Vijayanagar Educational Trust ®
EAST WEST COLLEGE OF ENGINEERING
(Affiliated to VTU, Belagavi, approved by AICTE, New Delhi, Recognized by Govt. of Karnataka)
13, 13th 'A' Main, Sector A, Yelahanka new Town, Bengaluru – 560064

Department of Computer Science and Engineering

LAB MANUAL

on

AngularJS (21CSL581)

V Semester
by

Prof. Shilpasree S
Assistant Professor

Prof. Pooja M V
Assistant Professor,

Computer Science and Engineering Department,
East West College of Engineering, Yelahanka New Town, Bangalore - 560064

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		

Course objectives:

- To learn the basics of AngularJS framework.
- To understand the AngularJS Modules, Forms, inputs, expression, data bindings and Filters
- To gain experience of modern tool usage in developing Web applications

Sl.NO	Experiments
1	Develop AngularJS 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	DevelopasimpleAngularJScalculatorapplicationthatcanperformbasicmathematicaloperations (addition, subtraction, multiplication, division) based on user input.
4	Write an AngularJS application that can calculate factorial and compute square based on given user input.
5	Develop AngularJS program to create a login form, with validation for the username and password fields.
6	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 maybe included in the program.
7	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 the mas needed. **Note**: The default values for items may be included in the program.
8	CreateAngularJSapplicationtoconvertstudentdetailstoUppercaseusingangularfilters. **Note**: The default details of students may be included in the program.
9	Create an AngularJS application that displays the date by using date filter parameters

NOTE: Include necessary HTML elements and CSS for the above Angular applications.

Course outcomes (Course Skill Set):

At the end of the course the student will be able to:

1. Develop AngularJS programs using basic features
2. Develop dynamic Web applications using AngularJS modules
3. Make use of form validations and controls for interactive applications
4. Apply the concepts of Expressions, data bindings and filters in developing AngularJS programs
5. Make use of modern tools to develop Web applications

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the **maximum** marks (20 marks).A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each course. The student has to secure not less than 35% (18 Marks out of 50) in the semester-end examination (SEE). The student has to secure a minimum of 40%(40marks out of 100) in the sum total of the CIE(Continuous Internal Evaluation) and SEE(Semester End Examination)taken together.

Continuous Internal Evaluation (CIE):

CIE marks for the practical course is **50Marks.**

The split-up of CIE marks for record / journal and test are in the ratio **60:40.**

- Each experiment to be evaluated for conduction with observation sheet and record write-up. Rubrics forthe evaluation of the journal / write- up for hardware/ software experiments designed by the faculty who is handling the laboratory session and is made known to students at the beginning of the practical session.
- Record should contain all the specified experiments in the syllabus and each experiment write-up willbe evaluated for10marks.
- Total marks scored by the students are scaled downed to 30marks(60% of maximum marks).
- Weightage to be given for neatness and submission of record/write-upon time.
- Department shall conduct 02 tests for 100 marks, the first test shall be conducted after the 8th week ofthe semester and the second test shall be conducted after the14th week of the semester.
- In each test, test write- up, conduction of experiment, acceptable result, and procedural knowledge will carry a weightage of 60%and the rest 40% for viva-voce.
- The suitable rubrics can be designed to evaluate each student's performance and learning ability .Rubrics suggested in Annexure-II of Regulation book
- The average of 02tests is scaled down to **20marks**(40% of **the maximum** marks). The Sum of scaled-down marks scored in the report write-up/journal and average marks of two tests is the total CIE marks scored by the student.

Semester End Evaluation (SEE):

- SEE marks for the practical course is 50Marks.
- SEE shall be conducted jointly by the two examiners of the same institute, examiners are appointed by the University
- All laboratory experiments are to be included for practical examination.

- (Rubrics) Breakup of marks and the instructions printed on the cover page of the answer script to be strictly adhered to by the examiners. OR based on the course requirement evaluation rubrics shall be decided jointly by examiners.
- Students can pick one question (experiment) from the questions lot prepared by the internal/external examiners jointly.
- Evaluation of test write-up /conduction procedure and result/ viva will be conducted jointly by examiners.
- General rubrics suggested for SEE are mentioned here, write up -20%, Conduction procedure and result in - 60%, Viva-voce 20% of maximum marks. SEE for practical shall be evaluated for 100 marks and scored marks shall be scaled down to 50 marks (however, based on course type, rubrics shall be decided by the examiners)
- The duration of SEE is02hours

Rubrics suggested in Annexure-II of Regulation book

Suggested Learning Resources:

Textbooks

1. ShyamSeshadri,BradGreen—"AngularJS:Up and Running: Enhanced Productivity with Structured Web Apps", A press, 0'ReillyMedia,Inc.
2. AgusKurniawan–"AngularJSProgrammingbyExample",FirstEdition,PEPress,2014

Weblinks and Video Lectures(e-Resources):

1. IntroductiontoAngularJS:https://www.youtube.com/watch?v=HEbphzK-0xE
2. AngularJS Modules :https://www.youtube.com/watch?v=gWm0KmgnQkU
3. https://www.youtube.com/watch?v=zKkUN-mJtPQ
4. https://www.youtube.com/watch?v=ICl7_i2mtZA
5. https://www.youtube.com/watch?v=Y2Few_nkze0
6. https://www.youtube.com/watch?v=QoptnVCQHsU

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- Demonstration of simple projects/applications (course project)

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

```
<html ng-app="nameApp">
<head>
    <title>AngularJS Full Name Example</title>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.0/angular.min.js"></script>
</head>
<body>
    <div ng-controller="nameCtrl">
        <!-- Input fields for first name and last name -->First Name:
        <input type="text" ng-model="firstName" placeholder="Enter your first name">
        <br>  <br>
        Last Name:
        <input type="text" ng-model="lastName" placeholder="Enter your last name">
        <br> <br>
        <!-- Button to display the full name -->
        <button ng-click="displayFullName()">Display Full Name</button>

        <!-- Display the full name -->
        <h1>Full Name is: {{ fullName }}</h1>
    </div>

    <script>
        angular.module('nameApp', [])
            .controller('nameCtrl', function ($scope) {
                // Default values for first name and last name
                $scope.firstName = 'Raj';
                $scope.lastName = 'Kumar';

                // Function to display the full name
                $scope.displayFullName = function () {
                    $scope.fullName = $scope.firstName + ' ' + $scope.lastName;
                };
            });
    </script>
</body>
```

```
</html>
```

Sample Output:

First Name: Raj

Last Name: Kumar

Display Full Name

Full Name: Raj Kumar

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 ng-app="shoppingApp">
<head>
    <title>AngularJS Shopping List</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.0/angular.min.js"></script>
</head>
<body ng-controller="shoppingCtrl">
    <h2>Shopping List</h2>
    <!-- Display the items in list
    <ul>
        <li ng-repeat="item in shoppingItems">{{ item }}  
            <button ng-click="removeItem($index)">Remove</button>
</li>
</ul> -->
    <table>
        <tr ng-repeat="item in shoppingItems">
            <td>{{ item }}</td>
            <td><button ng-click="removeItem($index)">Remove</button></td>
</tr>
</table>
    <!-- Input field and button to add a new item -->
    <input type="text" ng-model="newItem" placeholder="Add a new item">
    <button ng-click="addItem()">Add Item</button>
```

```
    <script>
        angular.module('shoppingApp', [])
            .controller('shoppingCtrl', function ($scope) {
                // Default values for shopping items
                $scope.shoppingItems = ['Apples', 'Bananas', 'Bread', 'Milk'];

                // Function to add a new item
                $scope.addItem = function () { if
                    ($scope.newItem) {
                        $scope.shoppingItems.push($scope.newItem);
                        $scope.newItem = ''; // Clear the input field after adding
                    }
                };

                // Function to remove an item
                $scope.removeItem = function (index) {
                    $scope.shoppingItems.splice(index, 1);
                };
            });
    </script>
</body>
</html>
```

Sample Output:

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

```
<html ng-app="calculatorApp">
<head>
  <title>AngularJS Calculator</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="calculatorController">
```

```
<h2>Simple Calculator</h2>
Enter Number 1:

<input type="number" ng-model="num1" />  
Select Operator:

<select ng-model="operator">

   <option value="+">Add</option><option value="-">Subtract</option>

   <option value="*">Multiply</option>
   <option value="/">Divide</option>

</select> 
Enter Number 2:

<input type="number" ng-model="num2" />

<button ng-click="calculate()">Calculate</button>
<p ng-show="result !== undefined">Result: {{ result }}</p>

<script>

   Var app = angular.module('calculatorApp', []);
   app.controller('calculatorController', function ($scope) {

      $scope.calculate = function () {switch
         ($scope.operator) {

            case '+':

               $scope.result = $scope.num1 + $scope.num2;break;

            case '-':
               $scope.result = $scope.num1 - $scope.num2;break;

            case '*':

               $scope.result = $scope.num1 * $scope.num2;break;
            case '/':

               if ($scope.num2 !== 0) {

                  $scope.result = $scope.num1 / $scope.num2;

               } else {

                  $scope.result = 'Cannot divide by zero';
               }

               break;


            }

         };
   });
</script>

</body>
```

```
</html>
```

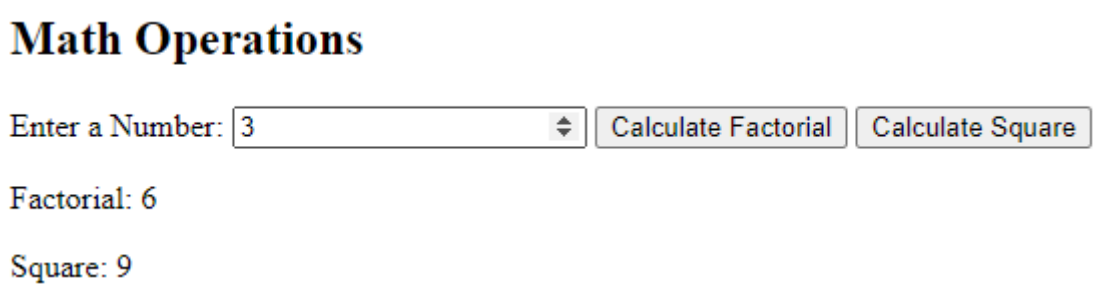
Sample Output:

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

```
<html ng-app="mathApp">
<head>
  <title>AngularJS Math Operations</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>

<body ng-controller="mathController">
  <h2>Math  Operations</h2>
  Enter a Number:
  <input type="number" ng-model="inputNumber" />
  <button ng-click="calculateFactorial()">Calculate Factorial</button>
  <button ng-click="calculateSquare()">Calculate Square</button>

  <p ng-show="factorialResult !== undefined">Factorial: {{ factorialResult }}</p>
  <p ng-show="squareResult !== undefined">Square: {{ squareResult }}</p>
  <script>
    var app = angular.module('mathApp', []);
    app.controller('mathController', function ($scope) {
      $scope.calculateFactorial = function () { if
        ($scope.inputNumber >= 0) {
          $scope.factorialResult = factorial($scope.inputNumber);
        } else {
          $scope.factorialResult = 'Cannot calculate factorial for negative numbers';
        }
      };

      $scope.calculateSquare = function () {
        $scope.squareResult = $scope.inputNumber * $scope.inputNumber;
      };
```

```
            function factorial(n) { if (n ==
               0 || n == 1) {
                  return 1;
               } else {
                  return n * factorial(n - 1);
               }
}
});
    </script>
  </body>
  </html>
```

Sample Output:

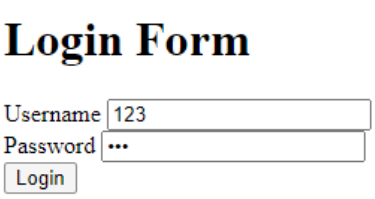


5.DevelopAngularJS program to create a login form, with validation for the username and password fields.

```
<html ng-app="loginApp">
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<body ng-controller="loginController">
    <h1>Login Form</h1>

    <!-- Form for login with validation -->
    <form ng-submit="login()">
```

```
    Username
        <input type="text" ng-model="username" required>
        <br>
        Password
        <input type="password" ng-model="password" required>
        <br>
        <button type="submit">Login</button>
    </form>
    <script>
        var app = angular.module('loginApp', []);
        app.controller('loginController', function ($scope) {
            $scope.login = function () {

            // Check if username is "Ram" and password is "Ram"
            if ($scope.username == 'ram' && $scope.password == 'ram') {alert('Login
                successful');
                // Add further logic for successful login
            } else {

                alert('Login failed. Invalid username or password.');
                // Add logic for failed login

            }
        };
    });
    </script>

</body>
</html>
```

Sample Output:

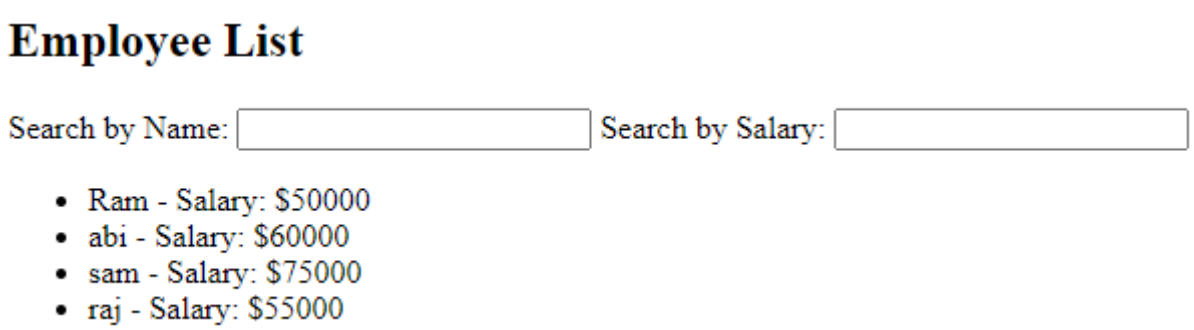


6.Create an AngularJS application that displays a list of employees and their salaries. Allow users to searchfor employees by name and salary. Note: Employee details may be included in the program.

```
<html ng-app="employeeApp">
<head>
   <title>AngularJS Employee Search</title>
   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="employeeController">
   <h2>Employee    List</h2>
Search by Name:
   <input type="text" ng-model="searchName" />

Search by Salary:
   <input type="number" ng-model="searchSalary" />
   <ul>
      <li ng-repeat="employee in employees | filter: {name: searchName, salary:searchSalary}">
         {{ employee.name }} - Salary: Rs{{ employee.salary }}
      </li>
   </ul>

   <script>
      var app = angular.module('employeeApp', []);
      app.controller('employeeController', function ($scope) {
         $scope.employees = [
            { name: 'Ram', salary: 50000 },
            { name: 'abi', salary: 60000 },
            { name: 'sam', salary: 75000 },
            { name: 'raj', salary: 55000 }
         ];

         $scope.searchName = '';
         $scope.searchSalary = '';
      });
   </script>
</body>
```

```
</html>
```

Sample Output:

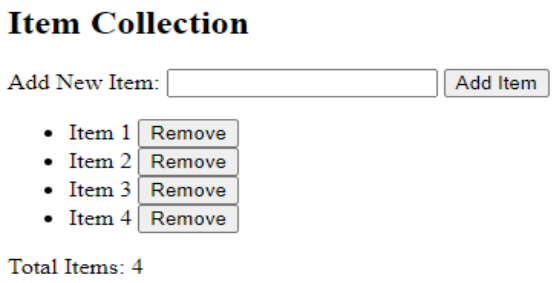


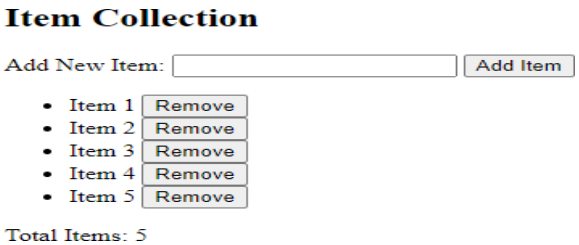
7.Create Angular JS application that allows users to maintain a collection of items. The application should display the current total number of items, and this count should automatically update as items are added or removed. Users should be able to add items to the collection and remove them as needed.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

```
<html ng-app="itemApp">
        <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

      <body ng-controller="itemController">
        <h2>Item Collection</h2>

        Add New Item:
        <input type="text" ng-model="newItem" />

        <button ng-click="addItem()">Add Item</button>

        <ul>
            <li ng-repeat="item in items track by $index">
              {{ item }}
              <button ng-click="removeItem($index)">Remove</button>
            </li>
</ul>
```

```
        <p>Total Items: {{ items.length }}</p>

        <script>
            var app = angular.module('itemApp', []);

            app.controller('itemController', function ($scope) {
              $scope.items = ['Item 1', 'Item 2', 'Item 3']; // Default items
              $scope.newItem = '';

              $scope.addItem = function () {
                  if ($scope.newItem) {
                    $scope.items.push($scope.newItem);
                    $scope.newItem = ''; // Clear the input field
                  }
              };

              $scope.removeItem = function (index) {
                  $scope.items.splice(index, 1);
};

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

Sample Output:





8. Create Angular JS application to convert student details to uppercase using angular filters. Note:The default details of students may be included in the program.

```
<html ng-app="studentApp">
                        <title>Student Name Converter</title>
                        <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

            <body ng-controller="studentController">

                        <h2>Student Names</h2>

                        <!-- Display the original student names -->
                        <h3>Original Names:</h3>
                        <ul>
                                    <li ng-repeat="name in names">
                                                {{ name }}
            </li>
        </ul>

        <!-- Display the student names in uppercase using filters -->
        <h3>Names in Uppercase:</h3>
        <ul>
<li ng-repeat="name in names">
  {{ name | uppercase }}
      </li>
  </ul>

        <script>
            var app = angular.module('studentApp', []);

            app.controller('studentController', function ($scope) {
                $scope.names = ['Raj', 'Ram', 'Sam'];
            });
        </script>
  </body>
  </html>
```

Sample Output:

Student Names

Original Names:

- Raj
- Ram
- Sam

Names in Uppercase:

- RAJ
- RAM
- SAM

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

```
<!DOCTYPE html>
<html ng-app="dateApp">
<head>
	<title>Date Display Application</title>
	<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="dateController">

	<h2>Date Display</h2>

	<!-- Display the current date with various filter parameters -->
	<p>Default Format: {{ currentDate | date }}</p>
	<p>Custom Format (yyyy-MM-dd): {{ currentDate | date:'yyyy-MM-dd' }}</p>
	<p>Short Date: {{ currentDate | date:'shortDate' }}</p>
	<p>Full Date: {{ currentDate | date:'fullDate' }}</p>

	<script>
		var app = angular.module('dateApp', []);
		app.controller('dateController', function ($scope) {
			$scope.currentDate = new Date();
		});
	</script>
</body>
</html>
```

Sample Output:

Date Display

Default Format: Nov 22, 2023

Custom Format (yyyy-MM-dd): 2023-11-22

Short Date: 11/22/23

Full Date: Wednesday, November 22, 2023