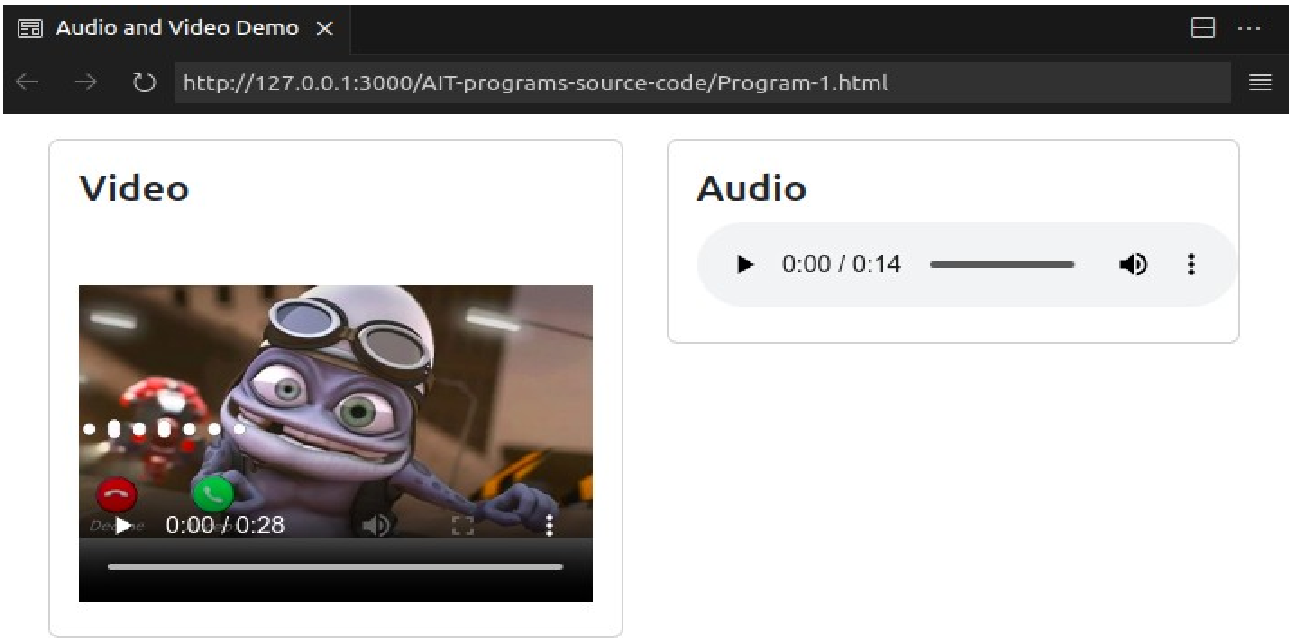
1. Program to implement Audio and Video features for your web page

Ans :

```
<!DOCTYPE html>
<html><head>
   <title>Audio and Video Demo</title>
   <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"/>
  </head><body>
   <div class="row m-3"><div class="col-sm-6">
      <div class="card">
        <div class="card-body">
          <h4>Video</h4>
          <video width="100%" height="240" controls autoplay>
            <source src="./files//crazy frog Ringtone.mp4" type="video/mp4" />
            Your browser does not support the video tag.
          </video>
        </div>
      </div></div>
     <div class="col-sm-6"><div class="card">
        <div class="card-body">
          <h4>Audio</h4>
          <audio controls>
            <source src="./files/my_crazy_frog_ringtone.mp3" type="audio/mpeg"/>
            Your browser does not support the audio element
          </audio>
        </div>
      </div>
     </div></div>
  </body></html>
```

Output :

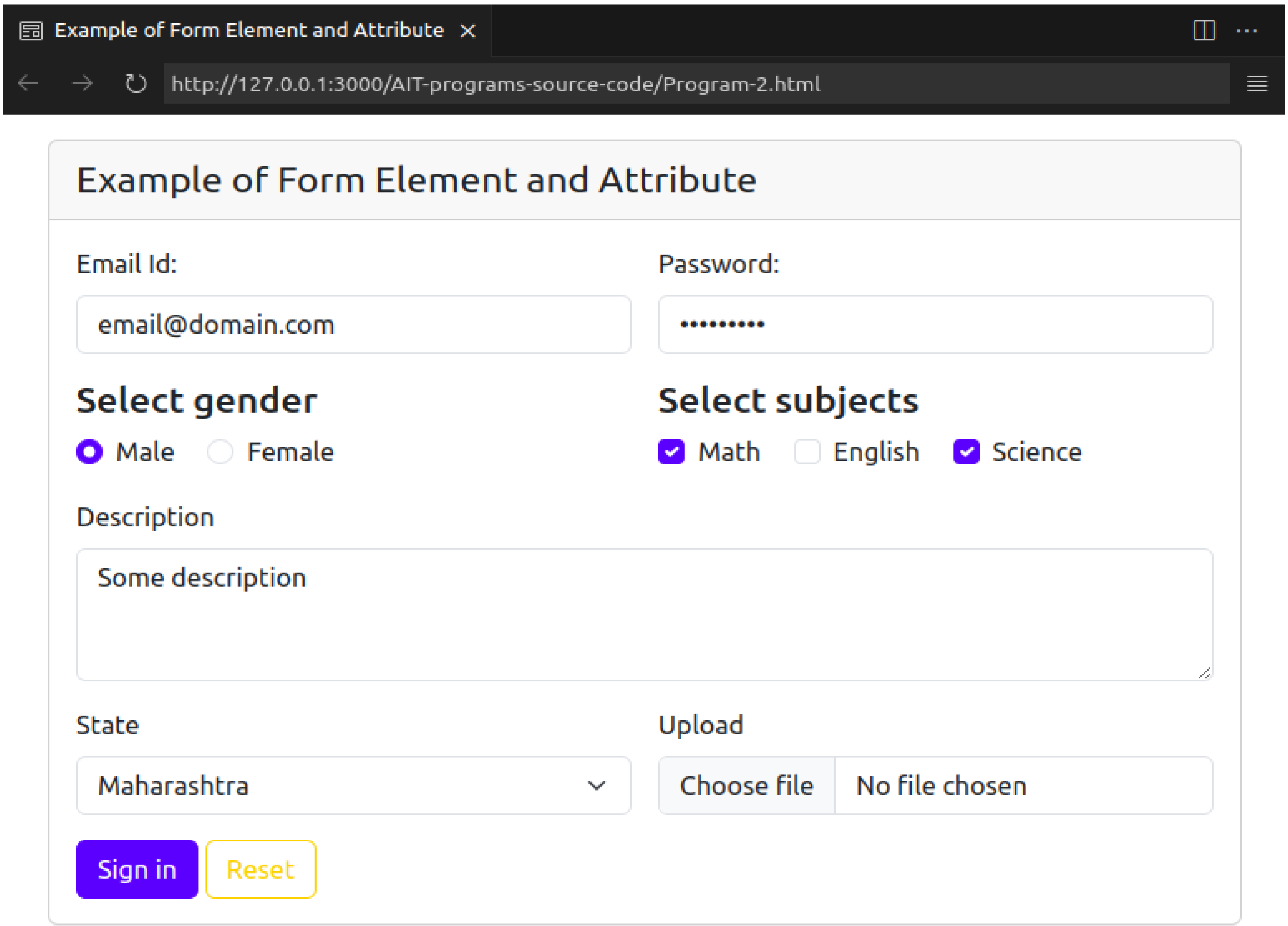

Audio and Video Demo
http://127.0.0.1:3000/AIT-programs-source-code/Program-1.html
Video
0:00 / 0:28
Audio
0:00 / 0:14

2. Program to design form using HTML5 elements, attributes and Semantics.

Ans :

```
<!DOCTYPE html><html>
  <title>Example of Form Element and Attribute</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"/>
  <body>
    <div class="container m-3">
      <div class="card" style="width: 45rem">
        <div class="card-header fs-4">Example of Form Element and Attribute</div>
        <div class="card-body">
          <form class="row g-3">
            <div class="col-md-6">
              <label for="Email Id:" class="form-label">Email Id:</label>
              <input type="email" placeholder="Enter your email id." class="form-control" id="Email
Id:"/>
            </div>
            <div class="col-md-6">
              <label for="password" class="form-label">Password:</label>
              <input type="password" class="form-control" id="password" placeholder="Enter
password"/>
            </div>
            <div class="col-6">
              <h4>Select gender</h4>
              <div class="form-check form-check-inline">
                <input class="form-check-input" type="radio" name="gender" id="inlineRadio1"
value="Male"/>
                <label class="form-check-label" for="inlineRadio1">Male</label>
              </div>
              <div class="form-check form-check-inline">
                <input class="form-check-input" type="radio" name="gender" id="inlineRadio2"
value="Female"/>
                <label class="form-check-label" for="inlineRadio2">Female</label>
              </div>
            </div>
            <div class="col-6">
              <h4>Select subjects</h4>
              <div class="form-check form-check-inline">
                <input class="form-check-input" type="checkbox" id="inlineCheckbox1"
name="subject[]" value="Math"/>
                <label class="form-check-label" for="inlineCheckbox1">Math</label>
              </div>
              <div class="form-check form-check-inline">
                <input class="form-check-input" type="checkbox" id="inlineCheckbox2"
name="subject[]" value="English"/>
                <label class="form-check-label" for="inlineCheckbox2" >English</label>
              </div>
              <div class="form-check form-check-inline">
                <input class="form-check-input" type="checkbox" id="inlineCheckbox3"
name="subject[]" value="Science"/>
                <label class="form-check-label" for="inlineCheckbox3" >Science</label>
      </div></div>
     <div class="col-12">
      <label for="description" class="form-label">Description</label>
      <textarea class="form-control" id="description" rows="3"></textarea>
     </div>
     <div class="col-6">
       <label for="inputState" class="form-label">State</label>
       <select id="inputState" class="form-select">
         <option selected>Select...</option>
         <option value="Maharashtra">Maharashtra</option>
         <option value="Gujrat">Gujrat</option>
       </select>
      </div>
     <div class="col-6">
      <label for="formFile" class="form-label">Upload</label>
      <input class="form-control" type="file" id="formFile" />
     </div>
     <div class="col-12">
      <button type="submit" class="btn btn-primary">Sign in</button>
      <button type="reset" class="btn btn-outline-warning">Reset</button>
     </div>
    </form>
   </div></div></div>
</body></html>
```

Output :



Example of Form Element and Attribute ×

http://127.0.0.1:3000/AIT-programs-source-code/Program-2.html

Example of Form Element and Attribute

Email Id: email@domain.com

Password: ••••••••••

Select gender

● Male ○ Female

Select subjects

☑ Math □ English ☑ Science

Description

Some description

State: Maharashtra

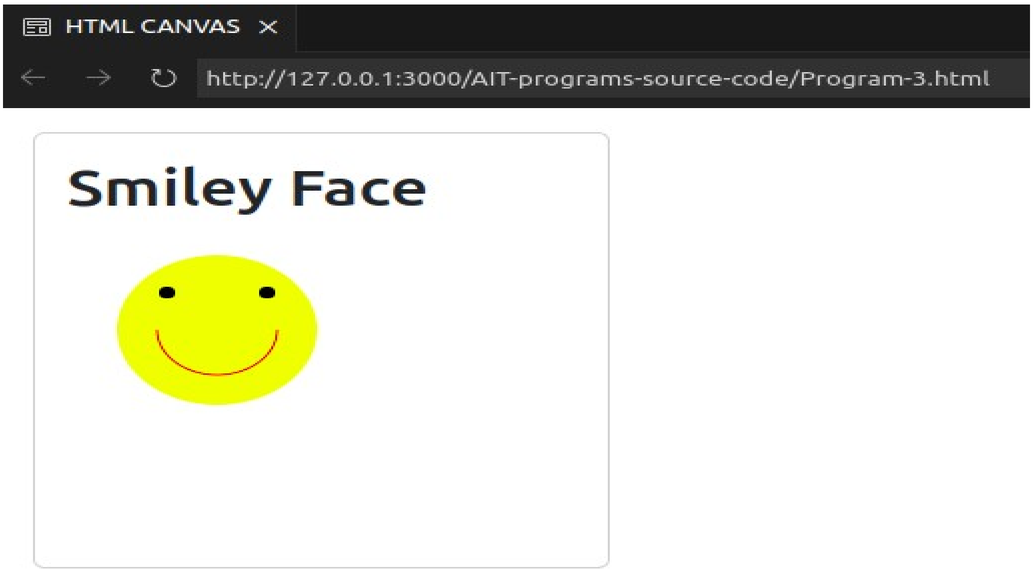
Upload: Choose file No file chosen

3. Program using Canvas to create Smiley face.

Ans :

```
<!DOCTYPE html>    <html>    <head>
     <title>    HTML CANVAS</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"/>
   </head><body onload="bodyLoad()">
    <div class="card m-3" style="width: 18rem;"><div class="card-body">
           <h1>Smiley Face</h>   <br>
           <canvas id="mycanvas" width="400" height="200"></canvas>
        </div></div>
        <script>
function bodyLoad() {      myFace();    myEye();    mySmile();    }
function myFace() {
   var c = document.getElementById("mycanvas");
   var draw = c.getContext("2d"); draw.beginPath();     draw.fillStyle = "yellow";
   draw.arc(75, 75, 50, 0, Math.PI * 2, true);    draw.closePath();    draw.fill();
}
function myEye() {
 var c = document.getElementById("mycanvas");   var eye = c.getContext("2d");
 eye.moveTo(55, 50);  eye.beginPath();  eye.fillStyle = "black";
 eye.arc(50, 50, 4, 0, Math.PI * 2, true);
 eye.closePath();  eye.fill();  eye.moveTo(103, 49);  eye.beginPath();
 eye.fillStyle = "black";  eye.arc(100, 50, 4, 0, Math.PI * 2, true);
 eye.closePath();  eye.fill();
}
function mySmile() {
   var c = document.getElementById("mycanvas");
   var smile = c.getContext("2d");    smile.moveTo(105, 75);
   smile.beginPath();    smile.strokeStyle = "red";
   smile.arc(75, 75, 30, 0, Math.PI, false);    smile.stroke();
}
   </script></body> </html>
```

Output :

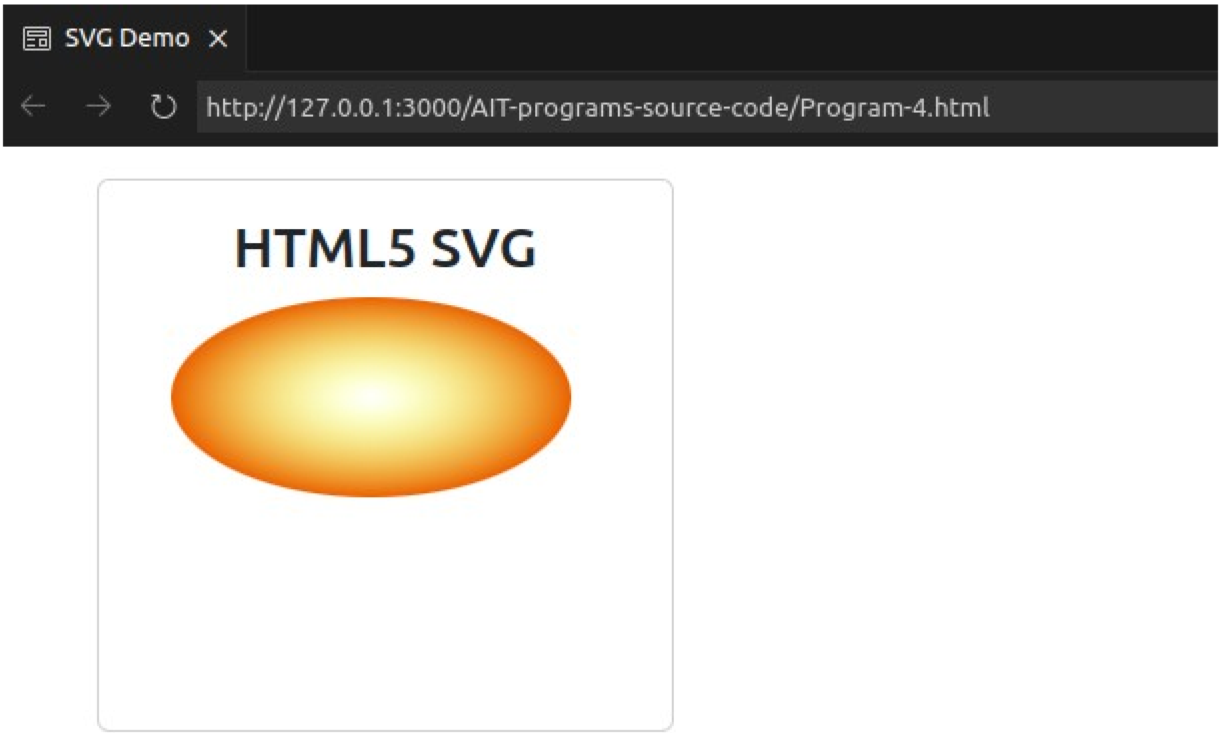

HTML CANVAS
http://127.0.0.1:3000/AIT-programs-source-code/Program-3.html
Smiley Face

4. Program to demonstrate Radial Ingredient using SVG.

Ans :

```
<!DOCTYPE html>
<html>
<head>
<title>SVG Demo</title>
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"/>
</head>
<body>
   <div class="card m-3 ms-5" style="width: 18rem;">
      <div class="card-body">
   <h2 align = "center">HTML5 SVG</h2>
   <svg height ="200" xmlns = "http://www.w3.org/2000/svg">
   <defs>
   <radialGradient id="rg" cx="50%" cy="50%" r="50%" fx="50%" fy="50%">
   <stop offset="0%" style="stop-color:rgb(255,255,0); stop-opacity:0" />
   <stop offset="100%" style="stop-color:rgb(230, 100, 25);stop-opacity:1" /> </radialGradient>
   </defs>
   <ellipse cx = "120" cy = "50" rx = "100" ry="50" fill="url(#rg)"/>
   </svg>
</div>
</div>
</body>
</html>
```

Output :


SVG Demo
http://127.0.0.1:3000/AIT-programs-source-code/Program-4.html

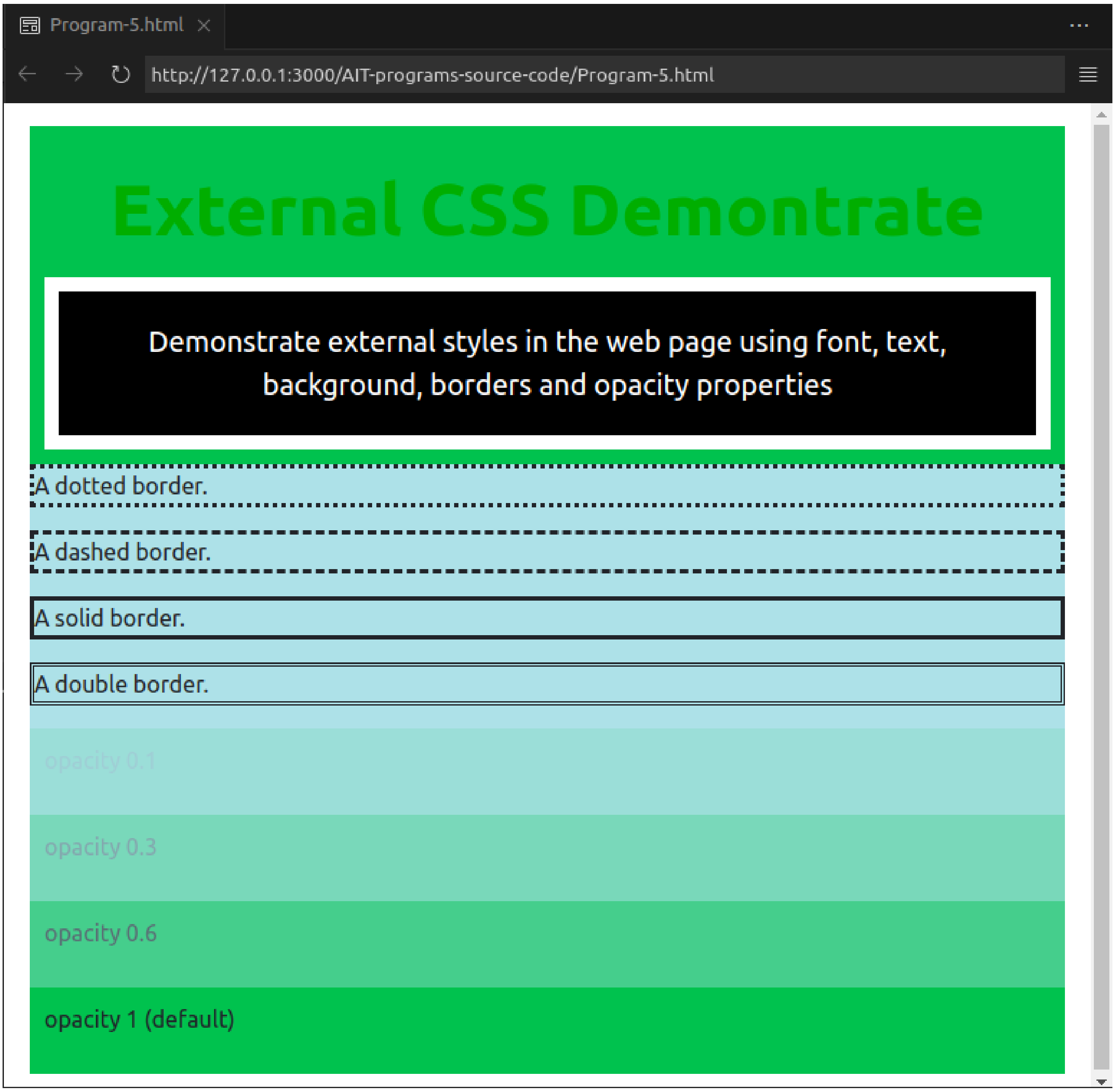
HTML5 SVG

5. Programs to demonstrate external styles in the web page using font, text, background, borders and opacity properties.

Ans :

```
<!DOCTYPE html><html><head>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"/>
  <style>
    .class1 {
      color: #009900;
      font-size: 50px;
      font-weight: bold;
    }
    #s1 {
      font-style: bold; font-size: 20px;
      display: table-cell;
      margin: 10;
      background: black;
      color: white;
      padding: 20px;
      border: 10px solid white;
      vertical-align: middle;
    }
    p.dotted {border-style: dotted;}
    p.dashed {border-style: dashed;}
    p.solid {border-style: solid;}
    p.double {border-style: double;}
    div {background-color: #4caf50;padding: 10px;}
    div.first { opacity: 0.1;}
    div.second { opacity: 0.3;}
    div.third { opacity: 0.6;}
  </style>
 </head>
 <body>
  <div class="container p-0 my-3" style="background: powderblue;">
    <div class="main text-center">
      <div class="class1">External CSS Demontrate</div>
      <div id="s1">
        Demonstrate external styles in the web page using font, text,
        background, borders and opacity properties
      </div>
    </div>
    <p class="dotted">A dotted border.</p>
    <p class="dashed">A dashed border.</p>
    <p class="solid">A solid border.</p>
    <p class="double">A double border.</p>
    <div class="first"><p>opacity 0.1</p></div>
    <div class="second"><p>opacity 0.3</p></div>
    <div class="third"><p>opacity 0.6</p></div>
    <div><p>opacity 1 (default)</p></div>
  </div>
 </body></html>
```

Output :



Program-5.html
http://127.0.0.1:3000/AIT-programs-source-code/Program-5.html

External CSS Demontrate

Demonstrate external styles in the web page using font, text, background, borders and opacity properties

A dotted border.

A dashed border.

A solid border.

A double border.

opacity 0.1

opacity 0.3

opacity 0.6

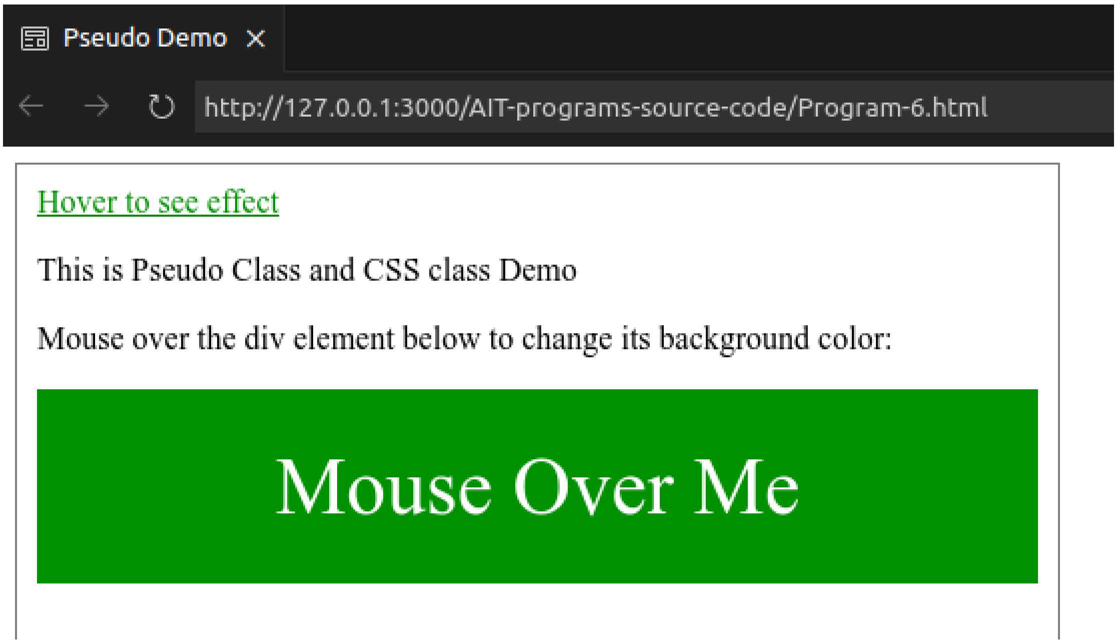
opacity 1 (default)

6. Programs to demonstrate any 3 CSS3 Pseudo-classes properties using internal styles in the web page.

Ans :

```
<html><head>
   <title>Pseudo Demo</title>
   <style>
     a:link { color: green;}
     a:visited  { color: red;}
     a.high:hover { color: black; cursor: progress;}
     a:active { color: yellow;}
     p.high:hover { color: white; font-family: Arial; background-color: red;}
     div { background-color: green; color: white; padding: 25px; text-align: center; font-size: 40px;}
     div:hover {background-color: blue;}
   </style>
  </head>
  <body style="width: 500px;border:1px solid gray; padding:10px;">
   <a href="hello.html" class="high">Hover to see effect</a>
   <p class="high">This is Pseudo Class and CSS class Demo</p>
   <p>Mouse over the div element below to change its background color:</p>
   <div>Mouse Over Me</div>
  </body></html>
```

Output :


Pseudo Demo
http://127.0.0.1:3000/AIT-programs-source-code/Program-6.html

Hover to see effect

This is Pseudo Class and CSS class Demo

Mouse over the div element below to change its background color:

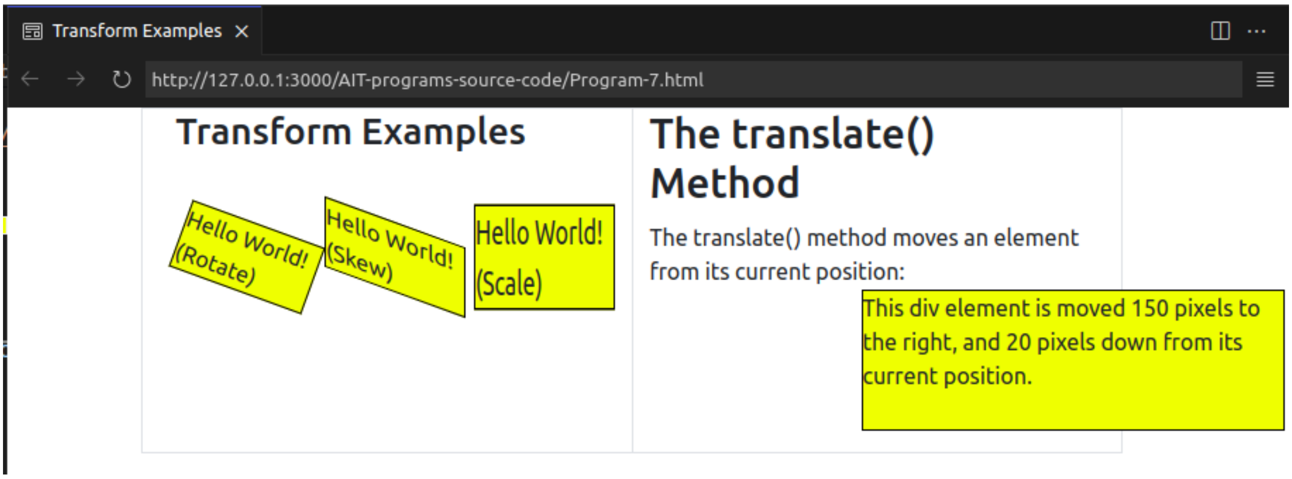
Mouse Over Me

7. Program to implement Transformation using Translation, Rotation and Scaling in your web page.

Ans :

```
<!DOCTYPE html><html lang="en">  <head>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"/>
    <title>Transform Examples</title><style>
      .box {width: 100px;height: 50px;background-color: yellow;border: 1px solid black;margin-bottom: 20px;}
      .rotate {transform: rotate(20deg);}
      .skew {transform: skewY(20deg);}
      .scale {transform: scaleY(1.5);}
      .translate {width: 300px;height: 100px;background-color: yellow;border: 1px solid
black;transform: translate(150px, -15px);}
    </style>
  </head><body class="container">
    <div class="row container border">
      <div class="col-6">
        <h3 class="mb-5">Transform Examples</h3>
        <div class="d-flex justify-content-between">
          <div class="box rotate">Hello World! (Rotate)</div>
          <div class="box skew">Hello World! (Skew)</div>
          <div class="box scale">Hello World! (Scale)</div>
        </div></div>
      <div class=" border-start col-6">
          <h2>The translate() Method</h2>
          <p>The translate() method moves an element from its current position:</p>
          <div class="translate">
            This div element is moved 150 pixels to the right, and 20 pixels down
            from its current position.
          </div>
      </div>
    </div>
  </body></html>
```

Output :


Transform Examples ×
http://127.0.0.1:3000/AIT-programs-source-code/Program-7.html

Transform Examples

Hello World! (Rotate)

Hello World! (Skew)

Hello World! (Scale)

The translate() Method

The translate() method moves an element from its current position:

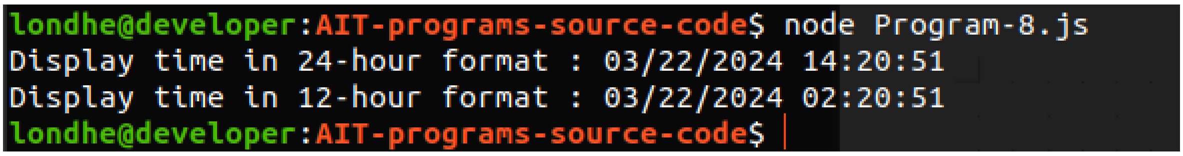
This div element is moved 150 pixels to the right, and 20 pixels down from its current position.

8. Program to show current date and time using user defined module in Node.JS

Ans :

```
// Function to convert single digit input to two digits
const formatDigit = (input) => (input > 9 ? input : `0${input}`);

// Function to convert 24-hour time to 12-hour time
const convertTo12HourFormat = (hour) => {
   if (hour > 12) {
      return hour - 12;
   }
   return hour;
};

// Get current date and time
const currentDate = new Date();

// Format date and time components
const formattedDate = {
   dd: formatDigit(currentDate.getDate()),
   mm: formatDigit(currentDate.getMonth() + 1),
   yyyy: currentDate.getFullYear(),
   HH: formatDigit(currentDate.getHours()),
   hh: formatDigit(convertTo12HourFormat(currentDate.getHours())),
   MM: formatDigit(currentDate.getMinutes()),
   SS: formatDigit(currentDate.getSeconds()),
};

// Function to display time in 24-hour format
const displayTimeIn24HourFormat = ({ dd, mm, yyyy, HH, MM, SS }) => {
   return `${mm}/${dd}/${yyyy} ${HH}:${MM}:${SS}`;
};

// Function to display time in 12-hour format
const displayTimeIn12HourFormat = ({ dd, mm, yyyy, hh, MM, SS }) => {
   return `${mm}/${dd}/${yyyy} ${hh}:${MM}:${SS}`;
};

// Display time in 24-hour format
console.log("Display time in 24-hour format : " + displayTimeIn24HourFormat(formattedDate));
// Display time in 12-hour format
console.log("Display time in 12-hour format : " + displayTimeIn12HourFormat(formattedDate));
```

Output :



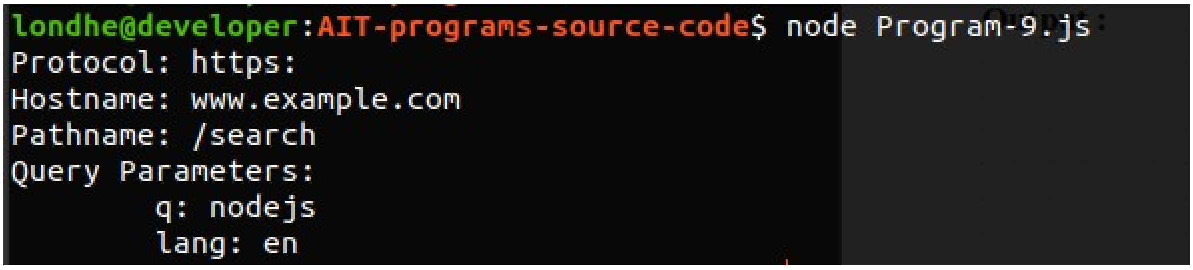
```
londhe@developer:AIT-programs-source-code$ node Program-8.js
Display time in 24-hour format : 03/22/2024 14:20:51
Display time in 12-hour format : 03/22/2024 02:20:51
londhe@developer:AIT-programs-source-code$
```

9. Program using built-in modules to split the query string into readable parts in Node.JS

Ans :

```
// Importing the built-in 'url' module
const url = require('url');

// Sample query string
const queryString = 'https://www.example.com/search?q=nodejs&lang=en';

// Parsing the query string using the 'url' module
const parsedUrl = url.parse(queryString, true);

// Extracting the readable parts from the parsed query string
const protocol = parsedUrl.protocol;
const hostname = parsedUrl.hostname;
const pathname = parsedUrl.pathname;
const queryParameters = parsedUrl.query;

// Displaying the readable parts
console.log('Protocol:', protocol);
console.log('Hostname:', hostname);
console.log('Pathname:', pathname);
console.log('Query Parameters:');
for (const [key, value] of Object.entries(queryParameters)) {
   console.log(`\t${key}: ${value}`);
}
```

Output :



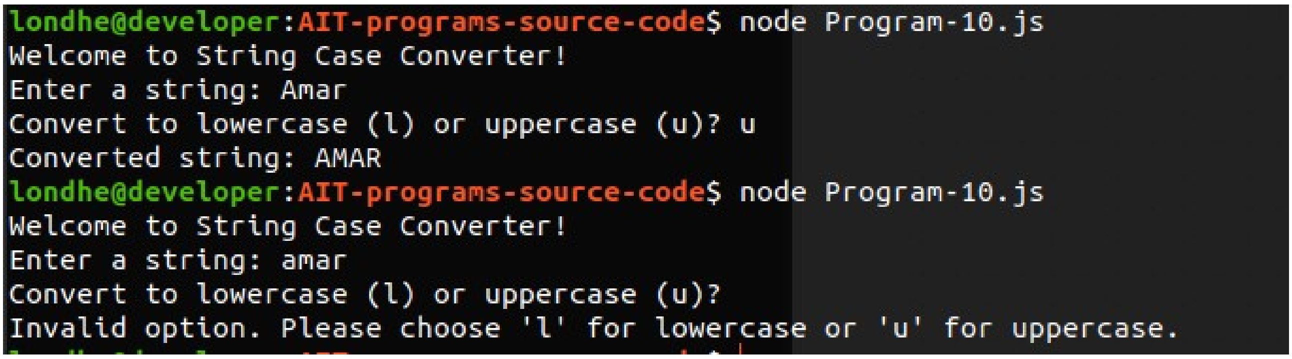
```
londhe@developer:AIT-programs-source-code$ node Program-9.js
Protocol: https:
Hostname: www.example.com
Pathname: /search
Query Parameters:
        q: nodejs
        lang: en
```

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

Ans :

```
const readline = require('readline-sync');
// Function to convert string to lowercase
function convertToLowercase(str) {
   return str.toLowerCase();
}
// Function to convert string to uppercase
function convertToUppercase(str) {
   return str.toUpperCase();
}
// Main function
function main() {
   console.log("Welcome to String Case Converter!");
   const inputString = readline.question("Enter a string: ");
   const option = readline.question("Convert to lowercase (l) or uppercase (u)? ");
   let result;
   switch (option.toLowerCase()) {
      case 'l': result = convertToLowercase(inputString);break;
      case 'u': result = convertToUppercase(inputString);break;
      default:
         console.log("Invalid option. Please choose 'l' for lowercase or 'u' for uppercase.");
         return;
   }
   // Display the converted string
   console.log(`Converted string: ${result}`);
}
// Call the main function to start the program
main();
```

Output :

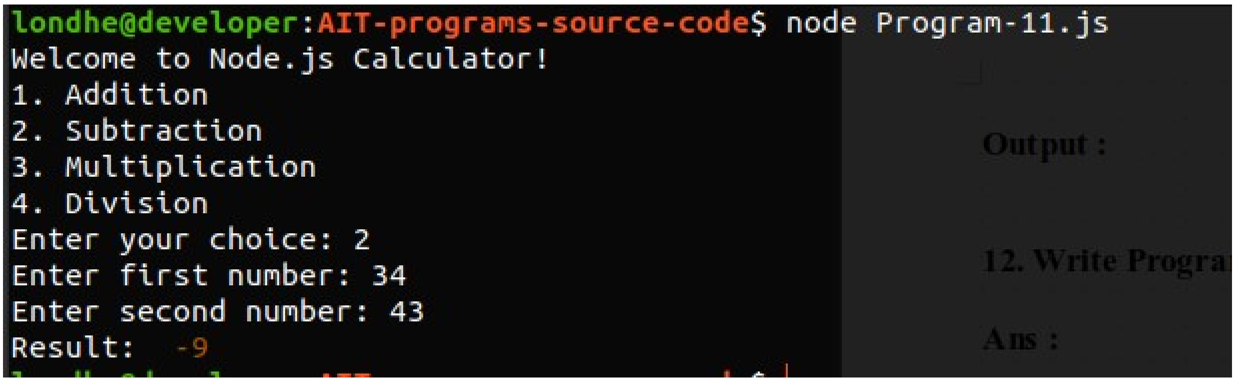


```
londhe@developer:AIT-programs-source-code$ node Program-10.js
Welcome to String Case Converter!
Enter a string: Amar
Convert to lowercase (l) or uppercase (u)? u
Converted string: AMAR
londhe@developer:AIT-programs-source-code$ node Program-10.js
Welcome to String Case Converter!
Enter a string: amar
Convert to lowercase (l) or uppercase (u)?
Invalid option. Please choose 'l' for lowercase or 'u' for uppercase.
```

11. Write a program to create a calculator using Node JS.
Ans :

```
const readline = require('readline-sync');
function add(a, b) { return a + b; } // Function to perform addition
function subtract(a, b) { return a - b; } // Function to perform subtraction
function multiply(a, b) { return a * b; } // Function to perform multiplication
function divide(a, b) { if (b === 0) { return "Error! Division by zero.";  } return a / b;}
function main() {
   console.log("Welcome to Node.js Calculator!");
   // Menu
   console.log("1. Addition"); console.log("2. Subtraction");console.log("3. Multiplication");
   console.log("4. Division");
   const choice = parseInt(readline.question("Enter your choice: "));
   let num1 = parseFloat(readline.question("Enter first number: "));
   let num2 = parseFloat(readline.question("Enter second number: "));
   let result;
   // Perform operation based on user choice
   switch (choice) {
      case 1:
         result = add(num1, num2);
         break;
      case 2:
         result = subtract(num1, num2);
         break;
      case 3:
         result = multiply(num1, num2);
         break;
      case 4:
         result = divide(num1, num2);
         break;
      default:
         console.log("Invalid choice.");
         return;
   }
   console.log("Result: ", result);
}
main();
```

Output :



```
londhe@developer:AIT-programs-source-code$ node Program-11.js
Welcome to Node.js Calculator!
1. Addition
2. Subtraction
3. Multiplication
4. Division
Enter your choice: 2
Enter first number: 34
Enter second number: 43
Result:  -9
```

12. Write Program for Form validation in Angular.

Ans :

index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Program12</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>Loading...</app-root>
</body>
</html>
```

main.ts

```
import { bootstrapApplication } from '@angular/platform-browser';
import { appConfig } from './app/app.config';
import { AppComponent } from './app/app.component';

bootstrapApplication(AppComponent, appConfig)
  .catch((err) => console.error(err));
```

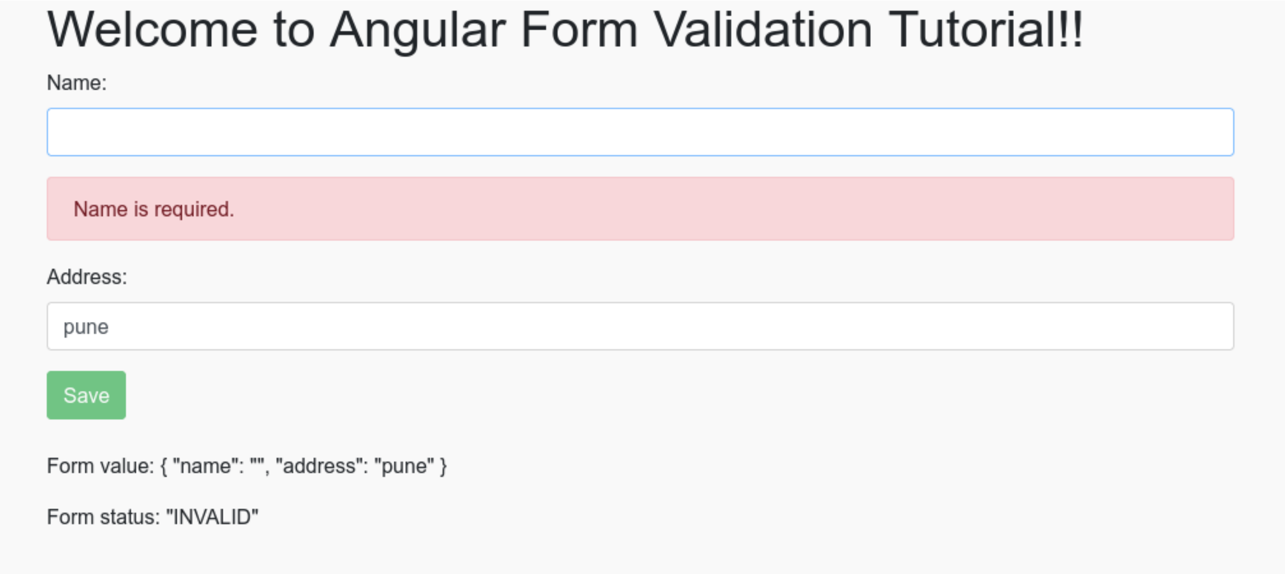
app.component.html

```
<div class="container">
  <h1>Welcome to {{ title }}!!</h1>
  <form [formGroup]="angForm" novalidate>
    <div class="form-group">
      <label>Name:</label><input class="form-control" formControlName="name" type="text" />
    </div>
    <div *ngIf="angForm.controls['name'].invalid &&(angForm.controls['name'].dirty ||
angForm.controls['name'].touched)" class="alert alert-danger">
      <div *ngIf="angForm.controls['name'].errors.required">Name is required.</div>
    </div>
    <div class="form-group">
      <label>Address:</label><input class="form-control" formControlName="address"
type="text" />
    </div>
    <div *ngIf="angForm.controls['address'].invalid && (angForm.controls['address'].dirty ||
angForm.controls['address'].touched)" class="alert alert-danger">
      <div *ngIf="angForm.controls['address'].errors.required">email is required.</div>
    </div>
    <button type="submit" [disabled]="angForm.pristine || angForm.invalid" class="btn btn-
success">Save</button>
  </form><br />
  <p>Form value: {{ angForm.value | json }}</p><p>Form status: {{ angForm.status | json
}}</p></div>
```

app.component.ts

```
import { Component } from '@angular/core';
import { FormGroup,  FormBuilder,  Validators } from '@angular/forms';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'Angular Form Validation Tutorial';
   angForm: FormGroup;
   constructor(private fb: FormBuilder) {
    this.createForm();
  }
   createForm() {
    this.angForm = this.fb.group({
       name: ['', Validators.required ]
    });
  }
}
```

Output :



Welcome to Angular Form Validation Tutorial!!

Name:

Name is required.

Address:

pune

Form value: { "name": "", "address": "pune" }

Form status: "INVALID"

13. Program to demonstrate the ngif, ngfor, ngswitch statements.

Ans :

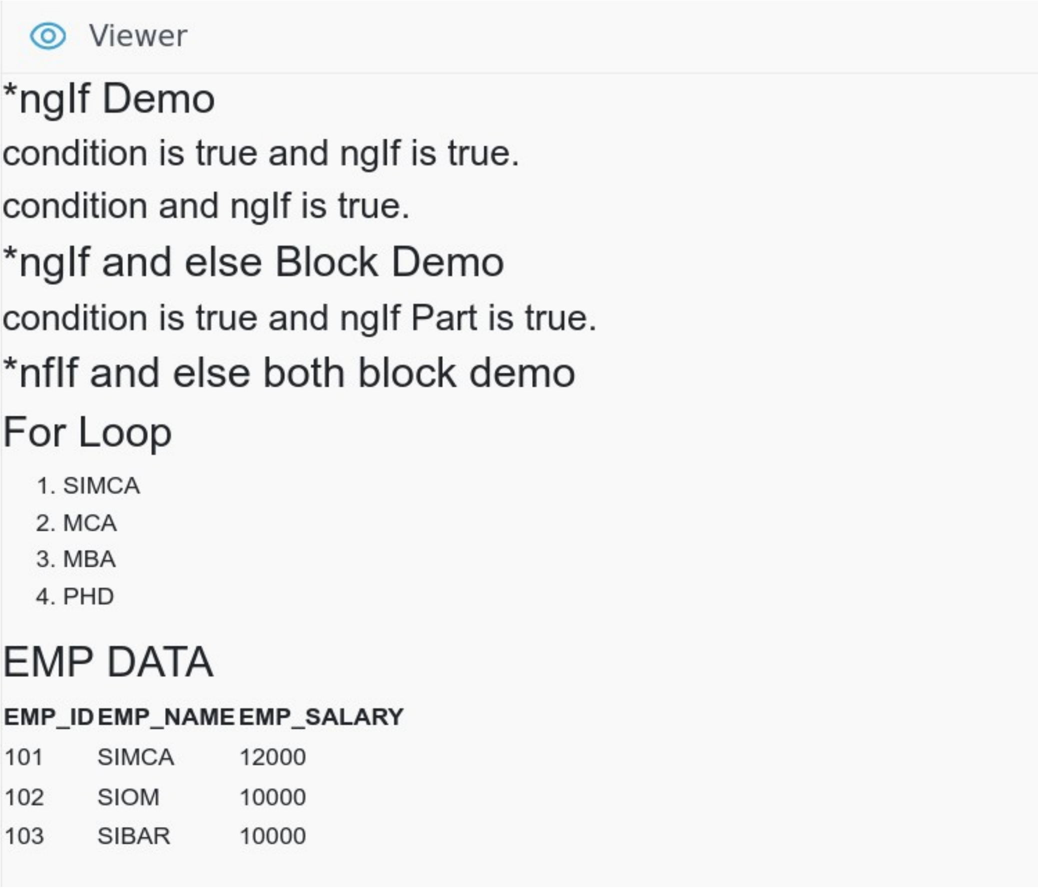
app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'HelloApp';
  show = true;
  show1 = false;
  msg = "h1";
  color1 = "blue";
  arr = ["SIMCA", "MCA", "MBA", "PHD"];
  Emp = [
    {
      Emp_id: 101,
      Emp_name: "SIMCA",
      Emp_sal: 12000
    },
    {
      Emp_id: 102,
      Emp_name: "SIOM",
      Emp_sal: 10000
    },
    {
      Emp_id: 103,
      Emp_name: "SIBAR",
      Emp_sal: 10000
    }
  ];
}
```

app.component.html

```
<h3>*ngIf Demo</h3>
<h4 *ngIf="show">condition is true and ngIf is true.</h4>
<h4 *ngIf="!show1">condition and ngIf is true.</h4>
<h3>*ngIf and else Block Demo</h3>
<h4 *ngIf="show">condition is true and ngIf Part is true.</h4>
<h3>*nfIf and else both block demo</h3>
<h4 *ngIf="msg == 'h1'; then ifBlock else eBlock">condition is true and ngIf Part is true.</h4>
<h3>For Loop</h3>
<ol><li *ngFor="let a of arr">{{ a }}</li></ol>
<h3>EMP DATA</h3>
<table>
  <tr><th>EMP_ID</th><th>EMP_NAME</th><th>EMP_SALARY</th></tr>
  <tr *ngFor="let em of Emp"><td>{{ em.Emp_id }}</td><td>{{ em.Emp_name
}}</td><td>{{ em.Emp_sal }}</td></tr>
</table>
```

Output :



*ngIf Demo

condition is true and ngIf is true.

condition and ngIf is true.

*ngIf and else Block Demo

condition is true and ngIf Part is true.

*nfIf and else both block demo

For Loop

1. SIMCA
2. MCA
3. MBA
4. PHD

EMP DATA

EMP_ID	EMP_NAME	EMP_SALARY
101	SIMCA	12000
102	SIOM	10000
103	SIBAR	10000

14. Create angular project which will demonstrate the usage of custom attribute directives

Ans :

highlight.directive.ts

```
import { Directive, HostListener, ElementRef } from '@angular/core';
@Directive({
  selector: '[appHighlight]'
})
export class HighlightDirective {
  constructor(private el: ElementRef) { }
  @HostListener('mouseenter') onMouseEnter() {
    this.highlight('yellow');
  }
  @HostListener('mouseleave') onMouseLeave() {
    this.highlight('green');
  }
  private highlight(color: string) {
    this.el.nativeElement.style.backgroundColor = color;
  }
}
```

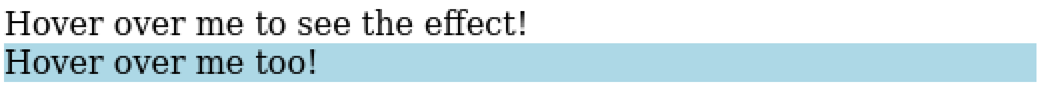
app.component.html

```
<div appHighlight>Hover over me to see the effect!</div>
<div appHighlight style="background-color: lightblue;">Hover over me too!</div>
```

app.component.ts

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet],
  templateUrl: './app.component.html',
  styleUrl: './app.component.css'
})
export class AppComponent {
  title = 'Program-14';
}
```

Output :



```
Hover over me to see the effect!
Hover over me too!
```

15. Create angular project which has HTML template and handle the click event on click of the button (Installation of Angular and Bootstrap 4 CSS Framework)

Install bootstrap : npm install bootstrap

Ans :

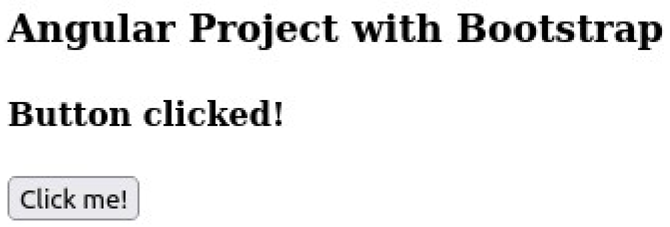
app.component.html

```
<div class="container">
  <h3>Angular Project with Bootstrap</h3>
  <h4 class="{'clickevent': clickEventText}" [textContent]="clickEventText"></h4>
  <button class="btn btn-primary" (click)="handleClick()">Click me!</button>
</div>
```

app.component.ts

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

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet],
  templateUrl: './app.component.html',
  styleUrl: './app.component.css'
})
export class AppComponent {
  title = 'Program-15';
  clickEventText: string = '';

  handleClick() {
    this.clickEventText = 'Button clicked!';
  }
}
```

Output :



Angular Project with Bootstrap

Button clicked!

16. Create an angular project to demonstrate Single Page Application (SPA).

Ans :

app.component.html

```
<div class="container">
  <ul class="nav">
    <li class="nav-item">
      <a class="nav-link active" routerLink="/" routerLinkActive="active">Home</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" routerLink="/contact" routerLinkActive="active">Contact</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" routerLink="/about" routerLinkActive="active">About</a>
    </li>
  </ul>
</div>

<h4>Hello, this is home page</h4>

<router-outlet></router-outlet>
```

app.component.ts

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

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet],
  templateUrl: './app.component.html',
  styleUrl: './app.component.css'
})
export class AppComponent {
  title = 'Program-16';
}
```

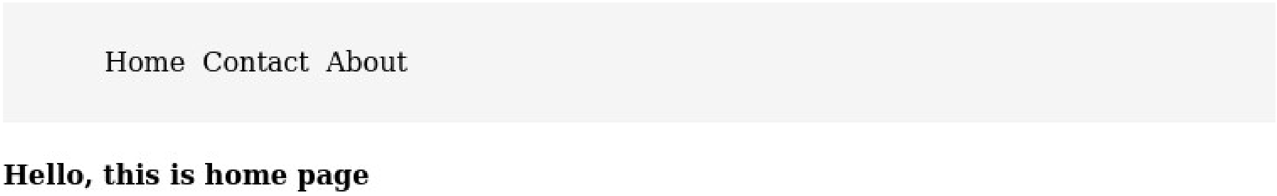
app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppComponent } from './app.component';
import { HomeComponent } from './home/home.component';
import { AboutComponent } from './about/about.component';
import { ContactComponent } from './contact/contact.component';
import { RouterModule } from '@angular/router';

@NgModule({
  declarations: [
    AppComponent,
    HomeComponent,
    AboutComponent,
    ContactComponent
  ],
  imports: [
    BrowserModule,
    RouterModule.forRoot([
        { path: '/', component: HomeComponent },
        { path: '/about', component: AboutComponent },
        { path: '/contact', component: ContactComponent },
    ])
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

app.component.css

```
.container{
    background: whitesmoke;
    padding: 10px ;
}
ul{
    display: flex;
    align-items: center;
    justify-content: start;
    list-style: none;
}

li{
    padding-left: 10px;
}
```

Output :



Home Contact About

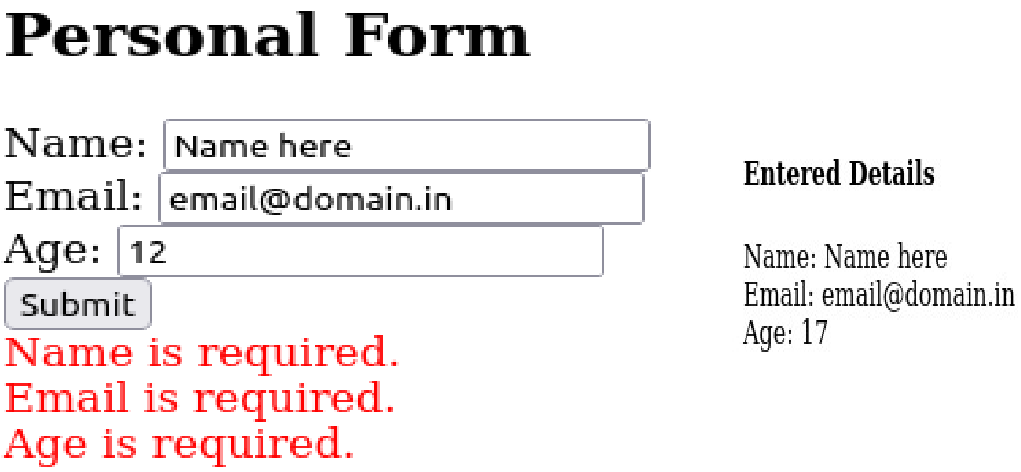
Hello, this is home page

17. Program for user interface handling like form validation and taking valid value through form and displaying it to another page.

Ans :

```
<?php $error = "";
$name = $email = $age = ""; $showDetails = false;
if (isset($_POST)) {
   if (empty($_POST["name"])) {$error .= "Name is required.<br>";} else { $name =
test_input($_POST["name"]);
      if (!preg_match("/^[a-zA-Z ]*$/", $name)) {$error .= "Only letters and white space allowed in
name.<br>";}}
   if (empty($_POST["email"])) {$error .= "Email is required.<br>";} else {
      $email = test_input($_POST["email"]);
      if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {$error .= "Invalid email
format.<br>";}}
   if (empty($_POST["age"])) {$error .= "Age is required.<br>";} else {
      $age = test_input($_POST["age"]);if (!is_numeric($age) || $age < 18 || $age > 100) {$error .=
"Age must be a number between 18 and 100.<br>";}}}
function test_input($data) {$data = trim($data);$data = stripslashes($data);$data =
htmlspecialchars($data);return $data;}
?>
<!DOCTYPE html><html><head><title>Personal Form</title></head><body>
<?php if($_POST["showDetails"] == "show"){
   echo "<h3>Entered Details : </h3>";
   echo "Name: " . $_POST["name"] . "<br>";echo "Email: " . $_POST["email"] . "<br>";echo
"Age: " . $_POST["age"] . "<br>";
} ?>
   <h2>Personal Form</h2><form method="post" action="">
   <input type="hidden" name="showDetails" value="show">
      Name: <input type="text" name="name" value="<?php echo $name; ?>"><br>Email: <input
type="text" name="email" value="<?php echo $email; ?>"><br>
      Age: <input type="text" name="age" value="<?php echo $age; ?>"><br><input type="submit"
name="submit" value="Submit">
   </form><div style="color: red;"><?php echo $error; ?></div></body></html>
```

Output :



Personal Form

Name: Name here
Email: email@domain.in
Age: 12
Submit
Name is required.
Email is required.
Age is required.

Entered Details

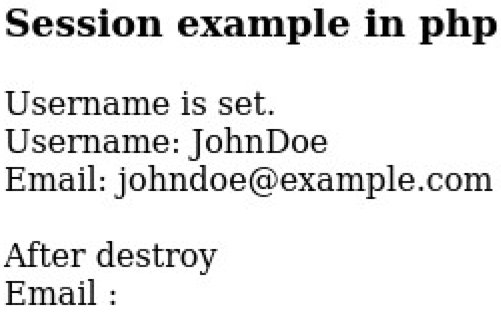
Name: Name here
Email: email@domain.in
Age: 17

18. Program to demonstrate session management using various techniques.

Ans :

```
<?php
// Starting the session
session_start();

echo "<h3>Session example in php</h3>";

// Technique 1: Setting session variables
$_SESSION['username'] = 'JohnDoe';
$_SESSION['email'] = 'johndoe@example.com';

// Technique 2: Accessing session variables
$username = $_SESSION['username'];
$email = $_SESSION['email'];

// Technique 3: Checking if a session variable is set
if (isset($_SESSION['username'])) {
   echo "Username is set. <br />";
} else {
   echo "Username is not set. <br />";
}

// Technique 4: Removing a session variable
unset($_SESSION['email']);

// Technique 5: Destroying the session
session_destroy();



// Displaying session data
echo "Username: $username<br>";
echo "Email: $email<br>";
echo "<br />";
echo "After destroy <br/>";
echo "Email : " . $_SESSION['email'];
?>
```

Output :



```
Session example in php

Username is set.
Username: JohnDoe
Email: johndoe@example.com

After destroy
Email :
```

19. Program to perform the CRUD Operations using PHP Script.

Ans :

```
<?php
function readUsersFromFile($filename) {
   $users = [];
   if (file_exists($filename)) {
      $data = file_get_contents($filename);
      $users = json_decode($data, true);
   }
   return $users;
}
function writeUsersToFile($filename, $users) {
   $data = json_encode($users);file_put_contents($filename, $data);
}
$filename = "users.json";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
   if ($_POST["action"] == "create") {
      $name = $_POST["name"];$email = $_POST["email"];
      $users = readUsersFromFile($filename);
      $users[] = ["name" => $name, "email" => $email];
      writeUsersToFile($filename, $users);echo "User created successfully.";
   }elseif ($_POST["action"] == "update") {
      $id = $_POST["id"]; $name = $_POST["name"];
      $email = $_POST["email"]; $users = readUsersFromFile($filename);
      $users[$id] = ["name" => $name, "email" => $email];
      writeUsersToFile($filename, $users); echo "User updated successfully.";
   }elseif ($_POST["action"] == "delete") {
      $id = $_POST["id"];
      $users = readUsersFromFile($filename);
      if (isset($users[$id])) {
         unset($users[$id]); writeUsersToFile($filename, $users); echo "User deleted successfully.";
      } else {echo "User not found.";} }
} elseif ($_SERVER["REQUEST_METHOD"] == "GET") {
   if ($_GET["action"] == "read") {$users = readUsersFromFile($filename); echo
json_encode($users);}
}
?>
```

Output :

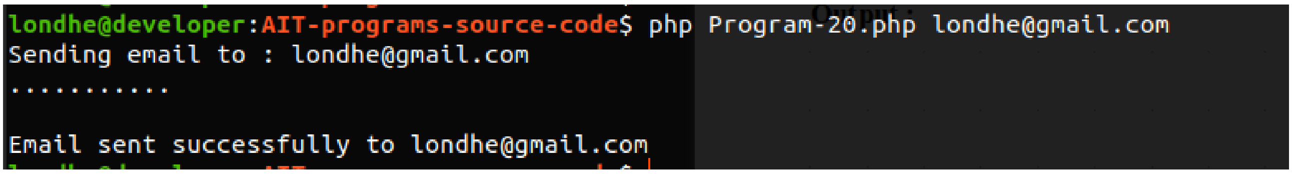
```
$ php Program-18.php

 1. Create user :  User created successfully.
  [  {
         "name": "John Doe",
          "email": "johndoe@example.com"
  }]

  2. Update user : User updated successfully.

  [{
           "name": "Jane Doe",
           "email": "janedoe@example.com"
  }]

  3. Delete user : User deleted successfully.

  [] // Blank array
```

20. Program to send email to specific email id.

Ans :

```
<?php
// Check if the email argument is provided
if ($argc < 2) {
    echo "Usage: php send_email.php <recipient_email>\n";
    exit(1);
}

$to = $argv[1];
$subject = 'Test Email';
$message = 'This is a test email sent from the command line.';
$headers = 'From: amar@ourlib.in' . "\r\n" .
    'Reply-To: amar@ourlib.in' . "\r\n" .
    'X-Mailer: PHP/' . phpversion();

echo "Sending email to : $to\n";
echo "...........\n";
echo "\n";

if (mail($to, $subject, $message, $headers)) {
    echo "Email sent successfully to $to\n";
} else {
    echo "Failed to send email\n";
}
```

Output :



```
londhe@developer:AIT-programs-source-code$ php Program-20.php londhe@gmail.com
Sending email to : londhe@gmail.com
...........

Email sent successfully to londhe@gmail.com
```