

## CPRG352 - WEB APPLICATION PROGRAMMING

FALL 2021

SAIT

**Due Date:** 9am, Monday 1 November, 2021.

**Submission:** Create a .zip file containing the entire NetBeans 12 project directory for your solution. Upload that .zip file to BrightSpace before the due date given above. Late assignments will be rejected by BrightSpace and cannot be accepted. Do not email assignments to your instructor, they will only be accepted via the process described here.

### **Assignment 2: Use of Servlets, JSPs and state management in a Java Web Application**

#### **Introduction**

In this second assignment you will create a Java web application called **Assign2**. This application represents a more fully-functional application that correctly uses servlets, JSPs, state management and appropriate data structures together.

When accessed the application will first present a login page to the user. If the user does not have an account with the application already then they can choose to navigate to another page to register a new account. Once the account is created they can then return to the login page and log in using that account.

When a user successfully logs in they are taken to a note management page. "Notes" in this case are just simple strings. Each note has an associated create date/time which will be displayed with it. Users can add and delete notes as required. They can also select a preferred page background colour in the notes page, which will be used for all pages in the application *including* the login page.

User account information and notes are stored in memory only for this application. The user account data will be stored in the *application*-level scope and each user's notes will be stored in the user's *session*. When a user logs out of the application their session is destroyed (invalidated) so all the notes they entered during that session will be lost. The user account data is only destroyed when the application server (Tomcat) is (re)started, or the web application is explicitly redeployed or stopped.

The preferred colour selected by the user will be stored in a *cookie*. This cookie value will be used to set the background colour for all the pages in the application.

## Sample Run of Application

### The Login Page

# Assign2 Login Page

Username:

Password:

[Register New User](#)

The login page is where a user will enter their login credentials to access the application. A user must have an account registered with the application consisting of a username and password. If a user does not have an account they can click on the **Register New User** hyperlink and go to the user registration page (more on that below). The login page can display three messages under the login form as necessary:

- **“Both username and password are required!”**: if the user submits incomplete data for login
- **“Invalid username or password!”**: if the user enters both a username and password, but the credentials do not match any registered user account
- **“Logged out”**: the user has returned to the login page because they logged out of the application

Here is an example of one of the messages:

# Assign2 Login Page

Username:

Password:

Both username and password are required!

[Register New User](#)

Registering a new User Account:

## Register New User

Enter username:   
Enter password:

[Back to Login Page](#)

On this page the user can enter a username and password for a new account. When submitted the system will check that there is not already an account registered with the same provided username. If there is then this page will display the following message:

## Register New User

Enter username:   
Enter password:   
  
Username is already taken, please select a different username  
[Back to Login Page](#)

If the account is created successfully then the following message is shown instead:

## Register New User

Enter username:   
Enter password:   
  
User 'bb' created successfully  
[Back to Login Page](#)

Note that in the above message “**bb**” is the name of the account that was just created (do not just hard-code the message to always say “**bb**” 😊).

Once the account is created the user can click on the **Back to Login Page** hyperlink and log in there using the account.

The Notes Page:

# Notes Page

Welcome, bb

## Add Note

Enter new note:

## Notes List

Date/Time	Note Text	Delete
-----------	-----------	--------

[Logout](#)

## Set Preferred Background Colour

☒ White ☐ Aqua ☐ Olive

The notes page is the main page of the application. At the top is a welcome message customized to show the username of the current user. Under that is a form the user can use to add a new note to the system. A table in the page will show any notes already added, and will also allow the user to delete existing notes. The **Logout** hyperlink will destroy the user's session and take them back to the login page.

The user can select a preferred page background colour. The change should be effective immediately and should be shown on all pages in the application, including the login page.

Adding a new Note:

# Notes Page

Welcome, bb

## Add Note

Enter new note:

New note added

## Notes List

Date/Time	Note Text	Delete
Fri Aug 28 13:15:54 MDT 2020	My first note	<a href="#">Delete</a>

[Logout](#)

## Set Preferred Background Colour

☒ White ☐ Aqua ☐ Olive

In the screenshot above the user entered “**My first note**” into the text field provided and clicked on the **Add** button. The system stored the note in the user’s session and returned to the page, which now shows it in the table. The date and time the note was created is also shown in the table, as well as a **Delete** hyperlink that the user can use to delete the note.

Under the **Add Note** form the message “**New note added**” is displayed.

The user adds a second note:

# Notes Page

Welcome, bb

## Add Note

Enter new note:

New note added

## Notes List

Date/Time	Note Text	Delete
Fri Aug 28 13:15:54 MDT 2020	My first note	<a href="#">Delete</a>
Fri Aug 28 13:18:52 MDT 2020	My second note	<a href="#">Delete</a>

[Logout](#)

## Set Preferred Background Colour

☒ White ☐ Aqua ☐ Olive

When the user adds a second note to the application both notes are stored and displayed.

User deletes a note:

# Notes Page

Welcome, bb

## Add Note

Enter new note:

Note deleted

## Notes List

Date/Time	Note Text	Delete
Fri Aug 28 13:18:52 MDT 2020	My second note	<a href="#">Delete</a>

[Logout](#)

## Set Preferred Background Colour

☒ White ☐ Aqua ☐ Olive

The user clicked on the **Delete** hyperlink beside the first note in the table so the application deleted it. The page is redrawn to show the remaining note(s). The message “**Note deleted**” is displayed under the **Add Note** form.

The user selects a preferred page colour:

## Notes Page

Welcome, bb

### Add Note

Enter new note:

Background colour set

### Notes List

Date/Time	Note Text	Delete
Fri Aug 28 13:18:52 MDT 2020	My second note	<a href="#">Delete</a>

[Logout](#)

### Set Preferred Background Colour

☒ White ☐ Aqua ☐ Olive

The user selected the **Aqua** option under **Set Preferred Background Colour** and clicked on the **Set Background Colour** button. The application created a *cookie* containing the preferred page colour information and refreshed the page using that colour. All other pages in the application must now also use this background colour. Note that the radio buttons show **White** as selected (as the default). That is fine, the radio button selection does not have to match the actual colour setting.

If the user clicks on the **Logout** hyperlink they will go back to the **Login** page, which will now appear as follows:

## Assign2 Login Page

Username:

Password:

Logged out

[Register New User](#)



## Application Requirements

Your solution for this assignment **must** adhere to the following:

- Be implemented in a **NetBeans 12.0/12.4** project and run on the **Tomcat 9** application server (i.e. the software we use in class)
- Be implemented as a “**Front Servlet**” application with any JSPs located under the WEB-INF directory in the project. The front servlet must be called **NotesNavigation**
- Be implemented using JSPs and servlets *appropriately*. Servlets must be used only as *controllers*, they must not themselves output anything to display to the user. The pages seen by users in a browser must all be implemented as JSPs
- Implement and use the following domain data classes:
  - **User**: a Java class that has attributes for **username** and **password** as Strings. Instances of this class must be used to store the data for a given user account when it is registered
  - **Note**: a Java class that has attributes for **note** (the note text entered by a user), and **dateTime** (the date and time the note was created by a user), also as Strings. The date/time attribute can be automatically set using a Java **Date** object)
- **User** account objects must be stored in an **ArrayList<User>** in the **application** scope of the application as they are created
- **Note** objects must be stored in an **ArrayList<Note>** in each individual user’s **session** scope
- Must use more than one servlet to perform its various operations, e.g. one servlet to handle account registration, user login and user logout. And another separate servlet to handle note creation and deletion.