**EX:4**

**Date:14.07.25**

**Smart Counter using Angular JS**

**AIM:**

To create an interactive, visually appealing counter application using AngularJS, featuring Increment, decrement, double, reset, randomize, and auto-increment controls , FizzBuzz and prime number detection , Confetti animation on multiples of 10 , Dark mode toggle ,Responsive and modern UI.

**PROCEDURE:**

**1. Initialize AngularJS App:**

* Create an AngularJS module and controller.

**2. Define State Variables:**

* `count`: stores the current counter value.
* `animated`, `darkMode`, `autoInc`, `fizzbuzz`, `isPrime`, `showConfetti`, `confettiArray`: manage UI state and features.

**3. Animation Helper:**

* Define a function to briefly set `animated` to true for counter animation.

**4. FizzBuzz Logic:**

Create a function that returns:

* "FizzBuzz" if `count` is divisible by 15 (and not zero),
* "Fizz" if divisible by 3,
* "Buzz" if divisible by 5,

**5. Prime Checker:**

* Create a function to check if `count` is a prime number.

**6. Confetti Animation:**

* When triggered, generate an array of confetti pieces with random colors, positions, and rotations, and display them for a short duration.

**7.Watch Counter Value:**

Use `$scope.$watch` on `count` to:

* Update FizzBuzz and prime status.
* Trigger confetti if `count` is a nonzero multiple of 10.

**8.Increment/Decrement:**

* `change(v)` increases or decreases `count` by `v` (if result is non-negative), then animates.

**9.Double:**

* `double()` multiplies `count` by 2 (if changed), then animates.

**10.Reset with Animation:**

* `reset()` animates the counter rolling to zero (stepwise), then animates.

**11.Randomize:**

* `randomize()` sets `count` to a random integer between 0 and 100 (if changed), then animates.

**12.Dark Mode Toggle:**

* `toggleDark()` toggles the `darkMode` flag, switching the UI theme.

**13.Auto Increment Toggle:**

* `toggleAuto()` toggles `autoInc` and starts/stops an interval that increments `count` every second.

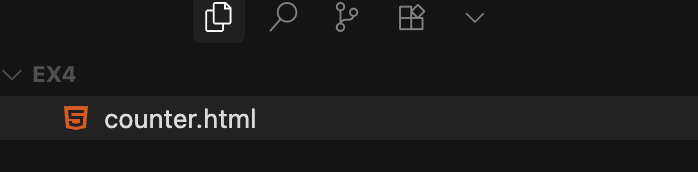
**14.Cleanup on Destroy:**

* On controller destroy, cancel the auto-increment interval if active.

**15.Responsive and Styled UI:**

* Use CSS for modern, responsive layout, button styles, counter animation, confetti, and dark mode.

**DESIGN:**

****

1. **Initialize AngularJS App:**

Set up the AngularJS module and controller to manage the app’s logic and data binding.

1. **Define State Variables:**

Declare variables in the controller’s scope to track the counter value, UI states (like animation, dark mode), and feature flags.

1. **Animation Helper:**

Create a function that briefly toggles an animation flag to trigger CSS-based counter animations.

1. **FizzBuzz Logic:**

Implement a function that checks the counter value and returns “Fizz”, “Buzz”, “FizzBuzz”, or an empty string based on divisibility by 3, 5, or 15.

1. **Prime Checker:**

Write a function to determine if the current counter value is a prime number.

1. **Confetti Animation:**

Generate and display animated confetti pieces on the screen when triggered, using random colors and positions for visual effect.

**IMPLEMENTATION:**

<!DOCTYPE html>

<html ng-app="counterApp">

<head>

<meta charset="UTF-8">

<title>AngularJS Counter</title>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<style>

body { display: flex; justify-content: center; align-items: center;

height: 100vh; margin: 0; background: linear-gradient(135deg,#f5f6fa,#dff9fb); }

.ctr { background: #fff; padding: 32px 24px; border-radius: 20px;

box-shadow: 0 8px 32px #0001;

display: flex; flex-direction: column; align-items: center;

min-width: 260px; position:relative; }

.ctitle { margin: 0 0 14px; font-size: 2rem; font-weight: 600; color: #0984e3;

text-align: center; }

.cval { font-size: 3.2rem; font-weight: bold; margin: 10px 0 18px;

background: linear-gradient(90deg,#74b9ff,#00b894);

-webkit-background-clip: text; -webkit-text-fill-color: transparent;

background-clip: text; text-shadow: 0 2px 8px #3498db22;

transition: transform .2s cubic-bezier(.68,-0.55,.27,1.55); }

.cval.a { transform: scale(1.12) rotate(-2deg); }

.cval.surprise { animation: pop 0.4s cubic-bezier(.68,-0.55,.27,1.55); }

@keyframes *pop* { 0%{transform:scale(1) rotate(0);} 30%{transform:scale(1.3) rotate(8deg);}

60%{transform:scale(0.9) rotate(-8deg);} 100%{transform:scale(1) rotate(0);} }

.cbtns { display: flex; gap: 12px; }

.cbtn { font-size: 1.5rem; width: 48px; height: 48px;

border: none; border-radius: 50%; color: #fff; cursor: pointer;

background: #0984e3; transition: background .2s,transform .1s,box-shadow .2s;

box-shadow: 0 2px 8px #0984e31a; display: flex; align-items: center;

justify-content: center; }

.cbtn:active { transform: scale(.95); }

.cbtn[title="Increment"] { background: #00b894; }

.cbtn[title="Decrement"] { background: #d63031; }

.cbtn[title="Double"] { background: #0984e3; }

.cbtn[title="Reset"] { border-radius: 16px; width: auto; padding: 0 16px;

font-size: 1rem; background: #636e72; }

.cbtn[title="Reset"]:hover { background: #b2bec3; color: #2d3436; }

.cbtn:hover { filter: brightness(1.12); box-shadow: 0 4px 16px #0984e32e; }

.ctr.dark { background: #23272f; color: #f5f6fa; }

.ctr.dark .ctitle { color: #74b9ff; }

.ctr.dark .cval { text-shadow: 0 2px 8px #222b; }

.ctr.dark .cbtn { background: #636e72; color: #fff; }

.ctr.dark .cbtn[title="Increment"] { background: #00b894; }

.ctr.dark .cbtn[title="Decrement"] { background: #d63031; }

.ctr.dark .cbtn[title="Double"] { background: #0984e3; }

.ctr.dark .cbtn[title="Reset"] { background: #636e72; }

.ctr.dark .cbtn:hover { filter: brightness(1.2); }

.confetti { pointer-events:none; position:absolute; left:0; top:0;

width:100vw; height:100vh; z-index:10; }

.piece { position:absolute; width:12px; height:12px; border-radius:3px; opacity:0.8; }

@media (max-width:600px){ .ctr{padding:14px 2vw;} .cval{font-size:2rem;}

.cbtn{width:36px;height:36px;font-size:1rem;} }

</style>

</head>

<body ng-controller="CounterController">

<div class="ctr" ng-class="{dark: darkMode}">

<h2 class="ctitle">Smart counter</h2>

<div class="cval" ng-class="{a: animated, surprise: surpriseAnim}">{{ count }}</div>

<div style="display:flex;gap:8px;align-items:center;margin-bottom:10px;">

<button class="cbtn" style="width:auto;border-radius:12px;font-size:1rem;padding:0 10px;" ng-click="randomize()">Randomize</button>

<button class="cbtn" style="width:auto;border-radius:12px;font-size:1rem;padding:0 10px;" ng-click="toggleAuto()">{{autoInc?'⏸️':'▶️'}} Auto</button>

<button class="cbtn" style="width:auto;border-radius:12px;font-size:1rem;padding:0 10px;" ng-click="toggleDark()">{{darkMode?'☀️':'🌙'}} Mode</button>

</div>

<div class="cbtns">

<button class="cbtn" ng-click="change(-1)" title="Decrement">&#8722;</button>

<button class="cbtn" ng-click="change(1)" title="Increment">&#43;</button>

<button class="cbtn" ng-click="double()" title="Double">&#10006;2</button>

<button class="cbtn" ng-click="reset()" title="Reset">&#8634; Reset</button>

</div>

<div style="margin-top:14px;font-size:1.1rem;min-height:1.5em;text-align:center;">

<span ng-if="fizzbuzz">{{fizzbuzz}}</span>

<span ng-if="isPrime">Prime!</span>

</div>

<div class="confetti" ng-if="showConfetti">

<div ng-repeat="n in confettiArray" class="piece" ng-style="n.style"></div>

</div>

</div>

<script>

angular.module('counterApp',[]).controller('CounterController',function(*$scope*,*$timeout*,*$interval*){

*$scope*.count=0;*$scope*.animated=false;*$scope*.darkMode=false;*$scope*.surpriseAnim=false;

*$scope*.autoInc=false;*$scope*.fizzbuzz='';*$scope*.isPrime=false;*$scope*.showConfetti=false;*$scope*.confettiArray=[];

let autoIntv=null;

function **anim**(){ *$scope*.animated=true; $timeout(()=>{*$scope*.animated=false;},200); }

function **checkFizzBuzz**(*v*){ return *v*%15===0&&*v*!==0?'FizzBuzz':*v*%3===0&&*v*!==0?'Fizz':*v*%5===0&&*v*!==0?'Buzz':''; }

function **isPrime**(*n*){ if(*n*<=1)return false; for(let i=2;i<=Math.sqrt(*n*);i++)if(*n*%i===0)return false; return true; }

function **confetti**(){ *$scope*.showConfetti=true; *$scope*.confettiArray=Array.from({length:24},()=>{ let color=['#00b894','#fdcb6e','#0984e3','#d63031','#636e72'][Math.random()\*5|0],left=Math.random()\*100,top=Math.random()\*40+20,rot=Math.random()\*360,dur=Math.random()\*0.7+0.8; return {style:{background:color,left:left+'vw',top:top+'vh',transform:'rotate('+rot+'deg)',transition:'top '+dur+'s cubic-bezier(.68,-0.55,.27,1.55)'}}; }); $timeout(()=>{*$scope*.showConfetti=false;},1200); }

*$scope*.$watch('count',*v*=>{ *$scope*.fizzbuzz=checkFizzBuzz(*v*); *$scope*.isPrime=isPrime(*v*); if(*v*&&*v*%10===0) confetti(); });

*$scope*.change=*v*=>{ let n=*$scope*.count+parseInt(*v*); if(n>=0){ *$scope*.count=n; anim(); } };

*$scope*.double=()=>{ let n=*$scope*.count\*2; if(n!==*$scope*.count){ *$scope*.count=n; anim(); } };

*$scope*.reset=()=>{ if(*$scope*.count!==0){ let s=*$scope*.count>0?-1:1; let **roll**=()=>{ if(*$scope*.count!==0){ *$scope*.count+=s; $timeout(roll,10); } else anim(); }; roll(); } };

*$scope*.randomize=()=>{ let v=Math.floor(Math.random()\*101); if(v!==*$scope*.count){ *$scope*.count=v; anim(); } };

*$scope*.toggleDark=()=>{ *$scope*.darkMode=!*$scope*.darkMode; };

*$scope*.toggleAuto=()=>{ *$scope*.autoInc=!*$scope*.autoInc; if(*$scope*.autoInc){ autoIntv=$interval(()=>{ *$scope*.change(1); },1000); }else if(autoIntv){ *$interval*.cancel(autoIntv); autoIntv=null; } };

*$scope*.$on('$destroy',()=>{ if(autoIntv) *$interval*.cancel(autoIntv); });

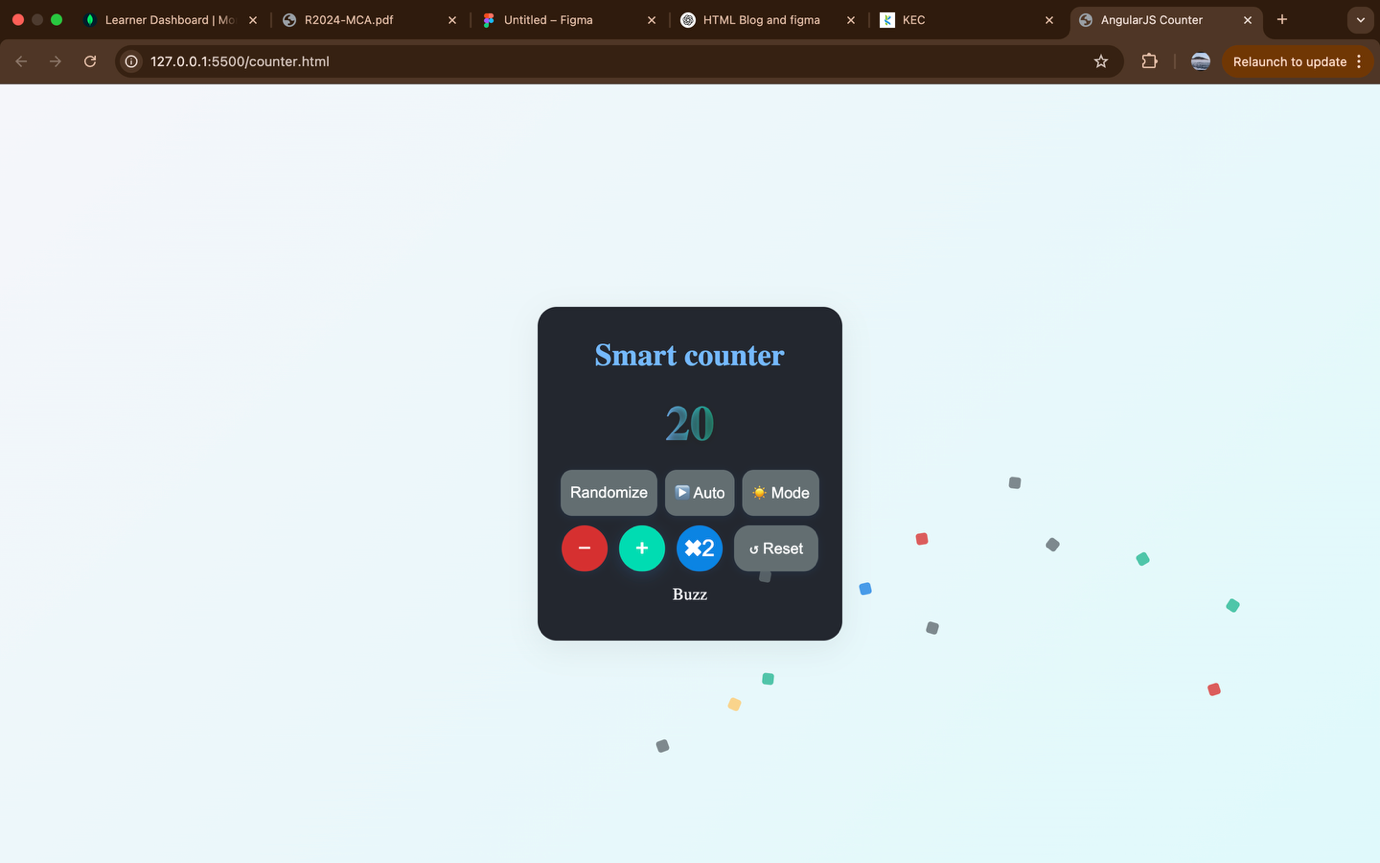
});

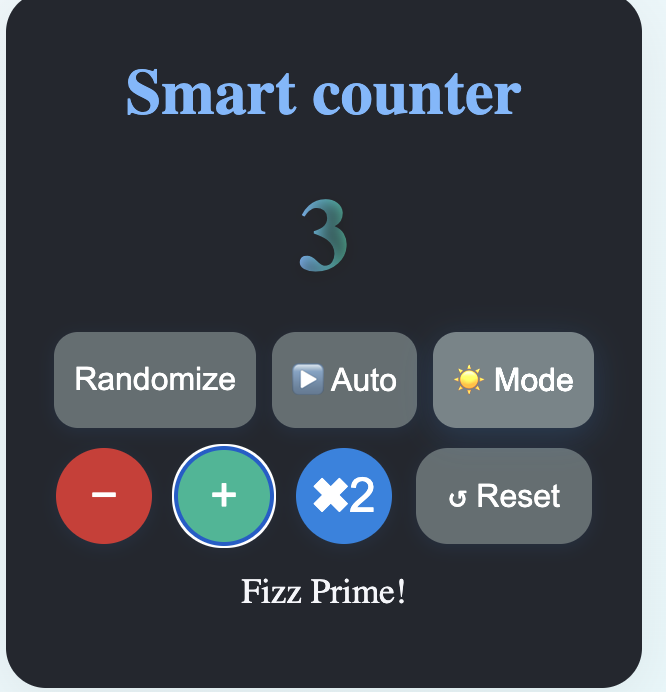
</script>

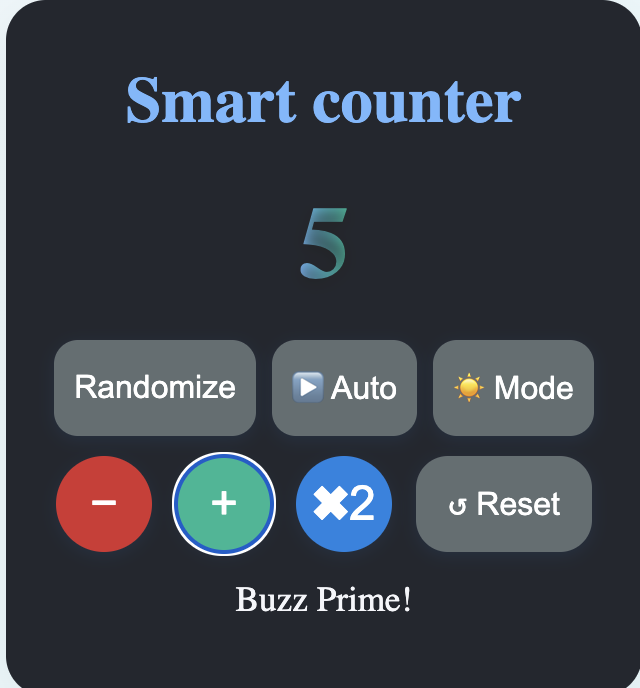
</body>

</html>

**OUTPUT:**

****

****

****

|  |  |
| --- | --- |
| **COE(30)** |  |
| **RECORD(10)** |  |
| **OBSERVATION(10)** |  |
| **VIVA(10)** |  |
| **TOTAL(60)** |  |

**RESULT:**

The smart counter was successfully developed using AngularJS,It features increment, decrement, double, reset (with animation), randomize, auto-increment, FizzBuzz and prime number detection, confetti animation for multiples of 10, and a dark mode toggle.