



## Angular JS Lab Manual - 21CSL581

Angular js Node (Visvesvaraya Technological University)



Scan to open on Studocu

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI**



**ANGULAR JS  
LAB MANUAL (21-SCHEME)  
21CSL581  
(Prepared By – VINUTHA PRASHANTH)**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
ACCREDITED BY NATIONAL BOARD OF ACCREDITATION  
RAO BAHADUR Y MAHABALESWARAPPA ENGINEERING  
COLLEGE  
ACCREDITED BY NAAC WITH A+  
CANTONMENT, BALLARI-583104, KARNATAKA  
2023-24**

## SYLLABUS

ANGULAR JS			
Course Code	21CSL581/ 21CBL583	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>SLNO</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		
<b>NOTE:</b> Include necessary HTML elements and CSS for the above Angular applications.			
<b>Course outcomes (Course Skill Set):</b> At the end of the course the student will be able to: <ul style="list-style-type: none"><li>Develop Angular JS programs using basic features</li><li>Develop dynamic Web applications using AngularJS modules</li><li>Make use of form validations and controls for interactive applications</li><li>Apply the concepts of Expressions, data bindings and filters in developing Angular JS programs</li><li>Make use of modern tools to develop Web applications</li></ul>			

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.

```
<!-- Using Interpolation Technique -->
<html>
  <head>
    <title>Angular JS LAB1</title>
    <script type="text/javascript" src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"> </script>
  </head>
  <body ng-app="">
    <h1>Angular Lab Program 1: Interpolation</h1>
    <p>Enter your name: <input type = 'text' ng-model =
'name'> </p>
    <p>Hello {{name}}</p>
  </body>
</html>
```

OR

```
<!-- Using ng-model and ng-bind Directive -->
<html>
  <head>
    <title>Angular Lab 1</title>
    <script type="text/javascript" src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
  </script>
  <script type="text/javascript">
    var app = angular.module("myApp",[]);
    app.controller("myCtrl", function($scope){
      $scope.fname = "First";
      $scope.lname = "Last";
    });
  </script>
</head >
<body ng-app="myApp">
```

```
<div ng-controller = "myCtrl">
  <h1>Angular Lab Program 1: Using ng-model and ng-bind </h1>
  <p>Enter First Name:<input type="text" ng-model="fname">
</p>
  <p>Enter Last Name:<input type="text" ng-model="lname">
</p>
  <h1>Hello, Welcome <span ng-bind="fname"> </span>
    <span ng-bind="lname"> </span>
</h1>
</div>
</body>
</html>
```



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.

```
<html>
  <head>
    <title>Lab Program 2</title>
    <script type = "text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/an
gular.min.js"></script>
    <script type="text/javascript">
      var app = angular.module("myApp2",[]);
      app.controller("myCtrl2", function($scope){
        $scope.items = ["chocolates", "grocessories"];
        $scope.addItem = function(){
          $scope.items.push($scope.newItem);
          $scope.newItem="";
        };
        $scope.deleteItem = function(){
          var index =
$scope.items.indexOf($scope.newItem);
          if(index != -1)
            $scope.items.splice(index, 1);
          $scope.newItem = "";
        }
      })
    </script>
  </head>
  <body ng-app="myApp2">
    <div ng-controller="myCtrl2">
      <h1>Shopping list</h1>
      Enter the item: <input type="text" ng-model="newItem">
      <button ng-click="addItem()"> Add Item</button>
      <button ng-click="deleteItem()">Delete Item</button>
      <ul>
        <li ng-repeat="x in items">{{x}}</li>
      </ul>
    </div>
  </body>
</html>
```

## ACADEMIC YEAR -2023-24

Lab Program 2

×

+

← → ↻ ⓘ File | E:/vinu/\_G%20Drive/2023-24/ODD/Angular%20JS/Angular%20JS%20Lab%20Programs/Lab22.htm

### Shopping list

Enter the item:

- chocolates
- grocessories

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>
<head>
  <meta http-equiv="CONTENT-TYPE" content="text/html; charset=UTF-8">
  <title>Lab 3</title>
  <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
</script>
<style>
  div {
    margin: auto;
    padding: 10px;
    width: 50%;
    text-align: center;
    border: 1px solid black;
  }
</style>
</head>
<body ng-app="">
  <div>
    <h1> Lab3 - Simple Calculator </h1>
    <p>Enter value 1: <input type="number" ng-model="num1" ng-init="num1=10">
    </p>
    <p>Enter value 2: <input type="number" ng-model="num2" ng-init="num2=5">
    </p>
    <div>
      <h3>Addition: {{num1}} + {{num2}} = {{num1+num2}}</h3>
      <h3>Subtraction: {{num1}} - {{num2}} = {{num1-num2}}</h3>
      <h3>Multiplication: {{num1}} * {{num2}} = {{num1*num2}}</h3>
      <h3>Division: {{num1}} / {{num2}} = {{num1/num2}}</h3>
    </div>
  </div>
</body>
</html>
```



### Lab3 - Simple Calculator

Enter value 1:

Enter value 2:

**Addition:  $10 + 5 = 15$**

**Subtraction:  $10 - 5 = 5$**

**Multiplication:  $10 * 5 = 50$**

**Division:  $10 / 5 = 2$**

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

```

<!DOCTYPE html>
<html>
<head> <title>Lab 4</title>

    <style>
        div {
            border: 1px solid black;
            margin: auto;
            width:50%;
            padding: 5px;
            text-align:center;
        }

    </style>
</head>
<body ng-app="app1">
    <div ng-controller = "ctrl1">
        <h1>
            Lab 4 - Square and Factorial
        </h1>
        <p>Enter value 1: <input type="number" ng-model="num1"
default=10></p>
        <p><button ng-click="calculate()">Calculate</button></p>

        <h3>Factorial = {{ fact }}</h3>
        <h3>Square = {{square}}</h3>
    </div>
    <script type="text/javascript"
        src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2
/angular.min.js">
    </script>
    <script type="text/javascript">
        var app = angular.module("app1", [])
        app.controller("ctrl1", function ($scope) {
            $scope.fact = 2;
            $scope.square = 4;
            $scope.num1 = 2;
            $scope.calculate = function () {
                $scope.fact = 1;
                for (i = 1; i <= $scope.num1; i++)
                    $scope.fact = $scope.fact * i;
                $scope.square = $scope.num1 * $scope.num1;
            }
        })
    </script>
</body>

```

</html>

## Lab 4 - Square and Factorial

Enter value 1:

Calculate

**Factorial = 24**

**Square = 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>
<head>
  <title>Document</title>
  <style>
    #one{
      margin:auto;
      padding:10px;
      width: 50%;
      text-align: center;
      border: 1px solid black;
    }
    table, th, td {
      border: 2px solid black;
      border-collapse: collapse;
      text-align: center;
      align-self: stretch;
      margin:auto;
    }
  </style>
</head>
<body ng-app="lab5" ng-controller="con5">
  <div id="one">
    <p>Want to insert new Student? <input type="checkbox" ng-
model="isInsertNew"></p>
    <div ng-show="isInsertNew">
      <p>Name: <input type="text" ng-model="name"></p>
      <p>USN: <input type="text" ng-model="usn"></p>
      <p>Sem: <input type="text" ng-model="sem"></p>
      <p>CGPA: <input type="text" ng-model="cgpa"></p>
      <p><button ng-click="insert()">Insert</button></p>
    </div>
    <table>
      <tr>
        <th>Name</th>
```

```

        <th>USN</th>
        <th>Sem</th>
        <th>CGPA</th>
    </tr>
    <tr ng-repeat="stu in students">
        <td>{{stu.name}}</td>
        <td>{{stu.usn}}</td>
        <td>{{stu.sem}}</td>
        <td>{{stu.cgpa}}</td>
    </tr>
    <tr>
        <td colspan="4">Number of Students: {{students.length}}
    </td>
    </tr>

</table>
</div>
<script type="text/javascript"
    src="https://ajax.googleapis.com/ajax/libs/angularjs/1.
8.2/angular.min.js">
    </script>
<script type="text/javascript">
    var lab5 = angular.module('lab5', []);
    lab5.controller("con5", function ($scope) {
        $scope.students = [
            { name: "ABC", usn: "3VC20CS001", sem: 5, cgpa: 9.25 },
            { name: "PQR", usn: "3VC20CS009", sem: 3, cgpa: 5.78 }
        ];
        $scope.insert = function () {
            $scope.students.push({ name: $scope.name, usn:
$scope.usn, sem: $scope.sem, cgpa: $scope.cgpa });
        }
    })
</script>

</body>
</html>

```

## ACADEMIC YEAR -2023-24

Want to insert new Student? ☒

Name:

USN:

Sem:

CGPA:

Name	USN	Sem	CGPA
ABC	3VC20CS001	5	9.25
PQR	3VC20CS009	3	5.78
XYZ	3VC21CS045	2	6.78
Number of Students: 3			

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>
<head>
  <title>To-Do List App</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.m
in.js"></script>
<style>
  div {
    margin: auto;
    padding: 10px;
    width: 50%;
    text-align: center;
    border: 1px solid black;
  }

  table, th, td {
    border: 2px solid black;
    border-collapse: collapse;
    text-align: center;
    align-self: stretch;
    margin: auto;
    padding: 5px;
  }
</style>
</head>
<body>

<div ng-app="lab6" ng-controller="cntr6">
  <h2>Lab6 - To-Do List</h2>
  <p>
    <input type="text" ng-model="newTask" placeholder="Add new
task" required>
    <button type="submit" ng-click="addTask()">Add Task</button>
  </p>
  <table>
    <tr>
      <th>Tasks</th>
      <th>Operations</th>
    </tr>
  </table>
</div>
```

```

    </tr>
    <tr ng-repeat="task in tasks">
        <td style="text-align: left;">
            <span ng-show="!task.editing">{{ task.name }}</span>
            <input type="text" ng-model="task.name"
                ng-show="task.editing">
        </td>
        <td>
            <button ng-click="editTask(task)">Edit</button>
            <button ng-click="deleteTask(task)">Delete</button>
        </td>
    </tr>
</table>
</div>

<script>
angular.module('lab6', [])
    .controller('cntr6', function($scope) {
        $scope.tasks = [
            { name: 'Revise DBMS'},
            { name: 'Practice Angular Programs' },
            { name: 'Practice SQL Queries' },
            { name: 'Design one page of DBMS Project' }
        ];

        $scope.addTask = function() {
            $scope.tasks.push({ name: $scope.newTask });
            $scope.newTask = '';
        };

        $scope.editTask = function(task) {
            task.editing = !task.editing;
        };

        $scope.deleteTask = function(task) {
            var index = $scope.tasks.indexOf(task);
            $scope.tasks.splice(index, 1);
        };
    });
</script>

</body>

```



&lt;/html&gt;

## Lab6 - To-Do List

Tasks	Operations
Revise DBMS	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Practice Angular Programs	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Practice SQL Queries	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Design one page of DBMS Project	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

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

```
<!DOCTYPE html>
<html>
  <head>
    <title>User Management</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.m
in.js"></script>
    <style>
      table {
        width: 100%;
        border-collapse: collapse;
      }
      th, td {
        border: 1px solid #ddd;
        padding: 8px;
        text-align: left;
      }
      th {
        background-color: #f2f2f2;
      }
      button {
        cursor: pointer;
      }
    </style>
  </head>
  <body ng-app="lab7">
    <div ng-controller="ctrl7">
      <h2>Lab 7 - CRUD Operations - User Management</h2>
      <input type="text" ng-model="newUser.name"
        placeholder="Name">
      <input type="email" ng-model="newUser.email"
        placeholder="Email">
      <button type="submit" ng-click="addUser()">
        Add User
      </button>
    </div>
  </body>
</html>
```

```

<table>
  <tr>
    <th>Name</th>
    <th>Email</th>
    <th>Action</th>
  </tr>
  <tr ng-repeat="user in users">
    <td><span ng-show="!user.editing">{{user.name}}
</span>

    <input type="text"
      ng-model="user.name"
      ng-show="user.editing"></td>
    <td><span ng-show="!user.editing">{{user.email}}
</span>

    <input type="text"
      ng-model="user.email"
      ng-show="user.editing"></td>
    <td>
      <button ng-click="editUser(user)"> Edit
</button>

      <button ng-click="deleteUser(user)"> Delete
</button>

    </td>
  </tr>
</table>
</div>

<script>
var lab7 = angular.module('lab7', []);
lab7.controller('ctrl17', function ($scope) {
  $scope.users = [
    {id: 1, name: 'Rashi', email: 'rashi@rymec.com' },
    {id: 2, name: 'Sai Reddy',
      email: 'sreddy@rymec.com' },
    { id: 3, name: 'Sandhya KR',
      email: 'sandykr@rymec.com' }
  ];
  $scope.newUser = {};

```

```

$scope.addUser = function () {
    $scope.newUser.id = $scope.users.length + 1;
    $scope.users.push(angular.copy($scope.newUser));
    $scope.newUser = {};
};
$scope.editUser = function (user) {
    user.editing = !user.editing;
};
$scope.deleteUser = function (user) {
    var index = $scope.users.indexOf(user);
    $scope.users.splice(index, 1);
};
});
</script>
</body>
</html>

```

#### Lab 7 - CRUD Operations - User Management

<input type="text" value="Name"/>	<input type="text" value="Email"/>	<input type="button" value="Add User"/>
Name	Email	Action
Rashi	rashi@rymec.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
<input type="text" value="Sai Reddy"/>	<input type="text" value="sreddy@rymec.com"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Sandhya KR	sandykr@rymec.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
rymec	rymec@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

8. Develop Angular-JS program to create a login form, with validation for the username and password fields.

```
<!DOCTYPE html>
<html>
<head>
  <title>Login Form</title>
  <script src="angular.min.js"></script>
  <style>
    .error {
      color: red;
    }
    #one {
      margin: auto;
      padding: 10px;
      width: 50%;
      text-align: center;
      border: 1px solid black;
    }
    input, button{
      margin:10px;
    }
  </style>
</head>

<body ng-app="lab8">

  <div id ="one" ng-controller="ctrl18">
    <h2>Lab 8 - Login Validation</h2>
    <form name="loginForm" novalidate>
      <div>
        <label>Username:</label>
        <input type="text"
          ng-model="username"
          name="username"
          required>
        <br />
        <span class="error"
          ng-show="loginForm.username.$touched &&
            loginForm.username.$error.required">
          Username is required
        </span>
      </div>
      <div>
        <label>Password:</label>
        <input type="password"
          ng-model=" password">
      </div>
    </form>
  </div>
```

```

        name="password"
        required>
        <br />
        <span class="error"
            ng-show="loginForm.password.$touched &&
            loginForm.password.$error.required">
            Password is required
        </span>
    </div>
    <button type="submit"
        ng-disabled="loginForm.$invalid"
        ng-click="login()"> Login </button>

</form>
</div>

<script>
    angular.module('lab8', [])
        .controller('ctrl8', function ($scope) {
            $scope.login = function () {
                if($scope.username == "angular" &&
                    $scope.password == "ang@123")
                    alert("Login successful!");
                else
                    alert("Login Unsuccessful!!! Username and
                        password doesn't match!!");
            };
        });
</script>
</body>
</html>

```

### Lab 8 - Login Validation

Username:

Username is required

Password:

Password is required

## ACADEMIC YEAR -2023-24

This page says

Login successful!

OK

Password: .....

Login

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>
  <head>
    <title> Employees Application </title>
    <script type = "text/javascript"
src="angular.min.js"></script>
    <style>
      div {
        margin: auto;
        padding: 10px;
        width: 50%;
        text-align: center;
        border: 1px solid black;
      }
      table, th, td {
        border: 2px solid black;
        border-collapse: collapse;
        text-align: center;
        align-self: stretch;
        margin:auto;
        padding: 10px;
      }
    </style>
  </head>
  <body ng-app='lab9'>
    <div ng-controller="ctrl9">
      <H2> Lab 9 - Employee App using Angular Filters</H2>
      <input type="text"
        placeholder="Search by name"
        ng-model="searchText.name" />
      <br /> <br />
      <input type="text"
        placeholder="Search by salary"
        ng-model="searchText.salary" />
      <br /> <br />
      <table>
        <tr>
          <th>Name</th>
```



```

        <th>Gender</th>
        <th>Salary</th>
        <th>City</th>
    </tr>
    <tr ng-repeat="e in employees | filter:searchText">
        <td>{{e.name}}</td>
        <td>{{e.gender}}</td>
        <td>{{e.salary}}</td>
        <td>{{e.city}}</td>
    </tr>
</table>
</div>
<script type="text/javascript">
    var lab9 = angular.module("lab9", []);
    lab9.controller("ctrl9", function($scope) {
        $scope.employees = [
            { name: "Ram", gender: "Male", salary: 55500.00,
city: "Bangalore" },
            { name: "Lakshman", gender: "Male", salary:
56000.00, city: "Delhi" },
            { name: "Seetha", gender: "Female", salary:
55500.00, city: "Hydrabad" },
            { name: "Ram", gender: "Male", salary: 55500.00,
city: "Delhi" },
            { name: "Ram", gender: "Male", salary: 44000.00,
city: "Mangalore" }
        ];
    });
</script>
</body>
</html>

```

**Lab 9 - Employee App using Angular Filters**

Name	Gender	Salary	City
Ram	Male	55500	Bangalore
Lakshman	Male	56000	Delhi
Seetha	Female	55500	Hydrabad
Ram	Male	55500	Delhi
Ram	Male	44000	Mangalore

**Lab 9 - Employee App using Angular Filters**

Name	Gender	Salary	City
Ram	Male	55500	Bangalore
Ram	Male	55500	Delhi
Ram	Male	44000	Mangalore

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>
<head>
  <title>To-Do List App</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.m
in.js"></script>
  <style>
    div {
      margin: auto;
      padding: 10px;
      width: 50%;
      text-align: center;
      border: 1px solid black;
    }

    table, th, td {
      border: 2px solid black;
      border-collapse: collapse;
      text-align: center;
      margin: auto;
      padding: 8px 20px;
      font-size: 20px;
    }
  </style>
</head>
<body ng-app="lab10">
  <div ng-controller="cntr10">
    <h1>Lab10 - List of Items</h1>
    <p>
      <input type="text"
        ng-model="newItem"
        placeholder="Add new Item"
        required>
      <button type="submit">
```

```

        ng-click="addItem()">Add Item</button>
    </p>
    <table>
        <tr>
            <th>Item</th>
            <th>Operations</th>
        </tr>
        <tr ng-repeat="item in items">
            <td> {{ item }} </td>
            <td>
                <button ng-click="deleteItem(item)">
Delete</button>
            </td>
        </tr>
    </table>
</div>

<script>
    var lab10 = angular.module('lab10', [])
    lab10.controller('cntr10', function ($scope) {
        $scope.items = [
            "Books", "Chairs", "Tables", "Fan", "AC", "TV"
        ];

        $scope.addItem = function () {
            $scope.items.push($scope.newItem);
            $scope.newItem = '';
        };

        $scope.deleteItem = function (item) {
            var index = $scope.items.indexOf(item);
            $scope.items.splice(index, 1);
        };
    });
</script>
</body>
</html>

```

## Lab10 - List of Items

Item	Operations
Books	<input type="button" value="Delete"/>
Chairs	<input type="button" value="Delete"/>
Tables	<input type="button" value="Delete"/>
Fan	<input type="button" value="Delete"/>
AC	<input type="button" value="Delete"/>
TV	<input type="button" value="Delete"/>

## Lab10 - List of Items

Item	Operations
Books	<input type="button" value="Delete"/>
Chairs	<input type="button" value="Delete"/>
Tables	<input type="button" value="Delete"/>
Fan	<input type="button" value="Delete"/>
AC	<input type="button" value="Delete"/>
TV	<input type="button" value="Delete"/>
Sofa	<input type="button" value="Delete"/>

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>
  <head>
    <title>Lab 11 - Student list</title>
    <style>
      #one {
        margin: auto;
        padding: 10px;
        width: 50%;
        border: 1px solid black;
      }
      ul, li{
        margin: 3px;
        font-size: 25px;
      }
      h1{
        text-align: center;
      }
    </style>
  </head>

  <body ng-app="lab11"
    ng-controller="con11">
    <div id="one">
      <h1>Lab 11 - Angular using String Filters</h1>
      <h2><p>Display in Uppercase: <input type="checkbox"
        ng-model="upper"></p>
      Student List:</h2>
      <UL ng-show="upper">
        <li ng-repeat="stu in students">{{stu | uppercase}}
        </li>
      </UL>
      <UL ng-hide="upper">
        <li ng-repeat="stu in students">{{stu | lowercase}}
        </li>
      </UL>
    </div>
  </body>
</html>
```

```
</div>
<script type="text/javascript"
      src="https://ajax.googleapis.com/ajax/libs/angularjs
/1.8.2/angular.min.js">
</script>
<script type="text/javascript">
  var lab11 = angular.module('lab11', []);
  lab11.controller("con11", function ($scope) {
    $scope.students = [ "Isaac Newton",
                        "Albert Einstein",
                        "Charles Darwin",
                        "Alexander Graham Bell"
                      ];
  });
</script>
</body>
</html>
```

## Lab 11 - Angular using String Filters

Display in Uppercase: ☐

Student List:

- isaac newton
- albert einstein
- charles darwin
- alexander graham bell

## **Lab 11 - Angular using String Filters**

**Display in Uppercase:** ☒

**Student List:**

- ISAAC NEWTON
- ALBERT EINSTEIN
- CHARLES DARWIN
- ALEXANDER GRAHAM BELL



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

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>AngularJS Date Filter Example</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.m
in.js"></script>
    <style>
      div {
        margin: auto;
        padding: 10px;
        width: 50%;
        text-align: center;
        border: 1px solid black;
      }
    </style>
  </head>

  <body ng-app="lab12">

    <div ng-controller="cnt12">
      <h1 style="color:red">Lab 12 - Date Filter Example</h1>
      <h3>Select the Format:
        <select ng-model="dateFilter">
          <option value="yyyy-MM-dd">yyyy-MM-dd</option>
          <option value="dd/MM/yyyy">dd/mm/yyyy</option>
          <option value="dd/MMM/yyyy">dd-mon-yyyy</option>
          <option value="MMMM dd, yyyy">Month dd, yyyy
          </option>
          <option value="EEEE - MMM dd, yyyy">Day - Month
            dd, yyyy </option>
          <option value="short">Short Date</option>
          <option value="medium">Medium Date</option>
        </select>
      </h3>
      <h2>
        <p>Current Date:{{ currentDate | date : dateFilter}}
        </p>
      </h2>
    </div>

    <script>
      var lab12 = angular.module('lab12', [])
      lab12.controller('cnt12', function ($scope) {
        $scope.currentDate = new Date();

```

```
});  
</script>  
</body>  
</html>
```

## Lab 12 - Date Filter Example

Select the Format:

**Current Date: 2024-02-20**

## Lab 12 - Date Filter Example

Select the Format:

**Current Date: Feb 20, 2024 4:44:41 PM**