<div class="container">

<div class="jumbotron">

<form #frm="ngForm" (ngSubmit)="addUser(frm)">

<div class="form-group">

<label for="name">Name:</label>

<input type="text" name="name" class="form-control" id="" ngModel>

</div>

<div class="form-group">

<label for="">Contact:</label>

<input type="text" name="contact" id="" class="form-control" ngModel>

</div>

<div class="form-group">

<label for="">Email:</label>

<input type="text" name="email" id="" class="form-control" ngModel>

</div>

<button type="submit" name="button" class="btn btn-success">Add User</button>

</form>

</div>

</div>

{{frm.value | json}}

import { Component, OnInit } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent implements OnInit{

title = 'demo-app';

addUser(form){

console.log(form.value);

}

constructor(){}

ngOnInit(){}

}

Create a model for the form

**User.ts**

export interface User{

name: string;

contact:string;

email:string;

}

Disable button when form entries are invalid.

<button type="submit" name="button" class="btn btn-success" [disabled]="frm.invalid">Add User</button>

Make all fields required

<input type="text" name="name" class="form-control" id="" required ngModel>

Add pattern matches to input fields.

Name cannot have digit.

pattern="[a-zA-Z][a-zA-Z]+"

Contact can have only characters.

pattern="[0-9]\*"

Email field validation

pattern="^(([A-Za-z0-9]+\_+)|([A-Za-z0-9]+\-+)|([A-Za-z0-9]+\.+)|([A-Za-z0-9]+\++))\*[A-Za-z0-9]+@((\w+\-+)|(\w+\.))\*\w{1,63}\.[a-zA-Z]{2,6}$"

Display error message when user enters invalid.

<div class="form-group">

<label for="name">Name:</label>

<input type="text" name="name" class="form-control" id="" required pattern="[a-zA-Z][a-zA-Z]+"

ngModel #username="ngModel">

</div>

<div \*ngIf="username?.touched" class="alert alert-danger">

<div \*ngIf="username.errors?.required">

Name is required.

</div>

</div>

<div class="container">

<div class="jumbotron">

<form #frm="ngForm" (ngSubmit)="addUser(frm)">

<div class="form-group">

<label for="name">Name:</label>

<input type="text" name="name" class="form-control" id="" required pattern="[a-zA-Z][a-zA-Z]+"

ngModel #username="ngModel">

</div>

<div \*ngIf="username?.touched" class="alert alert-danger">

<div \*ngIf="username.errors?.required">

Name is required.

</div>

<div \*ngIf="username.errors?.pattern">

Invalid Name!

</div>

</div>

<div class="form-group">

<label for="">Contact:</label>

<input type="text" name="contact" id="" class="form-control" required pattern="[0-9]\*"

#contact="ngModel" ngModel maxlength="10" minlength="10">

</div>

<div \*ngIf="contact?.touched" class="alert alert-danger">

<div \*ngIf="contact.errors?.required">

Contact is required.

</div>

<div [hidden]="!contact?.hasError('minlength')" >

Invalid Contact!

</div>

</div>

<div class="form-group">

<label for="">Email:</label>

<input type="text" name="email" id="" class="form-control" required pattern="^[\_\.0-9a-z-]+@([0-9a-z][0-9a-z-]+)+((\.)[a-z]{2,})+$"

ngModel #email="ngModel">

</div>

<div \*ngIf="email?.touched" class="alert alert-danger">

<div \*ngIf="email.errors?.required">

Email is required.

</div>

<div \*ngIf="email.errors?.pattern">

Invalid Email!

</div>

</div>

<button type="submit" name="button" class="btn btn-success" [disabled]="frm.invalid">Add User</button>

</form>

</div>

</div>

<ul class="list-group">

<li \*ngFor="let user of userList" class="list-group-item">

{{user?.name}}

</li>

</ul>

{{frm.value | json}}

import { Component, OnInit } from '@angular/core';

import {user} from './user';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent implements OnInit{

userList: user[]=[];

addUser(form){

var usr: user;

usr=form.value;

console.log(form.value);

}

constructor(){}

ngOnInit(){}

}