

## Create password validator with following constraints

### app.component.css

```
p {  
  font-family: Lato;  
}  
  
small {  
  padding: 0 0.25rem;  
}  
input {  
  margin-bottom: 1rem;  
}  
  
.errors {  
  color: red;  
}
```

### app.component.html

```
<form [formGroup]="myForms">  
  
  <label>Password :  
    <input type="text" name="PwdHtm" formControlName="password">  
  </label><br>  
  
  <div class="errors" *ngIf="myForms.get('password').invalid &&  
(myForms.get('password').touched || myForms.get('password').dirty)">  
  
    <div *ngIf="myForms.get('password').hasError('required')">  
      password is required  
    </div>  
  
    <div *ngIf="myForms.get('password').errors.minlength">  
      Password must be at least 8 characters  
    </div>
```

```

        <div class="error-text"
*ngIf="myForms.get('password').hasError('passwordStrength')">
            {{myForms.get('password').errors['passwordStrength']}}
        </div>
    </div>

    <button [disabled]="!myForms.valid">Submit</button>
</form>

```

### app.component.ts

```

import { Component } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from
"@angular/forms"
import { PasswordStrengthValidator } from "../password-strength.validators"

@Component({
  selector: 'my-app',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  public myForms: FormGroup;

  constructor(fb: FormBuilder) {

    this.myForms = fb.group({
      password: [null, Validators.compose([
        Validators.required, Validators.minLength(8), PasswordStrengthValidator])]
    });
  }
}

```

### app.module.ts

```

import { NgModule } from '@angular/core';

```

```

import { BrowserModule } from '@angular/platform-browser';
import { FormsModule , ReactiveFormsModule } from '@angular/forms';

import { AppComponent } from './app.component';

@NgModule({
  imports:    [ BrowserModule, FormsModule ,ReactiveFormsModule ],
  declarations: [ AppComponent ],
  bootstrap: [ AppComponent ]
})
export class AppModule { }

```

### [password-strength.validators.ts](#)

```

import { AbstractControl, ValidationErrors } from "@angular/forms"

export const PasswordStrengthValidator = function (control: AbstractControl):
ValidationErrors | null {

  let value: string = control.value || "";
  let msg = "";
  if (!value) {
    return null
  }

  let upperCaseCharacters = /[A-Z]+/g;
  let lowerCaseCharacters = /[a-z]+/g;
  let numberCharacters = /[0-9]+/g;
  let specialCharacters = /[!@#$%^&*()_+\-=\[\]{};':"\"|,.<>\/?]+/;
  if (upperCaseCharacters.test(value) === false || lowerCaseCharacters.test(value) ===
false || numberCharacters.test(value) === false || specialCharacters.test(value) === false)
  {
    return {
      passwordStrength: 'Password must contain at least two of the following: numbers,
lowercase letters, uppercase letters, or special characters.'
    }
  }
}

```

}

## 1. Create url validator

### app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
@NgModule({
  declarations: [AppComponent],
  imports: [BrowserModule, ReactiveFormsModule, FormsModule],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

### app.component.ts

```
import { Component } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
```

```

export class AppComponent {

  public myForm: FormGroup;

  constructor(private formBuilder: FormBuilder) {
    this.myForm = formBuilder.group({
      url: ['', [Validators.required,
Validators.pattern('(https?://)?([\\da-z.-]+)\\.([a-z.]{2,6})([/\\w .-]*/?')]]
    ])
  }

  get m(){
    return this.myForm.controls;
  }

  onSubmit(){
    console.log(this.myForm.value);
  }

}

```

## app.component.html

```

<div class="container mt-5">
  <h2>Angular Pattern URL Validation Example</h2>
  <form [formGroup]="myForm" (ngSubmit)="onSubmit()" novalidate>
    <div class="form-group">
      <label>Enter URL</label>
      <input type="text" formControlName="url" class="form-control mb-2">
    
```

```
<div *ngIf="m.url.touched && m.url.invalid" class="alert alert-danger">
  <div *ngIf="m.url.errors?.required">Please provide url</div>
  <div *ngIf="m.url.errors?.pattern">Please provide valid url</div>
</div>
</div>
<div class="d-grid mt-3">
  <button class="btn btn-dark" [disabled]="!myForm.valid"
type="submit">Store</button>
</div>
</form>
</div>
```