COMP 31, Week 5

Lab Assignment

Goal

The goal of this lab assignment is to bring together all of the work done so far in this course into a Spring application. The application will allow employees to login and to view their department web pages. For this lab, all but one of the pages will be nearly empty (except for a heading identifying the department.) For the **admin** page there will be a link to a page with an "Add New Employee" form**.**

Step 1 – Download Starter Project and Initialize Git Repo

- Make a fresh copy of the **Week 5 Lab Starter** from Blackboard (**Modules | Introduction to Spring MVC | Week 5**). (I.e. the starting point for this lab is the one posted on Blackboard, not the version with the registration form that was added in Tuesday's lab.)
- Test project by logging in as various registered and non-registered users.
- Open the Source Control Window (Ctrl+Shift+G) and click **Initialize Repository.**
- **Stage all Changes** and **Commit** with message **Initial Commit – Lab 5 Starter.**

Step 2 – Rename Person to Employee

- Open **Person.java**, right click on **Person** class name and select **Rename**. Change the class name from **Person** to **Employee** and press the Enter key. This will change all references to the Person class in the entire project from Person to Employee. It will also rename file namePerson.java to Employee.java and change all import statements from **model.person** to **model.employee**.
- From the **Edit** menu select **Find in Files** and search for all occurrences of the word **person** (or **Person**) remaining in the project. These will include parameter types and names, local variables, attribute names (in java files) and Thymeleaf variables (in html files). Rename all occurrences to employee (or Employee).
- Rename the **role** field to **department** in the **Employee** class.

- Test project by logging in as various registered and non-registered users.
- **Stage all Changes** and **Commit** with message **Create Employee class.**

Step 3 – Create New Users and Template Pages

- Add code to the **loginService** constructor to hardcode an initial set of users. Use the same userIds and departments as in Weeks 2 & 3. I.e.

UserId	Department
sam	sales
sally	sales
ollie	orders
olivia	orders
rachel	repairs

ralph	repairs
abbie	admin
arthur	admin

- Make up first and last names for each employee. For testing purposes set the password to be the same as the userId.
- Create template pages for each department. As in the **Solution to Lab 2** (posted on BB) , add the department pages to a **departments** folder inside the **templates** folder.
- Each page should display a heading that identifies the department.

Step 4 – Modify Login Validation Code

- In the **MainController**, modify the code in the **@PostMapping ("/login")** handler to validate userIds and passwords.
- If the user and password are correct, then they are routed to their department's page.
- In the case of **admin** logins, add an Employee backing object to a Model before going to the amind page.
- Test the changes by logging in as various employees to make sure they are sent to the correct department page.
- **Stage all Changes** and **Commit** with message **Add employees and department pages.**

Step 5 – Add 'Create Employee' Form

- On the **admin** web page, add a link that sends a get request the endpoint "/add-employee". In the MainController, add a handler a that navigates to a **new-employee-form** page. This page should display a form for creating new employees and should be based on the registration form (presented in the Week 5 Tuesday lab).
- When the submit button is clicked, the Employee data is sent to the server which sends it on to the **addEmployee** function in the loginService.
- Test the form by adding an Employee and then logging in under the new employee's userId.
- **Stage all Changes** and **Commit** with message **Add 'Create Employee' form.**

Hand In

At the end of the lab, zip your entire project folder and submit it to Blackboard. Hand in your work even if you haven't completed all the steps.

Looking Ahead