MEAN STACK TECHNOLOGIES

MODULE II- ANGULAR JS AND MONGODB

EXPERIMENT: 3A

Module Name: Attribute Directives - ngStyle
Apply multiple CSS properties to a paragraph in a component using ngStyle.

Solution:

Step 1: Install Angular CLI (if not already installed)

npm install -g @angular/cli

Step 2: Create a new Angular project

ng new loginDemo

cd loginDemo

Step 3: Generate a new component for the login form

ng generate component my-style-component

Step 4: Update the mystyle component

my-style-component.component.ts

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { LoginComponent } from './login/login.component';
import { CommonModule } from '@angular/common';
import { NgForComponent } from './ng-for/ng-for.component';
import { NgSwitchExampleComponent } from './ng-switch-example/ng-switch-
example.component';
import { RepeatDirective } from './repeat.directive';
import { MyStyleComponentComponent } from './my-style-component/my-style-
component.component';

@Component({
  selector: 'app-root',
  standalone: true,
```

```
  imports:
[RouterOutlet,LoginComponent,CommonModule,NgForComponent,NgSwitchExampleComponent
,RepeatDirective,MyStyleComponentComponent],
  templateUrl: './app.component.html',
  styleUrl: './app.component.css'
})
export class AppComponent {
  title = 'loginDemo';
}
```

Step 5: Update the login component.html

my-style-component.component.html

```
<!-- my-style-component.component.html -->
<p [ngStyle]="paragraphStyles">
   This paragraph has multiple CSS properties applied using ngStyle.
 </p>
```

Step 6: Update the app.component.ts

app.component.ts

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { LoginComponent } from './login/login.component';
import { CommonModule } from '@angular/common';
import { NgForComponent } from './ng-for/ng-for.component';
import   {   NgSwitchExampleComponent   }   from   './ng-switch-example/ng-switch-
example.component';
import { RepeatDirective } from './repeat.directive';
import   {   MyStyleComponentComponent   }   from   './my-style-component/my-style-
component.component';

@Component({
  selector: 'app-root',
  standalone: true,
                                                                    imports:
[RouterOutlet,LoginComponent,CommonModule,NgForComponent,NgSwitchExampleComponent
,RepeatDirective,MyStyleComponentComponent],
  templateUrl: './app.component.html',
  styleUrl: './app.component.css'
```

```
})
export class AppComponent {
  title = 'loginDemo';
}
```

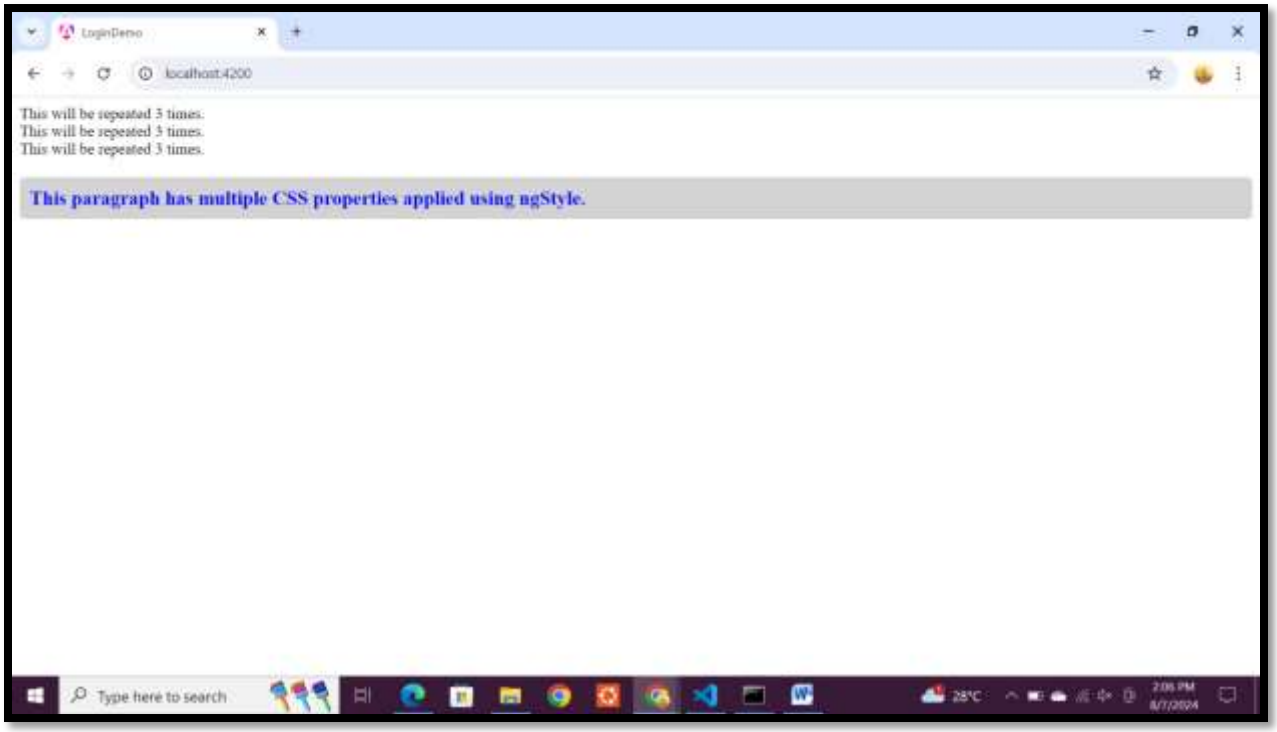
Step 7: Update the app.component.html Run the Angular application:

app.component.html

```
<app-my-style-component></app-my-style-component>
```

Step 8: Serve the application

ng serve –open



EXPERIMENT: 3B

Module Name: ngClass
Apply multiple CSS classes to the text using ngClass directive.

Solution:

Step 1: Generate a new component

ng generate component my-class-component

Step 2: Update the my-class-component component

my-class-component.component.ts

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

@Component({
  selector: 'app-my-class-component',
  standalone: true,
  imports: [CommonModule],
  templateUrl: './my-class-component.component.html',
  styleUrl: './my-class-component.component.css'
})
export class MyClassComponentComponent {
  textClasses = {
    'text-red': true,
    'text-bold': true,
    'text-italic': true,
    'text-large': true,
    'text-highlight': true
  };
}
```

my-class-component.component.html

```html
<!-- my-class-component.component.html -->
<p [ngClass]="textClasses">
  This text has multiple CSS classes applied using ngClass.
</p>
```

my-class-component.component.css

```css
/* my-class-component.component.css */
.text-red {
    color: red;
  }
```

```
  .text-bold {
    font-weight: bold;
  }

  .text-italic {
    font-style: italic;
  }

  .text-large {
    font-size: 24px;
  }

  .text-highlight {
    background-color: yellow;
  }
```

Step 3: Update the app.component.ts

app.component.ts

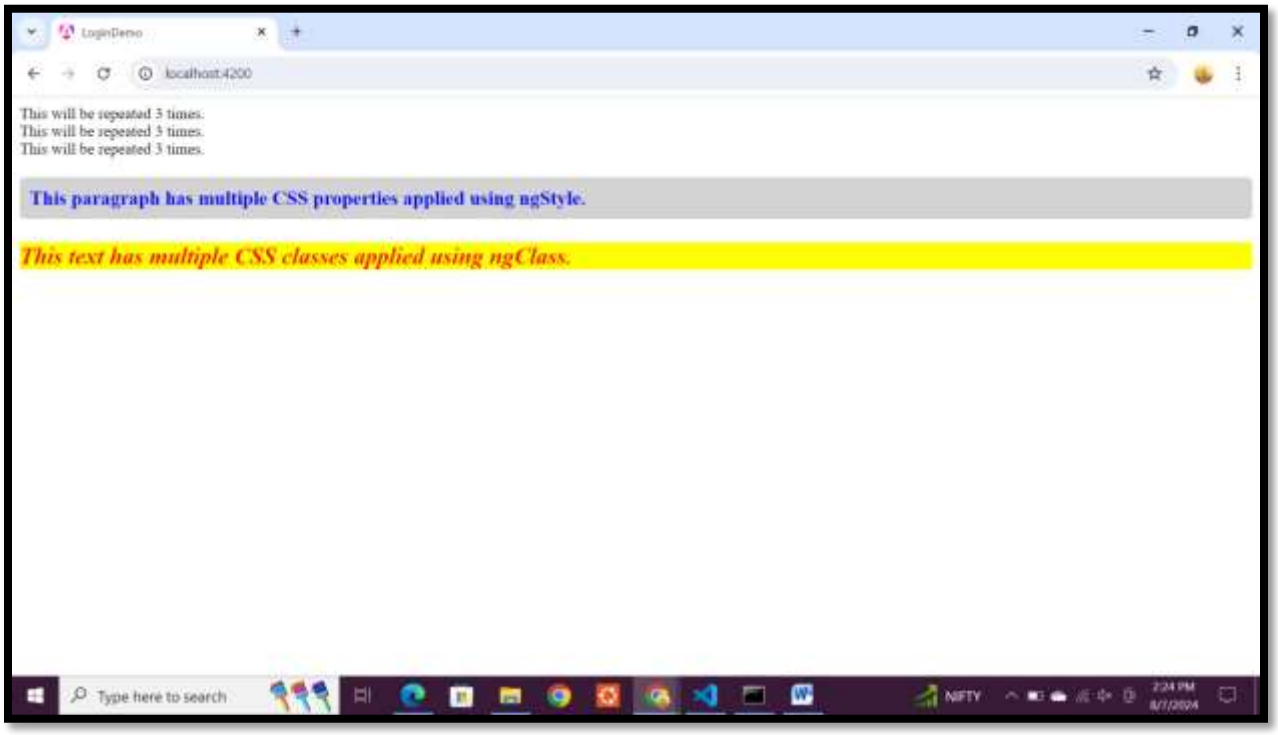
```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { LoginComponent } from './login/login.component';
import { CommonModule } from '@angular/common';
import { NgForComponent } from './ng-for/ng-for.component';
import { NgSwitchExampleComponent } from './ng-switch-example/ng-switch-example.component';
import { RepeatDirective } from './repeat.directive';
import { MyStyleComponentComponent } from './my-style-component/my-style-component.component';
import { MyClassComponentComponent } from './my-class-component/my-class-component.component';

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet,LoginComponent,CommonModule,NgForComponent,MyClassComponentComponent,NgSwitchExampleComponent,RepeatDirective,MyStyleComponentComponent],
  templateUrl: './app.component.html',
  styleUrl: './app.component.css'
})
export class AppComponent {
  title = 'loginDemo';
}
```

Step 4: Update the app.component.html

```
<app-my-class-component></app-my-class-component>
```

Step 5: Serve the application
Run the application to see your changes:
ng serve –open



EXPERIMENT: 3C

Module Name: Custom Attribute Directive

Create an attribute directive called 'showMessage' which should display the given message in a paragraph when a user clicks on it and should change the text color to red.

Solution:

1. Generate the Directive**: First, generate a new directive using Angular CLI.**

ng generate directive showMessage

2. Implement the Directive

Edit the generated directive file `repeat.directive.ts` as follows:

show-message.directive.ts

```
// show-message.directive.ts
import { Directive, ElementRef, HostListener, Input, Renderer2 } from
'@angular/core';

@Directive({
  selector: '[appShowMessage]',
  standalone: true
})
export class ShowMessageDirective {
  @Input('appShowMessage') message: string='';

  constructor(private el: ElementRef, private renderer: Renderer2) {

  }

  @HostListener('click') onClick() {
    const paragraph = this.renderer.createElement('p');
    const text = this.renderer.createText(this.message);
    this.renderer.appendChild(paragraph, text);
    this.renderer.setStyle(paragraph, 'color', 'red');
    this.renderer.appendChild(this.el.nativeElement, paragraph);
  }
}
```

Update app.component.ts

app.component.ts

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { LoginComponent } from './login/login.component';
import { CommonModule } from '@angular/common';
import { NgForComponent } from './ng-for/ng-for.component';
import { NgSwitchExampleComponent } from './ng-switch-example/ng-switch-example.component';
import { RepeatDirective } from './repeat.directive';
import { MyStyleComponentComponent } from './my-style-component/my-style-component.component';
import { MyClassComponentComponent } from './my-class-component/my-class-component.component';
import { ShowMessageDirective } from './show-message.directive';

@Component({
  selector: 'app-root',
  standalone: true,
  imports:
[RouterOutlet,AppComponent,ShowMessageDirective,LoginComponent,CommonModule,NgForComponent,MyClassComponentComponent,NgSwitchExampleComponent,RepeatDirective,MyStyleComponentComponent],
  templateUrl: './app.component.html',
  styleUrl: './app.component.css'
})
export class AppComponent {
  title = 'loginDemo';
}
```

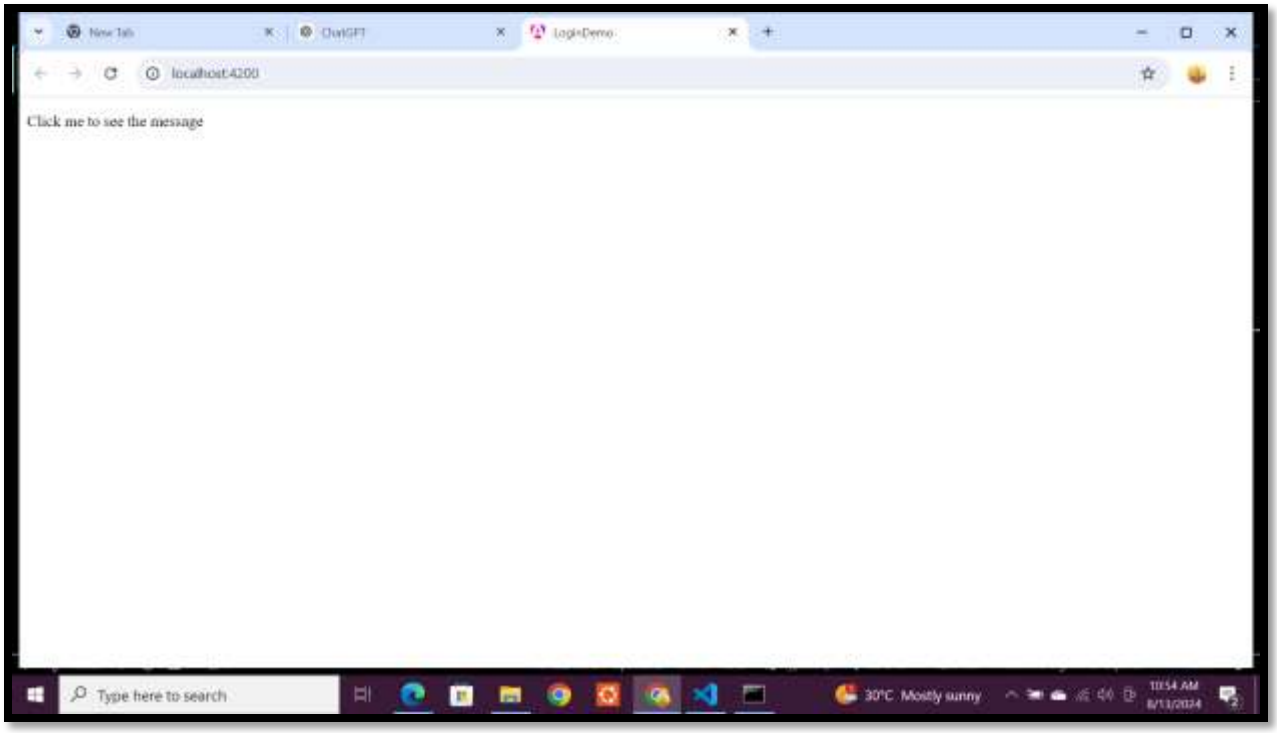
Update app.component.html

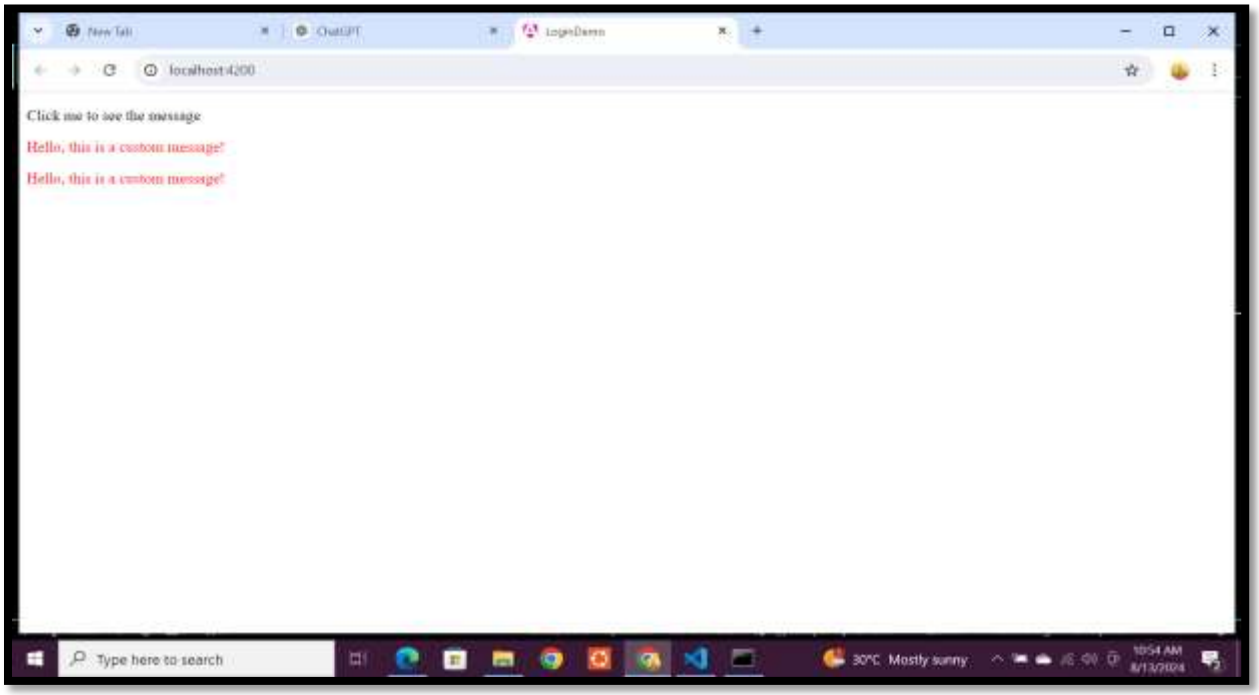
```
<p appShowMessage="Hello, this is a custom message!">Click me to see the message</p>
```

Step 5: Serve the application

Run the application to see your changes:

ng serve –open


localhost:4200
Click me to see the message
Type here to search
30°C Mostly sunny


localhost:4200
Click me to see the message
Hello, this is a custom message!
Hello, this is a custom message!
Type here to search
30°C Mostly sunny