ANJS 1

Program 1: Develop AngularJS program that allows user to input their firstname and lastname and display their full name. Note: The default values for first name and lastname 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/1.8.2/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 = 'Hello';
                $scope.lastName = 'world';
                // Function to display the full name
                $scope.displayFullName = function () {
                    $scope.fullName = $scope.firstName + ' ' + $scope.lastName;
                };
            });
    </script>
</body>

</html>
```

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 inthe program.

```
<html ng-app="shoppingApp">

<head>
    <title>AngularJS Shopping List</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
</script>
</head>

<body ng-controller="shoppingCtrl">
    <h2>Shopping List</h2>
    <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>
```

Program 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="calculatorCtrl">
    <h2>Simple Calculator</h2>
    Enter Number 1:
    <input type="number" ng-model="num1"> &nbsp;
    Select Operator:
    <select ng-model="operator">
        <option value="+">Add</option>
        <option value="-">Subtract</option>
        <option value="*">Multiply</option>
        <option value="/">Divide</option>
    </select>&nbsp;
    Enter Number 2:
    <input type="number" ng-model="num2">
    <button ng-click="calculate()">Calculate</button>
    <p ng-show="result !== undefined">Result: {{ result }}</p>
    <script>
        angular.module('calculatorApp', [])
            .controller('calculatorCtrl', 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>
```

Program 4: Write an Angular JS application that can calculate factorial and compute square based on given user 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="mathCtrl">
    <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('mathCtrl', 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>
```

Program 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.

```
<html ng-app="studentApp">

<head>
    <title>AngularJS Student Details</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
</script>
</head>

<body ng-controller="studentCtrl">
    <h2>Student Details</h2>
    Student Name:
    <input type="text" ng-model="name">
    CGPA:
    <input type="number" ng-model="cgpa" ng-min="1" ng-max="10">
    <button ng-click="addStudent()">Add Student</button>
    <p>Total Students: {{ students.length }}</p>
    <ul>
        <li ng-repeat="student in students">
            {{ student.name }} - CGPA: {{ student.cgpa }}
        </li>

    </ul>
    <script>
        angular.module('studentApp', [])
            .controller('studentCtrl', function ($scope) {
                $scope.students = [];
                $scope.addStudent = function () {
                    if ($scope.name && $scope.cgpa) {
                        $scope.students.push({
                            name: $scope.name,
                            cgpa: $scope.cgpa
                        });
                        // Clear the input fields
                        $scope.name = '';
                        $scope.cgpa = '';
                    }
                };
            });
    </script>
</body>

</html>
```

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.

```
<html ng-app="todoApp">

<head>
    <title>AngularJS Todo List</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
</script>
</head>

<body ng-controller="todoCtrl">
    <h1>Todo List</h1>
    <!-- Form for adding a new task -->
    <form ng-submit="addTask()">
        Task:
        <input type="text" ng-model="newTask" required>
        <button type="submit">Add Task</button>
    </form>
    <br>
    <table>
        <thead>
            <tr>
                <th>Task</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            <tr ng-repeat="task in tasks">
                <td>{{ task }}</td>
                <td>
                    <button ng-click="editTask($index)">Edit</button>
                    <button ng-click="deleteTask($index)">Delete</button>
                </td>
            </tr>
        </tbody>
    </table>
    <script>
        angular.module('todoApp', []).
            controller('todoCtrl', function ($scope) {
                $scope.tasks = [
                    'Task 1',
                    'Task 2',
                    'Task 3'
                ];
                $scope.newTask = '';
                $scope.addTask = function () {
                    $scope.tasks.push($scope.newTask);
                    $scope.newTask = '';
                };
                $scope.editTask = function (index) {
                    // Prompt for updated task with validation
                    var updatedTask = prompt('Enter updated task:');
                    // Check if the user pressed cancel
                    if (updatedTask !== null) {
                        // Update the task
                        // $scope.tasks.splice(index, 1, updatedTask);
                        $scope.tasks[index] = updatedTask;
                    }
                };
                $scope.deleteTask = function (index) {
                    $scope.tasks.splice(index, 1);
                };
            });
    </script>
</body>

</html>
```

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

```
<html ng-app="crudApp">

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

<body ng-controller="crudCtrl">
    <h1>User Management</h1>

    <form ng-submit="addUser()">
        User:
        <input type="text" ng-model="name" requied><br>
        Age:
        <input type="number" ng-model="age" required><br>
        <button type="submit">Add User</button>
    </form>
    <br>
    <table>
        <tr>
            <th>Name</th>
            <th>Age</th>
            <th>Action</th>
        </tr>
        <tr ng-repeat="user in users">
            <td>{{user.name}}</td>
            <td>{{user.age}}</td>
            <td>
                <button ng-click="editTask($index)">Edit</button>
                <button ng-click="deleteTask($index)">Delete</button>
            </td>
        </tr>
    </table>
    <script>
        angular.module('crudApp', [])
            .controller('crudCtrl', function ($scope) {
                $scope.users = [
                    { name: 'user1', age: 25 },
                    { name: 'uesr2', age: 30 }
                ];
                $scope.addUser = function () {
                    $scope.users.push({ name: $scope.name, age: $scope.age });
                };

                $scope.editTask = function ($index) {
                    var tmpName = prompt("Enter the new name");
                    var tmpAge = prompt("Enter the new age");
                    // $scope.users.splice($index, 1, {name: tmpName, age: tmpAge});
                    $scope.users[$index] = { name: tmpName, age: tmpAge };
                };

                $scope.deleteTask = function ($index) {
                    $scope.users.splice($index, 1);
                };
            });
    </script>
</body>

</html>
```

Program 8: Develop AngularJS program to create a login form, with validation for the username and password fields.

```
<html ng-app="loginApp">

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

<body ng-controller="loginCtrl">
    <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>
        angular.module('loginApp', [])
            .controller('loginCtrl', 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>
```

Program 9: 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="employeeCtrl">
    <h2>Employee List</h2>
    Search by Name:
    <input type="text" ng-model="searchName" /> <br>
    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>
        angular.module('employeeApp', [])
            .controller('employeeCtrl', 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>
```

Program 10: 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">

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

<body ng-controller="itemCtrl">
    <h2>ItemCollection</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>
        angular.module('itemApp', [])
            .controller('itemCtrl', function ($scope) {
                $scope.items = ['Item1', 'Item2', 'Item3'];//Defaultitems
                $scope.newItem = '';
                $scope.addItem = function () {
                    if ($scope.newItem) {
                        $scope.items.push($scope.newItem);
                        $scope.newItem = '';//Cleartheinputfield
                    }
                };
                $scope.removeItem = function (index) {
                    $scope.items.splice(index, 1);
                };
            });
    </script>
</body>

</html>
```

Program 11: Create AngularJS application to convert student details to upper case using angular filters. Note: The default details of students may be included in the program.

```
<html ng-app="studentApp">

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

<body ng-controller="studentCtrl">
    <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 upper case using filters-->
    <h3>Names in Uppercase:</h3>
    <ul>
        <li ng-repeat="name in names">
            {{name|uppercase}}
        </li>
    </ul>

    <script>
        angular.module('studentApp', [])
            .controller('studentCtrl', function ($scope) {
                $scope.names = ['Raj', 'Ram', 'Sam'];
            });
    </script>
</body>

</html>
```

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

```
<html ng-app="dateApp">

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

<body ng-controller="dateCtrl">
    <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>
        angular.module('dateApp', [])
            .controller('dateCtrl', function ($scope) {
                $scope.currentDate = new Date();
            });
    </script>
</body>

</html>
```