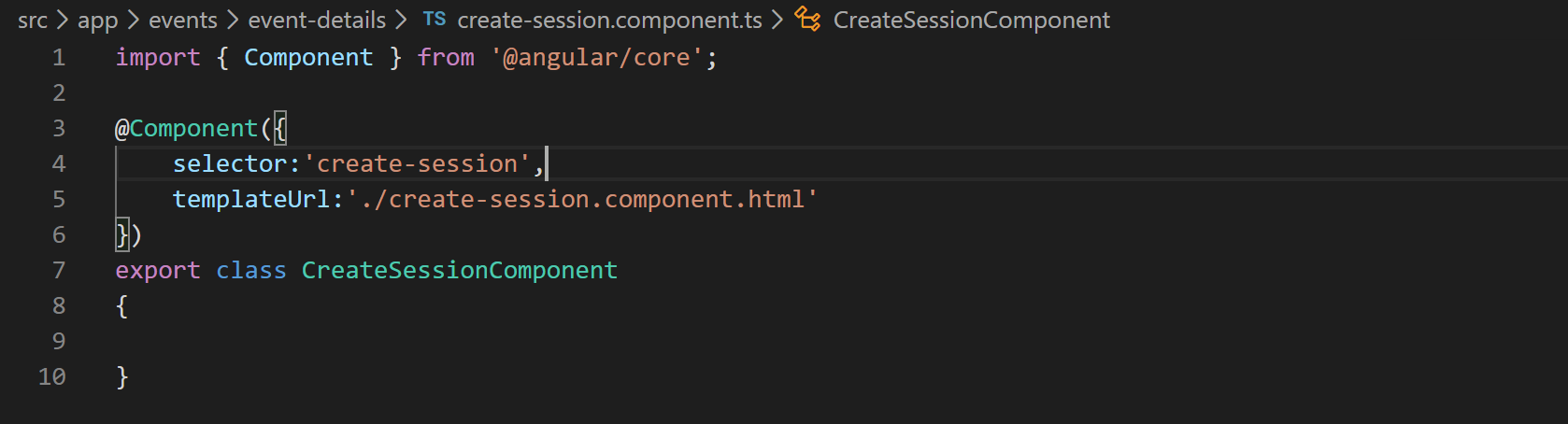
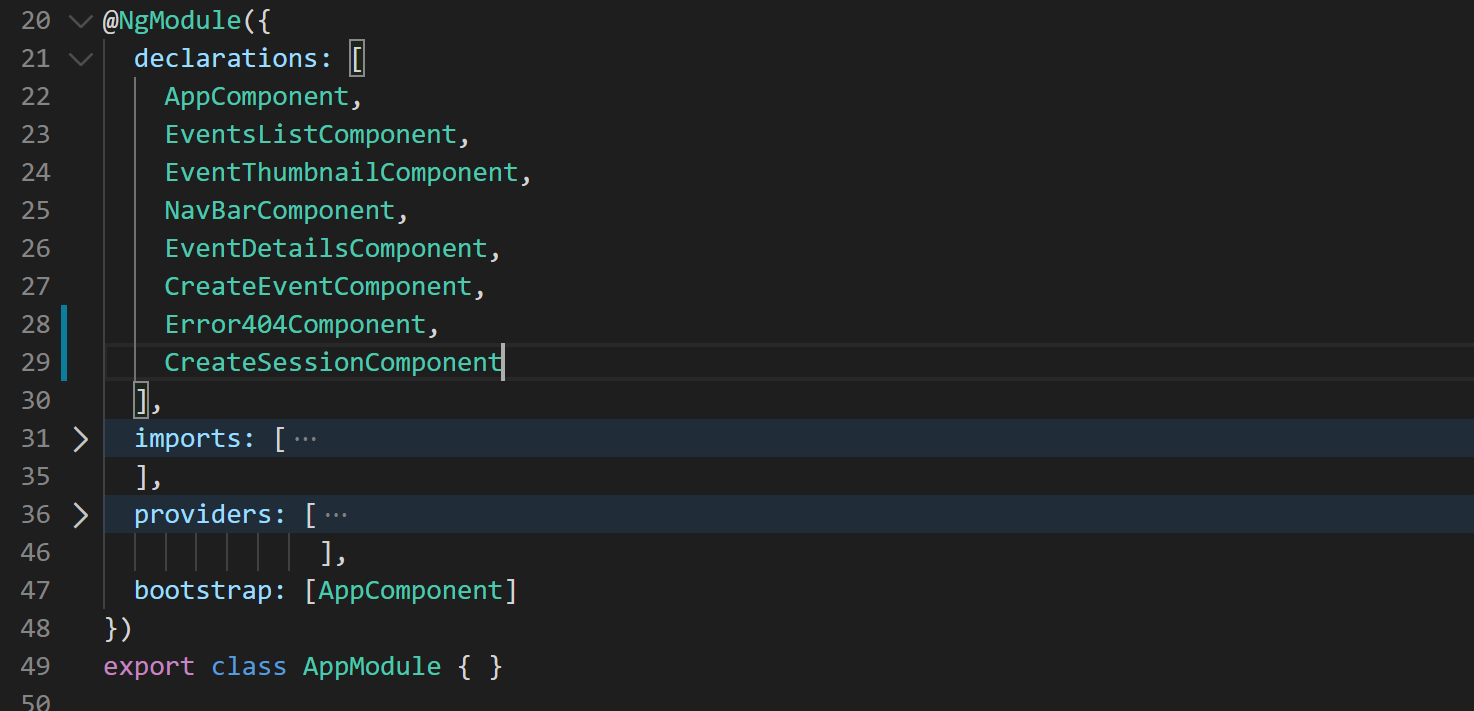
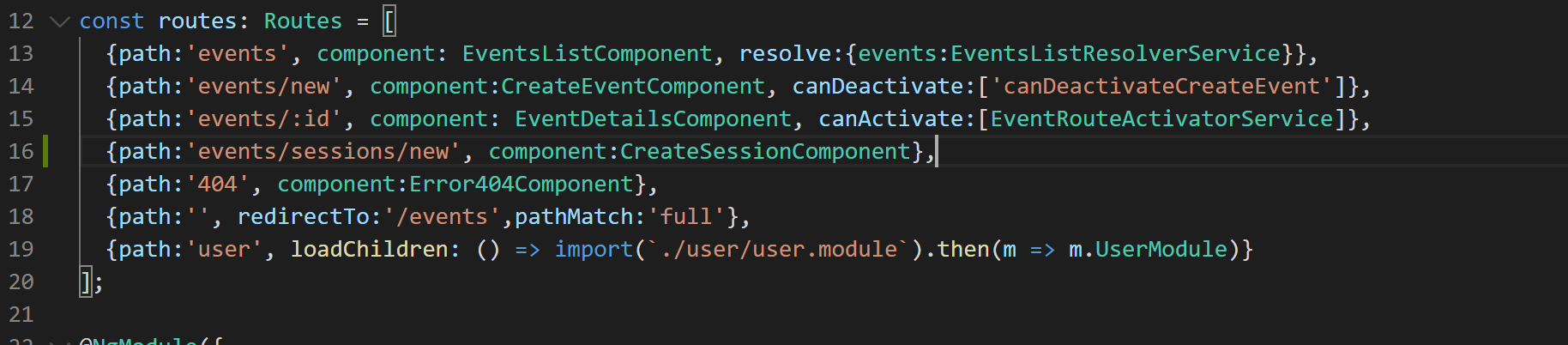
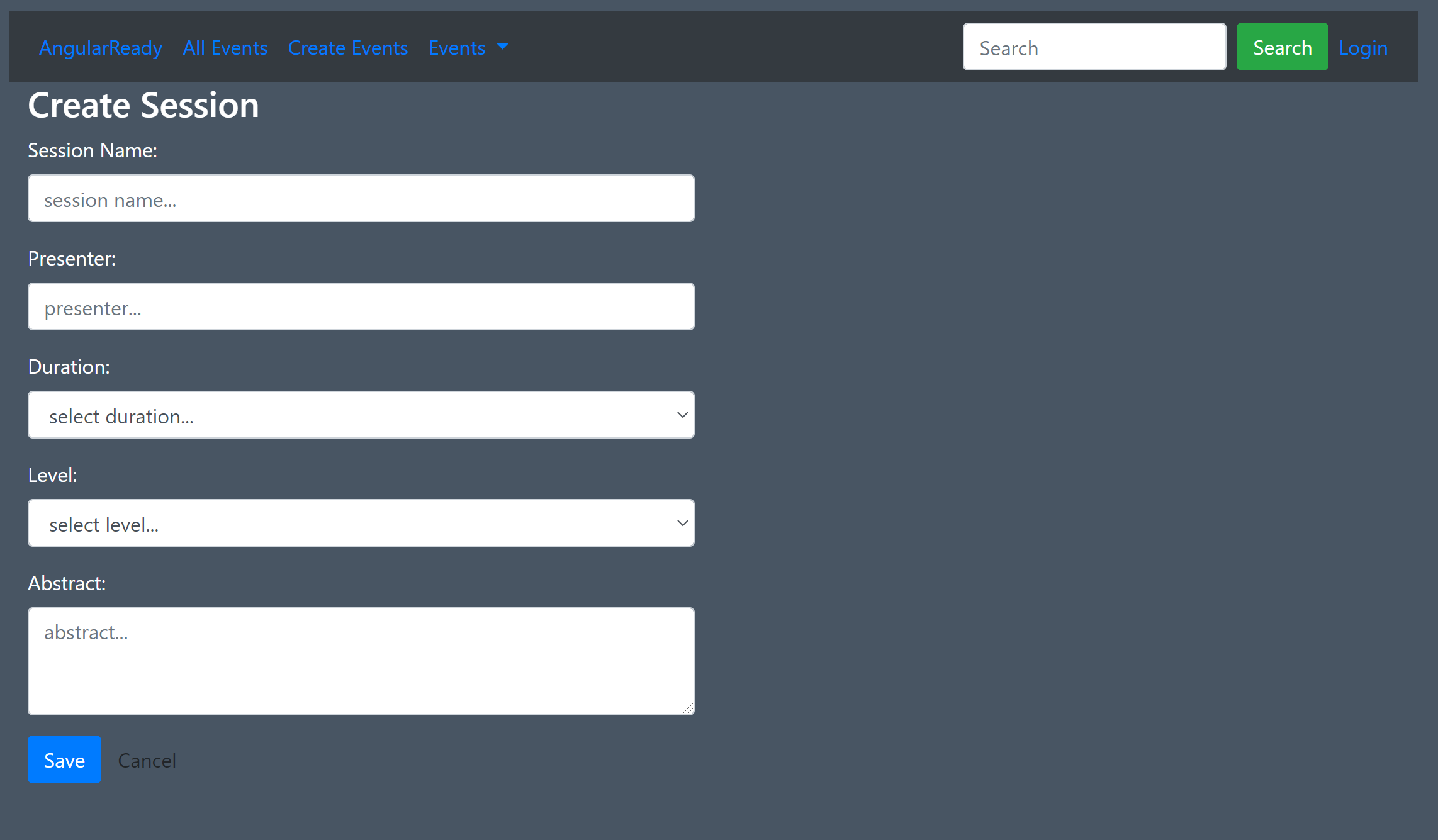
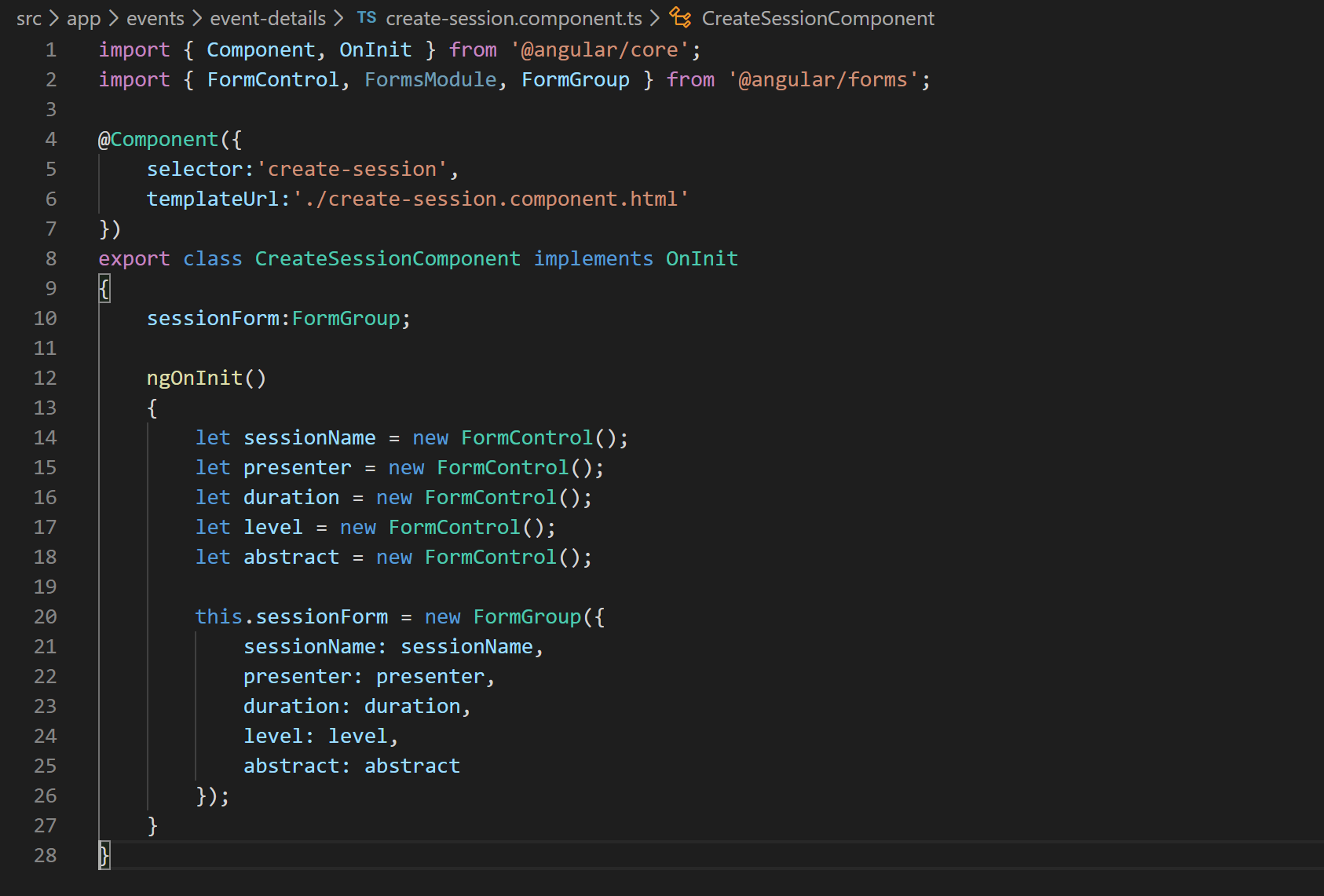
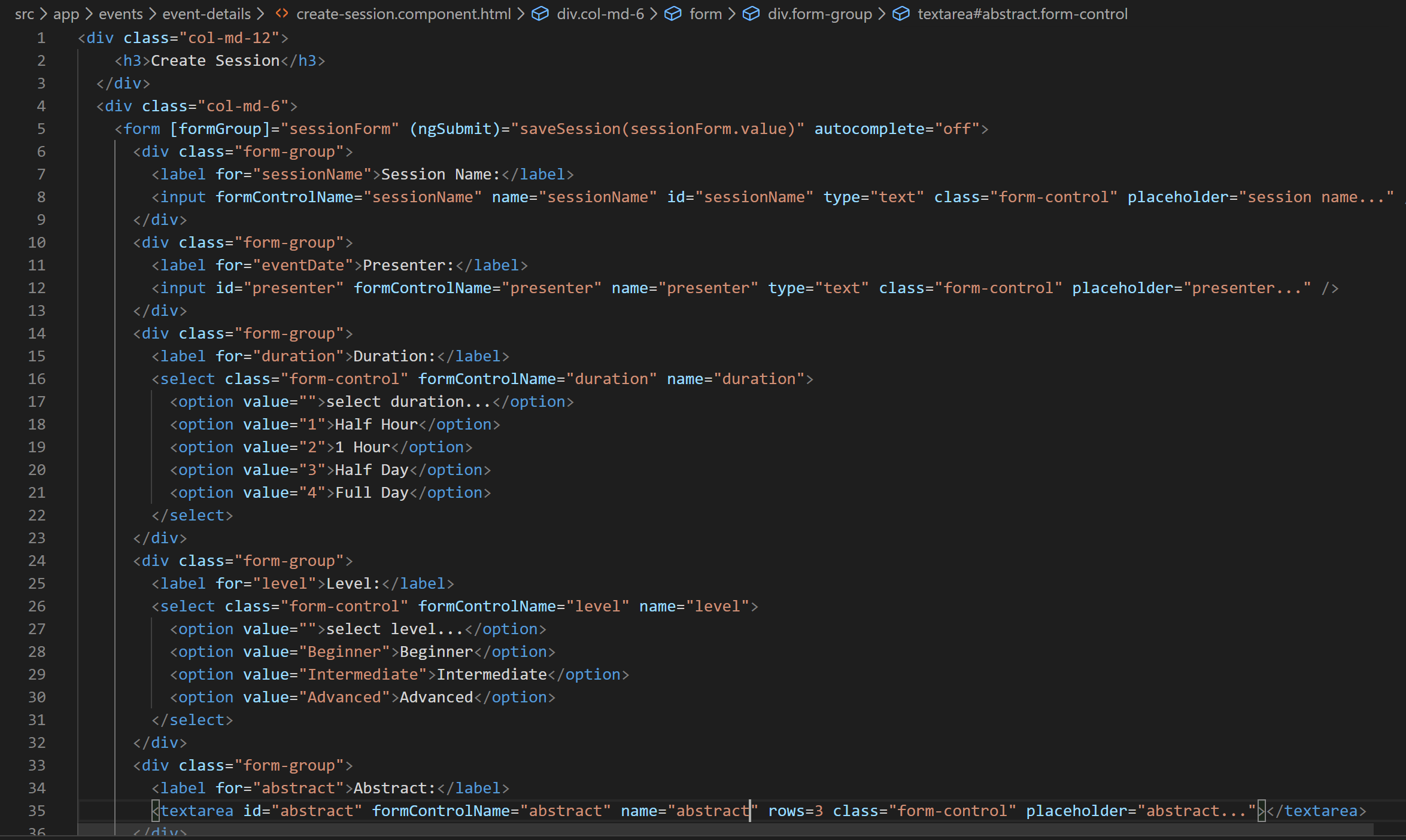
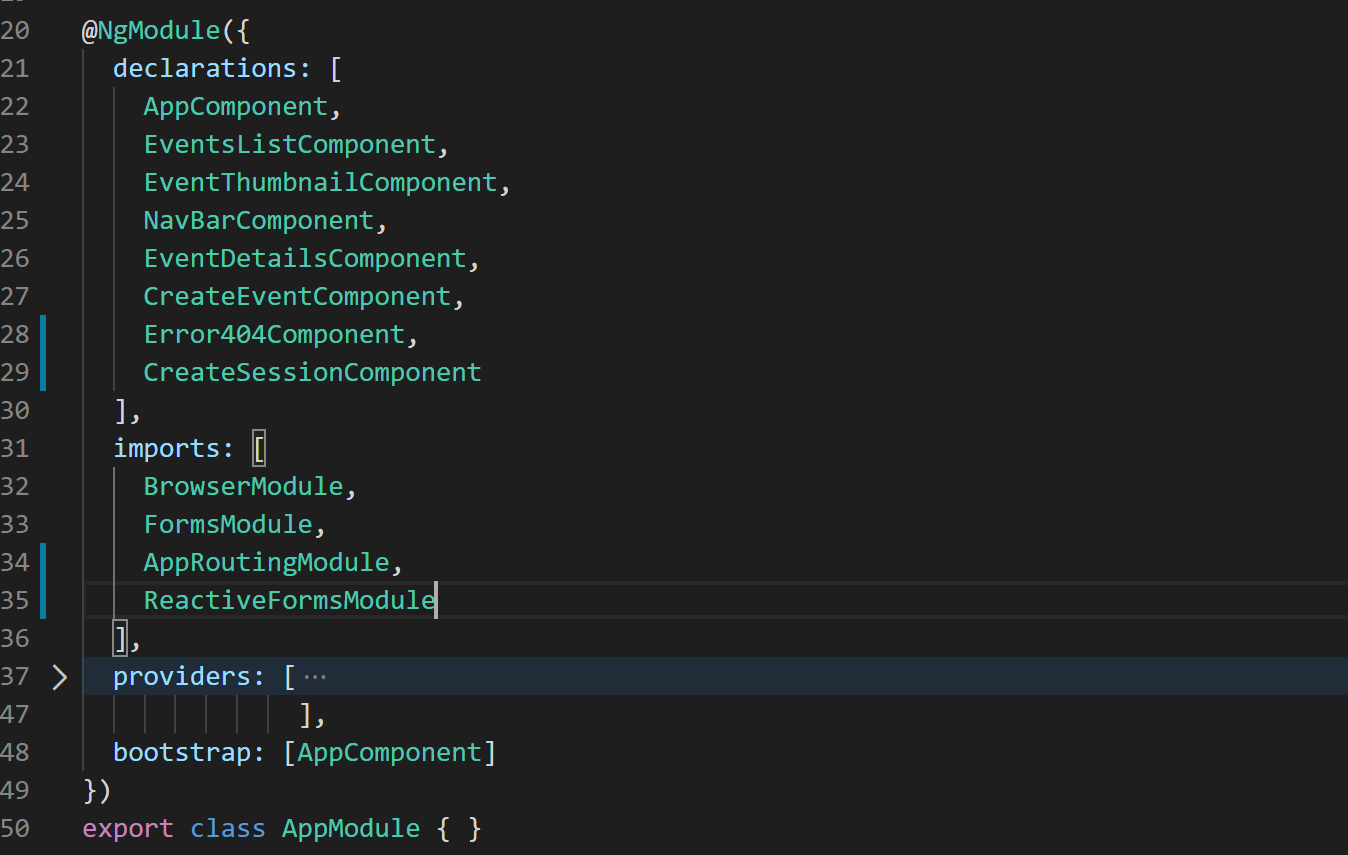
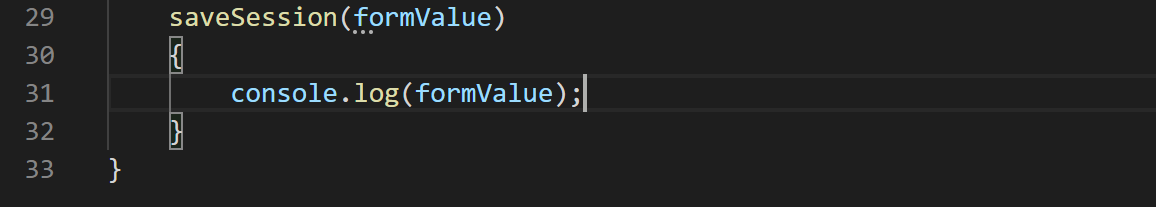
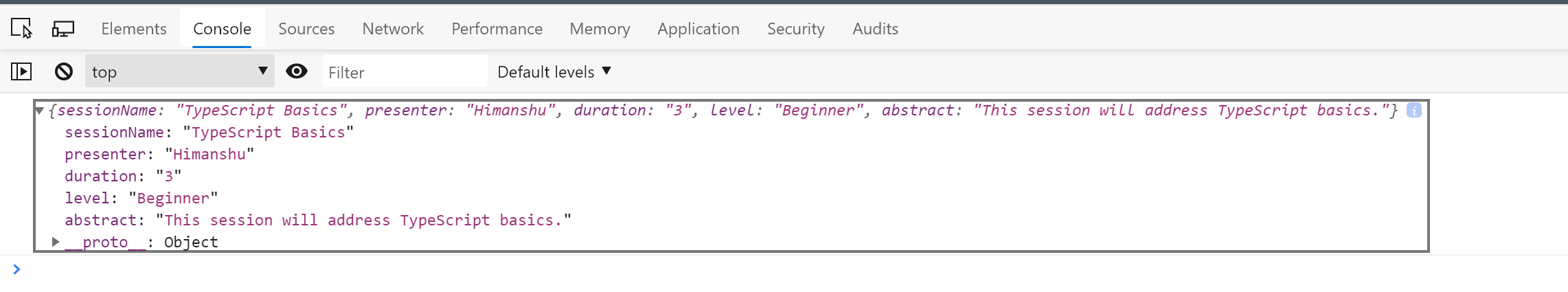
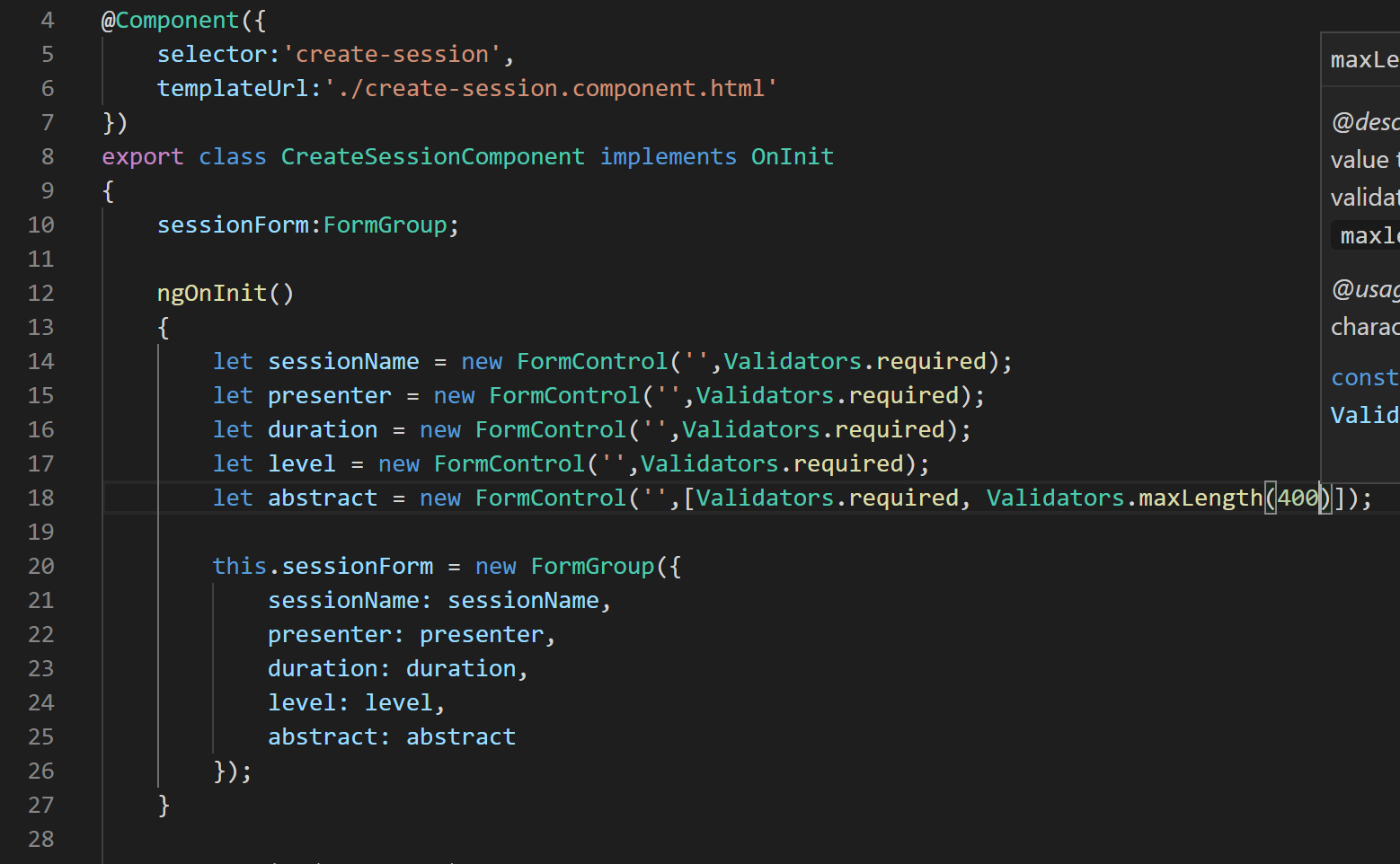
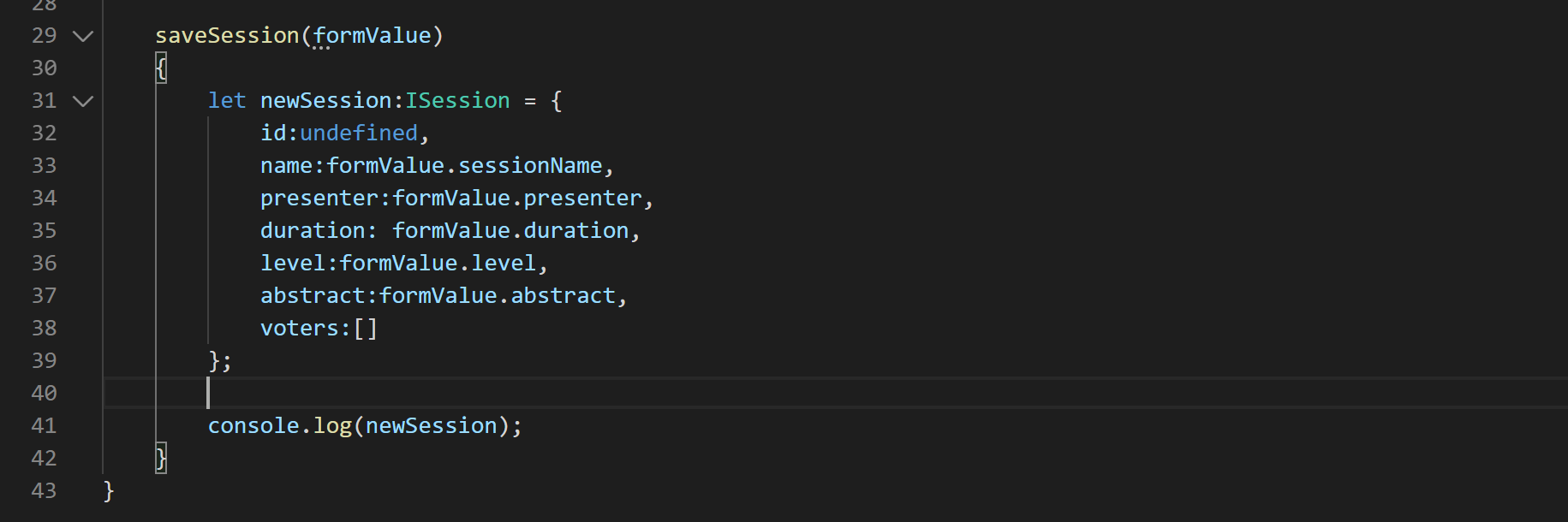
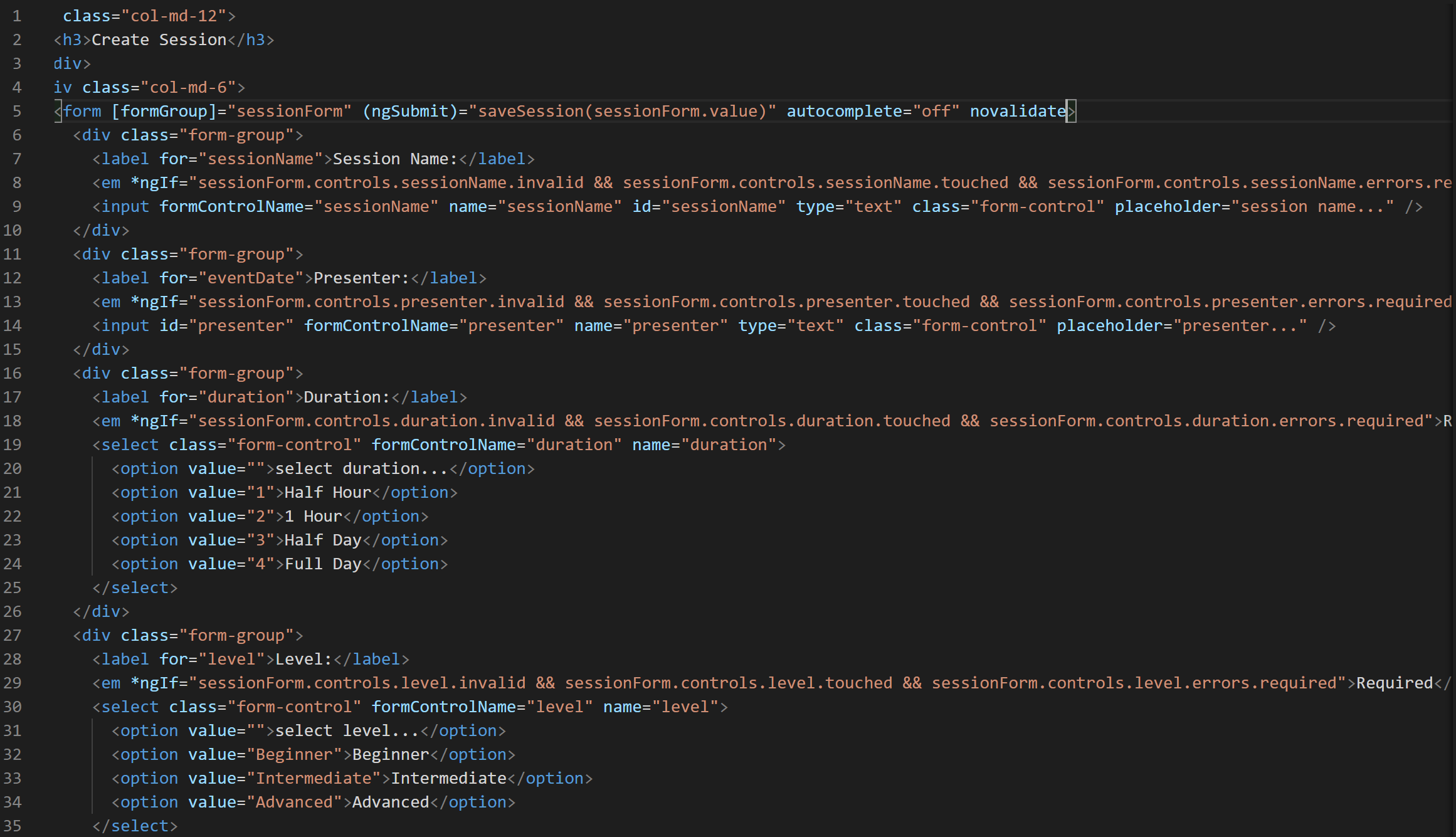
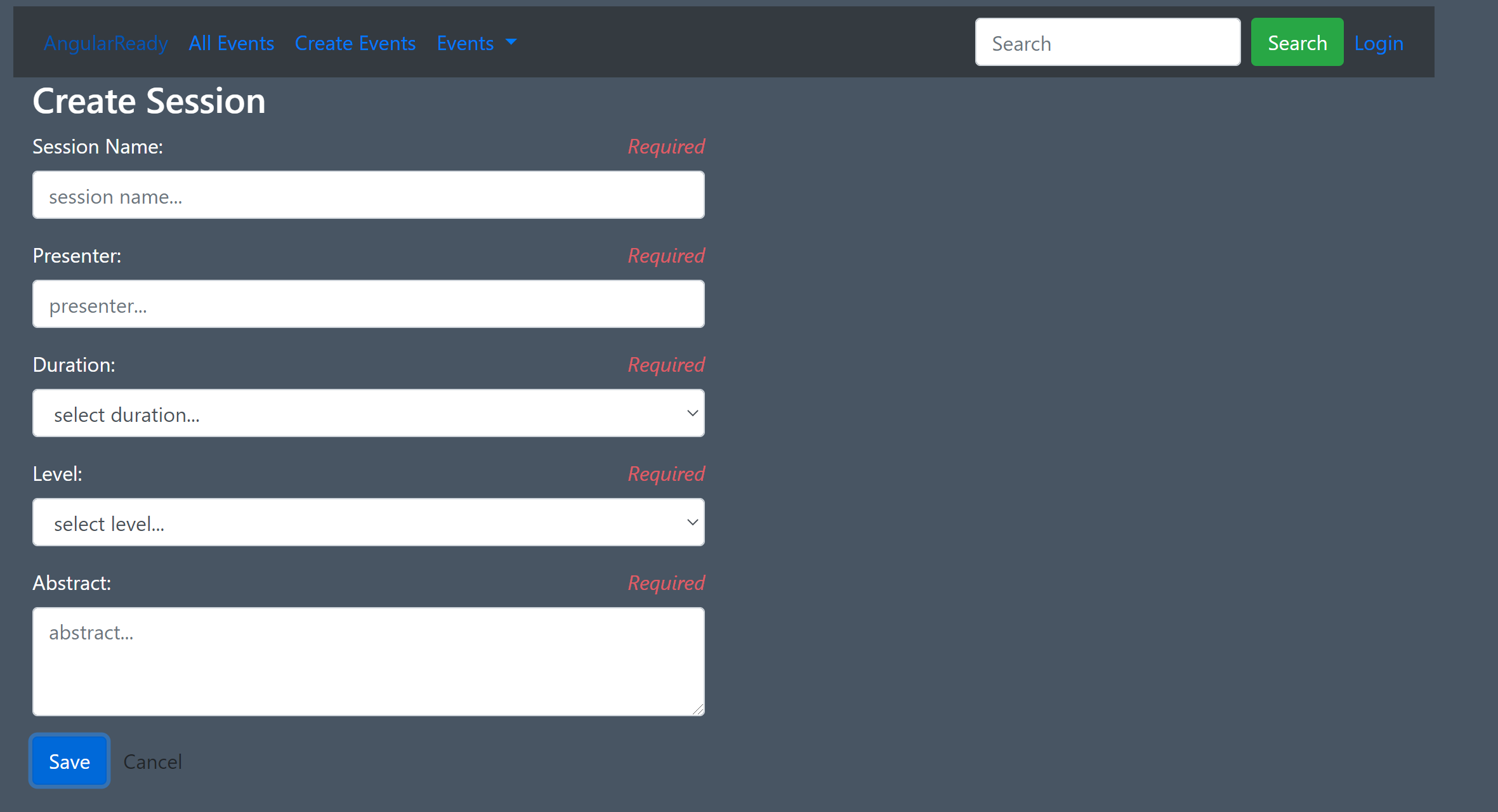
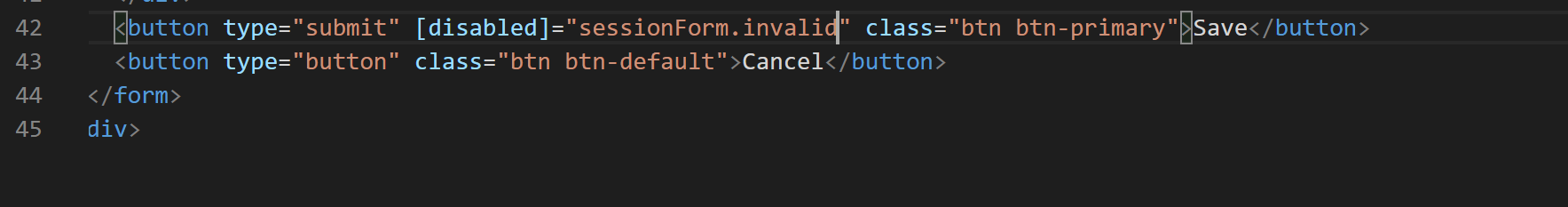
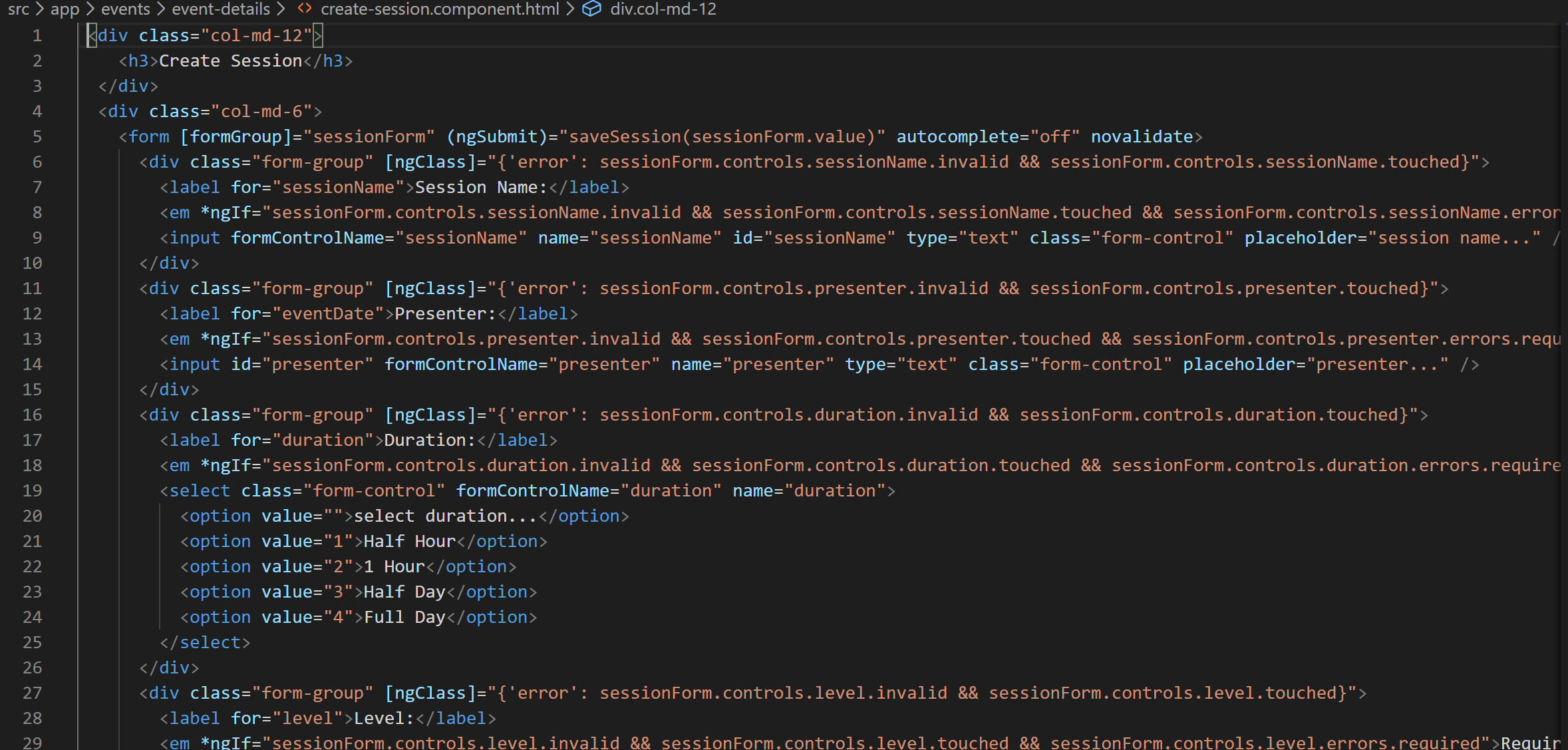
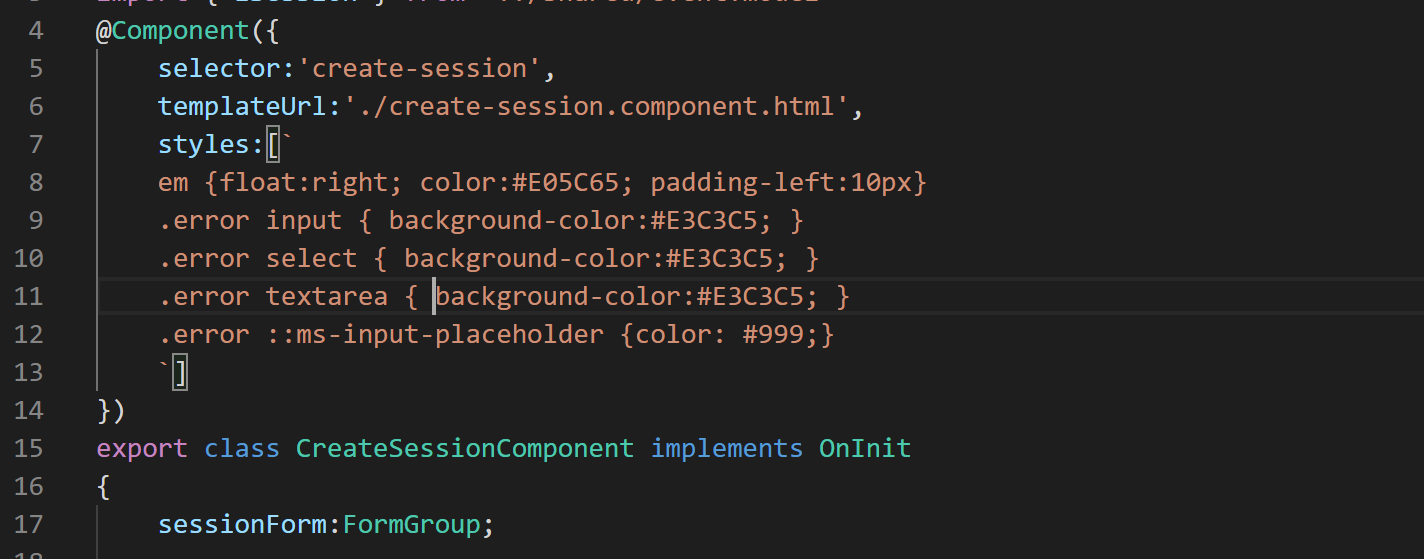
1. Now lets learn to create a complex reactive form to add sessions to our events.
2. Right click on “events” folder and create a file named “create-session.component.ts”.
3. Add basic component shell to it like below:
4. 
5. Lets create the template file we specified above in “event-details” folder.
6. Lets grab the HTML for this file from “misc” folder of cloned or downloaded repo.
7. Now lets add this component in declarations array in our app module like below:
8. 
9. Now lets add a route for it like below in our app-routing.module.ts:
10. 
11. We should be able to hit this newly added route now so lets go take a look. It should look like below:
12. 
13. Now lets define our reactive form in our ngOnInit method like below:
14. 
15. Now lets wire these up with HTML template like below:
16. 
17. Now if you notice the VS Code terminal we are getting an error saying that “formGroup” is not a property of “form” element. That is because we have not imported ReactiveFormsModule in our app module. So lets do that like below:
18. 
19. Alright with that error gone lets define the “saveSession” method in our “create-session.component.ts” like below:
20. 
21. With all errors gone from the VS Code terminal now lets see if the values are getting logged to console successfully or not.
22. So we can see that values are getting logged like below:
23. 
24. Now lets add some default value and validators to our form controls in “create-session.component.ts” file like below:
25. 
26. Now lets take advantage of type checking by importing the ISession interface and creating a new object and filling values like below:
27. 
28. Now lets add our error messages in our create-session.component.html file:
29. 
30. Now if we look at the form in our browser and invalidate all the fields we should see error messages like below:
31. 
32. Now lets disable the save button when form is invalid like below:
33. 
34. Now lets apply “error” class to all the “div” elements so that fields are highlighted in red like below:
35. 
36. Alright now we have some new control types like a ‘select’ dropdown and a textarea so we need to add error classes for them in our styles array in our “create-session.component.ts” like below:
37. 
38. Now if you again browse the form and invalidate the fields the form should look like below:
39. 