Q1)

README File:

This application will allow a user to register, login, and see some select research previously done by the authors of the application.

A non-authorized user trying to access a .jsp or .do endpoint will be redirected to the login page where they can either register or present their credentials.

In the event of an error, an error page will be displayed that will inform the user that an error occurred and present a link to the login page. An easy way to invoke an error is to attempt to access a random URL and get a 404 Not Found.

The application uses log4j2 to log some informational and debugging statements to the console. All logging is appended to a file located at C:\temp\hw3q1.log. This can be modified in the log4j2.xml file located in resources. This will likely error on a Unix based computer as it logs to C:\

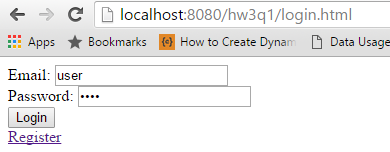
SETUP:

There is a sql script in the db folder that needs to be run to initialize the database. This creates the database and creates the users table and inserts some initial users.

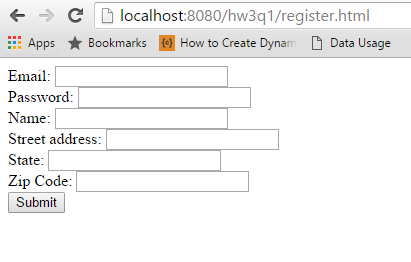
The web.xml file needs to be edited to include the credentials used to access the locally hosted database.

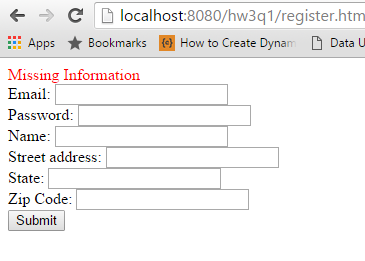
Tested on Tomcat7 & Tomcat8, MySQL 5.7.10, mysql-connector 5.1.38 (included), log4j2 2.5 (included), Windows 10, Windows 8, Java8, Google Chrome 49.0.2623.87, Firefox 45.0.1

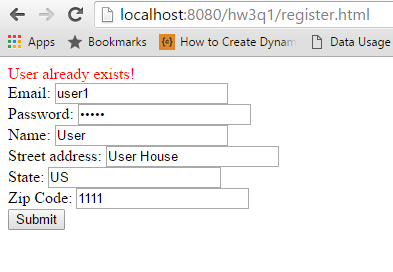
Screenshots of it working:



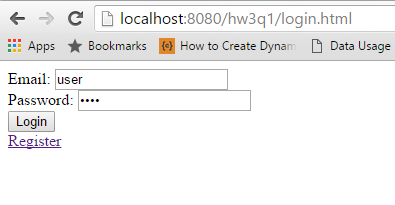
Registration requires javascript.



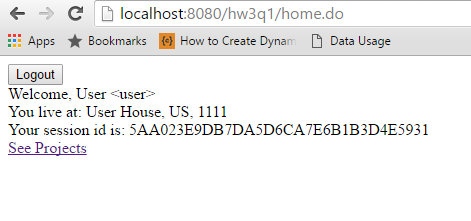




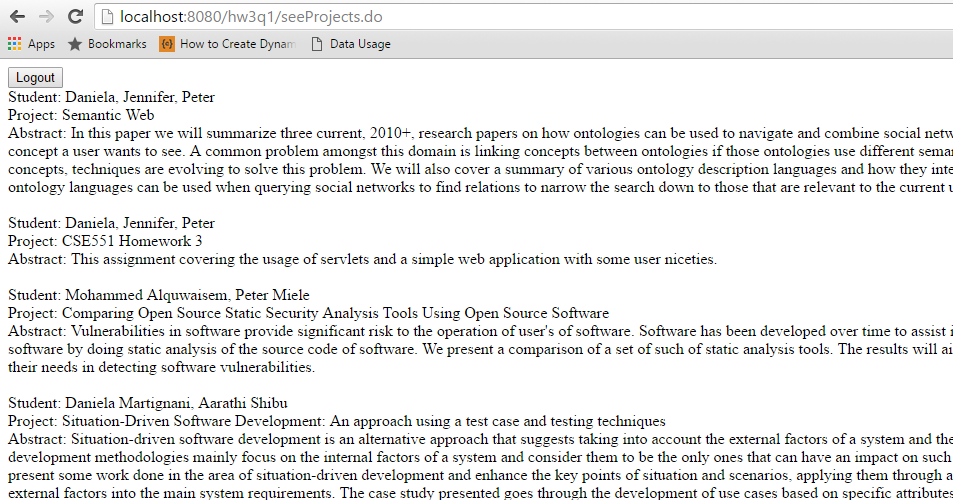
On success a user is redirected to the login page where they can log in to the application.



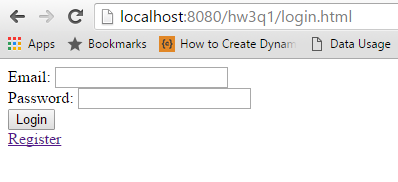
The user is presented with a home screen that prints out all the information about their account and the session id.



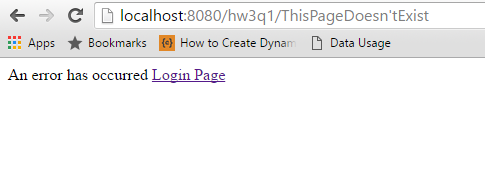
If a user clicks on See Projects they will see the research efforts of the software’s authors.



The user can click logout to invalidate their session and return to the login page



If an error occurs the user gets an error page with a useful link to the login page.



Q2)

The session in this application is used to store the selected items of the user even between multiple requests. This is similar to an online shopping cart experience in which the user can add items to the cart and then go find additional items to add to the cart.

* With two tabs of Chrome with cookies enabled
  + The session was shared between both tabs. Having tab 1 select the first book and tab two select the second book ended up with both tabs having both books selected
* With two tabs of Chrome with cookies disabled
  + The session was not shared between tabs, each tab could make selections independently of the other. When clicking the link the session information was stored in the URI. Session was lost if manual URI entering occurred that modified or removed the Session ID.
* One tab Chrome one tab Microsoft Edge
  + The session regardless of cookie state was not shared between browsers. Each one could work independently.

On a side note, with the introduction of Windows 10 Internet Explorer is no longer offered by Microsoft in lieu of their new Web Presenter Microsoft Edge™.