Advances In Web Technologies-(MDI3001)(L27+L28)

SRUTHI S (19MID0053)

LAB ASSESSMENT-3

1/10/2021

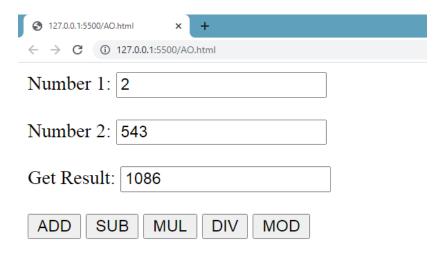
1.) Design a webpage to perform arithmetic operator

CODE:

```
<script language="javascript" type="text/javascript">
 function multiply(){
  a=Number(document.my_cal.first.value);
  b=Number(document.my_cal.second.value);
  c=a*b:
  document.my_cal.total.value=c;
  </script>
  <script language="javascript" type="text/javascript">
 function addition(){
  a=Number(document.my_cal.first.value);
  b=Number(document.my_cal.second.value);
  c=a+b:
  document.my_cal.total.value=c;
  </script>
```

```
<script language="javascript" type="text/javascript">
function subtraction(){
a=Number(document.my_cal.first.value);
b=Number(document.my_cal.second.value);
c=a-b;
document.my_cal.total.value=c;
</script>
<script language="javascript" type="text/javascript">
function division(){
a=Number(document.my_cal.first.value);
b=Number(document.my cal.second.value);
c=a/b;
document.my_cal.total.value=c;
}
</script>
<script language="javascript" type="text/javascript">
function modulus(){
a=Number(document.my_cal.first.value);
b=Number(document.my_cal.second.value);
c=a\%b;
```

```
document.my_cal.total.value=c;
}
</script>
<form name="my_cal">
Number 1: <input type="text" name="first"> <br><br>>
Number 2: <input type="text" name="second"><br><br>
Get Result: <input type="text" name="total"> <br><br>>
<input type="button" value="ADD" onclick="javascript:addition();">
<input type="button" value="SUB" onclick="javascript:subtraction();">
<input type="button" value="MUL" onclick="javascript:multiply();">
<input type="button" value="DIV" onclick="javascript:division();">
<input type="button" value="MOD" onclick="javascript:modulus();">
</form>
```



2.) design a webpage to read calculator and find the age

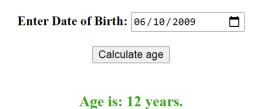
CODE:

```
<html>
<head>
<script>
function ageCalculator() {
  var userinput = document.getElementById("DOB").value;
  var\ dob = new\ Date(userinput);
  if(userinput==null || userinput==") {
   document.getElementById("message").innerHTML = "Choose a date please";
   return false;
  } else {
  //calculate month difference from current date in time
  var month_diff = Date.now() - dob.getTime();
  //convert the calculated difference in date format
  var age_dt = new Date(month_diff);
  //extract year from date
  var year = age_dt.getUTCFullYear();
  //now calculate the age of the user
  var\ age = Math.abs(year - 1970);
```

```
//display the calculated age
  return document.getElementById("result").innerHTML =
       "Age is: " + age + " years. ";
</script>
</head>
< body>
<center>
<h1 style="color: darkred;" align="center"> Calculate Age from Date of Birth <br>
<br> </h1>
<b> Enter Date of Birth: <input type=date id = DOB> </b>
<span id = "message" style="color:rgb(120, 21, 233)"> </span> <br><br>
<!-- Choose a date and enter in input field -->
<button type="submit" onclick = "ageCalculator()"> Calculate age </button>
<br><br>
<h3 style="color:32A80F" id="result" align="center"></h3>
</center>
</body>
</html>
```



Calculate Age from Date of Birth



3.) design a webpage to change colour based on user input

CODE:

```
<html>
<html>
<head>
<style>

#Myelement {

background-color: black;

width: 500px;

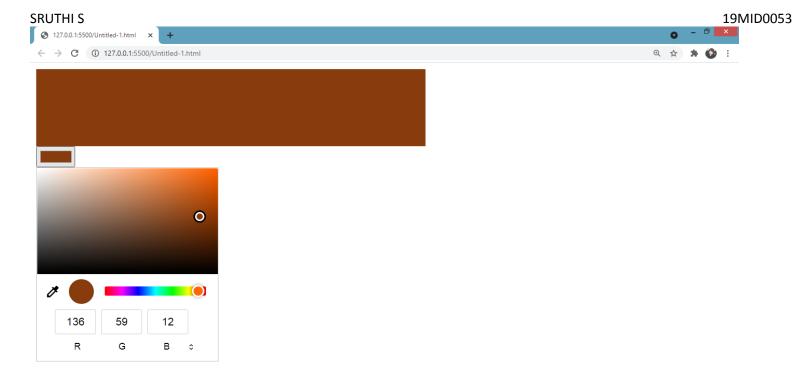
height: 100px;

}

</style>
```

```
<body>
  <div id="Myelement"</pre>
     onclick="changeColor()">
 </div>
  <input name="MyColorPicker"</pre>
      type="color"
      id="MyColorPicker"/>
  <script>
    function changeColor() {
       document.getElementById(
        "Myelement").style.backgroundColor =
         document.getElementById(
        "MyColorPicker").value;
  </script>
</body>
</html>
 RESULT:
```

SRUTHIS



4.) design a webpage to display the shapes based on user input. using any mouse event

```
CODE:
  <html>
  <style>
  body
{
  background-color: #32d132;
}
  </style>
<body>
```

<button id="circle"> Circle </button>


```
<button id="rectangle">Rectangle </button>
\langle br \rangle
<div id="shapes">
  <svg width='500' height='100'> <circle cx='50' cy='50' r='50'
style='stroke: none; fill: #3f1b94;'/></svg>
</div>
<script type="text/javascript">
var cir=document.getElementById("circle");
var rec=document.getElementById("rectangle");
cir.addEventListener("mouseover",()=>{
                 document.getElementById("shapes").innerHTML = "<svg
width='500'\ height='100'>< circle\ cx='50'\ cy='50'\ r='50'\ style='stroke:
none; fill: #CE00FF;'/></svg>";
            }):
cir.addEventListener("mouseout",()=>{
                 document.getElementById("shapes").innerHTML="<svg
width='100' \ height='100' > < circle \ cx='50' \ cy='50' \ r='40' \ style='stroke:
none; fill:#FF8181;'/></svg>";
});
rec.addEventListener("mouseover",()=>{
```

```
document.getElementById("shapes").innerHTML="<svg width='400'
height='110'><rect width='300' height='100' style='stroke: none;
fill:#CE00FF;'/></svg>";
});
```

rec.addEventListener("mouseout",()=>{

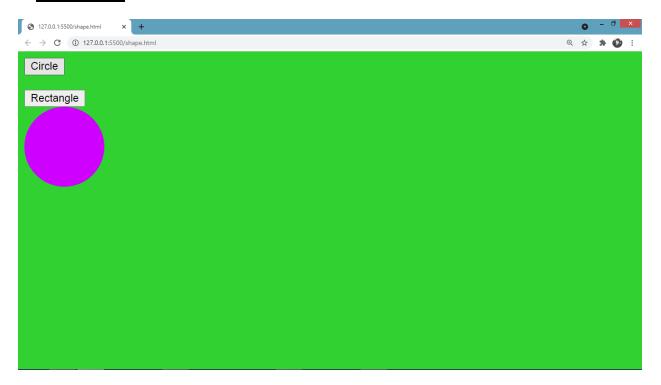
document.getElementById("shapes").innerHTML="<svg width='400' height='110'><rect width='300' height='100' style='stroke: none; fill:#FF8181;'/></svg>";

});

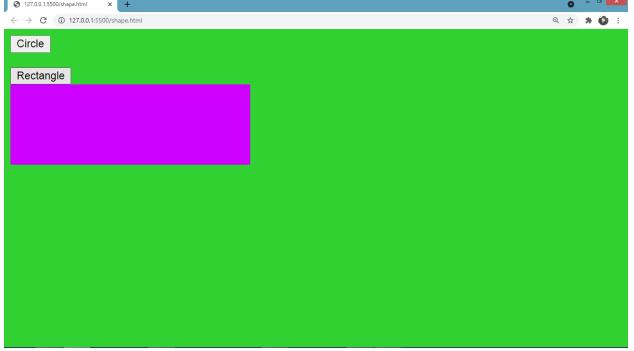
</script>

</body>

</html>







5.) design a webpage to show multiple mouse event handler on single function **CODE:**

```
<!DOCTYPE html>
<html lang="en">
<body>
  <button type="button" id="myBtn">click me</button>
  <br><br>
  <button type="button" id="myBtn1">click me twise</button>
  <br><br>
  <button type="button" id="myBtn2">right click</button>
  <script>
 function sayHelloo()
    alert("HELLO WELCOME!");
```

```
sruthis
}

var btn = document.getElementById("myBtn");

var btn1 = document.getElementById("myBtn1");

var btn2 = document.getElementById("myBtn2");

btn.addEventListener("click", sayHelloo);

btn1.addEventListener("dblclick", sayHelloo);

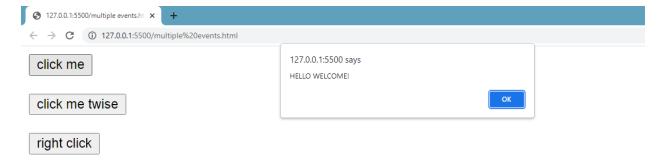
btn2.addEventListener("contextmenu", sayHelloo);

</script>

</body>

</html>
```

RESULT:



8/10/2021

1.)Angular Program:

```
TERMINAL
PS C:\Users\S.SARATH\my-first-project> ng serve

√ Browser application bundle generation complete.

Initial Chunk Files
                        Names
vendor.js
polyfills.js
                                           2.09 MB
                        vendor
                        polyfills
                                         510.59 kB
                        styles
                                          383.84 kB
                                           21.22 kB
main.js
                        main
                        runtime
                       | Initial Total |
Build at: 2021-10-15T17:25:14.582Z - Hash: 01706852053d93ffdfa0 - Time: 19909ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
```

app.component.html:

Personaldetails.components.html:

```
src > app > personaldetails > ♦ personaldetails.component.html > ♦ body > ♦ img
     <body bgcolor="blue">
<h1 align="center">PERSONAL DETAILS</h1>
    dimg src="C:/Users/S.SARATH/Desktop/sem 5/adc_web_tech/codes" alt="My picture" width="200" height="300" style="bor
     Name:
    {{name}}
    <P>AGE:</P>
     {{age}}
    DOB:
    <P>{{DOB}}</P>
     Email:
    <a href="">{{email}}</a>
    phone:
     {{phone}}
    HABITS:
    {{habits}}
```

Personaldetails.component.ts:

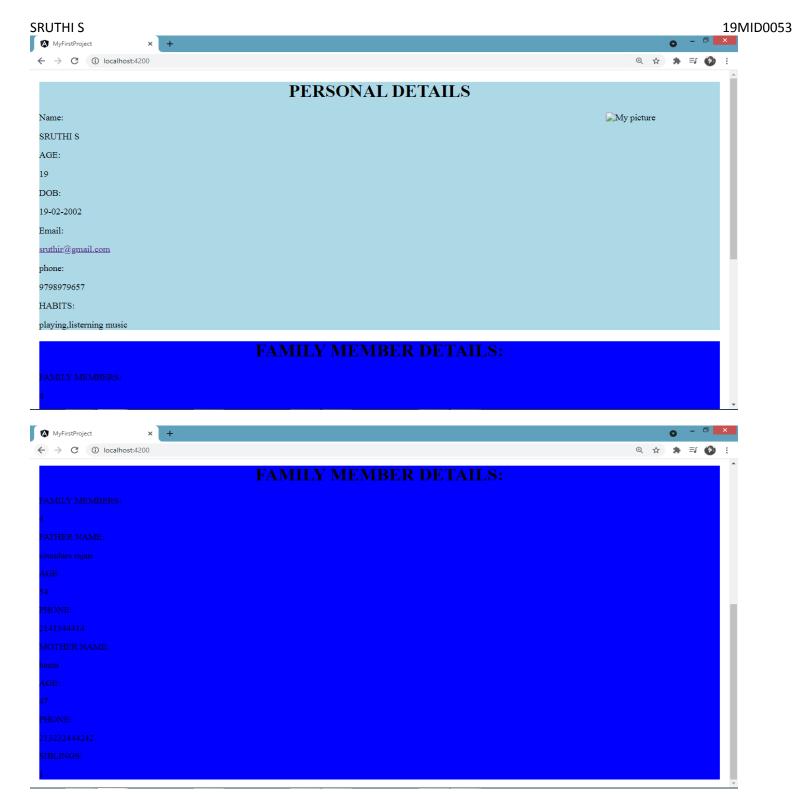
```
TS personaldetails.component.ts X
src > app > personaldetails > TS personaldetails.component.ts > 😭 PersonaldetailsComponent > 🔑 name
      import { Component, OnInit } from '@angular/core';
      @Component({
        selector: 'app-personaldetails',
        templateUrl: './personaldetails.component.html',
        styleUrls: ['./personaldetails.component.scss']
      export class PersonaldetailsComponent implements OnInit {
  9
        name:string="SRUTHI S";
        age:string="19";
        DOB:String="19-02-2002";
        email:string="sruthir@gmail.com";
        phone:string="9798979657";
         habits:string="playing,listerning music";
         constructor() { }
         ngOnInit(): void {
```

Familymemberdetails.component.html:

```
familymemberdetails.component.html X
                              personaldetails.component.html
src > app > familymemberdetails > ◆ familymemberdetails.component.html > ♦ body
     <body bgcolor="blue">
     <h1 align="center">FAMILY MEMBER DETAILS:</h1>
     FAMILY MEMBERS:
     {p>{{mem}}
     FATHER NAME:
     {{father}}
     AGE:
     {{age}}
     PHONE:
     {{phone}}
 11
     MOTHER NAME:
 12
     {{name1}}
 13
      AGE:
 14
     {{age1}}
     PHONE:
     {{phone1}}
 17
     SIBLINGS:
     {{bro}}
     </body>
 19
```

Familymemberdetails.component.ts:

```
TS familymemberdetails.component.ts •
src > app > familymemberdetails > 🏻 18 familymemberdetails.component.ts > 😭 FamilymemberdetailsComponent > 🔑 name1
      import { Component, OnInit } from '@angular/core';
      @Component({
        selector: 'app-familymemberdetails',
        templateUrl: './familymemberdetails.component.html',
        styleUrls: ['./familymemberdetails.component.scss']
      export class FamilymemberdetailsComponent implements OnInit {
       mem:string="4";
        father:string="soundara rajan";
        age:string="54";
        phone:string="2141344414";
       name1:string="hema";
        age1:string="47";
        phone1:string="213232444242";
        bro:string="1";
        constructor() { }
        ngOnInit(): void {
```



15/10/21

Using xml editor to construct the XML Tree. (<u>Online XML Editor</u> (<u>tutorialspoint.com</u>))

1.)Create XML tree model for employee management system

CODE:

<?xml version="1.0"?>

```
<EmployeeManagementSystem>
<Employee1>
```

<FirstName>SARATH</FirstName>

<LastName>RAJAN</LastName>

<Designation>CEO</Designation>

<*ContactNo>1234567890*</*ContactNo>*

< Email > sarathcrazy@gmail.com < / Email >

<Address>

<area>Aavadi</area>

<City>Chennai</City>

<State>TamilNadu</State>

<Pincode>600031</Pincode>

</Address>

</Employee1>

<*Employee2*>

<FirstName>SRUTHI</FirstName>

<LastName>RAJAN</LastName>

< Designation > Executive producer < / Designation >

<*ContactNo*>9452952552</*ContactNo*>

< Email > sruthilaya@gmail.com < / Email >

<Address>

<area>Adayar</area>

<City>Chennai</City>

<State>TamilNadu</State>

```
<Pincode>560212</Pincode>
   </Address>
</Employee2>
  <Employee3>
   <FirstName>PRIYA</FirstName>
   <LastName>KIRAN</LastName>
   <Designation>Manager</Designation>
   <ContactNo>1124413144</ContactNo>
   <Email>priya_reddy@gmail.com</Email>
   <Address>
      <area>ramapuram</area>
     <City>Chennai</City>
     <State>TamilNadu</State>
     <Pincode>560212</Pincode>
   </Address>
</Employee3>
</EmployeeManagementSystem>
```

Code Beautify

```
employeemanagementsystem ...
   employee1 ...
      firstname SARATH
      lastname RAJAN
      designation CEO
      contactno 1234567890
      email sarathcrazy@gmail.com
      address ...
         area Aavadi
         city Chennai
         state TamilNadu
         pincode 600031
   employee2 ...
      firstname SRUTHI
      lastname RAJAN
      designation Executive producer
      contactno 9452952552
      email sruthilaya@gmail.com
      address ...
         area Adayar
```

Code Beautify

```
contactno 9452952552
   email sruthilaya@gmail.com
   address ...
      area Adayar
      city Chennai
      state TamilNadu
      pincode 560212
employee3 ...
   firstname PRIYA
   lastname KIRAN
   designation Manager
   contactno 1124413144
   email priya_reddy@gmail.com
   address ...
      area ramapuram
      city Chennai
      state TamilNadu
      pincode 560212
```

2.) Create XML tree model to display our school faculty

CODE:

- <AcademicExperts>
- *<Department>*
- <Name>1.)SENSE</Name>
- <*Professor>*
- <Name>Venkat P</Name>
- <Designation>DEAN</Designation>
- </Professor>
- <*Professor>*
- <Name> Arjun</Name>
- <Designation>Senior Professor</Designation>
- <Subject>EEE</Subject>
- </Professor>
- <Name>2.)SCHEME</Name>
- <*Professor>*
- <Name>Aravindh</Name>
- <Designation>Associate Professor</Designation>
- </Professor>
 - <Name> 3.)SCOPE</Name>
- <*Professor>*
- <Name>Govindha k</Name>
- <Designation>Assistant Professor</Designation>
- <Subject>DSA</Subject>

```
</Professor>
<Professor>
```

<Name> Arun kumar</Name>

<Designation>Senior Professor</Designation>

<Subject>JAVA BASICS</Subject>

</Professor>

<Name>4.)SITE</Name>

<*Professor>*

<Name>Ganesh</Name>

<Designation>HOD</Designation>

</Professor>

<Name>5.)SELECT</Name>

<*Professor>*

<Name>Hari Niranjan</Name>

<Designation>Associate Professor</Designation>

<Subject>Stats</Subject>

</Professor>

</Department>

</AcademicExperts>

Code Beautify

```
academicexperts ...
   department ...
      name 1.)SENSE
      professor ...
        name Venkat P
         designation DEAN
      professor ...
         name Arjun
         designation Senior Professor
         subject EEE
      name 2.)SCHEME
      professor ...
         name Aravindh
         designation Associate Professor
      name 3.)SCOPE
      professor ...
         name Govindha k
         designation Assistant Professor
         subject DSA
```

Code Beautify

```
designation Assistant Professor

subject DSA

professor ..

name Arun kumar

designation Senior Professor

subject JAVA BASICS

name 4.)SITE

professor ..

name Ganesh

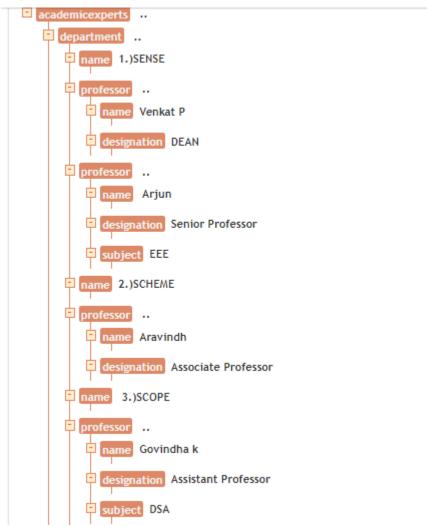
designation HOD

name 5.)SELECT

professor ..

subject Stats
```

Code Beautify



3.)Create XML tree model for e-content management

CODE:

<React>

<Intro>

<about>

React is a JavaScript library for building user interfaces.

React is used to build single-page applications.

React allows us to create reusable UI components.

</about>

</Intro>

```
cproperties>
```

<about>React.js is declarative

React.js is simple

React.js is component based

React.js supports server side

React.js is extensive

React.js is fast

React.js is easy to learn</about>

<photo>https://csharpcorner-mindcrackerinc.netdna-ssl.com/article/what-and-whyreactjs/Images/React%20Properties.jpg</photo>

</properties>

 $\langle JSX \rangle$

<about>In React, instead of using regular JavaScript for templating, it uses JSX.
JSX is a simple JavaScript that allows HTML quoting and uses these HTML tag syntax to render subcomponents. HTML syntax is processed into JavaScript calls of React Framework. We can also write in pure old JavaScript.</about>

</*JSX*>

<*CreateApp>*

<about>To learn and test React, you should set up a React Environment on your
computer.</about>

<how>This tutorial uses the create-react-app.

The create-react-app tool is an officially supported way to create React applications.

Node.js is required to use create-react-app.

```
Open your terminal in the directory you would like to create your application.</how>
  </CreateApp>
  < RunThe Application >
    <about>Run this command to move to the my-react-app directory:</about>
    <how>A new browser window will pop up with your newly created React App! If
not, open your browser and type localhost:3000 in the address bar.</how>
    <photo>https://www.w3schools.com/REACT/screenshot_myfirstreact.png</photo>
  </RunTheApplication>
  equisites>
    <about>we should have experience in the followig</about>
    <requires>
      HTML
      CSS
      JavaScript
    </requires>
  equisites>
</React>
```

