Routing Demo

App.module.ts

import { BrowserModule } from '@angular/platform-browser';

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

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

import { HomeComponent } from './home/home.component';

import { PageNotFoundComponent } from './page-not-found/page-not-found.component';

import{ Routes, RouterModule } from '@angular/router'

// Routes is an array of Route objects

// Each route maps a URL path to a component

// The 2nd route specifies the route to redirect to if the path

// is empty. In our case we are redirecting to /home

// The 3rd route (\*\*) is the wildcard route. This route is used

// if the requested URL doesn't match any other routes already defined

const appRoutes: Routes = [

{ path: 'home', component: HomeComponent },

{ path: '', redirectTo: '/home', pathMatch: 'full' },

{ path: '\*\*', component: PageNotFoundComponent }

];

// To let the router know about the routes defined above,

// pass "appRoutes" constant to forRoot(appRoutes) method of RouterModule

@NgModule({

declarations: [

AppComponent,

HomeComponent,

PageNotFoundComponent

],

imports: [

BrowserModule,

RouterModule.forRoot(appRoutes)

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

HomeComponent.ts

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

@Component({

//selector: 'app-home',

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

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

})

export class HomeComponent {}

PageNotFoundComponent.ts

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

@Component({

// selector: 'app-page-not-found',

templateUrl: './page-not-found.component.html',

styleUrls: ['./page-not-found.component.css']

})

export class PageNotFoundComponent { }

app.component.html

<!--The content below is only a placeholder and can be replaced.-->

<div style="text-align:center">

<h1>

Welcome to {{title}}!

</h1>

</div>

<!-- •  The routerLink directive tells

the router where to navigate when the user clicks the link.

-->

<ul>

<li> <h2><a routerLink="home">Home</a></h2> </li>

<li> <h2><a routerLink="employees">Employees</a></h2> </li>

<br/>

<!-- •  The router-outlet directive is used to specify

the location where we want the routed component's

view template to be displayed

-->

<router-outlet></router-outlet>

</ul>

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

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

title = 'YASH';

}

Index.html

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>ValidationDemo</title>

<base href="/">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="icon" type="image/x-icon" href="favicon.ico">

</head>

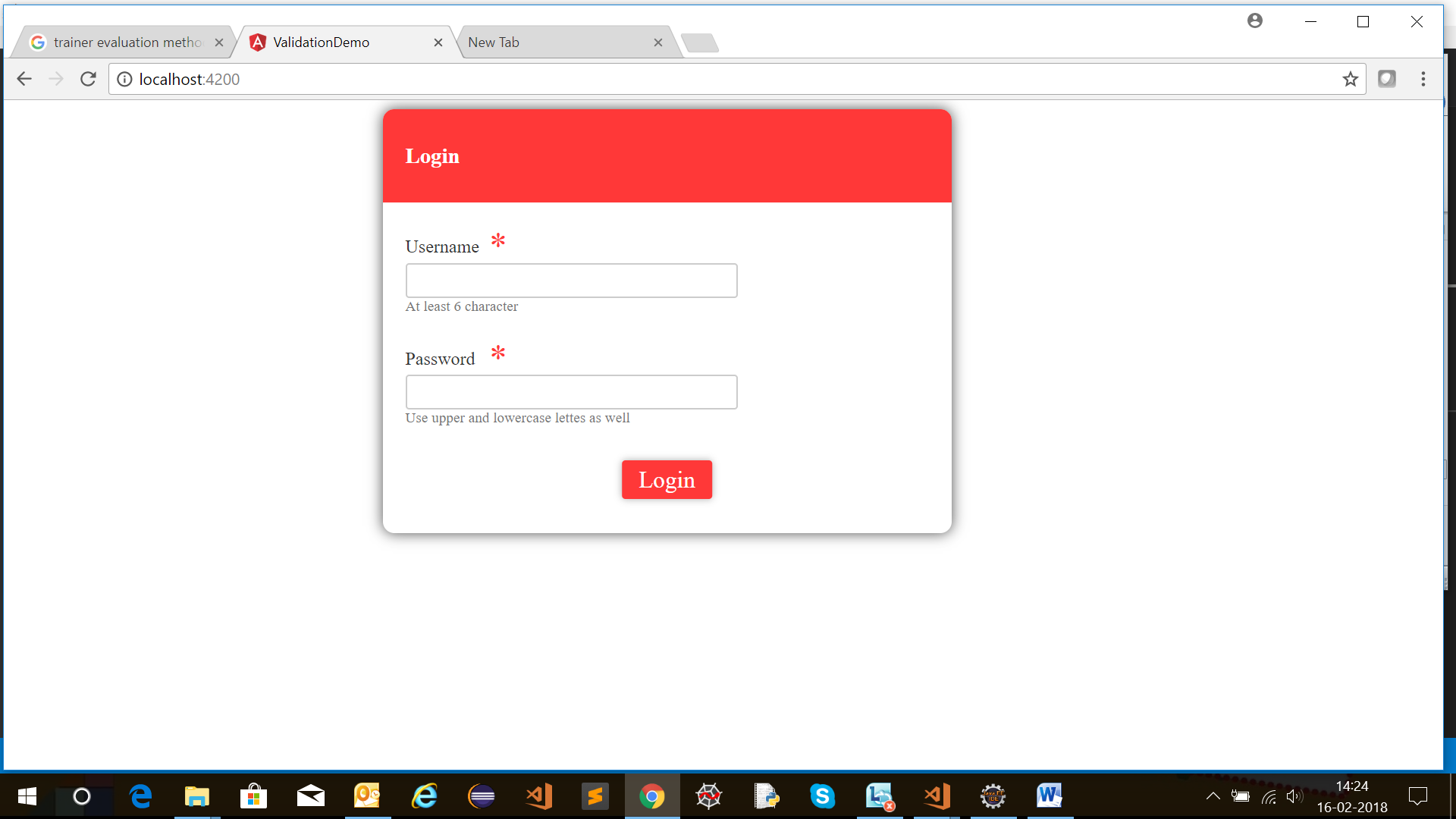
<body>

<app-root></app-root>

</body>

</html>

Validation Demo



AppModule.ts

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

import { BrowserModule } from '@angular/platform-browser';

import { ReactiveFormsModule } from '@angular/forms';

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

@NgModule({

imports: [BrowserModule, ReactiveFormsModule,], // we add FormsModule here

declarations: [AppComponent],

bootstrap: [AppComponent]

})

export class AppModule {}

AppComponent.ts

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

import { FormGroup, FormControl, Validators} from '@angular/forms';

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

formdata;

ngOnInit() {

this.formdata = new FormGroup({

uname: new FormControl("", Validators.compose([

Validators.required,

Validators.minLength(6)

])),

passwd: new FormControl("", this.passwordvalidation)

});

}

passwordvalidation(formcontrol) {

if (formcontrol.value.length < 5) {

return {"passwd" : true};

}

}

onClickSubmit(data) {

console.log(data.uname);

if (data.uname=="systemadmin" && data.passwd=="admin123") {

alert("Login Successful");

// this.router.navigate(['app-mainpage'])

} else {

alert("Invalid Login");

return false;

}

}

}

App.component.html

<!--The content below is only a placeholder and can be replaced.-->

<div class="form\_container">

<form [formGroup]="formdata" (ngSubmit) = "onClickSubmit(formdata.value)" >

<header>Login</header>

<label>Username <span>\*</span></label>

<input type="text" name="uname" formControlName="uname"/>

<div class="help">At least 6 character</div>

<label>Password <span>\*</span></label>

<input type="password" class="fortextbox" name="passwd" formControlName="passwd"/>

<div class="help">Use upper and lowercase lettes as well</div>

<button [disabled]="!formdata.valid" value="Login">Login</button>

</form>

</div>