Forms

Q.1) What is Form in Angular ?

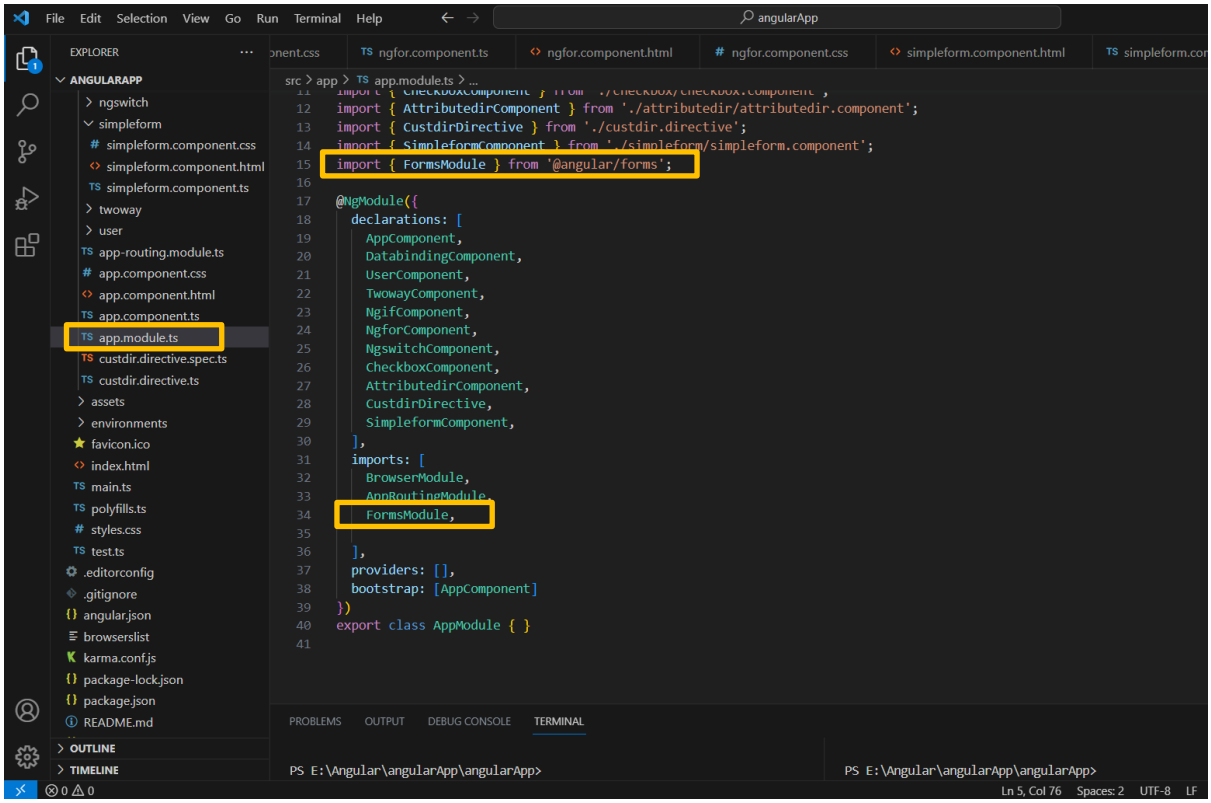
- Forms is use to handle the user input data.
- We need to import **FormsModule** in **app.module.ts.**
- Angular support two types of forms
- A) Template Driven Form
- B) Reactive Form

Q.2) What is Template Driven Forms?

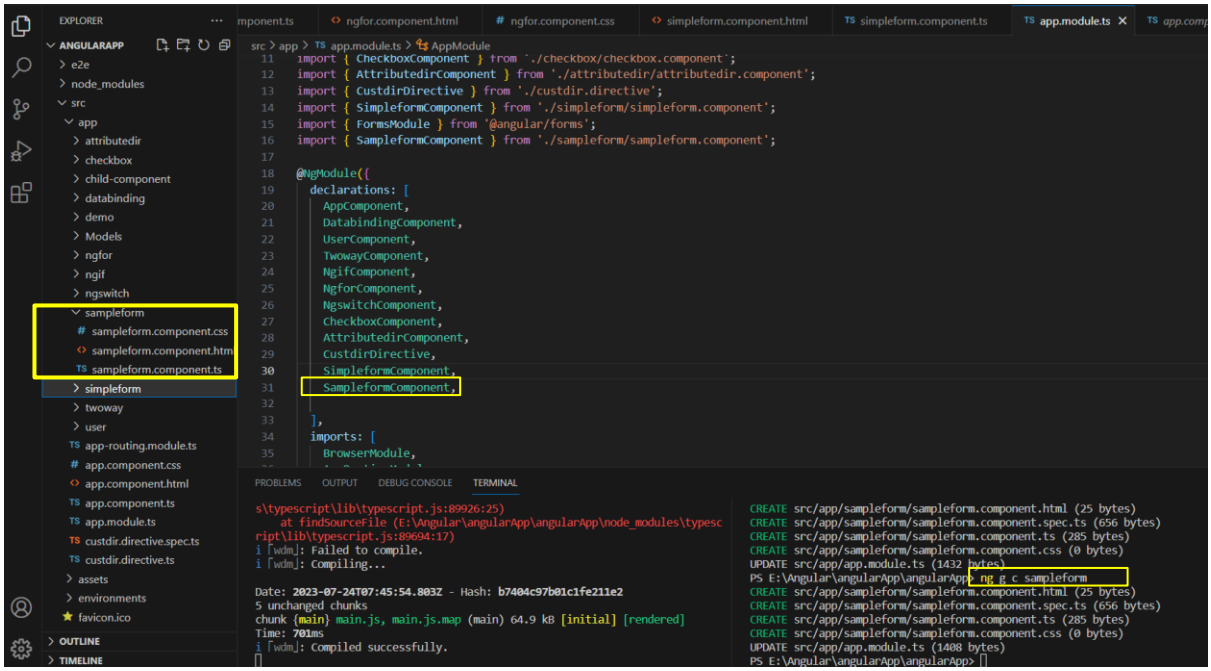
- It means if we have a Simple form that time we can use Template Driven Form.
- It is mainly used for creating a simple form Application.
- It is created using directives in the template.
- It is Supported Two Way Data Binding.
- It is Asynchronous.
- Most of the code we write in HTML.

Q.3) Create a Simple Login Form Using Template Driven Forms?

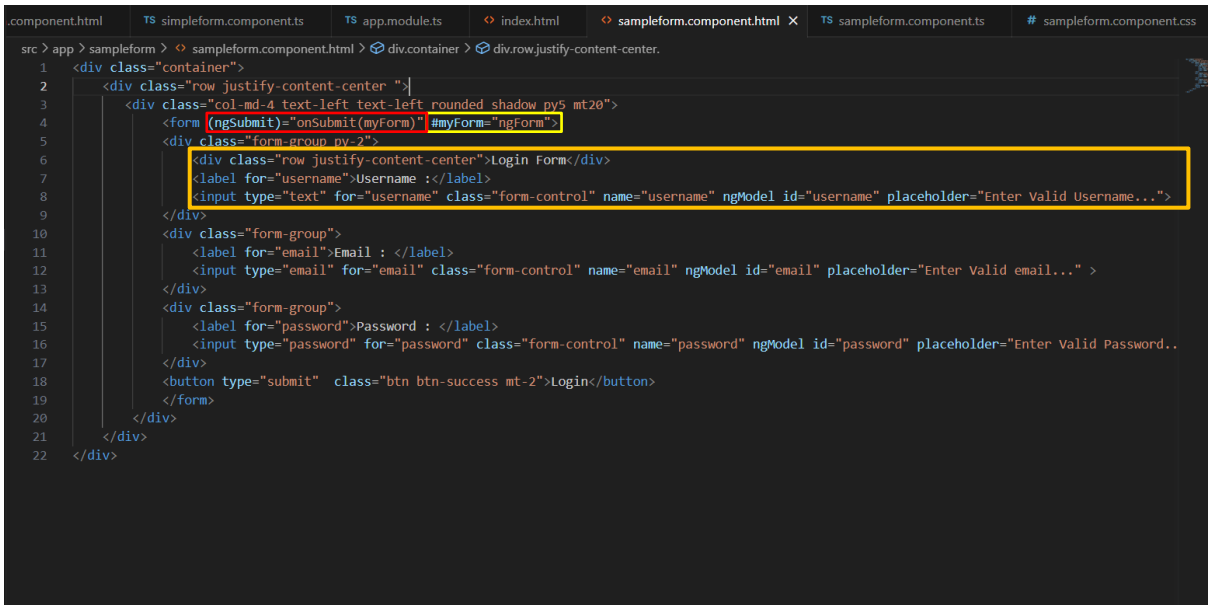
- First we need to import **FormsModule** in app.module.ts → imports: **[FormsModule]**.



- For login form we need to create one component **sampleform**.
- For that we use command **ng g c sampleform.**
- Our sampleform component is ready.



- Now create a simple form to display user entered text.
- For that we write our code in **sampleform.component.html.**



- U can see here I'm taking 3 formControls namely username, email and password.

- **class=”form-control”** →
  - Used on input, textarea, and select elements to span the entire width of the page and make them responsive.
- **ngModel →** ngModel directive creates an instance of form controls.

- **name = “username” →** As you see in the above form design that we are given name attribute to every element because if we are using ngModel with ngForm and not given name attribute that it gives an error like below.

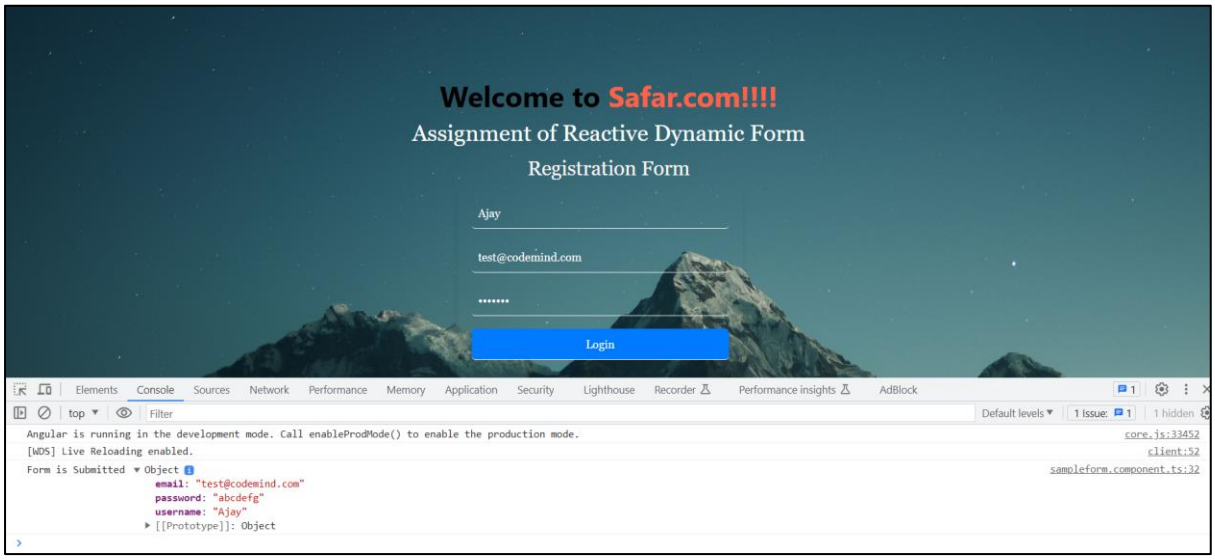
```
▼ERROR Error: If ngModel is used within a form tag, either   core.js:5980
the name attribute must be set or the form
      control must be defined as 'standalone' in ngModelOptions.

      Example 1: <input [(ngModel)]="person.firstName" name="first">
      Example 2: <input [(ngModel)]="person.firstName" [ngModelOptions]="
{standalone: true}">
    at Function.missingNameException (forms.js:4908)
    at NgModel._checkName (forms.js:5228)
    at NgModel._checkForErrors (forms.js:5211)
    at NgModel.ngOnChanges (forms.js:5150)
    at NgModel.rememberChangeHistoryAndInvokeOnChangesHook (core.js:1471)
    at callHook (core.js:2490)
    at callHooks (core.js:2457)
    at executeInitAndCheckHooks (core.js:2408)
    at refreshView (core.js:9207)
    at refreshComponent (core.js:10358)
```

  - **name attribute** helps us to register that form control on the forms.
  - **ngSubmit() :** The ng-submit directive specifies a function to run when the form is submitted.
  - **OnSubmit () :** Is a method which is write in **.ts** file. We are bind our ts file logic to our html form and apply also myForm.
  - **#myForm(NgForm) :**
    - Here we use Template Reference Variable myForm
    - And its denoted as # variableName.
    - Used to access all the properties of any element inside the DOM and here we apply this on form element.
    - So it will access all properties of form.
  - **ngForm** → It will create a top level instance of FormGroup. Also use to export that instance.
  - **So** Focus on this point
  - **<form (ngSubmit)="onSubmit(myForm)" #myForm="ngForm">**
  - ngSubmit → call after click on submit button.
  - onSubmit (myForm) → is a method implemented in **.ts** file and passing values as
  - myForm. → # myForm is a Template reference variable contain all the properties of the applied any element inside the DOM.
  - **ngForm** → Create a top level instance of FormGroup and export that instance using # myForm assign to .ts method onSubmit (myForm) using property binding (ngSubmit).

- Now Need to write a Business Logic in .ts file.

```
src > app > sampleform > TS sampleform.component.ts > SampleformComponent
1  import { Component, OnInit } from '@angular/core';
2  import { NgForm } from '@angular/forms';          <-- Here we need to import NgForm form angular/forms
3
4  @Component({
5    selector: 'app-sampleform',
6    templateUrl: './sampleform.component.html',
7    styleUrls: ['./sampleform.component.css']
8  })
9  export class SampleformComponent implements OnInit {
10
11   constructor() { }
12
13   ngOnInit() {
14   }
15
16   onSubmit(form : NgForm){
17     console.log(`Form is Submitted`, form.value);   <-- Using form.value we get form controls values
18
19   }
20
21
```

- NgForm used to import the properties from HTML form.
- After click on submit we get values like.



4) What is ngModel?

- ngModel will create an instance of form controls.
- Used to bind the control to particular form group we need to use ngModel directive.

5) What is NgForm ?

- NgForm directive is used with HTML <form> tag.
- It will create a top level instance of FormGroup.

6) What is name in form ?

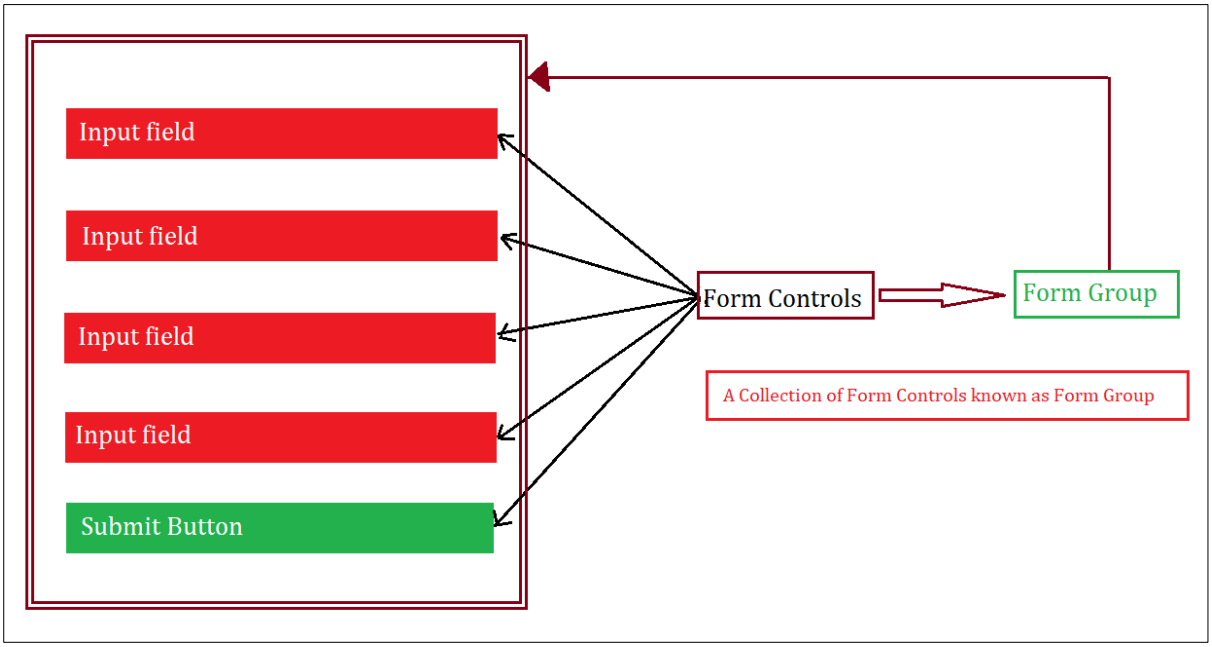
- To identify the control of form we use name attribute.
- If we want to use Template Driven Form then we need to use ngModel with ngForm tag we need to supply to name attribute to register control.

7) What is formControls ?

- Form controls is a class that can hold the data values and the validation information's of any form element.
- ngModel provide FormControls.

8) What is FormGroup ?

- A collection of FormControls is nothing but a FormGroup.
- NgForm provide FormGroup.



9) **What is Template Reference Variable ?**

- used to access all the properties of any element inside the DOM
- syntax #variableName

10) What is Validations ?

- Validations means we are applying conditions whether the user enter correct input format or not.

11) What are the form states/phases ?

- **ng-touched** : The user has interacted with form control.
- **ng-untouched** : The form control has not been interacted with by . any user.
- **ng-formControl** : Form controls hold the user data

- **ng-pristine** : The user has not modified the form control.
- **ng-dirty** : The user has modified the form control.
- **ng-invalid** : The forms controls values does not meet the . Validation rules.