

ASSIGNMENT 6

AIM: Implement student registration form in Angular JS

Theory:

Form validation in Angular allows you to ensure that user input meets certain requirements before it is submitted. Angular provides both template-driven and reactive approaches to form validation. Here's some theory about form validation in Angular:

1. Template-Driven Forms:

- In template-driven forms, validation is performed using directives such as **ngModel** and template reference variables.
- Angular provides several built-in validators like **required**, **minLength**, **maxLength**, **pattern**, etc.
- You can also create custom validators by writing your own functions.

2. Reactive Forms:

- Reactive forms are model-driven and defined programmatically using TypeScript.
- Validators are applied directly to form controls using functions provided by Angular's **Validators** class.
- Reactive forms offer more flexibility and control compared to template-driven forms, especially for complex validation scenarios.

3. Validation Directives:

- Angular provides various validation directives such as **ngModel**, **ngModelGroup**, **ngForm**, and **formControl** for handling form validation.
- These directives enable you to apply validation rules and display error messages based on the state of form controls.

4. Displaying Validation Errors:

- Angular provides several ways to display validation errors to users.
- You can use Angular's built-in directives like **ngIf** and **ngClass** to conditionally show error messages based on the validity state of form controls.
- You can also customize the appearance of error messages using CSS and Angular's template syntax.

5. Cross-Field Validation:

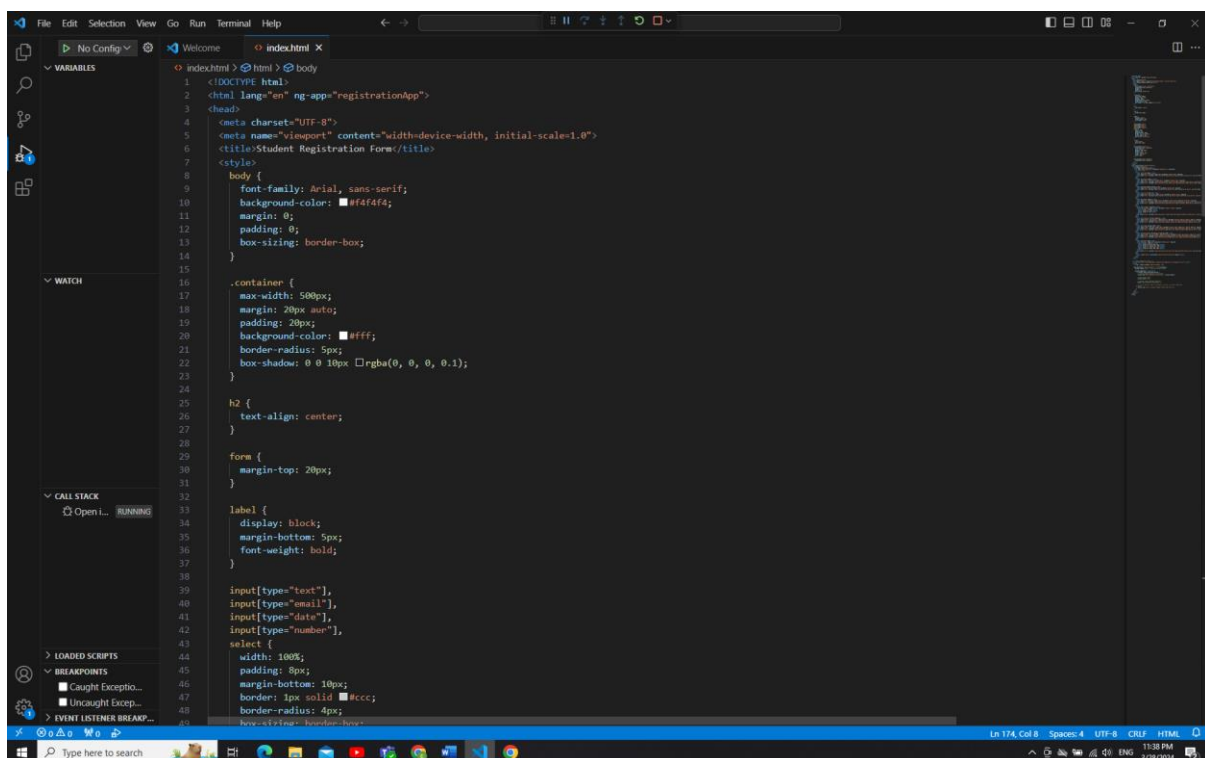
- Angular allows you to perform validation that depends on the values of multiple form controls.
- You can implement cross-field validation by creating custom validators or by using built-in validators like **Validators** class methods.

6. Asynchronous Validation:

- In addition to synchronous validation, Angular supports asynchronous validation for cases where validation rules depend on data fetched from external sources.
- Asynchronous validators return **Observable** or **Promise** objects and are applied using the **AsyncValidatorFn** function.

7. Handling Form Submission:

- Once form validation is successful, you can handle form submission by calling methods or functions defined in your component.
- You can prevent the default form submission behavior by calling **event.preventDefault()** in the submit handler function.



The screenshot shows a web browser window displaying a student registration form. The form is titled "Student Registration Form" and is styled with a dark background and light-colored text. It includes several input fields for registration details, a "Submit" button, and a "Cancel" button. The form is styled with a dark background and light-colored text. The browser's developer tools are open, showing the HTML and CSS for the page. The HTML structure includes a form with inputs for name, email, date, and number, and a select dropdown. The CSS defines the layout and styling of the form elements.

```

1 <!DOCTYPE html>
2 <html lang="en" ng-app="registrationApp">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Student Registration Form</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      background-color: #f4f4f4;
11      margin: 0;
12      padding: 0;
13      box-sizing: border-box;
14    }
15
16    .container {
17      max-width: 500px;
18      margin: 20px auto;
19      padding: 20px;
20      background-color: #fff;
21      border-radius: 5px;
22      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
23    }
24
25    h2 {
26      text-align: center;
27    }
28
29    form {
30      margin-top: 20px;
31    }
32
33    label {
34      display: block;
35      margin-bottom: 5px;
36      font-weight: bold;
37    }
38
39    input[type="text"],
40    input[type="email"],
41    input[type="date"],
42    input[type="number"],
43    select {
44      width: 100%;
45      padding: 5px;
46      margin-bottom: 10px;
47      border: 1px solid #ccc;
48      border-radius: 4px;
49      box-sizing: border-box;
50    }
51
52    .submit-button {
53      width: 100%;
54      padding: 10px;
55      background-color: #007bff;
56      color: white;
57      border: none;
58      border-radius: 4px;
59      cursor: pointer;
60    }
61
62    .cancel-button {
63      width: 100%;
64      padding: 10px;
65      background-color: #6c757d;
66      color: white;
67      border: none;
68      border-radius: 4px;
69      cursor: pointer;
70    }
71  </style>
72 </head>
73 <body>
74   <h2>Student Registration Form</h2>
75   <form>
76     <label>Name</label>
77     <input type="text">
78     <label>Email</label>
79     <input type="email">
80     <label>Date</label>
81     <input type="date">
82     <label>Number</label>
83     <input type="number">
84     <select>
85       <option>Gender</option>
86       <option>Male</option>
87       <option>Female</option>
88     </select>
89     <button type="submit">Submit</button>
90     <button type="button" value="Cancel">Cancel</button>
91   </form>
92 </body>
93 </html>

```

```
File Edit Selection View Go Run Terminal Help
index.html x
index.html > html > body
57 button[type="submit"] {
58   background-color: #4CAF50;
59   color: white;
60   padding: 10px 20px;
61   border: none;
62   border-radius: 4px;
63   cursor: pointer;
64   width: 100%;
65 }
66
67 button[type="submit"]:hover {
68   background-color: #45a049;
69 }
70 </style>
71 </head>
72 <body ng-controller="registrationCtrl">
73   <div class="container">
74     <h2>Student Registration Form</h2>
75     <form name="registrationForm" ng-submit="submitForm()" novalidate>
76       <!-- Form fields -->
77       <div>
78         <label for="name">Name:</label>
79         <input type="text" id="name" name="name" ng-model="student.name" required>
80         <span class="error" ng-show="registrationForm.name.$dirty && registrationForm.name.$error.required">Name is required.</span>
81       </div>
82       <div>
83         <label for="email">Email:</label>
84         <input type="email" id="email" name="email" ng-model="student.email" required>
85         <span class="error" ng-show="registrationForm.email.$dirty && registrationForm.email.$error.required">Email is required.</span>
86         <span class="error" ng-show="registrationForm.email.$dirty && registrationForm.email.$error.email">Invalid email format.</span>
87       </div>
88       <div>
89         <label for="dob">Date of Birth:</label>
90         <input type="date" id="dob" name="dob" ng-model="student.dob" required>
91         <span class="error" ng-show="registrationForm.dob.$dirty && registrationForm.dob.$error.required">Date of Birth is required.</span>
92       </div>
93       <div>
94         <label for="rollno">Roll No:</label>
95         <input type="text" id="rollno" name="rollno" ng-model="student.rollno" required>
96         <span class="error" ng-show="registrationForm.rollno.$dirty && registrationForm.rollno.$error.required">Roll No is required.</span>
97       </div>
98       <div>
99         <label for="phone">Phone:</label>
100        <input type="tel" id="phone" name="phone" ng-model="student.phone" pattern="[0-9]{10}" required>
101        <span class="error" ng-show="registrationForm.phone.$dirty && registrationForm.phone.$error.required">Phone number is required.</span>
102        <span class="error" ng-show="registrationForm.phone.$dirty && registrationForm.phone.$error.pattern">Invalid phone number format (should be 10 digits).</span>
103      </div>
104    </div>
105    <div>
106      <label for="year">Year:</label>
107      <select id="year" name="year" ng-model="student.year" required>
108        <option value="">Select</option>
109        <option value="YE">First Year (YE)</option>
110        <option value="SE">Second Year (SE)</option>
111        <option value="TE">Third Year (TE)</option>
112        <option value="BE">Final Year (BE)</option>
113      </select>
114      <span class="error" ng-show="registrationForm.year.$dirty && registrationForm.year.$error.required">Year is required.</span>
115    </div>
116    <div>
117      <button type="submit" ng-disabled="registrationForm.$invalid">Submit</button>
118    </div>
119  </div>
120 </body>
121 </html>
```

```
File Edit Selection View Go Run Terminal Help
index.html x
index.html > html > body
131 <div>
132   <label for="year">Year:</label>
133   <select id="year" name="year" ng-model="student.year" required>
134     <option value="">Select</option>
135     <option value="YE">First Year (YE)</option>
136     <option value="SE">Second Year (SE)</option>
137     <option value="TE">Third Year (TE)</option>
138     <option value="BE">Final Year (BE)</option>
139   </select>
140   <span class="error" ng-show="registrationForm.year.$dirty && registrationForm.year.$error.required">Year is required.</span>
141 </div>
142 <div>
143   <button type="submit" ng-disabled="registrationForm.$invalid">Submit</button>
144 </div>
145 </form>
146 </div>
147
148 <!-- Import AngularJS library -->
149 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
150 <script>
151   var app = angular.module('registrationApp', []);
152
153   app.controller('registrationCtrl', function($scope) {
154     $scope.student = {}; // Initialize student object
155
156     $scope.submitForm = function() {
157       if ($scope.registrationForm.$valid) {
158         // Log the student details in the console
159         console.log('Form submitted successfully:', $scope.student);
160
161         // Reset form fields
162         $scope.student = {};
163
164         // Display success message (optional)
165         alert('Form submitted successfully!');
166
167         // You can add your logic to send data to server or do other operations
168       } else {
169         console.log('Form is invalid. Please correct the errors.');
```

Student Registration Form

Name:

Email:

Date of Birth:

mm/dd/yyyy

Roll No:

Phone:

Live KTs:

Select

Current Semester:

Average CGPA:

Percentage of Average CGPA:

Year:

Select

Submit