

PRACTICAL 1

AIM - Write HTML/JavaScript code for the following : When name of the basic color is entered in the textbox and leave the textbox then it should take that color as background color.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>BackGround color Changer</title>
  <style>
    body {
      text-align: center;
      font-family: sans-serif;
      font-size: 15px;
    }
  </style>
</head>
<body>
  <h1>Color Changer</h1>
  Enter color name: <input type="text" placeholder="Enter the color name" id="InputColor"
onblur="colorChange()" ">

  <script>
    function colorChange() {
      var color = document.getElementById('InputColor').value;
      document.body.style.backgroundColor = color;
    }
  </script>
</body>
</html>
```



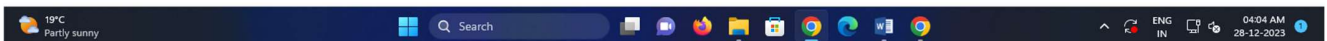
A D PATEL INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY
102046701-Advanced Web Development



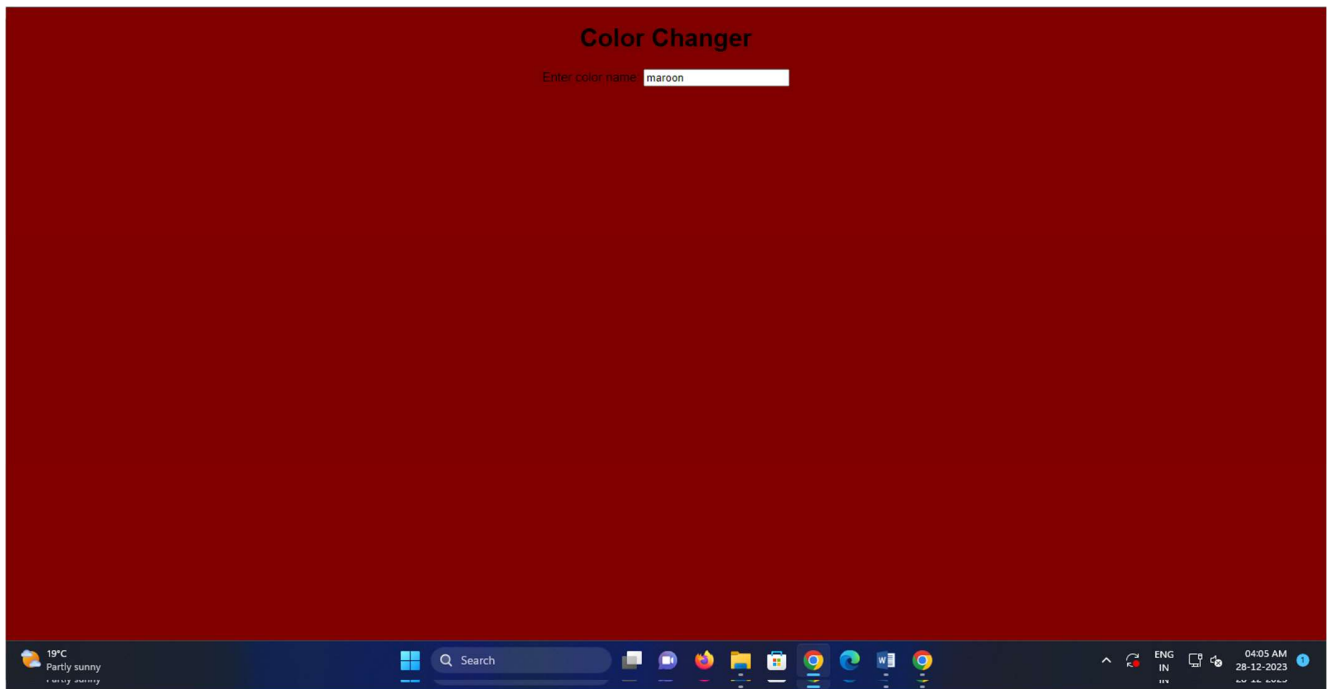
OUTPUT

Color Changer

Enter color name:



INPUT - maroon



AIM-Until you enter the complete mobile number (means less than 10 digits) in the textbox, it should take red color as background and when it becomes complete then show the alert message

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mobile Number Validator</title>
  <style>
    body {
      text-align: center;
      font-family: sans-serif;
      font-size: 15px;
    }
    #InputNumber {
      border: 2px solid black;
    }
  </style>
</head>
<body>
  <h1>Mobile Number Validator</h1>
  Enter mobile number: <input type="number" placeholder="Enter the mobile number"
id="InputNumber" oninput="validateNumber()">
  <script>
    function validateNumber() {
      var inputNumber = document.getElementById('InputNumber').value;

      if (inputNumber.length < 10) {
        InputNumber.style.backgroundColor = 'red';
      } else {
        InputNumber.style.backgroundColor = 'green';
        alert('Mobile number entered successfully: ' + inputNumber);
      }
    }
  </script>
</body>
```

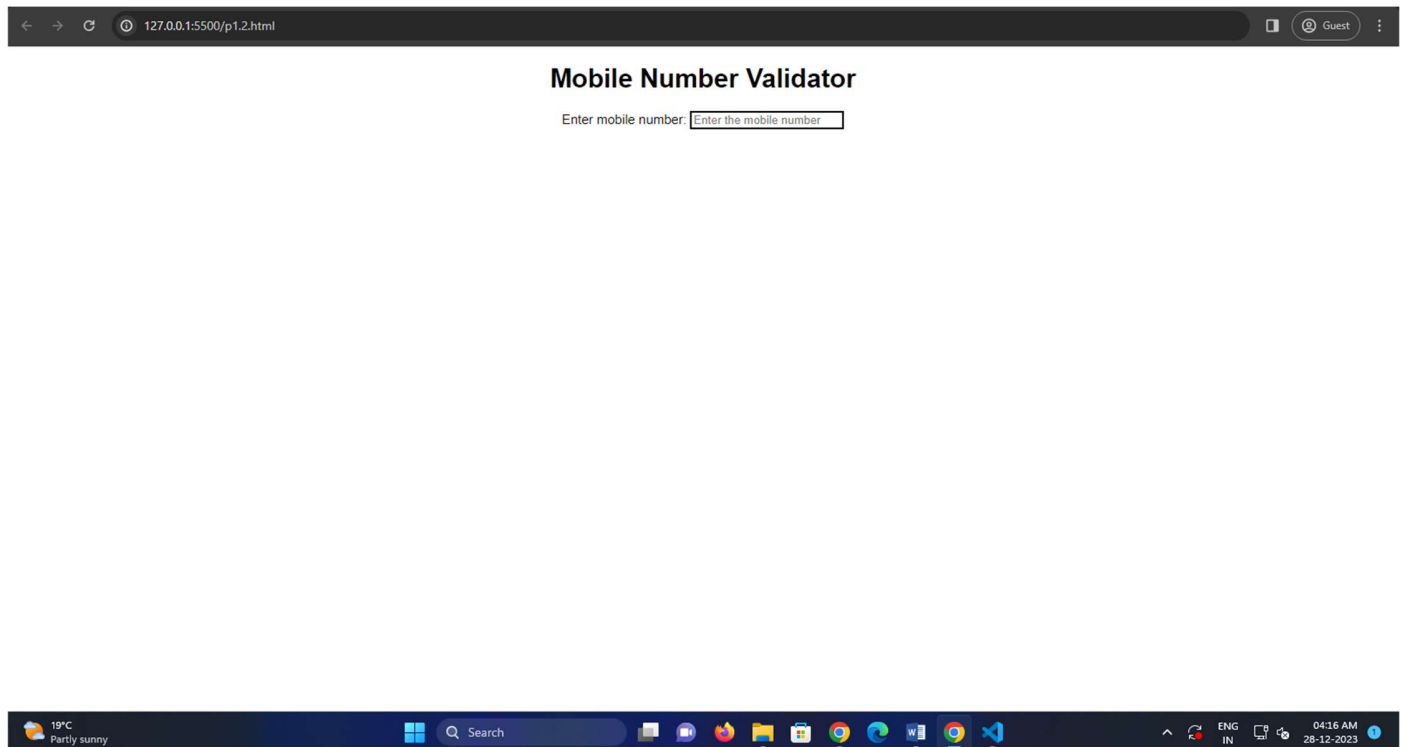


A D PATEL INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY
102046701-Advanced Web Development



</html>

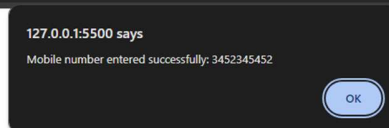
OUTPUT





Mobile Number Validator

Enter mobile number:





A D PATEL INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY
102046701-Advanced Web Development



Mobile Number Validator

Enter mobile number:



AIM-Write a Javascript code to create user defined object Car with carModelName, carColor, carPrice, carSpeed properties and with displaySpeed(s),increaseSpeed(s) and DecreaseSpeed(s) as methods.

```
<!DOCTYPE html>
<html>
<head>
  <title>Car Object Example</title>
  <style>
    body{
      background-image: url("Screenshot 2023-12-28 050621.png");
    }
  </style>
</head>
<body>

  <h1>Car Object Example</h1>
  <label for="increaseSpeedInput" style="color: aliceblue;">Increase Speed by (km/h) :
</label>
  <input type="number" id="increaseSpeedInput" min="0">
  <button onclick="increaseSpeed()">Increase Speed</button>
  <br><br>

  <label for="decreaseSpeedInput" style="color: aliceblue;">Decrease Speed by (km/h) :
</label>
  <input type="number" id="decreaseSpeedInput" min="0">
  <button onclick="decreaseSpeed()">Decrease Speed</button>
  <br><br>
  <p id="speedDisplay" style="color: aliceblue;"></p>

  <script>

    function Car(modelName, color, price, speed) {
      this.carModelName = modelName;
      this.carColor = color;
      this.carPrice = price;
```



```
this.carSpeed = speed;

this.increaseSpeed = function(s) {
    this.carSpeed += s;
    return `Speed increased by ${s} km/h. Current speed: ${this.carSpeed} km/h`;
};

this.decreaseSpeed = function(s) {
    if (this.carSpeed - s >= 0) {
        this.carSpeed -= s;
        return `Speed decreased by ${s} km/h. Current speed: ${this.carSpeed} km/h`;
    } else {
        return `Cannot decrease speed by ${s} km/h. Speed cannot be negative.`;
    }
};

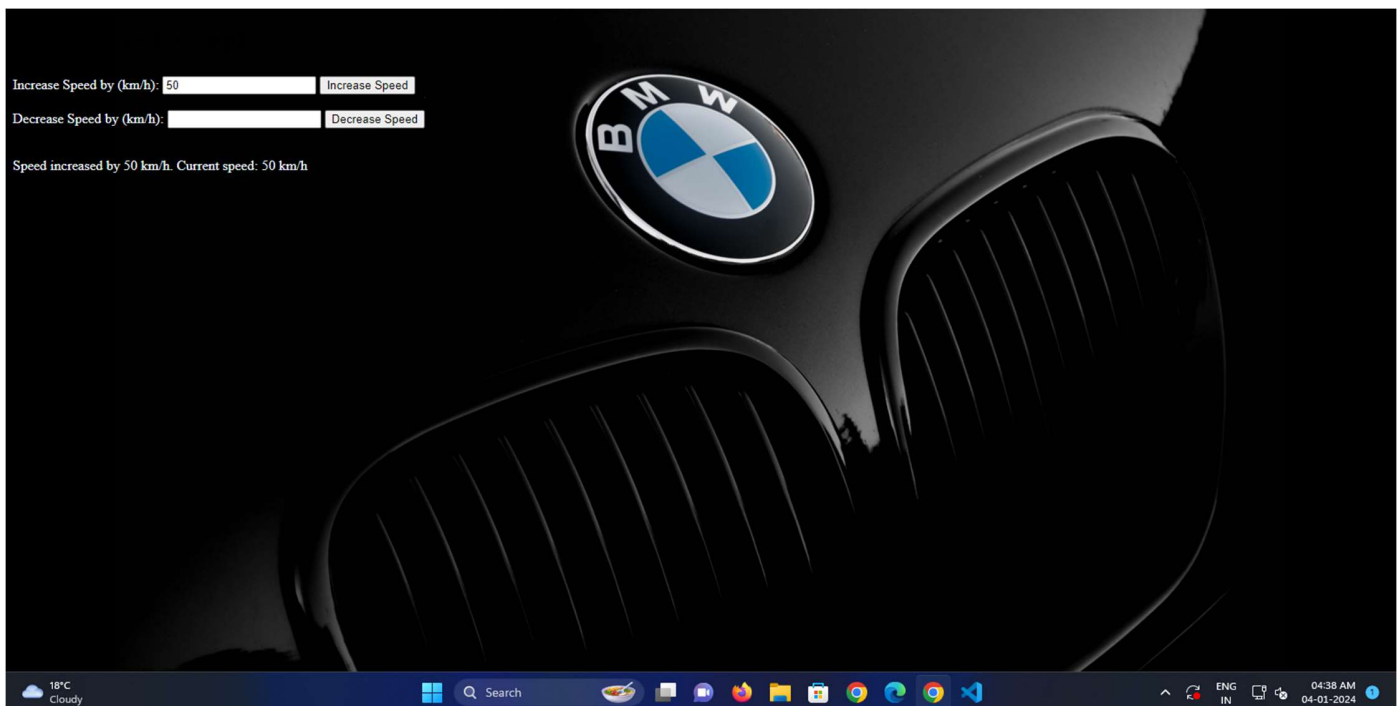
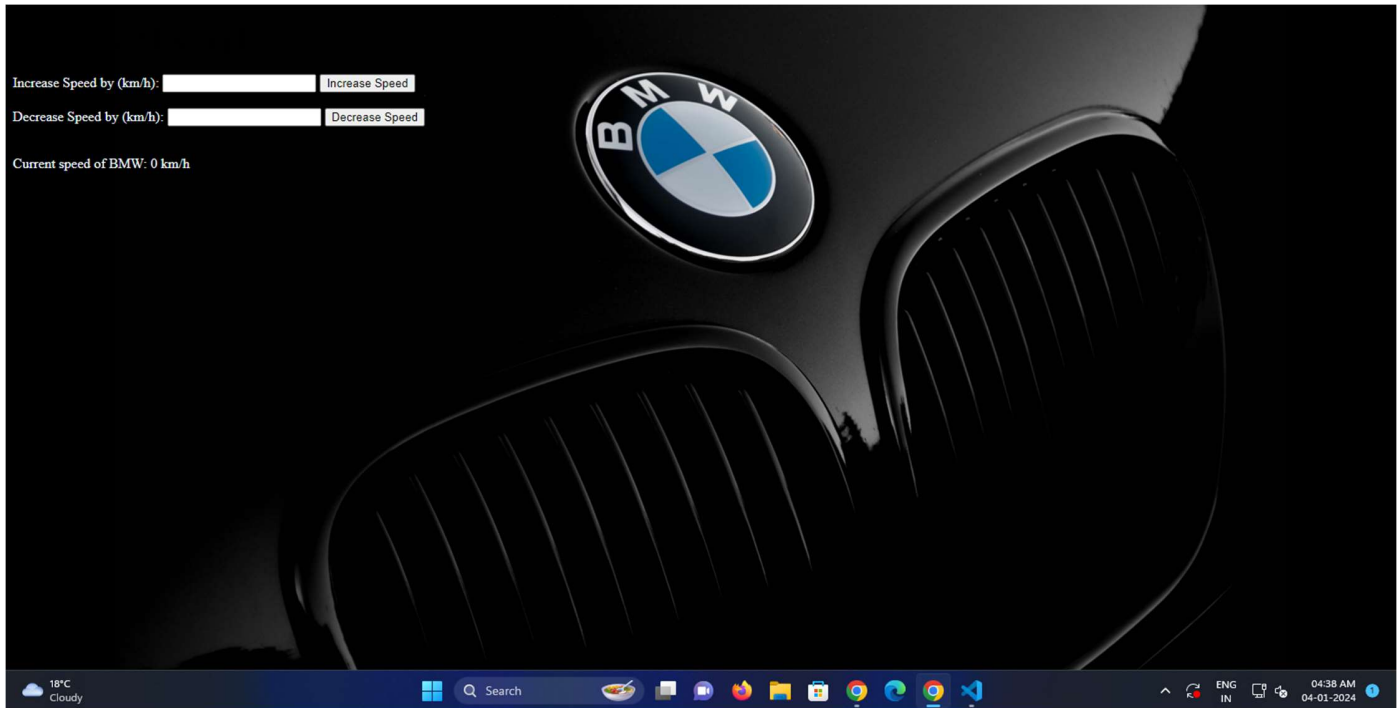
this.displaySpeed = function() {
    return `Current speed of ${this.carModelName}: ${this.carSpeed} km/h`;
};

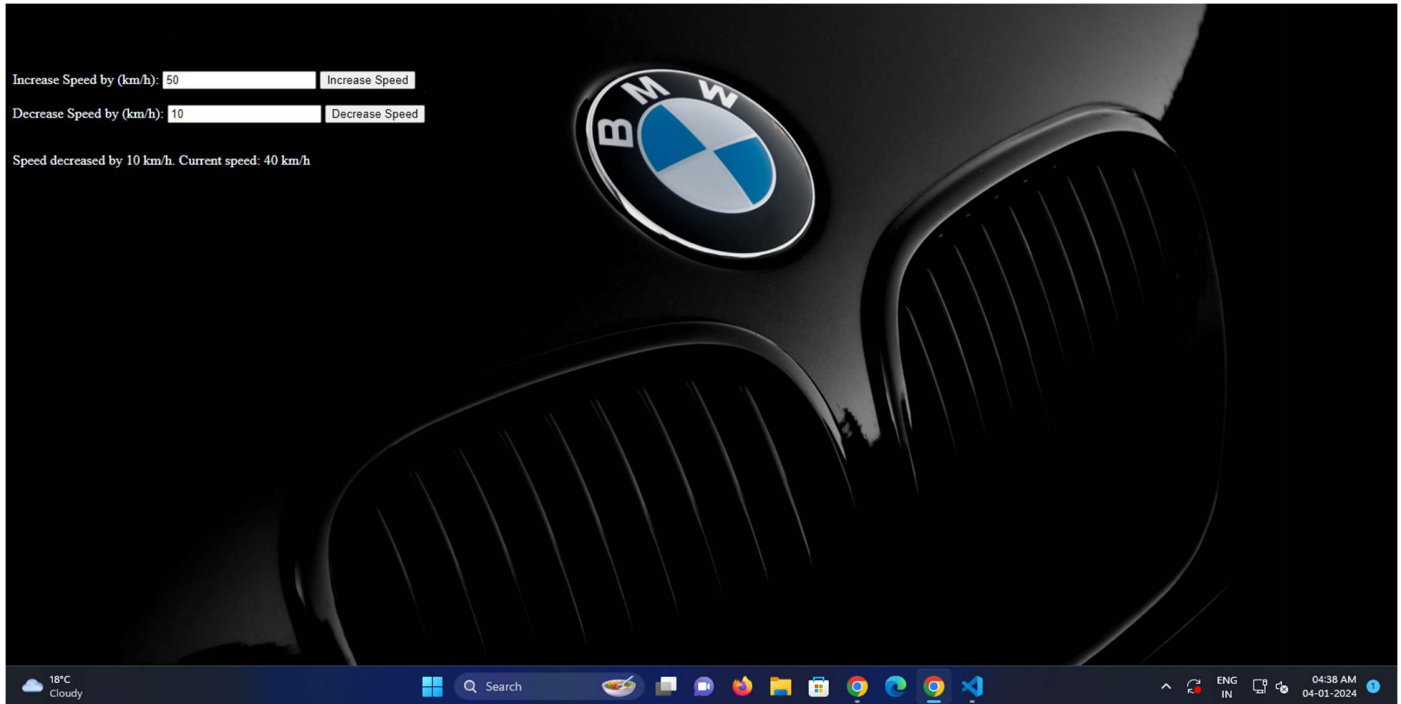
}
const myCar = new Car('BMW', 'Blue', 250000, 0);

function increaseSpeed() {
    const increaseInput = parseInt(document.getElementById('increaseSpeedInput').value);
    const result = myCar.increaseSpeed(increaseInput);
    document.getElementById('speedDisplay').innerText = result;
}

function decreaseSpeed() {
    const decreaseInput = parseInt(document.getElementById('decreaseSpeedInput').value);
    const result = myCar.decreaseSpeed(decreaseInput);
    document.getElementById('speedDisplay').innerText = result;
}
document.getElementById('speedDisplay').innerText = myCar.displaySpeed();
</script>
</body>
</html>
```

OUTPUT





Practical 2

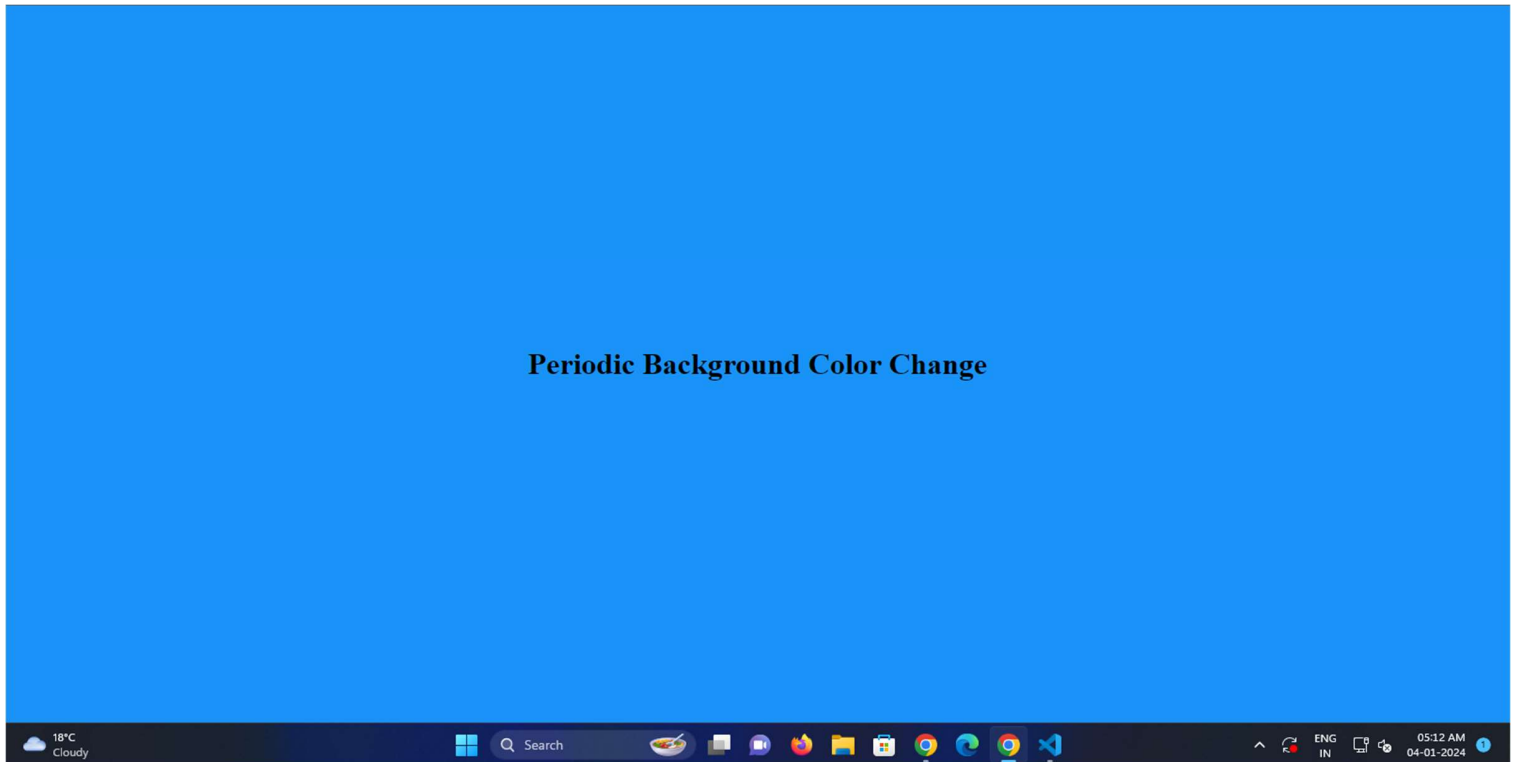
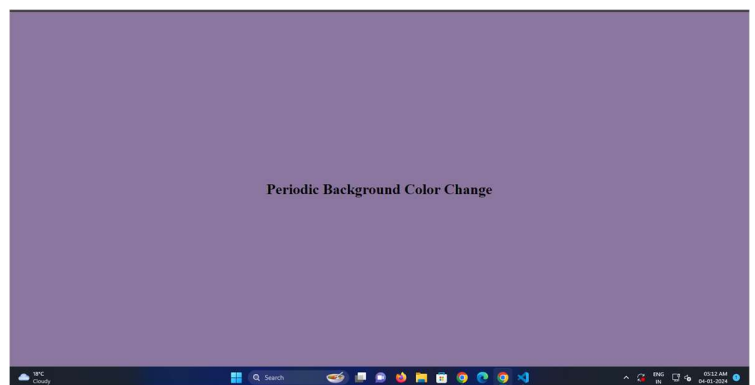
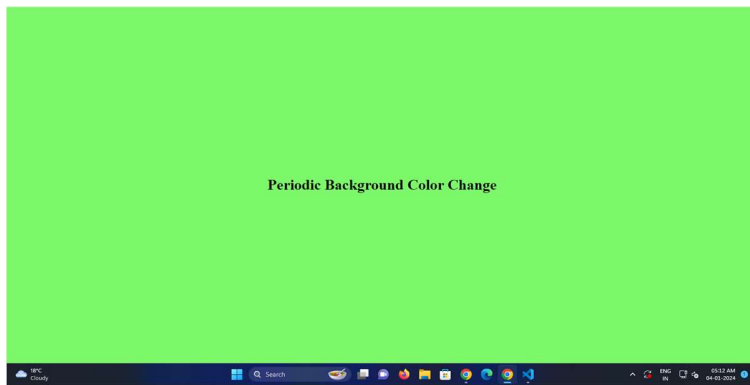
AIM : Write HTML/Javascript code to change the background color of div element periodically. For example, every 30 seconds it changes the background color.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Periodic Background Color Change</title>
  <style>
    body {
      transition: background-color 0.5s;
      margin: 0;
      height: 100vh;
      display: flex;
      align-items: center;
      justify-content: center;
      text-align: center;
    }
  </style>
</head>
<body>
  <h1>Periodic Background Color Change</h1>
  <script>
    function getRandomColor() {
      var letters = '0123456789ABCDEF';
      var color = '#';
      for (var i = 0; i < 6; i++) {
        color += letters[Math.floor(Math.random() * 16)];
      }
      return color;
    }
    function changeBackgroundColor() {
      document.body.style.backgroundColor = getRandomColor();
    }
  </script>
</body>
</html>
```

```
var intervalId = setInterval(changeBackgroundColor, 30000);  
changeBackgroundColor();
```

```
</script>  
</body>  
</html>
```

OUTPUT



AIM: Write HTML/Javascript code to make an animation for displaying set of images. It takes frequency (in milliseconds) of image repetition as input from the user. Also, it shows a button to start/stop (single button which toggles) the animation

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Image Animation</title>
  <style>
    #image-container {
      width: 100%;
      text-align: center;
    }
    #animation-button {
      margin-top: 10px;
      padding: 10px;
      font-size: 16px;
    }
  </style>
</head>
<body>

<div id="image-container"></div>
<button id="animation-button" onclick="toggleAnimation()">Start Animation</button>

<script>
  const imageContainer = document.getElementById('image-container');
  const animationButton = document.getElementById('animation-button');
  let isAnimating = false;
  let intervalId;

  const imagePaths = ['imagea.jpg', 'imageb.jpg', 'imagec.jpg'];

  let currentIndex = 0;

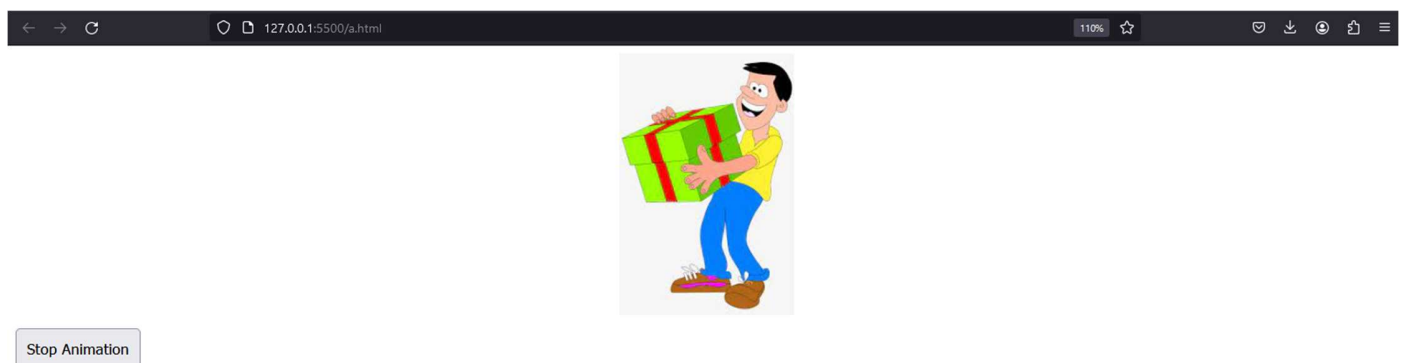
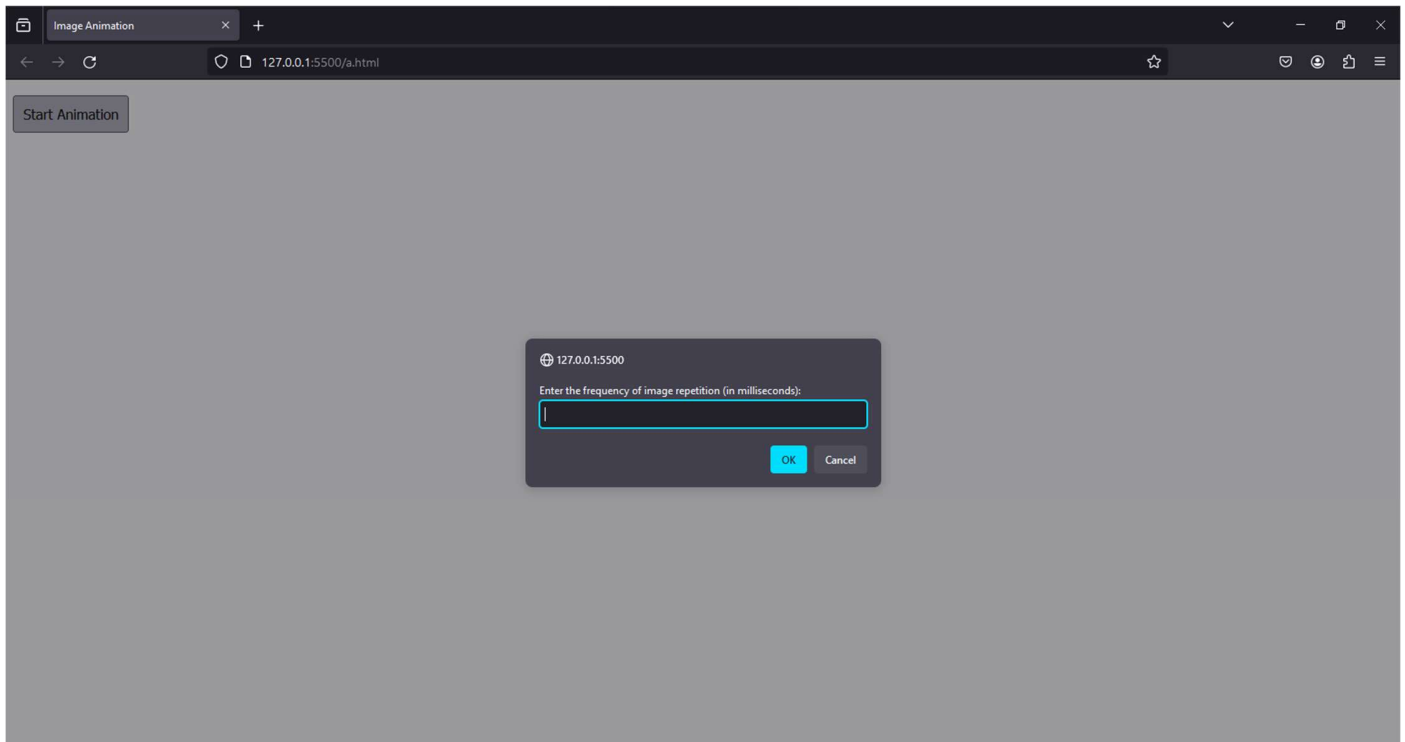
  function startAnimation(frequency) {
    isAnimating = true;
    animationButton.textContent = 'Stop Animation';

    intervalId = setInterval(() => {
      displayImage();
    }, frequency);
  }
  function stopAnimation() {
```

```
isAnimating = false;
animationButton.textContent = 'Start Animation';
clearInterval(intervalId);
}
function toggleAnimation() {
  if (isAnimating) {
    stopAnimation();
  } else {
    const frequency = prompt('Enter the frequency of image repetition (in
milliseconds:');
    if (frequency && !isNaN(frequency)) {
      startAnimation(parseInt(frequency, 10));
    } else {
      alert('Please enter a valid frequency.');
```

OUTPUT





Practical 3

AIM: Write AJAX script to display the student details like Enrollment Number and Name (in tabular form) from an XML file stored on the same web server. After displaying student details, when enrolment number is clicked then it fetches city, address, pincode and other details for the student.

```
<!DOCTYPE html>
<html>
<style>
    table, th, td {
        border: 1px solid black;
        border-collapse: collapse;
    }

    th, td {
        padding: 5px;
    }
</style>
<body>

    <h2>Student data</h2>

    <button type="button" onclick="loadDoc()">Get student data collection</button>
    <br><br>
    <table id="demo"></table>

    <script>
        function loadDoc() {
            const xhttp = new XMLHttpRequest();
            xhttp.onload = function () {
                myFunction(this);
            }
            xhttp.open("GET", "student.xml");
            xhttp.send();
        }
    </script>
```

```
function myFunction(xml) {  
    const xmlDoc = xml.responseXML;  
    const x = xmlDoc.getElementsByTagName("STUDENTDATA");  
    let table = "<tr><th>Enrollment Number</th><th>Name</th></tr>";  
    for (let i = 0; i < x.length; i++) {  
        const enrollmentNumber =  
x[i].getElementsByTagName("ERN")[0].childNodes[0].nodeValue;  
        table += "<tr id='row" + i + "'><td>" +  
            "<span id='enrollmentNumber" + i + "' style='cursor: pointer; color: black;'>" +  
enrollmentNumber +  
            "</span></td><td>" +  
x[i].getElementsByTagName("NAME")[0].childNodes[0].nodeValue +  
            "</td></tr>";  
    }  
    document.getElementById("demo").innerHTML = table;  
  
    for (let i = 0; i < x.length; i++) {  
        const enrollmentNumber =  
x[i].getElementsByTagName("ERN")[0].childNodes[0].nodeValue;  
        document.getElementById("enrollmentNumber" + i).addEventListener("click", function  
( ) {  
            alert("Details for enrollment number " + enrollmentNumber + ":\n\n" +  
                "Name: " + x[i].getElementsByTagName("NAME")[0].childNodes[0].nodeValue + "\n"  
+  
                "City: " + x[i].getElementsByTagName("CITY")[0].childNodes[0].nodeValue + "\n"  
+  
                "Department: " + x[i].getElementsByTagName("DEP")[0].childNodes[0].nodeValue);  
        });  
    }  
}  
  
</script>  
</body>  
</html>  
STUDENT.XML:  
<STUDENT>  
    <STUDENTDATA>  
        <ERN>101</ERN>
```



A D PATEL INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY
102046701-Advanced Web Development



```
<NAME>Bob Dylan</NAME>
<CITY>Delhi</CITY>
<DEP>FINANCE</DEP>
</STUDENTDATA>
<STUDENTDATA>
  <ERN>102</ERN>
  <NAME>Boa Dylan</NAME>
  <CITY>Cicago</CITY>
  <DEP>AUTOMOBILE</DEP>
</STUDENTDATA>
<STUDENTDATA>
  <ERN>103</ERN>
  <NAME>Boc</NAME>
  <CITY>Chennai</CITY>
  <DEP>CSD</DEP>
</STUDENTDATA>
<STUDENTDATA>
  <ERN>104</ERN>
  <NAME>Bod Dylan</NAME>
  <CITY>surat</CITY>
  <DEP>IT</DEP>
</STUDENTDATA>
</STUDENT>
```



Student data

Get student data collection



Student data

Get student data collection

Enrollment Number	Name
101	Bob Dylan
102	Boa
103	Boc
104	Bod

Practical 4

AIM: Design Order Form with a total price updated in real time, which contains name of five products and their prices. Create a bill amount for all the products and calculate GST on the billing amount and display total amount. Use AngularJS.

```
<!DOCTYPE html>
<html lang="en" ng-app="orderFormApp">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Order Form</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script src="app.js"></script>
</head>
<body>
  <div ng-controller="OrderFormController as orderCtrl">
    <h2>Order Form</h2>
    <table>
      <tr>
        <th>Product</th>
        <th>Price</th>
        <th>Quantity</th>
        <th>Total</th>
      </tr>
      <tr ng-repeat="product in orderCtrl.products">
        <td>{{ product.name }}</td>
        <td>{{ product.price | currency:'₹' }}</td>
        <td><input type="number" ng-model="product.quantity" ng-
change="orderCtrl.calculateTotal()"></td>
        <td>{{ product.total | currency:'₹' }}</td>
      </tr>
    </table>
    <p>Total Amount: {{ orderCtrl.totalAmount | currency:'₹' }}</p>
    <p>GST (10%): {{ orderCtrl.calculateGST() | currency:'₹' }}</p>
    <p>Total Amount + GST: {{ orderCtrl.totalAmountWithGST() | currency:'₹' }}</p>
  </div>
</body> </html>
```

App.js

```
angular.module('orderFormApp', [])
.controller('OrderFormController', function() {
    var vm = this;

    vm.products = [
        { name: 'Product 1', price: 10, quantity: 0, total: 0 },
        { name: 'Product 2', price: 20, quantity: 0, total: 0 },
        { name: 'Product 3', price: 15, quantity: 0, total: 0 },
        { name: 'Product 4', price: 25, quantity: 0, total: 0 },
        { name: 'Product 5', price: 30, quantity: 0, total: 0 }
    ];

    vm.calculateTotal = function() {
        vm.totalAmount = 0;
        angular.forEach(vm.products, function(product) {
            product.total = product.price * product.quantity;
            vm.totalAmount += product.total;
        });
    };

    vm.calculateGST = function() {
        return vm.totalAmount * 0.1;
    };

    vm.totalAmountWithGST = function() {
        return vm.totalAmount + vm.calculateGST();
    };
});
```

Order Form

Product	Price	Quantity	Total
Product 1	₹10.00	4	₹40.00
Product 2	₹20.00	2	₹40.00
Product 3	₹15.00	2	₹30.00
Product 4	₹25.00	2	₹50.00
Product 5	₹30.00	4	₹120.00

Total Amount: ₹280.00

GST (10%): ₹28.00

Total Amount + GST: ₹308.00

AIM: Write an AngularJS filter to check whether the entered number in the textbox is prime or not.

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

    <title>Practical-5</title>

</head>

<body>

    <div ng-app="myApp">

        <p>Enter a number:

        <input ng-model="number"></p><br/>

        Number you've entered is:{{number| prime}}

    </div>

    <script>

        var app=angular.module('myApp', []);

        app.filter('prime', function(){
```



```
var txt="prime";

return function(x) {

    for(var i=2; i<x; i++){

        if(x%i==0) {

            txt="not prime"

            break;

        }

    }

    return txt;

};

});
```

```
</script>
```

```
</body>
```

```
</html>
```

OUTPUT

Enter a number:

Number you've entered is: not prime

Practical - 6

AIM: Write AngularJS service which makes the square of the number when it is called.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
  <title>Practical-6</title>
</head>
<body> <div ng-app="myApp">
  Enter a number:<input ng-model="number"><br/>
  Square of {{number}} is:<square-of-the-number></square-of-the-number> </div>
<script>
  var app=angular.module("myApp", []);
  app.directive('squareOfTheNumber', function() {
    return {
      template: "{{number*number}}";});
</script>
</body>
</html>
```

OUTPUT

Enter a number:
Square of 126 is:15876

Practical - 7

AIM: Write AngularJS service which makes the addition/subtraction of two numbers entered in textboxes when the d Sub button is clicked.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src =
'https://ajax.googleapis.com/ajax/libs/angularjs/1.5.2/angular.min.js'></script>
  <title>Practical-7</title>
</head>
<body>
  <div ng-app="myApp" ng-controller="myCtrl">
    <label>Enter Number A:</label>
    <input type="text" ng-model="numA"><br>
    <label>Enter Number B:</label>
    <input type="text" ng-model="numB"><br>
    <button ng-click="calculate()">Sub</button><br/>
    <label>Result:</label>
    Subtraction is:<input type="text" ng-model="result1" disabled><br/>
    Addition is:<input type="text" ng-model="result2" disabled>
  </div>
  <script>
    var app=angular.module("myApp", []);
    app.controller('myCtrl', function($scope, CalculatorService) {
      $scope.calculate = function() {
        var result1 = CalculatorService.subtract(parseInt($scope.numA),
parseInt($scope.numB));
        $scope.result1 = result1;
        var result2 = CalculatorService.add(parseInt($scope.numA),
parseInt($scope.numB));
        $scope.result2 = result2;
      };
    });
    app.service('CalculatorService', function() {
      this.add = function(a, b) {
        return a + b;
      };
    });
  </script>
</body>
</html>
```

```
};  
this.subtract = function(a, b) {  
    return a - b;  
};  
});  
</script>  
</body>  
</html>
```

OUTPUT

Enter Number A:

Enter Number B:

Result: Subtraction is:

Addition is:

Practical - 8

AIM: Write AngularJS script to read the JSON data (Student's enrolment number and name) from the server and populate the dropdown list with enrolment number.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src =
'https://ajax.googleapis.com/ajax/libs/angularjs/1.5.2/angular.min.js'></script>
  <title>Practical-8</title>
</head>
<body>
  <div ng-app="myApp" ng-controller="myCtrl">
    <select ng-model="selectedEnrolmentNumber" ng-options="enrolmentNumber for
enrolmentNumber in enrolmentNumbers">
      {enrolmentNumber.enrollment}
    </select>
  </div>
  <script>
    var app = angular.module('myApp', []);
    app.controller('myCtrl', function($scope, $http) {
      $http.get('studentData.json')
        .then(function(response) {
          $scope.students = response.data;
          $scope.enrolmentNumbers = [];
          for (var i = 0; i < $scope.students.length; i++) {
            $scope.enrolmentNumbers.push($scope.students[i].enrollment);
          }
          $scope.selectedEnrolmentNumber = $scope.enrolmentNumbers[0];
        }, function(error) {
          console.error(error);
        });
    });
  </script>
</body>
</html>
```



A D PATEL INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY
102046701-Advanced Web Development
OUTPUT



12002040501001 ▾

- 12002040501001
- 12002040501002
- 12002040501003
- 12002040501004
- 12002040501005
- 12002040501006
- 12002040501007
- 12002040501008
- 12002040501009
- 12002040501010
- 12002040501011
- 12002040501012
- 12002040501013
- 12002040501014

Practical - 9

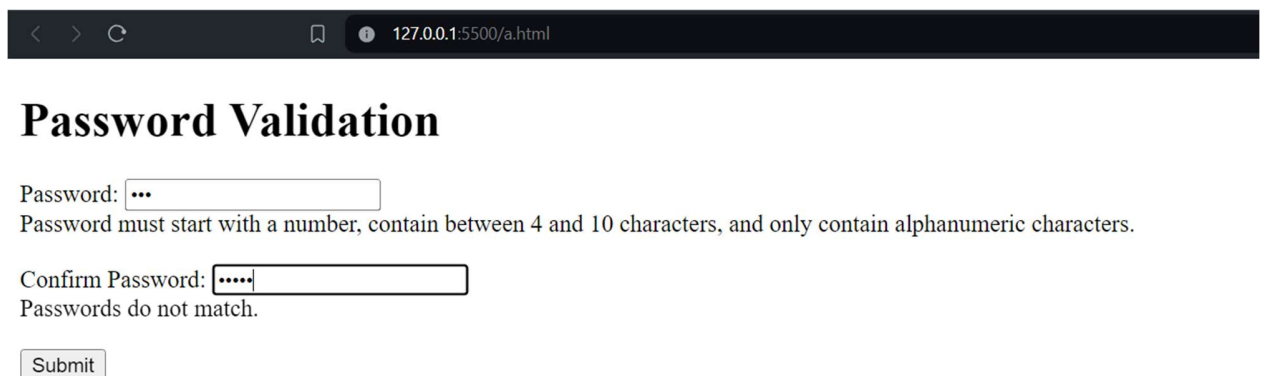
AIM: Write AngularJS validation script to validate the password and confirm password. Password must contain at least four and maximum 10 characters and must start with a number only. Content of the password and confirm password box must be the same otherwise it should display the appropriate message.

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
  <meta charset="utf-8">
  <title>Practical-9</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
</head>
<body ng-controller="myCtrl">
  <h1>Password Validation</h1>
  <form name="myForm" novalidate>
    Password:
    <input type="password" id="password" name="password" ng-model="password" ng-
pattern="/^[0-9][a-zA-Z0-9]{3,9}$/" ng-maxlength="10" required>
    <div ng-show="myForm.password.$error.required">Password is required.</div>
    <div ng-show="myForm.password.$error.pattern">{{ passwordError }}</div>
    <div ng-show="myForm.password.$error.maxlength">Password must be 10 characters or
less.</div>
    <br>
    Confirm Password:
    <input type="password" id="confirmPassword" name="confirmPassword" ng-
model="confirmPassword" ng-change="validatePassword()" required>
    <div ng-show="myForm.confirmPassword.$error.required">Confirm password is
required.</div>
    <div ng-show="confirmPasswordError">{{ confirmPasswordError }}</div>
    <br>
    <button type="submit">Submit</button>
  </form>
  <script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
  $scope.password = '';
  $scope.confirmPassword = '';

```

```
$scope.validatePassword = function() {  
    var passwordPattern = /^[0-9][a-zA-Z0-9]{3,9}$/;  
    if (!passwordPattern.test($scope.password)) {  
        $scope.passwordError = 'Password must start with a number, contain between  
4 and 10 characters, and only contain alphanumeric characters.';  
    } else {  
        $scope.passwordError = '';  
    }  
    if ($scope.password !== $scope.confirmPassword) {  
        $scope.confirmPasswordError = 'Passwords do not match.';  
    } else {  
        $scope.confirmPasswordError = '';  
    }  
};  
});  
</script>  
</body>  
</html>
```

OUTPUT



127.0.0.1:5500/a.html

Password Validation

Password:

Password must start with a number, contain between 4 and 10 characters, and only contain alphanumeric characters.

Confirm Password:

Passwords do not match.

Practical - 10

AIM: Write AngularJS validation script to validate the registration form and submit button gets enabled if all the validations are passed successfully. The form contains fields like First name, last name, email address, mobile number, sex (radio button) and department (select box).

```
<!doctype html>
<html ng-app="registrationFormApp">
<head>
  <title>Registration Form</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
  <div ng-controller="registrationFormCtrl">
    <form name="registrationForm" novalidate>
      First Name:
      <input type="text" name="firstName" ng-model="formData.firstName" required>
      <div ng-show="registrationForm.firstName.$error.required &&
registrationForm.firstName.$dirty">
        First Name is required.
      </div><br>
      Last Name:
      <input type="text" name="lastName" ng-model="formData.lastName" required>
      <div ng-show="registrationForm.lastName.$error.required &&
registrationForm.lastName.$dirty">
        Last Name is required.
      </div><br>
      Email Address:
      <input type="email" name="email" ng-model="formData.email" required>
      <div ng-show="registrationForm.email.$error.required &&
registrationForm.email.$dirty">
        Email is required.
      </div>
      <div ng-show="registrationForm.email.$error.email &&
registrationForm.email.$dirty">
        Invalid email format.
      </div><br>
      Mobile Number:
```

```
<input type="tel" name="mobile" ng-model="formData.mobile" ng-pattern="/^[0-9]{10}$/" required>
<div ng-show="registrationForm.mobile.$error.required &&
registrationForm.mobile.$dirty">
    Mobile Number is required.
</div>
<div ng-show="registrationForm.mobile.$error.pattern &&
registrationForm.mobile.$dirty">
    Invalid mobile number format. (10 digits only)
</div><br>
Sex:
<input type="radio" name="sex" ng-model="formData.sex" value="male" required>
Male
<input type="radio" name="sex" ng-model="formData.sex" value="female"
required> Female
<div ng-show="registrationForm.sex.$error.required &&
registrationForm.sex.$dirty">
    Please select your sex.
</div><br>
Department:
<select name="department" ng-model="formData.department" ng-
options="department for department in departments" required>
    <option value="">-- Select Department --</option>
</select>
<div ng-show="registrationForm.department.$error.required &&
registrationForm.department.$dirty">
    Please select your department.
</div><br>
<input type="submit" ng-disabled="registrationForm.$invalid"/>
</form>
</div>
<script>
    var app = angular.module('registrationFormApp', []);
    app.controller('registrationFormCtrl', function($scope) {
        $scope.departments = ['CP', 'IT', 'EC', 'CH', 'EE', 'CSD', 'IOT'];
    });
</script>
</body>
</html>
```

OUTPUT

First Name:
Last Name:
Email Address:
Mobile Number:
Invalid mobile number format. (10 digits only)

Sex: ☒ Male ☐ Female

Department: 

Practical - 11

AIM: Write AngularJS script to implement a single page web application. Make your own website which includes pages like about you, education, hobbies, contact me etc.

```
<!DOCTYPE html>
<html lang="en" ng-app="myApp">
<head>
  <meta charset="UTF-8">
  <title>My Single-Page Web Application</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-
route.js"></script>
  <script src="app.js"></script>
</head>
<body>
  <div ng-app="myApp">
    <ul>
      <li><a href="#/">Home</a></li>
      <li><a href="#!about">About Me</a></li>
      <li><a href="#!education">Education</a></li>
      <li><a href="#!hobbies">Hobbies</a></li>
      <li><a href="#!contact">Contact Me</a></li>
    </ul>
  </div>
  <main ng-view></main>
  <footer>
    <p>© 2023 My Single-Page Web Application</p>
  </footer>
  <script>
    var app = angular.module("myApp", ["ngRoute"]);
    app.config(function($routeProvider) {
      $routeProvider
        .when("/", {
          templateUrl : "home.htm"
        })
        .when("/about", {
          templateUrl : "about.htm"
        })
        .when("/education", {
```

```
        templateUrl : "education.htm"
    })
    .when("/hobbies", {
        templateUrl : "hobbies.htm"
    })
    .when("/contact", {
        templateUrl : "contact.htm"
    })
    .otherwise({redirectTo: '/'});
});
</script>
</body>
</html>
```

OUTPUT

- [Home](#)
- [About Me](#)
- [Education](#)
- [Hobbies](#)
- [Contact Me](#)

© 2023 My Single-Page Web Application

Practical - 12

AIM: Write AngularJS script to implement custom directive (reusable component) for calculator with operations like add, subtraction and multiplication.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <title>Practical-12</title>
</head>
<body>
  <div ng-app="myApp">
    <calculator></calculator>
  </div>
  <script>
    angular.module('myApp', []).directive('calculator', function() {
      return {
        restrict: 'E',
        template: '<div>' +
          '<input type="number" ng-model="num1">' +
          '<select ng-model="operator">' +
          '<option value="+">+</option>' +
          '<option value="-">-</option>' +
          '<option value="*">*</option>' +
          '</select>' +
          '<input type="number" ng-model="num2">' +
          '<button ng-click="calculate()">Calculate</button>' +
          '<p>Result: {{result}}</p>' +
          '</div>',
        controller: function($scope) {
          $scope.calculate = function() {
            if ($scope.operator === '+') {
              $scope.result = $scope.num1 + $scope.num2;
            } else if ($scope.operator === '-') {
              $scope.result = $scope.num1 - $scope.num2;
            } else if ($scope.operator === '*') {
```

```
$scope.result = $scope.num1 * $scope.num2;  
}  
};  
}  
};  
});  
</script>  
</body>  
</html>
```

OUTPUT

1232	- v	123	Calculate
------	-----	-----	-----------

Result: 1109

+

-

*

123	+ v	123	Calculate
-----	-----	-----	-----------

Result: 246

Practical - 13

AIM: Write NodeJS code to implement a web server to serve the different content based on URL.

```
const http = require('http');

const server = http.createServer((req, res) => {
  if (req.url === '/') {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.write('<h1>Welcome to the homepage</h1>');
    res.end();
  } else if (req.url === '/about') {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.write('<h1>About Us</h1><p>We are a company that sells widgets.</p>');
    res.end();
  } else if (req.url === '/contact') {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.write('<h1>Contact Us</h1><p>You can reach us at example@example.com</p>');
    res.end();
  } else {
    res.writeHead(404, { 'Content-Type': 'text/html' });
    res.write('<h1>404 Page Not Found</h1>');
    res.end();
  }
});

server.listen(3000, () => {
  console.log('Server running on port 3000');
});
```

OUTPUT



Practical - 14

AIM: Write NodeJS code to provide the file upload facility to the user. It also imposes the restriction of file size to be uploaded.

```
const http = require('http');
const formidable = require('formidable');

const server = http.createServer((req, res) => {
  if (req.url === '/upload' && req.method.toLowerCase() === 'post') {
    const form = new formidable.IncomingForm({
      uploadDir: './pract11',
      maxFileSize: 5 * 1024 * 1024
    });
    form.parse(req, (err, fields, files) => {
      if (err) {
        console.error(err);
        res.writeHead(500, { 'Content-Type': 'text/plain' });
        res.write('An error occurred during file upload');
        res.end();
      } else {
        const file = files.upload;
        console.log(`File uploaded: ${file.name} (${file.size} bytes)`);
        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.write(`<h1>File uploaded successfully</h1><p>Filename:
${file.name}</p><p>Size: ${file.size} bytes</p>`);
        res.end();
      }
    });
  } else {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.write(`
<form action="/upload" method="post" enctype="multipart/form-data">
  <input type="file" name="upload"><br>
  <input type="submit" value="Upload">
</form>
`);
    res.end();
  }
});

server.listen(3000, () => {
```



A D PATEL INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY
102046701-Advanced Web Development



```
console.log('Server running on port 3000');  
});
```

OUTPUT

Choose File nodejs.pptx
Upload

File uploaded successfully

Filename: undefined

Size: 3198918 bytes

Practical - 15

AIM: Create a simple web application using AngularJS, NodeJS and MongoDB. For example, you may design Library Management System, Hospital Management System etc. with minimal functionalities.

Home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.1/css/all.min.css" integrity="sha512-
MV7K8+y+gLIBoVD59lQIYicR65iaqukzvf/nwasF0nqhPay5w/9lJmVM2hMDcnK1OnMGCdVK+iQrJ7lzpPJQdlw=="
crossorigin="anonymous" referrerpolicy="no-referrer" />
  <title>Weather App</title>
</head>
<body>
  <div class="box">
    <div class="wave">
      <div class="wave-one"></div>
      <div class="wave-two"></div>
      <div class="wave-three"></div>
    </div>
    <div id="weathercon">
      <i class="fa-solid fa-sun" style="color:#eccc68"></i>
    </div>
    <div class="info">
      <h2 class="location"><i class="fa-solid fa-street-
view"></i>{%location%},{%country%}</h2>
      <p id="date">TUE | JAN 10 | 9:48PM</p>
      <h1 class="temp">{%tempVal%}&deg;C</h1>
      <h3 class="tempMinMax">MIN {%tempmin%}&deg;C | MAX {%tempmax%}&deg;C</h3>
    </div>
  </div>
  <script>
    const currDate=document.getElementById('date');
    let weathercon=document.getElementById('weathercon');
```

```
const tempStatus="{%tempstatus%}";
if(tempStatus=="Sunny") {
    weathercon.innerHTML("<i class='fa-solid fa-sun' style='color:#eccc68'></i>");
}else
if(tempStatus=="Clouds") {
    weathercon.innerHTML("<i class='fa-solid fa-cloud'
style='color:#f1f2f6'></i>");
}else
if(tempStatus=="Rainy") {
    weathercon.innerHTML("<i class='fa-solid fa-cloud-rain'
style='color:#a4b0be'></i>");
}else{
    weathercon.innerHTML("<i class='fa-solid fa-cloud'
style='color:#44c3de'></i>");
}
const getCurrentDay=()=>{
    var weekday=new Array(7);
    weekday[0]="Sunday";
    weekday[1]="Monday";
    weekday[2]="Tuesday";
    weekday[3]="Wednesday";
    weekday[4]="Thursday";
    weekday[5]="Friday";
    weekday[6]="Saturday";
    let currentTime=new Date();
    let day=weekday[currentTime.getDay()];
    return day;
};
const getCurrentTime=()=>{
    var months=[
        "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"
    ];
    var now=new Date();
    var month=months[now.getMonth()];
    var date=now.getDate();
    var hours=now.getHours();
    var min=now.getMinutes();
    let period="AM";
    if(hours>11) {
```

```
        period="PM";
        if(hours>12){
            hours-=12;
        }
    }
    if(min<10){
        mins="0"+min;
    }
    return `${month} ${date} | ${hours}:${min}${period}`;
}
currDate.innerHTML=getCurrentDay()+"|"+getCurrentTime();
</script>
</body>
</html>
```

Index.js

```
const http=require("http");
const fs=require("fs");
var requests=require('requests');
const replaceVal=(tempVal, orgVal)=>{
    let temperature=tempVal.replace("%tempVal%",orgVal.main.temp);
    temperature=temperature.replace("%tempmin%",orgVal.main.temp_min);
    temperature=temperature.replace("%tempmax%",orgVal.main.temp_max);
    temperature=temperature.replace("%location%",orgVal.name);
    temperature=temperature.replace("%country%",orgVal.sys.country);
    temperature=temperature.replace("%tempstatus%",orgVal.weather[0].main);
    return temperature;
}

const homeFile=fs.readFileSync('home.html','utf8');
const server=http.createServer((req, res)=>{
    if(req.url=='/'){
        requests('https://api.openweathermap.org/data/2.5/weather?q=Surat&appid=dfdcde5d3679f9c9d7ddd8f9b63a129f')
            .on('data',(chunk)=>{
                const objData=JSON.parse(chunk);
                const arrData=[objData];
```

```
// console.log(arrData[0].main.temp);  
const realTime=arrData.map(val=>replaceVal(homeFile, val)).join("");  
res.write(realTime);  
// console.log(realTime);  
}).on('end', (err) => {  
  if(err) return console.log("Connection closed",err);  
  res.end();  
});  
}  
});  
server.listen(8000,"127.0.0.1");
```

OUTPUT

