CJC

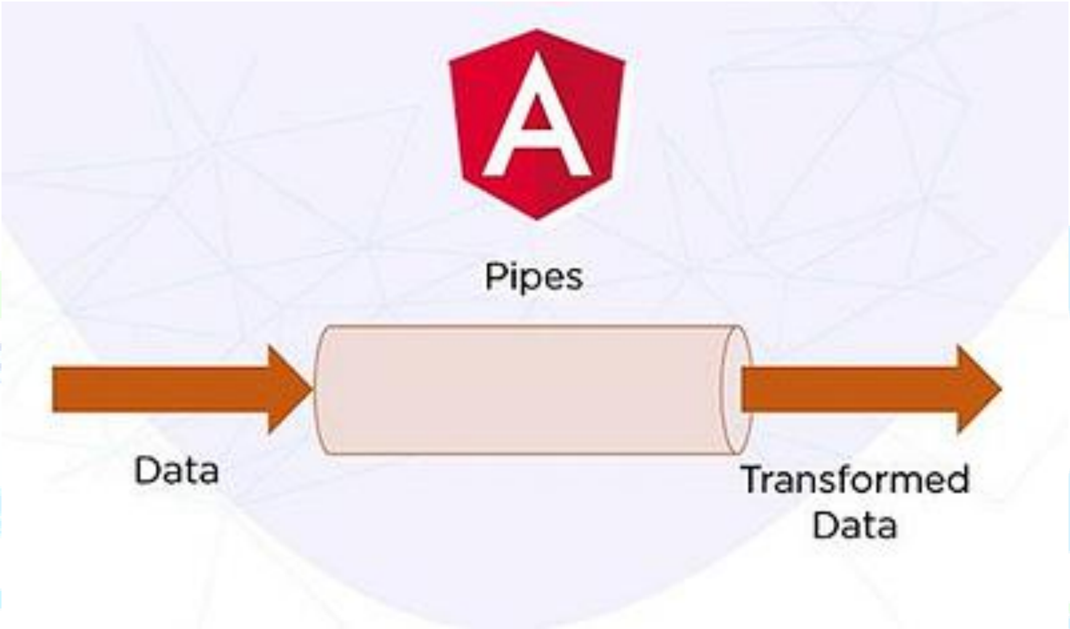
Complete Java Classes

by Kunal Sir

Angular Pipes

❖ What is angular pipes?

Angular Pipes transform the output. You can think of them as makeup rooms where they beautify the data into a more desirable format. They do not alter the data but change how they appear to the user.



In other words, pipes are nothing but simple functions that accept input value , and applies the business logic on input value and return the output value , that we can display on browser instead of displaying actual input value.

➢ There are some some special features of Angular pipe :-

1. Pipe are invoke by symbol ‘ | ‘.
2. Pipe can be chained.
3. We can pass parameter argument to pipes using ‘ : ’ symbol.

Angular Pipes

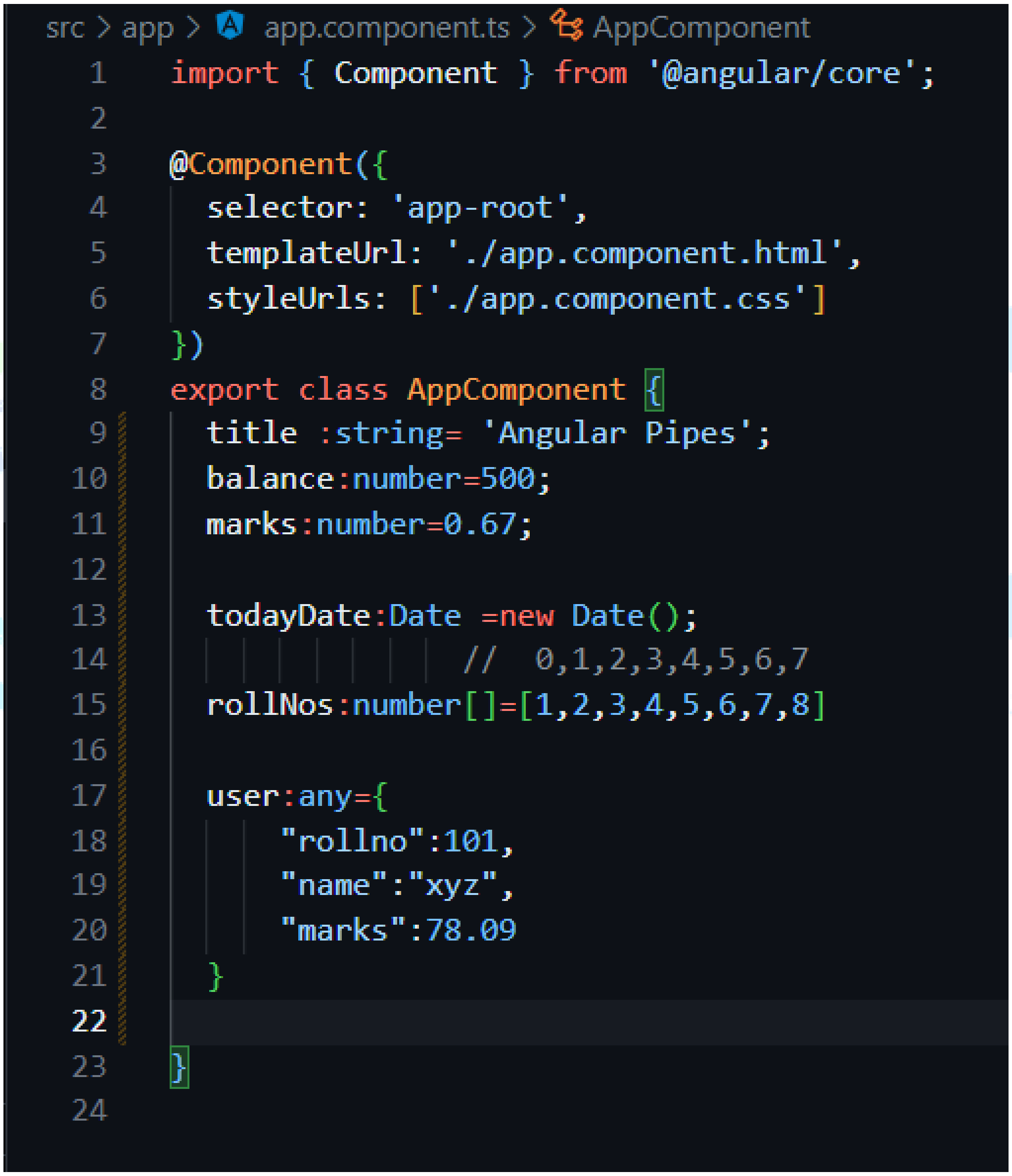
- **There are some build in Angular Pipes as below :-**

1. Uppercase.
2. Lowercase.
3. Currency.
4. Percent.
5. Date.
6. Slice.
7. Json.

Angular Pipes

❖ Example on build in pipes in angular :-

1. **Snippit on app.component.ts**

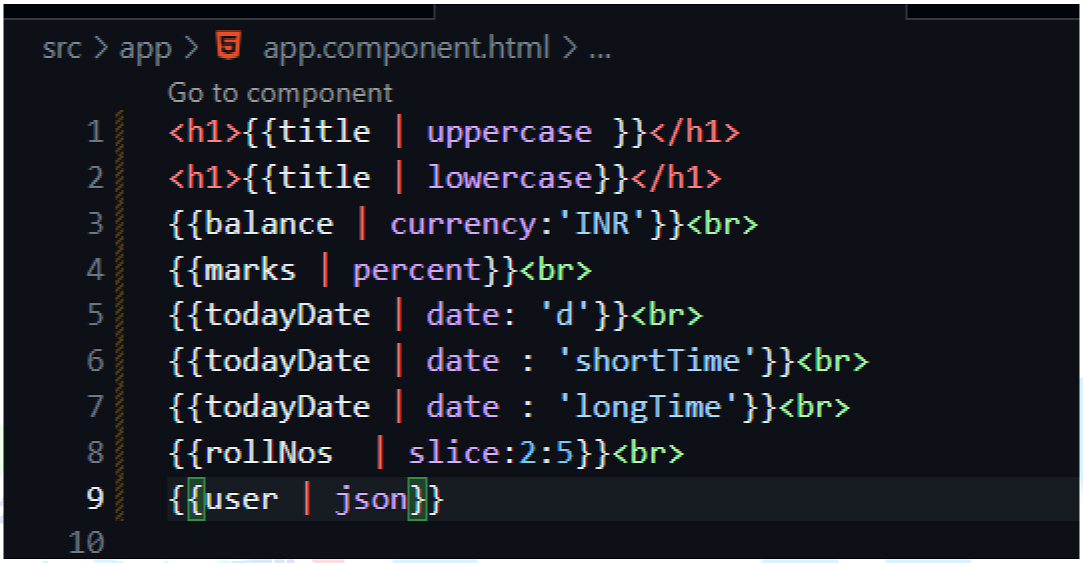


src > app > app.component.ts > AppComponent

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title :string= 'Angular Pipes';
  balance:number=500;
  marks:number=0.67;

  todayDate:Date =new Date();
                //  0,1,2,3,4,5,6,7
  rollNos:number[]=[1,2,3,4,5,6,7,8]

  user:any={
      "rollno":101,
      "name":"xyz",
      "marks":78.09
  }

}

```

Angular Pipes

2. **Snippit on app.component.html :-**



```
<h1>{{title | uppercase }}</h1>
<h1>{{title | lowercase}}</h1>
{{balance | currency:'INR'}}<br>
{{marks | percent}}<br>
{{todayDate | date: 'd'}}<br>
{{todayDate | date : 'shortTime'}}<br>
{{todayDate | date : 'longTime'}}<br>
{{rollNos  | slice:2:5}}<br>
{{user | json}}
```

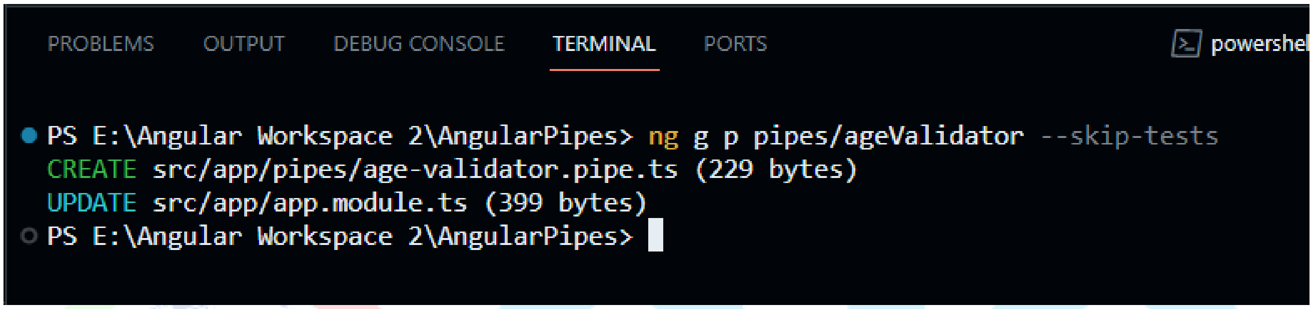
❖ How to create custom pipe :-

Whenever , build in pipes of angular are not able to satisfy need of our application or to transform the input value into desired outcome. then we have to create our own custom Pipes.

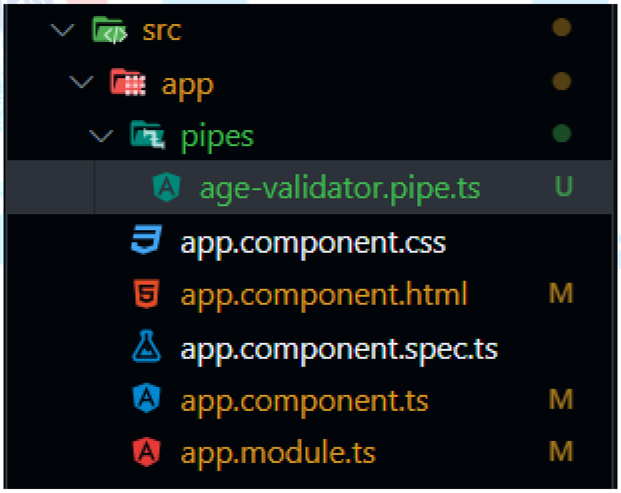
Angular Pipes

Steps to create and use custom pipes:-

- **Step 1 -** On terminal to create pipe use command - ng g p pipes/ageValidator - -skip-tests



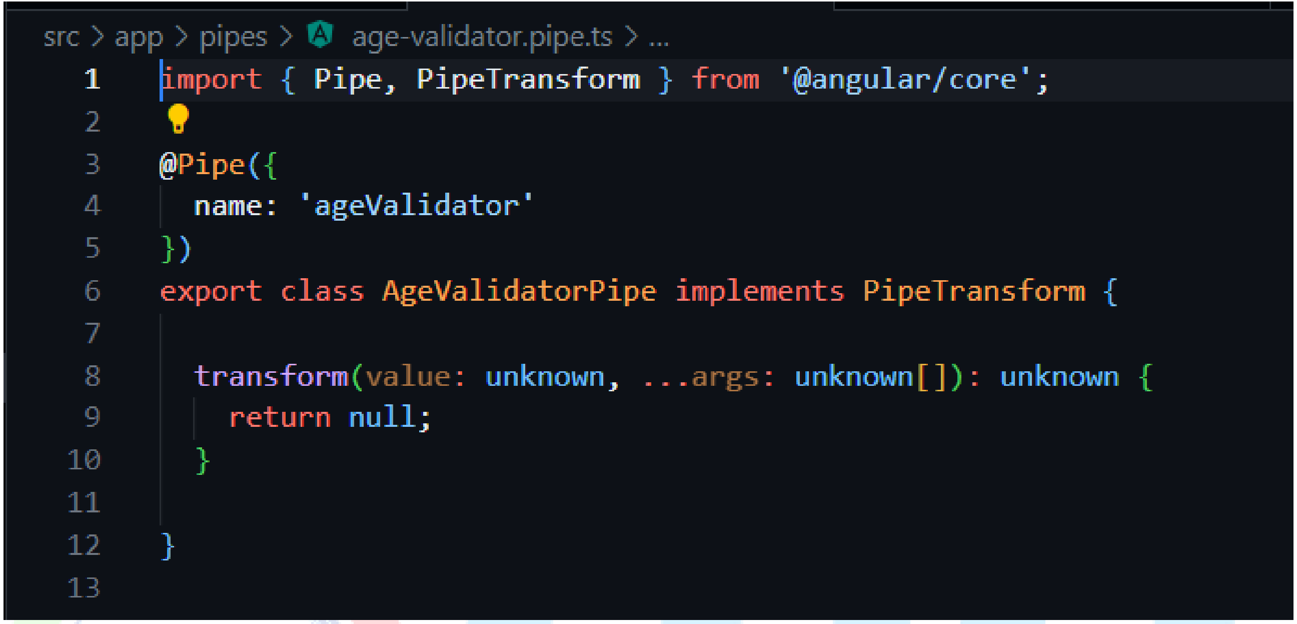
```
● PS E:\Angular Workspace 2\AngularPipes> ng g p pipes/ageValidator --skip-tests
  CREATE src/app/pipes/age-validator.pipe.ts (229 bytes)
  UPDATE src/app/app.module.ts (399 bytes)
○ PS E:\Angular Workspace 2\AngularPipes> 
```



```
⌄ src
  ⌄ app
    ⌄ pipes
        age-validator.pipe.ts      U
      app.component.css
      app.component.html           M
      app.component.spec.ts
      app.component.ts             M
      app.module.ts                M
```

Our AgeValidator pipe is created , and it is declared in AppModule. Lets open age-validator.pipe.ts.

Angular Pipes



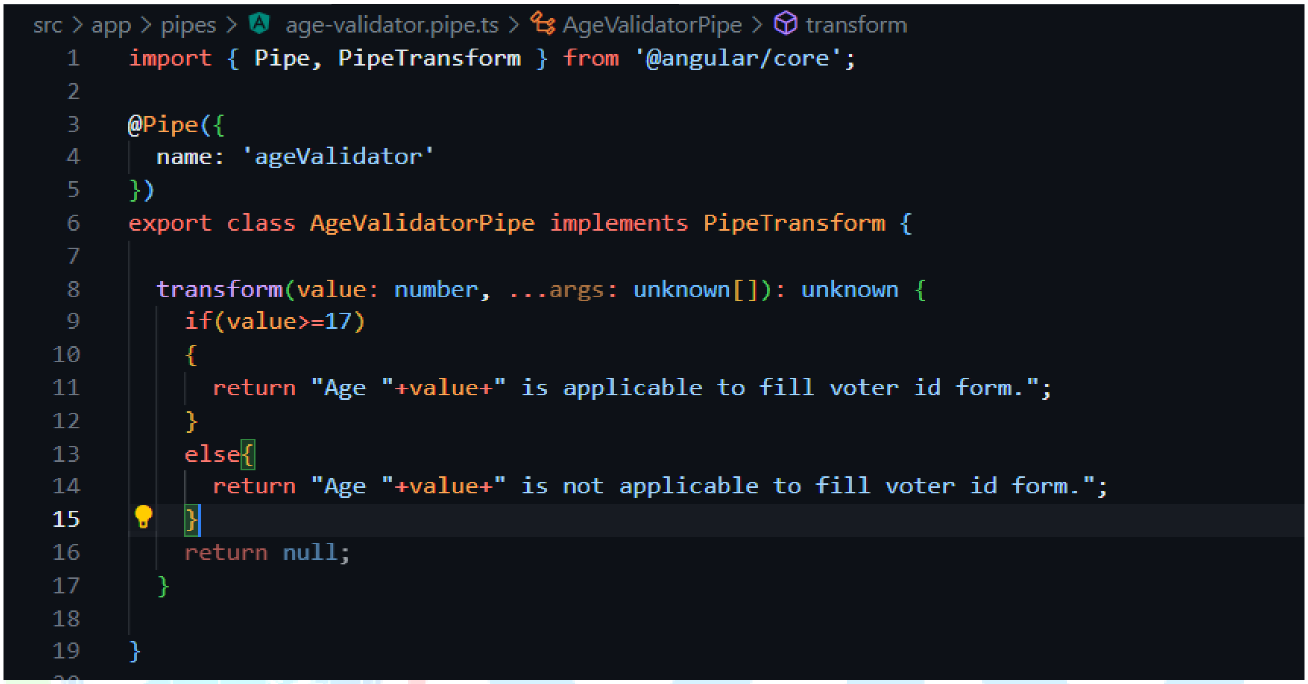
```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'ageValidator'
})
export class AgeValidatorPipe implements PipeTransform {

  transform(value: unknown, ...args: unknown[]): unknown {
    return null;
  }

}
```

@Pipe :- @Pipe decorator is a class level decorator and it is use to declare the class as angular pipe. In the parameter of pipe decorator we can see name property ,which describes the name of pipes through we can call this pipe in our template (view).

PipeTransform (I) :- An interface that is implemented by pipes in order to perform a transformation. Angular invokes the transform method with the value of a binding as the first argument, and any parameters as the second argument in list form.

Angular Pipes

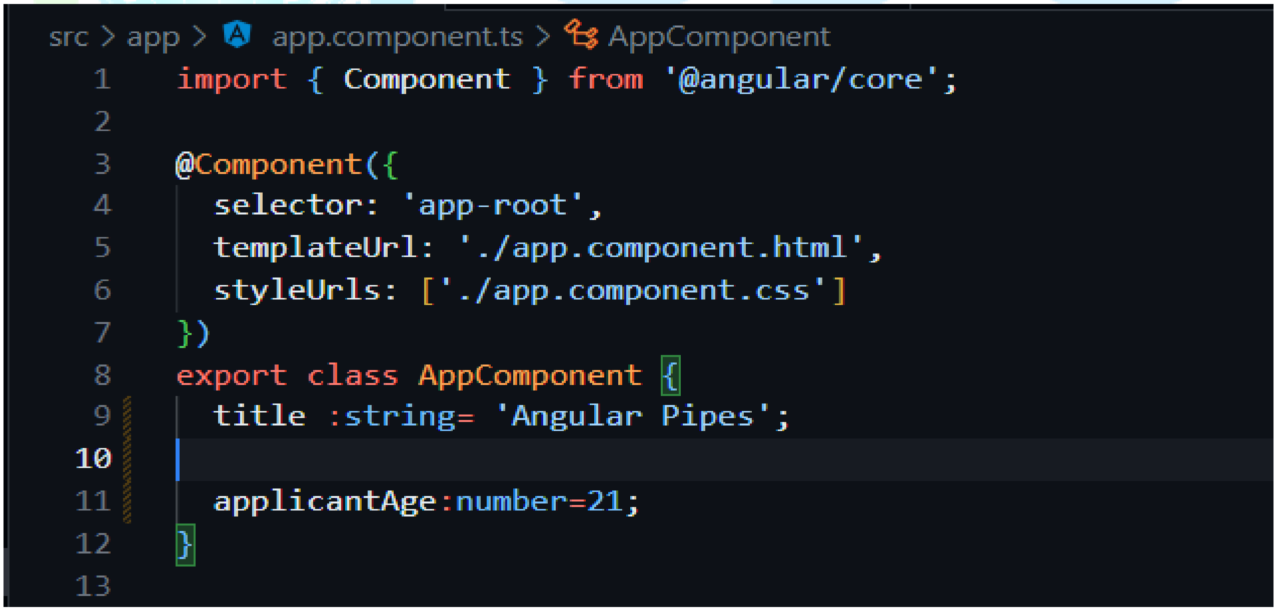
- **Step 2 -** Implement the business logic in the pipe to get your desired outcome.



src > app > pipes > age-validator.pipe.ts > AgeValidatorPipe > transform

```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'ageValidator'
})
export class AgeValidatorPipe implements PipeTransform {

  transform(value: number, ...args: unknown[]): unknown {
    if(value>=17)
    {
      return "Age "+value+" is applicable to fill voter id form.";
    }
    else{
      return "Age "+value+" is not applicable to fill voter id form.";
    }
    return null;
  }

}
```

- **Step 3 -** declare the data in app.component.ts to which you want to transform.

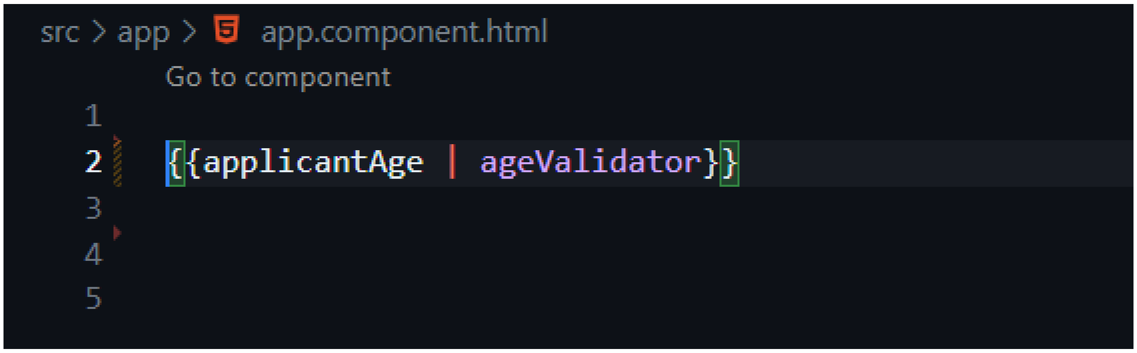


src > app > app.component.ts > AppComponent

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title :string= 'Angular Pipes';

  applicantAge:number=21;
}

```

Angular Pipes

- **Step 4 -** Use ageValidator pipe while binding applicantAge property in template using string interpolation.

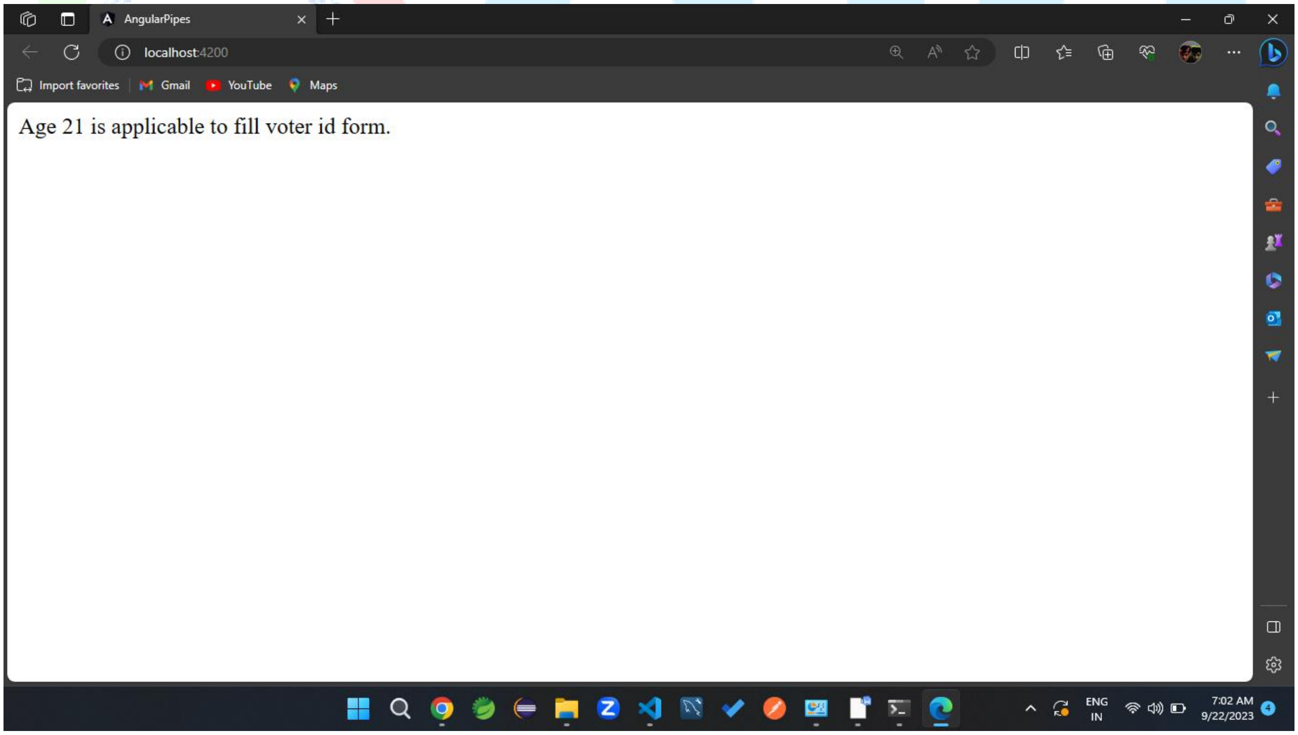


src > app > app.component.html

Go to component

```
{{applicantAge | ageValidator}}
```

OUTPUT :-



Age 21 is applicable to fill voter id form.

by Kunal Sir

Angular Pipes