CPRG 352 – Web Application Programming

SAIT – Fall 2021

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 COULD RESULT IN DISCIPLINARY ACTION AGAINST YOU. IF YOU ARE NOT SURE HOW MUCH COLLABORATION IS TOO MUCH, ASK YOUR INSTRUCTOR FIRST!**

Deadline: ***5pm, Friday, 10th December 2021***

Submission: When finished zip the entire NetBeans project folder containing your solution into a file called ***FinalProjectFall2021.zip*** and upload it to BrightSpace. Late submissions cannot be accepted and you will receive a mark of zero for the project. In the event of exceptional circumstances please contact your instructor.

Introduction

For this project you will create a Java web application called **finalProjectFall2021**. It is the first phase in a web-based business 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
  - Change registered username
- *Admin* functionality:
  - Listing all user accounts
  - User account deletion
  - Account type toggling between *admin* and *normal* user types
  - Toggle account status between "locked" and "unlocked"[1]

The application works with user data from the **users** table in the provided **finalProjectFall2021** MySQL database.

1 "locked" accounts cannot be used to log in to the system.

The structure of the database as given to you consists of a single table called ***users***, with the following columns:

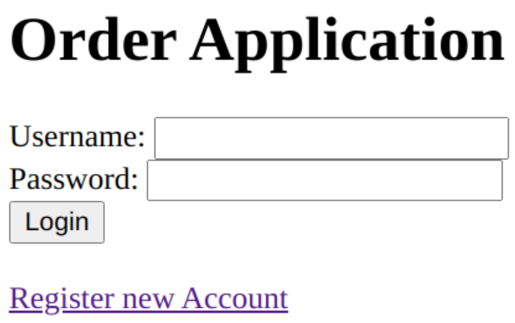
- **username**: name of the user account, varchar(20), not null, the primary key value for the table
- **password**: password for the user account, varchar(20), not null
- **usertype**: type of the user account (admin or normal), tinyint(1), defaults to 0 (normal user), can be set to 1 (admin user)
- **locked**: locked/unlocked status of the user account, tinyint(1), defaults to 0 (unlocked), can be set to 1 (locked)

Any account can be of either user type (the fact that there is a pre-created 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, i.e. too much like the example screenshots given in this document, will lose marks.***

1. ***Viewing default page (Login)***

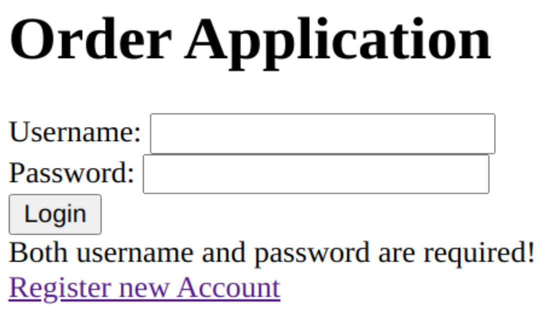

Order Application

Username:
Password:
Login

Register new Account

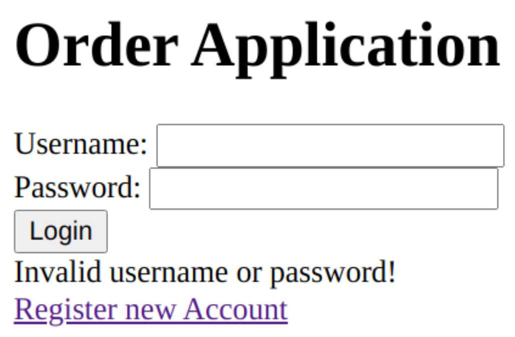
The login page is the default page for the application. The user can log into the application here, or select the **Register new Account** link to move to another page if they want to create a new account for themselves with the application.

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


Order Application

Username:
Password:
Login
Both username and password are required!
Register new Account

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



Order Application

Username:

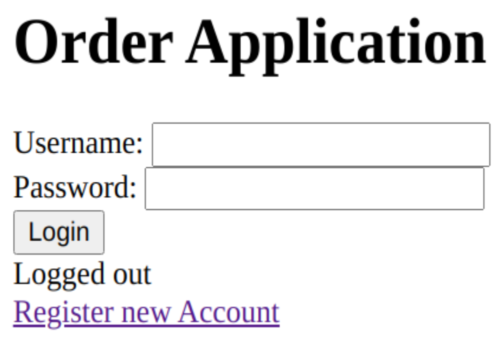
Password:

Login

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:

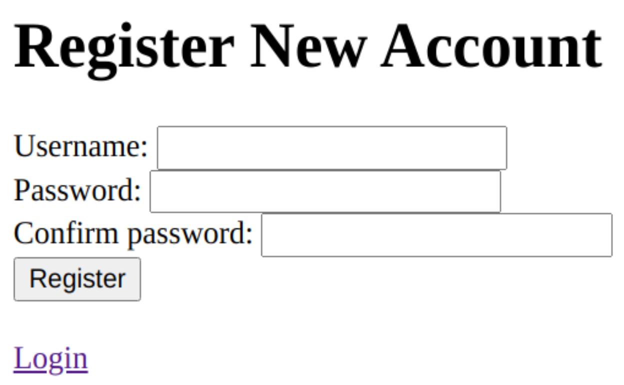
Login

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:

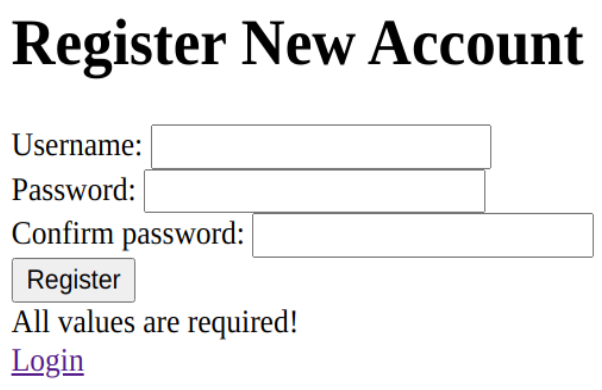
Confirm password:

Register

Login

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

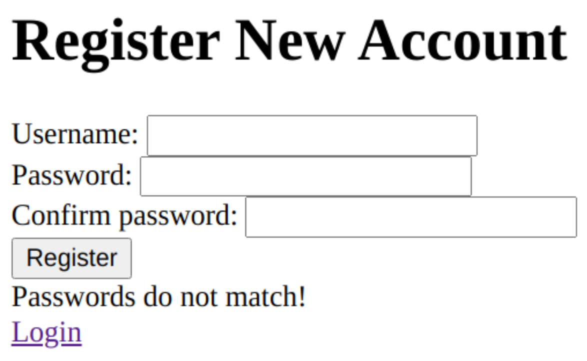
The user can enter a username and a password, with a second confirmatory password (which must match the first password to be valid). If the user does not enter data into all the fields the following message is displayed:


Register New Account

Username:
Password:
Confirm password:

All values are required!
Login

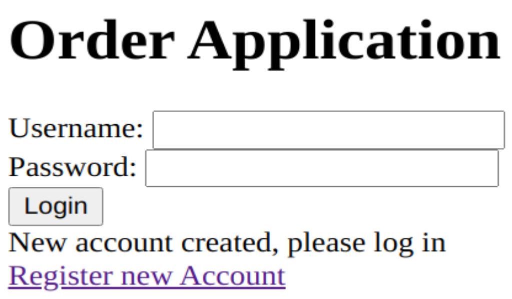
If the user enters data but the two passwords do not match then the application displays:


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:

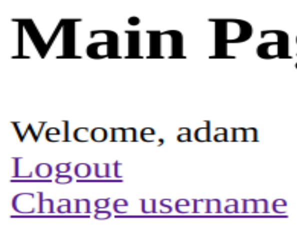
Login

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:


Main Page

Welcome, adam

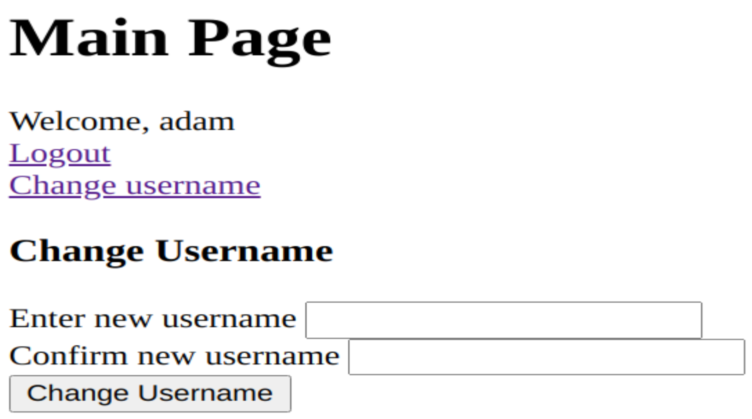
Logout

Change username

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. Beneath that is a **Logout** link that logs the user out and takes him/her back to the login page.

The **Change username** link refreshes the page and causes it to show a **Change Username** form in the current page[2] that the user can use to change their password, e.g.:

2 This form must be shown in the *current page* (Main Page), you cannot move to another page to show it!

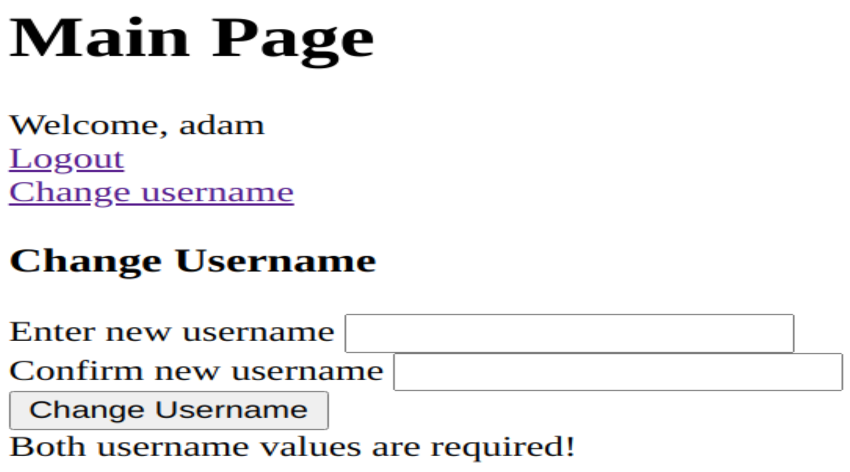

Main Page

Welcome, adam
Logout
Change username

Change Username

Enter new username
Confirm new username

If the user does not fill in both fields and clicks on the "**Change Username**" button then they will see the following message:


Main Page

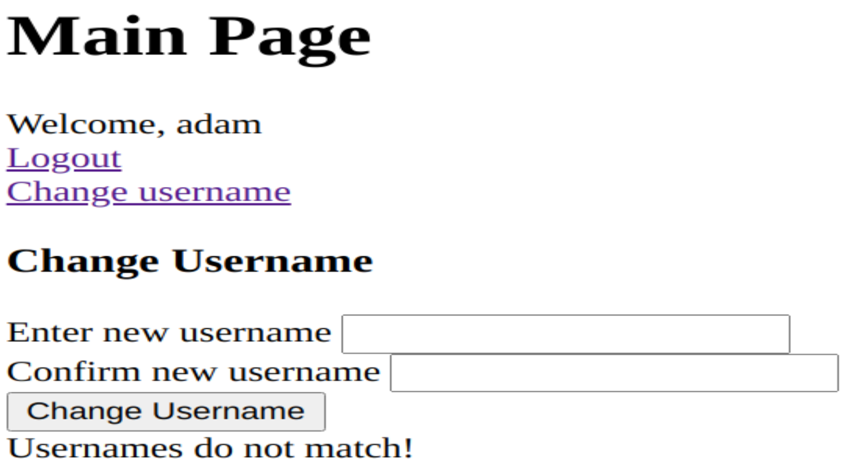
Welcome, adam
Logout
Change username

Change Username

Enter new username
Confirm new username

Both username values are required!

Entering in usernames that do not match and clicking on **Change Username** shows:


Main Page

Welcome, adam

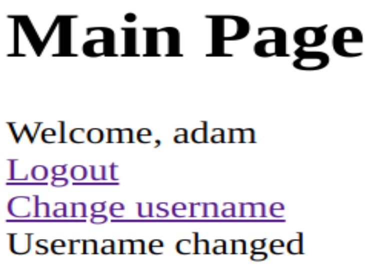
Change Username

Enter new username

Confirm new username

Usernames do not match!

Entering in matching usernames ("***Adam***" in this example) shows:


Main Page

Welcome, adam

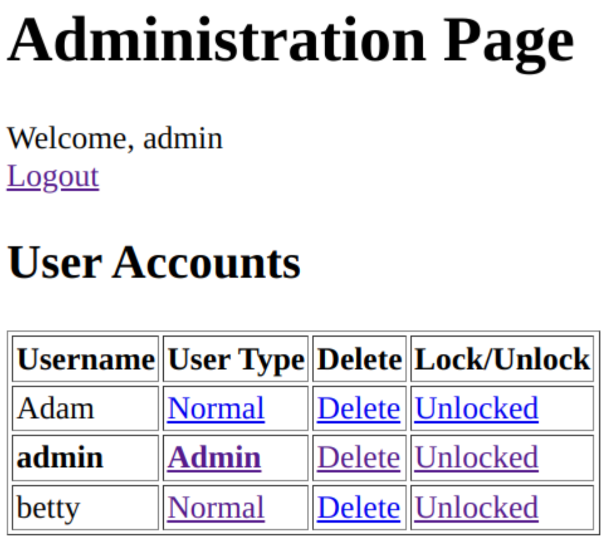
Username changed

The username is changed for the user, and the page refreshes without the **Change Username** form being displayed. Note the "***Username changed***" message displayed beneath the **Change username** link in the above.

Note: you will not see the changed username displayed in the "***Welcome***" message for the page until you log out and log back in as "***Adam***".

4. ***<u>Administrative user logs in</u>***

If a user with the administrator user type, e.g. "***admin***" in the initial version of the database, logs in they should see the administration page for the application:



Administration Page

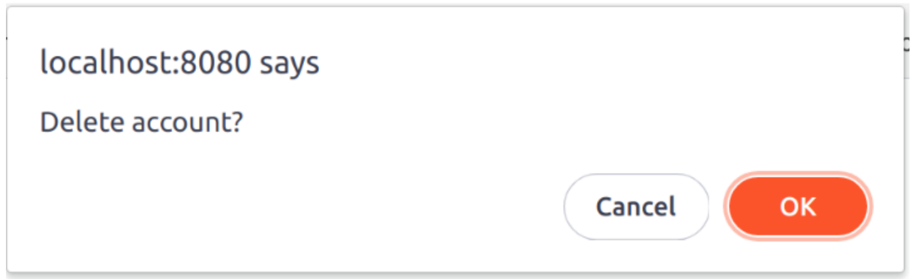
Welcome, admin
Logout

User Accounts

Username	User Type	Delete	Lock/Unlock
Adam	Normal	Delete	Unlocked
admin	**Admin**	Delete	Unlocked
betty	Normal	Delete	Unlocked

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 ***JavaScript*** message should be displayed in the browser *before* the account is actually deleted, e.g.:



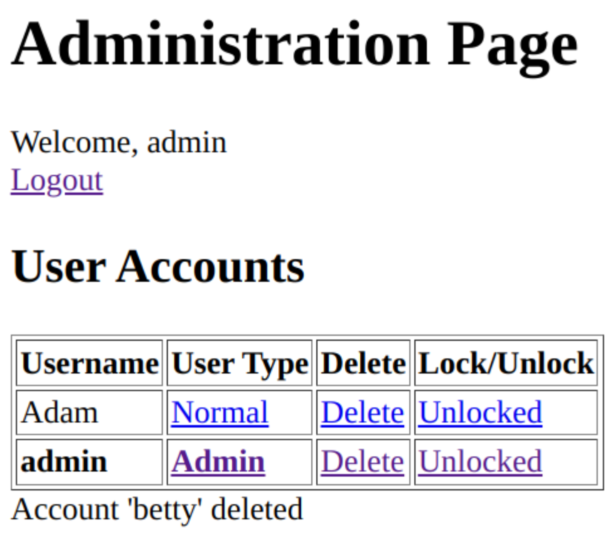
localhost:8080 says

Delete account?

Note: the exact representation on-screen of this JavaScript dialog box is determined by the browser being used.

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

If the user chooses to delete the account, for example "***betty***", then the following message should be shown (which includes the username of the account that was deleted):



Administration Page

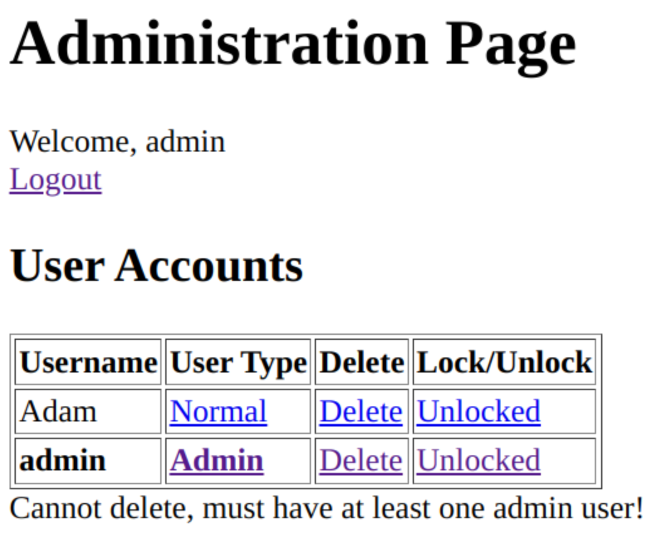
Welcome, admin
Logout

User Accounts

Username	User Type	Delete	Lock/Unlock
Adam	Normal	Delete	Unlocked
admin	**Admin**	Delete	Unlocked

Account 'betty' deleted

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



Administration Page

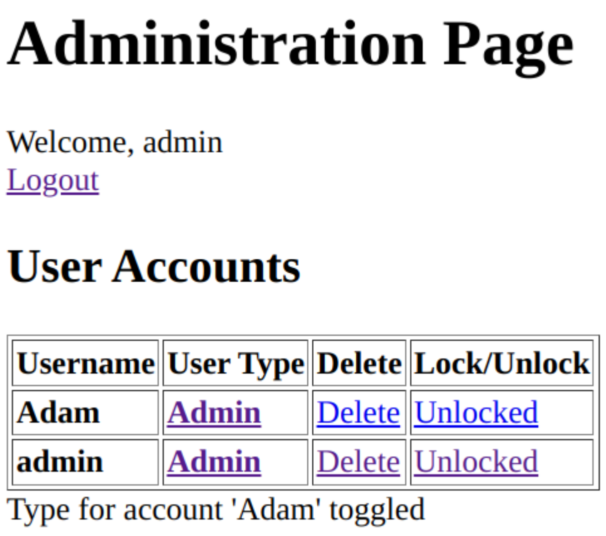
Welcome, admin
Logout

User Accounts

Username	User Type	Delete	Lock/Unlock
Adam	Normal	Delete	Unlocked
admin	**Admin**	Delete	Unlocked

Cannot delete, must have at least one admin user!

Clicking on the link in the **User Type** column for a given user account toggles the account type between “***Admin***” and “***Normal***”, e.g.:



Administration Page

Welcome, admin
Logout

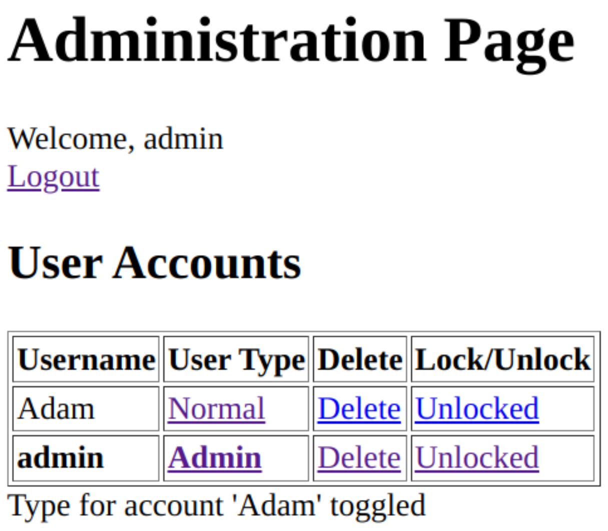
User Accounts

Username	User Type	Delete	Lock/Unlock
Adam	**Admin**	Delete	Unlocked
admin	**Admin**	Delete	Unlocked

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 **User Type** link beside **Adam**'s account when he is an "***Admin***" would change him back to a "***Normal***" user, e.g.:



Administration Page

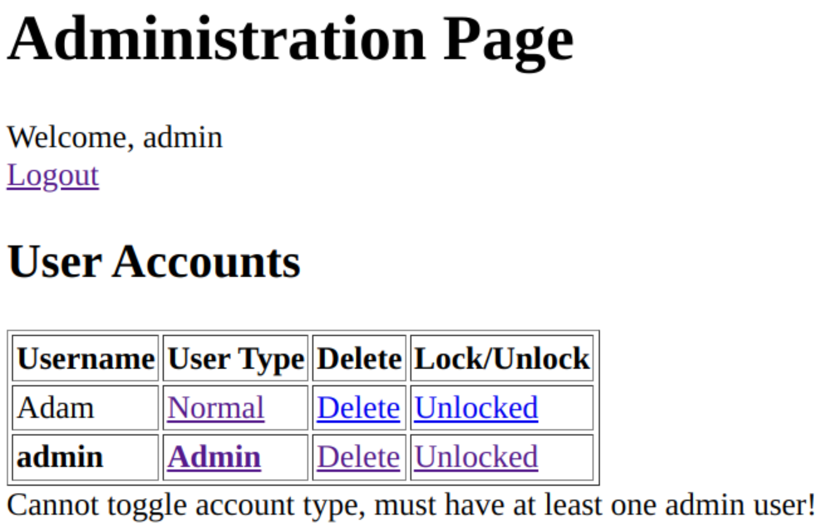
Welcome, admin
Logout

User Accounts

Username	User Type	Delete	Lock/Unlock
Adam	Normal	Delete	Unlocked
admin	**Admin**	Delete	Unlocked

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

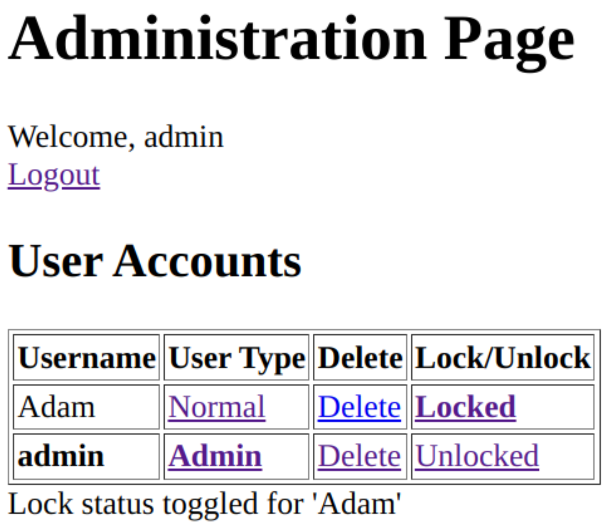
Welcome, admin
Logout

User Accounts

Username	User Type	Delete	Lock/Unlock
Adam	Normal	Delete	Unlocked
admin	**Admin**	Delete	Unlocked

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

Selecting the link in the **Lock/Unlock** column beside a user toggles the status of that user's account between "***Locked***" and "***Unlocked***". This option is used to stop a particuar account from being used to log into the system, e.g. clicked on "Unlocked" beside "Adam" gives:



Administration Page

Welcome, admin
Logout

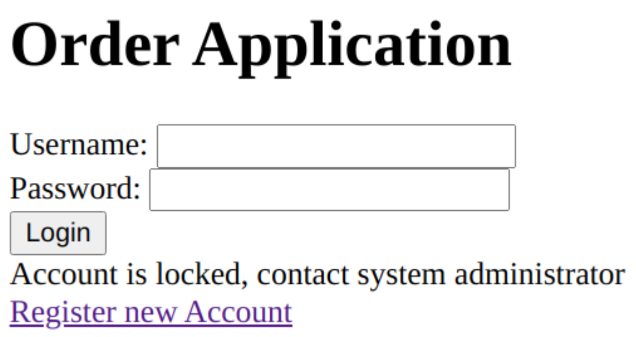
User Accounts

Username	User Type	Delete	Lock/Unlock
Adam	Normal	Delete	**Locked**
admin	**Admin**	Delete	Unlocked

Lock status toggled for 'Adam'

Note the message displayed under the table.

Attempting to log in using a locked account should disallow the login attempt and show the following message in the login page:


Order Application

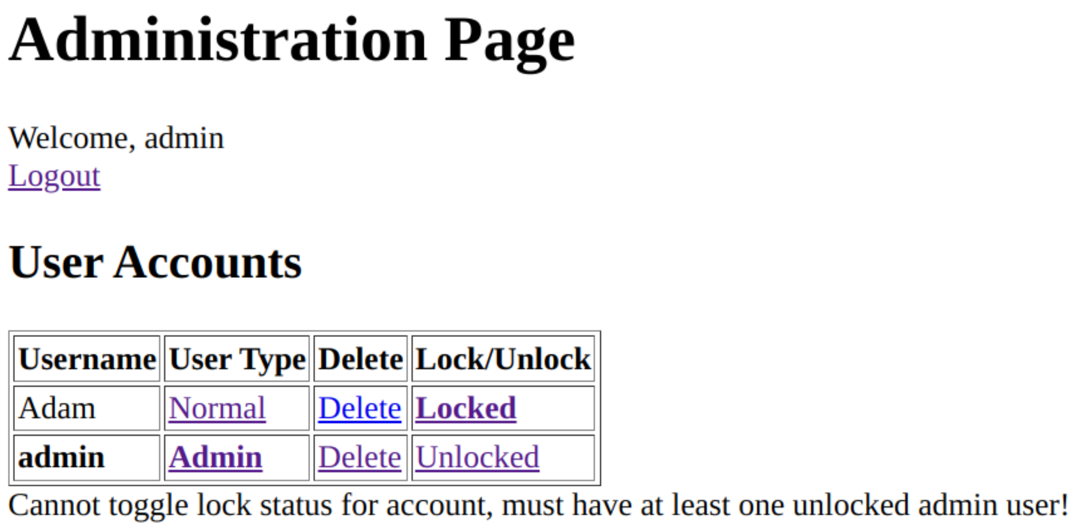
Username:

Password:

Account is locked, contact system administrator

Register new Account

If there is only one **Admin** user in the system then that user's account cannot be locked. For example, attempting to lock the "***admin***" account should give the following message:


Administration Page

Welcome, admin

User Accounts

Username	User Type	Delete	Lock/Unlock
Adam	Normal	Delete	**Locked**
admin	**Admin**	Delete	Unlocked

Cannot toggle lock status for account, must have at least one unlocked admin user!

APPLICATION REQUIREMENTS

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

- Use **CSS**[3], etc., to provide an attractive look-and-feel to users
- Be a Java web application
- Use a front-servlet design
- Use a connection pool called **jdbc/finalProjectFall2021** to get connections to the database
- Use only ***prepared statements*** to send queries to the database (***no Statements, CallableStatements or JPA***)
- All changes to data in the database must be done directly against the database, do not store data in memory except as described below
- Use the database as provided to you. You can change the data, but not the structure (I will be testing your solution form grading purposes using a copy of the database as it is given to you)
- *Include*[4] 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 (i.e. the controlling servlet should pass an ***ArrayList<User>*** to the JSP as a request attribute for display, not a comma-semicolon formatted string!)
- 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 username changing
    - **AdminServices**: account deletion, account type toggling, account lock status toggling
- It must implement and use a **DBoperations** class to contain the database-related operation methods

3 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.

4 Take a look back at the <%@include> tag as used in JSPs.