COMP 3095 – Java Web Application Development

Due Date: Friday, October 21st, 2016 (4:00 pm)
Team Size: 1 – 4 Team Members.

Problem statement:
Having recently graduated from George Brown College, you've recently been offered and accepted a position at a lucrative start-up company. The company consists of a number of experienced engineers and developers, who believe a new graduate, such as you, will bring fresh and innovative development ideas to the team. Your manager is a little uncertain of your abilities, but is open to giving the opportunity to prove yourself immediately, and with your very first job assignment.

Your manager is under pressure from the company Director to produce a website that achieves the functionality as described below. Your manager has been assigned the task of assembling a team to construct the initial phase of a website. The project is under a limited budget, and as such, the Director, Management team and Architects all agree, that to build the site, utilizing Tomcat (version 8), Java (version 8) and Eclipse would be the most cost effective strategy.

Your manager recalls during your interview that you mentioned you know Java, Servlets and JSPs (at a later time). Giving your knowledge of these technologies, your manager sees you as an ideal fit for the project. This task gives your manager the opportunity to evaluate your performance, while dually evaluating your worth into future tasks and assignments within the organization. Quite simply, this is your opportunity to prove yourself and set the stage for future growth within the company.

Your assignment has been scoped-out for you ahead of time by the senior engineers and architects. Wireframes (screenshots) have been constructed and provided. Your assignment must contain the following items described below. Don't let your manager or yourself down.

In your first assignment, we will be creating an online journal. The journal will allow its users to post questions and answers for all to view.

Registration

Like any online application, before any user can post and ask a question on the site, they first need to register to the site. In order to register, your website must create an account through an appropriate registration page.

The screenshot below is the wireframe of the registration page that your website must contain and that your website should match.

Figure 1: Registration Page

If the registration data supplied by the user is valid and all fields have been vetted successfully, your application is required to notify the user of the successful registration.

Figure 2: Example Successful Registration

It is expected that the user fills out all fields correctly. However, if the user submits the form with empty fields or with incorrect data, your web application should redisplay the form with the appropriate errors educating the user accordingly of the error or errors.

The wireframe below is an example of a registration page that has been submitted **incorrectly**.

Figure 3: Submitted Registration Page with Errors

The HTML content for this first assignment is dynamically created and is produced via a single servlet, RegisterServlet. Rephrased, the single RegisterServlet is qualified to generate the form with and without error messages.

The RegisterServlet should be accessible and mapped to the following path:

http://localhost:8080/comp3095/assignment1/RegisterServlet

Login

Once students can register, they require the ability to login. You website is required to have a login page that will authenticate a user based on a username / password pair.

When a user successfully completes registration, you should be creating the necessary object and storing the information so it is accessible from any servlet (for example) on your site.
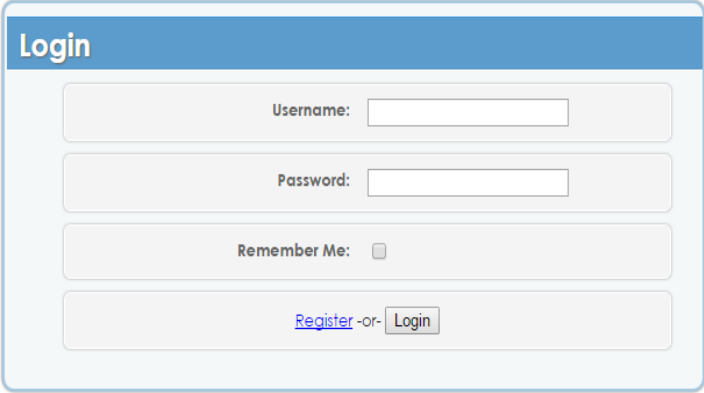
The user object should contain the following properties.

FirstName
LastName
Email
Phone
Year
Major
Username
Password

Login Servlet

The login page should match the wireframe provided below:

Figure 4: Login Page

The login servlet will prompt the user for the necessary credentials to login. It will contain a "Remember Me" check box, that will allow the user to remain "logged in" even after they have closed their browser window.

The page will allow the user to submit their credentials and thus login, or alternatively a link is provided that will allow the user to navigate to the registration page.

The LoginServlet will also be responsible for displaying error messages in the case of the following two (2) events:

- Invalid username and password
  - o Occurs when a User enters an invalid username/password combination.
- A User requests a page that requires a User to be logged in
  - o Occurs when a User enters the URL of a restricted page without first authenticating using the Login Servlet.
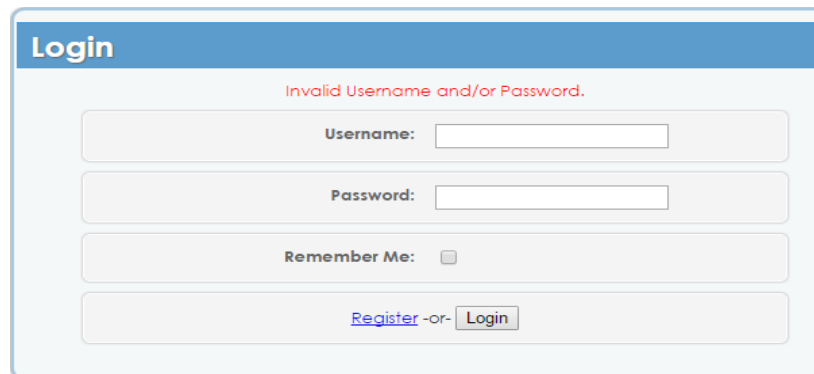
Authenticate Servlet

The Authentication Servlet is primarly used for validating a forwarded username and password against a List of Registered Users. This servlet will not generate any HTML. The servlet expects two parameters (username and password). If any manadatory parameters are missing, the servlet should redirect the Uer back to the Login Servlet without displaying any error messages.

If the User authenticates sucessfully and is found, the servlet should store the User's information in a Session object and redirect the User to the Welcome Servlet.

If the User fails authenticatoin, the servlet shoud redirect the User back to the LoginServlet with the appropriate error message, similar to the wireframe below:

Figure 5: Login Error

If the third (optional) parameter (rememberMe) is set to true, the servlet should implement some facility for remembering the User even after the User's browser has been closed.

Welcome Servlet

The wireframe for the Welcome Page is located below:

Figure 6: Welcome Page

The actual content of the page will not be generated until your second assignment, for now it should consist of a simple "Under Construction" image.

The Welcome Servlet should not be visible to Users who either have not have not authenticated or failed to authenticate. If a non-authenticated User attempts to access the Welcome Servlet, they should be redirected to the LoginServlet with an appropriate error message displayed.

Figure 7: Login Page Redirect with Error

You should notice the new "LoginStatus" bar located at the top of Welcome Servlet. It displays the message "Welcome, FIRSTNAME LASTNAME!", where the User's first and last name are replaced accordingly. Next to the welcome message is a link that, when clicked, sends the User to the login page and effectively logs the User out of your site. This "Logout" should occur in the Login Servlet. Note, not only should the current Session be terminated when the User logs out, but if the User had requested your site to "Remember Me", the facility that was implemented to remember them should also be removed.

Validation Requirements
- Both first and last names must contain only alphabets.
- Both e-mail address must match, and the value entered must match the pattern of an e-mail address.
- The telephone number must consist of 10 numerals. Decimals and non-numeric values are not permitted.
- A year and major must be selected.
- The username must only contain alphabets.
- The passwords must match.

Important Assignment Guidelines:

1. The project name and project navigation must be strictly followed as described above.
2. You must upload your assignment to your manager (Professor Santilli) via **blackboard**.
3. Within the body of the email, clarify course code, team name, team members and student numbers. Title the email accordingly COMP 3095 – Assignment 1.

    Example:

    Course: COMP 3095
    Team Name: The Hackers
    Team Members: John Smith - 1234567
                      Sally Jones - 7654321
                      Jane Wilson - 2342342

4. When uploading, cc' a copy to yourself for backup and time verification.
5. The uploaded compressed file must be in **.rar** or **.zip** format.
6. The contents of the compressed file must include:
    1. **.war** file of the project (with source code, no source code, no marks)
    2. **scripts** folder (only if applicable) – .sql scripts located inside.
7. The .zip/.rar file naming convention as follows:

    COMP3095_**YOUR_TEAM_NAME.**rar or COMP3095_**YOUR_TEAM_NAME.**zip

    Example: COMP3095_The_Hackers**.**zip

    *where **YOUR_TEAM_NAME** should be replaced with your team name in the company.

8. Each java file (.java) should include a header.

    ```
    //*************************************************************************
    *******************
    * Project:         < project name … >
    * Assignment:      < assignment # >
    * Author(s):       < author name …>
    * Student Number: < student number … >
    * Date:
    * Description:        <describe the java file and its purpose briefly only – 1 or 2 lines>
    *************************************************************************
    ***************//
    ```

9. Your code should be <u>modular</u> and should show no signs of dry (don't repeat yourself) code.
10. You are required to devise and use a form of persistent data storage.
11. The preferred data storage persistence is MySQL.
12. All required **.sql** files must be provided and included within the uploaded compressed file. There should be two .sql files, 1 for schema creation, table creation, seed data (if necessary) and another for schema deletion to remove the database completely from the system (original state).
13. Test your deployments.
14. Your manager should need only to deploy your application (possibly run some .sql scripts) and be able to use and evaluate your application immediately (navigating to the pages described). Your manager will be somewhat disappointed if otherwise.
15. Be cautious **DO NOT** share your application with others. Complete failures will be assigned if code is shared. All assignments will be reviewed and analyzed strictly within these regards.
16. Late assignments are assigned a penalty of 15% per day.

Marking Scheme:

Functionality	50%
Coding Style	25%
Design / Usability	25%

Good Luck!