

CPRG352 - Web Application Programming

Fall 2021

Topic: JDBC data access

Problem 1: Retrieve data using varying sort orders

Create a project called **GetAndSort**. This project will open allows the user to retrieve and display data from a database (the *users* database we created in class) and display it in either ascending or descending order (where the order is determined by the SQL query run).

Example

When first accessed the application shows the following page:

List Users

Sort Order

Ascending 

Get Data

User List

The “**User List**” at the bottom of the page will show a list of the usernames in the **users** table of the **users1** database once the “**Get Data**” button has been clicked. The user can select options in the dropdown list to view the list in either *Ascending* or *Descending* sort order based upon the usernames.

If the user clicks on the button with the option set to *Ascending* they might see:

List Users

Sort Order

Ascending 

User List

alice
bob
charlie

If *Descending* had been selected in the list they would see:

List Users

Sort Order

Ascending 

User List

charlie
bob
alice

Application Structure:

The page should be displayed from a file called **index.jsp**. This does not have to be under WEB-INF.

The form on the page with the button should submit to a servlet called **DBcontroller**. The servlet should not open any database connections itself, but instead should use an object of type **DatabaseOps** to retrieve the list of usernames to show in **index.jsp**.

Problem 2: Manage notes in a database

Create a project called **notesLab**. This application will allow the user to add and delete notes (strings) from the **notes** table in the **notes** database (you are being given a SQL script called *notesDBscript.sql* to use to create this database).

When first accessed (with an empty **notes** table) the user will see:

Manage Notes

Add Note

Enter note:

Note List

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

The user can enter a new note in the text box provided. When the **Add** button is clicked upon the form submits the note to a servlet called **DBcontroller**, which itself uses an object of type **DBoperations** to add the new note to the database (and also supports database deletions). The user is then forwarded back to the page which will display the note.

For example, if the user enters “**first note**” and clicks on **Add** they should see:

Manage Notes

Add Note

Enter note:

Add

Note List

Note Text	Date/Time	Delete
first note	2020-08-24 15:36:40	Delete

The “***Date/Time***” column is automatically populated by the database with the computers date/time when a new row is added to the **notes** table (you do not have to provide data for this column).

The “**Delete**” link to the right of the table allows the user to delete the note it is beside.

Application Structure:

This application should use a “*front servlet*” approach (content under **WEB-INF**).

Database access should be done using Java **Statement** objects only.

The table in the JSP should be generated using **JSTL** tags only.

Problem 3: Revised problem 2 using PreparedStatement

Make a copy of your solution for the previous project called **notesLabPS** and revise it to use **PreparedStatement** objects instead of **Statements** to access and manipulate the database.