

**LABORATORY MANUAL**

**21CSL581**

**ANGULAR JS LABORATORY**

**2023-2024**



**COMPILED BY**

**Prof. Keerthi P**

**COMPUTER SCIENCE AND ENGINEERING**

**IMPACT COLLEGE OF ENGINEERING & APPLIED SCIENCES**

**Bangalore - 560 092**

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		
<b>Course objectives:</b> <ul style="list-style-type: none"><li>• To learn the basics of Angular JS framework.</li><li>• To understand the Angular JS Modules, Forms, inputs, expression, data bindings and Filters</li><li>• To gain experience of modern tool usage (VS Code, Atom or any other] in developing Web applications</li></ul>			
<b>Sl.NO</b>	<b>Experiments</b>		
1	Develop Angular JS program that allows user to input their first name and last name and display their full name. <b>Note:</b> 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. <b>Note:</b> 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. <b>Note:</b> 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. <b>Note:</b> 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. <b>Note:</b> 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. <b>Note:</b> The default values for items may be included in the program.		
11	Create AngularJS application to convert student details to Uppercase using angular filters. <b>Note:</b> 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		

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="Keerthi"
    $scope.lastName="P"
  });
</script>
</head>
<body ng-app="myApp">
  <h2>Angular 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:**

← ↻ ⓘ File | D:/pgm1.html

### Anjular JS Application to Display Full Name

Enter First Name:

Enter Last Name:

Your Full Name: keerthi p

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=['COSMETICS','GROCERIES','CLOTHING','VEGETABLES']
    $scope.addItem=function(){
      if($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:**

← ↻ ⓘ File | D:/pgm2.html

## Shopping Application

**List of Shopping Items**

SLNO	Item
1	COSMETICS
2	GROCERIES
3	CLOTHING
4	VEGETABLES

Enter an Item to Add:

Select an Item to Remove:

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:

A browser window showing the Angular JS Simple Calculator. The title bar says "File | D:/pgm3.html". The page title is "Angular JS Simple Calculator". The form has two input fields: "Enter First Number:" with value 3, and "Enter Second Number:" with value 2. A dropdown menu for "Select Operator:" shows "\*" selected. A "Compute" button is on the right. Below the inputs, the result is displayed as "3 mul 2=6".

A browser window showing the Angular JS Simple Calculator. The title bar says "File | D:/pgm3.html". The page title is "Angular JS Simple Calculator". The form has two input fields: "Enter First Number:" with value 3, and "Enter Second Number:" with value 2. A dropdown menu for "Select Operator:" shows "-" selected. A "Compute" button is on the right. Below the inputs, the result is displayed as "3 sub 2=1".

A browser window showing the Angular JS Simple Calculator. The title bar says "File | D:/pgm3.html". The page title is "Angular JS Simple Calculator". The form has two input fields: "Enter First Number:" with value 3, and "Enter Second Number:" with value 2. A dropdown menu for "Select Operator:" shows "/" selected. A "Compute" button is on the right. Below the inputs, the result is displayed as "3 div 2=1.5".

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

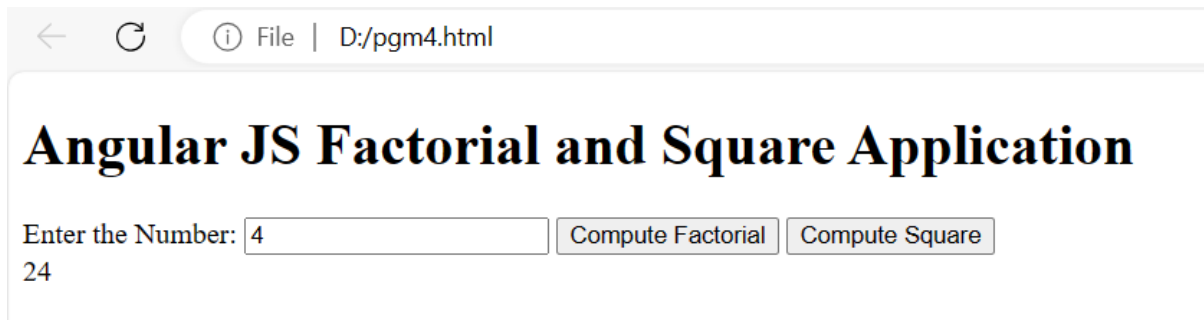
            $scope.factorial=function()
            {
                if($scope.num==0)
                {
                    $scope.result=1
                }
                else{
                    $scope.fact=1
                    for(var i=$scope.num; i>=1; i--)
                    {
                        $scope.fact=$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>
</div>
</body>
</html>
```

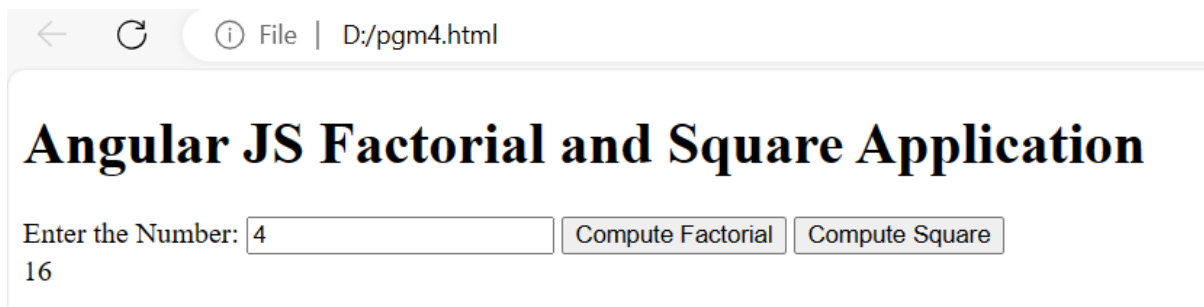
```
<br/>

    {{result}}

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

**Output:**

A screenshot of a web browser displaying the application. The browser's address bar shows 'D:/pgm4.html'. The page title is 'Angular JS Factorial and Square Application'. Below the title, there is a text input field with the value '4', followed by two buttons: 'Compute Factorial' and 'Compute Square'. The output of the application is displayed below the input field, showing the number '24'.



A screenshot of the same web browser displaying the application. The browser's address bar shows 'D:/pgm4.html'. The page title is 'Angular JS Factorial and Square Application'. Below the title, there is a text input field with the value '4', followed by two buttons: 'Compute Factorial' and 'Compute Square'. The output of the application is displayed below the input field, showing the number '16'.

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="studDetailsAppCtrl">
    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:**

Student Details Application

Enter the Number of Students to Generate the Data:

SLNO	NAME	CGPA
1	Student-1	7.21
2	Student-2	4.02
3	Student-3	8.10
4	Student-4	6.36
5	Student-5	4.16

Number of Students=5

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)

        {
```

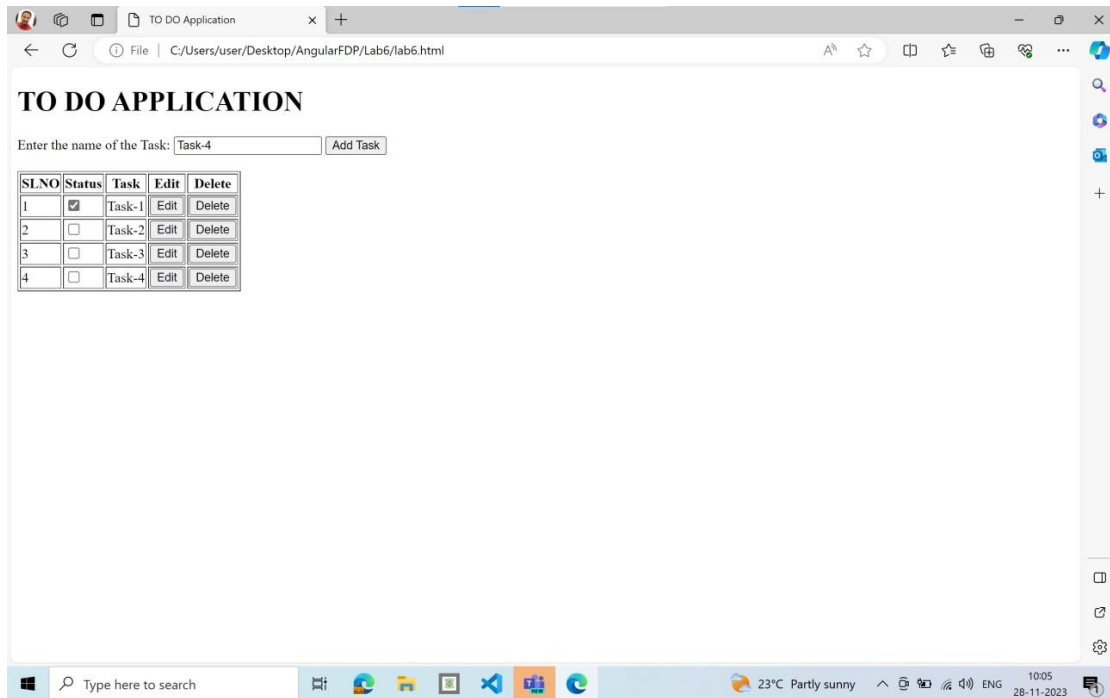
```

        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':"PRABHAS", 'email':PRABHI@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
        }
      }
    </script>
  </head>
</html>
```

[illegible]

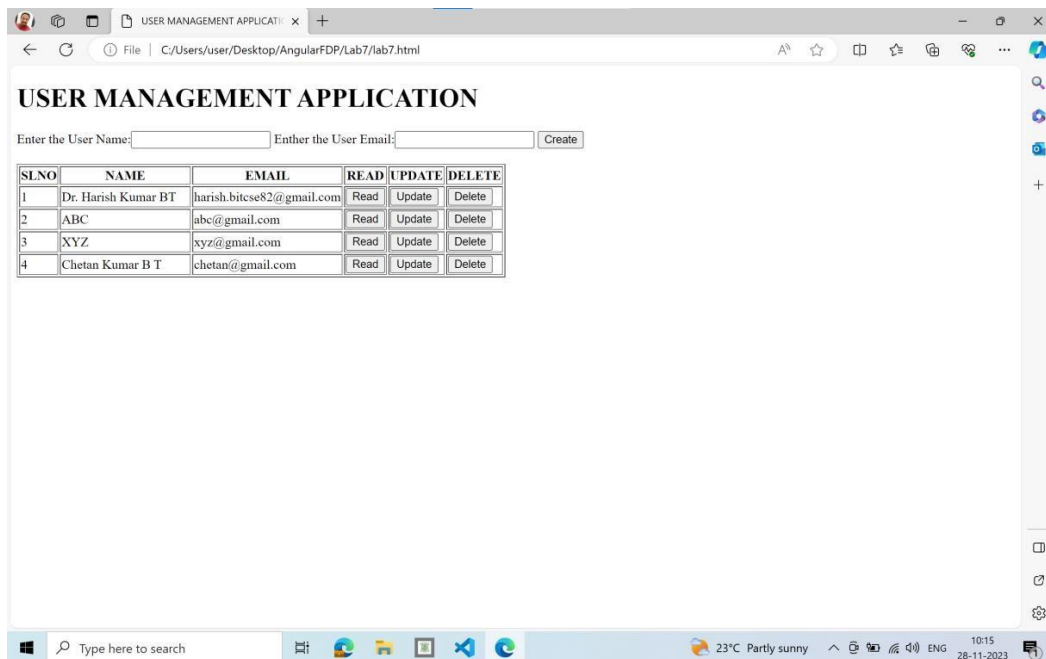
```

        </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="loginAppCtrl">

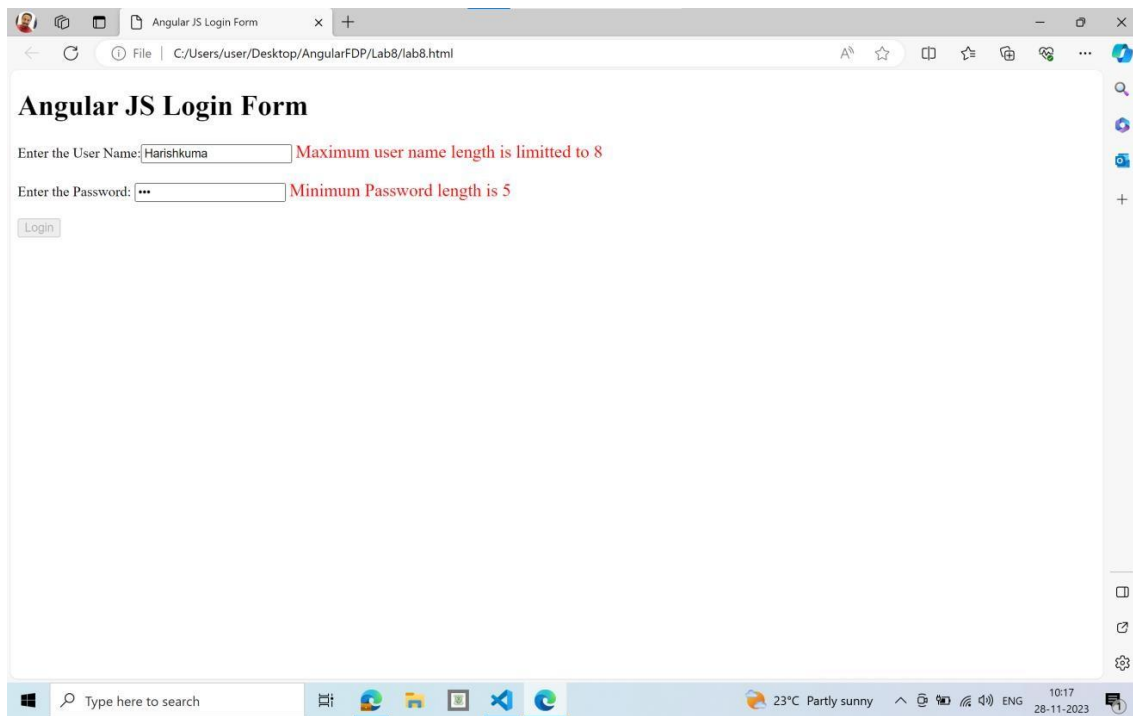
  <h1>Angular JS Login Form</h1>
  <form name="loginForm" ng-submit="submitForm()">

    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 limited
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 limited
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.min.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">

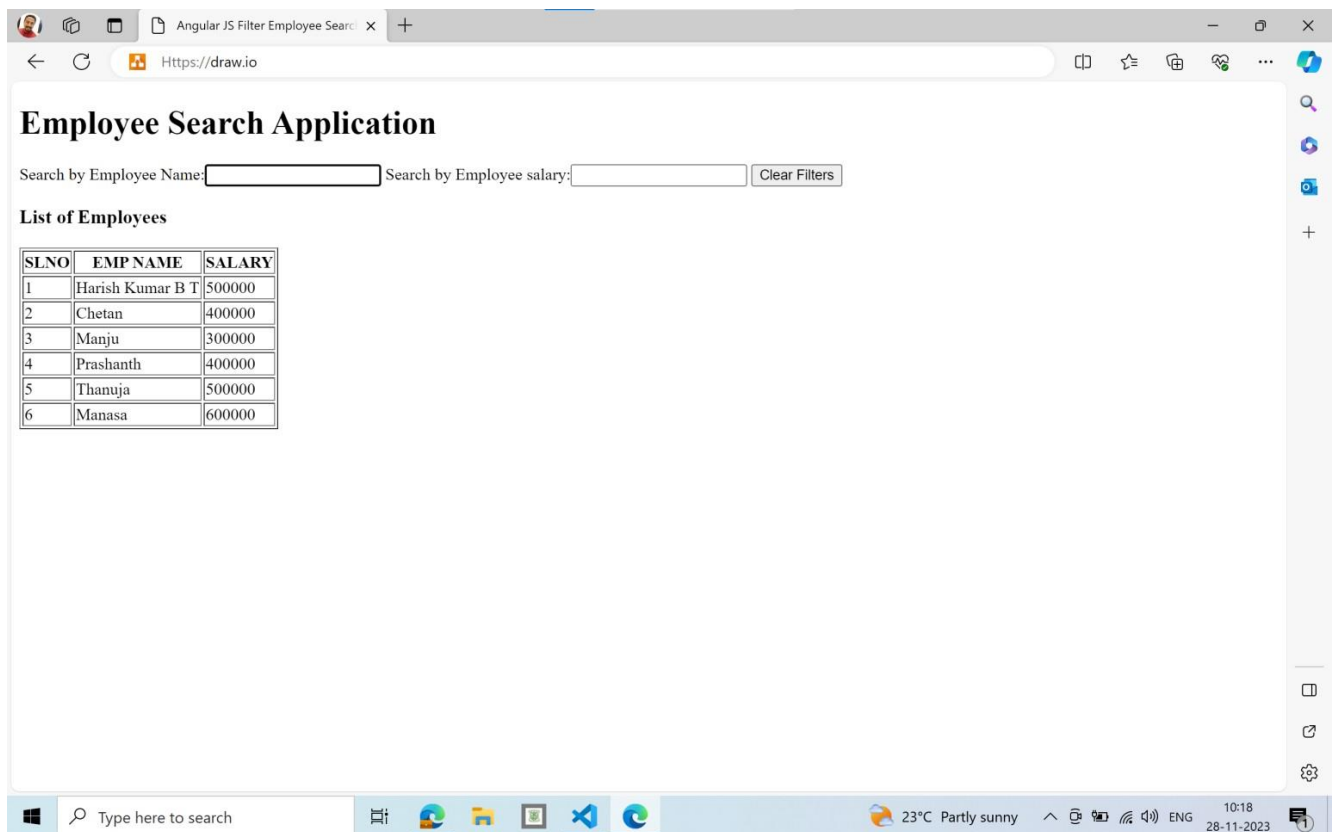
      <button ng-click="clearFilters()">Clear Filters</button>
      <br/>
      <h3>List of Employees</h3>
      <table border="1">
```

```

        <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:





The screenshot shows a web browser window with the title "Angular JS Filter Employee Search". The address bar shows "https://draw.io". The application is titled "Employee Search Application". It features two search filters: "Search by Employee Name:" with a text input containing the letter "H", and "Search by Employee salary:" with an empty text input. A "Clear Filters" button is located to the right of the salary input. Below the filters, the text "List of Employees" is displayed above a table. The table has three columns: "SLNO", "EMP NAME", and "SALARY". It contains four rows of employee data. The Windows taskbar at the bottom shows the search bar, task icons, system tray with weather and time (10:18, 28-11-2023), and language (ENG).

**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	Prashanth	400000
4	Thanuja	500000

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>
</html>
```

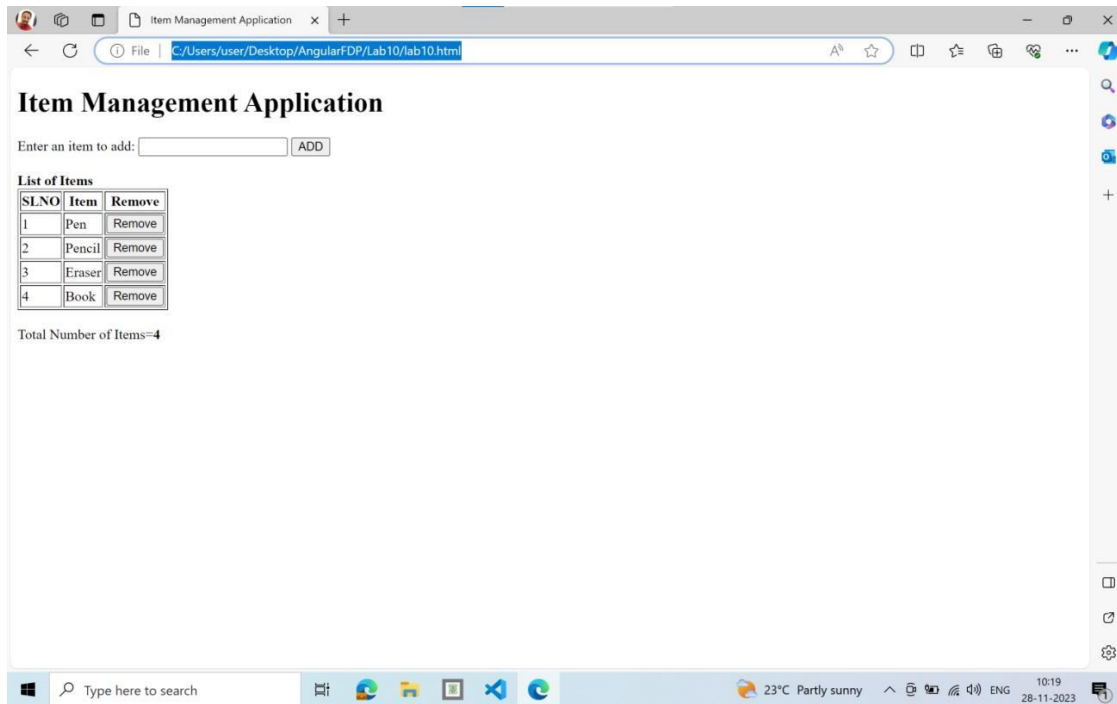
```
    });
  </script>
</head>
<body ng-app="itemMgmtApp">
  <h1>Item Management Application</h1>

  <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
        }

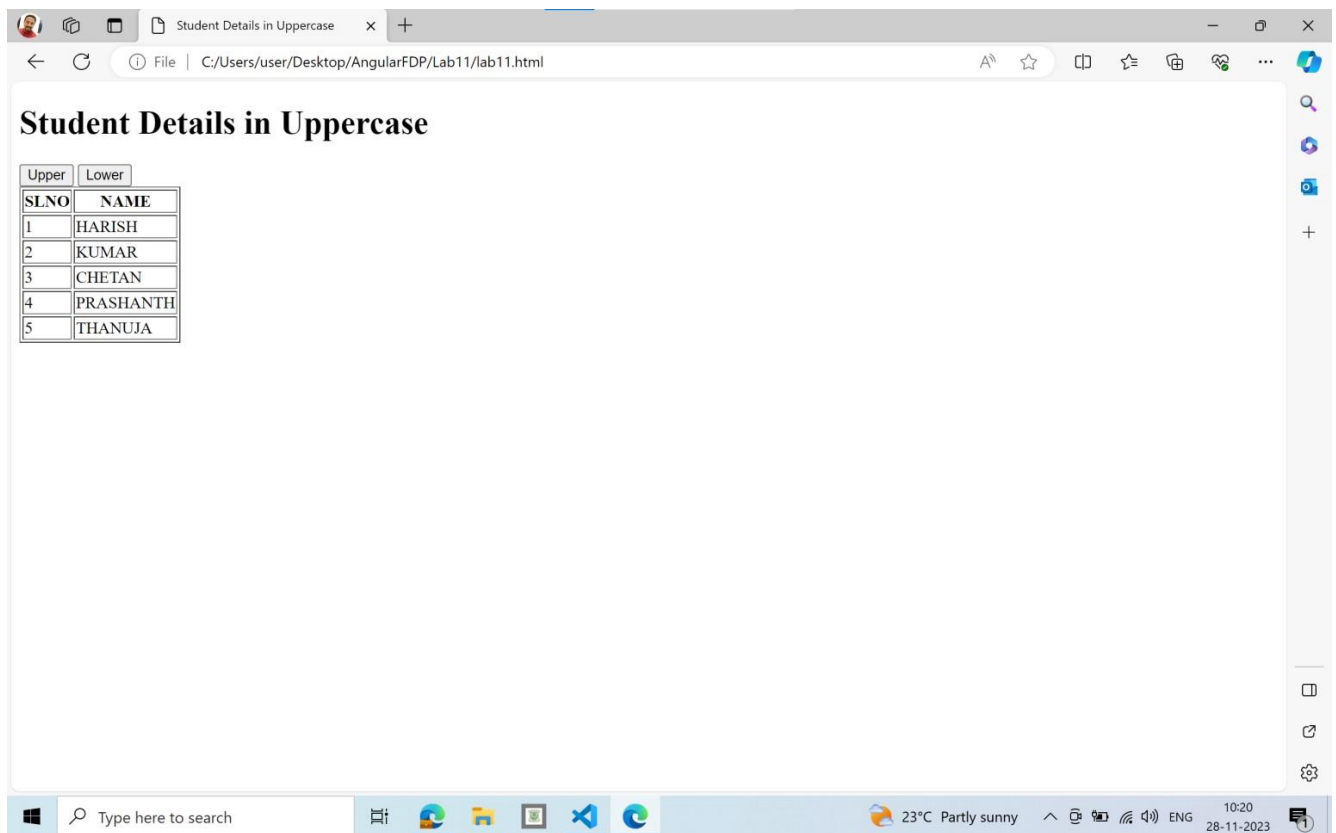
        $scope.Upper=function()
        {
          $scope.upper=true
          $scope.lower=false
        }
      });
    </script>
  </head>

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

```
        </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:**