

Lesson 01 Demo 03

Service doGet() and doPost() in Servlet

Objective: To explore the doGet() and doPost() services in Servlet

Tools Required: Eclipse IDE

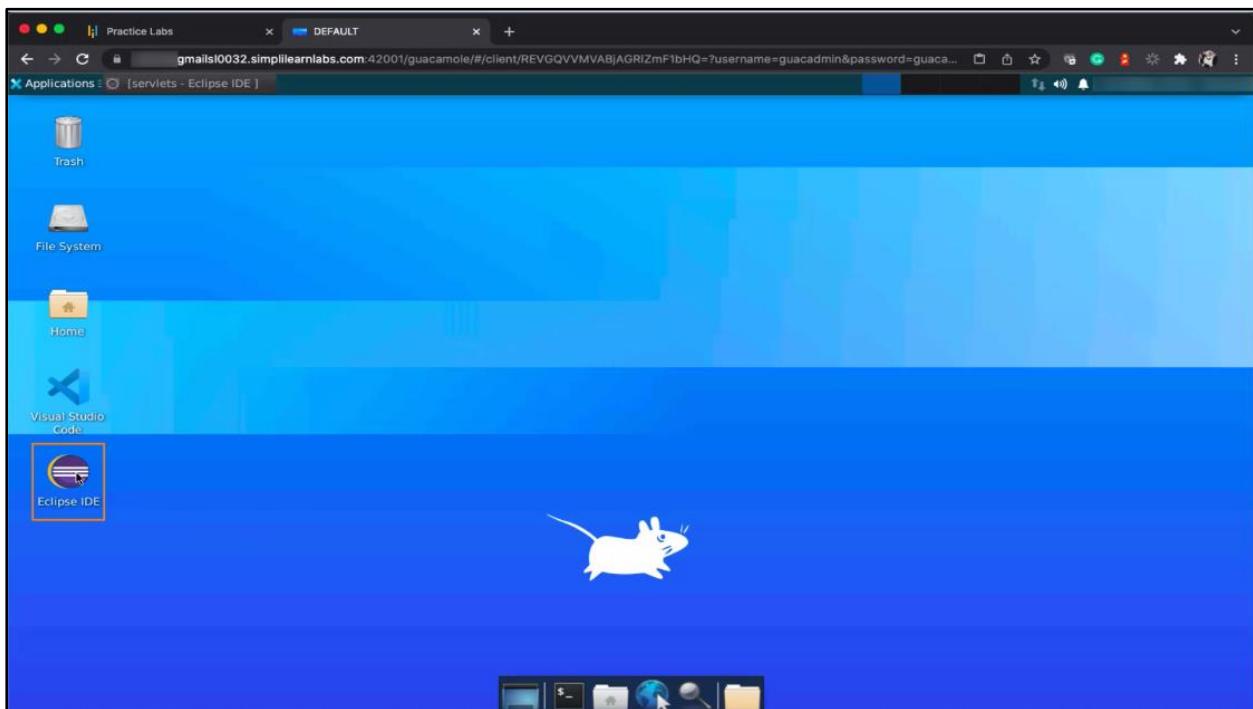
Prerequisites: None

Steps to be followed:

1. Perform the HTTP POST method
2. Perform the HTTP GET method
3. Create a doGet() method
4. Create a doPost() method

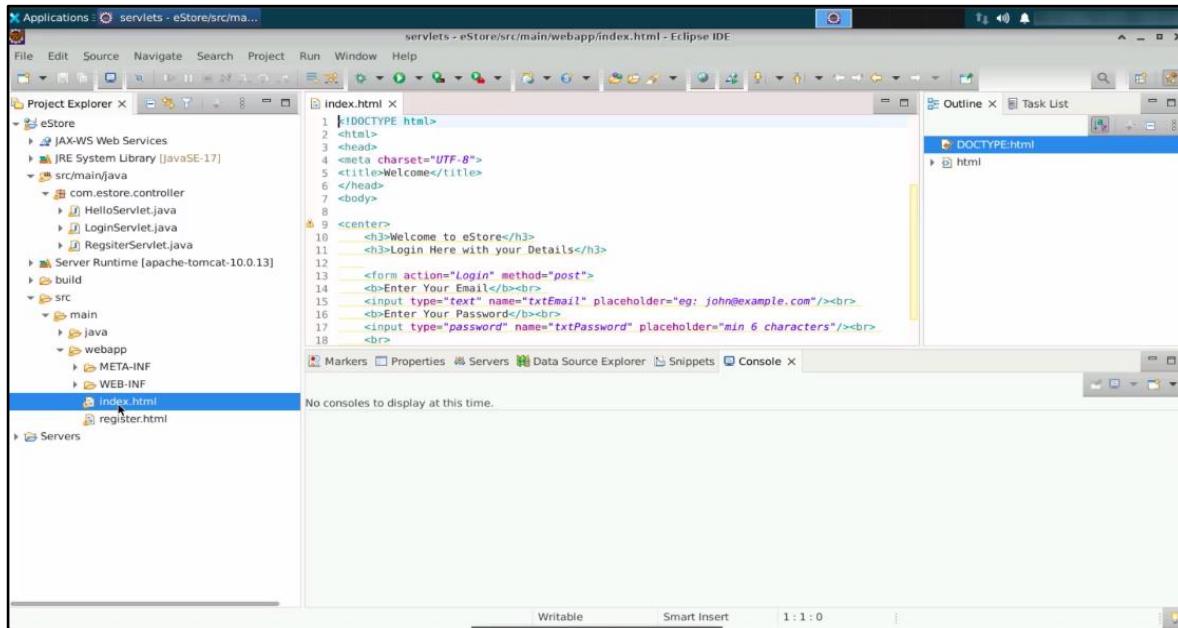
Step 1: Perform the HTTP POST method

1.1 Open Eclipse IDE

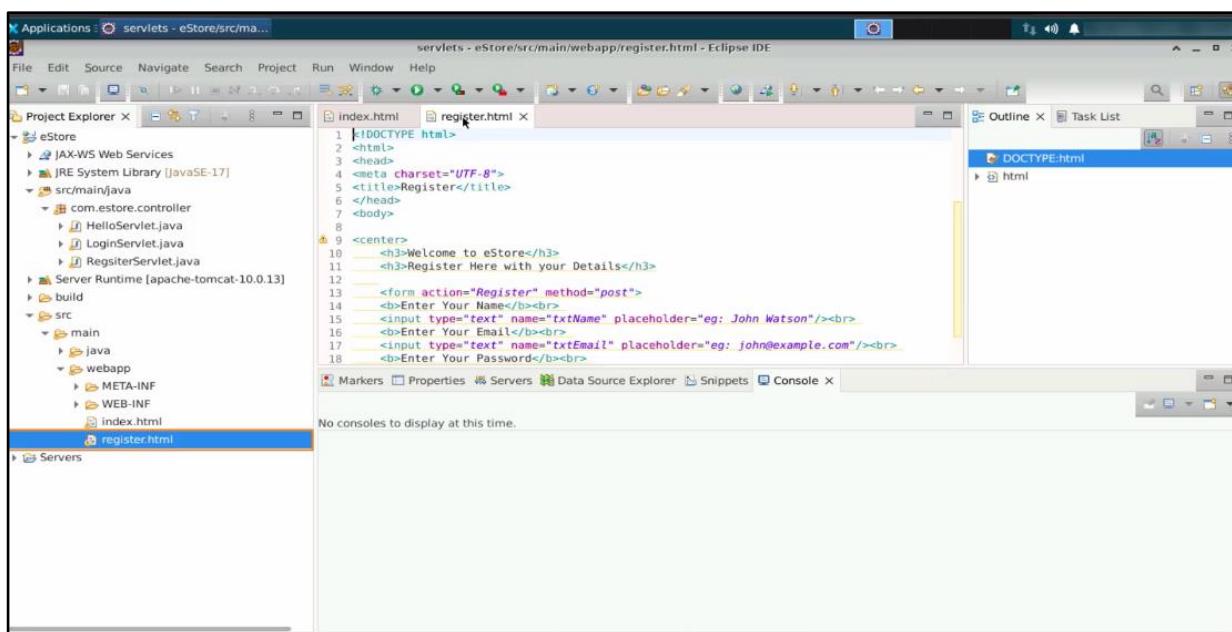


1.2 Open the **index.html** page inside the **webapp** folder.

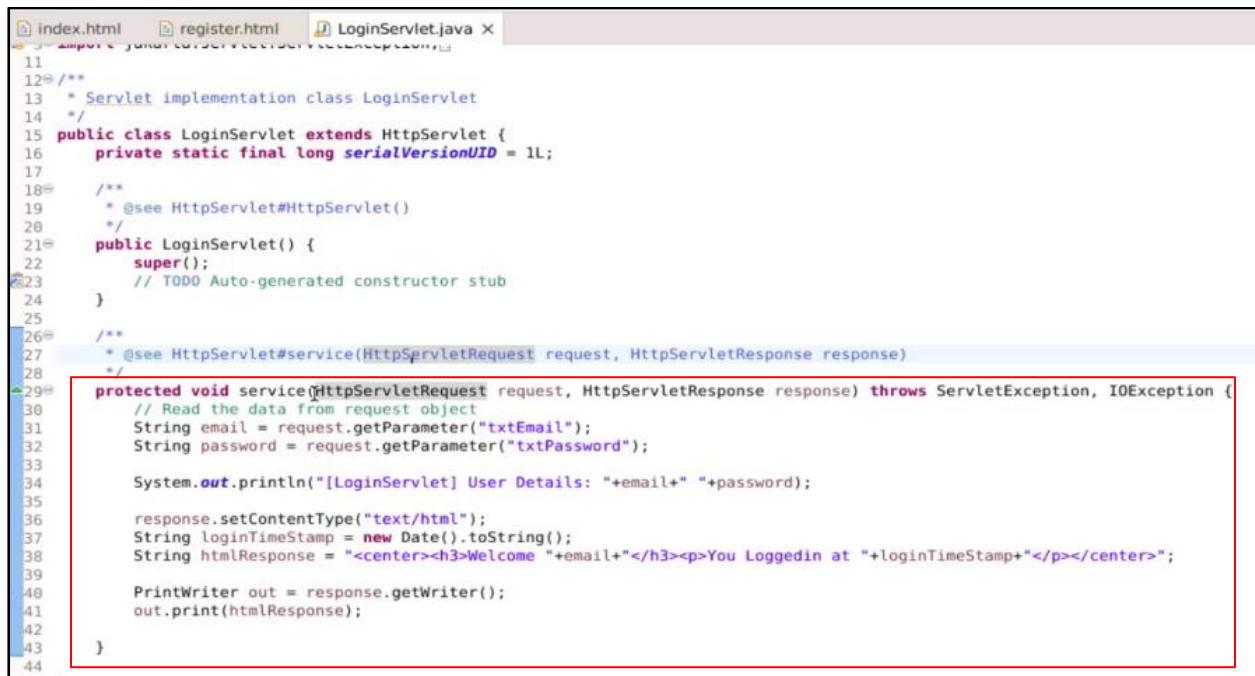
Note: eStore is a dynamic web project that has already been created in **Demo 02**.



1.3 Open the **register.html** file and find a registration form that takes the register action. It is like a Servlet which sends an email and the password.



1.4 Now, open the **LoginServlet.java** file and search for the **service** method. A Service method can handle **GET** and **POST** requests.

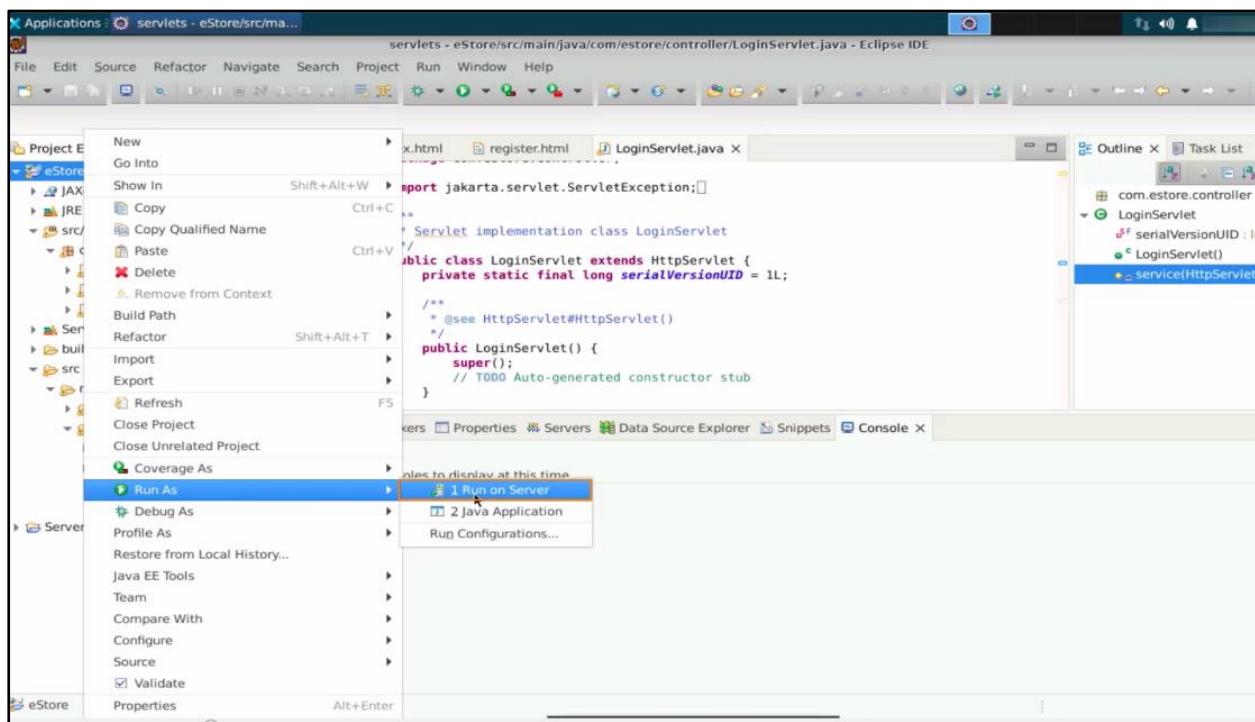


```

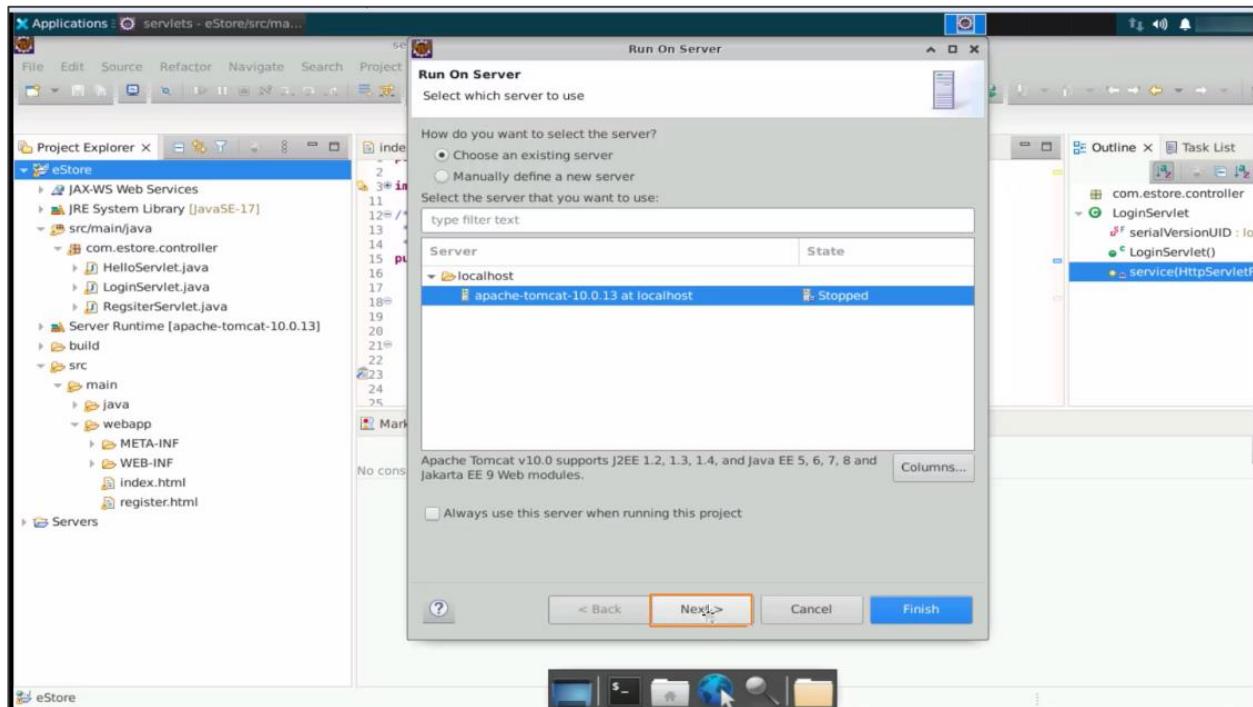
11
12 /**
13  * Servlet implementation class LoginServlet
14 */
15 public class LoginServlet extends HttpServlet {
16     private static final long serialVersionUID = 1L;
17
18     /**
19      * @see HttpServlet#HttpServlet()
20     */
21     public LoginServlet() {
22         super();
23         // TODO Auto-generated constructor stub
24     }
25
26     /**
27      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
28     */
29     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
30         // Read the data from request object
31         String email = request.getParameter("txtEmail");
32         String password = request.getParameter("txtPassword");
33
34         System.out.println("[LoginServlet] User Details: "+email+" "+password);
35
36         response.setContentType("text/html");
37         String loginTimeStamp = new Date().toString();
38         String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"</p></center>";
39
40         PrintWriter out = response.getWriter();
41         out.print(htmlResponse);
42     }
43 }
44

```

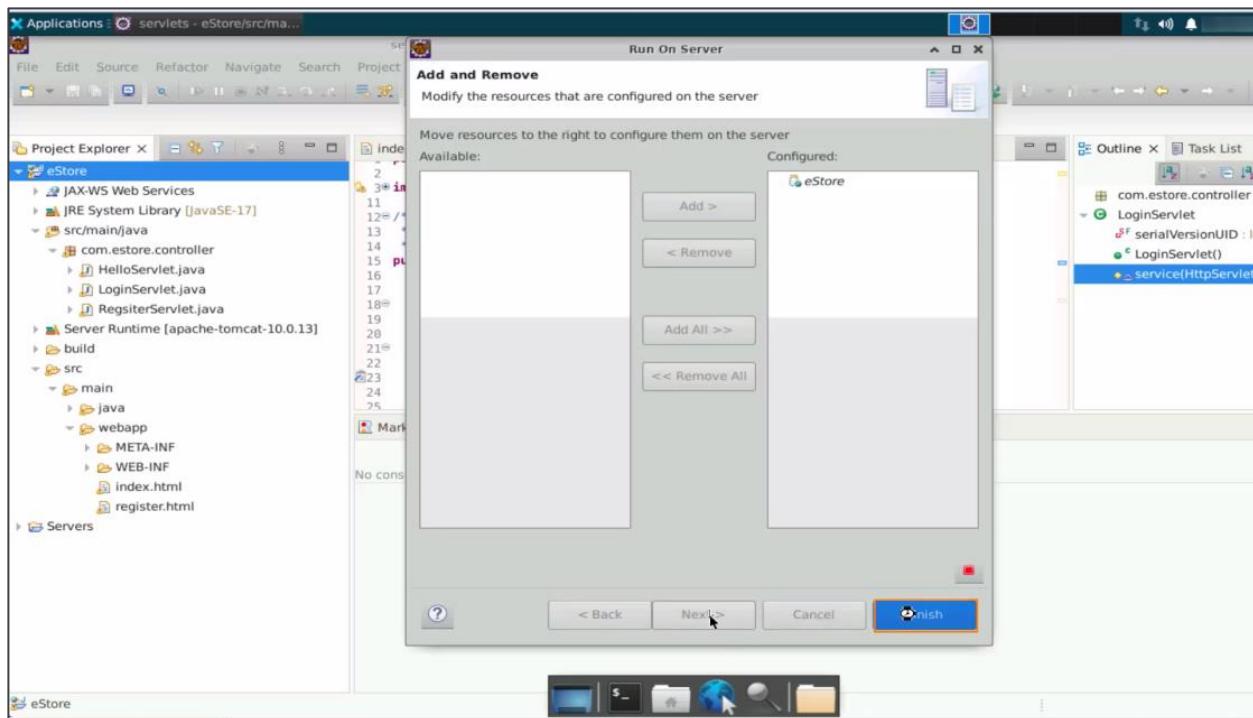
1.5 Run the **eStore** project. Right-click on **eStore**, select **Run As**, and click on **Run on Server**



1.6 Choose the server **apache-tomcat-10.0.13** and click **Next**



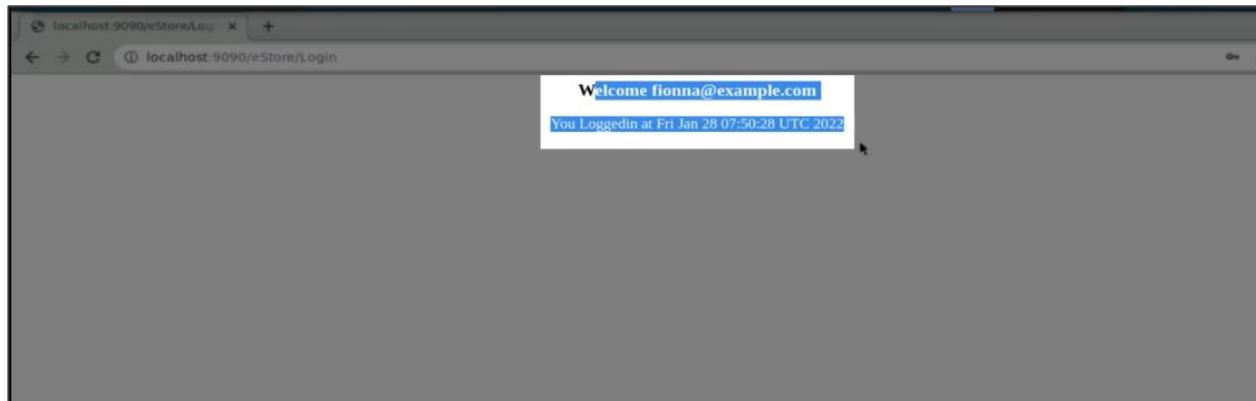
1.7 Click **Finish**



1.8 The login page would appear in the browser. Enter the information as required and click the **LOGIN** button



The below data is being shown, which says **Welcome fionna@example.com**.



Step 2: Performing the HTTP GET method

2.1 Go to the index.html page, change the method to get, and save it

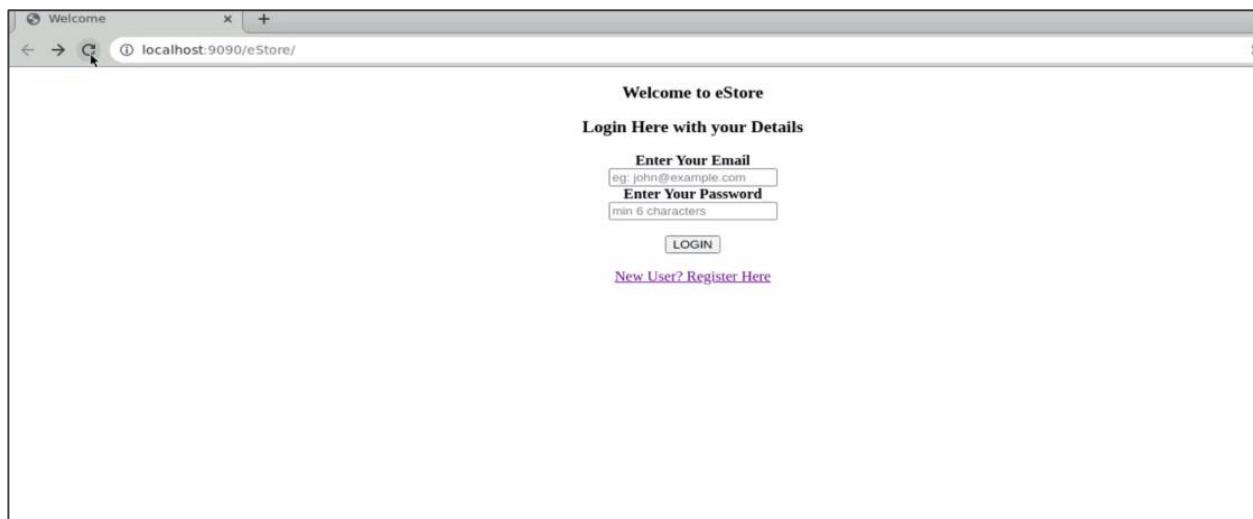
```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Welcome</title>
</head>
<body>
<center>
<h3>Welcome to eStore</h3>
<h3>Login Here with your Details</h3>
<form action="Login" method="get">
<b>Enter Your Email:</b><br>
<input type="text" name="txtEmail" placeholder="eg: john@example.com"/><br>
<b>Enter Your Password:</b><br>
<input type="password" name="txtPassword" placeholder="min 6 characters"/><br>
<br>

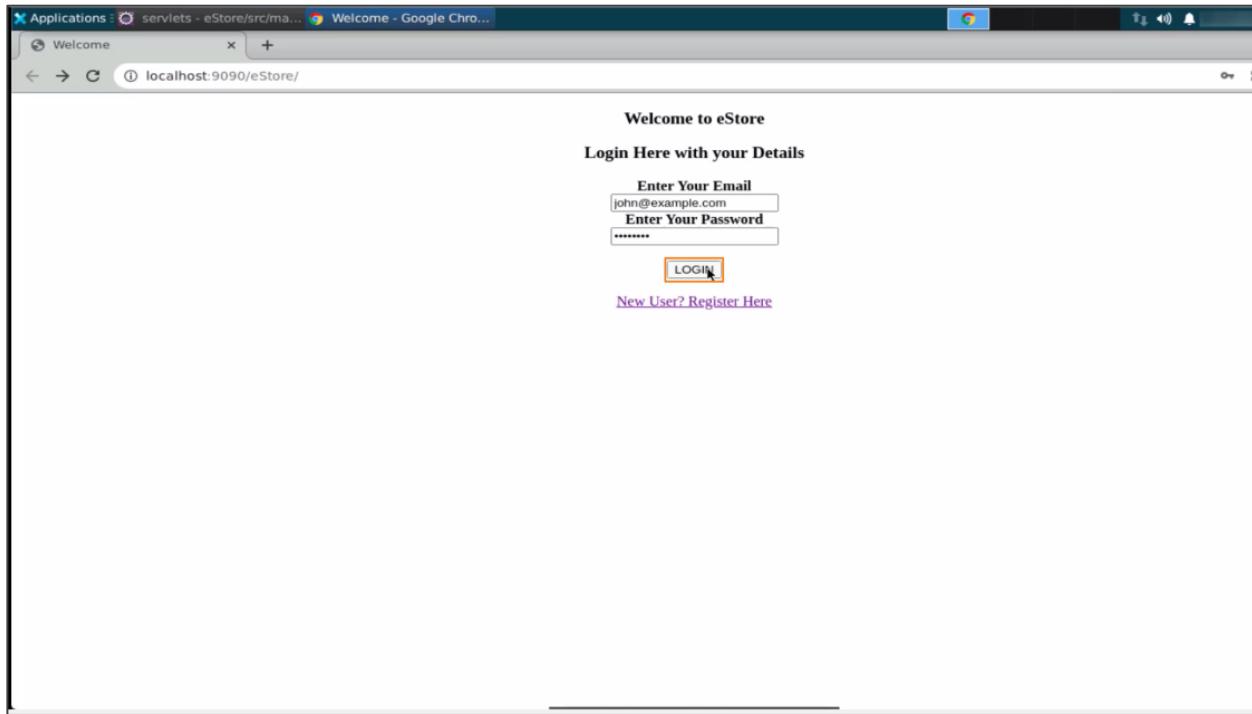
```

The screenshot shows the Eclipse IDE interface with the Project Explorer, Outline, and Task List. The code editor is open with the file index.html selected. The line of code `<form action="Login" method="get">` is highlighted in yellow, indicating it is being edited. The status bar at the bottom shows the file path as `html/body/center/form/method` and the current line as `13= <form action="Login" method="get">`.

2.2 Again, go to the browser and refresh the page



2.3 Enter the username as **john@example.com**, the password as **john@123**, and click **LOGIN**



We will now be able to view the URL data. Thus, a GET request is created, and the data becomes part of the URL. It is not recommended to use this for sensitive information.



2.4 To handle both the **GET** and the **POST** requests from the client, go to the **LoginServlet.java** page and add a comment for them

```

11
12 /**
13  * Servlet implementation class LoginServlet
14 */
15 public class LoginServlet extends HttpServlet {
16     private static final long serialVersionUID = 1L;
17
18     /**
19      * @see HttpServlet#HttpServlet()
20     */
21     public LoginServlet() {
22         super();
23         // TODO Auto-generated constructor stub
24     }
25
26     /**
27      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
28     */
29     // get and post both can be handled by service method
30     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
31         // Read the data from request object
32         String email = request.getParameter("txtEmail");
33         String password = request.getParameter("txtPassword");
34
35         System.out.println("[LoginServlet] User Details: "+email+" "+password);
36
37         response.setContentType("text/html");
38         String loginTimeStamp = new Date().toString();
39         String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"</p></center>";
40
41         PrintWriter out = response.getWriter();
42         out.print(htmlResponse);
43
44     }
45
46 }

```

Note: As the service method does not allow the addition of restrictions, users need to comment on the method to add restrictions.

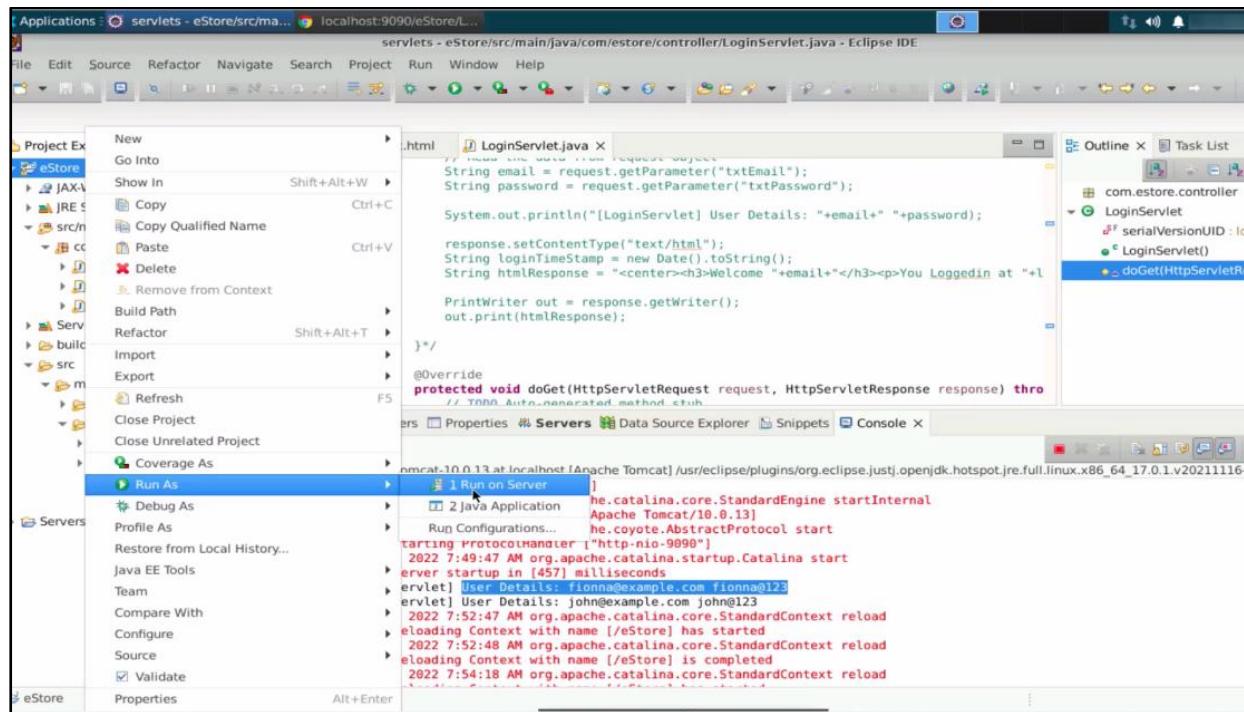
2.5 Replace the service method with doGet() and copy the complete doGet() code

```

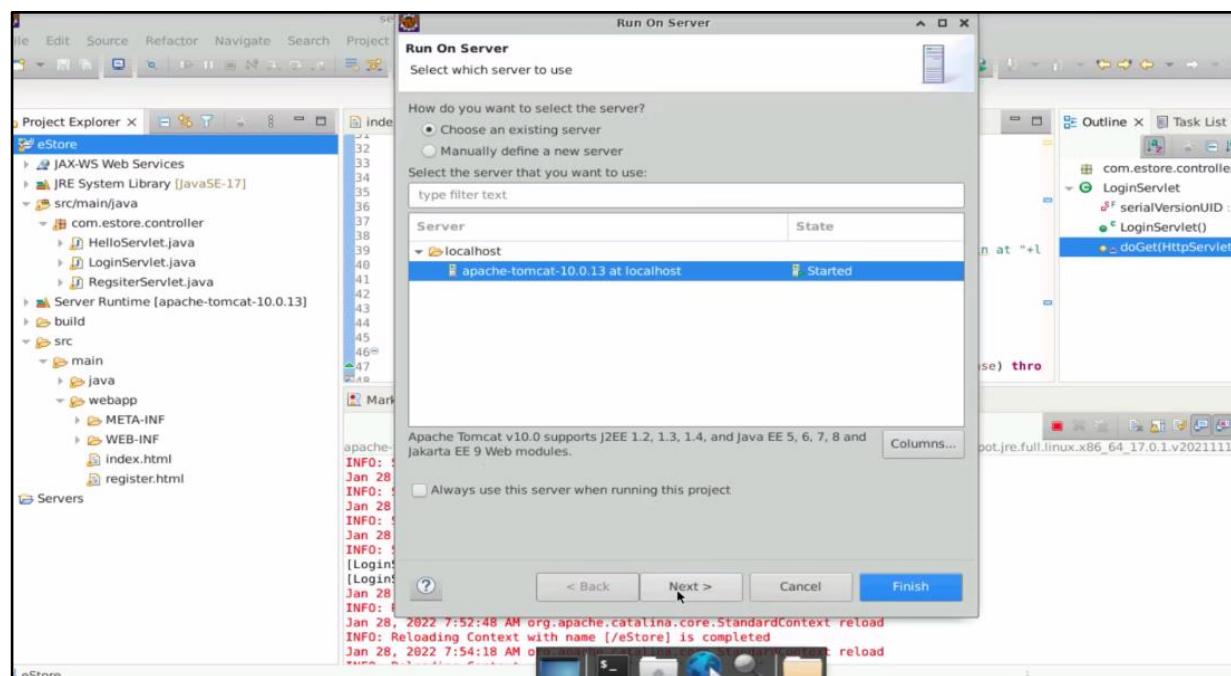
11
12 /**
13  * Servlet implementation class LoginServlet
14 */
15 public class LoginServlet extends HttpServlet {
16     private static final long serialVersionUID = 1L;
17
18     /**
19      * @see HttpServlet#HttpServlet()
20     */
21     public LoginServlet() {
22         super();
23         // TODO Auto-generated constructor stub
24     }
25
26     /**
27      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
28     */
29     // get and post both can be handled by service method
30     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
31         // Read the data from request object
32         String email = request.getParameter("txtEmail");
33         String password = request.getParameter("txtPassword");
34
35         System.out.println("[LoginServlet] User Details: "+email+" "+password);
36
37         response.setContentType("text/html");
38         String loginTimeStamp = new Date().toString();
39         String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"</p></center>";
40
41         PrintWriter out = response.getWriter();
42         out.print(htmlResponse);
43
44     }
45
46     /**
47      * @Override
48      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
49     */
50     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
51         // TODO Auto-generated method stub
52         //super.doGet(request, response);
53
54         // Read the data from request object
55         String email = request.getParameter("txtEmail");
56         String password = request.getParameter("txtPassword");
57
58         System.out.println("[LoginServlet] User Details: "+email+" "+password);
59
60         response.setContentType("text/html");
61         String loginTimeStamp = new Date().toString();
62         String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"</p></center>";
63
64         PrintWriter out = response.getWriter();
65         out.print(htmlResponse);
66     }
67
68 }

```

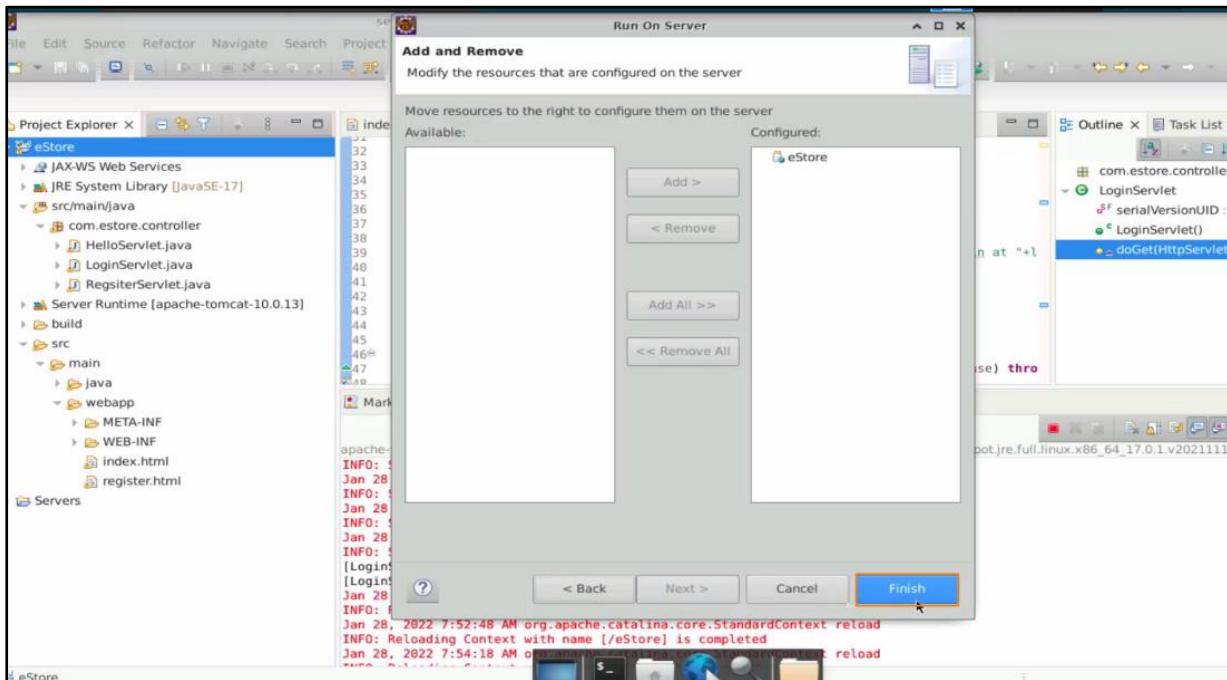
2.6 Since the server is modified, recompile the entire project. Right-click on **eStore**, select **Run As**, and click **Run on Server**



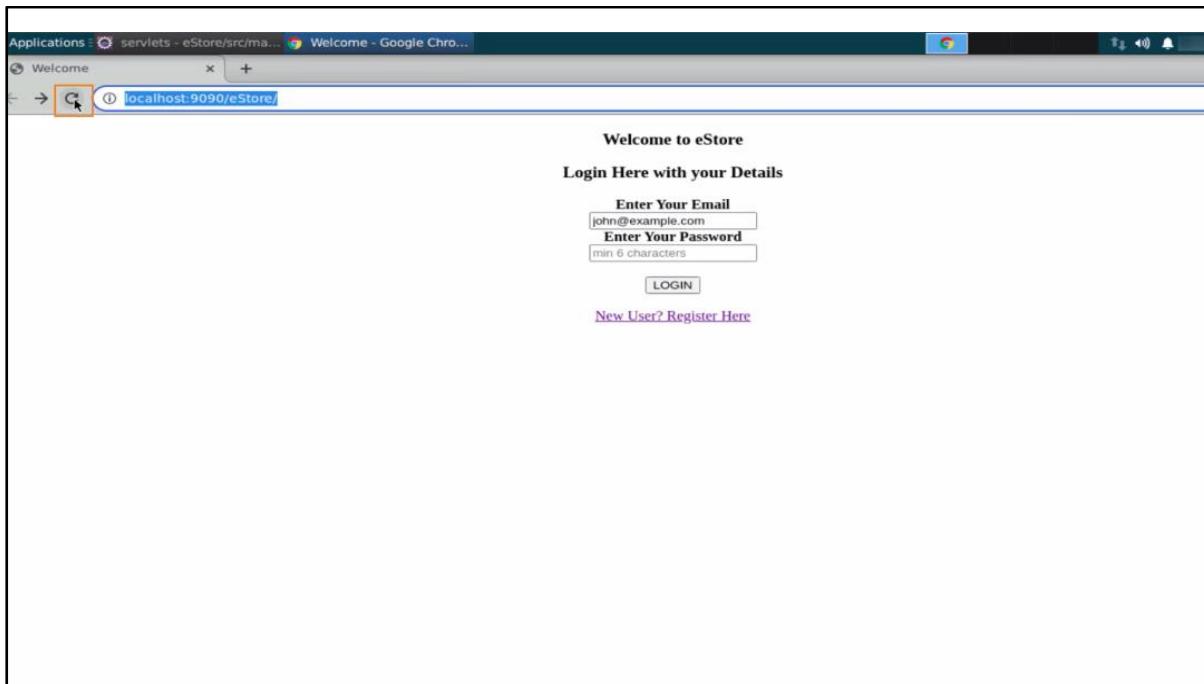
2.7 Select the tomcat server as **apache-tomcat-10.0.13** and click **Next >**



2.8 Click Finish



2.9 Refresh the browser



2.10 Enter the username **dave@example.com** and the password **dave@123**, and click on **LOGIN**



The response displayed will be as shown in the screenshot:



Step 3: Creating a doGet() method

3.1 To manipulate the response, go to the **LoginServlet.java** file and write the **doGet()** method

Eclipse IDE - Java EE IDE for Web Developers

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer X index.html *LoginServlet.java X Outline X Task List

eStore

JAX-WS Web Services

IREE System Library [javaSE-17]

src/main/java

com.estore.controller

HelloServlet.java

LoginServlet.java

RegisterServlet.java

Server Runtime [apache-tomcat-10.0.13]

build

src

main

webapp

META-INF

WEB-INF

index.html

register.html

index.html

```

44
45     @Override
46     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
47         // TODO Auto-generated method stub
48         //super.doGet(request, response);
49
50         // Read the data from request object
51         String email = request.getParameter("txtEmail");
52         String password = request.getParameter("txtPassword");
53
54         System.out.println("[LoginServlet] doGet[User Details: "+email+" "+password]);
55
56         response.setContentType("text/html");
57         String loginTimeStamp = new Date().toString();
58         String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"</p></center>";
59
60         PrintWriter out = response.getWriter();
61         out.print(htmlResponse);
62
63     }
64 }
65
66 }
```

Markers Properties Servers Data Source Explorer Snippets Console X

apache-tomcat-10.0.13 at localhost [Apache Tomcat] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_17.0.1.v20211116-Jan 28, 2022 7:49:47 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in [457] milliseconds
[LoginServlet] User Details: fionna@example.com fionna@123
Jan 28, 2022 7:52:47 AM org.apache.catalina.core.StandardContext reload
INFO: Reloading Context with name [/eStore] has started
Jan 28, 2022 7:52:48 AM org.apache.catalina.core.StandardContext reload
INFO: Reloading Context with name [/eStore] is completed
Jan 28, 2022 7:54:18 AM org.apache.catalina.core.StandardContext reload
INFO: Reloading Context with name [/eStore] has started
Jan 28, 2022 7:54:18 AM org.apache.catalina.core.StandardContext reload
INFO: Reloading Context with name [/eStore] is completed
[LoginServlet] User Details: dave@example.com dave@123

Writable Smart Insert 55 : 49 : 1854

3.2 Add the **doGet()** parameter in the HTML response as **Handled by [doGet]** and save it

Eclipse IDE - Java EE IDE for Web Developers

File Edit Source Refactor Navigate Search Project Run Window Help

index.html *LoginServlet.java X Outline X Task List

eStore

JAX-WS Web Services

IREE System Library [javaSE-17]

src/main/java

com.estore.controller

HelloServlet.java

LoginServlet.java

RegisterServlet.java

Server Runtime [apache-tomcat-10.0.13]

build

src

main

webapp

META-INF

WEB-INF

index.html

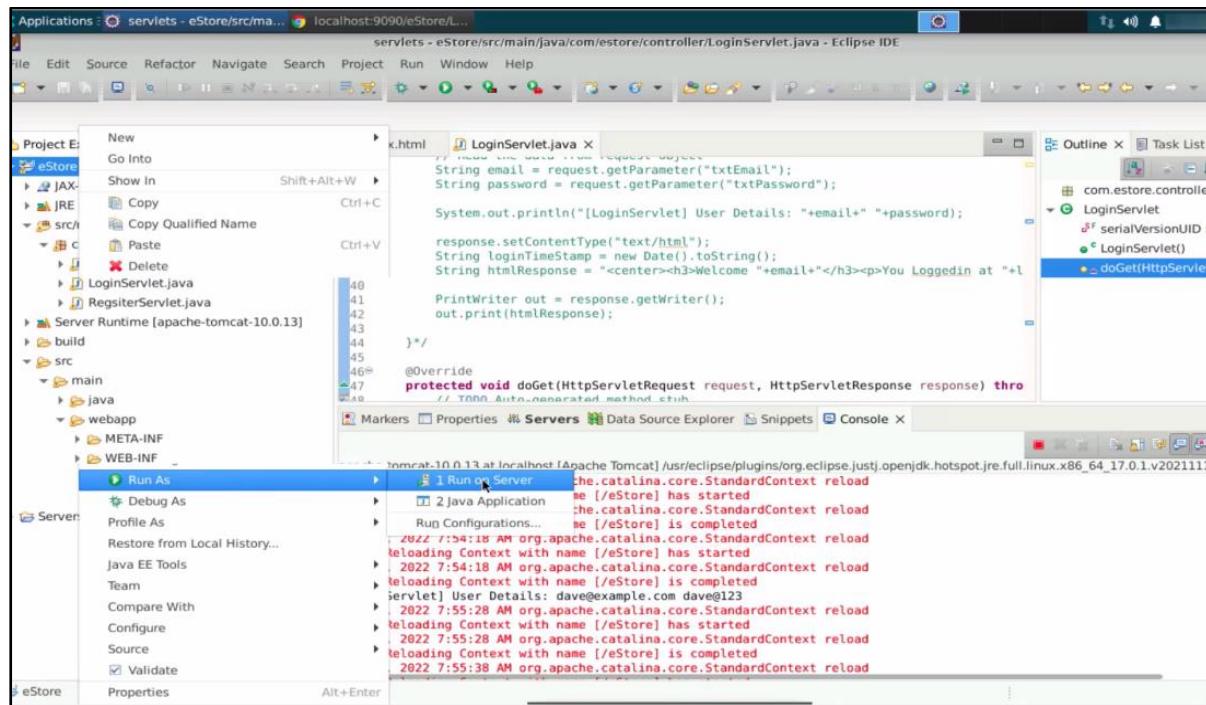
register.html

index.html

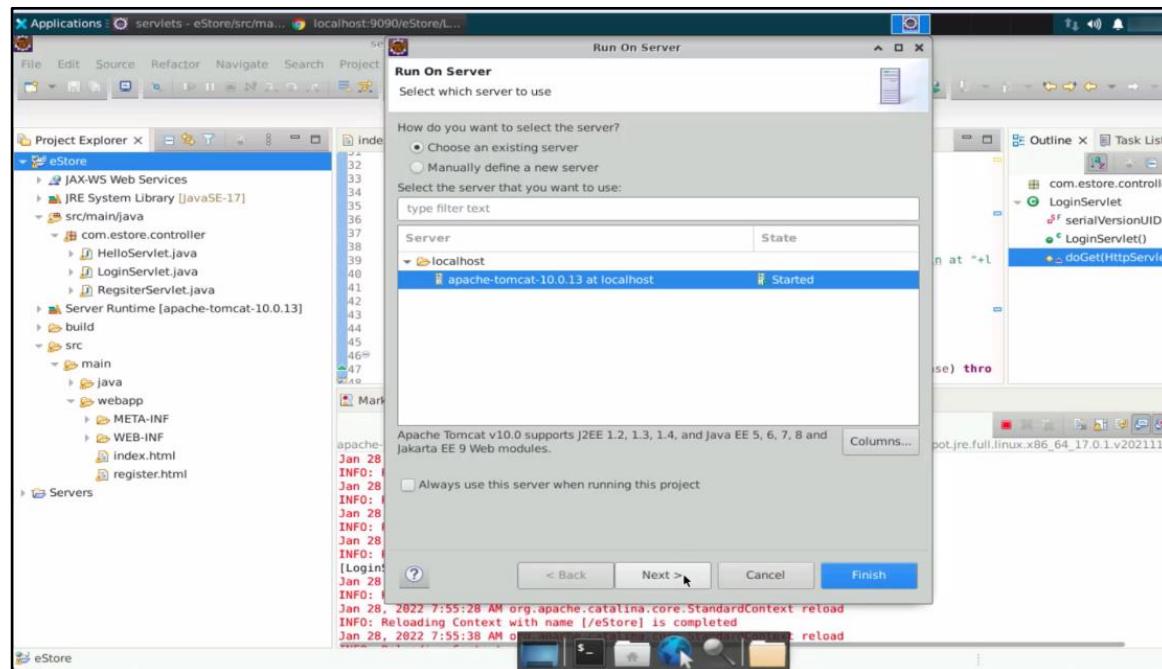
```

32
33     String email = request.getParameter("txtEmail");
34     String password = request.getParameter("txtPassword");
35
36     System.out.println("[LoginServlet] User Details: "+email+" "+password);
37
38     response.setContentType("text/html");
39     String loginTimeStamp = new Date().toString();
40     String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"</p></center>";
41
42     PrintWriter out = response.getWriter();
43     out.print(htmlResponse);
44
45 }
46
47 @Override
48 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
49     // TODO Auto-generated method stub
50     //super.doGet(request, response);
51
52     // Read the data from request object
53     String email = request.getParameter("txtEmail");
54     String password = request.getParameter("txtPassword");
55
56     System.out.println("[LoginServlet] doGet[User Details: "+email+" "+password]);
57
58     response.setContentType("text/html");
59     String loginTimeStamp = new Date().toString();
60     String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"<br> Handled by [doGet()]</p></center>";
61
62     PrintWriter out = response.getWriter();
63     out.print(htmlResponse);
64
65 }
66 }
```

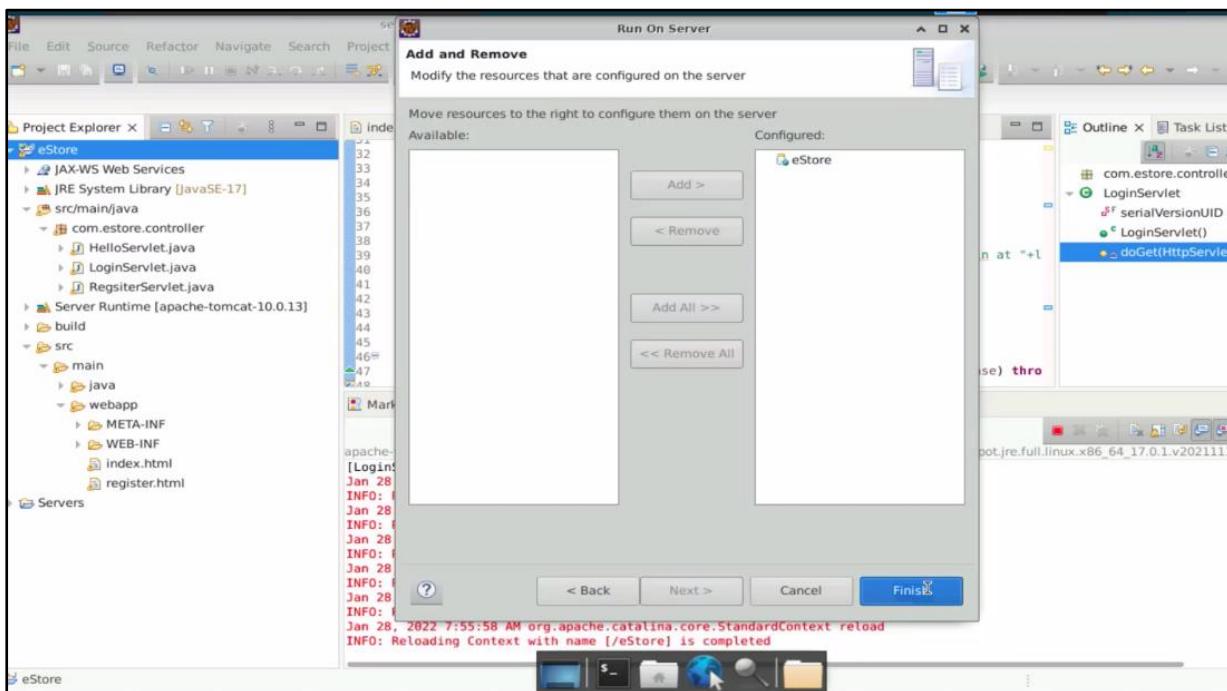
3.3 Re-run the project on the server by right-clicking on eStore, selecting Run As, and then selecting Run on Server



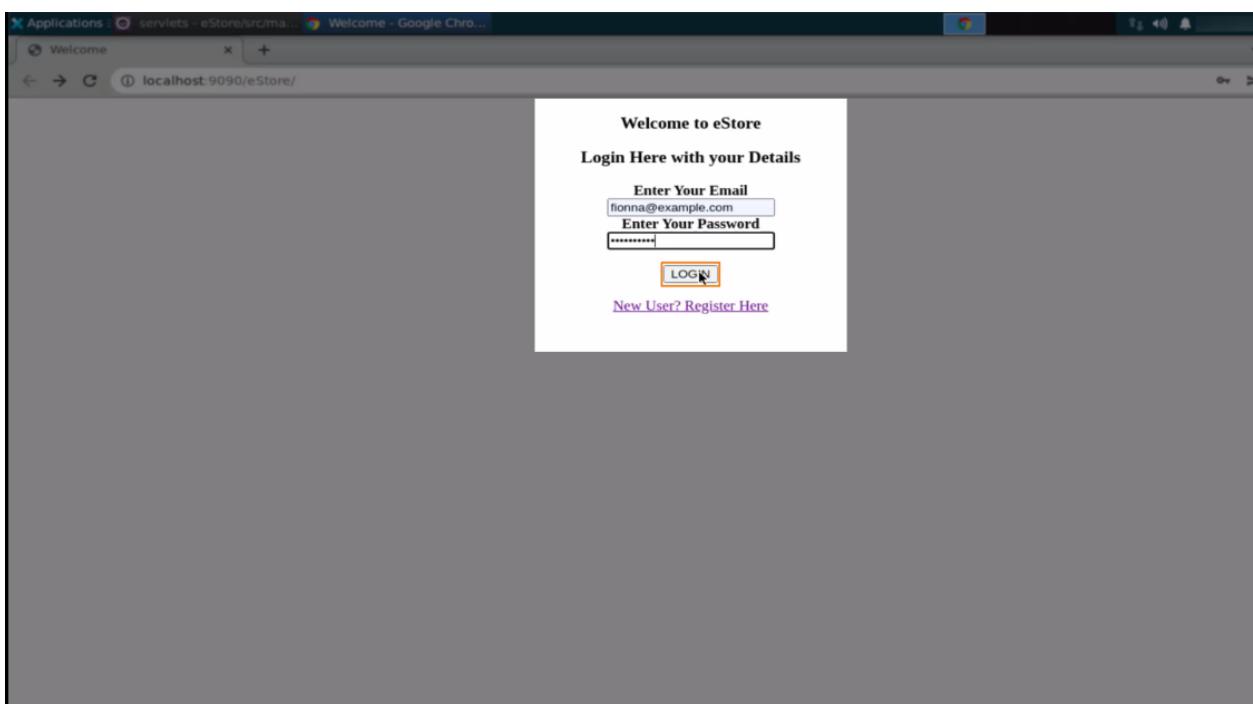
3.4 Select Apache Tomcat as apache-tomcat-10.0.13 at localhost and click Next



3.5 Click Finish



3.6 Refresh the browser, enter the username and password, and click LOGIN

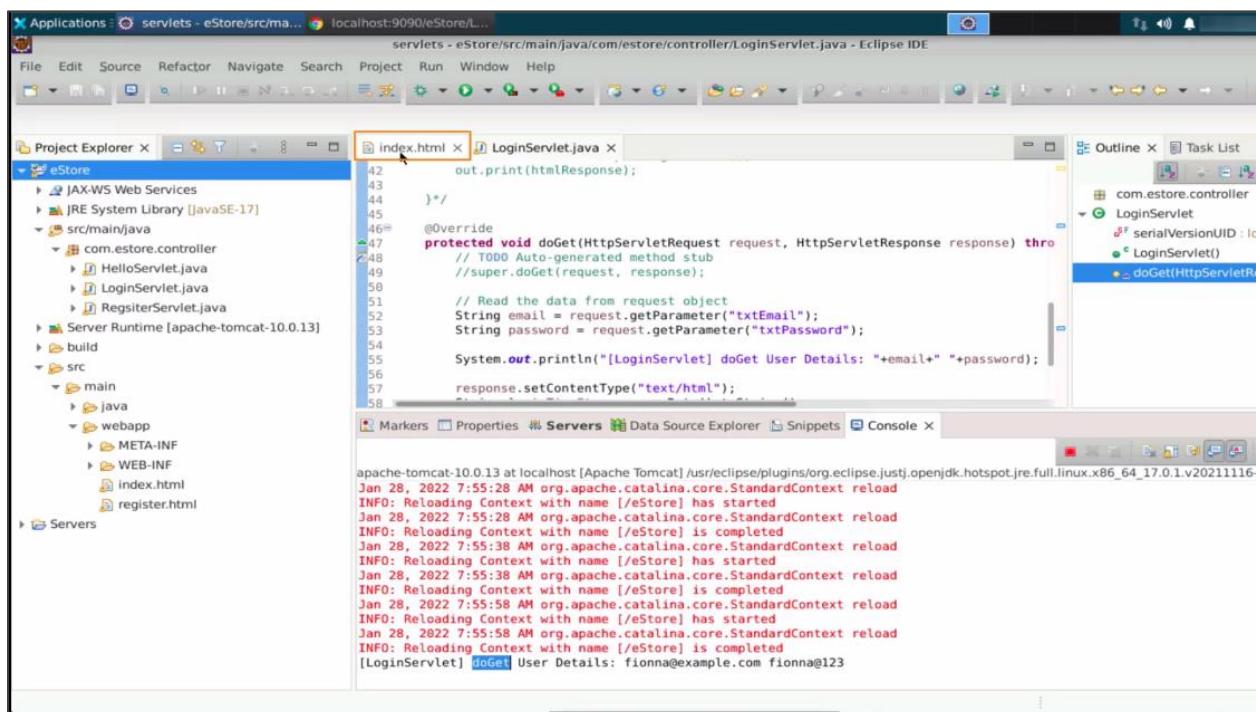


The output appears as shown below:



Step 4: Creating a doPost() method

4.1 Go to the index.html page



4.2 Change the method to doPost()

```

    4 <meta charset="UTF-8">
    5 <title>Welcome</title>
    6 </head>
    7<body>
    8
    9<center>
    10 <h2>Welcome to eStore</h2>
    11 <h2>Login Here with your Details</h2>
    12
    13<= <form action="Login" method="post">
    14 <b>Enter Your Email</b><br>
    15 <input type="text" name="txtEmail" placeholder="eg: john@example.com"/><br>
    16 <b>Enter Your Password</b><br>
    17 <input type="password" name="txtPassword" placeholder="min 6 characters"/><br>
    18 <br>
    19 <input type="submit" value="LOGIN"/>
    20
    21 </center>

```

The code editor shows the Java code for LoginServlet.java. The line 13 is highlighted with a yellow background, indicating the change made.

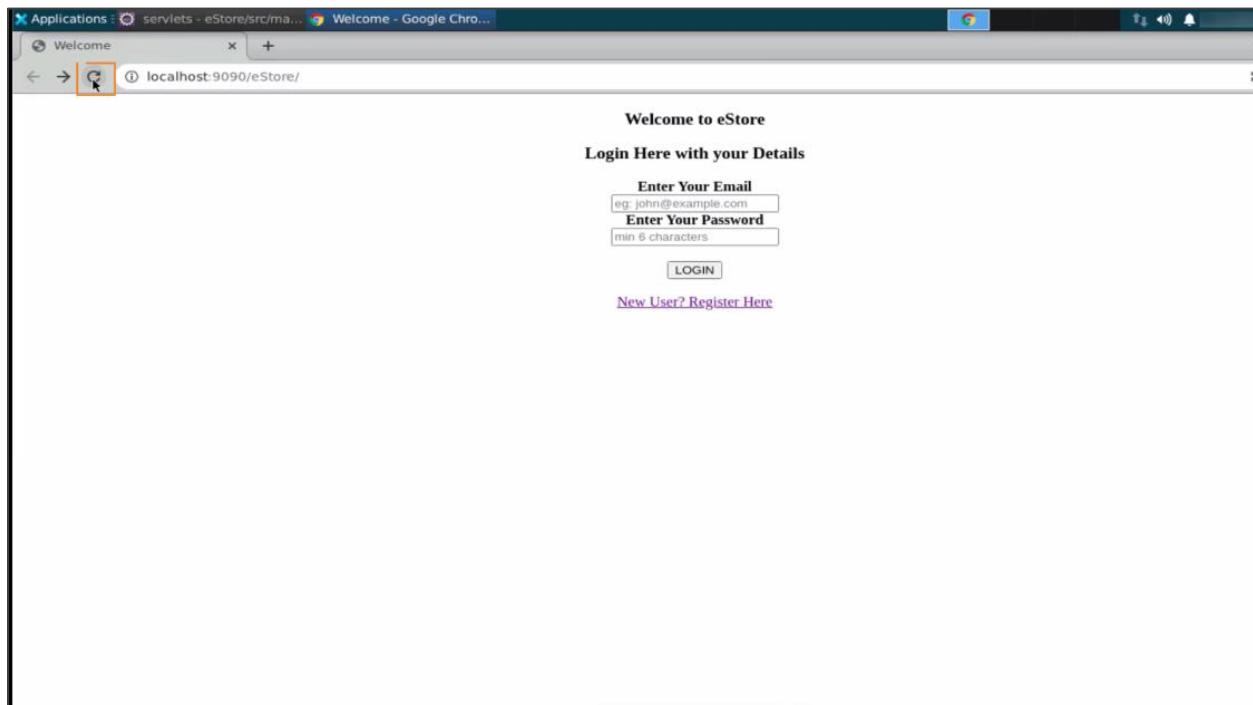
The Eclipse interface includes the Project Explorer, Outline, and Console panes. The Console pane shows the Tomcat logs:

```

    apache-tomcat-10.0.13 at localhost [Apache Tomcat] /usr/eclipse/plugins/org.eclipse.just/jopenjdk.hotspot.jre.full.linux.x86_64_17.0.1.v20211116
    [LoginServlet] User Details: dave@example.com dave@123
    Jan 28, 2022 7:55:28 AM org.apache.catalina.core.StandardContext reload
    INFO: Reloading Context with name [/eStore] has started
    Jan 28, 2022 7:55:28 AM org.apache.catalina.core.StandardContext reload
    INFO: Reloading Context with name [/eStore] is completed
    Jan 28, 2022 7:55:38 AM org.apache.catalina.core.StandardContext reload
    INFO: Reloading Context with name [/eStore] has started
    Jan 28, 2022 7:55:38 AM org.apache.catalina.core.StandardContext reload
    INFO: Reloading Context with name [/eStore] is completed
    Jan 28, 2022 7:55:58 AM org.apache.catalina.core.StandardContext reload
    INFO: Reloading Context with name [/eStore] has started
    Jan 28, 2022 7:55:58 AM org.apache.catalina.core.StandardContext reload
    INFO: Reloading Context with name [/eStore] is completed
    [LoginServlet] doGet User Details: fionna@example.com fionna@123

```

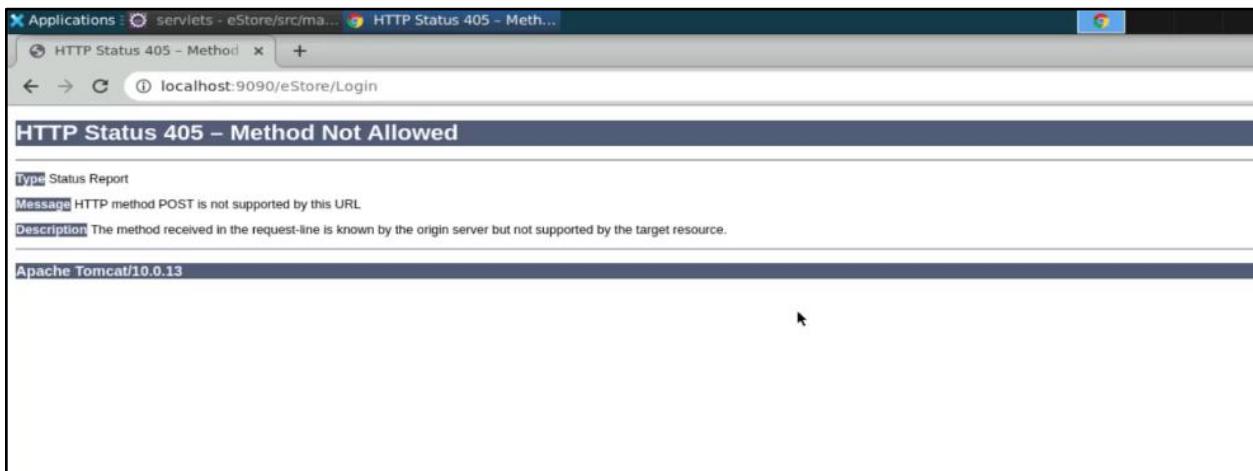
4.3 Refresh the browser



4.4 Enter the username and password and click **LOGIN**



Observation: The server throws an exception, which indicates that the login servlet is not capable of entertaining the request in the form of a post. This means that the URL does not support the HTTP method POST.



4.5 Go back to the **LoginServlet.java** file and override the **doPost()** method (lines 66 to 84). Change it to **Handled by [doPost]** and save it.

```

    50
    51     // Read the data from request object
    52     String email = request.getParameter("txtEmail");
    53     String password = request.getParameter("txtPassword");
    54
    55     System.out.println("[LoginServlet] doGet User Details: "+email+" "+password);
    56
    57     response.setContentType("text/html");
    58     String loginTimeStamp = new Date().toString();
    59     String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"<br> Handled by [doGet] </p></center>";
    60
    61     PrintWriter out = response.getWriter();
    62     out.print(htmlResponse);
    63
    64 }
    65
    66@override
    67 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    68     // TODO Auto-generated method stub
    69     //super.doPost(req, resp);
    70
    71     String email = request.getParameter("txtEmail");
    72     String password = request.getParameter("txtPassword");
    73
    74     System.out.println("[LoginServlet] doPost User Details: "+email+" "+password);
    75
    76     response.setContentType("text/html");
    77     String loginTimeStamp = new Date().toString();
    78     String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"<br> Handled by [doPost] </p></center>";
    79
    80     PrintWriter out = response.getWriter();
    81     out.print(htmlResponse);
    82 }
    83
    84 }
    85

```

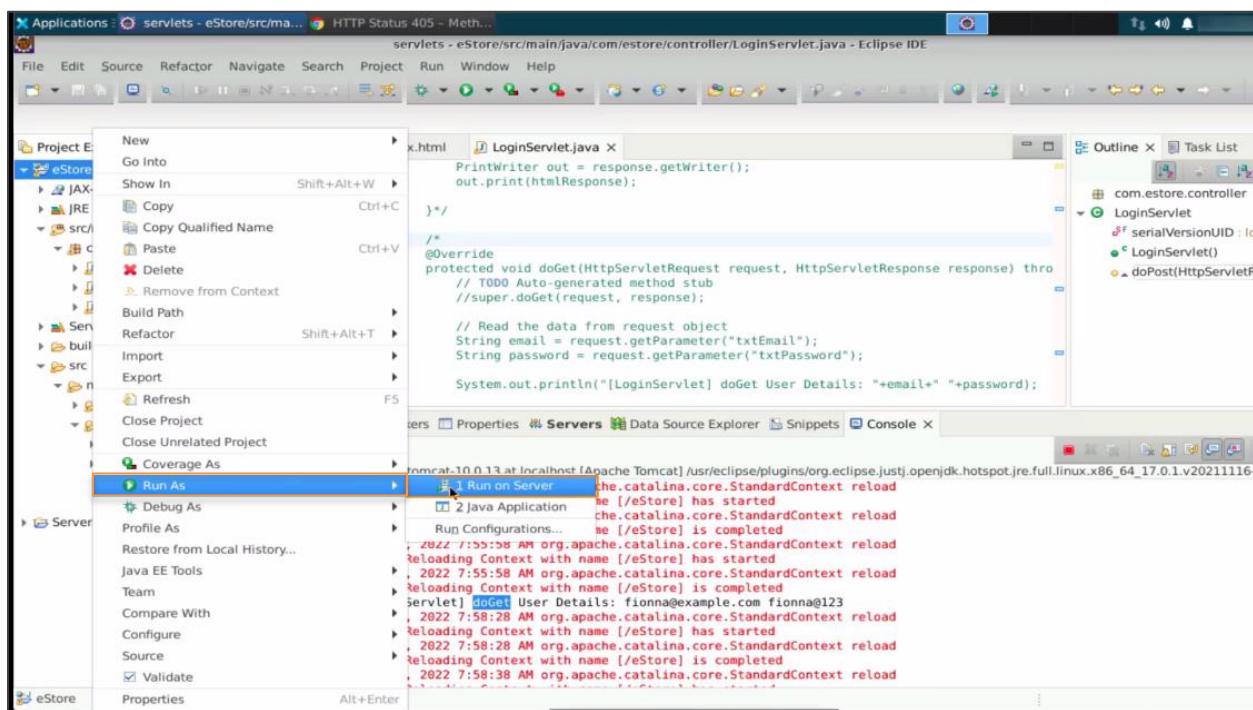
4.6 Comment the **doGet()** method and save it

```

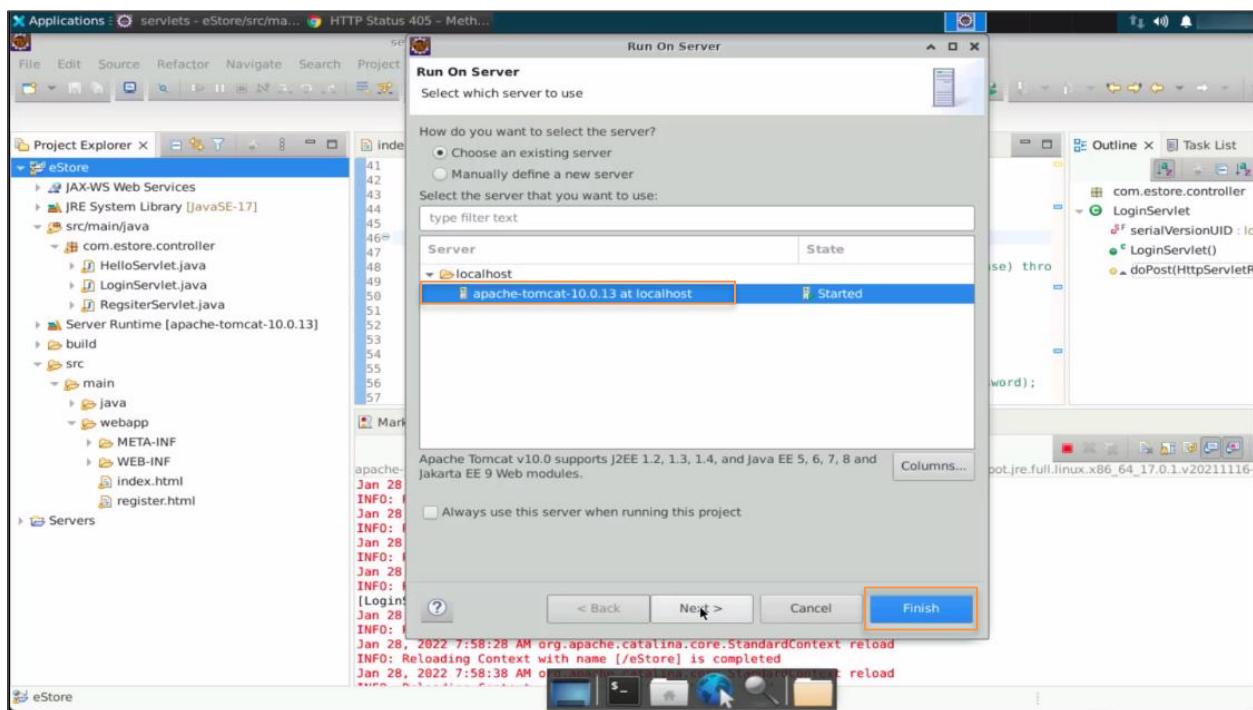
    41     PrintWriter out = response.getWriter();
    42     out.print(htmlResponse);
    43
    44 }
    45
    46 /**
    47 @Override
    48 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    49     // TODO Auto-generated method stub
    50     //super.doGet(request, response);
    51
    52     // Read the data from request object
    53     String email = request.getParameter("txtEmail");
    54     String password = request.getParameter("txtPassword");
    55
    56     System.out.println("[LoginServlet] doGet User Details: "+email+" "+password);
    57
    58     response.setContentType("text/html");
    59     String loginTimeStamp = new Date().toString();
    60     String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"<br> Handled by [doGet] </p></center>";
    61
    62     PrintWriter out = response.getWriter();
    63     out.print(htmlResponse);
    64
    65 }
    66
    67
    68@override
    69 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    70     // TODO Auto-generated method stub
    71     //super.doPost(req, resp);
    72
    73     String email = request.getParameter("txtEmail");
    74     String password = request.getParameter("txtPassword");
    75
    76     System.out.println("[LoginServlet] doPost User Details: "+email+" "+password);
    77
    78 }
    79
    80
    81
    82
    83
    84
    85

```

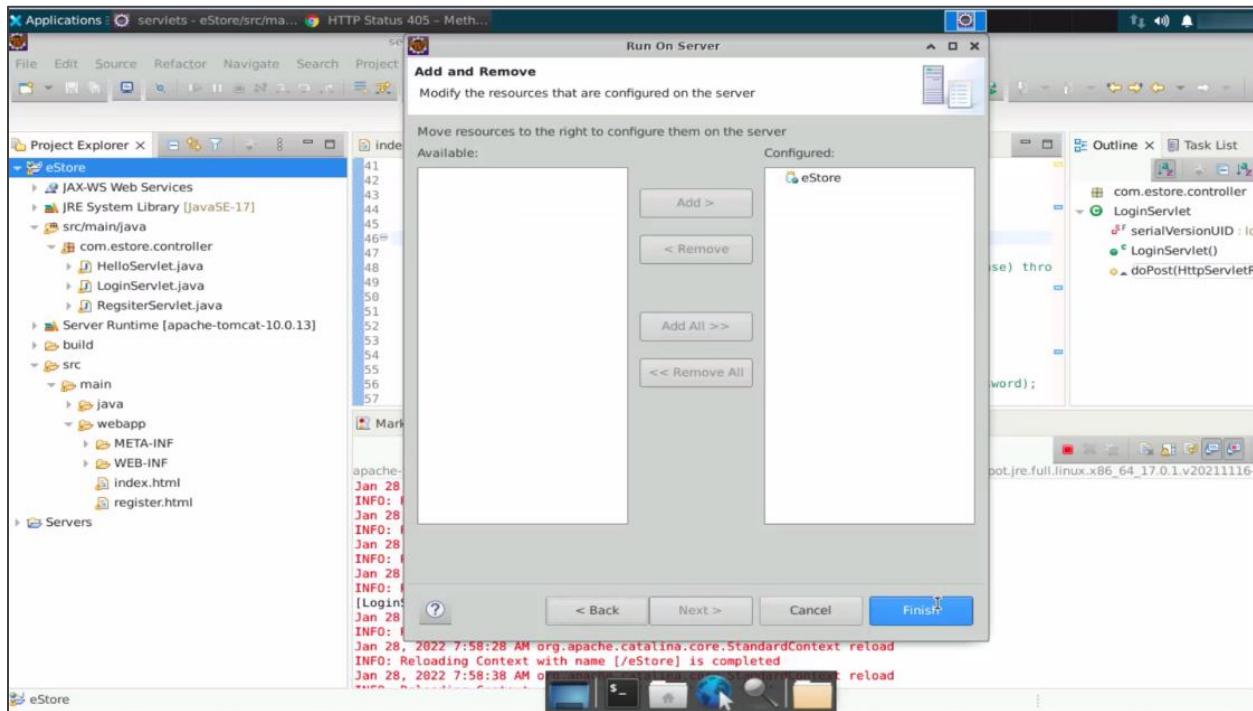
4.7 Run the code on the server by clicking **Run As** and **Run on Server**



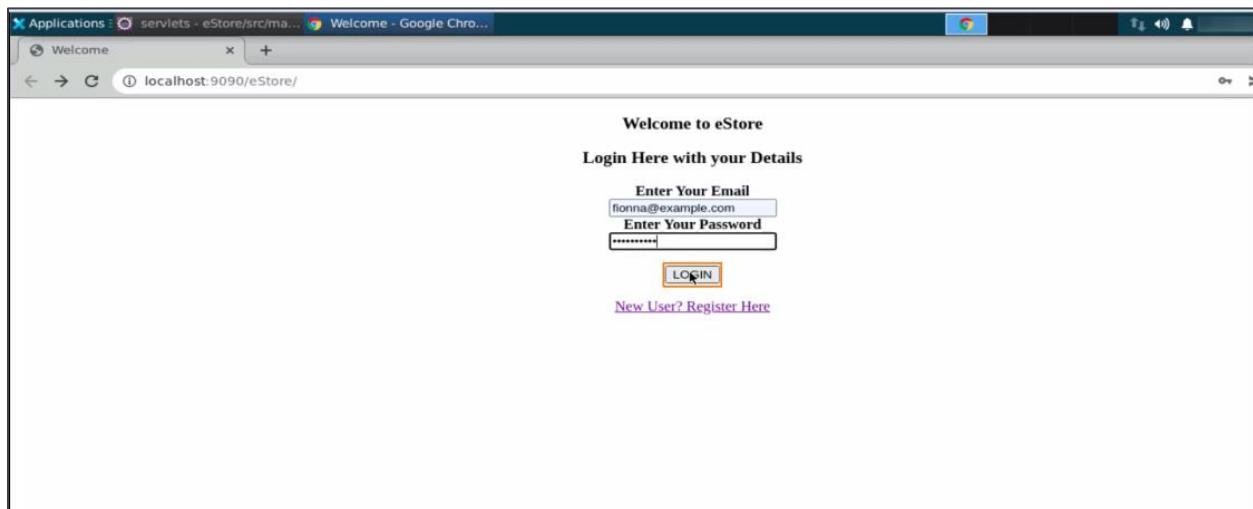
4.8 Select the Apache Tomcat server and click **Next**



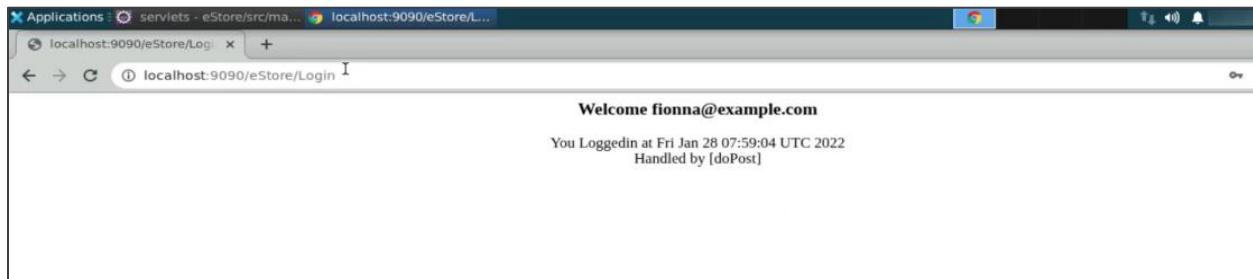
4.9 Click Finish



4.10 Go to the browser and refresh the page. Enter the username and password for Fionna, and click LOGIN



You can see the output as **Handled by [doPost]**. This means that the **doPost()** method is now working.



This indicates that the Servlet is handling the doGet and doPost methods per the requirements. The service method can also be used to address this issue. There are also other methods such as doPut and doDelete but in general, GET and POST are used.

Following these steps, you have successfully explored the doGet() and doPost() services in Servlet.