Lesson 02 Demo 01
Session Tracking in Servlets

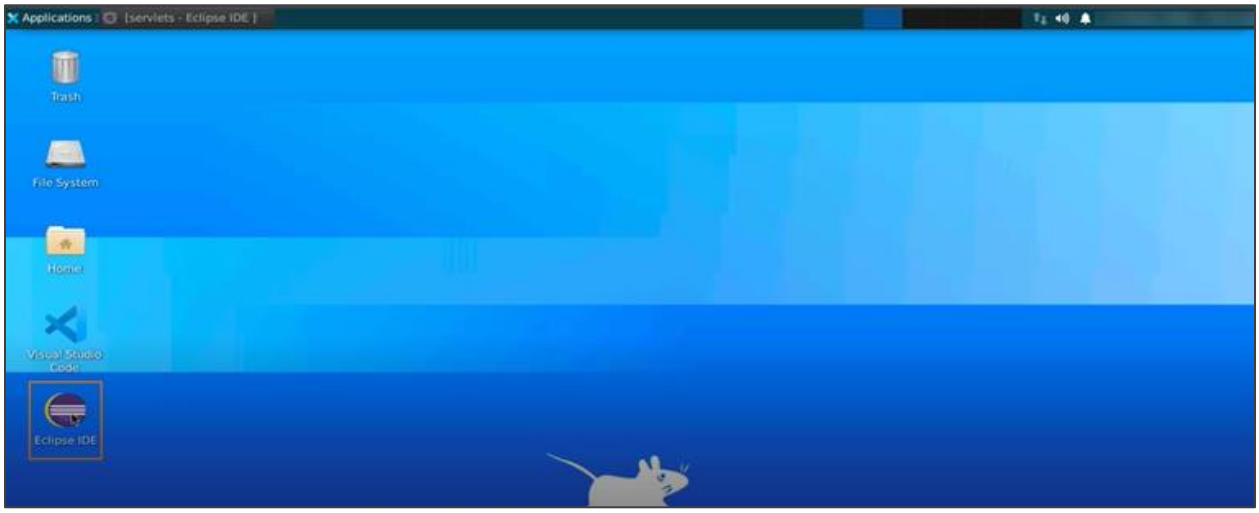
Objective: To demonstrate the implementation of session tracking in servlets using cookies **Tools Required:** Eclipse IDE **Prerequisites:** None

Steps to be followed:

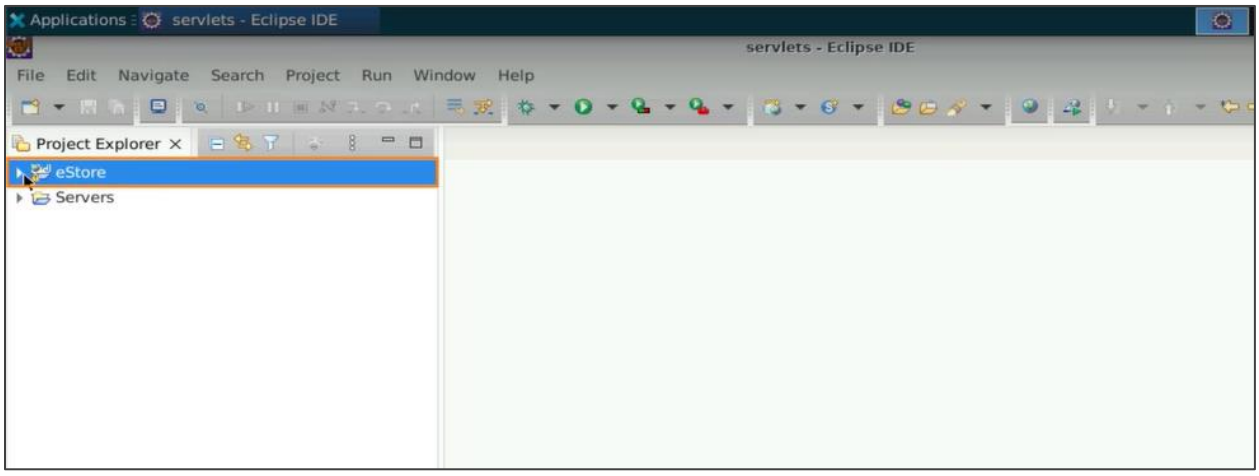
1. Set up and authenticate the user
2. Implement session tracking with cookies

Step 1: Set up and authenticating the user

1.1 Open **Eclipse IDE**

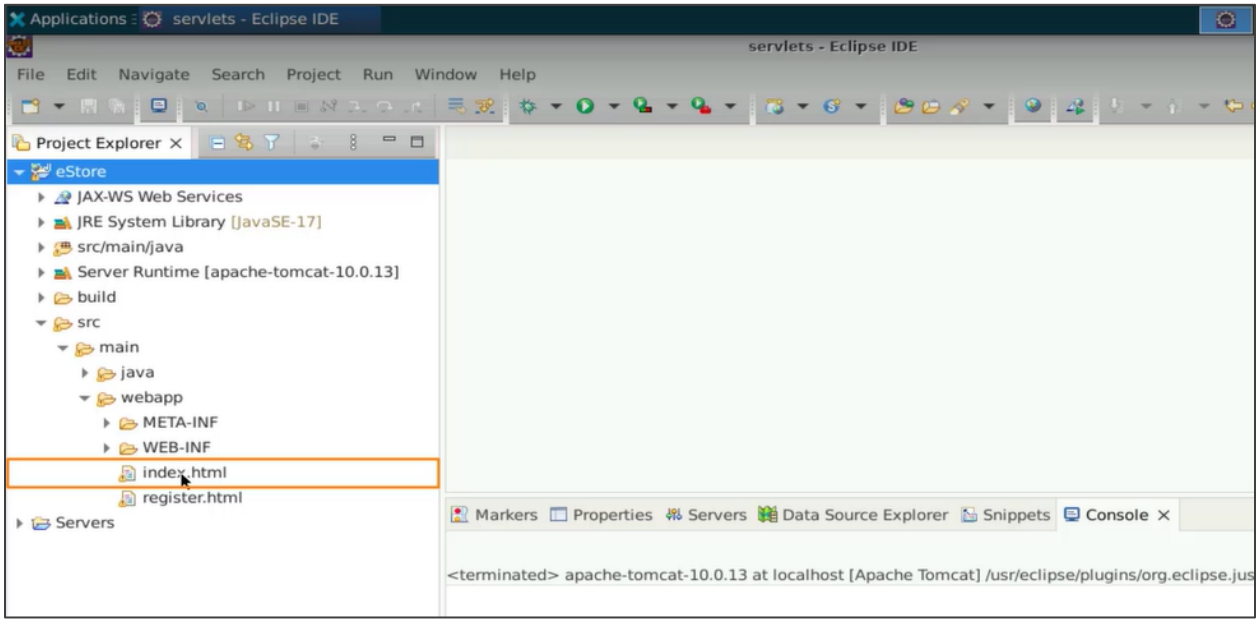


1.2 Open the **eStore** project



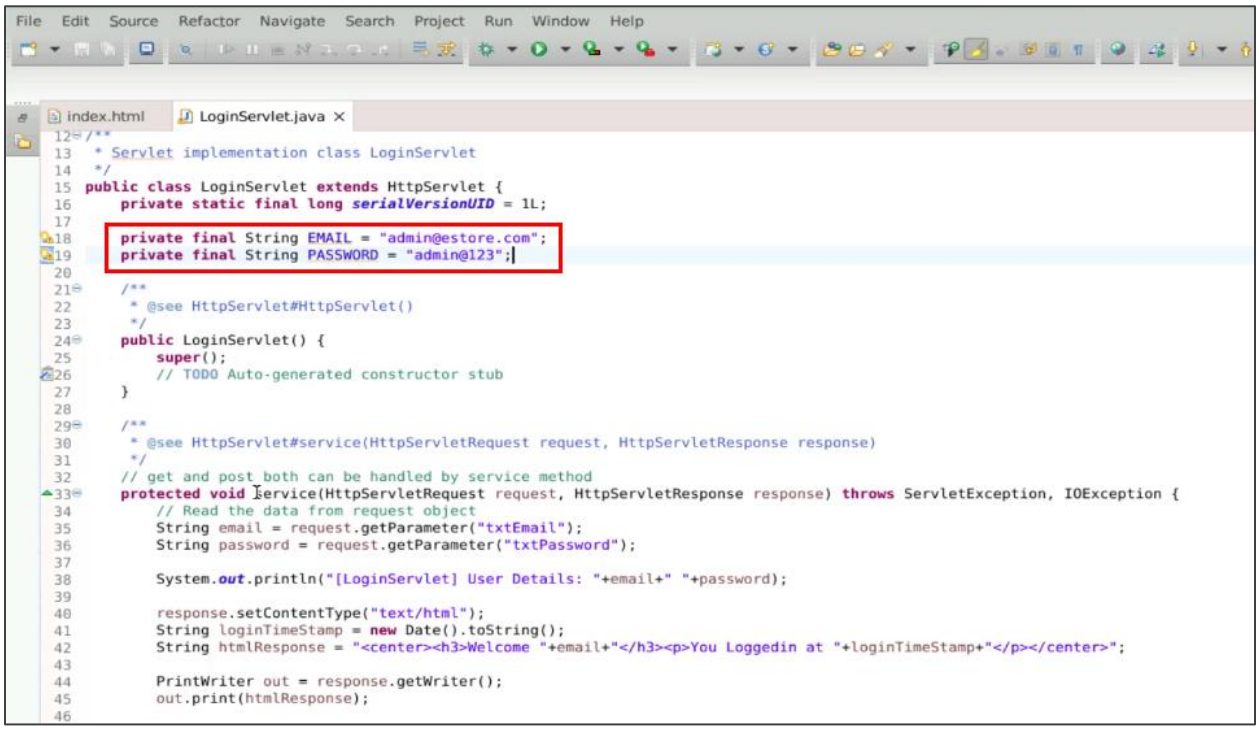
Note: Please refer to the previous demo on how to create the **eStore** project

1.3 Navigate to the **index.html** page under the **webapp** folder, which serves as the registration page

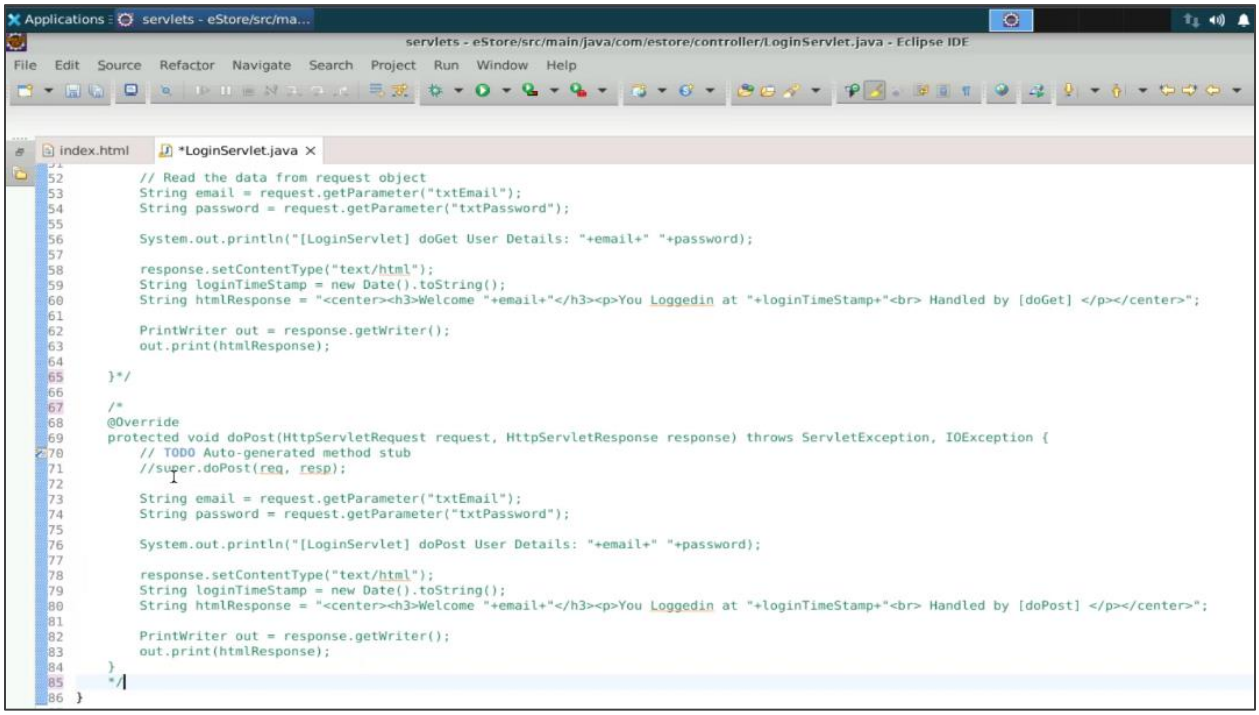


Note: In the **index.html** file, there is a login form as **Login** for the user with the action set and a link to navigate to **register.html** for account registration.

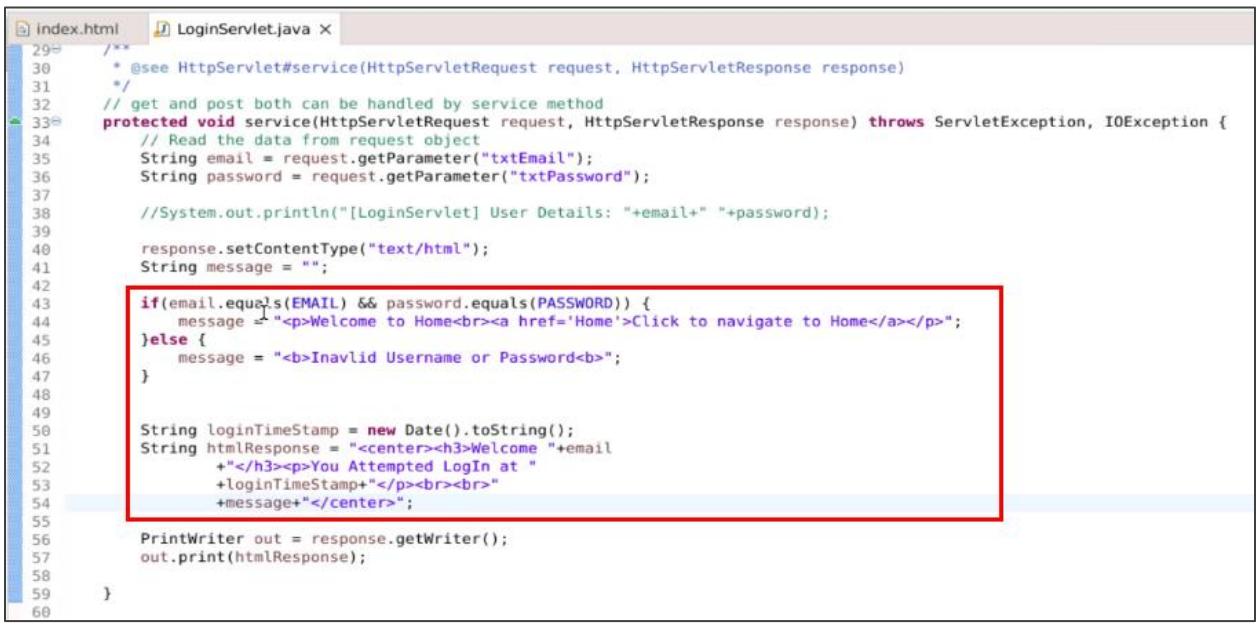
1.4 In the **LoginServlet.java**, globally define the attributes such as **EMAIL** and **PASSWORD** as private and final



1.5 Comment the **doGet** and **doPost** methods in the **LoginServlet.java** file

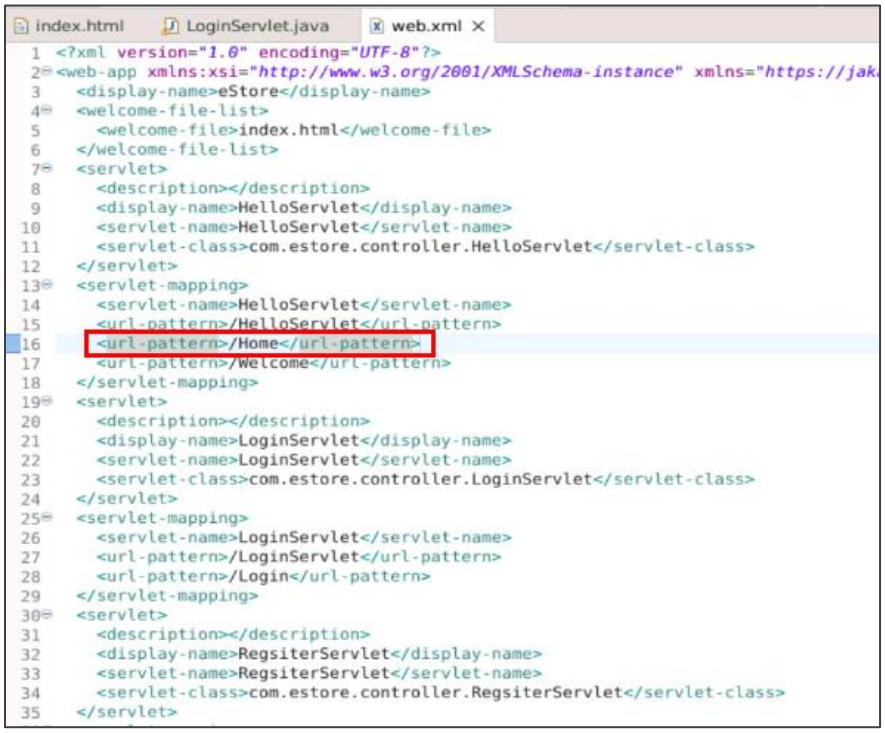


1.6 In the service method of **LoginServlet.java**, add an if-else condition to validate if the email and password match. If they match, show the welcome page with a button to navigate to the home page. Otherwise, display a message saying **Invalid Username or Password** along with the timestamp of the login.

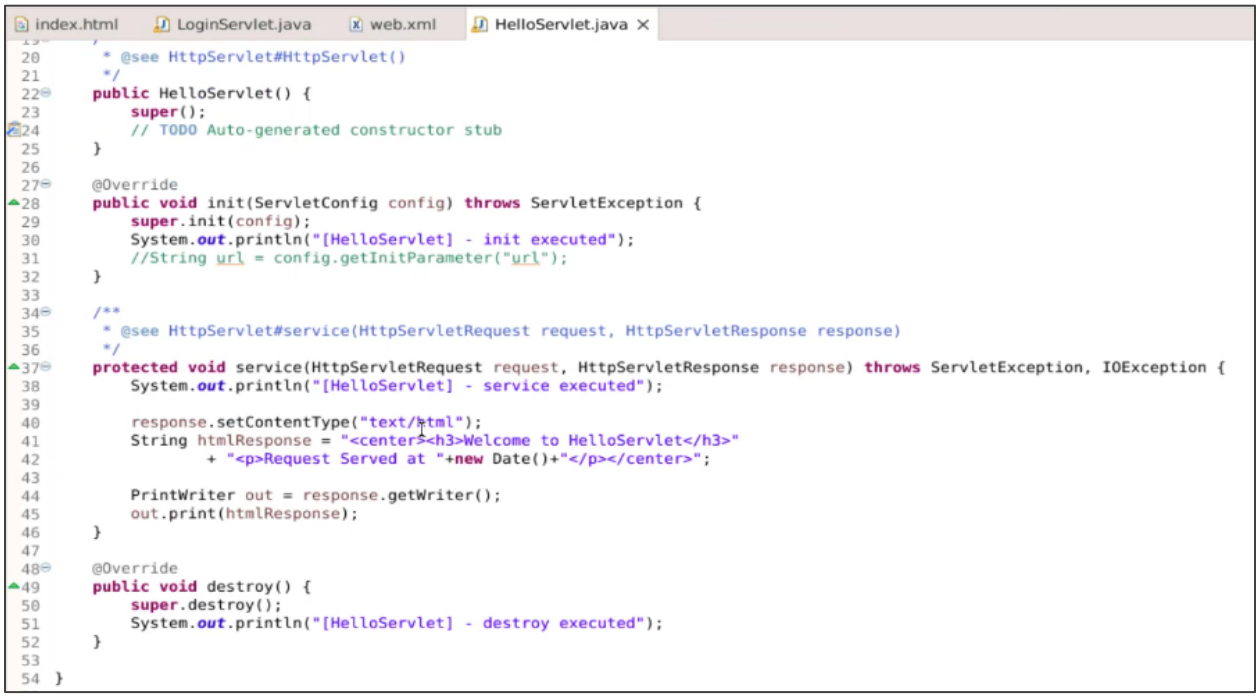


Observation: From the index page, a request will be sent to the **LoginServlet**, which will capture the email and password from the request.

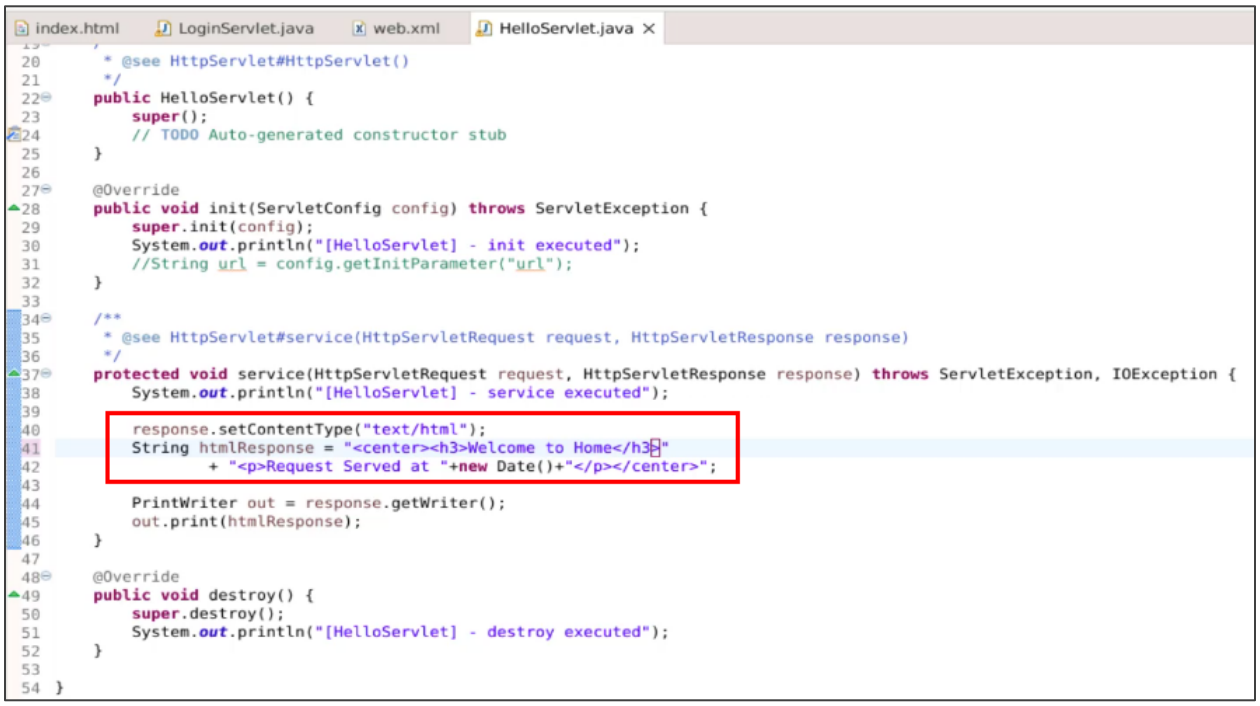
1.7 Open the **web.xml** file and update the **url-pattern** of HelloServlet from **/Hello** to **/Home**



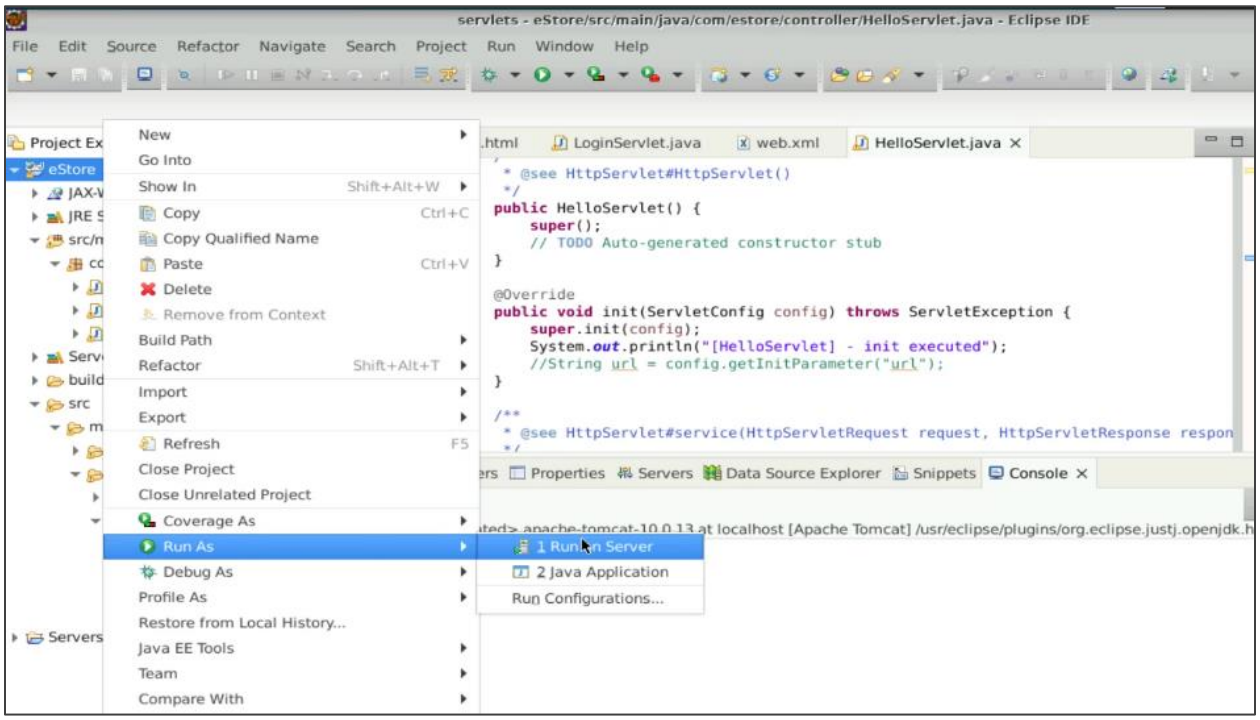
1.8 Open the **HelloServlet.java** servlet file where the init and destroy methods were previously configured



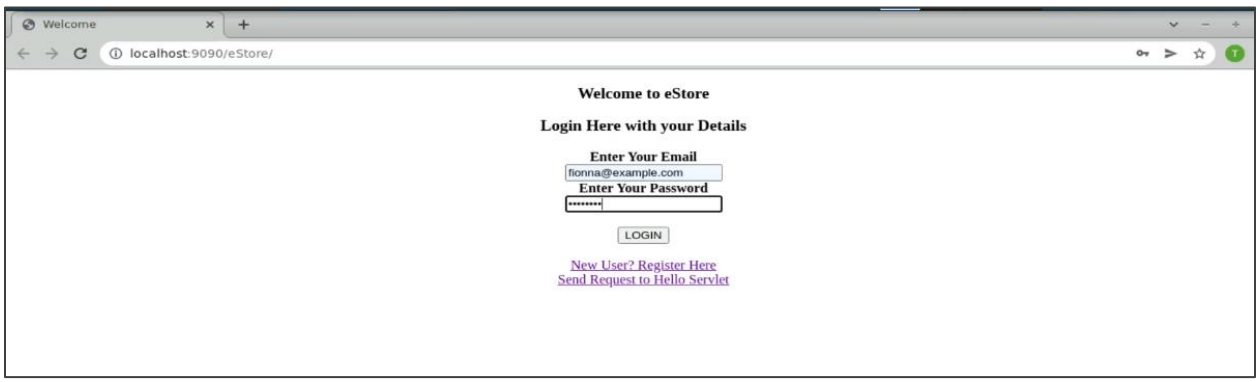
1.9 Update the **htmlResponse** message in the service method of **HelloServlet.java** to display **Welcome to Home** along with the current date and timestamp



1.10 Save and run the code by right clicking the **eStore** project, selecting **Run As**, and selecting **Run on Server**



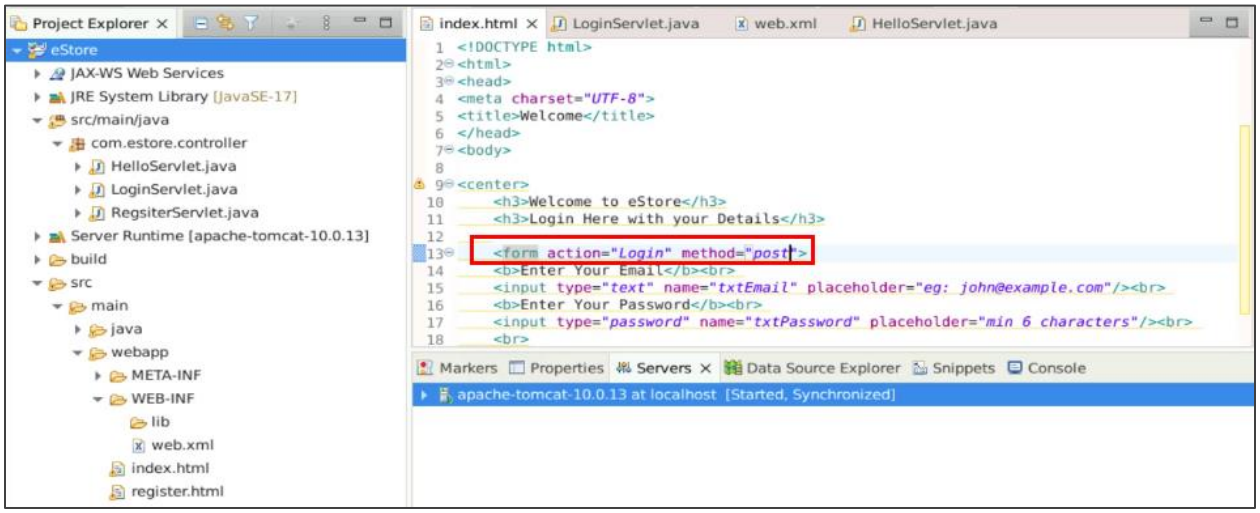
1.11 Enter invalid credentials and click the **LOGIN** button



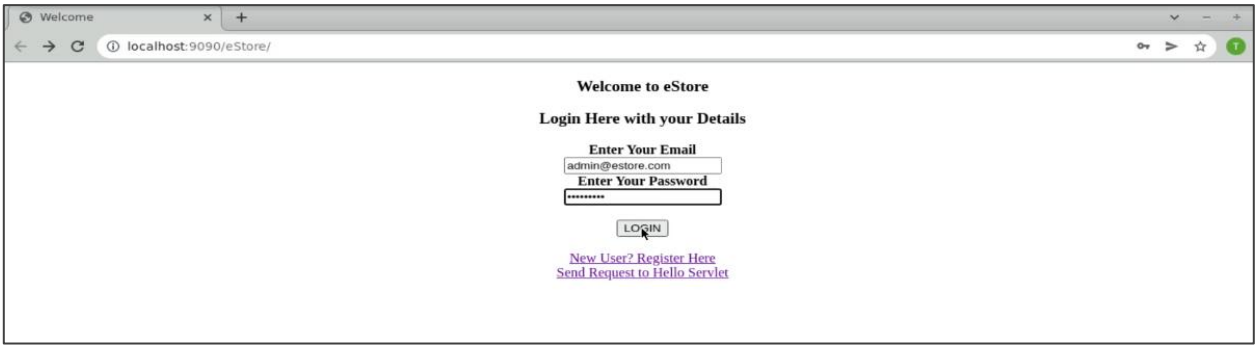
Observation: You will observe the output displaying **Welcome username@example.com** with the attempted login timestamp. This information indicates a successful login attempt.

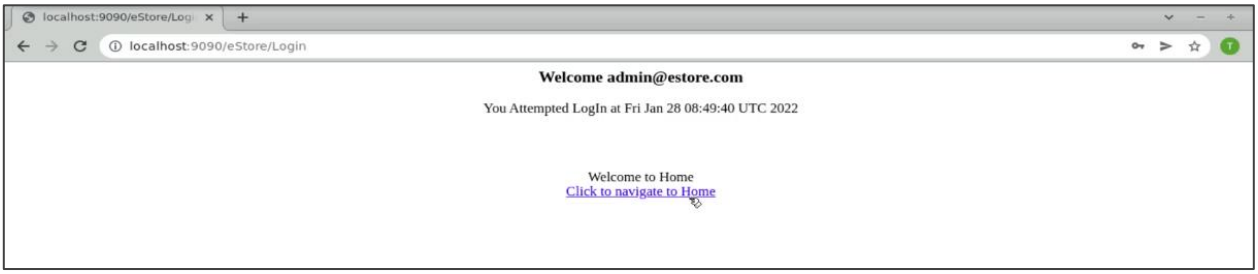
In case of an invalid username or password, you will receive the message **Invalid Username or Password**. Additionally, you can view the GET request sent during the login process in the URL.

1.12 Go to the **index.html** file and change the method to **POST** to ensure that the data is not present in the URL



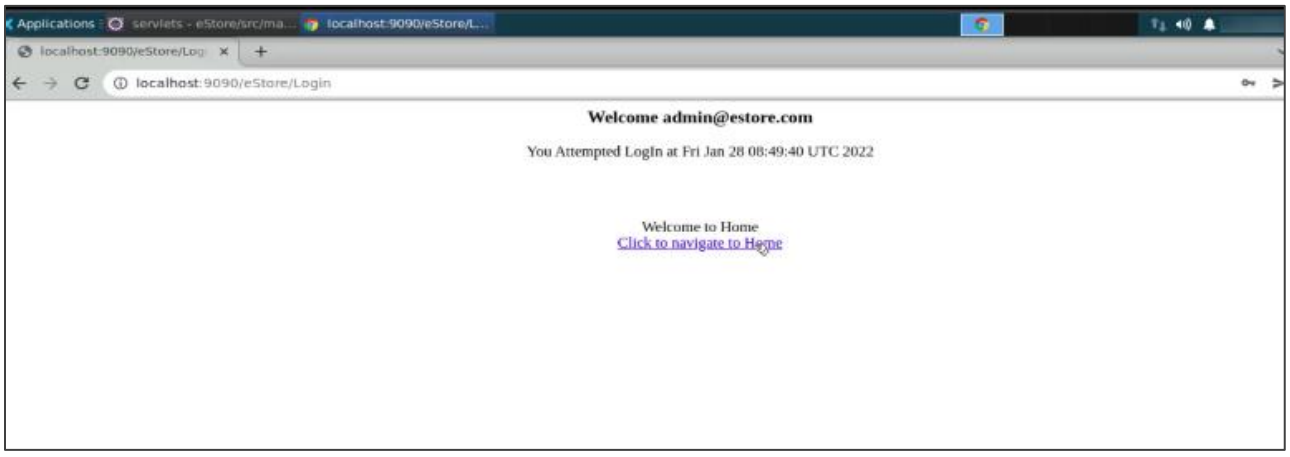
1.13 Enter the correct credentials for the admin and click on the **LOGIN** button





Observation: Now you will observe that the URL doesn't contain any data and the message **Welcome to Home** is shown.

1.14 Click on the **Click to navigate to Home** link



The result will be displayed as **Welcome to Home**, indicating that you have successfully navigated to the home URL.

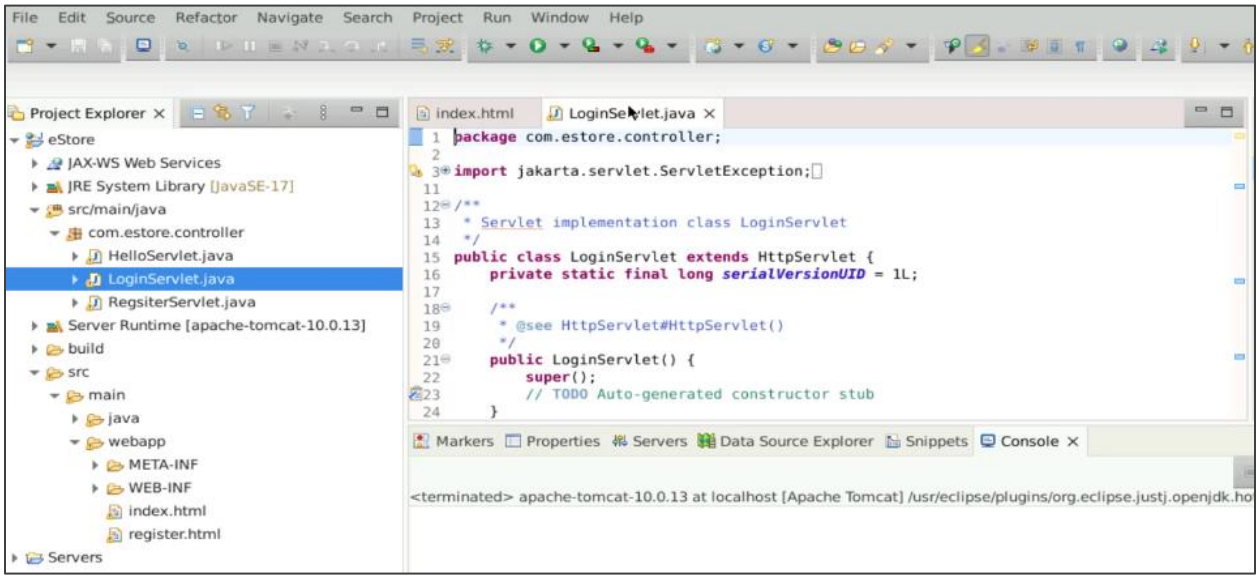
1.15 Add a private final String in the **LoginServlet.java** file for the variables **NAME** and **TOTAL_SALES**, with the values set as **John** and **3000,** respectively

index.html | *LoginServlet.java | web.xml | HelloServlet.java

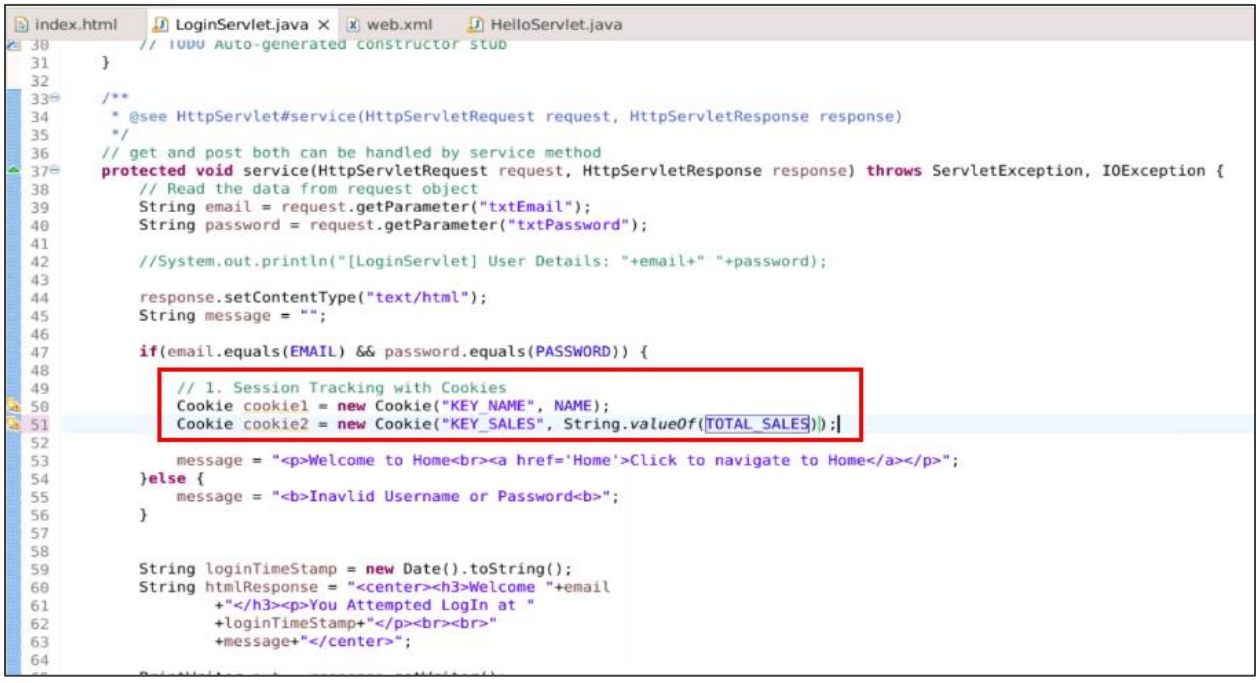
```
1  package com.estore.controller;
2
3  import jakarta.servlet.ServletException;
11
12 /**
13  * Servlet implementation class LoginServlet
14  */
15 public class LoginServlet extends HttpServlet {
16     private static final long serialVersionUID = 1L;
17
18     private final String EMAIL = "admin@estore.com";
19     private final String PASSWORD = "admin@123";
20
21     private final String NAME = "John";
22     private final int TOTAL_SALES = 30000;
23
24     /**
25      * @see HttpServlet#HttpServlet()
26      */
27     public LoginServlet() {
28         super();
29         // TODO Auto-generated constructor stub
30     }
31
32     /**
33      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
34      */
35     // get and post both can be handled by service method
36     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
37         // Read the data from request object
38         String email = request.getParameter("txtEmail");
39         String password = request.getParameter("txtPassword");
40
41         //System.out.println("[LoginServlet] User Details: "+email+" "+password);
42
```

Step 2: Implement session tracking with cookies

2.1 Go to the **LoginServlet.java** file to implement session tracking with cookies

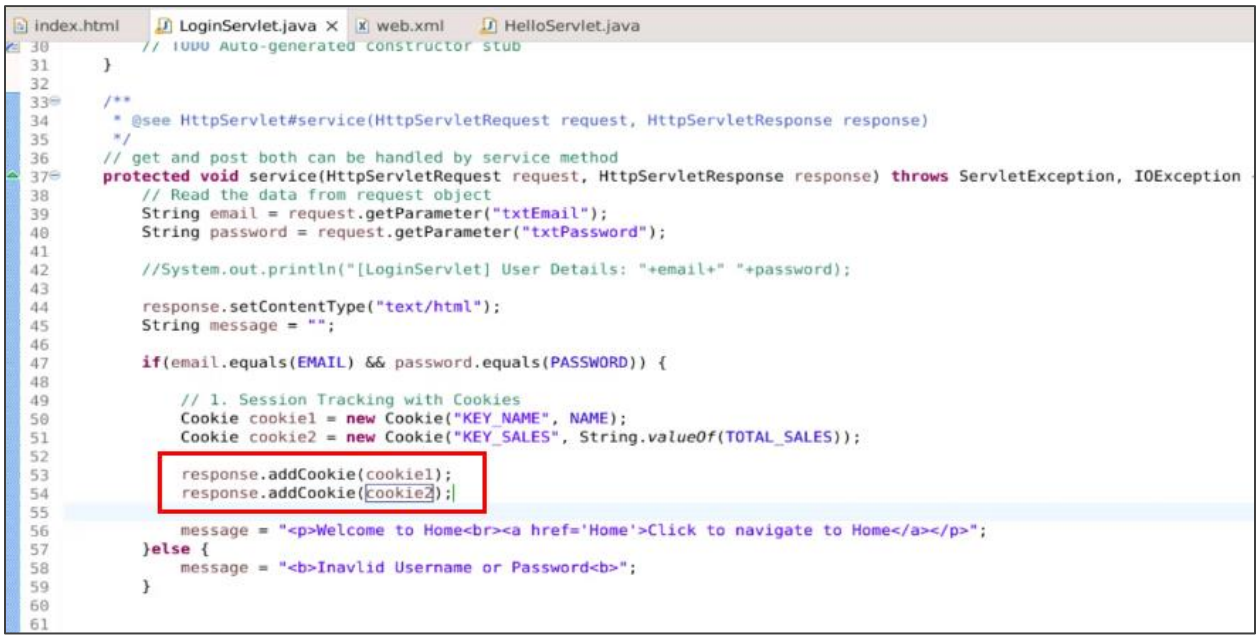


2.2 Under the if clause in the **LoginServlet**, create Cookie objects named **cookie1** and **cookie2** to store the values **NAME** (**John**) and **TOTAL_SALES (3000)**



Since cookies cannot store integer values, use the **String.valueOf()** method to convert the value of **TOTAL_SALES** into a string.

2.3 Add the cookie objects as responses using the r**esponse.addCookie()** function



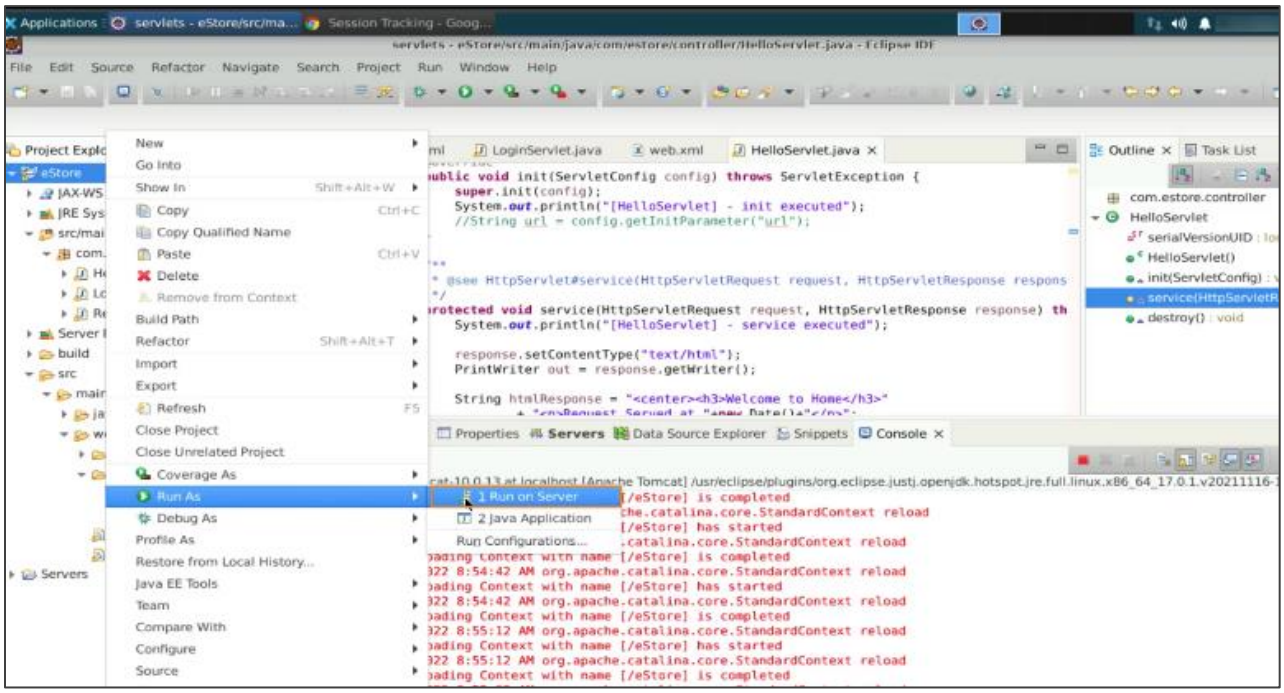
For session tracking, it is important to read cookies. First, we create the cookies, write them, and then read them.

2.4 Read the cookies from the request using the **Cookie[]** array and the **getCookies()** method. Iterate within the array using an enhanced for loop to print the name and value of each cookie

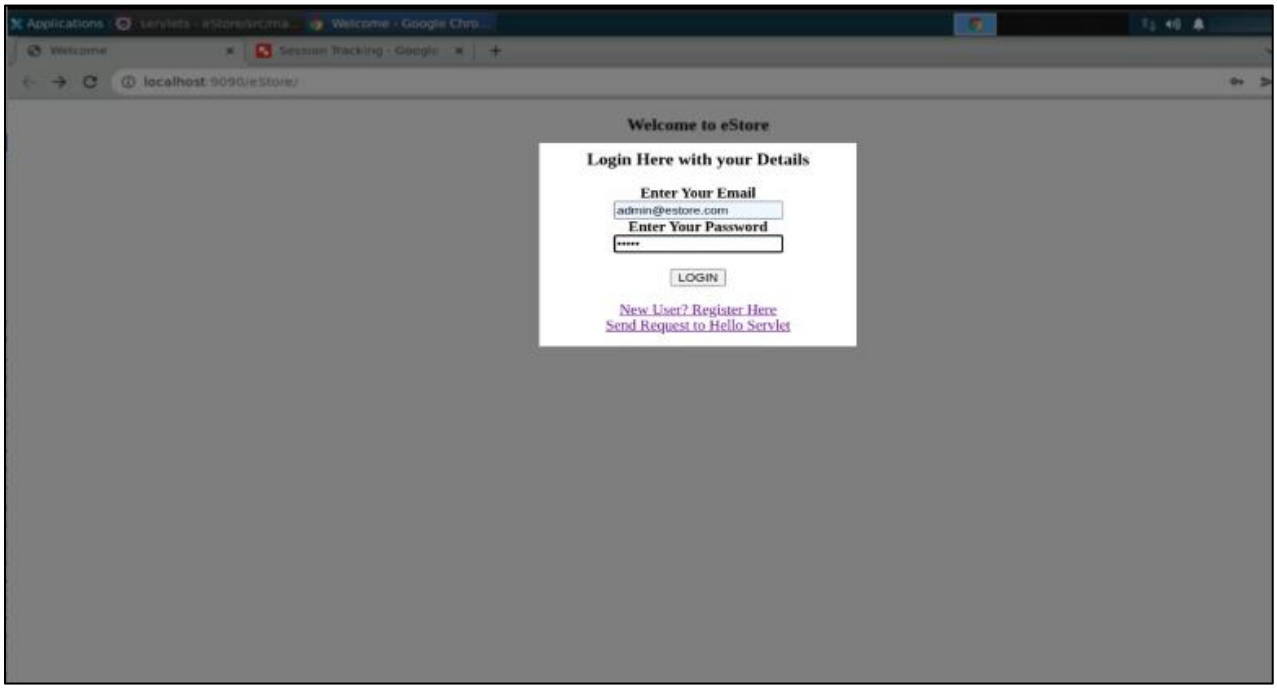
```
index.html   LoginServlet.java   web.xml   HelloServlet.java ×

29  public void init(ServletConfig config) throws ServletException {
30      super.init(config);
31      System.out.println("[HelloServlet] - init executed");
32      //String url = config.getInitParameter("url");
33  }
34
35  /**
36   * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
37   */
38  protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
39      System.out.println("[HelloServlet] - service executed");
40
41      response.setContentType("text/html");
42      PrintWriter out = response.getWriter();
43
44      String htmlResponse = "<center><h3>Welcome to Home</h3>"
45              + "<p>Request Served at "+new Date()+"</p>";
46      out.print(htmlResponse);
47
48      // 1. Session Tracking with Cookies
49      //    Reading Cookies
50
51      Cookie[] array = request.getCookies();
52      for(Cookie cookie : array) {
53          out.print("<p>"+cookie.getName()+" - "+cookie.getValue()+"</p>");
54      }
55
56      out.print("</center>");
57
58
59  }
```

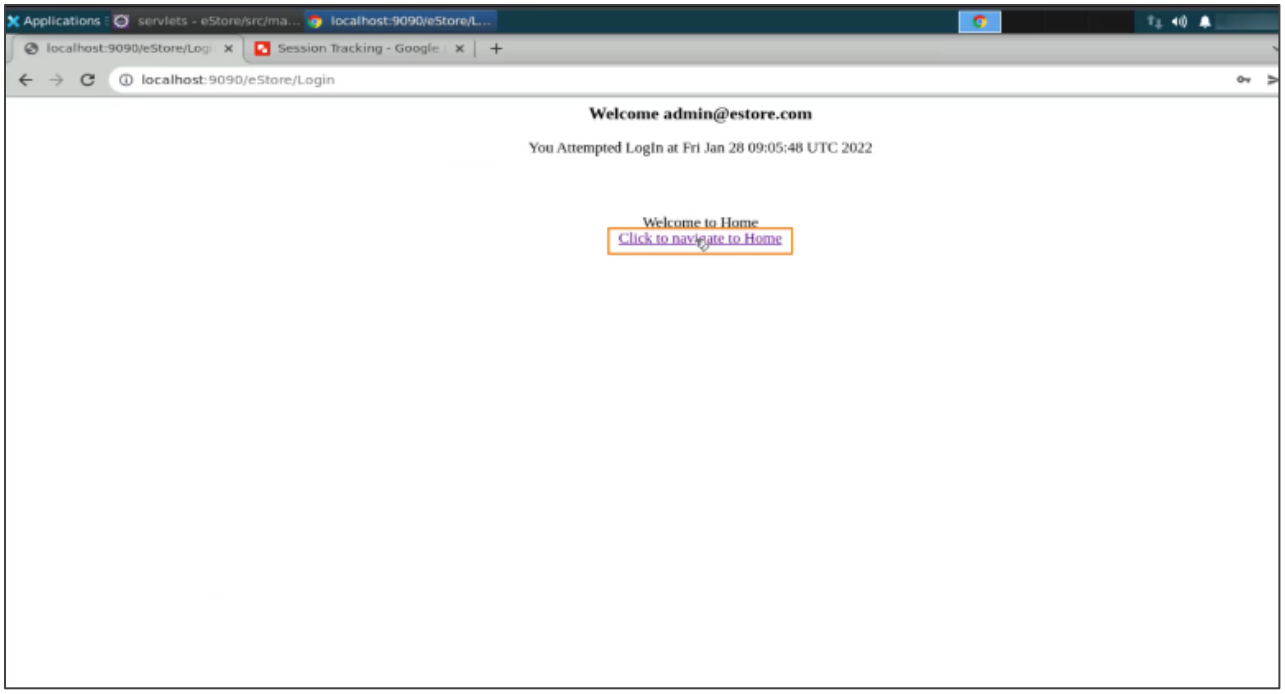
2.5 Save and rerun the code on the server

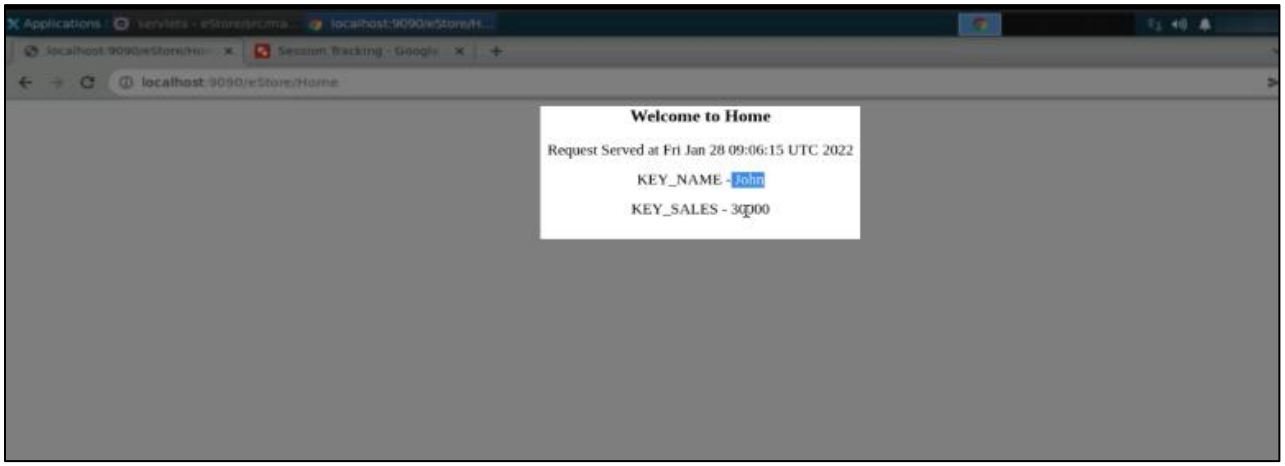


2.6 Return to the browser, log in again with the admin email and password, and click on the **LOGIN** button



2.7 Click on the **Click to navigate to Home** link





Observation: The result can be seen when it prints **the KEY_NAME** and **KEY_SALES** values. This is the first way to work with the session tracking technique. However, please note that this method may not work if cookies are disabled.

Following these steps, you have successfully implemented session tracking in servlets using cookies.