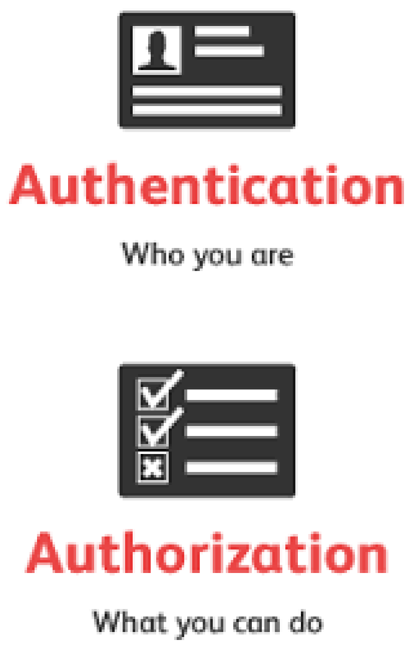
B.Sc. In Software Development. Year 4.
Enterprise Application Development. Semester I.
Authentication and Authorisation With Shiro

LIMERICK INSTITUTE
OF TECHNOLOGY
SCHOOL OF SCIENCE,
ENGINEERING & I.T.
Department of Information Technology

Introduction

- **Authentication**: the process of ascertaining that somebody really is who they claim to be.

- **Authorisation:** rules that determine what authenticated users are allowed to do.



Introduction

- You can restrict access to certain parts of a Web Application by writing custom Servlets and JSP’s to work directly with HTTP requests and responses.
    - Time-consuming and error-prone.

- Servlet containers provide built-in ways to restrict access to certain parts of a web app.
    - Container managed security/authentication.

- Shrio is a security framework that performs authentication, authorization, cryptography, and session management

Key Terminology

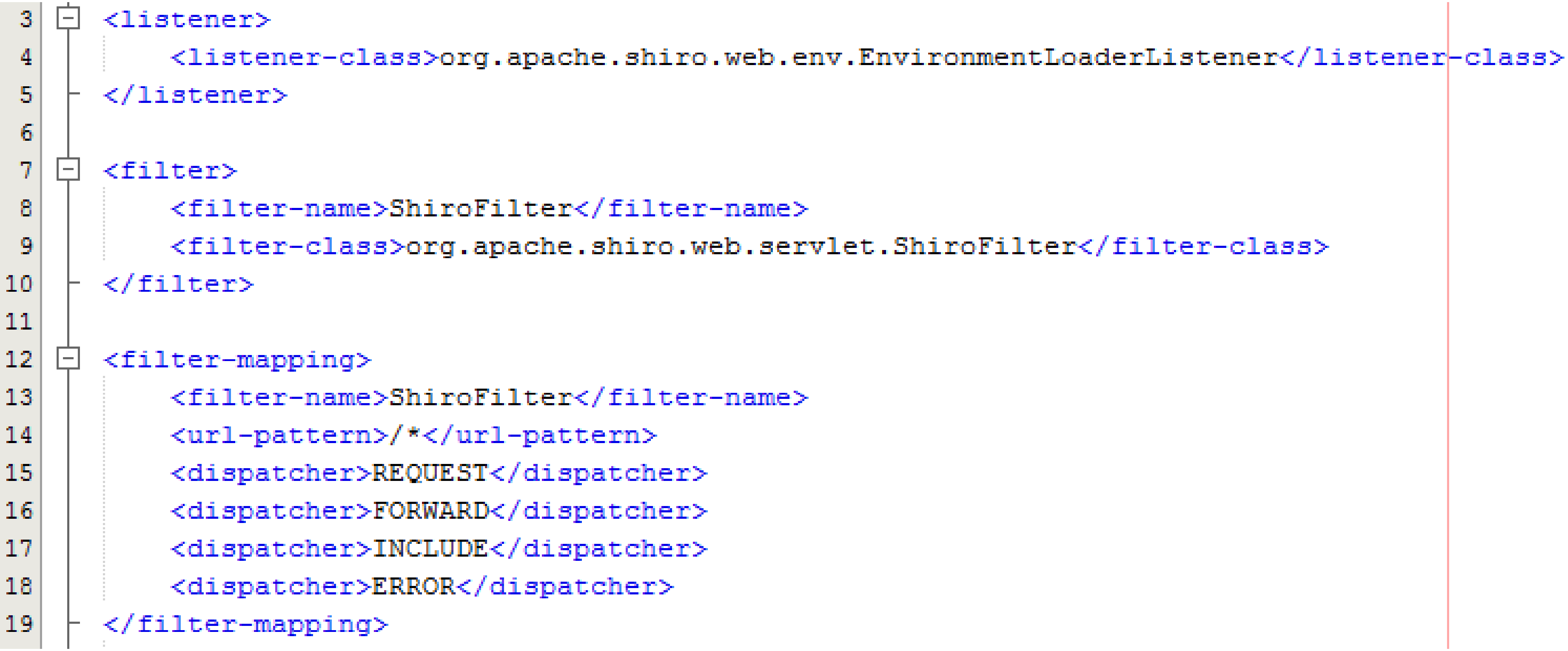
- **Subject:** It can be a human being, a third-party process, a server etc.

- **Principal:** A subjects identifying attributes. First name, last name, knumber, username etc.

- **Credentials** - secret data that are used to verify identities. Passwords, Biometric data etc.

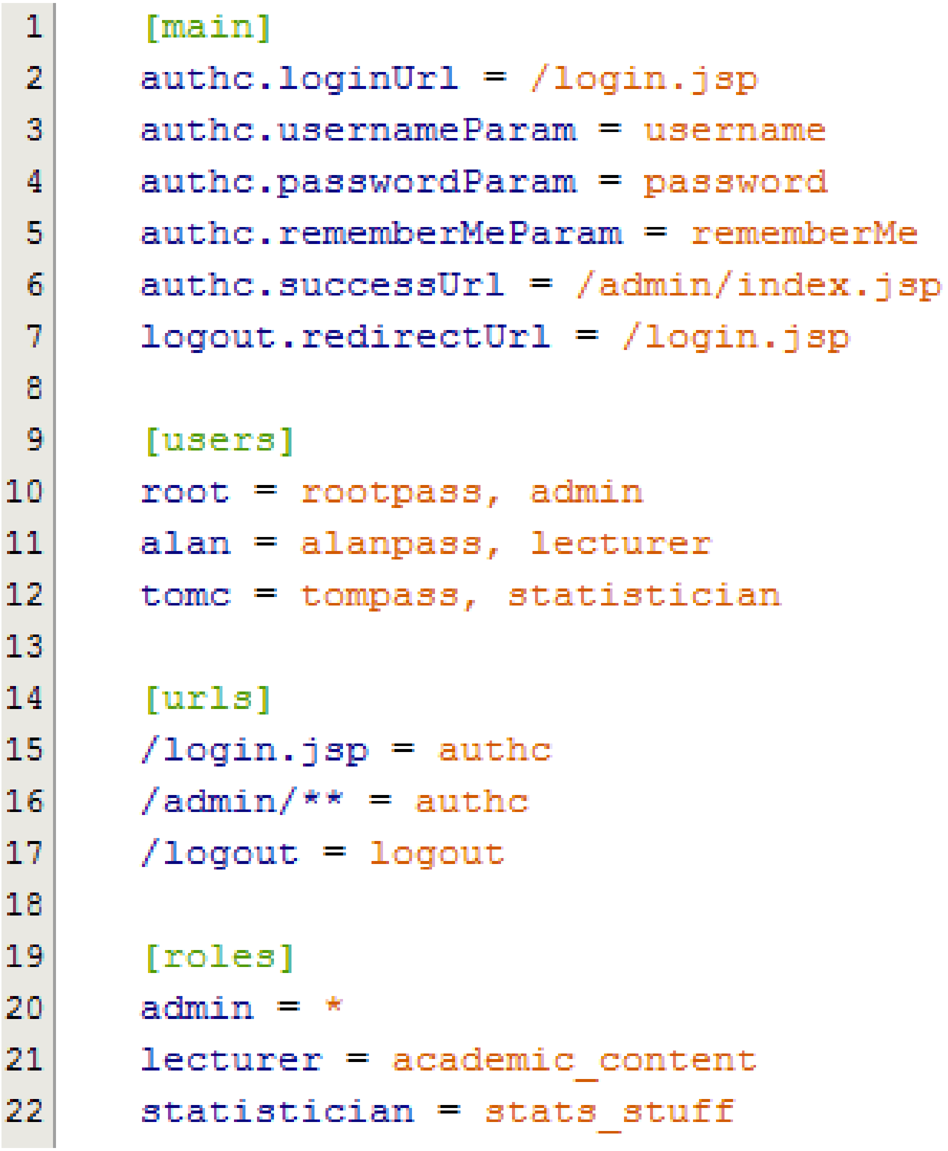
- **Realm:** A mechanism used for protecting Web application resources.

Configuring Shiro - web.xml



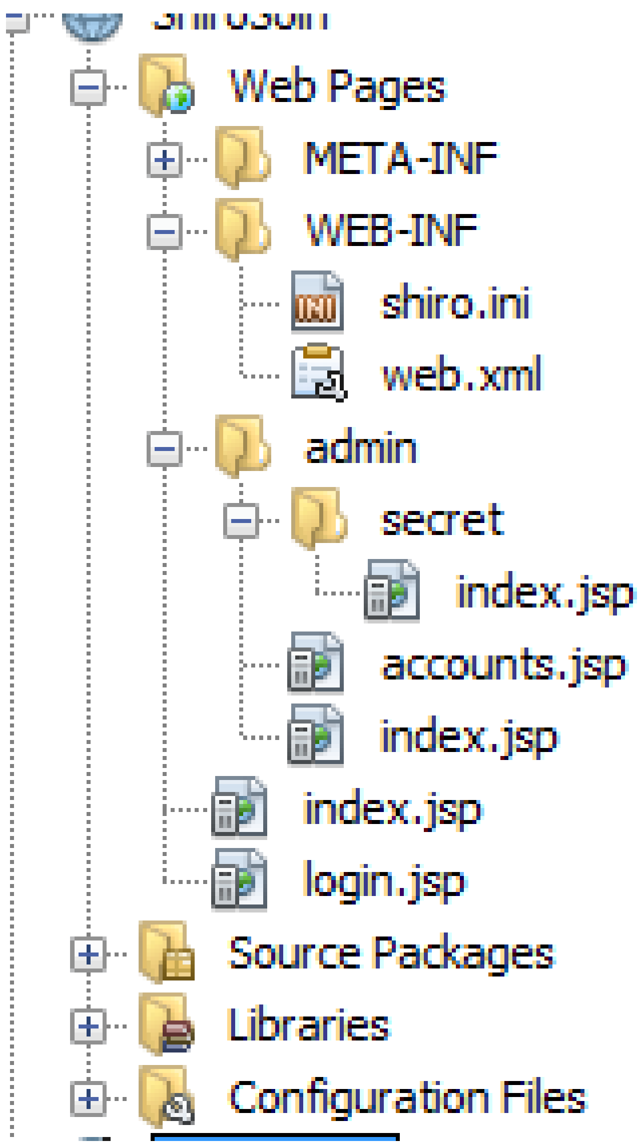
```xml
<listener>
    <listener-class>org.apache.shiro.web.env.EnvironmentLoaderListener</listener-class>
</listener>

<filter>
    <filter-name>ShiroFilter</filter-name>
    <filter-class>org.apache.shiro.web.servlet.ShiroFilter</filter-class>
</filter>

<filter-mapping>
    <filter-name>ShiroFilter</filter-name>
    <url-pattern>/*</url-pattern>
    <dispatcher>REQUEST</dispatcher>
    <dispatcher>FORWARD</dispatcher>
    <dispatcher>INCLUDE</dispatcher>
    <dispatcher>ERROR</dispatcher>
</filter-mapping>
```

Configuring Shiro - shiro.ini



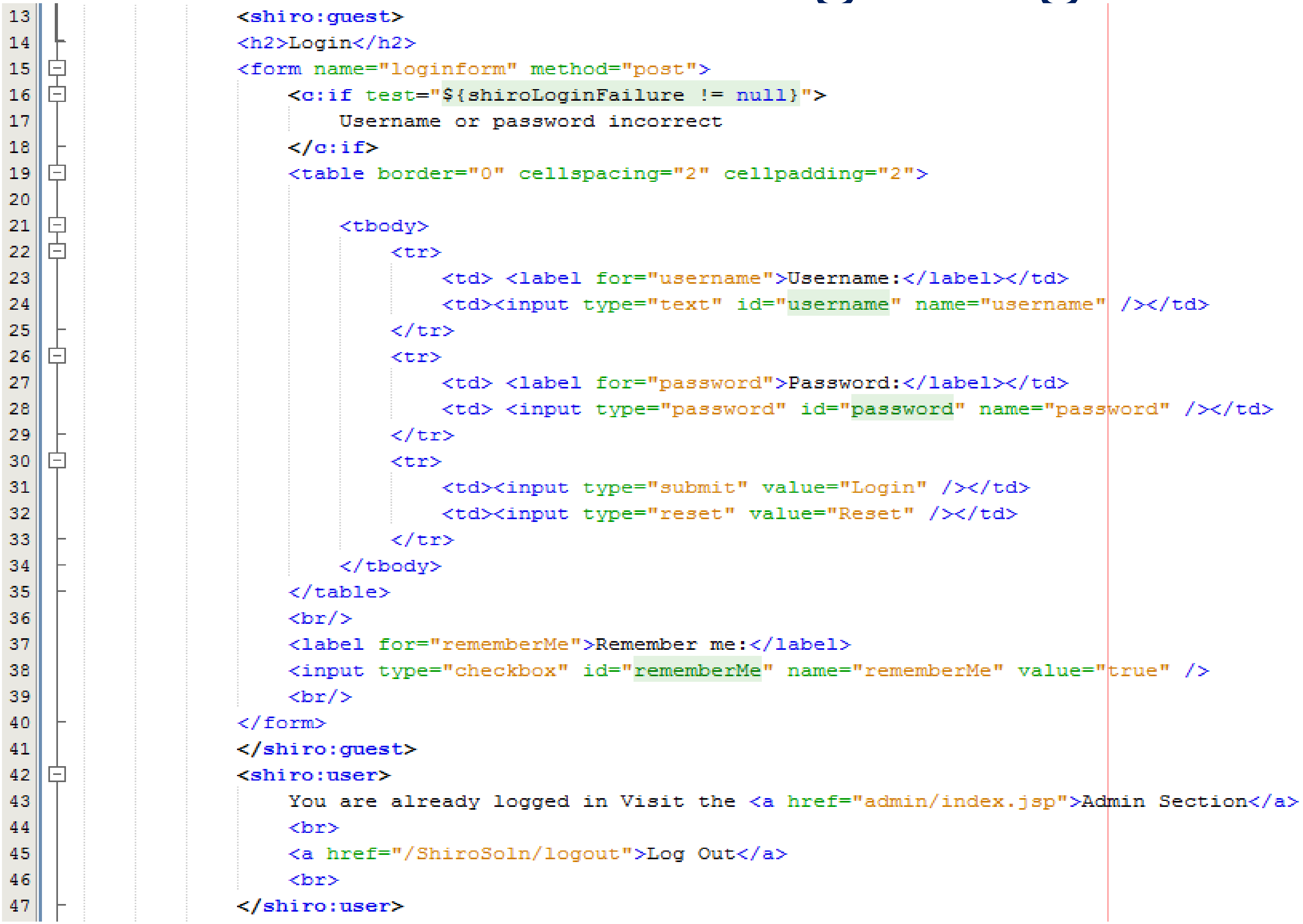
```
[main]
authc.loginUrl = /login.jsp
authc.usernameParam = username
authc.passwordParam = password
authc.rememberMeParam = rememberMe
authc.successUrl = /admin/index.jsp
logout.redirectUrl = /login.jsp

[users]
root = rootpass, admin
alan = alanpass, lecturer
tomc = tompass, statistician

[urls]
/login.jsp = authc
/admin/** = authc
/logout = logout

[roles]
admin = *
lecturer = academic_content
statistician = stats_stuff
```

Project Anatomy



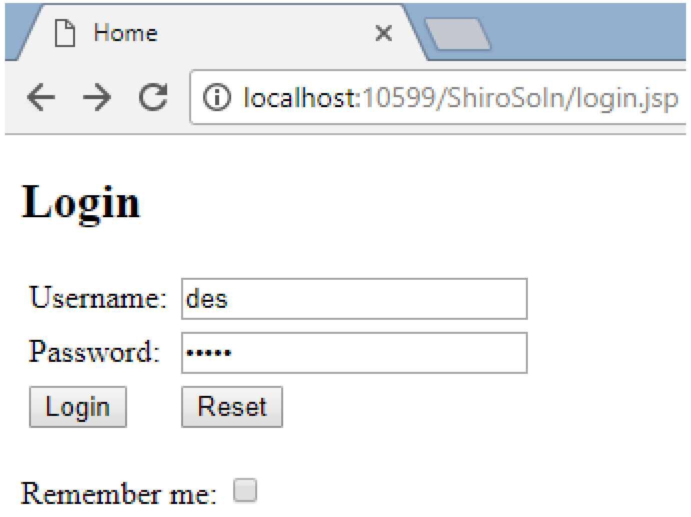
```
Web Pages
    META-INF
    WEB-INF
        shiro.ini
        web.xml
    admin
        secret
            index.jsp
        accounts.jsp
        index.jsp
    index.jsp
    login.jsp
Source Packages
Libraries
Configuration Files
```

Login Page



```
<shiro:guest>
<h2>Login</h2>
<form name="loginform" method="post">
    <c:if test="${shiroLoginFailure != null}">
        Username or password incorrect
    </c:if>
    <table border="0" cellspacing="2" cellpadding="2">

        <tbody>
            <tr>
                <td> <label for="username">Username:</label></td>
                <td><input type="text" id="username" name="username" /></td>
            </tr>
            <tr>
                <td> <label for="password">Password:</label></td>
                <td> <input type="password" id="password" name="password" /></td>
            </tr>
            <tr>
                <td><input type="submit" value="Login" /></td>
                <td><input type="reset" value="Reset" /></td>
            </tr>
        </tbody>
    </table>
    <br/>
    <label for="rememberMe">Remember me:</label>
    <input type="checkbox" id="rememberMe" name="rememberMe" value="true" />
    <br/>
</form>
</shiro:guest>
<shiro:user>
    You are already logged in Visit the <a href="admin/index.jsp">Admin Section</a>
    <br>
    <a href="/ShiroSoln/logout">Log Out</a>
    <br>
</shiro:user>
```

Application Flow



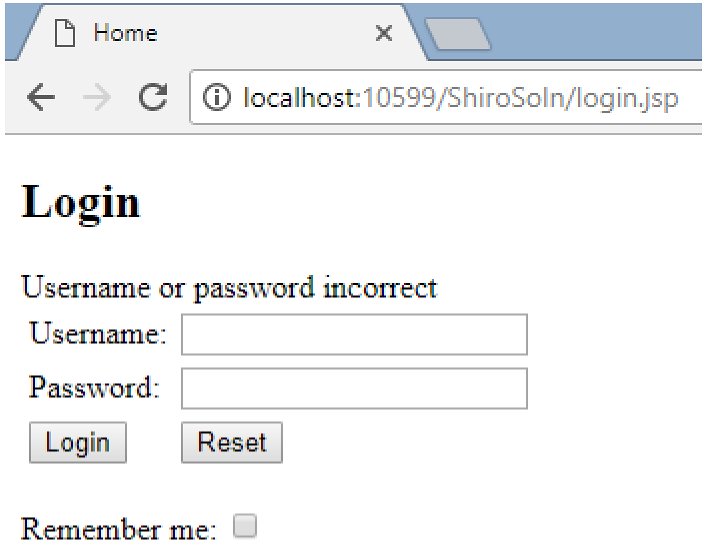
Login

Username: des

Password: •••••

Login Reset

Remember me: □



Login

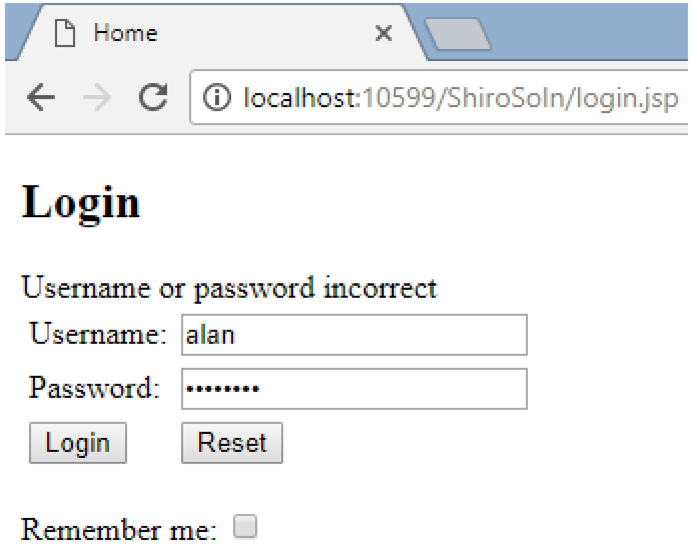
Username or password incorrect

Username:

Password:

Login Reset

Remember me: □



Login

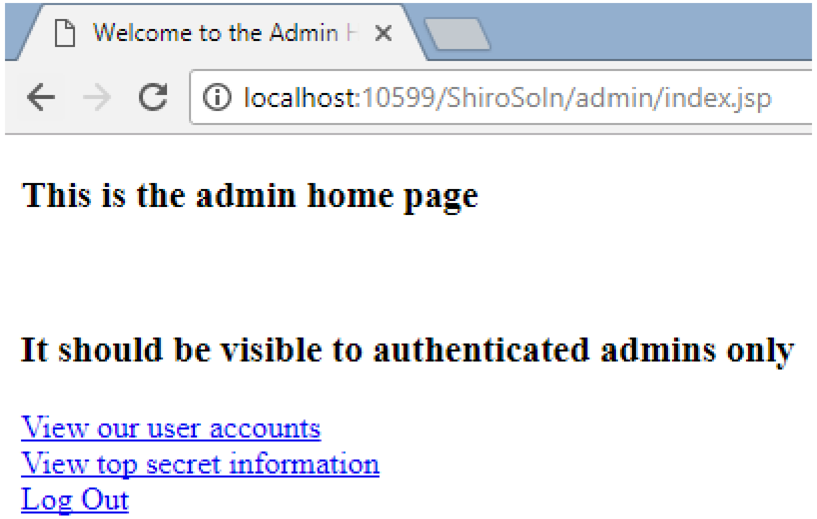
Username or password incorrect

Username: alan

Password: ••••••••

Login Reset

Remember me: □

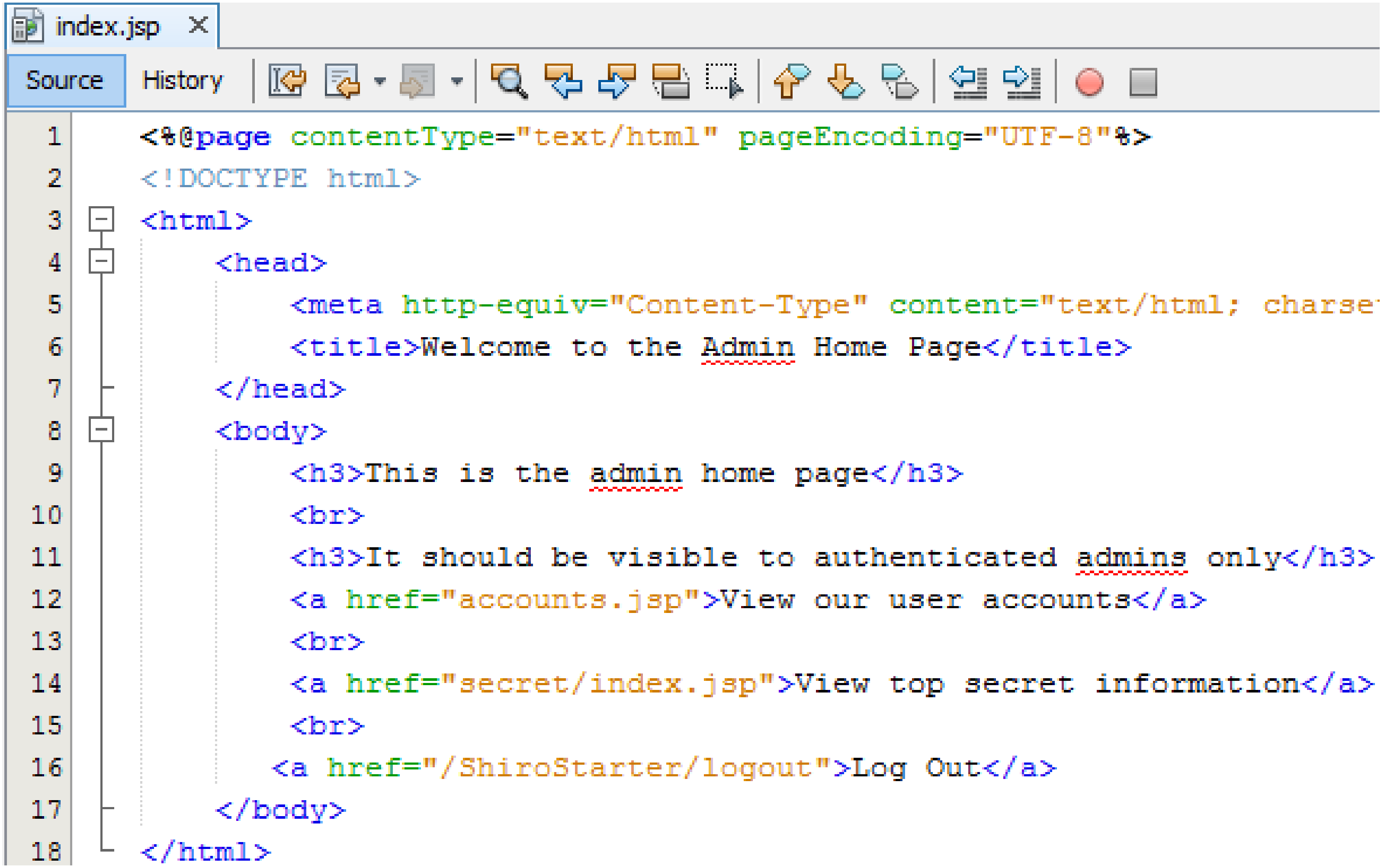


This is the admin home page

It should be visible to authenticated admins only

[View our user accounts](#)
[View top secret information](#)
[Log Out](#)

Admin index page



```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charse
        <title>Welcome to the Admin Home Page</title>
    </head>
    <body>
        <h3>This is the admin home page</h3>
        <br>
        <h3>It should be visible to authenticated admins only</h3>
        <a href="accounts.jsp">View our user accounts</a>
        <br>
        <a href="secret/index.jsp">View top secret information</a>
        <br>
       <a href="/ShiroStarter/logout">Log Out</a>
    </body>
</html>
```

Shiro JSP tags

Tag	Description
<shiro:guest/>	Displays content only if the current Subject IS NOT known to the system, either because they have not logged in or they have no corresponding 'RememberMe' identity.
<shiro:user/>	Displays content only if the current Subject has a known identity, either from a previous login or from 'RememberMe' services
<shiro:principal/>	Displays the user's principal or a property of the user's principal.
<shiro:hasPermission/>	Displays content only if the current Subject (user) 'has' (implies) the specified permission (i.e the user has the specified ability).

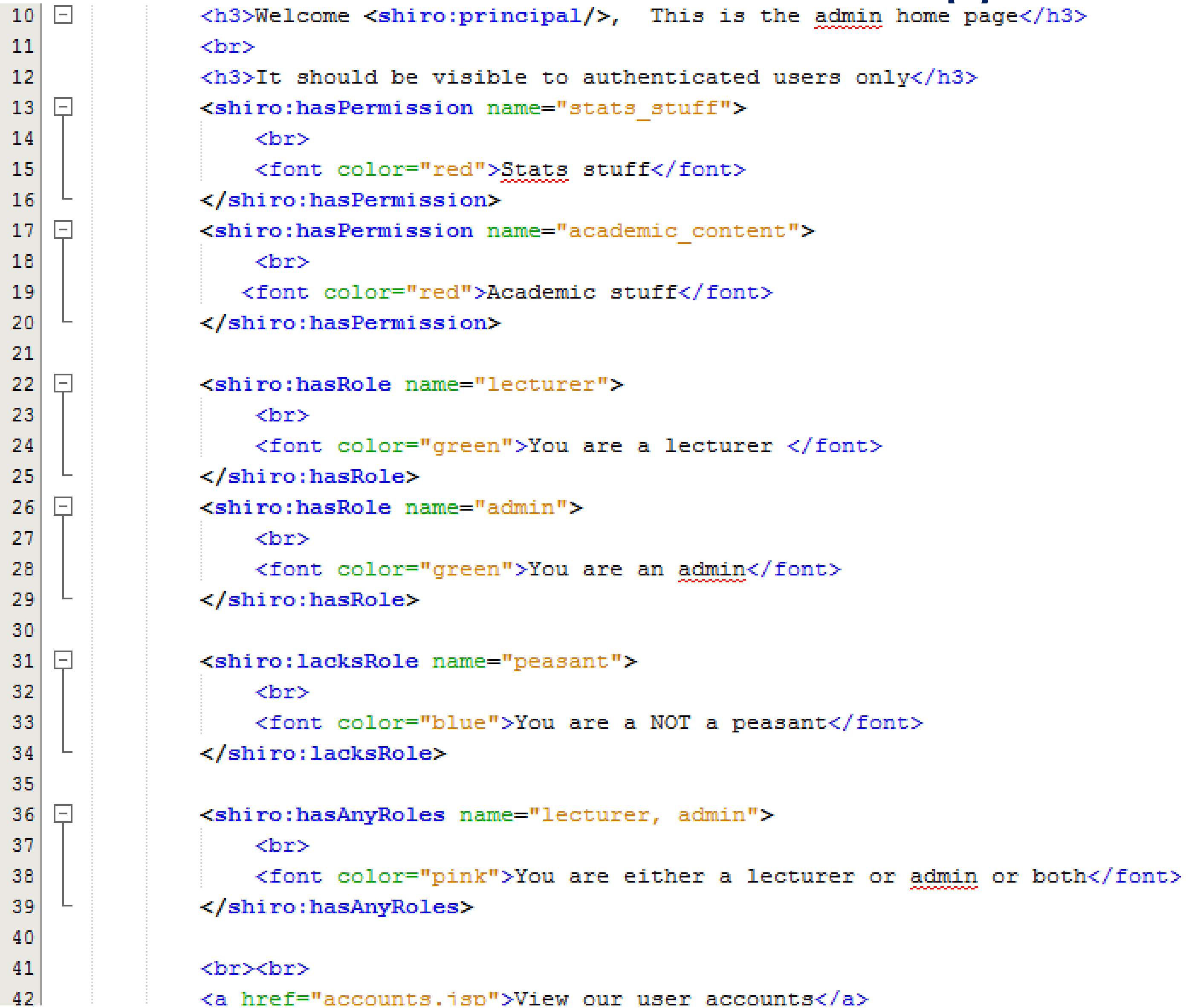
Shiro JSP tags

Tag	Description
<shiro:lacksPermission/>	Displays content only if the current Subject (user) does NOT have (not imply) the specified permission (i.e. the user lacks the specified ability)
<shiro:hasRole/>	Displays content only if the current user has the specified role.
<shiro:lacksRole/>	Displays content only if the current user does NOT have the specified role.
<shiro:hasAnyRoles/>	Displays content only if the current user has one of the specified roles from a comma-separated list of role names

Shiro JSP tags

Tag	Description
<shiro:authenticated/>	Displays content only if the current user has successfully authenticated *during their current session*.
<shiro:notAuthenticated/>	Displays content only if the current user has NOT successfully authenticated *during their current session*.

Shiro JSP tags example



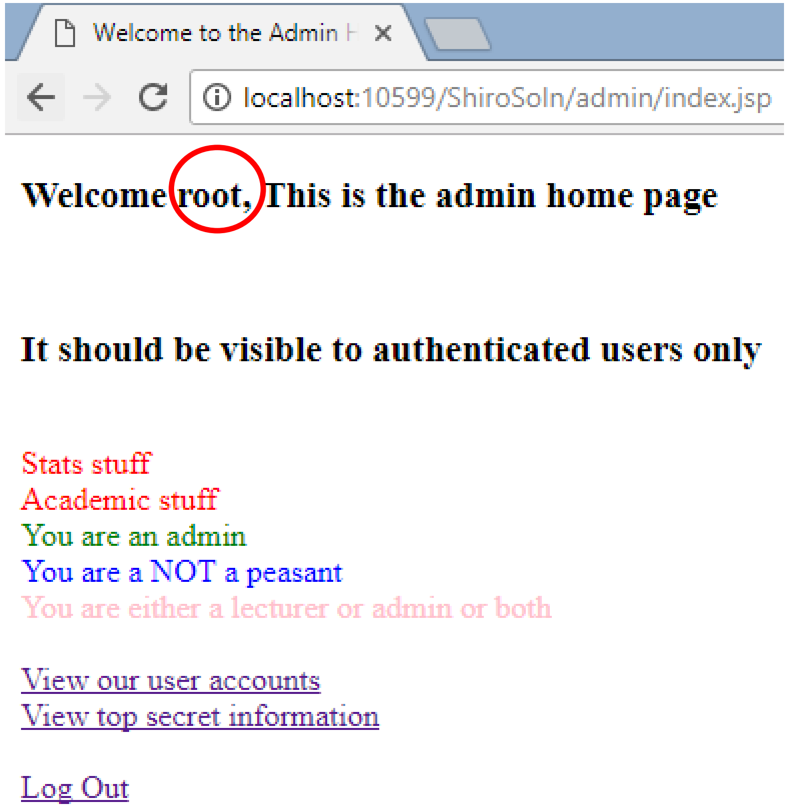
```
<h3>Welcome <shiro:principal/>,  This is the admin home page</h3>
<br>
<h3>It should be visible to authenticated users only</h3>
<shiro:hasPermission name="stats_stuff">
    <br>
    <font color="red">Stats stuff</font>
</shiro:hasPermission>
<shiro:hasPermission name="academic_content">
    <br>
   <font color="red">Academic stuff</font>
</shiro:hasPermission>

<shiro:hasRole name="lecturer">
    <br>
    <font color="green">You are a lecturer </font>
</shiro:hasRole>
<shiro:hasRole name="admin">
    <br>
    <font color="green">You are an admin</font>
</shiro:hasRole>

<shiro:lacksRole name="peasant">
    <br>
    <font color="blue">You are a NOT a peasant</font>
</shiro:lacksRole>

<shiro:hasAnyRoles name="lecturer, admin">
    <br>
    <font color="pink">You are either a lecturer or admin or both</font>
</shiro:hasAnyRoles>

<br><br>
<a href="accounts.jsp">View our user accounts</a>
```

Reworked index page for the admin section

]

Shiro JSP tags example

Output for various user accounts



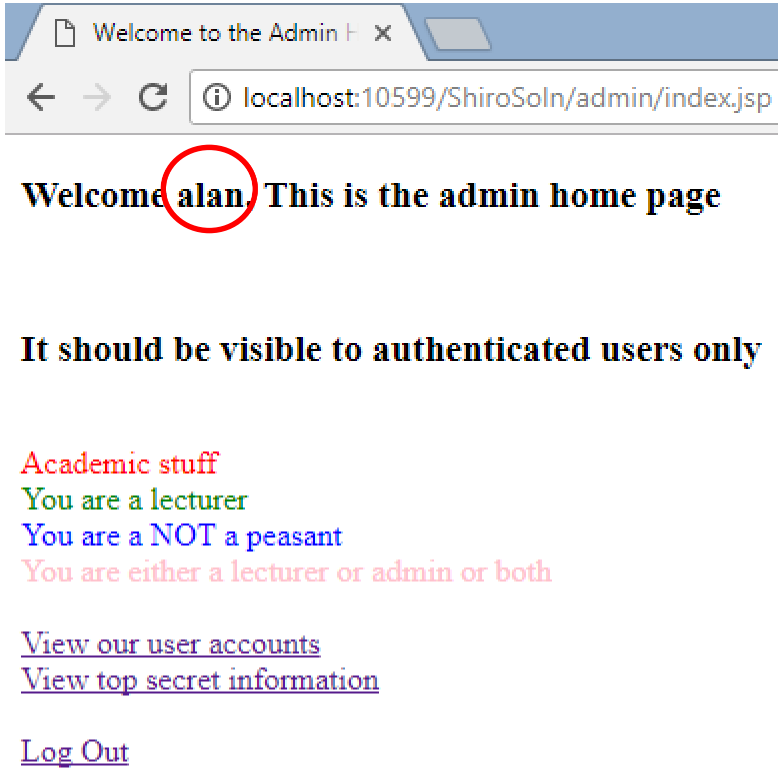
Welcome root, This is the admin home page

It should be visible to authenticated users only

Stats stuff
Academic stuff
You are an admin
You are a NOT a peasant
You are either a lecturer or admin or both

View our user accounts
View top secret information

Log Out



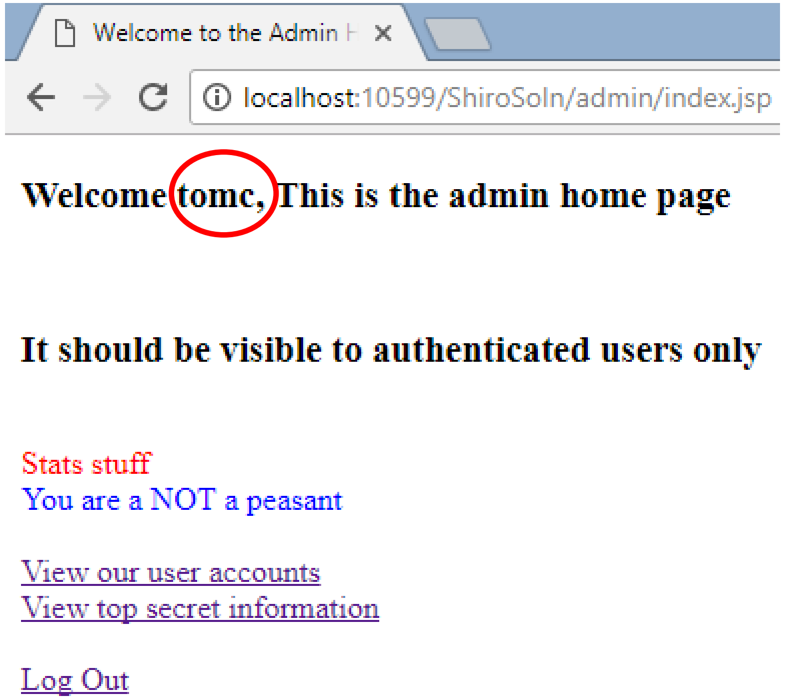
Welcome alan, This is the admin home page

It should be visible to authenticated users only

Academic stuff
You are a lecturer
You are a NOT a peasant
You are either a lecturer or admin or both

View our user accounts
View top secret information

Log Out



Welcome tomc, This is the admin home page

It should be visible to authenticated users only

Stats stuff
You are a NOT a peasant

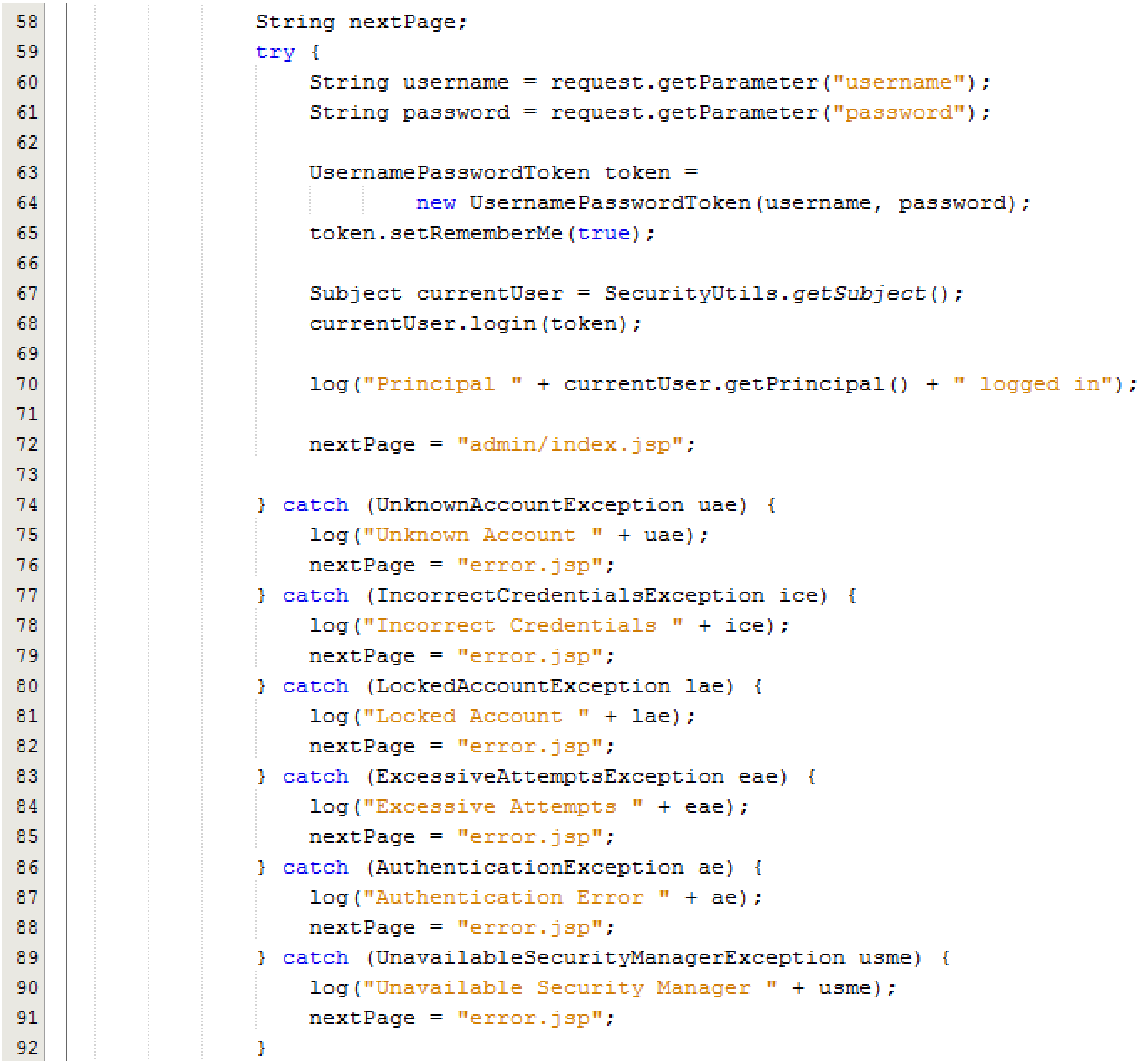
View our user accounts
View top secret information

Log Out

Shiro Filters

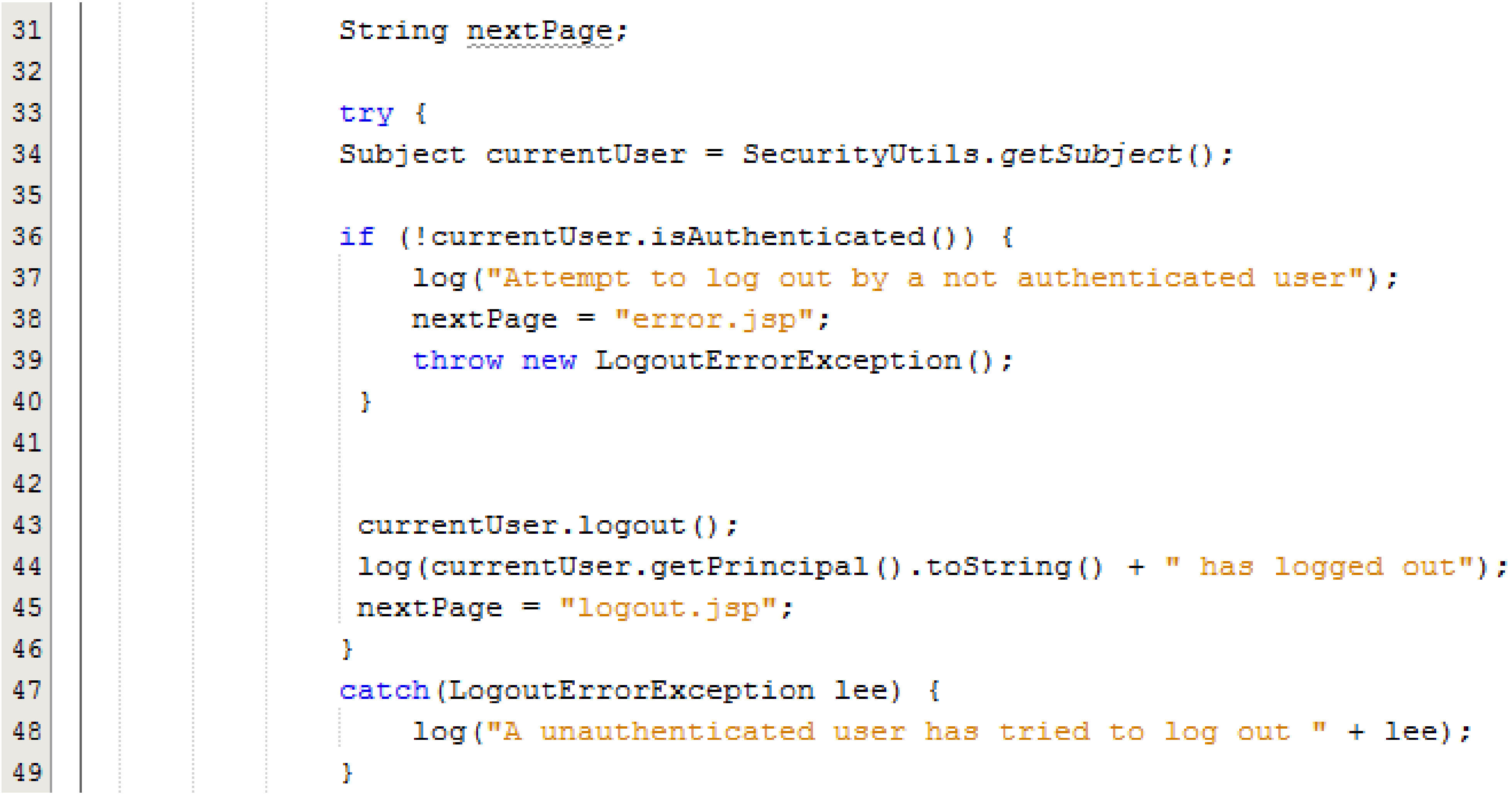
Filter	Class
anon	org.apache.shiro.web.filter.authc.AnonymousFilter
authc	org.apache.shiro.web.filter.authc.FormAuthenticationFilter
authcBasic	org.apache.shiro.web.filter.authc.BasicHttpAuthenticationFilter
logout	org.apache.shiro.web.filter.authc.LogoutFilter
noSessionCreation	org.apache.shiro.web.filter.session.NoSessionCreationFilter
perms	org.apache.shiro.web.filter.authz.PermissionsAuthorizationFilter
port	org.apache.shiro.web.filter.authz.PortFilter
rest	org.apache.shiro.web.filter.authz.HttpMethodPermissionFilter
roles	org.apache.shiro.web.filter.authz.RolesAuthorizationFilter
ssl	org.apache.shiro.web.filter.authz.SslFilter
user	org.apache.shiro.web.filter.authc.UserFilter

Equivalent Servlet code - Login



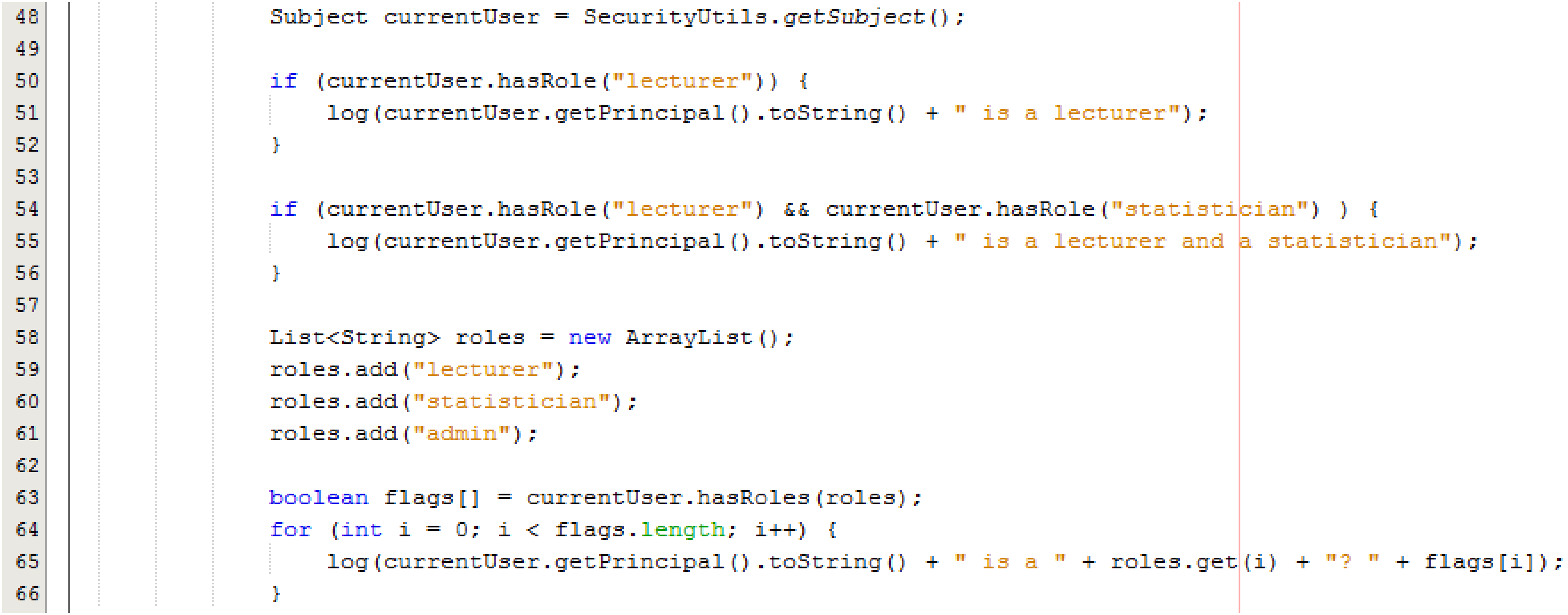
```java
String nextPage;
try {
    String username = request.getParameter("username");
    String password = request.getParameter("password");

    UsernamePasswordToken token =
            new UsernamePasswordToken(username, password);
    token.setRememberMe(true);

    Subject currentUser = SecurityUtils.getSubject();
    currentUser.login(token);

    log("Principal " + currentUser.getPrincipal() + " logged in");

    nextPage = "admin/index.jsp";

} catch (UnknownAccountException uae) {
    log("Unknown Account " + uae);
    nextPage = "error.jsp";
} catch (IncorrectCredentialsException ice) {
    log("Incorrect Credentials " + ice);
    nextPage = "error.jsp";
} catch (LockedAccountException lae) {
    log("Locked Account " + lae);
    nextPage = "error.jsp";
} catch (ExcessiveAttemptsException eae) {
    log("Excessive Attempts " + eae);
    nextPage = "error.jsp";
} catch (AuthenticationException ae) {
    log("Authentication Error " + ae);
    nextPage = "error.jsp";
} catch (UnavailableSecurityManagerException usme) {
    log("Unavailable Security Manager " + usme);
    nextPage = "error.jsp";
}
```

Equivalent Servlet code - Logout



```
String nextPage;

try {
Subject currentUser = SecurityUtils.getSubject();

if (!currentUser.isAuthenticated()) {
    log("Attempt to log out by a not authenticated user");
    nextPage = "error.jsp";
    throw new LogoutErrorException();
 }


 currentUser.logout();
 log(currentUser.getPrincipal().toString() + " has logged out");
 nextPage = "logout.jsp";
}
catch(LogoutErrorException lee) {
    log("A unauthenticated user has tried to log out " + lee);
}
```

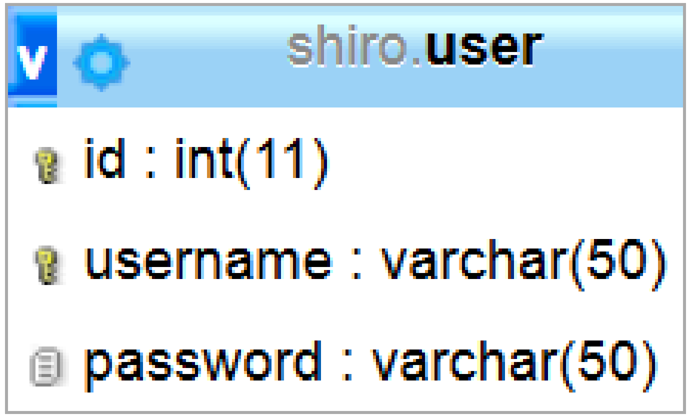
Equivalent Servlet code – Check Roles



```java
Subject currentUser = SecurityUtils.getSubject();

if (currentUser.hasRole("lecturer")) {
    log(currentUser.getPrincipal().toString() + " is a lecturer");
}

if (currentUser.hasRole("lecturer") && currentUser.hasRole("statistician") ) {
    log(currentUser.getPrincipal().toString() + " is a lecturer and a statistician");
}

List<String> roles = new ArrayList();
roles.add("lecturer");
roles.add("statistician");
roles.add("admin");

boolean flags[] = currentUser.hasRoles(roles);
for (int i = 0; i < flags.length; i++) {
    log(currentUser.getPrincipal().toString() + " is a " + roles.get(i) + "? " + flags[i]);
}
```

Creating a database realm

Create the necessary tables within the database

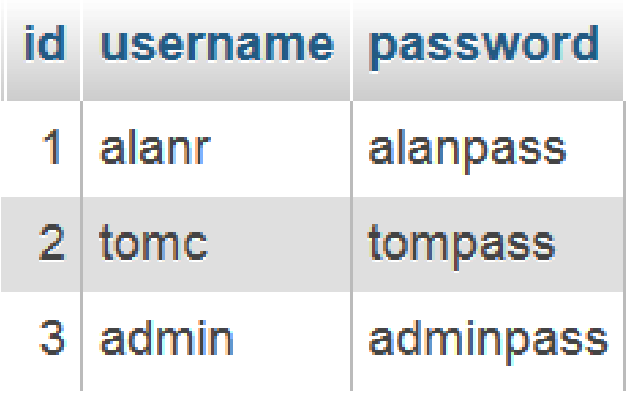


shiro.user

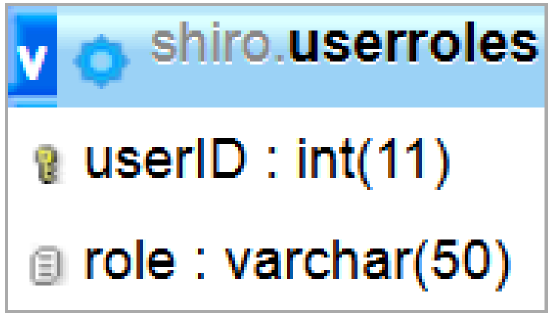
id : int(11)

username : varchar(50)

password : varchar(50)



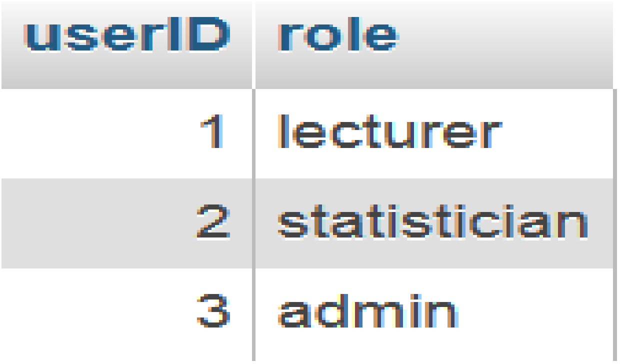
id	username	password
1	alanr	alanpass
2	tomc	tompass
3	admin	adminpass



shiro.userroles

userID : int(11)

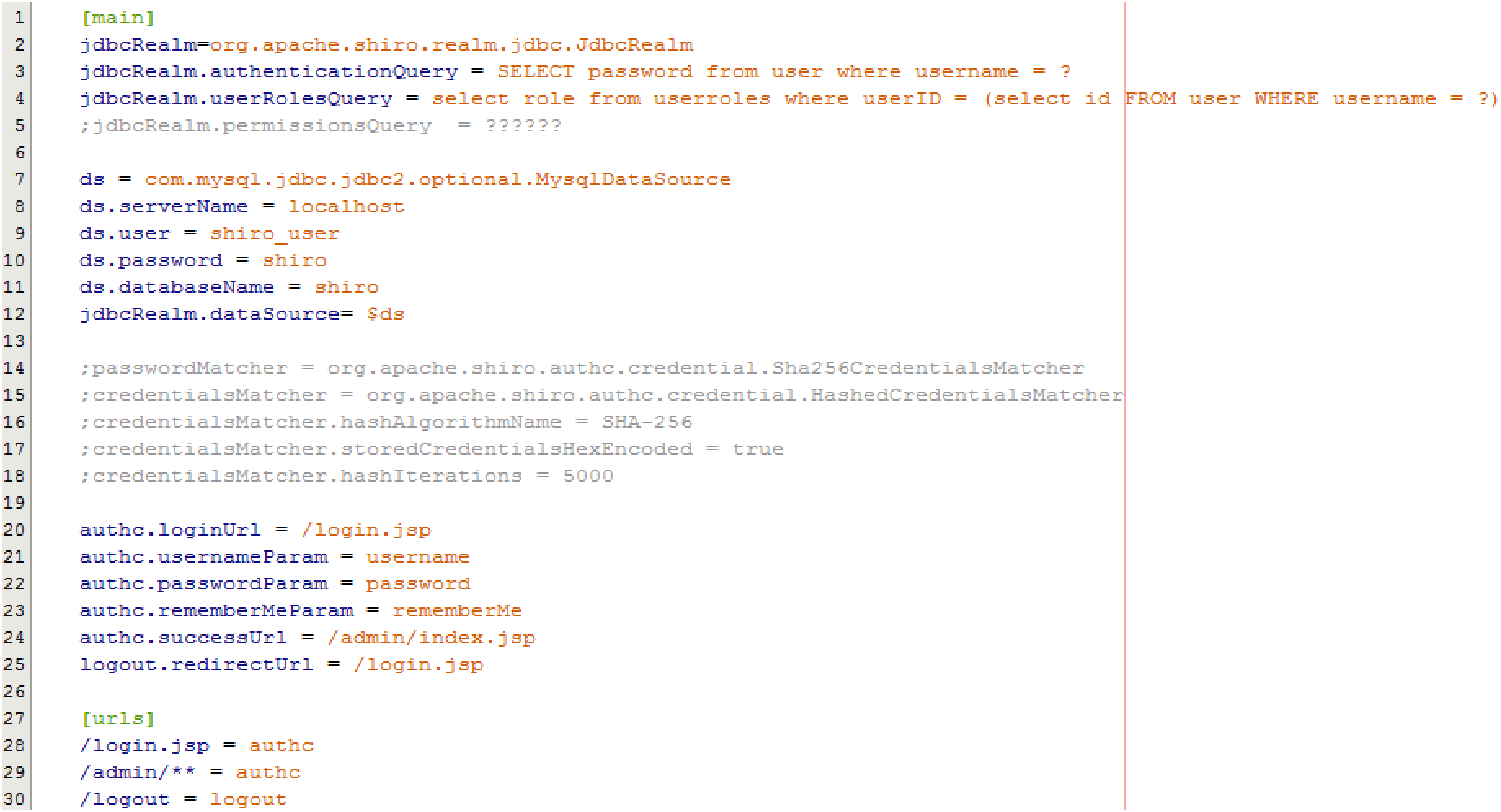
role : varchar(50)



userID	role
1	lecturer
2	statistician
3	admin

Creating a database realm

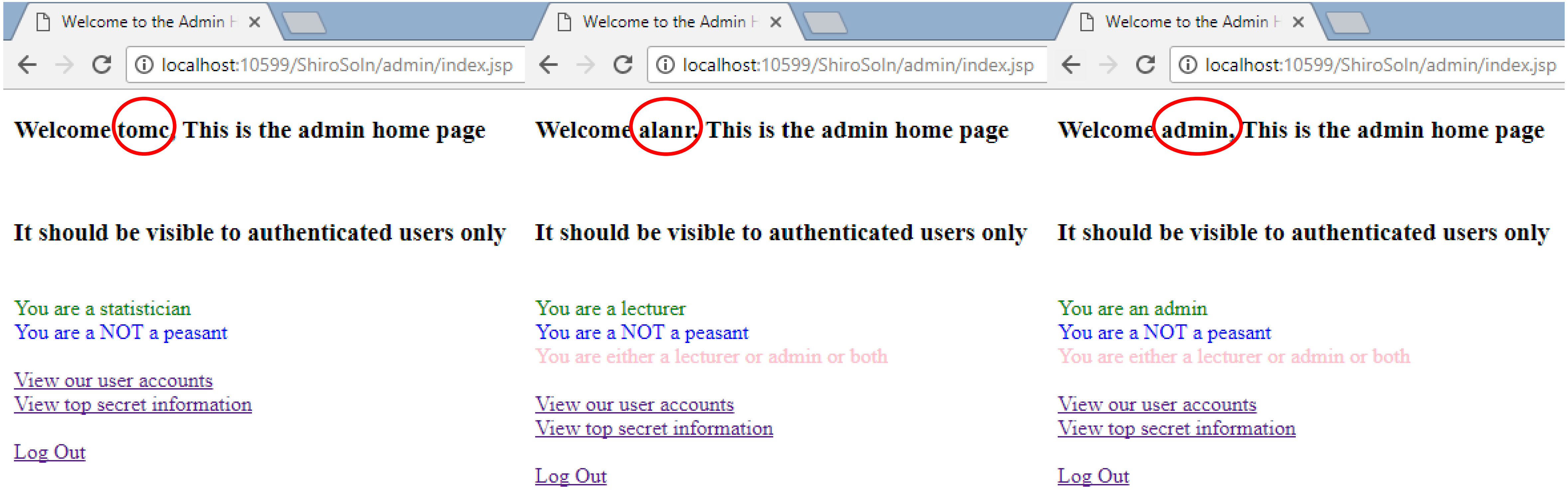
Adjusted *shiro.ini*



```
[main]
jdbcRealm=org.apache.shiro.realm.jdbc.JdbcRealm
jdbcRealm.authenticationQuery = SELECT password from user where username = ?
jdbcRealm.userRolesQuery = select role from userroles where userID = (select id FROM user WHERE username = ?)
;jdbcRealm.permissionsQuery  = ??????

ds = com.mysql.jdbc.jdbc2.optional.MysqlDataSource
ds.serverName = localhost
ds.user = shiro_user
ds.password = shiro
ds.databaseName = shiro
jdbcRealm.dataSource= $ds

;passwordMatcher = org.apache.shiro.authc.credential.Sha256CredentialsMatcher
;credentialsMatcher = org.apache.shiro.authc.credential.HashedCredentialsMatcher
;credentialsMatcher.hashAlgorithmName = SHA-256
;credentialsMatcher.storedCredentialsHexEncoded = true
;credentialsMatcher.hashIterations = 5000

authc.loginUrl = /login.jsp
authc.usernameParam = username
authc.passwordParam = password
authc.rememberMeParam = rememberMe
authc.successUrl = /admin/index.jsp
logout.redirectUrl = /login.jsp

[urls]
/login.jsp = authc
/admin/** = authc
/logout = logout
```

Creating a database realm

Output for various user accounts


Welcome to the Admin H
localhost:10599/ShiroSoln/admin/index.jsp
Welcome to the Admin H
localhost:10599/ShiroSoln/admin/index.jsp
Welcome to the Admin H
localhost:10599/ShiroSoln/admin/index.jsp

Welcome tomc, This is the admin home page

It should be visible to authenticated users only

You are a statistician
You are a NOT a peasant

View our user accounts
View top secret information

Log Out

Welcome alanr, This is the admin home page

It should be visible to authenticated users only

You are a lecturer
You are a NOT a peasant
You are either a lecturer or admin or both

View our user accounts
View top secret information

Log Out

Welcome admin, This is the admin home page

It should be visible to authenticated users only

You are an admin
You are a NOT a peasant
You are either a lecturer or admin or both

View our user accounts
View top secret information

Log Out

References

https://shiro.apache.org/

http://meri-stuff.blogspot.ie/2011/03/apache-shiro-part-1-basics.html

http://www.jjoe64.com/2014/01/apache-shiro-with-hibernatesql-full.html

http://balusc.omnifaces.org/2013/01/apache-shiro-is-it-ready-for-java-ee-6.html

http://www.baeldung.com/apache-shiro