Lesson 02 Demo 03

Filters

Objective: To explore filters by creating a new filter and working on the login filter's response, pre-processing, and post-processing

Tools Required: Eclipse IDE

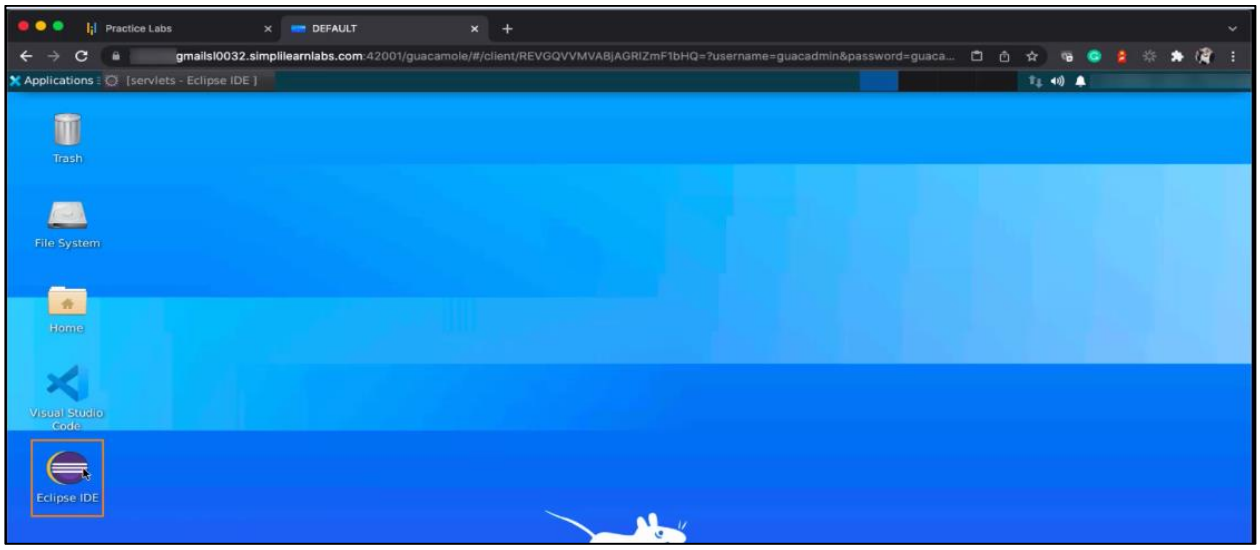
Prerequisites: None

Steps to be followed:

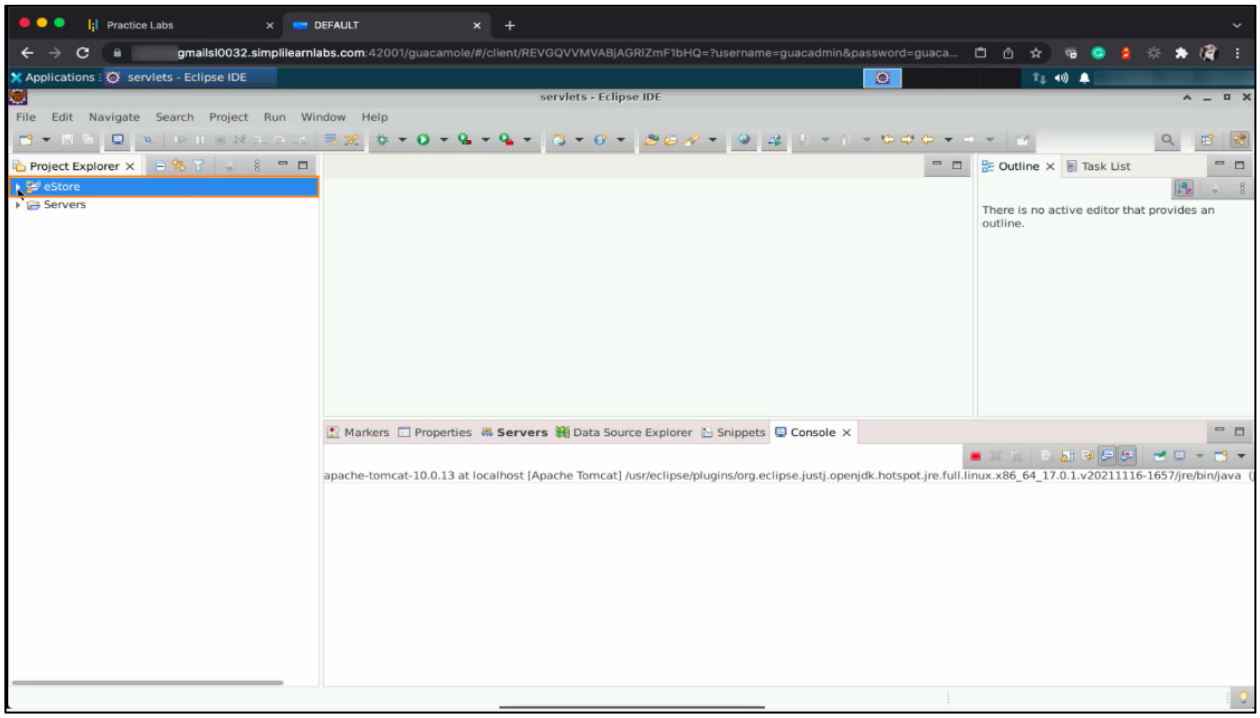
1. Create a new filter
2. Work with mandatory methods
3. Work with a filter mechanism
4. Work on the login filter's response
5. Make a filter for login parameters

Step 1: Create a new filter

1.1 Open **Eclipse IDE**

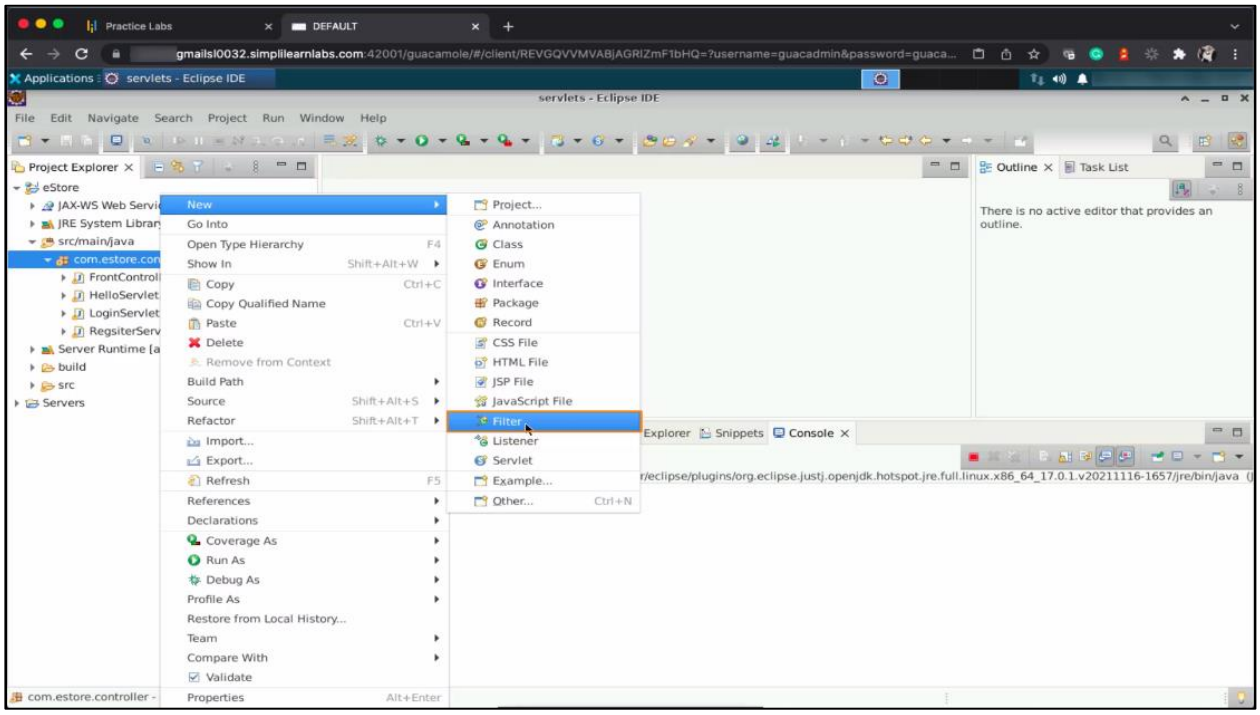


1.2 Navigate to the **eStore** project



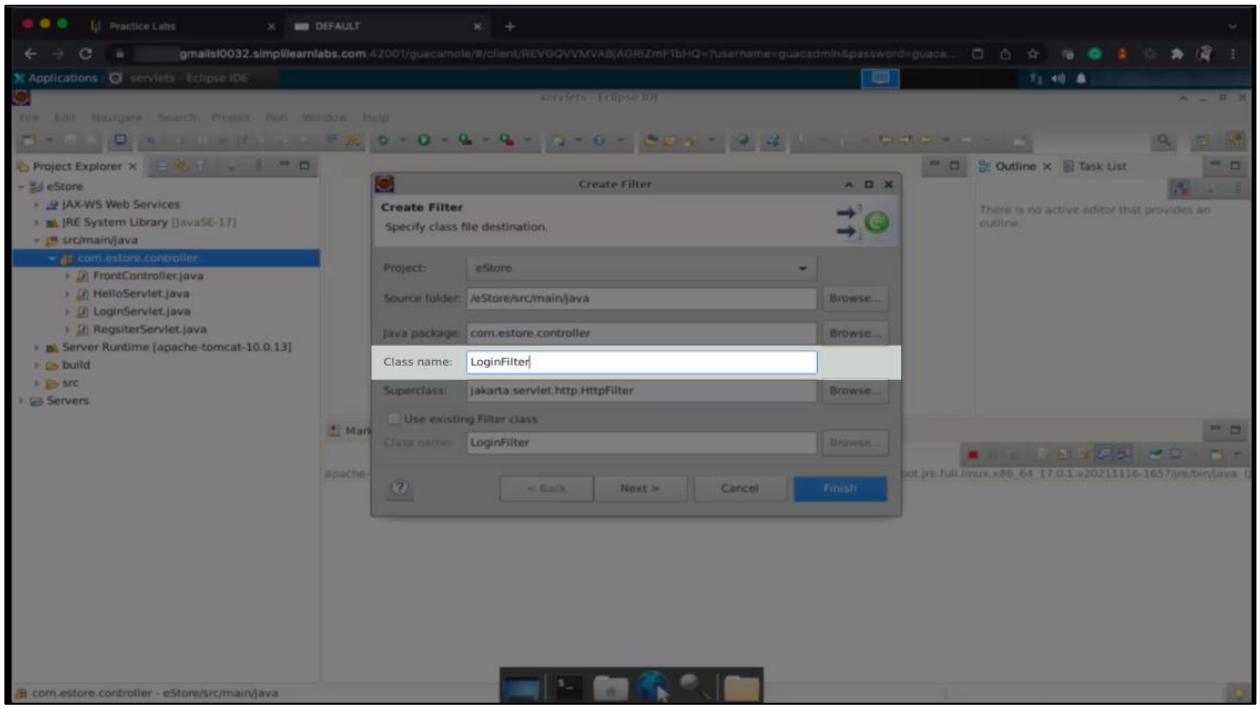
Note: Please refer to the previous demo on how to create the **eStore** project

1.3 Create a new filter by right-clicking on the project, selecting **New**, and clicking on **Filter**

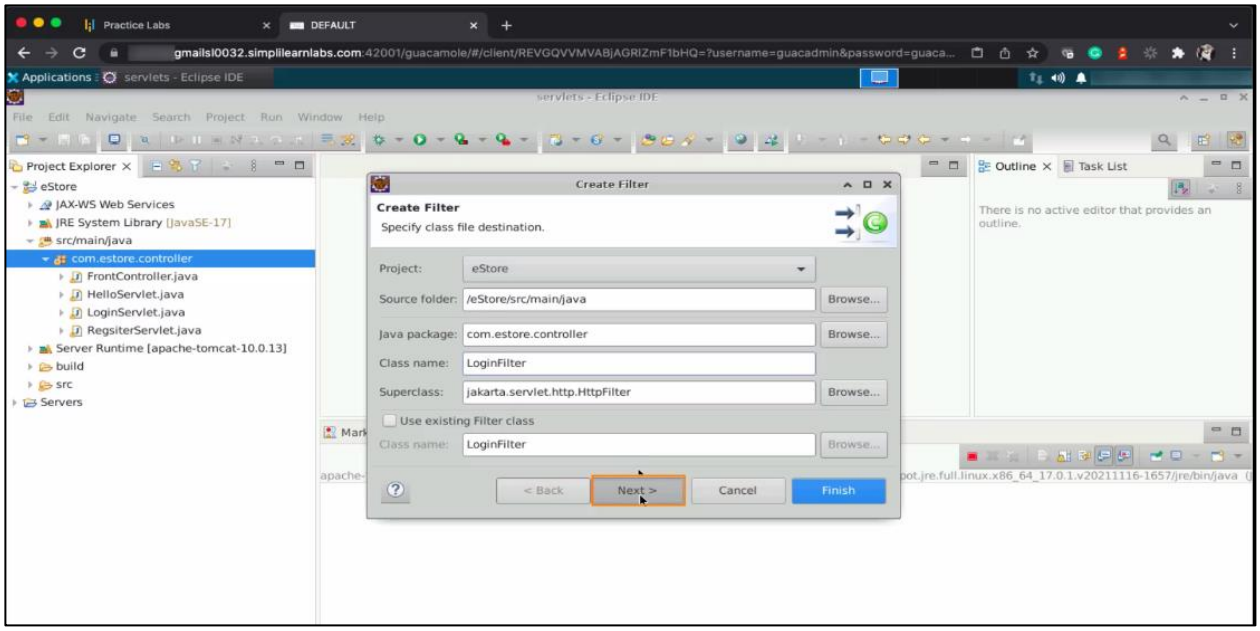


Observation: Rather than requests going directly to the servlet, the request will first navigate to the filter which will then send the request to the servlet.

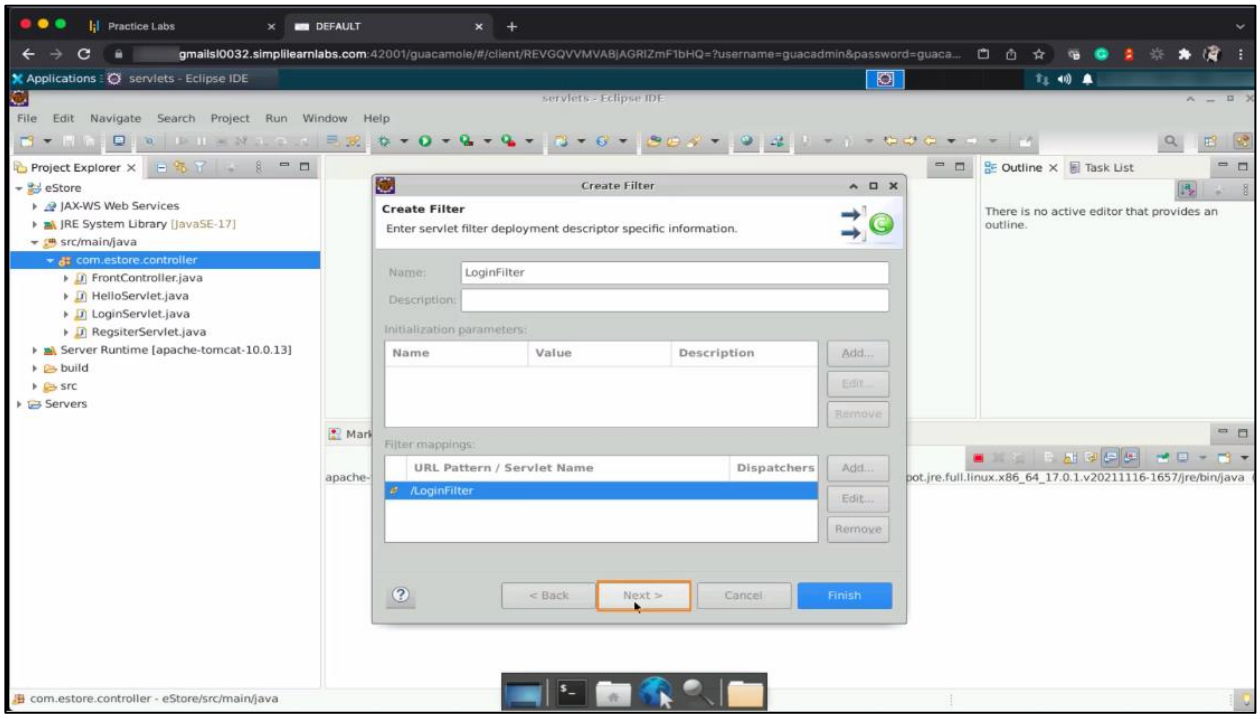
1.4 Name the filter as **LoginFilter**



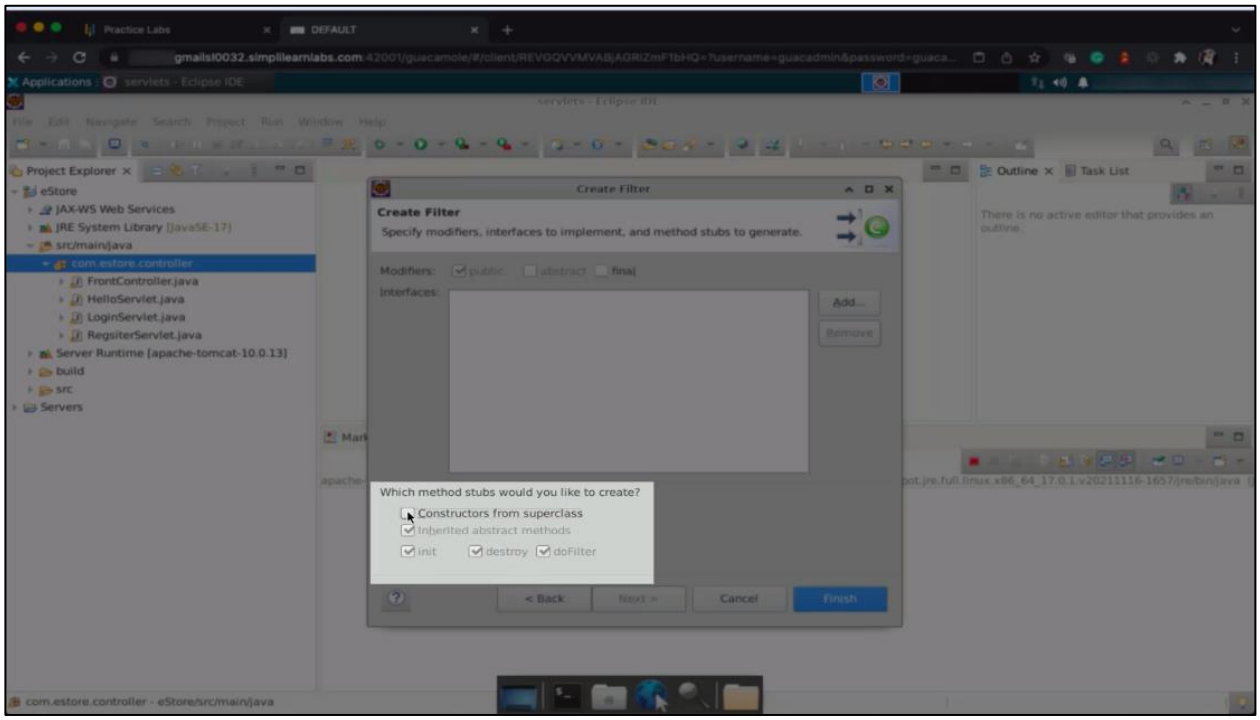
1.5 Click on **Next**



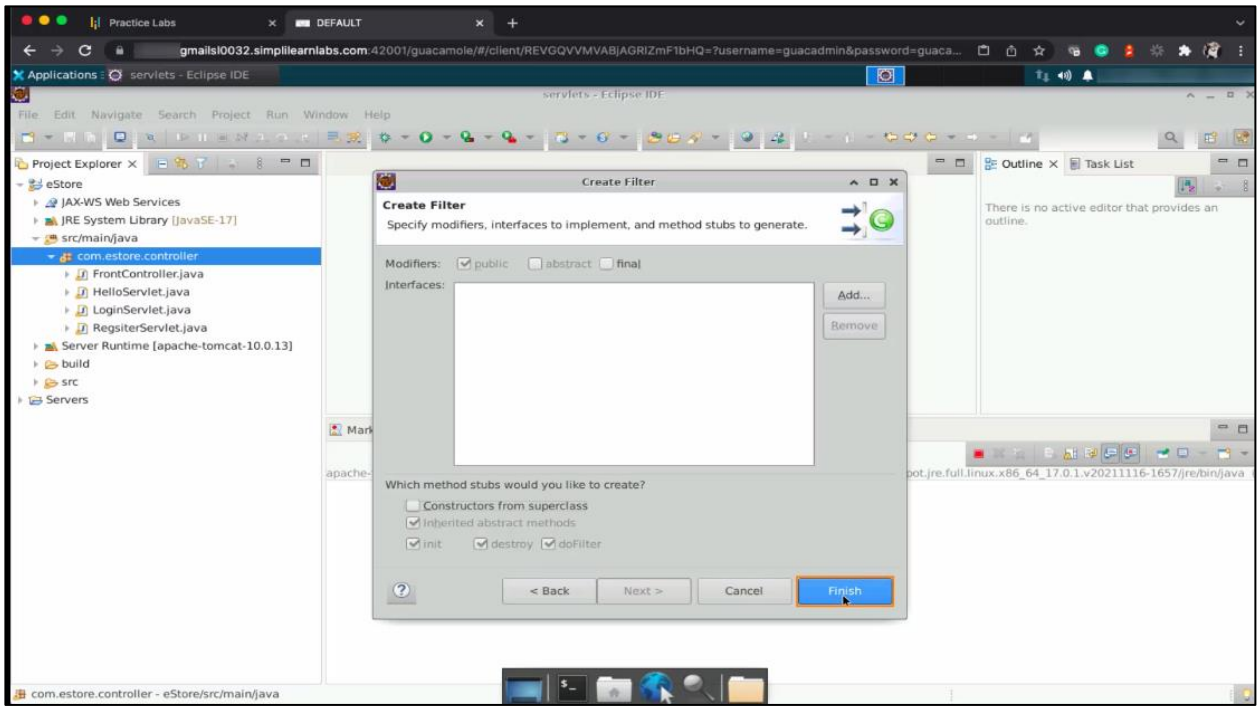
1.6 Specify the URL patterns by selecting the **/LoginFilter** as the URL-pattern and clicking on **Next**



1.7 Unselect the checkbox for **Constructor from superclass**



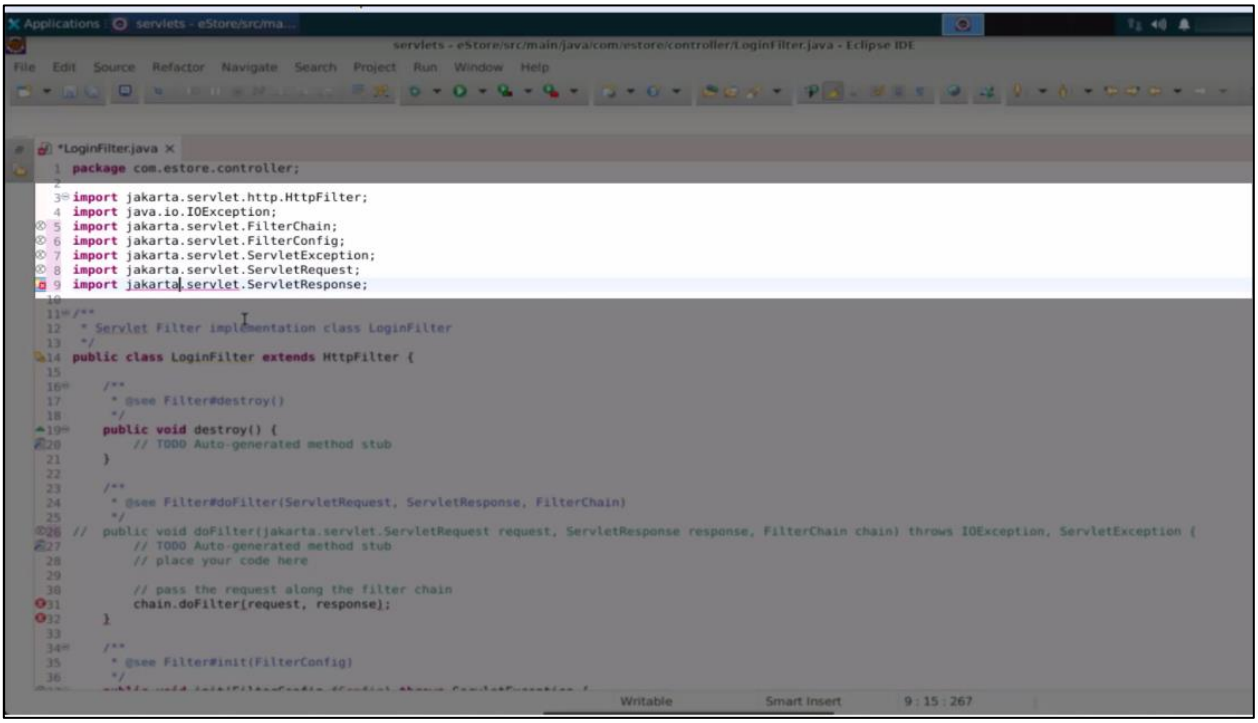
1.8 Click on **Finish**



Step 2: Work with mandatory methods

2.1 Navigate to the **LoginFilter.java** file and import the following packages:

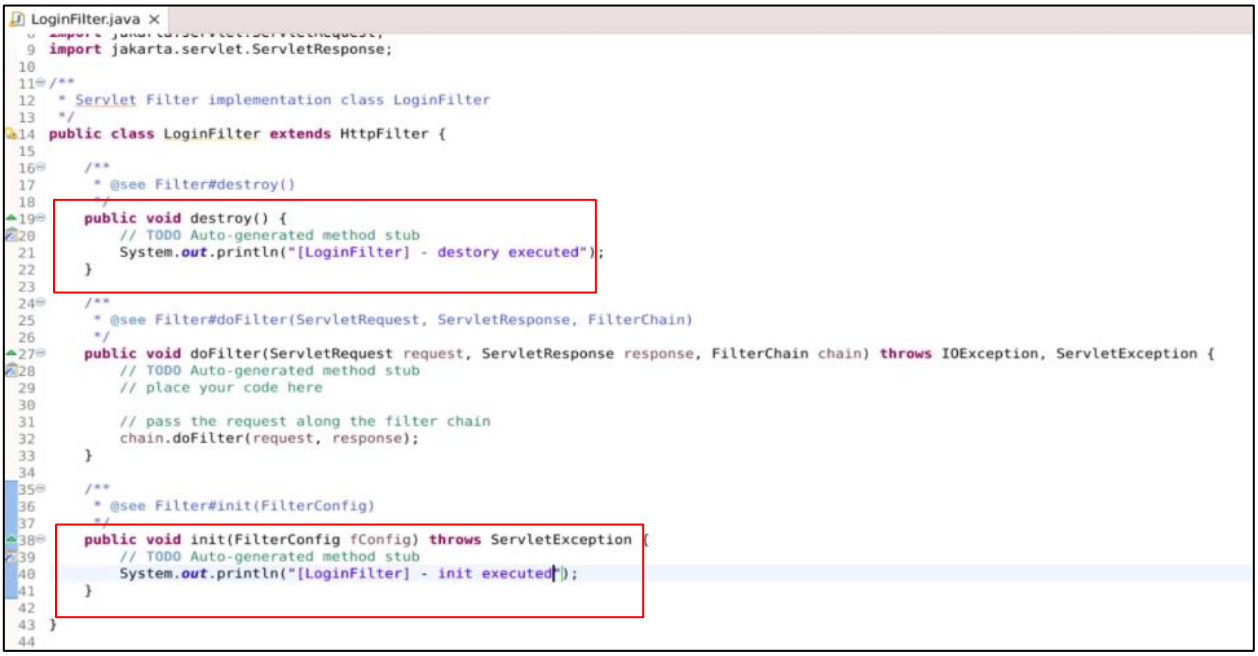
import.jakarta.servlet.FilterChain;
import.jakarta.servlet.FilterConfig;
import.jakarta.servlet.ServletException;
import.jakarta.servlet.ServletRequest;
import.jakarta.servlet.ServletResponse;



2.2 Specify the **LoginFilte**r and include the execution of the destroy and init methods, you can modify the code as follows:

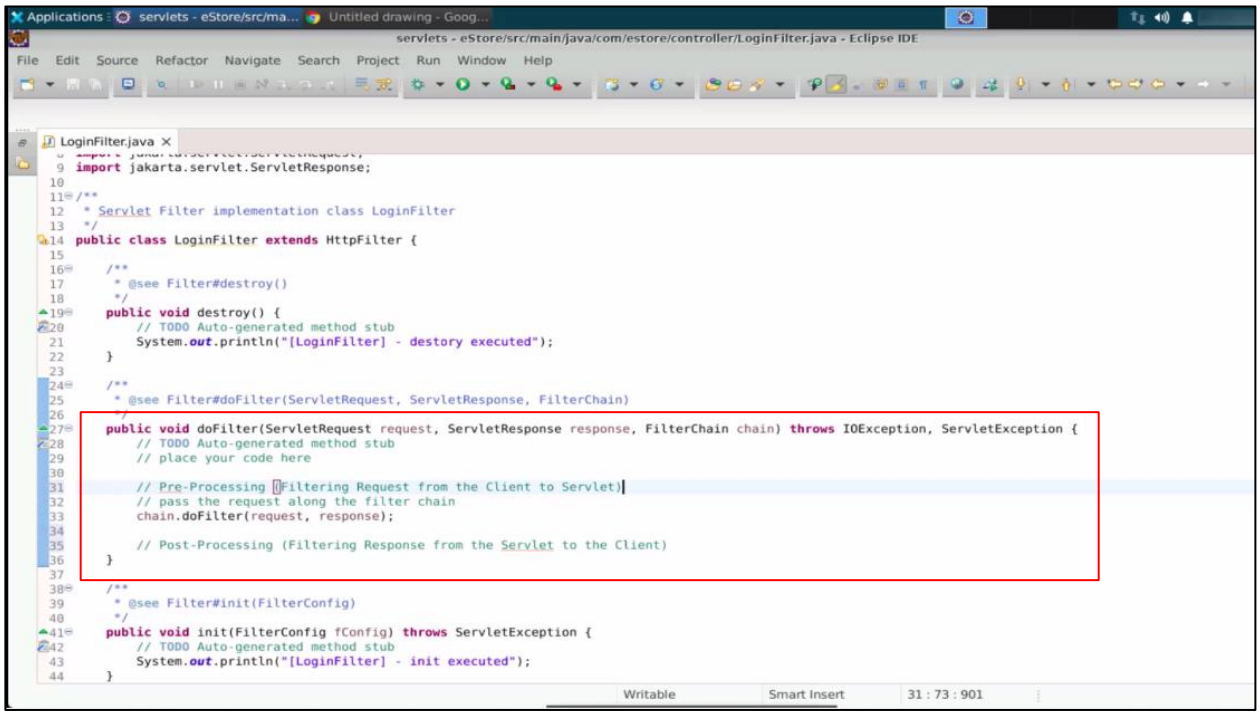
System.out.println (“[LoginFilter] – destroy executed”);

System.out.println (“[LoginFilter] – init executed”);

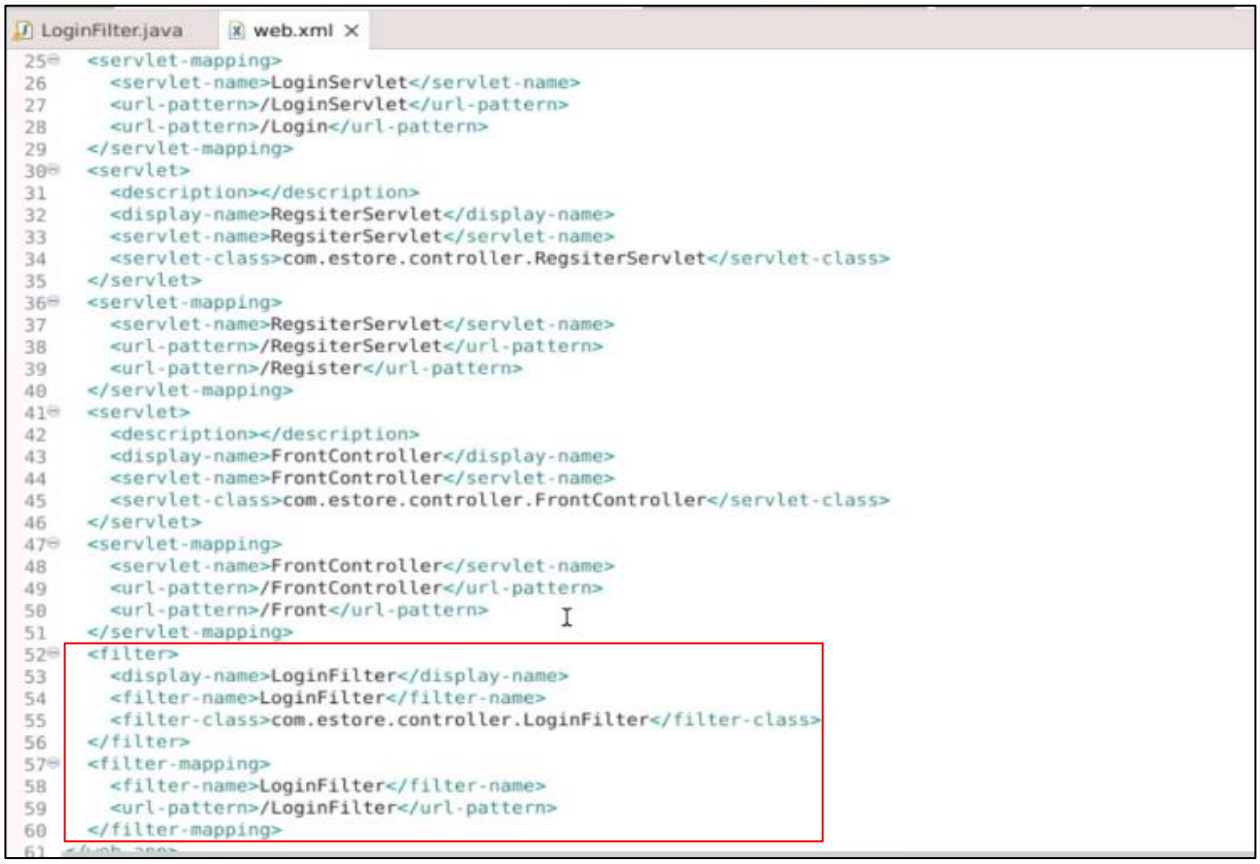


Step 3: Work with a filter mechanism

3.1 Return to the **LoginServlet.java** file and specify the pre-processing and post-processing of data

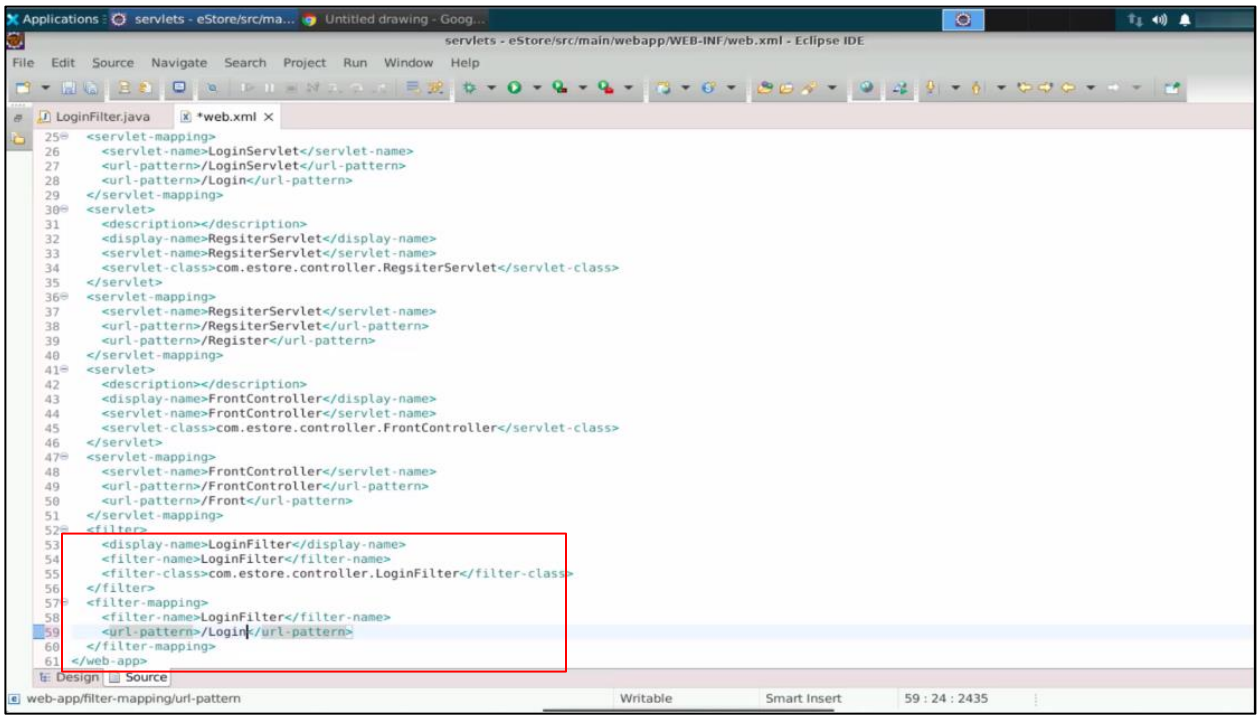


3.2 Open the **web.xml** file where you will find the login filter and filter mapping that has been created with the name **LoginFilter**



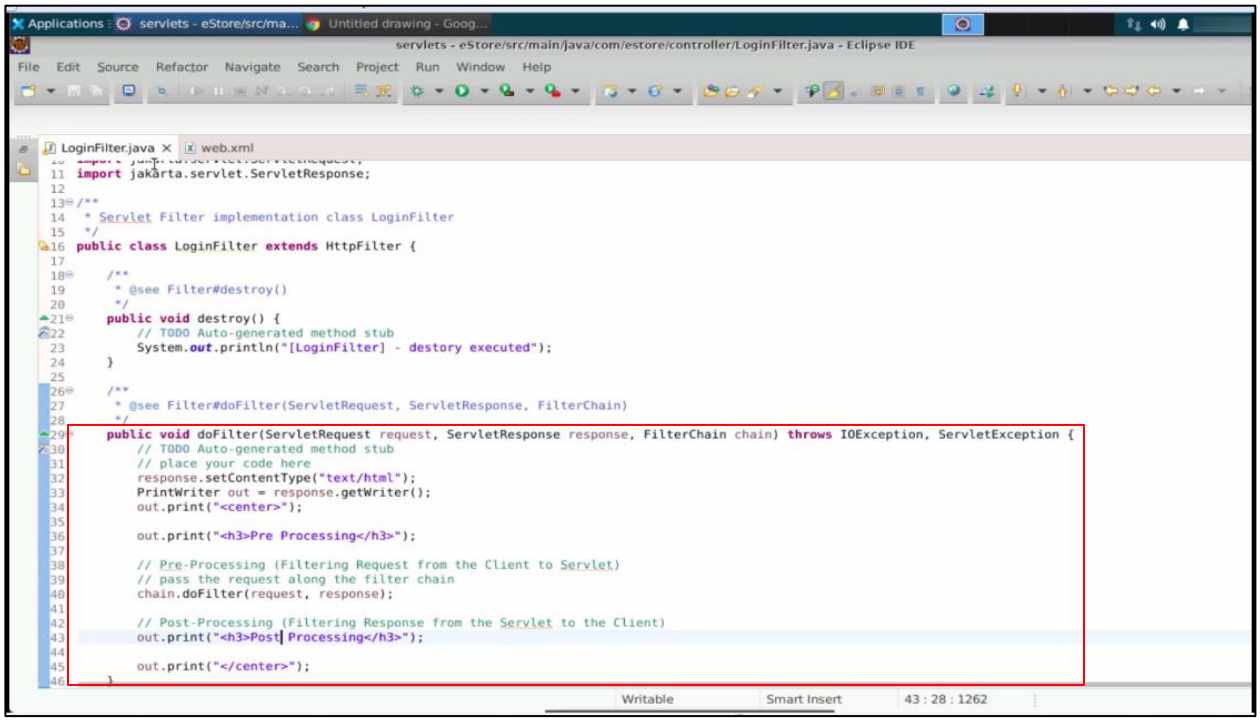
Step 4: Work on the login filter's response

4.1 Change the value of the url-pattern from **/LoginFilter** to **/Login**

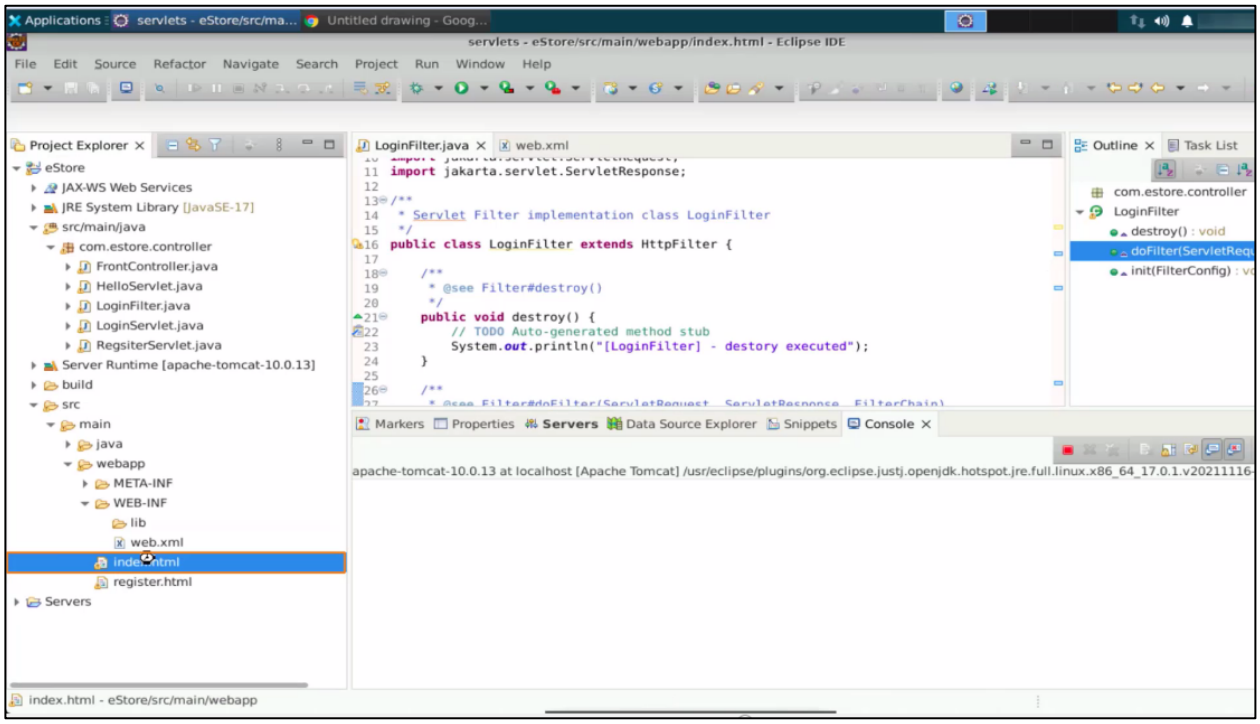


This is because the URL pattern specified for the filter will match that of the servlet. As a result, the request will be intercepted by the filter before it reaches the servlet directly.

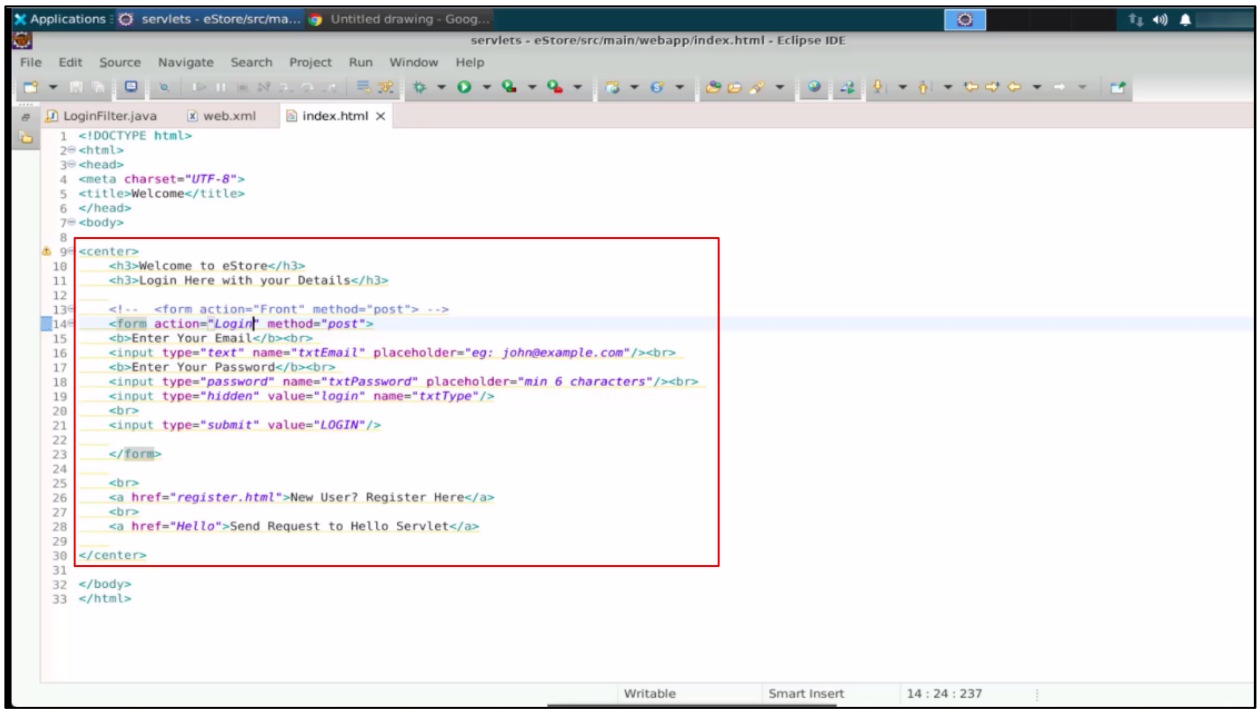
4.2 In **LoginFilter.java**, modify the response to include pre-processing and post-processing messages using the **setContentType()** method and **<h3>** tags



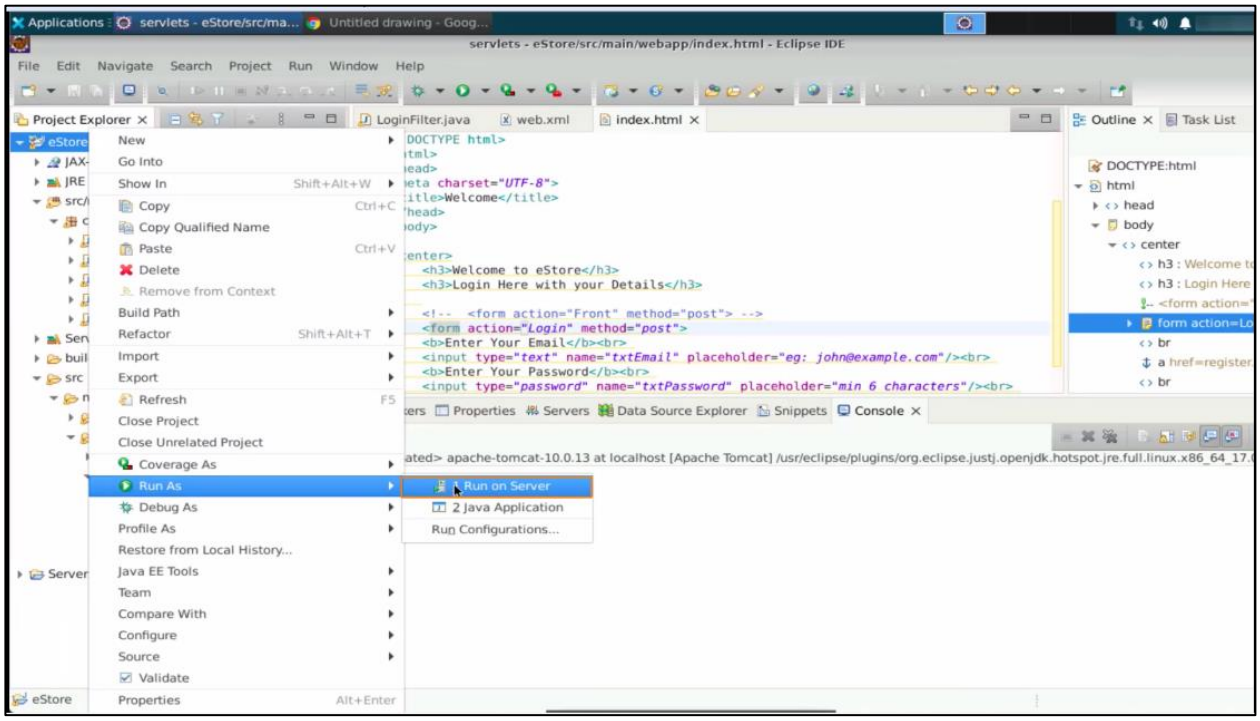
4.3 Open the **index.html** file



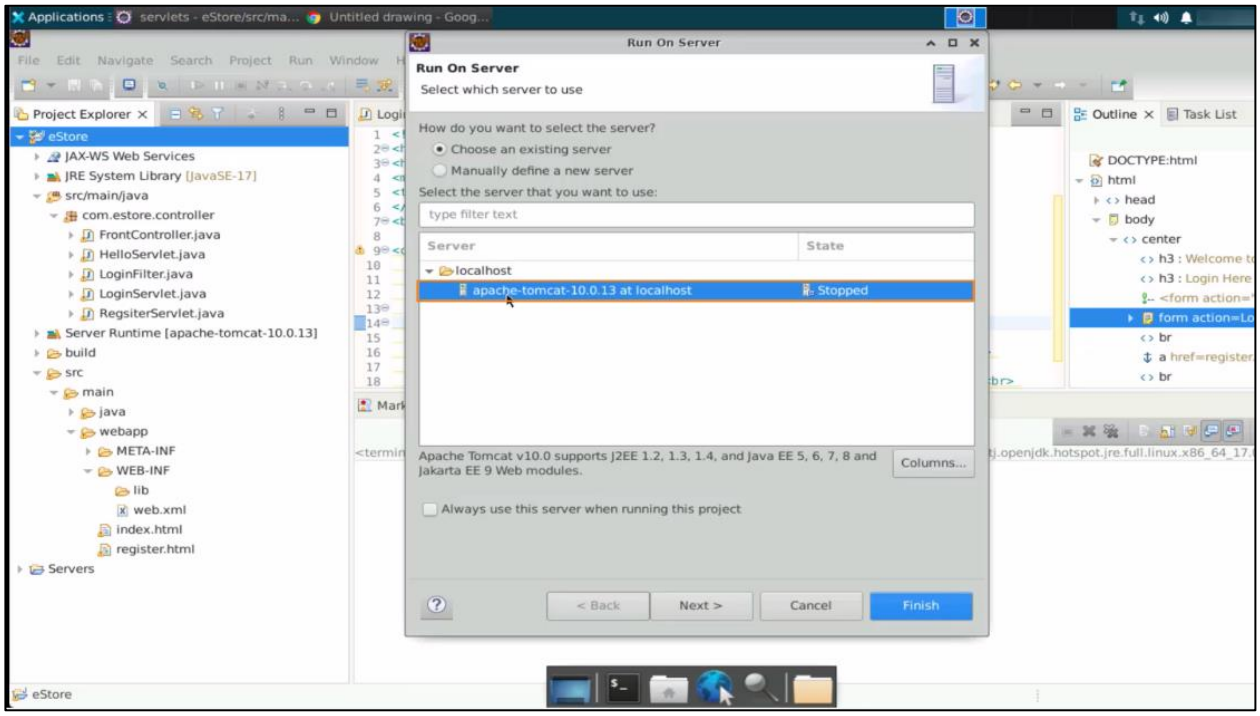
4.4 In the **index.html** file, change the action to **Login** and the method to **post** (13 and 14)



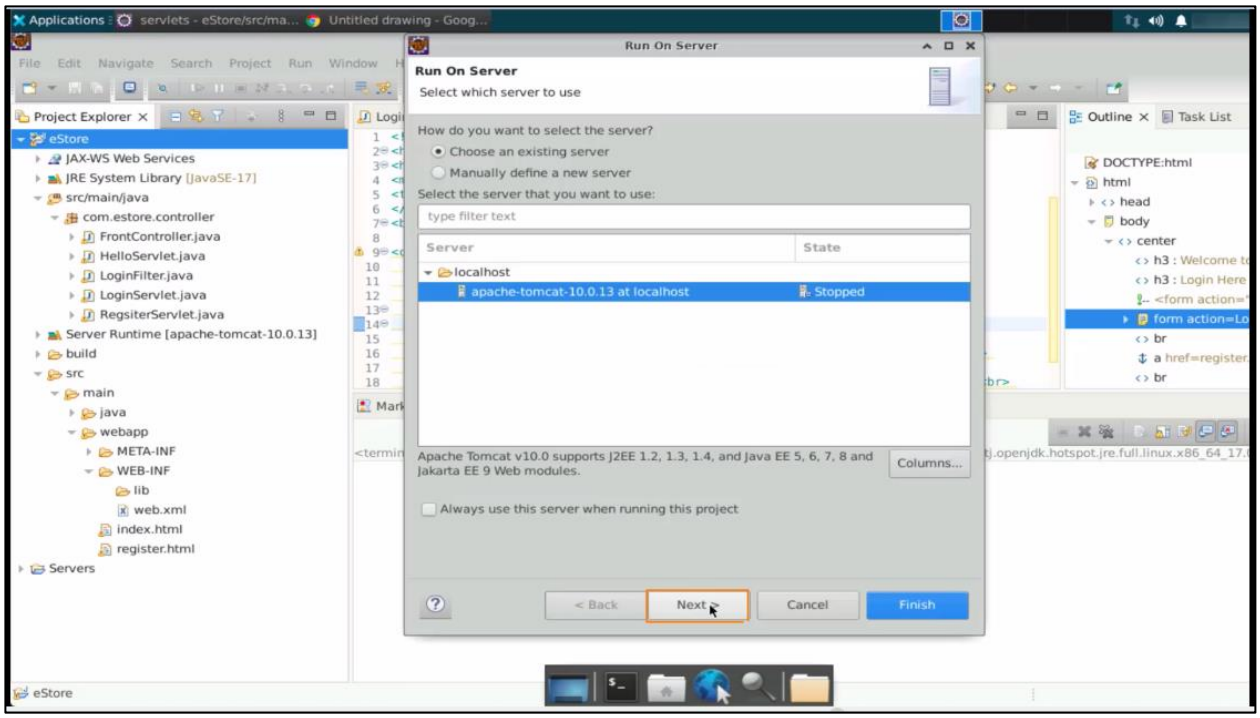
4.5 Navigate to the **estore** project, right-click on it, select **Run As**, and click on **Run on Server**



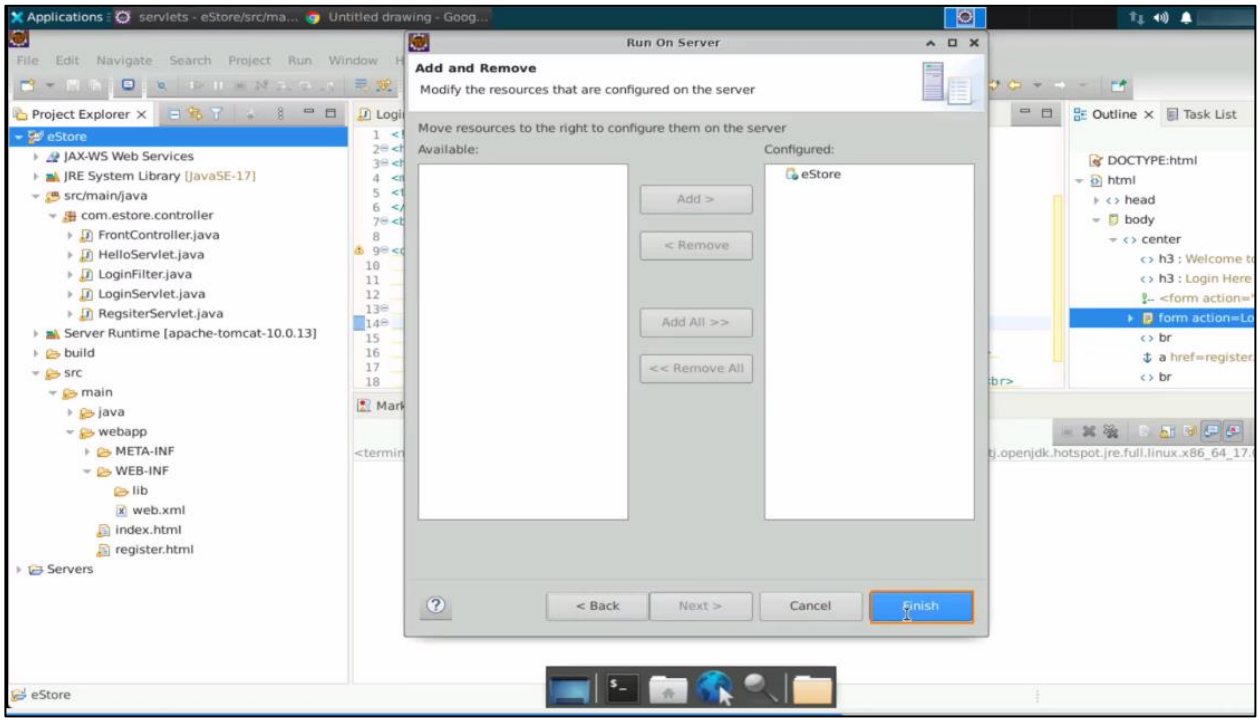
4.6 Select the **apache-tomcat-10.0.13** as the server



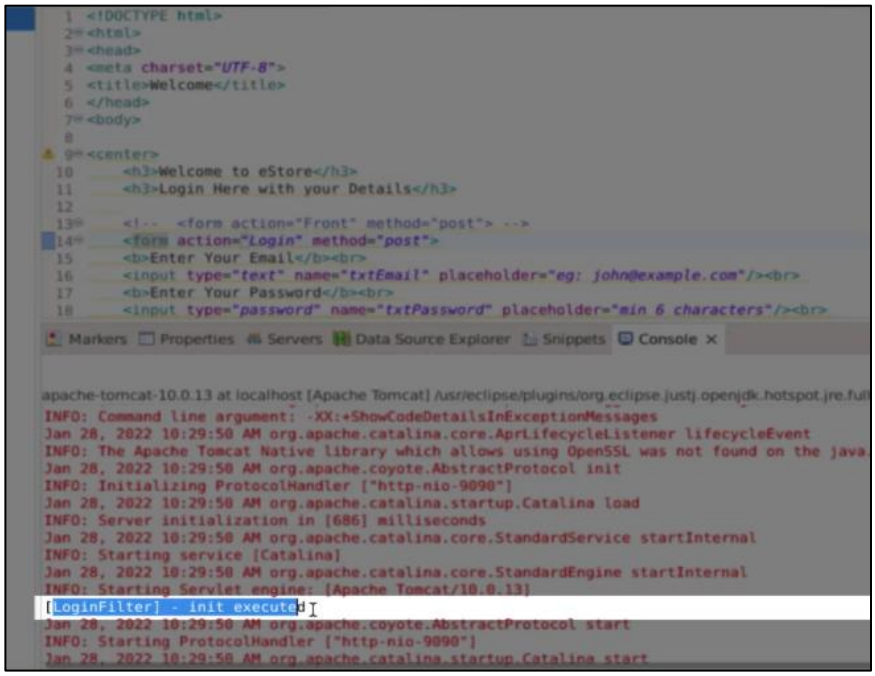
4.7 Click on **Next**



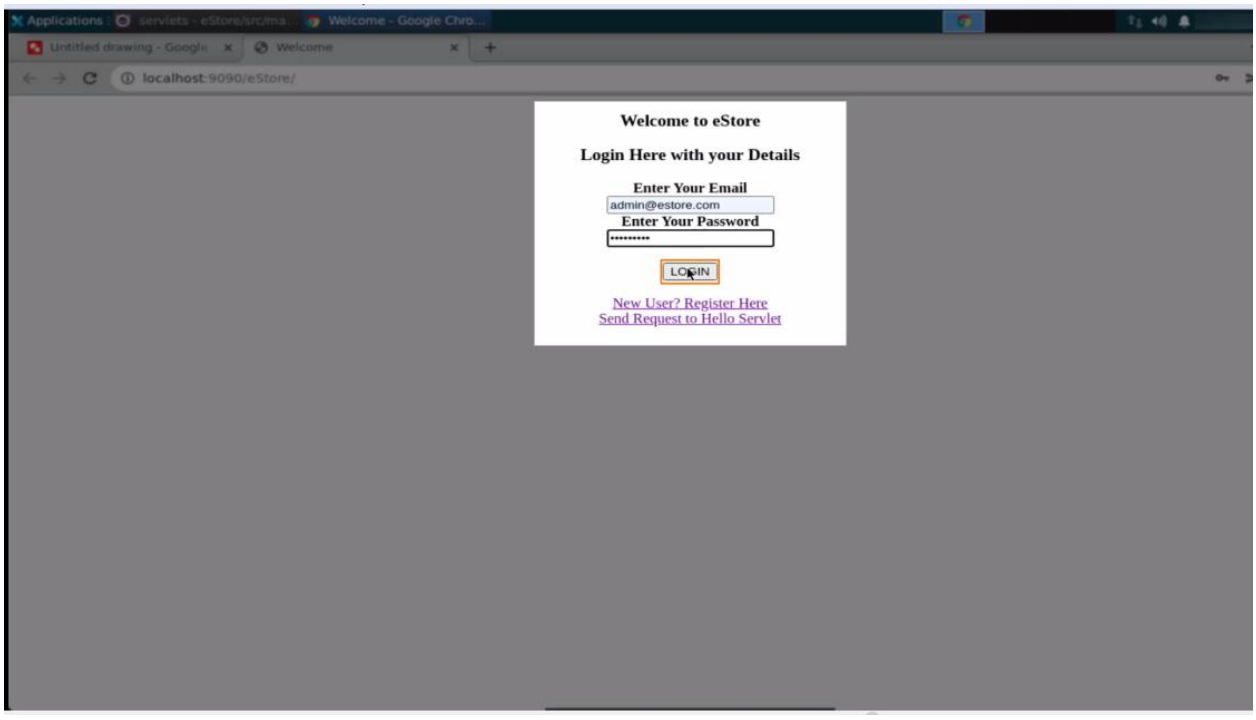
4.8 Click on **Finish**



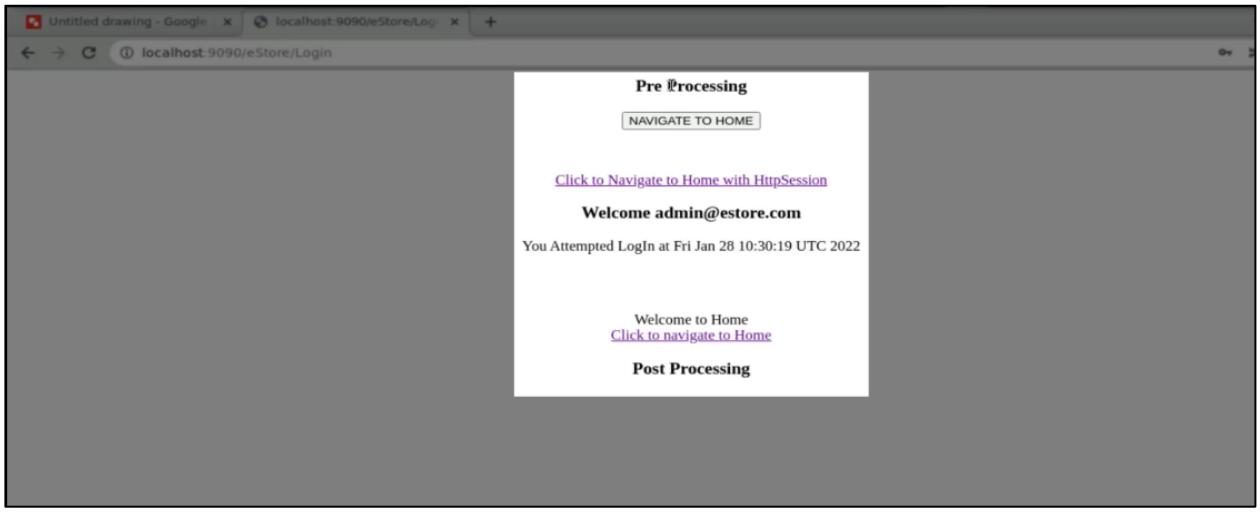
4.9 Navigate to the Eclipse Console tab where the message **[LoginFilter] – init executed** indicates the successful initialization of the **LoginFilter**



4.10 Enter the login details and click on **LOGIN**



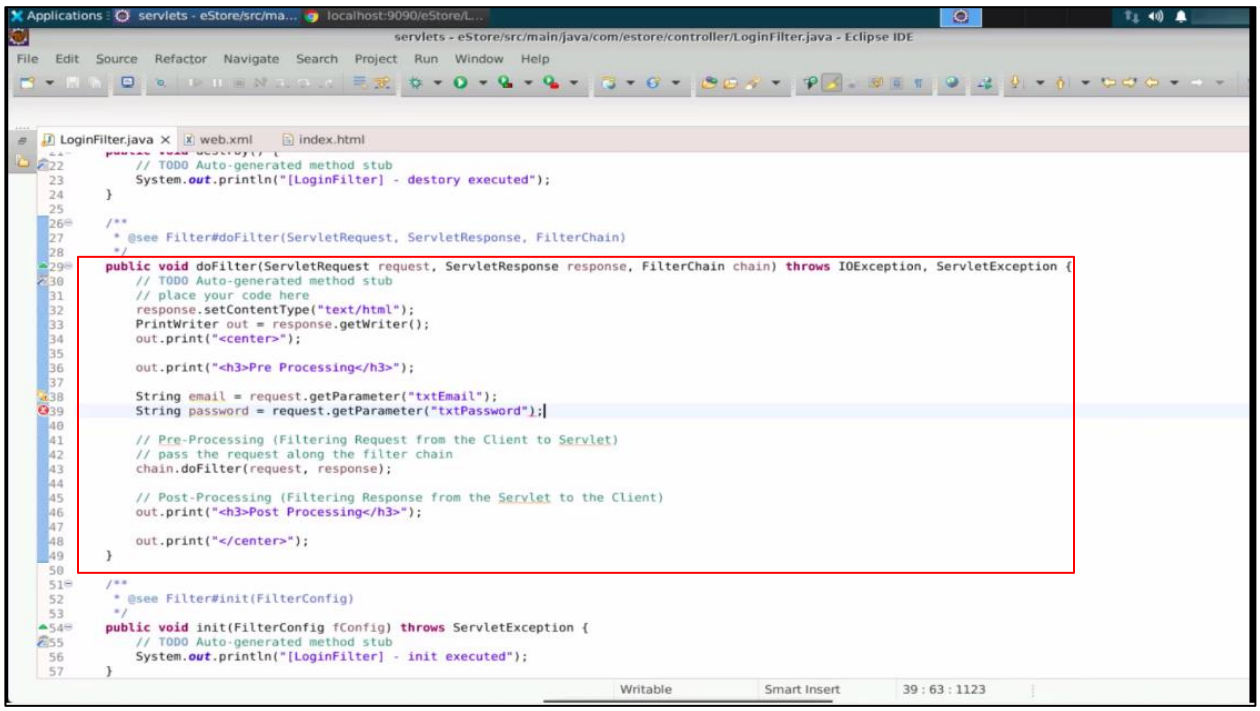
You can see the output as **Pre-processing** and **Post-processing**.



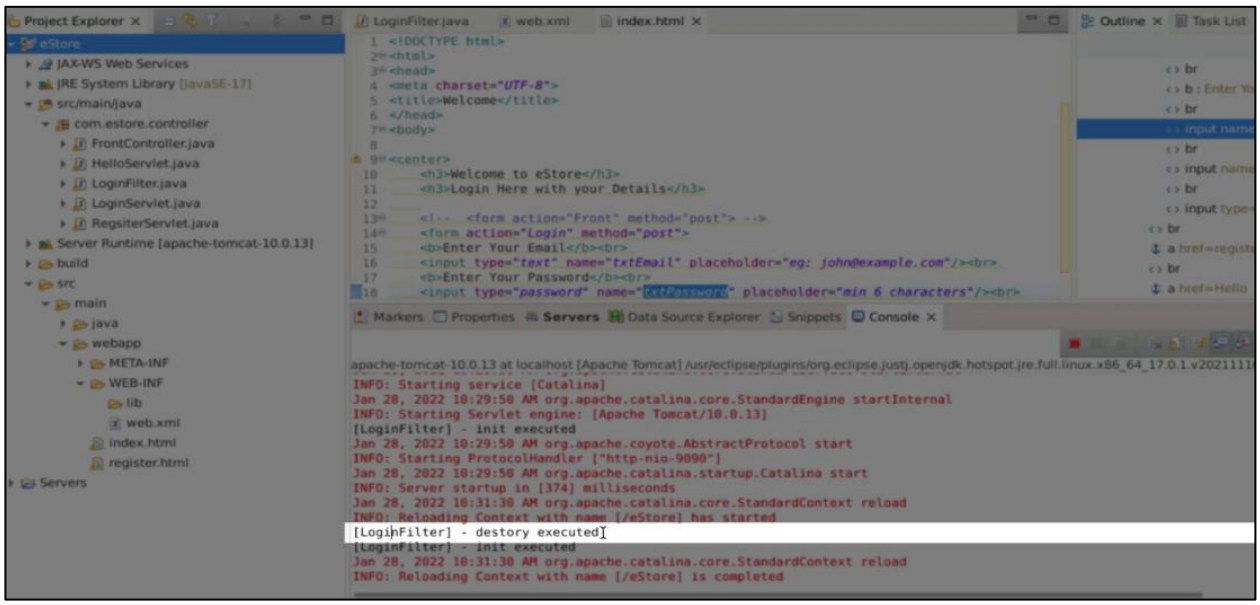
This means that the request is sent from the filter to the Servlet, and the response is sent from the Servlet to the filter.

Step 5: Make the filter work for login parameters

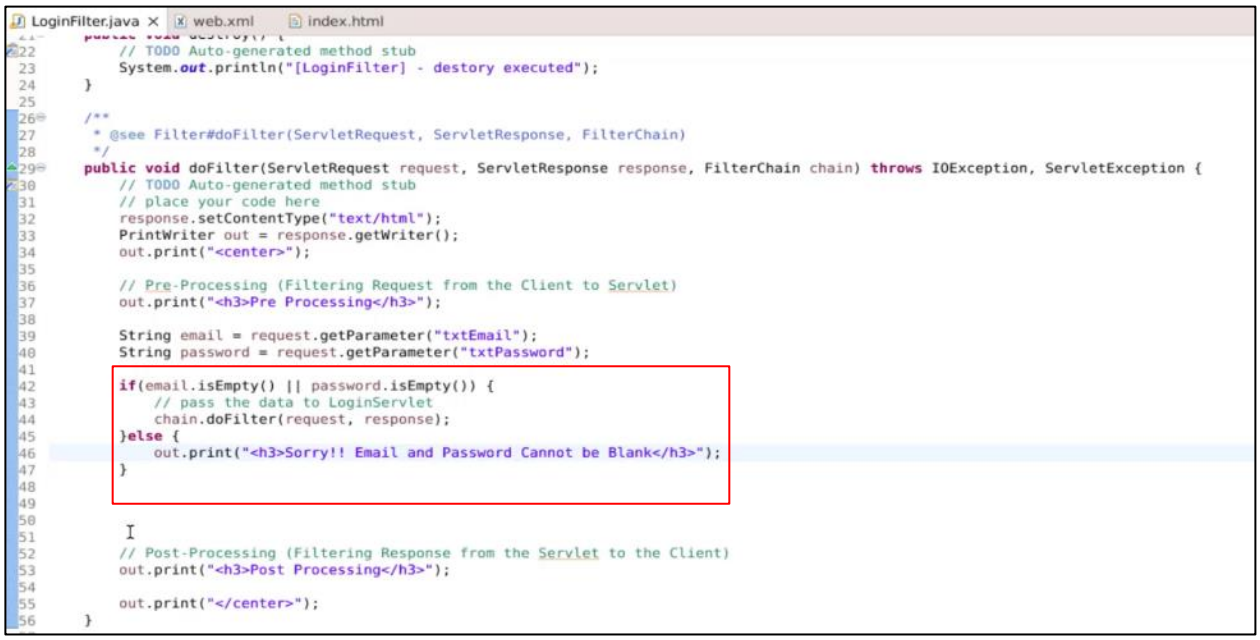
5.1 Navigate to **LoginFilter.java** and extract the email and the password from the request using the **request.getParameter()** method (lines 38 and 39)



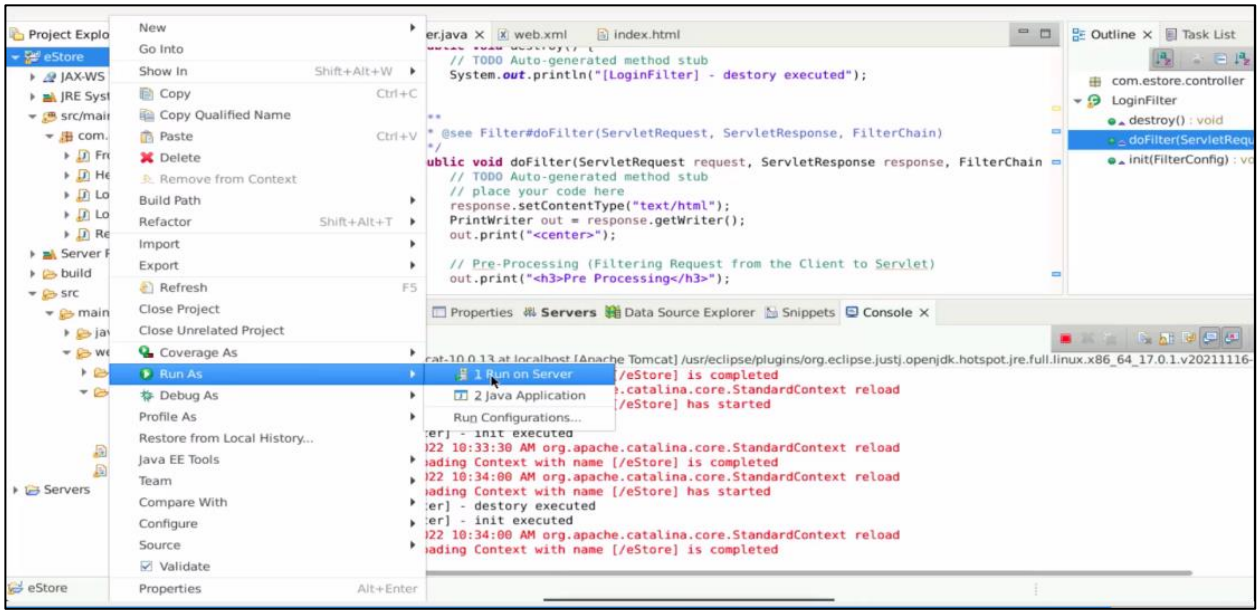
5.2 Navigate to the console logs where you will see that the filter has been destroyed



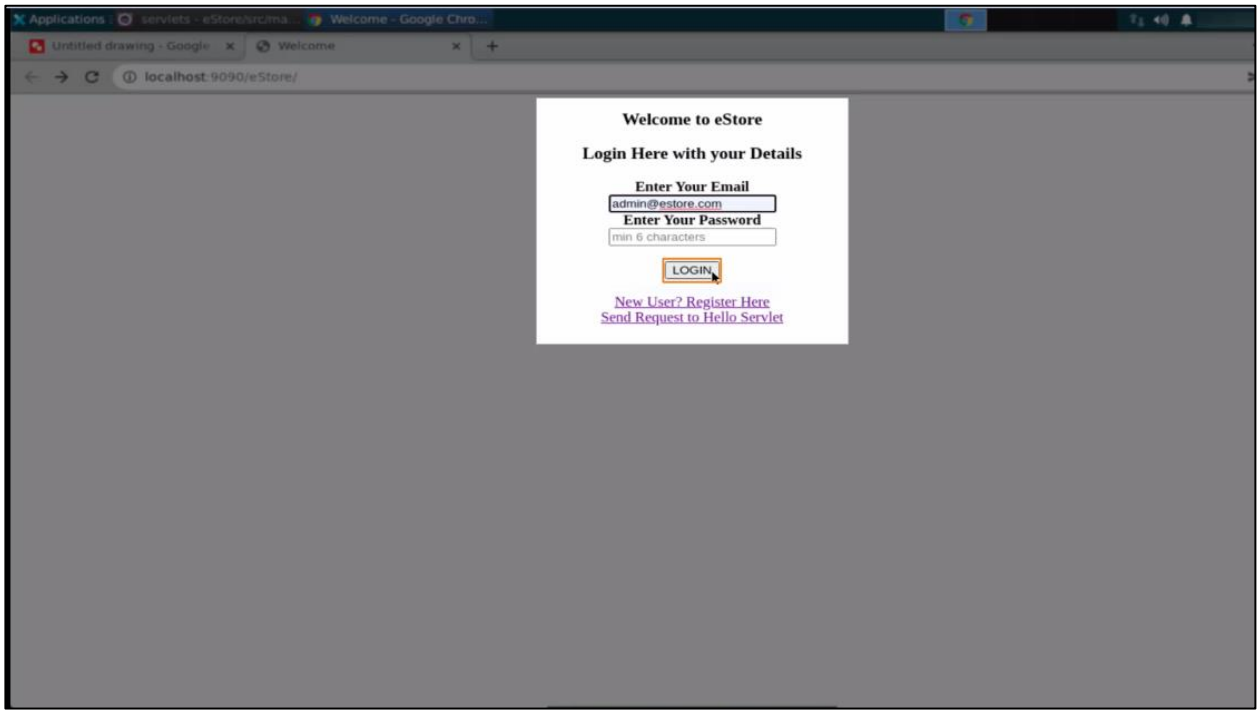
5.3 For general pre-processing, check if the email or password is empty by using the **isEmpty()** function and use the **doFilter()** method to perform the filtering in pre-processing



5.4 Next, rerun the code by right-clicking on the **eStore** project, selecting **Run As**, and clicking on **Run on Server**

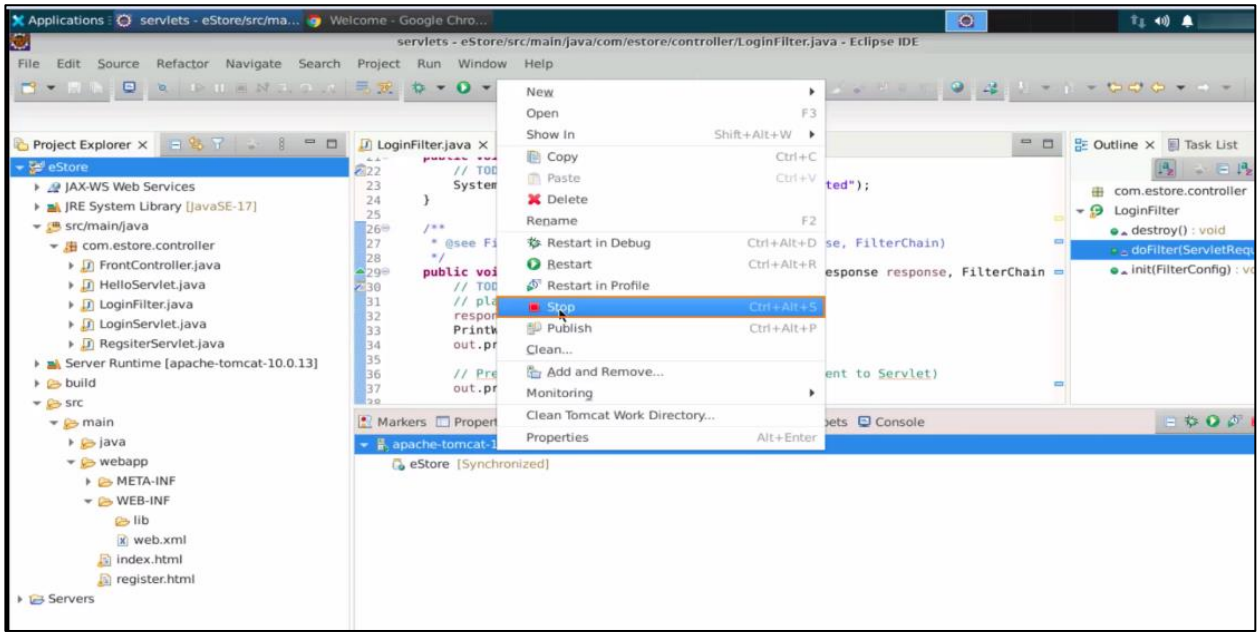


5.5 Refresh the page and enter only the email, and click on **LOGIN**

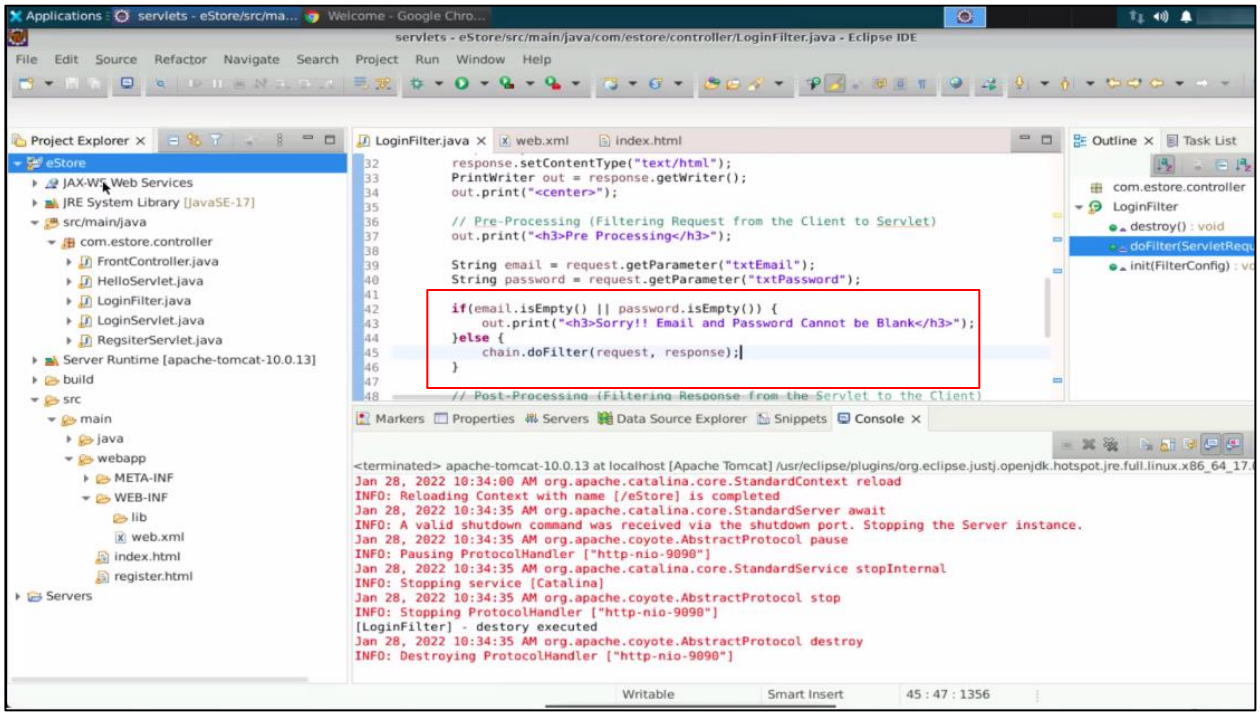


You can see that the server is not yet initialized.

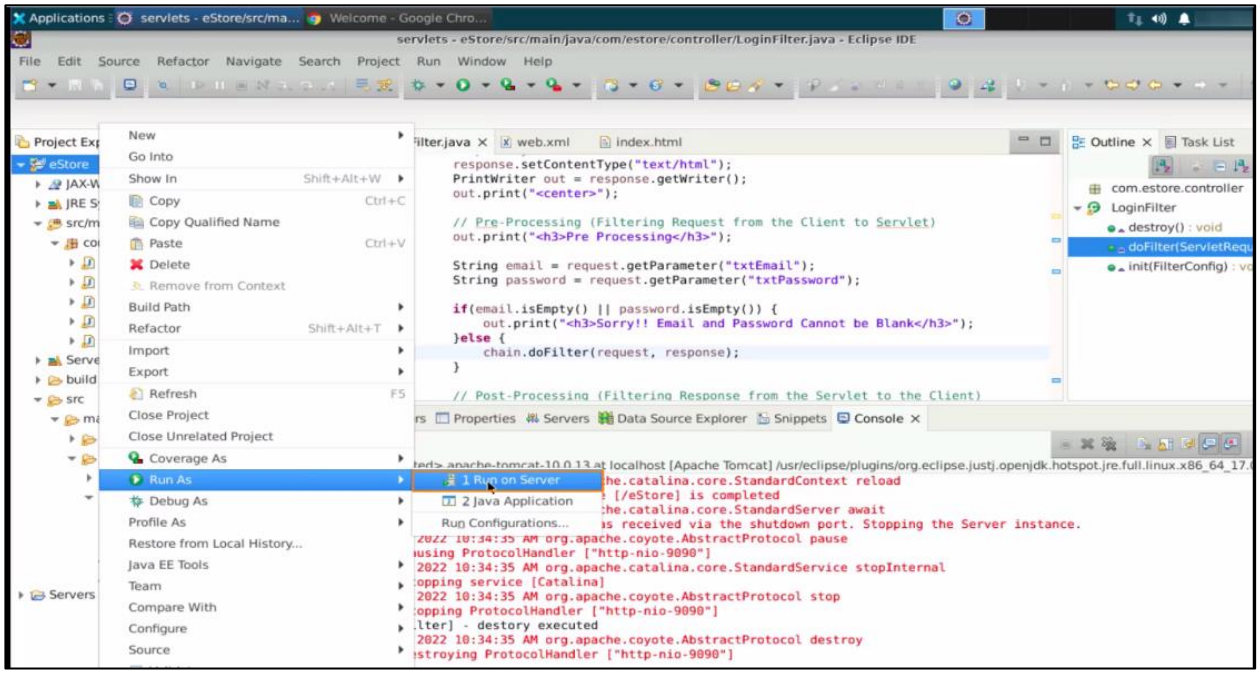
5.6 Go back and stop the server by clicking on **Stop**



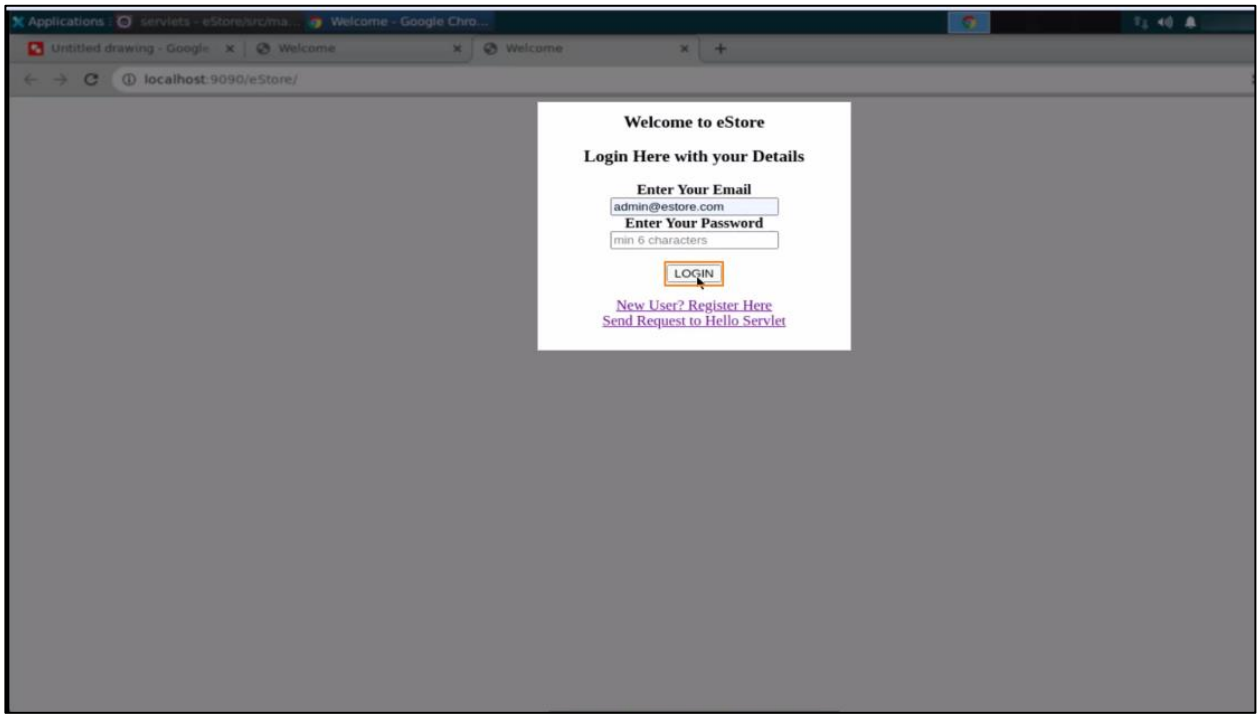
5.7 If the email or password is blank, filter the request and the response using the **chain.doFilter()** method



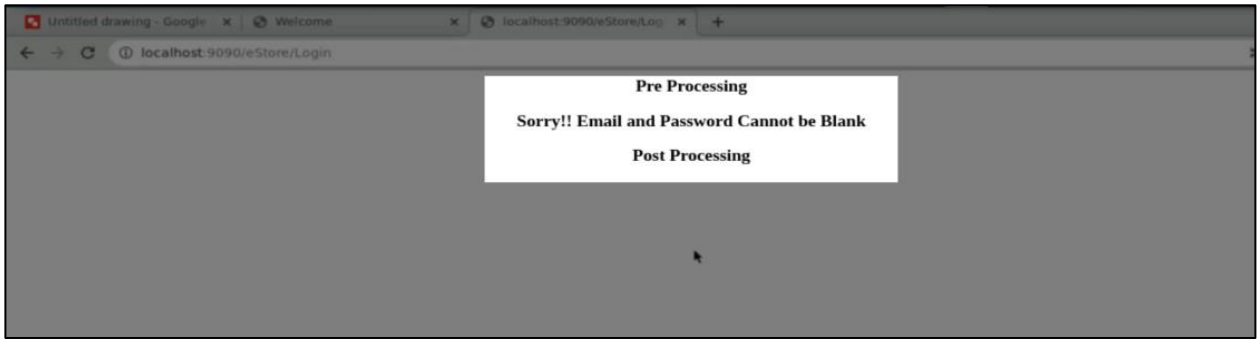
5.8 Right-click on **eStore**, select **Run As**, and **Run on Server**



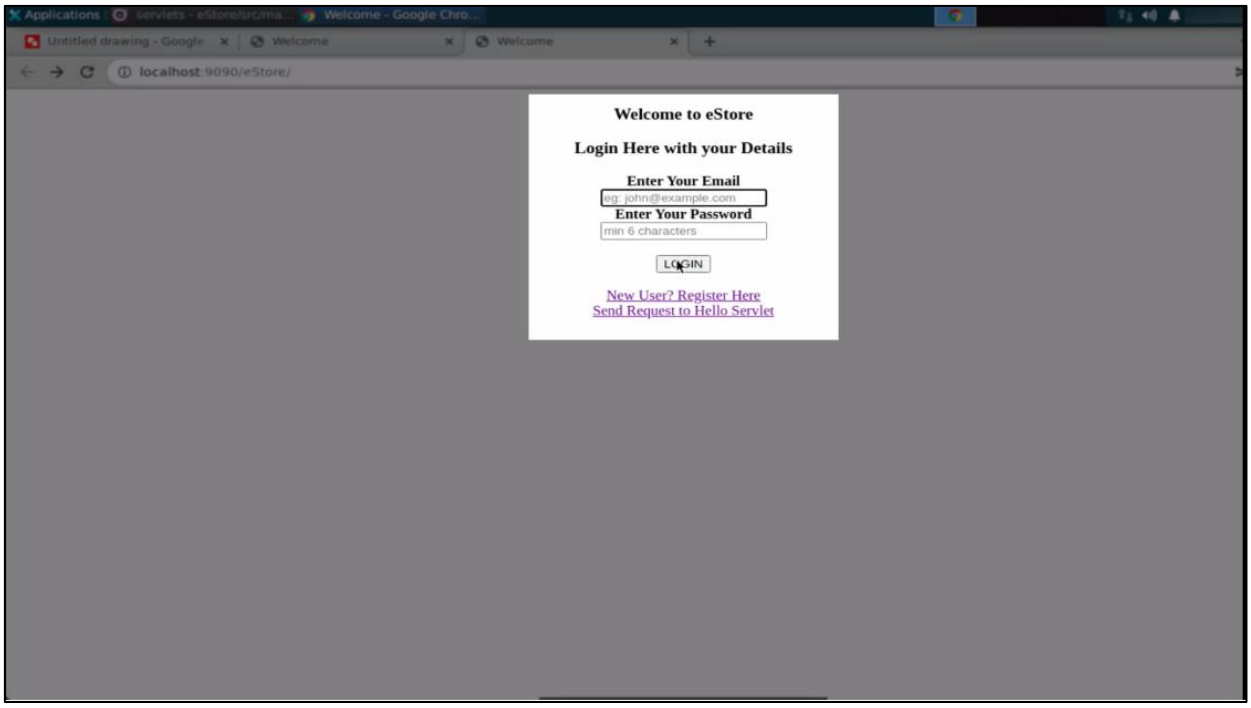
5.9 Enter the email and click on **LOGIN**



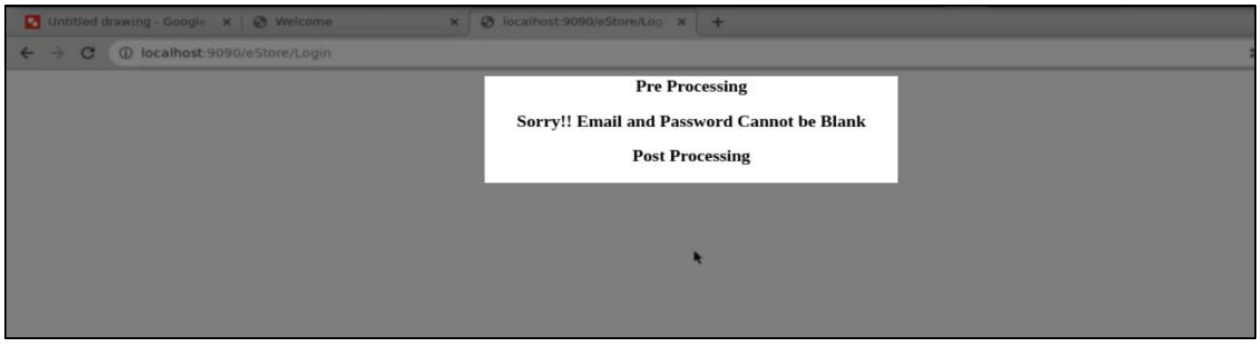
You will see the output: **Sorry!! Email and Password Cannot be Blank**. This means that the filter is working.



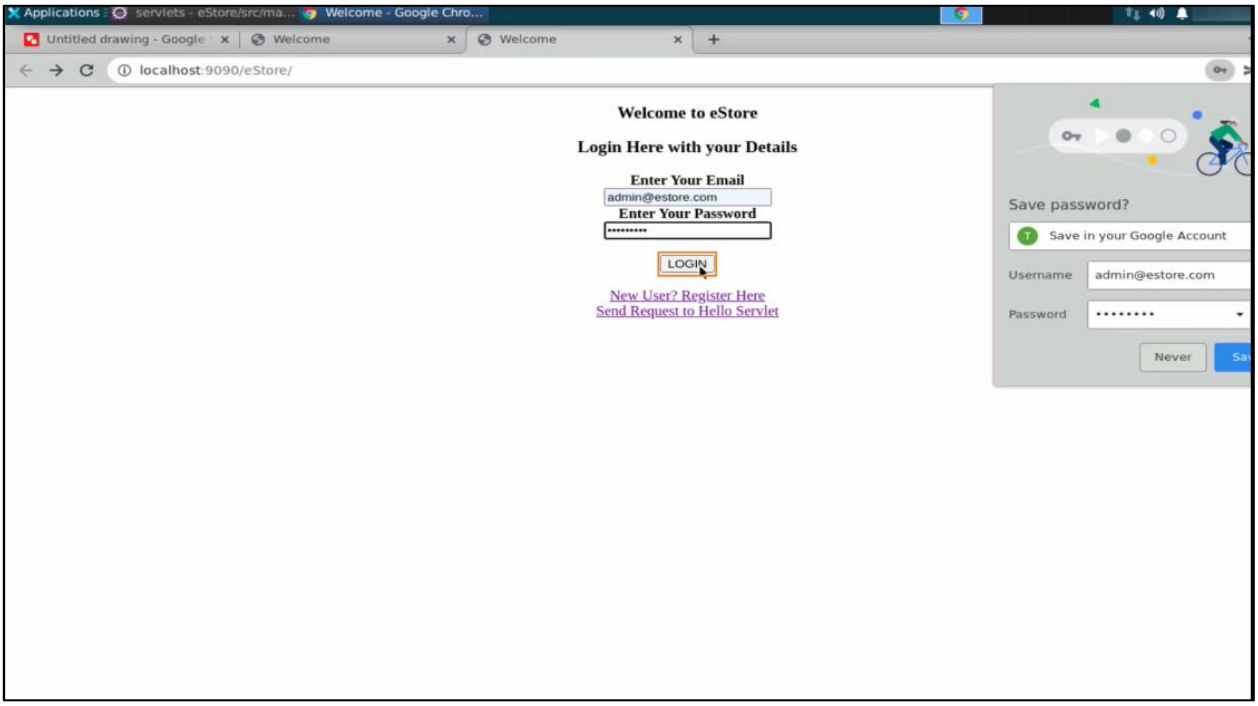
5.10 Keep both the text boxes empty, and click on **LOGIN**



The filter will check for both the username and password, and if either of them is empty, the same output will be displayed.



5.11 Enter the username and password, and click on **LOGIN**



Observation: You can observe the output as **Welcome admin@estore.com**, indicating that the filter has successfully forwarded the request to the Servlet and returned the response to the client.

Following these steps, you have successfully created a new filter and observed the login filter's response, pre-processing, and post-processing.