# ATME COLLEGE OF ENGINEERING

**13th KM Stone, Bannur Road, Mysore - 560 028**


A T M E
College of Engineering

## DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

## (DATA SCIENCE)

## (ACADEMIC YEAR 2023-24)

# LABORATORY MANUAL

**SUBJECT: ANGULAR JS LABORATORY MANUAL**

**SUB CODE: 21CSL581**

**SEMESTER: V- 2021 CBCS Scheme**

| Composed by | Verified by | Approved by |
|---|---|---|
| Mrs. ASHA L | Mrs. KHATEEJA AMBAREEN | Dr. J V GORABAL |
| PROGRAMMER | FACULTY CO-ORDINATOR | HOD, CSE-DS |

# INSTITUTIONAL MISSION AND VISION

## Objectives

- To provide quality education and groom top-notch professionals, entrepreneurs and leaders for different fields of engineering, technology and management.
- To open a Training-R & D-Design-Consultancy cell in each department, gradually introduce doctoral and postdoctoral programs, encourage basic & applied research in areas of social relevance, and develop the institute as a center of excellence.
- To develop academic, professional and financial alliances with the industry as well as the academia at national and transnational levels
- To develop academic, professional and financial alliances with the industry as well as the academia at national and transnational levels.
- To cultivate strong community relationships and involve the students and the staff in local community service.
- To constantly enhance the value of the educational inputs with the participation of students, faculty, parents and industry.

## Vision

- Development of academically excellent, culturally vibrant, socially responsible and globally competent human resources.

## Mission

- To keep pace with advancements in knowledge and make the students competitive and capable at the global level.
- To create an environment for the students to acquire the right physical, intellectual, emotional and moral foundations and shine as torch bearers of tomorrow’s society.

- To strive to attain ever-higher benchmarks of educational excellence.

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND ENGINEERING**

**(DATA SCIENCE &ENGINEERING)**

## Vision of The Department

- To impart technical education in the field of data science of excellent quality with a high level of professional competence, social responsibility, and global awareness among the students

## Mission

- To impart technical education that is up to date, relevant and makes students competitive and employable at global level
- To provide technical education with a high sense of discipline, social relevance in an intellectually, ethically and socially challenging environment for better tomorrow
- Educate to the global standards with a benchmark of excellence and to kindle the spirit of innovation.

## Program Outcomes(PO)

- **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

- **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

- **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental
  considerations.
- **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

- **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

- **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

- **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

- **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

- **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and

write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

- **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## Program Specific Outcomes (PSOs)

- PSO1: Develop relevant programming skills to become a successful data scientist
- PSO2: Apply data science concepts and algorithms to solve real world problems of the society
- PSO3: Apply data science techniques in the various domains like agriculture, education healthcare for better society

## Program Educational Objectives (PEOs):

**PEO1**: Develop cutting-edge skills in data science and its related technologies, such as machine learning, predictive analytic, and data engineering.

**PEO2**: Design and develop data-driven solutions to real-world problems in a business, research, or social environment.

**PEO3**: Apply data engineering and data visualization techniques to discover, investigate, and interpret data.

**PEO4:** Demonstrate ethical and responsible data practices in problem solving

**PEO5**: Integrate fields within computer science, optimization, and statistics to develop better solutions

| ANGULAR JS | | | |
|---|---|---|---|
| Course Code | **21CSL581** | CIE Marks | 50 |
| Teaching Hours/Week (L:T:P: S) | 0:0:2:0 | SEE Marks | 50 |
| Credits | 01 | Total marks | 100 |
| Examination type (SEE) | PRACTICAL | | |

**Course objectives:**

- To learn the basics of Angular JS framework.
- To understand the Angular JS Modules, Forms, inputs, expression, data bindings and Filters
- To gain experience of modern tool usage (VS Code, Atom or any other] in developing Web applications

| Sl.NO | Experiments |
|---|---|
| 1 | Develop Angular JS program that allows user to input their first name and last name and display their full name. **Note**: The default values for first name and last name may be included in the program. |
| 2 | Develop an Angular JS application that displays a list of shopping items. Allow users to add and remove items from the list using directives and controllers. **Note**: The default values of items may be included in the program. |
| 3 | Develop a simple Angular JS calculator application that can perform basic mathematical operations (addition, subtraction, multiplication, division) based on user input. |
| 4 | Write an Angular JS application that can calculate factorial and compute square based on given user input. |
| 5 | Develop AngularJS application that displays a details of students and their CGPA. Allow users to read the number of students and display the count. **Note**: Student details may be included in the program. |
| 6 | Develop an AngularJS program to create a simple to-do list application. Allow users to add, edit, and delete tasks. **Note**: The default values for tasks may be included in the program. |
| 7 | Write an AngularJS program to create a simple CRUD application (Create, Read, Update, and Delete) for managing users. |
| 8 | Develop AngularJS program to create a login form, with validation for the username and password fields. |
| 9 | Create an AngularJS application that displays a list of employees and their salaries. Allow users to search for employees by name and salary. **Note**: Employee details may be included in the program. |
| 10 | Create AngularJS application that allows users to maintain a collection of items. The application should display the current total number of items, and this count should automatically update as items are added or removed. Users should be able to add items to the collection and remove them as needed.<br>**Note**: The default values for items may be included in the program. |
| 11 | Create AngularJS application to convert student details to Uppercase using angular filters. **Note**:<br>The default details of students may be included in the program. |
| 12 | Create an AngularJS application that displays the date by using date filter parameters |

# Introduction

In 2009, Adam Abrons and Misko Hevery created AngularJS for the first time. However, it is now maintained by Google. Angular is entirely a JavaScript framework that is used for building dynamic and interactive applications. It is one of the popular front-end frameworks that is predominantly used for creating SPAs - Single Page Web Applications. Today, Angular is one such framework that has owned a remarkable position in the Web Development industry. Also, it is persistently evolving with the emerging needs in the industry and has also offered better solutions and support for developing Web applications for the front-end developers. Here in this Angular tutorial, you will have access to know about Angular and its role in the front-end development industry.

Intended Audiences: This AngularJS Tutorial is tailored for aspirants who want to kick-start their career in the Web Development domain and also for the Software Professionals who wish to become familiar with AngularJS concepts.

Prerequisites: There are no requisites for learning the Angular Concepts. However, knowing HTML, Javascript, AJAX, and CSS will help you grasp the ideas more quickly.

Reasons to learn AngularJS

AngularJS is the most popular open-source and front-end Web Application framework. It is a structural framework that is used for developing dynamic web applications. It permits you to make use of the HTML as the fundamental template language and allows you to extend HTML syntax for expressing the application components in a precise manner. It has the feature of dependency injection and data binding and it eliminates all the code that you are required to write. Also, all this takes place within the browser and thus makes it the best option for the server technology.

Here in the AngularJS Tutorial session, you will get to know about the features, advantages, and disadvantages of the AngularJS framework.

**Common Features of the AngularJS**

The common features of AngularJS are listed below:

- It is an efficient framework that could be developed using Rich Internet Applications.

- AngularJS offers the developers a wide range of options for writing client-side applications with the support of Javascript and it offers the clean Model View Controller.
- The Applications that are written in AngularJS are compatible with cross-browsers. AngularJS is capable of handling all the JavaScript code that is perfect for all browsers.
- AngularJS is completely free and it could be used by thousands of developers across the world. It is released under the Apache License 2.0.
- On the whole, AngularJS is the framework that is used for developing large-scale and high performance, and easy-to-handle web applications.

**Core Features**

Given below are the important features of the AngularJS

**Scope** - These are the set of objects that indicates the model. They usually act as a glue between the view and controller.

**Data Binding** - It is the automatic synchronization of data between view components and models.

**Services** - The AngularJS comes with numerous built-in services like $http for making the HTTP requests. These are the singleton objects that are instantiated only once in the app.

**Controller** - This is the list of JavaScript functions that are bound for a specific scope.

**Directives** - It is the markers that are found on the DOM elements namely CSS, attributes, and elements. It can be used for building custom HTML tags which serve as the new custom widgets like AngularJS has built-in directives like ngModel, and ngBind.

**Filters** - This is used for choosing the subset of all the items from the array and it returns to as a new array.

**Routing** - It applies the concept of switching the views.

**Templates** - It is mainly rendered to view the information right from the model and controller. It can be used as a single file or multiple views on one specific page using the partials.

**Deep Linking** - Deep Linking permits you to encode the specific state of the application on the URL so it could be bookmarked easily. Also, the applications could be restored right from the URL to the previous state.

**Model View Whatever** - It is also called MVW. It is the design pattern that is used for separating the application into different parts and it is called Model, View, and Controller. Each one of them has unique responsibilities. Further, AngularJS will not implement the MVC in the traditional aspects, rather it is something that is similar to the MVVM. The Angular JS is the team that denotes the Model View Whatever.

**Dependency Injection** - The AngularJS has numerous built-in dependency injections and subsystems that aids the developers to build and understand the test and applications easier. The AngularJS Training in Chennai at FITA Academy renders a holistic training of the AngularJS concepts and their applications clearly to the students.

**Angular Concepts**

To become more familiar with Angular, you must have a clear idea of the AngularJS Basics. See-through the illustration that is given below, we will further discuss these topics elaborately in the later session of this AngularJS Tutorial for beginners.

**Merits of AngularJS**

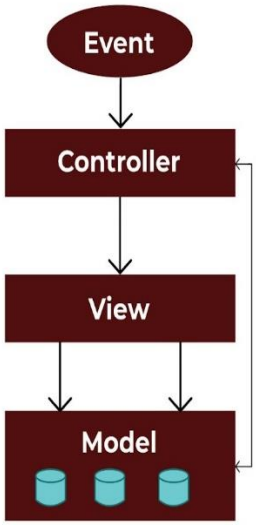
The Benefits of using AngularJS are:

- It supports you in building Single Page Applications clearly and easily.
- It offers data binding capacity for HTML. Hence, it offers its users a responsive and opulent experience.
- The AngularJS codes are highly unit testable.
- AngularJS bestows its users with reusable components.
- AngularJS makes use of the dependency injections and it utilizes the separation of concerns also.
- With the support of AngularJS, the developers could easily reach more functionality with lesser code.

- With AngularJS, the views are simple HTML pages and the controllers are written using JavaScript for performing the business processing.
- Keeping all these things aside, AngularJS applications are capable of running on all the important browsers and operating platforms like Android and iOS, tablets, and desktops.

**Angular MVC Architecture**

Model View Controller is commonly called MVC. It is the software design pattern that is used in building web applications. It is highly popular because it can separate the application logic from a UI layer and further lends its support in the concerns of the separation as well.




The above is the structure of the MVC, and it is made of three different parts, and they are:

**Model**: It is important to handle the application data. It can respond to the requests right from the view till it receives the instruction from a controller for updating itself.

**View**: It is more important for showcasing all the data or only a specific part of data to its users. Also, it specifies all the data is in a specific format that could be triggered by a controller's decision for presenting all the data. It is a script-based template system, and they are PHP, ASP, JSP, and these could be integrated easily with AJAX technology.

**Controlle**r: It is more important for controlling the relation between the views and models. It easily responds to the user input and executes the interactions on data model objects. Also, the controller receives the input, validates them, and performs all the operations of a business that alters the state of a data model.

## Sample Programs are given at the end for the students to practice

**Since Internet is not allowed during the exams it is not possible to use the angular.min.js from online. Hence, instead of using the online script, please download the offline angular.min.js from the link below**

**https://drive.google.com/file/d/18WE3NQA7p2isaLzSnMDJYXLWfWiW3NvT/view?usp=drive_link**

**and paste it inside the same directory from where the program is being run and use the below script to include the angular.min.js.**

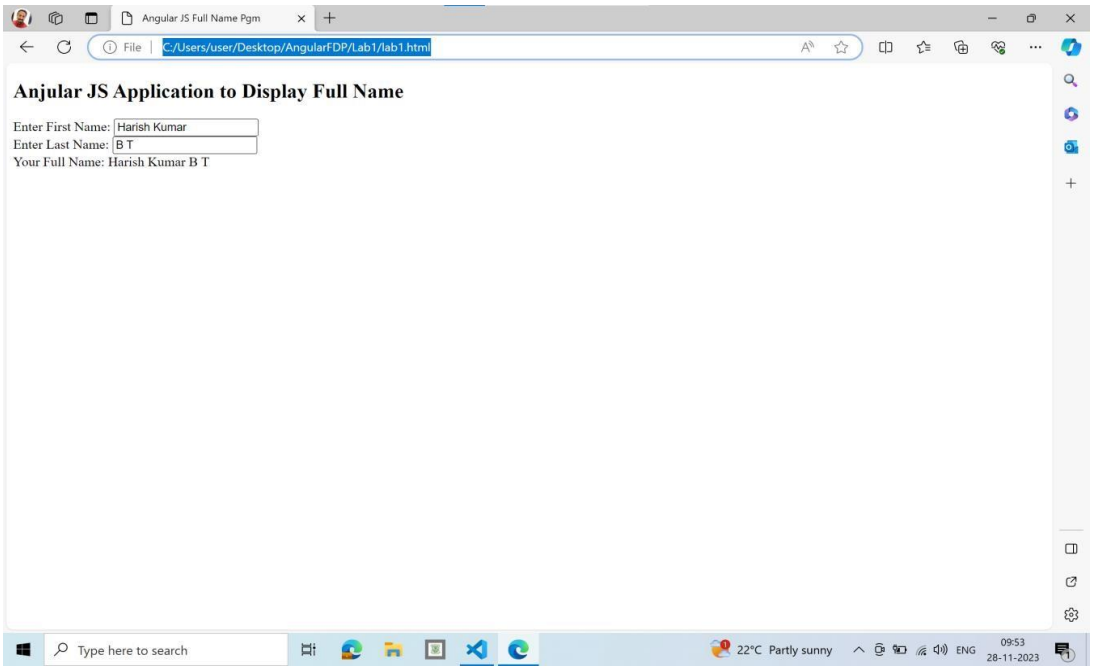
**<script type="text/javascript" src="angular.min.js"></script>**

**1. Develop Angular JS program that allows user to input their first name and last name and display their full name. Note: The default values for first name and last name may be included in the program.**

```
<!DOCTYPE html>
<html>
<title>
   Angular JS Full Name Pgm
</title>
<head>
   <script type="text/javascript"
   src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
</script> <script>     var app=angular.module("myApp",[]);
app.controller("myCntrl",function($scope){
      $scope.firstName="Harish Kumar"
      $scope.lastName="B T"
   });
</script>
</head>
```

```
<body ng-app="myApp">
   <h2>Anjular JS Application to Display Full Name</h2>
   <div ng-controller="myCntrl">
      Enter First Name: <input type="text" ng-model="firstName"><br/>
      Enter Last Name: <input type="text" ng-model="lastName"><br/>
      Your Full Name: {{firstName +" "+ lastName}}
   </div>
</body> </html>
```

**Output:**



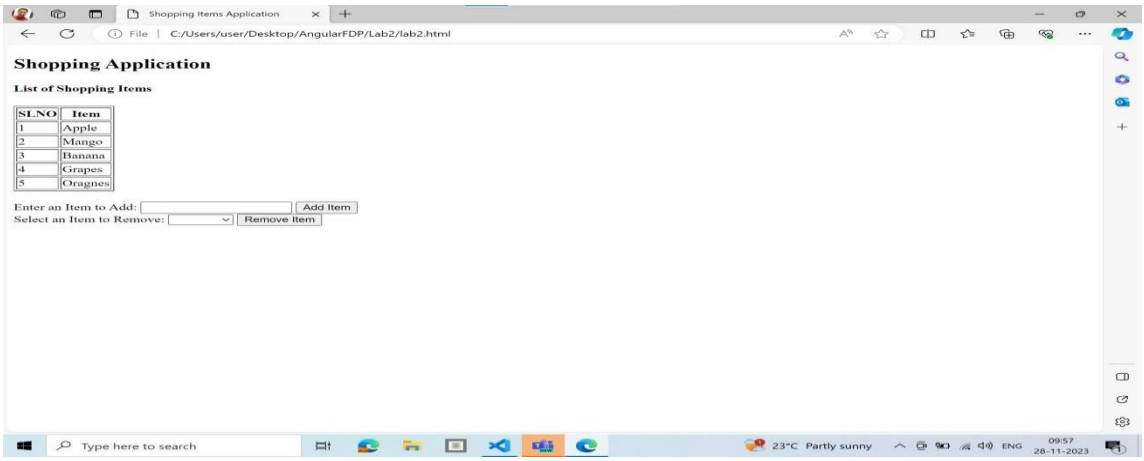

**2. Develop an Angular JS application that displays a list of shopping items. Allow users to add and remove items from the list using directives and controllers. Note: The default values of items may be included in the program.**

```
<!DOCTYPE html>
<html>
<title>
   Shopping Items Application
```

```
</title>
<head>
   <script type="text/javascript"
   src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
</script> <script>     var app=angular.module("myApp",[]);
app.controller("myCntrl",function($scope){
   $scope.shoppingItems=['Apple','Mango','Banana','Grapes']
   $scope.addItem=function(){
 if($scope.newItem && $scope.shoppingItems.indexOf($scope.newItem)==-1)
         {
            $scope.shoppingItems.push($scope.newItem)
            $scope.newItem=""
         }
else
         {
            if($scope.newItem)
            alert("This item is already there in the shopping list")
else
           alert("Please enter an item to add")
         }
      }

      $scope.removeItem=function(){
//console.log("function called")
         if($scope.shoppingItems.indexOf($scope.selectItem)==-1)
         {
            alert("Please select an item to remove")
         }
else{
            var index=$scope.shoppingItems.indexOf($scope.selectItem)
            $scope.shoppingItems.splice(index,1)
            $scope.selectItem=""
         }
      }
   });
</script>


</head>
<body ng-app="myApp">
<div ng-controller="myCntrl">
   <h2>Shopping Application</h2>
   <h4>List of Shopping Items</h4>
   <table border="1">
```

```
        <tr>
          <th>SLNO</th>
          <th>Item</th>
        </tr>
        <tr ng-repeat="items in shoppingItems">
          <td>{{$index+1}}</td>
          <td>{{items}}</td>
        </tr>
      </table>
      <br/>
      <div>
      Enter an Item to Add: <input type="text" ng-model="newItem">
      <button ng-click="addItem()">Add Item</button>
    </div>

    <div>
     Select an Item to Remove:
     <select ng-model="selectItem" ng-options="item for item in shoppingItems"></select>
      <button ng-click="removeItem()">Remove Item</button>
    </div>
    </div>
    </body>
    </html>
```

**Output:**

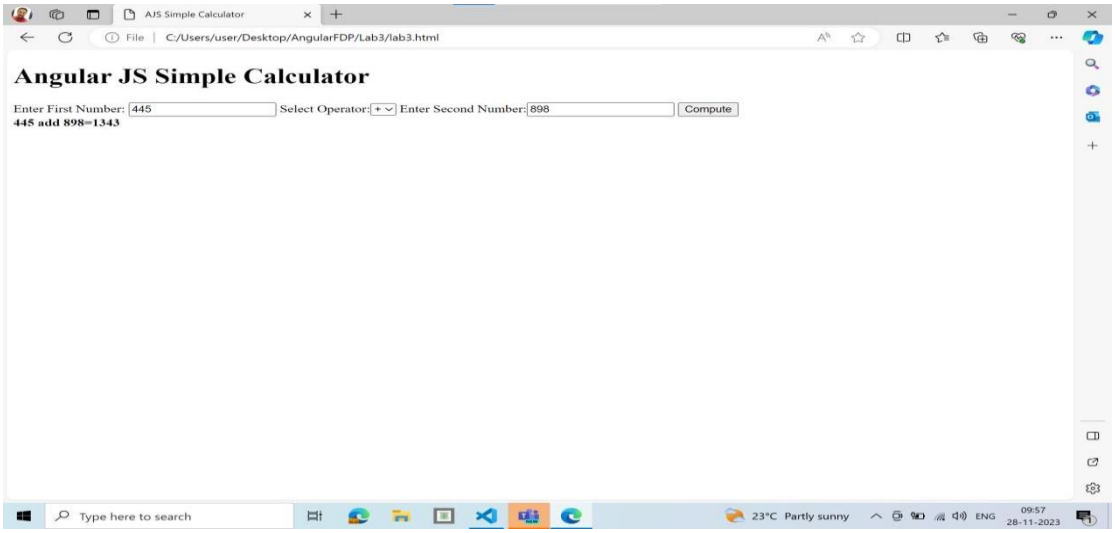



**3. Develop a simple Angular JS calculator application that can perform basic mathematical operations (addition, subtraction, multiplication, division) based on user input.**

```
<!DOCTYPE html>
<html>
<title>
   AJS Simple Calculator
</title>
<head>
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
</script> <script> var
app=angular.module("calcApp",[]);
app.controller("calcCntrl",function($scope)
 {
       $scope.num1=0
       $scope.num2=0
       $scope.result=0
       $scope.operator="add"

       $scope.compute=function(){
                switch($scope.operator){              case 'add':
$scope.result=$scope.num1 + $scope.num2
                                          break              case 'sub':
$scope.result=$scope.num1 - $scope.num2

break
             case 'mul': $scope.result=$scope.num1 * $scope.num2
break              case 'div': if($scope.num2==0){
                   alert("Divide by zero error")
                   }                        else{
                     $scope.result=$scope.num1/$scope.num2
                   }}}
 });

</script>
</head>
<body ng-app="calcApp">
   <h1>Angular JS Simple Calculator</h1>

   <div ng-controller="calcCntrl">


          Enter First Number: <input type="number" ng-model="num1">
```

```
Select Operator:<select ng-model="operator">
  <option value="add">+</option>
  <option value="sub">-</option>
  <option value="mul">*</option>
  <option value="div">/</option>
</select>
Enter Second Number:<input type="number" ng-model="num2">
<button ng-click="compute()">Compute</button>
<br/>
<b>{{num1 + " "+operator+ " "+ num2+ "="+result}}</b>
</div>
</body>
</html>
```

**Output:**



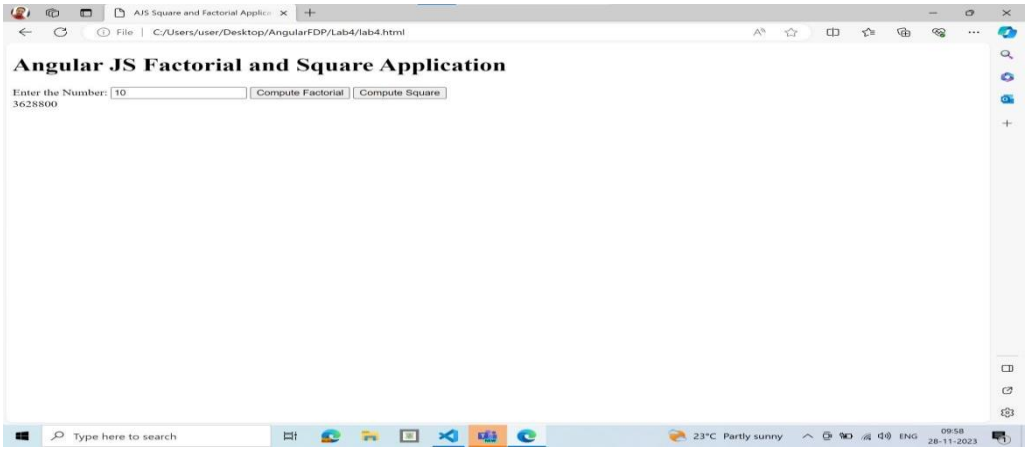

**4. Write an Angular JS application that can calculate factorial and compute square based on given user input.**

```
<!DOCTYPE html>
<html>
<title>
  AJS Square and Factorial Application
</title>
<head>
```

```
  <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
</script>  <script>     var
app=angular.module("mySqFct", []);
   app.controller("mySqFctCntrl", function($scope){
      $scope.num=0
      $scope.result

      $scope.factorial=function()
      {           if($scope.num==0)
         {
            $scope.result=1
         }           else{
            $scope.fact=1              for(var i=$scope.num;
i>=1; i--)
            {
                  $scope.fact=$scope.fact*i
            }
            $scope.result=$scope.fact
         }
      }
      $scope.square=function(){
         $scope.result=$scope.num*$scope.num
      }    });
</script>
</head>
<body ng-app="mySqFct">
<h1> Angular JS Factorial and Square Application</h1>
<div ng-controller="mySqFctCntrl">
    Enter the Number: <input type="number" ng-model="num">
      <button ng-click="factorial()">Compute Factorial</button>

      <button ng-click="square()">Compute Square</button>
```

```
<br/>

{{result}}

</div>
</body>
</html>
```

**Output:**



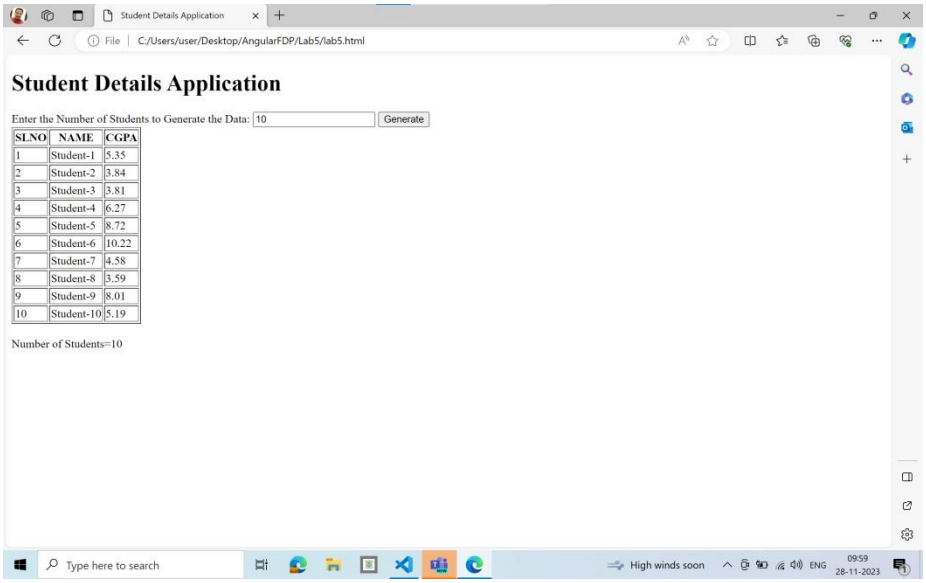

**5. Develop AngularJS application that displays a details of students and their CGPA. Allow users to read the number of students and display the count. Note: Student details may be included in the program.**

```
<!DOCTYPE html>
<html>
  <title>Student Details Application</title>
  <head>
  <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
  </script>  <script>  var
app=angular.module("studDetailsApp",[]);
```

```
app.controller("studDetailsAppCntrl",function($scope){
  $scope.studData=[]

  $scope.generateData=function()
  {
    $scope.studData=[]
    for(var i=1;i<=$scope.num;i++)
    {          var stud={
        "SLNO":i,
        "NAME":'Student-'+i,
        "CGPA":(Math.random()*10+1).toFixed(2)
      }
      $scope.studData.push(stud)
    }
  }
});
</script>
</head>
<body ng-app="studDetailsApp">
  <h1>Student Details Application</h1>
  <div ng-controller="studDetailsAppCntrl">
    Enter the Number of Students to Generate the Data:
    <input type="number" ng-model="num">
    <button ng-click="generateData()">Generate</button>
    <br/>
    <table border="1" ng-show="studData.length>0">
      <tr>
        <th>SLNO</th>
        <th>NAME</th>
<th>CGPA</th>
      </tr>
      <tr ng-repeat="student in studData">
```

```
                <td>{{student.SLNO}}</td>


                        <td>{{student.NAME}}</td>
                    <td>{{student.CGPA}}</td>
                </tr>
            </table>
            <br/>
            Number of Students={{studData.length}}
        </div>
    </body>
</html>
```

**Output:**



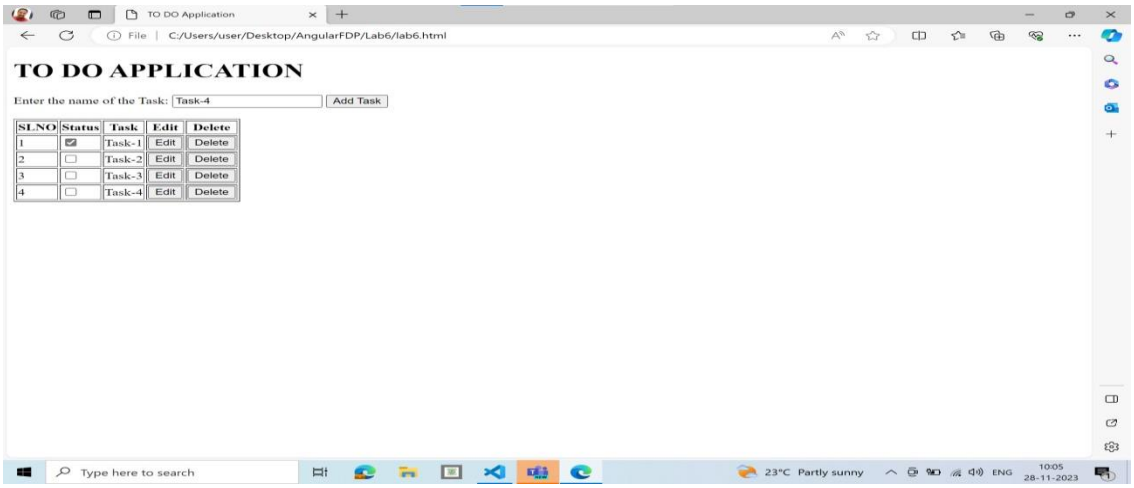

**6. Develop an AngularJS program to create a simple to-do list application. Allow users to add, edit, and delete tasks. Note: The default values for tasks may be included in the program.**

```
<!DOCTYPE html>
<html>
    <title>TO DO Application</title>
    <head>
```

```
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"> </script>

<script> var
app=angular.module("toDoApp",[]);
app.controller("toDoAppCntrl",function($scope){ $scope.tasks=[
{'TITLE':'Task-1','COMPLETED':true,'EDITING':false},
{'TITLE':'Task-2','COMPLETED':false,'EDITING':false},
{'TITLE':'Task-3','COMPLETED':false,'EDITING':false}
]

$scope.addTask=function(){ if($scope.newTask)
{ var
t={
'TITLE':$scope.newTask,
'COMPLETED':false,
'EDITING':false
}

$scope.tasks.push(t)
} else{ alert("Please enter the task
to add")
}
}

$scope.editTask=function(task)
{
task.EDITING=true
}

$scope.turnOffEditing=function(task){ task.EDITING=false
}

$scope.deleteTask=function(task)

{

var index=$scope.tasks.indexOf(task)
$scope.tasks.splice(index,1)
}

});
```

```
    </script>
  </head>

  <body ng-app="toDoApp">
    <h1>TO DO APPLICATION</h1>
    <div ng-controller="toDoAppCntrl">
      Enter the name of the Task:
      <input type="text" ng-model="newTask">
      <button ng-click="addTask()">Add Task</button>
      <br/>
      <br/>
      <table border="1">
        <tr>
          <th>SLNO</th>
          <th>Status</th>
          <th>Task</th>
          <th>Edit</th>
          <th>Delete</th>
        </tr>
        <tr ng-repeat="task in tasks">
          <td>{{$index+1}}</td>
          <td>
          <input type="checkbox" ng-model="task.COMPLETED">
          </td>
          <td>
      <span ng-show="!task.EDITING">{{task.TITLE}}</span> <input
            type="text" ng-show="task.EDITING"
            ng-model="task.TITLE" ng-    blur="turnOffEditing(task)">
          </td>
          <td>
            <button ng-click="editTask(task)">Edit</button>
          </td>
          <td>
            <button ng-click="deleteTask(task)">Delete</button>
          </td>
        </tr>
      </table>
      </div>
    </body>
  </html>
```

**Output:**



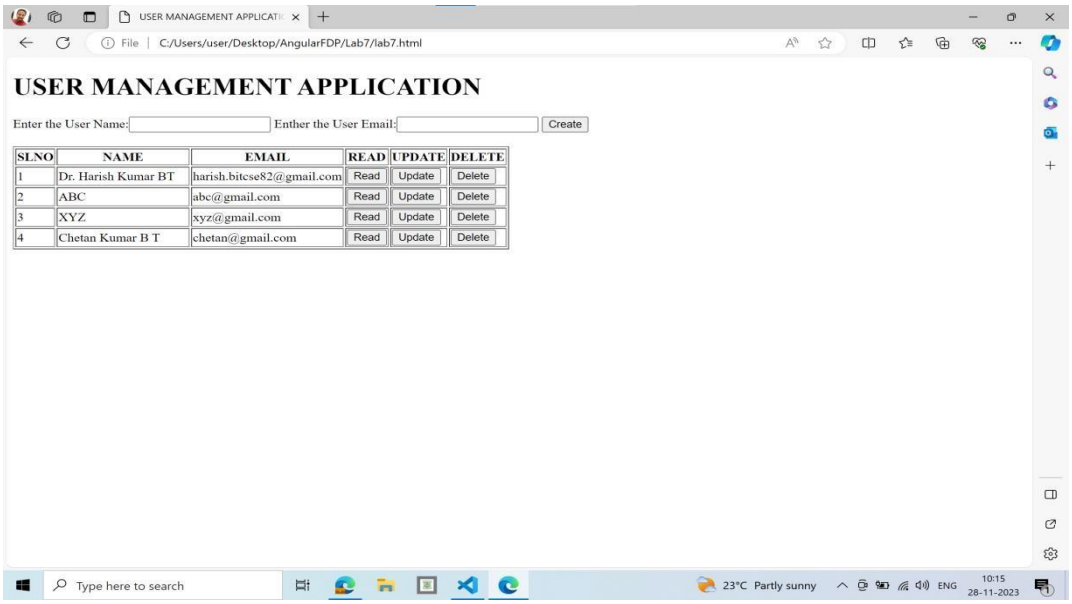

**7. Write an AngularJS program to create a simple CRUD application (Create, Read, Update, and Delete) for managing users.**

```
<!DOCTYPE html>
<html>
  <title>USER MANAGEMENT APPLICATION</title>
  <head>
    <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
    </script>          <script>          var
app=angular.module("userMgmtApp",[]);
        app.controller("userMgmtAppCntrl",function($scope){
          $scope.users=[
 {'name':"Dr. Harish Kumar BT",
'email':'harish.bitcse82@gmail.com','editing':false},
 {'name':'ABC','email':'abc@gmail.com','editing':false},
  {'name':'XYZ','email':'xyz@gmail.com','editing':false}
          ]

          $scope.createUser=function()
          {
            if($scope.newUserName && $scope.newUserEmail)
            {          var
u={
              'name':$scope.newUserName,
              'email':$scope.newUserEmail,
              'editing':false
            }

            $scope.users.push(u)
            $scope.newUserName="
```

```
            $scope.newUserEmail=''
            }           else{           alert("Please provide the user name and
email id")          }

        }

            $scope.readUser=function(user)
            {           user.editing=true
            }


           $scope.updateUser=function(user){           user.editing=false
            }


            $scope.deleteUser=function(user)
            {
              var yes=confirm("Are you sure you want to delete")
if(yes==true)
            {           var index=$scope.users.indexOf(user)

            $scope.users.splice(index,1)
            }
            }

      });

    </script>

  </head>

  <body ng-app="userMgmtApp">

    <h1>USER MANAGEMENT APPLICATION</h1>
    <div ng-controller="userMgmtAppCntrl">

      Enter the User Name:<input type="text" ng-model="newUserName">
      Enther the User Email:<input type="text" ng-model="newUserEmail">
      <button ng-click="createUser()">Create</button>
      <br/>
      <br/>
      <table border="1">
        <tr>
```

```
                <th>SLNO</th>
                <th>NAME</th>
                <th>EMAIL</th>
                <th>READ</th>
                <th>UPDATE</th>
                <th>DELETE</th>
            </tr>
            <tr ng-repeat="user in users">
                <td>{{$index+1}}</td>
                <td>          <span
ng-how="!user.editing">{{user.name}}</span>    
 <input type="text" ng-show="user.editing" ng-model="user.name">



                        </td>
                <td>
                    <span ng-show="!user.editing">{{user.email}}</span>
          <input type="text" ng-show="user.editing" ng-model="user.email">
                </td>
                <td>
                    <button ng-click="readUser(user)">Read</button>
                </td>
                <td>
                    <button ng-click="updateUser(user)">Update</button>
                </td>
                <td>
                    <button ng-click="deleteUser(user)">Delete</button>
                </td>
            </tr>
          </table>

        </div>
    </body>
</html>
```

**Output:**

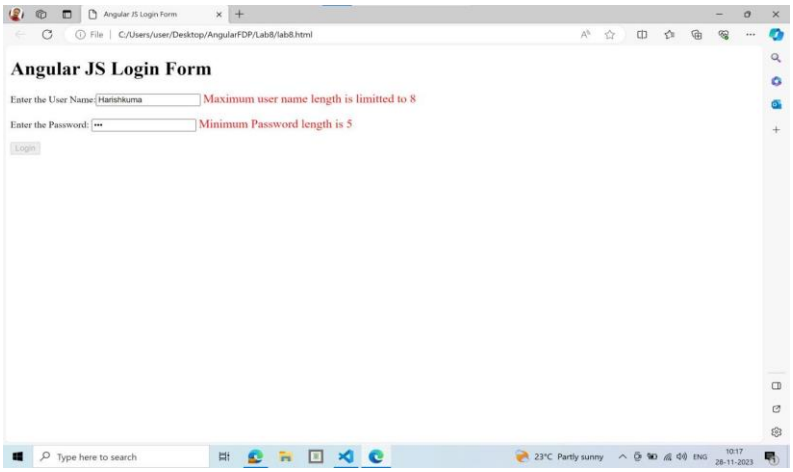



**8. Develop AngularJS program to create a login form, with validation for the username and password fields.**

```
<!DOCTYPE html>
<html>
   <title>Angular JS Login Form</title>
   <head>
      <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">        </script>

      <script>          var
app=angular.module("loginApp",[]);
         app.controller('loginAppCntrl',function($scope){
            $scope.userName=''
            $scope.password=''

            $scope.noAttempts=0
            $scope.login=function(){
             //  console.log("Inside login function")
              if($scope.userName=="harish" &&
$scope.password=="12345678")              {
alert("Login Successfull")
            }              else{
              $scope.noAttempts++              if($scope.noAttempts<=3)
              {              alert("Incorrect user name/password! Attempt No.
"+$scope.noAttempts)
}              else{
               document.getElementById("loginButton").disabled=true              }
```

```
                }
            }


        });
    </script>
    <style>
        .error-message{        color:red;
    font-size: 20px;    }
    </style>
</head>

<body ng-app="loginApp" ng-controller="loginAppCntrl">

  <h1>Angular JS Login Form</h1>
  <form name="loginForm" ng-submit="submitForm()">
    Enter the User Name:<input type="text" name="userName"
ng-model="userName" ng-minlength="5" ng-maxlength="8" required placeholder="Enter User
Name">
    <span class="error-message"
 ng-show="loginForm.userName.$error.required && loginForm.userName.$dirty">User
Name is Required</span>  <span class="error-message"  ng-
show="loginForm.userName.$error.minlength">Minimum Length Must be 5</span>
    <span class="error-message"  ng-
show="loginForm.userName.$error.maxlength">Maximum user name length is limitted to
8</span>        <br/>
    <br/>
    Enter the Password: <input type="password" name="password"  ng-model="password" ng-
minlength="5" ng-maxlength="8" required placeholder="Enter your password">
    <span class="error-message" ng-show="loginForm.password.$error.required
&& loginForm.password.$dirty">Password is required</span>
    <span class="error-message"  ng-show="loginForm.password.$error.minlength">Minimum
Password length is 5</span>
    <span class="error-message"  ng-
show="loginForm.password.$error.maxlength">Maximum password length is limitted to
8</span>        <br/>
    <br/>
    <button type="submit" ng-disabled="loginForm.$invalid"  ng-
click="login()" id="loginButton">Login</button>
  </form>
  </body>
</html>
```

**Output:**

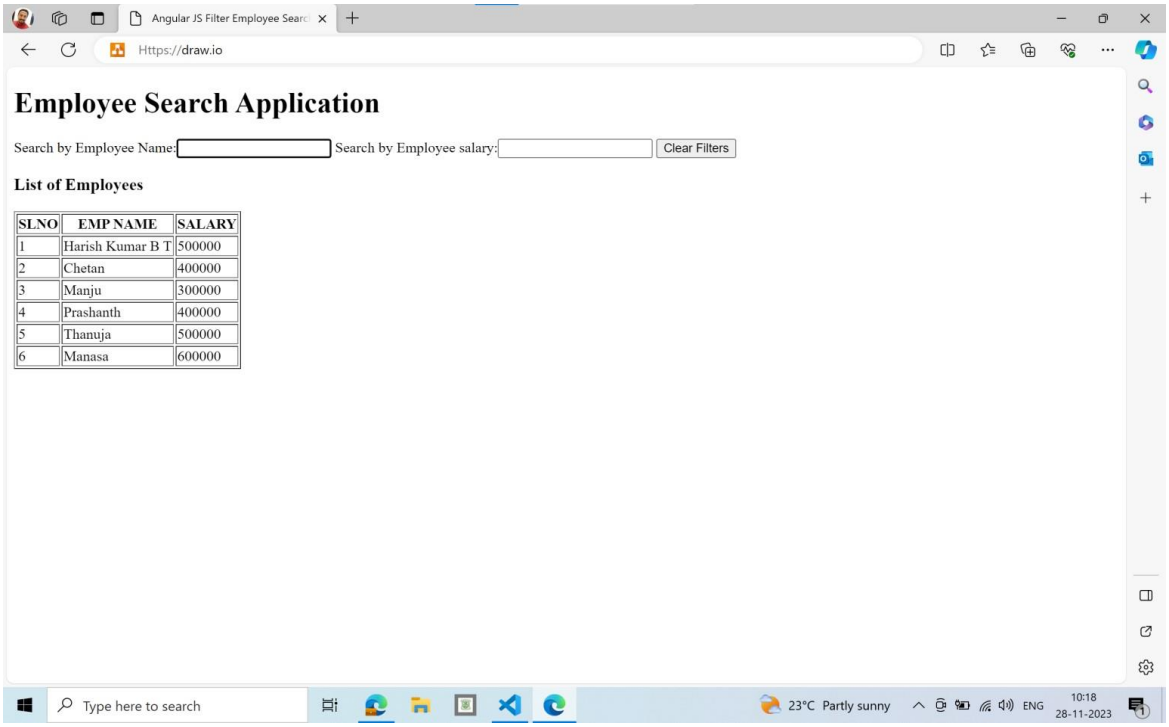


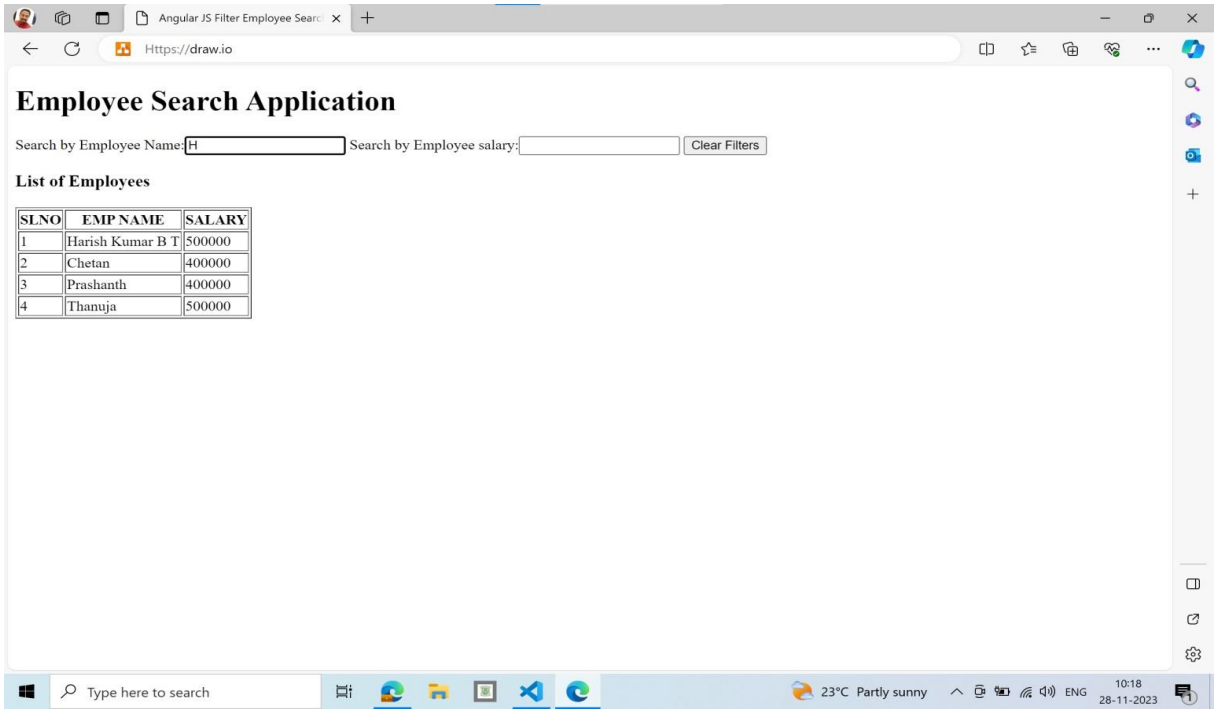

**9. Create an AngularJS application that displays a list of employees and their salaries. Allow users to search for employees by name and salary. Note: Employee details may be included in the program.**

```
<!DOCTYPE html>
<html>
  <title>Angular JS Filter Employee Search Application</title>
  <head>
    <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.mi n.js">
    </script>

    <script>          var
app=angular.module("empSearchApp",[]);
        app.controller("empSearchAppCntrl",function($scope){          $scope.empList=[
            {'name':'Harish Kumar B T','salary':500000},
            {'name':'Chetan','salary':400000},
            {'name':'Manju','salary':300000},
            {'name':'Prashanth','salary':400000},
            {'name':'Thanuja','salary':500000},
            {'name':'Manasa','salary':600000}
          ]

            $scope.clearFilters=function()
            {
              $scope.searchName=''
              $scope.searchSalary=''
```

```
                }

            });
        </script>
    </head>

    <body ng-app="empSearchApp">
        <h1>Employee Search Application</h1>
        <div ng-controller="empSearchAppCntrl">
            Search by Employee Name:<input type="text" ng-model="searchName">          Search
by Employee salary:<input type="number"  ng-model="searchSalary">

            <button ng-click="clearFilters()">Clear Filters</button>
            <br/>
            <h3>List of Employees</h3>
            <table border="1">
                <tr>
                    <th>SLNO</th>
        <th>EMP NAME</th>
        <th>SALARY</th>
    </tr>
    <tr ng-repeat="emp in empList | filter:{name:searchName,salary:searchSalary}">
        <td>{{$index+1}}</td>
        <td>{{emp.name}}</td>
        <td>{{emp.salary}}</td>
    </tr>

</table>

    </div>
  </body>
</html>
```

**Output:**


Angular JS Filter Employee Searc
Https://draw.io
Employee Search Application
Search by Employee Name:
Search by Employee salary:
Clear Filters
List of Employees
SLNO EMP NAME SALARY
1 Harish Kumar B T 500000
2 Chetan 400000
3 Manju 300000
4 Prashanth 400000
5 Thanuja 500000
6 Manasa 600000
Type here to search
23°C Partly sunny
ENG
10:18
28-11-2023

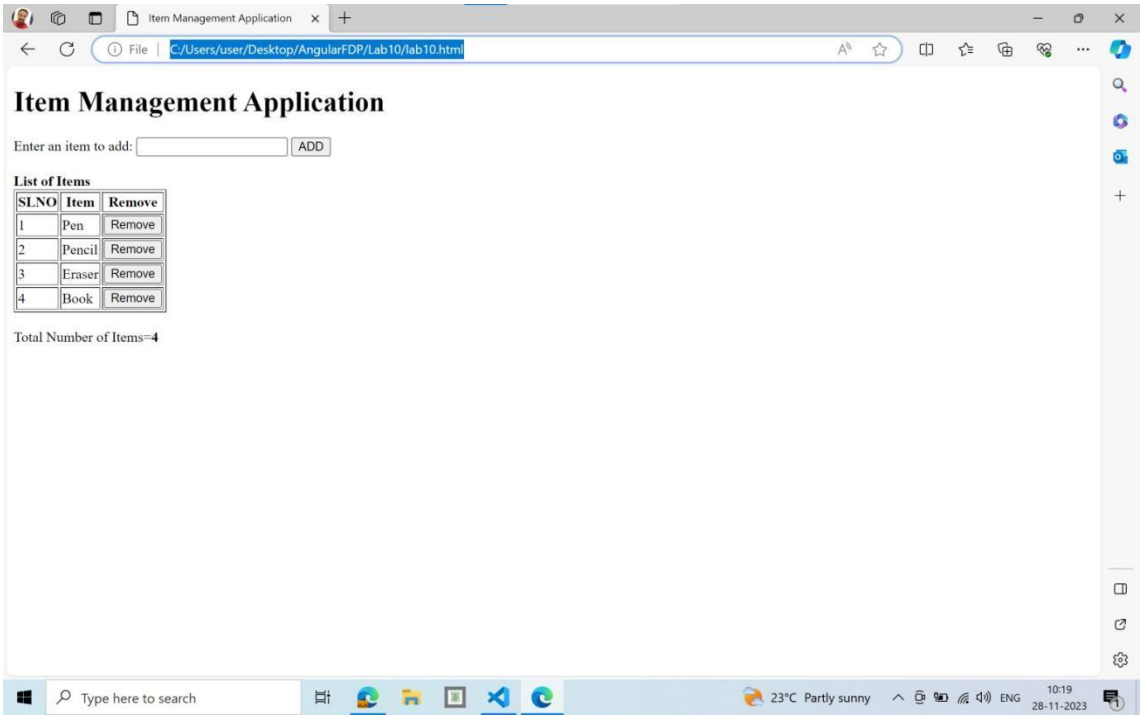

Angular JS Filter Employee Searc
Https://draw.io
Employee Search Application
Search by Employee Name:H
Search by Employee salary:
Clear Filters
List of Employees
SLNO EMP NAME SALARY
1 Harish Kumar B T 500000
2 Chetan 400000
3 Prashanth 400000
4 Thanuja 500000
Type here to search
23°C Partly sunny
ENG
10:18
28-11-2023

**10. Create AngularJS application that allows users to maintain a collection of items. The application should display the current total number of items, and this count should automatically update as items are added or removed. Users should be able to add items to the collection and remove them as needed. Note: The default values for items may be included in the program.**

```
<!DOCTYPE html>
<html>
  <title>Item Management Application</title>
  <head>
    <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">      </script>

      <script>          var app=angular.module("itemMgmtApp",[]);
app.controller("itemMgmtAppCntrl",function($scope){
          $scope.itemList=['Pen','Pencil','Eraser','Book']

          $scope.addItem=function()
          {              if($scope.newItem)
            {                  if($scope.itemList.indexOf($scope.newItem)==-1)
{
                $scope.itemList.push($scope.newItem)
              }                  else{              alert('This item is already there in the item
collection')
              }              }              else{
alert('Please Enter the item to add')
            }
          }

          $scope.removeItem=function(item)
          {
            var yes=confirm("Are you sure you want to delete "+item)              if(yes==true)
            {
              var index=$scope.itemList.indexOf(item)
              $scope.itemList.splice(index,1)
            }


          }

    });
  </script>
</head>
<body ng-app="itemMgmtApp">
```

```
    <h1>Item Management Application</h1>

    <div ng-controller="itemMgmtAppCntrl">
        Enter an item to add: <input type="text" ng-model="newItem">
        <button ng-click="addItem()">ADD</button>
        <br/><br/>

        <b>List of Items</b>
        <table border="1">
            <tr>
                <th>SLNO</th>
                <th>Item</th>
                <th>Remove</th>
            </tr>
            <tr ng-repeat="item in itemList">
                <td>{{$index+1}}</td>
                <td>{{item}}</td>
                <td><button ng-click=removeItem(item)>Remove</button></td>

            </tr>
        </table>
        <br/>

        Total Number of Items=<b>{{itemList.length}}</b>
    </div>

    </body>
</html>
```

**Output:**

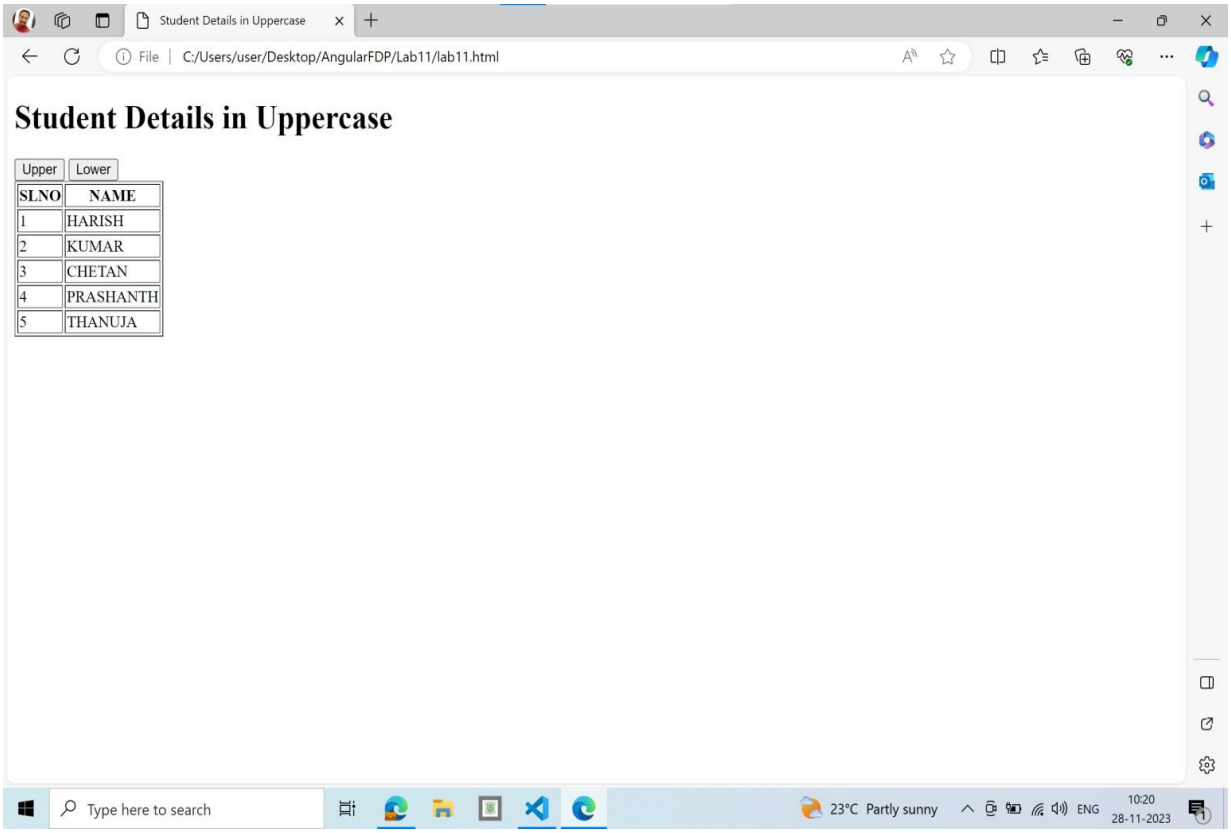



**11. Create AngularJS application to convert student details to Uppercase using angular filters.**
**Note: The default details of students may be included in the program.**

```
<!DOCTYPE html>
<html>
   <title>Student Details in Uppercase</title>
   <head>
      <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">      </script>

      <script>          var
app=angular.module("studDetailsUpperApp",[]);
         app.controller("studDetailsUpperAppCntrl",function($scope){
      $scope.studDetails=['harish','kumar','chetan','prashanth','thanuja']
$scope.upper=true
            $scope.lower=false


            $scope.Lower=function()
            {
              //console.log('called')
              $scope.upper=false
              $scope.lower=true
            }
```

```
            $scope.Upper=function()
            {
               $scope.upper=true
               $scope.lower=false
            }          });
        </script>
    </head>

    <body ng-app="studDetailsUpperApp">
        <h1>Student Details in Uppercase</h1>
        <div ng-controller="studDetailsUpperAppCntrl">
            <button ng-click="Upper()">Upper</button>
            <button ng-click="Lower()">Lower</button>
            <table border="1">
                <tr>
                    <th>SLNO</th>
                    <th>NAME</th>



                </tr>
            <tr ng-repeat="student in studDetails">
                <td>{{$index+1}}</td>
                <td ng-show="upper">{{student|uppercase}}</td>
                <td ng-show="lower">{{student|lowercase}}</td>
            </tr>
        </table>
    </div>

</body>
</html>
```

**Output:**

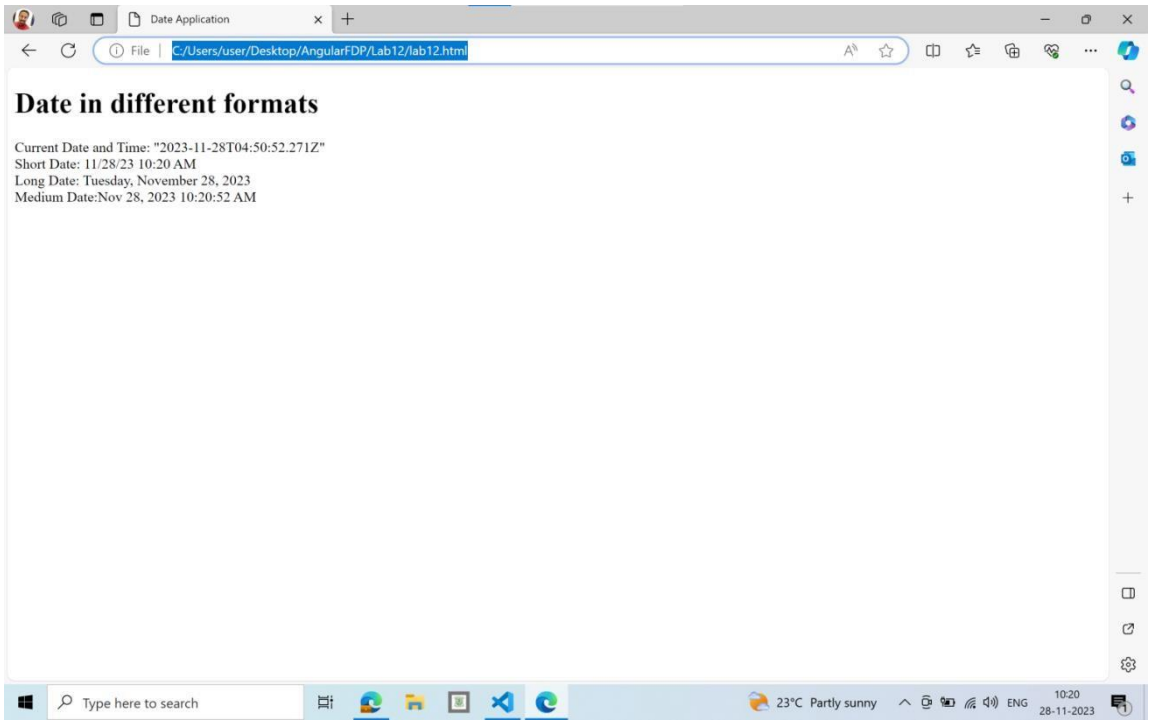



**12. Create an AngularJS application that displays the date by using date filter parameters.**

```
<!DOCTYPE html>
<html>
   <title>Date Application</title>
   <head>
      <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">      </script>

    <script>       var
app=angular.module("dateApp",[]);
      app.controller("dateAppCntrl",function($scope){
         $scope.currentDate=new Date();
      });
    </script>
   </head>
   <body ng-app="dateApp">
      <h1>Date in different formats</h1>
      <div ng-controller="dateAppCntrl">

         Current Date and Time: {{currentDate}}<br/>
```

```
            Short Date: {{currentDate|date: 'short'}}<br/>
            Long Date: {{currentDate |date: 'fullDate'}}<br/>
            Medium Date:{{currentDate| date: 'medium'}}
        </div>
    </body>
</html>
```

**Output:**




## Sample Programs

## 1. Example Program on Angular expression

```
<!DOCTYPE html>
<html>
  <title>
     This is my first angular progarm
  </title>
  <head>
  <script type="text/javascript"
src="https:\\ajax.googleapis.com\ajax\libs\angularjs\1.8.2\angular.min.js" >
  </script>
  </head>
  <body>
    <div ng-app="">
       {{5+5}}
    </div>

  </body> </html>
```

## 2. Example on ng-model, ng-bind, ng-init

```
<!DOCTYPE html>
<html>
   <title>Demo example on ng-model</title>
   <head>
      <script type="text/javascript"
  src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js ">
   </script>
   </head>
   <body>
      <div ng-app="" ng-init="age=41">
         Enter Your Name: <input type="text" ng-model="name"><br/>
Your Name is using angular expression: {{name}} <br/>          Your Name is
using ng-bind:  <p ng-bind="name"></p><br/>          your age is: <p ng-
bind="age"></p>
      </div>
  </body>
</html>
```

## 3. Example on ng-click

```
<!DOCTYPE html>
```

```
<html>
   <title>
      Demo on ng-click directive
   </title>
   <head>
      <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js ">
   </script>

   <script>       var app=angular.module("myApp",[]);
app.controller("myCntrl",function($scope){
         $scope.num1=20
         $scope.num2=30
         $scope.result

      $scope.add=function()
      {           console.log("Function Called")
$scope.result=$scope.num1 + $scope.num2
        // return $scope.result
      }

      });
   </script>
   </head>

   <body>
      <div ng-app="myApp" ng-controller="myCntrl">
         Click the button to get the result
         <button ng-click="add()">Click Here</button><br/>
         Result: {{result}}
      </div>
   </body>
</html>
```

## 4. Addition program

```
<!DOCTYPE html>

<html>
   <title>
      Addition Program
   </title>
```

```
<head>
  <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js ">

  </script>

  <script>

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

    app.controller("myCntrl",function($scope){
       $scope.num1=0
       $scope.num2=0
       $scope.result=0

       $scope.add=function()
       {
          $scope.result=$scope.num1 + $scope.num2
       }

    });

  </script>

</head>

<body ng-app="myApp">

    <h1>Addition Program</h1>

    <div ng-controller="myCntrl">

       Enter First Number: <input type="number" ng-model="num1"><br/>
       Enter Second Number: <input type="number" ng-model="num2"><br/>
       <button ng-click="add()">ADD</button><br/>
       Sum of {{num1}} and {{num2}} = {{result}}

    </div>

</body> </html>
```

## 5. Example on rootScope and Controller scope variables

```
<!DOCTYPE html>

<html>
   <title>Demo on root scope variable</title>
<head>
   <script type="text/javascript"
 src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"
>
</script>

<script>
var app=angular.module("myApp",[]);
 app.run(function($rootScope){
$rootScope.dept1="CSE"
   $rootScope.dept2="ECE"
});
app.controller("myCntrl", function($scope){
   $scope.name1="Harish Kumar B T"
});
app.controller("myAnotherCntrl", function($scope){
   $scope.name2="Chetan Kumar B T"
});

</script>

</head>

<body ng-app="myApp">
   <div ng-controller="myCntrl">
    Dept. of {{dept1}} Faculties<br/>
1. {{name1}}
2. {{name2}}

   </div>

   <div ng-controller="myAnotherCntrl">
   Dept. of {{dept2}} Faculties<br/>
1. {{name1}}
2. {{name2}}
   </div>
```

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

## 6. -options

```
<!DOCTYPE html>

<html>
<title> Demo on Using the ng-options to fill the dropdown list from the list of values</title>
<head>
    <script type="text/javascript"
   src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
</script>
 <script>
    var app=angular.module("myApp",[]);

    app.controller("myCntrl",function($scope){
        $scope.myList=['Apple','Mango','Banana','Orange']
    });
</script>
</head>
<body ng-app="myApp">

    <div ng-controller="myCntrl">
        List of Fruits: <select ng-model="selectFruit" ng-options="fruit for fruit in myList"></select>

    </div>
</body>
</html>
```

## 7. -repeat

```
<!DOCTYPE html>

<html>
```

```
<title>
   Demo on using ng-repeat
</title>

<head>
   <script type="text/javascript"
  src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js
">
</script>
 <script>
   var app=angular.module("myApp",[]);

   app.controller("myCntrl",function($scope){
      $scope.myList=['Cow','Lion','Tiger','Deer','Sheep']
   });
</script>
</head>

<body ng-app="myApp">

   <div ng-controller="myCntrl">

      <ul>
         <li ng-repeat="animal in myList">{{animal}}</li>
      </ul>
   </div>
</body>
</html>
```

## 8. -show

```
<!DOCTYPE html>

<html>
   <title>
      Demo pgm on using ng-show directive
   </title>

   <head>
```

```
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
</script>

<script>
 var app=angular.module("myApp",[]);
app.controller("myCntrl",function($scope){
  $scope.display=true;
});
</script>
</head>

<body ng-app="myApp">
  <div ng-controller="myCntrl">
    <li ng-show="display">Harish</li>
  </div>
</body>
</html>
```

## 9. Example on random number generation

```
<!DOCTYPE html>
<html>
<title>Random Number Generation Demo</title>
<head>
<script type="text/javascript"
 src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js" >
</script>

<script>
 var app=angular.module("myApp",[]);
app.controller("myCntrl",function($scope){
  $scope.randomList=[]

  $scope.generateRandomNumbers=function(){
    $scope.randomList=[]
    for(var i=1;i<=$scope.num;i++)
    {
```

```
            $scope.randomList.push((Math.random()*$scope.num+1).toFixed(2 ))
          }
        }
    });
    </script>
    </head>
    <body ng-app="myApp">
        <div ng-controller="myCntrl">
            Enter the Number of random numbers to be generated: <input
type="numbers" ng-model="num">            <button ng-
click="generateRandomNumbers()">Generate</button><br/>
<li ng-repeat="x in randomList">{{x}}</li>
        </div>
    </body>
</html>
```

## 10. Example on student List Object

```
<!DOCTYPE html>

<html>
<title> Demo on List of Student Objects</title>

<head>
   <script type="text/javascript"
 src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"> </script>

<script>
 var app=angular.module("studListApp",[]);

 app.controller("studListAppCntrl", function($scope){
    $scope.studList=[
      {'USN':'1BI20CS001','NAME':'ABC','DEPT':'CSE','SEM':7,'MARKS':78},
      {'USN':'1BI20EC002','NAME':'XYZ','DEPT':'ECE','SEM':6,'MARKS':89},
      {'USN':'1BI20EE003','NAME':'PQR','DEPT':'EEE','SEM':7,'MARKS':56},
      {'USN':'1BI20CS002','NAME':'LMN','DEPT':'CSE','SEM':7,'MARKS':90},
            ]
 });

</script>
</head>
<body ng-app="studListApp">

   <h1>Student List</h1>

   <div ng-controller="studListAppCntrl">

      <table border="1">
         <tr>
            <th>USN</th>
            <th>NAME</th>
            <th>DEPT</th>
            <th>SEM</th>
            <th>MARKS</th>
         </tr>
               <tr ng-repeat="student in studList">
```

```
            <td>{{student.USN}}</td>
            <td>{{student.NAME}}</td>
            <td>{{student.DEPT}}</td>
            <td>{{student.SEM}}</td>
            <td>{{student.MARKS}}</td>

        </tr>
      </table>

  </div>
</body>
</html>
```

## 11.Example on ng-blur

```
<!DOCTYPE html>
<html>
  <title>
     Demo on ng-blur directive
  </title>
  <head>
  <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
  </script>
  </head>
  <body ng-app="">
     Keep focus and keep losing focus from me to get the action done:
<input type="text" ng-model="name" ng-init="count=0"  ng-blur="count=count+1">
     <br/>
     {{count}}
  </body>
</html>
```