NAME: ABHIK DEY

REG.NO: 201900073

SL LAB ASSIGNMENT

Q) Create a calculator app using Angular which is capable of performing following operations:

- Addition of two numbers
- Subtraction of two numbers
- Multiplication of two numbers
- Division of two numbers
- Factorial of a number
- Checking if a given number is Prime or not

->Snapshots and codes

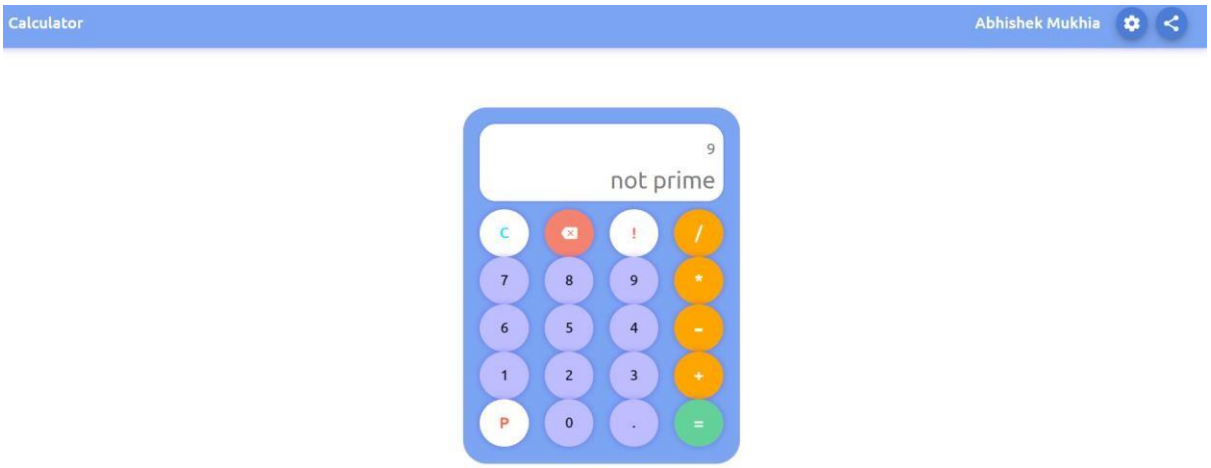
```
D:\Angular projects\calculator>ng serve
? Would you like to share anonymous usage data about this project with the Angular Team at
Google under Google's Privacy Policy at https://policies.google.com/privacy? For more
details and how to change this setting, see https://angular.io/analytics. Yes

Thank you for sharing anonymous usage data. Should you change your mind, the following
command will disable this feature entirely:

    ng analytics project off

- Generating browser application bundles (phase: setup)...Compiling @angular/cdk/keycodes : es2015 as esm2015
Compiling @angular/cdk/platform : es2015 as esm2015
Compiling @angular/cdk/observers : es2015 as esm2015
Compiling @angular/cdk/a11y : es2015 as esm2015
Compiling @angular/cdk/bidi : es2015 as esm2015
Compiling @angular/forms : es2015 as esm2015
Compiling @angular/material/core : es2015 as esm2015
Compiling @angular/material/grid-list : es2015 as esm2015
Compiling @angular/material/toolbar : es2015 as esm2015
Compiling @angular/common/http : es2015 as esm2015
Compiling @angular/material/icon : es2015 as esm2015
Compiling @angular/material/button : es2015 as esm2015
Compiling @angular/material/card : es2015 as esm2015
Compiling @angular/cdk/collections : es2015 as esm2015
Compiling @angular/cdk/scrolling : es2015 as esm2015
Compiling @angular/cdk/portal : es2015 as esm2015
Compiling @angular/cdk/overlay : es2015 as esm2015
Compiling @angular/cdk/layout : es2015 as esm2015
Compiling @angular/material/tooltip : es2015 as esm2015
√ Browser application bundle generation complete.

Initial Chunk Files   | Names         |      Size
vendor.js             | vendor        |   3.33 MB
polyfills.js          | polyfills     | 510.58 kB
styles.css, styles.js | styles        | 461.02 kB
main.js               | main          |  38.05 kB
runtime.js            | runtime       |   6.62 kB

                      | Initial Total |   4.32 MB

Build at: 2021-11-17T05:37:33.902Z - Hash: 194d9bdfa9c3717019ca - Time: 19629ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **


√ Compiled successfully.
√ Browser application bundle generation complete.

5 unchanged chunks

Build at: 2021-11-17T05:37:34.581Z - Hash: 3543949e4bd70010e50c - Time: 220ms

√ Compiled successfully.
```

Using angular components

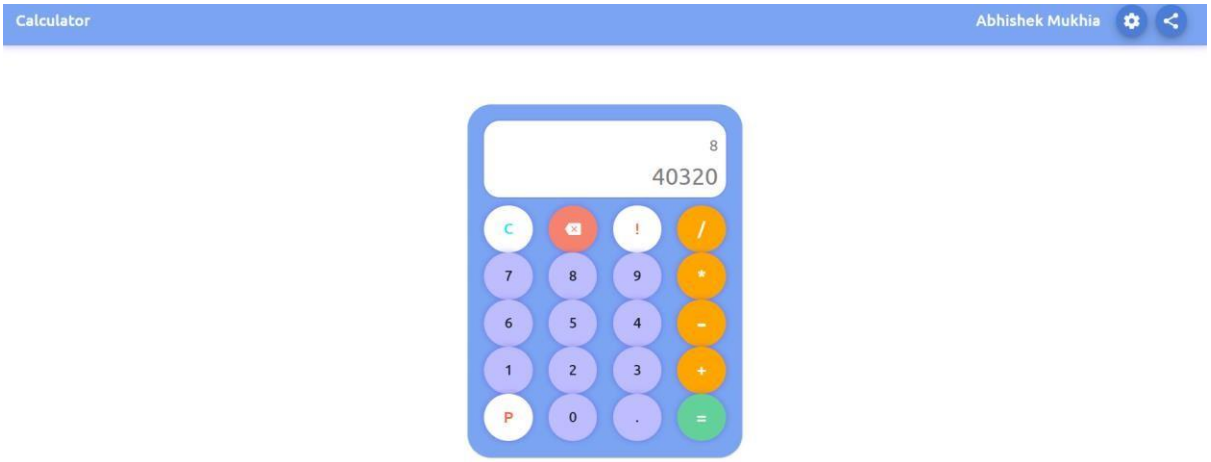
```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import { BrowserAnimationsModule } from '@angular/platform-
browser/animations';
import { CalculatorComponent } from './calculator/calculator.component';
import { MatGridListModule } from '@angular/material/grid-list';
import { MatToolbarModule } from '@angular/material/toolbar';
import { MatIconModule } from '@angular/material/icon';
import { Toolbar } from './toolbar/toolbar.component';
import { MatButtonModule } from '@angular/material/button';
import { MatCardModule } from '@angular/material/card';
import { MatTooltipModule } from '@angular/material/tooltip';
@NgModule({
  declarations: [AppComponent, CalculatorComponent, Toolbar],
  imports: [
    BrowserModule,
    BrowserAnimationsModule,
    MatGridListModule,
    MatToolbarModule,
    MatIconModule,
    MatButtonModule,
    MatCardModule,
    MatTooltipModule,
  ],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

Final Output:

Prime Button



Factorial



The Function logic code:-

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

@Component({
  selector: 'app-calculator',
  templateUrl: './calculator.component.html',
  styleUrls: ['./calculator.component.scss'],
})
export class CalculatorComponent {
  symbols = ['C', 'del', '!', '/', '*', '-', '+', '=', 'P', '.'];

  setList = [
    [this.symbols[0], this.symbols[1], this.symbols[2], this.symbols[3]],
    [7, 8, 9, this.symbols[4]],
    [6, 5, 4, this.symbols[5]],
    [1, 2, 3, this.symbols[6]],
    [this.symbols[8], 0, this.symbols[9], this.symbols[7]],
```

```
];

inp: any = '';
res: any = '';

delete() {
  const vals = this.inp.value.split('');
  vals.splice(-1, 1);
  this.inp.value = vals.length > 0 ? vals.join('') : null;
  this.inp.value ? null : this.reset();
}

reset() {
  this.inp.value = null;
  this.res.value = null;
}

setAnsToDisplay(ans: any) {
  this.inp.value = ans;
}

extrude(item: any) {
  const copy = this.inp.value.split('');
  copy.splice(-1, 1);
  this.inp.value = copy.join('') + item;
}

isPrime(num: any) {
  for (var i = 2; i < num; i++) if (num % i === 0) return false;
  return num > 1;
}
factorial(num: any) {
  let pro = 1;
  for (let i = num; i > 0; i--) {
    pro *= i;
  }

  return pro;
}
getLength() {
  return this.inp.value.split(/([-+*\/])/g).length;
}

evaluate() {
  var ans = eval(this.inp.value);
  var prevValue = this.inp.value;
  this.reset();
  this.res.value = prevValue;
```

```
    this.setAnsToDisplay(ans);
  }

  getPrime() {
    if (this.getLength() === 1) {
      const ans = this.isPrime(this.inp.value);
      var prevValue = this.inp.value;
      this.reset();
      this.res.value = prevValue;
      this.setAnsToDisplay(ans ? 'prime' : 'not prime');
      setTimeout(() => {
        this.reset();
      }, 1000);
    }
  }
  checkExtrude(item: any) {
    this.symbols.includes(this.inp.value.at(-1)) &&
this.symbols.includes(item)
      ? this.extrude(item)
      : (this.inp.value += item);
  }

  getFactorial() {
    if (this.getLength() === 1) {
      const ans = this.factorial(Number(this.inp.value));
      var prevValue = this.inp.value;
      this.reset();
      this.res.value = prevValue;
      this.setAnsToDisplay(ans);
    }
  }

  getValue(item: any) {
    this.inp = <HTMLInputElement>document.querySelector('.inp-field');
    this.res = <HTMLInputElement>document.querySelector('.res-field');

    switch (item) {
      case 'C':
        this.reset();
        break;
      case 'del':
        this.delete();
        break;
      case '=':
        this.evaluate();
        break;
      case 'P':
        this.res.value = this.inp.value;
```

```
        this.getPrime();
        break;
      case '!':
        this.res.value = this.inp.value;
        this.getFactorial();
        break;
      default:
        this.checkExtrude(item);
        break;
    }
  }
}
```