THE OXFORD COLLEGE OF ENGINEERING

Hosur Road, Bommanahalli, Bengaluru-560068

2023-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SUBJECT NAME/CODE: ANGULAR JS (21CSL581)

SCHEME : 2021 Scheme

SEMESTER : V Sem –A, B&C sec

INTRODUCTION

AngularJS is a JavaScript framework. It can be added to an HTML page with a <script> tag. AngularJS extends HTML attributes with Directives, and binds data to HTML with Expressions. AngularJS is distributed as a JavaScript file, and can be added to a web page with a script tag:

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

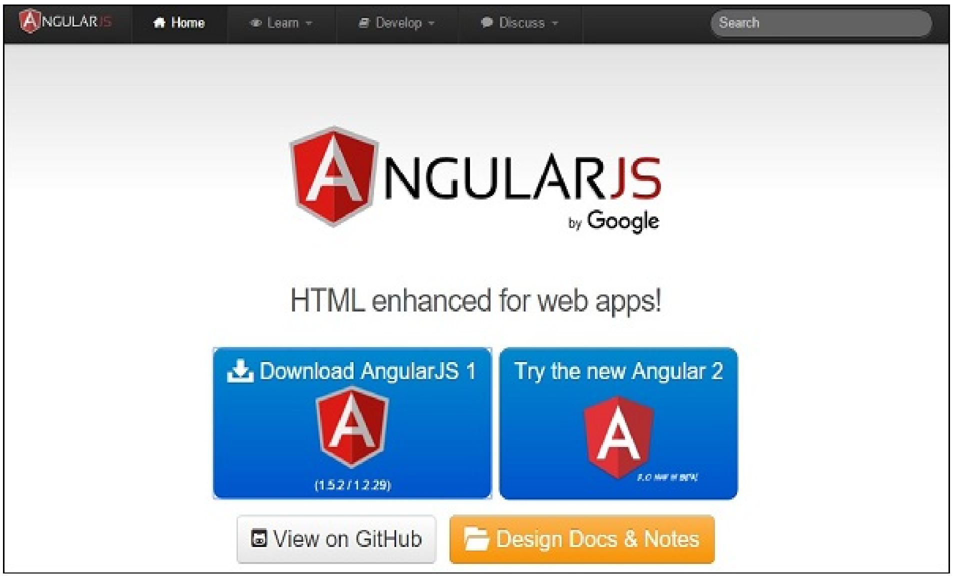
What Are the Basics Required to Learn AngularJS?

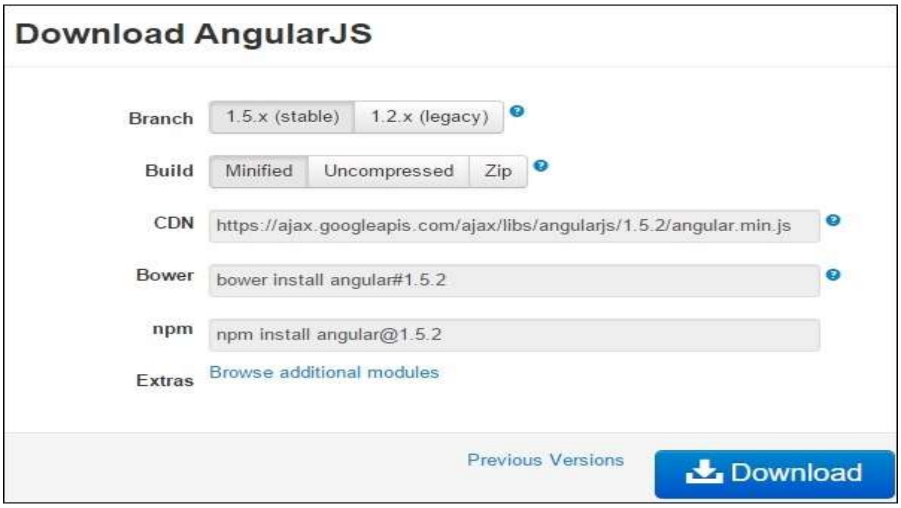
Overall, the following may ease the process of learning AngularJS for beginners:

- Moderate knowledge of HTML, CSS, and JavaScript
- Basic Model-View-Controller (MVC) concepts
- The Document Object Model (DOM)
- JavaScript functions, events, and error handling.

AngularJS - Environment Setup

- Open the link https://angularjs.org/, you will see there are two options to download AngularJS library –





Download AngularJS

Branch: 1.5.x (stable) | 1.2.x (legacy)

Build: Minified | Uncompressed | Zip

CDN: https://ajax.googleapis.com/ajax/libs/angularjs/1.5.2/angular.min.js

Bower: bower install angular#1.5.2

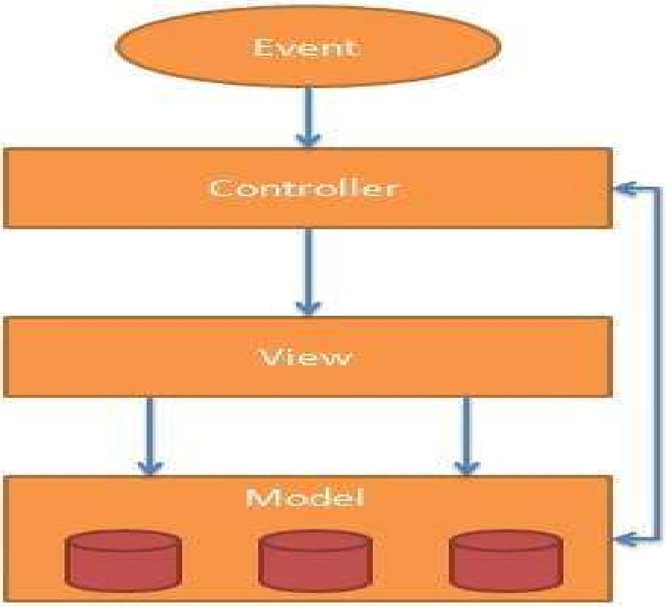
npm: npm install angular@1.5.2

Extras: Browse additional modules

AngularJS - MVC Architecture

Model **V**iew **C**ontroller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts −

- Model − It is the lowest level of the pattern responsible for maintaining data.
- View − It is responsible for displaying all or a portion of the data to the user.
- Controller − It is a software Code that controls the interactions between the Model and View.



Creating AngularJS Application

Step 1: Load framework
Being a pure JavaScript framework, it can be added using <Script> tag.

```
<script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">
</script>
```

Step 2: Define AngularJS application using ng-app directive

```
<div ng-app = "">
...
</div>
```

Step 3: Define a model name using ng-model directive

```
<p>Enter your Name: <input type = "text" ng-model = "name"></p>
```

Step 4: Bind the value of above model defined using ng-bind directive

```
<p>Hello <span ng-bind = "name"></span>!</p>
```

AngularJS – Directives

AngularJS directives are used to extend HTML. They are special attributes starting with ng-prefix.

- ng-app − This directive starts an AngularJS Application.
- ng-init − This directive initializes application data.
- ng-model − This directive defines the model that is variable to be used in AngularJS.
- ng-repeat − This directive repeats HTML elements for each item in a collection.

AngularJS – Expressions

Expressions are used to bind application data to HTML. Expressions are written inside double curly braces such as in {{ expression}}.
Expressions behave similar to ngbind directives. AngularJS expressions are pure JavaScript expressions and output the data where they are used.

AngularJS – Controllers

AngularJS application mainly relies on controllers to control the flow of data in the application. A controller is defined using *ng-controller* directive. A controller is a JavaScript object that contains attributes/properties, and functions. Each controller accepts $scope as a parameter, which refers to the application/module that the controller needs to handle.

```
<div ng-app = "" ng-controller = "studentController">
...
</div>
```

AngularJS - Filters

Filters are used to modify the data. They can be clubbed in expression or directives using pipe (|) character. The following list shows the commonly used filters.

Sr.No.	Name & Description
1	**uppercase** converts a text to upper case text.
2	**lowercase** converts a text to lower case text.
3	**currency** formats text in a currency format.
4	**filter** filter the array to a subset of it based on provided criteria.
5	**orderby** orders the array based on provided criteria.

Semester -V

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 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	DevelopAngularJS 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. **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**: 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

NOTE: Include necessary HTML elementsand 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 Angular JS programs using basic features
2. Develop dynamic Web applications using AngularJS modules
3. Make use of form validations and controls for interactive applications
4. Appy the concepts of Expressions, data bindings and filters in developing Angular JS programs
5. Make use of modern tools to develop Web applications

Updated as Practical Course on 03.11.2023

Semester -V

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% (40 marks out of 100) in the sum totaloftheCIE(ContinuousInternalEvaluation)andSEE (SemesterEndExamination)takentogether.

Continuous Internal Evaluation (CIE):

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

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 for the 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 practicalsession.
- Record should contain all the specified experiments in the syllabus and each experiment write-up will be evaluated for 10marks.
- Total marks scored by the students are scaled downed to 30 marks (60% of maximummarks).
- Weightage to be given for neatness and submission of record/write-up ontime.
- Department shall conduct 02 tests for 100 marks, the first test shall be conducted after the 8$^{t\wedge}$ week of the semester and the second test shall be conducted after the 14$^{'h}$ week of thesemester.
- In each test, test write-up, conduction of experiment, acceptable result, and procedural knowledge will carry a weightage of 60% and the rest 40% forviva-voce.
- The suitable rubrics can be designed to evaluate each student’s performance and learning ability. Rubrics suggested in Annexure-II of Regulationbook
- The average of 02 tests is scaled down to **20 marks** (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 theUniversity
- All laboratory experiments are to be included for practicalexamination.
- (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 byexaminers.
- Students can pick one question (experiment) from the questions lot prepared by theinternal/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 procedureand 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 bythe examiners)
- The duration of SEE is 02hours

Rubrics suggested in Annexure-lI of Regulation book

Updated as Practical Course on 03.11.2023

Suggested Learning Resources:

Textbooks

1. ShyamSeshadri, Brad Green —"AngularJS: Up and Running: Enhanced Productivity with Structured Web Apps", Apress, 0'Reilly Media,Inc.
2. AgusKurniawan–"AngularJS Programming by Example", First Edition, PE Press, 2014

Weblinks and Video Lectures (e-Resources):

1. Introduction to Angular JS :https://www.youtube.com/watch?v=HEbphzK-0xE
2. Angular JS 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 simpleprojects/applications (course project)

Updated as Practical Course on 03.11.2023

ANGULAR JS (21CSL581)

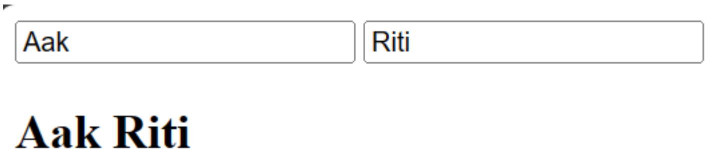
Q1. Develop Angular JS program that allows user to input their first name and last name and display their full name.

Note: The default values for first name and last name may be included in the program.

Program: -

```
<!DOCTYPE html>
<html>
<head>
    <!-- Link Your CSS If You Want  -->
    <link rel="stylesheet" href="Stylesheet.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-app="" ng-init="firstName='Aak'; lastName='Riti'">
    <input type="text" ng-model="firstName">
    <input type="text" ng-model="lastName">
    <h2>{{ firstName + ' ' + lastName }}</h2>
</body>
</html>
```

Output: -



Aak Riti

Aak Riti

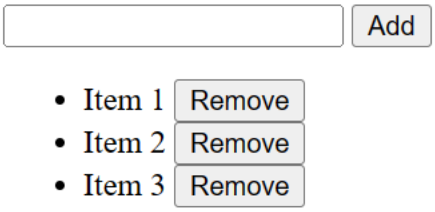
Q2. Develop an Angular JS application that displays a list of shopping items. Allow users to add and remove items from the list using directives and controllers.

Note: The default values of items may be included in the program.

Program: -

```
<!DOCTYPE html>
<html>
<head>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
    <script>
        angular.module('shoppingApp', []).controller('ShoppingController',
['$scope', function($scope) {
            $scope.items = ['Item 1', 'Item 2', 'Item 3'];
            $scope.updateItem = function(item) {
                item ? $scope.items.push(item) : $scope.items.pop();
                $scope.newItem = '';
            };
        }]);
    </script>
</head>
<body ng-app="shoppingApp" ng-controller="ShoppingController">
    <input ng-model="newItem">
    <button ng-click="updateItem(newItem)">Add</button>
    <ul>
        <li ng-repeat="item in items">{{ item }}
            <button ng-click="updateItem()">Remove</button>
        </li>
    </ul>
</body>
</html>
```

Output: -



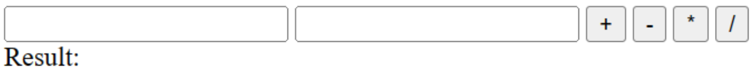
- Item 1 Remove
- Item 2 Remove
- Item 3 Remove

Q3. Develop a simple Angular JS calculator application that can perform basic mathematical operations (addition, subtraction, multiplication, division) based on user input.

Program: -

```
<!DOCTYPE html>
<html>
<head>
    <!-- Link Your CSS If You Want  -->
    <link rel="stylesheet" href="Stylesheet.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>
<body ng-app="">
    <input type="number" ng-model="num1">
    <input type="number" ng-model="num2">
    <button ng-click="result = num1 + num2">+</button>
    <button ng-click="result = num1 - num2">-</button>
    <button ng-click="result = num1 * num2">*</button>
    <button ng-click="result = num1 / num2">/</button>
    <div>Result: {{ result }}</div>
</body>
</html>
```

Output: -



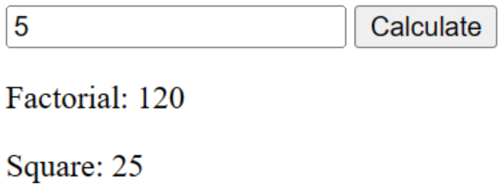
Result:

Q4. Write an Angular JS application that can calculate factorial and compute square based on given user input.

Program: -

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <!-- Link Your CSS If You Want  -->
    <link rel="stylesheet" href="Stylesheet.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
    <script>
        angular.module('myApp', []).controller('myCtrl', function($scope) {
            $scope.fact = function(n) {
                return n < 2 ? 1 : n * $scope.fact(n - 1);
            };
        });
    </script>
</head>
<body ng-controller="myCtrl">
    <input type="number" ng-model="n">
    <button ng-click="fr = fact(n); sr = n * n">Calculate</button>

    <p>Factorial: {{ fr }}</p>
    <p>Square: {{ sr }}</p>
</body>
</html>
```

Output: -



```
Factorial: 120

Square: 25
```

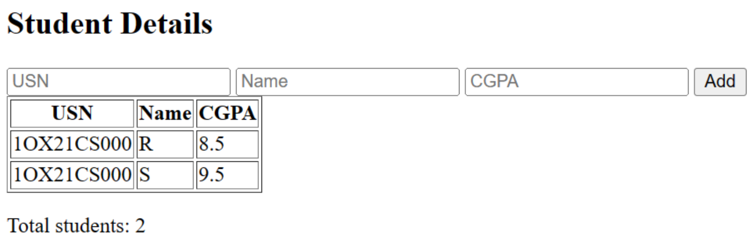
Q5. Develop AngularJS application that displays a detail of students and their CGPA. Allow users to read the number of students and display the count.

Note: Student details may be included in the program.

Program: -

```
<html>
<head>
    <!-- Link Your CSS If You Want  -->
    <!-- <link rel="stylesheet" href="Stylesheet.css">  -->
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.0/angular.min.js"></s
cript>
</head>
<body ng-app ng-init="s=[{ usn: '1OX21CS000', name: 'S', cgpa: 8.5 }]">
    <h2>Student Details</h2>
    <input ng-model="ns.usn" placeholder="USN">
    <input ng-model="ns.name" placeholder="Name">
    <input type="number" ng-model="ns.cgpa" placeholder="CGPA">
    <button ng-click="s.push(ns); ns={}">Add</button>
    <table border="1">
        <tr>
            <th>USN</th>
            <th>Name</th>
            <th>CGPA</th>
        </tr>
        <tr ng-repeat="st in s | orderBy:'usn'">
            <td>{{ st.usn }}</td>
            <td>{{ st.name }}</td>
            <td>{{ st.cgpa }}</td>
        </tr>
    </table>
    <p>Total students: {{ s.length }}</p>
</body>
</html>
```

Output: -



Student Details

USN	Name	CGPA
1OX21CS000	R	8.5
1OX21CS000	S	9.5

Total students: 2

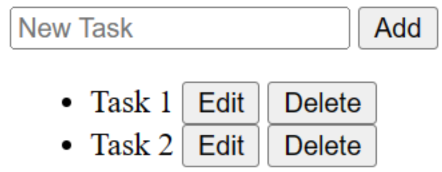
Q6. Develop an AngularJS program to create a simple to-do list application. Allow users to add, edit, and delete tasks.

Note: The default values for tasks may be included in the program.

Program: -

```
<!DOCTYPE html>
<html ng-app="">
<head>
    <!-- Link Your CSS If You Want  -->
    <!-- <link rel="stylesheet" href="Stylesheet.css">  -->
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>
<body ng-init="tasks=[{text:'Task 1', editing:false}, {text:'Task 2',
editing:false}]">
    <input ng-model="newTask" placeholder="New Task">
    <button ng-click="tasks.push({text: newTask, editing: false});
newTask=''">Add</button>
    <ul>
        <li ng-repeat="task in tasks">
            <span ng-hide="task.editing" ng-bind="task.text"></span>
            <input ng-show="task.editing" ng-model="task.text">
            <button ng-click="task.editing = !task.editing">{{task.editing ?
'Done' : 'Edit'}}</button>
            <button ng-click="tasks.splice($index, 1)">Delete</button>
        </li>
    </ul>
</body>
</html>
```

Output: -



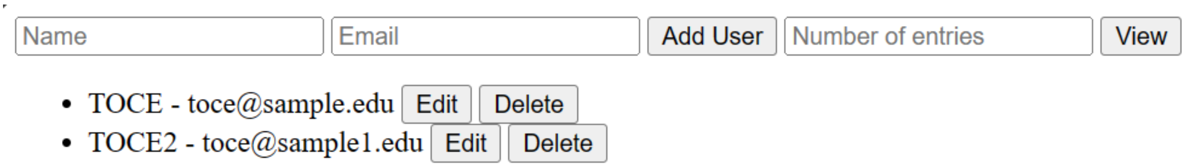
```
New Task  Add

• Task 1  Edit  Delete
• Task 2  Edit  Delete
```

Q7.Write an AngularJS program to create a simple CRUD application (Create, Read, Update, and Delete) for managing users.

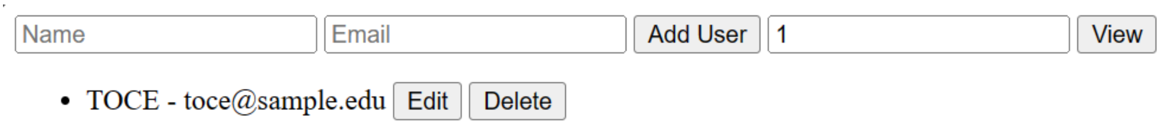
Program: -

```
<!DOCTYPE html>
<html lang="en" ng-app>
<head>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>
<body ng-init="uList=[]; nUser={}; editIndex=-1; numEntries=1">
    <input ng-model="nUser.name" placeholder="Name">
    <input ng-model="nUser.email" placeholder="Email">
    <button ng-click="uList.push({name: nUser.name, email: nUser.email});
nUser={}">Add User</button>
    <input type="number" ng-model="numEntries" placeholder="Number of
entries">
    <button ng-click="selectedEntries = numEntries">View</button>
    <ul>
        <li ng-repeat="user in uList | limitTo:selectedEntries">
            <span ng-if="editIndex !== $index">{{ user.name }} - {{ user.email
}}</span>
            <button ng-click="editIndex = editIndex === $index ? -1 : $index">
                {{ editIndex === $index ? 'Save' : 'Edit' }}
            </button>
            <button ng-click="uList.splice($index, 1)">Delete</button>
            <div ng-if="editIndex === $index">
                <input ng-model="user.name">
                <input ng-model="user.email">
            </div>
        </li>
    </ul>
</body>
</html>
```

Output: -



- TOCE - toce@sample.edu Edit Delete
- TOCE2 - toce@sample1.edu Edit Delete



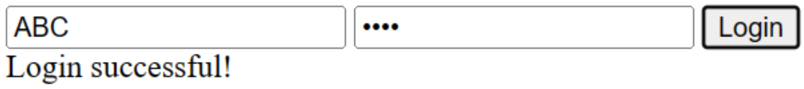
- TOCE - toce@sample.edu Edit Delete

Q8. Develop AngularJS program to create a login form, with validation for the username and password fields.

Program: -

```
<!DOCTYPE html>
<html>
<head>
    <!-- Link Your CSS If You Want  -->
    <!-- <link rel="stylesheet" href="Stylesheet.css">  -->
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>
<body ng-app>
    <form ng-submit="submitted = true">
        <input type="text" ng-model="user.username" placeholder="Username"
required>
        <input type="password" ng-model="user.password" placeholder="Password"
required>
        <button type="submit">Login</button>
    </form>
    <div ng-if="submitted">Login successful!</div>
</body>
</html>
```

Output: -



ABC

••••

Login successful!

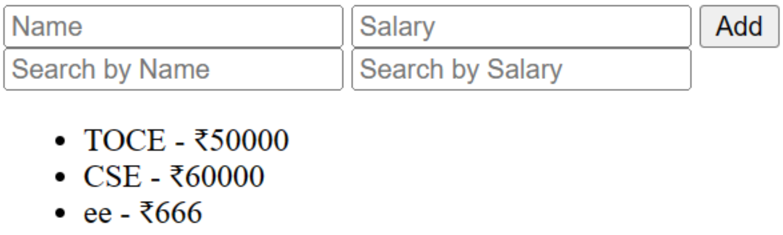
Q9. Create an AngularJS application that displays a list of employees and their salaries. Allow users to search for employees by name and salary.

Note: Employee details may be included in the program.

Program: -

```
<!DOCTYPE html>
<html ng-app="">
<head>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>
<body ng-init="emps=[{n: 'TOCE ', s: 50000}, {n: 'CSE', s: 60000}]">
    <form ng-submit="emps.push({n: n, s: s}); n=''; s=null">
        <input ng-model="n" placeholder="Name">
        <input type="number" ng-model="s" placeholder="Salary">
        <button type="submit">Add</button>
    </form>
    <input ng-model="sName" placeholder="Search by Name">
    <input ng-model="sSalary" type="number" placeholder="Search by Salary">
    <ul>
        <li ng-repeat="emp in emps | filter:{n: sName, s: (sSalary ||
undefined)}">
            {{ emp.n }} - ₹{{ emp.s }}
        </li>
    </ul>
</body>
</html>
```

Output: -



```
• TOCE - ₹50000
• CSE - ₹60000
• ee - ₹666
```

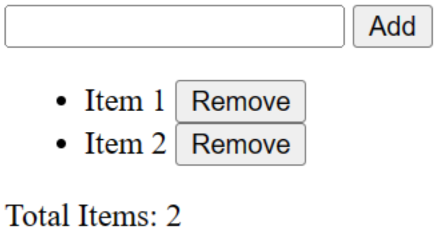
Q10. Create AngularJS application that allows users to maintain a collection of items. The application should display the current total number of items, and this count should automatically update as items are added or removed. Users should be able to add items to the collection and remove them as needed.

Note: The default values for items may be included in the program.

Program: -

```
<!DOCTYPE html>
<html ng-app>
<head>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>
<body ng-init="items=['Item 1', 'Item 2']">
    <input ng-model="newItem">
    <button ng-click="newItem && items.push(newItem); newItem =
'';">Add</button>
    <ul>
        <li ng-repeat="item in items">
            {{ item }}
            <button ng-click="items.splice($index, 1)">Remove</button>
        </li>
    </ul>
    Total Items: {{ items.length }}
</body>
</html>
```

Output: -



```
[            ] Add

  • Item 1 Remove
  • Item 2 Remove

Total Items: 2
```

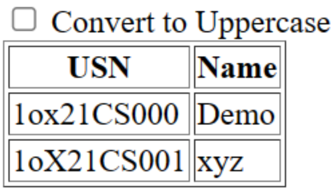
Q11. Create AngularJS application to convert student details to Uppercase using angular filters.

Note: The default details of students may be included in the program.

Program: -

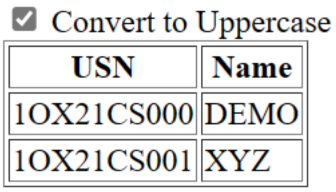
```
<!DOCTYPE html>
<html>
<head>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>
<body ng-app ng-init="students=[{usn:'1ox21CS000',
name:'Demo'},{usn:'1oX21CS001', name:'xyz'}]; uppercase=false">
    <label>
        <input type="checkbox" ng-model="uppercase"> Convert to Uppercase
    </label>
    <table border="1">
        <tr>
            <th>USN</th>
            <th>Name</th>
        </tr>
        <tr ng-repeat="student in students">
            <td>{{ uppercase ? (student.usn | uppercase) : student.usn }}</td>
            <td>{{ uppercase ? (student.name | uppercase) : student.name
}}</td>
        </tr>
    </table>
</body>
</html>
```

Output: -



□ Convert to Uppercase

USN	Name
1ox21CS000	Demo
1oX21CS001	xyz



☑ Convert to Uppercase

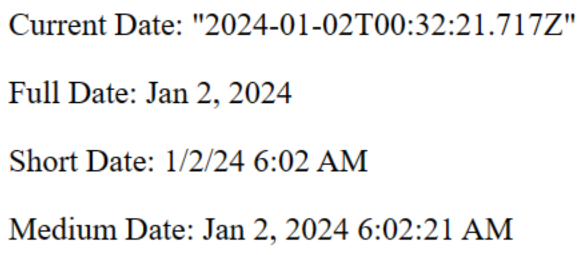
USN	Name
1OX21CS000	DEMO
1OX21CS001	XYZ

Q12. Create an AngularJS application that displays the date by using date filter parameters.

Program: -

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s
cript>
</head>
<body ng-controller="myController">
    <p>Current Date: {{ currentDate }}</p>
    <p>Full Date: {{ currentDate | date }}</p>
    <p>Short Date: {{ currentDate | date: 'short' }}</p>
    <p>Medium Date: {{ currentDate | date: 'medium' }}</p>
</body>
</html>
<script>
angular.module('myApp', []).controller('myController', function($scope) {
    $scope.currentDate = new Date();
});
</script>
```

Output: -



```
Current Date: "2024-01-02T00:32:21.717Z"

Full Date: Jan 2, 2024

Short Date: 1/2/24 6:02 AM

Medium Date: Jan 2, 2024 6:02:21 AM
```

VIVA QUESTIONS:

1. **What is AngularJS?**
   Ans: AngularJS is an **open-source JavaScript framework** used to build rich and extensible web applications. It is developed by Google and follows the MVC (Model View Controller) pattern.

2. **What are the main advantages of AngularJS?**
   Ans: Some of the main advantages of AngularJS are given below:
   - Allows us to create a **single page application**.
   - Follows **MVC** design pattern.
   - Predefined form validations.
   - Supports animations.
   - Open-source.
   - Cross-browser compliant.

3. **What are the disadvantages of AngularJS?**
   Ans:
   - JavaScript Dependent
   - Not Secured
   - Time Consumption in Old Devices
   - Difficult to Learn

4. **What are the features of AngularJS?**
   Ans: Some important features of AngularJS are given below:
   - MVC- In AngularJS, you just have to **split your application code** into MVC components, i.e., Model, View, and the Controller.
   - Validation- It performs client-side form validation.
   - Module- It defines an application.
   - Directive- It specifies behavior on the DOM element.
   - Template- It renders the dynamic view.
   - Scope- It joins the controller with the views.
   - Expression- It binds application data to HTML.

5. **Describe MVC in reference to angular.**
   Ans: AngularJS is based on MVC framework, where MVC stands for Model-View-Controller.

   MVCperforms the following operations:

- A **model** is the lowest level of the pattern responsible for **maintaining data**.
- **A controller** is responsible for a view that contains the logic to **manipulate that data**. It is basically a software code which is used for taking control of the interactions between the Model and View.
- A **view** is the HTML which is responsible for **displaying the data**.

For example, a $scope can be defined as a model, whereas the functions written in angular controller modifies the $scope and HTML displays the value of scope variable.

6. **What IDE's are currently used for the development of AngularJS?**
   - Eclipse
   - Visual Studio Code
   - WebStorm

7. **What are the controllers in AngularJS?**
   Ans: AngularJS **controllers** are used to **control the flow of data** of AngularJS application. A controller is defined using **ng-controller** directive.

8. **What are the uses of controllers in AngularJS?**
   Ans: AngularJS controllers are used for:
   - Setting the initial state of the $scope object
   - Adding behavior to the $scope object

9. **What is the module in AngularJS?**
   Ans: An AngularJS module defines an application. The module is a container for the different parts of an application. The module is a container for the application controllers. Controllers always belong to a module.

10. **What are the expressions in AngularJS?**
    Ans: AngularJS expressions are those that are written inside double braces {{expression}}. AngularJS evaluates the specified expression and binds the result data to HTML. AngularJS expressions can also be written inside a directive: ng-bind="expression".

11. **What are the key differences between Angular expressions and JavaScript expressions?**
    Ans: Unlike JavaScript expressions, AngularJS expressions can be written inside HTML. AngularJS expressions do not support conditionals, loops, and exceptions, while JavaScript expressions do. AngularJS expressions support filters, while JavaScript expressions do not.

12. **What are the different types of directives available in AngularJS?**

Ans: AngularJS provides support for creating custom directives for the following type of elements:

- **Element Directive** : Element directives are activated when a matching element is encountered.
- **Attribute**: Attribute directives are activated when a matching attribute is encountered.
- **CSS:** CSS directives are activated when a matching CSS style is encountered.
- **Comment**: Comment directives are activated when a matching comment is encountered.

13. **Explain ng-app directive.**

Ans: ng-app directive defines and links an AngularJS application to HTML. It also indicate the start of the application.

14. **What is $scope?**

Ans: A $scope is an object that represents the application model for an Angular application.Each AngularJS application can have only one root scope but can have multiple child scopes.

For example: var app = angular.module('myApp', []).app.controller('myCtrl', function($scope){
$scope.carname = "Volvo"; });

15. **What are the key features of $scope?**

Ans:

- It provides observers to check for all the model changes
- It provides the ability to propagate model changes through the application as well as outside the system to other associated components.
- Scopes can be nested in a way that they can isolate functionality and model properties.
- It provides an execution environment in which expressions are evaluated.