Assignment 4

Securing your Application

Points: 100

Due 04/17/16 EOD

This assignment is to be done by a team of two students. Your first step of the assignment is to ensure that you have documented your team membership to course staff in the manner your instructor has specified. There is a fair amount of work to be done, but there are two people working the assignment together. You should establish a sensible way to share your project development work. You **MUST NOT** make your assignment code available publicly; as this would have to be treated as a violation of UNCC academic integrity policy. So do not use a repository service such as GitHub where all projects are publicly viewable – any such repository must be private. You may consider using the UNCC Git repository server (https://cci-git.uncc.edu).

This assignment is intended to familiarize you with the different techniques used to secure your web application against known attacks (Cross-site Scripting, SQL injection, and social engineering). In addition, it requires you to add new features into your project such as sending emails and Facebook share button. This assignment is an extension to the previous assignment. To be able to meet the requirements for this step successfully you are encouraged to resolve any issues or missing requirements from Assignment 2. To start, you should coordinate with your team member(s) on which submission your will use as a starting point for this solution.

Assignment Description

In this assignment, you will add new features and security countermeasures which entail some modifications to your previous JSP/Servlet MVC web application, according to the following specifications:

1. Correct any errors identified or partial/missing functionality from the previous assignment.
2. All structure, design, and content requirements from previous assignments are mandatory, unless explicitly updated in this assignment description.
3. Use JavaBeans to implement the business layer of the application (**model**).
4. Use JSP pages to present the **view** to the browser.
5. Use Servlet pages to **control** the flow of the application.
6. Functionality that does not follow the assignment specifications will not receive credit.

Adding A new Feature:

Add activate the account feature to your existing project (assignment 3):

1. Update the UserController servlet –the part when the action is “create”
    a. The user is on the signup.jsp page, fills in the entire fields and then hits the “Create Account” button.
    b. The UserController servlet is called with an action value “create”
        i. UserController servlet checks the http request for parameters: name, email, password, and confirm password.
        ii. Validates the above information for possible errors.
            1. If there is any error:
                a. Adds an error message to the http request object, call the parameter “msg”.
                b. Adds the above information to the http request.
                c. Dispatches to the signup.jsp page.
            2. If there is no errors
                a. The UserController servlet generates a unique token.
                b. Adds the **user record and the token** to the DB (TempUser Table (New)).
                c. Sends an activation email to the supplied email address with an activation link that includes the unique token.
                d. Dispatches to the login.jsp page.

2. The user opens her email and click the activation link, the UserController servlet is called with an action value of “create” (Update the UserController servlet by adding a new action value)
    i. If there is an action parameter, validates that its value is either “login”, “create”, “how”, “about”, “home” , “main” or “activate”
    ii. If action is “activate”
        1. Checks the http request for the token parameter.
        2. Validates the token in the TempUser table.
            a. If there is any error:
                i. Adds an error message to the http request object, call the parameter “msg”.
                ii. Dispatches to the signup.jsp page.
            b. If there is no errors:
                i. Move the user record from the TempUser table to User tabe. (delete the record from the TempUser table).
                ii. Creates a User bean for the user.
                iii. Adds the User bean to the current session as “theUser”.
                iv. Dispatches to the main.jsp page.

For generating the token you may consider using the Java’s UUID.randomUUID()method.

Database Creation

Add a new table to your MySQL database.

```
Create Table TempUser(
UName VARCHAR(40),
Email VARCHAR(50),
Password VARCHAR(50),
IssueDate datetime,
Token VARCHAR(50));
```

- Create a new TempUserDB class to add/retrieve data from the TempUser table. Here are some methods that you may suggested methods:
  - User getUser(String Token)
  - addTempUser(String UName, String pass, String Email, Date date, String token)

Applying Security Measures:

The Murach book talks about three main attacks against web applications. The book also describes the possible countermeasures for each of these attacks. The following table summarizes the attacks, their countermeasures, and the chapters where these topics are been discussed. Feel free to use any of the utility classes that are provided in the book (e.g. a utility class for hashing passwords). Apply the changes on all pages, classes, and methods that have the vulnerability. For instance, for the SQL injection, use prepared statements in all the utility classes that accept input from the user. You need to study the four chapters very well to gain depth understanding of these attacks and their countermeasures. After that, you need to define the places in your project that are vulnerable to any of these attacks, then implement the necessary courntermeasure.

Attack	**Prevention**	**Chapter**
Cross-site Scripting	JSTL out tag	Chapter 9
SQL injection	Prepared Statements	Chapter 12 & Chapter 13
Social engineering	Hash and Salt Password	Chapter 17

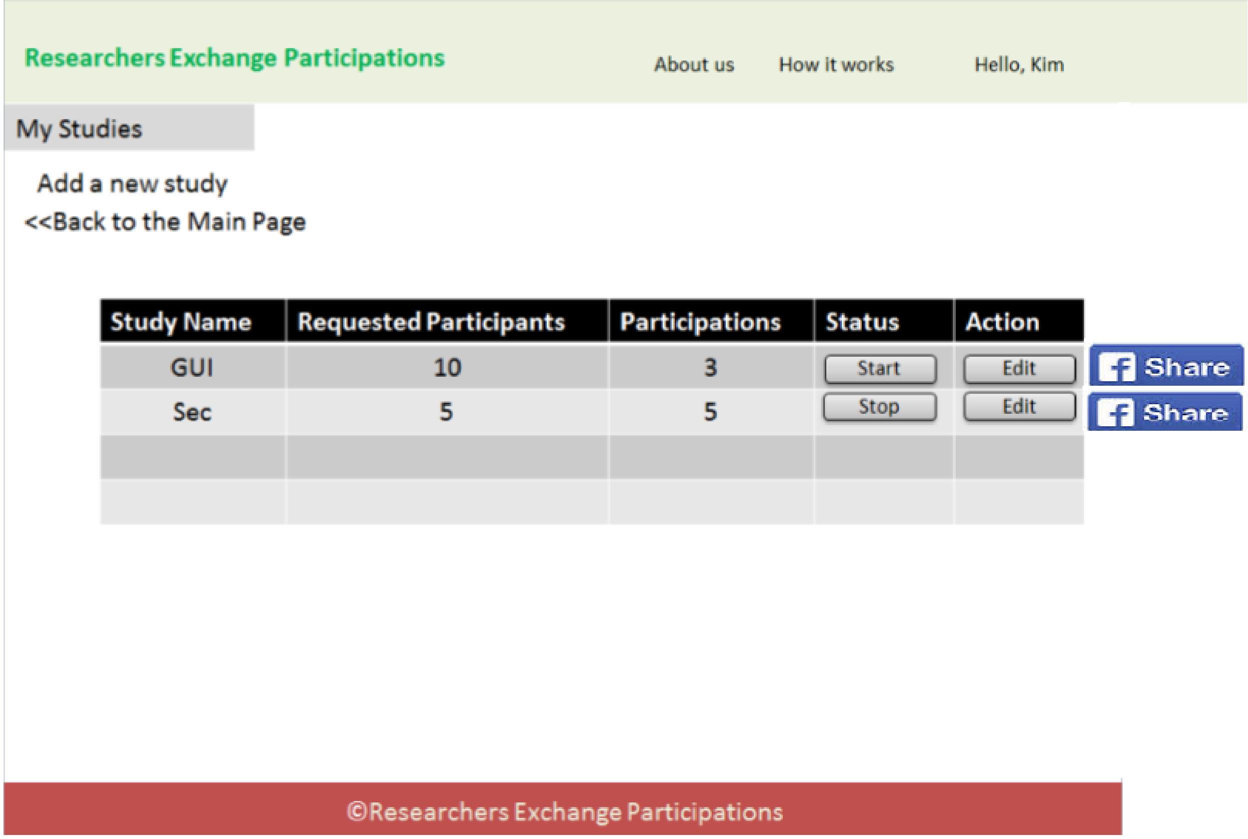
HTML5 Validation/Cross-browser testing:

Validate your HTML5 on http://html5.validator.nu/. Also, make sure your app works (and looks right) in both Google chrome and Firefox. Check the document in the moodle page on how to conduct the HTML5 validation (Resource Links >> How to? HTML 5 Validation.

<u>(The maximum number of credits that you can obtain is 20, you can use these extra credits in the assignment category only)</u>

Extra Credit (20 points):

You will receive up to 20 extra points on this assignment if you implement any social media sharing button on the studies.jsp page, the user must be able to share her completed studies results on a given social media website (e.g. Facebook, Twitter, Flickr, etc.).



Study Name	Requested Participants	Participations	Status	Action
GUI	10	3	Start	Edit
Sec	5	5	Stop	Edit

Extra Credit (10 points):

You will receive up to 10 extra points on this assignment if you implement forgot password feature. You need to follow the steps mentioned in this link (answer by clement): http://stackoverflow.com/questions/27313352/implementing-forgot-password-functionality-in-java

Extra Credit (10 points)

You will receive up to 10 extra points on this assignment if you implement the functionality "The user receives 2 coins when his friend signs up", see the recommend jsp page. The logic here should be similar to the forgot password feature.

Assignment Submissions

What to submit using Moodle (Email submissions will NOT be accepted):
1**. firstname_assignment4.war** - An archive of the entire web application (project) stored in a standard WAR file. **You must ensure that the java source files are included as part of the archive**. The WAR file will be imported into Netbeans for grading and may be required to be deployed as part of the submission (Check the help document in the Resource links >> How to create a war file in netbeans).

2. **info.pdf** – PDF document with the following assignment information :
a) Explanation of status and stopping point, if incomplete.
b) Explanation of additional features, if any**.** **New**: add a table that has three columns, the first column shows the counter measure, the second column shows the file name where that measure is implemented, and a comment column. Describe the changes or the additions that you made on the controllers, model, and view layers to implement the extra features.
c) Discuss the easy and challenging parts of the assignment. How did you overcome all or some of the challenges?
d) Discuss division of labor specifying who did what and why this is a fair and equal split.

3. **Openshift link**. Upload your project into openshift and post the openshift link in the online text session (so we have easy access when grading).

Finally, set up a time to demo your assignment to your grader using the process described in the course Moodle site. Students should be prepared to answer questions posed by the grader about their work. Failing to demonstrate the assignment to the grader will result in no credit for all team members.