps will not work .
* *With this imported angular will actually automatically create a form for you, so a JavaScript representation of that form when it detects a form element in HTML code like it does here .*
* *So you can think of that form element serving as a selector for some angular directive which then creates such a JavaScript representation of the form for you .*
* Of course, you can't see that form representation as of now and it would be very empty because one thing does not happen automatically .
* ***Angular will not automatically detect your inputs in this form .***
* And the simple reason for this is that while you could argue that it should be able to scan your HTML code and detect that you have an input here and here and that you have a select dropdown here you still might not want to add all these elements as controls to your form .
* So with control, I'm referring to what is in the JavaScript object .
* And again, not every input in your HTML code might be a control you want to have in your JavaScript form .
* Maybe you have a dropdown which value only changes something you view on the UI, but the import should not actually be part of what gets submitted .
* Maybe you use some third party package which adds some custom form controls, which are not labeled input, which don't use input as a selector and then Angular would have no chance of detecting that this is a control of your form .
* *So you still need to register controls manually .*
* You need to tell Angular, "Hey, within that form element what should be an actual control of my form?" And this is what we're going to do now .
* **2) Telling angular about controls, how the form looks etc:**
* *Tell Angular how does our form look like? Which controls do we want to have? In the template of an approach, this is super simple .*
* You simply pick the input which you want to add as a control, like this input here .
* And I'm just going to structure it a bit different to split it up over multiple lines to make it easier to read .
* And then you add ***NG model*** like this .



* Now you might already know NG model from the two-way binding, two-way data binding and it actually is the same directive in two-way data binding, though you saw that we used it with square brackets and parentheses wrapping NG model .
* Now we will have a look at this later again, but for now let's add without any parentheses without any squared brackets, just like this .
* This will be enough to tell Angular, "Hey, this input is actually a control of my form .
* " So NNG model in the end is a directive made available in the forms module .
* Something I mentioned earlier in the course when we had a look at two-way data binding .
* This is key to understand you can use it to get two-way data binding but it actually is part of a bigger module with more features giving you full control over forms .
* Now for this to work, for this to be recognized as a control in your form*, we need to give Angular one hour piece of information, the name of this control right now it would say, "Okay, this input should be part of the JavaScript object representation of this forum .*
* " So whatever the user enters here as a value should be the value of my well of this control .
* But what's the name of that control? We need to give that information to Angular .
* And we do this by adding the normal HTML attribute name .
* So name is nothing, Angler is too specific .
* Name is the default attribute you can add to any HTML control .
* Now here the name might be username because that is what we can enter in this input .
* And with this, this control will be registered in this JavaScript representation of the form .
* Now, I'll do the same for the email restructure it so that it's easier to read add NG model and add a name like for example, email here .



* Well, and the same is true for the select here which is just another type of HTML input .
* Here we can also add NNG model and we can add a name like secret because am I asking for a secret question here? And with that we registered all the controls .



* Of course we can't see that much though .
* Well, that is something we're going to have a look at in the next lecture when we see how we can submit such a form and therefore how we can see these key value pairs representing what the user entered into which input .