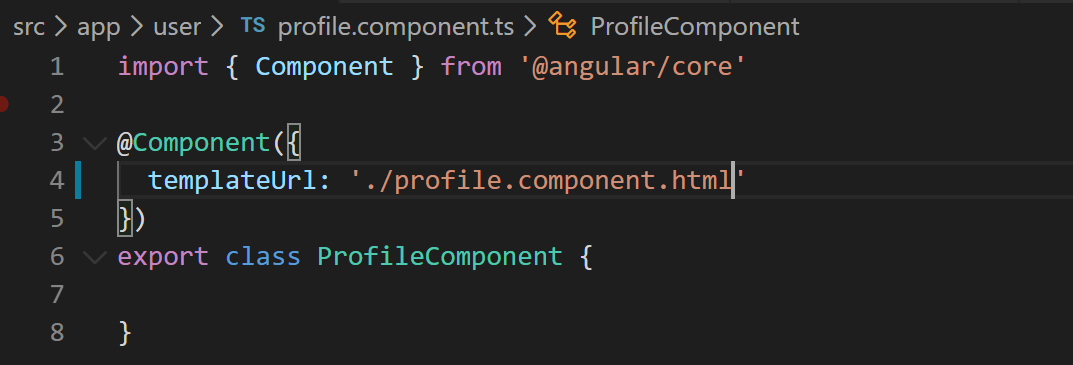
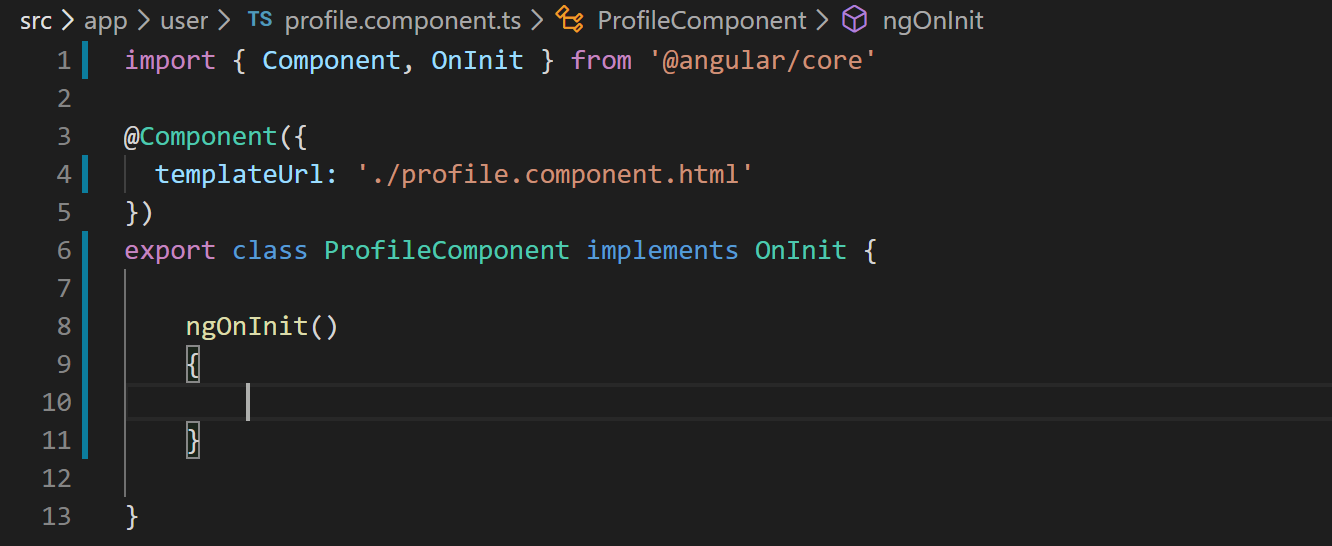
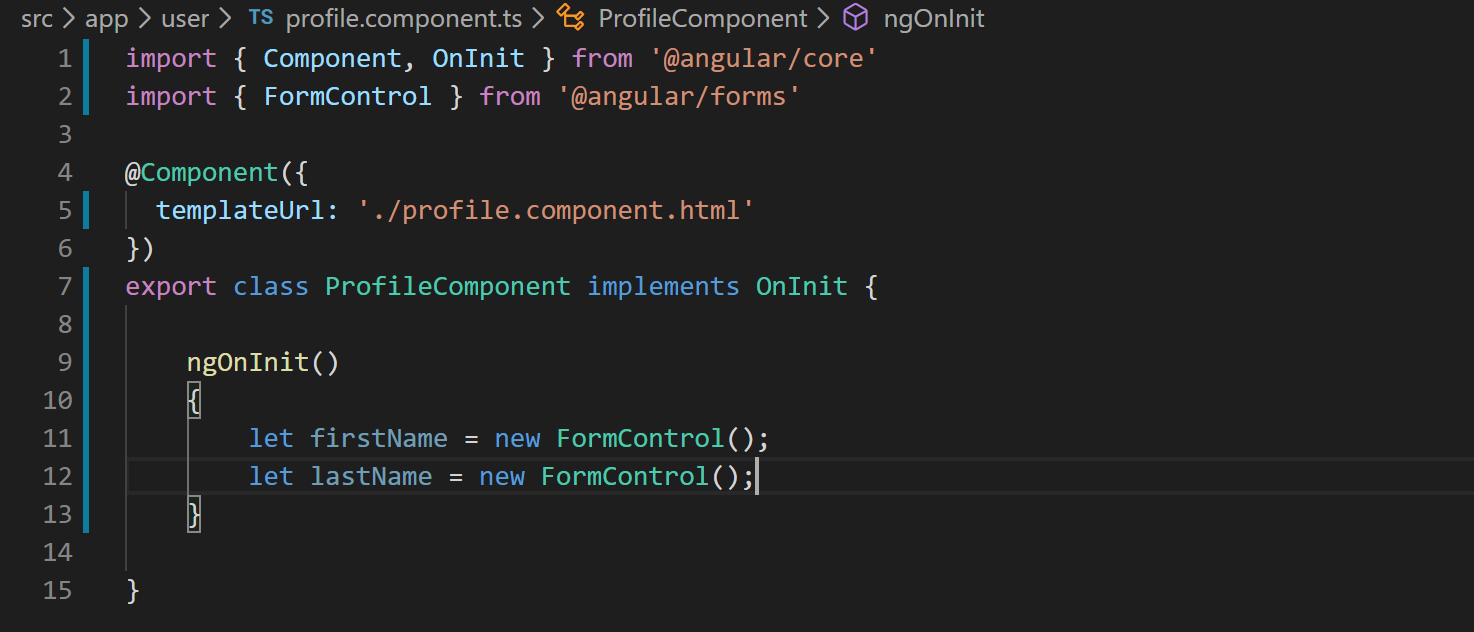
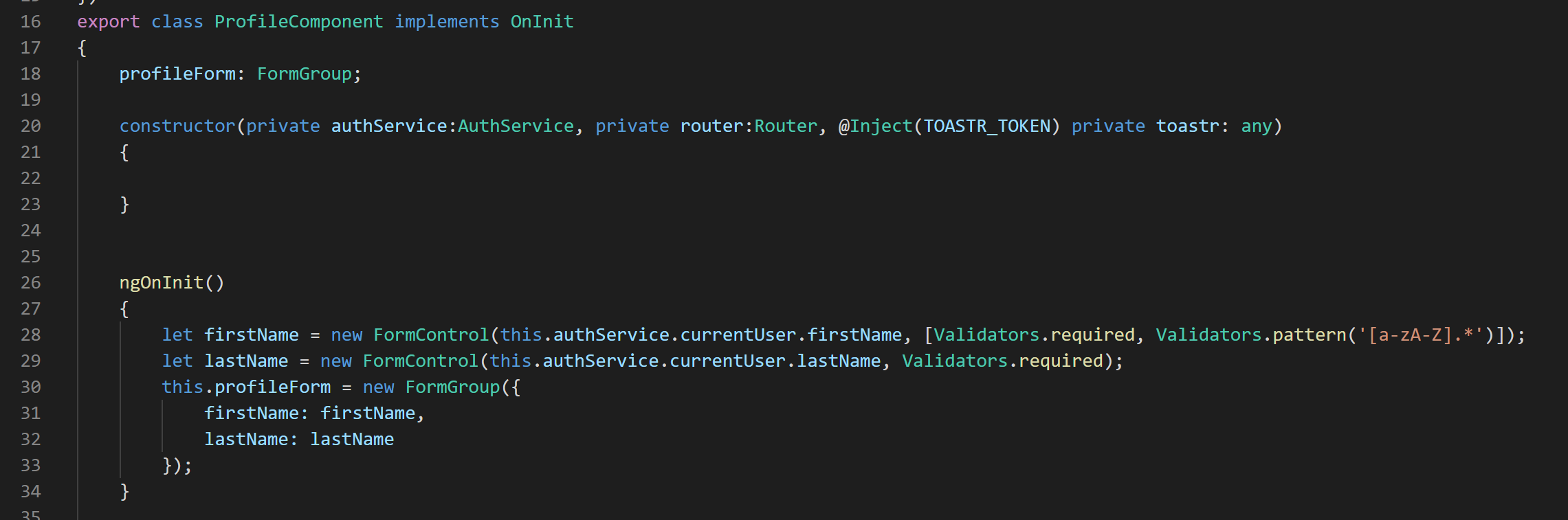
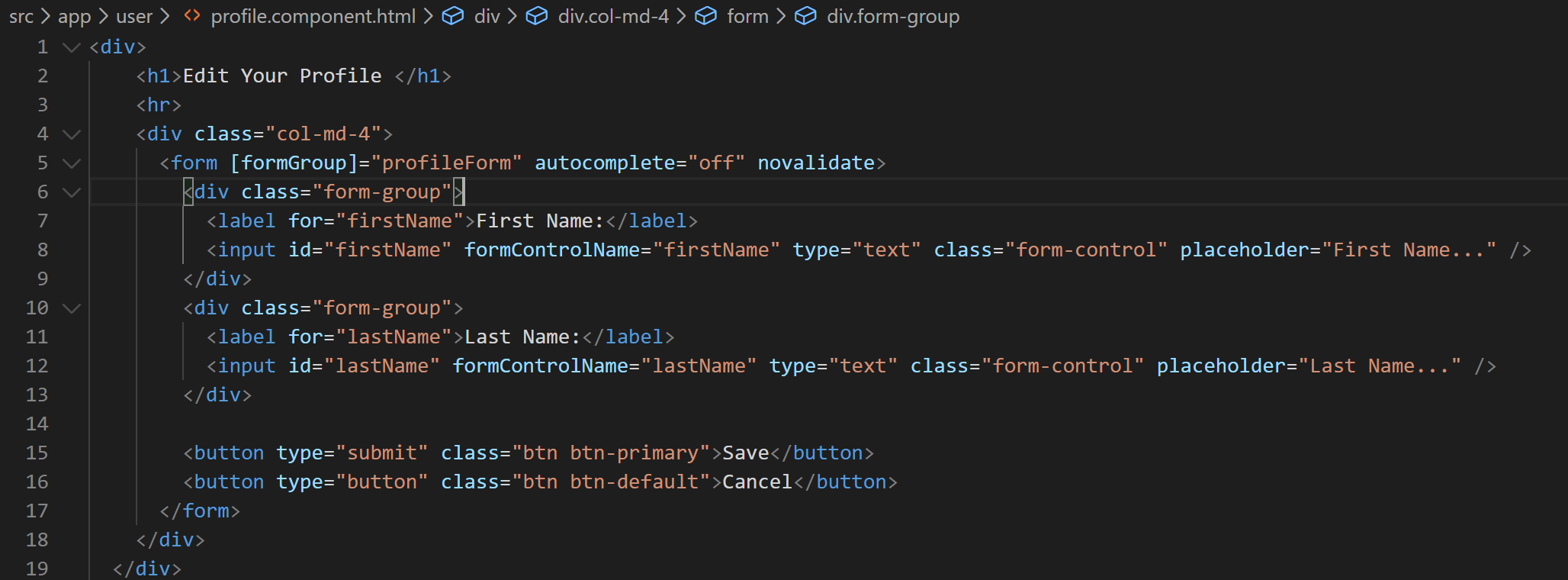
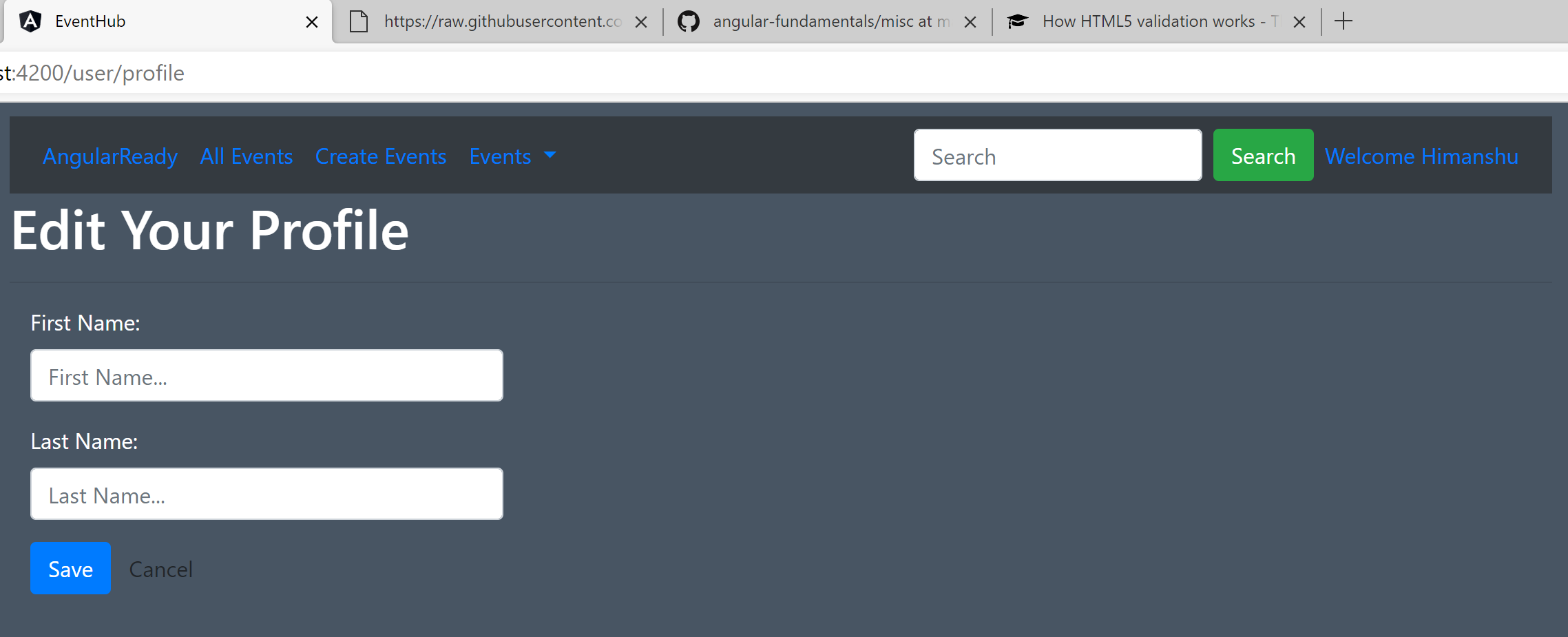
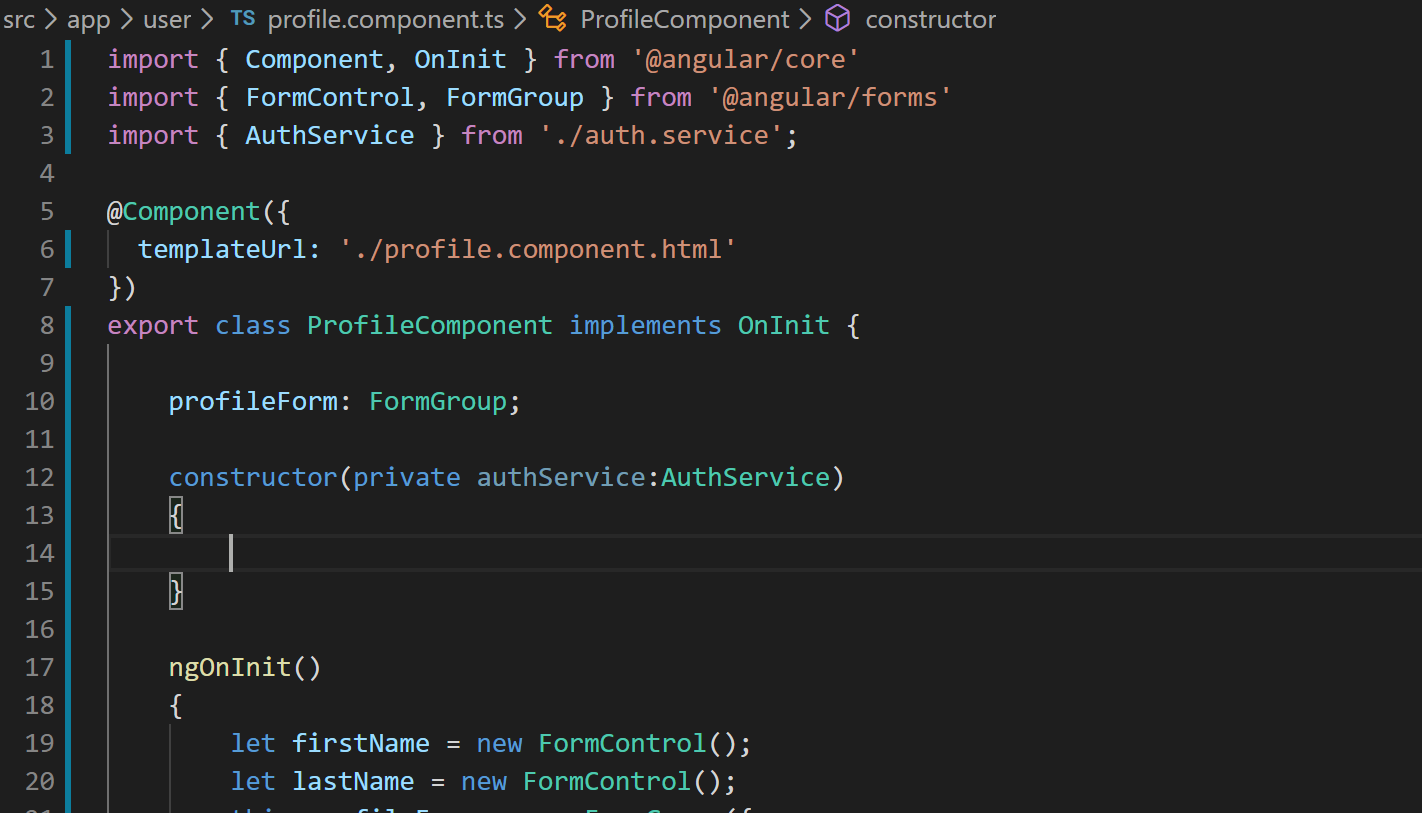
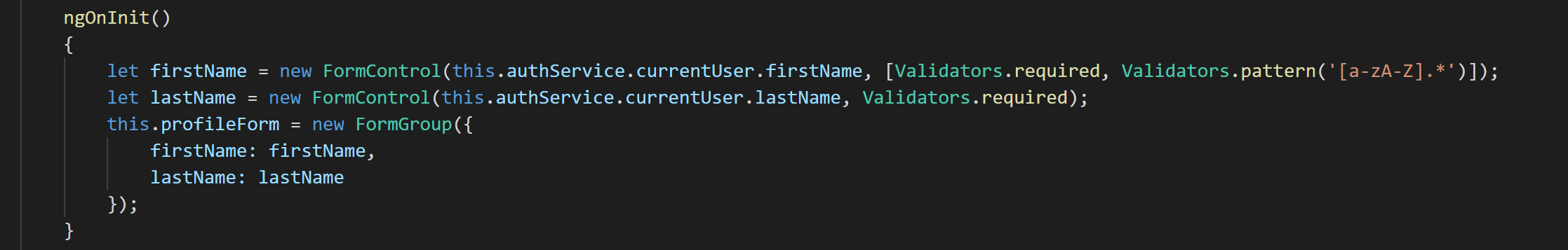
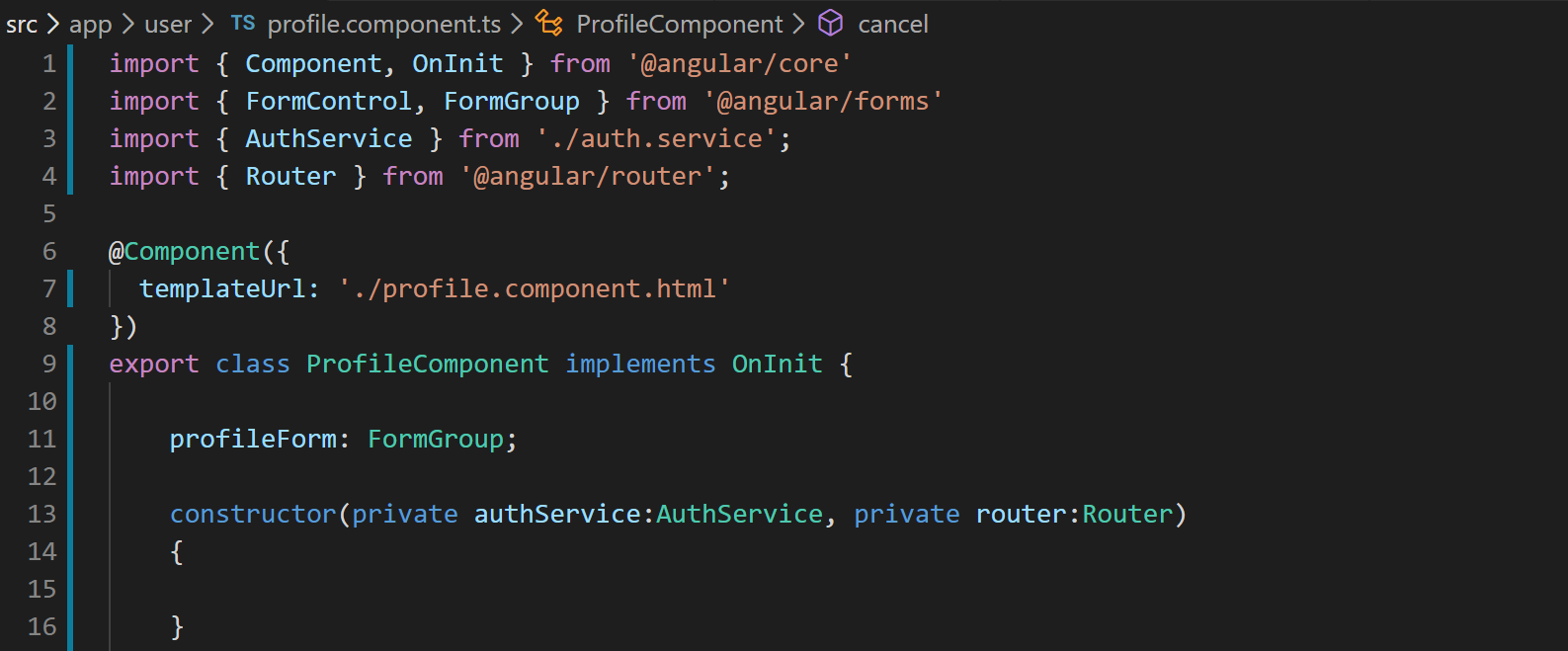
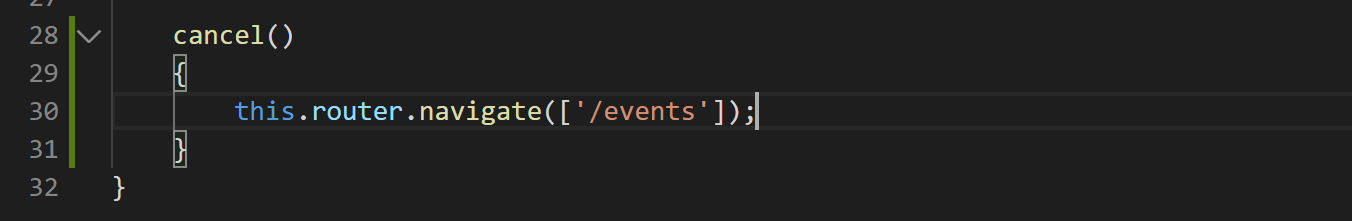
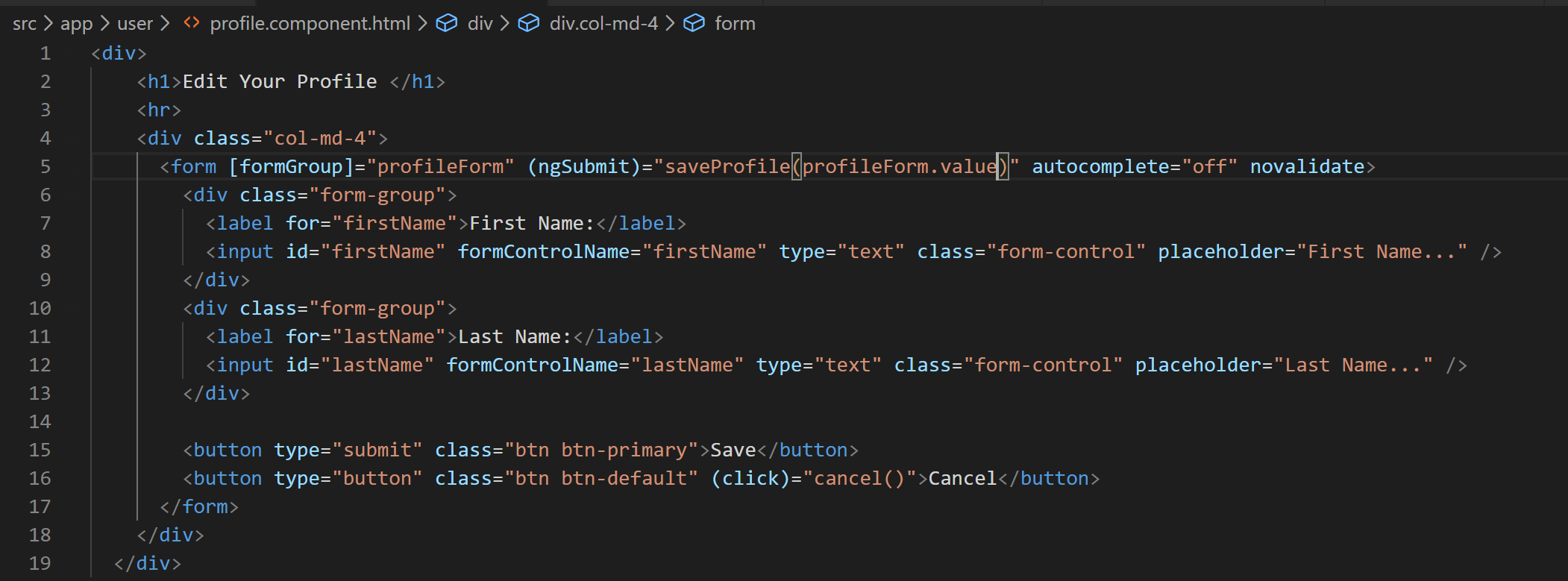
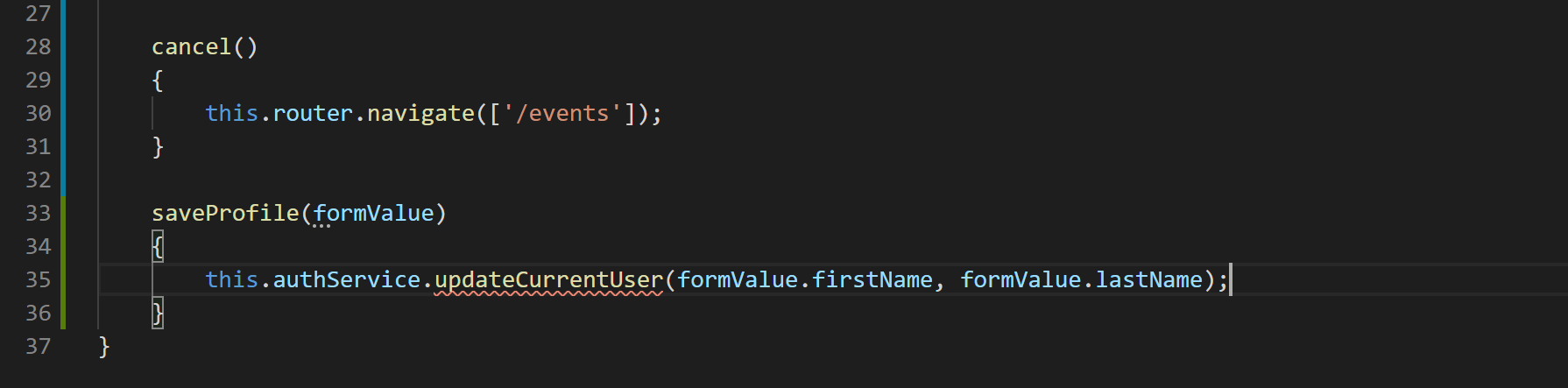
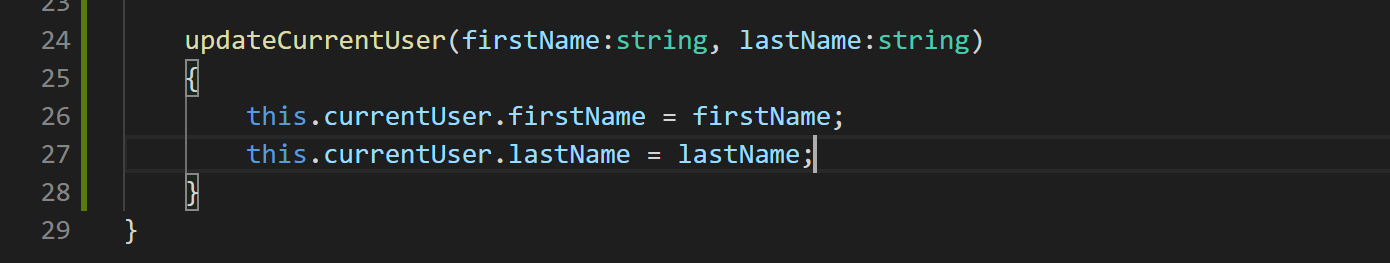
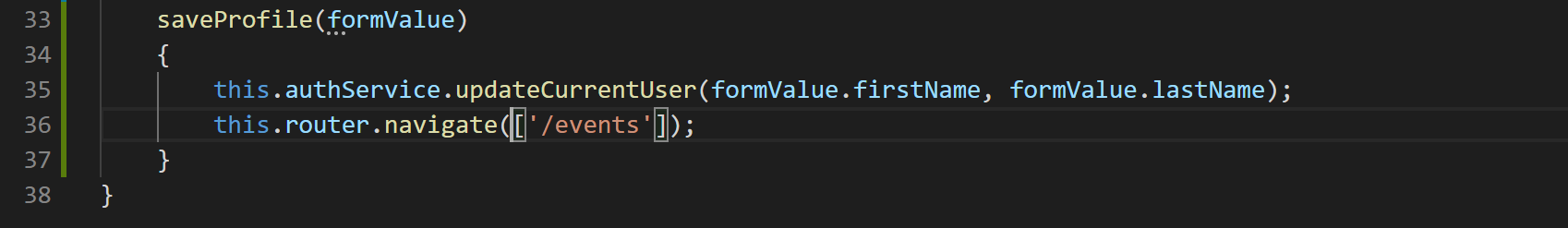
1. Now let's create the reactive form for our edit profile page.
2. Go to file “profile.component.ts” and remove the HTML template that we have in line.
3. Paste this template in a template file called “profile.component.html”. Since file is not there please create the file called “profile.component.html”.
4. Now lets specify the “templateUrl” in the component decorator of profile component in file “profile.component.ts” like below:
5. 
6. Now lets grab the HTML for this template from “misc” folder of our cloned or downloaded repository.
7. We will initialize the form and its validation in OnInit method to lets implement it like below in our “profile.component.ts”.
8. 
9. Now lets create form control for each of our input control on the form. We have two of them first name and last name. So lets create the controls like below:
10. 
11. Now we need to add these controls to a form group object. We do that by using “FormGroup” class like below:
12. 
13. Though we have created the form and its controls we still need to tell Angular to which HTML controls these form controls correspond to. So lets do that using properties like “[formGroup]” and “formControlName” like below:
14. 
15. If you look at VS Code terminal you will notice we are getting an error saying “formGroup” is not a property of “form”. That is because we need to import the “ReactiveFormsModule” in our user module like below:
16. 
17. Now if you go take a look at our application the edit profile form will look like below:
18. 
19. If you notice the form above you will see that even though we are logged in we are not seeing any value in the first name and last name input boxes. So lets take care of that by getting values for current user from the auth service.
20. Inject the auth service in the profile component like below:
21. 
22. Now lets pass the value of first name and last name from current user property of auth service into the FormControl constructor like below:
23. 
24. If you look at the edit profile form now you will notice that values are filled already.
25. Lets us now wire up the submit and cancel buttons. Create an cancel method that will redirect to all events page and bind it to the cancel button in the template. Don’t forget to inject Angular’s router service.
26. 
27. 
28. Lets create a “saveProfile” method on the component like below:
29. Now lets wire it up with the HTML template using (ngSubmit) like below:
30. 
31. Ok now lets implement the saveProfile method in our component by calling a method on auth service to update the currently logged in user property like below:
32. 
33. 
34. Update the saveProfile method to redirect the user to events page like below:
35. 
36. Now if you go to the edit profile page and update the first name of the user and click save you will be redirected to the events page and the first name of the user will be updated in the navigation bar automatically.