Lesson 01 Demo 02

Create an HTML Page with a Form to Send Request to a Servlet

Objective: To create an HTML page that contains a login and registration form for sending a request to a Servlet

Tools Required: Eclipse IDE

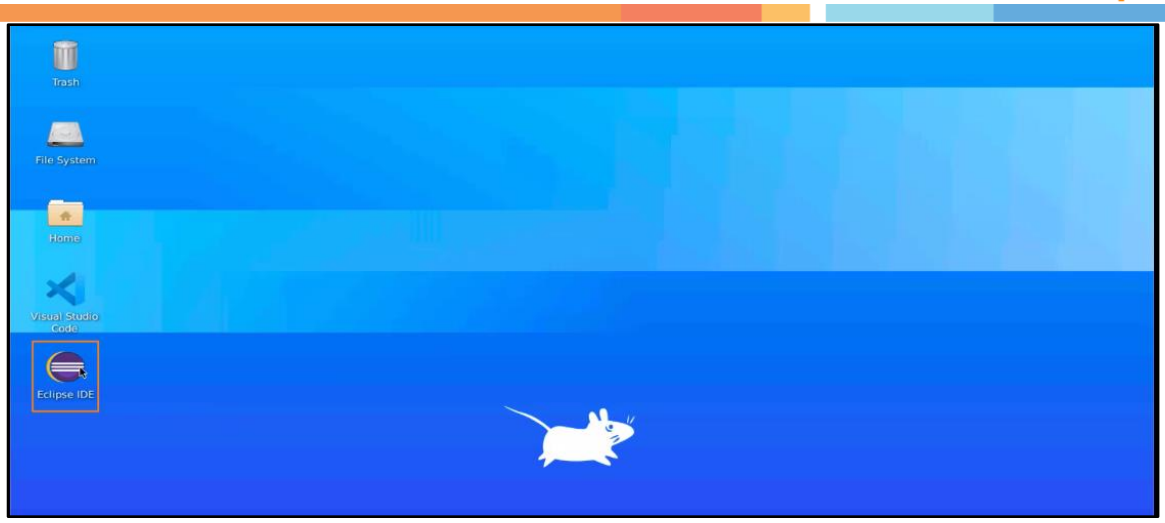
Prerequisites: None

Steps to be followed:

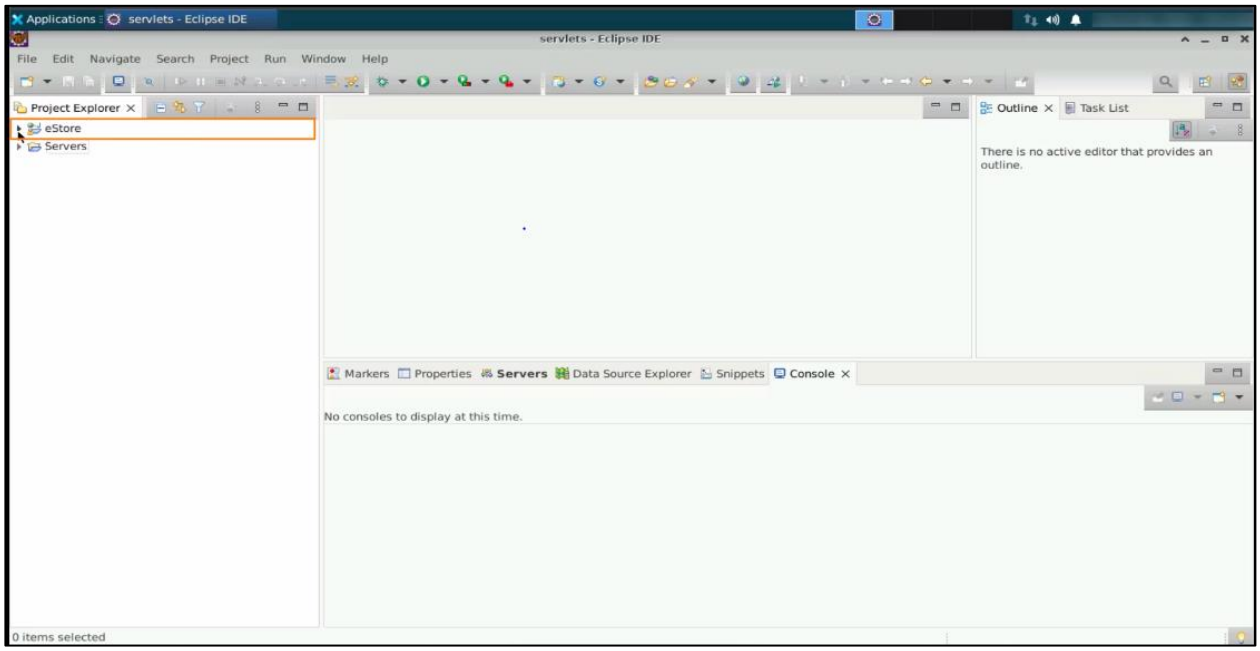
1. Create a login form in HTML
2. Create a new Servlet for the login form
3. Create a method in the login Servlet for an HTTP Servlet request
4. Create a form to register users
5. Create a new Servlet for the registration form
6. Create a method in the registered Servlet for an HTTP Servlet request

Step 1: Create a login form in HTML

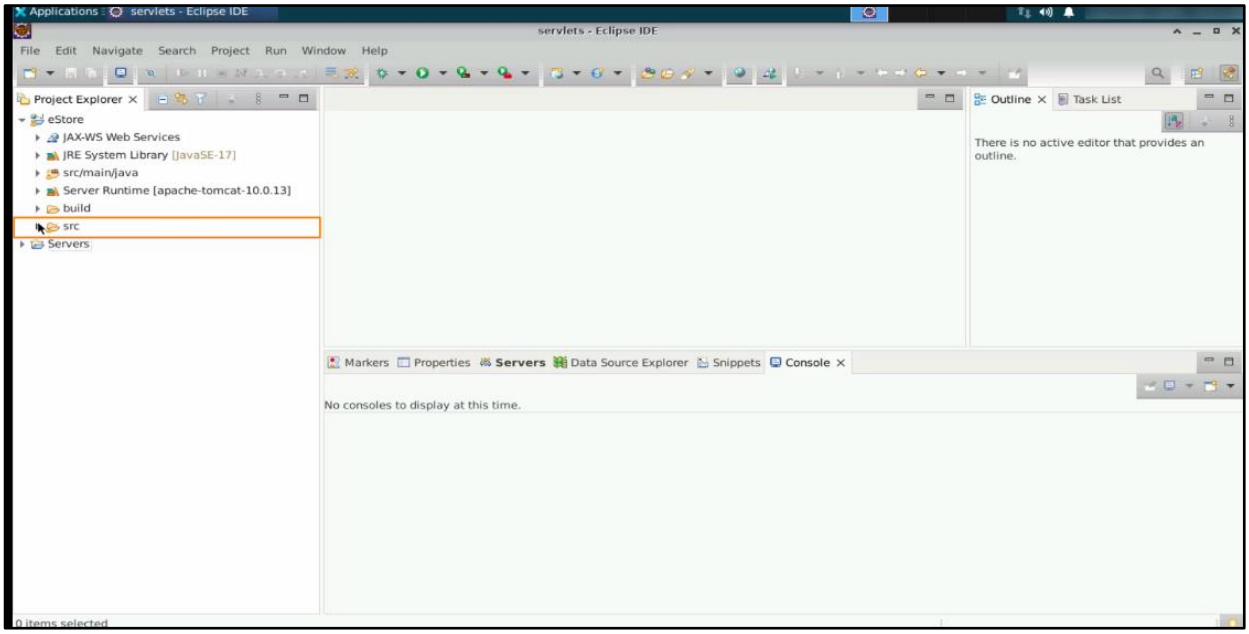
1.1 Open the **Eclipse IDE**



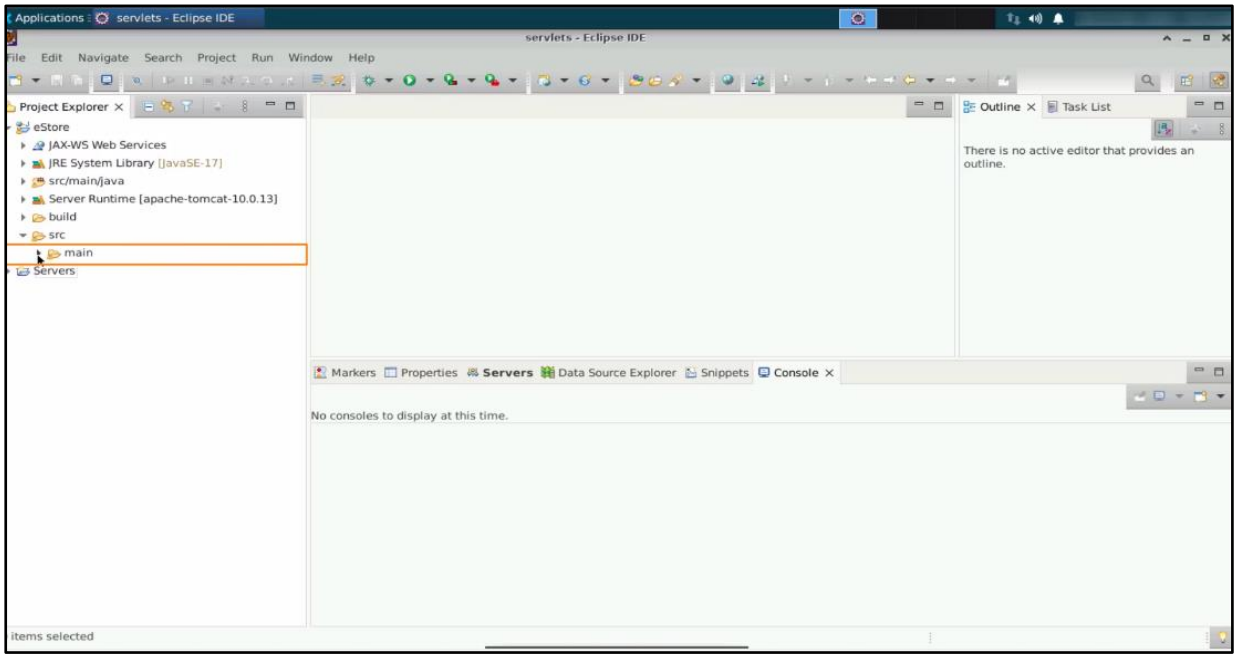
1.2 Open the **eStore** project



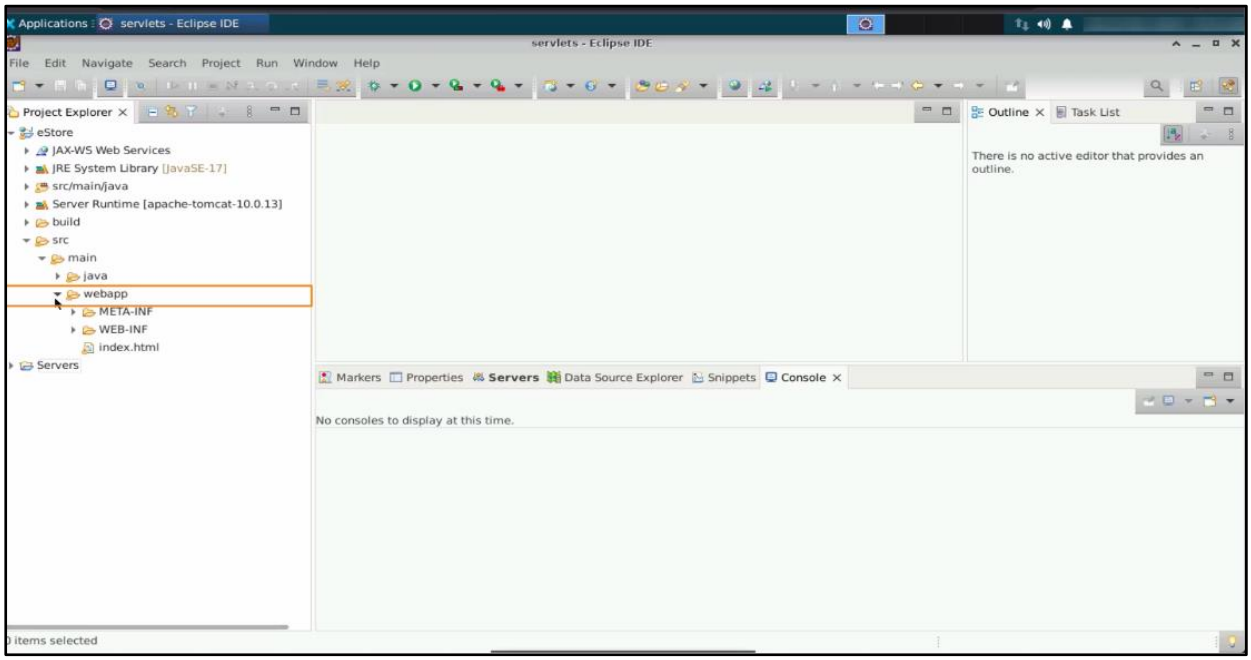
1.3 Go to the **src** folder



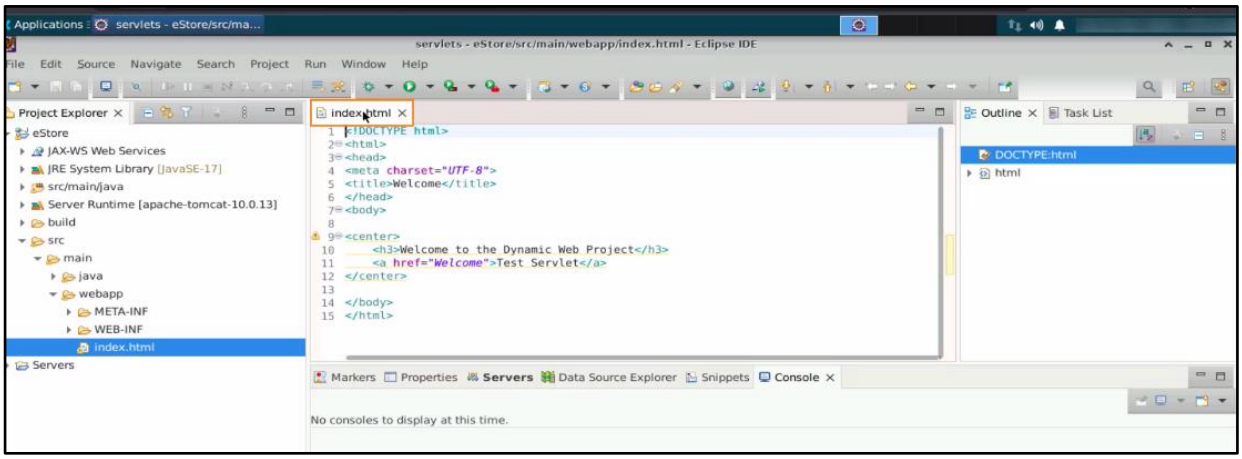
1.4 Go to the **main**



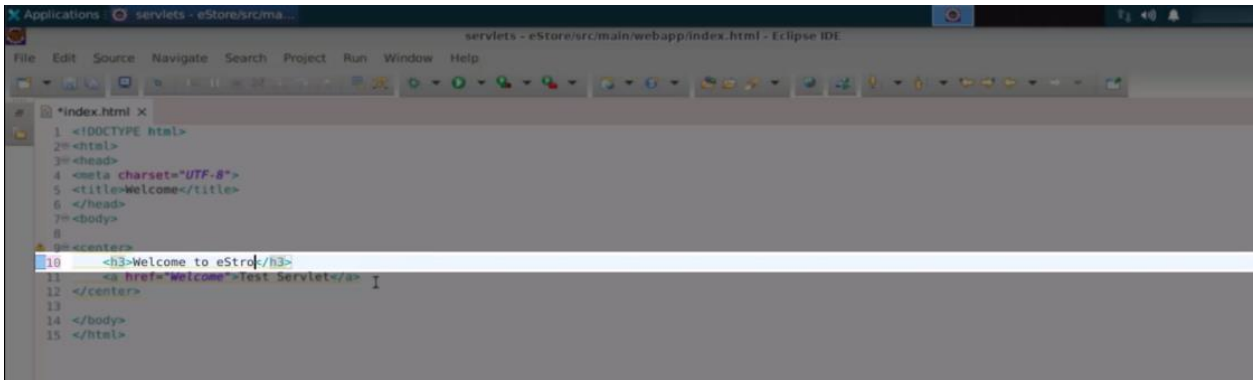
1.5 Then, go to **webapp**



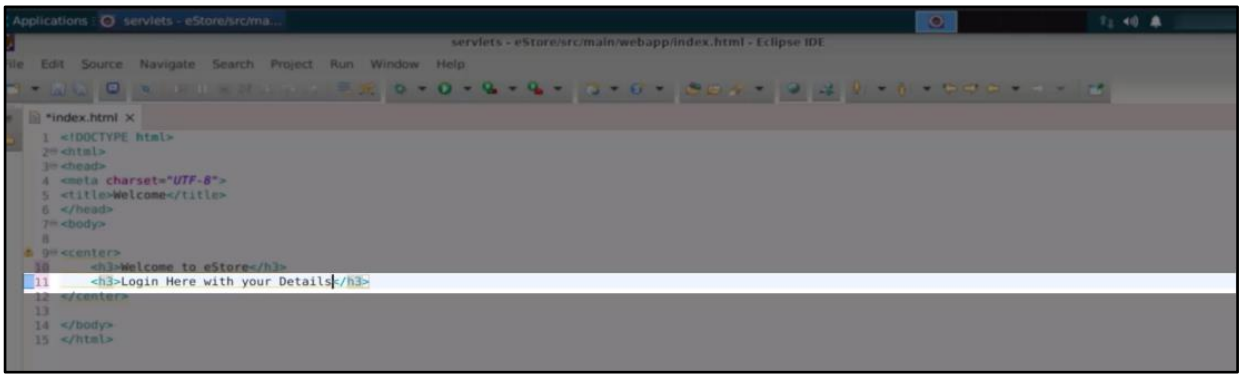
1.6 Open **index.html** file



1.7 Enter **Welcome to eStore** in the **h3** tag

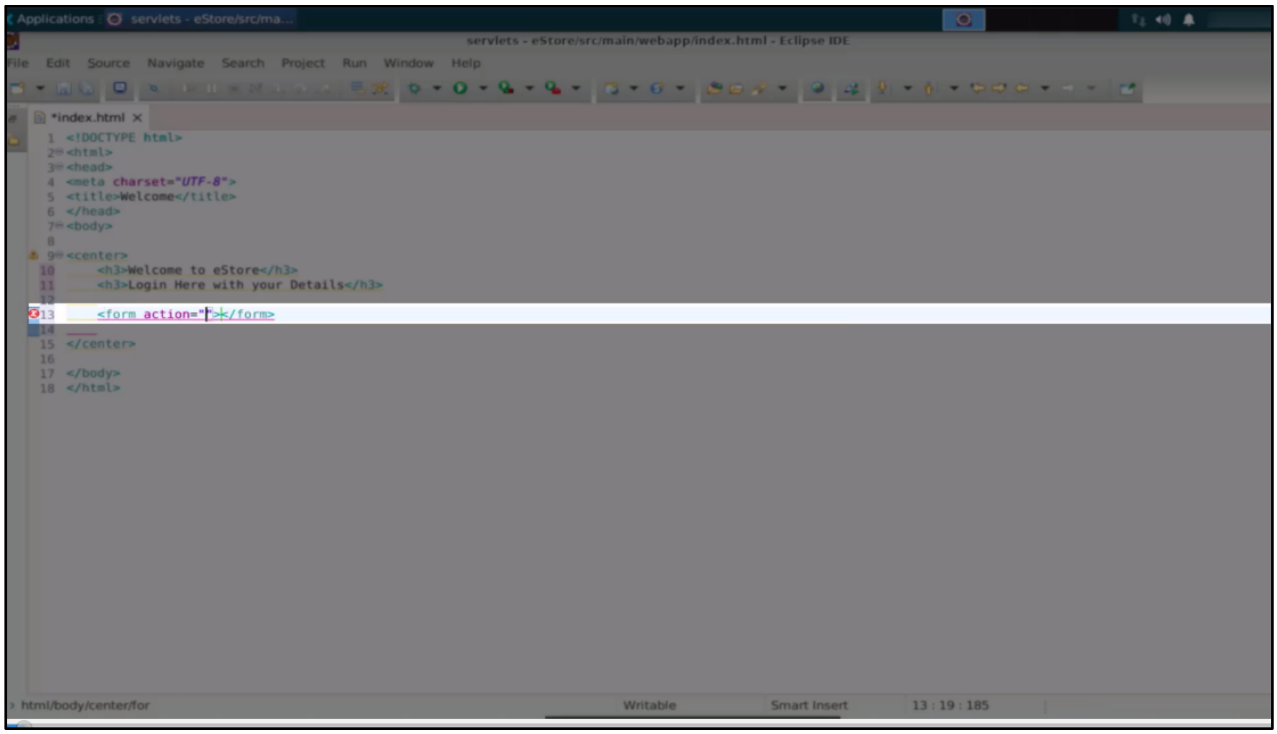


1.8 Remove the **hyperlink** and add an **h3** tag. Enter **Login Here with your Details** as a display message.



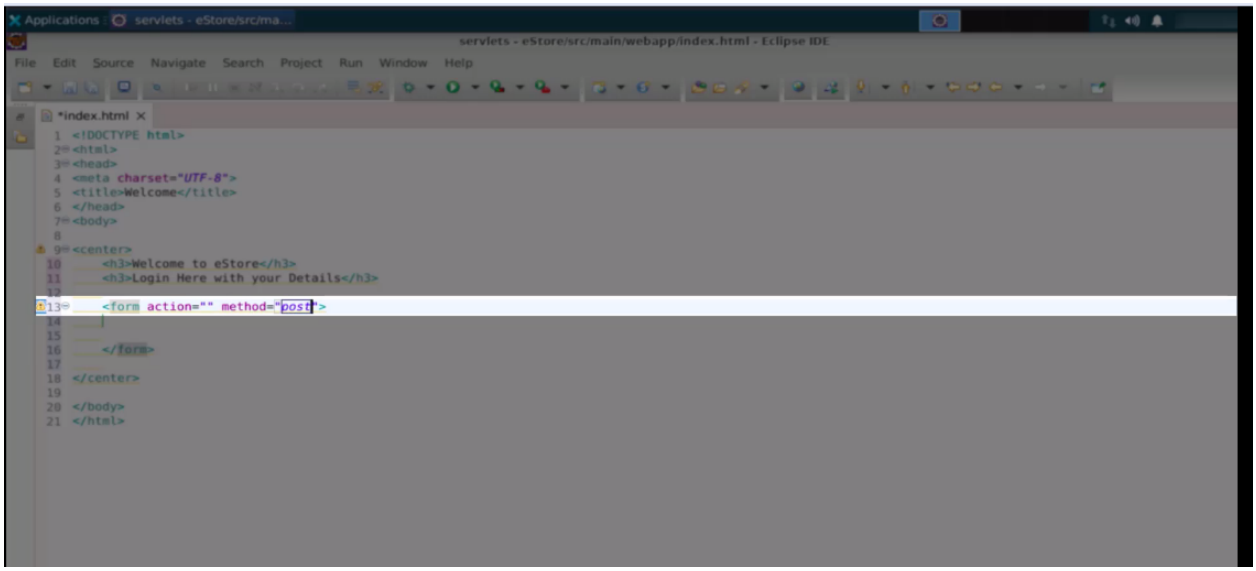
1.9 Create the **login** form

<form action = “”></form>



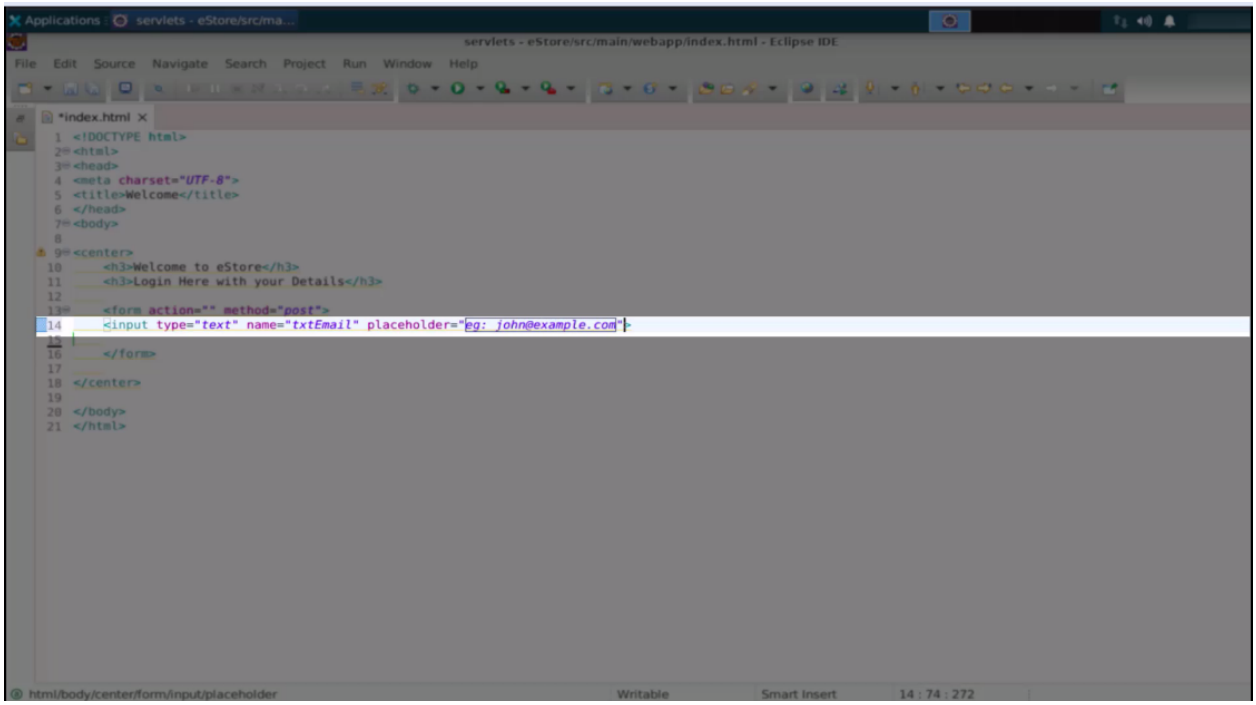
1.10 Enter the **post** method as you are working with user-sensitive data

<form action = “” method=”post”></form>

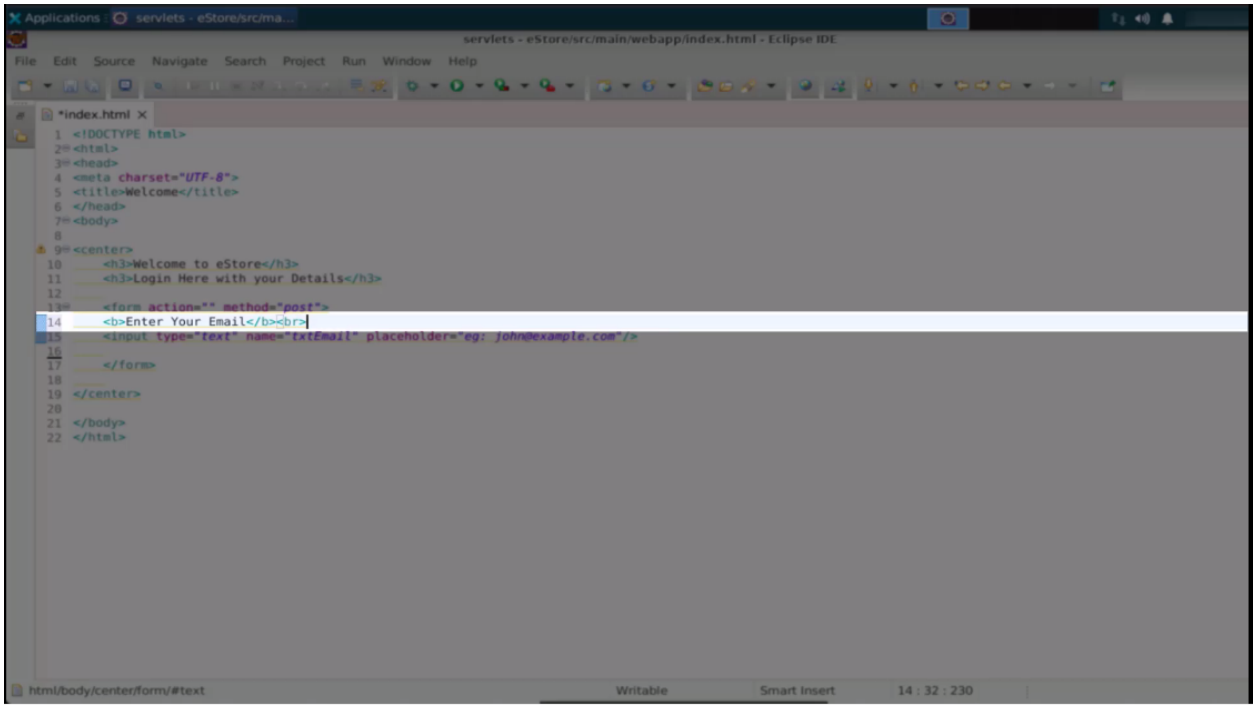


1.11 Enter the **input** with type text for email

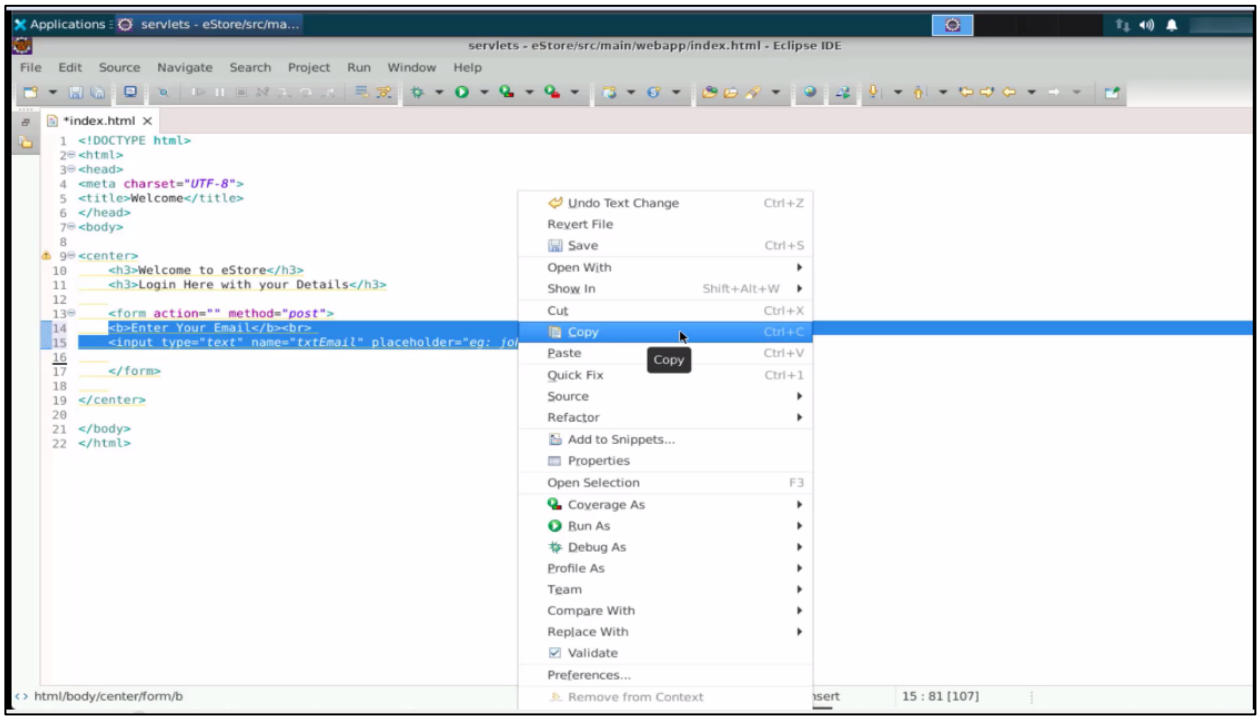
<input type="text" name="txtEmail" placeholder="eg: john@example.com"/></br>



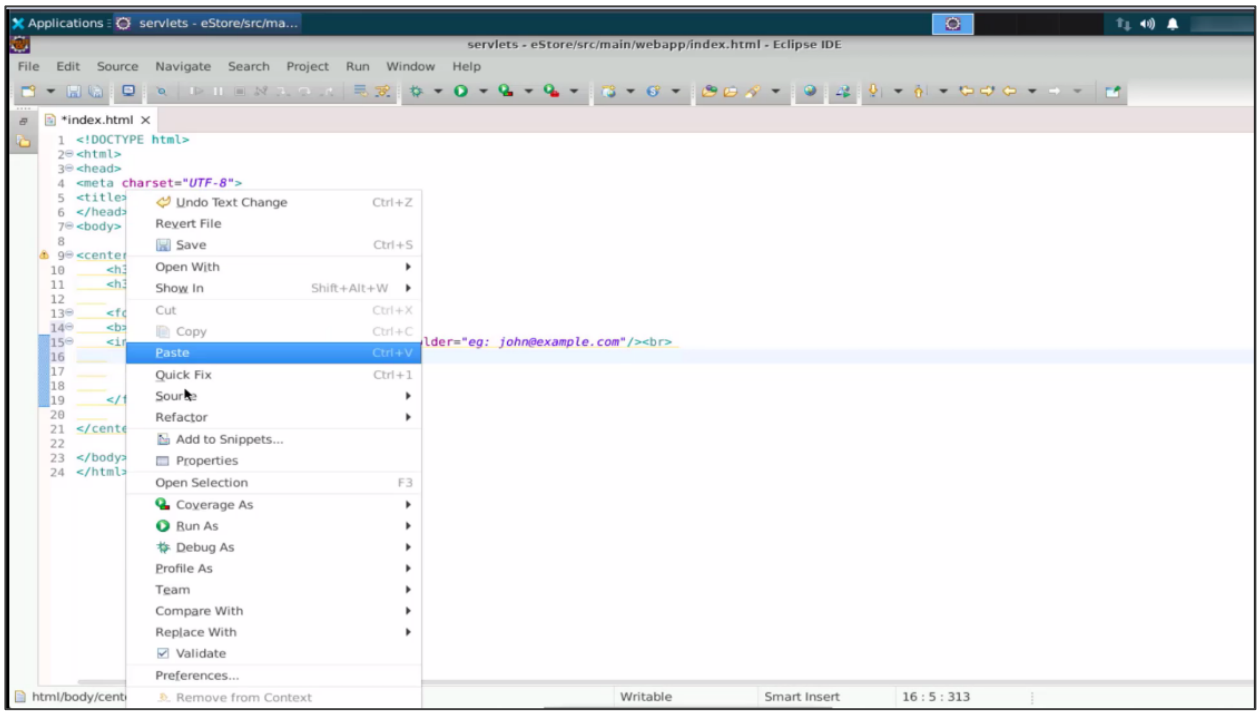
1.12 Before the input tag, use the **bold** tag and enter the **Enter Your Email** line



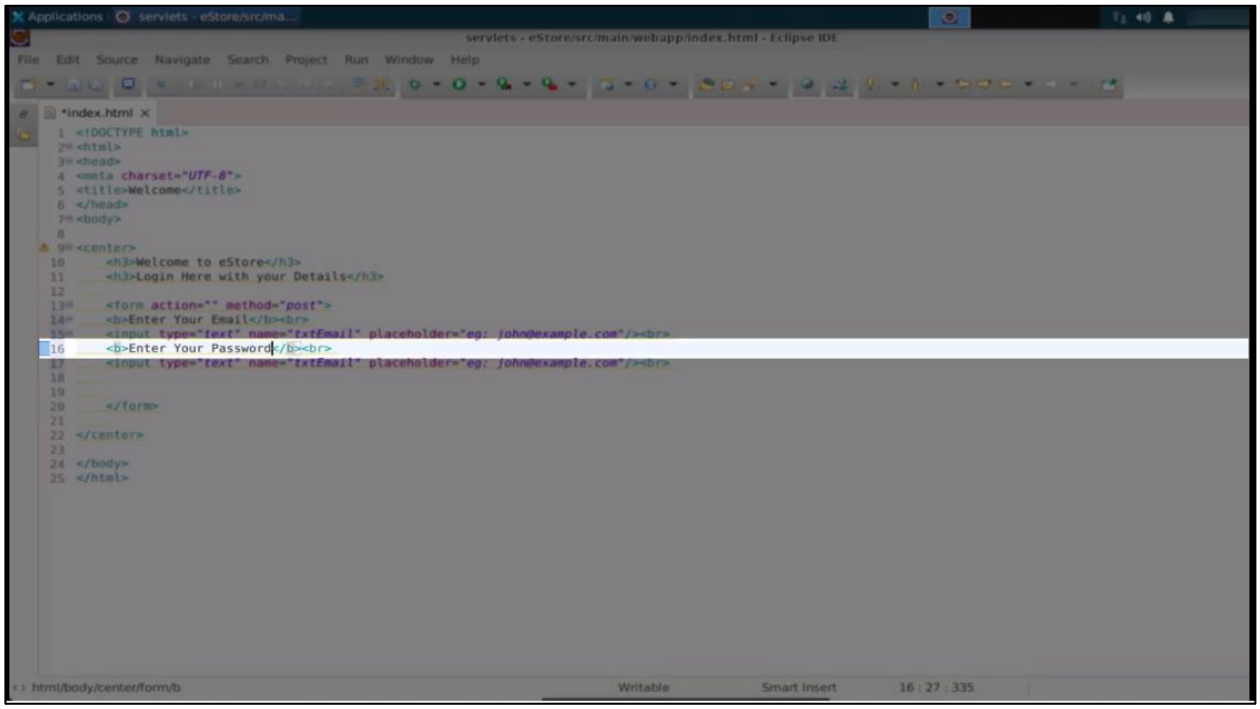
1.13 Copy the input fields from the form



1.14 Then, paste it as shown below:

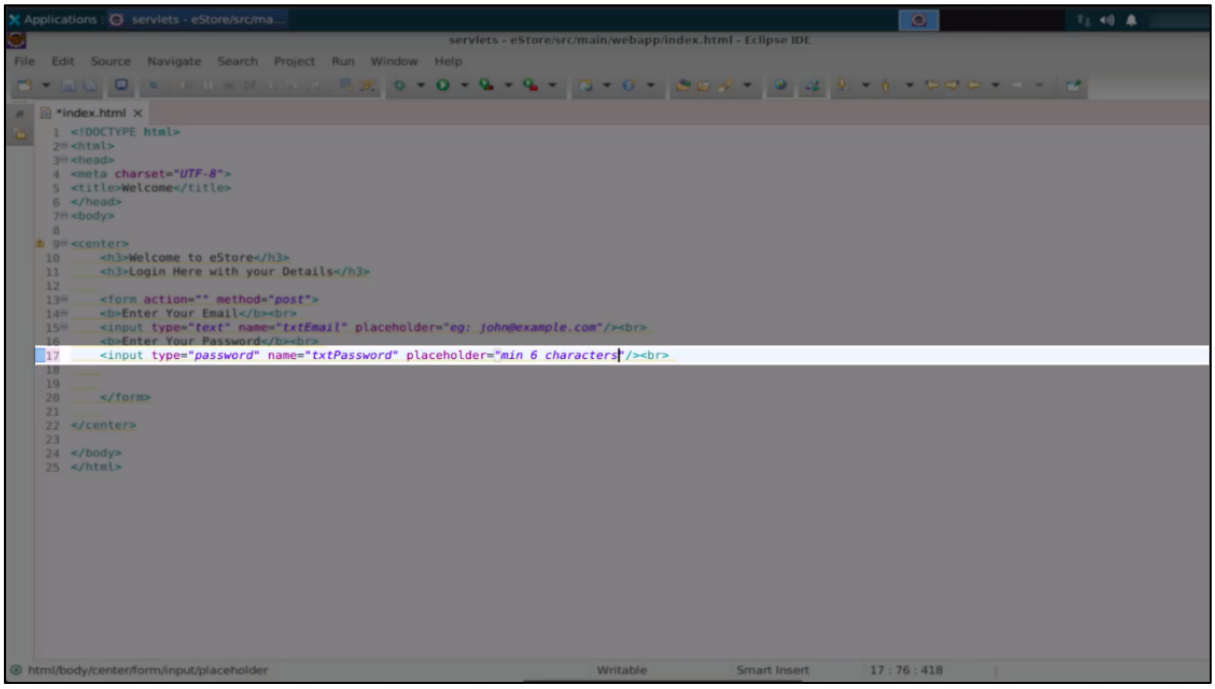


1.15 Replace **Enter Your Email** with **Enter Your Password**



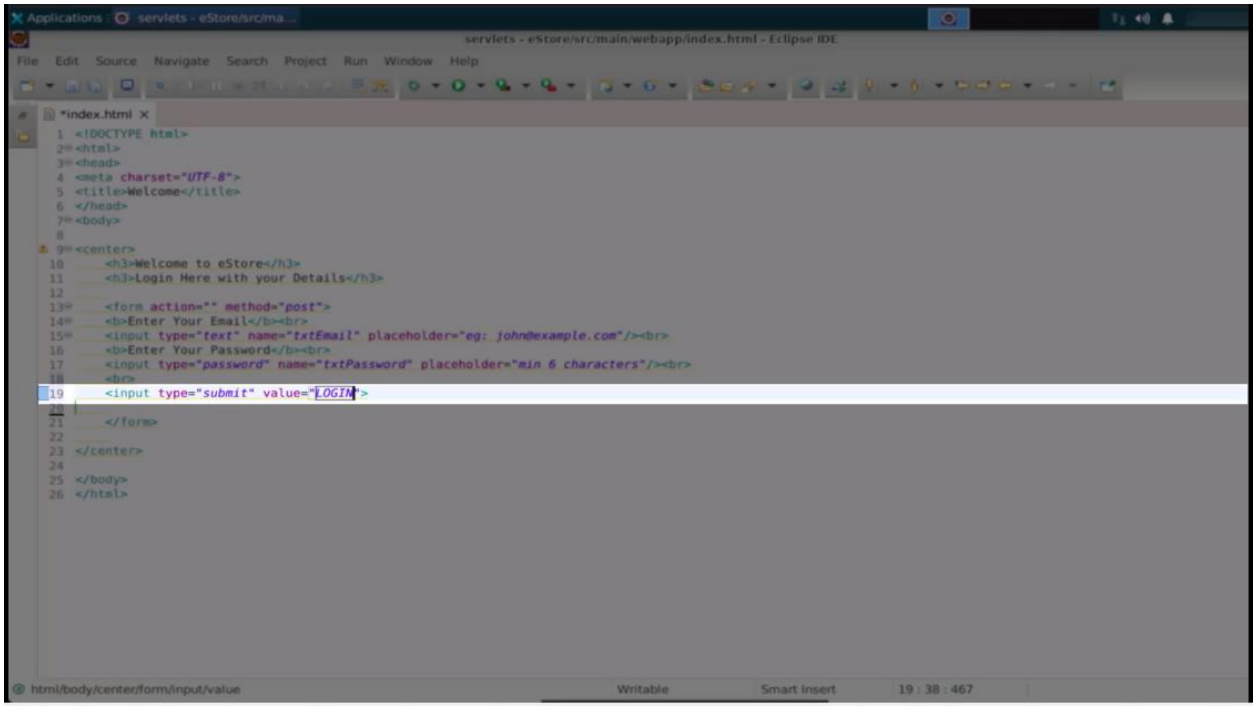
1.16 Enter the given line of code in the input tag for the password functionality:

**<input type="password" name="txtPassword" placeholder="min 6 characters"/>
**

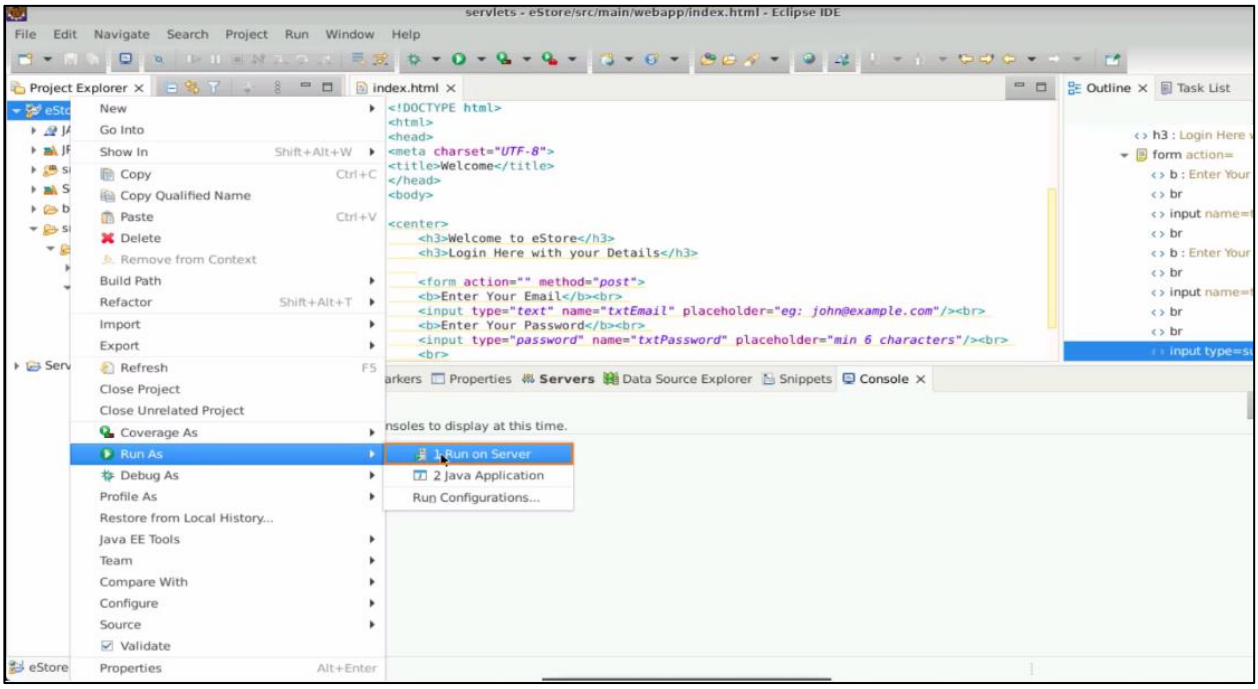


1.17 Enter the given line of code in the input tag for the login button:

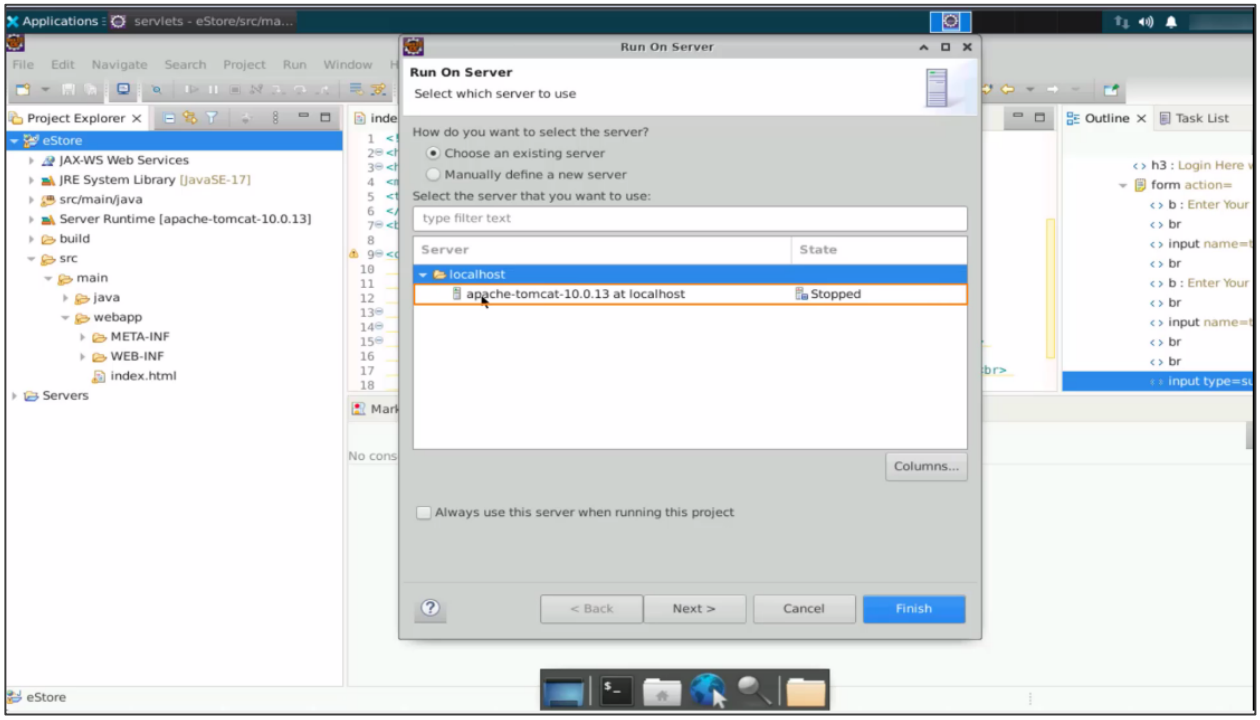
<input type="submit" value="LOGIN">



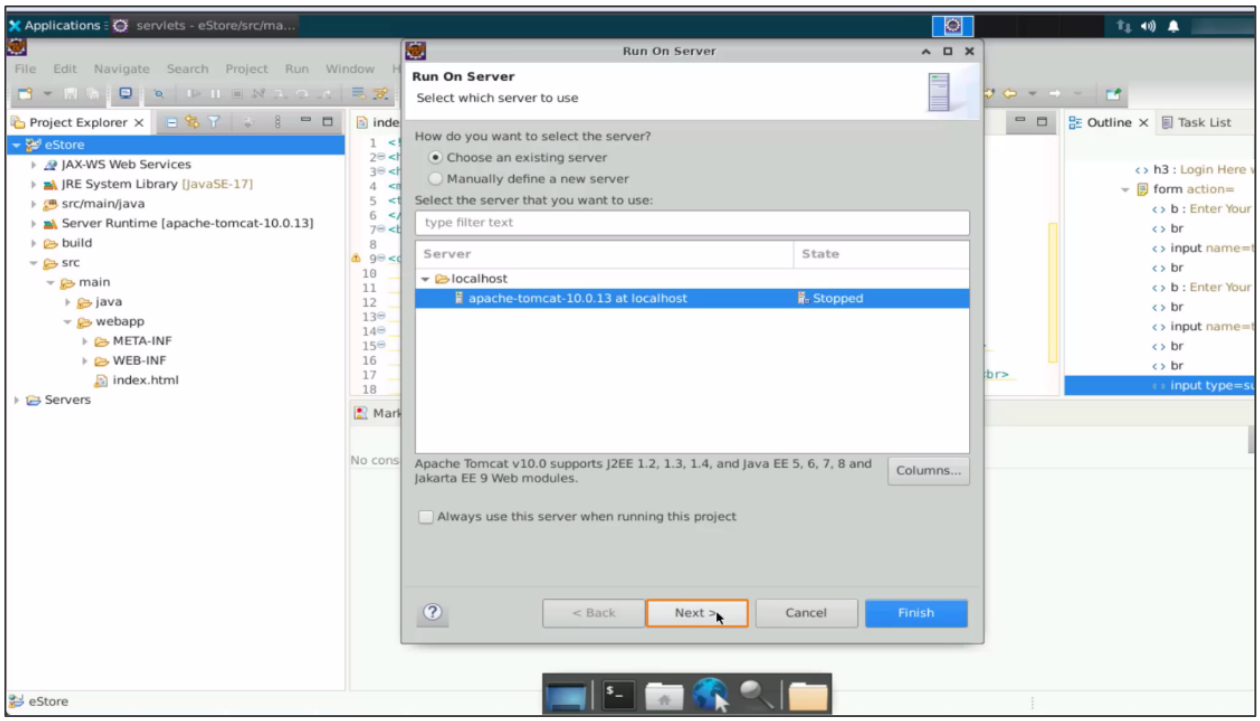
1.18 Now, right-click on the **eStore** project name, select **Run As,** and click **Run on Server**



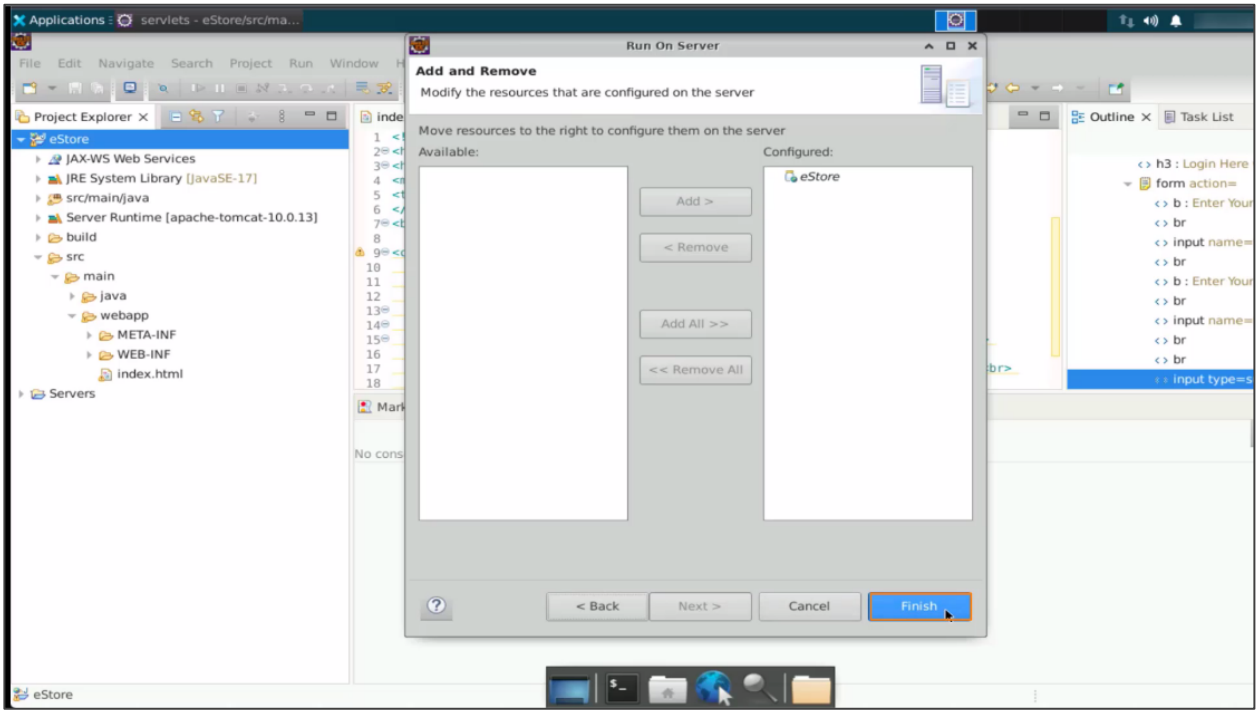
1.19 Choose the **apache-tomcat-10.0.13** server



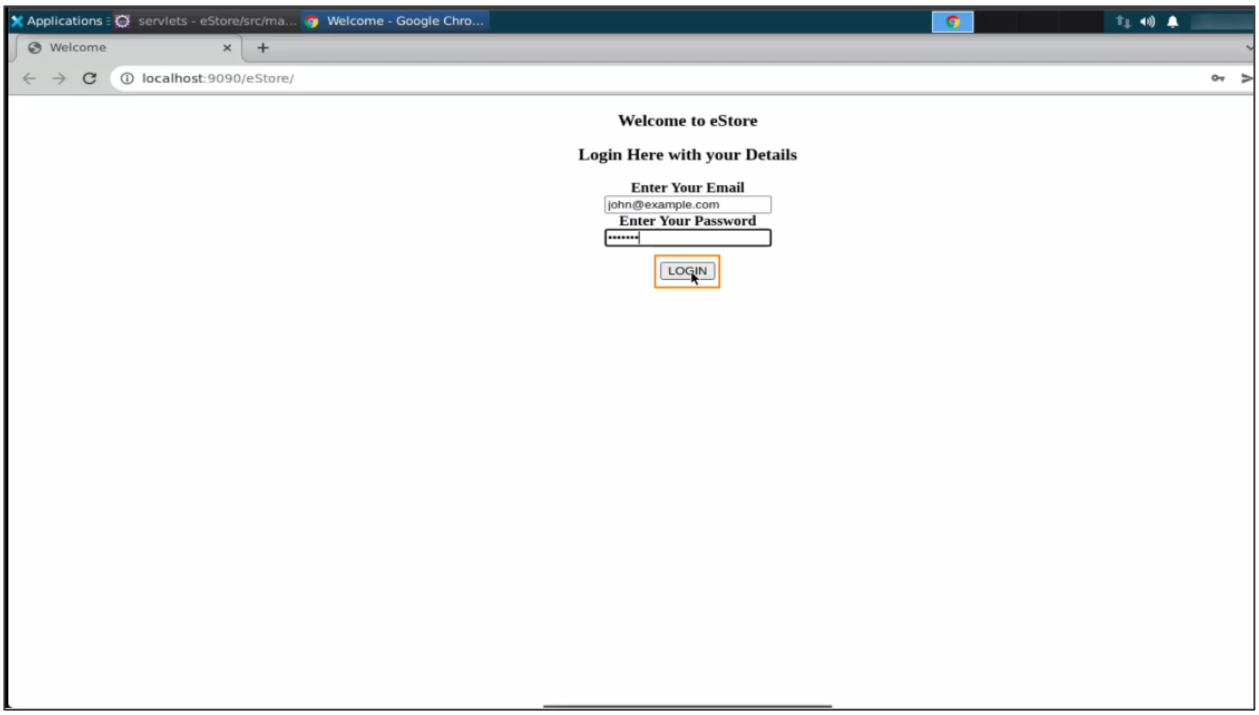
1.20 Click on **Next**



1.21 Click on **Finish**

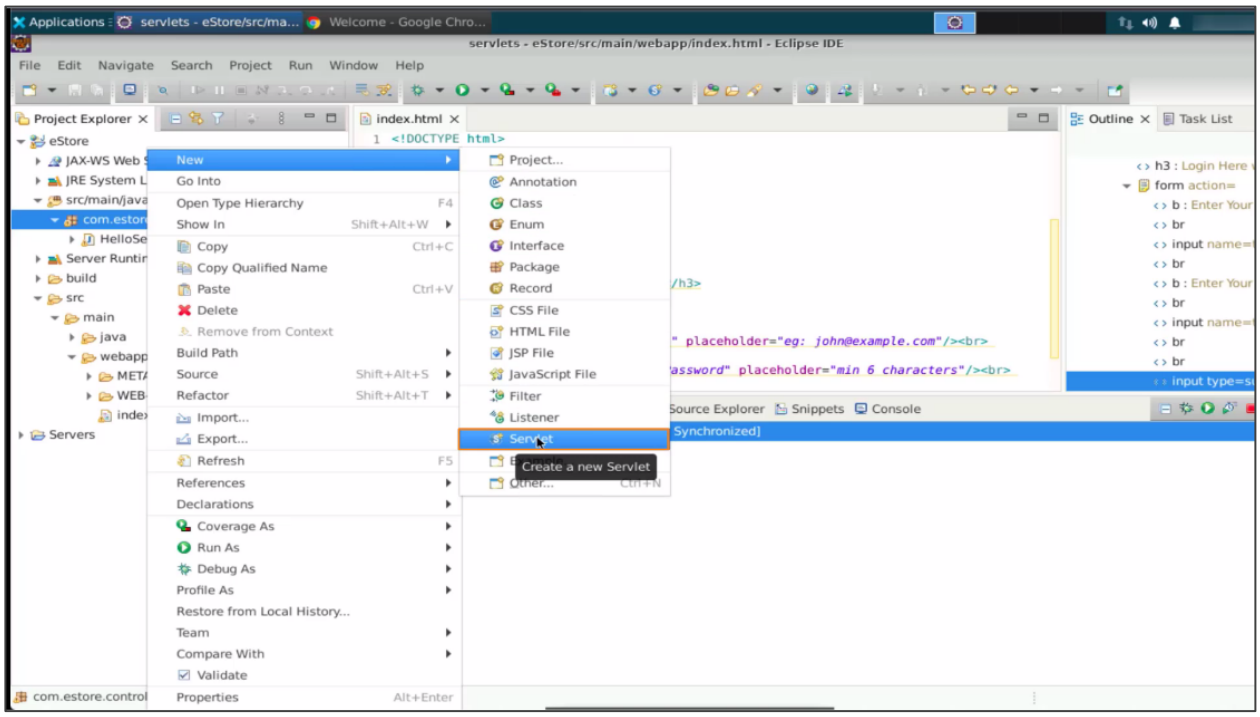


1.22 Enter the details and click on **LOGIN**

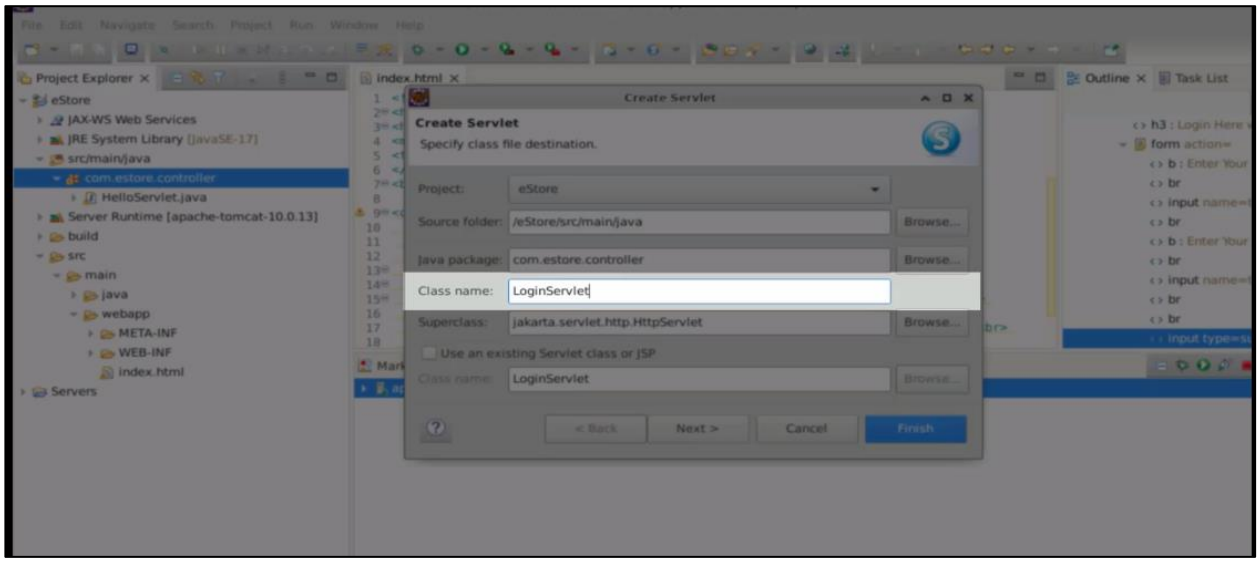


Step 2: Create a new Servlet for the login form

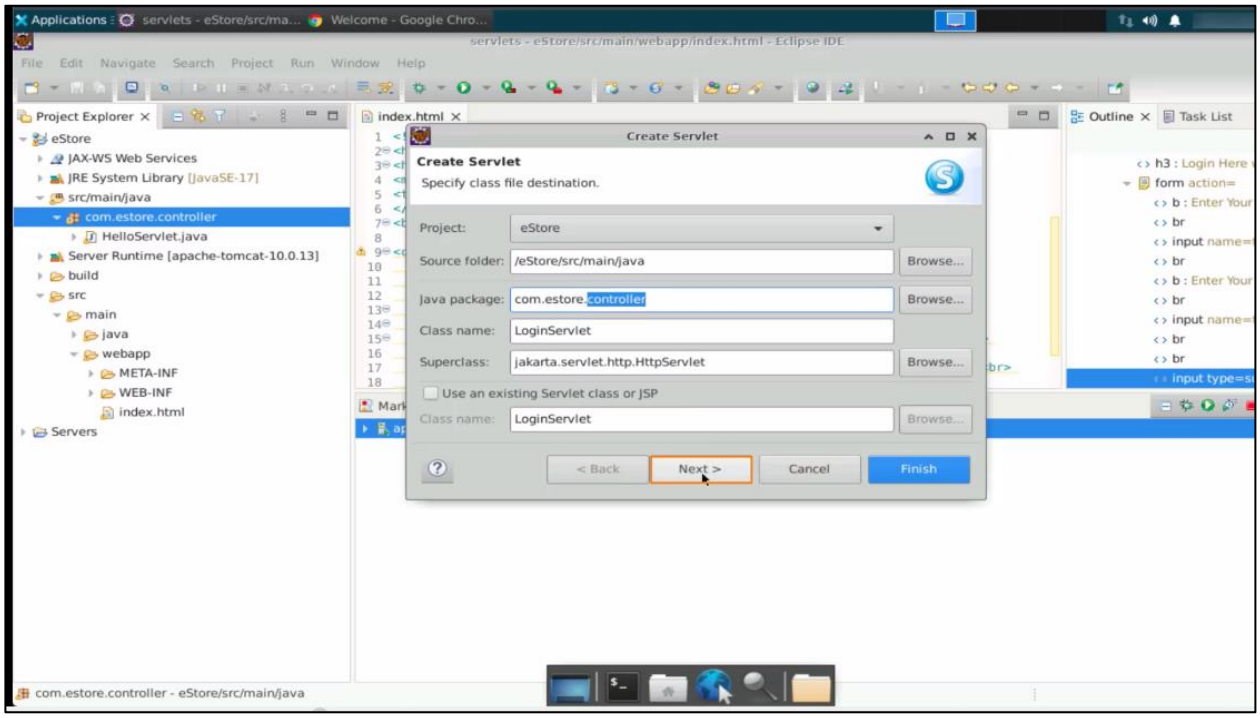
2.1 Go to the **controller package**, right-click to select **New**, and click **Servlet**



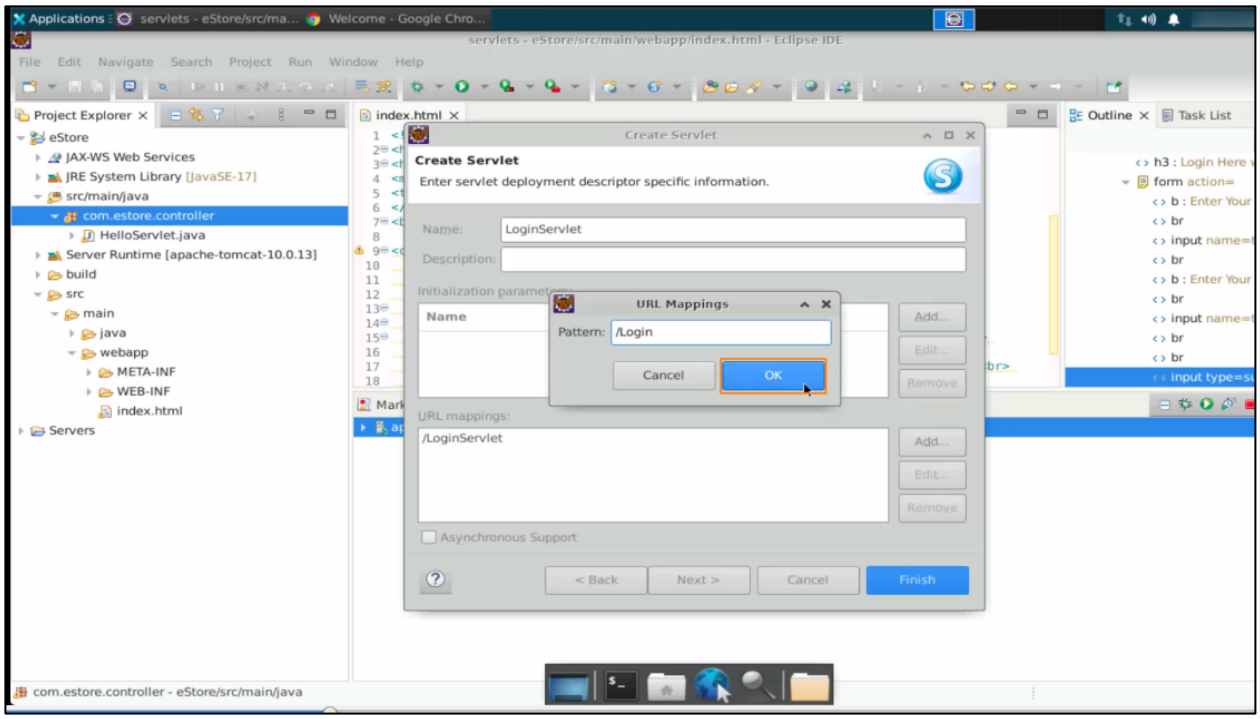
2.2 Enter **LoginServlet** in the **Class name** field



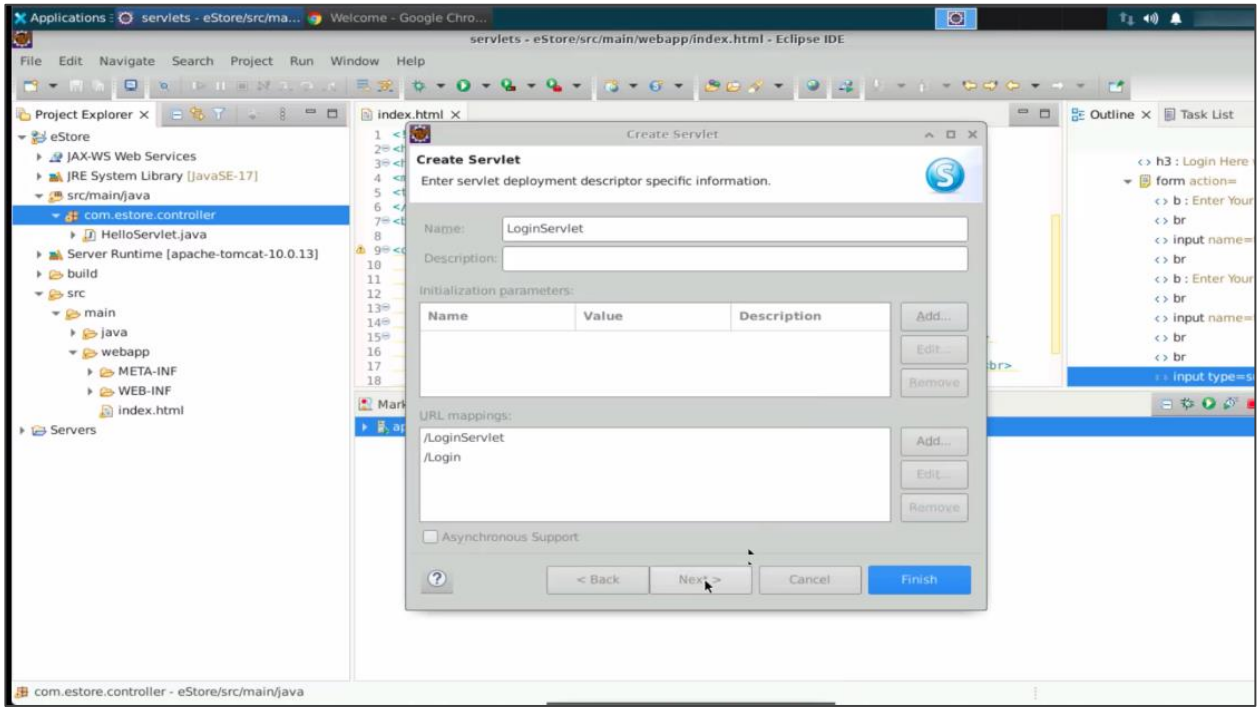
2.3 Click **Next**



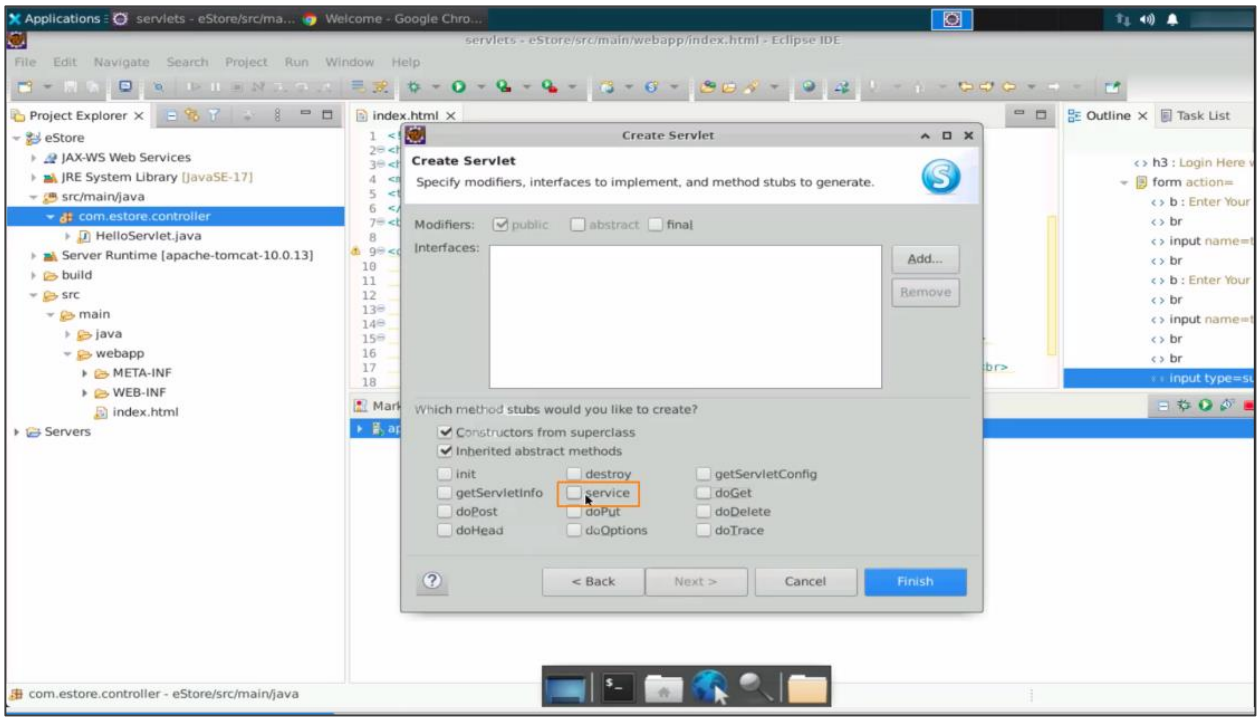
2.4 Enter the URL mapping as **/Login,** then click **OK**



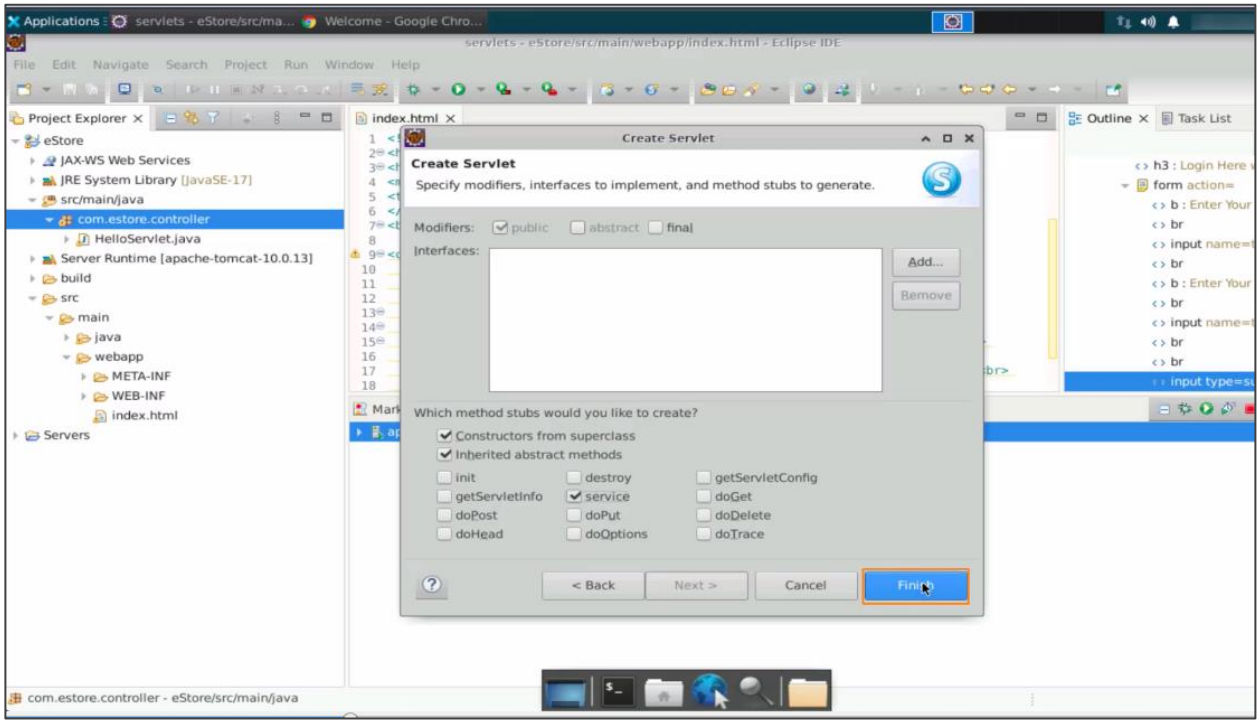
2.5 Click on **Next**



2.6 Select only the **service** method which can work on both the GET and POST methods

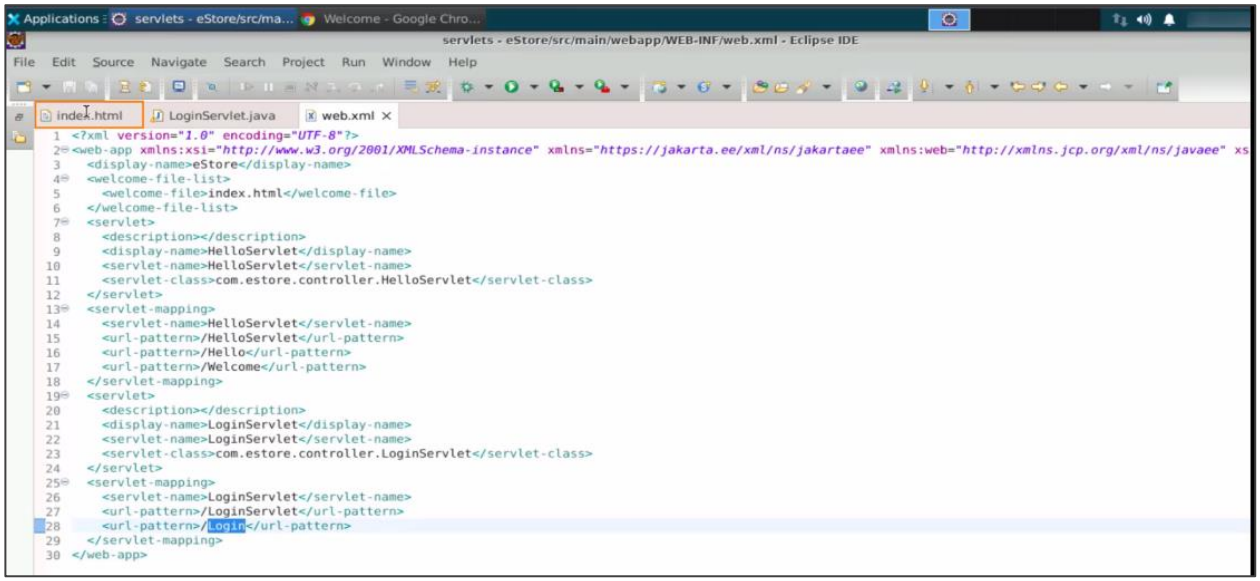


2.7 Click **Finish**

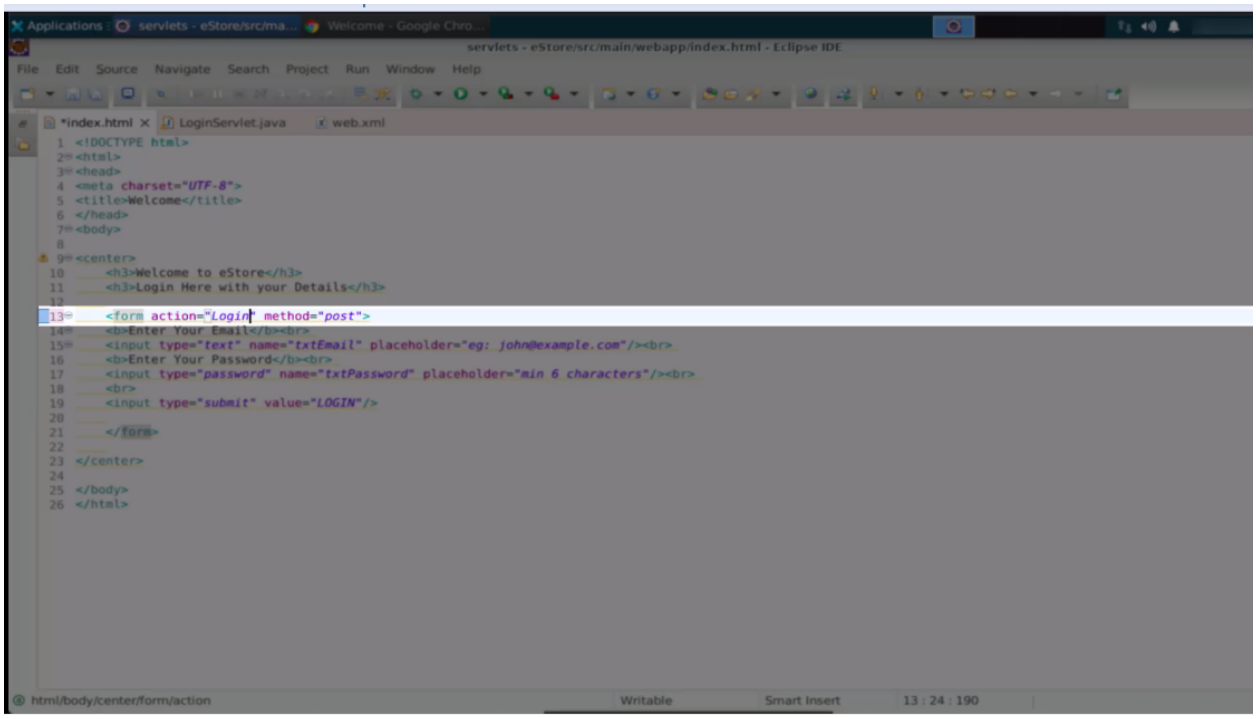


Step 3: Create a method in the login Servlet for the HTTP Servlet request

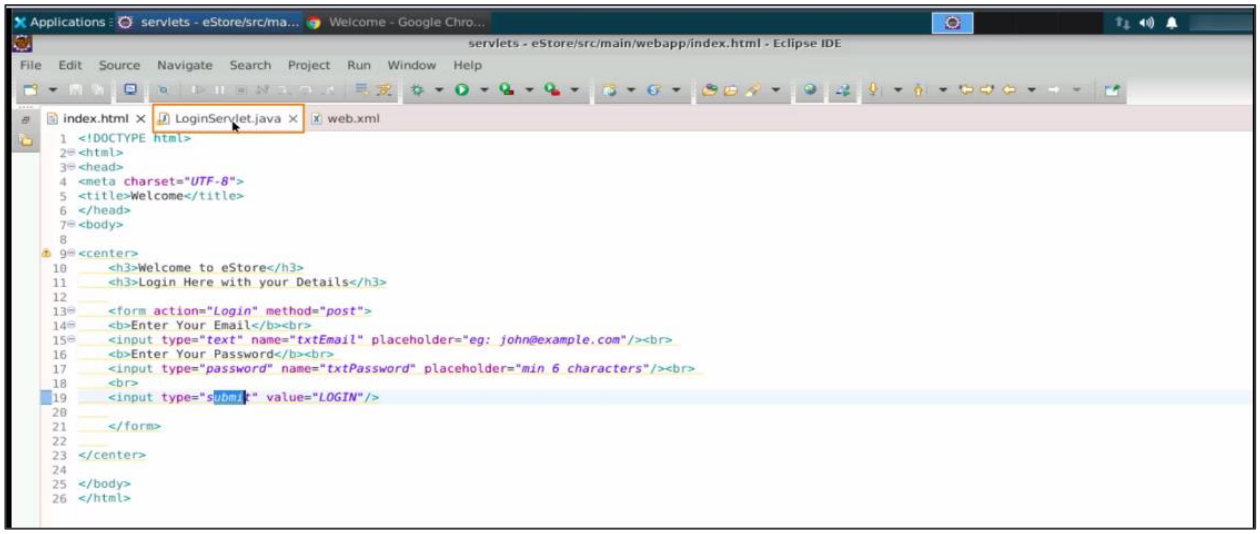
3.1 Go to the **index.html** file



3.2 Enter action= **Login**. As a result, whenever you submit data, it will be submitted to the login Servlet.



3.3 Open the **LoginServlet.java** file



3.4 Enter **getParameter()** to capture the **email**

```
package com.estore.controller;

import jakarta.servlet.ServletException;

/**
 * Servlet implementation class LoginServlet
 */
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String email = request.getParameter("txtEmail");

    }

}
```

3.5 Enter **getParameter()** to capture the **password**

```
package com.estore.controller;

import jakarta.servlet.ServletException;

/**
 * Servlet implementation class LoginServlet
 */
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // Read the data from request object
        String email = request.getParameter("txtEmail");
        String password = request.getParameter("txtPassword");

    }

}
```

3.6 Enter **System.out.println()** to print the details on the console

```
package com.estore.controller;

import jakarta.servlet.ServletException;

/**
 * Servlet implementation class LoginServlet
 */
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // Read the data from request object
        String email = request.getParameter("txtEmail");
        String password = request.getParameter("txtPassword");

        System.out.println("User Details: "+email+" "+password);

    }

}
```

3.7 To send **acknowledgment** from the Servlet, enter the **contentType** command and set the content type for the **response object**

```
package com.estore.controller;

import jakarta.servlet.ServletException;

/**
 * Servlet implementation class LoginServlet
 */
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // Read the data from request object
        String email = request.getParameter("txtEmail");
        String password = request.getParameter("txtPassword");

        System.out.println("User Details: "+email+" "+password);
        response.setContentType("text/html");

    }

}
```

3.8 Create an HTML response by entering the **HTML response** string

```
package com.estore.controller;

import jakarta.servlet.ServletException;

/**
 * Servlet implementation class LoginServlet
 */
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // Read the data from request object
        String email = request.getParameter("txtEmail");
        String password = request.getParameter("txtPassword");

        System.out.println("User Details: "+email+" "+password);

        response.setContentType("text/html");
        String htmlResponse = "<center><h3>Welcome "+email+"</h3></center>";

    }
}
```

3.9 Enter the **LoginTimeStamp** function to capture the date and to convert it into a string

```
package com.estore.controller;

import jakarta.servlet.ServletException;

/**
 * Servlet implementation class LoginServlet
 */
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // Read the data from request object
        String email = request.getParameter("txtEmail");
        String password = request.getParameter("txtPassword");

        System.out.println("User Details: "+email+" "+password);

        response.setContentType("text/html");
        String loginTimeStamp = new Date().toString();
        String htmlResponse = "<center><h3>Welcome "+email+"</h3></center>";

    }
}
```

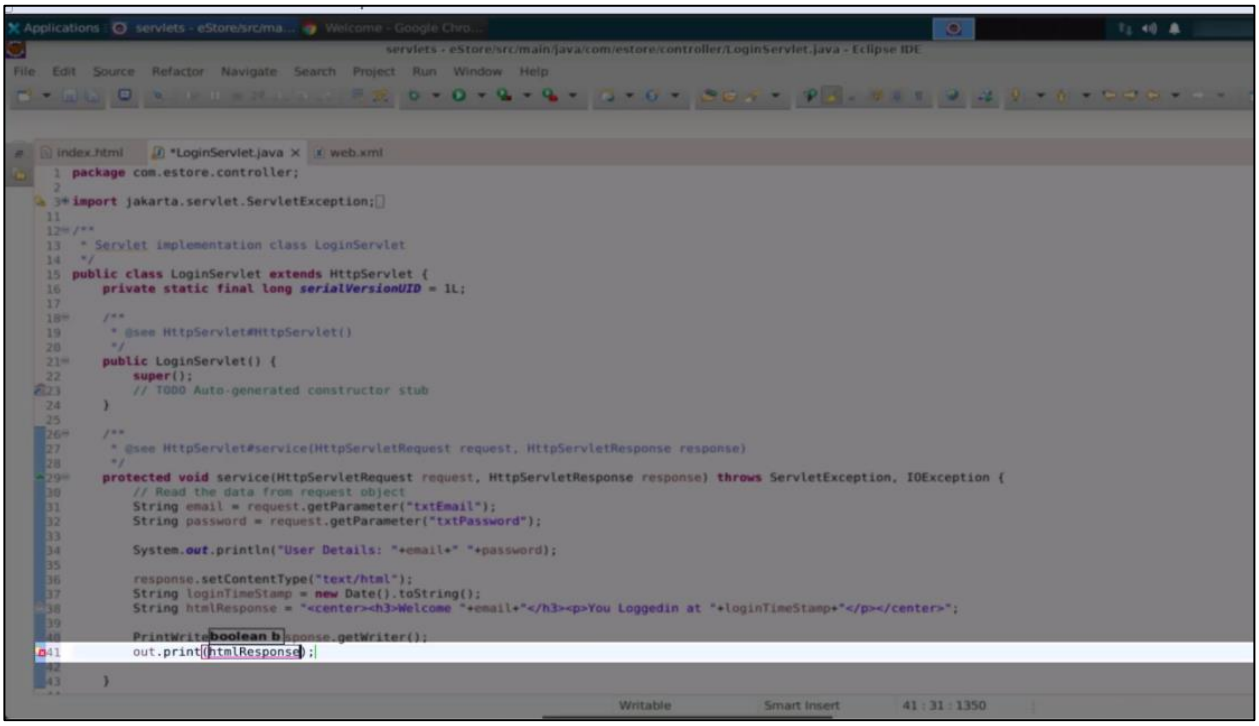
3.10 Add a **paragraph** tag for **LoginTimeStamp**

```java
package com.estore.controller;

import jakarta.servlet.ServletException;

/**
 * Servlet implementation class LoginServlet
 */
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // Read the data from request object
        String email = request.getParameter("txtEmail");
        String password = request.getParameter("txtPassword");

        System.out.println("User Details: "+email+" "+password);

        response.setContentType("text/html");
        String loginTimeStamp = new Date().toString();
        String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"</p></center>";

    }

}
```

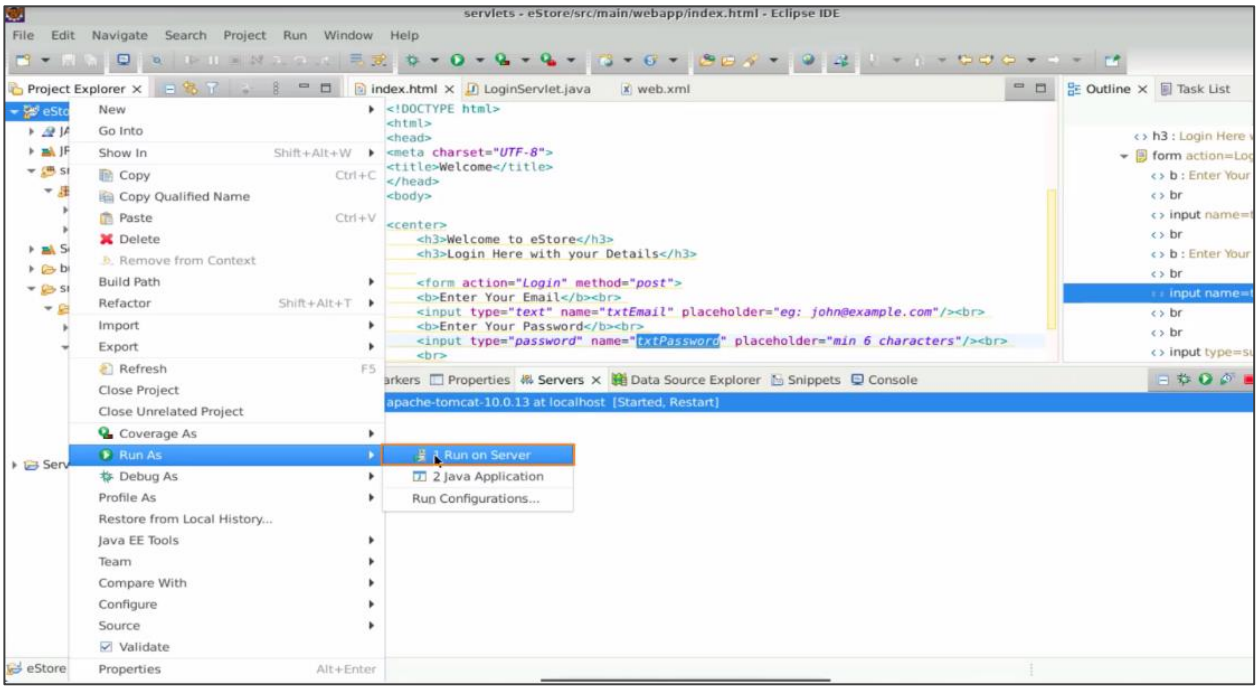
3.11 Enter **PrintWriter** to capture the response

```java
import jakarta.servlet.ServletException;

/**
 * Servlet implementation class LoginServlet
 */
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // Read the data from request object
        String email = request.getParameter("txtEmail");
        String password = request.getParameter("txtPassword");

        System.out.println("User Details: "+email+" "+password);

        response.setContentType("text/html");
        String loginTimeStamp = new Date().toString();
        String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"</p></center>";

        PrintWriter out = response.getWriter();

    }
```

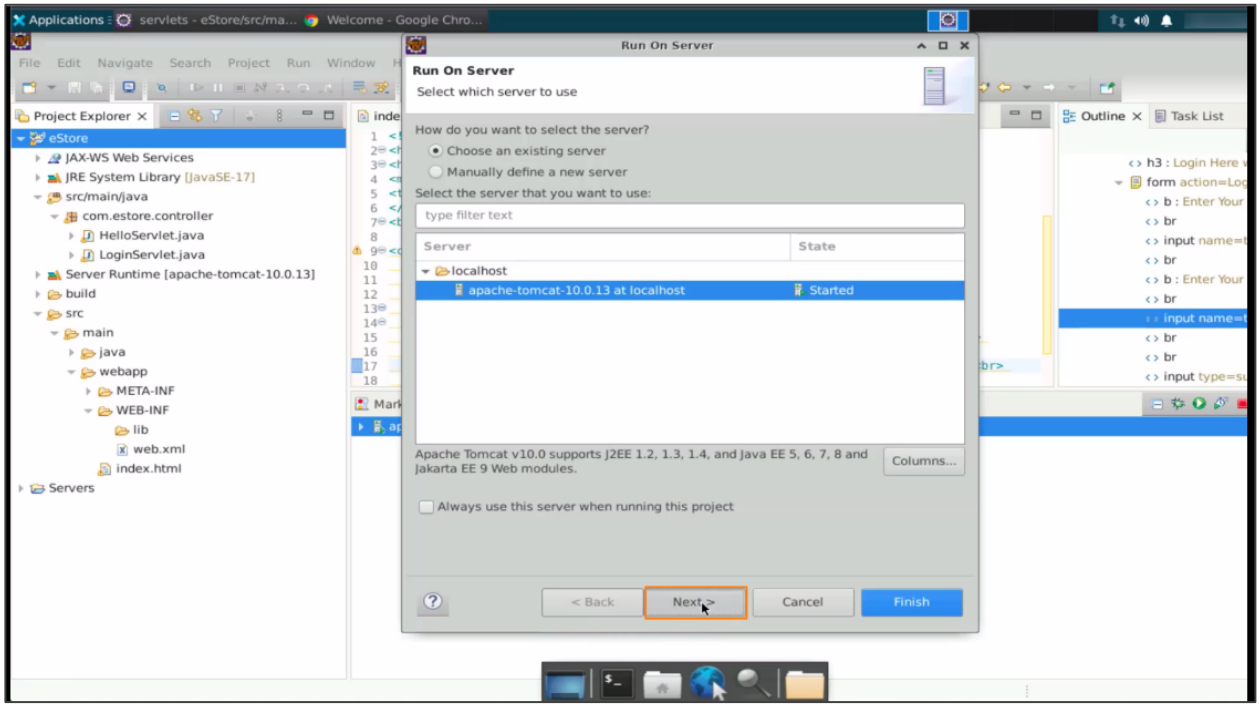
3.12 Enter the **out.print** method to map the index form



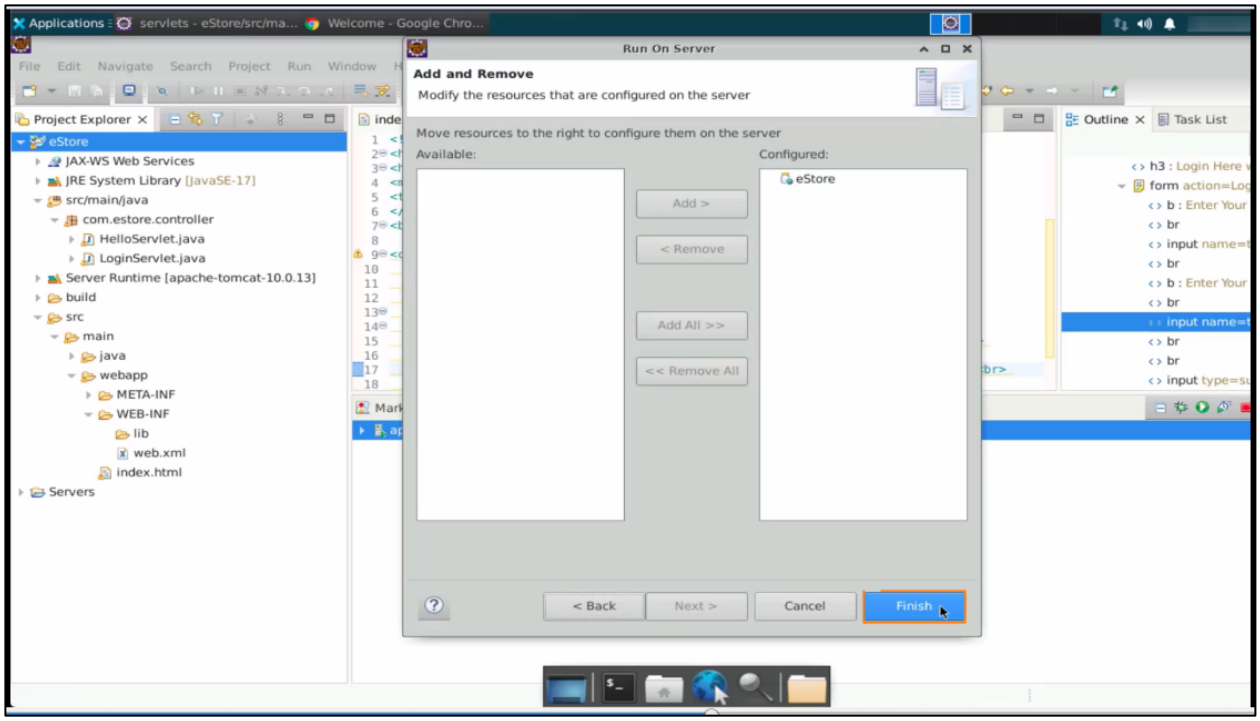
3.13 To run the project on the server, right-click on the **project**, select **Run As**, and click on **Run on Server**



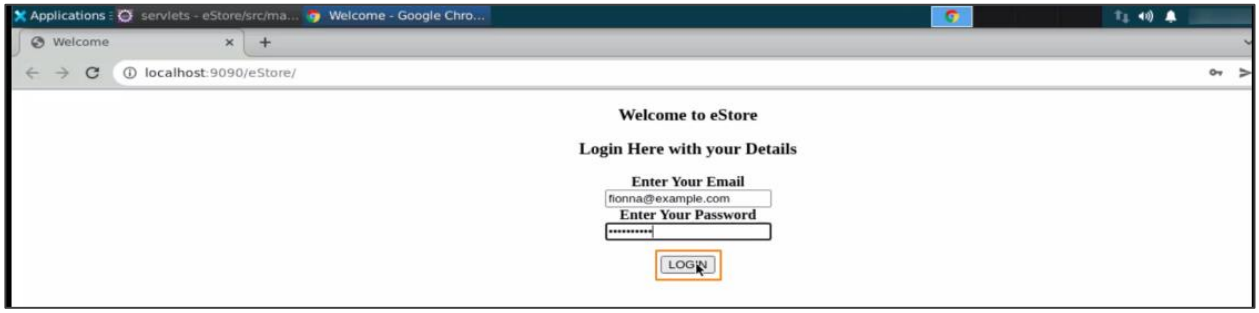
3.14 Select **apache-tomcat-10.0.13 at localhost**, then click on **Next**



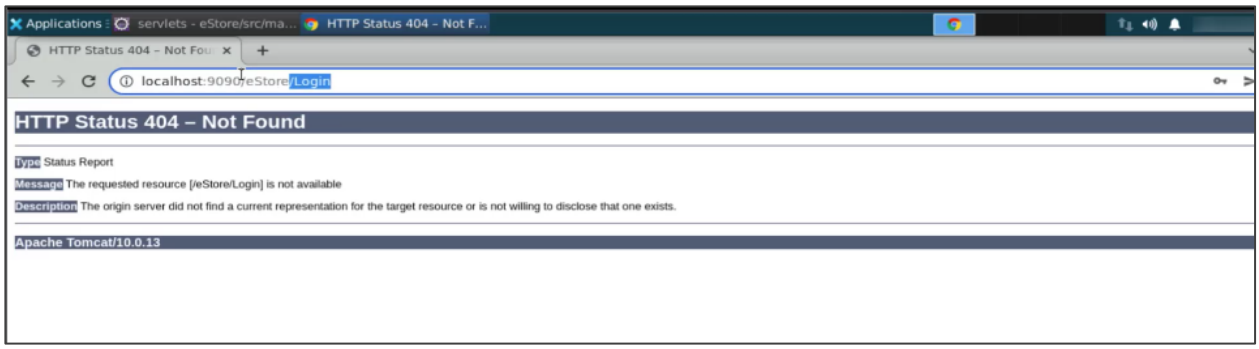
3.15 Click **Finish**



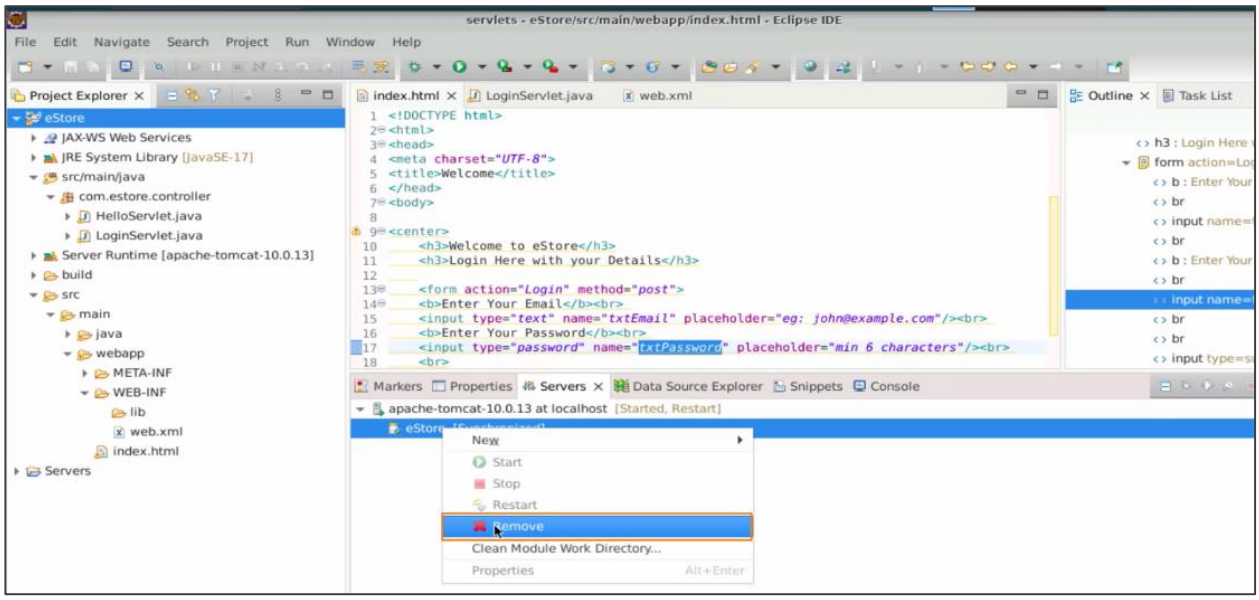
3.16 Refresh the page, enter the details, and click on the **LOGIN** button



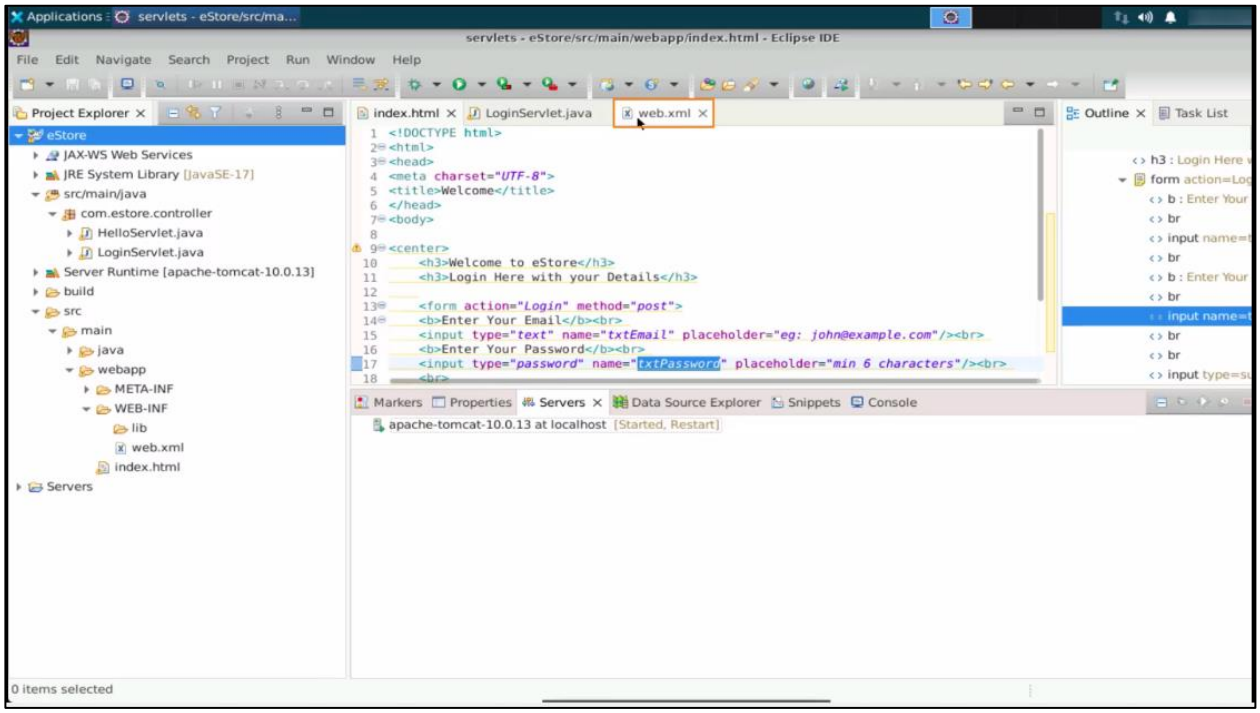
The URL is called successfully, but you will notice that project syncing is still needed.



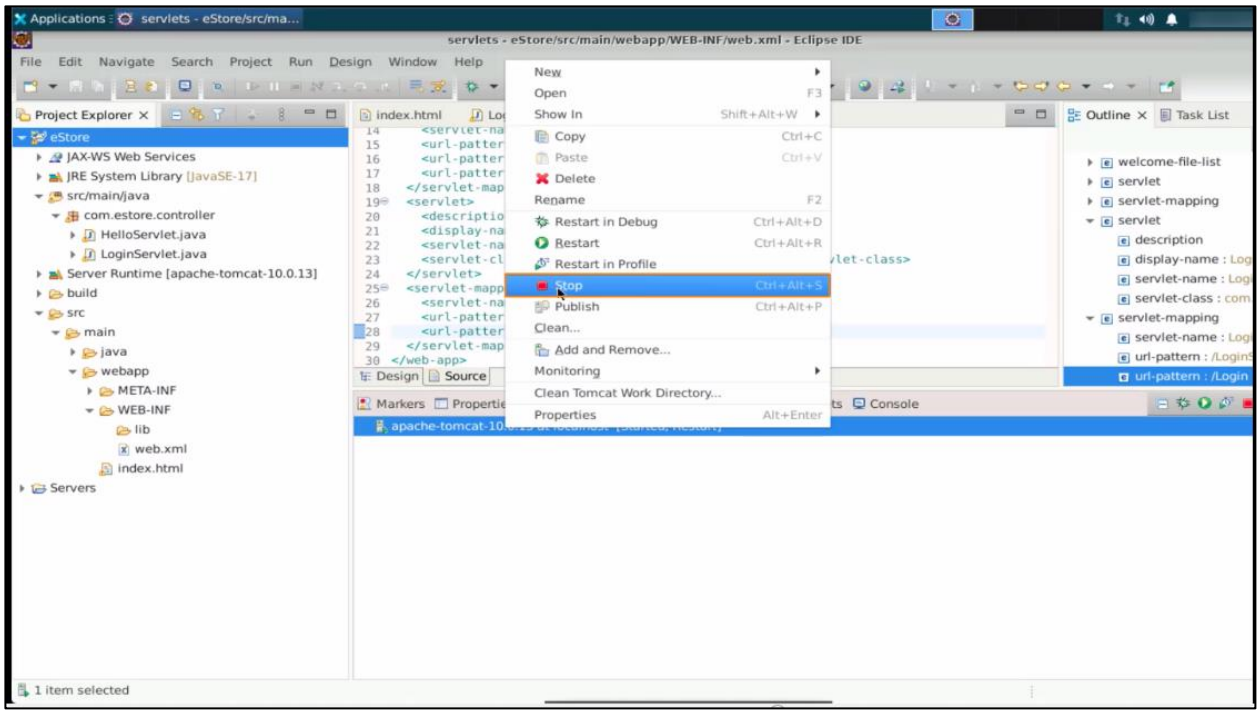
3.17 Go back to **Eclipse**, right-click on the server, and select **Remove**



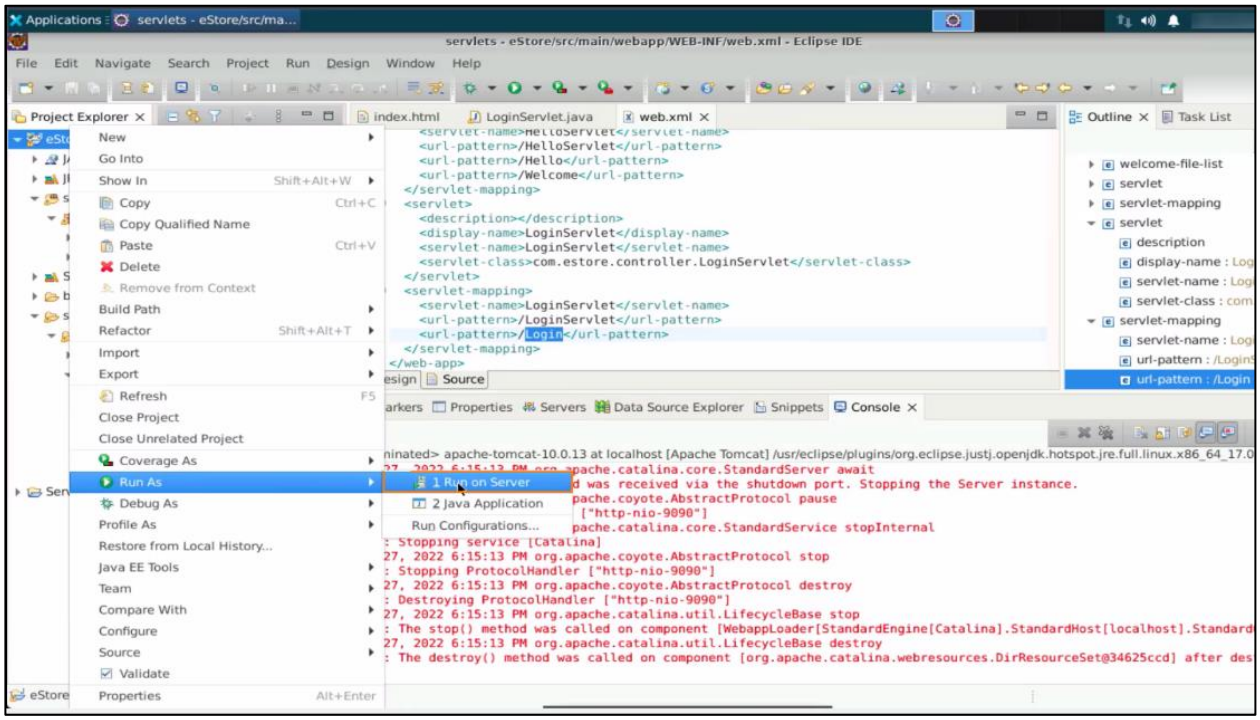
3.18 Go to the **web.xml** file



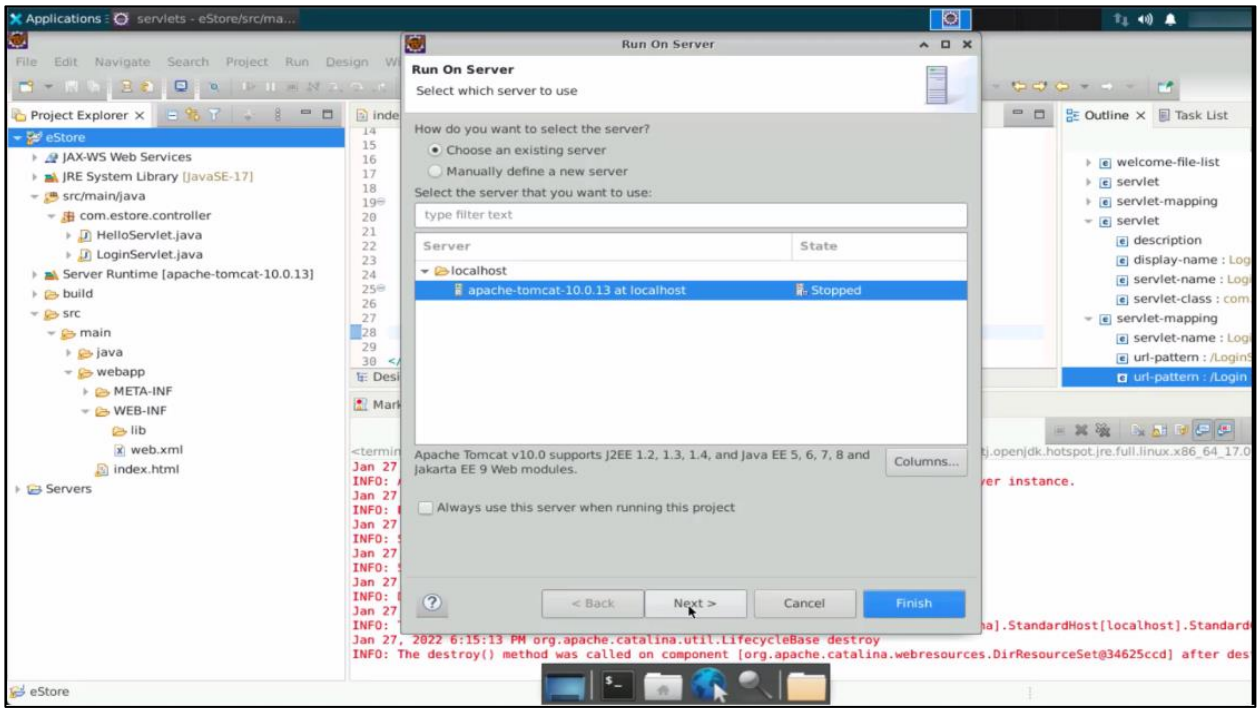
3.19 Stop the **server**



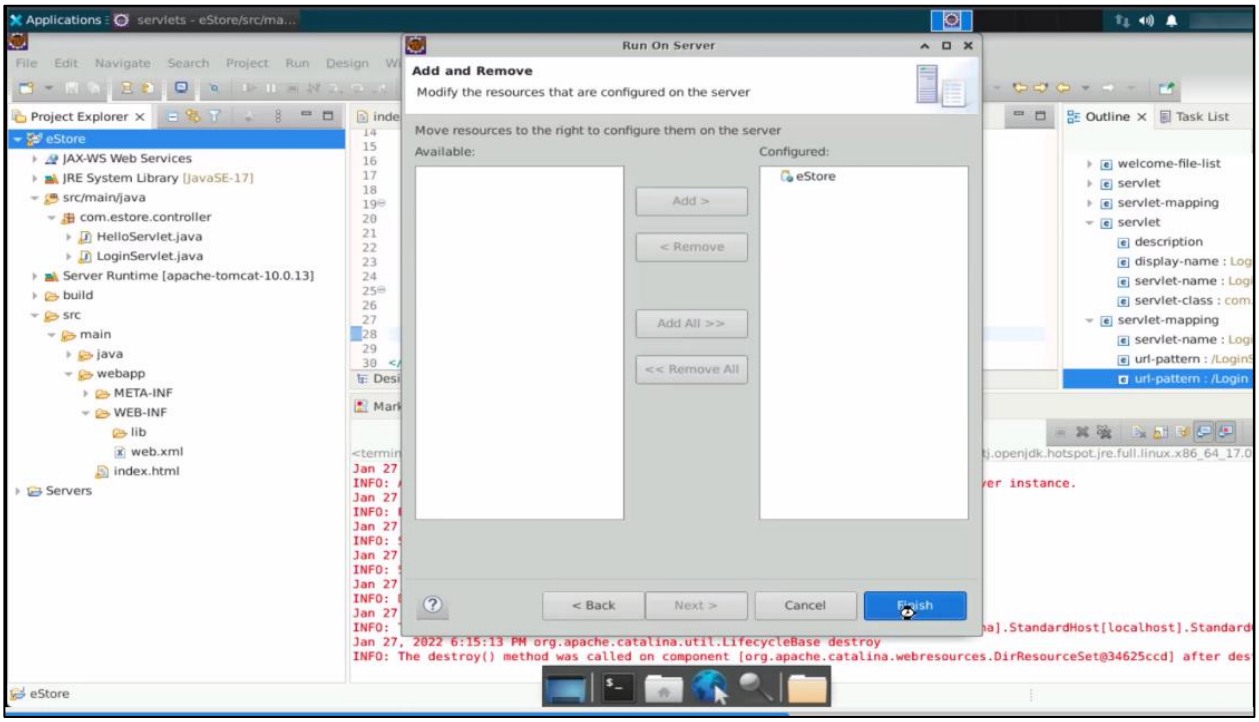
3.20 Now, right-click on the project name, select **Run As**, and click **Run on Server**



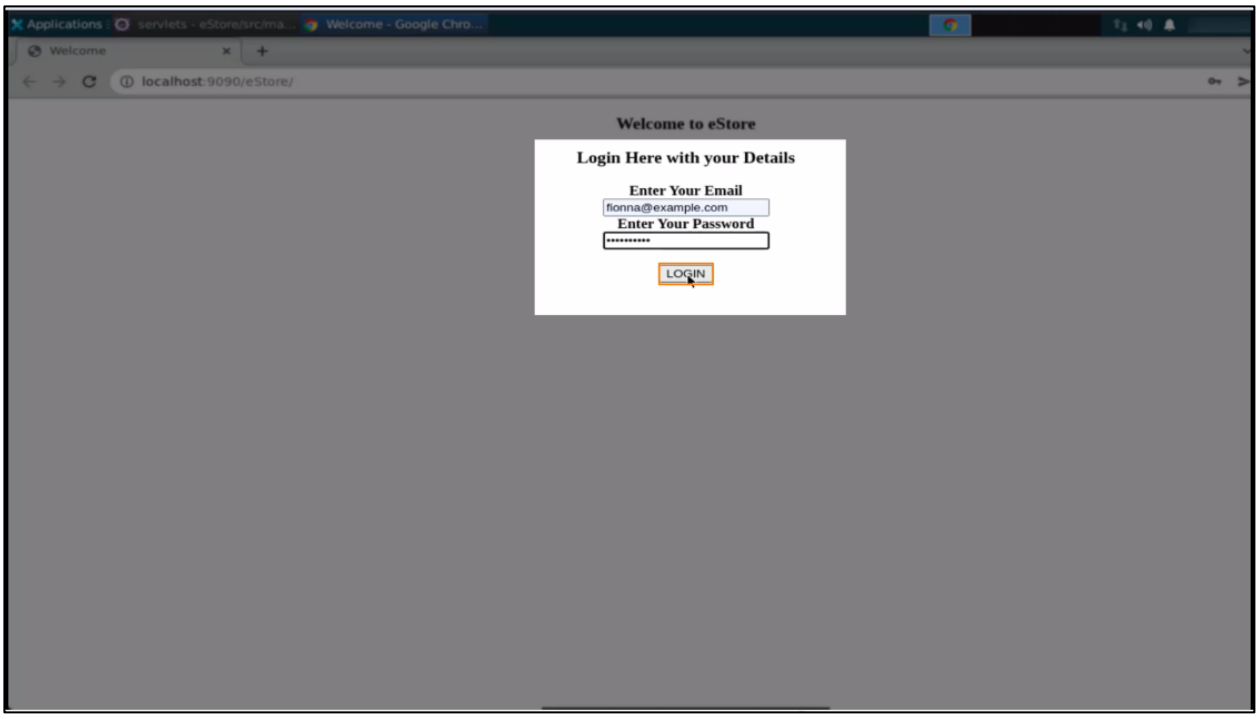
3.21 Select **apache-tomcat-10.0.13** and click **Next**



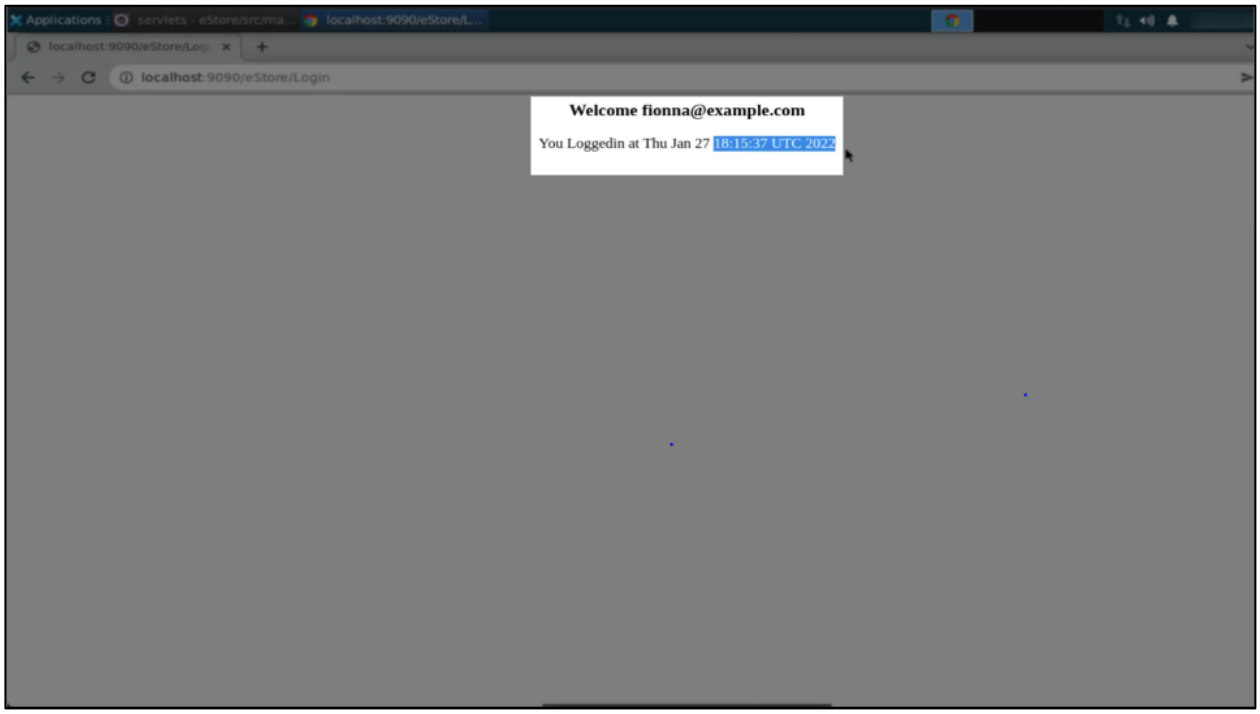
3.22 Click on **Finish**

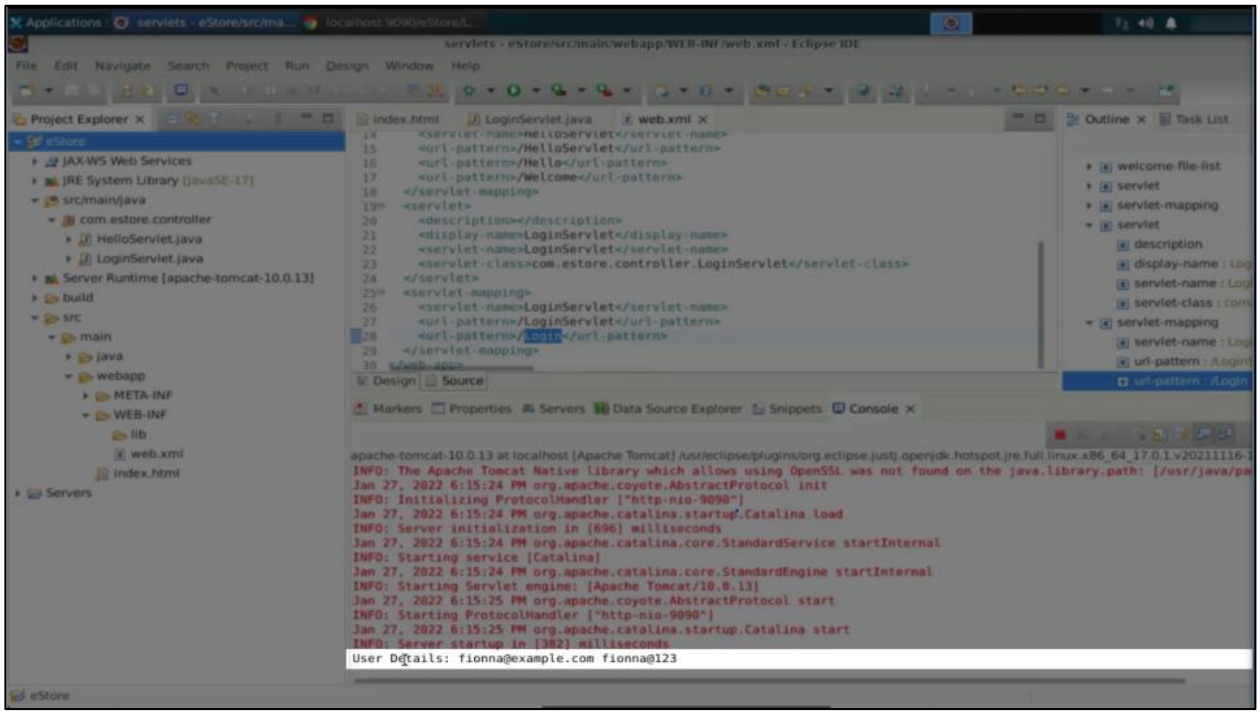


3.23 Refresh the page, enter the data, and click **LOGIN**



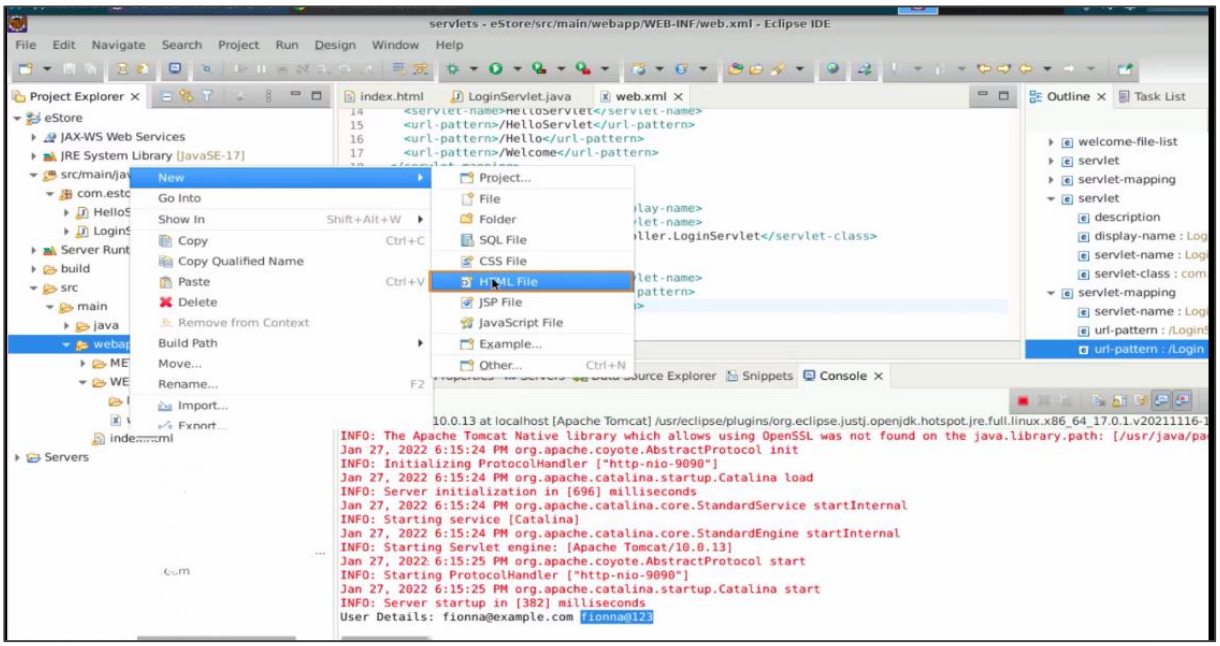
It has logged in successfully, and you will be able to see the user details in the **console.**



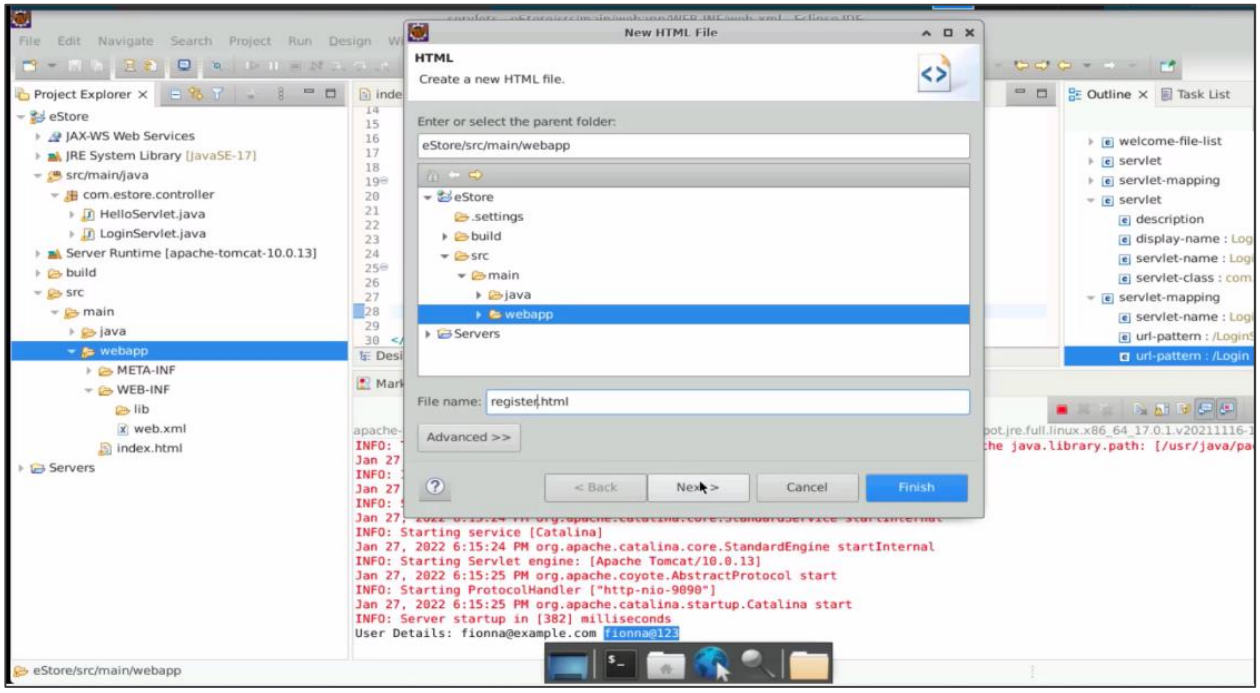


Step 4: Create a form to register the user

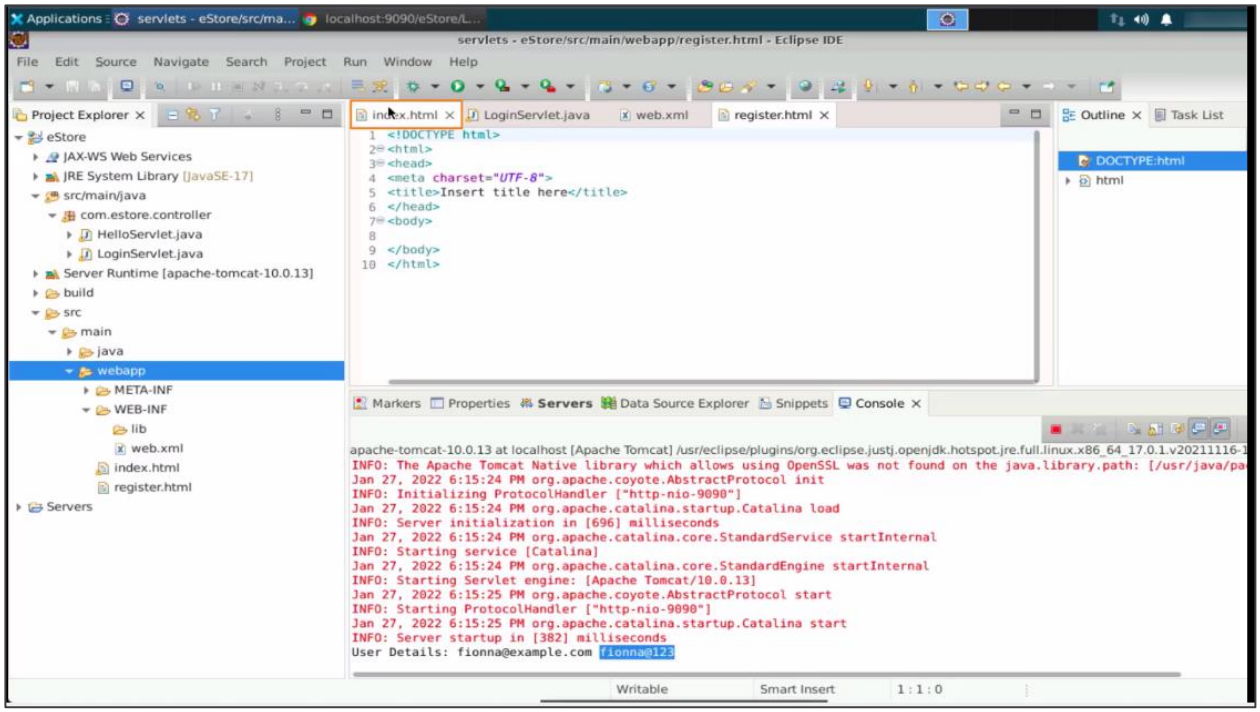
4.1 Create a new HTML file in **webapp**. Right-click on the **webapp**, select **New**, and click **HTML File**



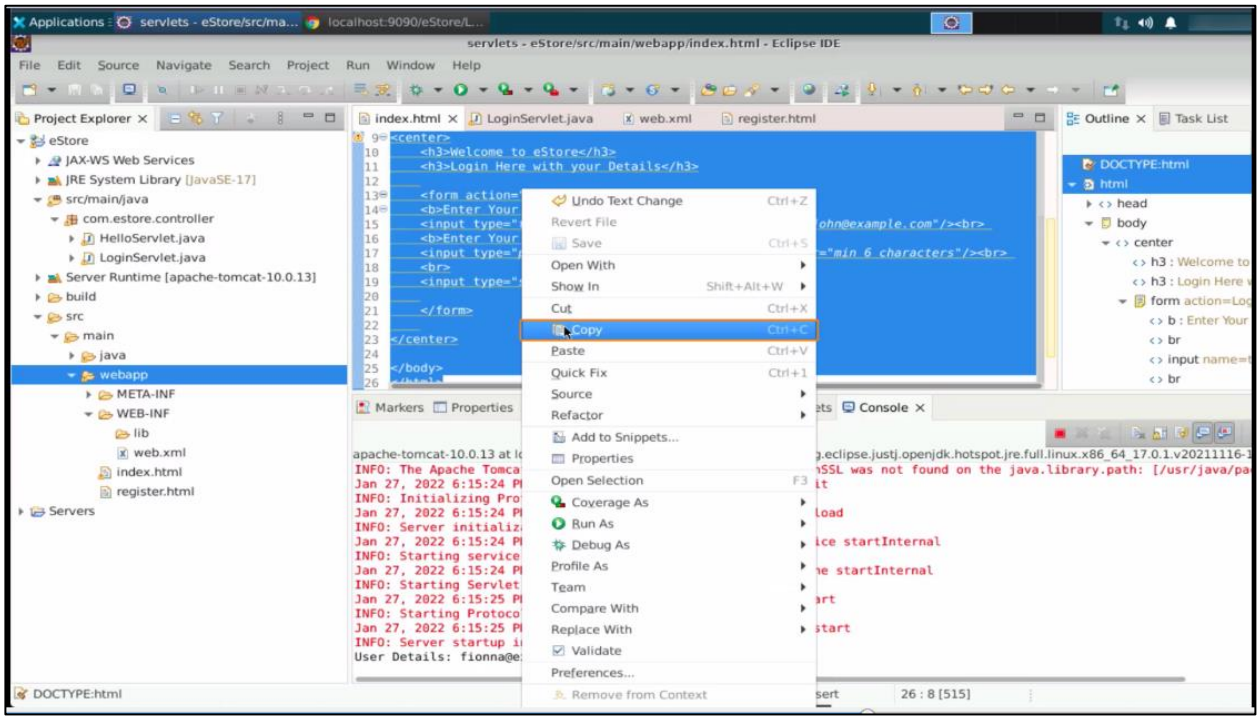
4.2 Name the file **register.html**, click on **Next**, and then click on **Finish**



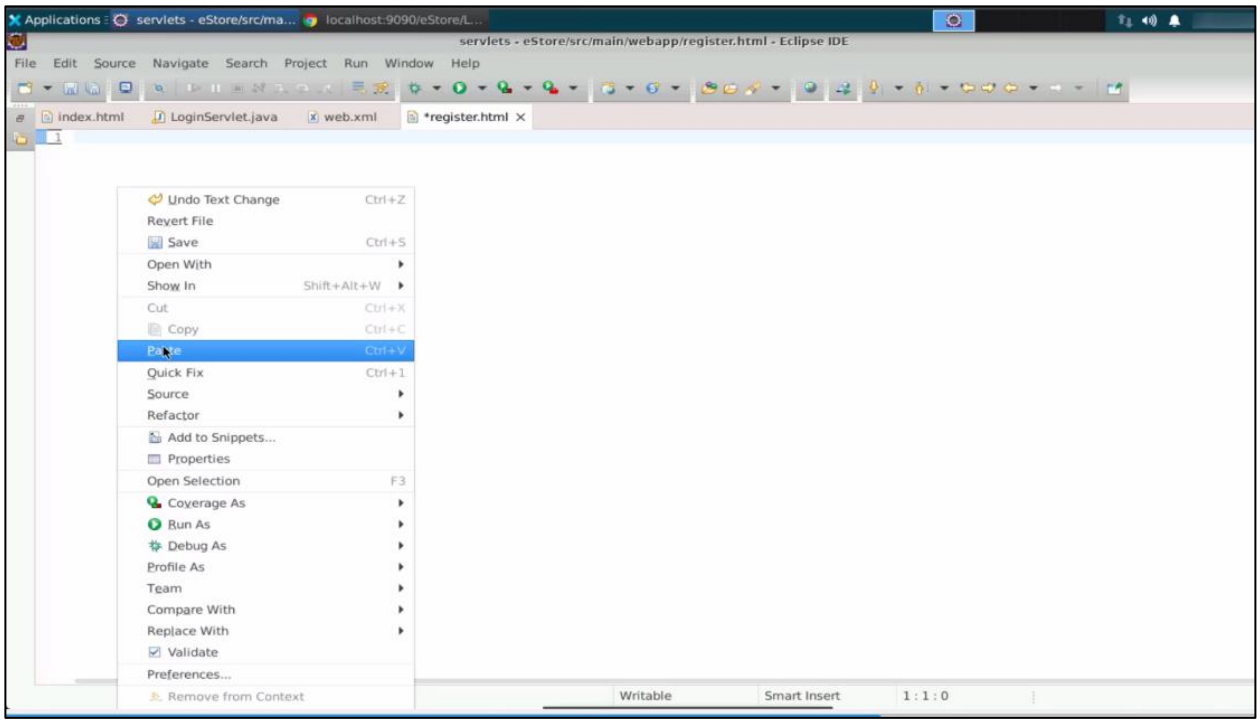
4.3 Return to the **index.html** file



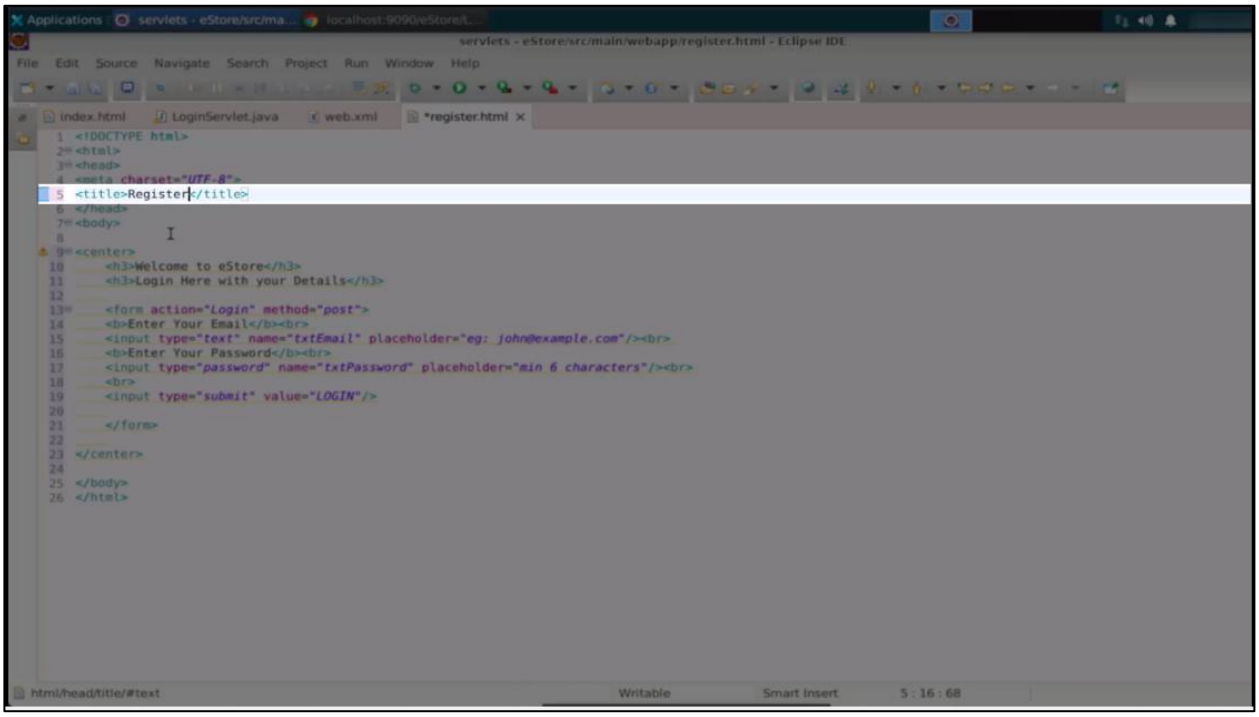
4.4 Copy the entire code



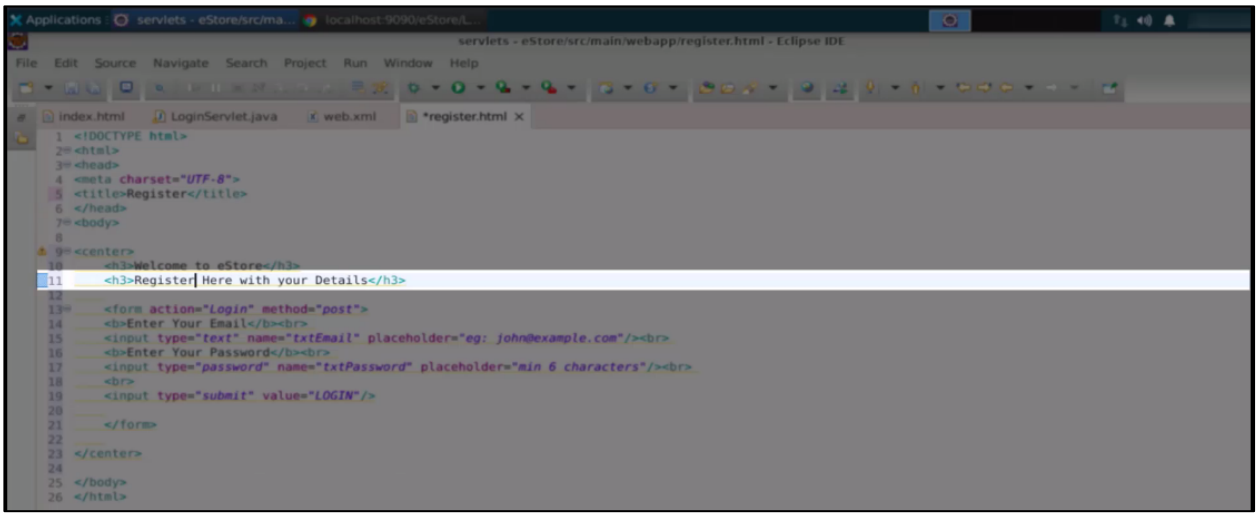
4.5 Paste the code in the **register.html** file



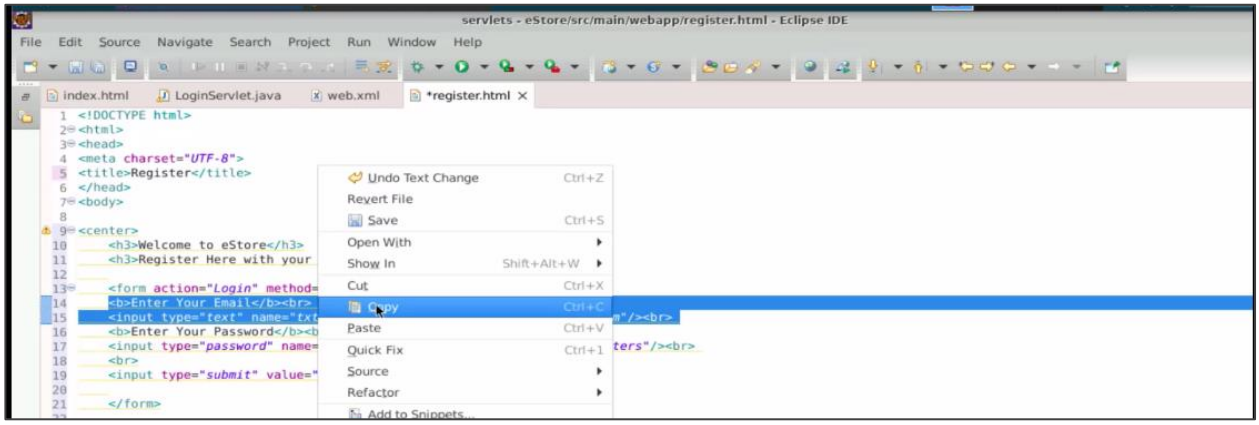
4.6 Enter **Register** in the title

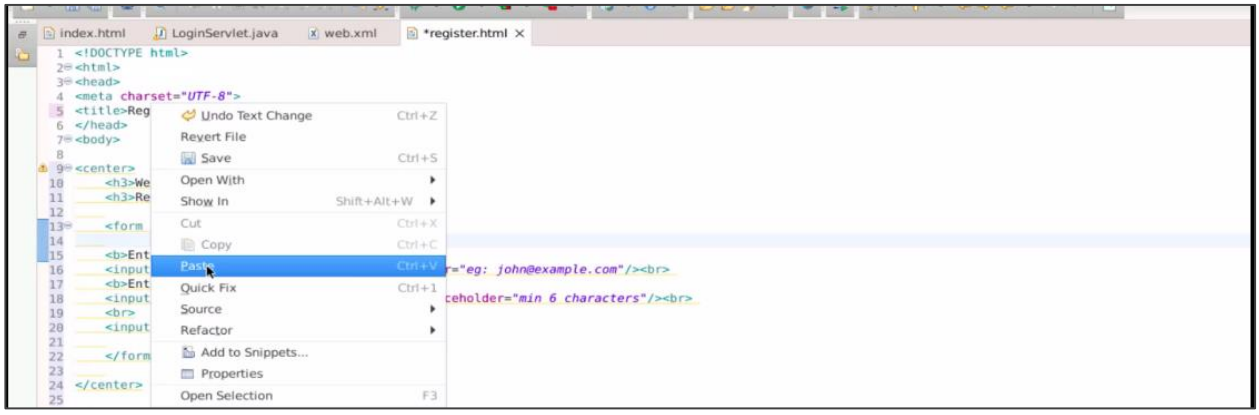


4.7 Replace **Login** with **Register** in the **h3** tag

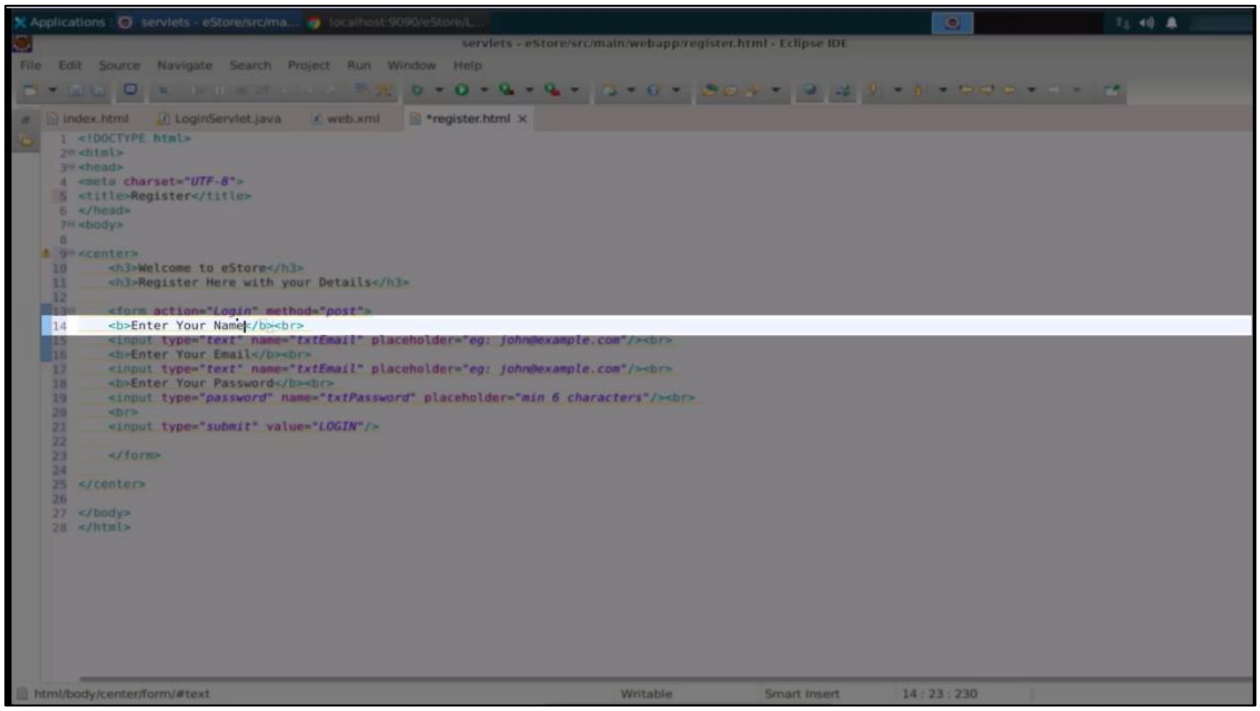


4.8 Copy the **input type** line and paste it before the **Email** line

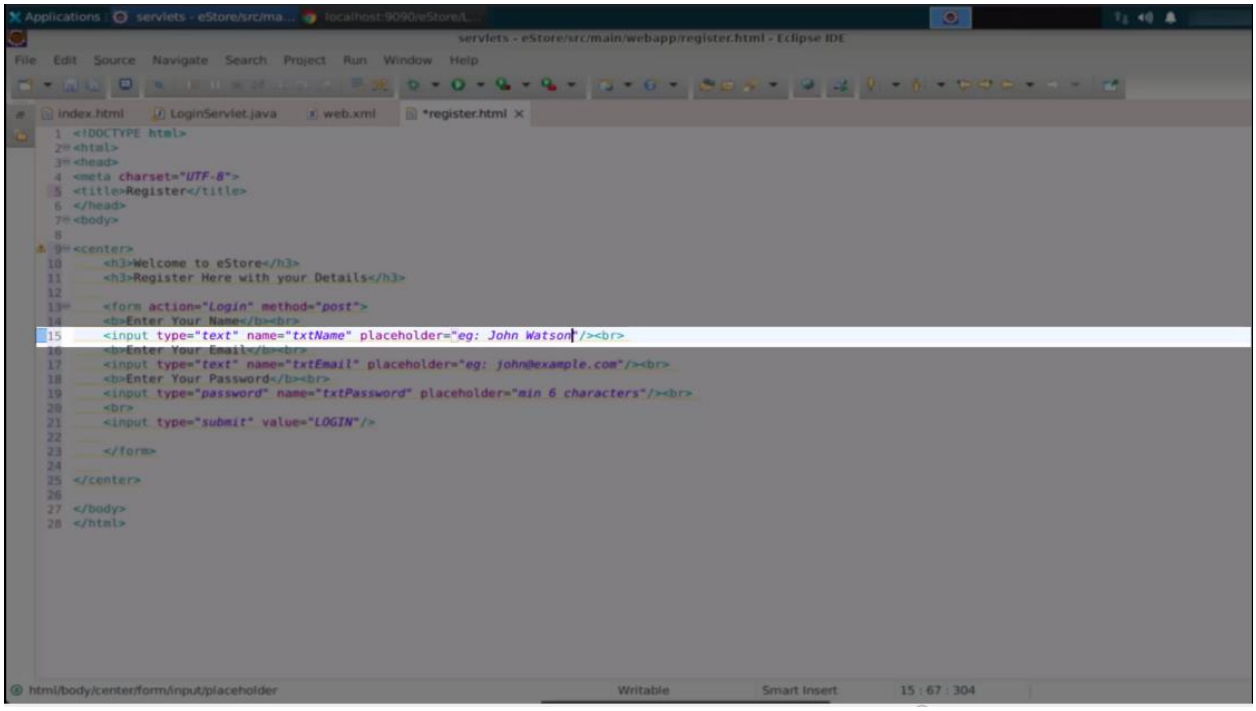




4.9 Replace email with a name



4.10 Make the changes in the **input type** field to capture the name as shown in the screenshot:



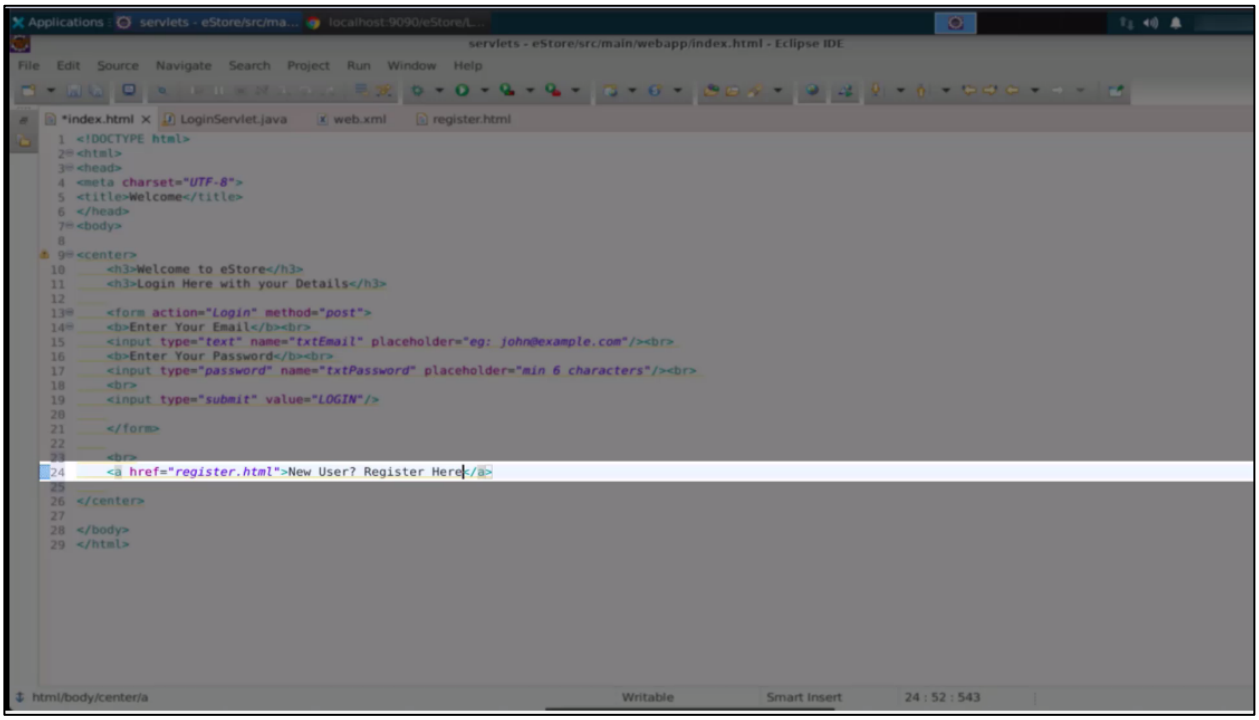
4.11 Replace login with **REGISTER** to call the register servlet

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Register</title>
</head>
<body>

<center>
    <h3>Welcome to eStore</h3>
    <h3>Register Here with your Details</h3>

    <form action="Login" method="post">
    <b>Enter Your Name</b><br>
    <input type="text" name="txtName" placeholder="eg: John Watson"/><br>
    <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>
    <input type="submit" value="REGISTER"/>

    </form>

</center>

</body>
</html>
```

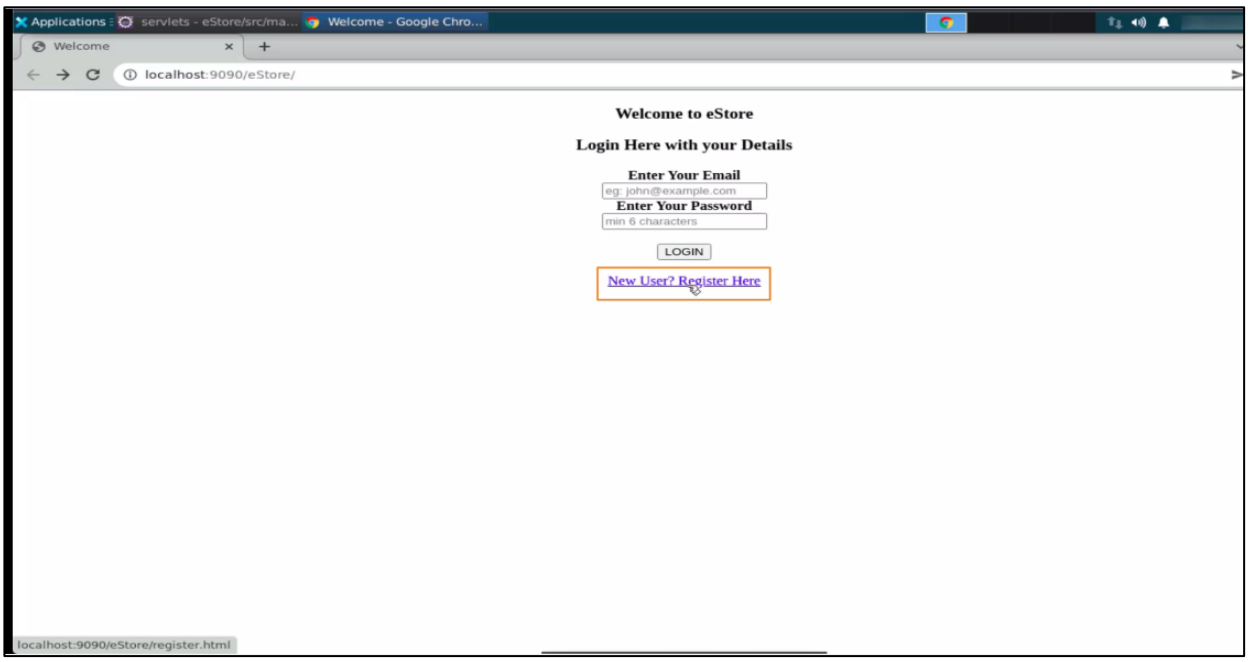
4.12 Add a **
** tag in the **index.html** file

```
<!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="post">
    <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>
    <input type="submit" value="LOGIN"/>

    </form>

    <br>

</center>

</body>
</html>
```

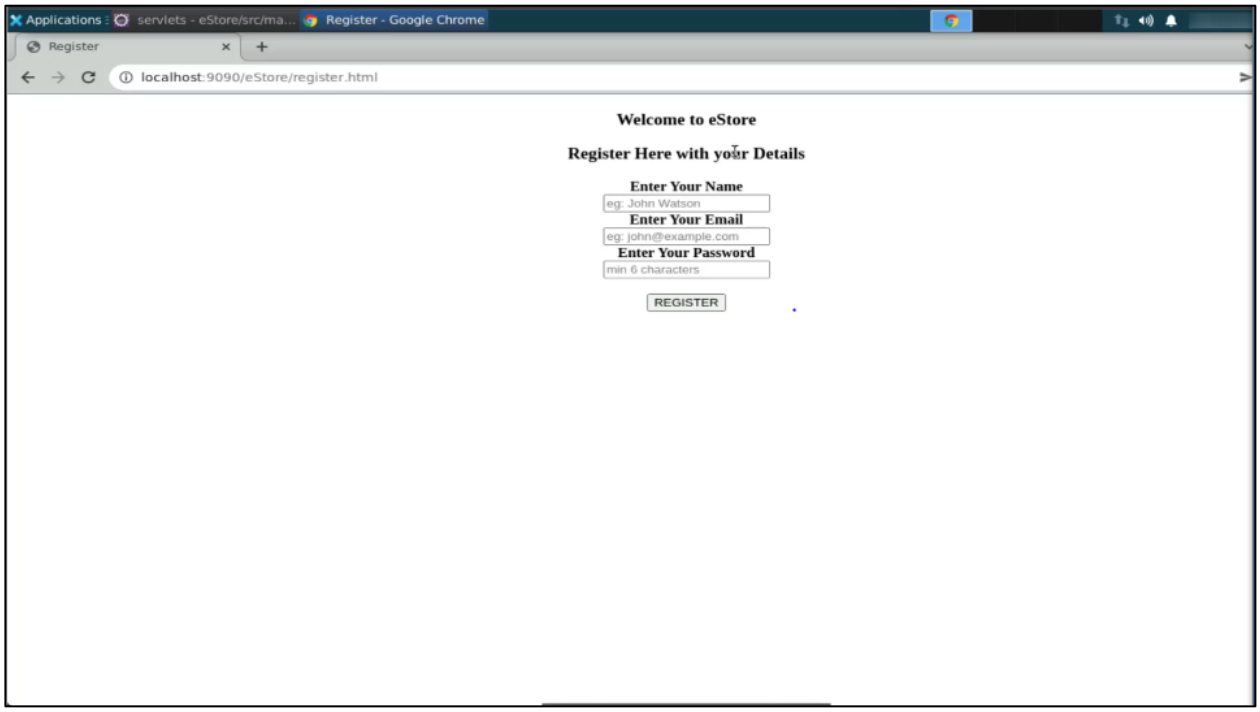
4.13 Enter a **href** tag and link it to the register page



4.14 Go to the web page and refresh it, you will be able to see the link for **New User? Register Here** link. Click on this link to create a new user or register

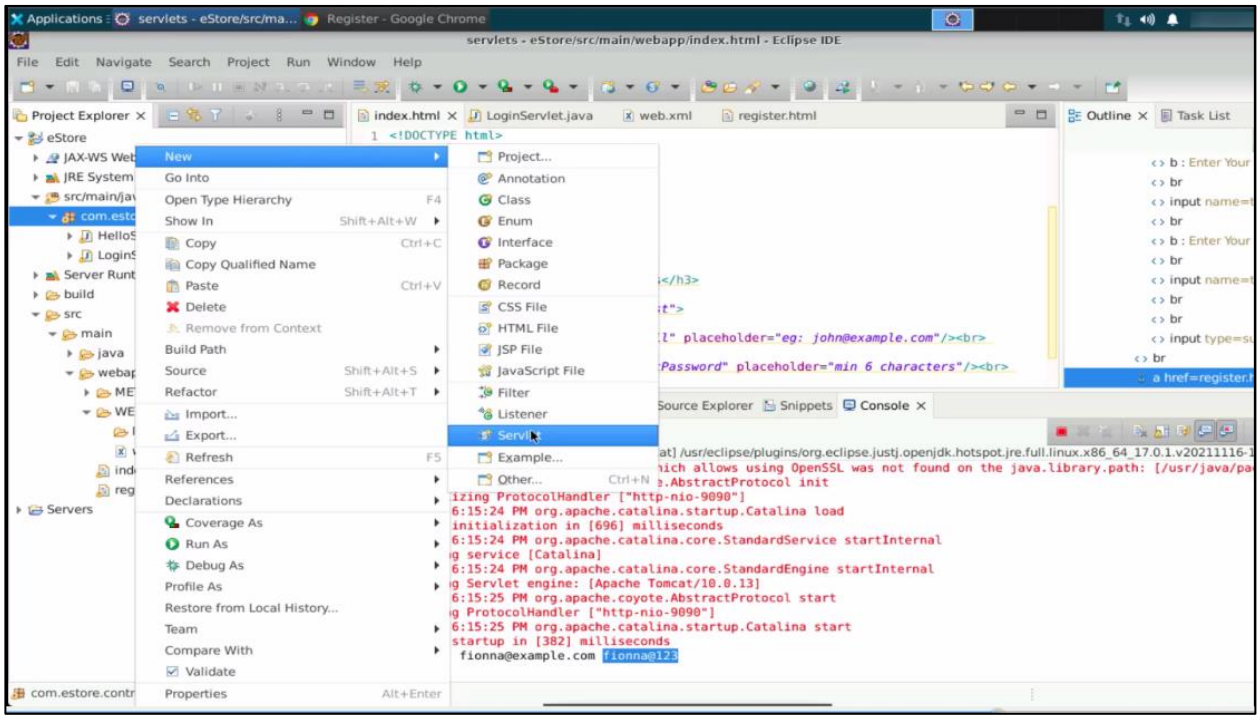


This will navigate the user to the register page.

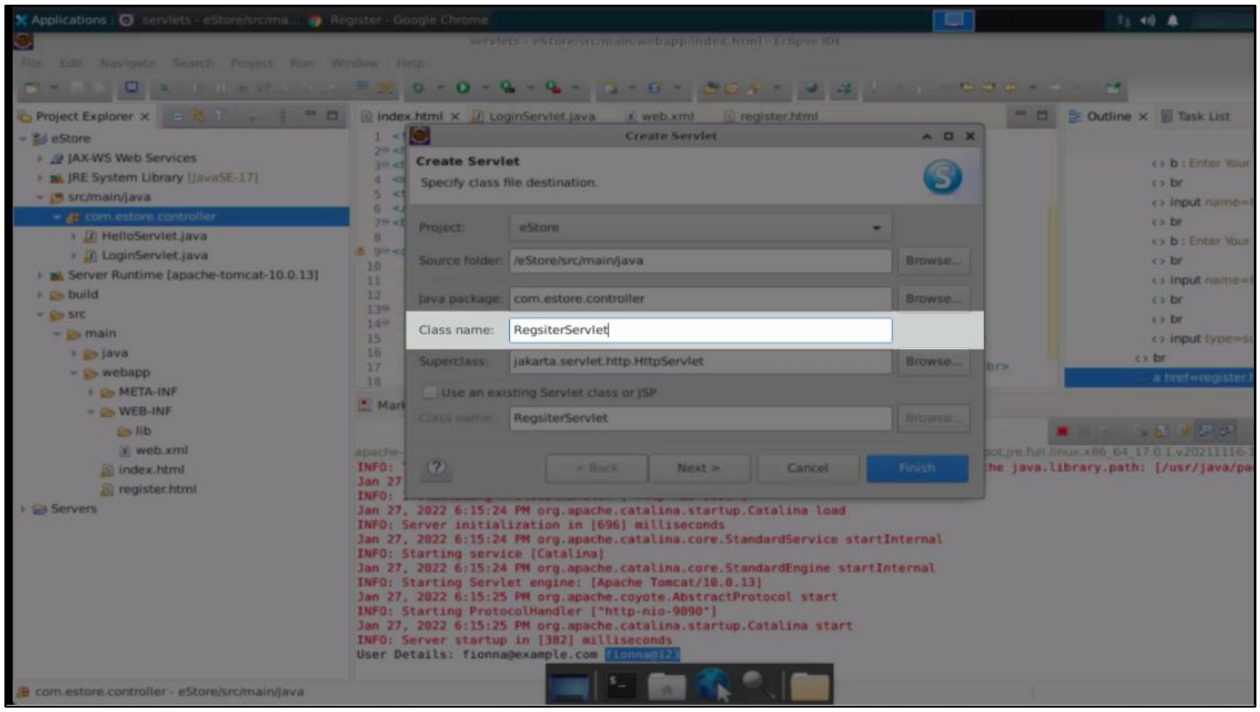


Step 5: Create a new Servlet for the register form

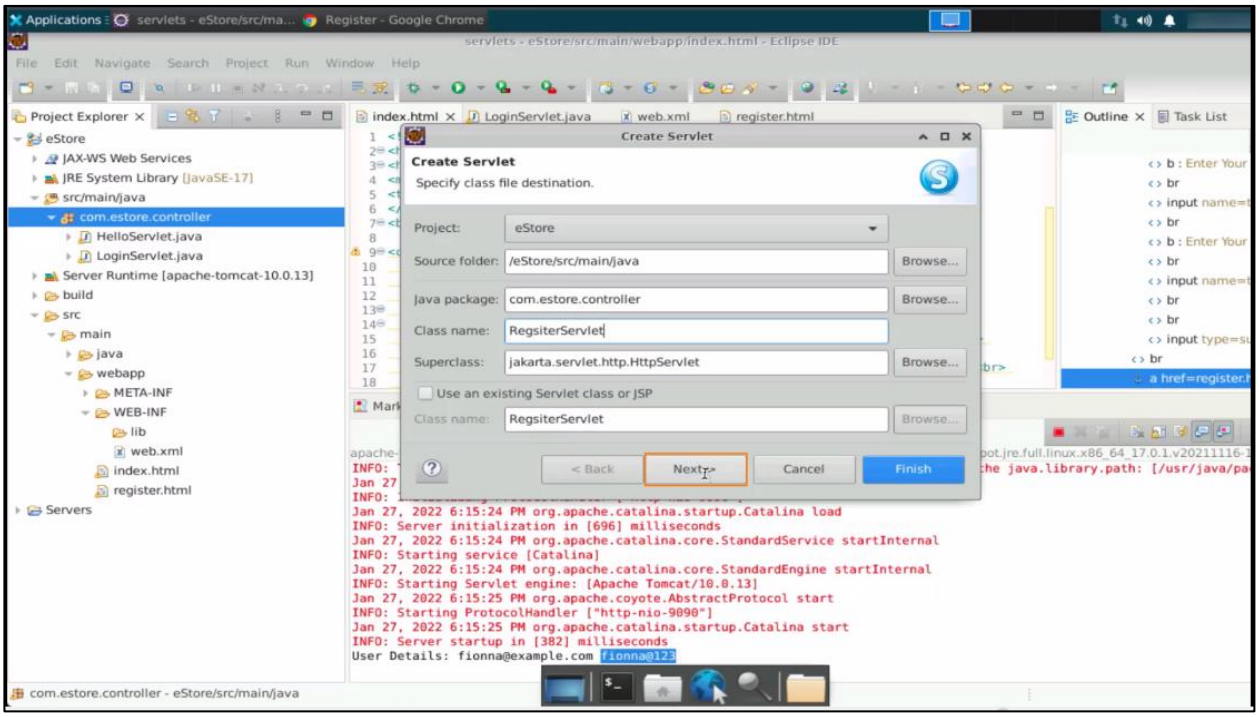
5.1 Create a new Servlet for the registration page. Go to **Eclipse,** right-click on the project, select **New**, and click on **Servlet.**



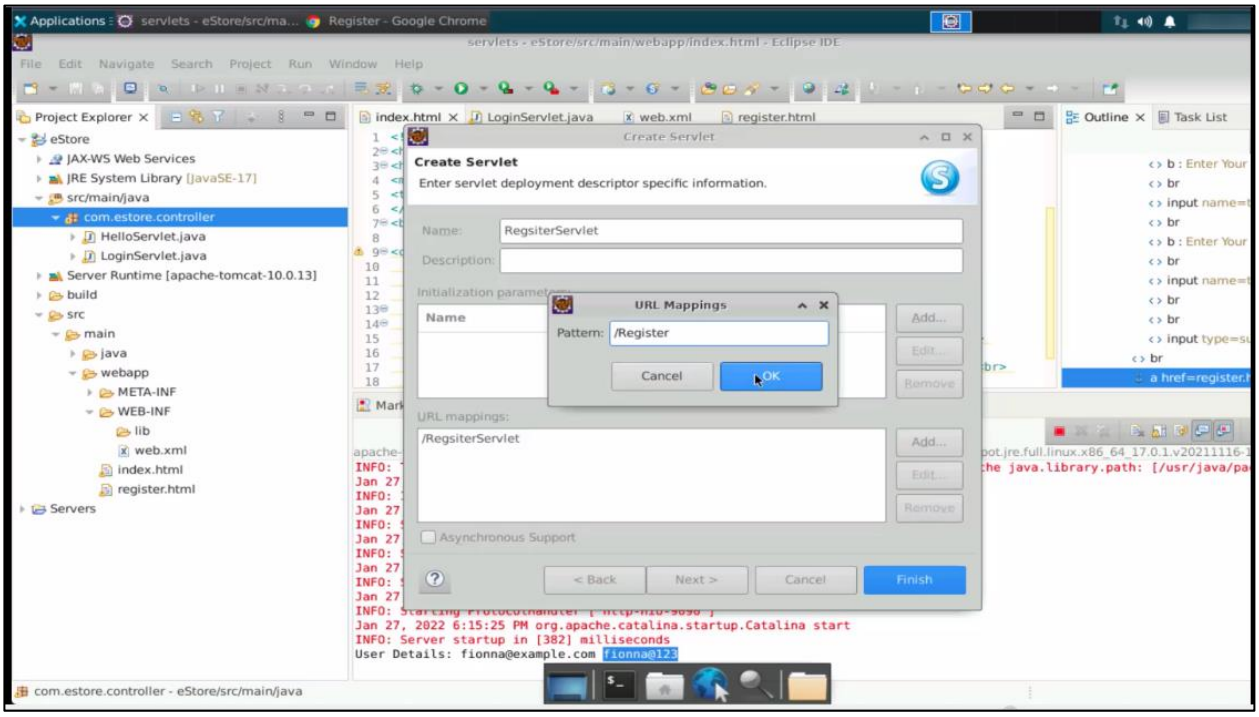
5.2 Name the Servlet as **RegisterServlet**



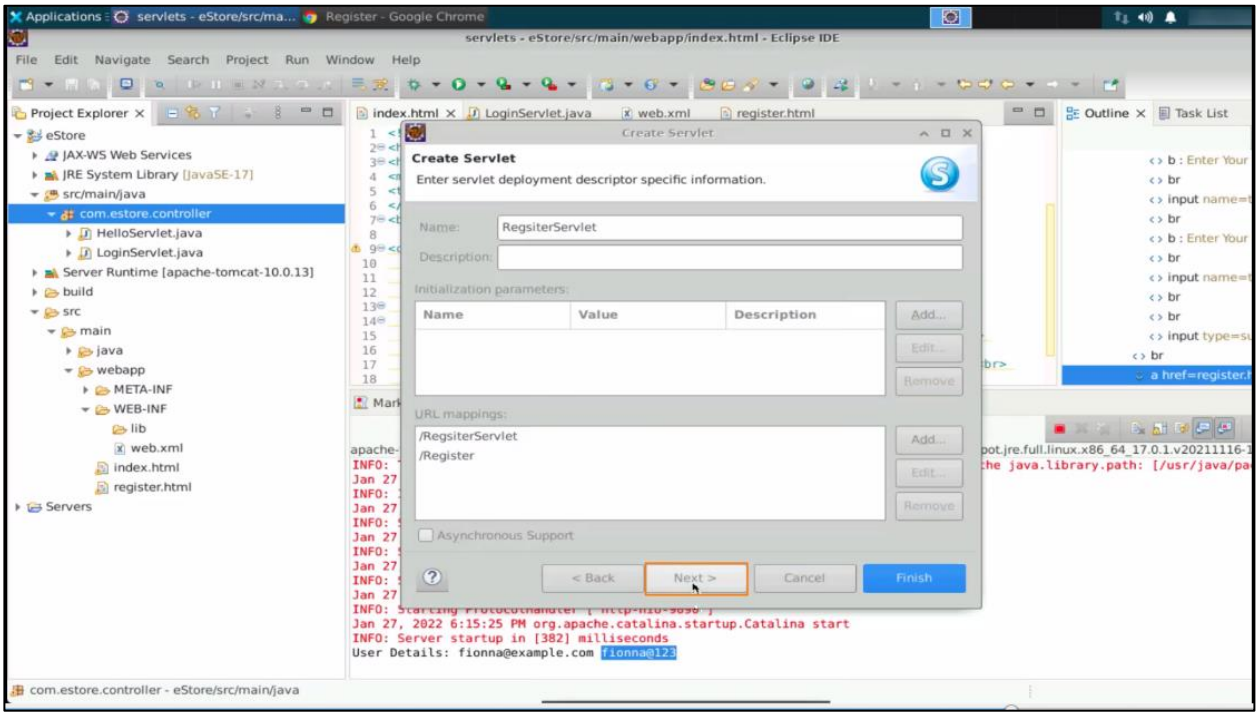
5.3 Select **Next**



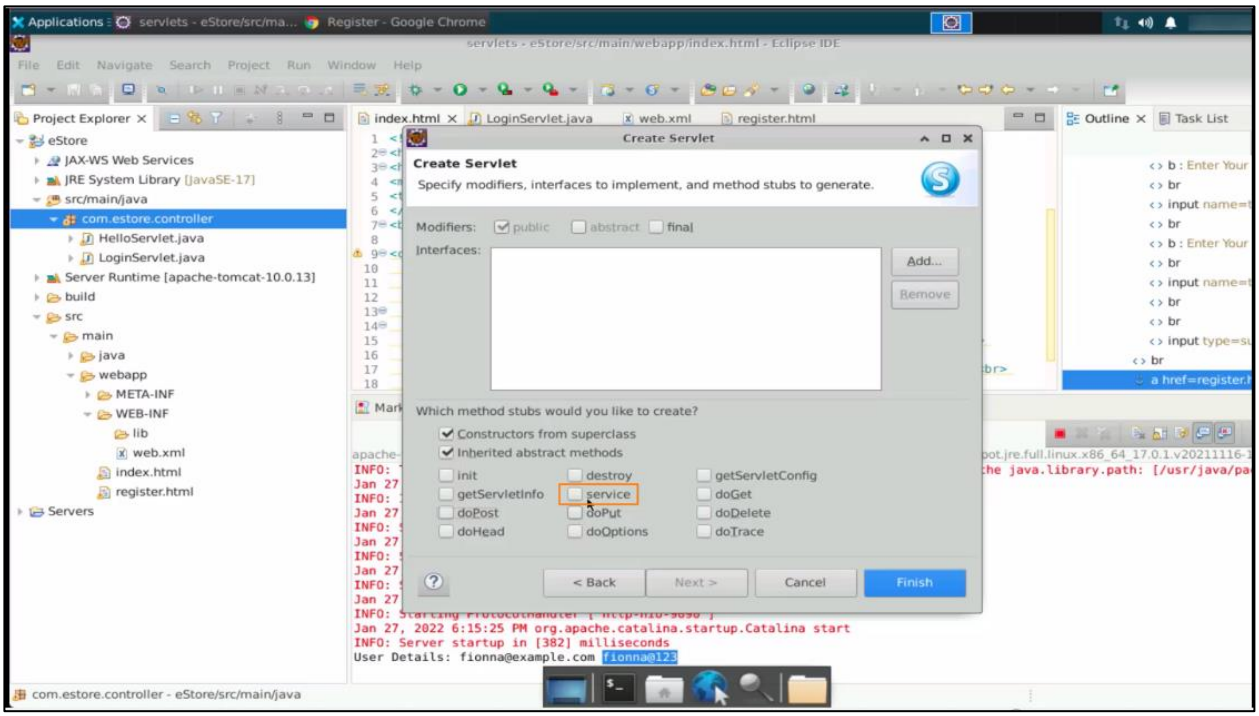
5.4 Enter **Register** in the **Pattern** field and click on **OK**



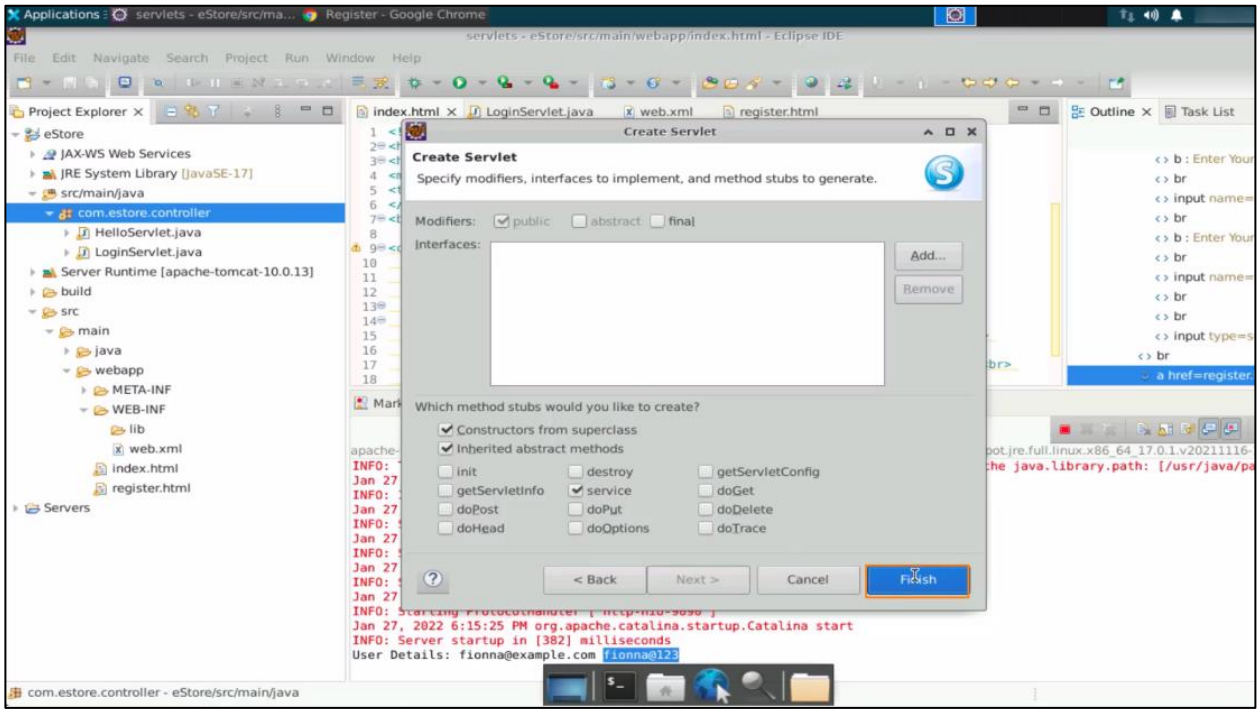
5.5 Click on **Next**



5.6 Select the **service** method

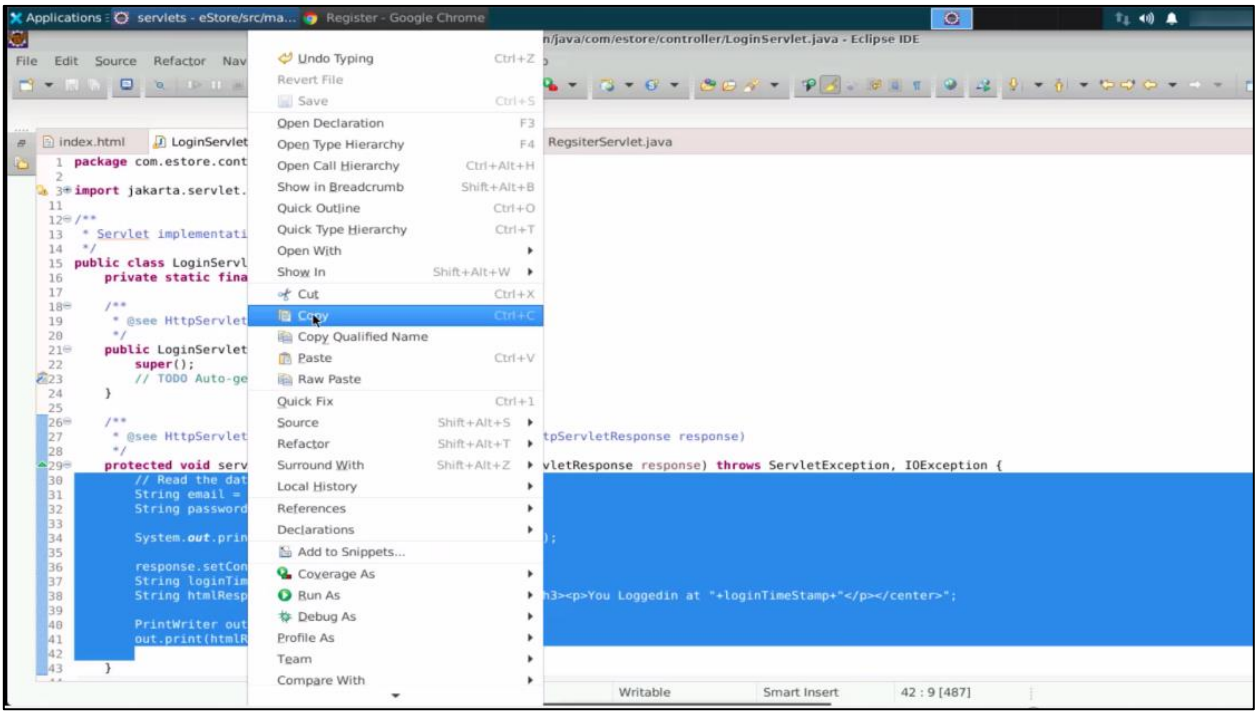


5.7 Click on **Finish**

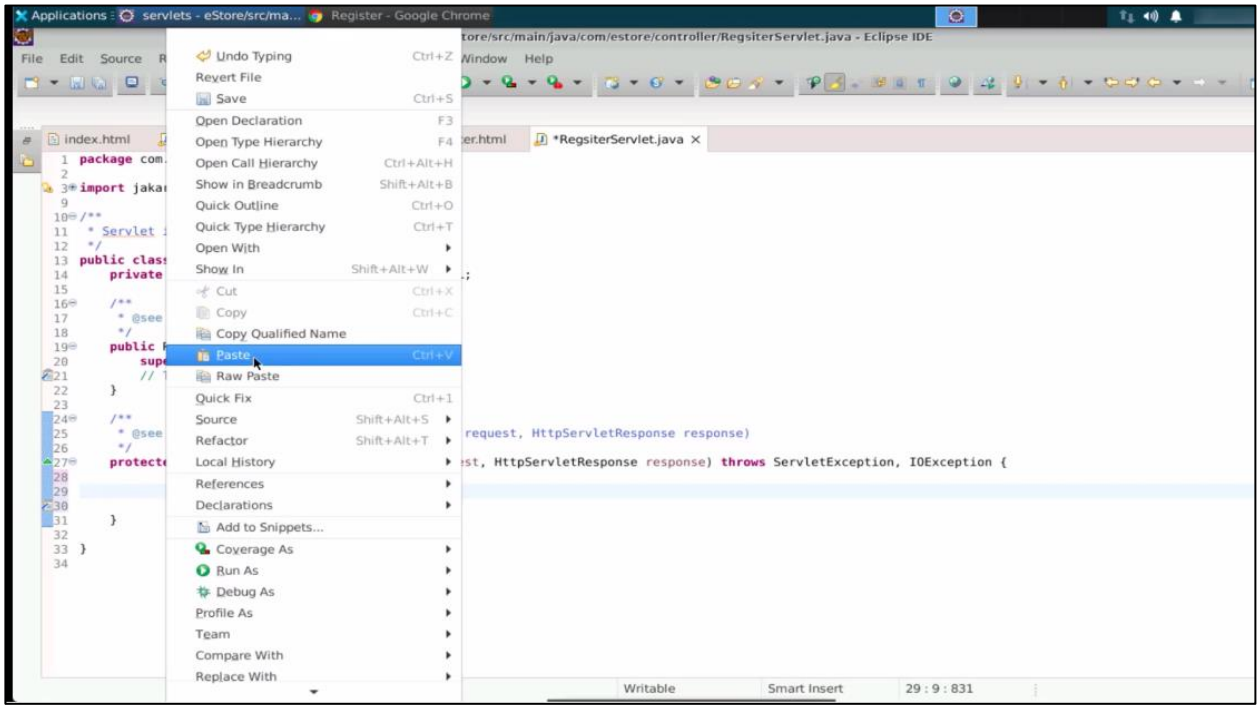


Step 6: Create a method in the register Servlet for the HTTP Servlet request

6.1 Copy the code from the **LoginServlet.java** file



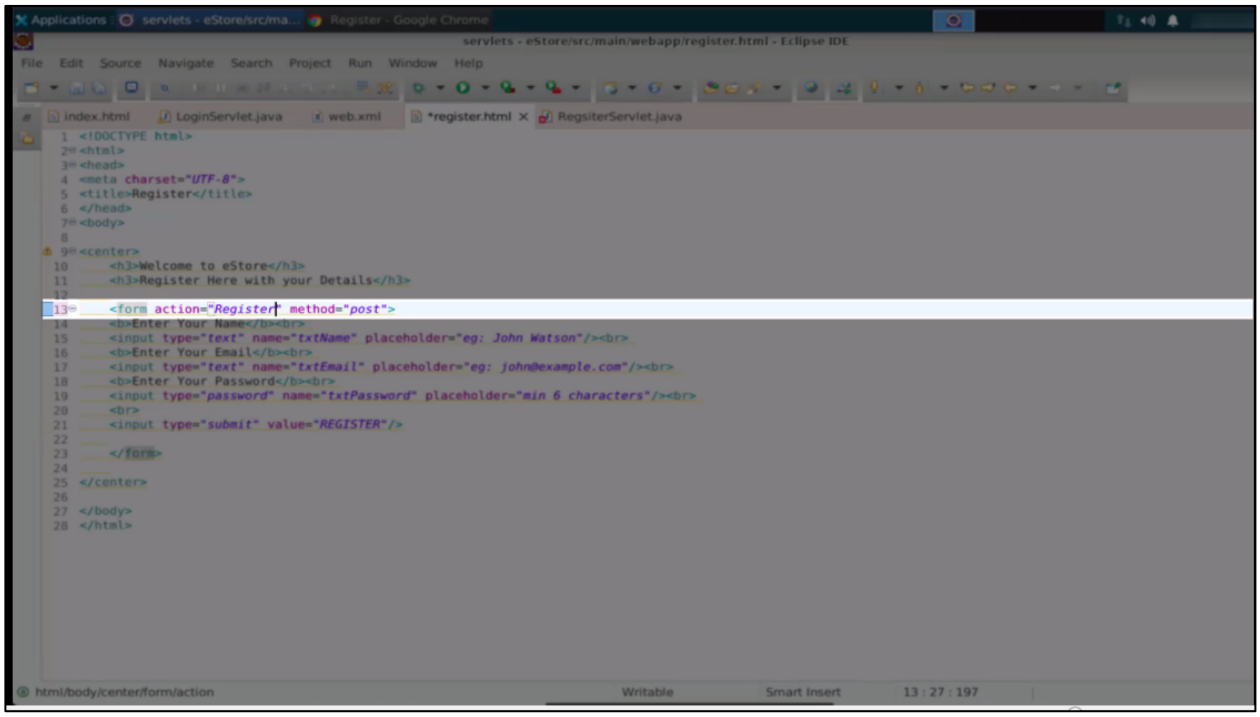
6.2 Paste the code in the **RegisterServlet.java** file



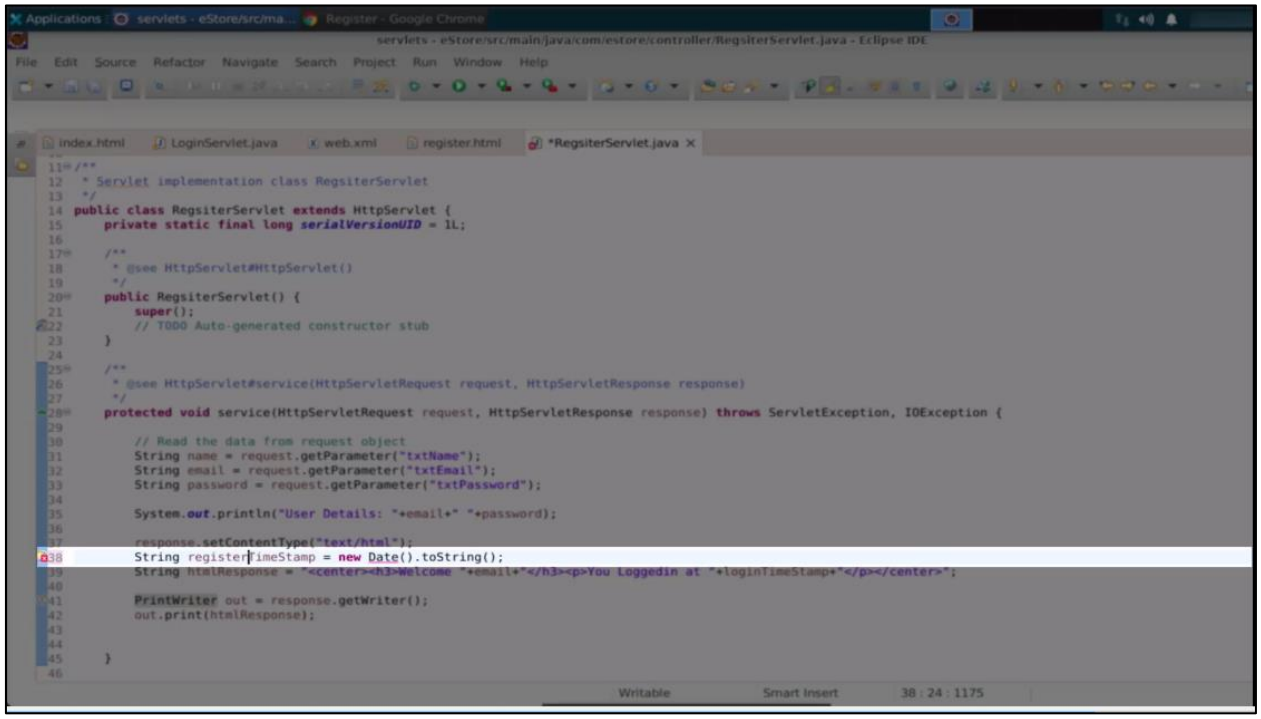
6.3 Add string **name** for reading the data of name field

```
/**
 * Servlet implementation class RegsiterServlet
 */
public class RegsiterServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public RegsiterServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        // Read the data from request object
        String name = request.getParameter("txtName");
        String email = request.getParameter("txtEmail");
        String password = request.getParameter("txtPassword");

        System.out.println("User Details: "+email+" "+password);

        response.setContentType("text/html");
        String loginTimeStamp = new Date().toString();
        String htmlResponse = "<center><h3>Welcome "+email+"</h3><p>You Loggedin at "+loginTimeStamp+"</p></center>";

        PrintWriter out = response.getWriter();
        out.print(htmlResponse);


    }

}
```

6.4 On the **Register.html** page, change the action to **Register** to send data to the registered Servlet



6.5 Replace **LoginTimeStamp** with **registerTimeStamp** in the **RegisterServlet.java** file



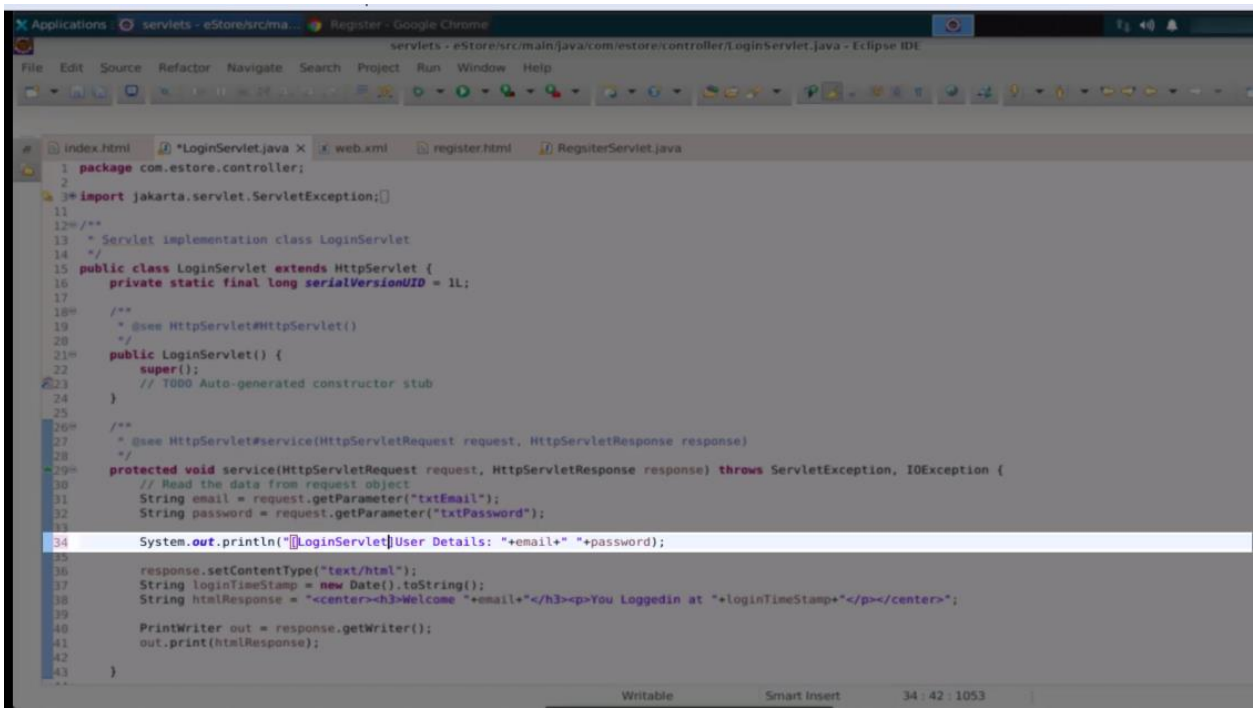
6.6 Make the changes in the response code as required. Change the message to display in the paragraph tag.

```
/**
 * Servlet implementation class RegsiterServlet
 */
public class RegsiterServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public RegsiterServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        // Read the data from request object
        String name = request.getParameter("txtName");
        String email = request.getParameter("txtEmail");
        String password = request.getParameter("txtPassword");

        System.out.println("User Details: "+email+" "+password);

        response.setContentType("text/html");
        String registerTimeStamp = new Date().toString();
        String htmlResponse = "<center><h3>Thank you "+name+"</h3><p>You are Registered with email "+email+" at "+registerTimeStamp+"</p></center>";

        PrintWriter out = response.getWriter();
        out.print(htmlResponse);


    }
```

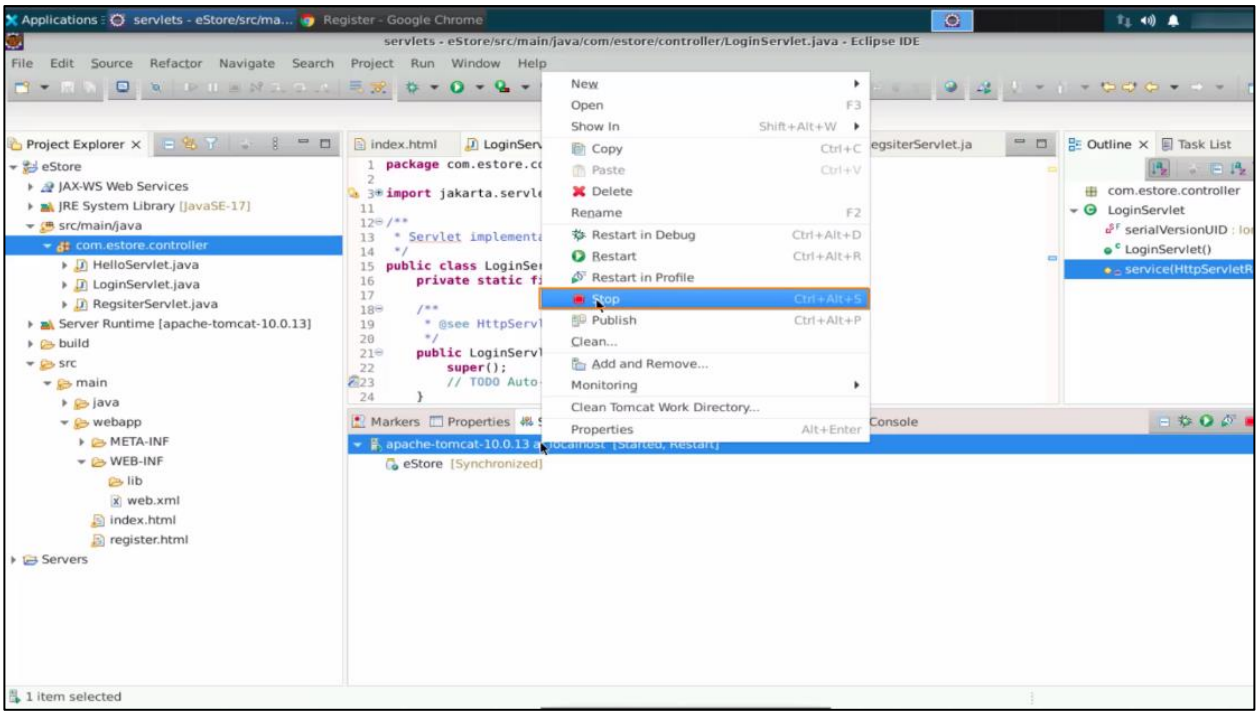
6.7 Change the console output to show **user details** with their **names**

```
/**
 * Servlet implementation class RegsiterServlet
 */
public class RegsiterServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public RegsiterServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        // Read the data from request object
        String name = request.getParameter("txtName");
        String email = request.getParameter("txtEmail");
        String password = request.getParameter("txtPassword");

        System.out.println("RegisterServlet User Details: "+name+" "+email+" "+password);

        response.setContentType("text/html");
        String registerTimeStamp = new Date().toString();
        String htmlResponse = "<center><h3>Thank you "+name+"</h3><p>You are Registered with email "+email+" at "+registerTimeStamp+"</p></center>";

        PrintWriter out = response.getWriter();
        out.print(htmlResponse);


    }
```

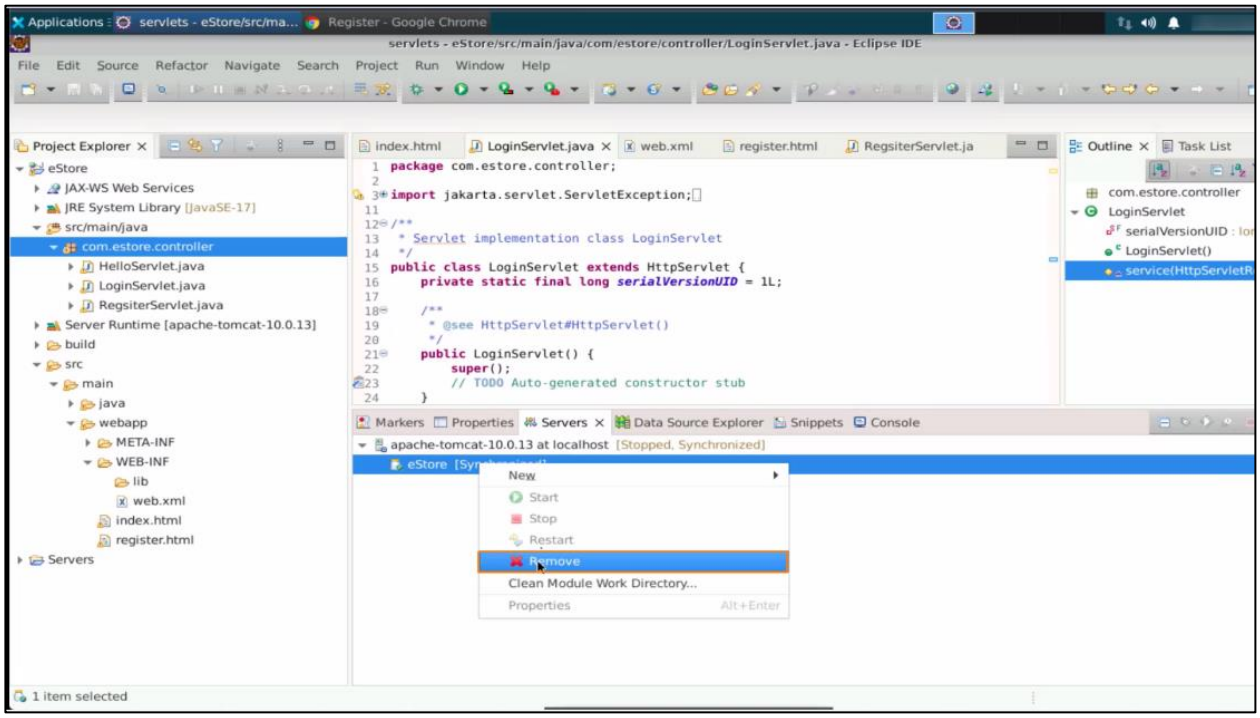
6.8 In the **LoginServlet**, type **LoginServlet** and save



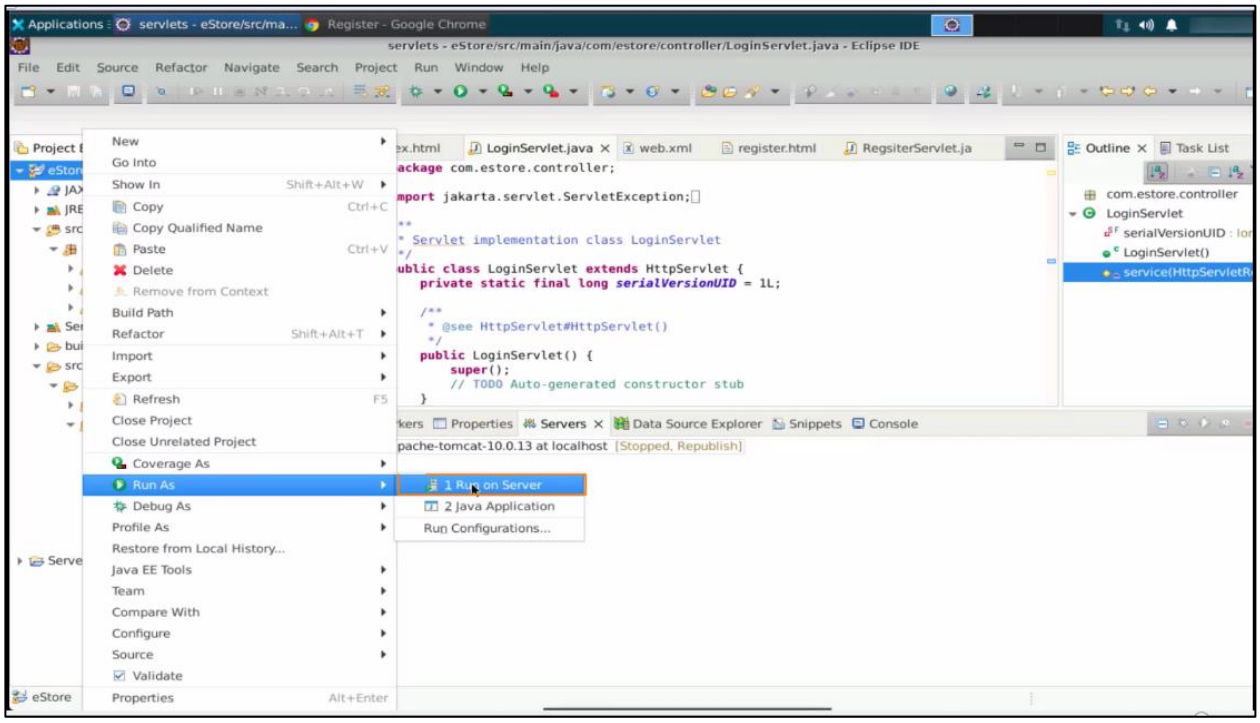
6.9 Stop the server



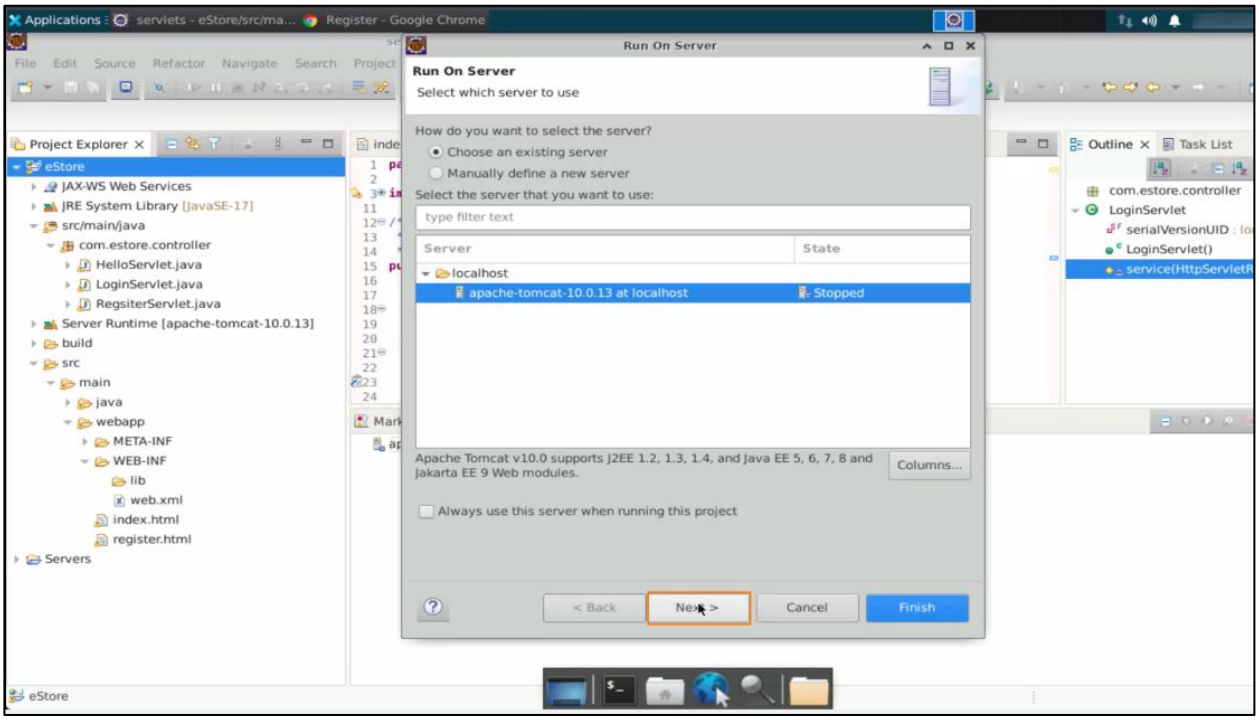
6.10 Remove the **eStore** file from the server by right-clicking on the file and selecting **Remove**



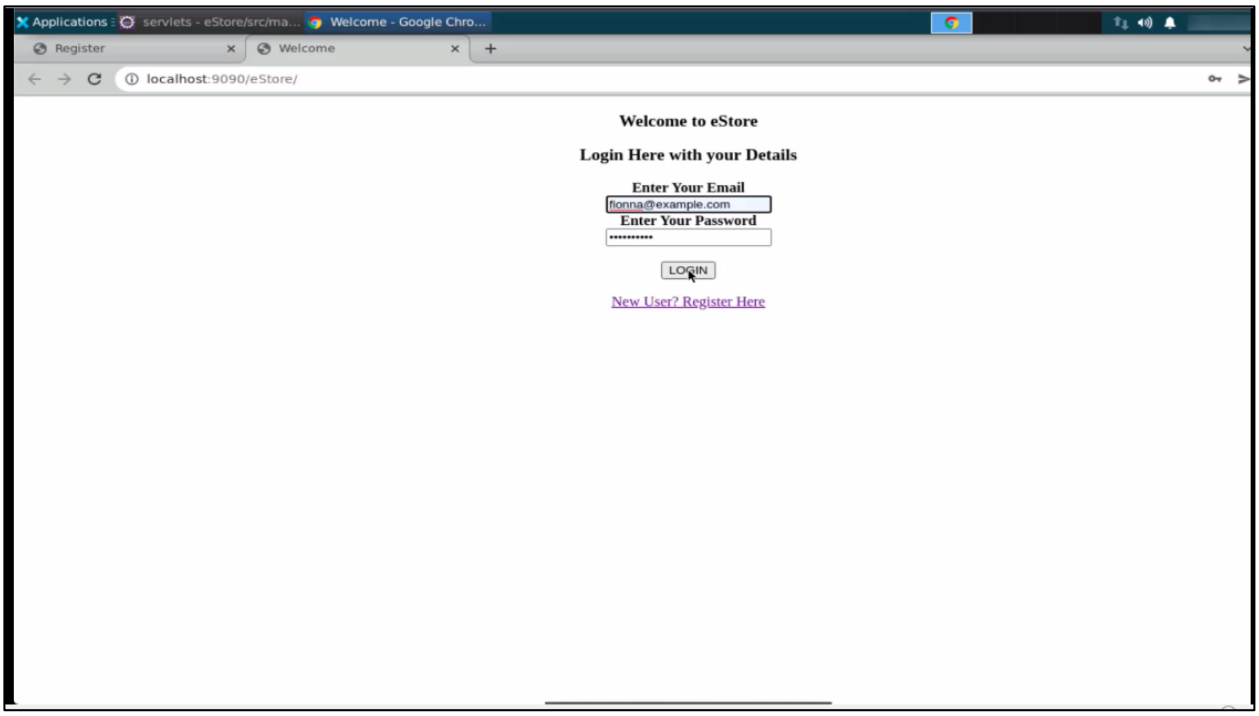
6.11 Then, run it as the server by right-clicking on the project, selecting **Run As**, and then **Run on Server**



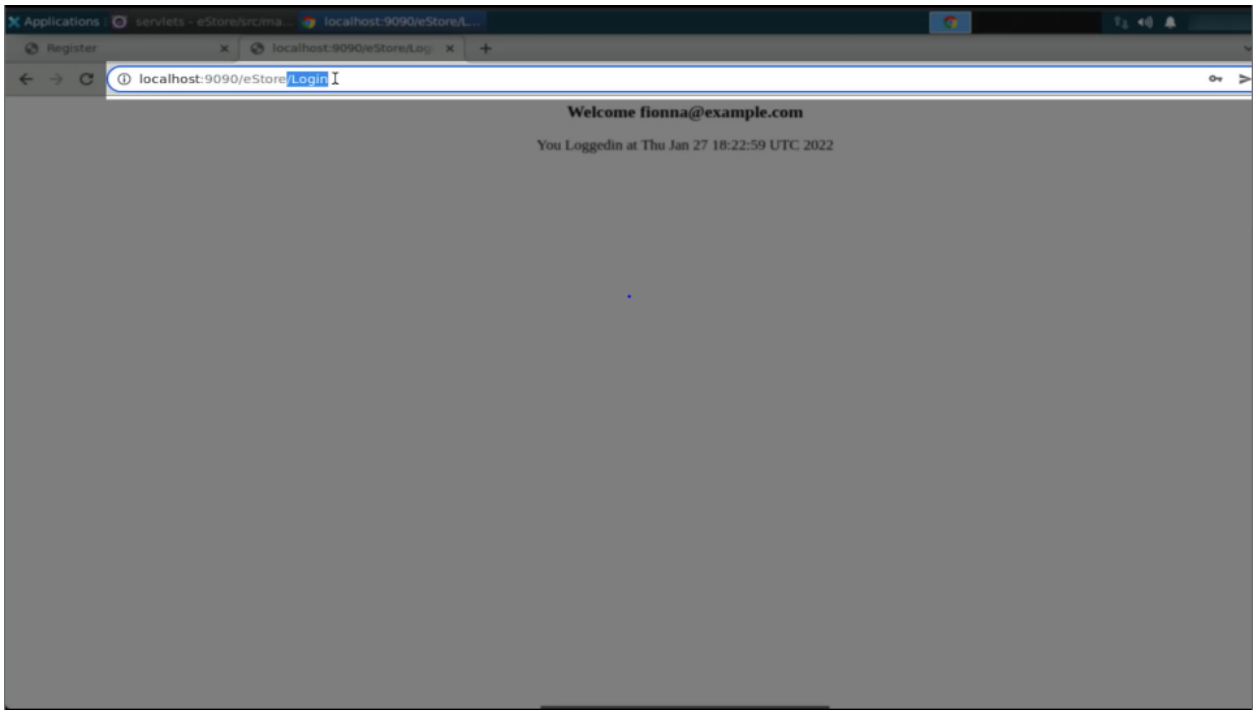
6.12 Choose the server as **apache-tomcat-10.0.13**, click on **Next,** and then click on **Finish**



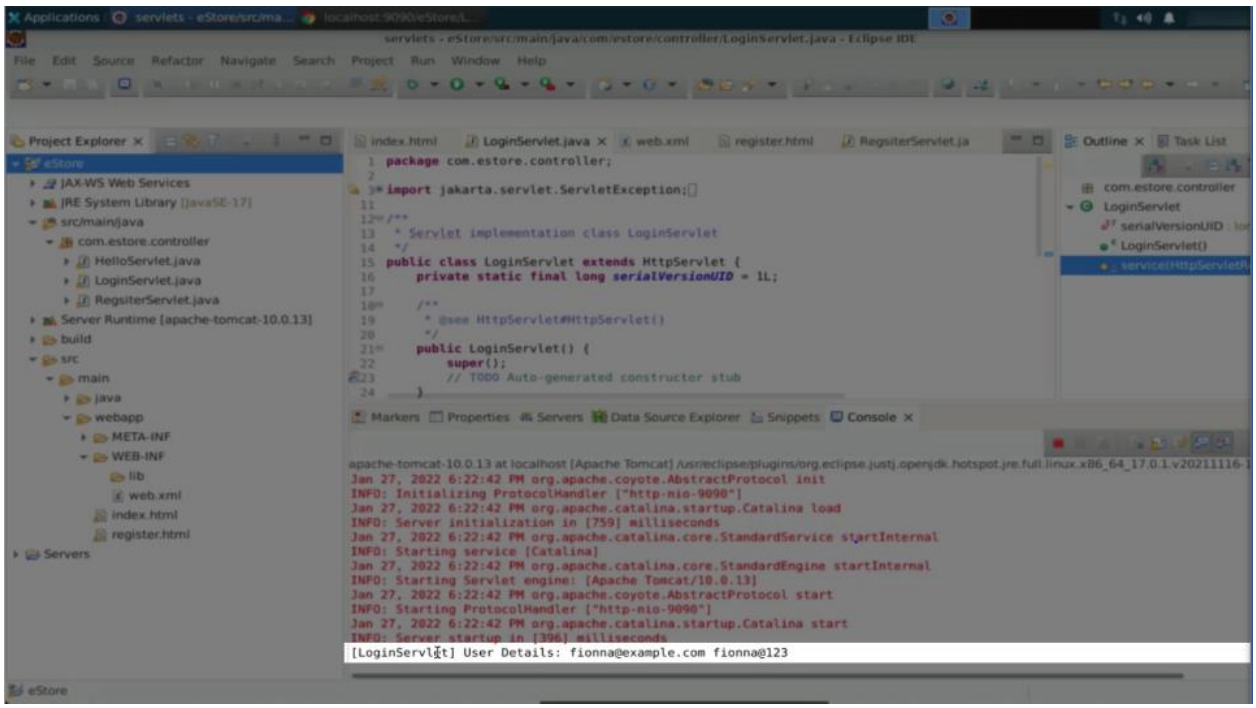
6.13 Refresh the page, enter data, and click on **LOGIN**



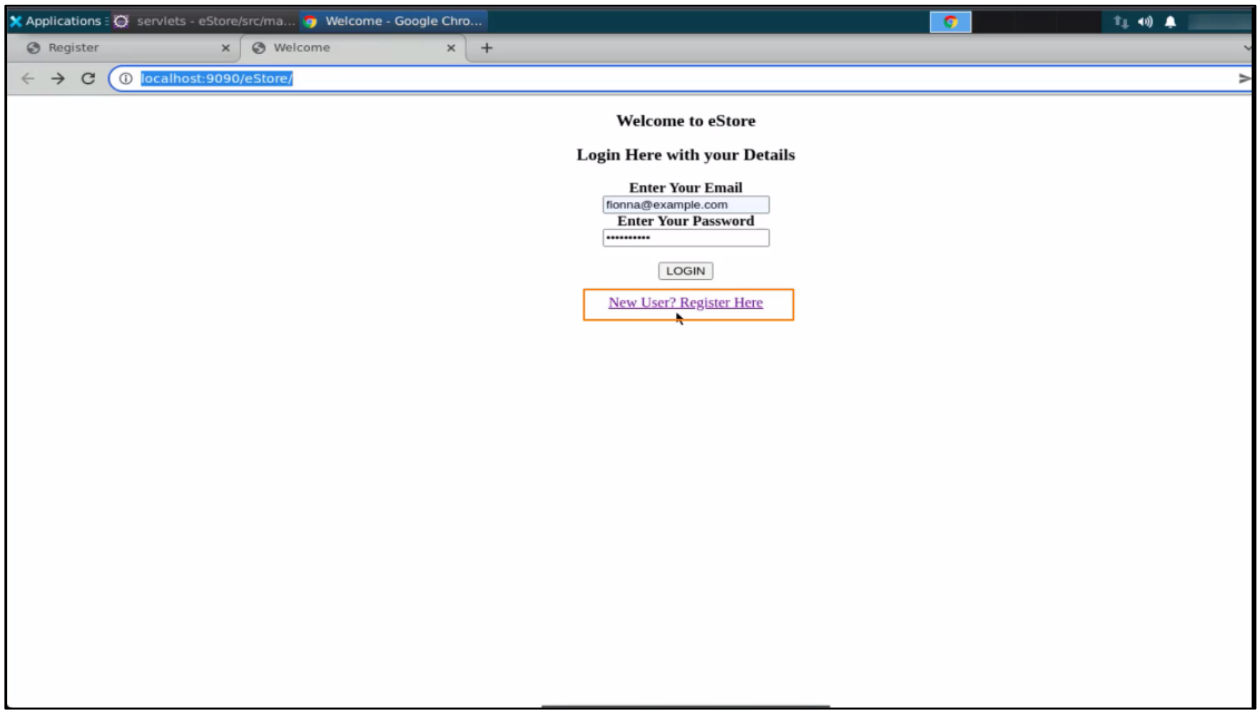
You can see that you are redirected to the login page successfully.



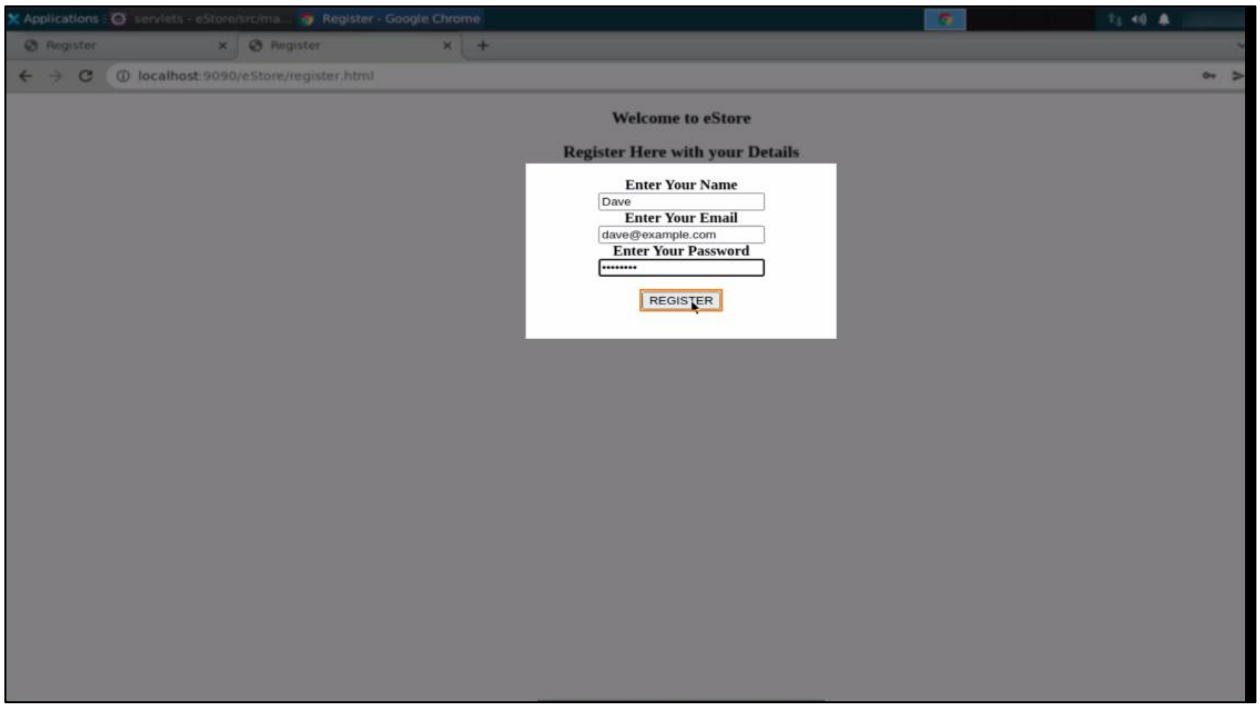
Here in the console, you can see the user details:



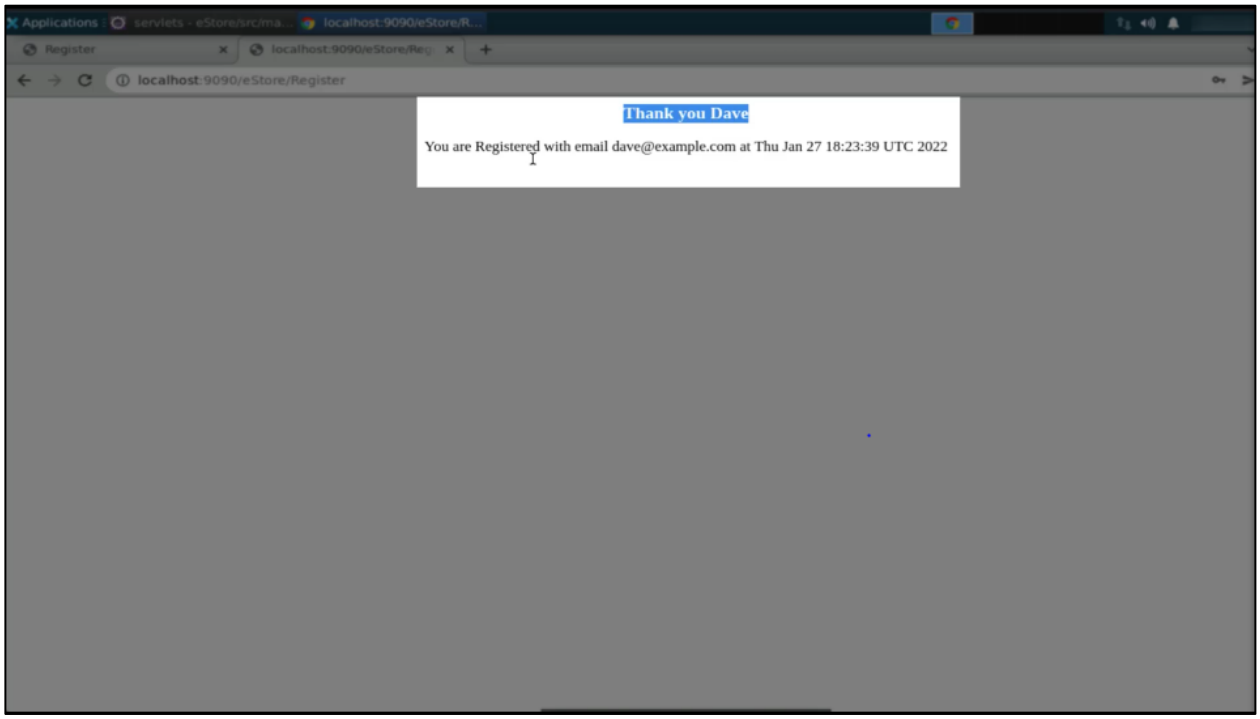
6.14 Click on **New User? Register Here** on the first page of the project



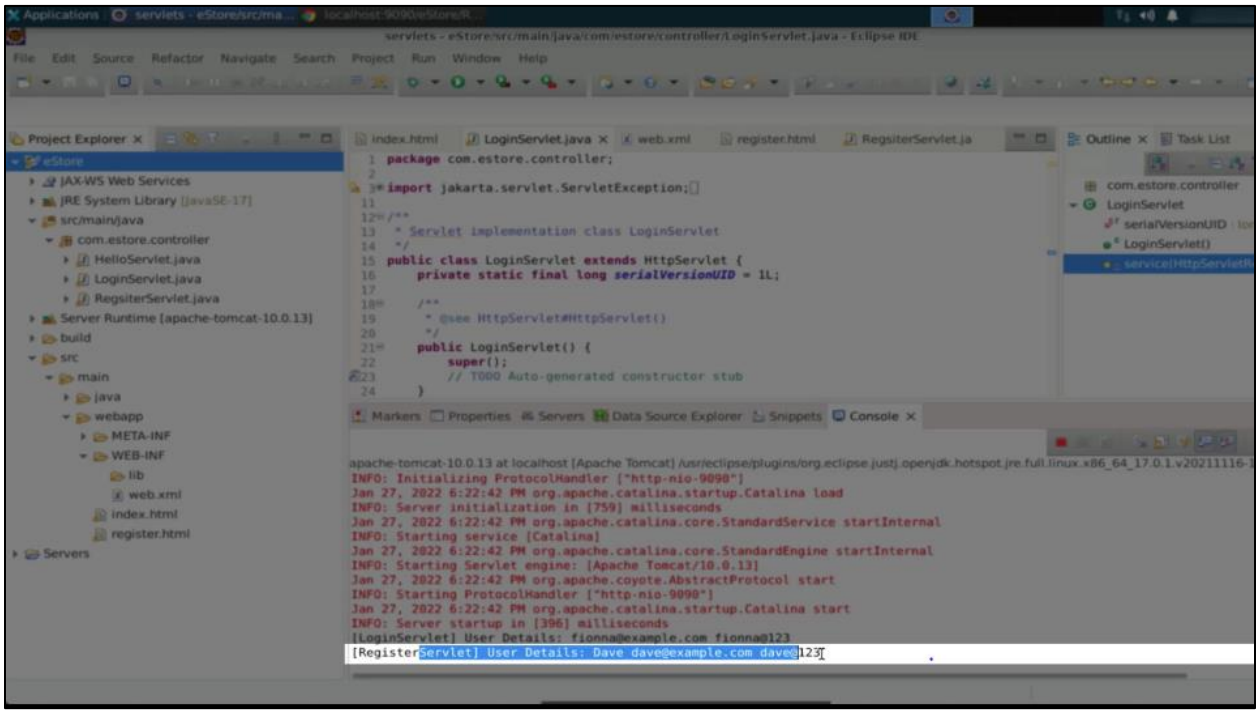
6.15 Enter the new user data and click on **REGISTER**



You can see that the new user data is registered successfully, and **Thank you, Dave** is displayed.



Go to the Eclipse console, you can see that the registered Servlet has shared the data on the console as well.



Following these steps, you will create an HTML page that contains a login and registration form for sending a request to a Servlet.