

UNNOBA

Programación Orientada a Objetos

2018

Práctica 3: Webapp - MVC - Servlet - HTML-Javascript - CSS

Presentación y objetivos

Se la implementación de la funcionalidad de login para la aplicación web de objetivos de desarrollo sustentable utilizando html, javascript, css y Java Servlet. El objetivo es brindar una introducción práctica al desarrollo de web apps utilizando las herramientas más elementales a efectos de comprender el propósito de cada capa y interacción entre las mismas, al igual que introducir el protocolo HTTP.

Materiales y recursos

Para el desarrollo de la actividad deberán utilizarse los siguientes materiales y recursos:

- Clases 3 y 4.
- Foro de actividades prácticas.
- Bibliografía disponible en biblioteca:
 - An introduction to object-oriented programming.
 - Análisis y diseño orientado a objetos con aplicaciones.
 - Fundamentos del diseño y la programación orientada a objetos.
 - Head first object-oriented analysis and design.
 - The java programming language.
- Software de desarrollo:
 - Netbeans o Eclipse.

Actividades

1. En el archivo index.jsp, crear un formulario html que permita ingresar el email y password de un usuario, además debe tener un botón “Ingresar” para que el usuario pueda enviar los datos para que los mismo sean validados. Los datos del formulario deben ser enviados por el método de post de http a la url <http://localhost:8008/ods/login>.
2. Crear dentro de la carpeta webapp/resources dos carpetas una llamada css que contendrá los archivos de estilos y una carpeta js que contendrá los archivos de script de la aplicación.
3. Dentro de la carpeta webapp/resources/css crear el archivo style.css, el mismo debe especificar los estilos para la web, en particular deberá dar estilo al formulario de la funcionalidad de login y a la página html que lo contiene.

4. Dentro de la carpeta webapp/resources/js crear el archivo script.js, el mismo debe especificar algunas funciones de javascript necesarias para la implementación de login. Específicamente se pide crear la función validar_login que reciba un formulario html con los campos email y password. La función debe retornar el valor booleano true si email y password no están vacío y además el campo email tiene un formato de email válido y password tiene una longitud mínima de 8 caracteres y está compuesto por letras y números. En caso contrario retorna el valor booleano false.
5. Implementar dentro del package ar.edu.unnoba.poo2018.ods.controller el controlador para la funcionalidad de login. La clase debe llevar el nombre de LoginController e implementar la annotation @WebServlet y extender de la clase HttpServlet. Sobrecribir el método doPost.
 - a. Sobre la anotación @WebServlet especificar como urlPatterns “/login”
 - b. En el método doPost, tomar del HttpServletRequest los atributos email y password crear una instancia del modelo Usuario con los datos recibidos.
 - c. Si usuario y contraseña coinciden con “poo2018@unnoba.edu.ar” y “unnoba2018!” agregar el usuario en session y redirigir al servlet welcome.
 - d. En caso contrario, establecer el status code del response con el estado HttpServletResponse.SC_UNAUTHORIZED y mostrar nuevamente la vista del index con el formulario de login mostrando el mensaje “Usuario y/o contraseña incorrecta”.
6. Implementar dentro del package ar.edu.unnoba.poo2018.ods.controller el controlador. La clase debe llevar el nombre de WelcomeController y al igual que en el caso de login implementar @WebServlet y extender de HttpServlet
 - a. En este caso se deberá sobrecribir el método doGet.
 - b. El servlet debe validar que una instancia de Usuario esté session. Si el usuario está en session simplemente deberá mostrar el contenido del archivo welcome.jsp (este archivo simplemente saluda mostrando el email de usuario en session)
7. El proyecto tiene seguir las especificaciones de una aplicación web y utilizando Maven como herramienta para la gestión y construcción del proyecto.
8. Se pide respetar la convenciones de nombrados de Java para clases, métodos, atributos y constantes.
9. Se deberá utilizar Git como herramienta de versionado para el código del proyecto.
 - i. Seguir el modelo de branches propuesto:
 1. Tener un branch con el nombre development. Este será el branch que tendrá todo el código del trabajo incremental. Luego, al momento de comenzar a trabajar en una nueva práctica se deberá crear un branch a partir de development para poder trabajar sobre las actividades específicas de la práctica. Al finalizar la práctica el branch deberá ser mergeado con el branch development. La convención de nombrado para cada uno de los branches de las prácticas es features-poo2018-tp[N], donde [N] será el número de la práctica
 - ii. El versionado de código será local, es decir, no será solicitado requerido subir los cambios de código a un repositorio central.

Entrega

La entrega podrá ser grupal con un máximo de 2 personas por grupo y se deberá entregar subiendo a Virtual un único archivo adjunto en formato “.zip”. El archivo adjunto deberá contener la imagen del diagrama de clases, todo el código fuente de la solución planteada, notas e información adicional que considere pertinente para la correcta ejecución del programa realizado.