

CPRG 352 – Web Application Programming

SAIT – Fall 2019

FINAL PROJECT

NOTE: ALL WORK FOR THIS COURSE IS INDIVIDUAL WORK. YOU MAY NOT WORK CLOSELY WITH OTHERS TO IMPLEMENT YOUR SOLUTION TO THIS PROBLEM. FAILURE TO COMPLETE THE ASSIGNMENT INDIVIDUALLY WILL RESULT IN PENALTIES BEING ASSESSED AGAINST YOU. IF YOU ARE NOT SURE HOW MUCH COLLABORATION IS TOO MUCH, ASK YOUR INSTRUCTOR FIRST!

Deadline: *5pm, Thursday, 5th December 2019*

Submission: When finished zip the entire NetBeans project folder containing your solution into a file called ***FinalProjectFall2019.zip*** and upload it to BrightSpace. Late submissions cannot be accepted and you will receive a mark of zero for the assignment. In the event of exceptional circumstances please contact your instructor.

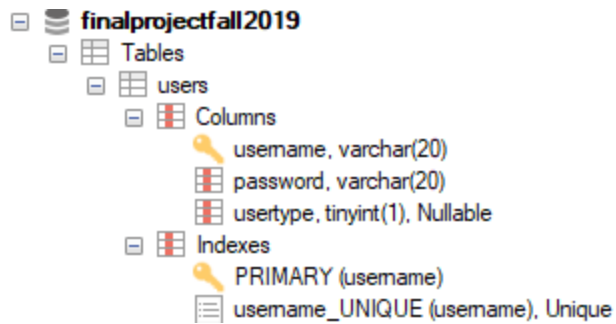
Introduction

For this project you will create a Java web application called **finalProjectFall2019**. It is the first phase in an order management application. The main focus on this first phase is basic user account management, including:

- *Normal* user functionality:
 - self-registration with the application to create an account
 - Password change
- *Admin* functionality:
 - Listing user accounts
 - User account deletion
 - Account type toggling between admin and normal user types
 - Password reset (to “***password***”) for a selected user account

The application works with user data from the users table in the provided **finalprojectfall2019** MySQL database.

The structure of the database as given to you is:



The initial set of data in the users table in the data is:

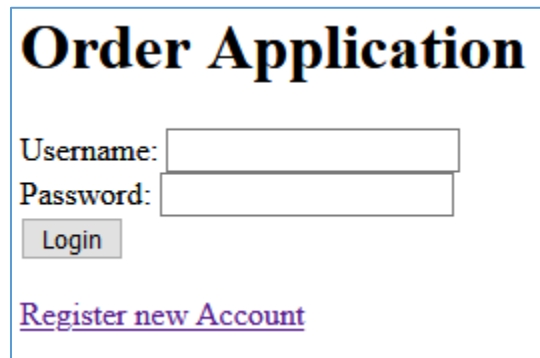
username	password	usertype
adam	adam	0
admin	admin	1
betty	betty	0

The **usertype** column is a Boolean (**tinyint(1)**) field, if the value in it is **0** then the user is a normal user, if it is **1** the user is an administrator. Any account can be of either user type (the fact that there is an account named “**admin**” is not significant, it could be set to be a normal account type if you wanted to do so).

SAMPLE RUN OF APPLICATION

NOTE: The screenshots shown in this sample run show required application functionality only. For this project you must also use CSS to provide an attractive look to your application. Solutions that look very basic or crude on-screen will lose marks.

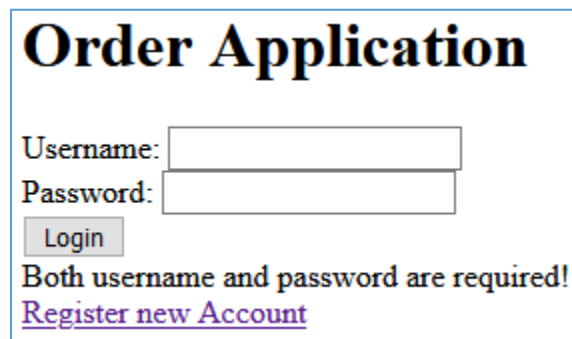
1. Viewing default page (Login)



The screenshot shows a web form titled "Order Application". It contains two input fields: "Username:" and "Password:". Below the password field is a "Login" button. At the bottom of the form is a link that says "Register new Account".

The login page is the default page for the application. The user can log into the application here, or select the **Register new Account** link to move to another page if they want to create a new account for themselves with the application.

If the user clicks on the **Login** button without filling in both username and password then the message below should be shown:



The screenshot shows the same "Order Application" login form as before, but with an additional message: "Both username and password are required!". The "Register new Account" link is still present at the bottom.

If invalid credentials are entered then the application should show the message:

Order Application

Username:
Password:

Invalid username or password!

[Register new Account](#)

If the user has already logged in and has now logged out this page should show:

Order Application

Username:
Password:

Logged out

[Register new Account](#)

2. User registers a new account (Register)

Clicking on the **Register new Account** link in the login page should show a separate registration page:

Register New Account

Username:
Password:
Confirm password:

[Login](#)

Clicking on the **Login** link takes the user back to the login page.

The user can enter a username and a password, with a second confirmatory password (which must match the first password to be valid). If the user does not enter data into all the fields the following message is displayed:



Register New Account

Username:

Password:

Confirm password:

All values are required!

[Login](#)

If the user enters data but the two passwords do not match then the application displays:



Register New Account

Username:

Password:

Confirm password:

Passwords do not match!

[Login](#)

Entering valid data for a new user creates the account in the system and takes the user back to the login page, which displays the following message:

Order Application

Username:

Password:

Login

New account created, please log in

[Register new Account](#)

3. Normal User Type Logs In

If a user with the normal user account type logs in then they should see the following page:

Orders

Welcome, adam

[Logout](#)

[Set password](#)

Orders will be here in version 2...

The “**Welcome**” message at the top of the page includes the username of the current user (“**adam**” in this case), which should be stored in a **session** object for each user. Beside that is a **Logout** link that logs the user out and takes him/her back to the login page.

The **Set password** link refreshes the page and causes it to show a **Set Password** form in the current page¹ that the user can use to change their password, e.g.:

¹ This form must be shown in the current page, you cannot move to another page to show it!

Orders

Welcome, adam

[Logout](#)

[Set password](#)

Set Password

Enter password:

Confirm password:

Orders will be here in version 2...

If the user does not fill in both fields and clicks on the “**Set Password**” button then they will see the following message:

Orders

Welcome, adam

[Logout](#)

[Set password](#)

Set Password

Enter password:

Confirm password:

Both password values are required!

Orders will be here in version 2...

Entering in password that do not match and clicking on **Set Password** shows:

Orders

Welcome, adam

[Logout](#)

[Set password](#)

Set Password

Enter password:

Confirm password:

Passwords do not match!

Orders will be here in version 2...

Entering in matching passwords shows:

Orders

Welcome, adam

[Logout](#)

[Set password](#)

Password reset

Orders will be here in version 2...

The password is set for the user, and the page refreshes without the **Set Password** form being displayed. Note the “**Password Reset**” message displayed beside the **Set Password** link in the above.

4. Administrative user logs in

If a user with the administrator user type logs in they should see the administration page for the application:

Administration Page

Welcome, admin

[Logout](#)

User Accounts

Username	User Type	Delete	Toggle Account Type	Reset Password
adam	Normal	Delete	Toggle	Reset
admin	Admin	Delete	Toggle	Reset
betty	Normal	Delete	Toggle	Reset
charlie	Normal	Delete	Toggle	Reset

Again a “**Welcome**” message that includes the logged-in user’s name and a **Logout** link should be displayed at the top of the page. Beneath that a table called **User Accounts** is shown. The admin user can see all of the user accounts in the system in this table.

Clicking on the “**Delete**” link will delete the relevant user account from the system. A confirmatory message should be displayed in the browser before the account is actually deleted, e.g.:

Administration Page

Welcome, admin
[Logout](#)

User Accounts

Username	User Type	Delete	Toggle Account Type	Reset Password
adam	Normal	Delete	Toggle	Reset
admin	Admin	Delete	Toggle	Reset
betty	Normal	Delete	Toggle	Reset
charlie	Normal	Delete	Toggle	Reset

Delete account?

The user can choose to delete the account (“**OK**”), or cancel the deletion operation (“**Cancel**”).

If the user chooses to delete the account then the following message should be shown (which includes the username of the account that was deleted):

Administration Page

Welcome, admin
[Logout](#)

User Accounts

Username	User Type	Delete	Toggle Account Type	Reset Password
adam	Normal	Delete	Toggle	Reset
admin	Admin	Delete	Toggle	Reset
betty	Normal	Delete	Toggle	Reset

Account 'charlie' deleted

The system must have *at least one administrative user*, so if the **Delete** link beside the “**admin**” user is clicked and the user confirms the deletion then the account must not be deleted, and the following message should be displayed:

Administration Page

Welcome, admin
[Logout](#)

User Accounts

Username	User Type	Delete	Toggle Account Type	Reset Password
adam	Normal	Delete	Toggle	Reset
admin	Admin	Delete	Toggle	Reset
betty	Normal	Delete	Toggle	Reset

Cannot delete, must have at least one admin user!

Clicking on the **Toggle** link beside a given user account toggles the account type between “**Admin**” and “**Normal**”, e.g.:

Administration Page

Welcome, admin
[Logout](#)

User Accounts

Username	User Type	Delete	Toggle Account Type	Reset Password
adam	Admin	Delete	Toggle	Reset
admin	Admin	Delete	Toggle	Reset
betty	Normal	Delete	Toggle	Reset

Type for account 'adam' toggled

In the above **adam**’s user account type was changed from “**Normal**” to “**Admin**”. Note the message below the table also. It contains the name of the account which had its type changed.

Clicking on the **Toggle** link beside **adam**’s account would change him back to a “**Normal**” user, e.g.:

Administration Page

Welcome, admin
[Logout](#)

User Accounts

Username	User Type	Delete	Toggle Account Type	Reset Password
adam	Normal	Delete	Toggle	Reset
admin	Admin	Delete	Toggle	Reset
betty	Normal	Delete	Toggle	Reset

Type for account 'adam' toggled

Since the system must have at least one administrative user, if the **Toggle** link beside the only “**Admin**” user (**admin**) is clicked then the following message must be displayed:

Administration Page

Welcome, admin
[Logout](#)

User Accounts

Username	User Type	Delete	Toggle Account Type	Reset Password
adam	Normal	Delete	Toggle	Reset
admin	Admin	Delete	Toggle	Reset
betty	Normal	Delete	Toggle	Reset

Cannot toggle account type, must have at least one admin user!

Selecting the **Reset** link beside a user sets that user’s password to the string “**password**” in the database. This option is used when a user forgets their password, e.g. if the password for **adam** is reset:

Administration Page

Welcome, admin
[Logout](#)

User Accounts

Username	User Type	Delete	Toggle Account Type	Reset Password
adam	Normal	Delete	Toggle	Reset
admin	Admin	Delete	Toggle	Reset
betty	Normal	Delete	Toggle	Reset

Password reset for 'adam'

Note the message displayed under the table.

The database should now contain “*password*” as **adam**’s password, e.g.:

<input type="checkbox"/>	username	password	usertype
<input type="checkbox"/>	adam	password	0

APPLICATION REQUIREMENTS

Your solution for this project must follow the technical requirements below:

- Be a Java web application
- Use a front-servlet design
- Use a connection pool called **jdbc/finalProjectFall2019** to get connections to the database
- Use only ***prepared statements*** to send queries to the database (***no Statements, no JPA!***)
- Use the database as provided to you. You can change the data, but not the structure (I will be testing your solution from grading purposes using a copy of the database as it is given to you)
- Include the “**Welcome**” message and “**Logout**” link into the normal and admin user pages from a JSP fragment (called a “*JSP segment*” in the NetBeans IDE) stored under **/WEB-INF/jspf**. This is to save you from having to replicate this common functionality across multiple pages
- Define and use a domain class called **User** to hold user data to be shown in the “**User Accounts**” table in the administrative page
- Use JSTL/EL to generate any dynamic content in the normal and admin user pages (no Java scriptlet code, no Java method calls in JSTL!)
- The application must implement and use two servlets which provide the services as described here:
 - **UserServices**: login validation, logout, new account registration, normal user password setting
 - **AdminServices**: account deletion, account type toggling, password resets to “*password*”
- It must implement and use a **DBoperations** class to contain the database-related operation methods
- Use **CSS²**, etc., to provide an attractive look-and-feel to users

² You cannot store CSS files under the /WEB-INF folder in a NetBeans project because the browser would not be able to access them. It is alright to store CSS files under the public “/Web Pages” folder and then link them into HTML or JSP pages that are stored under /WEB-INF.