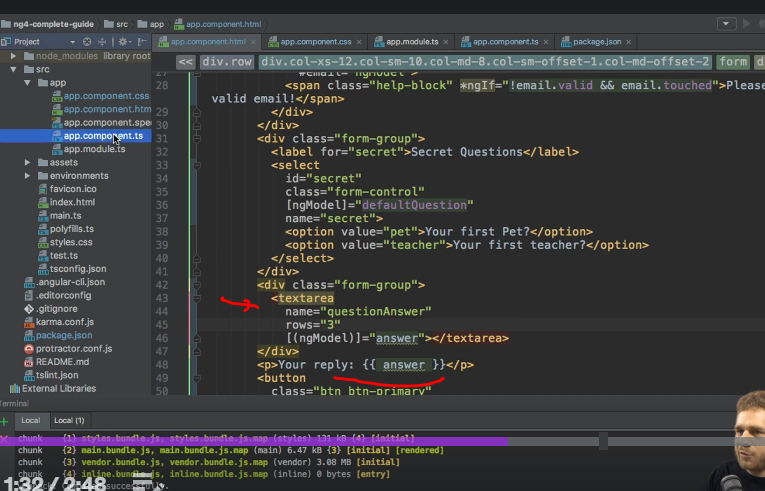
* -: Sometimes, you not only want to pre-populate the default user entered, but you also want to instantly react to any changes.
* Right now, everything about this form here only updates once I click "Submit," then I get this form object, where I can retrieve the value.
* We don't use two-way binding.
* ngModel is either used without any binding or with one-way binding.
* But sometimes you instantly want to check something or simply repeat whatever the user entered.
* Let's show an example.
* For this question here, I'm going to add a new "form-group," bootstrap style here, below the dropdown.
* I want to have a "textarea" let's say to also introduce a new input.
* Let's name it "questionAnswer.
* " Don't need the "id" here, don't need columns, I want to have 3 rows, let's say and I'll also distribute this over model lines, and now we could add "ngModel" to get access to whatever the user entered as a reply.
* But maybe, for some reason we instantly want to repeat this reply.
* So below this we could say "Your reply:" and now we want to output the answer to the question.
* Well, for this, we could still use two-way binding on ngModel here, and bind it to the answer property here.

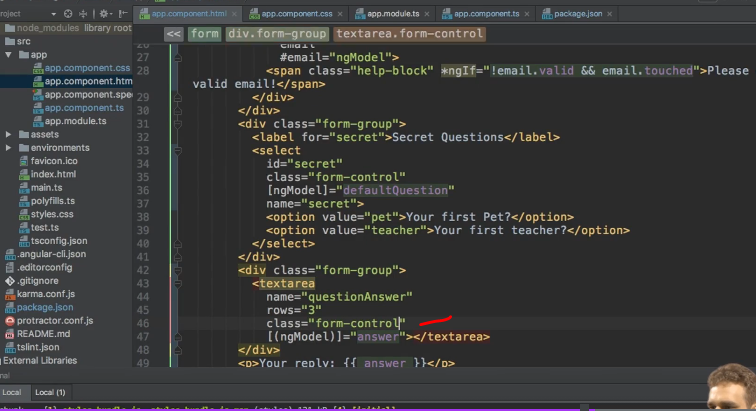


* Of course, we need to introduce this answer property then there, an answer could be an empty string by default.

Text

Description automatically generated

* With this in place, let me also add a CSS class to the text area "form-control," to give it some nice styling.



* If this reloads, we see our text area here, and if I enter valid data here, for my first pet let's say, a rabbit named Nelly, you instantly see that here.

Graphical user interface, text, application, email

Description automatically generated

* So you can still use two-way binding.
* And actually, if I submit the whole form and open up the developer tools, you see in the value object this is still part of this value object, it just is updated with every key stroke, but of course if we submit it, we will get a snapshot of the value at a point of time we hit "Submit.
* " So two-way binding is still possible.

Graphical user interface, text, application, email

Description automatically generated

* You can still use ngModel with two-way binding, and with that, ***you saw all 3 forms, no binding to just tell Angular data input is a control, one-way binding to give that control a default value, and two-way binding to instantly output it or do whatever you want to do with that value.***