# Lab Exercise- 1

## IT-23: Advanced Internet Technologies

## Name: Harshal Bhaskarrao Wandhare

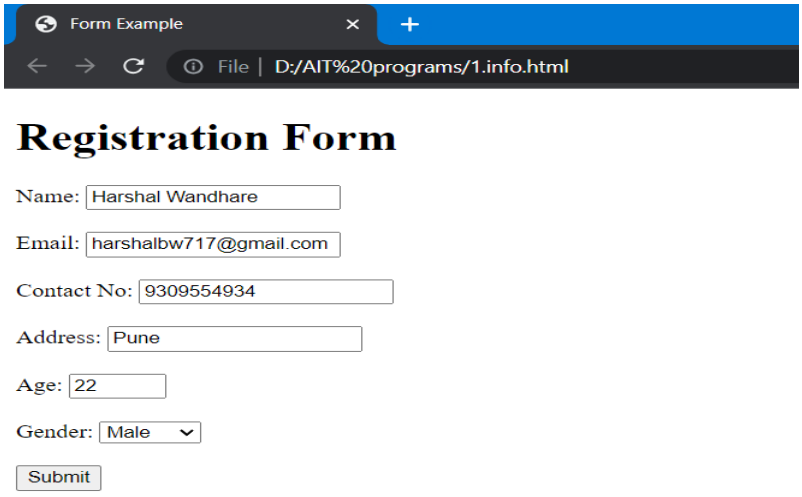
## Roll no: 74

1. **Write a Program to design form using HTML5 elements, attributes and semantics.**

**Code:**

```
<html>
<head>
  <title>Form Example</title>
</head>
<body>
  <h1>Registration Form</h1>
  <form action="/submit" method="post">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required><br><br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required><br><br>
    <label for="Contact">Contact No:</label>
    <input type="text" id="Contact" name="contant" required><br><br>
    <label for="address">Address:</label>
    <input type="text" id="address" name="address" required><br><br>
    <label for="age">Age:</label>
    <input type="number" id="age" name="age" min="18" max="99" required><br><br>
    <label for="gender">Gender:</label>
    <select id="gender" name="gender" required>
      <option value="">Select</option>
      <option value="male">Male</option>
      <option value="female">Female</option>
      <option value="other">Other</option>
    </select><br><br>
    <button type="submit">Submit</button>
  </form>
</body>
</html>
```

**Output:**

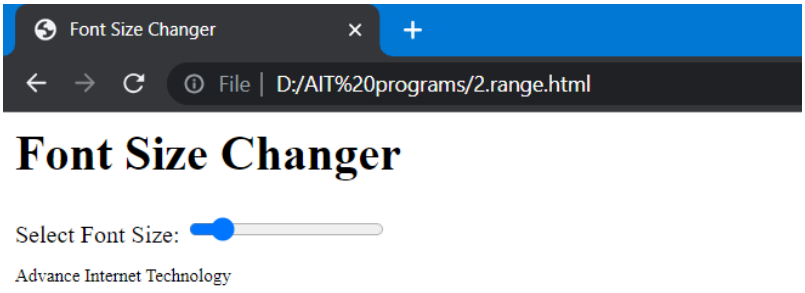


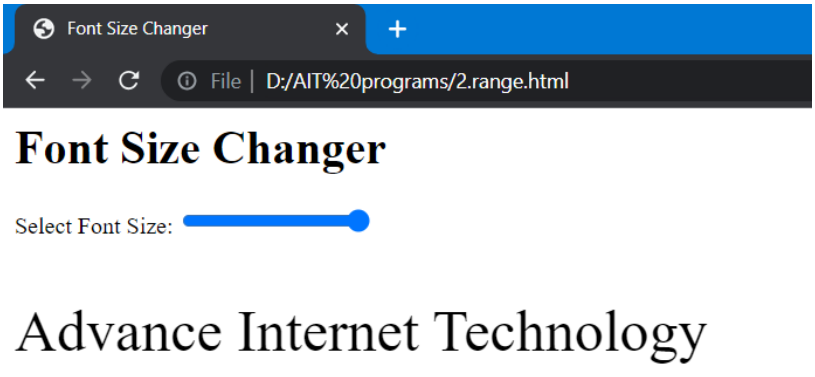
2. **WAP to change font size of a text as per the ‘range’ control value, ranging from 8 to 40.**

**Code:**

```
<html>
<head>
  <title>Font Size Changer</title>
</head>
<body>
  <h1>Font Size Changer</h1>

  <label for="font-size-range">Select Font Size:</label>
  <input type="range" id="font-size-range" min="8" max="40" value="16">

  <p class="text" id="text">Advance Internet Technology</p>

  <script>
    const fontSizeRange = document.getElementById('font-size-range');
    const text = document.getElementById('text');

    fontSizeRange.addEventListener('input', function() {
      const fontSize = fontSizeRange.value;
      text.style.fontSize = fontSize + 'px';
    });
  </script>
</body>
</html>
```

**Output:**



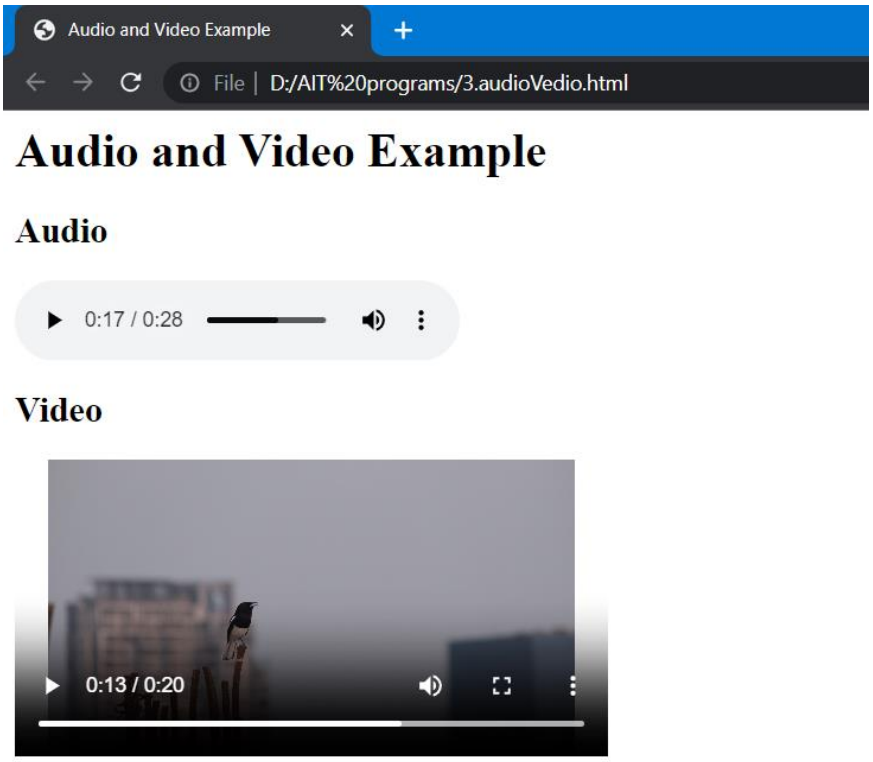



3. **Write a program to embed audio and video elements use controls.**

**Code:**

```
<html>
<head>
  <title>Audio and Video Example</title>
</head>
<body>
  <h1>Audio and Video Example</h1>

  <h2>Audio</h2>
  <audio src="Bird Sound Message Tune.mp3" controls>
    Your browser does not support the audio element.
  </audio>

  <h2>Video</h2>
  <video src="bird tune.mp4" controls height="200" width="400">
    Your browser does not support the video element.
  </video>
</body>
</html>
```

**Output:**

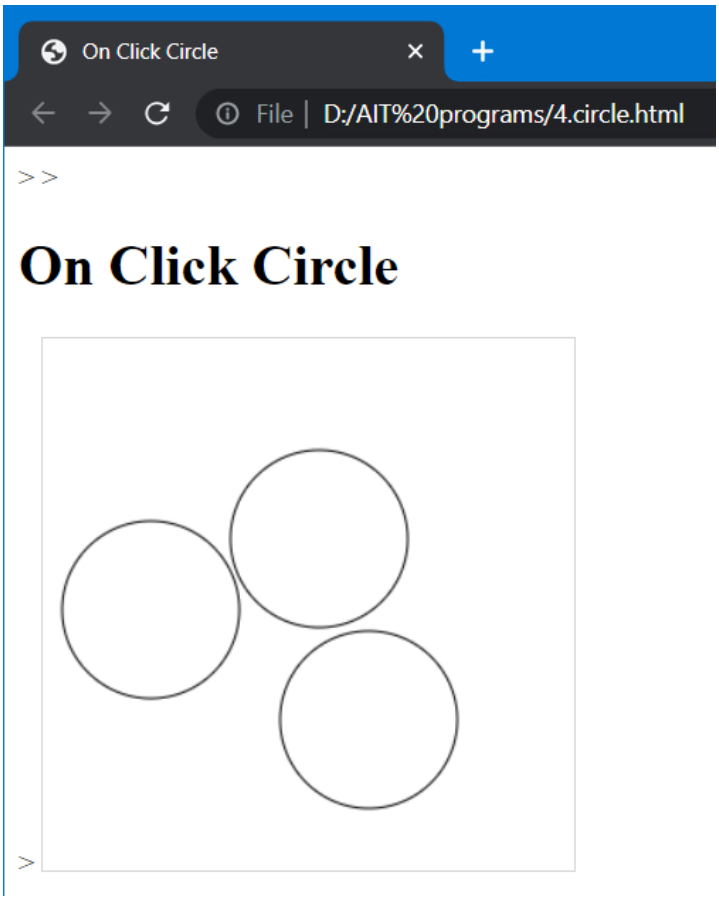


4. **WAP to draw a circle considering mouse click position as a centre.(Radius can be considered as 50)**

**Code:**

```
<html>
<head>
   <title>On Click Circle</title>>
</head>>
<body>
   <h1>On Click Circle</h1>>
   <canvas id="myCanvas" width="300" height="300" style="border:1px solid #d3d3d3;"
   onclick="showCoords(event)"></canvas>
   <script>
   function showCoords(event) {
   var x = event.clientX;
   var y = event.clientY;
   var c = document.getElementById("myCanvas");
   var ctx = c.getContext("2d");
   ctx.beginPath();
   ctx.arc(x, y, 50, 0, 2 * Math.PI);
   ctx.stroke();
   }
   </script>
   <canvas>
</body>
</html>
```

**Output:**

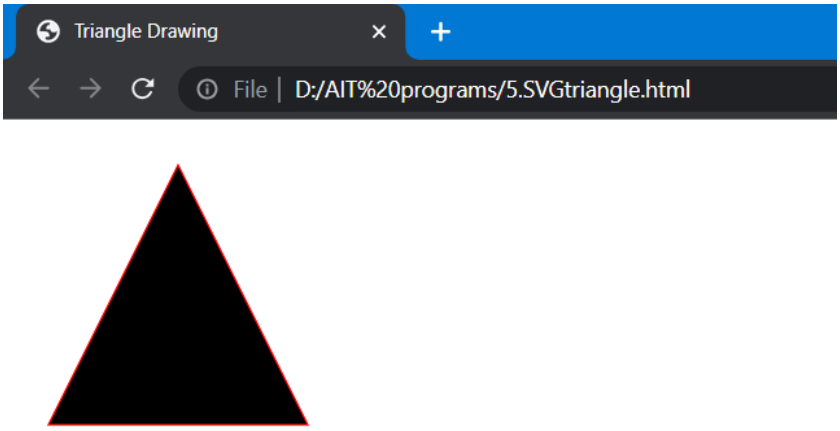


5. **WAP to draw triangle using SVG and canvas-path.**

**Code:**

```
<!DOCTYPE html>
<html>
<head>
   <title>Triangle Drawing</title>
</head>
<body>
   <svg xmlns="http://www.w3.org/2000/svg" width="200" height="200">
      <path d="M 100 20 L 20 180 L 180 180 Z" fill="coloe:black" stroke="red" />
   </svg>
</body>
</html>
```

**Output:**

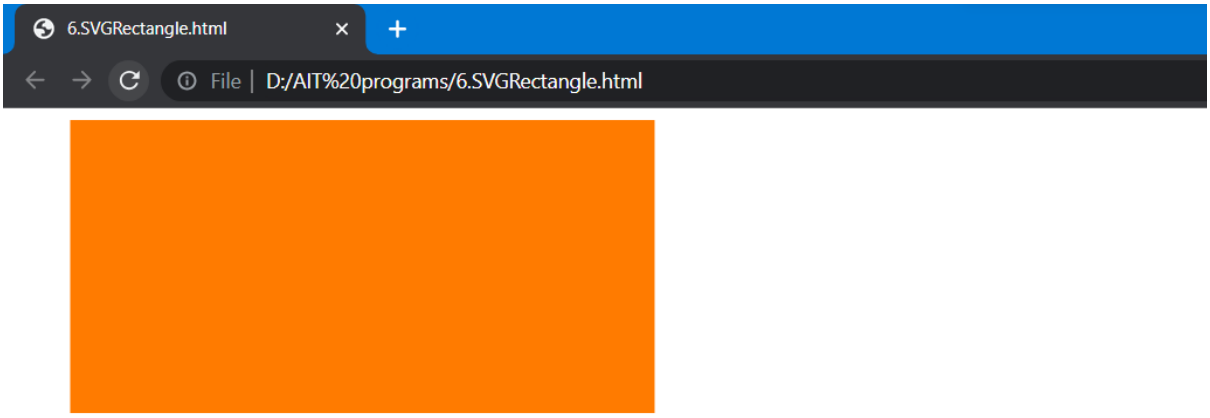


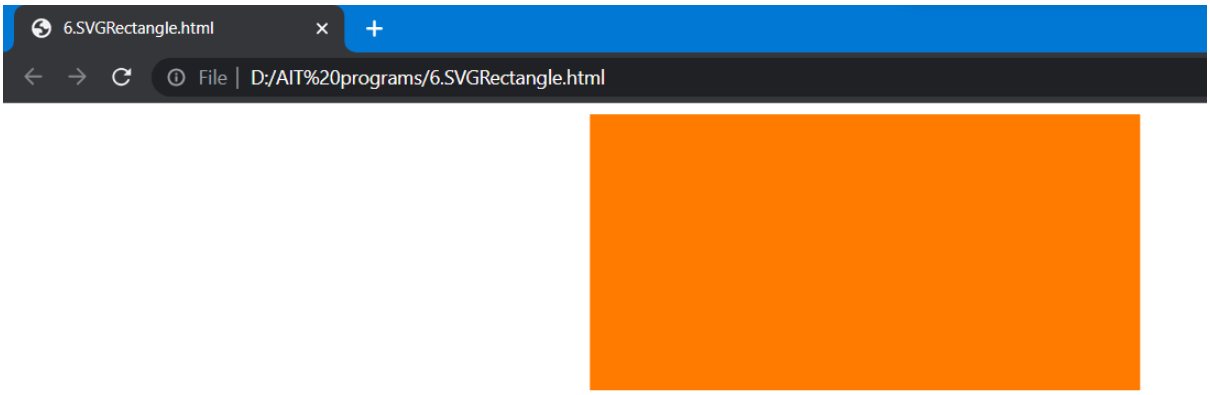
6. **Write a program to draw the rectangle shape using SVG and apply animation effect.**

**Code:**

```
<html>
<body>
<svg width="2000" height="1000" >
<rect width="400" height="200" style= "fill:rgb(255, 123, 0)">
<animate attributeName="x" from="0" to="600" dur="8s" repeatCount= "indefinite"/>
</rect>
</svg>
</body>
</html>
```

**Output:**



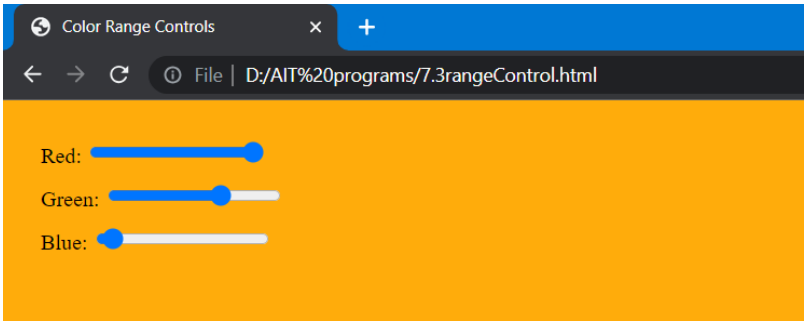



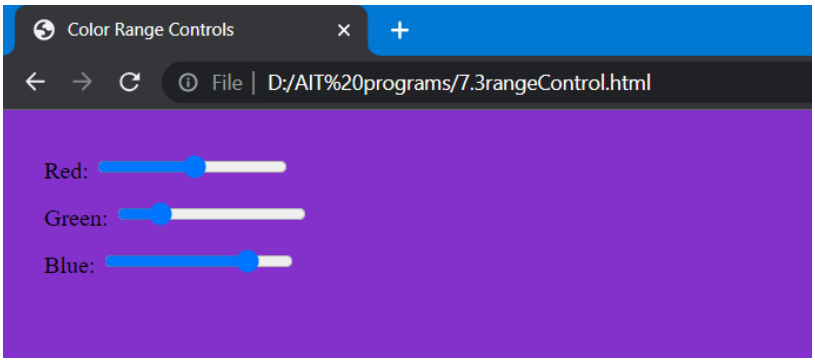
**7. WAP to add 3 ‘range’ control for colors red, green, blue. Change the background color as the valuesof range. Start the values from 0 to maximum 256.**

**Code:**

```
<!DOCTYPE html>
<html>
<head>
    <title>Color Range Controls</title>
    <style>
        body {
            padding: 20px;
        }
        .color-control {
            margin-bottom: 10px;
        }
    </style>
</head>
<body>
    <div class="color-control">
        <label for="redRange">Red:</label>
        <input type="range" id="redRange" min="0" max="256" value="0">
    </div>
    <div class="color-control">
        <label for="greenRange">Green:</label>
        <input type="range" id="greenRange" min="0" max="256" value="0">
    </div>
    <div class="color-control">
        <label for="blueRange">Blue:</label>
        <input type="range" id="blueRange" min="0" max="256" value="0">
    </div>

    <script>
        const redRange = document.getElementById("redRange");
        const greenRange = document.getElementById("greenRange");
        const blueRange = document.getElementById("blueRange");

        function changeBackgroundColor() {
            const redValue = redRange.value;
            const greenValue = greenRange.value;
            const blueValue = blueRange.value;

            const backgroundColor = `rgb(${redValue}, ${greenValue}, ${blueValue})`;
            document.body.style.backgroundColor = backgroundColor;
        }
        redRange.addEventListener("input", changeBackgroundColor);
        greenRange.addEventListener("input", changeBackgroundColor);
        blueRange.addEventListener("input", changeBackgroundColor);
    </script>
</body>
</html>
```

**Output:**


Color Range Controls
File | D:/AIT%20programs/7.3rangeControl.html
Red:
Green:
Blue:


Color Range Controls
File | D:/AIT%20programs/7.3rangeControl.html
Red:
Green:
Blue:

8. **Apply pseudo selectors to set the background color for visited and unvisited links to ‘lightblue’, and the background color for the hover and active link states to ‘green’.**

**Code:**

```
<!DOCTYPE html>
<html>
<head>
   <title>Link Styling</title>
   <style>
      a:link {
         background-color: lightblue;
      }
      a:visited {
         background-color: lightblue;
      }
      a:hover {
         background-color: green;
      }
      a:active {
         background-color: green;
      }
   </style>
</head>
<body>
   <h1>Link Styling Example</h1>
   <p><a href="#">Unvisited link</a></p>
   <p><a href="#">Visited link</a></p>
   <p><a href="#">Click me</a></p>
</body>
</html>
```

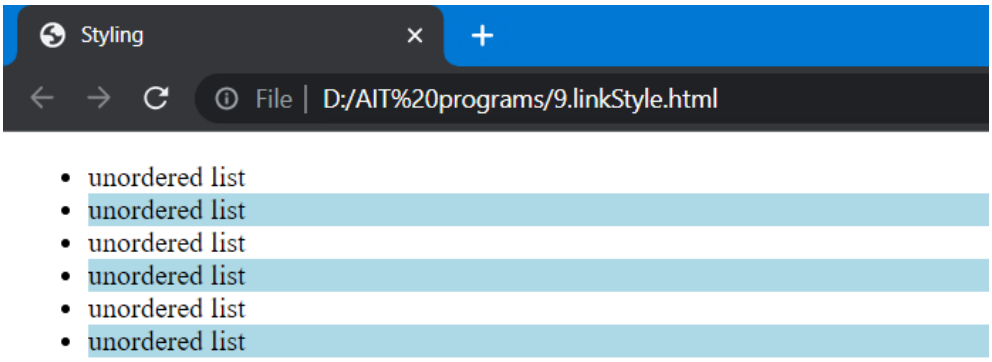
**Output:**

**9. Add a CSS rule and appropriate html to style every other line of the unordered list with a light bluebackground color. Specify the background color using the rgba method.**

**Code:**

```
<!DOCTYPE html>
<html>
<head>
   <title>Styling</title>
   <style>
      ul li:nth-child(even) {
         background-color:lightblue;
      }
   </style>
</head>
<body>
   <ul>
      <li> unordered list</li>
      <li> unordered list</li>
      <li> unordered list</li>
      <li> unordered list</li>
      <li> unordered list</li>
      <li> unordered list</li>
   </ul>
</body>
</html>
```

**Output:**

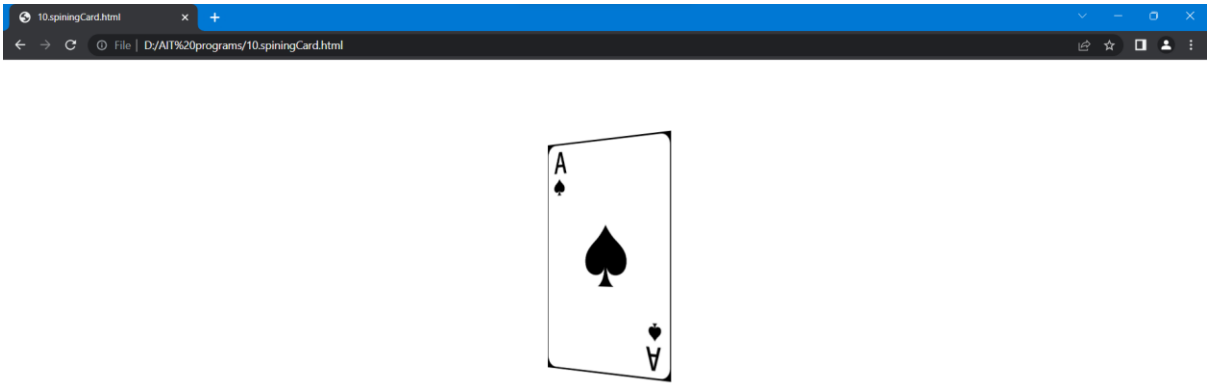


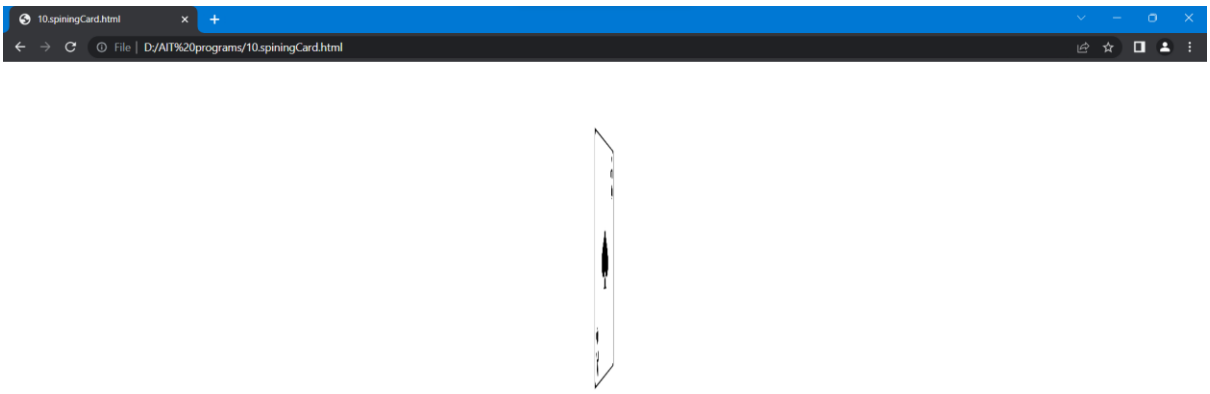
10. **Using the image of the playing card, use transformations and animations to make the playing card spin around endlessly.**

**Code:**

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .card-container {
      width: 200px; /* Adjust the dimensions as per your card image */
      height: 300px;
      position: relative;
      margin: 100px auto;
      perspective: 1000px;
    }
    .card-container img {
      width: 100%;
      height: 100%;
      position: absolute;
      transform-style: preserve-3d;
      animation: spin 5s infinite linear;
    }
    @keyframes spin {
      0% {
        transform: rotateY(0deg);
      }
      100% {
        transform: rotateY(360deg);
      }
    }
  </style>
</head>
<body>
  <div class="card-container">
    <img src="p1.jpg" alt="Playing Card">
  </div>
</body>
</html>
```

**Output:**



10.spiningCard.html
File | D:/AIT%20programs/10.spiningCard.html

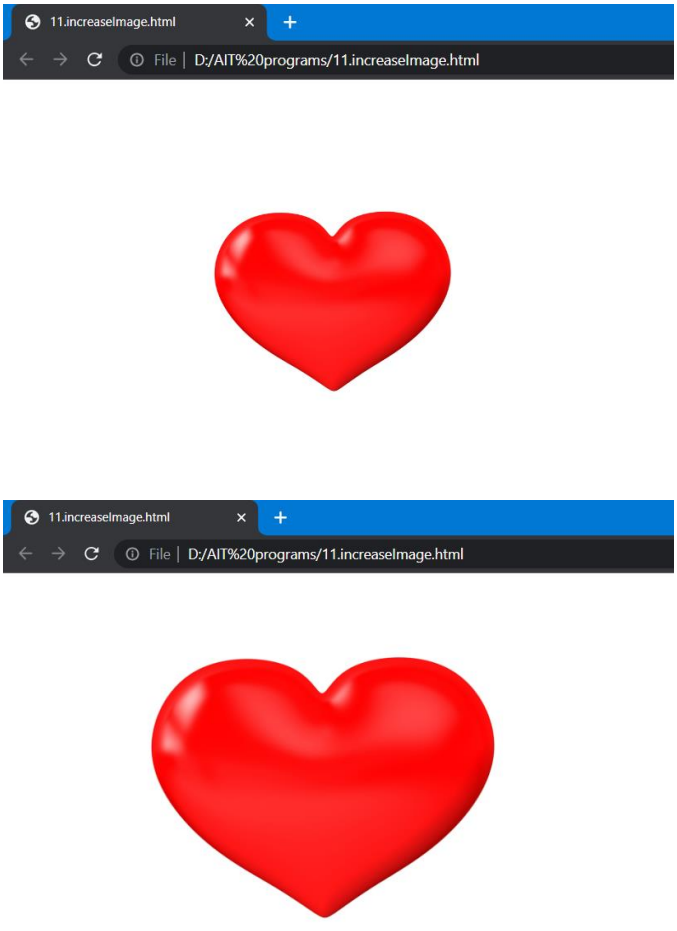

10.spiningCard.html
File | D:/AIT%20programs/10.spiningCard.html

11. **Consider the heart shape as an image and apply Animation on hover effect where image will increase the size and revert back to original size with shadow effect.**

**Code:**

```
<html lang="en">
<head>
<style>
img {
transition: transform .2s;
width: 400px;
height: 200px;
margin: 100px;
}
img:hover {
transform: scale(1.5);
}
</style>
</head>
<body>
<img src="heart.jpg" width="400px" height="200px" alt="heart">
</body>
</html>
```

**Output:**

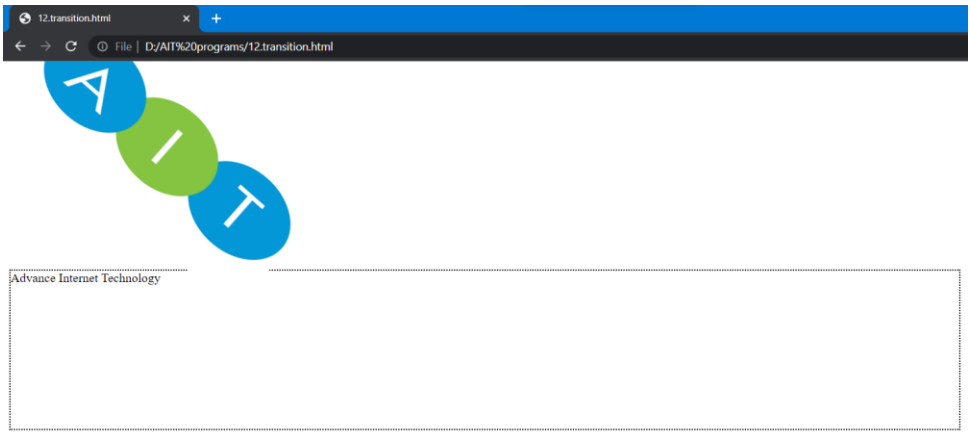


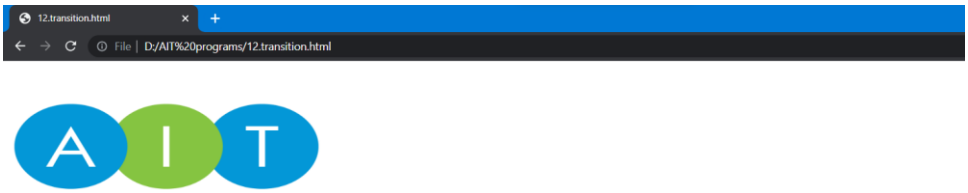
12. **Apply transition and transformation to perform following task: i. On hover of an image, it should rotate in 360 degrees. ii. On hover of a div, text should increase the size and removing cursor from div, size of the textshould be restored.**

**Code:**

```
<html>
<style>
img:hover{
transition: 4s;
transform: rotate(360deg);
}
div
{
border: 2px dotted #000 ;
height: 200px;
width: 1200px;
}
div:hover {
font-family:Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif;
font-size: 50px;
}
</style>
<body>
<img width=400 height=200 src="p2.jpg"></img><br><br><br><br>
<div>Advance Internet Technology</div>
</body>
</html>
```

**Output:**



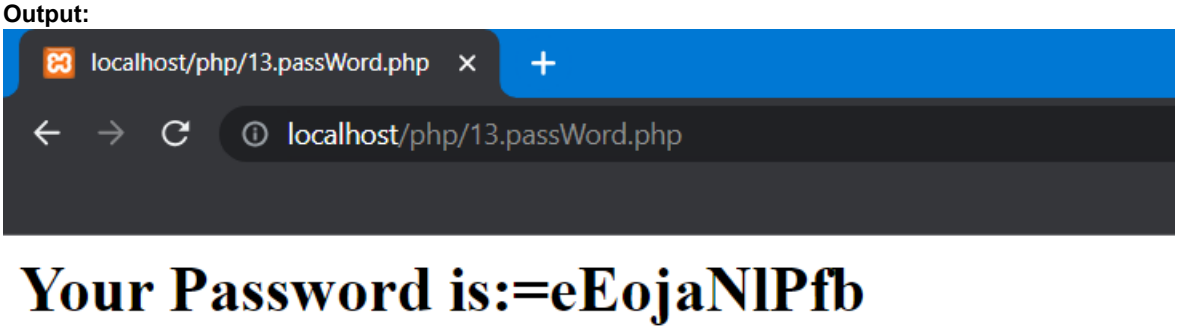




13. **Write a PHP script to generate simple random password of 10 characters from a given string. Assume astring contains characters from A-Z, a-z, 0- 9.**

**Code:**

```
<?php

$str="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789";

$str1=str_shuffle($str);

$password=substr($str1,0,10);

echo "<h1>Your Password is:=$password</h1>"

?>
```

**Output:**



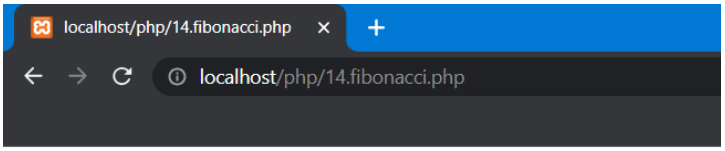

**Your Password is:=eEojaNlPfb**

14. **Write a PHP script to print Fibonacci Series using both ways: loops, recursive function**

**Code:**

```
<h1>Using loop</h1>
<?php
function fib($n){
 $num1 = 0;
 $num2 = 1;
 echo"<b>Fibonacci Series :</b>";
 $count = 0;
 while ($count< $n){
 echo ' '.$num1;
 $num3 = $num2 + $num1;
 $num1 = $num2;
 $num2 = $num3;
 $count = $count + 1;
 }
}
$n =10;
fib($n);
?>


<br><br><h1>Using recursion</h1>
<?php
echo"<b>Fibonacci Series : </b>";
function Fibonacci($num){
    if ($num == 0)
    return 0;
    else if ($num == 1)
    return 1;
    else
    return (Fibonacci($num-1) + Fibonacci($num-2));
}
$num=10;
for ($count= 0; $count < $num; $count++){
 echo Fibonacci($count),' ';
}
?>
```

**Output:**




# Using loop

**Fibonacci Series :** 0 1 1 2 3 5 8 13 21 34

# Using recursion

**Fibonacci Series :** 0 1 1 2 3 5 8 13 21 34

15. **Create a table Emp with fileds: empno, name, address, salary, deptno. Design HTML & PHP pages forfollowing: a. Insert a record b. Delete record c. View record d. Display department number wise group report.**

**Code:**

```
<html>
<head>
<style>
input[type=submit]
{background-color:#3CBC8D;
color:white;}
</style>
</head>
<?php
$servername="localhost";
$username="root";
$pass="";
$db="ait";
$con=mysqli_connect($servername,$username,$pass,$db);
if(!$con)
{die('Could no =t connect to database'.mysql_error());}
$result=mysqli_query($con,"select * from emp");
if(mysqli_num_rows($result)>0)
{?>
 <center>
<table border=2px style="padding:2px"><tr>
<th>Employe Id</th>
<th>Name</th>
<th>Address</th>
<th>Salary</th>
<th>Department No</th>
<th>Delete</th>
</tr>
<?php
while($row=mysqli_fetch_assoc($result))
{ $empid=$row['empid'];
 $name=$row['name'];
 $address=$row['address'];
 $salary=$row['salary'];
 $deptno=$row['deptno'];
 echo "
<tr><td>".$empid."</td><td>".$name."</td><td>".$address."</td><td>".$salary."</td><td>".$deptno."</t
d><td><a href=delete.php?empid=$empid>Delete</a></td></tr>";
}
}
else{
echo " No Record Found ";
}
?>
</table>
<form method="post">
<input type="submit" value="New Employee" name=new > <input type="submit" value="Display By
Department" name=display >
</form>
</center>
<?php
if(isset($_POST['new']))
{
        header('location:new.php');
}
```

```
if(isset($_POST['display']))
{
	header('location:report.php');
}
?>
</html>
<html>
<body>
<h1 align=center>Register New Employee Here</h1>
<form style="border:solid 2px black;margin:auto auto; width:50%;padding-left:20px;padding-
top:20px;padding-bottom:20px;" method="post">
<lable>Name:</lable>
<input type="text" name="name"><br><br>
<lable>Address:</lable>
<textarea cols=20 rows=4 name="address"></textarea><br><br>
<lable>Salary:</lable>
<input type="text" name="sal"><br><br>
<lable>Department:</lable>
<input type="number" name="deptno" min="1" max="5"><br><br>
<center><input type="submit"  name="submit" value="Submit"></center>
</form>
</body>
</html>
<?php
include "connection.php";
if(isset($_POST['submit']))
{
   $name=$_POST["name"];
   $address=$_POST["address"];
	$sal=$_POST["sal"];
	$deptno=$_POST["deptno"];
	$sql="INSERT INTO `emp`(`name`, `address`, `salary`, `deptno`) VALUES
('$name','$address','$sal','$deptno')";
	if(mysqli_query($con,$sql))
	{
		header('location:emp.php');
	}
	else
	{
		echo"Insertion Failed";
	}
}
?>
<?php
 include "connection.php";
 $sql="select * from emp order by deptno";
  $result=mysqli_query($con,$sql);
  if(mysqli_num_rows($result)>0)
{
?>
 <center>
<table border=2px style="padding:2px">
<tr>
<th>Employe Id</th>
<th>Name</th>
<th>Address</th>
<th>Salary</th>
<th>Department No</th>
</tr>
<?php
while($row=mysqli_fetch_assoc($result))
{
```

```
 $empid=$row['empid'];
 $name=$row['name'];
 $address=$row['address'];
 $salary=$row['salary'];
 $deptno=$row['deptno'];
 echo "
<tr><td>".$empid."</td><td>".$name."</td><td>".$address."</td><td>".$salary."</td><td>".$deptno."</t
d></tr>";
}
}
else{
echo " No Record Found ";
}
echo "<h4><a href='emp.php'>Back To Previous Page</a></h4+>";
?>
```

**Output:**

| Employe Id | Name | Address | Salary | Department No | Delete |
|---|---|---|---|---|---|
| 101 | Ram | Pune | 43000 | 1 | Delete |

New Employee | Display By Department

## Register New Employee Here

Name:

Address:

Salary:

Department:

Submit

Back To Previous Page

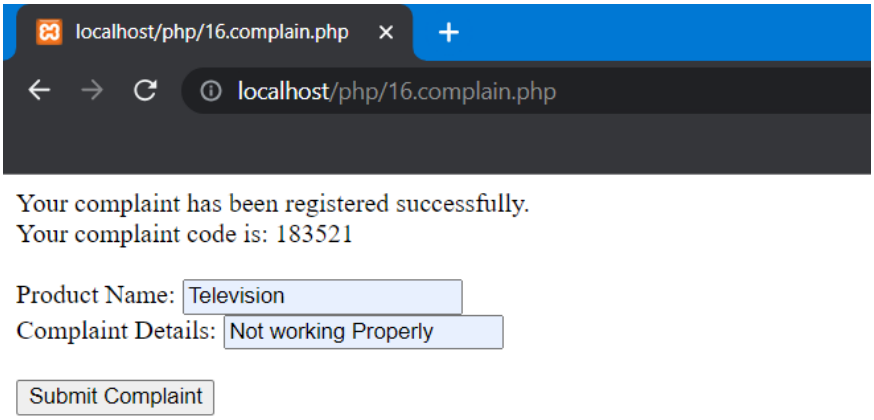
| Employe Id | Name | Address | Salary | Department No |
|---|---|---|---|---|
| 101 | Ram | Pune | 43000 | 1 |

16. **Write PHP code to register complaint of an electronic product. After receiving complaint,give/display complaint code to user.**

**Code:**

```
<?php
function generateComplaintCode(){
        return mt_rand(100000, 999999);
}
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $productName = $_POST['product_name'];
    $complaintDetails = $_POST['complaint_details'];
    $complaintCode = generateComplaintCode();
    echo "Your complaint has been registered successfully.<br> Your complaint code is:
$complaintCode";
}
?>
<form method="POST" action=""><br>

    <label for="product_name">Product Name:</label>
    <input type="text" name="product_name" id="product_name" required><br>
    <label for="complaint_details">Complaint Details:</label>
    <input type="text" name="complaint_details" id="complaint_details" required><br><br>

    <input type="submit" value="Submit Complaint">
</form>
```

**Output:**



17. **Create book shopping cart application in PHP. Display all books details on page using Book table.**

**Code:**

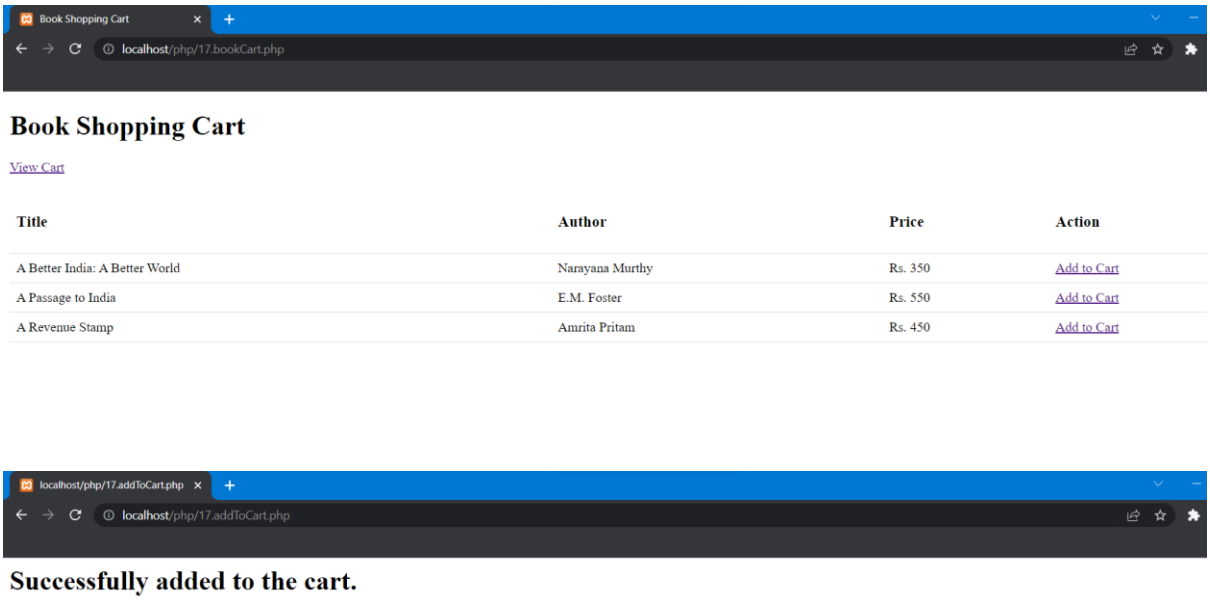
**bookCart.php file:**

```
<!DOCTYPE html>
<html>
<head>
    <title>Book Shopping Cart</title>
    <style>
        table {
            border-collapse: collapse;
            width: 100%;
        }
        th, td {
            padding: 8px;
            text-align: left;
            border-bottom: 1px solid #ddd;
        }
    </style>
</head>
<body>
    <h1>Book Shopping Cart</h1>
    <a href="17.bookCart.php">View Cart</a>
    <br><br>
    <table>
        <tr>
            <th><h3>Title</h3></th>
            <th><h3>Author</h3></th>
            <th><h3>Price</h3></th>
            <th><h3>Action</h3></th>
        </tr>

        <?php
        $conn = mysqli_connect('localhost', 'root', '', 'book');
        if (!$conn) {
            die("Connection failed: " . mysqli_connect_error());
        }

        $sql = "SELECT * FROM books";
        $result = mysqli_query($conn, $sql);

        if (mysqli_num_rows($result) > 0) {
            while ($row = mysqli_fetch_assoc($result)) {
                echo "<tr>";
                echo "<td>" . $row['title'] . "</td>";
                echo "<td>" . $row['author'] . "</td>";
                echo "<td>Rs. " . $row['price'] . "</td>";
                echo "<td><a href='17.addToCart.php'>Add to Cart</a></td>";
                echo "</tr>";
            }
        } else {
            echo "<tr><td colspan='4'>No books available</td></tr>";
        }
        mysqli_close($conn);
        ?>
    </table>
</body>
</html>
```

**addToCard.php file:**

```
<h1>Successfully added to the cart.</h1>
```

**Output:**



**18. Program to show current date and time using user defined module.**

**Code:**

**date.js file:**

```
function getCurrentDateTime(){
   const now = new Date();
   const currentDateTime = now.toISOString();
   return currentDateTime;
}
module.exports = { getCurrentDateTime };
```

**main.js:**

```
const datetimeModule = require("./date");
const currentDateTime = datetimeModule.getCurrentDateTime();
console.log("Current date and time:", currentDateTime);
```

**Output:**

```
C:\Program Files\nodejs\node.exe .\main.js
Current date and time: 2023-06-28T06:42:19.266Z
```

**19. Program using built-in modules to split the query string into readableparts.**

**Code:**

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

const string = 'name=Harshal&age=22&location=Pune';
const query = querystring.parse(string);

console.log('Name:', query.name);
console.log('Age:', query.age);
console.log('Location:', query.location);
```

**Output:**

```
C:\Program Files\nodejs\node.exe .\queryString.js
Name: Harshal
Age: 22
Location: Pune
```

**20. Program using NPM which will convert entered string into either case.**

**Code:**

```
const readline=require('readline');
let r1=readline.createInterface(
process.stdin,process.stdout
);
r1.question("Enter a string : ",(s)=>{
  console.log("String in lower case : ",s.toLowerCase());
  console.log("String in upper case : ",s.toUpperCase());
  console.log("String in title case : ",s.charAt(0).toUpperCase() +
  s.substr(1).toLowerCase());
  r1.close();
});
```

**Output:**

```
PS D:\AIT programs> node npm.js
Enter a string : harshal wandhare
String in lower case :  harshal wandhare
String in upper case :  HARSHAL WANDHARE
String in title case :  Harshal wandhare
PS D:\AIT programs>
```

**21. Write a program to create a calculator using Node JS.**

**Code:**

```
const readline = require('readline');
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});
function calculator() {
  rl.question('Enter the first number: ', (num1) => {
    rl.question('Enter the operator (+, -, *, /): ', (operator) => {
      rl.question('Enter the second number: ', (num2) => {
        num1 = parseFloat(num1);
        num2 = parseFloat(num2);
        let result;
        switch (operator) {
          case '+':
            result = num1 + num2;
            break;
          case '-':
            result = num1 - num2;
            break;
          case '*':
            result = num1 * num2;
            break;
          case '/':
            result = num2 !== 0 ? num1 / num2 : 'Error: Cannot divide by zero';
            break;
          default:
            result = 'Error: Invalid operator';
        }
        console.log('Result:', result);
        rl.close();
      });
    });
  });
}
calculator();
```

**Output:**

```
PS D:\AIT programs> node calculator.js
Enter the first number: 8
Enter the operator (+, -, *, /): +
Enter the second number: 7
Result: 15
PS D:\AIT programs> node calculator.js
Enter the first number: 8
Enter the operator (+, -, *, /): -
Enter the second number: 5
Result: 3
PS D:\AIT programs> node calculator.js
Enter the first number: 7
Enter the operator (+, -, *, /): *
Enter the second number: 7
Result: 49
PS D:\AIT programs> node calculator.js
Enter the first number: 10
Enter the operator (+, -, *, /): /
Enter the second number: 2
Result: 5
PS D:\AIT programs>
```

**22. Create angular project which will demonstrate the usage of componentdirective, structural directive and attribute directives.**

**Code:**

**For ngIf:**

**app.component.html:**

```
<h1>This is my Angular program</h1>
<h1 *ngIf="show"=='yes' ; then ifBlock else elseBlock></h1>
<ng-template #ifBlock><h1>If Block</h1></ng-template>
<ng-template #elseBlock><h1>Else Block</h1></ng-template>
<div [ngSwitch]="color">
  <div *ngSwitchCase="'red'"><h1>You picked Red Colour</h1></div>
  <div *ngSwitchCase="'blue'"><h1>You picked Blue Colour</h1></div>
  <div *ngSwitchCase="'green'"><h1>You picked Green Colour</h1></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 {
title = 'directives';
color='red';
show='yes';
}
```

**Output:**


# This is my Angular program

# You picked Red Colour

**For ngFor:**

**Code:**

**app.component.html:**

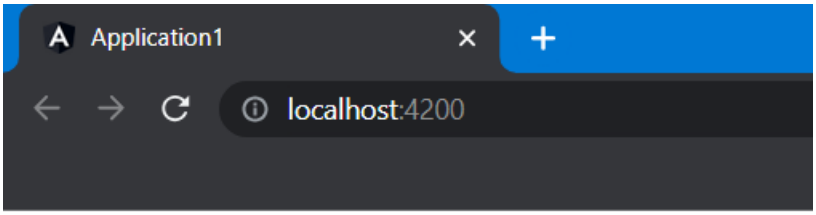
```
<table border="2">
  <tr>
  <td>
```

```
Name
</td>
<td>
Rollno
</td>
</tr>
<tr *ngFor="let item of data">
<td>{{item.name}}</td>
<td>{{item.rollno}}</td>
</tr>
</table>
```

**app.component.ts:**

```
import { Component } from '@angular/core';
@Component({
selector: 'app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent {
title = 'directives1';
data=[
{
name:'John',
rollno:114
},
{
name:'Harry',
rollno:112
},
{
name:'michel',
rollno:117
}
]
}
```

**Output:**




| Name | Rollno |
|---|---|
| John | 114 |
| Harry | 112 |
| michel | 117 |

23. **Create angular project which has HTML template and handle the clickevent on click of the button.**

**Code:**

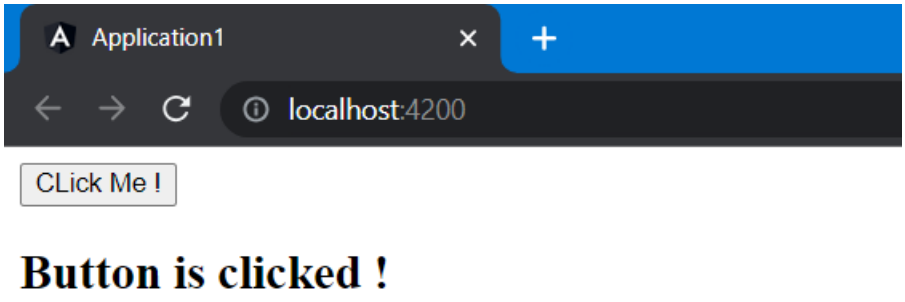
**App.component.html:**

```
<button (click)="clickEvent()">
  CLick Me !
  </button>
  <h2>
  {{mesg}}
  </h2>
```

**App.component.ts:**

```
import { Component } from '@angular/core';
@Component({
selector: 'app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent {
title = 'click';
message="Button is clicked ! ";
mesg:string|undefined;
clickEvent(){
this.mesg="Button is clicked !";
return this.mesg;
}
}
```

**Output:**




24. **Write Program for Form validation in Angular.**

**Code:**

**App.component.html:**

```
<form (ngSubmit)="onSubmit()" #myForm="ngForm">
  <label for="name">Full Name : </label>
  <input required id="name" type="text" name="name" ngModel #name="ngModel">
  <br>
  <div><small *ngIf="name.invalid && name.touched">Name is
  required</small></div>
  <label for="email">Email : </label>
  <input required name="email" type="text" email #email="ngModel" ngModel>
  <br>
  <p>
  <small *ngIf="email.invalid && email touched">Please enter valid
  email</small>
  </p>
  <label for="password">Password : </label>
  <input required name="password" type="password" minlength="4" maxlength="10"
  #password="ngModel" ngModel>
  <p><small *ngIf="password.invalid && password.touched">Please enter valid
  password</small></p>
  <input type="submit" id="btn" value="submit" [disabled]="!myForm.valid">
  </form>
```

**App.component.ts:**

```
import { Component } from '@angular/core';
@Component({
selector: 'app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent {
title = 'valform';
onSubmit()
{
console.log("Form submitted....");
}
}
```

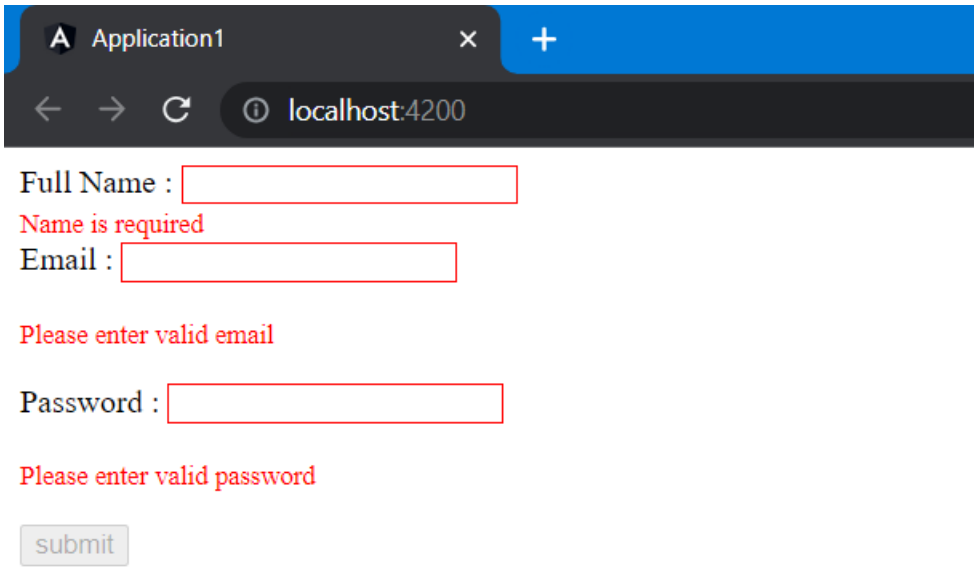
**App.module.ts:**

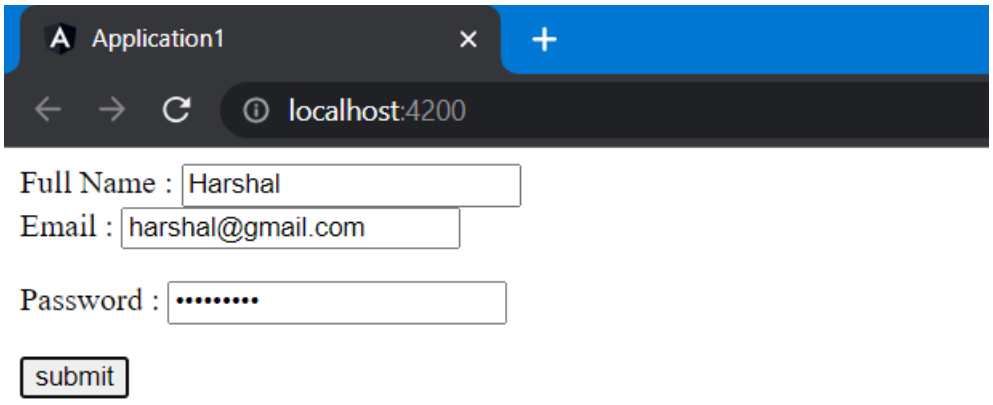
```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import {FormsModule } from '@angular/forms';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
@NgModule({
declarations: [
AppComponent
],
imports: [
BrowserModule,
AppRoutingModule,
FormsModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }
```

**App.component.css:**

```
input.ng-invalid.ng-touched{
   border:red 1px solid;
   }
   small{
   color:red;
   }
```

**Output:**



25. **Program to demonstrate the Services and Module in Angular.**

**Code:**

**App.service.ts:**

```
import { Injectable } from '@angular/core';
@Injectable({
providedIn: 'root'
})
export class DataService {
private data: string = ''; // Assigning an initial value
constructor() { }
setData(data: string) {
this.data = data;
}
getData() {
return this.data;
}
}
```

**App.component.ts:**

```
import { Component, OnInit } from '@angular/core';
import { DataService } from '../data.service';
@Component({
selector: 'app-my-component',
template: `
<h1>{{ data }}</h1>
<button (click)="setData()">Set Data</button>
`,
styleUrls: ['./my-component.component.css']
})
export class MyComponentComponent implements OnInit {
data!: string; // Definite assignment assertion
constructor(private dataService: DataService) { }
ngOnInit() {
this.data = this.dataService.getData();
}
setData() {
this.dataService.setData('Hello, Angular!');
this.data = this.dataService.getData();
}
}
```

**App.module.ts:**

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppComponent } from './app.component';
import { MyComponentComponent } from './my-component/my-component.component';
import { DataService } from './data.service';
@NgModule({
declarations: [
AppComponent,
MyComponentComponent
IICMR,Nigdi,Pune
31|Advanced Internet Technologies],
imports: [
```

```
BrowserModule
],
providers: [DataService],
bootstrap: [AppComponent]
})
export class AppModule { }
```

app.component.html

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

**Output:**