Task 1: Setup

1. Start by creating a new *Web Project* in Netbeans
2. Right-click on Libraries >> *Add Library* >> *MySQL JDBC Driver*
3. Go to the `handouts` folder on N: drive. Locate folder `Servlets_and_JSP_Examples/LargeInClassJSPExercise_Incomplete`
   a. Copy contents of the `web` subfolder (<mark>THE CONTENTS NOT THE FOLDER ITSELF</mark>) to your project's `Web Pages` folder
   b. Copy contents of the `src` subfolder folder (<mark>THE CONTENTS NOT THE FOLDER ITSELF</mark>) to your project's `Source Packages` folder
4. Go over classes `User.java`, `UserController.java` and `UserDBLibrary.java` and figure out what they do. Ask questions if there is anything that you don't understand.

Task 2: Completing pages index.jsp & Login_action.jsp

Page `index.jsp` submits the *username* and *password* to page `Login_action.jsp`. Page `Login_acion.jsp` should then create an instance of the `UserConroller` class and call method `login` on the created instance using the submitted *username* and *password*. If the return value is anything other than <mark>0</mark> (i.e., success), the user should be sent back to `index.jsp` and an appropriate error message should be displayed there describing the error: <mark>−1</mark> means that the provided *username* is not valid, <mark>−2</mark> means that the provided *password* is not valid, and <mark>−3</mark> means that there was an error while attempting to access the database.

If the `login` method returns <mark>0</mark> (i.e., a successful login), then we need to create a session variable to store the created `UserConroller` object so that it is accessible on any page during the user's session (remember that the Web is stateless; once we leave a Webpage and go to another one, nothing can be recalled from the previous Webpage). Session variables help us make certain variables persist during a session. To do that, we can use a statement like below:

```
session.setAttribute(String nameForYourSessionVariable, Object valueForYourSessionVariable);
```

Any Webpage where the session variable is needed can now access it using the following (assuming that your session variable contains a variable of type `UserConroller`)

```
UserConroller uc = (UserConroller) session.getAttribute(String nameForYourSessionVariable);
```

After creating the session variable, forward the user to page `Menu.jsp`.

Task 3: Completing page Menu.jsp

When complete, page `Menu.jsp` will display something like the figure below which starts with a greeting for the logged-in user ("*Hello User imad*" in my case) followed by a table.

The table's first row (which spans 8 columns) contains a hyperlink to page `Add.jsp` which allows the user to add a new user. The second row contains 8 cells, the first and last of which are empty. The rest of the cells contain headers for the corresponding columns.

Hello User imad

	Full name	Username	Password	Age	City	State	
Edit	Adam Andersson	adam	andersson	19	Collegeville	MN	Delete
Edit	Alyssa Anderson	alyssa	anderson	19	St. Joseph	MN	Delete
Edit	Anders Langley	anders	langley	19	Collegeville	MN	Delete
Edit	Andrew Mikos	andrewm	mikos	19	Collegeville	MN	Delete
Edit	Andrew Zurn	andrewz	zurn	19	Collegeville	MN	Delete
Edit	Benjamin Maakestad	benjamin	maakestad	19	Collegeville	MN	Delete
Edit	Bernard Ferguson	bernard	ferguson	19	Collegeville	MN	Delete
Edit	Imad Rahal	imad	rahal	33	St. Joseph	MN	Delete
Edit							Delete

The rest of the rows contain user information from the database with one row for every user. The first cell contains a form with two fields: an *edit submit button* which submits to page `Edit.jsp` and a *hidden field* which contains the username of the corresponding user. The reason for the hidden field is that the target page `Edit.jsp` needs to know which user was clicked in order to edit the correct one. The last cell is pretty similar but its delete button should submit to `Delete.jsp` instead. The remainder of the cells contains the user information.

Page `Menu.jsp` should use the `UserConroller` session variable created in page Login_action.jsp. This will allow the page to get the username of the logged-in user and call method `getAllUsers()` in `UserConroller` in order to loop over all the user records and create rows for them.