DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

Department of Artificial Intelligence and Machine Learning

Academic Year – 2023- 2024

Lab Manual

Angular 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

Program Number: 1

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

```
<!DOCTYPE html>
<html ng-app="fullNameApp">
<head>
  <title>Full Name Display</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
  <div ng-controller="FullNameController">
    <label>First Name:</label>
    <input type="text" ng-model="firstName"><br>
    <label>Last Name:</label>
    <input type="text" ng-model="lastName"><br>
    <p>Full Name: {{fullName()}}</p>
  </div>

  <script>
    var app = angular.module('fullNameApp', []);
    app.controller('FullNameController', function($scope) {
      // Default values for first name and last name
      $scope.firstName = "John";
      $scope.lastName = "Doe";

      // Function to concatenate first name and last name
      $scope.fullName = function() {
        return $scope.firstName + " " + $scope.lastName;
      };
    });
  </script>
</body>
</html>
```

Program Number : 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.

Solution:

```
<!DOCTYPE html>
<html ng-app="shoppingApp">
<head>
  <title>Shopping List</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
  <div ng-controller="ShoppingController">
    <h2>Shopping List</h2>
    <ul>
      <li ng-repeat="item in items">{{item}} <button ng-
click="removeItem($index)">Remove</button></li>
    </ul>
    <label>Add Item:</label>
    <input type="text" ng-model="newItem">
    <button ng-click="addItem()">Add</button>
  </div>

  <script>
    var app = angular.module('shoppingApp', []);
    app.controller('ShoppingController', function($scope) {
      // Default items
      $scope.items = ["Milk", "Bread", "Eggs"];

      // Function to add a new item to the list
      $scope.addItem = function() {
        if ($scope.newItem) {
          $scope.items.push($scope.newItem);
          $scope.newItem = ""; // Clear the input field
        }
      };

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

Program Number : 3

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

Solution:

```
<!DOCTYPE html>
<html ng-app="calculatorApp">
<head>
  <title>Simple Calculator</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
  <div ng-controller="CalculatorController">
    <h2>Simple Calculator</h2>
    <label>Number 1:</label>
    <input type="number" ng-model="num1"><br>
    <label>Number 2:</label>
    <input type="number" ng-model="num2"><br>
    <button ng-click="add()">Add</button>
    <button ng-click="subtract()">Subtract</button>
    <button ng-click="multiply()">Multiply</button>
    <button ng-click="divide()">Divide</button><br>
    <p>Result: {{result}}</p>
  </div>

  <script>
    var app = angular.module('calculatorApp', []);
    app.controller('CalculatorController', function($scope) {
      // Initialize variables
      $scope.num1 = 0;
      $scope.num2 = 0;
      $scope.result = 0;

      // Function to perform addition
      $scope.add = function() {
        $scope.result = $scope.num1 + $scope.num2;
      };

      // Function to perform subtraction
      $scope.subtract = function() {
        $scope.result = $scope.num1 - $scope.num2;
      };

      // Function to perform multiplication
      $scope.multiply = function() {
        $scope.result = $scope.num1 * $scope.num2;
      };

      // Function to perform division
      $scope.divide = function() {
```

```
                    if ($scope.num2 !== 0) {
                        $scope.result = $scope.num1 / $scope.num2;
                    } else {
                        $scope.result = "Cannot divide by zero";
                    }
                };
            });
    </script>
</body>
</html>
```

Program Number : 4

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

Solution:

```
<!DOCTYPE html>
<html ng-app="mathApp">
<head>
  <title>Factorial and Square Calculator</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
  <div ng-controller="MathController">
    <h2>Factorial and Square Calculator</h2>
    <label>Number:</label>
    <input type="number" ng-model="number"><br>
    <button ng-click="calculateFactorial()">Calculate Factorial</button>
    <button ng-click="calculateSquare()">Calculate Square</button><br>
    <p ng-if="factorial !== null">Factorial: {{factorial}}</p>
    <p ng-if="square !== null">Square: {{square}}</p>
  </div>

  <script>
    var app = angular.module('mathApp', []);
    app.controller('MathController', function($scope) {
      // Initialize variables
      $scope.number = 0;
      $scope.factorial = null;
      $scope.square = null;

      // Function to calculate factorial
      $scope.calculateFactorial = function() {
        if ($scope.number < 0) {
          $scope.factorial = "Invalid input";
          return;
        }
        var result = 1;
        for (var i = 2; i <= $scope.number; i++) {
          result *= i;
        }
        $scope.factorial = result;
      };
      // Function to calculate square
      $scope.calculateSquare = function() {
        $scope.square = $scope.number * $scope.number;
      };
    });
  </script>
</body>
</html>
```

Program Number : 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

Solution:

```
<!DOCTYPE html>
<html ng-app="studentApp">
<head>
   <title>Student Details</title>
   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
   <div ng-controller="StudentController">
      <h2>Student Details</h2>
      <label>Enter the number of students:</label>
      <input type="number" ng-model="numStudents" ng-change="generateStudents()"><br>

      <div ng-repeat="student in students">
         <h3>Student {{$index + 1}}</h3>
         <p>Name: {{student.name}}</p>
         <p>CGPA: {{student.cgpa}}</p>
      </div>

      <p>Total Students: {{numStudents}}</p>
   </div>

   <script>
      var app = angular.module('studentApp', []);
      app.controller('StudentController', function($scope) {
         $scope.numStudents = 0;
         $scope.students = [];

         $scope.generateStudents = function() {
            $scope.students = [];
            for (var i = 0; i < $scope.numStudents; i++) {
               $scope.students.push({
                  name: "Student " + (i + 1),
                  cgpa: Math.floor(Math.random() * 4) + 1 // Random CGPA between 1 and 4
               });
            }
         };
      });
   </script>
</body>
</html>
```

Program Number 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

Solution:

```
<!DOCTYPE html>
<html ng-app="todoApp">
<head>
  <title>To-Do List</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
  <div ng-controller="TodoController">
    <h2>To-Do List</h2>
    <label>Add Task:</label>
    <input type="text" ng-model="newTask">
    <button ng-click="addTask()">Add</button><br>

    <ul>
      <li ng-repeat="task in tasks">
        <span>{{task}}</span>
        <button ng-click="editTask($index)">Edit</button>
        <button ng-click="deleteTask($index)">Delete</button>
      </li>
    </ul>
  </div>

  <script>
    var app = angular.module('todoApp', []);
    app.controller('TodoController', function($scope) {
      // Default tasks
      $scope.tasks = ["Task 1", "Task 2", "Task 3"];

      // Function to add a task
      $scope.addTask = function() {
        if ($scope.newTask) {
          $scope.tasks.push($scope.newTask);
          $scope.newTask = ""; // Clear the input field
        }
      };

      // Function to edit a task
      $scope.editTask = function(index) {
        var editedTask = prompt("Edit Task", $scope.tasks[index]);
        if (editedTask !== null) {
          $scope.tasks[index] = editedTask;
        }
      };

      // Function to delete a task
      $scope.deleteTask = function(index) {
        $scope.tasks.splice(index, 1);
      };
    });
  </script>
</body>
</html>
```

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

Solution:

```
<!DOCTYPE html>
<html ng-app="crudApp">
<head>
  <title>User Management</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
  <div ng-controller="UserController">
    <h2>User Management</h2>
    <label>Name:</label>
    <input type="text" ng-model="newUser.name"><br>
    <label>Email:</label>
    <input type="email" ng-model="newUser.email"><br>
    <button ng-click="addUser()">Add User</button><br>

    <ul>
      <li ng-repeat="user in users">
        <span>Name: {{user.name}}, Email: {{user.email}}</span>
        <button ng-click="editUser(user)">Edit</button>
        <button ng-click="deleteUser(user)">Delete</button>
      </li>
    </ul>

    <div ng-show="editingUser">
      <h3>Edit User</h3>
      <label>Name:</label>
      <input type="text" ng-model="editedUser.name"><br>
      <label>Email:</label>
      <input type="email" ng-model="editedUser.email"><br>
      <button ng-click="updateUser()">Update</button>
      <button ng-click="cancelEdit()">Cancel</button>
    </div>
  </div>

  <script>
    var app = angular.module('crudApp', []);
    app.controller('UserController', function($scope) {
      $scope.users = []; // Array to store users
      $scope.newUser = {}; // Object to store new user data
      $scope.editedUser = {}; // Object to store edited user data
      $scope.editingUser = false; // Flag to indicate if user is being edited

      // Function to add a new user
      $scope.addUser = function() {
        $scope.users.push($scope.newUser);
        $scope.newUser = {}; // Clear the new user object
      };

      // Function to edit a user
      $scope.editUser = function(user) {
        $scope.editedUser = angular.copy(user);
        $scope.editingUser = true;
      };

      // Function to update a user
      $scope.updateUser = function() {
        var index = $scope.users.indexOf($scope.editedUser);
        $scope.users[index] = $scope.editedUser;
        $scope.editingUser = false;
        $scope.editedUser = {};
      };

      // Function to cancel editing
      $scope.cancelEdit = function() {
        $scope.editingUser = false;
        $scope.editedUser = {};
      };

      // Function to delete a user
      $scope.deleteUser = function(user) {
        var index = $scope.users.indexOf(user);
        $scope.users.splice(index, 1);
      };
    });
  </script>
</body>
</html>
```

Program Number 8:

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

Solution:

```
<!DOCTYPE html>
<html ng-app="loginApp">
<head>
  <title>Login Form</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
  <div ng-controller="LoginController">
    <h2>Login Form</h2>
    <form name="loginForm" ng-submit="submitForm()">
      <label>Username:</label>
      <input type="text" ng-model="username" required><br>
      <div ng-show="loginForm.username.$dirty && loginForm.username.$invalid">
        <span ng-show="loginForm.username.$error.required">Username is required.</span>
      </div>

      <label>Password:</label>
      <input type="password" ng-model="password" required><br>
      <div ng-show="loginForm.password.$dirty && loginForm.password.$invalid">
        <span ng-show="loginForm.password.$error.required">Password is required.</span>
      </div>

      <button type="submit" ng-disabled="loginForm.$invalid">Login</button>
    </form>
  </div>

  <script>
    var app = angular.module('loginApp', []);
    app.controller('LoginController', function($scope) {
      $scope.submitForm = function() {
        // Implement your login logic here
        console.log('Username:', $scope.username);
        console.log('Password:', $scope.password);
      };
    });
  </script>
</body>
</html>
```

Program Number 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.

Solution:

```
<!DOCTYPE html>
<html ng-app="employeeApp">
<head>
  <title>Employee List</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
  <div ng-controller="EmployeeController">
    <h2>Employee List</h2>
    <label>Search by Name:</label>
    <input type="text" ng-model="searchName"><br>
    <label>Search by Salary:</label>
    <input type="number" ng-model="searchSalary"><br>

    <ul>
      <li ng-repeat="employee in filteredEmployees">
        <span>Name: {{employee.name}}, Salary: {{employee.salary}}</span>
      </li>
    </ul>
  </div>

  <script>
    var app = angular.module('employeeApp', []);
    app.controller('EmployeeController', function($scope) {
      $scope.employees = [
        { name: 'John', salary: 50000 },
        { name: 'Jane', salary: 60000 },
        { name: 'Doe', salary: 55000 },
        { name: 'Smith', salary: 70000 },
        { name: 'Alice', salary: 65000 }
      ];

      $scope.searchName = '';
      $scope.searchSalary = '';

      $scope.filteredEmployees = $scope.employees;

      $scope.$watchGroup(['searchName', 'searchSalary'], function(newValues, oldValues) {
        var name = newValues[0];
        var salary = newValues[1];
        $scope.filteredEmployees = $scope.employees.filter(function(employee) {
          return employee.name.toLowerCase().includes(name.toLowerCase()) &&
              (salary === '' || employee.salary >= salary);
        });
      });
```

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

Program Number 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. Include the default values for items in the program

Solution:

```
<!DOCTYPE html>
<html ng-app="itemApp">
<head>
    <title>Item Collection</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
    <div ng-controller="ItemController">
        <h2>Item Collection</h2>
        <label>Add Item:</label>
        <input type="text" ng-model="newItem"><br>
        <button ng-click="addItem()">Add</button><br>

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

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

    <script>
        var app = angular.module('itemApp', []);
        app.controller('ItemController', function($scope) {
            // Default items
            $scope.items = ["Item 1", "Item 2", "Item 3"];

            // Function to add an item
            $scope.addItem = function() {
                if ($scope.newItem) {
                    $scope.items.push($scope.newItem);
                    $scope.newItem = ""; // Clear the input field
                }
            };
            // Function to remove an item
            $scope.removeItem = function(index) {
                $scope.items.splice(index, 1);
            };
        });
    </script>
</body>
</html>
```

Program Number 11:

Create AngularJS application to convert student details to Uppercase using angular filters. Include the default details of students in the program.

Solution:

```
<!DOCTYPE html>
<html ng-app="studentApp">
<head>
    <title>Student Details</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
    <div ng-controller="StudentController">
        <h2>Student Details</h2>

        <ul>
            <li ng-repeat="student in students">
                <span>Name: {{student.name | uppercase}}, Grade: {{student.grade | uppercase}}</span>
            </li>
        </ul>
    </div>

    <script>
        var app = angular.module('studentApp', []);
        app.controller('StudentController', function($scope) {
            $scope.students = [
                { name: 'John Doe', grade: 'A' },
                { name: 'Alice Smith', grade: 'B' },
                { name: 'Bob Johnson', grade: 'C' }
            ];
        });
    </script>
</body>
</html>
```

Program Number 12:

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

Solution:

```
<!DOCTYPE html>
<html ng-app="dateApp">
<head>
  <title>Date Display</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>
  <div ng-controller="DateController">
    <h2>Date Display</h2>
    <p>Today's Date: {{ currentDate | date: 'fullDate' }}</p>
    <p>Current Time: {{ currentDate | date: 'shortTime' }}</p>
  </div>

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