CPRG352 - WEB APPLICATION PROGRAMMING

FALL 2021

SAIT

Due Date: 11:59pm, Friday 15th October, 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 1: Use of Servlets including output, input, navigation and basic logic

Introduction

For this first assignment you will create a Java web application called ***Assign1***. This application will consist of ***only servlets***, each one performing a particular task as outlined below. It is designed to help you demonstrate your learning in the design and creation of a simple Java web application using the techniques discussed in class.

The application will first show a login page, where a user can enter their username and password and be validated by the application. Depending upon who the user is ("**alice**" or "**bob**"), the user will then be taken to a different page in the application. If the user provided incomplete or invalid login data the application will take them back to the login page and display a suitable message.

Alice's page contains a *password complexity checker*, so that she can check that a password is acceptable according to given rules.

Bob's page contains a *palindrome checker*, wherein he can enter some text and the application will tell him whether or not that text is a palindrome (text that reads the same back-to-front as it does front-to-back, e.g. "***noon***", "***ABBA***", etc.).

Once logged into the application both Alice and Bob can log out again using a ***Logout*** hyperlink on each of their respective pages.

Sample Run of Application

Here we see a series of screen shots showing a sample run of the application for both Alice and Bob.

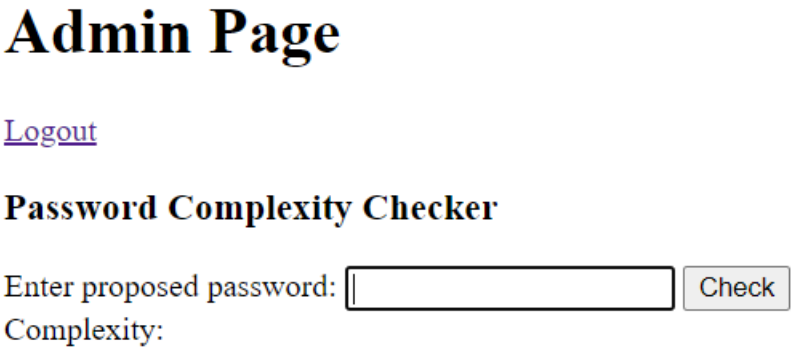
Alice's Sample Run

When Alice first accesses the application she sees the login page:

If she clicks on the Login button without entering data into both text fields, she will stay on the login page and see the following message:

If she types in incorrect login data, she will again stay on the login page, but this time she will see a different message:

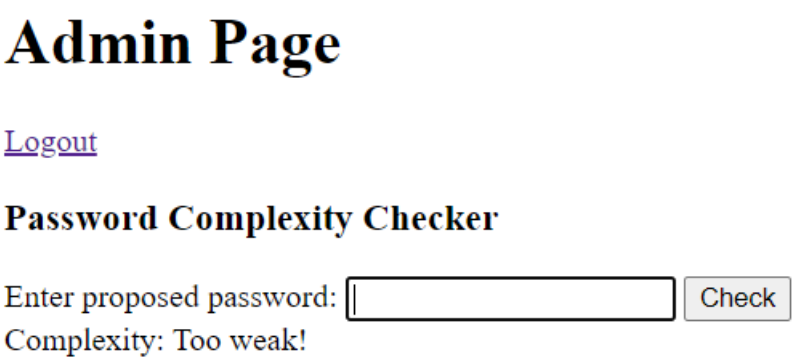
When she enters valid login data (**NOTE:** a username of "***alice***" and a password of "***pass***" should be accepted by the application) she should see a different page:



She can enter a password she might want to use into the text field provided and submit it to see if it follows the rules required for passwords in her organization. The rules are:

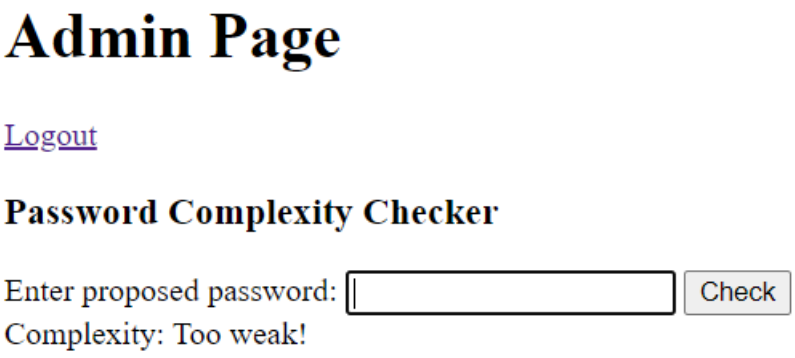
- Must be at least 10 characters long
- Must contain at least one upper case letter
- Must contain at least one lower case letter
- Must contain at least one numeric digit (0-9)
- Must contain at least one punctuation symbol from the ASCII character set, between positions 33 to 47 inclusive. (See the decimal values in www.asciitable.com if you are not sure which symbols are acceptable here)

*Example: she enters "**abc**"*



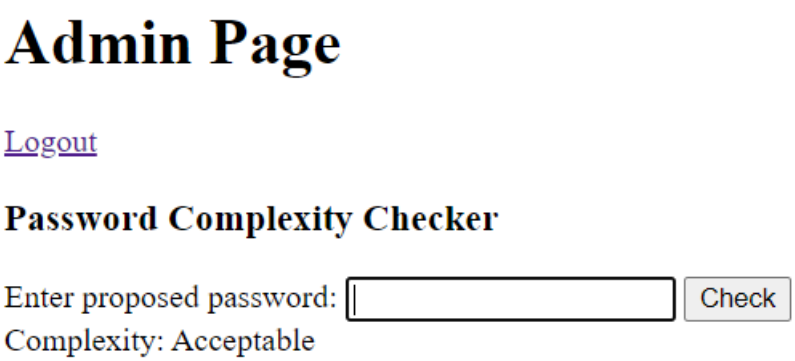
The message "***Too weak!***" is displayed under the form because "***abc***" does not satisfy the password rules.

*Example: she enters "**abcdefghij**"*



Again the message "***Too weak!***" appears because although the provided password was long enough, it fails to satisfy the other password rule requirements.

*Example: she enters "**123456789Aa!**"*



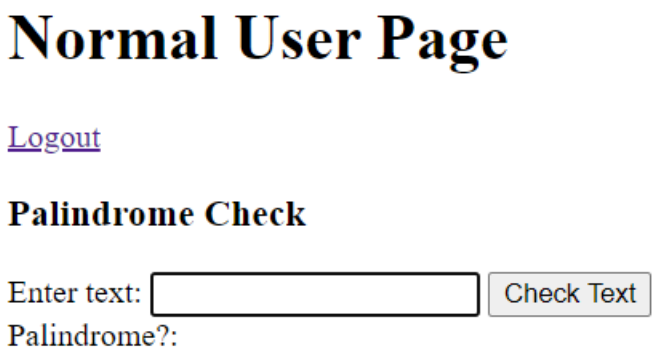
This time the message "***Acceptable***" is displayed because this password does satisfy all the rules.

When she is finished with the password checker Alice can log out of the application by clicking on the Logout hyperlink in the page, and she will be taken back to the login page again.

Bob's Sample Run

Bob logs in the same way as Alice (with a username of "***bob***" and a password of "***pass***"). The application takes him to a different page, however, where he can check if text that he enters is a palindrome, or not.

Example: Bob has logged in and is looking at the palindrome page



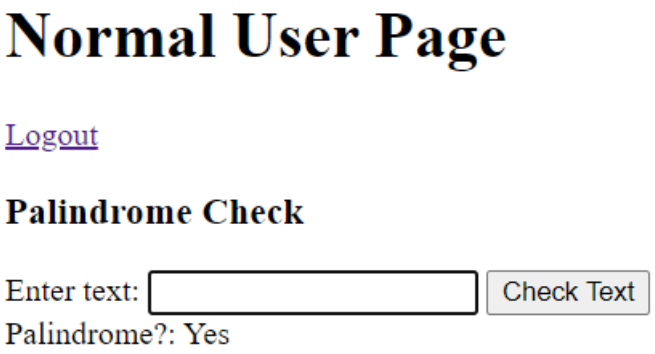
The rules for checking a piece of text to see if it is a palindrome, or not, are simple:

- Consider only alphabetic (A to Z) and numeric digit (0 to 9) characters in the text being checked
- Do not consider letter case (upper and lower cases don't matter!)
- Ignore spaces and any other punctuation symbols

Based upon these rules "***ABBA***" is a palindrome. So are "***AbBa***", and "***A bB a%***".

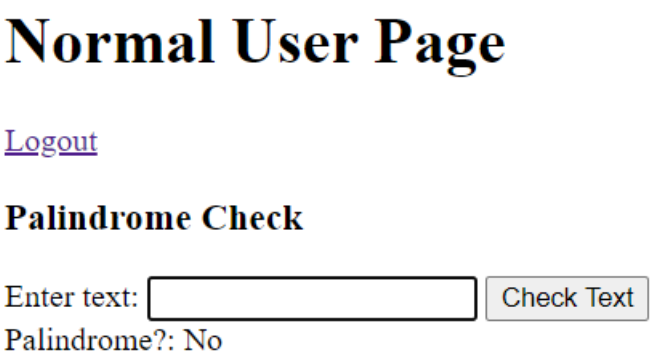
"**121**" is also a palindrome, as are "**1 21**" and "**12%%% %%1**".

*Example: Bob enters "**ABBA**" into the text field and submits*



The message "***Yes***" is displayed under the form becauses "***ABBA***" is a palindrome.

*Example: Bob enters "**ABC**" into the text field and submits*



This time the message "***No***" is displayed under the form because "***ABC***" is not a palindrome.

If Bob clicks on the ***Logout*** hyperlink he will be taken back to the login page, just as Alice was.

Application Requirements

Your solution for this assignment must adhere to the following:

- Be implemented in a **NetBeans 12** project and run on the **Tomcat 9** application server (the software we use in class)
- Can use **only servlets** for all its pages, no HTML pages are allowed

- Must implement and use the following servlets:
    - **Login**: displays the login page and submits the provided username and password to the Validate servlet. Also shows messages about failed login attempts, e.g. invalid login data
    - **Validate**: this servlet receives the username and password from the Login page and checks them to see if they are valid. If they are not then this servlet redirects the user back to the login page with an appropriate message. If they are then this servlet checks the username and redirects the username to the appropriate page (Admin or Normal User). This servlet does not itself display anything to the user (it is a *controller*)
    - **Admin**: This is the page Alice goes to, and it shows the form for, and processes, the password complexity checks
    - **NormalUser**: This is the page that Bob goes to. It shows the form for, and processes, the palindrome check functionality

For doing the password complexity and palindrome detection work you can add POJO (Plain Old Java Objects) to the application and call methods in them from the ***Admin*** and ***NormalUser*** servlets, rather than code that functionality into the servlets themselves (which gets messy fairly quickly and can even cause problems in a multi user application!). This is not a requirement, however, just a suggestion.