

ANGULAR 4 - EXAMPLES

https://www.tutorialspoint.com/angular4/angular4_examples.htm

Copyright © tutorialspoint.com

Advertisements

In this chapter, we will discuss a few examples related to Angular 4.

To begin with, we have created an example which shows a login form with input as **username** and **password**. Upon entering the correct values, it will enter inside and show another form wherein, you can enter the customer details. In addition, we have created four components - header, footer, userlogin and mainpage.

The components are created using the following command –

ng g component header

```
C:\ngexamples\aexamples>ng g component header
installing component
  create src\app\header\header.component.css
  create src\app\header\header.component.html
  create src\app\header\header.component.spec.ts
  create src\app\header\header.component.ts
  update src\app\app.module.ts
```

ng g component footer

```
C:\ngexamples\aexamples>ng g component footer
installing component
  create src\app\footer\footer.component.css
  create src\app\footer\footer.component.html
  create src\app\footer\footer.component.spec.ts
  create src\app\footer\footer.component.ts
  update src\app\app.module.ts
```

ng g component userlogin

```
C:\ngexamples\aexamples>ng g component userlogin
installing component
  create src\app\userlogin\userlogin.component.css
  create src\app\userlogin\userlogin.component.html
  create src\app\userlogin\userlogin.component.spec.ts
  create src\app\userlogin\userlogin.component.ts
  update src\app\app.module.ts
```

ng g component mainpage

```
C:\ngexamples\aexamples>ng g component mainpage
installing component
  create src\app\mainpage\mainpage.component.css
  create src\app\mainpage\mainpage.component.html
  create src\app\mainpage\mainpage.component.spec.ts
```

```
create src\app\mainpage\mainpage.component.ts
update src\app\app.module.ts
```

In the **app.module.ts**, the parent module has all the components added when created. The file looks as follows –

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { ReactiveFormsModule } from '@angular/forms';

import { RouterModule, Routes } from '@angular/router';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { MdTableModule } from '@angular/material';

import { HttpClientModule } from '@angular/http';
import { MdInputModule } from '@angular/material';
import { AppComponent } from './app.component';

import { HeaderComponent } from './header/header.component';
import { FooterComponent } from './footer/footer.component';
import { UserloginComponent } from './userlogin/userlogin.component';
import { MainpageComponent } from './mainpage/mainpage.component';

const appRoutes: Routes = [
  {
    path: '',
    component: UserloginComponent
  },
  {
    path: 'app-mainpage',
    component: MainpageComponent
  }
];

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    FooterComponent,
    UserloginComponent,
    MainpageComponent
  ],
  imports: [
    BrowserModule,
    ReactiveFormsModule,
    RouterModule.forRoot(appRoutes),
    BrowserAnimationsModule,
    HttpClientModule,
    MdTableModule,
    MdInputModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

The components created above are added –

```
import { HeaderComponent } from './header/header.component';
import { FooterComponent } from './footer/footer.component';
import { UserloginComponent } from './userlogin/userlogin.component';
import { MainpageComponent } from './mainpage/mainpage.component';
```

The components are added in the declarations too –

```
declarations: [
  AppComponent,
  HeaderComponent,
  FooterComponent,
  UserloginComponent,
  MainpageComponent
],
```

In the parent **app.component.html**, we have added the main structure of the file that will be seen by the user.

```
<div class="mainpage">
  <app-header></app-header>
  <router-outlet></router-outlet>
  <app-footer></app-footer>
</div>
```

We have created a div and added **<app-header></app-header>**, **<router-outlet></router-outlet>** and **<app-footer></app-footer>**.

The **<router-outlet></router-outlet>** is used for navigation between one page to another. Here, the pages are login-form and once it is successful it will redirect to the mainpage, i.e., the customer form.

To get the login-form first and later get the mainpage.component.html, the changes are done in **app.module.ts** as shown below –

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { ReactiveFormsModule } from '@angular/forms';

import { RouterModule, Routes } from '@angular/router';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { MdTableModule } from '@angular/material';

import { HttpModule } from '@angular/http';
import { MdInputModule } from '@angular/material';
import { AppComponent } from './app.component';

import { HeaderComponent } from './header/header.component';
import { FooterComponent } from './footer/footer.component';
import { UserloginComponent } from './userlogin/userlogin.component';
import { MainpageComponent } from './mainpage/mainpage.component';

const appRoutes: Routes = [
  {
    path: '',
    component: UserloginComponent
```

```

    },
    {
      path: 'app-mainpage',
      component: MainpageComponent
    }
  ];

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    FooterComponent,
    UserloginComponent,
    MainpageComponent
  ],

  imports: [
    BrowserModule,
    ReactiveFormsModule,
    RouterModule.forRoot(appRoutes),
    BrowserAnimationsModule,
    HttpClientModule,
    MdTableModule,
    MdInputModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

We have imported **RouterModule** and **Routes** from **@angular/router**. In imports, the RouterModules takes appRoutes as the param which is defined above as –

```

const appRoutes: Routes = [
  {
    path: '',
    component: UserloginComponent
  },
  {
    path: 'app-mainpage',
    component: MainpageComponent
  }
];

```

Routes take the array of components and by default the userloginComponent is called.

In **userlogin.component.ts**, we have imported the router and navigated to mainpage.component.html based on the condition as shown below –

```

import { Component, OnInit } from '@angular/core';
import { FormGroup, FormControl, Validators } from '@angular/forms';
import { Router } from '@angular/router';

@Component({
  selector: 'app-userlogin',
  templateUrl: './userlogin.component.html',

```

```

    styleUrls: ['./userlogin.component.css']
  })

export class UserloginComponent implements OnInit {
  formdata;
  constructor(private router: Router) { }
  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;
      }
    }
  }
}

```

Following is the .ts file for **app.component.ts**. Only the default details are present in it.

```

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {title = 'app';}

```

Let us now display the details of each of the components files. To start with, we will first take the header component. For the new component, four files are created **header.component.ts**, **header.component.html**, **header.component.css**, and **header.component.spec.ts**.

header.component.ts

```

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

@Component({
  selector: 'app-header',

```

```
    templateUrl: './header.component.html',
    styleUrls: ['./header.component.css']
  })

export class HeaderComponent implements OnInit {
  constructor() { }
  ngOnInit() {}
}
```

header.component.html

```
<div>
  <hr />
</div>
```

We have not added any css. This makes the header.component.css file empty. Also, the **header.component.spec.ts** file is empty as the test cases are not considered here.

For the header, we will draw a horizontal line. A logo or any other detail can be added to make the header look more creative.

Let us now consider creating a footer component.

For the footer component, **footer.component.ts**, **footer.component.html**, **footer.component.spec.ts** and **footer.component.css** files are created.

footer.component.ts

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

@Component({
  selector: 'app-footer',
  templateUrl: './footer.component.html',
  styleUrls: ['./footer.component.css']
})

export class FooterComponent implements OnInit {
  constructor() { }
  ngOnInit() { }
}
```

footer.component.html

```
<hr/>
```

As we have not added any css, the **footer.component.css** file is empty. Also, the **footer.component.spec.ts** file is empty as the test cases are not considered here.

For the footer, we will just draw a horizontal line as shown in the .html file.

Let us now see how the userlogin component works. The following files for userlogin component created are **userlogin.component.css**, **userlogin.component.html**, **userlogin.component.ts**, and **userlogin.component.spec.ts**.

The details of the files are as follows –

userlogin.component.html

```
<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>
```

Here, we have created form with two input controls **Username** and **Password**. This is a model driven form approach and the details of the same are explained in Chapter 14 - Forms.

We consider the username and password mandatory, hence the validation for the same is added in the ts. Upon clicking the submit button, the control is passed to the **onClickSubmit**, which is defined in the ts file.

userlogin.component.ts

```
import { Component, OnInit } from '@angular/core';
import { FormGroup, FormControl, Validators } from '@angular/forms';
import { Router } from '@angular/router';

@Component({
  selector: 'app-userlogin',
  templateUrl: './userlogin.component.html',
  styleUrls: ['./userlogin.component.css']
})

export class UserloginComponent implements OnInit {
  formdata;
  constructor(private router: Router) { }
  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'])
    }
}
}

```

For the formcontrol and validation, the modules are imported as shown below

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

We need a router to navigate to a different component when the user and password are correct. For this, the router is imported as shown below –

```
import { Router} from '@angular/router';
```

In **ngOnInit**, the validation for the form is done. We need the username to be more than six characters and the field is mandatory. The same condition applies to password too.

Upon clicking submit, we can check if the username is **systemadmin** and the password is **admin123**. If yes, a dialog box appears that shows **Login Successful** and the router navigates to the app-mainpage, which is the selector of the mainpage component.

There is css added for the form in **userlogin.component.css** file –

```

.form_container{
  margin : 0 auto;
  width:600px;
}

form {
  background: white;
  width: 500px;
  box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.7);
  font-family: lato;
  position: relative;
  color: #333;
  border-radius: 10px;
}

form header {
  background: #FF3838;
  padding: 30px 20px;
  color: white;
  font-size: 1.2em;
  font-weight: 600;
  border-radius: 10px 10px 0 0;
}

form label {
  margin-left: 20px;
  display: inline-block;
  margin-top: 30px;
  margin-bottom: 5px;
}

```



```
    position: relative;
  }

  form label span {
    color: #FF3838;
    font-size: 2em;
    position: absolute;
    left: 2.3em;
    top: -10px;
  }

  form input {
    display: block;
    width: 50%;
    margin-left: 20px;
    padding: 5px 20px;
    font-size: 1em;
    border-radius: 3px;
    outline: none;
    border: 1px solid #ccc;
  }

  form .help {
    margin-left: 20px;
    font-size: 0.8em;
    color: #777;
  }

  form button {
    position: relative;
    margin-top: 30px;
    margin-bottom: 30px;
    left: 50%;
    transform: translate(-50%, 0);
    font-family: inherit;
    color: white;
    background: #FF3838;
    outline: none;
    border: none;
    padding: 5px 15px;
    font-size: 1.3em;
    font-weight: 400;
    border-radius: 3px;
    box-shadow: 0px 0px 10px rgba(51, 51, 51, 0.4);
    cursor: pointer;
    transition: all 0.15s ease-in-out;
  }

  form button:hover {
    background: #ff5252;
  }
}
```

The **userlogin.component.spec.ts** file is empty as there no test cases right now.

Let us now discuss how the mainpage component works. The files created for mainpage component are **mainpage.component.ts**, **mainpage.component.html**, **mainpage.component.css**, and **mainpage.component.spec.ts**.

mainpage.component.ts

```

import { Component, OnInit, ViewChild } from '@angular/core';
import { FormGroup, FormControl, Validators } from '@angular/forms';

import { Http, Response, Headers, RequestOptions } from "@angular/http";
import 'rxjs/add/operator/map';

@Component({
  selector: 'app-mainpage',
  templateUrl: './mainpage.component.html',
  styleUrls: ['./mainpage.component.css']
})

export class MainpageComponent implements OnInit {
  formdata;
  cutomerdata;
  constructor(private http: Http) { }
  stateCtrl: FormControl;
  ngOnInit() {
    this.formdata = new FormGroup({
      fname: new FormControl("", Validators.compose([
        Validators.required,
        Validators.minLength(3)
      ])),
      lname: new FormControl("", Validators.compose([
        Validators.required,
        Validators.minLength(3)
      ])),
      address: new FormControl(""),
      phoneno: new FormControl("")
    });
  }
  onClickSubmit(data) {
    document.getElementById("custtable").style.display="";
    this.cutomerdata = [];
    for (var prop in data) {
      this.cutomerdata.push(data[prop]);
    }
    console.log(this.cutomerdata);
  }
}

```

We have created a customer form with firstname, lastname , address and phone number. The validation of the same is done with the **ngOnInit** function. Upon clicking submit, the control comes to the function **onClickSubmit**. Here, the table which is used to display the entered details is made visible.

The customerdata is converted from json to array so that we can use the same in ngFor on the table, which is done in the .html file as shown below.

mainpage.component.html

```

<div class="form_container">
  <form [formGroup]="formdata" (ngSubmit) = "onClickSubmit(formdata.value)" >
    <header>Customer Details</header>
    <label>FirstName <span>*</span></label>
    <input type="text" name="fname" formControlName="fname"/>
    <label>LastName <span>*</span></label>

```

```

<input type="text" name="lname" formControlName="lname"/>
<label>Address <span></span></label>
<input type="text" name="address" formControlName="address"/>
<label>Phone No <span></span></label>
<input type="text" name="phoneno" formControlName="phoneno"/>
<button [disabled]="!formdata.valid" value="Submit">Submit</button>
</form>
</div>
<br/>

<div id="custtable" style="display:none;margin:0 auto;">
  <table>
    <tr>
      <td>FirstName</td>
      <td>LastName</td>
      <td>Address</td>
      <td>Phone No</td>
    </tr>
    <tr>
      <td *ngFor="let data of cutomerdata">
        <h5>{{data}}</h5>
      </td>
    </tr>
  </table>
</div>

```

Here, the first div has the customer details and the second div has the table, which will show the entered details. The display of the userlogin and the customer details is as shown below. This is the page with login form and header and footer.

Login

Username *

At least 6 character

Password *

Use upper and lowercase lettes as well

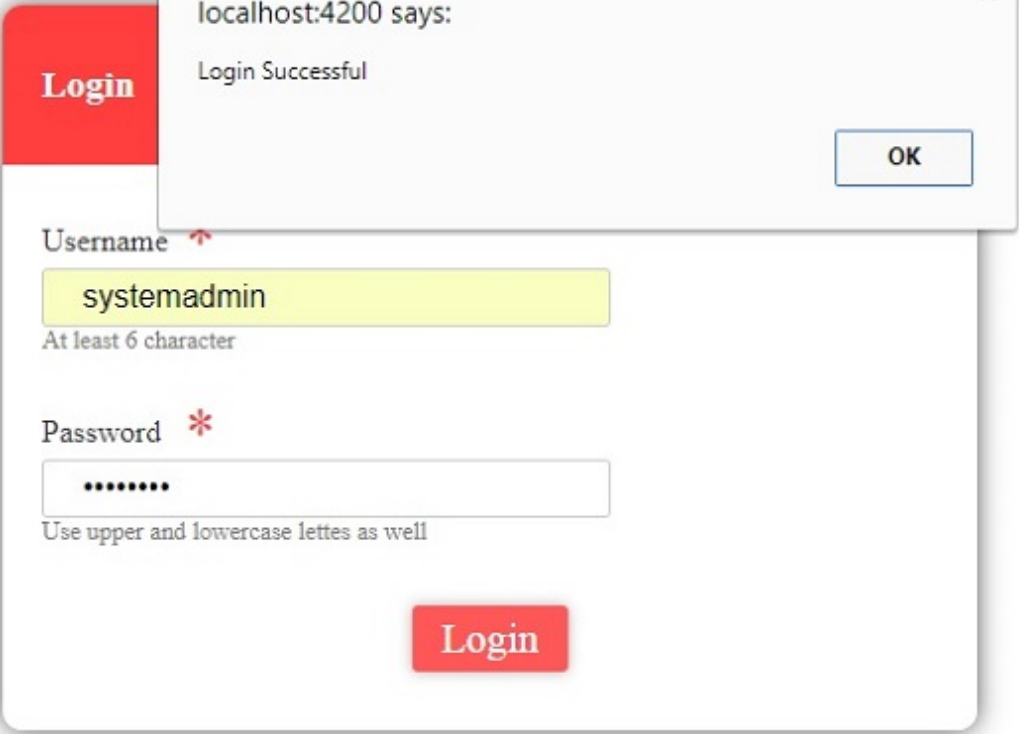
Login

Once you enter the details, the display is as shown below



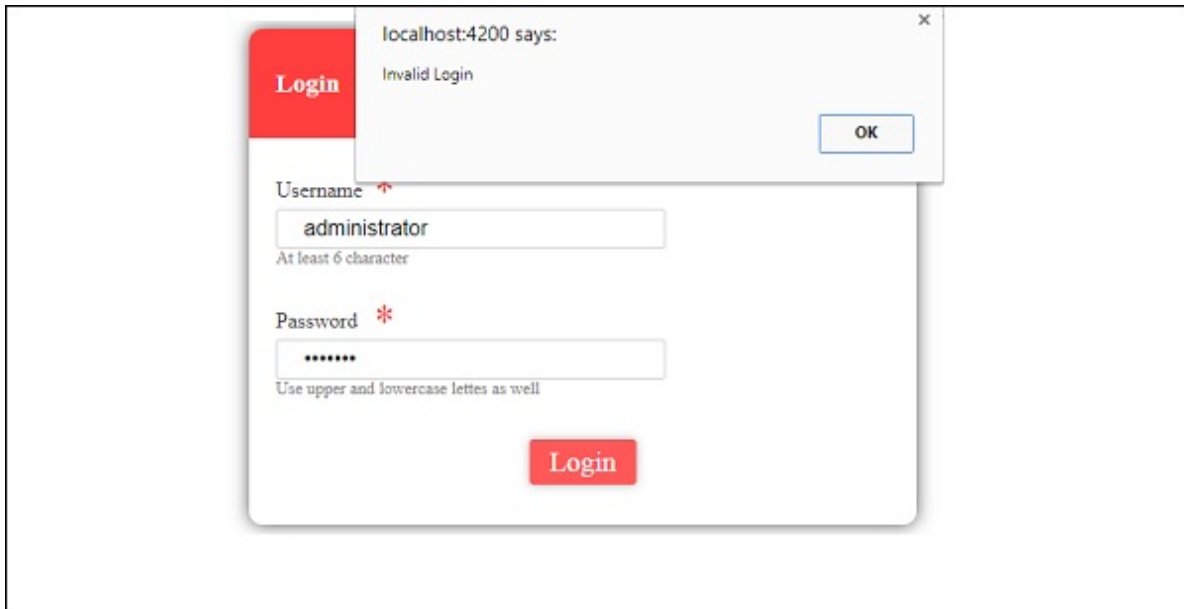
The screenshot shows a login form with a red header bar containing the word "Login". Below the header, there are two input fields. The first is labeled "Username" with a red asterisk icon. The input field contains the text "systemadmin" and has a yellow highlight. Below the input field, there is a hint text "At least 6 character". The second input field is labeled "Password" with a red asterisk icon. The input field contains seven dots. Below the input field, there is a hint text "Use upper and lowercase lettes as well". At the bottom of the form, there is a red button labeled "Login".

Upon clicking submit, a dialog box appears which shows Login Successful.



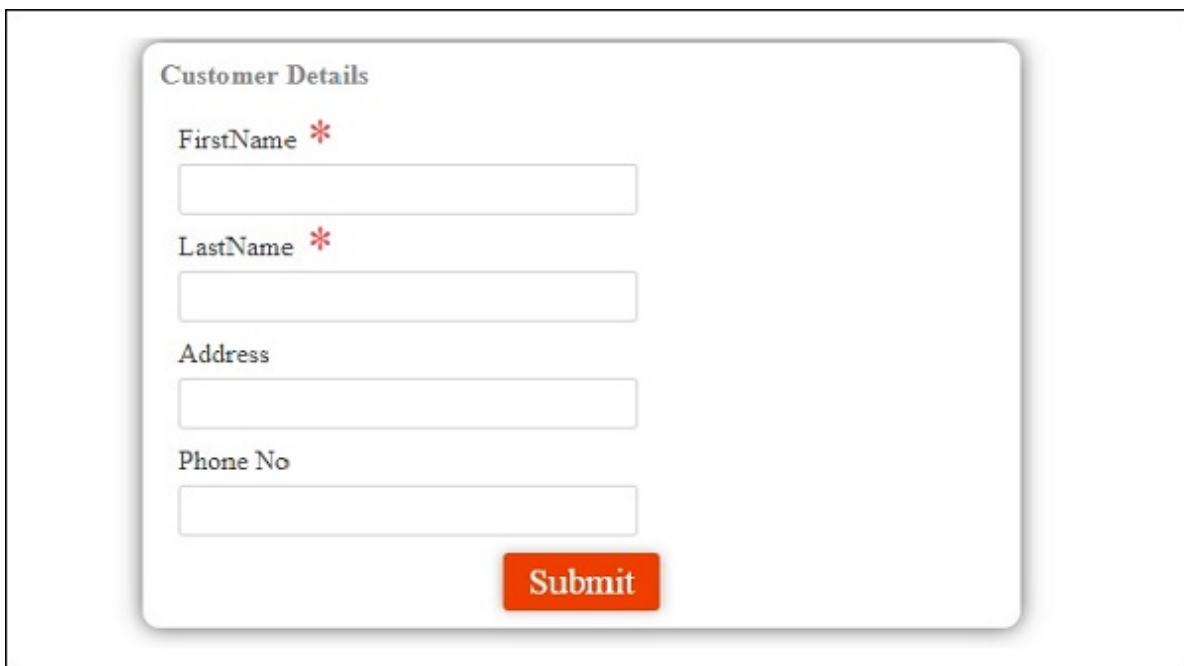
The screenshot shows the same login form as before, but with a dialog box overlaid on top. The dialog box has a title bar that says "localhost:4200 says:" and a close button (X). The main content of the dialog box says "Login Successful". There is an "OK" button at the bottom right of the dialog box. The login form is partially visible behind the dialog box.

If the details are invalid, a dialog box appears which shows Invalid Login as shown below –



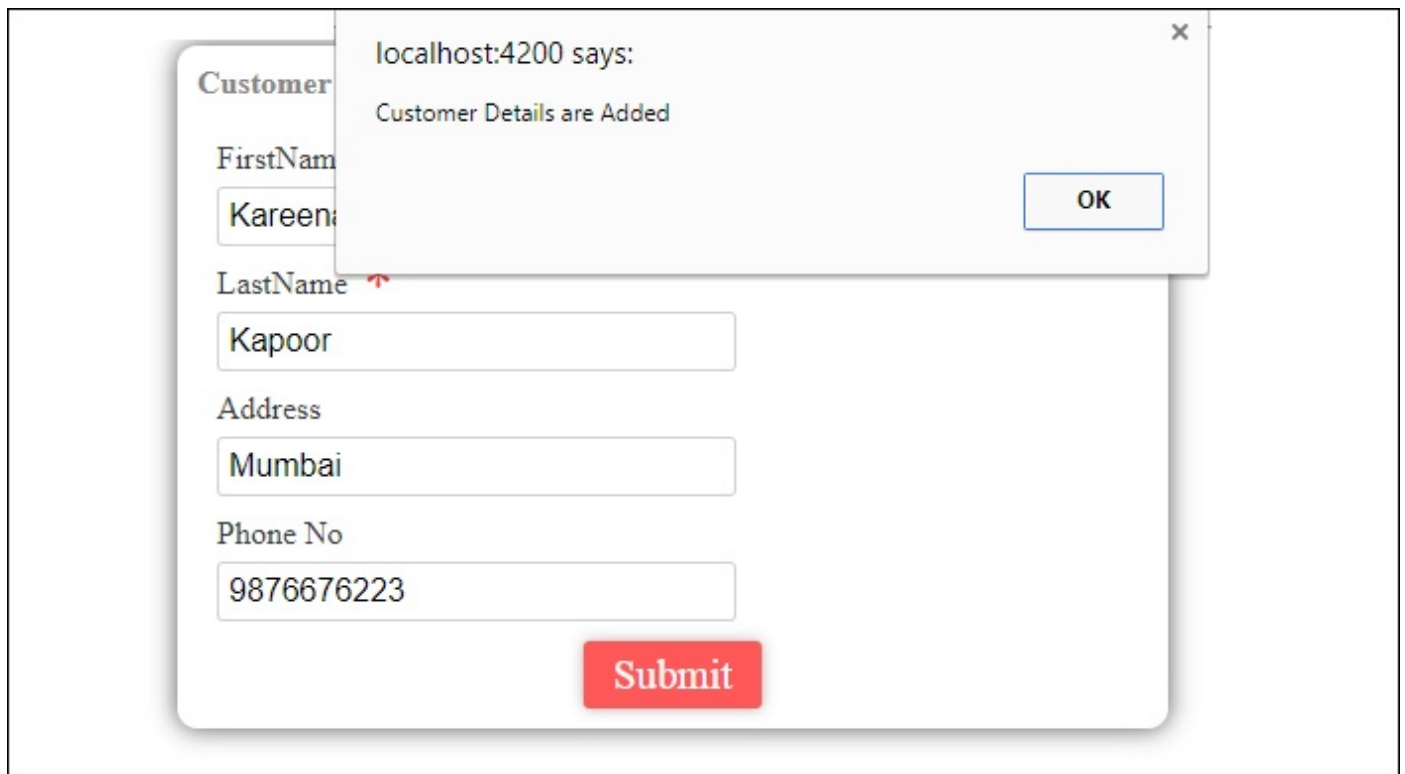
The screenshot shows a login form with a red 'Login' button at the top left. A dialog box titled 'localhost:4200 says:' is open, displaying the message 'Invalid Login' and an 'OK' button. The login form contains two input fields: 'Username' with the value 'administrator' and a red asterisk icon, and 'Password' with masked characters '*****' and a red asterisk icon. Below the password field is a hint: 'Use upper and lowercase letters as well'. A red 'Login' button is at the bottom right of the form.

If the login is successful, it will proceed to the next form of Customer Details as shown below –



The screenshot shows a 'Customer Details' form with four input fields: 'FirstName' with a red asterisk, 'LastName' with a red asterisk, 'Address', and 'Phone No'. A red 'Submit' button is at the bottom right of the form.

Once the details are entered and submitted, a dialog box appears which shows the Customer Details are added as shown in the screenshot below –



The screenshot shows a web application interface for adding customer details. The main form is titled "Customer" and contains four input fields: "First Name" (with the value "Kareena"), "Last Name" (with the value "Kapoor"), "Address" (with the value "Mumbai"), and "Phone No" (with the value "9876676223"). A red "Submit" button is located at the bottom right of the form. A modal dialog box is open on top of the form, displaying the message "localhost:4200 says: Customer Details are Added" and an "OK" button.

When we click OK in the above screenshot, the details will appear as shown in the screenshot below –

Customer Details

FirstName *

Kareena

LastName *

Kapoor

Address

Mumbai

Phone No

9876545454

Submit

FirstName	LastName	Address	Phone No
Kareena	Kapoor	Mumbai	9876545454