

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

Department of Artificial Intelligence and Machine Learning



Academic Year – 2022- 2023

Lab Manual

Angular JS and Node JS

Subject – code – 21CSL581

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND
MANAGEMENT

Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore-560 082

(Affiliated to Visvesvaraya Technological University, Belagavi and Approved by AICTE,

New Delhi) CE, CSE, ECE, EEE, ISE, ME Courses Accredited by NBA, New Delhi,

NAAC A+

2022-2023

Course	ANGULAR JS AND NODE JS		
Semester	5th Semester		
Course Code	21CSL581	CIE Marks	50
Teaching Hours/Week (L: T: P: S)	1:0:0:0	SEE Marks	50
Total Hours of Pedagogy	12T+12P	Total Marks	100
Credits	01	Exam Hours	02

5th Semester**Syllabus - ANGULAR JS AND NODE JS**

Module-1	
Introduction To Angular JS: Introduction – Features – Angular JS Model-View-Controller – Expression Directives and Controllers.	
Teaching-Learning Process	Chalk and board, Active Learning, practical based Learning
Module-2	
Angular JS Modules: Arrays –Working with ng-model – Working with Forms – Form Validation – Error Handling with Forms – Nested Forms with ng-form – Other Form Controls.	
Teaching-Learning Process	Chalk and board, Active Learning, Demonstration, presentation, problem solving
Module-3	
Directives& Building Databases: Part I- Filters – Using Filters in Controllers and Services – Angular JS Services – Internal Angular JS Services – Custom Angular JS Services	
Teaching-Learning Process	Chalk and board, Demonstration, problem solving
Module-4	
Directives& Building Databases: Part-II- Directives – Alternatives to Custom Directives – Understanding the Basic options – Interacting with Server –HTTP Services – Building Database, Front End and BackEnd	
Teaching-Learning Process	Chalk and board, Practical based learning, practical's

Module-5	
Introduction to NODE .JS: Introduction –Using the Terminals – Editors –Building a Webserver with Node – The HTTPModule – Views and Layouts.	
Teaching-Learning Process	Chalk and board, MOOC

Sl. No.	Name of the program
1	Implement AngularJS First Application to print the name using web browser
2	Implement the AngularJS Directives using web browser to print list of Countries with locate
3	Implement the AngularJS Expressions using web browser to print student name, marks, roll number and expense on books.
4	Implement the Angular JS Filters using web browser to print Name in Upper Case, Name in Lower Case, fees and Subjects.
5	Implement AngularJS - HTML DOM using web browser to Disable Button to Click Me! ,Show Button, Hide Button to Click Me! and to show total.
6	Implement Node.js to print hello world
7	Implement Node.js to get the current date and time
8	Implement the Node.js using web browser to display sentence.
9	Implement the Node.js using web browser to display My Header My paragraph.
10	Implement Node.js to creating new files: <ul style="list-style-type: none"> • fs.appendFile() • fs.open() • fs.writeFile()

1) Implement AngularJS First Application to print the name using a web browser

```
<!DOCTYPE html>
<html lang="en" ng-app="myApp">
<head>
  <meta charset="UTF-8">
  <title>AngularJS First Application</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="myController">
  <h1>Hello, {{ name }}</h1>
</div>

<script>
  angular.module('myApp', [])
    .controller('myController', function($scope) {
      $scope.name = 'John'; // Change 'John' to any name you want to display
    });
</script>

</body>
</html>
```

2) Implement the AngularJS Directives using web browser to print list of Countries with locate

```
<!DOCTYPE html>
```

```
<html lang="en" ng-app="myApp">
<head>
  <meta charset="UTF-8">
  <title>AngularJS Directives - Countries List</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="myController">
  <h1>Countries List with Continents</h1>
  <country-list countries="countryData"></country-list>
</div>

<script>
angular.module('myApp', [])
.controller('myController', function($scope) {
  $scope.countryData = [
    { name: 'USA', continent: 'North America' },
    { name: 'Canada', continent: 'North America' },
    { name: 'Brazil', continent: 'South America' },
    { name: 'France', continent: 'Europe' },
    { name: 'Germany', continent: 'Europe' },
    { name: 'Australia', continent: 'Oceania' },
    { name: 'China', continent: 'Asia' },
    { name: 'South Africa', continent: 'Africa' }
  ];
})
.directive('countryList', function() {
  return {
    restrict: 'E',
    scope: {
      countries: '='
    },
    template: `
    <ul>
      <li ng-repeat="country in countries">
        <strong>{{ country.name }}</strong> - {{ country.continent }}
      </li>
    </ul>
    `
  };
});
```

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

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

3. Implement the AngularJS Expressions using web browser to print student name, marks, roll number and expense on books.

```
<!DOCTYPE html>
<html lang="en" ng-app="myApp">
<head>
  <meta charset="UTF-8">
  <title>AngularJS Expressions - Student Information</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="myController">
  <h1>Student Information</h1>
  <p>Name: {{ student.name }}</p>
  <p>Roll Number: {{ student.rollNumber }}</p>
  <p>Marks: {{ student.marks }}</p>
  <p>Expense on Books: {{ student.expenseOnBooks | currency }}</p>
</div>

<script>
angular.module('myApp', [])
  .controller('myController', function($scope) {
    $scope.student = {
      name: 'Alice',
      rollNumber: '2023001',
      marks: 85,
      expenseOnBooks: 75.50 // Assume the expense is in dollars
    };
  });
</script>

</body>
</html>
```

4. Implement the Angular JS Filters using web browser

to print Name in Upper Case, Name in Lower Case, fees and Subjects.

```
<!DOCTYPE html>
<html lang="en" ng-app="myApp">
<head>
  <meta charset="UTF-8">
  <title>AngularJS Filters - Student Details</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="myController">
  <h1>Student Details</h1>

  <p>Name in Upper Case: {{ student.name | uppercase }}</p>
  <p>Name in Lower Case: {{ student.name | lowercase }}</p>

  <p>Fees: {{ student.fees | currency }}</p>

  <p>Subjects:
    <span ng-repeat="subject in student.subjects">
      {{ subject }} {{ $last ? " : ", ' ' }}
    </span>
  </p>
</div>

<script>
angular.module('myApp', [])
.controller('myController', function($scope) {
  $scope.student = {
    name: 'John Doe',
    fees: 250.75, // Assume fees are in dollars
    subjects: ['Mathematics', 'Science', 'History']
  };
});
</script>

</body>
</html>
```

5. Implement AngularJS - HTML DOM using web browser to Disable Button to Click Me!

,Show Button, Hide Button to Click Me! and to show total.

```
<!DOCTYPE html>

<html lang="en" ng-app="myApp">

<head>

  <meta charset="UTF-8">

  <title>AngularJS - Button Toggle and Total Display</title>

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

</head>

<body>

  <div ng-controller="myController">

    <h1>Button Toggle and Total Display</h1>

    <button ng-click="clickMe()" ng-disabled="disableClickMe">Click Me!</button>

    <button ng-click="showButton = true">Show Button</button>

    <button ng-click="showButton = false">Hide Button</button>

    <p ng-show="showTotal">Total: {{ total }}</p>

  </div>

  <script>

    angular.module('myApp', [])

      .controller('myController', function($scope) {

        $scope.disableClickMe = false;
```

```
$scope.showButton = true;

$scope.total = 0;


$scope.clickMe = function() {

    $scope.disableClickMe = true;

    $scope.showTotal = true;

    // Assuming some calculation is done here to get the total

    $scope.total = 100; // Change this value to calculate a different total

};

});

</script>

</body>

</html>
```

6. Implement Node.js to print hello world

```
// helloWorld.js
console.log("Hello, World!");
-----
```

node helloWorld.js

7. Implement Node.js to get the current date and time

```
// getCurrentDateTime.js  
const currentDate = new Date();
```

```
console.log(`Current Date and Time: ${currentDate}`);
```

```
-----
```

```
node getCurrentDateTime.js
```

```
// getCurrentDateTimeFormatted.js
```

```
const currentDate = new Date();
```

```
const formattedDateTime = `${currentDate.getFullYear()}-${(currentDate.getMonth() + 1).toString().padStart(2, '0')}-${currentDate.getDate().toString().padStart(2, '0')}
```

```
${currentDate.getHours().toString().padStart(2, '0')}:${currentDate.getMinutes().toString().padStart(2, '0')}:${currentDate.getSeconds().toString().padStart(2, '0')}`;
```

```
console.log(`Formatted Date and Time: ${formattedDateTime}`);
```

```
node getCurrentDateTimeFormatted.js
```

8. Implement the Node.js using web browser to display sentence.

```
// simpleServer.js
```

```
const http = require('http');
```

```
const hostname = '127.0.0.1';

const port = 3000;

const server = http.createServer((req, res) => {

  res.statusCode = 200;

  res.setHeader('Content-Type', 'text/html');

  res.end('<h1>Hello from Node.js Server!</h1><p>This is a simple sentence displayed  
in the browser.</p>');

});

server.listen(port, hostname, () => {

  console.log(`Server running at http://${hostname}:${port}/`);

});
```

9. Implement the Node.js using web browser to display

```
// app.js
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
```

```
res.send('<h1>Hello from Node.js Server!</h1><p>This content is displayed in the  
browser.</p>');  
});  
  
app.listen(port, () => {  
  console.log(`Server running at http://localhost:${port}/`);  
});
```

10. Implement Node.js to creating new files:

- ☐ **fs.appendFile()**
- ☐ **fs.open()**
- ☐ **fs.writeFile()**

```
const fs = require('fs');
```

```
// Appending content to a file (or creating a new file)
fs.appendFile('newFile_append.txt', 'This is the content that will be appended.', (err) => {
  if (err) {
    console.error(err);
    return;
  }
  console.log('File created (or content appended) successfully!');
});
-----

const fs = require('fs');

// Creating a new file using fs.open()
fs.open('newFile_open.txt', 'w', (err, file) => {
  if (err) {
    console.error(err);
    return;
  }
  console.log('File created successfully!');
});
-----

const fs = require('fs');

// Writing content to a file (or creating a new file)
fs.writeFile('newFile_write.txt', 'This is the content that will be written.', (err) => {
  if (err) {
    console.error(err);
    return;
  }
  console.log('File created (or content written) successfully!');
});
```

