ITEC 4020M: Internet Client-Server Systems

Professor: Jimmy Huang

Assignment 2: Implementing a Prototype secured Web Information System via Servlet and XML

March 16, 2023

Group 1

Keeran Ravishankar (216387771)

CheukWai Chan (218464057)

Domenico Lasalandra(217125105)

Peipei Yu (215869753)

## 

## **Introduction**

The internet is one of the most useful things ever created as it provides people with fast and easy access to communication and information. But with all this access to social media, electronic mail, audio/video transmission a new threat to safety emerges. This threat is compromised online security. Security is one the most important measures to have when using or accessing the internet as personal information can be easily leaked and accessed by the wrong people.

So for this assignment, We are investigating and learning about advanced security themes and issues for major web information systems. The objective of the assignment is to build a web information system by using XML and servlet. The programming portion should contain two parts. The first being dynamic web pages and the second is the prototype secured web information system. The web information system must contain login pages, information content pages and a logout button on each page. Most importantly the web security portion should contain restrictions on what a logged in user can/can’t do. These restrictions can include no copying/pasting text, no printing of the webpage, no access unless the user is logged in and so fourth. Our report will focus on the methods we took to create our web information system and the steps we took to implement our security objectives.

**Methods and Implementation**

Our group has created 3 Java servlets which are loginHandle, logoutHandle, and dataServlet. The Java servlets allow us to extend the functions of servers that host applications accessed by the use of a request response programming model. There are also other files such as Books.Xsl, Catalog.XML, Web.XML and login.jsp. These files define the style, the security measures, and contents of the website and also the communication of clients with the server.

**Description Of Website**

The website consists of a login and the home page. The contents of the homepage consists of the books sorted by the title, genre, author, and description. The page allows a user to login using an account and you can look at the library of books. There are also several security measures on the page such as users not being able to use the back button to login, user login is the only way to access the page, Images/text are not copyable, printable or savable and the server side is protected.

**Login Handle Servlet**

The loginHandle is a java servlet that handles the login requests and implements the doPost methods of the HTTPServlet class to handle the HTTP Post request method. The doPost method retrieves the userId and pwd parameters from the HTTP request object using getParameter method. An HttpSession object is created and it is used to store session attributes. The type of the response is html and a PrinterWriter object is the one that writes the response to the object. If the User ID and password match then you are able to access the home page as the session attribute login\_status is set to login. If the User ID and password doesn't match then the login\_status has failed and the user is just taken back to the login page.

**Logout Handle Servlet**

The logoutHandle is another Java Servlet that is used to handle the logout requests. It extends the HttpServlet class and overrides the doGet and doPost methods to handle both the HTTP GET and POST requests. The doPost method invalidates the current session using the invalidate method and redirects the user to the login page using the sendRedirect method.

**Data Servlet**

The Data servlet is the servlet that handles all the GET requests. The servlet converts XML files to HTML using an XSL stylesheet and it returns the HTML to the client. The servlet takes the parameter GenreType which is used to filter data that is displayed in HTML.

The code starts getting some response headers to ensure that the page is not cached. It then checks if the user is logged in by checking if the login\_status attribute is set for the session. If the user isn't logged in the servlet would then redirect the user to the login page.

The code then reads the XML file and creates an XSL stylesheet based on the genreType parameter. The XSL stylesheet is used to transform the XML file to HTML. The HTML is then written to the response and sent back to the client. The current implementation of the code reads the files from a hard-coded file path.

**Web XML**

The web.xml file is for the Java web application. The file starts with the declaration of the web-app element and it specifies the XML namespaces being used in the version of the Java servlet specification. The display-name element names the web application. The welcome-file-list element gives a list of the default pages that the web container needs to look for when a client makes a request to the root of the application. The servlet elements define servlets that handle requests from clients. Each servlet element has a servlet-name and a servlet-class. Servlet-mapping elements map servlets to URL patterns that clients use to access them. The URL patterns must match the value of the action attribute in either the Html page or the Jsp page. The three Java servlets loginHandle, logoutHandle, and dataServlet have their own unique URL pattern.

**Catalog XML**

The catalogue XML code contains the characteristics of elements such as author, title, genre, price and publish date. The Catalog XML file is also linked to the XSL stylesheet called books.Xsl. The XSL stylesheet helps the XML file become presentable to the user viewing the web page in the browser.

**Books XSL**

The stylesheet defines the design and layout of the HTML page and it displays a catalogue of books. The background colour, font family, size, text, and styling for the navigation bar and buttons are all defined in this XSL sheet. The style sheet contains XSLT code that selects and iterates over a set of XML elements which are the books in the catalogue. Each book creates an HTML <div> element that displays the book's characteristics such as title, author, genre, etc. The Stylesheet includes a form with buttons as well to filter out books in the catalogue by genre. It also prevents the context menu from appearing when the user right-clicks on the page.

**Login JSP**

The login.jsp code defines a login form in HTML and CSS and it contains two input fields for the user ID, password and a submit button. The form authenticates users and allows them access to restricted areas of the website. The form action attribute is set to login where the form data will be received. The form method is set to post so the form data will be sent in the request body. The CSS styles define the appearance of the form and its elements. The form has a container with a fixed width and a white background colour. There are also input fields which are border, border radius, and padding and they are centred with a margin-left and right set to auto. The submit button has a yellow background with white text which are in all capital letters. When the button is hovered over it changes colour to orange. Warning messages are also displayed if any users enter incorrect credentials.

**Test Case**

Test number: 01

Test name: Login

Test case description: To check the login functionality of the website

Test scenario: validating after user entering ID and password, login successfully for right credentials

Test step: 1. Enter ID and password 2. Click sign in button

Test input: ID = user, password = 12345

Result: Display the home page, with a navigation bar at the top and all the books listed.

Test number: 02

Test name: Logout

Test case description: To check the logout functionality of the website

Test scenario: user clicks the logout button and is redirected back to the login page.

Test step: 1. Click the logout button

Test input: -

Result: Destroy the current session and the attributes, and display the login page.

Test number: 03

Test name: Book filtering

Test case description: To check if the website can display the right content after filtering.

Test scenario: user chooses a genre by clicking the corresponding button, then the website will only display the books of the chosen genre.

Test step: 1. Click one of the genre choice buttons

Test input: -

Result: A list of books based on the genre user chosen.

**Web Information System Security Measures**

**User Authentication**

A user's credentials must match the credentials of authorised users. A user will have to provide the right ID and password to login to the page. If the user entered the wrong ID or password, “login\_status” will be set as “failed” and a message will be shown to tell the user the login failed. In this case, the user will not be redirected to the home page. It helps block out unauthorised access to the website.

After validating the user login, a new session will be created. Session attribute “login\_status” is set as “login” and redirects the user to the home page. In the home page, it will check the “login\_status” of the current session. If the attribute is not equal to the “login”, then it will be redirected to the login page for user validation. Therefore, if the user logs out, the current session will be destroyed, and “login\_status” will become null. Even if the user types in the URL directly, it will be redirected back to the home page and cannot access the content inside.

**Access control**

Authorised users can only view the data that is displayed on the home page, they are not allowed to save any data from the website, including text and images. For security measures on the client side, the right click and selecting function are disabled:

CSS: body{-webkit-user-select: none ;-ms-user-select: none; user-select: none; }

JavaScript: document.addEventListener('contextmenu', event => event.preventDefault());

So, users cannot select and copy the text or download images using the “save as” function. It also makes it harder to view the source code directly from the website. Moreover, if the users try to print out the webpage, all the content will not be displayed on print outcome.

CSS: @media print {html, body {display: none;}}

Furthermore, for security measures on the server side. All images are converted to base64 and are used in the html response to the client. So the path of the image will not be shown on the source code. Hence, users will not be able to download images and see the path of images on the server directory.

**Results/Conclusion**

In the beginning of this project, one of the challenges was the configuration and setting up the server. It is important to make sure that all required jar.files are included in the project, so the website can run smoothly. Another challenge of this project is the security part of the website. Some of the security measures can be done on the client side, however it is hard to do it on the server side. We found that it is difficult to keep 100% safe for the information on the website. So what we can do is to limit the possibility of unauthorised access and have appropriate security control on the website. Also, our project can only run on the local computer, it does not work when we put those files on the SIT server. Therefore both IS security measures and configuration on SIT server will be further studied in the future to achieve a more secured web information system.