CIS 1056 Register Lab

In this lab, you shall create a JSP page named register.jsp that allows a user to register (insert a record into the web_user table). To assist you with this, sample code shows you various stages of implementing a registration page that inserts a record into a customer table. Here is a description of the sample code:

- In project 1056_simpleRegister:
  - 01_insert_too_much_JSP_code.jsp: tries to put all the code needed for insert into a single JSP page. This page does use DbConn (otherwise, even the demos are too error prone with having to edit the connection string URLs all the time).
  - 02_insert_less_JSP_code.jsp: moves most of the code from the JSP page into a single large class called Insert.

Lab requirements (plus some recommendations):

Your web application shall contain a new jsp page called register.jsp. This page shall be modeled after 1056_simpleRegister.02_insert_less_JSP_code.jsp

Your code shall not be similar to 01_insert_too_much_JSP_code.jsp, because in this version, the JSP page has too many different responsibilities. This JSP code cannot be reused by other JSP pages (e.g., edit functionality should you decide to do the optional edit feature).

- Handling null values:
  - To get full credit for this lab, membership_fee and birthday shall be optional in the database (using mySqlWorkbench, unclick “Not Null”).
  - If the user enters “” empty string for membership_fee and/or birthday, your program shall accept this input and encode null into the prepared statement (for membership_fee and/or birthday).
  - As always, do regression testing to ensure that all of your pages continue to work each week. Check that the page that displays all the users does not “blow up” due to null values being the database. To prevent this, see how methods in dbUtils.FormatUtils() carefully check for null first and substitute the appropriate display (like “” empty string) for null values.
• User friendly Database Exception messages:
  o For database exceptions that have the potential to be thrown even when your code is bug free, the user error messages shall include the “full/technical” exception error message but they shall be preceded with a “user friendly” error message. For help with this, look at method “modelCustomer.dbMods.insert()” of the 1056_register sample code. Here are some examples of db error messages that need user friendly messages inserted before the full/technical error message:
    ▪ SQLState:23000, Error message: Duplicate entry 'r' for key 'email_address_UNIQUE' → User friendly message could be: Cannot register. A user is already registered with that email address.
    ▪ Communications link failure The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server. → User friendly message could be: The database is unavailable right now. Please try again later.

• In your web application, every “Log On” link, shall be expanded to two links: “Log On / Register”.

• Your web application shall meet all of the requirements from previous labs, and these help you prepare your project for later submission:
  o All HTML/CSS code is syntactically correct (checked by View Source from firefox).
  o JSP include statements are employed for UI code reuse.
  o Good naming (representative names, camelCase) is used for all variables, objects, methods, and classes. Only classes begin with a capital letter, the rest begin with a lower case letter.
  o Good class and method design using SRP (single responsibility principal). For example, instead of creating one method that does two different things, you create two methods that each do one thing.
  o Your project does not have any code that is not used. (Back up before you start cleaning up.)
  o All of your JSP pages still work locally (so all your java source files are still in the zip that you submit).
  o Your code works properly after publishing.
  o You have a blog in your labs page that tells what you did for this lab. The blog links to register.jsp.

• If you have not already done so, make sure that you have a blog in your labs page for your database lab. Upload the word document(s) that you submitted for the database lab and link to those word documents from this blog.

EXTRA CREDIT: You can get 3 points extra credit if you implement insert functionality for your second table. Call this jsp page just insert.jsp. Link to this page from your “other” list page (as well as from a blog in the labs page).