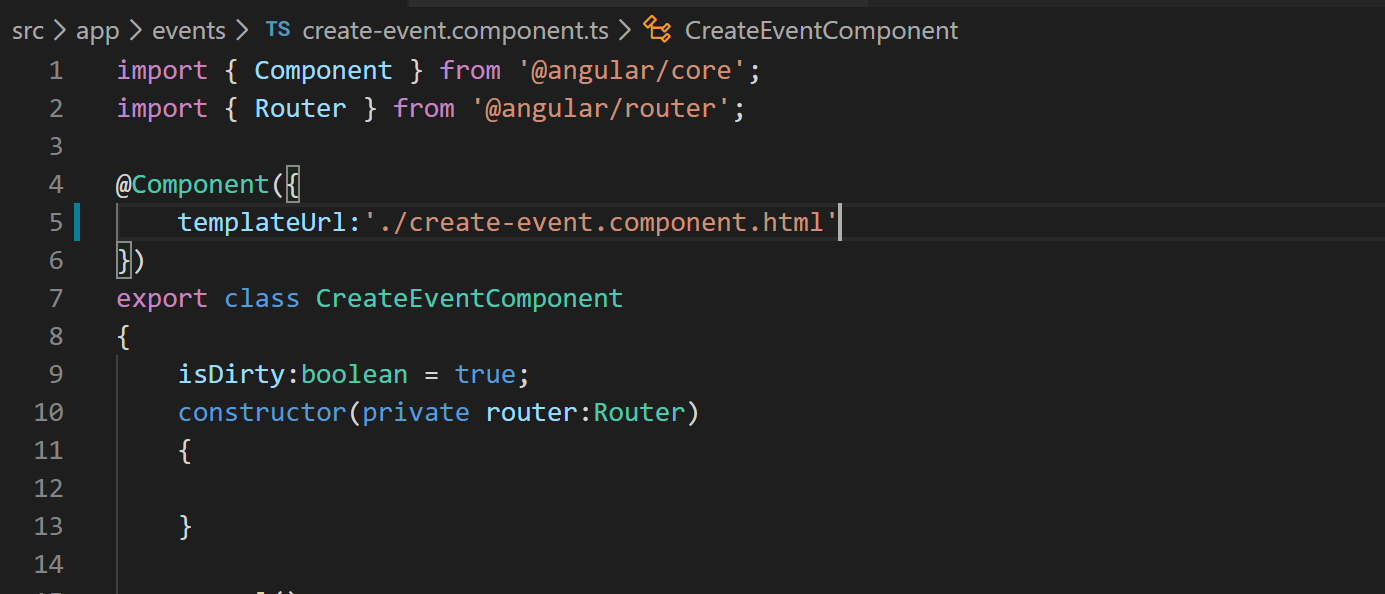
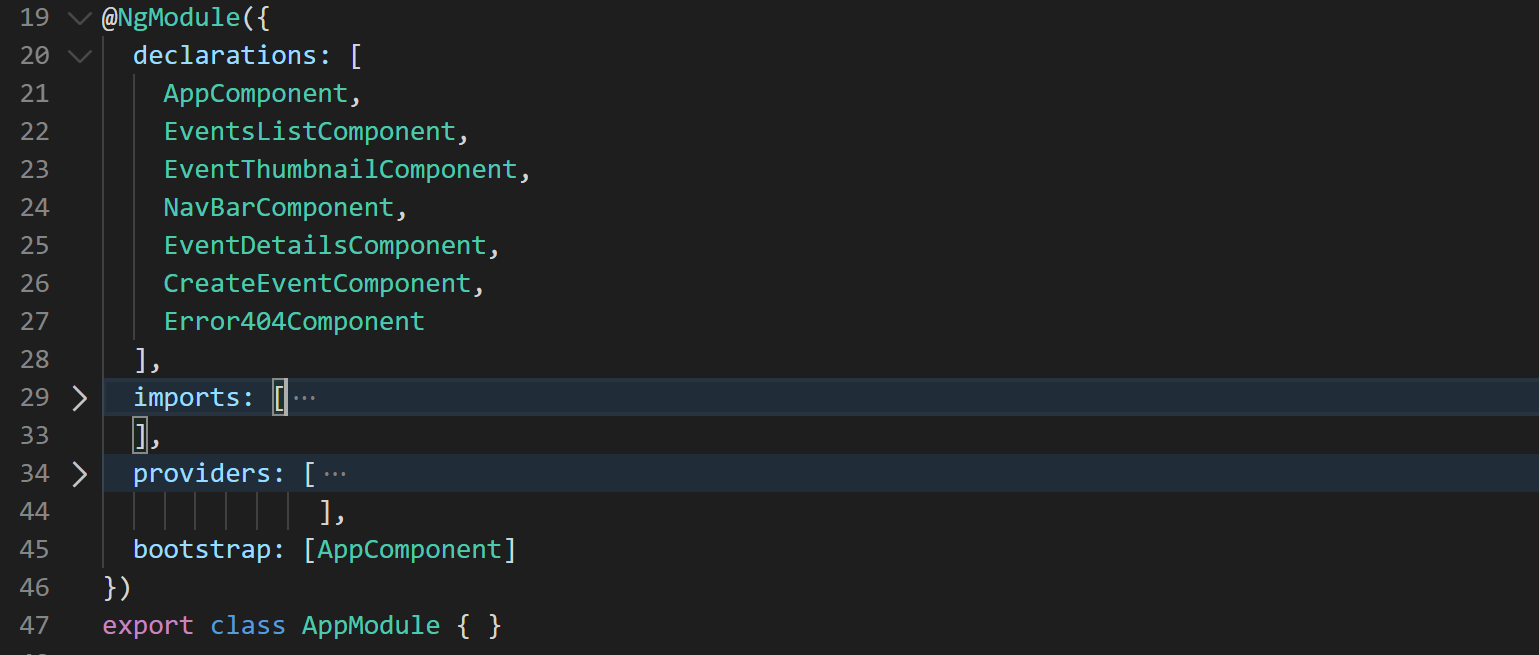
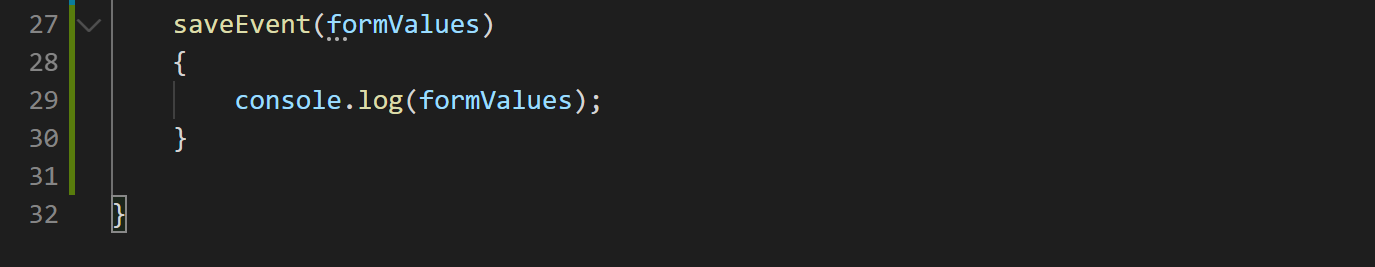
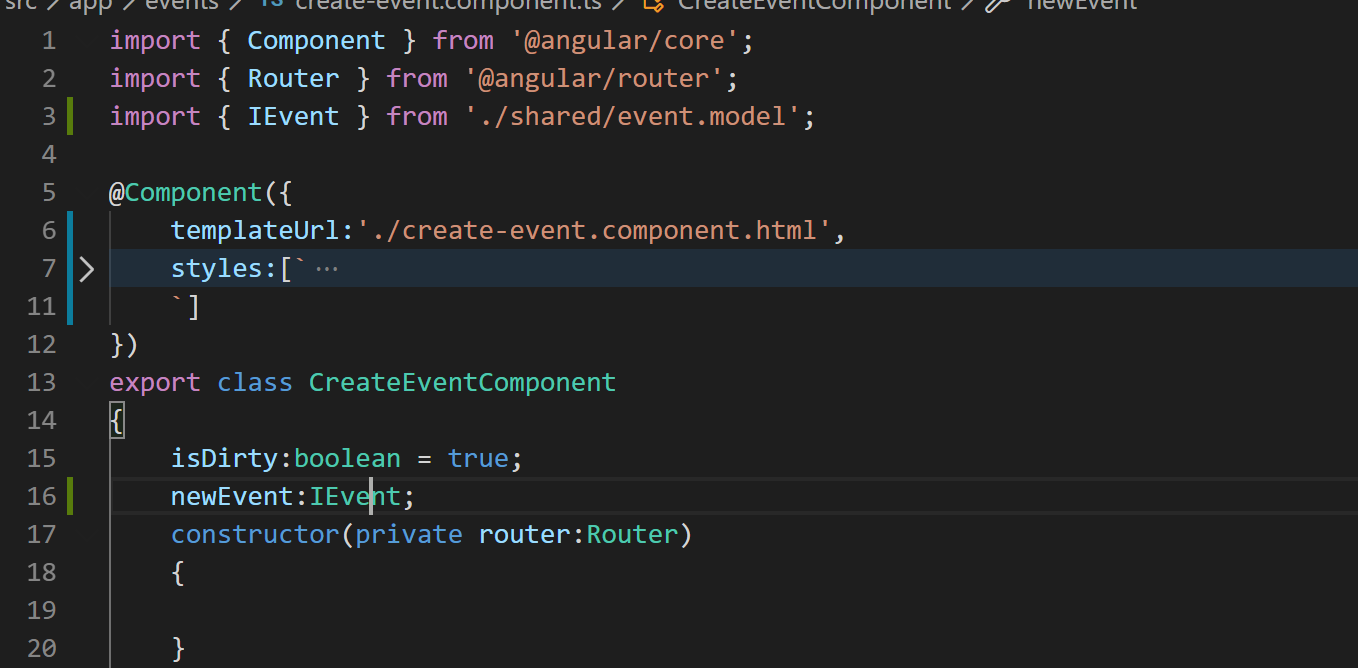
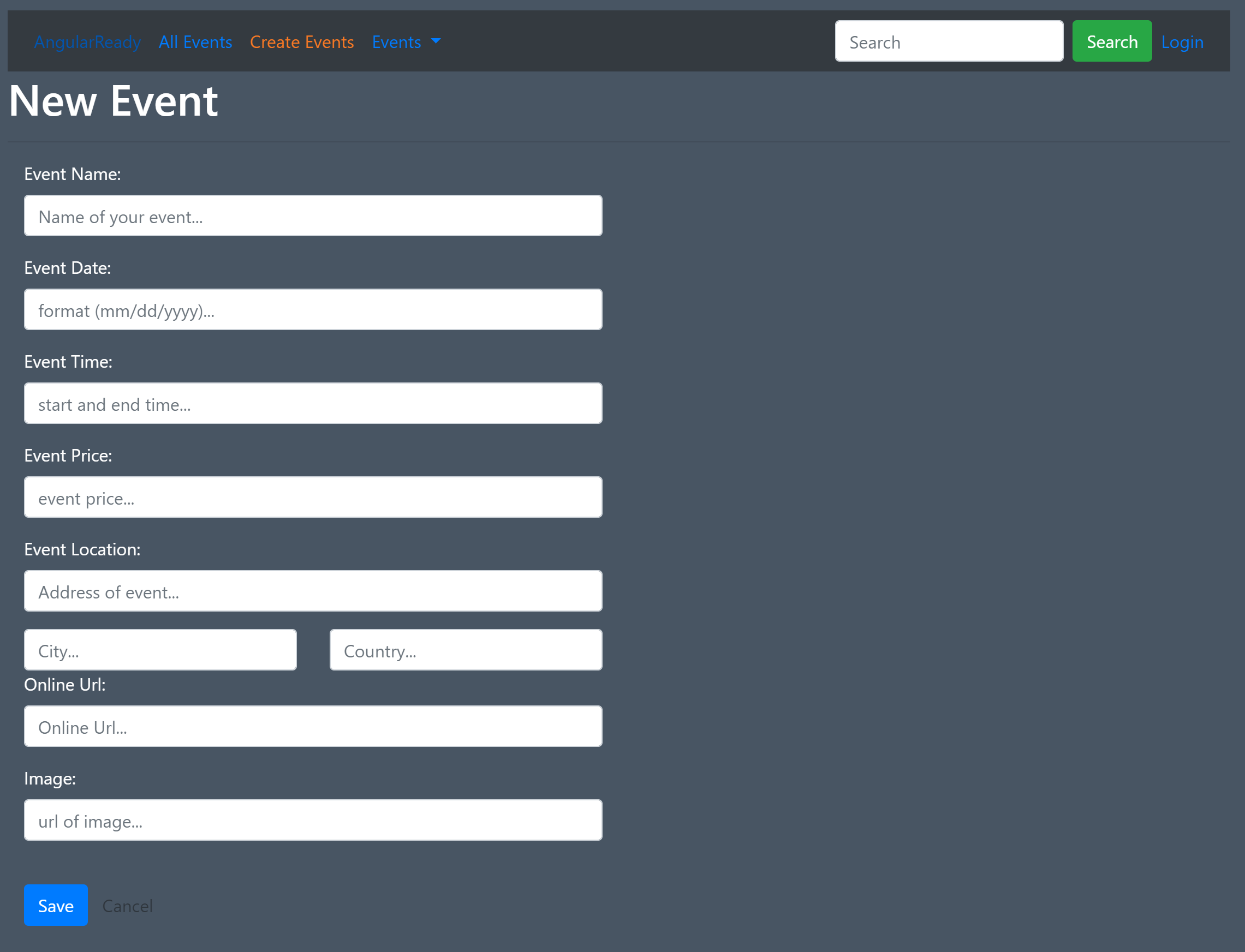
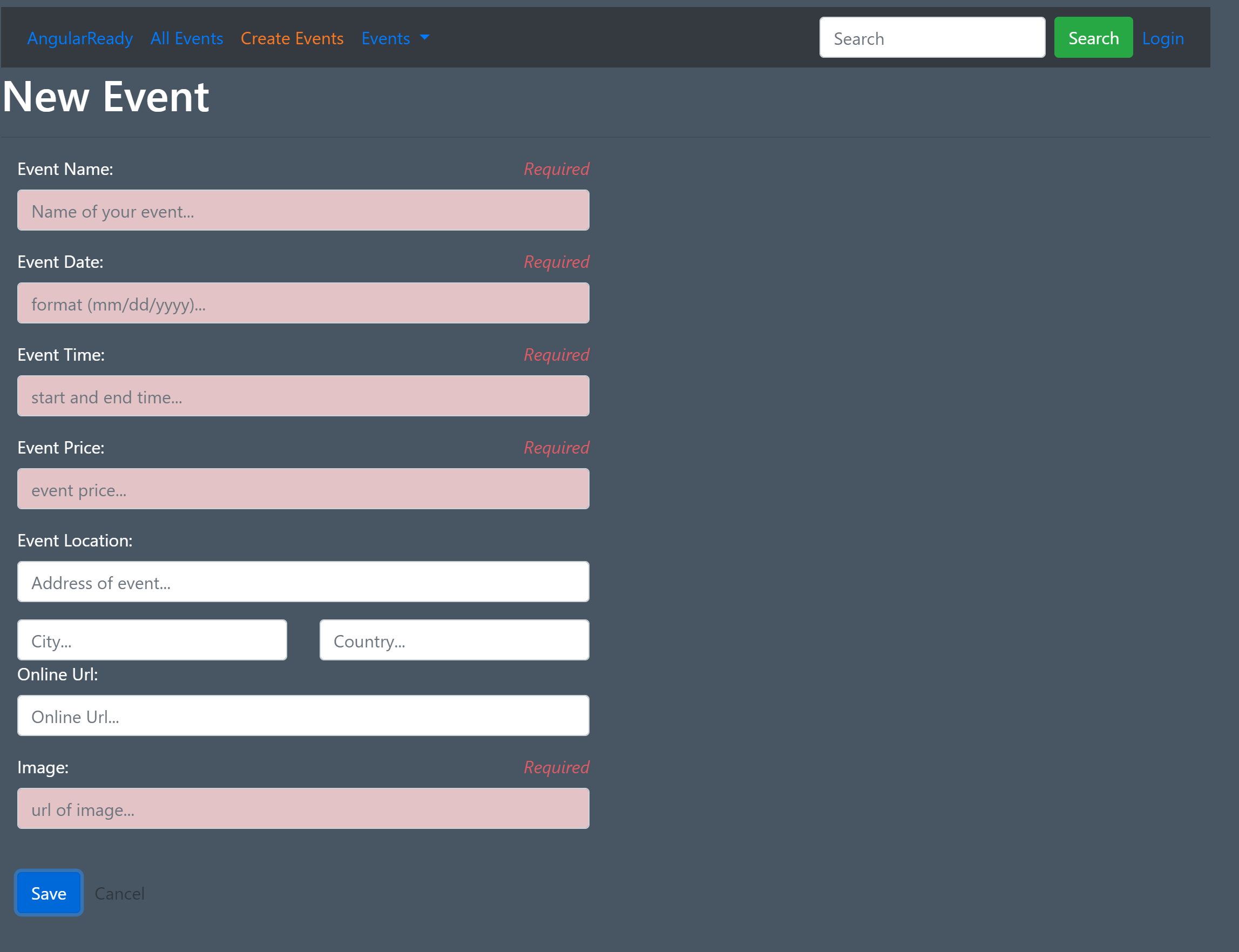
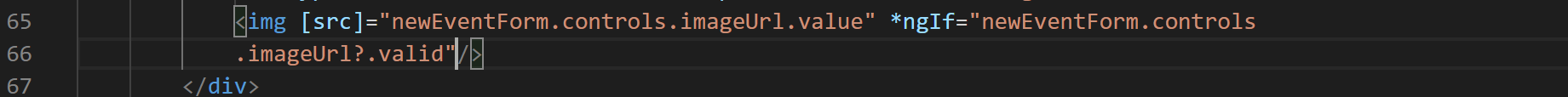
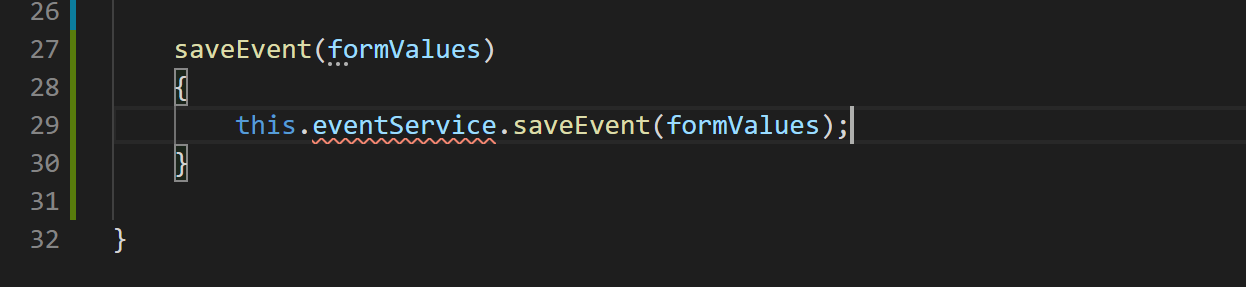
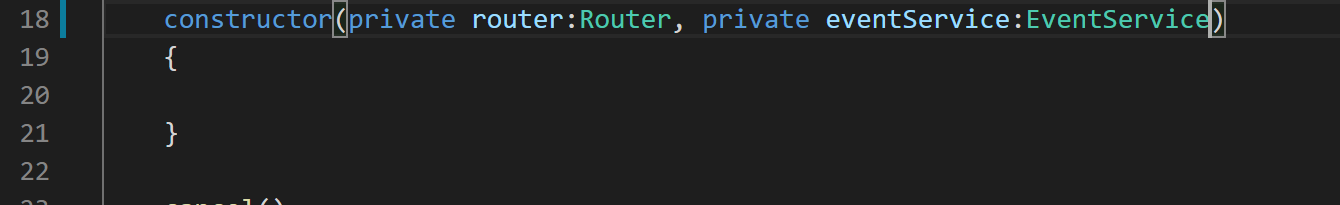
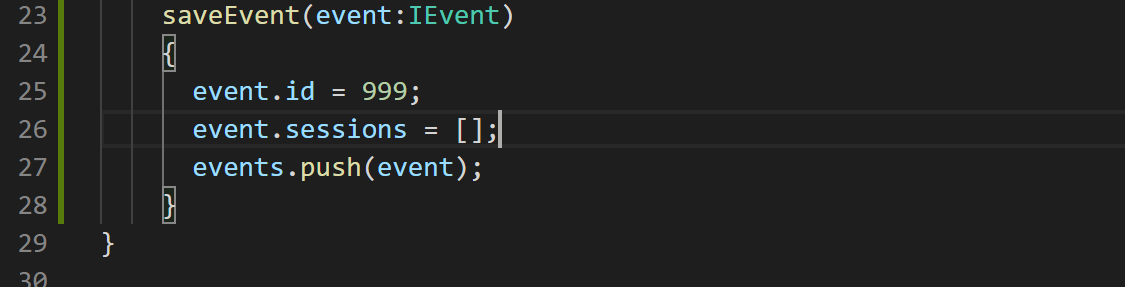
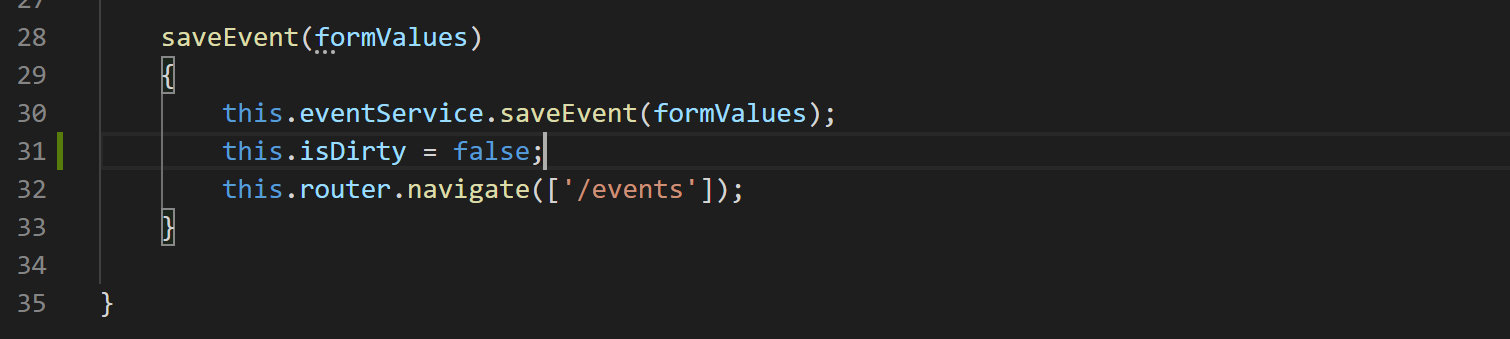
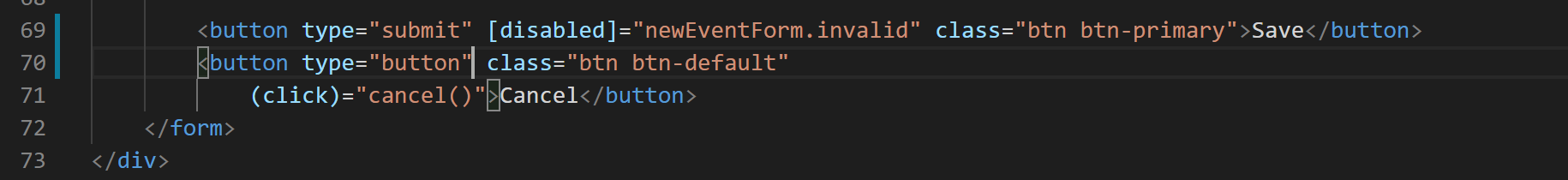
1. So far the forms that we have created using either template based approach or model based approach for very simple forms. Now let's try to create a complex form.
2. We will create a complex form to create a new event. So open the file “create-event.component.ts” and remove the in line HTML template and replace it with a templateUrl property like below:
3. 
4. Now lets create the “create-event.component.html” file in “events” folder.
5. Lets grab the HTML for it from the “misc” folder or cloned or downloaded repo.
6. Now lets import “FormsModule” in our app module so that we can remove the error that we are noticing in the VS Code terminal. So modify the “app.module.ts” like below:
7. 
8. Lets copy the error styles from “profile.component.ts” file and paste them here.
9. Now lets create the “saveEvent” method and console log the formValue that we are getting from the form like below:
10. 
11. Now you can notice in the VS Code terminal that we are getting bunch of errors for the ngModel properties that we have defined in our template. So lets bind them to an IEvent type property. For that we will have to update all the property binding like below:
12. 
13. Now lets define the property on our “create-event.component.ts” file like below:
14. 
15. Now if you go to our application and browse the event creation form it should look like below:
16. 
17. Now if you just tab out of all the fields without entering any value you will notice that all validators are firing and form looks like below:
18. 
19. If you fill out all the fields in the form and click save. You will notice all values logged into the console like below:
20. 
21. If you notice the console output you will notice that fields related to location are not inside location object which does not match our IEvent object. So to fix that we just need to wrap the fields we want in an object in something called “ngModelGroup” like below:
22. 
23. Now if you fill in the form again and look at the console output you will notice that fields are now wrapped in a location object:
24. 
25. Now lets try to show a preview of the image that users specify as a logo for the event. Lets bind the image tag at the bottom of the form with the value of image URL field and show it only when it is a valid URL like below:
26. 
27. Now lets wire up the saveEvent method with out event service’s save event method like below:
28. 
29. Now lets inject the event service in the component like below:
30. 
31. Now lets add the saveEvent method to our service. So go to “event.service.ts” and add the method like below:
32. 
33. Now lets redirect the user to events page once the event has been saved like below in “create-event.component.ts”:
34. 
35. Now if you fill in the values and click save you should see a new event on events list page like below:
36. 
37. Ok you must have noticed that we are still getting a confirm dialog box asking us if you have saved the event or not. To remove this from appearing we just need to set the dirty state to false in saveEvent method like below in “create-event.component.ts” file:
38. 
39. Ok with that out of the way now lets disable the save button on the form like below:
40. 
41. With that our create event form is complete.