

پروژه Registration و Login

در ساخت این پروژه‌ها از موارد زیر استفاده شده است :

- نرم افزار Eclipse
- زبان برنامه نویسی Java
- JSP
- Servlet
- JDBC
- MySQL

Overview

ابتدا sign-up page میسازم که user باید username و password را وارد کند بعد از آن اگر بر روی دکمه log in کلیک کند ، validation اطلاعات وارد شده با دیتابیس چک میشود و اگر اطلاعات وارد شده صحیح باشند وارد Homepage میشود.

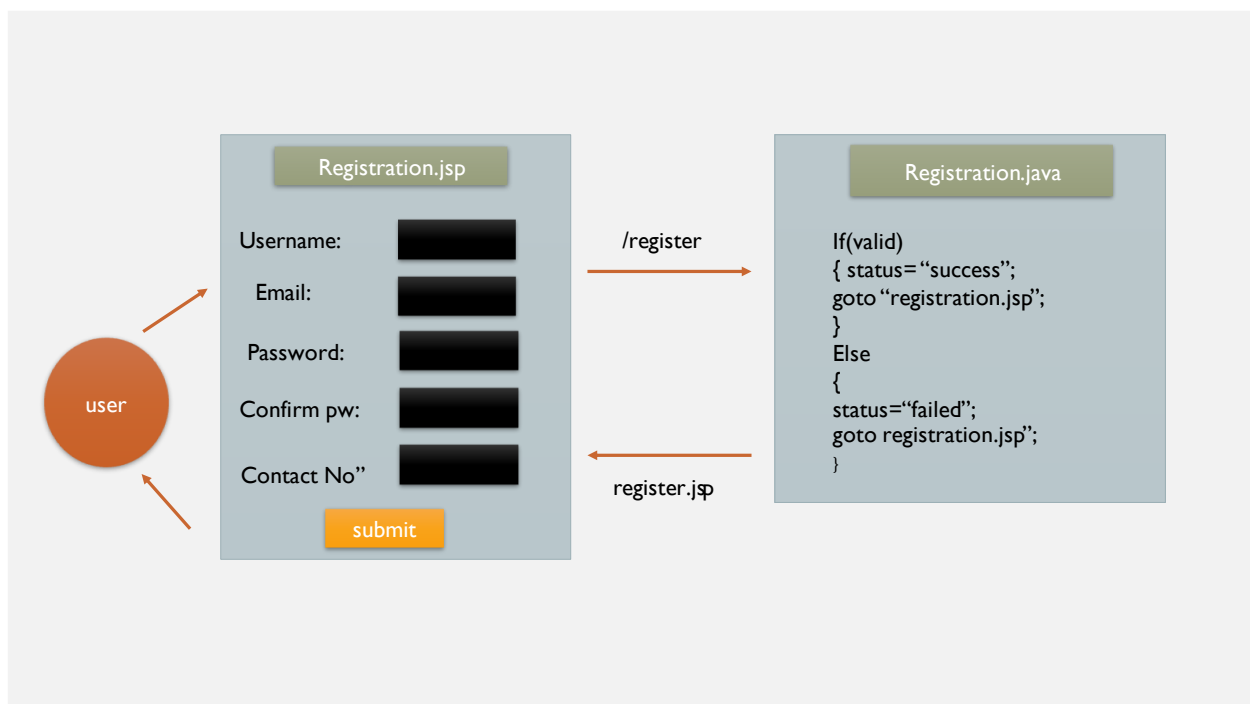
هنگامی که وارد Homepage میشود میتواند اسم User را در بالای صفحه سمت راست مشاهده کند.

اگر بر روی log out کلیک کنید دوباره به صفحه sign-up برمیگردد.

اگر user اکانت ندارد، بر روی create an account کلیک کند و وارد صفحه sign-up شود و با پر کردن فرم میتواند اکانت جدید بسازد.

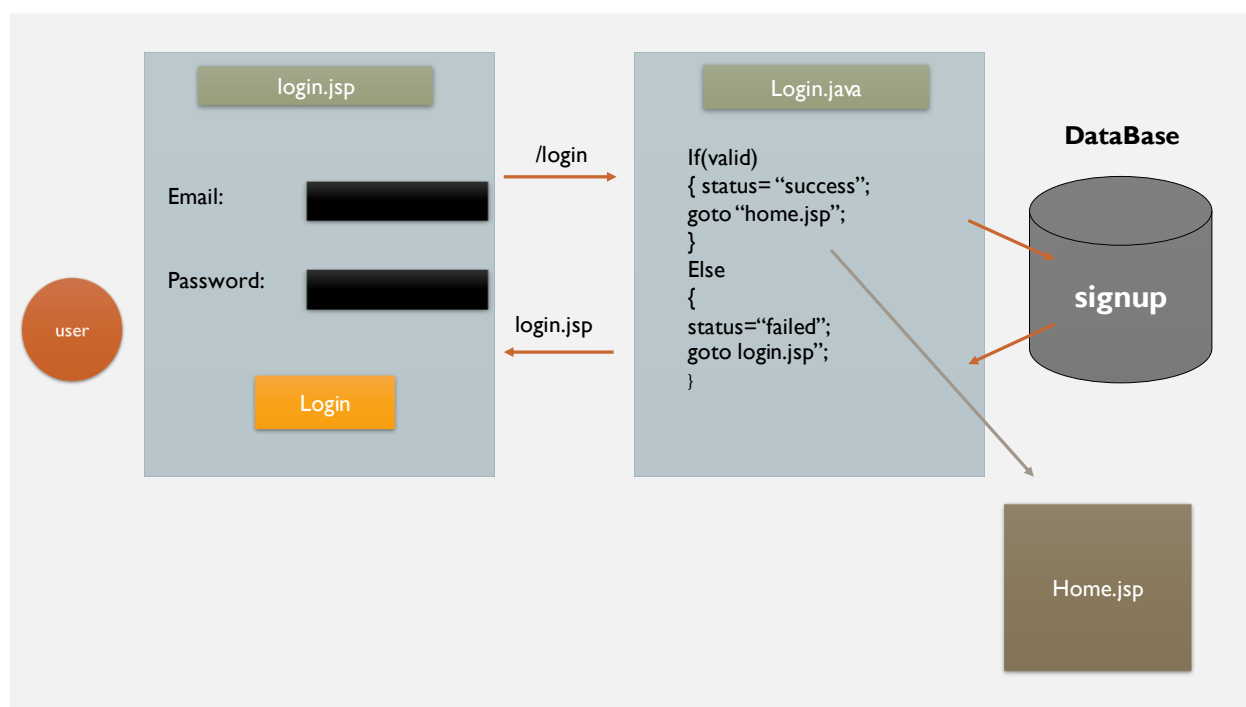
User Registration Flow

در دیتا بیس یک جدول داریم به اسم کاربر در این جدول فیلدهای نام کاربر، ایمیل، رمز عبور، شماره تلفن وجود دارد، کاربر باید این اطلاعات را وارد کند در صورتی که `status=success` بود رجیستر نمایش داده میشود و در صورت `status = failed` پیام عدم موفقیت نمایش داده میشود.



User Login Flow

کاربر وارد login پیج میشود و اطلاعات را وارد میکند و سپس loginServletPage.java را بررسی میکند، اگر username و password را با دیتابیس چک میشود و اطلاعات درست باشد status=success وارد Homepage میشود اگر status = failed باشد وارد login page میشود و یک پیام عدم موفقیت نمایش داده میشود.



Steps

- ابتدا یک dynamic web project ایجاد میکنیم.
اسم پروژه را sign up میگذاریم و برای target run هم apache tomcat v9.0 قرار میدهیم.
بر روی next کلیک میکنم و حتما چک ایجاد فایل web.xml را میزنم.
پروژه ما ساخته شد.
به فایل web.xml میرویم و فایل های welcome موجود را پاک کرده و یک تگ ولکام فایل جدید میسازم و در آن فایل index.jsp را قرار می دهیم.
- جار های مورد نیاز را اضافه میکنیم.
باید MySQL را به فولدر lib اضافه کنیم.
عبارت MySQL Connector for jar 8.0 را سرچ میکنیم.
از وب سایت jar.download آن را دانلود میکنیم.
فایل را اکستراکت کرده و جار فایل را به فولدر lib اضافه میکنیم.
- ایجاد دیتابیس
ابتدا MySQL Databasel را دانلود میکنیم و آن را روی سیستم نصب میکنیم.
MySQL Workbench را باز میکنیم .
دیتابیس به اسم company ایجاد میکنیم .
سپس جدول users با فیلد های مشخص شده در تصویر را ایجاد میکنیم.

```

Query 1 x
Limit to 1000 rows
1 • create database company;
2 • use company;
3 • create table users(
4   id int primary key auto_increment,
5   uname varchar(50),
6   upwd varchar(50),
7   uemail varchar(50),
8   umobile varchar(20)
9 );
10

```

جدول ایجاد شده :

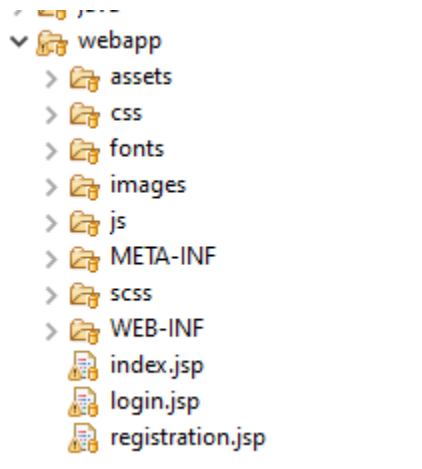
	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	uname	varchar(50)	YES		NULL	
	upwd	varchar(50)	YES		NULL	
	uemail	varchar(50)	YES		NULL	
	umobile	varchar(20)	YES		NULL	

4. ایجاد login page و register page

بخش فرانت اند پروژه قبل آماده شده و به صورت template آماده دانلود شده

فایل ها و فولدر های مربوط به این بخش را در فولدر webapp قرار میدهیم

3 فایل اصلی : index.jsp – login.jsp -registration.jsp



در فایل لاگین حتما name attribute چک شود چون از این طریق میتوانیم در سرولت value که user وارد میکند را fetch کنیم.

در فایل رجستر یک form داریم که method و action آن خالی است.

در این فرم یک فیلد confirm password دارد که فقط برای user هست و در جدول دیتابیس قرار نمیگردد.

در این فایل هم name attribute چک شود.

برای ایمیل، تایپ را email انتخاب میکنیم که مطمئن شویم user حتما ایمیل وارد میکند.

اگه الان پروژه را Run کنیم مستقیما وارد Homepage میشویم ولی ما میخواهیم ابتدا مطمئن شویم که کاربر login است، برای این کار بالای index.jsp کد زیر را مینویسیم

```
<%  
if(session.getAttribute("name")==null){  
    response.sendRedirect("login.jsp");  
}  
%>
```

بنابراین هر موقع کاربر login کند این attribute را اسم کاربر را return میکند و اگر username را return نکند یعنی null باشد به این معنی است که کاربر login نکرده است و ما user را به پیج login, redirect میکنیم.

• در فولدر src یک پکیج جدید ایجاد میکنیم

به اسم com.myapp.registration

همه کلاس های مربوط به رجستر در این پکیج قرار میگیرن ، در این پکیج یک سرولت ایجاد میکنیم به نام RegistrationServlet ، تنها متودی که نیاز داریم doPost هست، سپس به فایل registration.jsp رفته و متود فرم را post قرار میدهیم.

سپس اتریبوت اکشن را برابر با register قرار میدیم و در فایل سرولت :

```
@WebServlet("/register")
```

Value این 2 بخش باید یکسان باشد بنابراین وقتی کاربر بر روی دکمه رجیستر کلیک میکند، به دنبال سرولتی میگردد که WebServlet آن برابر با رجستر باشد.

سپس در متود doPost متود زیر را مینویسیم:

```
PrintWriter out = response.getWriter();  
out.print("Working");
```

در این صورت اگر به صفحه ایجاد اکانت برویم و فرم را پر کنیم و دکمه رجستر را بزنیم میتوانیم ورکینگ را مشاهده کنیم این به این معنی است که request ما به سرولت میرسد.

حال برای اینکه اطلاعاتی که user وارد میکند را دریافت کنیم:

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
    String uname = request.getParameter("name");  
    String uemail = request.getParameter("email");  
    String upwd = request.getParameter("pass");  
    String umobile = request.getParameter("contact");  
    RequestDispatcher dispatcher = null;  
    Connection con = null;  
  
    PrintWriter out = response.getWriter();  
    out.print(uname);  
    out.print(uemail);  
    out.print(upwd);  
    out.print(umobile);  
}
```

با استفاده از name attribute میتوانیم اطلاعاتی که کاربر وارد میکند را بگیریم

در این صورت اگر به صفحه ایجاد اکانت برویم و فرم را پر کنیم و دکمه رجستر را بزنیم میتوانیم تمام اطلاعاتی که وارد کردیم را مشاهده کنیم.

• در گام بعدی باید دیتابیس کد را بنویسیم

ابتدا trycatch را مینویسیم، سپس پکیج و کلاس مورد نظر را مینویسیم (این کلاس ها در MySQL Connector هستند).

سپس باید connection ایجاد کنیم بعد از آن insert query را ایجاد میکنیم.

بعد از این مراحل باید value های این 4 تا پارمتر رو set کنیم.

برای چک کردن موفق بودن یا نبودن request شرط مینویسیم.

اگر row count بیشتر از 4 باشد (تعداد record های ایجاد شده) یعنی اطلاعات وارد شده در غیر این صورت یعنی موفق نبوده و اطلاعات وارد دیتابیس نشده.

اگر اطلاعات وارد شد یک setter ایجاد میکنیم و redirect میکنیم به سمت رجستر پیج، برای اینکار ابتدا باید یک dispatcher ایجاد کنیم و آدرس رجستر پیج را قرار دهیم.

در قسمت exception , catch را پرینت میکنیم .

در قسمت finally به کانکشن خاتمه میدهیم و برای اینکه con قابل استفاده در finally باشد آن را خارج از try block قرار میدهیم.

```
try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    con = DriverManager.getConnection("jdbc:mysql://localhost:3306/company?useSSL=false", "root", "password");
    PreparedStatement pst = con.prepareStatement("insert into users(username,upwd,umail,umobile) values(?,?,?,?)");
    pst.setString(1,uname);
    pst.setString(2,upwd);
    pst.setString(3,umail);
    pst.setString(4,umobile);

    int rowCount = pst.executeUpdate();
    dispatcher = request.getRequestDispatcher("registration.jsp");

    if (rowCount > 0) {
        request.setAttribute("status", "success");
    } else {
        request.setAttribute("status", "failed");
    }
    dispatcher.forward(request, response);
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        con.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

اگر پروژه را run کنیم و به یه اکانت جدید بسازیم میتوانیم آن را در دیتابیس مشاهده کنیم.

در گام بعدی برای مشاهده پیام موفقیت آمیز بودن از کتابخانه sweetalert استفاده میکنیم .

به فایل registration.jsp رفته و در پایین فایل script اضافه میکنیم.

```
</div>
<!-- JS -->
<script src="vendor/jquery/jquery.min.js"></script>
<script src="js/main.js"></script>
<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
<link rel="stylesheet" href="alert/dist/sweetalert.css">
```


سپس در بالای فایل یک hidden فیلد ایجاد میکنیم:

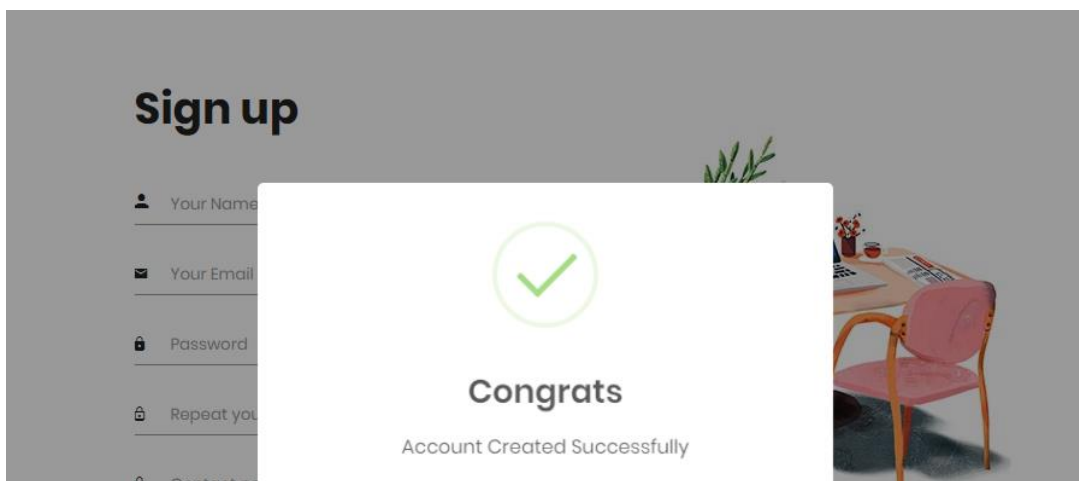
```
<input type="hidden" id="status" value="<%= request.getAttribute("status") %>">
```

ما اینجا status value را از registration.jsp سرولت میگیریم ، status میتواند success یا failed باشد اگر status=success باشد اسکریپت زیر اجرا میشود.

```
<script type="text/javascript">
    var status = document.getElementById("status").value;
    if (status == "success") {
        swal("Congrats","Account Created Successfully","success");
    }
</script>
```

Swal برای کتابخانه sweetalert و آنچه ما میخواهیم را نمایش میدهد.

اگر پروژه را Run کنیم و به یه اکانت جدید بسازیم و فرایند موفق باشد یک پیام موفقیت مشاهده خواهیم کرد.



بخش register تکمیل است.

• ایجاد بخش login

به فایل login.jsp رفته و متود فرم را post قرار میدهیم و action را login قرار دهیم.

سپس یک servlet دیگر در پکیج com.myapp.registration کلاسی به اسم login میسازیم. تنها متودی که نیاز داریم doPost هست و /login webServlet باشد.

برای دریافت اطلاعات که کاربر وارد میکند:

```
String uemail = request.getParameter("username");  
String upwd = request.getParameter("password");  
HttpSession session= request.getSession();  
RequestDispatcher dispatcher = null;
```

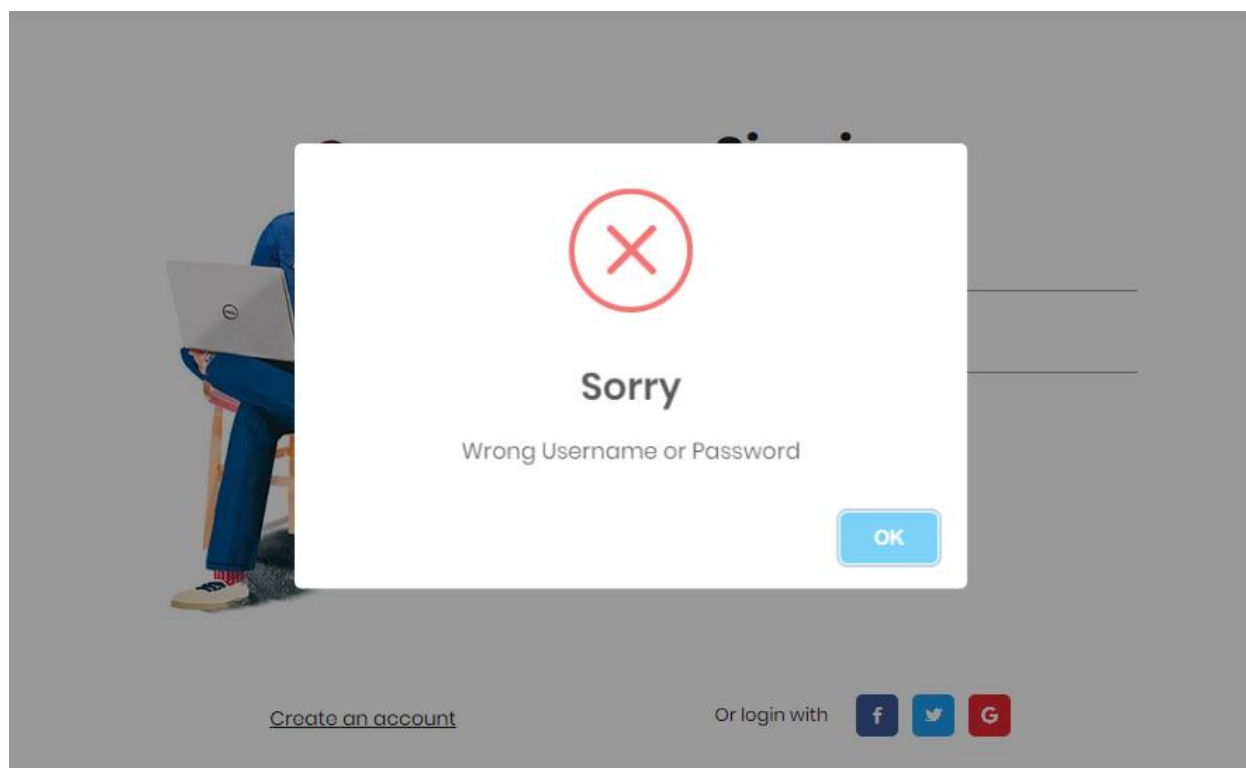
برای ایجاد کانکشن با دیتابیس:

اگر result set (rs) دیتا دارد به این معنی است که user در دیتا بیس وجود دارد و user را به redirect Homepage میکنیم.

اگر کاربر وجود نداشته باشد status = failed و redirect همیشه به login.

```
try {  
    Class.forName("com.mysql.jdbc.Driver");  
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/company?useSSL=false","root","password");  
    PreparedStatement pst = con.prepareStatement("select * from users where uemail = ? and upwd = ?");  
    pst.setString(1, uemail);  
    pst.setString(2, upwd);  
  
    ResultSet rs = pst.executeQuery();  
    if (rs.next()) {  
        session.setAttribute("name", rs.getString("uname"));  
        dispatcher = request.getRequestDispatcher("index.jsp");  
    }  
    else {  
        request.setAttribute("status", "failed");  
        dispatcher = request.getRequestDispatcher("login.jsp");  
    }  
    dispatcher.forward(request, response);  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

اگر username یا password اشتباه باشد می‌خواهیم یک پیام برای user نمایش داده شود. همان روندی که پیام موفقیت در رجسترانجام دادیم اینجا در فایل login.jsp تکرار میکنیم فقط متن پیام و آیکون آن متفاوت است.



• ایجاد logout

به فایل ایندکس جی اس پی میریم و کدهای زیر را مینویسیم

```
class="nav-link py-3 px-0 px-lg-3 rounded" href="Logout">Logout</a></li>
<li class="nav-item mx-0 mx-lg-1 bg-danger"><a
class="nav-link py-3 px-0 px-lg-3 rounded" href="Logout"><%= session.getAttribute("name") %></a></li>
```

با اینکار username را return میکند.

در گام بعدی یک سرولت جدید می‌سازیم به نام logout، تنها methodه که نیاز داریم doGet هست و وب سرولت هم باید logout باشد.

در doGet، باید session را دریافت کنیم و ری دایرکت کنیم user را به login page.

```
*/
@WebServlet("/logout")
public class Logout extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        HttpSession session = request.getSession();
        session.invalidate();
        response.sendRedirect("login.jsp");
    }
}
```

پروژه تکمیل است.