

## Lesson 02 Demo 02

### Create a FrontController Design Pattern

**Objective:** To create a FrontController design pattern and send a request to the FrontController servlet

**Tools Required:** Eclipse IDE

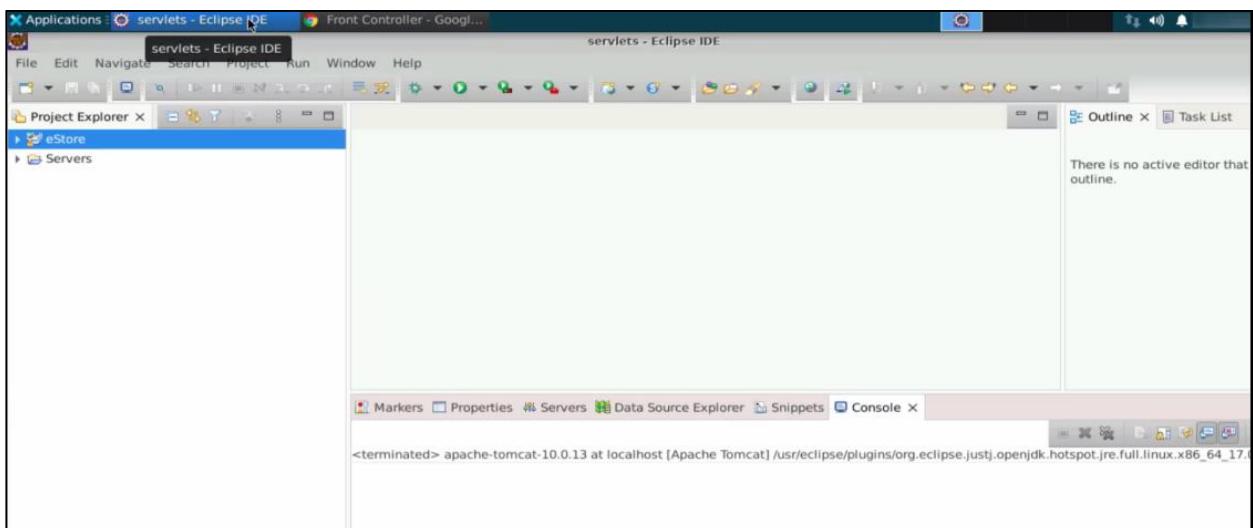
**Prerequisites:** None

#### Steps to be followed:

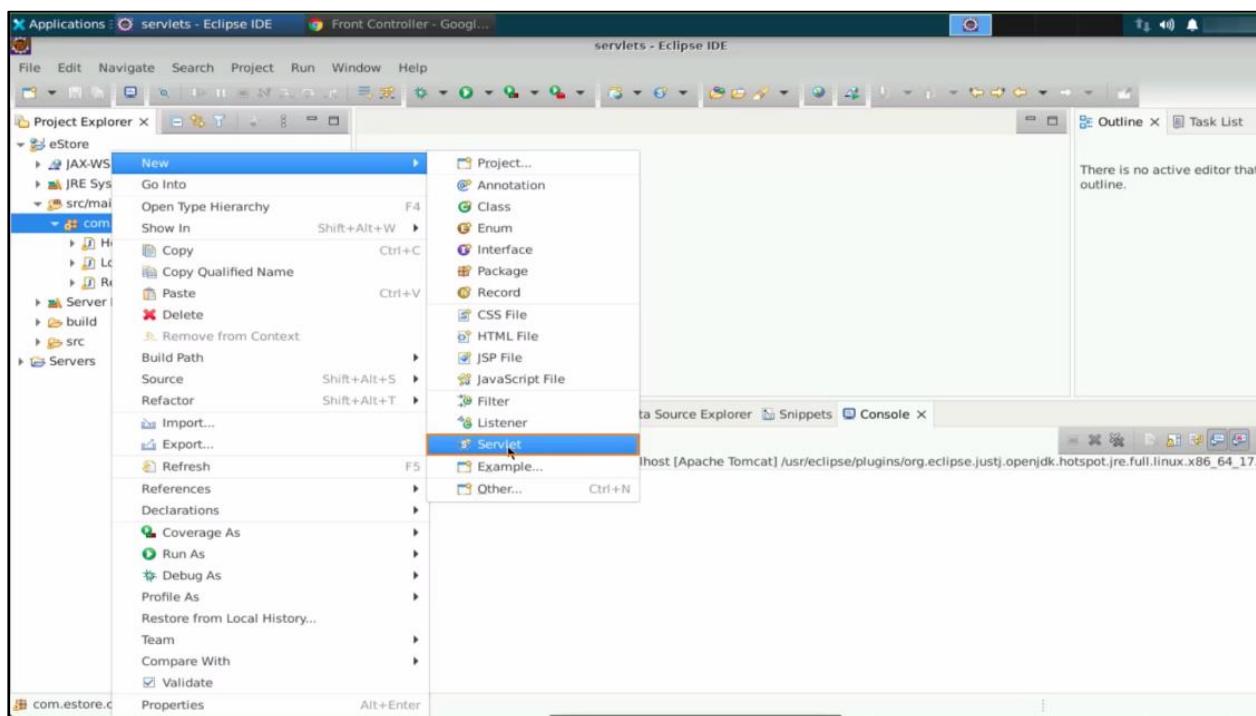
1. Create a FrontController servlet
2. Send a request to the FrontController servlet
3. Create a method in the FrontController servlet
4. Update responses in the FrontController servlet

#### Step 1: Create a FrontController servlet

##### 1.1 Open Eclipse IDE

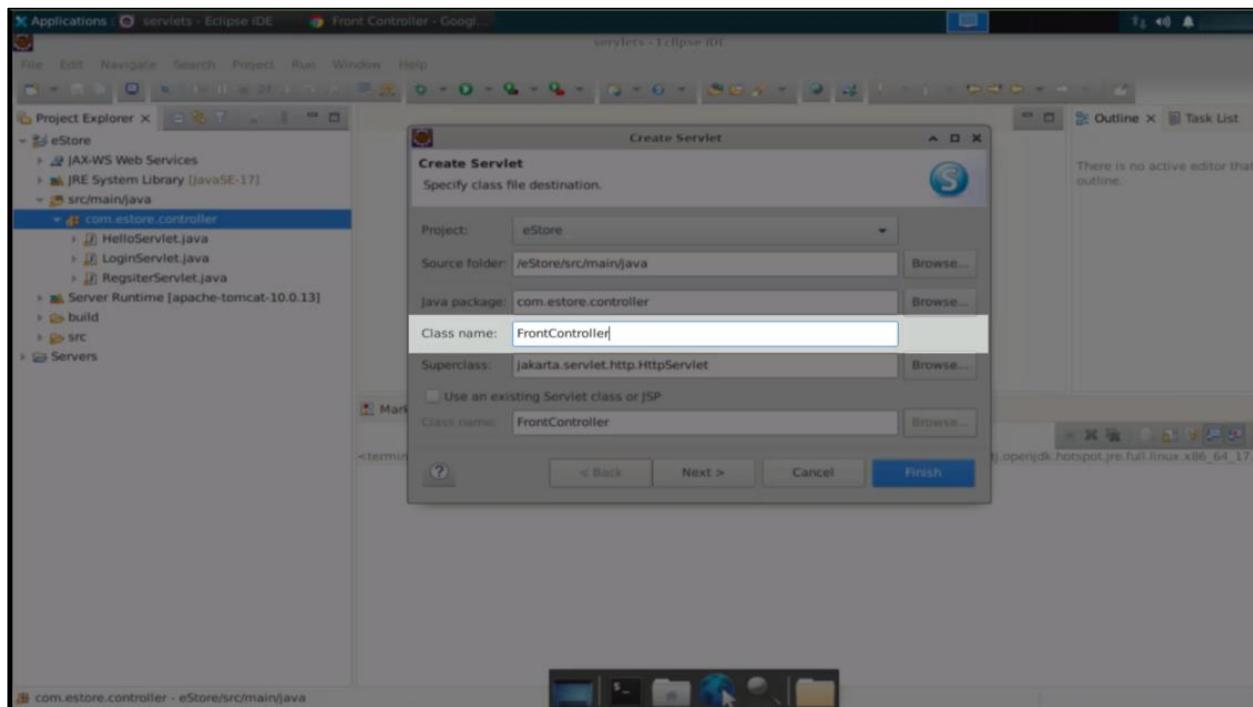


## 1.2 Create a new Servlet under the eStore project by selecting New and Servlet

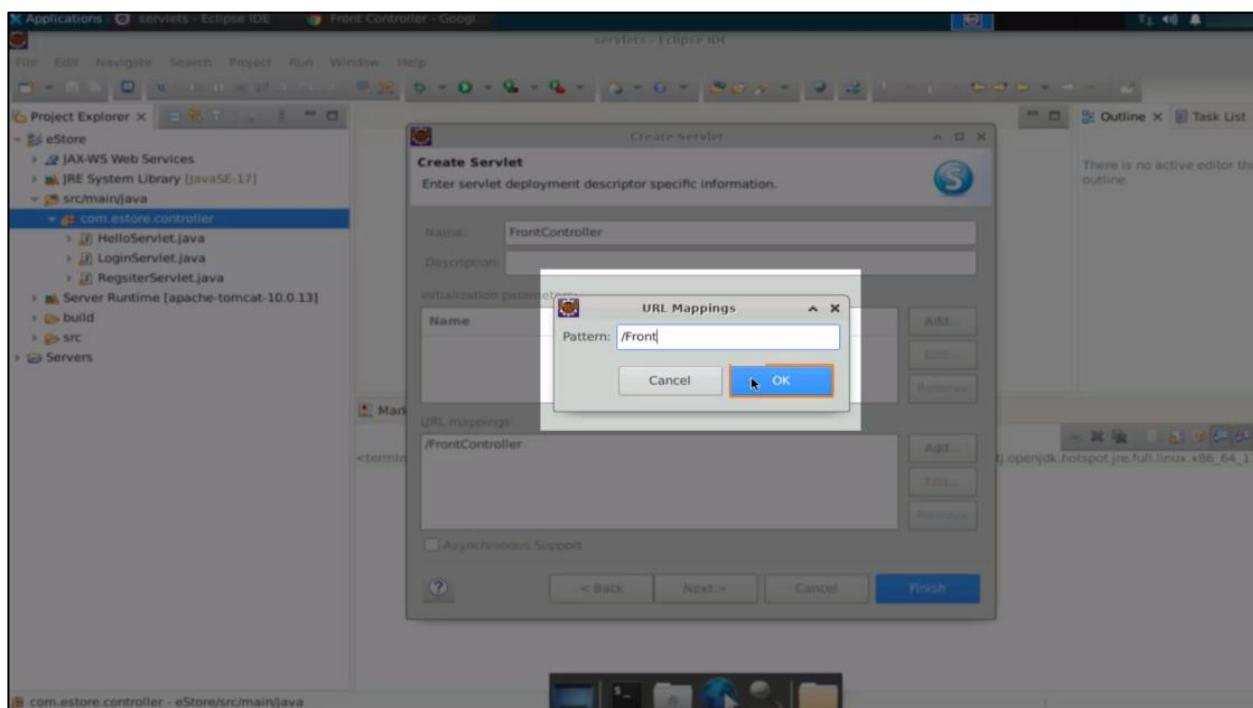


**Note:** Please refer to the previous demo on how to create the **eStore** project

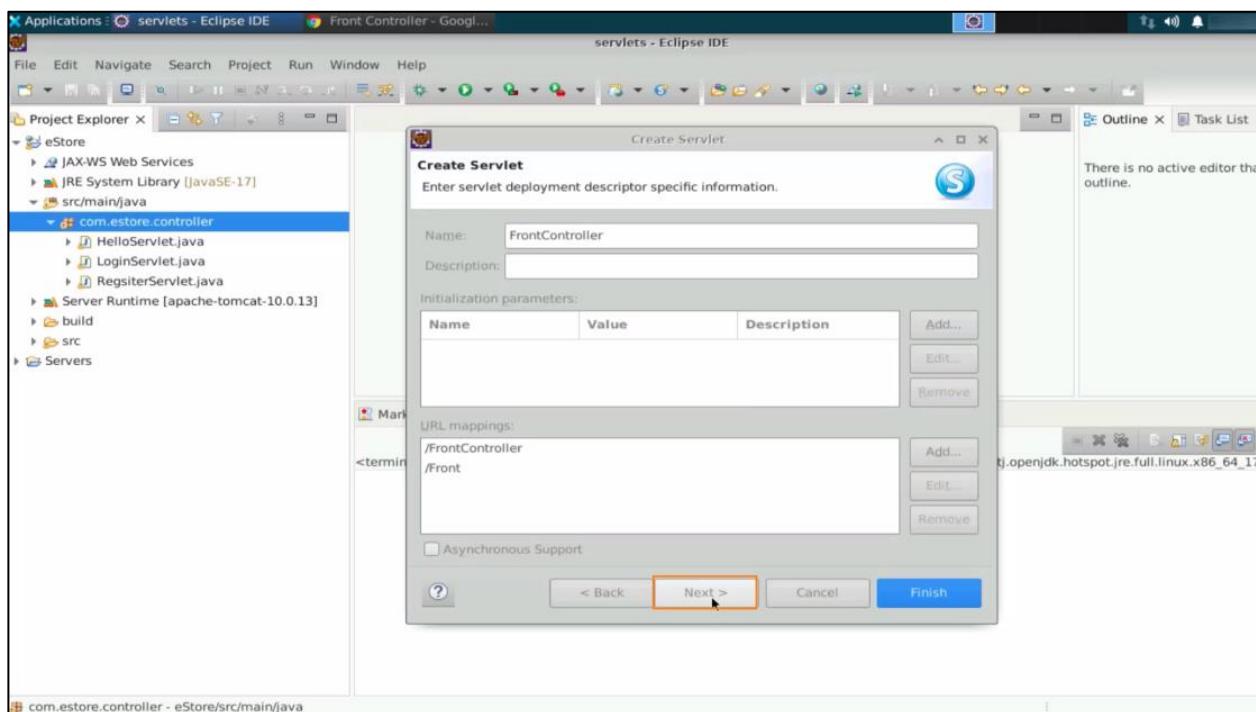
### 1.3 Enter the class name as **FrontController**



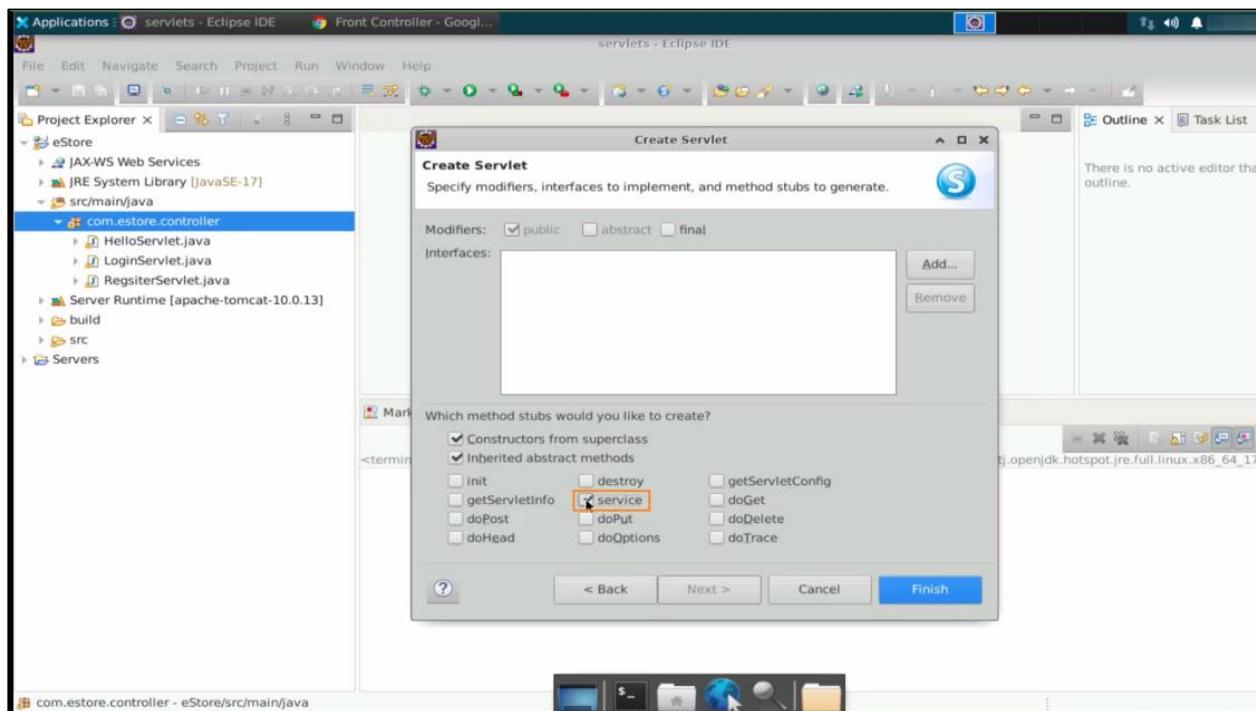
### 1.4 Enter the URL Mappings as **/Front** and click **OK**



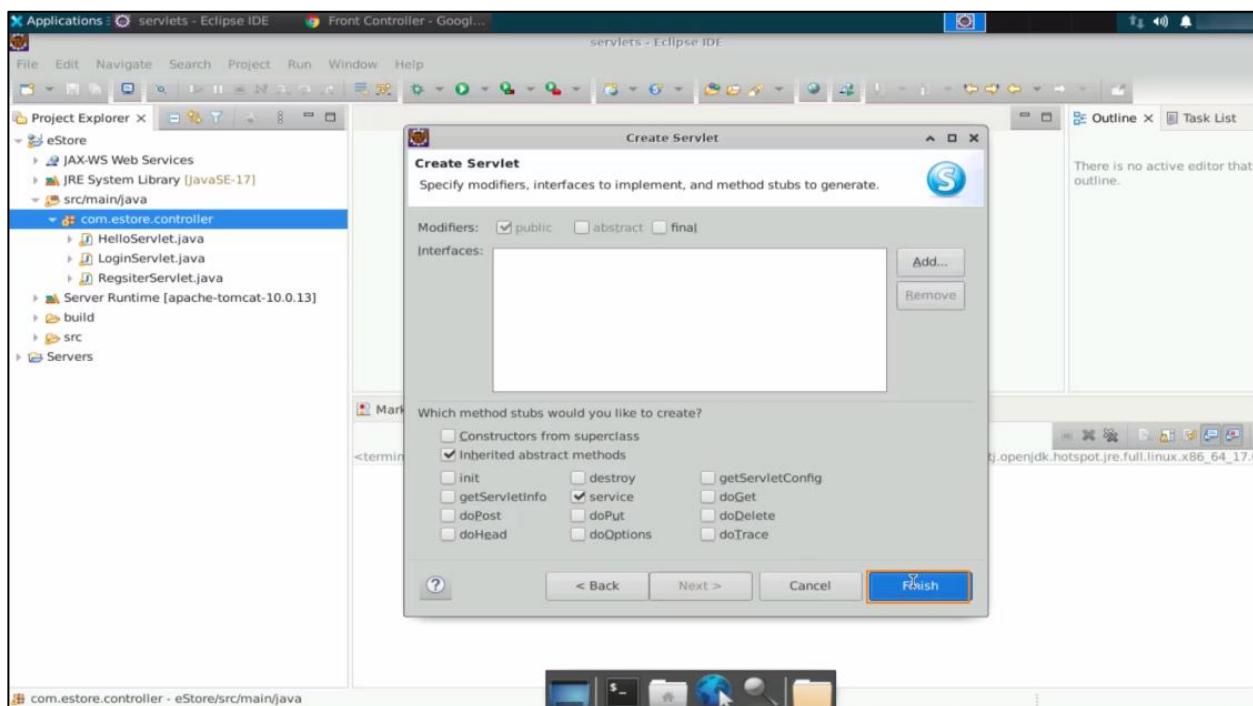
## 1.5 Click on Next



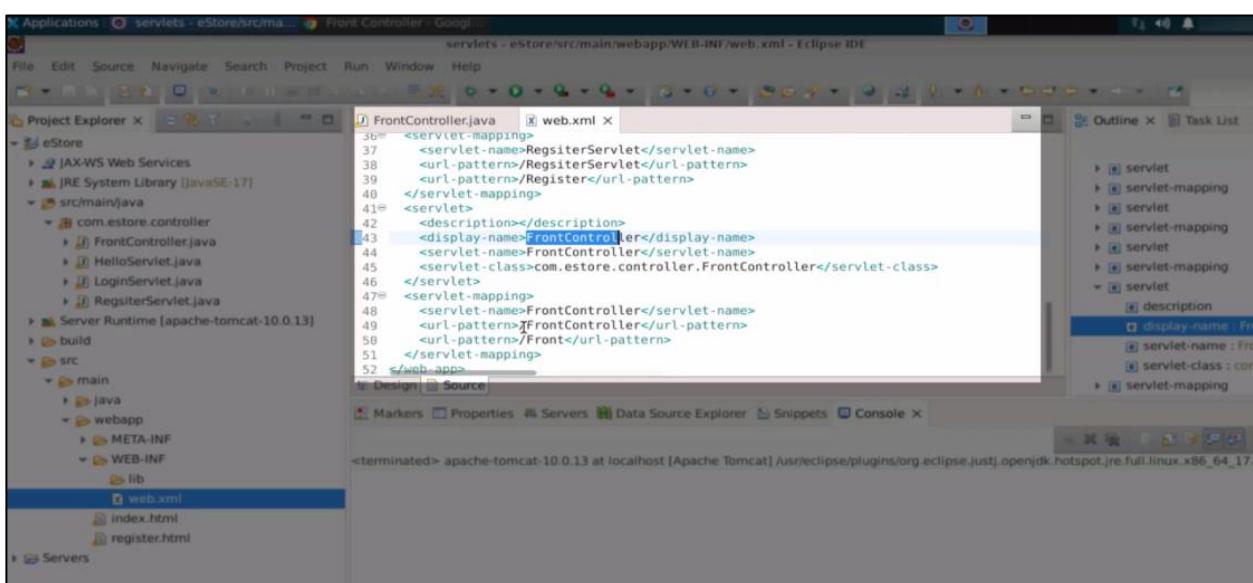
## 1.6 Select service as the method to be included



## 1.7 Click on Finish

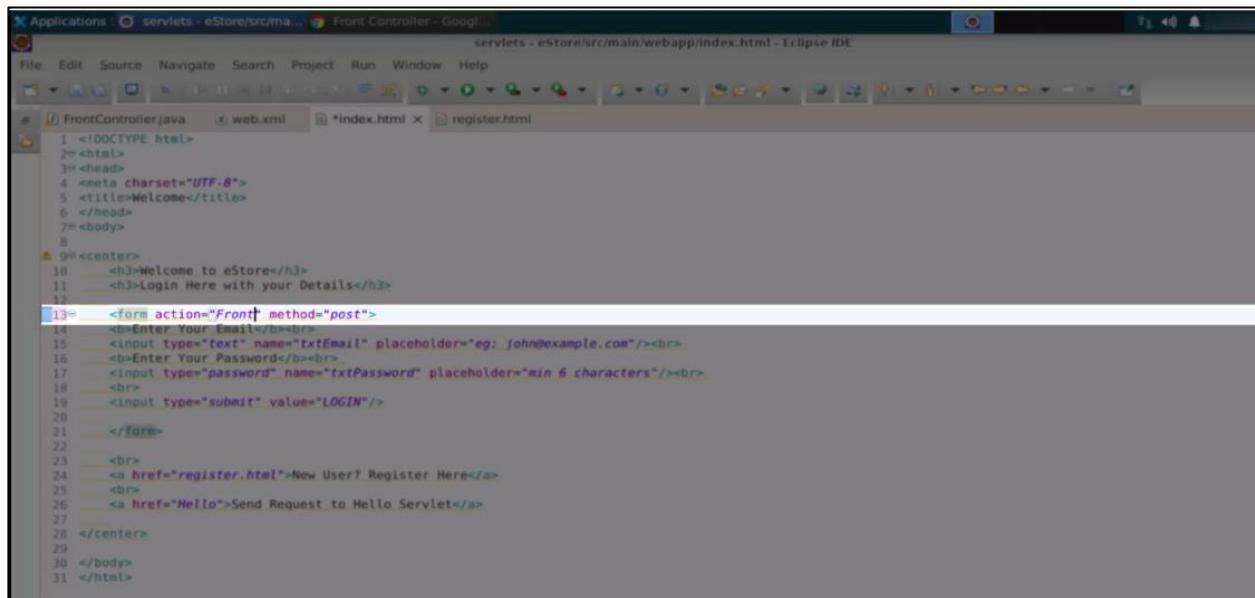


## 1.8 Open the **web.xml** file located in the **webapp** folder where you will find the specified URL pattern as **/FrontController**



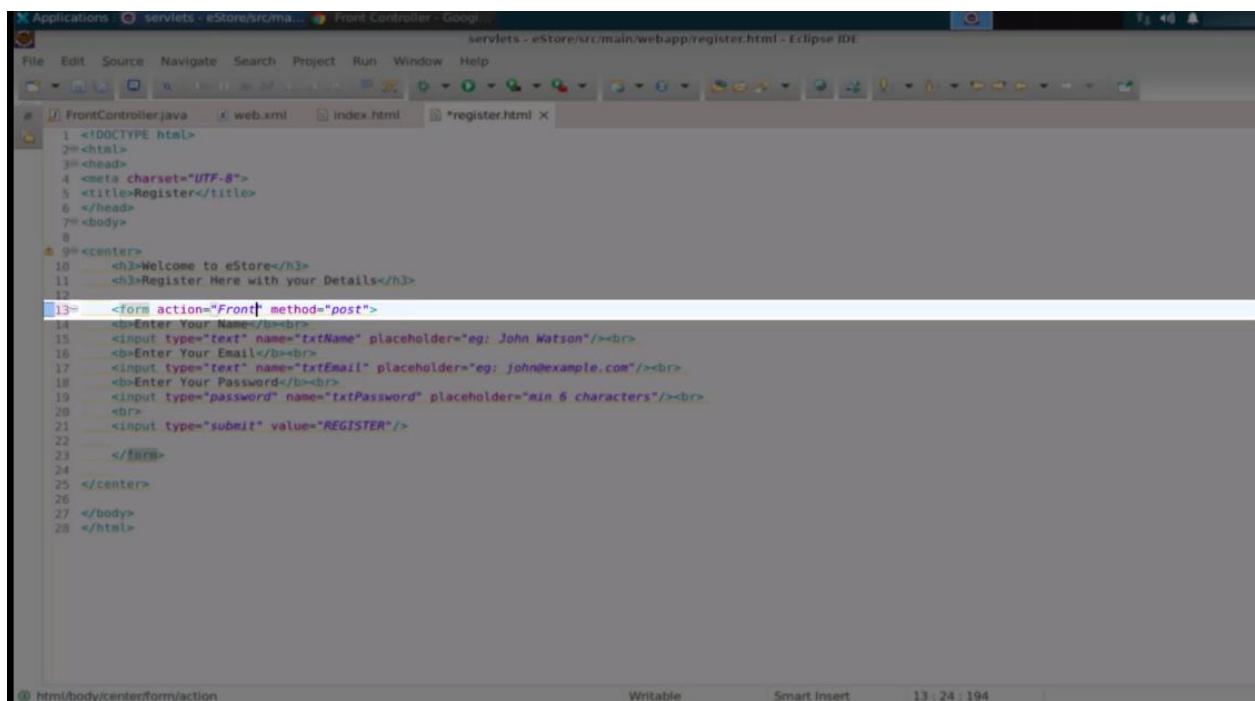
## Step 2: Send a request to the FrontController servlet

2.1 Open the **index.html** file and change the action Register to **Front**. This means you will send the request to the Front server, not the Register.



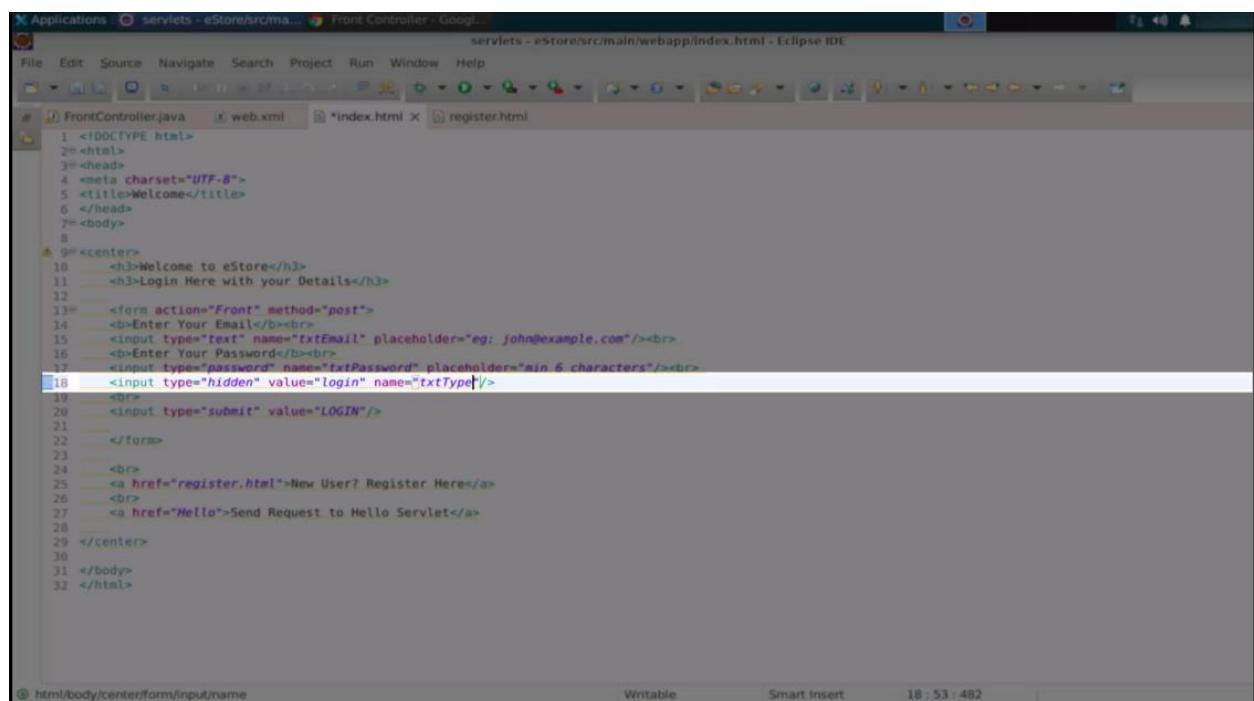
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Welcome</title>
</head>
<body>
<center>
<h3>Welcome to eStore</h3>
<h3>Login Here with your Details</h3>
<form action="Front" method="post">
<br><input type="text" name="txtEmail" placeholder="eg: john@example.com"/><br>
<br><input type="password" name="txtPassword" placeholder="min 6 characters"/><br>
<br><input type="submit" value="LOGIN"/>
</form>
<br>
<br>
<a href="register.html">New User? Register Here</a>
<br>
<a href="Hello">Send Request to Hello Servlet</a>
</center>
</body>
</html>
```

2.2 Open the **register.html** file and change the action to **Front**



```
<!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="Front" method="post">
<br><input type="text" name="txtName" placeholder="eg: John Watson"/><br>
<br><input type="text" name="txtEmail" placeholder="eg: john@example.com"/><br>
<br><input type="password" name="txtPassword" placeholder="min 6 characters"/><br>
<br><input type="submit" value="REGISTER"/>
</form>
<br>
<br>
</center>
</body>
</html>
```

## 2.3 Return to the **index.html** and create one input type with the **hidden** and value as **login**

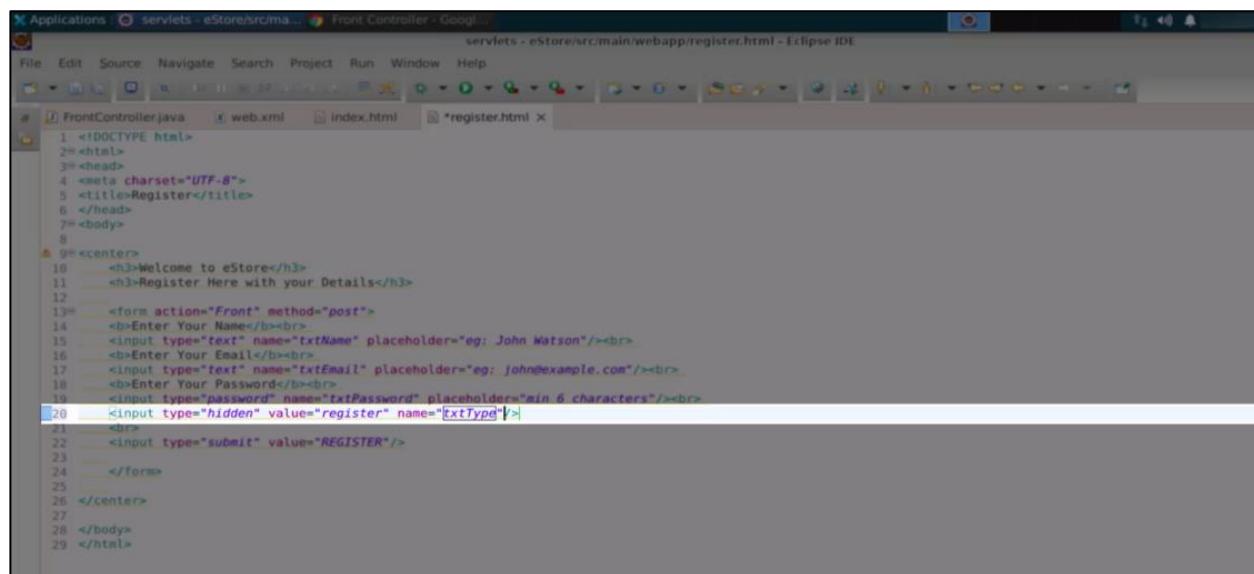


The screenshot shows the Eclipse IDE interface with the file "index.html" open. The code in the editor is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Welcome</title>
6 </head>
7 <body>
8
9 <center>
10 <h3>Welcome to eStore</h3>
11 <h3>Login Here with your Details</h3>
12
13 <form action="Front" method="post">
14 <b><br>
15 <input type="text" name="txtEmail" placeholder="eg: john@example.com"/><br>
16 <b><br>
17 <input type="password" name="txtPassword" placeholder="min 6 characters"/><br>
18 <input type="hidden" value="login" name="txtType"/>
19 <b><br>
20 <input type="submit" value="LOGIN"/>
21
22 </form>
23
24 <br>
25 <a href="register.html">New User? Register Here</a>
26 <br>
27 <a href="Hello">Send Request to Hello Servlet</a>
28
29 </center>
30
31 </body>
32 </html>
```

The cursor is positioned at line 18, column 1, where the input type is defined.

## 2.4 Return to the **register.html** and create one input type with the **hidden** and value as **register**



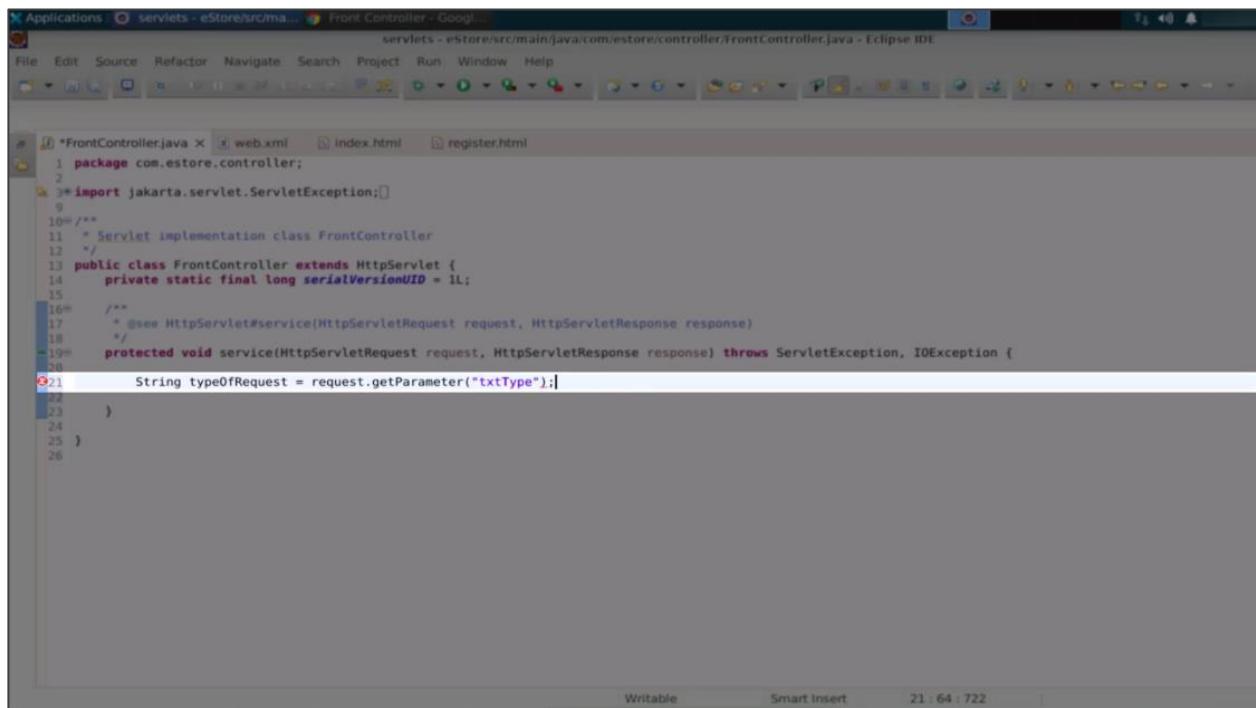
The screenshot shows the Eclipse IDE interface with the file "register.html" open. The code in the editor is as follows:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Register</title>
6 </head>
7 <body>
8
9 <center>
10 <h3>Welcome to eStore</h3>
11 <h3>Register Here with your Details</h3>
12
13 <form action="Front" method="post">
14 <b><br>
15 <input type="text" name="txtName" placeholder="eg: John Watson"/><br>
16 <b><br>
17 <input type="text" name="txtEmail" placeholder="eg: john@example.com"/><br>
18 <b><br>
19 <input type="password" name="txtPassword" placeholder="min 6 characters"/><br>
20 <input type="hidden" value="register" name="txtType"/>
21 <b><br>
22 <input type="submit" value="REGISTER"/>
23
24 </form>
25
26 </center>
27
28 </body>
29 </html>
```

The cursor is positioned at line 20, column 1, where the input type is defined.

## Step 3: Create a method in the FrontController servlet

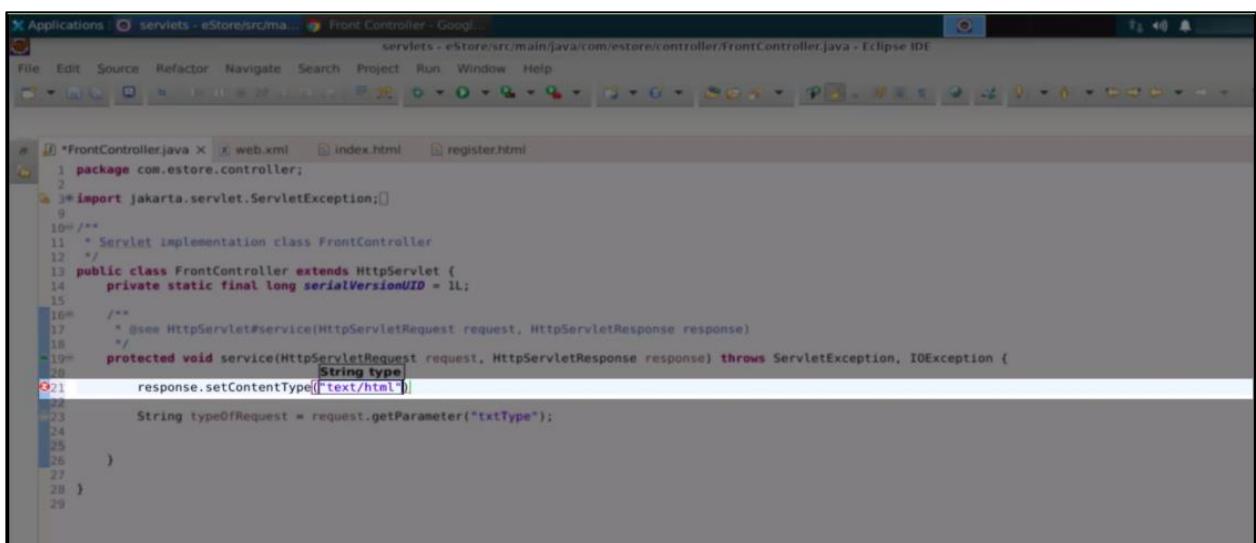
3.1 Return to the **FrontController.java** and add the **typeOfRequest** attribute using the **request.getParameter()** function of the **txt** type



The screenshot shows the Eclipse IDE interface with the file **FrontController.java** open. The code defines a class **FrontController** that extends **HttpServlet**. It includes imports for **com.estore.controller** and **jakarta.servlet.ServletException**. The **service** method takes **HttpServletRequest** and **HttpServletResponse** parameters and throws **ServletException** and **IOException**. Inside the method, the line **String typeOfRequest = request.getParameter("txtType");** is highlighted.

```
1 package com.estore.controller;
2
3 import jakarta.servlet.ServletException;
4
5 /**
6  * Servlet implementation class FrontController
7  */
8 public class FrontController extends HttpServlet {
9     private static final long serialVersionUID = 1L;
10
11     /**
12      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
13     */
14     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
15
16         /**
17          * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
18         */
19         protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
20
21             String typeOfRequest = request.getParameter("txtType");
22
23         }
24
25     }
26 }
```

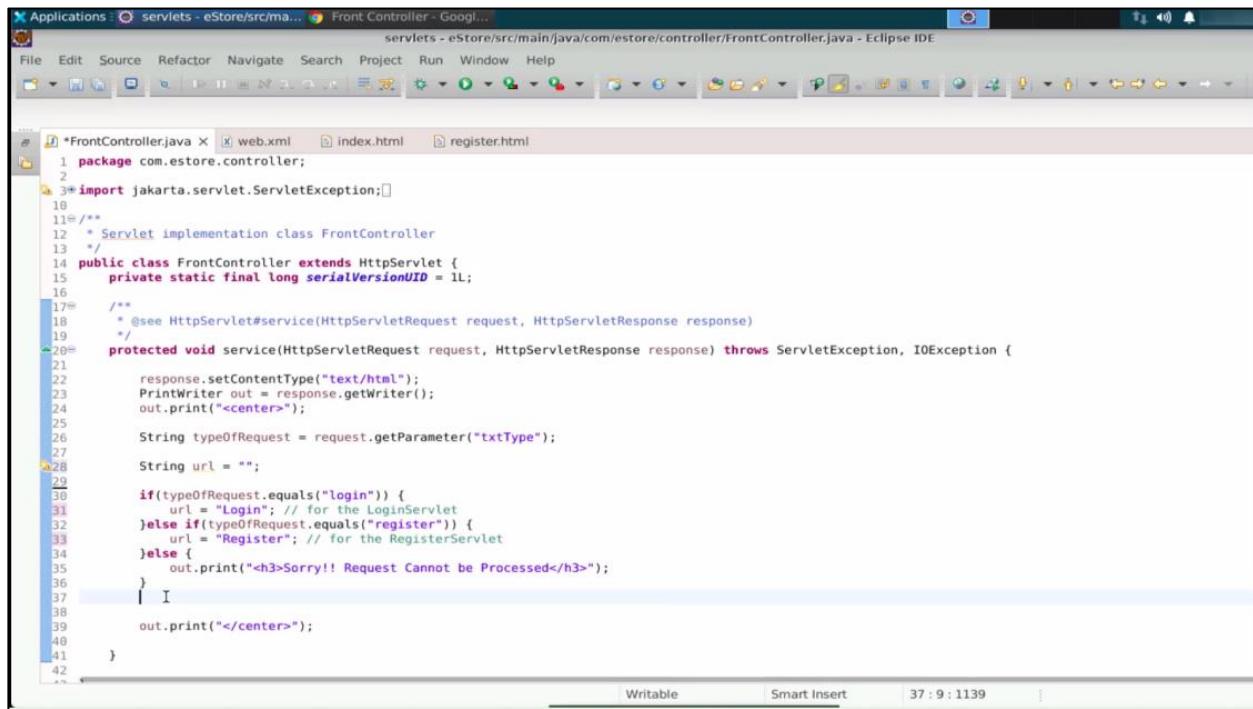
3.2 Use the **response.setContentType()** function to set the response type as **HTML**



The screenshot shows the Eclipse IDE interface with the file **FrontController.java** open. The code is identical to the previous screenshot, but the line **String typeOfRequest = request.getParameter("txtType");** has been replaced by **response.setContentType("text/html");**, which is highlighted.

```
1 package com.estore.controller;
2
3 import jakarta.servlet.ServletException;
4
5 /**
6  * Servlet implementation class FrontController
7  */
8 public class FrontController extends HttpServlet {
9     private static final long serialVersionUID = 1L;
10
11     /**
12      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
13     */
14     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
15
16         /**
17          * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
18         */
19         protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
20
21             response.setContentType("text/html");
22
23             String typeOfRequest = request.getParameter("txtType");
24
25         }
26
27     }
28 }
```

3.3 Create an object for the response and write the conditions using the **response.getWriter()** function. Also, check if the type of request is register (lines 22 to 36). Please note that the URL is **Login** for the LoginServlet and **Register** for the RegisterServlet.

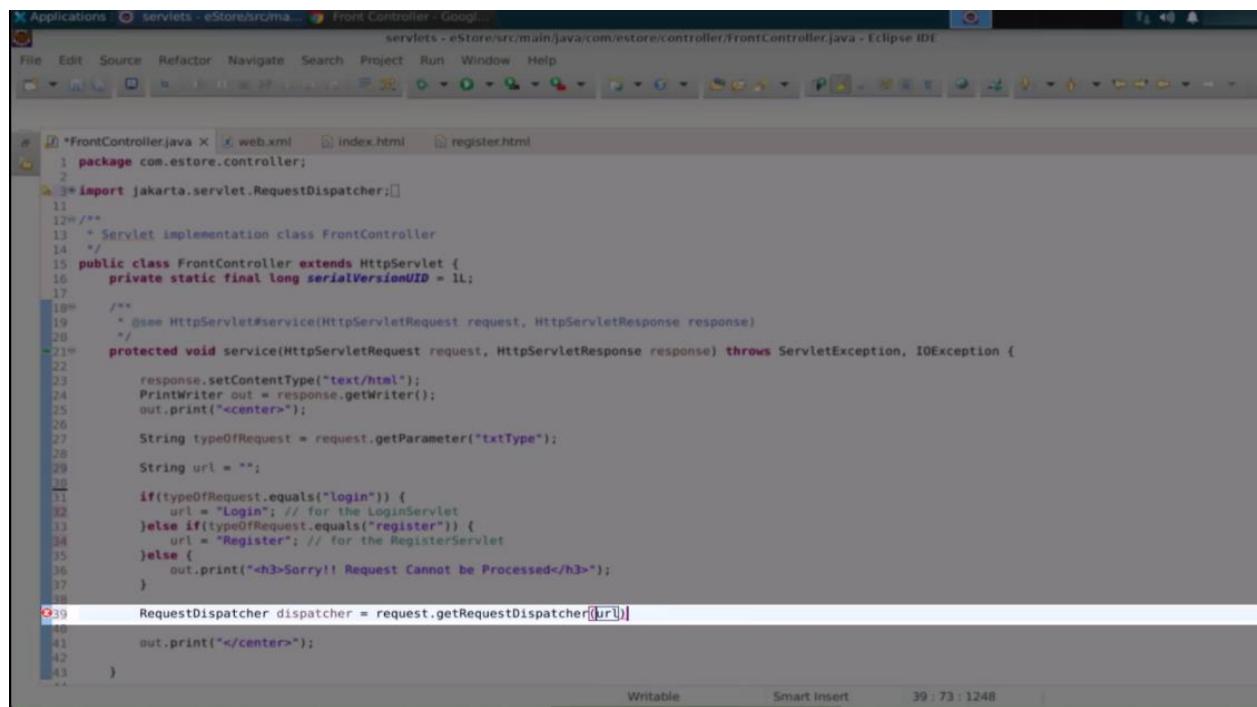


The screenshot shows the Eclipse IDE interface with the title bar "Applications : servlets - eStore/src/main/java/com/estore/controller - Google Chrome". The code editor displays "FrontController.java" with the following content:

```
1 package com.estore.controller;
2
3 import jakarta.servlet.ServletException;
4
5 /**
6  * Servlet implementation class FrontController
7 */
8 public class FrontController extends HttpServlet {
9     private static final long serialVersionUID = 1L;
10
11     /**
12      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
13     */
14     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
15         response.setContentType("text/html");
16         PrintWriter out = response.getWriter();
17         out.print("<center>");
18
19         String typeOfRequest = request.getParameter("txtType");
20
21         String url = "";
22
23         if(typeOfRequest.equals("login")) {
24             url = "Login"; // for the LoginServlet
25         }else if(typeOfRequest.equals("register")) {
26             url = "Register"; // for the RegisterServlet
27         }else {
28             out.print("<h3>Sorry!! Request Cannot be Processed</h3>");
29         }
30         out.print("</center>");
31     }
32 }
33
34
35
36
37
38
39
40
41
42 }
```

The code implements a servlet named **FrontController** that handles requests. It checks the parameter **txtType** to determine the URL to forward. If **txtType** is **"login"**, it sets the URL to **"Login"**. If **txtType** is **"register"**, it sets the URL to **"Register"**. Otherwise, it prints an error message. The code is annotated with JavaDoc comments and imports the **Jakarta Servlet API**.

3.4 Use the **getRequestDispatcher()** function to specify the URL in the dispatcher, which represents the path to where you want to navigate



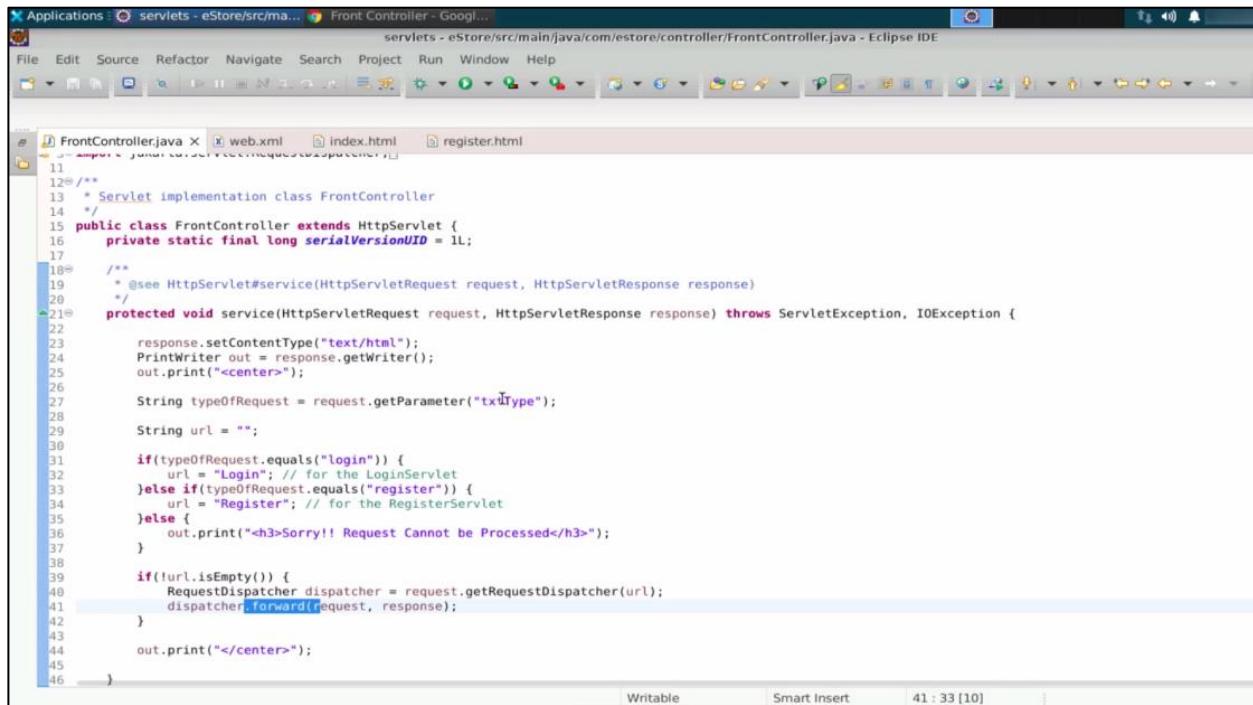
The screenshot shows the Eclipse IDE interface with the file "FrontController.java" open. The code implements a Servlet named "FrontController" that handles requests based on their type. It uses the `getRequestDispatcher()` method to forward requests to specific URLs ("Login" or "Register") and prints an error message if the type is neither.

```
File Edit Source Refactor Navigate Search Project Run Window Help

FrontController.java [x] web.xml [x] index.html [x] register.html

1 package com.estore.controller;
2
3 import jakarta.servlet.RequestDispatcher;
4
5 /**
6  * Servlet implementation class FrontController
7  */
8 public class FrontController extends HttpServlet {
9     private static final long serialVersionUID = 1L;
10
11     /**
12      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
13     */
14     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
15         response.setContentType("text/html");
16         PrintWriter out = response.getWriter();
17         out.print("<center>");
18
19         String typeOfRequest = request.getParameter("txtType");
20
21         String url = "";
22
23         if(typeOfRequest.equals("Login")) {
24             url = "Login"; // for the LoginServlet
25         }else if(typeOfRequest.equals("register")) {
26             url = "Register"; // for the RegisterServlet
27         }else {
28             out.print("<h3>Sorry!! Request Cannot be Processed</h3>");
29         }
30
31         RequestDispatcher dispatcher = request.getRequestDispatcher(url);
32         out.print("</center>");
33     }
34 }
```

3.5 If the URL is empty, forward the dispatcher to an error page or any corresponding Servlet. The final code will look like this:

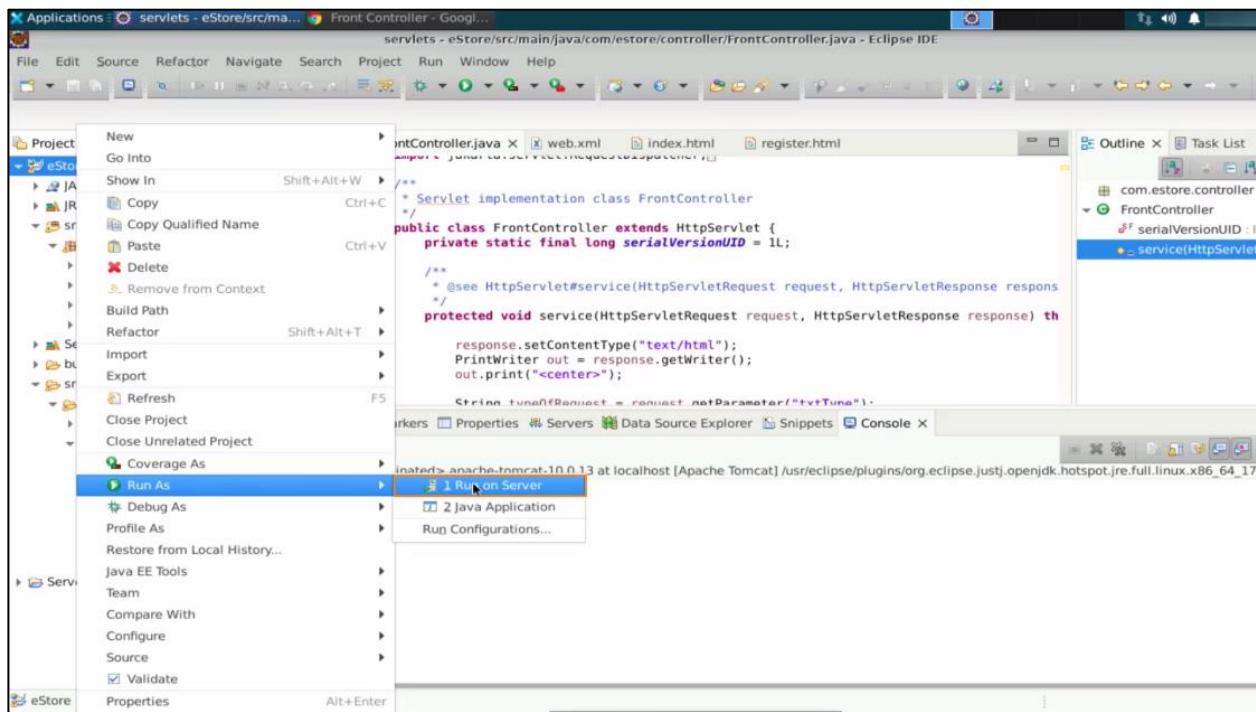


The screenshot shows the Eclipse IDE interface with the 'Front Controller - Google Chrome' tab active. The code editor displays 'FrontController.java' with the following content:

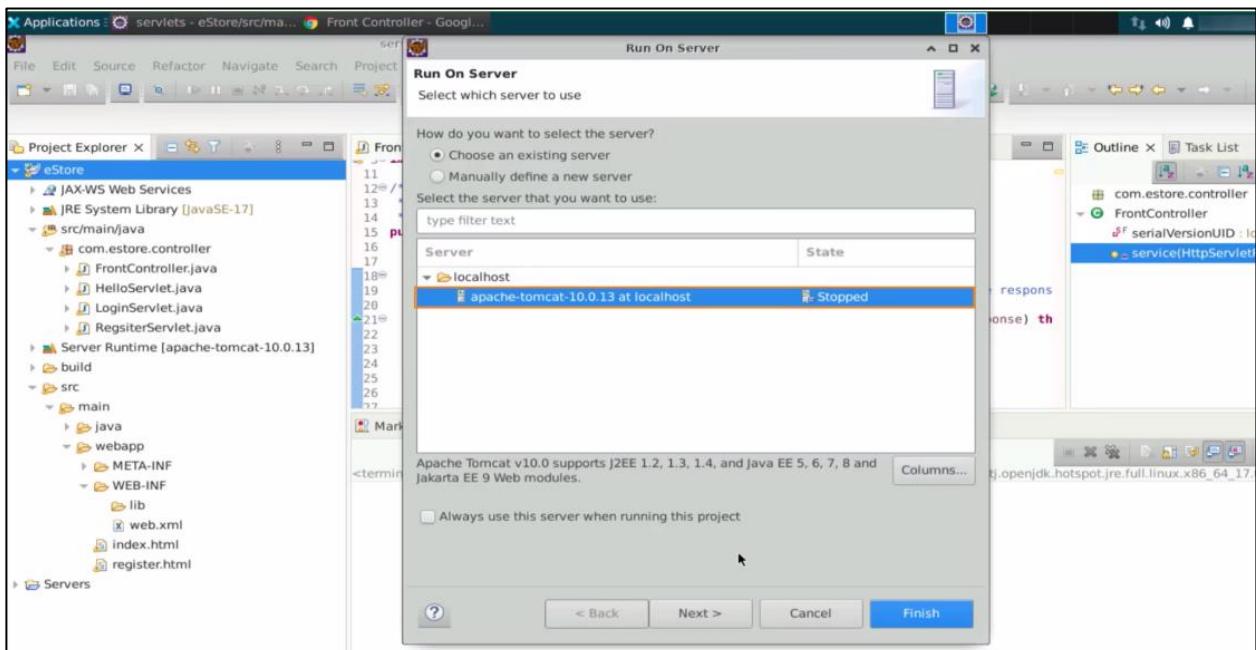
```
FrontController.java X web.xml index.html register.html
File Edit Source Refactor Navigate Search Project Run Window Help
11
12 /**
13  * Servlet implementation class FrontController
14 */
15 public class FrontController extends HttpServlet {
16     private static final long serialVersionUID = 1L;
17
18     /**
19      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
20     */
21     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
22
23         response.setContentType("text/html");
24         PrintWriter out = response.getWriter();
25         out.print("<center>");
26
27         String typeOfRequest = request.getParameter("txtType");
28
29         String url = "";
30
31         if(typeOfRequest.equals("login")) {
32             url = "Login"; // for the LoginServlet
33         }else if(typeOfRequest.equals("register")) {
34             url = "Register"; // for the RegisterServlet
35         }else {
36             out.print("<h3>Sorry!! Request Cannot be Processed</h3>");
37         }
38
39         if(!url.isEmpty()) {
40             RequestDispatcher dispatcher = request.getRequestDispatcher(url);
41             dispatcher.forward(request, response);
42         }
43
44         out.print("</center>");
45     }
46 }
```

The code implements a servlet named 'FrontController' that handles requests. It checks the parameter 'txtType' to determine the URL ('Login' or 'Register'). If the URL is empty, it prints an error message. Otherwise, it forwards the request to the specified URL using a RequestDispatcher.

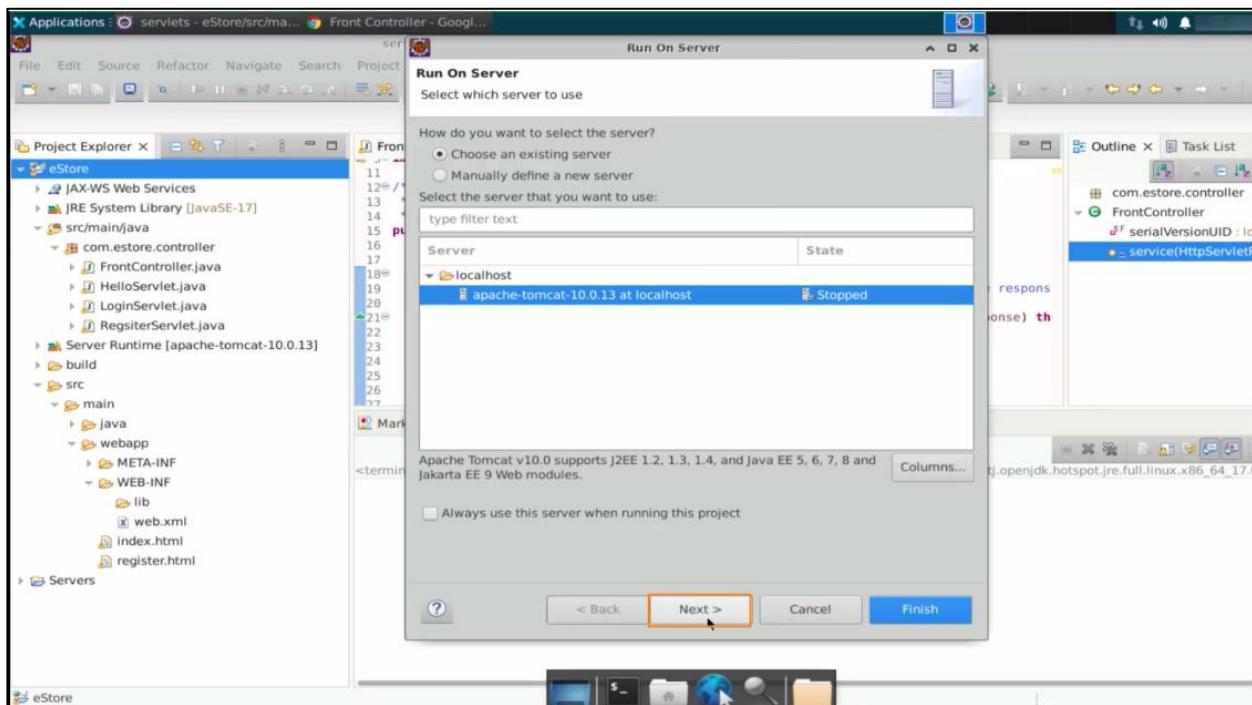
3.6 Save and run the project on the server. Right-click on the project, select **Run As**, and click on **Run on Server**:



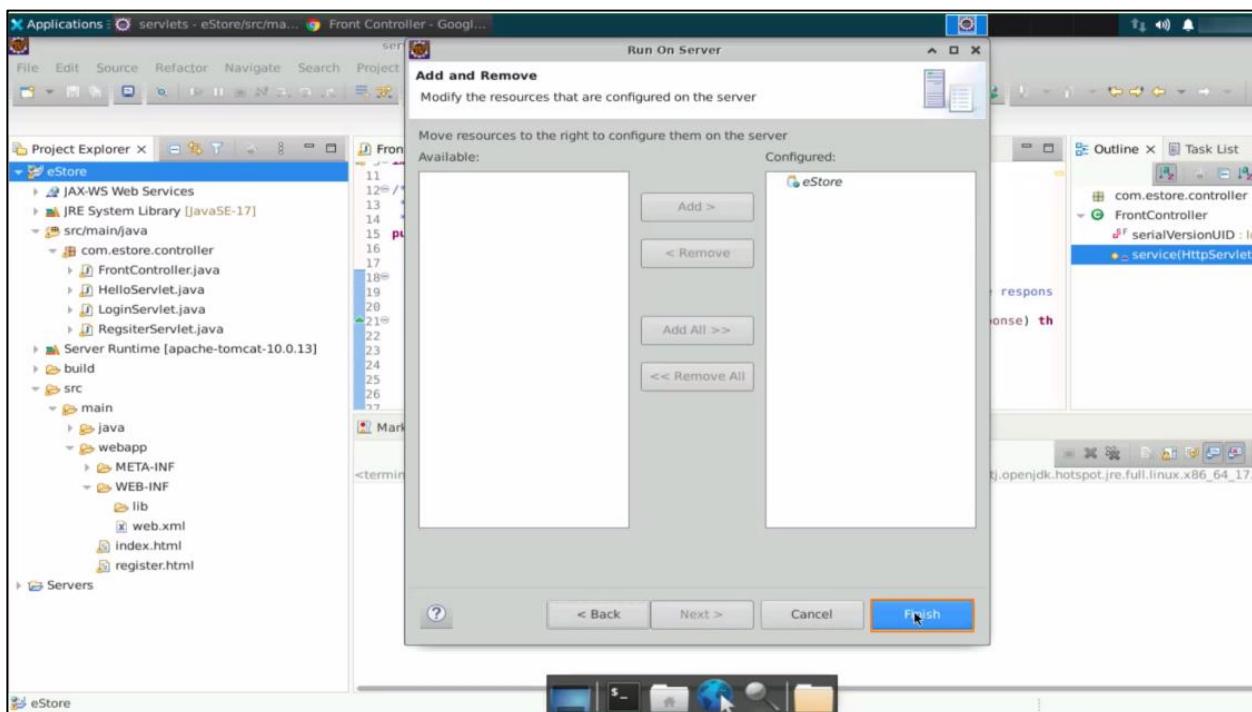
3.7 Select **apache-tomcat-10.0.13 at localhost** as the server



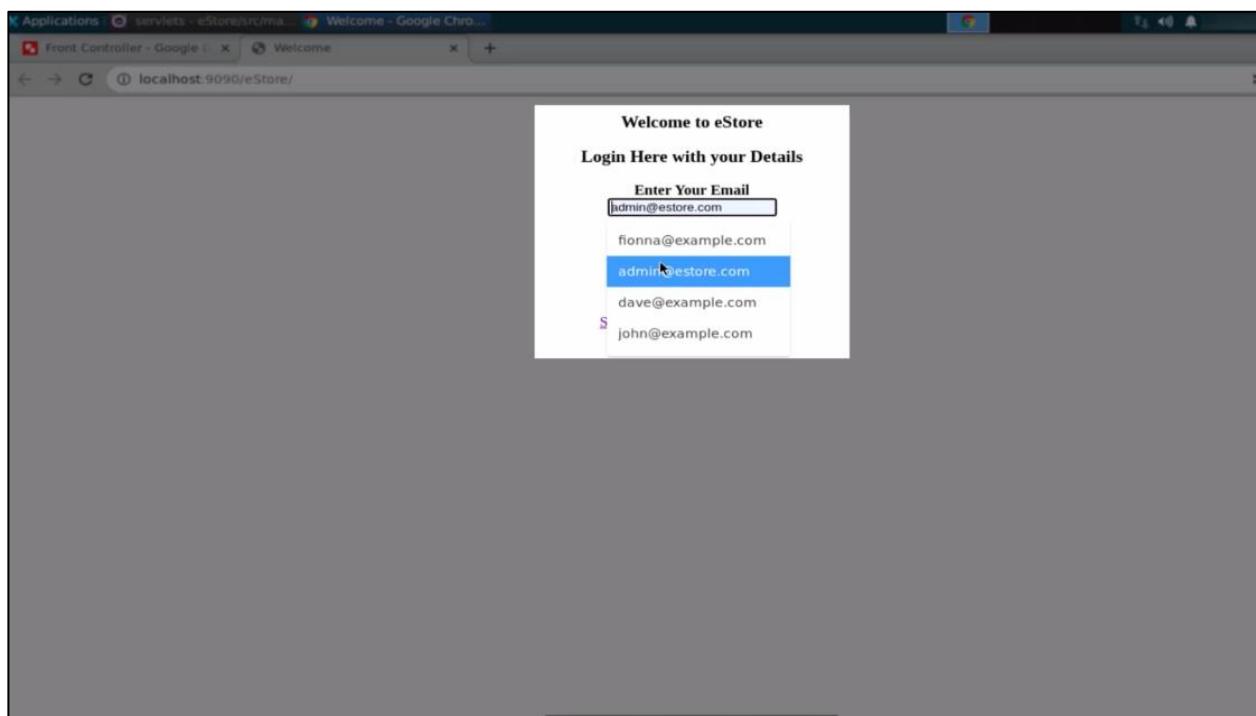
### 3.8 Click on Next



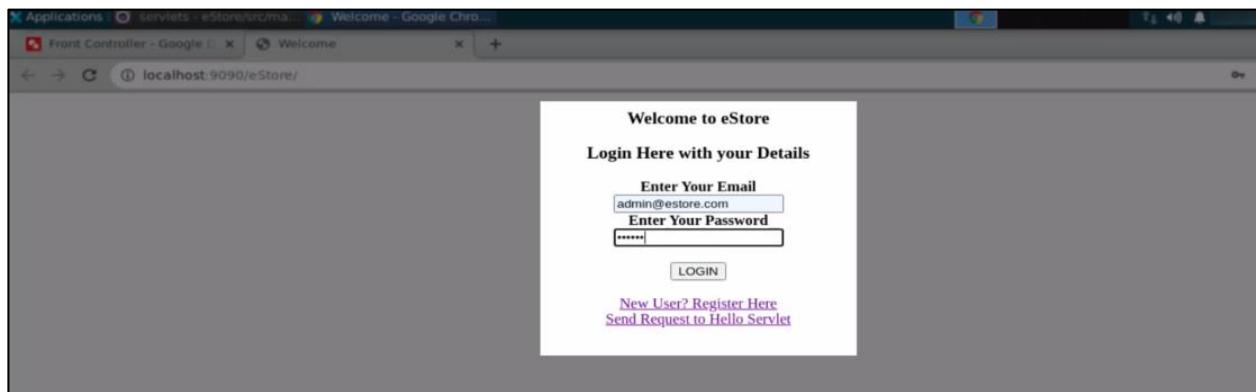
### 3.9 Click on Finish



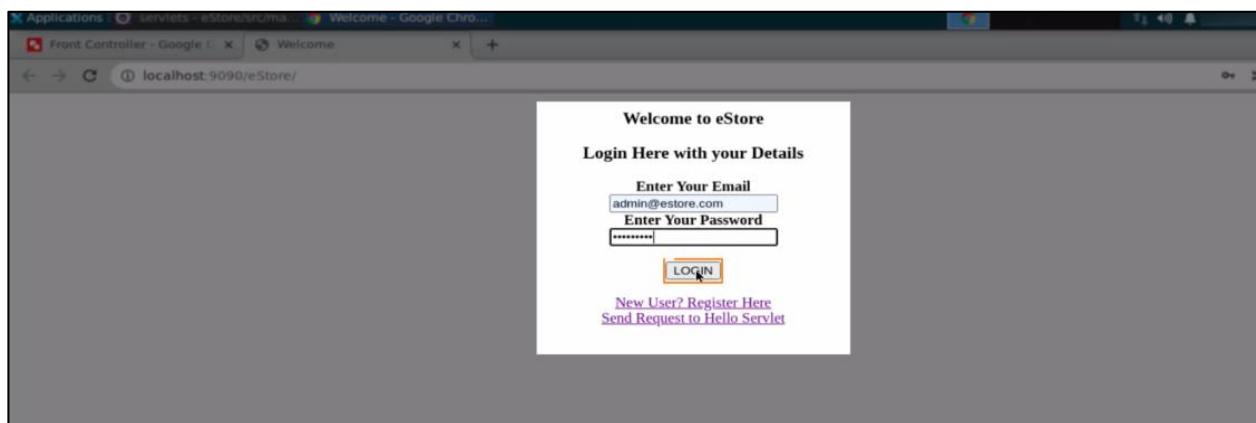
### 3.10 Enter the email as **admin@estore.com**



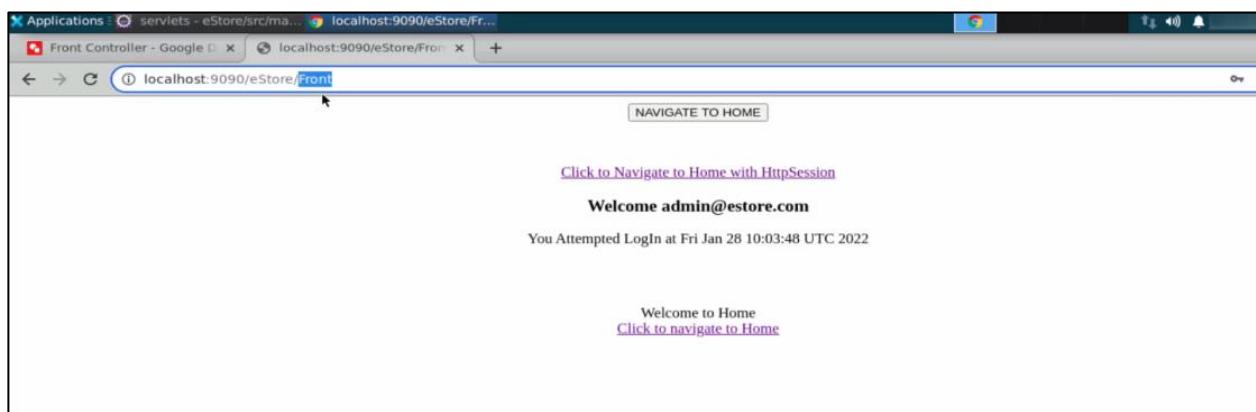
### 3.11 Enter the password as **admin@123**



### 3.12 Click on LOGIN



You will be redirected to the front controller, which corresponds to the URL pattern specified in the front controller.



### 3.13 Return and click on New User? Register Here



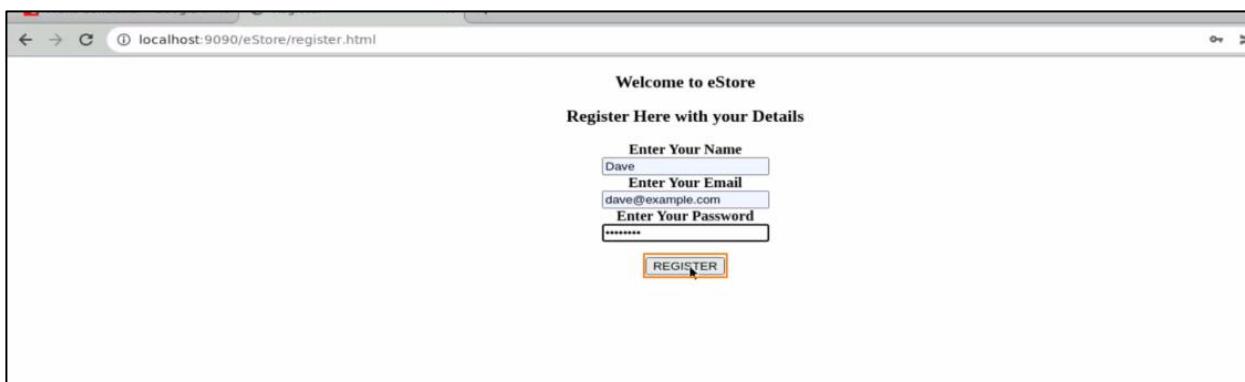
The registration page will look like this:



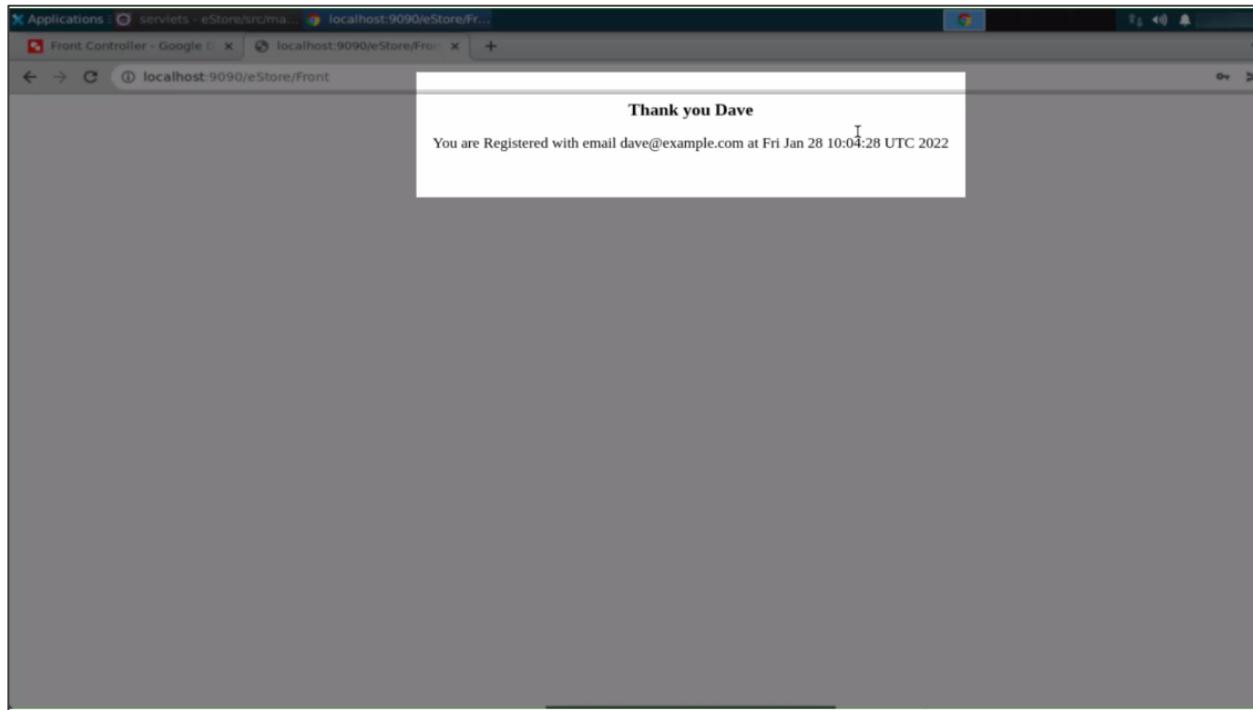
3.14 Enter the name as **Dave**, email as **dave@example.com**, and password as **dave@1**



3.15 Click on the **REGISTER** button

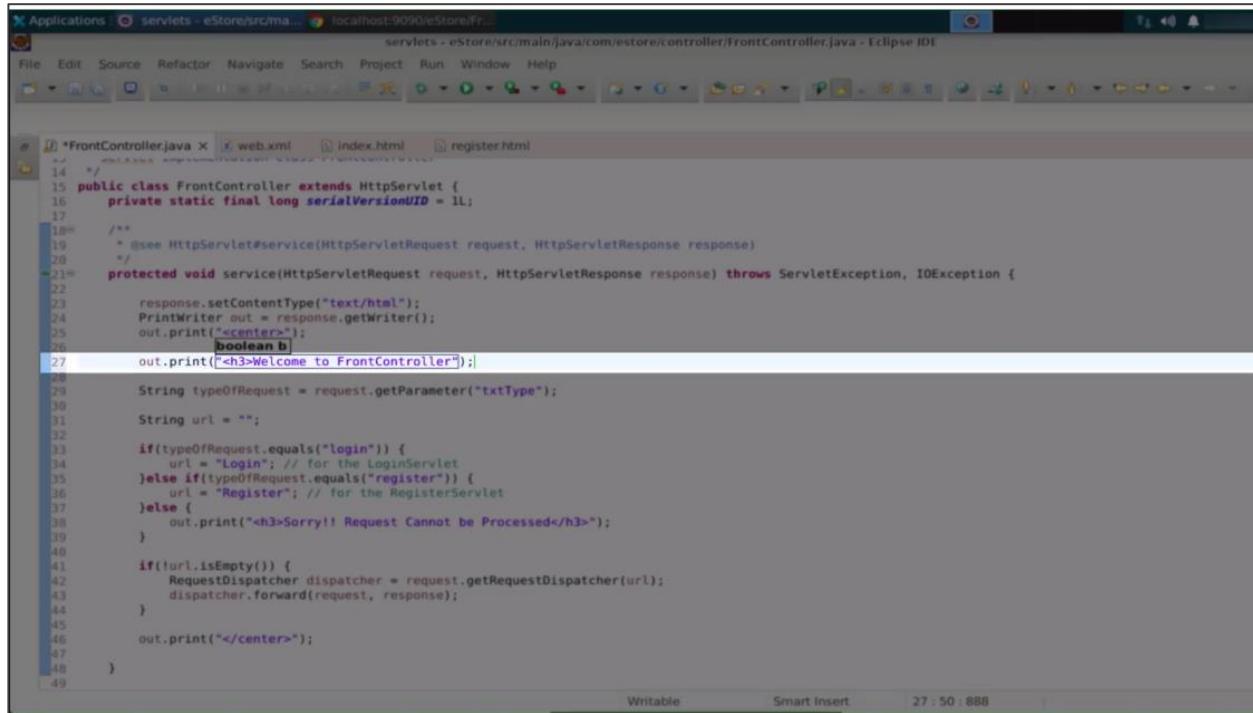


On navigating to the FrontController URL again, you will receive a response from the corresponding Servlet stating **Thank you Dave.**



#### Step 4: Update responses in the FrontController Servlet

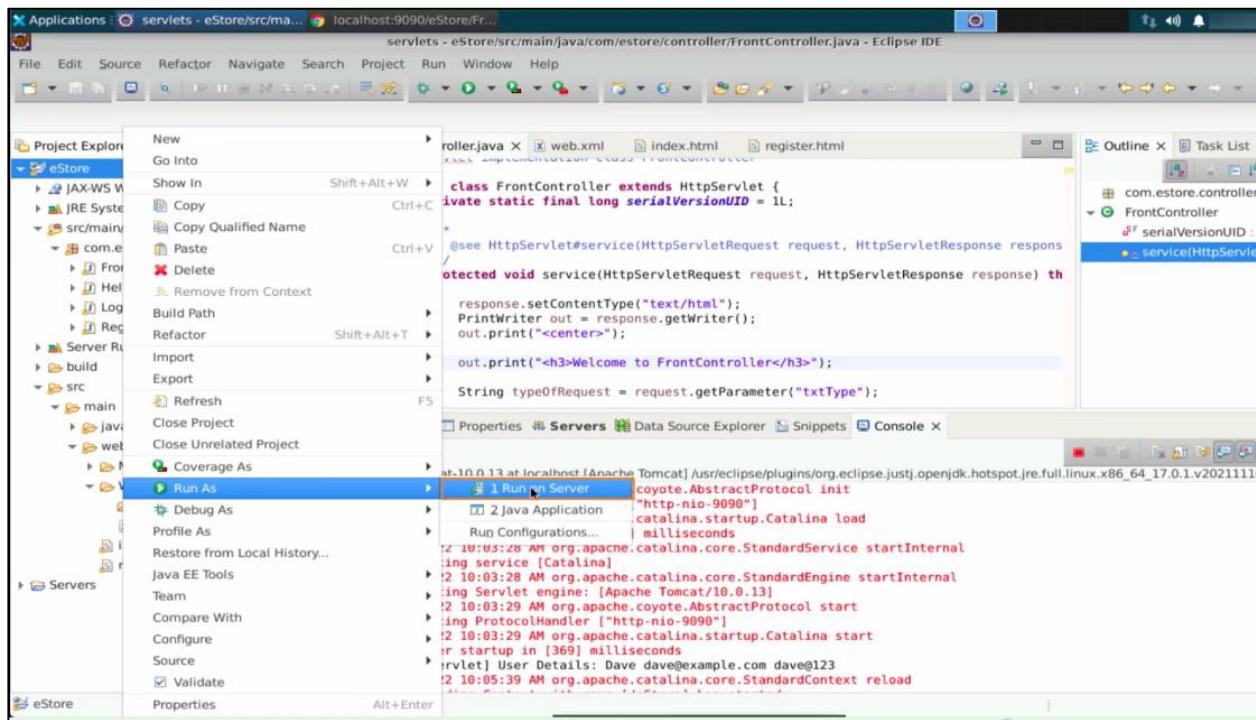
4.1 Return to the **FrontController.java** and enter the response for your Servlet using the **out.print()** function



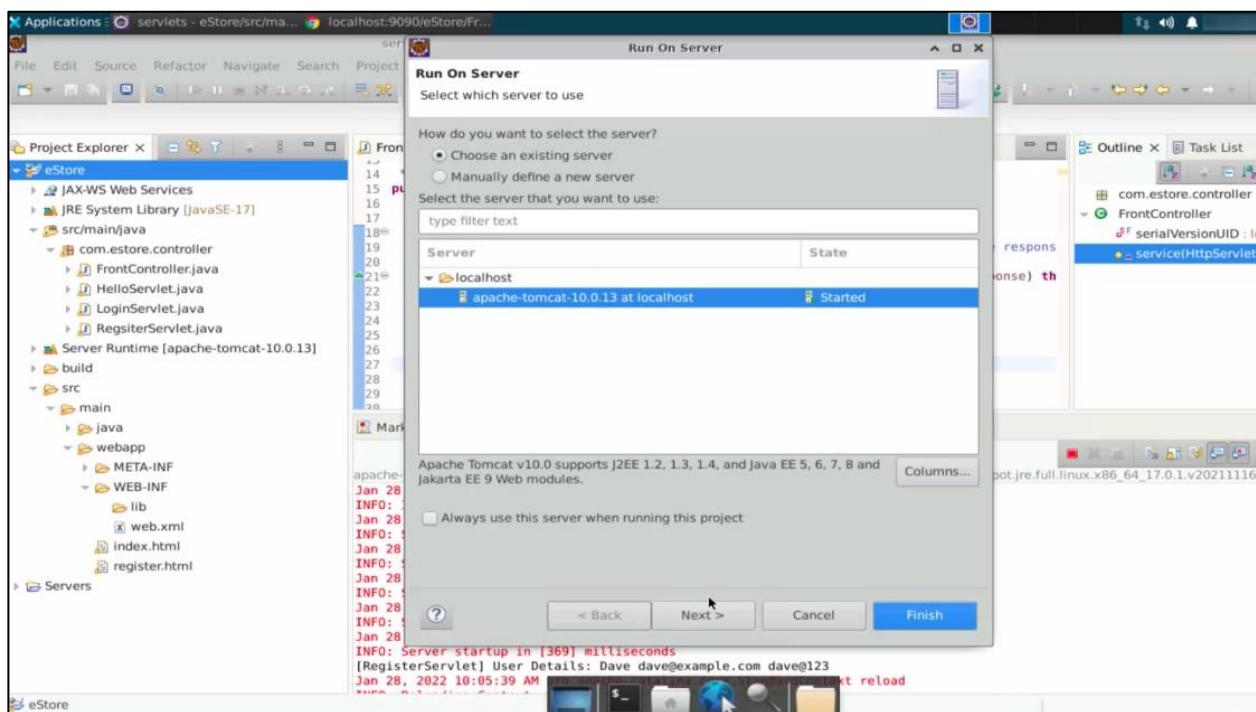
The screenshot shows the Eclipse IDE interface with the code editor open to the **FrontController.java** file. The code implements a **HttpServlet** with a **service** method. Inside the service method, it sets the content type to **text/html**, creates a **PrintWriter** object, and starts printing HTML content. A cursor is positioned at the start of the second line of the printed content, which is **<center>**. The code also handles request parameters and dispatches them to appropriate URLS based on the request type.

```
14  /*
15  * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
16  */
17  protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
18      /*
19      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
20      */
21      response.setContentType("text/html");
22      PrintWriter out = response.getWriter();
23      out.print("<center>");
24      boolean b;
25      b = true;
26      out.print("<h3>Welcome to FrontController</h3>");
27
28      String typeOfRequest = request.getParameter("txtType");
29
30      String url = "";
31
32      if(typeOfRequest.equals("login")) {
33          url = "Login"; // for the LoginServlet
34      }else if(typeOfRequest.equals("register")) {
35          url = "Register"; // for the RegisterServlet
36      }else {
37          out.print("<h3>Sorry!! Request Cannot be Processed</h3>");
38      }
39
40      if(!url.isEmpty()) {
41          RequestDispatcher dispatcher = request.getRequestDispatcher(url);
42          dispatcher.forward(request, response);
43      }
44
45      out.print("</center>");
46
47  }
48
49 }
```

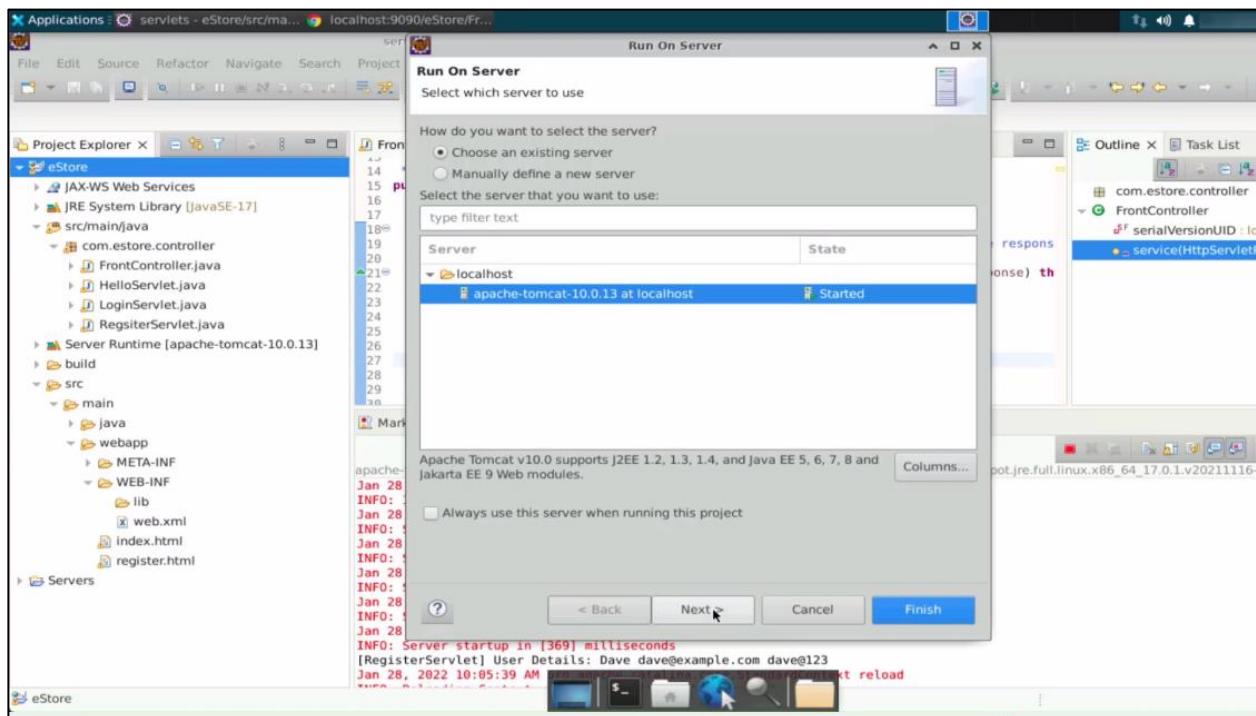
## 4.2 Run the code by selecting Run on Server



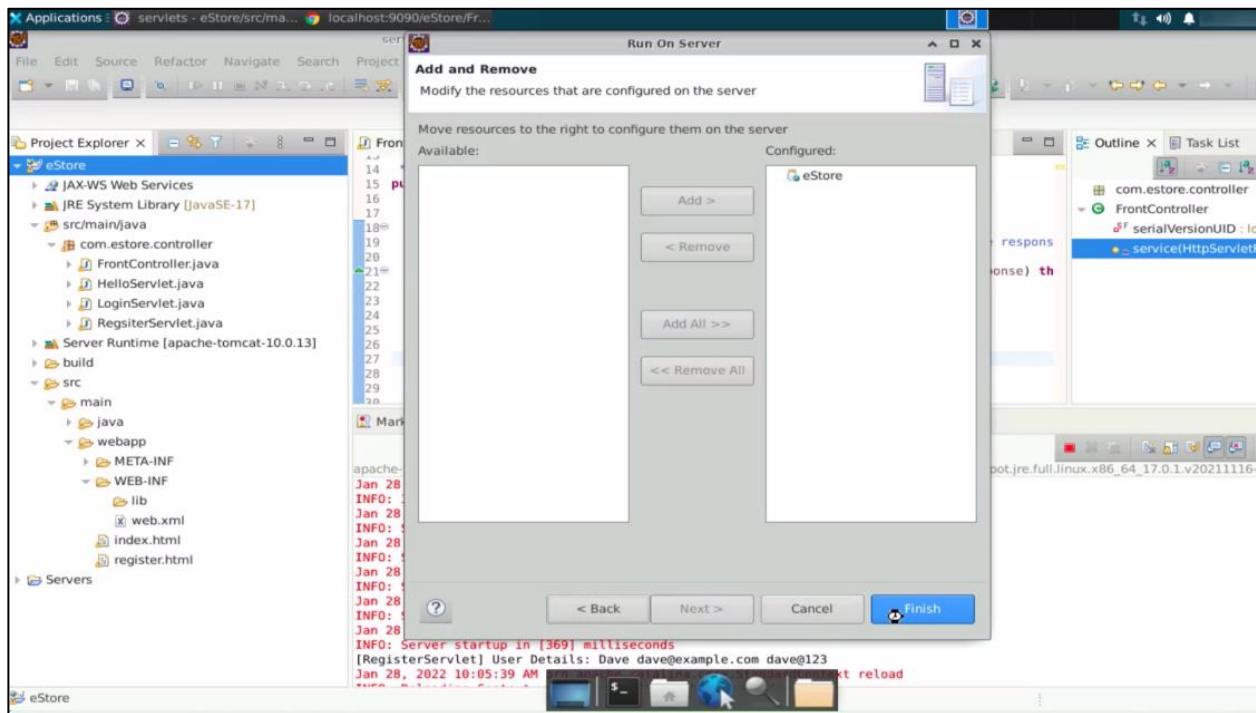
#### 4.3 Select apache-tomcat-10.0.13 at localhost as the server



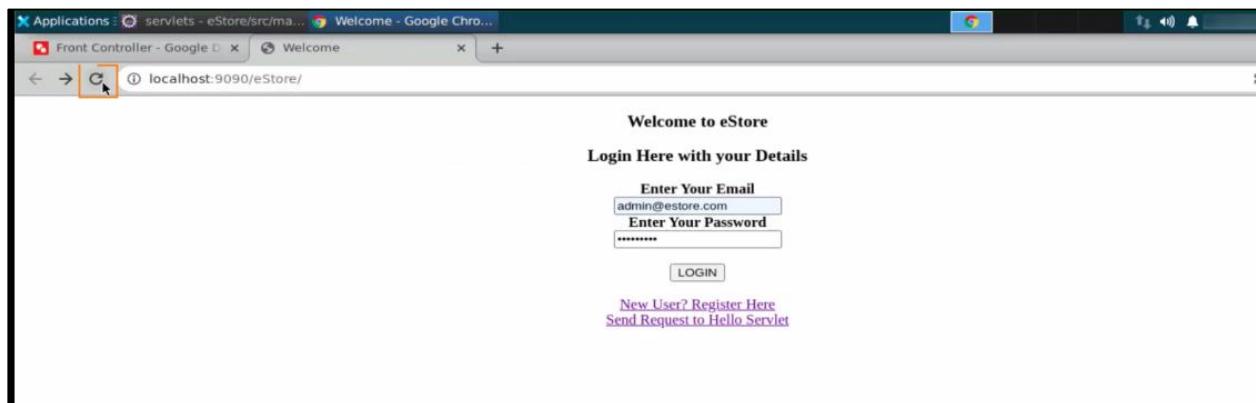
#### 4.4 Click on Next



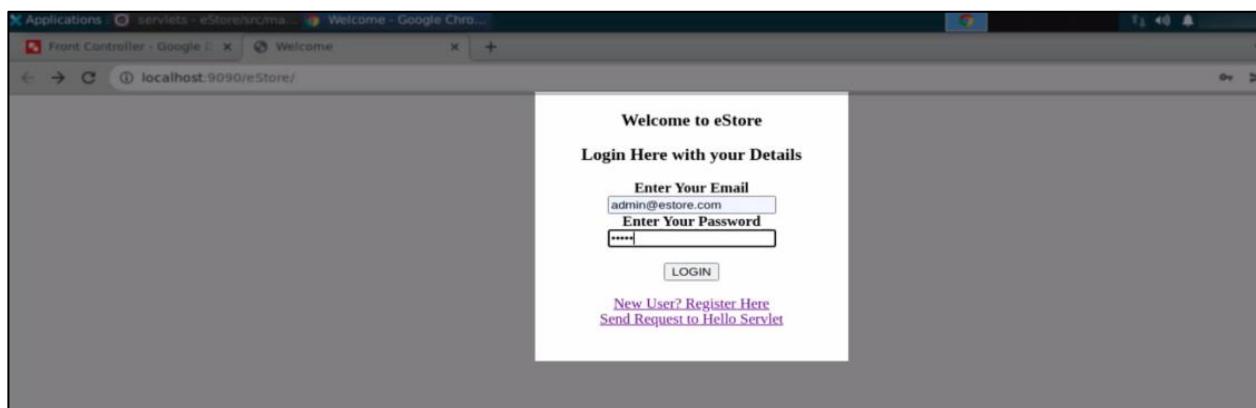
#### 4.5 Click on Finish



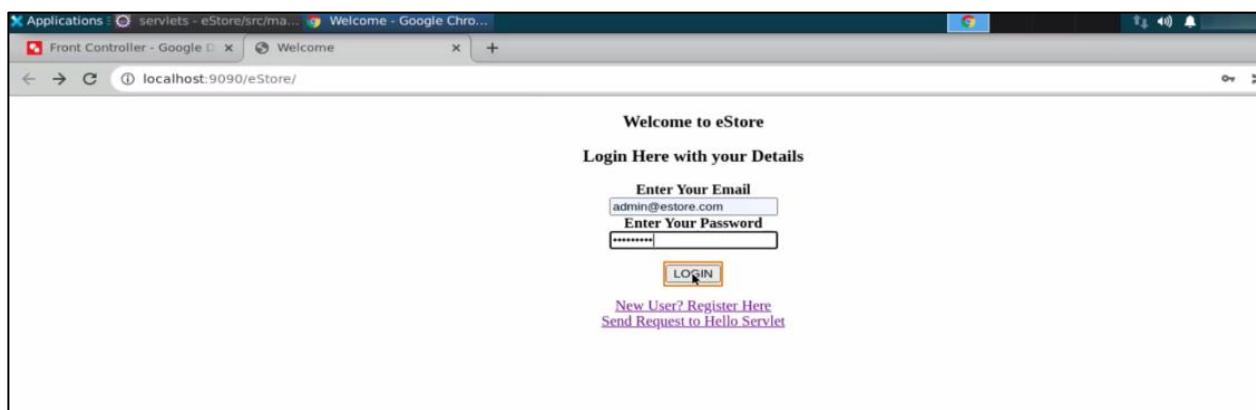
#### 4.6 Return to the browser and refresh the screen



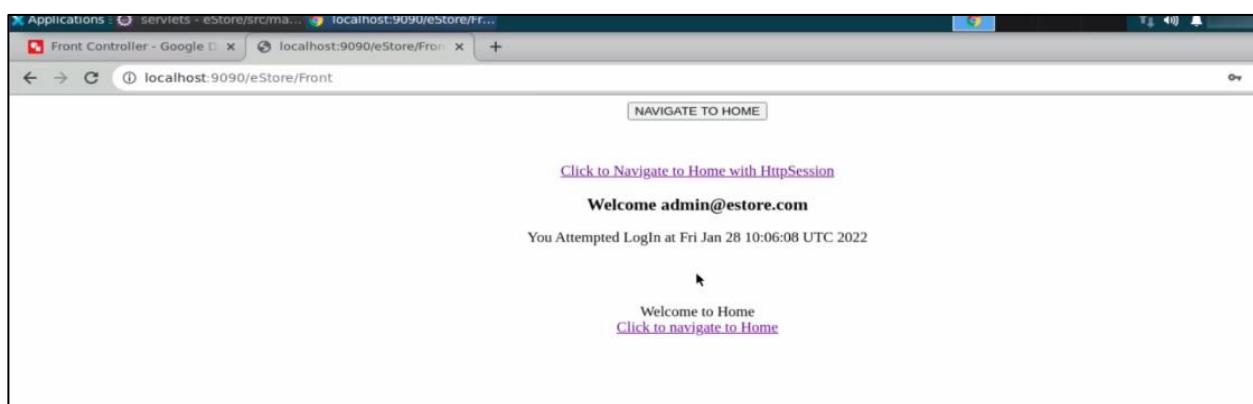
#### 4.7 Enter the email and password for the admin



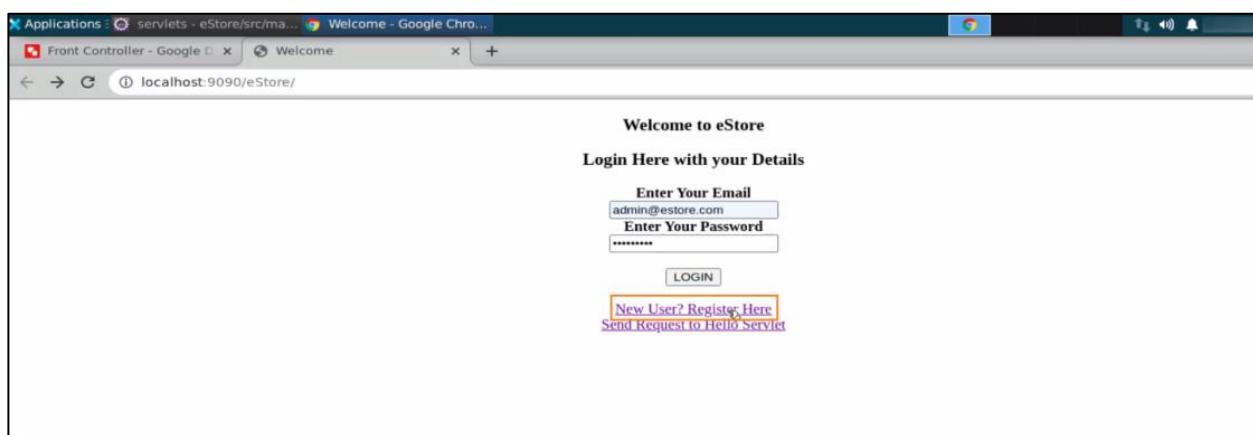
#### 4.8 Click on LOGIN



When you click on the **LOGIN** button, you may not be able to see the response from the front controller due to the use of the **forward** method. This method performs a forward action in both the request and response. The same situation applies to the registration page.



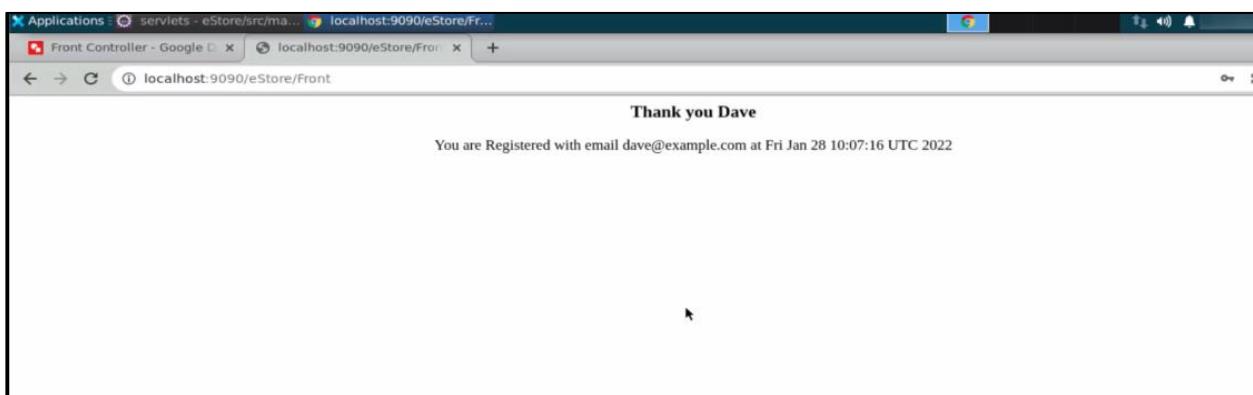
#### 4.9 Return and click on New User? Register Here



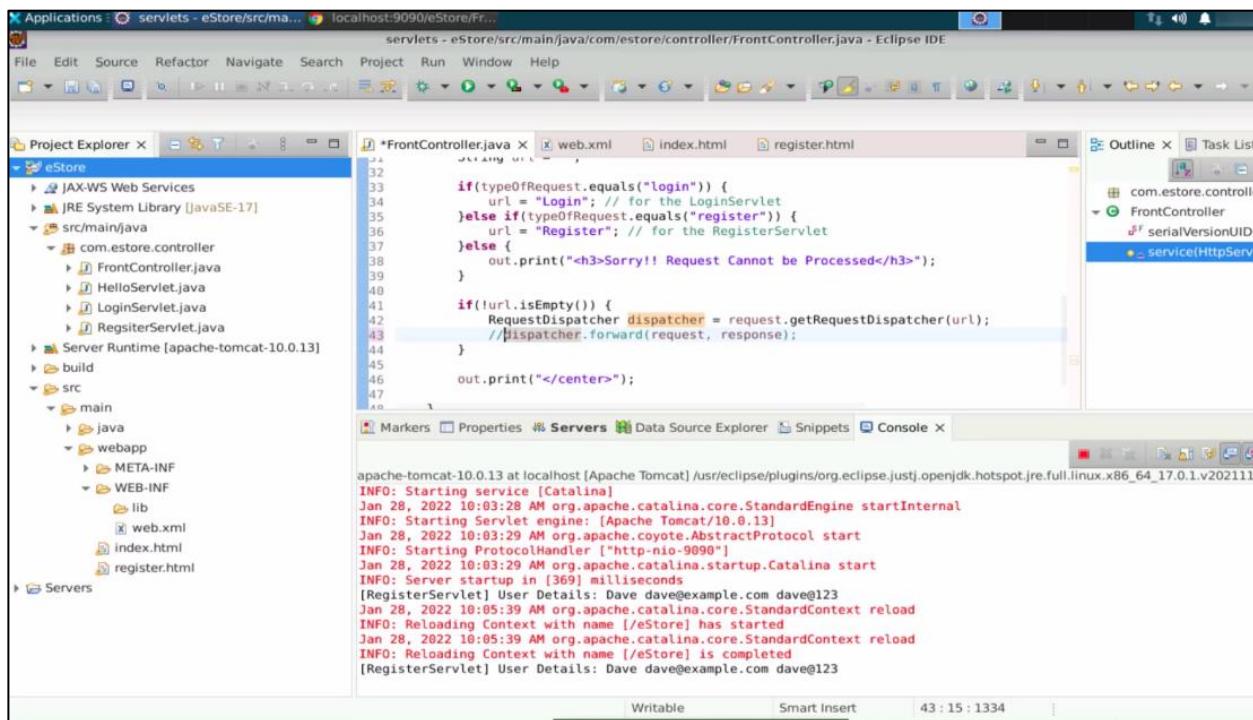
#### 4.10 Enter the name, email, and password, and click on REGISTER



You can see that the response from the front controller is not coming back to your page. This is because of the **dispatcher.forward()** method.



#### 4.11 Return the `FrontController.java` file and mark the `dispatcher.forward()` method as a comment

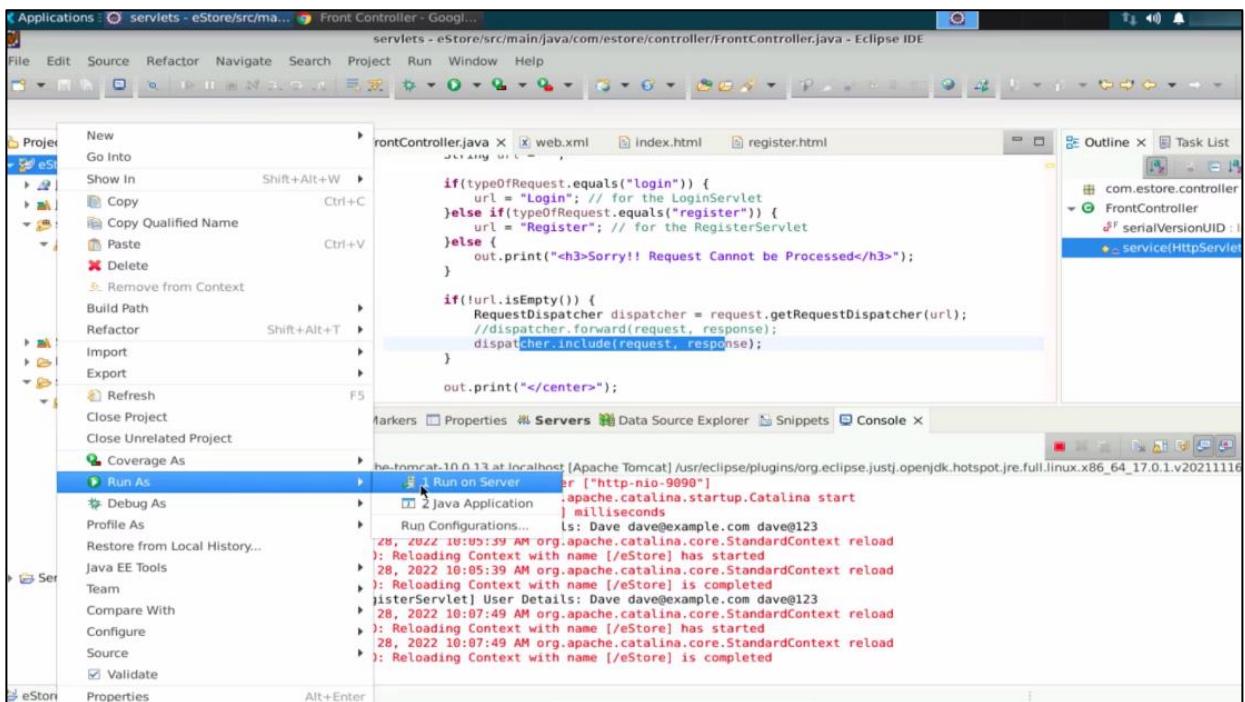


#### 4.12 Add another method called `dispatcher.include()` that includes the response from the front controller

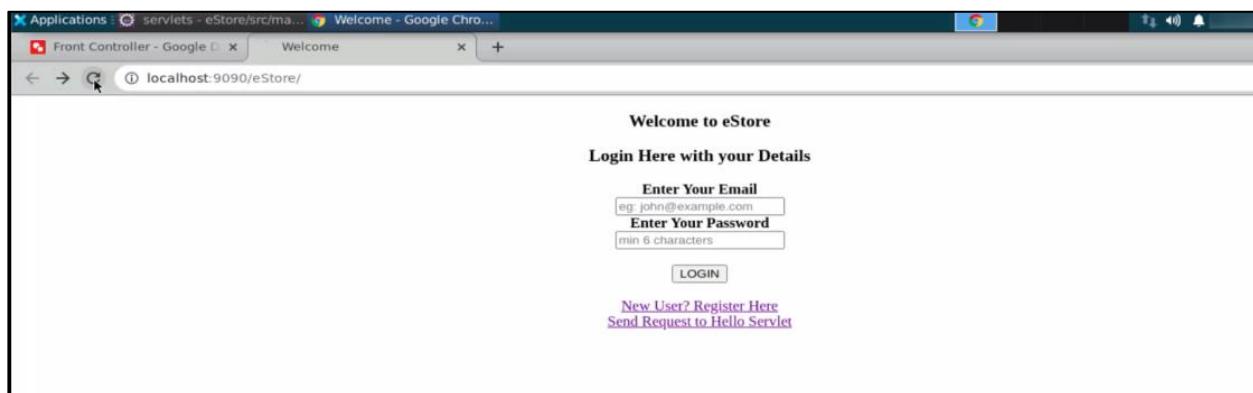
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "eStore" with packages like "JAX-WS Web Services", "src/main/java" containing "com.estore.controller" (with files "FrontController.java", "LoginServlet.java", "RegisterServlet.java"), and "Server Runtime [apache-tomcat-10.0.13]".
- Code Editor:** Displays the Java code for `FrontController.java`. The code handles requests for "login" and "register". It prints an error message if the URL is empty and then uses `dispatcher.include(request, response);` to include the response from the front controller.
- Console View:** Shows the Tomcat startup logs. Key entries include:
  - INFO: Starting service [Catalina]
  - INFO: Starting Server engine: [Apache Tomcat/10.0.13]
  - INFO: Starting ProtocolHandler ["http-nio-9090"]
  - INFO: Server startup in [369] milliseconds
  - [RegisterServlet] User Details: Dave dave@example.com dave@123
  - [RegisterServlet] User Details: Dave dave@example.com dave@123
  - INFO: Reloading Context with name [/eStore] has started
  - INFO: Reloading Context with name [/eStore] is completed
  - [RegisterServlet] User Details: Dave dave@example.com dave@123

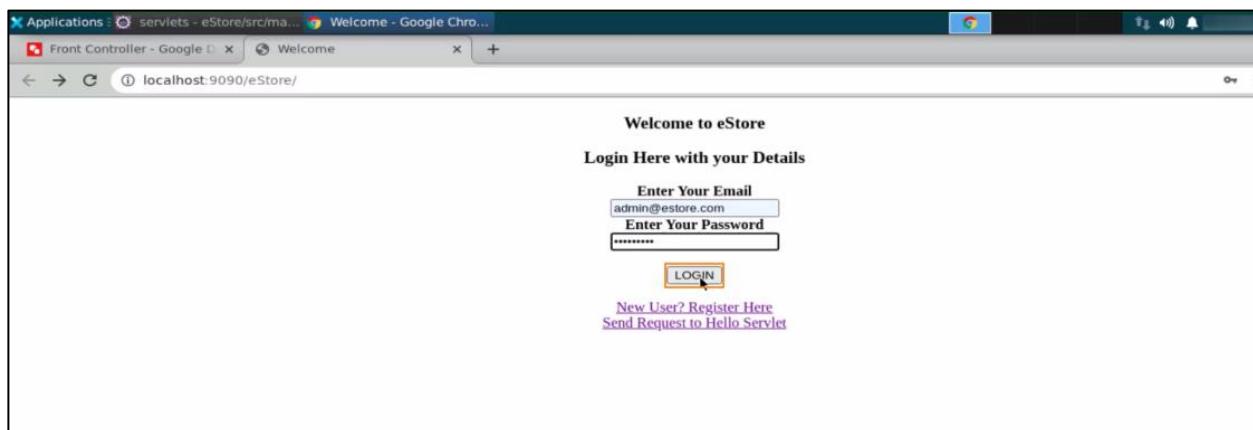
#### 4.13 Save and run the code. Right-click on the project, select **Run As**, and select **Run on Server**



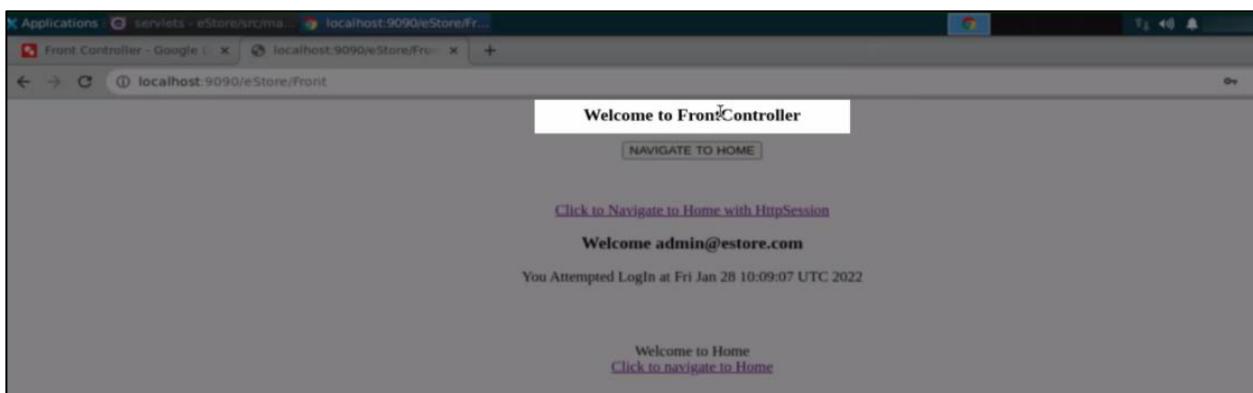
#### 4.14 Return to the browser and refresh the screen



#### 4.15 Enter the email ID and password, and click on the **LOGIN** button



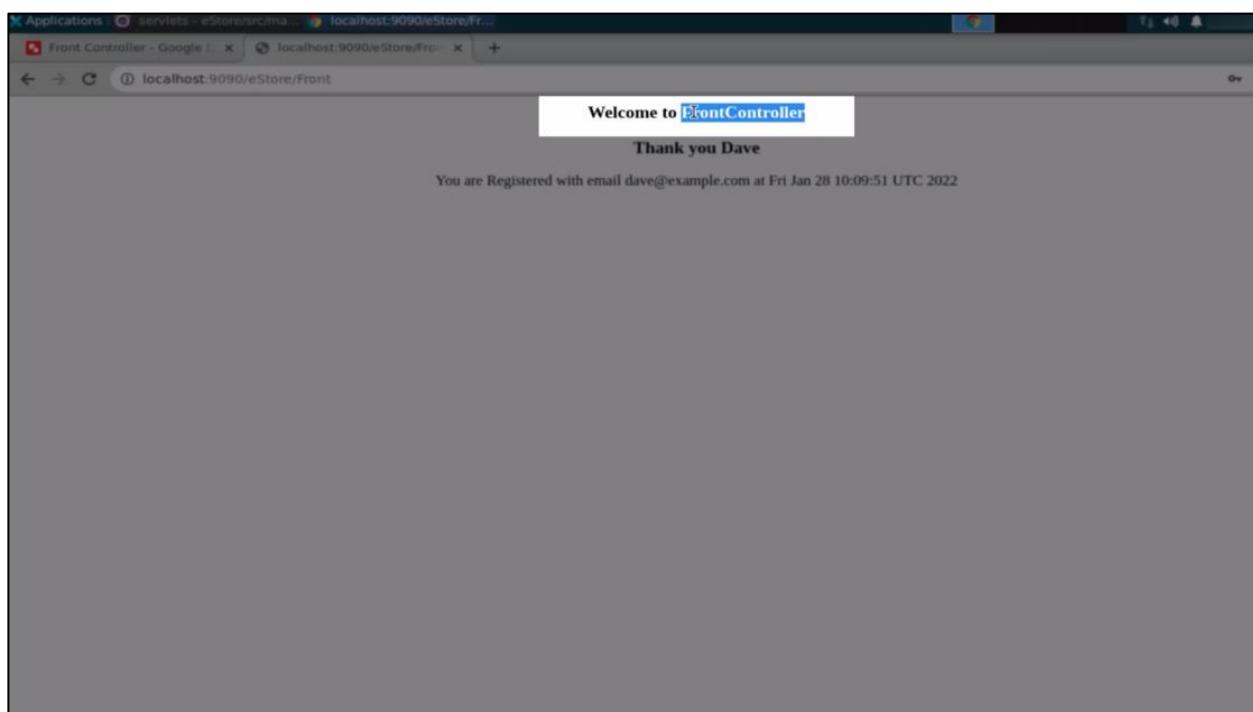
You can see the response as **Welcome to FrontController**. This means that the response is coming from the **FrontController**, but it is a perfect abstraction for the end user.



#### 4.16 Return and click on New User? Register User



#### 4.17 Enter the details and click on Register



You can see the output as **Welcome to FrontController**.

With these steps, you have successfully created a FrontController design pattern and sent a request to the FrontController servlet.