Ex.no:1 Application identification

AIM:

To identify an application to design build and display it as a composite web application.

1.Indroduction

1.1Purpose

The purpose of this document is to present a detailed description of the project Assesing management system.It will explain the purpose and features of the system,the interfaces of the system,what system will do,the constraints under which it much operate and how the system will react to external stimuli.

1.2 Scope of work

Initially assesing a project weekly was done manually and it may be difficult to store and retrive data and status of project.The application developing makes the assesment accurate and easy to grade and retrive status of the project.It stores students details and their project status.

1.3 Objectives

1.3.1 Details storage of project

This will help faculty to know which project is done by which student on which domaine etc.,

1.3.2 Status update

This helps the faculty to track the status of the project just like sprits in agile methodology.

1.3.3 Reduce paper work

As most of the will be performed online,it will reduce the usage of paper.

1.3.4 Improves efficiency

This system will make things easier for the faculty for assesing the project.

2.Functional requirements

Requirements:-

* Provides a clear user friendly UI
* Provides login/signup fuctionality
* Enter students name to navigate to that page.
* Select the action update grade or check details.
* Adding new student
* Retrive the data of student
* Remove the student
* Alter the details
* CalculatingAggregate and display

As a goal of the system is to make the process of assesing a project as simple and accurate.The functionality provided through the web assesing system is provided to the respective faculty.All the functionalies can be performed alone by the faculty and can submit the project assesment sheet at the end of the project.

3.Software Requirements

* HTML
* CSS
* MEAN STACK

MONGO DB

EXPRESS JS

ANGULAR JS

NODE JS

4.Usecase Diagram

5.Usecase Text

Actors:-

User:Any entity using the application for the purpose of Assesing project.

Usecase:-

1.Visit Site:-User can visit the project assesing site

2.Login:-Authenticates user into website

3.Add Student:-User can add student into the list and add details to store information.

4.Search for Student:-User can enter the name of the student to perform actions by navigating into that students page

5.Update:-In the weekly project assesment User can update the project status and award marks or grade.

6.Delete:-User can delete Students details or anything.

7.Alter:-User can alter details if necessary

8.Calculate Aggregate:-For final assesment by calculate aggregate action it calculates and producesses cumulative marks.

9.Logout..

RESULT:An application has been choosen and its associated Usecase/Features were discussed and finalized.

EX. No.2a CREATION OF WEB PAGE USING HTML

**Aim :**To create a web page using HTML

**Description:** A Short description about the various pages of the web application

The following tags must be incorporated in the web page:

1. Text formatting tags

|  |
| --- |
| 2. tag with borders and proper spacing, spanning and padding  3Top of Form  Tag with all types of input boxes(Text, Submit, Reset, Checkbox, Radio Buttons, Text area, Select)  4. Use of tag  5. Insert an image using **Error! Filename not specified.**tag and use tag and create four hot spots(circle, square, rectangle, polygon), that show all the related information when the hot spots are clicked  6. Create a frame with table of contents on the left side of the window, and have each entry in the table of contents use internal linking to scroll down the document to the appropriate subsection.  7. Use tag to link web pages & Internal hyper linking  8. Ordered and unordered list  9. Include HTML 5.0 tags (Audio, Video, Form creation, Canvas API, SVG, and Geo location)  Bottom of Form |

**CODE:**

<html>

<head>

<meta name="description" content="exercise 2">

<meta name="author" content="abirami">

</head>

<body bgcolor="white" text="black">

<center><h1>SSN PROJECT ASSESMENT</h1>

<hr>

<h2>PROJECT ASSESMENT MANAGEMENT SYSTEM</h2>

<img src="E:\static\uploadimg.jpg" alt="Workplace" usemap="#workmap" width="400" height="379">

<map name="workmap">

<area shape="rect" coords="34,44,270,350" alt="Computer" href="https://www.google.co.in/">

</map>

<hr>

<a href="https://kappanonline.org/assessing-student-directed-projects-brennan-blum-smith-haduong/">Have some idea about project assesment!</a>

<br>

<p style="text-align:left;">outcomes:</p>

<ul style="text-align:left;">

<li>knowledge on angular</li>

<li>Advanced Technologies</li>

</ul>

<p style="text-align:left;">Best selling Products:</p>

<ol style="text-align:left;">

<li>Aim</li>

<li>problem statement</li>

<li>implementation</li>

</ol>

<br><table border="2px">

<tr>

<td>STUDNET NAME</td>

<td>PROJECT NAME</td>

</tr>

<tr>

<td>Samanvitha</td>

<td>project assesing management</td>

</tr>

<tr>

<td>thanmayi</td>

<td>grocery store</td>

</tr>

<tr>

<td>sree</td>

<td>attendance automation</td>

</tr>

</table>

<br><form>

<label>student name</label><br>

<input type="text"><br><br>

<label>project name</label><br>

<input type="text"><br><br>

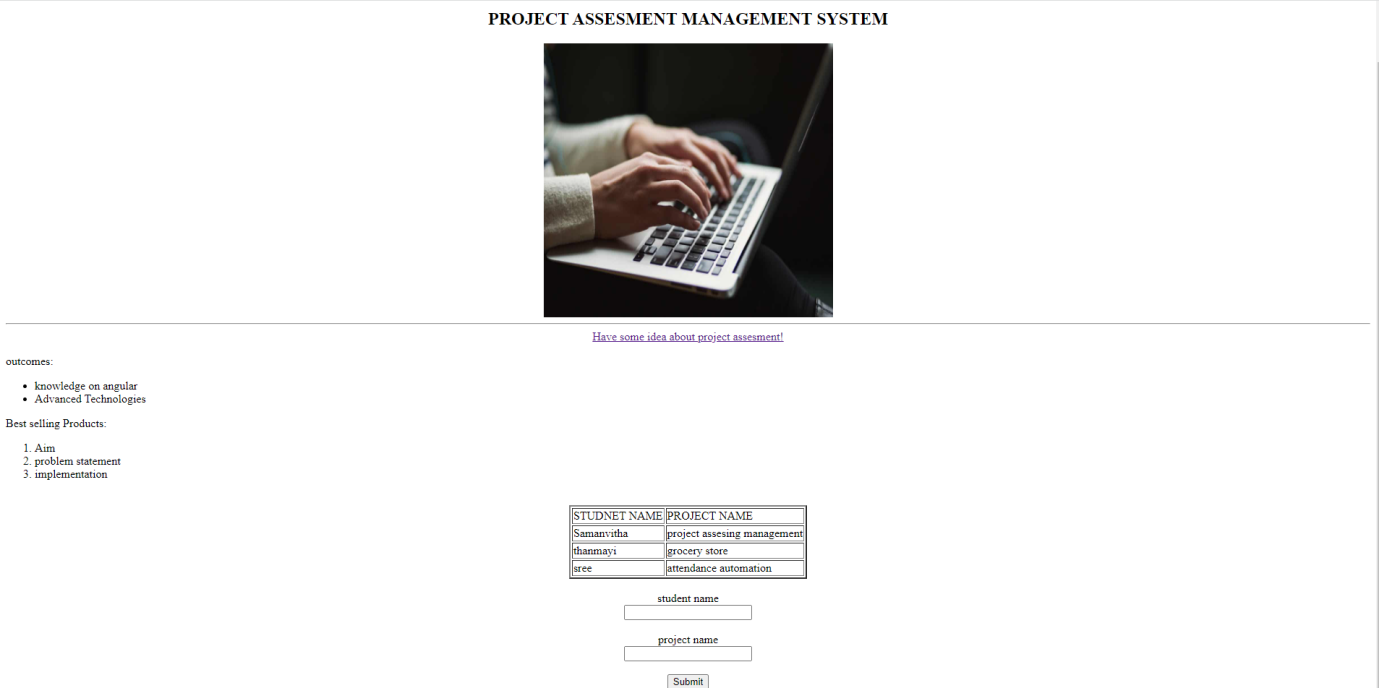
<input type="submit">

</form>

</center>

</body>

</html>



2b. HTML 5: New Structural Elements, Audio, Video, Form creation, Canvas API, SVG, and Geo location

<!DOCTYPE html>

<html>

<body style="background-color: grey">

<article>

<header>

<h1 style="background-color: lightseagreen">project assesment management</h1>

<p></p>

</header>

<footer>

<p>

Bgalkoti Samanvitha Sree<br />

<a href="mailto:hege@example.com">samanvithasree24@gmail.com</a>

</p>

</footer>

<section>

<h2 style="background-color: lightseagreen">About</h2>

<p>

The purpose of this document is to present a detailed description of the project Assesing management system.It will explain the purpose and features of the system,the interfaces of the system,what system will do,the constraints under which it much operate and how the system will react to external stimuli.

</p>

</section>

<section>

<h2 style="background-color: lightseagreen">How it works</h2>

<p>

• Provides a clear user friendly UI</p><p>

• Provides login/signup fuctionality</p><p>

• Enter students name to navigate to that page.</p><p>

• Select the action update grade or check details.</p><p>

• Adding new student</p><p>

• Retrive the data of student</p><p>

• Remove the student</p><p>

• Alter the details</p><p>

• CalculatingAggregate and display</p>

</p>

</section>

</article>

<br />

<br />

<center>

<video width="320" height="240" controls>

<source src="breadjam.mp4" type="video/mp4" />

</video>

<br />

<audio controls>

<source

src="https://filesamples.com/samples/audio/mp3/sample3.mp3"

type="audio/mp3"

/>

</audio>

</center>

<hr />

<p>Click to give your current lcation for delivery:</p>

<button onclick="getLocation()">Current location</button>

<p id="demo"></p>

<script>

var x = document.getElementById('demo');

function getLocation() {

if (navigator.geolocation) {

navigator.geolocation.getCurrentPosition(showPosition);

} else {

x.innerHTML = 'Geolocation is not supported by this browser.';

}

}

function showPosition(position) {

x.innerHTML =

'Latitude: ' +

position.coords.latitude +

'<br>Longitude: ' +

position.coords.longitude;

}

</script>

<h4>Enter your queries here:</h4>

<form>

<label for="fname">First name:</label><br />

<input type="text" id="fname" name="fname" value="John" /><br />

<label for="lname">Last name:</label><br />

<input type="text" id="lname" name="lname" value="Doe" /><br /><br />

<label for="select"></label>

<select name="select" id="select">

<option value="Damage">Reason</option>

<option value="Damage">Damage</option>

<option value="Expired">Expired</option>

<option value="Breakage">Breakage</option>

<option value="Others">Others</option>

</select>

<br />

<label for="lname1">Comments</label><br />

<textarea name="lname1" id="lname1" cols="60" rows="5"></textarea>

</form>

<canvas

id="myCanvas"

width="200"

height="100"

style="border: 1px solid #d3d3d3 >

Your browser does not support the HTML canvas tag.</canvas>

<script>

var c = document.getElementById('myCanvas');

var ctx = c.getContext('2d');

ctx.beginPath();

ctx.arc(95, 50, 40, 0, 2 \* Math.PI);

ctx.stroke();

</script>

<center>

<svg height="130" width="500">

<defs>

<linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">

<stop

offset="0%"

style="stop-color: rgb(255, 255, 0); stop-opacity: 1"

/>

<stop

offset="100%"

style="stop-color: rgb(255, 0, 0); stop-opacity: 1"

/>

</linearGradient>

</defs>

<ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />

<text fill="#ffffff" font-size="35" font-family="Verdana" x="50" y="86">

Nestle

</text>

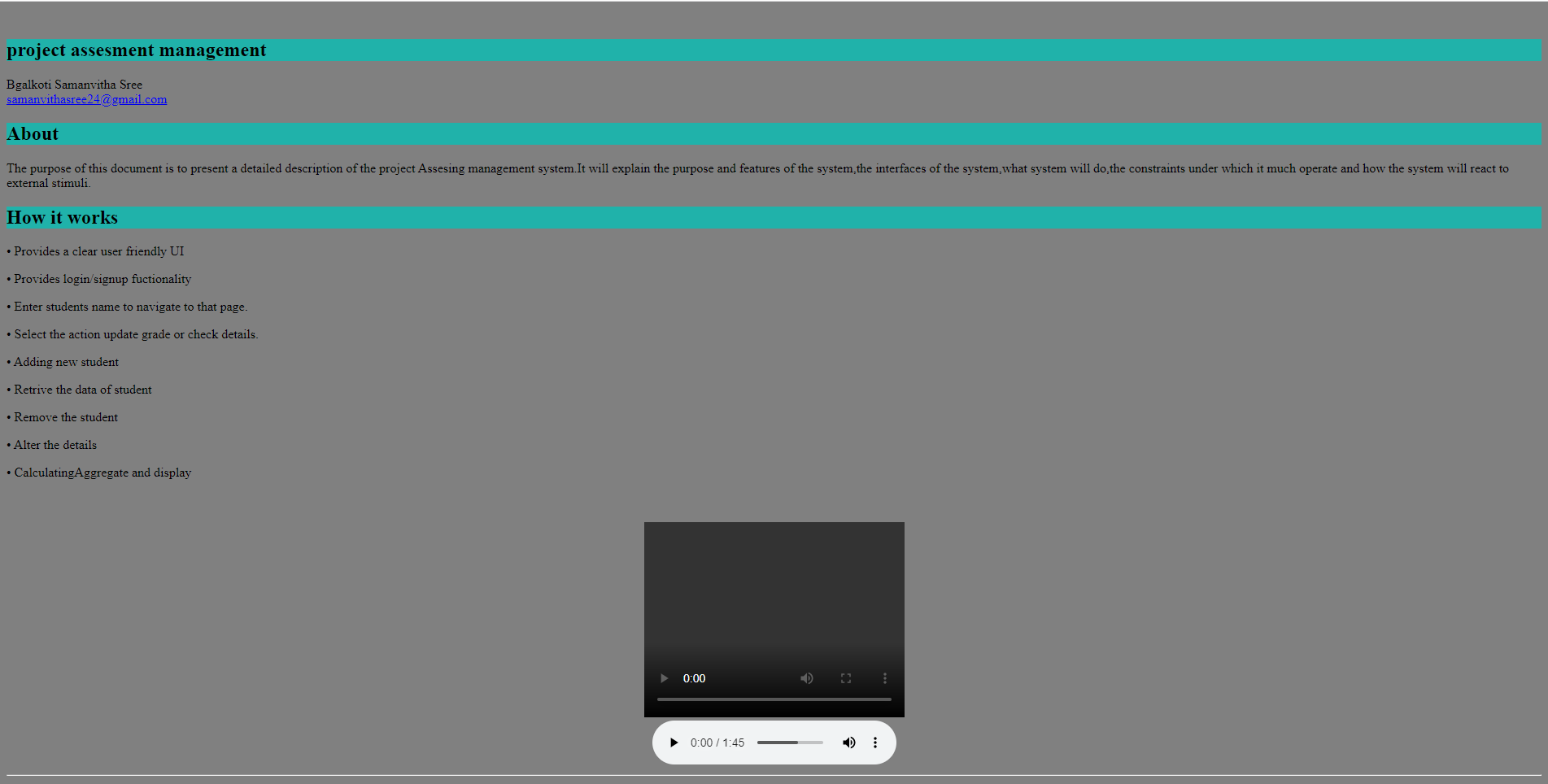
Sorry, your browser does not support inline SVG.

</svg>

</center>

</body>

</html>



RESULT: Hence web page is created using HTML and New Structural Elements, Audio, Video, Form creation, Canvas API, SVG, and Geo location are implemented.

EXERCISE-3

3a. CSS 2: Types of CSS (inline, internal and external), Selectors, Box model, Layout, and Positioning

<!DOCTYPE html>

<html>

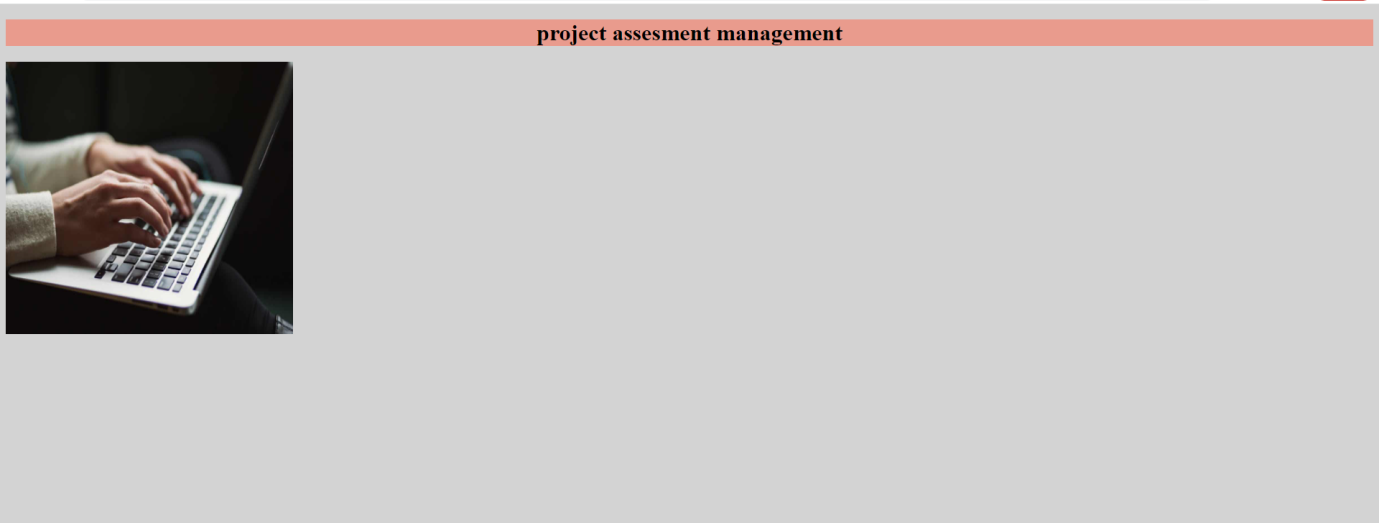
<body style="background-color:lightgrey" >

<h1 align="center" style="background-color:hsla(9, 100%, 64%, 0.5);;" style="color:blue;">project assesment management</h1>

<img src="E:\static\uploadimg.jpg" width="400" height="379" align="center">

</body>

</html>



<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

img {

display: block;

margin-left: auto;

margin-right: auto;

border:15px solid rgb(153, 9, 35);

}

#para1 {

text-align: center;

color: red;

}

div {

background-color: grey;

width: 1750px;

border: 15px solid green;

padding: 50px;

margin: 20px;

}

body{

background-color:lightgrey;

}

h1{

text-align:center;

background-color:hsla(9, 100%, 64%, 0.5);

}

</style>

</head>

<body>

<h1>project assesment management</h1>

<img src="E:\static\uploadimg.jpg" alt="Paris" style="width:50%;">

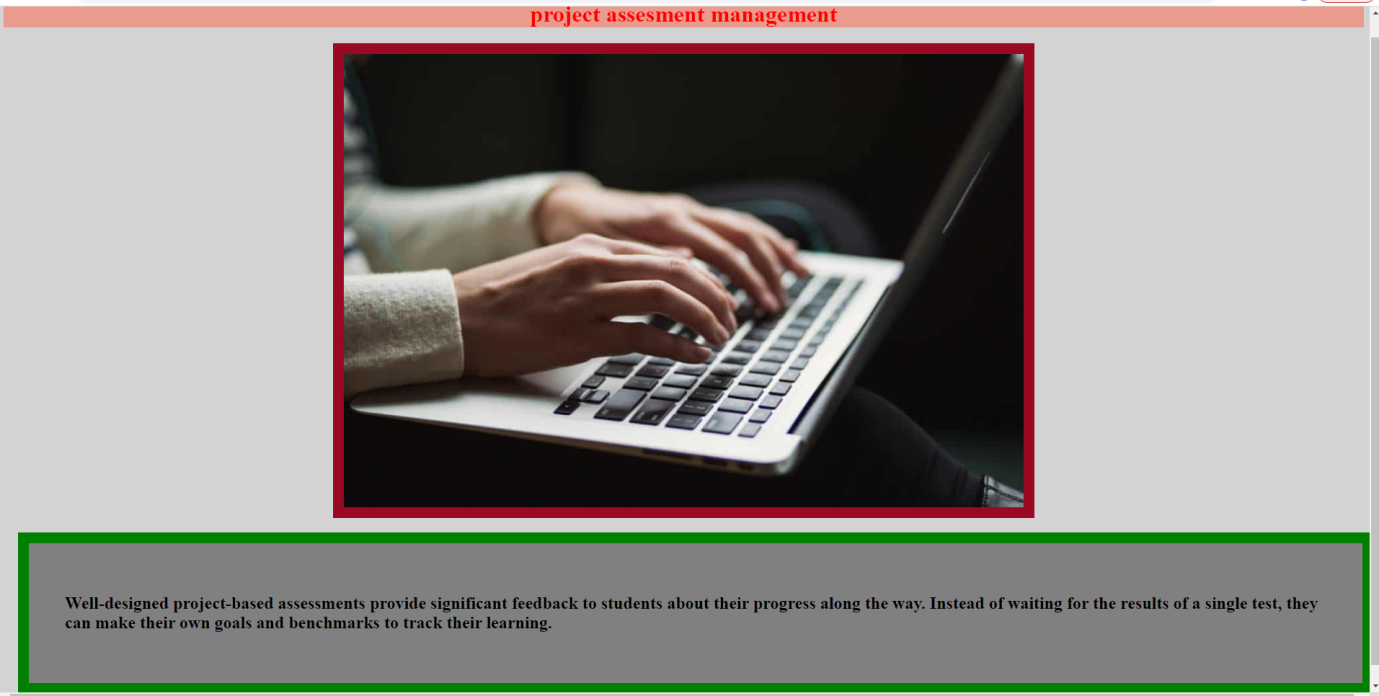
<div>

<h2>Well-designed project-based assessments provide significant feedback to students about their progress along the way. Instead of waiting for the results of a single test, they can make their own goals and benchmarks to track their learning.

</h2> </div>

</body>

</html>



External.css

input[type=text], select, input[type=password] {

width: 100%;

padding: 12px 20px;

margin: 8px 5px;

display: inline-block;

border: 1px solid #ccc;

border-radius: 14px;

box-sizing: border-box;

}

input[type=email], select, input[type=password] {

width: 100%;

padding: 12px 20px;

margin: 8px 5px;

display: inline-block;

border: 1px solid #ccc;

border-radius: 14px;

box-sizing: border-box;

}

input[type=submit] {

width: 100%;

background-color: #4CAF50;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

}

input[type=submit]:hover {

background-color: #45a049;

}

div {

border-radius: 35px;

background-color: #666666;

padding: 40px;

margin:180px 750px ;

font-family:sans-serif;

}

body{

background-image: url("../static/uploadimg.jpg");

margin: 0%;

min-height: 680px;

background-position: center;

background-repeat: no-repeat;

background-size: cover;

position: relative;

}

h2{

text-align: center;

}

.bg-image {

/\* The image used \*/

background-image: url("../static/uploadimg.jpg");

/\* Add the blur effect \*/

filter: blur(8px);

-webkit-filter: blur(8px);

/\* Full height \*/

height: 100%;

/\* Center and scale the image nicely \*/

background-position: center;

background-repeat: no-repeat;

background-size: cover;

}

Login.html

<html>

<head>

<link rel="stylesheet" href="../static/external.css">

<script>

function validateForm() {

var x = document.forms["login-form"]["username"].value;

if (x==null || x=="") {

alert("Name must be filled out");

return false;

}

var y = document.forms["login-form"]["password"].value;

if (y==null || y=="") {

alert("Password name must be filled out");

return false;

}

}

</script>

</head>

<body>

<div >

<form name="login-form" action="/login" onsubmit="return validateForm()" target="\_self" method="POST" >

<h2>Login:</h2>

{% if error %}

<p>{{ error }}</p>

{% endif %}

<input type ="text" name ="username" placeholder="Username"><br>

<input type="password" name="password" placeholder="Password"><br>

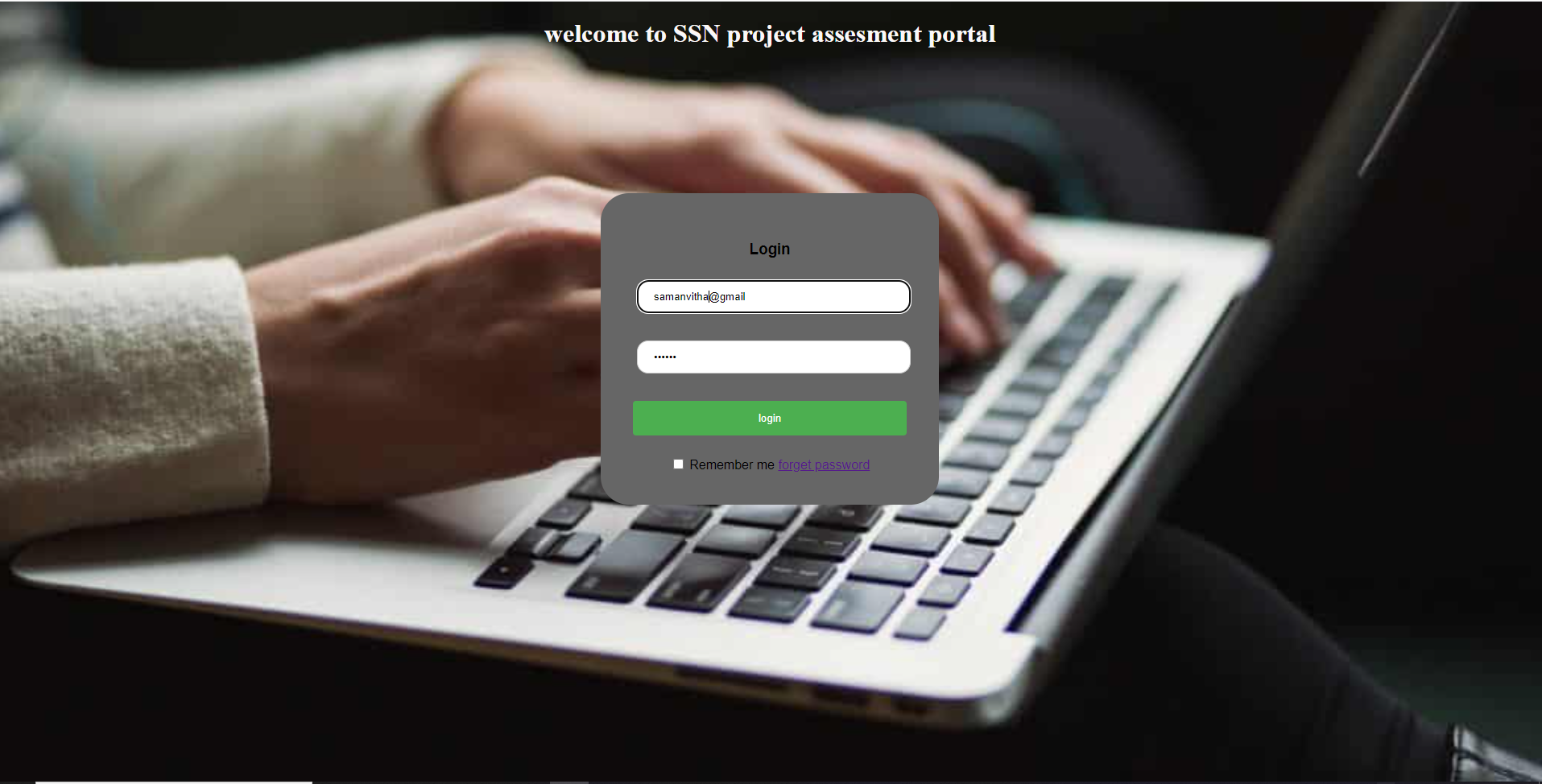
<input type="submit" value="Submit"><br>

</form>

</div>

</body>

</html>



3b. CSS 3: Borders, Text effects, Animation, and Transitions

<!DOCTYPE html>

<html>

<head>

<style>

div {

transition: width 2s, height 4s;

width: 700px;

height: 250px;

background-color: purple;

position: relative;

animation-name: example;

animation-duration: 7s;

animation-delay: 1s;

border-style:dotted;

}

@keyframes example {

0% {background-color:red; left:0px; top:0px;}

25% {background-color:yellow; left:200px; top:0px;}

50% {background-color:blue; left:200px; top:200px;}

75% {background-color:green; left:0px; top:200px;}

100% {background-color:red; left:0px; top:0px;}

}

</style>

</head>

<body>

<h1 style= color:green;>Project Assesment management</h1>

<p>About us :</p>

<div>

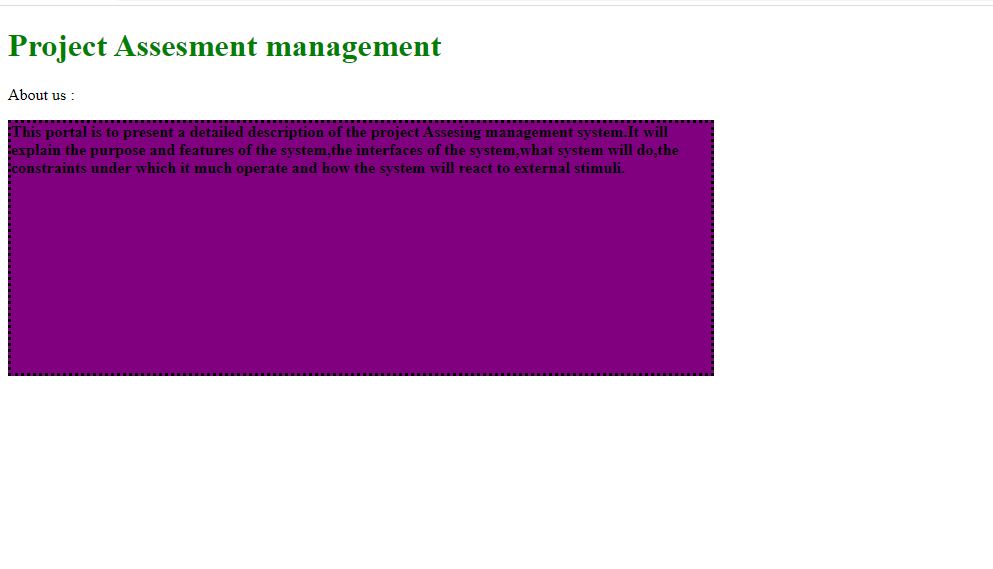
<b >

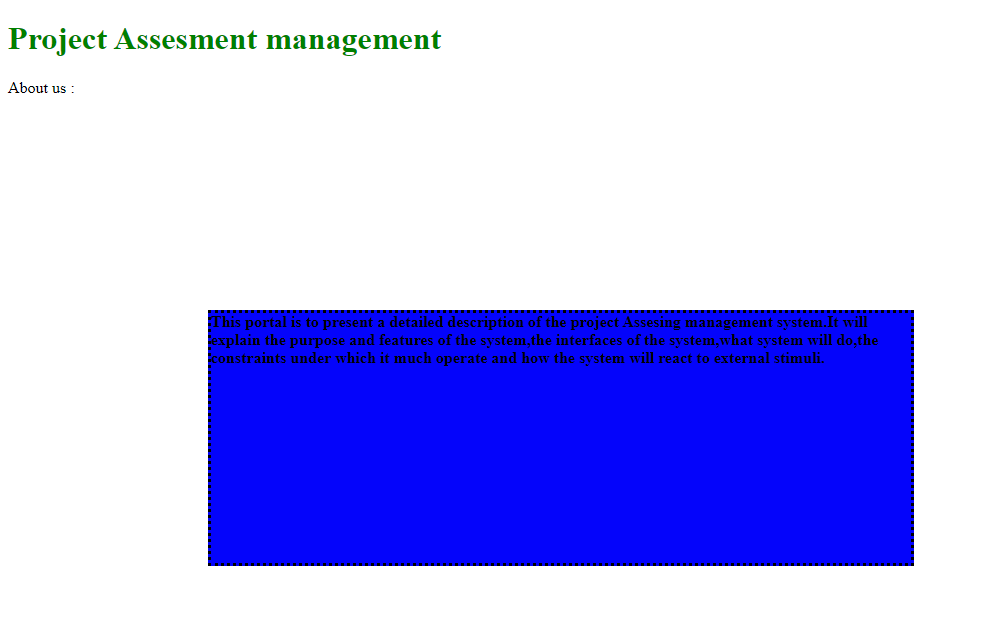
This portal is to present a detailed description of the project Assesing management system.It will explain the purpose and features of the system,the interfaces of the system,what system will do,the constraints under which it much operate and how the system will react to external stimuli.

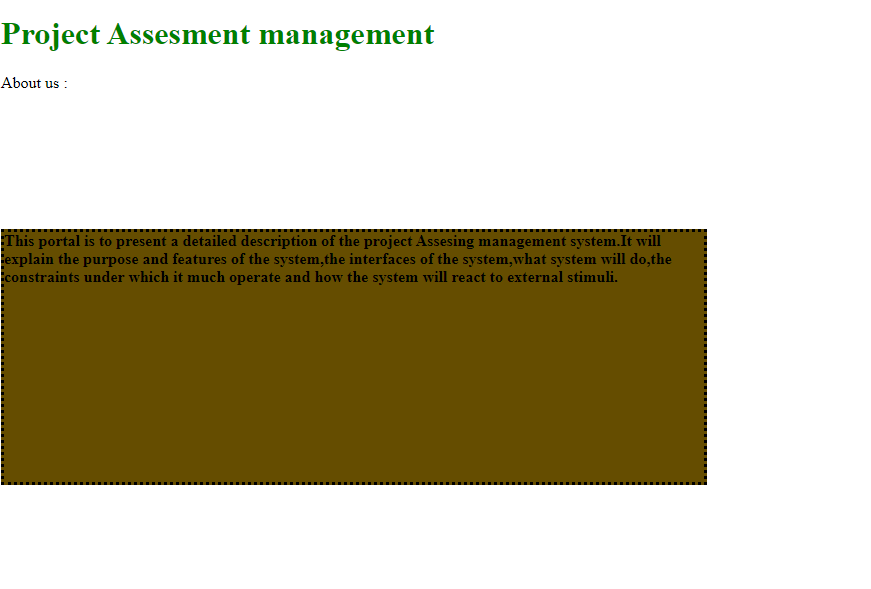
</b>

</body>

</html>







**Result:-** Inline Style sheets –Inline.Html 2. Embedded style sheets – Internal.Html 3. External style sheets – External.Html are implemented successfully

**Exercise 4a**

1. Write a Java Script that asks the user for a name, then greets the user with "Hello" followed by the user name on the page based on time.

<!DOCTYPE html>

<html lang="en">

<head>

<title>Exercise 4a</title>

<script>

function findGreeting() {

let greet = "";

let curDate = new Date();

let curTimeHrs = curDate.getHours();

if(curTimeHrs >= 5 && curTimeHrs <= 11) {

greet = 'Morning';

} else if(curTimeHrs >= 12 && curTimeHrs <= 15) {

greet = 'Afternoon';

} else if(curTimeHrs >= 16 && curTimeHrs <= 19) {

greet = 'Evening';

} else if(curTimeHrs >= 20 && curTimeHrs <= 23) {

greet = 'Night';

} else if(curTimeHrs >= 0 && curTimeHrs <= 4) {

greet = 'Night';

}

return greet;

}

let name = prompt("Enter Name");

alert('Hello ' + name + ' Good' + findGreeting());

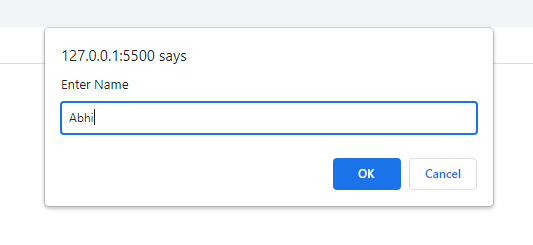
</script>

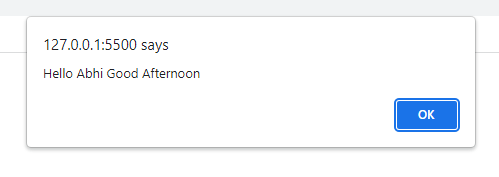
</head>

<body>

</body>

</html>





1. Write a Java Script that collects the numbers from a page and then adds them up and prints them to a blank field on the page.

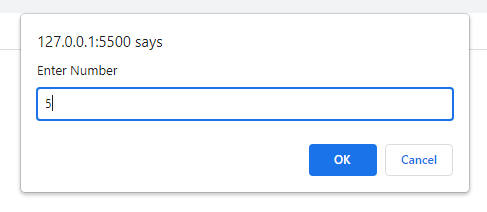
<script>

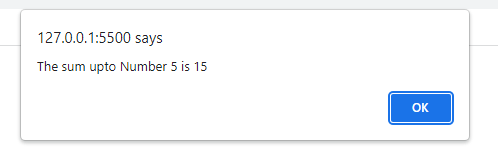
let num = prompt("Enter Number");

let sum = (num \* (num+1)) / 2;

alert('The sum upto Number ' + num + ' is ' + sum);

</script>





1. Write a Java Script that prompts the user for a number and then counts them from 1 to that number displaying only the odd numbers.

<script>

let num = prompt("Enter Number");

let arr = [];

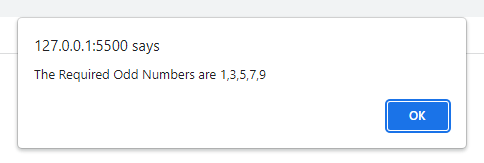
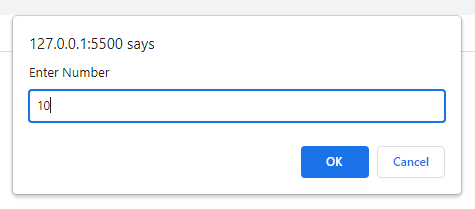
for(var i=1; i <=num; i++) {

if(i%2 != 0) arr.push(i);

}

alert('The Required Odd Numbers are ' + arr);

</script>



4. Write a Java Script to create a form with labels and submit button to provide feedback/hint to the user.

<body>

<form action="" method="POST" onsubmit="alert('Thanks for the Feedback')">

<label for="fullName">Your Name</label> <br>

<input type="text" name="fullName" required><br><br>

<label for="gender">Your Gender</label> <br>

<input type="radio" name="gender" value="male">Male<br>

<input type="radio" name="gender" value="female">Female<br><br>

<label for="email">Your E-mail</label> <br>

<input type="text" name="email" required><br>

<br>

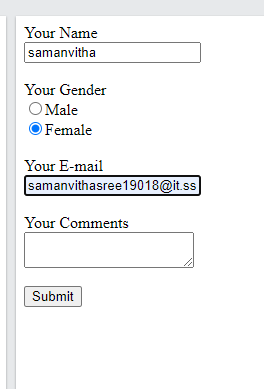
<label for="comments">Your Comments</label> <br>

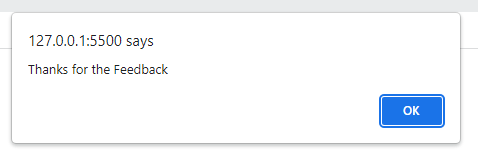
<textarea name="comments"></textarea><br><br>

<input type="submit" value="Submit">

</form>

</body>





1. Write a Java Script to create a form with a text box to ensure the field is not empty and have proper http:// else notify the user.

<head>

<title>Exercise 4a</title>

<script>

function check() {

let usrUrl = document.getElementById('usrurl').value;

if(!usrUrl.startsWith('http://')) {

alert("url must start with 'http://'");

}

}

</script>

</head>

<body>

<form action="">

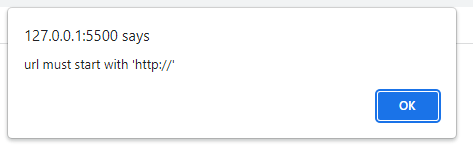
<label for="usrurl">Enter Url</label> <br>

<input type="text" name="usrurl" required id="usrurl"><br><br>

<input type="submit" value="Submit" onclick="check()">

</form>

</body>



**EXERCISE-4b**

<!DOCTYPE html>

<html>

<head>

<title>project system</title>

<style>

legend {

display: block;

padding-left: 2px;

padding-right: 2px;

border: none;

}

tr{

padding-left: 90%;

padding-top: 90%;

padding-bottom: 90%;

padding-right: 90%;

}

</style>

<script type="text/javascript">

function validate() {

var user = document.getElementById("e").value;

var user2 = document.getElementById("e");

var re = /^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/;

if (re.test(user)) {

alert("done");

return true;

}

else {

user2.style.border = "red solid 3px";

return false;

}

}

</script>

</head>

<body bgcolor="beige">

<center>

<h1>Registration of student project</h1>

<form>

<fieldset style="width:300px">

<legend>Form</legend>

<table>

<tr>

<input type="text"

placeholder="firstname"

maxlength="10">

</tr>

<br><br>

<tr>

<input type="text"

placeholder="lastname"

maxlength="10">

</tr>

<br><br>

<tr>

<input type="text"

placeholder="project\_title" id="e">

</tr>

<br><br>

<tr>

<input type="text" placeholder="domain">

</tr>

<br><br>

<tr>

<input type="text" placeholder="contact">

</tr>

<br><br>

<tr><input type="submit"

onclick="validate()" value="create">

</tr>

</table>

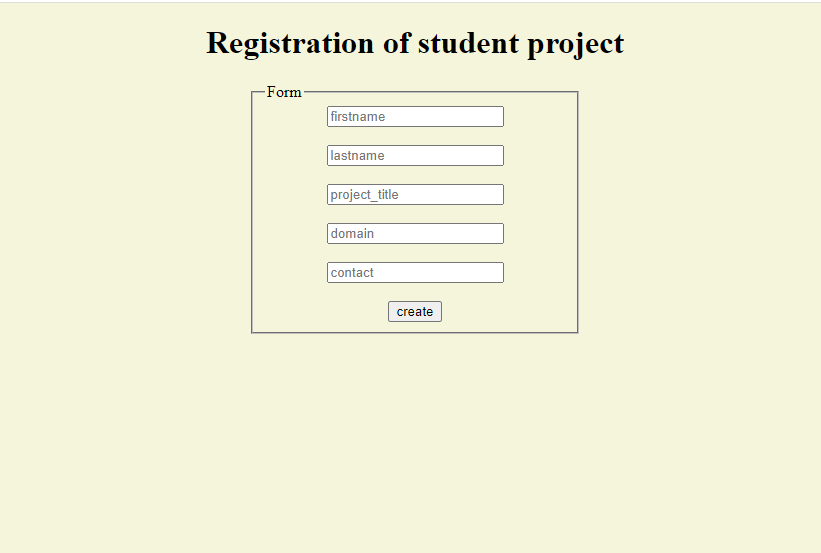
</fieldset>

</form>

</center>

</body>

</html>



Exercise-4c

HTML DOM Implement a HTML DOM using JavaScript

The steps are

Get a document builder using document builder factory and parse the html file to create a DOM object

Get a list of elements from the DOM

Manipulate - DOM tree.(Insert, Delete, Search, Modify, Display)

<!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="stylee.css">

<title>15 puzzle</title>

</head>

<body>

<div class='game'>

<table class="tableClass">

<tr>

<td class='normal' id="1" onclick="action(this)"></td>

<td class='normal' id="2" onclick="action(this)">1</td>

<td class='normal' id="3" onclick="action(this)">2</td>

<td class='normal' id="4" onclick="action(this)">3</td>

</tr>

<tr>

<td class='normal' id="5" onclick="action(this)">4</td>

<td class='normal' id="6" onclick="action(this)">5</td>

<td class='normal' id="7" onclick="action(this)">6</td>

<td class='normal' id="8" onclick="action(this)">7</td>

</tr>

<tr>

<td class='normal' id="9" onclick="action(this)">8</td>

<td class='normal' id="10" onclick="action(this)">9</td>

<td class='normal' id="11" onclick="action(this)">10</td>

<td class='normal' id="12" onclick="action(this)">11</td>

</tr>

<tr>

<td class='normal' id="13" onclick="action(this)">12</td>

<td class='normal' id="14" onclick="action(this)">13</td>

<td class='normal' id="15" onclick="action(this)">14</td>

<td class='normal' id="16" onclick="action(this)">15</td>

</tr>

</table>

<button class="reset" onclick="randomize()">Reset</button>

<p class='hide' id='issue'> Not Adjacent!! </p>

<p class='hide' id='finished'> Congratulations!! </p>

</div>

<script defer src="script.js"></script>

</body>

</html>

STYLE.CSS

.tableClass td{

border:1px solid black;

}

.normal{

border:1px solid black !important;

}

.valid{

border: 1px solid blue !important;

}

.invalid{

border:1px solid red !important;

}

.table\_final{

border:1px solid white;

}

.game

{

position: relative;

top: 25rem;

left: 50%;

}

.hide{

visibility: hidden;

}

.show{

visibility: visible;

}

SCRIPT.JS

var selected=0;

var row=0;

var column=0;

function randomize()

{

numbers=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15];

document.getElementById(1).innerHTML='';

i=2;

while(numbers.length!=0)

{

x=Math.floor(Math.random() \* numbers.length);

val=numbers[x];

document.getElementById(i).innerHTML=val;

numbers.splice(x,1);

i+=1;

}

}

function hideagain(x,issue)

{

x.className='normal';

document.getElementById(row\*4+column+1).className='normal';

document.getElementById(issue).className='hide';

}

function completed()

{

for(var i=2;i<=16;i++)

{

if(document.getElementById(i).innerHTML!=i-1)

return 0;

}

return 1;

}

function neighbour(r,c)

{

var arr=[[row-1,column],[row,column-1],[row+1,column],[row,column+1]];

if(arr.findIndex((val)=>{return val[0]==r && val[1]==c})!=-1)

return 1;

return 0;

}

function action(element)

{

if(selected==0)

{

if(element.innerHTML=='')

{

element.className='valid';

selected=1;

var tot=element.id;

row=Math.floor((tot-1)/4);

column=tot-(row\*4)-1;

}

}

else

{

if(element.innerHTML=='')

{

element.className='normal';

}

else

{

var tot=element.id;

row\_selected=Math.floor((tot-1)/4);

column\_selected=tot-(row\_selected\*4)-1;

if(neighbour(row\_selected,column\_selected))

{

var temp=element.innerHTML;

var emptyval=document.getElementById(row\*4+column+1);

element.innerHTML=emptyval.innerHTML;

emptyval.innerHTML=temp;

emptyval.className='normal';

row=row\_selected;

column=column\_selected;

}

else

{

element.className='invalid';

document.getElementById(row\*4+column+1).className='invalid';

document.getElementById('issue').className='show';

showfor1s=setTimeout(hideagain, 1000, element,'issue');

}

}

if(completed())

{

document.getElementById('finished').className='show';

showfor1s=setTimeout(() => {document.getElementById('finished').className='hide';randomize();}, 1000);

}

selected=0;

}

}

**EXERCISE 5**

**5a.ANGULARJS - SIMPLE CALCULATOR**

**AIM:**

To design a webpage for simple calculator

**ALGORITHM:**

1. IncludetheappropriateCDN linksof AngularJS
2. Addng-app to theenclosing tag(divor body)always.

# Calculator

* + Inadd-marksform,useAngularJSmodel-viewbindingtocalculatethetotalandaverageautomatically
  + Thedirectivesusedareng-init,ng-bindand ng-model
  + *Example*:
    - Importtheangularscript
    - Getoperand1 andoperand 2 asinput fromthe userusinginput tag
    - Getthe operationto beperformed fromtheuser usingselecttag
    - Thedivenclosingthecalculatorisdirectedtotherespectiveng-appandng-controller.
    - Insidetheangularcontroller,definefunctionresult()andusing$scope,gettheoperation required
    - Usingifstatementsperformthenecessaryoperationontheoperandsaandbwhich arelinked by ng-model and returnit
    - Resultisinvoked inthecalculator whichwillgive therequiredoutput

**CODE:**

<!DOCTYPE html>

<head>

<title>Simple calculator</title>

<script src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.8.3/angular.min.js"></script>

<script>

angular.module('CalculatorApp', [])

.controller('CalculatorController', function ($scope) {

$scope.result = function () {

if ($scope.operator == '+') {

return $scope.a + $scope.b;

}

if ($scope.operator == '-') {

return $scope.a - $scope.b;

}

if ($scope.operator == '\*') {

return $scope.a \* $scope.b;

}

if ($scope.operator == '/') {

return $scope.a / $scope.b;

}

};

});

</script>

</head>

<body>

<h1>SIMPLE CALCULATOR</h1>

<div ng-app="CalculatorApp" ng-controller="CalculatorController">

<p>Enter a number: <input type="number" ng-model="a"></p>

<p>Enter a number:<input type="number" ng-model="b"></p>

<p>Select an operation: <select ng-model="operator">

<option>+</option>

<option>\*</option>

<option>-</option>

<option>/</option>

</select></p>

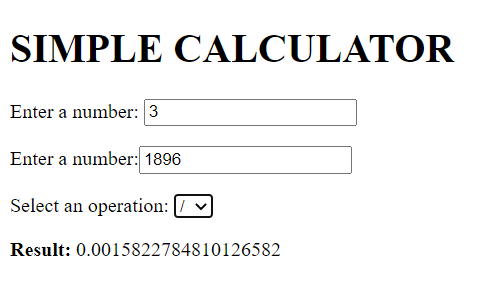
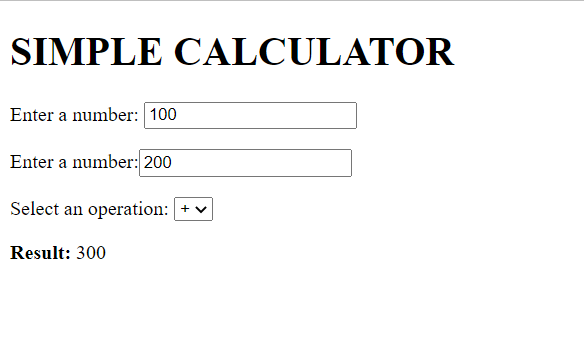
<p><strong>Result: </strong>{{ result() }}</p>

</div>

</body>

</html>

**OUTPUT:**



# 

# 

# 5b. Angular program to implement Components, Service, and Controllers

Aim: implement components, controllers, service and filters.

# Service

* + Usingapp.service(), wecan createcustomservices
  + Thecustomservice,takesinastringandreturnsastringwithstyledquotesaroundit.
  + *Example:*
    - Here,inthisapplication,wearedisplaying thedataof differentIPLteams.
    - Forthis,thedatatobedisplayedisobtainedusingthehttpserviceandthefileobtained is of JSON format.
    - Usingthisdataandwiththehelpofangularcommands,wedisplaythenecessarydata.
    - Thisprogrammustberunusingtheliveserverasweareusingthehttpserviceand ifrun locally, wewon’t be gettingthedesiredoutput.

# Controller

* + Controllerscan be usedto setinitialdata and definefunctions.
  + app.controller()canbeusedtocreatecontrollersandng-controllerdirectiveisusedto usethem
  + Thecontrollerstoresthedatalistontoascope variable.
  + Thecomponentalsousesacontroller,whichhasthefunctionfetchingrandomquotes.

# Component

* + Youhavecreateacomponentcalled“quotes”whichfetchesrandomquotesfroman API and displaysit.
  + Youuse theinbuilt$http, $intervalservices and alsoyourcustomservice.
  + Touse the inbuilt services,we haveto firstincludethemin the dependency array.
  + app.component()canbeusedtocreatethecomponentandthecomponentnameisusedlike a html tag.
  + Example:
    - Importtheangularscript
    - Createa divand directittotherespectiveng-appandng-controller
    - Defineacomponent forthe angularmodule
    - Insidethetemplateuseinputtagstoinputvaluesofname,age,email,passwordfromthe user and create alogin buttonusing HTML tags
    - Inthecontrollerinvokethe inbuilttimeserviceinordertodisplaytime
    - Insidethedivtag invokethis component byusing thename ofyourcomponent as a tag

<!DOCTYPE html>

<html>

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

<body>

<h2>ng-controller directive</h2>

<div ng-app="myApp" ng-controller="myCtrl">

Full Name: {{firstName + " " + lastName}}

</div>

<script>

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

app.controller('myCtrl', function ($scope) {

$scope.firstName = "Samanvitha";

$scope.lastName = "sree";

});

</script>

</body>

</html>

//app.js

//component.js

angular.module('newApp', []);

angular.module('newApp').

component('newList', {

template:

'<ul>' +

'<li ng-repeat="item in $ctrl.names">' +

'<span>{{item.name}}</span>' +

'<p>{{item.department}}</p>' +

'</li>' +

'</ul>',

controller: function newListController() {

this.names = [

{

name: 'Abirami S',

department: 'Information Technology '

}, {

name: 'Jothilaxmi H',

department: 'Chemical Engineering'

}, {

name: 'Mohasin S',

department: 'Mechanical Engineering'

}, {

name: 'Dharshan S',

department: 'Computer Science Engineering'

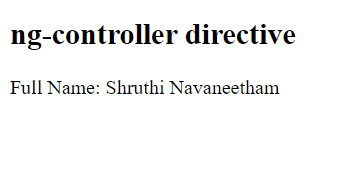
}

];

}

});

OUTPUT:



samanvitha

Services:

Creating a custom service:

CODE:

<!DOCTYPE html>

<html>

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

<body>

<div ng-app="myApp" ng-controller="myCtrl">

<p>The hexadecimal value of 255 is:</p>

<h1>{{hex}}</h1>

</div>

<p>A custom service with a method that converts a given number into a hexadecimal number.</p>

<script>

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

app.service('hexafy', function () {

this.myFunc = function (x) {

returnx.toString(16);

}

});

app.controller('myCtrl', function ($scope, hexafy) {

$scope.hex = hexafy.myFunc(255);

});

</script>

</body>

</html>

<!DOCTYPE html>

<html ng-app="myapp">

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

<link rel="stylesheet" href="../static/style.css">

<body align="center" ng-init="name='welcome to SSN project assesment portal'" >

<p><h1 style="color:rgb(251, 248, 248);">{{name}}</h1></p>

<div >

<form ng-controller="mycntrl" name="loginform" action="E:/templates/studentdetails.html">

<h3>Login</h3>

<input type="email" name="email" ng-model="email" required>

<span style="color:red" ng-show="loginform.email.$dirty && loginform.email.$invalid">

<p ng-show="loginform.email.$error.required">Email is required.</p>

<p ng-show="loginform.email.$error.email">Invalid email address.</p>

</span><br><br>

<input type="password" ng-model="pass" required><br><br>

<input type="submit" value="login" ng-init="email='saman@gmail.com'" ><br><br>

<input type="checkbox" id="remember" name="remember" value="Remember me">

<label for="remember"> Remember me</label>

<a href="https://www.google.co.in/">forget password</a>

</form>

</div>

<script>

var app=angular.module("myapp",[]);

app.controller("mycntrl",function($scope){

$scope.email="samanvitha@gmail.com",

$scope.pass="Abcd#1"

$scope.student=[{'name':'saman@gmail.com','password':'12345'}

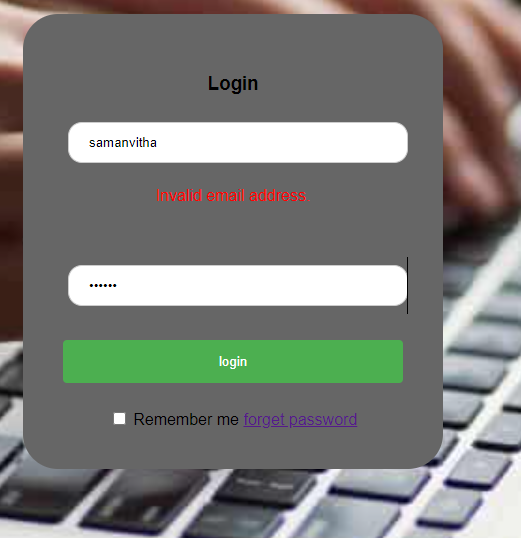
]

});

</script>

</body>

</html>



Added angular to the login form using controller and components

**5c. Angular program to implement search to filter items**

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

\* {

box-sizing: border-box;

}

#myInput {

background-image: url('/css/searchicon.png');

background-position: 10px 12px;

background-repeat: no-repeat;

width: 100%;

font-size: 16px;

padding: 12px 20px 12px 40px;

border: 1px solid #ddd;

margin-bottom: 12px;

}

#myUL {

list-style-type: none;

padding: 0;

margin: 0;

}

#myUL li a {

border: 1px solid #ddd;

margin-top: -1px; /\* Prevent double borders \*/

background-color: #f6f6f6;

padding: 12px;

text-decoration: none;

font-size: 18px;

color: black;

display: block

}

#myUL li a:hover:not(.header) {

background-color: #eee;

}

</style>

</head>

<body>

<h2>Project List</h2>

<input type="text" id="myInput" onkeyup="myFunction()" placeholder="Search for names.." title="Type in a name">

<ul id="myUL">

<li><a href="#">attendance automation</a></li>

<li><a href="#">project assesment management</a></li>

<li><a href="#">chatbot</a></li>

<li><a href="#">grocery management</a></li>

<li><a href="#">restaurent management</a></li>

<li><a href="#">class timetable</a></li>

<li><a href="#">blog management</a></li>

</ul>

<script>

function myFunction() {

var input, filter, ul, li, a, i, txtValue;

input = document.getElementById("myInput");

filter = input.value.toUpperCase();

ul = document.getElementById("myUL");

li = ul.getElementsByTagName("li");

for (i = 0; i < li.length; i++) {

a = li[i].getElementsByTagName("a")[0];

txtValue = a.textContent || a.innerText;

if (txtValue.toUpperCase().indexOf(filter) > -1) {

li[i].style.display = "";

} else {

li[i].style.display = "none";

}

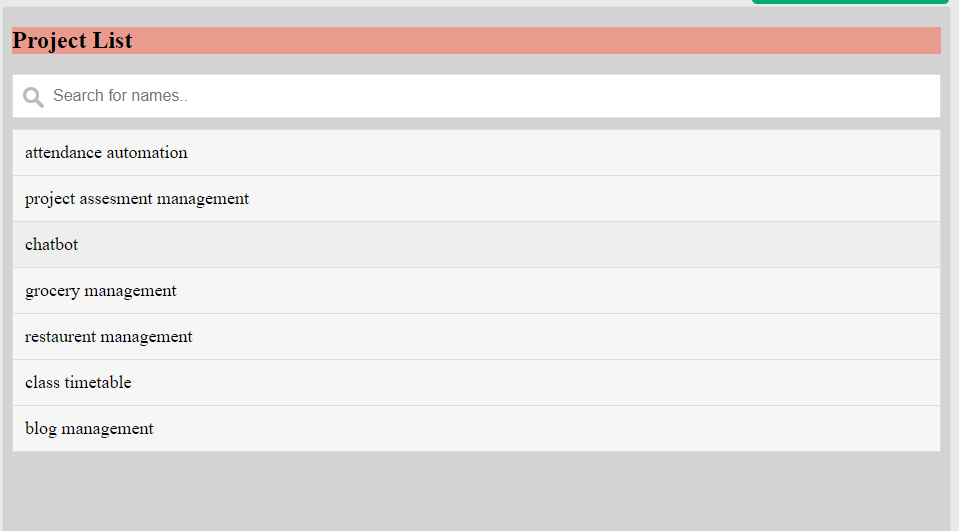
}

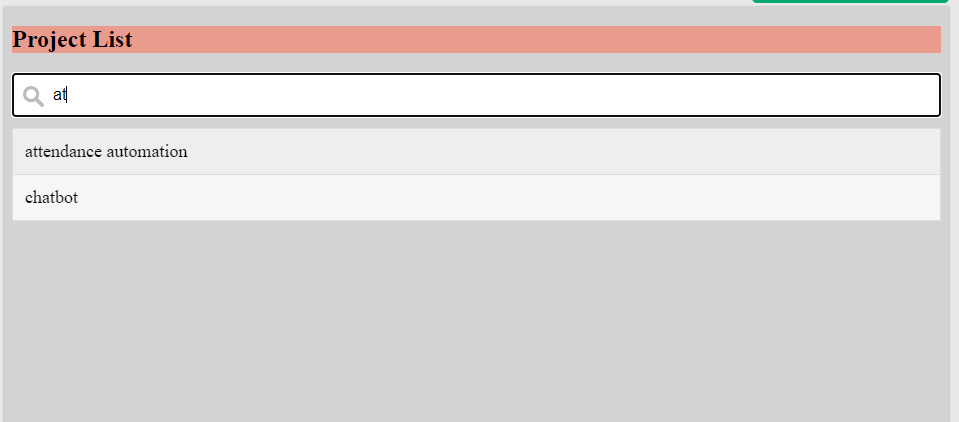
}

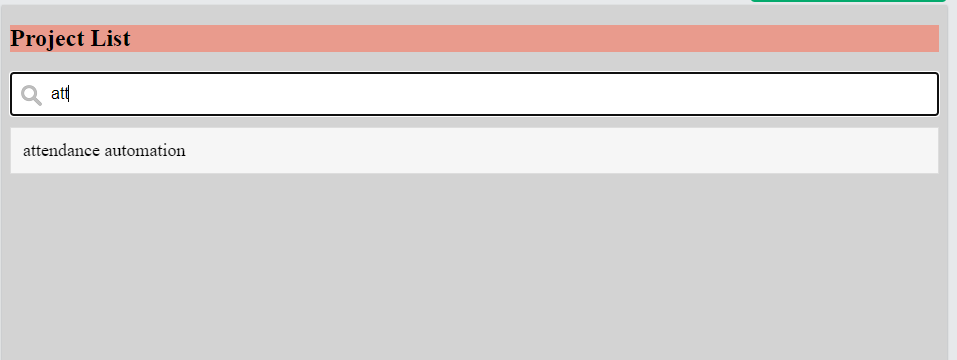
</script>

</body>

</html>







**Ex. No. 6 Date:**

**ANGULARJS – SWITCH AND ANIMATIONS**

**AIM:**

To design a webpage using Angular JS switch and animations.

**ALGORITHM:**

1. Include the AngularJS (load framework) using the script tag.
2. Define an ng-app and ng-model directives for the program.
3. Get the required inputs using the input tag
4. Use ng-click to add onclick functionality to the buttons, and ng-controller to control the data flow.
5. Use ng-hide and ng-show to hide and show elements. (animations)
6. Use CSS to function along with ng-hide for animations.
7. Use ng-switch to switch between layouts.
8. Use above procedure and implement the at least 2 layouts (list and grid).

**TOOLS NEEDED:**

* Notepad – Windows Notepad is a simple text editor and it creates and edit plain text documents and here used for html code.
* Browser – Chrome browser is used here for viewing the web page created using Html.

**CODE:**

<!DOCTYPEhtml>

<html>

<head>

<title>Animations</title>

<style>

div{

transition: all linear 0.5s;background-color: lightblue;height:100px;

}

.ng-hide {height:0;

}

</style>

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

<scriptsrc="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-animate.js"></script>

</head>

<bodyng-app="myApp">

<h1>Angularanimations</h1>

<h2>ClickHere!<inputtype="checkbox"ng-model="myCheck"></h2>

<divng-hide="myCheck"></div>

<script>

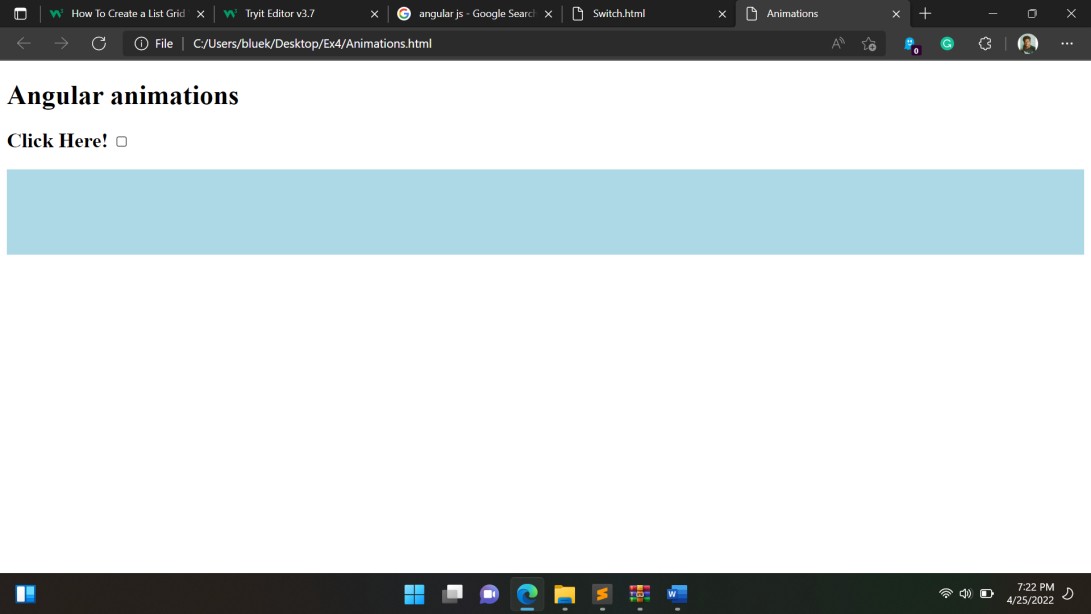
varapp=angular.module('myApp',['ngAnimate']);

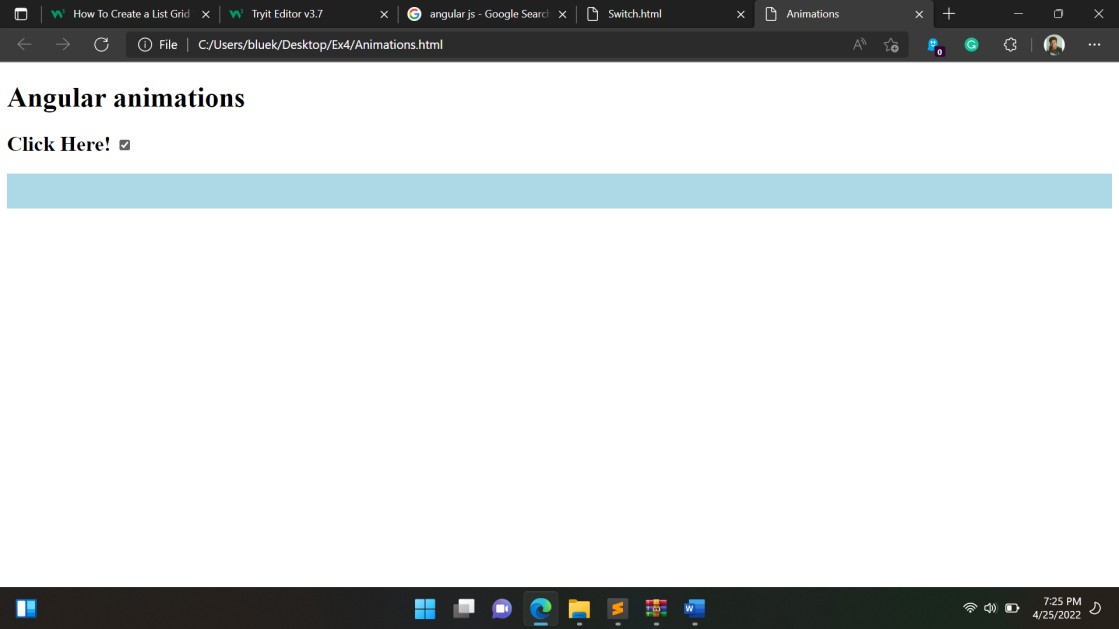
</script>

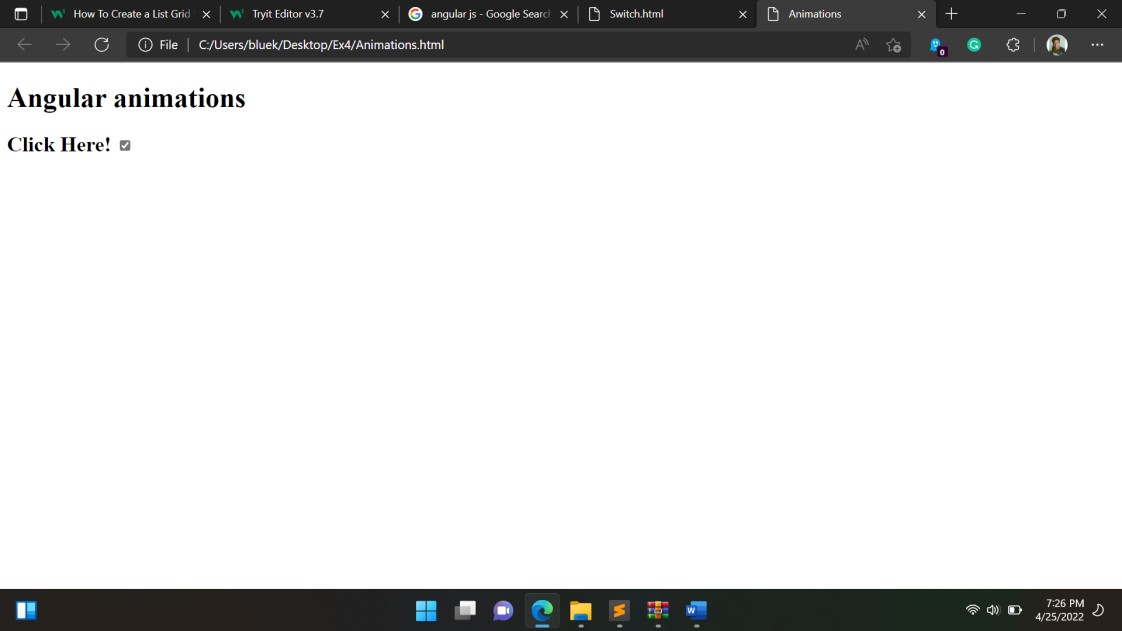
</body>

</html>

# OUTPUT:







**SWITCHINGBETWEENGRIDANDLIST**

# CODE:

<!DOCTYPEhtml>

<html>

<head>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<style>

\*{

box-sizing:border-box;

}

/\*Create twoequalcolumnsthatfloatsnexttoeachother\*/

.column {float: leftwidth:50%;

padding:10px;

}

/\*Clear floatsafterthecolumns \*/

.row:after {content: "";display: table;clear:both;

}

/\*Stylethe buttons\*/

.btn{

border: none;

outline:none;

padding: 12px 16px;background-color:#f1f1f1;cursor: pointer;

}

.btn:hover{

background-color:#ddd;

}

.btn.active{

background-color: #666;color:white;

}

</style>

</head>

<body>

<h2>ListVieworGridView</h2>

<p>Clickonabuttonto choose listview orgridview.</p>

<divid="btnContainer">

<buttonclass="btn"onclick="listView()"><iclass="fafa-bars"></i>List</button>

<buttonclass="btnactive"onclick="gridView()"><iclass="fafa-th-large"></i>Grid</button>

</div>

<br>

<divclass="row">

<divclass="column"style="background-color:#aaa;">

<h2>Html</h2>

<p>TheHyperTextMarkupLanguage orHTMListhestandardmarkuplanguagefordocumentsdesigned to bedisplayed ina webbrowser.</p>

</div>

<divclass="column"style="background-color:#bbb;">

<h2>CSS</h2>

<p>CascadingStyleSheetsisastylesheetlanguageusedfordescribingthepresentationofadocumentwritten inamarkuplanguagesuchas HTML.</p>

</div>

</div>

<divclass="row">

<divclass="column"style="background-color:#ccc;">

<h2>JS</h2>

<p>JavaScript, often abbreviated JS, is a programming language that is one of the core technologies oftheWorldWideWeb, alongsideHTML andCSS.</p>

</div>

<divclass="column"style="background-color:#ddd;">

<h2>AngularJs</h2>

<p>AngularJS was a JavaScript-based open-source front-end web framework for developing single-page applications.</p>

</div>

</div>

<script>

//Gettheelementswithclass="column"

varelements=document.getElementsByClassName("column");

// Declare a loop variablevar i;

// List ViewfunctionlistView(){

for (i = 0; i <elements.length; i++) {elements[i].style.width="100%";

}

}

//GridViewfunctiongridView(){

for (i = 0; i <elements.length; i++) {elements[i].style.width="50%";

}

}

/\* Optional: Add active class to the current button (highlight it) \*/varcontainer=document.getElementById("btnContainer");

varbtns = container.getElementsByClassName("btn");for (var i = 0; i <btns.length; i++) {btns[i].addEventListener("click",function(){

var current = document.getElementsByClassName("active");current[0].className=current[0].className.replace("active","");this.className += "active";

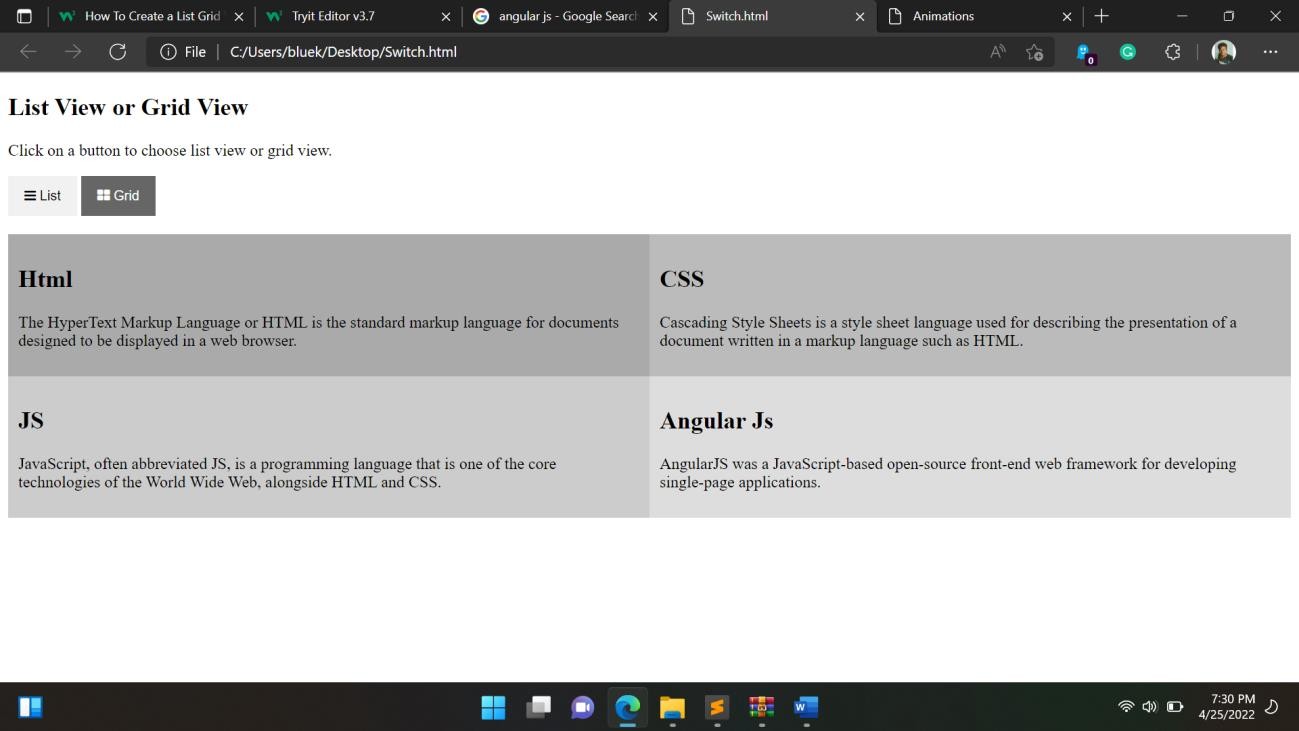
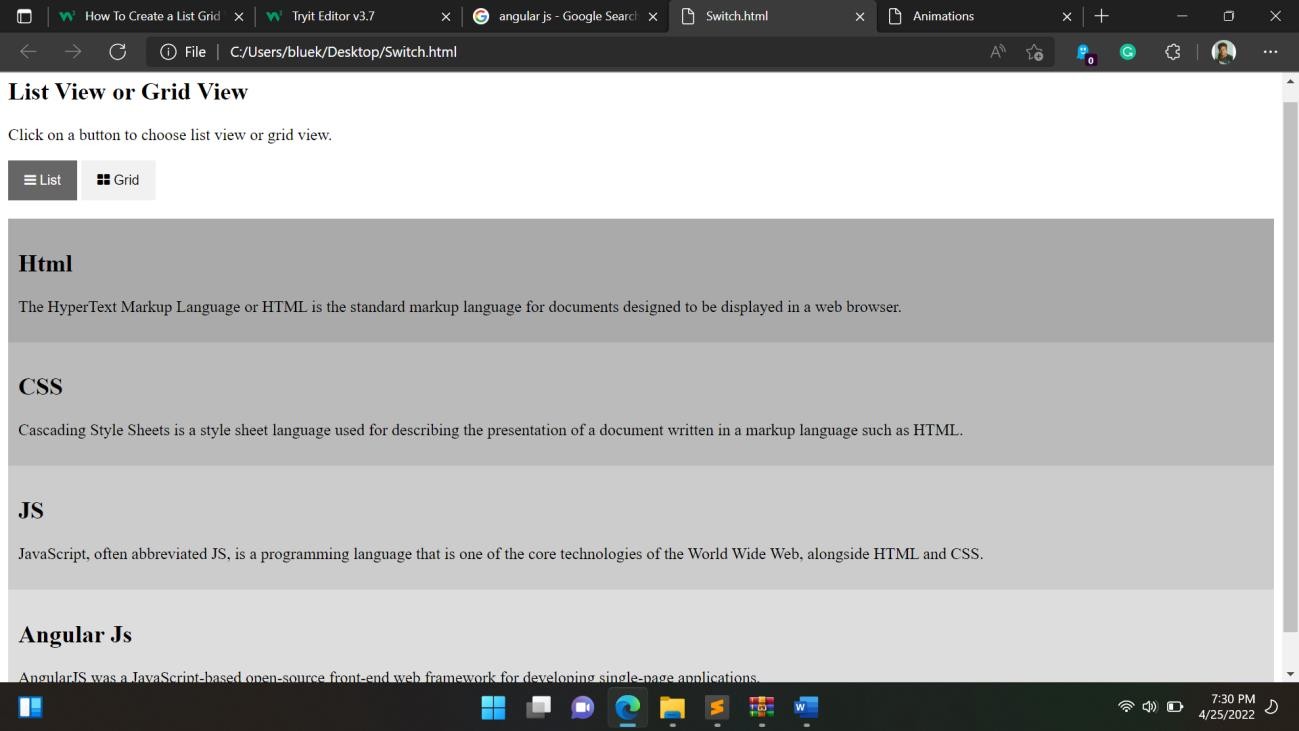
});

}

</script>

</body>

</html>



**RESULT:**

The Angular JS animations and switch has been explored and implemented in html and their outputs were verified.

**Ex. No. 7 Date:**

**FORM WITH REAL-TIME (PHP)**

**AIM:**

To design a real-time webpage using forms and PHP.

**ALGORITHM:**

1. Import angular script.
2. Create the form using theTop of Form tag and add labels and inputs to the same.
3. Set appropriate input types for all parameters.
4. Write an angular function to set patterns or Regular expressions as constraints for each input as required.
5. Install PHP and MySQL database.
6. Create the required database on MySQL with the appropriate database name, username, password, and table.
7. Connect the form to the PHP file using the form action tag (refer the last page for code).
8. Validate the form by calling the angular function after submitting the form using the ngcontroller tag.
9. Create a display file, to show all the inputs received into the database, in a table like manner.
10. Send alerts if there is an error and after successful submission.
11. Display the real-time entries received in the table.

**TOOLS NEEDED:**

* Notepad – Windows Notepad is a simple text editor and it creates and edit plain text documents and here used for html code.
* Browser – Chrome browser is used here for viewing the web page created using Html.

**CODE:**

# Index.php

<!DOCTYPE html>

<html>

<head>

<title>WebslessonTutorial|SubmitFormDatabyusingAngularJSwithValidationusing PHP</title>

<link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"/>

<script[src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>](http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js)

</head>

<body>

<br /><br />

<div class="container" style="width:750px;">

<h3align="center">Submit FormDatabyusingAngularJSwithValidationusingPHP</h3>

<br /><br />

<div ng-app="myapp" ng-controller="formcontroller">

<form name="userForm" ng-submit="insertData()">

<label class="text-success" ng-show="successInsert">{{successInsert}}</label>

<div class="form-group">

<label>First Name <span class="text-danger">\*</span></label>

<input type="text" name="first\_name" ng-model="insert.first\_name"class="form-control" />

<span class="text-danger" ng-show="errorFirstname">{{errorFirstname}}</span>

</div>

<div class="form-group">

<label>Last Name <span class="text-danger">\*</span></label>

<input type="text" name="last\_name" ng-model="insert.last\_name"class="form-control" />

<span class="text-danger" ng-show="errorLastname">{{errorLastname}}</span>

</div>

<br />

<div class="form-group">

<input type="submit" name="insert" class="btnbtn-info" value="Insert"/>

</div>

</form>

</div>

</div>

</body>

</html>

<script>

var application = angular.module("myapp", []);application.controller("formcontroller",function($scope,$http){

$scope.insert = {};

$scope.insertData = function(){

$http({method:"POST",url:"insert.php",data:$scope.insert,

}).success(function(data){if(data.error)

{

$scope.errorFirstname=data.error.first\_name;

$scope.errorLastname=data.error.last\_name;

$scope.successInsert = null;

}

else

{

$scope.insert = null;

$scope.errorFirstname = null;

$scope.errorLastname = null;

$scope.successInsert = data.message;

}

});

}

});

</script>

# Insert.php:

# <?php

//insert.php

$connect = mysqli\_connect("localhost", "root", "", "testing");

$form\_data = json\_decode(file\_get\_contents("php://input"));

$data = array();

$error = array();

if(empty($form\_data->first\_name))

{

$error["first\_name"] = "First Name is required";

}

if(empty($form\_data->last\_name))

{

$error["last\_name"] = "Last Name is required";

}

if(!empty($error))

{

$data["error"] = $error;

}

else

{

$first\_name = mysqli\_real\_escape\_string($connect, $form\_data->first\_name);

$last\_name = mysqli\_real\_escape\_string($connect, $form\_data->last\_name);

$query = "

INSERTINTOtbl\_user(first\_name,last\_name)VALUES('$first\_name','$last\_name')";

if(mysqli\_query($connect, $query))

{

$data["message"] = "Data Inserted...";

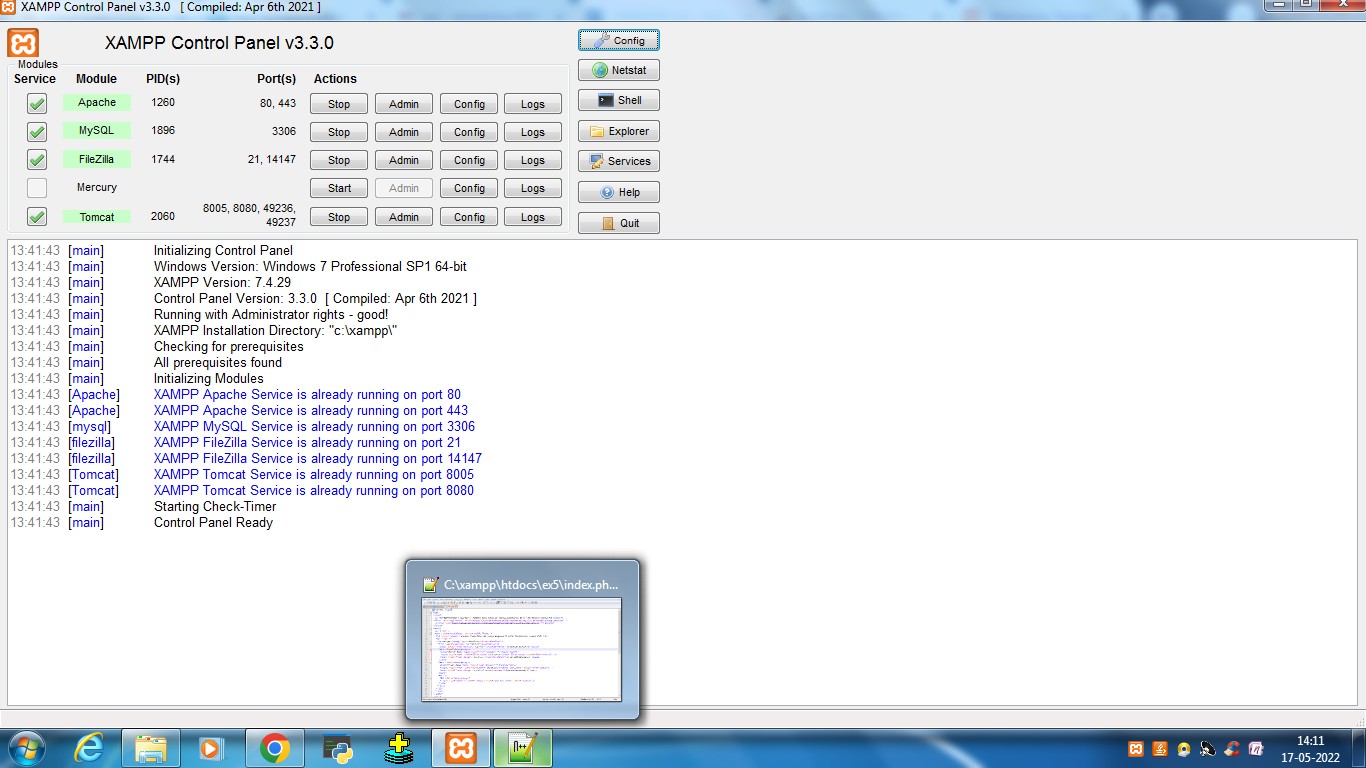
}

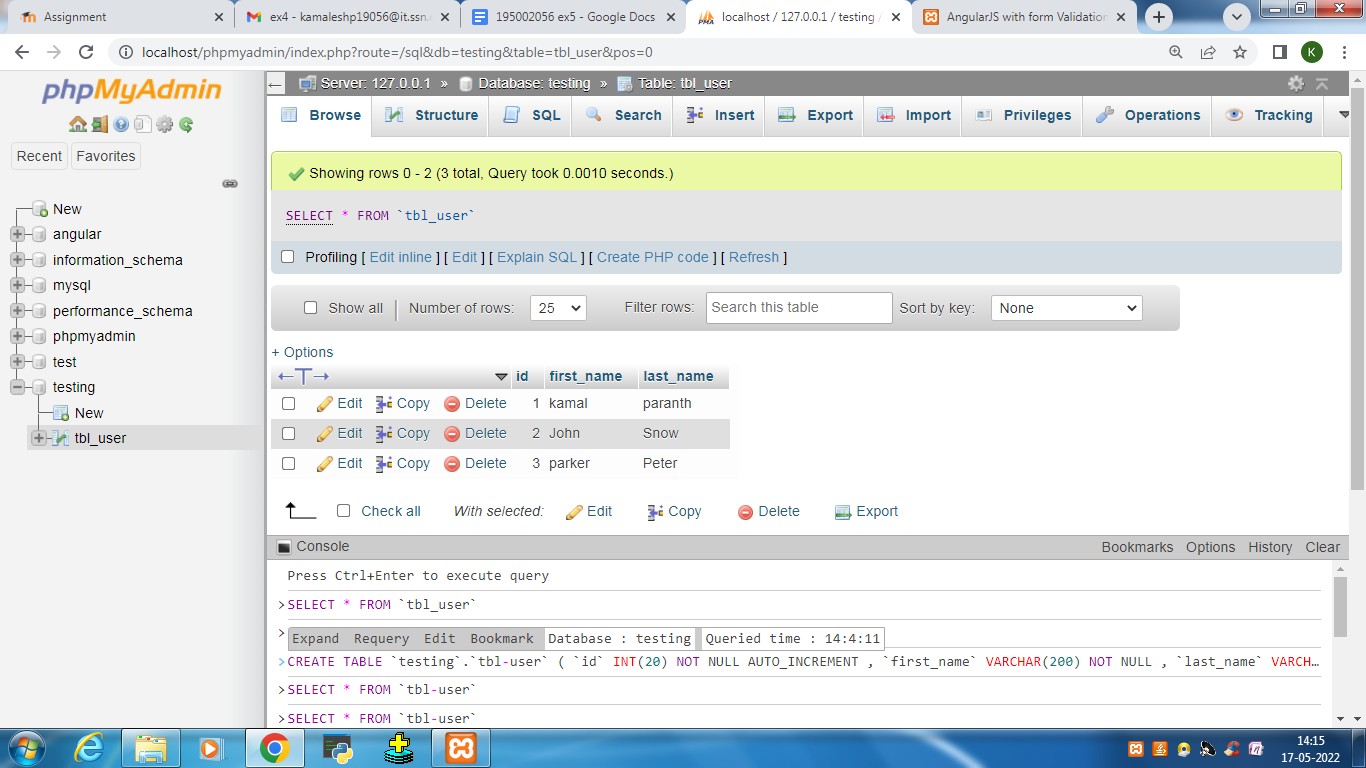
}

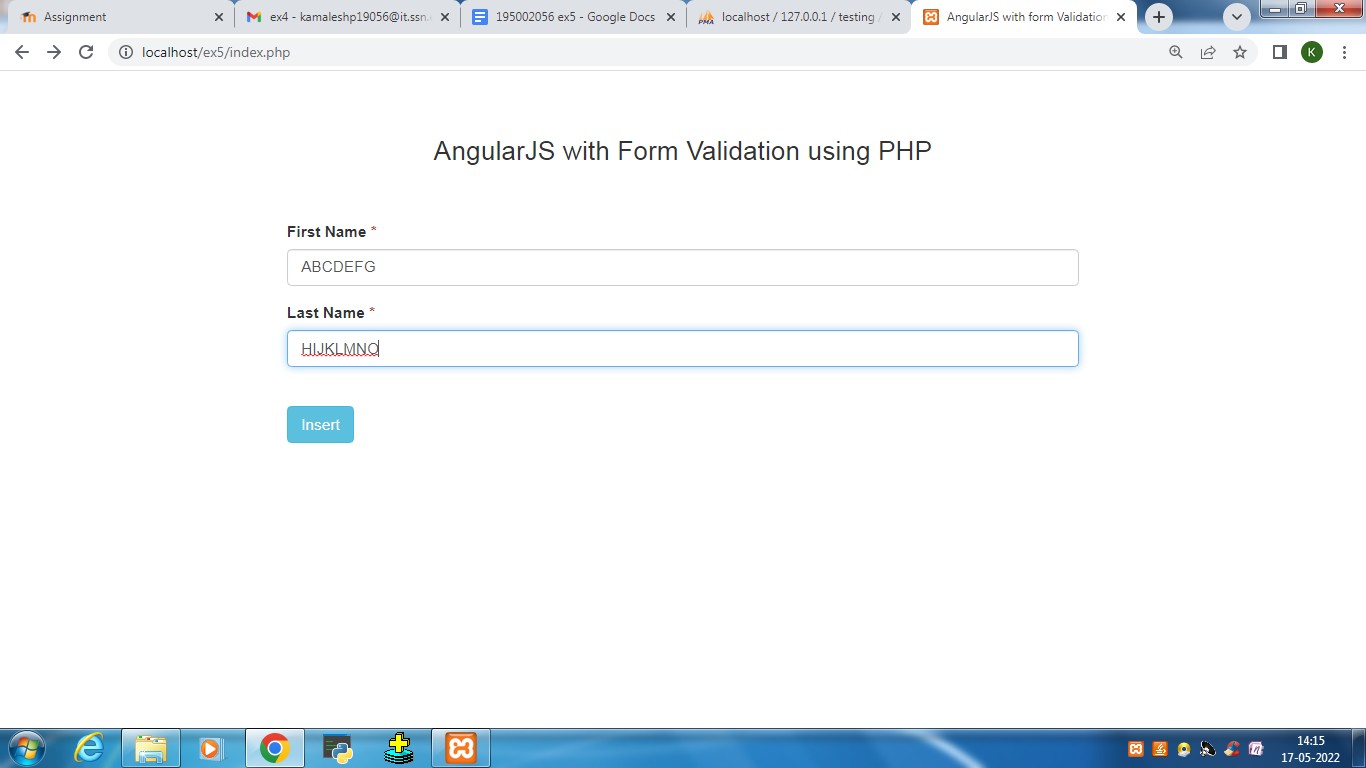
echojson\_encode($data);

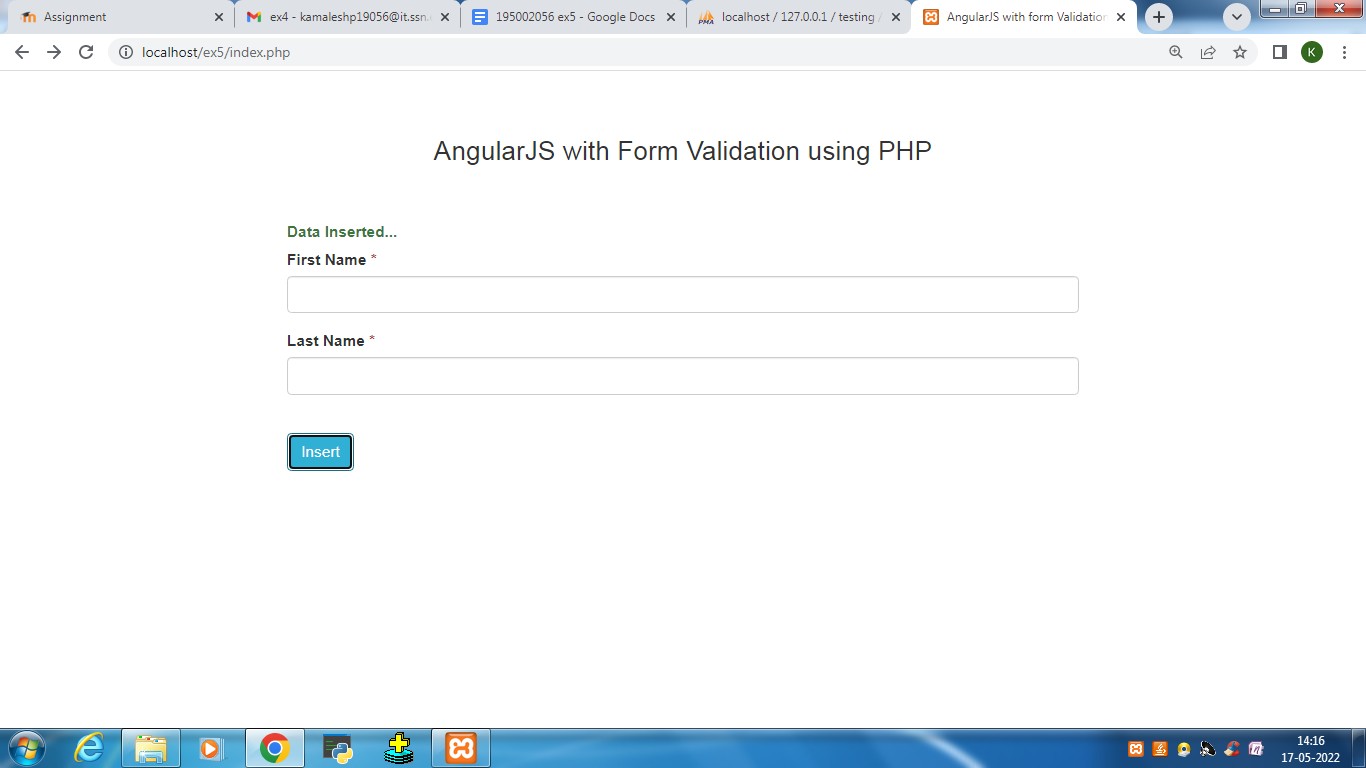
?>

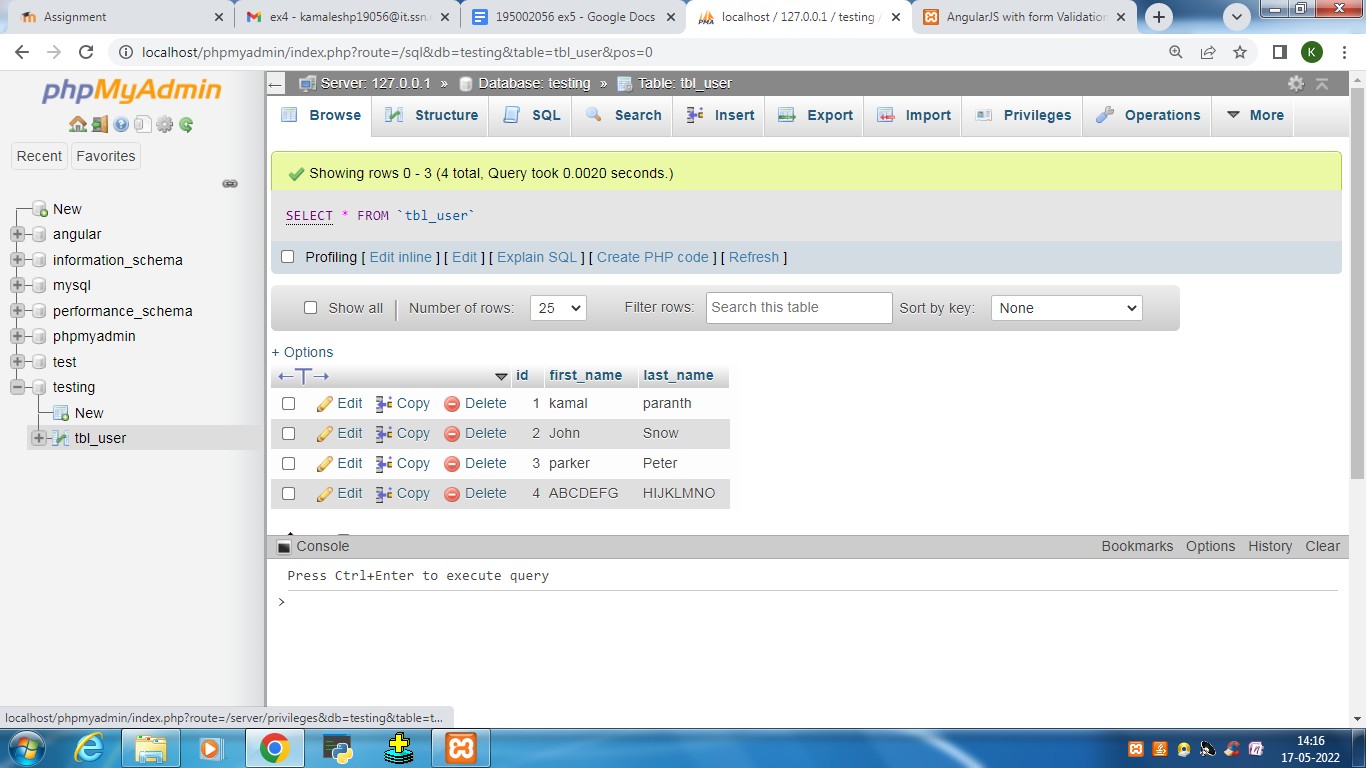
# OUTPUT:

Started apache and mysql:









**Angular JS – Navigation Menu**

**Ex. No. 8**

**Objectives:**

* Design a Navigation Menu for the website using angular JS.

**What to do by the students (Implementation)**

* Include Angular JS and Angular route links using script tag.
* Define the separate pages under type=”text/ng-template” using script
* Use ng-route to route to the respective pages based on the value of $routeprovider.
* Defining separate controllers for each application.
* In this application, users can navigate to the respective webpage using the navigation menu at the top. Calculator and To Do List are provided as samples to navigate.

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<style>

.center {

display: block;

margin-left: auto;

margin-right: auto;

width: 50%;

}

body {

margin: 0;

font-family: Arial, Helvetica, sans-serif;

}

.topnav {

overflow: hidden;

background-color: #333;

}

.topnav a {

float: left;

display: block;

color: #f2f2f2;

text-align: center;

padding: 14px 16px;

text-decoration: none;

font-size: 17px;

}

.topnav a:hover {

background-color: #ddd;

color: black;

}

.topnav a.active {

background-color: #04AA6D;

color: white;

}

.topnav .icon {

display: none;

}

@media screen and (max-width: 600px) {

.topnav a:not(:first-child) {display: none;}

.topnav a.icon {

float: right;

display: block;

}

}

@media screen and (max-width: 600px) {

.topnav.responsive {position: relative;}

.topnav.responsive .icon {

position: absolute;

right: 0;

top: 0;

}

.topnav.responsive a {

float: none;

display: block;

text-align: left;

}

}

</style>

</head>

<body>

<div class="topnav" id="myTopnav">

<a href="#home" class="active">Home</a>

<a href="#news">Add details</a>

<a href="#contact">Alter</a>

<a href="#about">Register project</a>

<a href="javascript:void(0);" class="icon" onclick="myFunction()">

<i class="fa fa-bars"></i>

</a>

</div>

<div style="padding-left:16px">

</div>

<script>

function myFunction() {

var x = document.getElementById("myTopnav");

if (x.className === "topnav") {

x.className += " responsive";

} else {

x.className = "topnav";

}

}

</script>

<img src="E:\static\uploadimg.jpg " class="center">

</body>

</html>

