

HTML LAB 6

Index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Calci</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

App.component.html

```
<div class="calculator">
  <h1>Calculator</h1>

  <div>
    <br>
    <input class="result" type="text" readonly value="{{ result }}"><br><br><br>
  </div>

  <div>
    <button class="responsive long" *ngFor="let longButton of longButtons" (click)="addToExpression(longButton)">
      {{ longButton }}
    </button>
    <button class="responsive " *ngFor="let button of buttons" (click)="addToExpression(button)">
      {{ button }}
    </button>
    <button class="responsive long1" *ngFor="let longButton1 of longButtons1" (click)="addToExpression(longButton1)">
      {{ longButton1 }}
    </button>
  </div>
</div>
```

App.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})

export class AppComponent {

  result: string = '';
  longButtons: string[] = ['AC', 'CE'];
  buttons: string[] = ['7','8', '9','/', '4','5','6','*', '1','2','3','-','.', '0','=','+'];
  longButtons1: string[] = ['Prime', 'Factorial'];

  private prevValue: string = '';
  private curValue: string = '';

  addToExpression(value: string) {

    if (this.result !== "") {
      this.prevValue = this.curValue;
      this.curValue = value;
    }

    if (value == 'AC') {
      this.result = '';
    }
    else if (value == 'CE'){
      this.result = this.prevValue != "" ? this.result.slice(0, -1) : this.result;
    }
    else if (value == '=') {
      this.result = eval(this.result);
    }
    else if (value == 'Prime') {
      var c=0;
      var n=parseInt(this.result);
      for(let i=1; i<=n/2;i++ ){
        c+=1;
      }
      if(c==1){
        this.result='Prime';
      }
    }
  }
}
```

```

    }
    else{
        this.result='Composite';
    }
}
else if(value=='Factorial'){
    var f=1;
    var n=parseInt(this.result);
    for(let i=1;i<=n;i++)
        f=f*i;
    this.result=f.toString();
}
else {
    this.result += value;
}
}
}

```

App.component.css

```

h1{
    margin-top: 0%;
    padding-top: 15px;
    padding-bottom: 0;
    font-size: 2em;
    font: bold small-caps bold 16px/2 cursive;
    color: rgb(198, 219, 240);
    font-family: Asul ;
}

.calculator{
    height:40em; background-size:cover; width:auto;
    background-position:50% 50%;
    background-color:rgb(23, 40, 58);
    position: relative;
    margin: 7% auto auto auto;
    width: 400px;
    height: 600px;
    text-align: center;
    border-color: rgb(99, 94, 97);
    border-radius: 20px;
    box-shadow: inset 2px 0px 10px -5px rgb(85, 80, 83),
    inset -5px 0px 15px -5px rgb(85, 80, 83),
    inset 10px 0px 10px -12px rgb(89, 80, 83),
    inset -2px 0px 4px 2px rgb(85, 80, 83),
    inset -2px 0px 10px 0px rgb(89, 80, 83);
}

```

```
}

.result{
border-radius: 8px;
}

button{
    font-family: Asul;
    border-radius: 8px;
    width: 80px;
    height: 55px;
    font-size: 20px;
    font-weight: bold;
    border: 2px solid;
    border-color: rgb(87, 85, 85);
    margin: 2px 2px 8px 2px;
    transition-duration: 0.4s;
    box-shadow: inset 5px 15px -5px rgb(109, 103, 106),
    inset -3px 15px -5px rgb(109, 103, 106),
    inset 0px -5px 0px -3px rgb(109, 103, 106),
    inset 0 0 2px 2px rgb(109, 103, 106),
    inset 0 5px 15px rgb( 109, 103, 106);
}

.long{
    width: 170px;
    background-color: black;
    color: whitesmoke;
}

.long1{
    width: 160px;
}

.responsive: hover {
    background-color: rgb(138, 134, 134);
    color: rgb(247, 240, 240);
    box-shadow: inset 5px 0 15px -5px rgb(199, 192, 182),
    inset -3px 0 15px -5px rgb(199, 192, 182),
    inset 0px -5px 0px -3px rgb(199, 192, 182),
    inset 0 0 2px 2px rgb(199, 192, 182),
    inset 0 5px 15px 0 rgb(199, 192, 182),
}

input{
    font-weight: bold;
    background-color: yellowgreen;
    width: 350px;
```

```
height: 60px;  
text-align: right;  
font-size: 1.7em;  
margin-top: 0%;  
padding-top: 0%;  
}
```

APP

