JSF Application and Grid

Daniel R. Cender

Grand Canyon University: CST-235-O500

July 19, 2020

**All Source Code Retrievable At:** [**https://github.com/DanielCender/CST-235**](https://github.com/DanielCender/CST-235)

**Explanation**

The applications built in this assignment build on the assignment code from last week. Validation is added to the Java Managed Beans and JSF forms to enforce input data restrictions. I replaced the bare XHTML with use of a general layout, comprised of smaller ui:composition JSF view components. Now, every page has the same header/footer layout. Furthermore, the use and styling flexibility of JSF data grids was explored on the TestResponse.xhtml page.

Below is a simple breakdown of all the classes/views in the assignment.

**Classes**

assignment3

* + User.java (Managed Bean)
    - Methods: constructor, overloaded constructor, getters/setters
  + Order.java (POJO)
    - Methods: constructor, getters/setters
  + Orders.java (Managed Bean)
    - Methods: constructor, getters/setters
  + FormController.java (Managed Bean)
    - Methods: onSubmit, onFlash

**Pages**

assignment3

* + TestForm.xhtml
  + TestResponse.xhtml
  + TestResponse2.xhtml

**assignment3**

Adding onto the last assignment, Java Bean property validators are added to enforce restrictions on user input. The first screenshot below shows the view result when a string is entered that is too short for the validation. The second picture shows the correct redirect result from entering valid-length inputs.

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

I next updated the JSF application to use predefined and styled layouts for structuring the form and response pages. After creating a layout file and JSF fragments, I restarted the JBoss server and ran the TestForm.xhtml page again. Then, the form and response pages contained the centered header and footer.

A screenshot of a computer screen

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Per the assignment instructions, I wrote a basic CSS stylesheet for the login and response pages, then imported it into the commonLayout.xhtml template. The new header and footer styles, along with an additional “back” button on the response pages, are shown in screenshots below.

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

The last requirement for this assignment involved implementing an Orders Managed Bean and a data grid on the TestResponse.xhtml view displaying a list of Order items. I created a few custom CSS classes to slightly style the grid, which can be seen in the picture below.

A screenshot of a computer screen

Description automatically generated