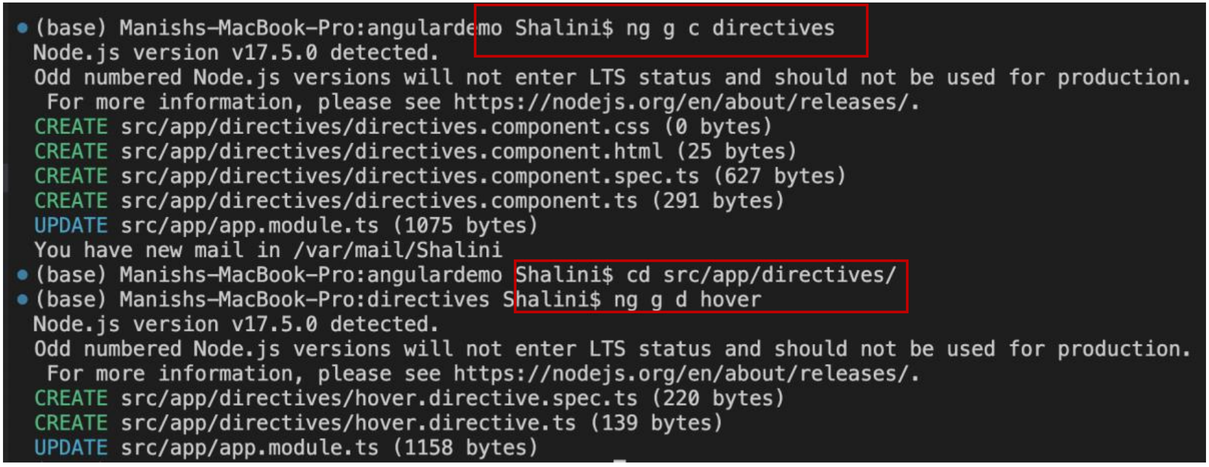
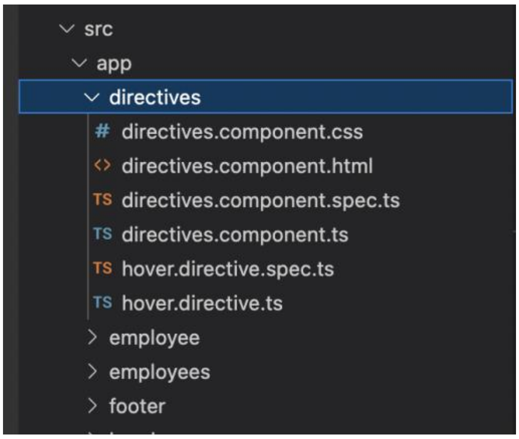
STEP 9: Directives

1. Add Directive component by executing the below command from the angulardemo folder:
   ng g c directives
2. Add a custom directive **from within the directives folder** using the below command:
   ng g d hover
   
   ```
   ● (base) Manishs-MacBook-Pro:angulardemo Shalini$ ng g c directives
     Node.js version v17.5.0 detected.
     Odd numbered Node.js versions will not enter LTS status and should not be used for production.
      For more information, please see https://nodejs.org/en/about/releases/.
     CREATE src/app/directives/directives.component.css (0 bytes)
     CREATE src/app/directives/directives.component.html (25 bytes)
     CREATE src/app/directives/directives.component.spec.ts (627 bytes)
     CREATE src/app/directives/directives.component.ts (291 bytes)
     UPDATE src/app/app.module.ts (1075 bytes)
     You have new mail in /var/mail/Shalini
   ● (base) Manishs-MacBook-Pro:angulardemo Shalini$ cd src/app/directives/
   ● (base) Manishs-MacBook-Pro:directives Shalini$ ng g d hover
     Node.js version v17.5.0 detected.
     Odd numbered Node.js versions will not enter LTS status and should not be used for production.
      For more information, please see https://nodejs.org/en/about/releases/.
     CREATE src/app/directives/hover.directive.spec.ts (220 bytes)
     CREATE src/app/directives/hover.directive.ts (139 bytes)
     UPDATE src/app/app.module.ts (1158 bytes)
   ```
3. Folder looks as follows:

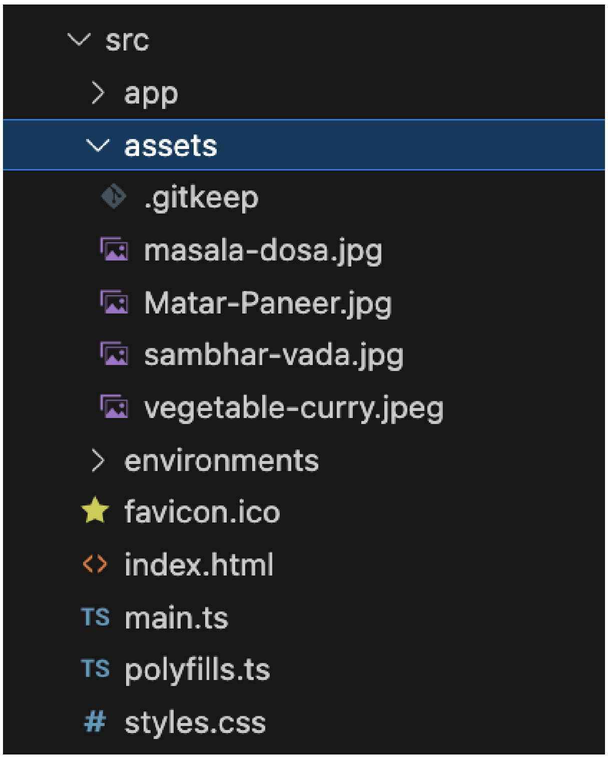
   
   ```
   ∨ src
     ∨ app
       ∨ directives
         # directives.component.css
         <> directives.component.html
         TS directives.component.spec.ts
         TS directives.component.ts
         TS hover.directive.spec.ts
         TS hover.directive.ts
       > employee
       > employees
       > footer
   ```
4. Comment out the previous tags from app.component.html file and add <app-directives> tag
5. To use ngFor directive add below code in directives.component.html file:
   <ul><li *ngFor="let pno of [1,2,3,5,7,11,13,17];let i = index">
   {{i}} : {{ pno }} </li></ul>
6. To use switch case, add below data in directives.component.ts file :
   people: any[] = [
   {
   "name": "Douglas Pace",
   "age": 35,
   "country": 'MARS'
   },
   {
   "name": "Mcleod Mueller",

```
"age": 32,
"country": 'USA'
},
{
"name": "Aguirre Ellis",
"age": 34,
"country": 'UK'
},
{
"name": "Cook Tyson",
"age": 32,
"country": 'USA'
}
];
```

7. Display style conditionally as follows:

```
<ul *ngFor="let person of people”  [ngSwitch]="person.country">
<li *ngSwitchCase="'UK'" class="text-success">
        {{ person.name }} ({{ person.country }})
</li>
<li *ngSwitchCase="'USA'"class="text-primary">
        {{ person.name }} ({{ person.country }})
</li>
<li *ngSwitchDefault  class="text-warning">
        {{ person.name }} ({{ person.country }})
</li>
</ul>
```

8. Create custom directives:
   a. This activity needs some dummy images to be added in the assets folder, which is available on the GIT REPO :



```
src
  app
  assets
    .gitkeep
    masala-dosa.jpg
    Matar-Paneer.jpg
    sambhar-vada.jpg
    vegetable-curry.jpeg
  environments
  favicon.ico
  index.html
  main.ts
  polyfills.ts
  styles.css
```

   b. Add below property in directives.component.ts file:

```
itemname:string='masala-dosa.jpg'
```

   c. Add below code in directives.component.html file:

```
<h1>Custom Directive</h1>
<select [(ngModel)]="itemname">
   <option value="masala-dosa.jpg">Masala Dosa</option>
   <option value="Matar-Paneer.jpg">Matar Paneer</option>
   <option value="sambhar-vada.jpg">Sambhar Vada</option>
   <option value="vegetable-curry.jpeg">Vegetable Curry</option>
</select>
<img src="../../assets/masala-dosa.jpg" width="400" height="400"/>
<p>para </p>
```

   d. Custom directives are accessed like attributes. To add a border when an image or paragraph is hovered, add below code in constructor of hover.directive.ts. Inject the references of ElementRef and Renderer as follows:

```
constructor(private el:ElementRef, private render:Renderer2) {
            console.log(el.nativeElement)
            render.setStyle(el.nativeElement,'border','2px solid #eee');
            render.setStyle(el.nativeElement,'padding','10px');
        }
```

   To use this directive, modify img and p tag of directives.component.html as follows:

```
<img src="../../assets/masala-dosa.jpg" width="400" height="400"
appHover/>
<p appHover >para </p>
```

e. To modify the behaviour on mouse over and out, add below listeners in hover.directive.ts file:

```
@HostListener('mouseenter')
  mouseenter(){
    this.render.setStyle(this.el.nativeElement,'border','2px solid #56ABEE');
  }
  @HostListener('mouseleave')
  mouseleave(){
   this.render.setStyle(this.el.nativeElement,'border','2px solid #eee');
 }
```

f. To get the dynamic itemname from component add below in hover.directive.ts file:

```
@Input()
itemname:string='';
```

g. To change the image src attribute dynamically, need to bind this with img tag src attribute. So add below in hover.directive.ts file:

```
@HostBinding("src")
 imgSrc:string='';
```

Modify the img tag of directives.component.html as follows:

```
<img src="../../assets/masala-dosa.jpg" width="400" height="400" appHover
[itemname]= "itemname"/>
```

h. To listen for changes add the below lifecycle methods in hover.directive.ts file:

```
// this method listens for only first time and displays the image for the
default itemname
ngOnInit(): void {
    this.imgSrc=`../../assets/${this.itemname}`
  }
// this method listens for any subsequent changes and displays the image for
that itemname
  ngOnChanges(changes: SimpleChanges): void {
    this.imgSrc=`../../assets/${this.itemname}`
  }
```

https://blog.angulartraining.com/how-to-control-native-html-elements-with-angular-40aecb9e6796