

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual
PART A

(Part A: TO BE REFERRED BY STUDENTS)

Experiment No. 08

A.1 AIM:

Implement Table and Form validation through Angular JS by using directive, controller, expression, modules etc.

A.2 Pre requisite:

HTML, CSS, JavaScript

A.3 Outcome:

After successful completion of this experiment students will be able to:

1. Understand and implement directives, expressions, controllers, scope in Angular JS.
2. Understand the principles behind data binding in Angular JS.

A.4 Theory:

Tables

Table data is generally repeatable. The ng-repeat directive can be used to draw table easily. The following example shows the use of ng-repeat directive to draw a table –

```
<table>
  <tr>
    <th>Name</th>
    <th>Marks</th>
  </tr>

  <tr ng-repeat = "subject in student.subjects">
    <td>{{ subject.name }}</td>
    <td>{{ subject.marks }}</td>
  </tr>
</table>
```

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

Example

The following example shows the use of all the above-mentioned directives.

```
<html>
<head>
  <title>Angular JS Table</title>
  <script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

  <style>
    table, th , td {
      border: 1px solid grey;
      border-collapse: collapse;
      padding: 5px;
    }
    table tr:nth-child(odd) {
      background-color: #f2f2f2;
    }
    table tr:nth-child(even) {
      background-color: #ffffff;
    }
  </style>
</head>

<body>
  <h2>AngularJS Sample Application</h2>
  <div ng-app = "mainApp" ng-controller = "studentController">

    <table border = "0">
      <tr>
        <td>Enter first name:</td>
        <td><input type = "text" ng-model = "student.firstName"></td>
      </tr>
      <tr>
        <td>Enter last name: </td>
        <td>
          <input type = "text" ng-model = "student.lastName">
        </td>
      </tr>
      <tr>
        <td>Name: </td>
        <td>{{ student.fullName() }}</td>
      </tr>
      <tr>
        <td>Subject:</td>
      </tr>
    </table>
  </div>
</body>
</html>
```

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

```
<td>
  <table>
    <tr>
      <th>Name</th>
      <th>Marks</th>
    </tr>
    <tr ng-repeat = "subject in student.subjects">
      <td>{{ subject.name }}</td>
      <td>{{ subject.marks }}</td>
    </tr>
  </table>
</td>
</tr>
</table>
</div>

<script>
  var mainApp = angular.module("mainApp", []);

  mainApp.controller('studentController', function($scope) {
    $scope.student = {
      firstName: "Mahesh",
      lastName: "Parashar",
      fees:500,

      subjects:[
        { name:'Physics',marks:70},
        { name:'Chemistry',marks:80},
        { name:'Math',marks:65},
        { name:'English',marks:75},
        { name:'Hindi',marks:67}
      ],
      fullName: function() {
        var studentObject;
        studentObject = $scope.student;
        return studentObject.firstName + " " + studentObject.lastName;
      }
    };
  });
</script>

</body>
</html>
```

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

Form Validation: AngularJS performs form validation on the client side. AngularJS monitors the state of the form and input fields (input, text-area, select), and notify the user about the current state. AngularJS also holds information about whether the input fields have been touched, modified, or not. Form input fields have the following states:

- `$untouched`: It shows that field has not been touched yet.
- `$touched`: It shows that field has been touched.
- `$pristine`: It represents that the field has not been modified yet.
- `$dirty`: It illustrates that the field has been modified.
- `$invalid`: It specifies that the field content is not valid.
- `$valid`: It specifies that the field content is valid.

AngularJS includes the following validation directives.

- `ng-required`: Sets required attribute on an input field.
- `ng-minlength`: Sets minlength attribute on an input field.
- `ng-maxlength`: Sets maxlength attribute on an input field. Setting the attribute to a negative or non-numeric value, allows view values of any length.
- `ng-pattern`: Sets pattern validation error key if the `ngModel` value does not match the specified RegEx expression.

These all are the properties of the input field which can be either true or false. Forms have the following states:

- `$pristine`: It represents that the fields have not been modified yet.
- `$dirty`: It illustrates that one or more fields have been modified.
- `$invalid`: It specifies that the form content is not valid.
- `$valid`: It specifies that the form content is valid.
- `$submitted`: It specifies that the form is submitted.

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

These all are the properties of the form which can be either true or false. These states can be used to show meaningful messages to the user.

ng-click

Reset data of a form using on-click directive of a button.

```
<input name = "firstname" type = "text" ng-model = "firstName" required>
<input name = "lastname" type = "text" ng-model = "lastName" required>
<input name = "email" type = "email" ng-model = "email" required>
<button ng-click = "reset()">Reset</button>

<script>
function studentController($scope) {
    $scope.reset = function() {
        $scope.firstName = "Mahesh";
        $scope.lastName = "Parashar";
        $scope.email = "MaheshParashar@tutorialspoint.com";
    }

    $scope.reset();
}
</script>
```

Example

The following example will showcase all the above-mentioned directives.

testAngularJS.htm

```
<html>
<head>
    <title>Angular JS Forms</title>
    <script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

    <style>
        table, th , td {
            border: 1px solid grey;
            border-collapse: collapse;
            padding: 5px;
        }
        table tr:nth-child(odd) {
```

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

```
background-color: #f2f2f2;
}
table tr:nth-child(even) {
background-color: #ffffff;
}
</style>

</head>
<body>

<h2>AngularJS Sample Application</h2>
<div ng-app = "mainApp" ng-controller = "studentController">

  <form name = "studentForm" novalidate>
    <table border = "0">
      <tr>
        <td>Enter first name:</td>
        <td><input name = "firstname" type = "text" ng-model = "firstName" required>
          <span style = "color:red" ng-show = "studentForm.firstname.$dirty &&
studentForm.firstname.$invalid">
            <span ng-show = "studentForm.firstname.$error.required">First Name is
required.</span>
          </span>
        </td>
      </tr>

      <tr>
        <td>Enter last name: </td>
        <td><input name = "lastname" type = "text" ng-model = "lastName" required>
          <span style = "color:red" ng-show = "studentForm.lastname.$dirty &&
studentForm.lastname.$invalid">
            <span ng-show = "studentForm.lastname.$error.required">Last Name is
required.</span>
          </span>
        </td>
      </tr>

      <tr>
        <td>Email: </td><td><input name = "email" type = "email" ng-model = "email" length =
"100" required>
          <span style = "color:red" ng-show = "studentForm.email.$dirty &&
studentForm.email.$invalid">
            <span ng-show = "studentForm.email.$error.required">Email is required.</span>
            <span ng-show = "studentForm.email.$error.email">Invalid email address.</span>
          </span>
        </td>
      </tr>
    </table>
  </form>
</div>
```

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

```
        </td>
    </tr>

    <tr>
        <td>
            <button ng-click = "reset()">Reset</button>
        </td>
        <td>
            <button ng-disabled = "studentForm.firstname.$dirty &&
            studentForm.firstname.$invalid || studentForm.lastname.$dirty &&
            studentForm.lastname.$invalid || studentForm.email.$dirty &&
            studentForm.email.$invalid" ng-click="submit()">Submit</button>
        </td>
    </tr>

</table>
</form>
</div>

<script>
    var mainApp = angular.module("mainApp", []);

    mainApp.controller('studentController', function($scope) {
        $scope.reset = function() {
            $scope.firstName = "Mahesh";
            $scope.lastName = "Parashar";
            $scope.email = "MaheshParashar@tutorialspoint.com";
        }

        $scope.reset();
    });
</script>

</body>
</html>
```

A.5 Procedure/Task:

1. Create an Angular JS application to implement Angular JS table to calculate total pay of order placed.
2. Design client-side validation for student registration form using angular JS. The angular JS validation should apply at input control level and formcontrol level with

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

following validations.

- User Name is RequiredField with Maxlength of 15-character Minlength of 05 character.
- Email is valid and can't be blank
- Age cannot be less than 18
- If form is not modified the background color is aqua else if one of field gets modified, thebackground color is lawn green.
- The submit button is disabled if form data is invalid.

3. To implement registration page form and form validation in Angular JS.

Create a form with the fields given below and validation is to be done for

1. Must not be empty **username**, start with capital letter, only have alphabets.
2. Must not be empty **password**, 5 and 15 characters long, must be alphanumeric.
3. Confirm **password**.
4. **Email** field and its basic validation.
5. Must not be empty **checkbox**. When the page loads, one checkbox must always be checked.
6. Atleast one **Radio button** must be selected.
7. Dynamic **dropdown** from controller. (i.e dropdown items must be populated from within the controller). Atleast one dropdown item must be selected.
8. **Phone Number**- 10 digits
9. **Pan Card Validation**

3. Prepare the document. Save and close the file and name it as **EXP08_Name of Student**

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual
PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll No. :E258	Name: Kundan S. Patil
Class :BTech CS	Batch :A3
Date of Experiment :18-03-2025	Date/Time of Submission :19-03-2025
Grade :	

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

B.1 Code:

```
1  <!DOCTYPE html>
2  <html Lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7      <script src="https://code.angularjs.org/1.8.2/angular.min.js"></script>
8  </head>
9  <body>
10     <div ng-app="MyApp" ng-controller="ctrl">
11         <h2>Food Items</h2>
12         <table border="1">
13             <tr>
14                 <th>Product Item</th>
15                 <th>Price</th>
16                 <th>Quantity</th>
17             </tr>
18             <tr ng-repeat="x in FoodItems">
19                 <td>{{x.name}}</td>
20                 <td>{{x.price}}</td>
21                 <td><input type="number" ng-model="x.qyt"></td>
22             </tr>
23         </table>
24         <br>
25         <input type="button" ng-click="show()" value="Calculate">
26         <h1>Payment Order: {{total}}</h1>
27     </div>
28
29     <script>
30         var app = angular.module('MyApp', []);
31         app.controller('ctrl', function($scope) {
32             // Food items array
33             $scope.FoodItems = [
34                 {name: 'Pizza', price: 200, qyt: 1},
35                 {name: 'Bread', price: 50, qyt: 1},
36                 {name: 'Banana', price: 5, qyt: 1}
37             ];
38
39             // Function to calculate total payment
40             $scope.show = function() {
41                 var tot = 0;
42                 // Loop through the items to calculate the total payment
43                 for (var i = 0; i < $scope.FoodItems.length; i++) {
44                     tot += $scope.FoodItems[i].price * $scope.FoodItems[i].qyt;
45                 }
46                 $scope.total = tot; // Update the total value
47             };
48         });
49     </script>
50 </body>
51 </html>
```

B.2 Output:

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

Food Items

Product Item	Price	Quantity
Pizza	200	3
Bread	50	2
Banana	5	1

Calculate

Payment Order: 705

B.1 Code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <script src="https://code.angularjs.org/1.8.2/angular.min.js"></script>
7   <title>Form Validation</title>
8   <style>
9     form.ng-dirty {
10       background-color: ■ rgb(226, 253, 199);
11     }
12
13     form.ng-pristine {
14       background-color: ■ rgb(148, 255, 255);
15     }
16   </style>
17 </head>
18 <body ng-app="MyApp">
19
20   <div ng-controller="FormController">
21     <h2>Student Registration Form</h2>
22
23     <form name="myform" ng-submit="submitForm()" ng-class="{ 'ng-dirty': myform.$dirty, 'ng-pristine': myform.$pristine }" novalidate>
24       <label for="name">User Name (5-15 characters, starts with capital letter)</label>
25       <input type="text" name="username" ng-model="username" required minlength="5" maxlength="15" ng-pattern="/^[A-Z][a-zA-Z]*$/ " ng-class="{ 'is-invalid': myform.username.$invalid && myform.username.$dirty }" />
26       <span ng-show="myform.username.$dirty && myform.username.$invalid">
27         User Name is required, should start with a capital letter, and contain only alphabets (5-15 characters).
28       </span>
29       <br><br>
30
31       <label for="email">Email:</label>
32       <input type="email" name="email" ng-model="email" required ng-class="{ 'is-invalid': myform.email.$invalid && myform.email.$dirty }" />
33       <span ng-show="myform.email.$dirty && myform.email.$invalid">
34         Please enter a valid email.
35       </span>
36       <br><br>
37
38       <label for="age">Age:</label>
39       <input type="number" name="age" ng-model="age" required min="18" ng-class="{ 'is-invalid': myform.age.$invalid && myform.age.$dirty }" />
40       <span ng-show="myform.age.$dirty && myform.age.$invalid">
41         Age must be at least 18.
42       </span>
43       <br><br>
44
45       <input type="submit" value="Submit" ng-disabled="myform.$invalid" />
46     </form>
47   </div>
48 </body>
49 </html>
```

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

```
47     </form>
48 </div>
49
50 <script>
51     var app = angular.module("MyApp", []);
52
53     app.controller("FormController", function($scope) {
54 +         $scope.username = "";
55         $scope.email = "";
56         $scope.age = "";
57
58         $scope.submitForm = function() {
59             if ($scope.myform.$valid) {
60                 alert("Form submitted successfully!");
61             } else {
62                 alert("Please correct the errors in the form.");
63             }
64         };
65     });
66 </script>
67 </body>
68 </html>
```

B.2 Output:

Student Registration Form

User Name (5-15 characters, starts with capital letter):

Email:

Age:

B.1 Code:

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Registration Form with Validations</title>
7   <script src="https://code.angularjs.org/1.8.2/angular.min.js"></script>
8   <style>
9     .is-invalid {
10       border-color: red;
11     }
12     span {
13       color: red;
14       font-size: 0.9em;
15     }
16     form.ng-dirty {
17       background-color: lawngreen;
18     }
19     form.ng-pristine {
20       background-color: aqua;
21     }
22   </style>
23 </head>
24 <body ng-app="MyApp">
25
26   <div ng-controller="FormController">
27     <h2>User Registration Form</h2>
28
29     <form name="registrationForm" ng-submit="submitForm()" novalidate>
30
31       <label for="username">User Name (Start with a capital letter and only alphabets)</label>
32       <input type="text" name="username" ng-model="username" required ng-pattern="/^[A-Z][a-zA-Z]*$/> ng-class="{ 'is-invalid': registrationForm.username.$invalid && registrationForm.username.$dirty}" />
33       <span ng-show="registrationForm.username.$dirty && registrationForm.username.$invalid">User Name is required and must start with a capital letter.</span>
34       <br><br>
35
36       <label for="password">Password (5-15 characters, alphanumeric)</label>
37       <input type="password" name="password" ng-model="password" required ng-minlength="5" ng-maxlength="15" ng-pattern="/^[a-zA-Z0-9]*$/> ng-class="{ 'is-invalid': registrationForm.password.$invalid && registrationForm.password.$dirty}" />
38       <span ng-show="registrationForm.password.$dirty && registrationForm.password.$invalid">Password is required, must be alphanumeric, and 5-15 characters long.</span>
39       <br><br>
40
41       <label for="confirmPassword">Confirm Password</label>
42       <input type="password" name="confirmPassword" ng-model="confirmPassword" required ng-match="password" ng-class="{ 'is-invalid': registrationForm.confirmPassword.$invalid && registrationForm.confirmPassword.$dirty}" />
43       <span ng-show="registrationForm.confirmPassword.$dirty && registrationForm.confirmPassword.$invalid">Passwords must match.</span>
44       <br><br>
45
46       <label for="email">Email</label>
47       <input type="email" name="email" ng-model="email" required ng-class="{ 'is-invalid': registrationForm.email.$invalid && registrationForm.email.$dirty}" />
48       <span ng-show="registrationForm.email.$dirty && registrationForm.email.$invalid">Please enter a valid email.</span>
49       <br><br>
50
51       <label for="terms">Accept Terms and Conditions</label>
52       <input type="checkbox" name="terms" ng-model="terms" required ng-checked="true" />
53       <span ng-show="registrationForm.terms.$invalid && registrationForm.terms.$dirty">You must accept the terms and conditions.</span>
54       <br><br>
55
56       <label>Gender</label>
57       <input type="radio" name="gender" ng-model="gender" value="Male" required /> Male
58
59       <input type="radio" name="gender" ng-model="gender" value="Female" required /> Female
60       <span ng-show="registrationForm.gender.$invalid && registrationForm.gender.$dirty">Please select your gender.</span>
61       <br><br>
62
63       <label for="country">Country</label>
64       <select name="country" ng-model="country" ng-options="country for country in countries" required ng-class="{ 'is-invalid': registrationForm.country.$invalid && registrationForm.country.$dirty}">
65         <option value="">-- Select Country --</option>
66       </select>
67       <span ng-show="registrationForm.country.$invalid && registrationForm.country.$dirty">Please select a country.</span>
68       <br><br>
69
70       <label for="phone">Phone Number (10 digits)</label>
71       <input type="text" name="phone" ng-model="phone" ng-pattern="/^\d{10}$/> required ng-class="{ 'is-invalid': registrationForm.phone.$invalid && registrationForm.phone.$dirty}" />
72       <span ng-show="registrationForm.phone.$dirty && registrationForm.phone.$invalid">Phone number must be 10 digits long.</span>
73       <br><br>
74
75       <label for="panCard">PAN Card (Format: ABCDE1234F)</label>
76       <input type="text" name="panCard" ng-model="panCard" ng-pattern="^[A-Z](5)[0-9](4)[A-Z](1)$> required ng-class="{ 'is-invalid': registrationForm.panCard.$invalid && registrationForm.panCard.$dirty}" />
77       <span ng-show="registrationForm.panCard.$dirty && registrationForm.panCard.$invalid">Please enter a valid PAN card number.</span>
78       <br><br>
79
80       <input type="submit" value="Submit" ng-disabled="registrationForm.$invalid" />
81     </form>
82   </div>
83
84   <script>
85     var app = angular.module('MyApp', []);
86
87     app.controller('FormController', function($scope) {
88
89       $scope.countries = ['India', 'USA', 'UK', 'Canada', 'Australia'];
90
91       $scope.username = '';
92       $scope.password = '';
93       $scope.confirmPassword = '';
94       $scope.email = '';
95       $scope.terms = false;
96       $scope.gender = '';
97       $scope.country = '';
98       $scope.phone = '';
99       $scope.panCard = '';
100
101       $scope.submitForm = function() {
102         if ($scope.registrationForm.$valid) {
103           alert('Form submitted successfully!');
104         } else {
105           alert('Please correct the errors in the form.');
```

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

B.2 Output:

User Registration Form

User Name (Start with a capital letter and only alphabets): Kundanpatil

Password (5-15 characters, alphanumeric): *****

Confirm Password: *****

Email: kundan@gmail.com

Accept Terms and Conditions: ☒

Gender: ☒ Male ☐ Female

Country: India

Phone Number (10 digits): 9309411001

PAN Card (Format: ABCDE1234F): KPRYX1354M

Submit

B.3 Conclusion:

In the context of all the AngularJS implementations above, we have explored how to build a **student registration form** with complex validations, a **food order calculator**, and **dynamic form handling**. The key objective was to ensure that forms collect accurate data while providing users with an interactive and responsive interface.

Key Points:

1. Form Validation with AngularJS:

- We utilized AngularJS's built-in directives such as `ng-model`, `ng-required`, `ng-pattern`, `ng-minlength`, and `ng-maxlength` for both **field-level** and **form-level validation**. This allows for real-time feedback, ensuring that users input valid data before submitting the form.

2. Dynamic Content Handling:

- For dynamic dropdowns (like the country list) and calculation-based outputs (such as the total price of items), AngularJS's **two-way data binding** and **directives** such as `ng-repeat` and `ng-options` made it easy to update and reflect data without needing manual DOM manipulation.

3. User Feedback and Experience:

- AngularJS provided **real-time validation feedback**, changing the form's background color and displaying error messages when data was invalid. The **submit button** was disabled until all required fields passed validation, enhancing user experience and preventing incorrect data submission.

4. Efficient Form Submission and Data Integrity:

SVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering (Shirpur Campus)
Computer Science Department (B Tech Sem IV)
Web Programming
Lab Manual

- With **dynamic validation rules** and **clear error messages**, users were guided step-by-step in correctly filling out the form. The use of **ng-disabled** for the submit button ensured that the form could not be submitted until all validation checks were passed.

B.3 Observations and Learning:

Learnings:

- **AngularJS Makes Form Validation Easy:** The use of AngularJS's directives simplifies form validation and handling, making it easier to manage complex forms with multiple validation rules.
- **Two-Way Data Binding is Powerful:** AngularJS's two-way data binding allows dynamic updates to both the form and the model, which ensures that the UI and underlying data are always synchronized.
- **User Experience is Key:** Real-time feedback, color changes, and error messages ensure that users can easily fill out the form, reducing frustration and error rates.

Observations:

- **Maintainability:** The modular approach with AngularJS makes the code easy to maintain and extend for future features.
- **Dynamic Forms:** AngularJS's flexibility allows for dynamically generated content and easy management of forms with different input types.
- **Validation Flexibility:** With AngularJS, handling complex validations (like custom email formats, phone number lengths, etc.) was straightforward.