Day 14 Social Media II notes

Show solutions to homework

1 In each of our html pages, we've included several css and js files. Break this out into a fragment and then include the fragment on each page.

Create head.html in templates/fragements, including <head> tags

Add the following to each html page

<head th:include="fragments/head">

<title id="title">Feed</title>

</head>

2 Add a sign up button to the login page so that it's easier to navigate. This will need to be a mini

form that just makes a get request

<form th:action="@{/?????}" method="get">

<button class="btn btn-md btn-primary btn-block" type="Submit">Sign up</button>

</form>

3 In our AuthorizationController, we sent back a success message in the model if the registration was successful. We haven't done anything with it yet. On the registration page, if a success message comes back, hide the form, display the success message, and display a button to navigate to the login page. If a user is filling out form still (no success message included in the model), then the page should look the same as it does now.

In registration.html

<div th:if=*"!${success}"*>

// wrap this div around whatever you want to hide after a successful registration

<div>

<div th:if="${success}">

<h2><span class="text-success" th:utext="${success}"></span></h2>

<br/>

<form th:action="@{/??????}" method="get">

<button class="btn btn-md btn-primary btn-block" type="Submit">Go to login page</button>

</form>

</div>

4. On the main feed page, display a message saying no tweets are available when that is the case, rather than showing an entirely blank page like we are right now.

<div th:if="${#lists.isEmpty(tweetList)}">

</div>

5 On the main feed page, improve the styling of the tweets. Check out the bootstrap list-group and list-group-item classes and use them to display each tweet inside of a box.

**Run the app, have students add users and tweets**

**Walk through the slides with the students adding the new code.**

Homework 2

1 (DUPICATE, IGNORE). Let's think back to day 1. In our AuthorizationController, we sent back a success message in the model if the registration was successful. We haven't done anything with it yet. On the registration page, if a success message comes back, hide the form, display the success message, and display a button to navigate to the login page. If a user is filling out form still (no success message included in the model), then the page should look the same as it does now.

2. Add username links to the main feed, just like we did in the user and hashtag pages.

3. On the main feed, we currently show all tweets in the system. Change it so that it only shows tweets for the accounts the user is following.

a) You need to modify the routing of the users from /tweets to /tweets?filter=following

b) Need to update Tweet controller, replace getFeed with the following

@GetMapping(value= {"/tweets", "/"})

**public** String getFeed(@RequestParam(value="filter", required=**false**) String filter, Model model){

User loggedInUser = userService.getLoggedInUser();

List<Tweet> tweets = **new** ArrayList<>();

**if** (filter == **null**) {

filter = "all";

}

**if** (filter.equalsIgnoreCase("following")) {

List<User> following = loggedInUser.getFollowing();

tweets = tweetService.findAllByUsers(following);

model.addAttribute("filter", "following");

}

**else** {

tweets = tweetService.findAll();

model.addAttribute("filter", "all");

}

model.addAttribute("tweetList", tweets);

**return** "feed";

}

4. On the users page, display the text '(Me)' next to the username of the currently logged in user

Need to research to solve