Form validations

This feature improves data quality by validating user input for accuracy and completeness.

So that it can validate user input in the UI and display useful validation messages.

HTML5 introduced new mechanisms for forms: it added new semantic types for the <input> element and *constraint validation* to ease the work of checking the form content on the client side

Intrinsic and basic constraints

The intrinsic constraints for the `type` attribute are

Input type	Constraint description	Associated violation
`<input type="URL">`	The value must be an absolute URL, as defined in the URL Living Standard.	**TypeMismatch** constraint violation
`<input type="email">`	The value must be a syntactically valid email address, which generally has the format *`username@hostname.tld`*.	**TypeMismatch** constraint violation

Validation-related attributes

In addition to the type attribute described above, the following attributes are used to describe basic constraints:

Attribute	Description
required	There must be a value
minlength	The number of characters must not be less than the value of the attribute
maxlength	The number of characters must not exceed the value of the attribute.
pattern	The value must match the pattern

Form validation in Angular

To add validation to a template-driven form, you add the same validation attributes. Angular uses directives to match these attributes with validator functions in the framework.

Every time the value of a form control changes, Angular runs validation and generates either a list of validation errors, which results in an INVALID status, or null, which results in a VALID status

We can then inspect the control's state by exporting ngModel to a local template variable.

Angular form validation properties(classes)

ng-valid	**ng-invalid**
ng-touched	**ng-untouched**
ng-pristine	**ng-dirty**

With the help of above classes ,angular can check validity of data of an input field. All of the above properties can return boolean value. To check the status of these classes with respect to an input field,use the following

```
<input type=”text”  #ref  name=”name”  ngModel>

{{ref.className}}
```

It will print the class names to web page which are true. One can observe the changes by interacting with that input field.

test for required

***ngIf="ref.touched && ref.invalid"**

test for minlength

***ngIf="ref.errors.minlength"**

test for pattern

***ngIf="ref.errors.pattern"**

test for required&minlength

<div *ngIf="r.invalid && (r.dirty || r.touched)" class="alert alert-danger">

<div *ngIf="r.errors?.required">

Name is required.

** **

<div *ngIf="r.errors?.minlength">

Name must be at least 4 characters long.

** **

** **

Sample code for your verification

```
<div class="row">
    <div class="col-sm-3"></div>
    <div class="col-sm-6">
        <form  #ref="ngForm" (ngSubmit)="doLogin(ref)">
            <!--create radio buttons to represent role-->
            <label for="">Choose role</label>
            <div class="form-check">
                <input type="radio" name="role" id="adm" class="form-check-
input"  value="admin" #r="ngModel" ngModel required>
                <label for="adm" class="form-check-label">Admin</label>
            </div>
            <div class="form-check">
                <input type="radio" name="role" id="usr" class="form-check-
input" value="user" ngModel required>
```

```
                <label for="usr" class="form-check-label">User</label>
            </div>


            <div *ngIf="r.invalid &&ref.submitted" class="alert alert-
danger">*Role selection is required</div>









            <!--username-->
            <div class="form-group">
                <label for="un">Username</label>
                <input type="text" name="username" id="un"  ngModel class="form-
control" required #ref1="ngModel">
            </div>

            <!--violation of username-->
            <div *ngIf="ref1.touched&&ref1.invalid" class="alert alert-danger">
                *Username is required
            </div>

            <!--password-->
            <div class="form-group">
                <label for="pw">Password</label>
                <input type="text" name="password" id="pw"  ngModel class="form-
control"  required  pattern="\d[abc]\d.*"  #ref2="ngModel">
            </div>


          <!--violation of password-->

                            <div *ngIf="ref2.invalid && (ref2.dirty || ref2.touched)"
                            class="alert alert-danger">

                                    <div *ngIf="ref2.errors?.required">
                                    Name is required.
                                    </div>
                                    <div *ngIf="ref2.errors?.pattern">
                                   Pattern is not matched
                                    </div>
                            </div>

            <!--login button-->
            <div class="text-center">
                <button type="submit" class="btn btn-
success" [disabled]="ref.invalid">Login</button>
            </div>
        </form>
    </div>
    <div class="col-sm-3"></div>
```

```
</div>
```