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

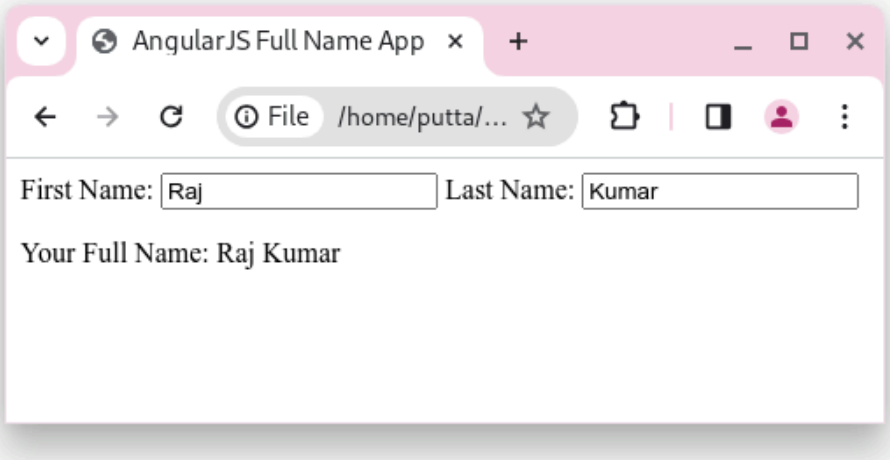
Sl.NO	Experiments
1	Develop Angular JS program that allows user to input their first name and last name and display their full name. **Note**: The default values for first name and last name may be included in the program.
2	Develop an Angular JS application that displays a list of shopping items. Allow 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.	

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>AngularJS Full Name App</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="fullNameCtrl">
  <label for="firstName">First Name:</label>
  <input type="text" id="firstName" ng-model="firstName" placeholder="Enter your first name">

  <label for="lastName">Last Name:</label>
  <input type="text" id="lastName" ng-model="lastName" placeholder="Enter your last name">

  <p>Your Full Name: {{ getFullName() }}</p>
</div>

<script>
  // AngularJS application module
  var app = angular.module('fullNameApp', []);
```

```
 // AngularJS controller
 app.controller('fullNameCtrl', function ($scope) {
  // Default values for first name and last name
  $scope.firstName = 'Raj';
  $scope.lastName = 'Kumar';

  // Function to get the full name
  $scope.getFullName = function () {
   return $scope.firstName + ' ' + $scope.lastName;
  };
 });
</script>

</body>
</html>
```

Output



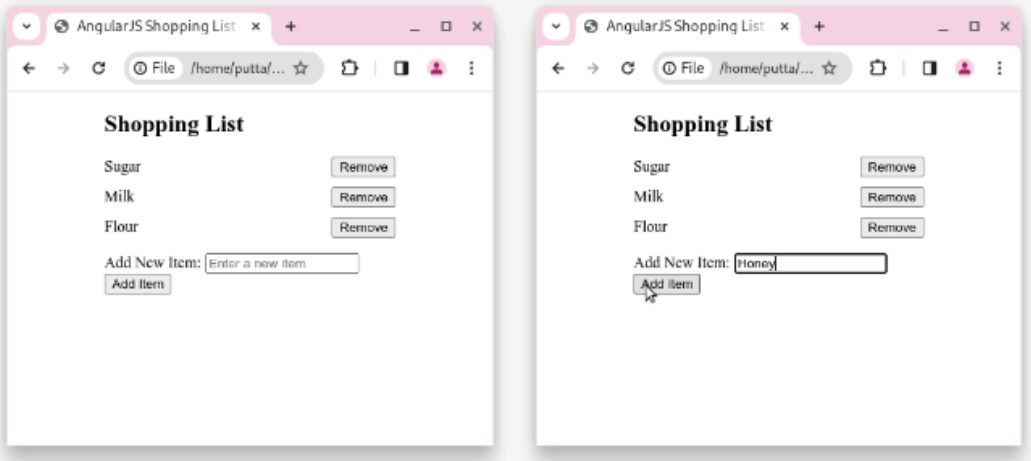
2. Develop an Angular JS application that displays a list of shopping items. Allow users to add and remove items from the list using directives and controllers.Note: The default values of items may be included in the program.

```
<!DOCTYPE html>
<html ng-app="shoppingListApp">
<head>
  <title>AngularJS Shopping List App</title>
  <style>
    #shoppingListContainer {
      width: 300px;
      margin: auto;
    }

    ul {
      list-style-type: none;
      padding: 0;
    }

    li {
      margin-bottom: 10px;
      display: flex;
      justify-content: space-between;
      align-items: center;
    }

    input {
      margin-right: 10px;
```

```
  }
 </style>
 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="shoppingListCtrl" id="shoppingListContainer">
 <h2>Shopping List</h2>

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

 <div>
  <label for="newItem">Add New Item:</label>
  <input type="text" id="newItem" ng-model="newItem" placeholder="Enter a new item">
  <button ng-click="addItem()">Add Item</button>
 </div>
</div>

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

 app.controller('shoppingListCtrl', function ($scope) {
  $scope.shoppingItems = ['Sugar', 'Milk', 'Flour'];
```

```
  $scope.addItem = function () {
    if ($scope.newItem) {
      $scope.shoppingItems.push($scope.newItem);
      $scope.newItem = '';
    }
  };

  $scope.removeItem = function (index) {
    $scope.shoppingItems.splice(index, 1);
  };
 });
</script>

</body>
</html>
```

Output



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

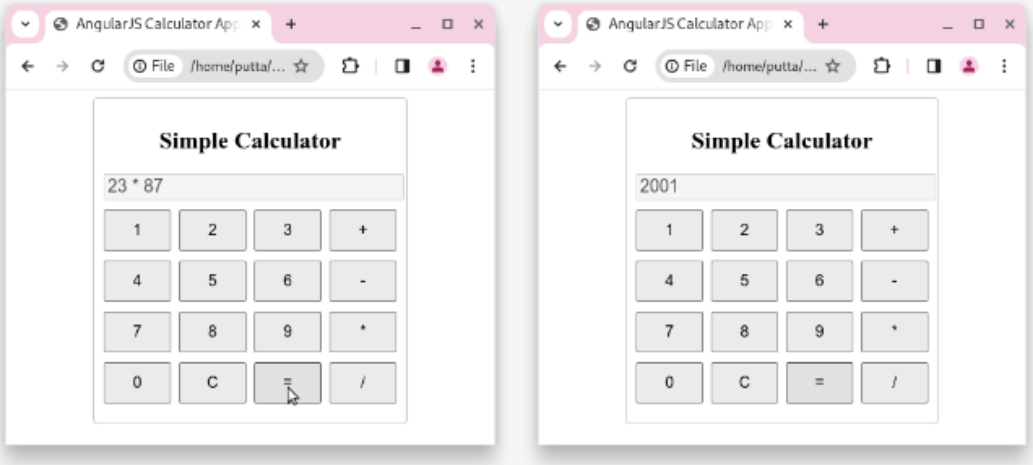
```
<!DOCTYPE html>
<html ng-app="calculatorApp">
<head>
  <title>AngularJS Calculator App</title>
  <style>
    #calculator {
      width: 300px;
      margin: auto;
      border: 1px solid #ccc;
      padding: 10px;
      border-radius: 5px;
      text-align: center;
    }

    input {
      width: 100%;
      margin-bottom: 10px;
      font-size: 18px;
    }

    .row {
      display: flex;
      justify-content: space-between;
    }

    button {
      width: 23%;
```

```
    padding: 10px;
    font-size: 16px;
    margin-bottom: 10px;
   }
  </style>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="calculatorCtrl" id="calculator">
  <h2>Simple Calculator</h2>

  <input type="text" ng-model="display" disabled>

  <div class="row">
    <button ng-click="appendToDisplay('1')">1</button>
    <button ng-click="appendToDisplay('2')">2</button>
    <button ng-click="appendToDisplay('3')">3</button>
    <button ng-click="performOperation('+')">+</button>
  </div>

  <div class="row">
    <button ng-click="appendToDisplay('4')">4</button>
    <button ng-click="appendToDisplay('5')">5</button>
    <button ng-click="appendToDisplay('6')">6</button>
    <button ng-click="performOperation('-')">-</button>
  </div>

  <div class="row">
```

```
    <button ng-click="appendToDisplay('7')">7</button>
    <button ng-click="appendToDisplay('8')">8</button>
    <button ng-click="appendToDisplay('9')">9</button>
    <button ng-click="performOperation('*')">*</button>
  </div>

  <div class="row">
    <button ng-click="appendToDisplay('0')">0</button>
    <button ng-click="clearDisplay()">C</button>
    <button ng-click="calculateResult()">=</button>
    <button ng-click="performOperation('/')">/</button>
  </div>
</div>

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

  app.controller('calculatorCtrl', function ($scope) {
    $scope.display = '';

    $scope.appendToDisplay = function (value) {
      $scope.display += value;
    };

    $scope.clearDisplay = function () {
      $scope.display = '';
    };

    $scope.performOperation = function (operator) {
```

```
    $scope.appendToDisplay(' ' + operator + ' ');
  };

  $scope.calculateResult = function () {
    try {
      $scope.display = eval($scope.display);
    } catch (error) {
      $scope.display = 'Error';
    }
  };
});
</script>

</body>
</html>
```

Output



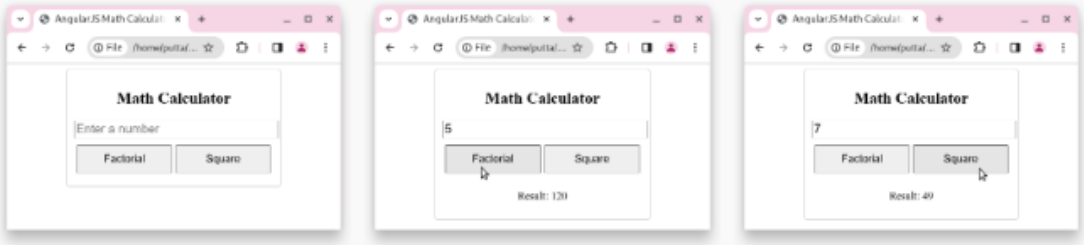
4. Write an Angular JS application that can calculate factorial and compute square based on given user input.

```
<!DOCTYPE html>
<html ng-app="mathApp">
<head>
 <title>AngularJS Math Calculator</title>
 <style>
  #calculator {
   width: 300px;
   margin: auto;
   border: 1px solid #ccc;
   padding: 10px;
   border-radius: 5px;
   text-align: center;
  }

  input {
   width: 100%;
   margin-bottom: 10px;
   font-size: 18px;
  }

  button {
   width: 48%;
   padding: 10px;
   font-size: 16px;
   margin-bottom: 10px;
  }
```

```
    button:last-child {
      margin-right: 0;
    }
  </style>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="mathCtrl" id="calculator">
  <h2>Math Calculator</h2>

  <input type="number" ng-model="inputNumber" placeholder="Enter a number">

  <div>
    <button ng-click="calculateFactorial()">Factorial</button>
    <button ng-click="calculateSquare()">Square</button>
  </div>

  <p ng-show="result !== undefined">Result: {{ result }}</p>
</div>

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

  app.controller('mathCtrl', function ($scope) {
    $scope.inputNumber = '';
    $scope.result = undefined;

    $scope.calculateFactorial = function () {
```

```
    if ($scope.inputNumber >= 0) {
      $scope.result = factorial($scope.inputNumber);
    } else {
      $scope.result = 'Invalid input';
    }
  };

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

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

</body>
</html>
```

Output



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.

```
<!DOCTYPE html>
<html ng-app="studentApp">
<head>
 <title>AngularJS Student Details</title>
 <style>
  #studentDetails {
   display: flex;
   width: 800px;
   margin: auto;
  }

  #inputForm {
   width: 33.33%;
   border: 1px solid #ccc;
   padding: 20px;
   border-radius: 5px;
   text-align: center;
  }
```

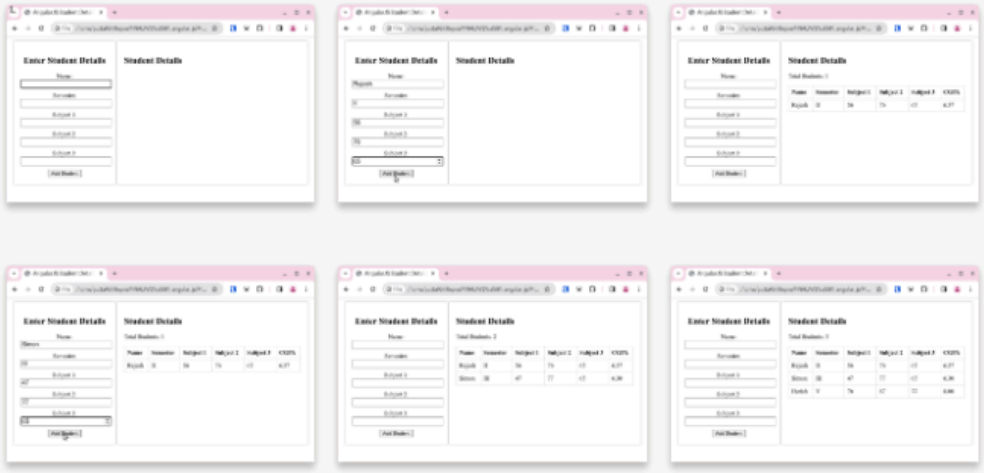
```
#displayDetails {
  width: 66.66%;
  border: 1px solid #ccc;
  padding: 20px;
  border-radius: 5px;
}

input, select {
  width: 100%;
  margin-bottom: 10px;
  font-size: 16px;
}

table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 10px;
}

th, td {
  border: 1px solid #ddd;
  padding: 8px;
  text-align: left;
}

th:nth-child(3), th:nth-child(4), th:nth-child(5) {
  white-space: nowrap;
}
```

```
    td:last-child {
      max-width: 80px; /* Adjust the max-width as needed */
      overflow: hidden;
      text-overflow: ellipsis;
    }
  </style>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="studentCtrl" id="studentDetails">
  <div id="inputForm">
    <h2>Enter Student Details</h2>
    <form ng-submit="enterStudentDetails()">
      <label for="studentName">Name:</label>
      <input type="text" id="studentName" ng-model="newStudent.name" required>

      <label for="studentSemester">Semester:</label>
      <input type="text" id="studentSemester" ng-model="newStudent.semester" required>

      <label for="subject1">Subject 1:</label>
      <input type="number" id="subject1" ng-model="newStudent.marks.subject1" required>

      <label for="subject2">Subject 2:</label>
      <input type="number" id="subject2" ng-model="newStudent.marks.subject2" required>

      <label for="subject3">Subject 3:</label>
      <input type="number" id="subject3" ng-model="newStudent.marks.subject3" required>
```

```
    <button type="submit">Add Student</button>
  </form>
 </div>

 <div id="displayDetails">
  <h2>Student Details</h2>
  <p ng-show="students.length > 0">Total Students: {{ students.length }}</p>

  <table ng-show="students.length > 0">
   <tr>
    <th>Name</th>
    <th>Semester</th>
    <th>Subject 1</th>
    <th>Subject 2</th>
    <th>Subject 3</th>
    <th>CGPA</th>
   </tr>
   <tr ng-repeat="student in students">
    <td>{{ student.name }}</td>
    <td>{{ student.semester }}</td>
    <td>{{ student.marks.subject1 }}</td>
    <td>{{ student.marks.subject2 }}</td>
    <td>{{ student.marks.subject3 }}</td>
    <td>{{ calculateCGPA(student.marks) }}</td>
   </tr>
  </table>
 </div>
</div>
```

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

  app.controller('studentCtrl', function ($scope) {
    $scope.newStudent = {};
    $scope.students = [];

    $scope.enterStudentDetails = function () {
      $scope.students.push(angular.copy($scope.newStudent));
      $scope.newStudent = {};
    };

    $scope.calculateCGPA = function (marks) {
      var averageMarks = (marks.subject1 + marks.subject2 + marks.subject3) / 3;
      return (averageMarks / 10).toFixed(2);
    };
  });
</script>

</body>
</html>
```

Output



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.

```
<!DOCTYPE html>
<html ng-app="todoApp">
<head>
 <title>AngularJS To-Do List</title>
 <style>
  #todoList {
    width: 400px;
    margin: auto;
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 20px;
    margin-top: 20px;
    text-align: center;
  }
```

```
ul {
  list-style-type: none;
  padding: 0;
  text-align: left;
}

li {
  margin-bottom: 10px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

button {
  background-color: #dc3545;
  color: #fff;
  border: none;
  padding: 5px 10px;
  border-radius: 3px;
  cursor: pointer;
}

input[type="text"] {
  padding: 5px;
  width: 60%;
  margin-right: 5px;
}
```

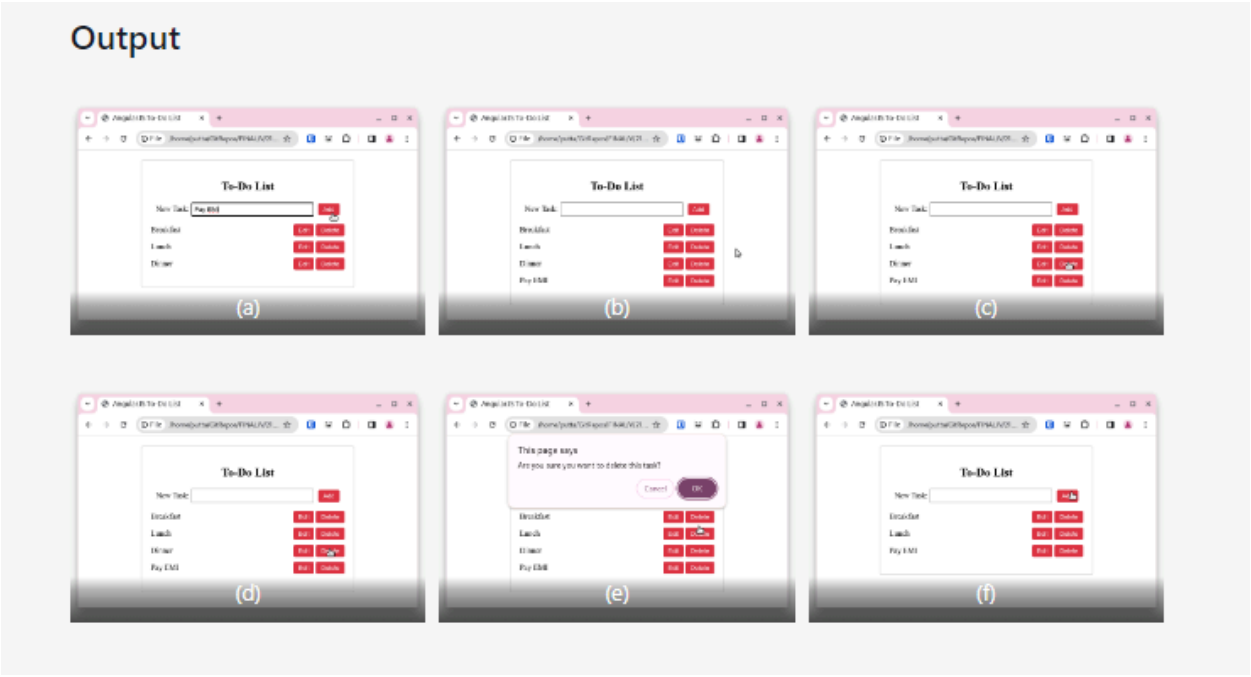
```
    form {
      margin-top: 10px;
    }
  </style>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="todoCtrl" id="todoList">
  <h2>To-Do List</h2>

  <form ng-submit="addTask()">
    <label for="newTask">New Task:</label>
    <input type="text" id="newTask" ng-model="newTask" required>
    <button type="submit">Add</button>
  </form>

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

<script>
```

```
  var app = angular.module('todoApp', []);

  app.controller('todoCtrl', function ($scope) {
    $scope.tasks = ["Breakfast", "Lunch", "Dinner"];
    $scope.newTask = "";

    $scope.addTask = function () {
      if ($scope.newTask.trim() !== "") {
        $scope.tasks.push($scope.newTask);
        $scope.newTask = "";
      }
    };

    $scope.editTask = function (index) {
      var editedTask = prompt("Edit task:", $scope.tasks[index]);
      if (editedTask !== null) {
        $scope.tasks[index] = editedTask;
      }
    };

    $scope.deleteTask = function (index) {
      var confirmDelete = confirm("Are you sure you want to delete this task?");
      if (confirmDelete) {
        $scope.tasks.splice(index, 1);
      }
    };
  });
</script>
```

</body>

</html>

Output



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

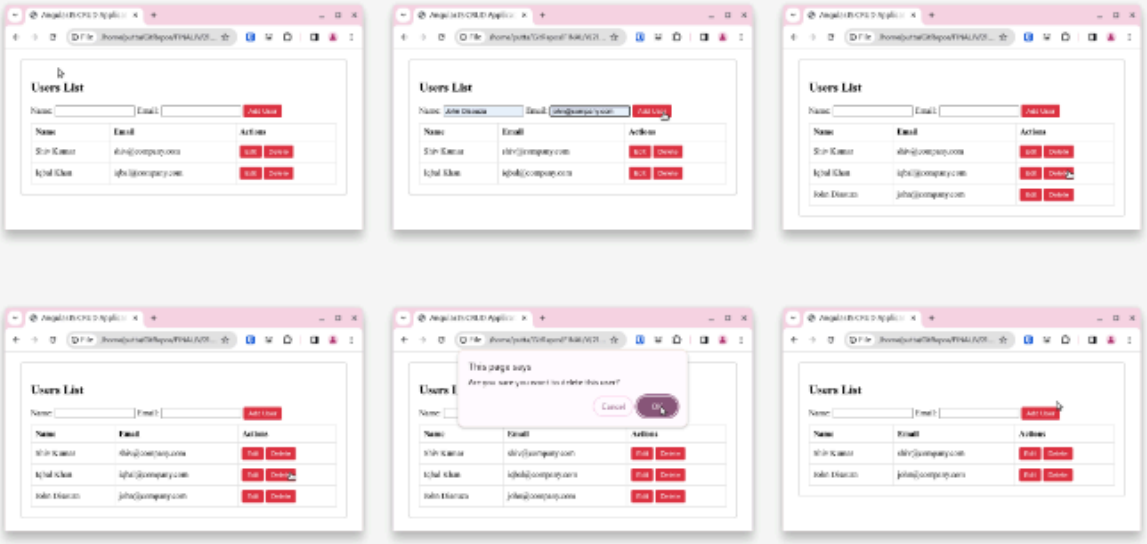
```
<!DOCTYPE html>
<html ng-app="crudApp">
<head>
 <title>AngularJS CRUD Application</title>
 <style>
  #userList {
   width: 600px;
   margin: auto;
   border: 1px solid #ccc;
   border-radius: 5px;
   padding: 20px;
   margin-top: 20px;
  }

  table {
   width: 100%;
   border-collapse: collapse;
```

```
    margin-top: 10px;
   }

   th, td {
    border: 1px solid #ddd;
    padding: 8px;
    text-align: left;
   }

   button {
    background-color: #dc3545;
    color: #fff;
    border: none;
    padding: 5px 10px;
    border-radius: 3px;
    cursor: pointer;
   }
  </style>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="crudCtrl" id="userList">
  <h2>Users List</h2>

  <form ng-submit="addUser()">
   <label for="userName">Name:</label>
   <input type="text" id="userName" ng-model="newUser.name" required>

   <label for="userEmail">Email:</label>
   <input type="email" id="userEmail" ng-model="newUser.email" required>

   <button type="submit">Add User</button>
  </form>

  <table ng-show="users.length > 0">
   <tr>
    <th>Name</th>
    <th>Email</th>
    <th>Actions</th>
   </tr>
   <tr ng-repeat="user in users">
    <td>{{ user.name }}</td>
    <td>{{ user.email }}</td>
```

```
      <td>
        <button ng-click="editUser(user)">Edit</button>
        <button ng-click="deleteUser(user)">Delete</button>
      </td>
    </tr>
  </table>
</div>

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

  app.controller('crudCtrl', function ($scope) {
    $scope.users = [
      { name: 'Shiv Kumar', email: 'shiv@company.com' },
      { name: 'Iqbal Khan', email: 'shiv@company.com' }
    ];

    $scope.newUser = {};

    $scope.addUser = function () {
      if ($scope.newUser.name && $scope.newUser.email) {
        $scope.users.push(angular.copy($scope.newUser));
        $scope.newUser = {};
      }
    };

    $scope.editUser = function (user) {
      var editedName = prompt("Edit user's name:", user.name);
      var editedEmail = prompt("Edit user's email:", user.email);

      if (editedName !== null && editedEmail !== null) {
        user.name = editedName;
        user.email = editedEmail;
      }
    };

    $scope.deleteUser = function (user) {
      var confirmDelete = confirm("Are you sure you want to delete this user?");
      if (confirmDelete) {
        var index = $scope.users.indexOf(user);
        $scope.users.splice(index, 1);
      }
    };
  });
```

```
</script>

</body>
</html>
```

Output



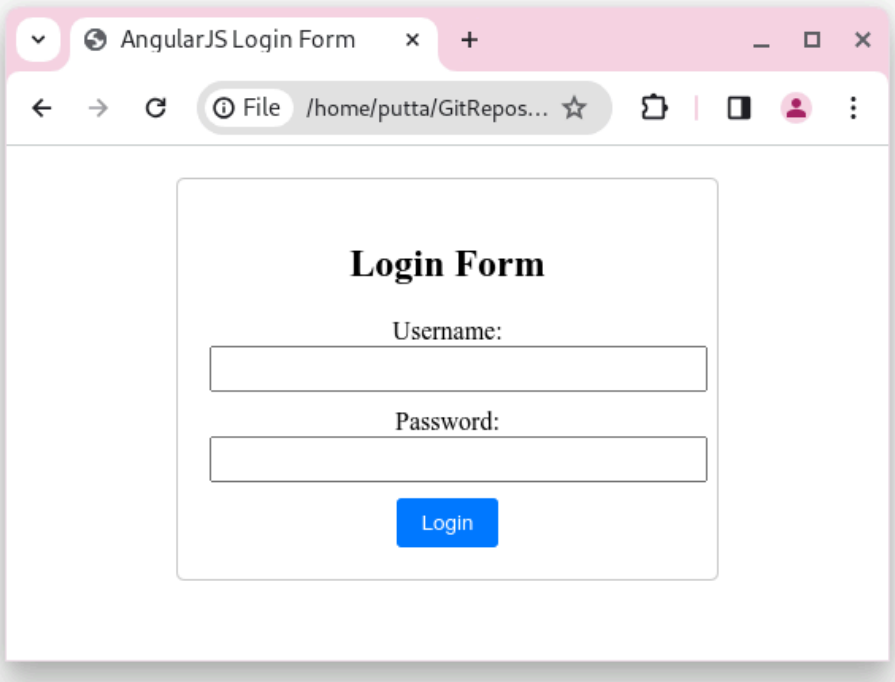
8. Develop AngularJS program to create a login form, with validation for the username and password fields.

```
<!DOCTYPE html>
<html ng-app="loginApp">
<head>
 <title>AngularJS Login Form</title>
 <style>
  #loginForm {
   width: 300px;
   margin: auto;
   border: 1px solid #ccc;
   border-radius: 5px;
   padding: 20px;
   margin-top: 20px;
   text-align: center;
  }
```

```
    input {
      width: 100%;
      margin-bottom: 10px;
      padding: 5px;
    }

    button {
      background-color: #007bff;
      color: #fff;
      border: none;
      padding: 8px 16px;
      border-radius: 3px;
      cursor: pointer;
    }

    .error {
      color: #dc3545;
      margin-top: 5px;
    }
  </style>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="loginCtrl" id="loginForm">
  <h2>Login Form</h2>

  <form name="loginForm" ng-submit="login()">
    <label for="username">Username:</label>
    <input type="text" id="username" ng-model="username" ng-pattern="/^[a-zA-Z_]{4,}$/"
required>
    <div class="error" ng-show="formSubmitted &&
loginForm.username.$error.required">Username is required.</div>
    <div class="error" ng-show="formSubmitted && loginForm.username.$error.pattern">
      Username should contain only alphabets and underscore and be at least 4 characters.
    </div>

    <label for="password">Password:</label>
    <input type="password" id="password" ng-model="password" ng-minlength="6" required>
    <div class="error" ng-show="formSubmitted &&
loginForm.password.$error.required">Password is required.</div>
    <div class="error" ng-show="formSubmitted && loginForm.password.$error.minlength">
      Password should be at least 6 characters.
    </div>
```

```
    <button type="submit">Login</button>
    <div class="error" ng-show="loginFailed">Invalid username or password.</div>
  </form>
</div>

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

  app.controller('loginCtrl', function ($scope) {
    $scope.username = "";
    $scope.password = "";
    $scope.formSubmitted = false;
    $scope.loginFailed = false;

    // In a real-world scenario, replace this with actual authentication logic.
    var validUsername = "user123";
    var validPassword = "pass123";

    $scope.login = function () {
      $scope.formSubmitted = true;

      if ($scope.loginForm.$valid) {
        if ($scope.username === validUsername && $scope.password === validPassword) {
          // Authentication successful
          console.log('Login successful!');
          $scope.loginFailed = false;
        } else {
          // Authentication failed
          console.log('Login failed!');
          $scope.loginFailed = true;
        }
      }
    };
  });
</script>

</body>
</html>
```

Output



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.

```
<!DOCTYPE html>
<html ng-app="employeeApp">
<head>
 <title>AngularJS Employee List</title>
 <style>
  #employeeList {
   width: 400px;
   margin: auto;
   border: 1px solid #ccc;
   border-radius: 5px;
   padding: 20px;
```

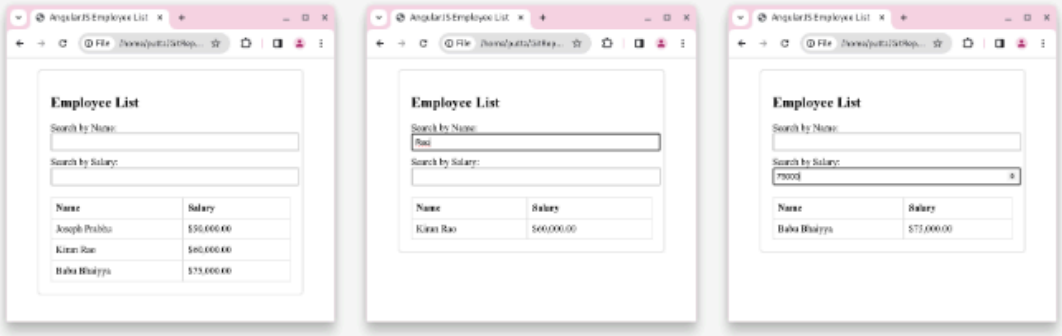
```
    margin-top: 20px;
   }

   input {
    width: 100%;
    margin-bottom: 10px;
    padding: 5px;
   }

   table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 10px;
   }

   th, td {
    border: 1px solid #ddd;
    padding: 8px;
    text-align: left;
   }
  </style>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="employeeCtrl" id="employeeList">
  <h2>Employee List</h2>

  <label for="searchName">Search by Name:</label>
```

```
  <input type="text" id="searchName" ng-model="searchName">

  <label for="searchSalary">Search by Salary:</label>
  <input type="number" id="searchSalary" ng-model="searchSalary">

  <table>
    <tr>
      <th>Name</th>
      <th>Salary</th>
    </tr>
    <tr ng-repeat="employee in employees | filter:{name: searchName, salary: searchSalary}">
      <td>{{ employee.name }}</td>
      <td>{{ employee.salary | currency }}</td>
    </tr>
  </table>
</div>

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

  app.controller('employeeCtrl', function ($scope) {
    $scope.employees = [
      { name: 'Joseph Prabhu', salary: 50000 },
      { name: 'Kiran Rao', salary: 60000 },
      { name: 'Babu Bhaiyya', salary: 75000 },
      // Add more employee details as needed
    ];
  });
</script>
```

</body>

</html>

Output

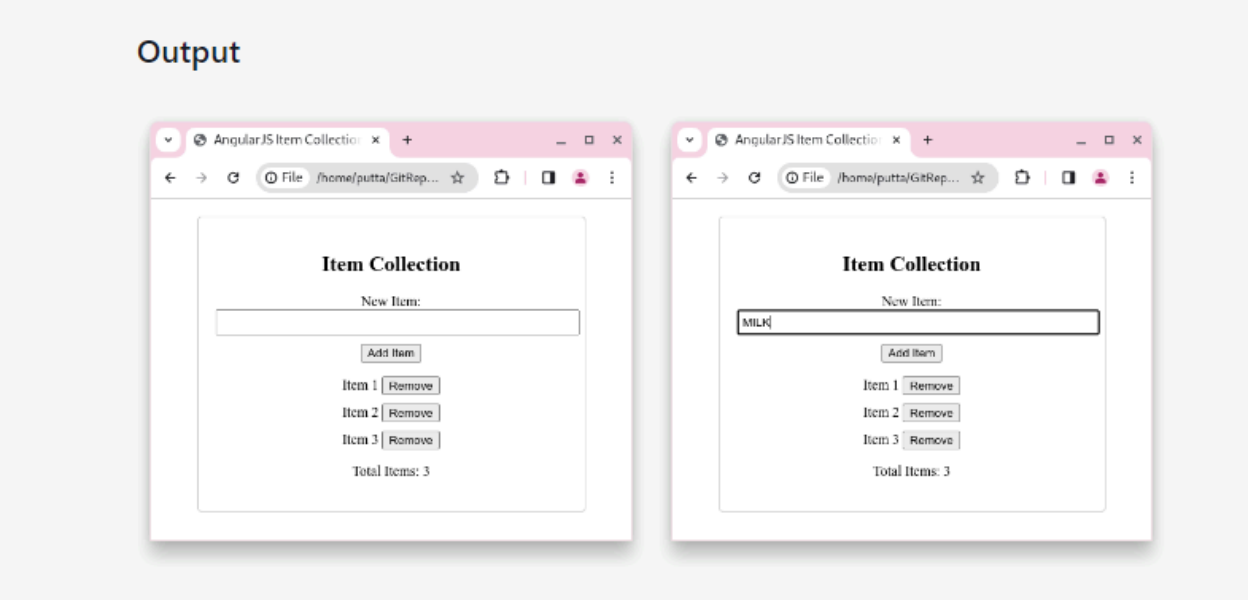


10. Create AngularJS application that allows users to maintain a collection of items. The application should display the current total number of items, and this count should automatically update as items are added or removed. Users should be able to add items to the collection and remove them as needed.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

```
<!DOCTYPE html>
<html ng-app="itemApp">
<head>
 <title>AngularJS Item Collection</title>
 <style>
  #itemCollection {
   width: 400px;
   margin: auto;
   border: 1px solid #ccc;
   border-radius: 5px;
   padding: 20px;
   margin-top: 20px;
   text-align: center;
  }

  input {
   width: 100%;
```

```
    margin-bottom: 10px;
    padding: 5px;
   }

   ul {
    list-style-type: none;
    padding: 0;
   }

   li {
    margin-bottom: 10px;
   }
  </style>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="itemCtrl" id="itemCollection">
  <h2>Item Collection</h2>

  <label for="newItem">New Item:</label>
  <input type="text" id="newItem" ng-model="newItem" required>
  <button ng-click="addItem()">Add Item</button>

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

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

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

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

   $scope.addItem = function () {
    if ($scope.newItem.trim() !== "") {
     $scope.items.push($scope.newItem);
```

```
    $scope.newItem = "";
   }
  };

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

</body>
</html>
```

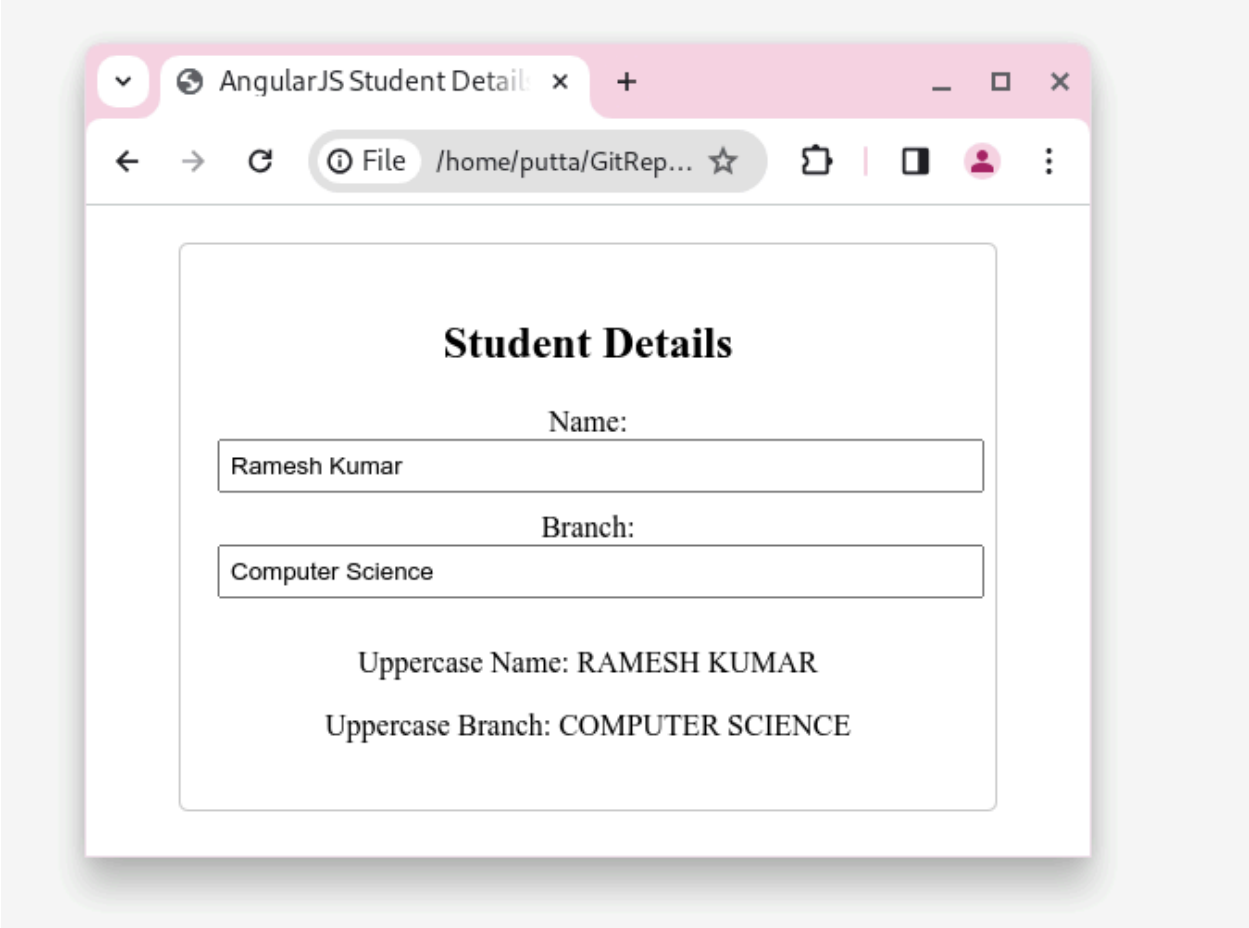


11. Create AngularJS application to convert student details to Uppercase using angular filters. Note: The default details of students may be included in the program.

```
    <!DOCTYPE html>
<html ng-app="studentApp">
<head>
 <title>AngularJS Student Details</title>
 <style>
  #studentDetails {
   width: 400px;
   margin: auto;
```

```
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 20px;
    margin-top: 20px;
    text-align: center;
   }

   input {
    width: 100%;
    margin-bottom: 10px;
    padding: 5px;
   }
  </style>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="studentCtrl" id="studentDetails">
  <h2>Student Details</h2>

  <label for="studentName">Name:</label>
  <input type="text" id="studentName" ng-model="student.name" required>

  <label for="studentBranch">Branch:</label>
  <input type="text" id="studentBranch" ng-model="student.branch" required>

  <p>Uppercase Name: {{ student.name | uppercase }}</p>
  <p>Uppercase Branch: {{ student.branch | uppercase }}</p>
</div>
```

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

  app.controller('studentCtrl', function ($scope) {
    $scope.student = {
      name: 'John Doe',
      branch: 'Computer Science'
    };
  });
</script>

</body>
</html>
```

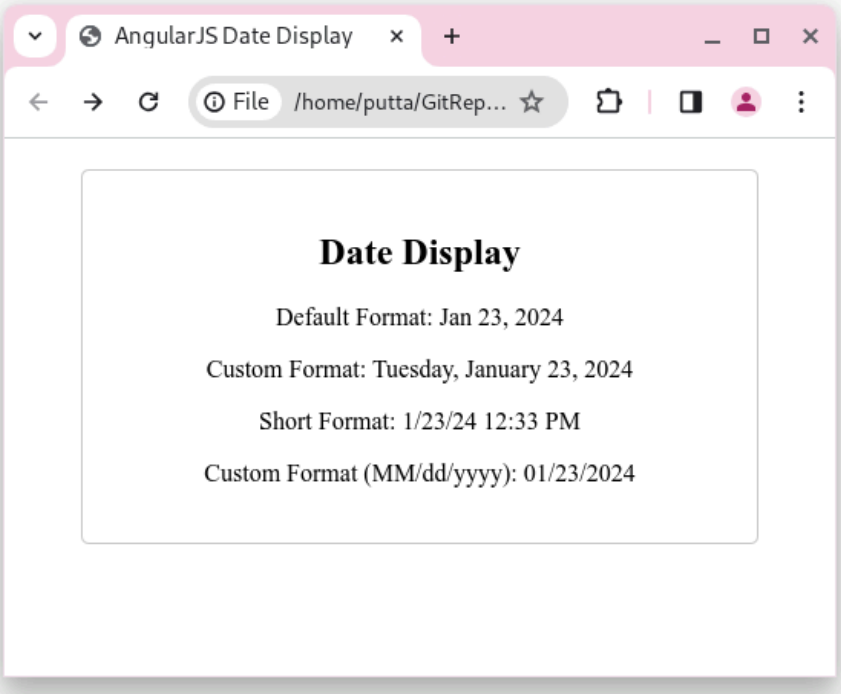


12. Create an AngularJS application that displays the date by using date filter parameters

```
<!DOCTYPE html>
<html ng-app="dateApp">
<head>
 <title>AngularJS Date Display</title>
 <style>
  #dateDisplay {
   width: 400px;
   margin: auto;
   border: 1px solid #ccc;
   border-radius: 5px;
   padding: 20px;
```

```
    margin-top: 20px;
    text-align: center;
   }
  </style>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body>

<div ng-controller="dateCtrl" id="dateDisplay">
  <h2>Date Display</h2>

  <p>Default Format: {{ currentDate | date }}</p>
  <p>Custom Format: {{ currentDate | date:'fullDate' }}</p>
  <p>Short Format: {{ currentDate | date:'short' }}</p>
  <p>Custom Format (MM/dd/yyyy): {{ currentDate | date:'MM/dd/yyyy' }}</p>
</div>

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

  app.controller('dateCtrl', function ($scope) {
    $scope.currentDate = new Date();
  });
</script>

</body>
</html>
```

Output


AngularJS Date Display
File
/home/putta/GitRep...
Date Display
Default Format: Jan 23, 2024
Custom Format: Tuesday, January 23, 2024
Short Format: 1/23/24 12:33 PM
Custom Format (MM/dd/yyyy): 01/23/2024