**PHASE 2**

**1. DIFFERENCE BETWEEN GET VS POST HANDLER**

**getpost.html**

**<?xml version="1.0" encoding="UTF-8"?>**

**<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" id="WebApp\_ID" version="4.0">**

**<display-name>LoginLogout</display-name>**

**<welcome-file-list>**

**<welcome-file>index.html</welcome-file>**

**<welcome-file>index.htm</welcome-file>**

**<welcome-file>index.jsp</welcome-file>**

**<welcome-file>default.html</welcome-file>**

**<welcome-file>default.htm</welcome-file>**

**<welcome-file>default.jsp</welcome-file>**

**</welcome-file-list>**

**<servlet>**

**<servlet-name>GetHandler</servlet-name>**

**<servlet-class>GetHandler</servlet-class>**

**</servlet>**

**<servlet>**

**<servlet-name>PostHandler</servlet-name>**

**<servlet-class>PostHandler</servlet-class>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>GetHandler</servlet-name>**

**<url-pattern>/gethandler</url-pattern>**

**</servlet-mapping>**

**<servlet-mapping>**

**<servlet-name>PostHandler</servlet-name>**

**<url-pattern>/posthandler</url-pattern>**

**</servlet-mapping>**

**</web-app>**

**getpostproject.java**

**package finalProject;**

**import java.io.\*;**

**import java.io.PrintWriter;**

**import javax.servlet.ServletException;**

**import javax.servlet.annotation.WebServlet;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import javax.servlet.http.HttpSession;**

**@WebServlet("/GetHandler")**

**public class getpostproject<HttpServletRequest, HttpServletResponse> extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**public getpostproject() {**

**super();**

**}**

**protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**// TODO Auto-generated method stub**

**PrintWriter out = response.getWriter();**

**out.println("<html><body>");**

**String name = request.getParameter("name");**

**String address = request.getParameter("address");**

**out.println("Name=" + name + "<br>Address=" + address);**

**out.println("</body></html>");**

**}**

**protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**// TODO Auto-generated method stub**

**doGet(request, response);**

**}**

**}**

**posthandler.java**

**package finalProject;**

**import java.io.\*;**

**import java.io.PrintWriter;**

**import javax.servlet.ServletException;**

**import javax.servlet.annotation.WebServlet;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**public class posthandler {**

**@WebServlet("/PostHandler")**

**public class PostHandler extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**public PostHandler() {**

**super();**

**}**

**protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**}**

**protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**// TODO Auto-generated method stub**

**PrintWriter out = response.getWriter();**

**out.println("<html><body>");**

**String name = request.getParameter("name");**

**String address = request.getParameter("address");**

**out.println("Name=" + name + "<br>Address=" + address);**

**out.println("</body></html>");**

**}**

**}**

**}**

**getpost.xml**

**<?xml version="1.0" encoding="UTF-8"?>**

**<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" id="WebApp\_ID" version="4.0">**

**<display-name>LoginLogout</display-name>**

**<welcome-file-list>**

**<welcome-file>index.html</welcome-file>**

**<welcome-file>index.htm</welcome-file>**

**<welcome-file>index.jsp</welcome-file>**

**<welcome-file>default.html</welcome-file>**

**<welcome-file>default.htm</welcome-file>**

**<welcome-file>default.jsp</welcome-file>**

**</welcome-file-list>**

**<servlet>**

**<servlet-name>GetHandler</servlet-name>**

**<servlet-class>GetHandler</servlet-class>**

**</servlet>**

**<servlet>**

**<servlet-name>PostHandler</servlet-name>**

**<servlet-class>PostHandler</servlet-class>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>GetHandler</servlet-name>**

**<url-pattern>/gethandler</url-pattern>**

**</servlet-mapping>**

**<servlet-mapping>**

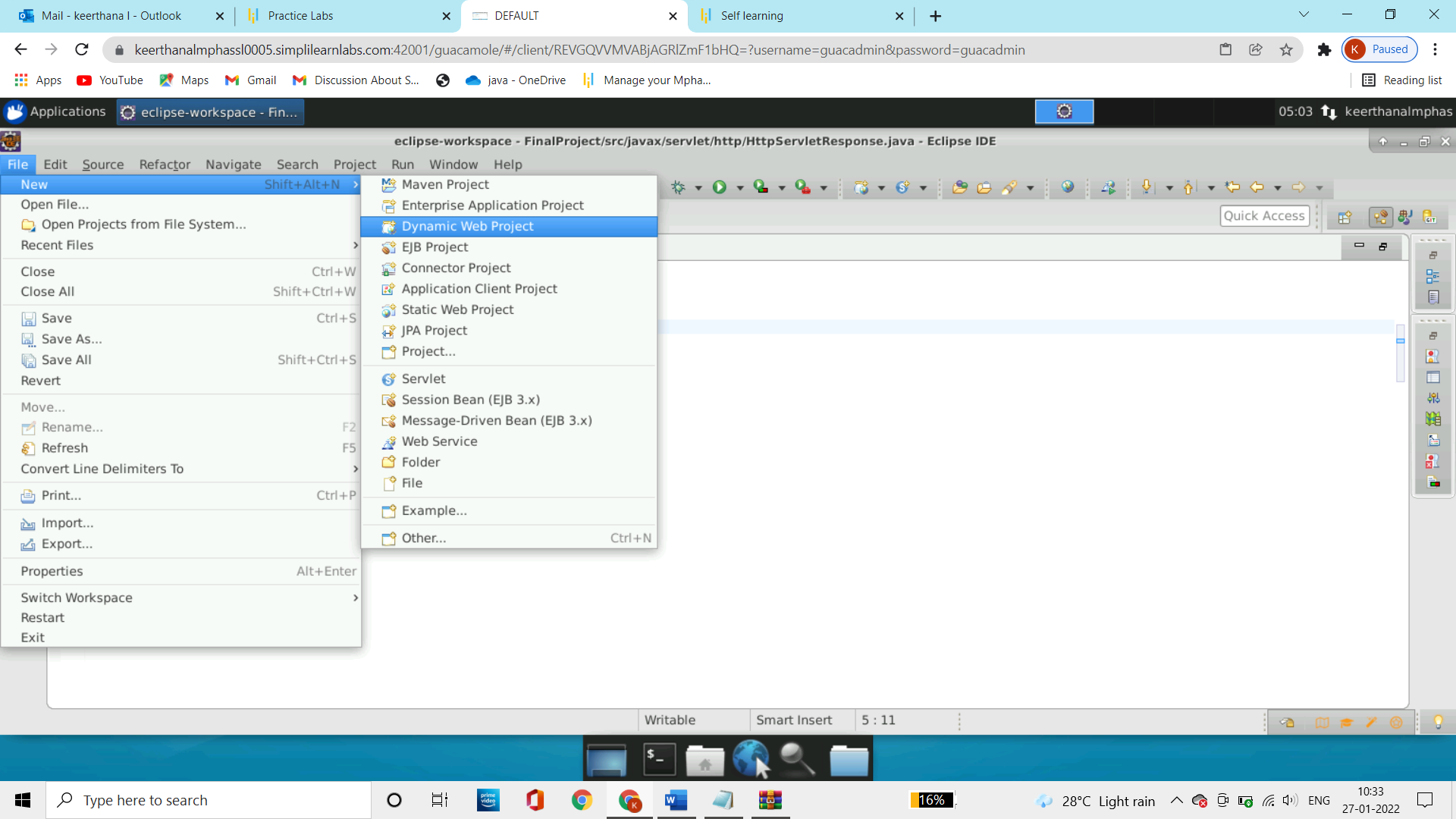
**<servlet-name>PostHandler</servlet-name>**

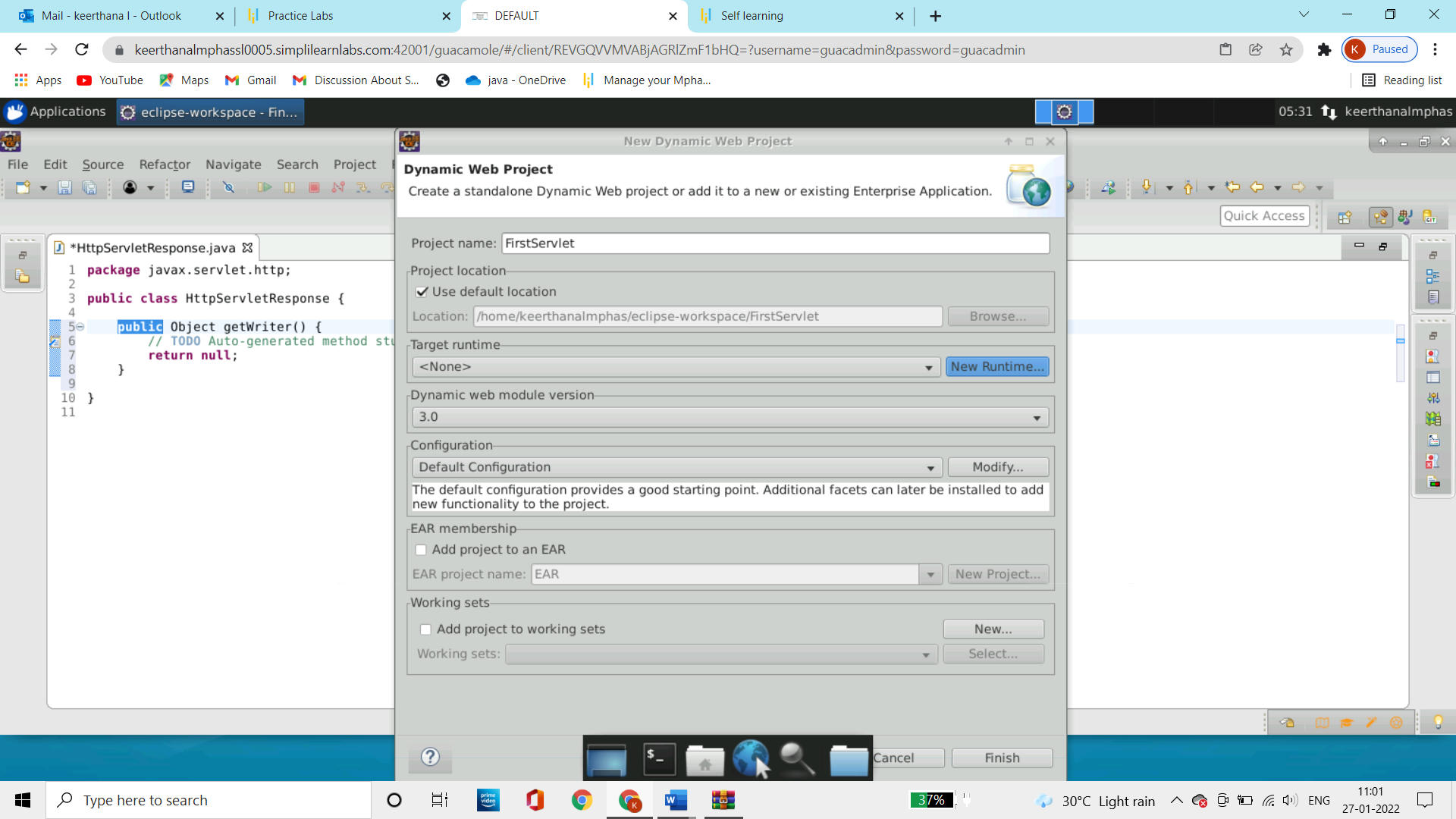
**<url-pattern>/posthandler</url-pattern>**

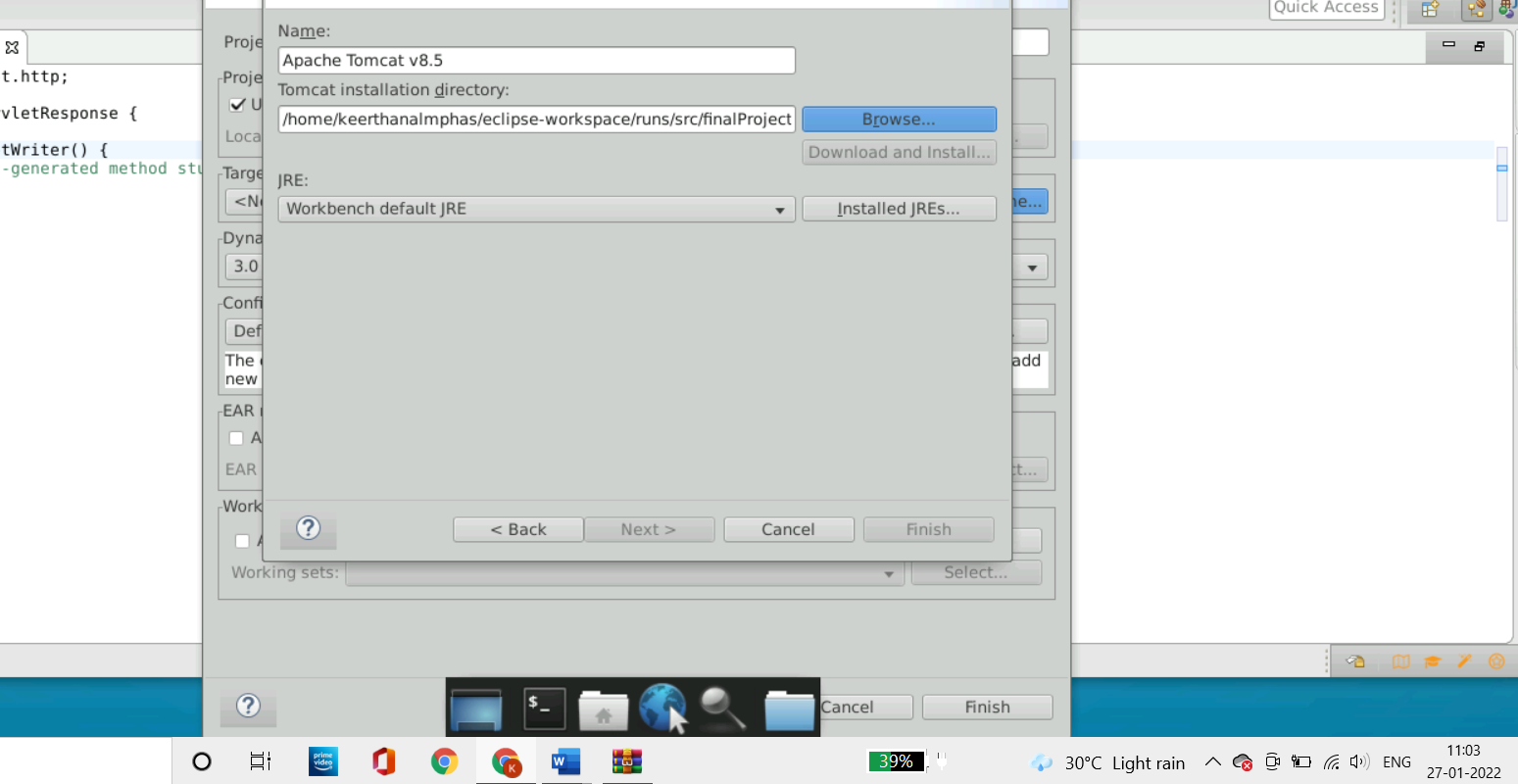
**</servlet-mapping>**

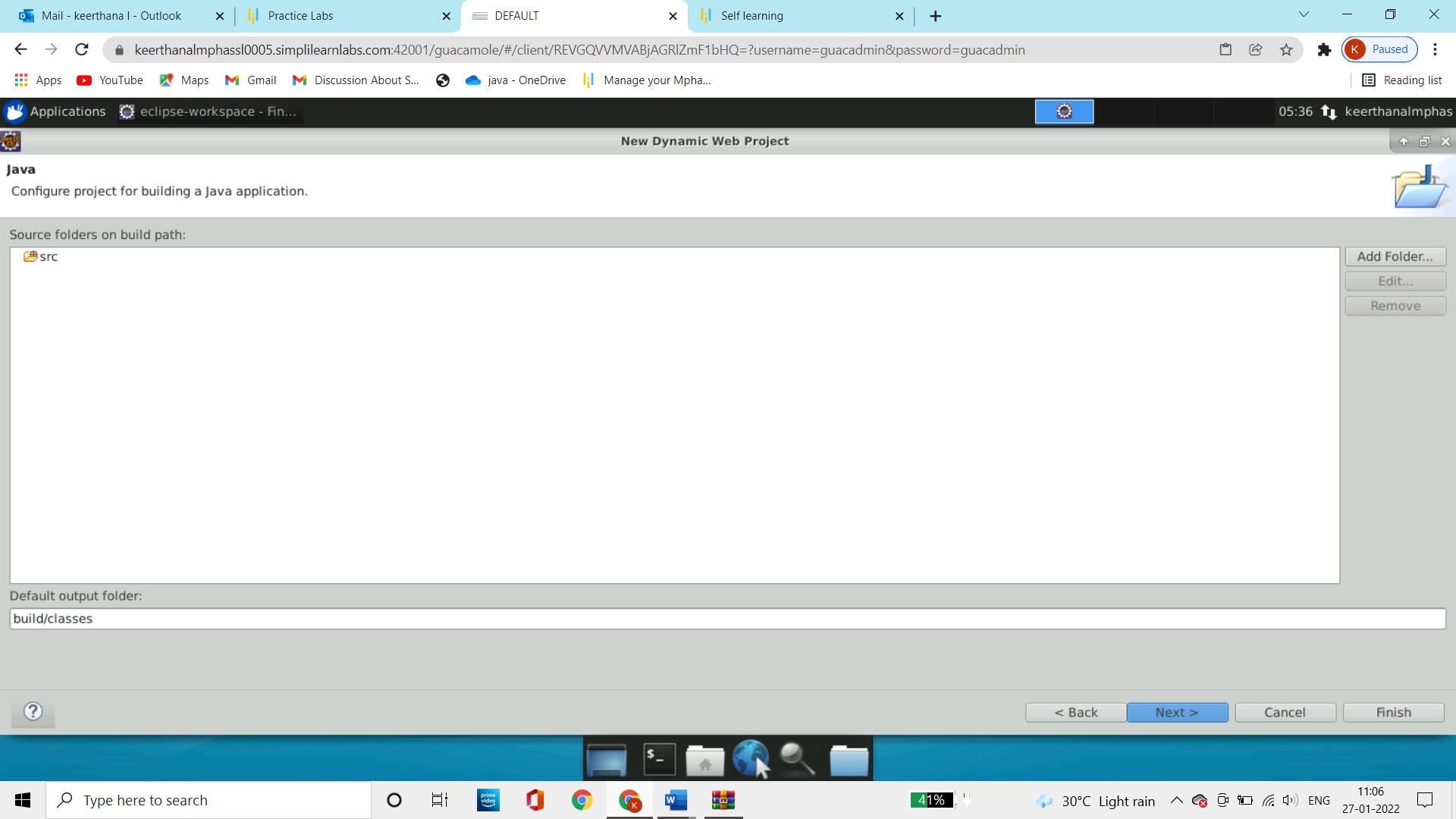
**</web-app>**

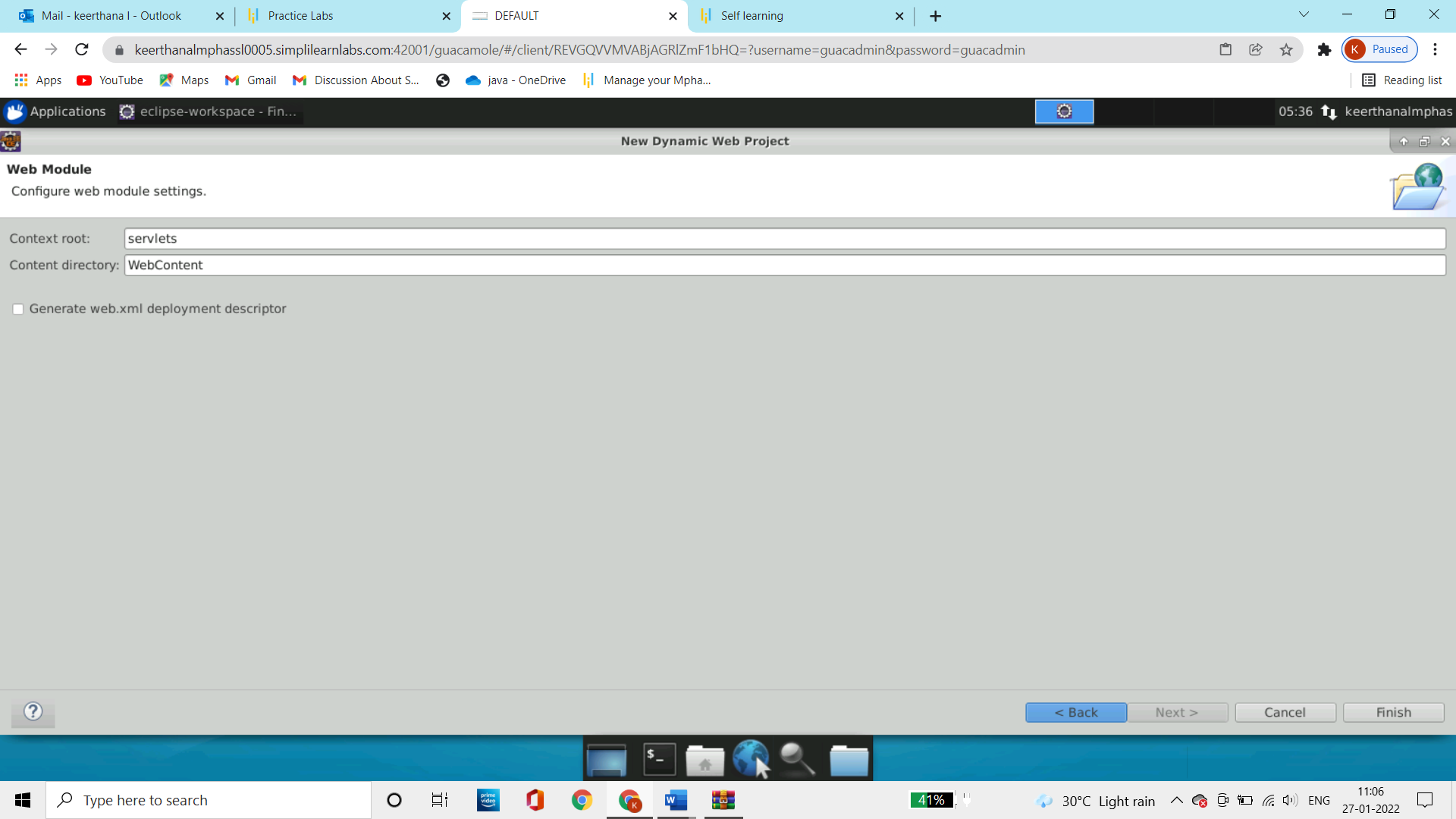
**2. CONFIGURING A SERVLET WITH ECLIPSE**

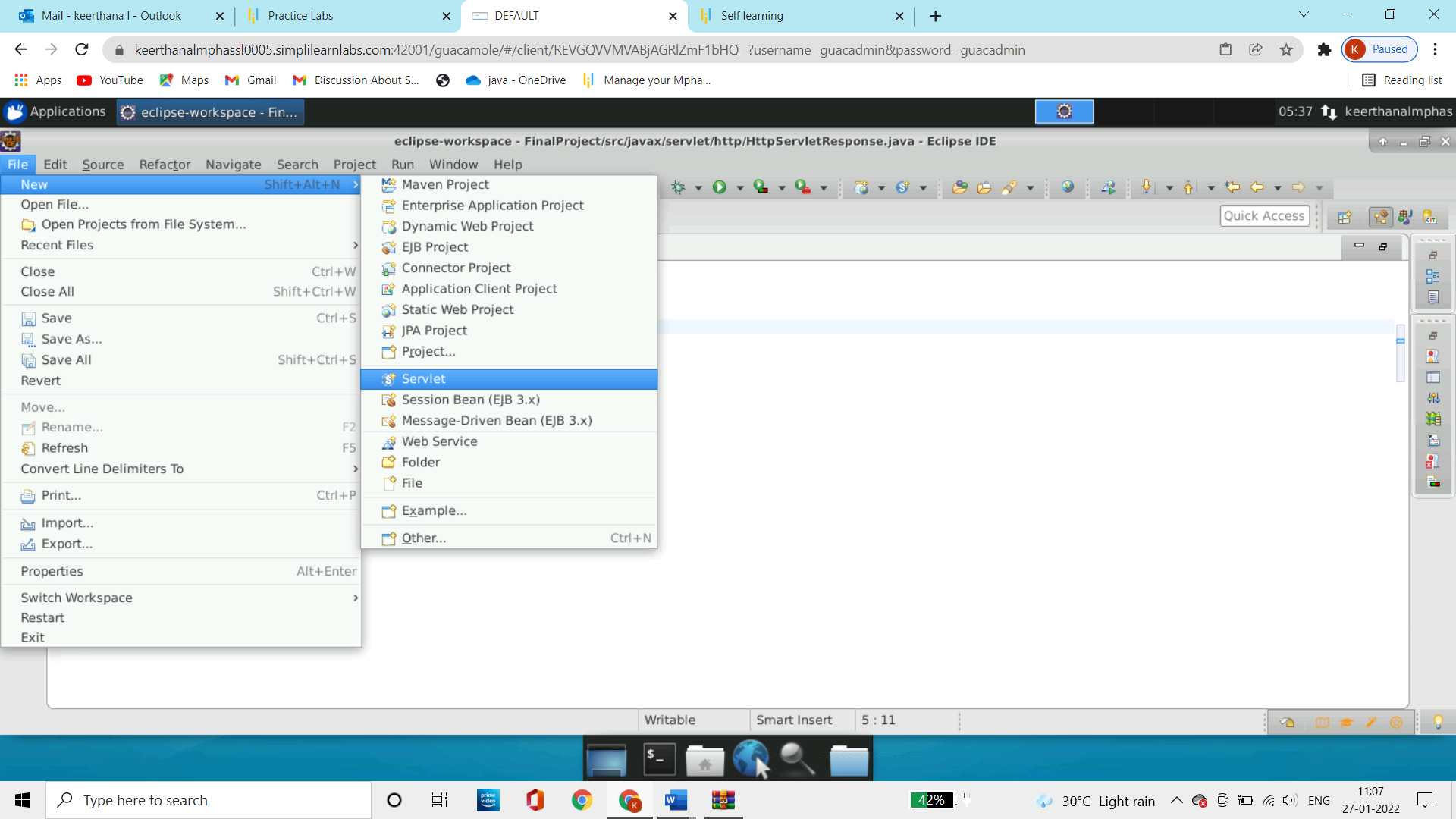


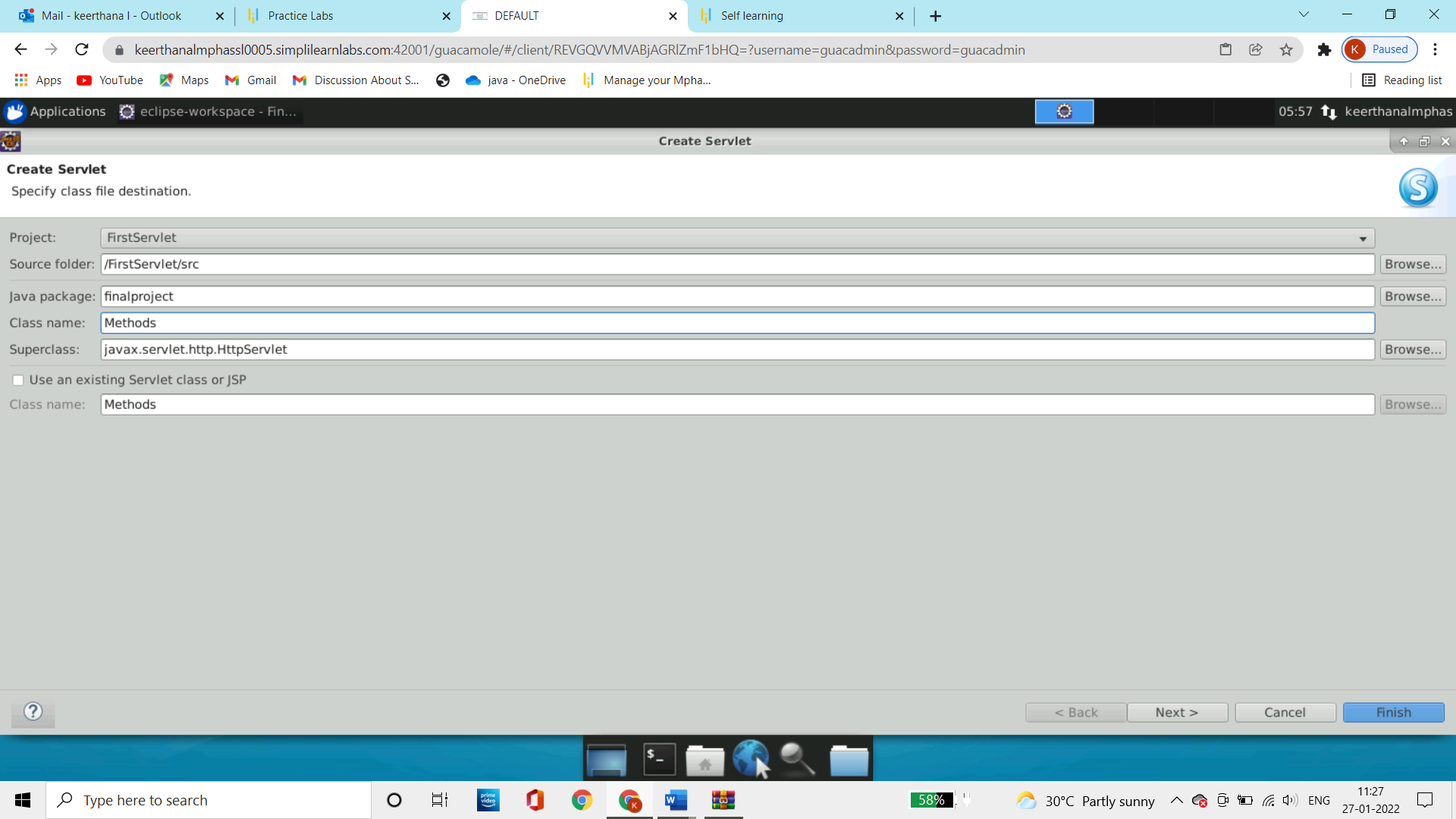


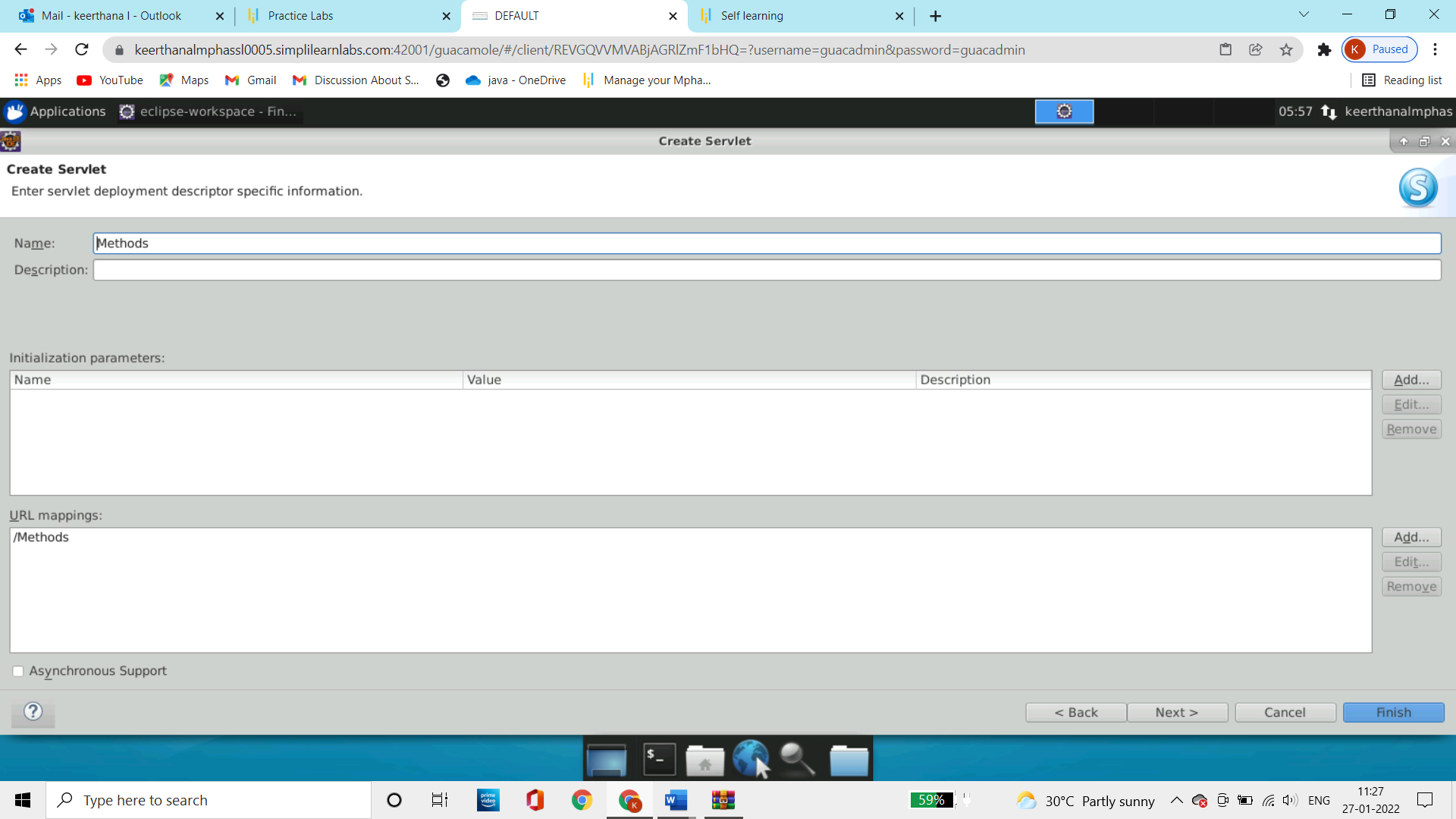


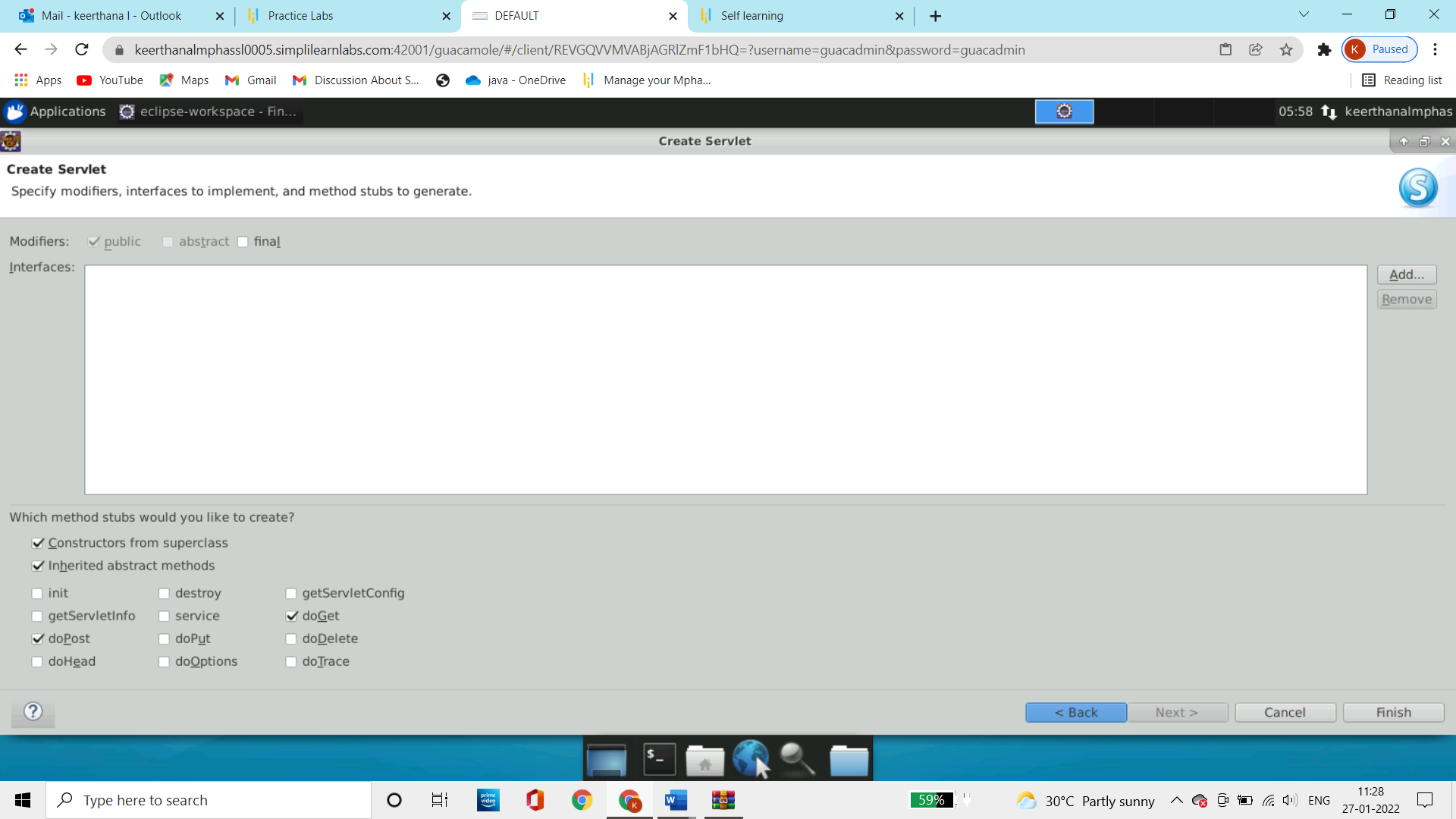


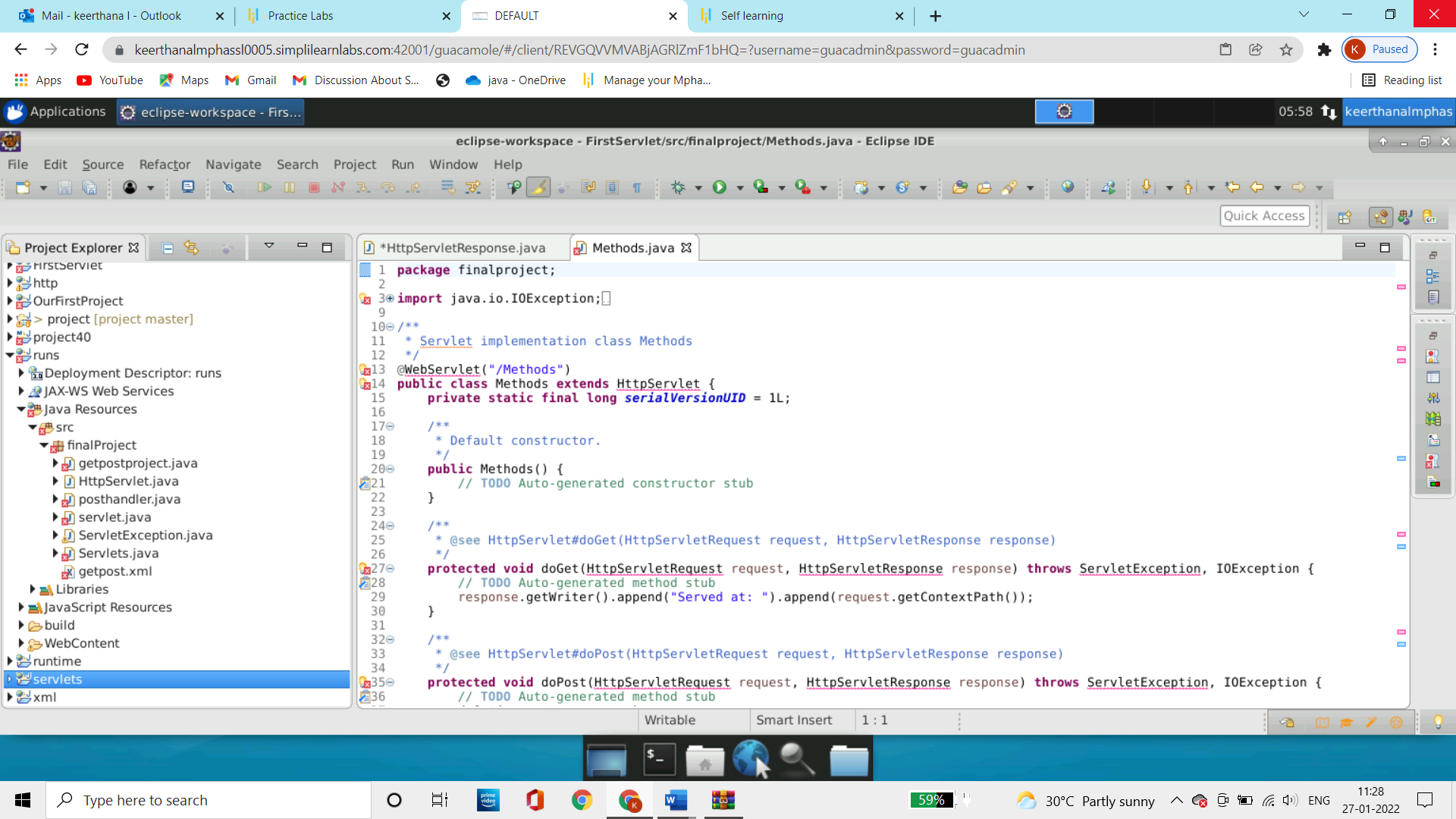


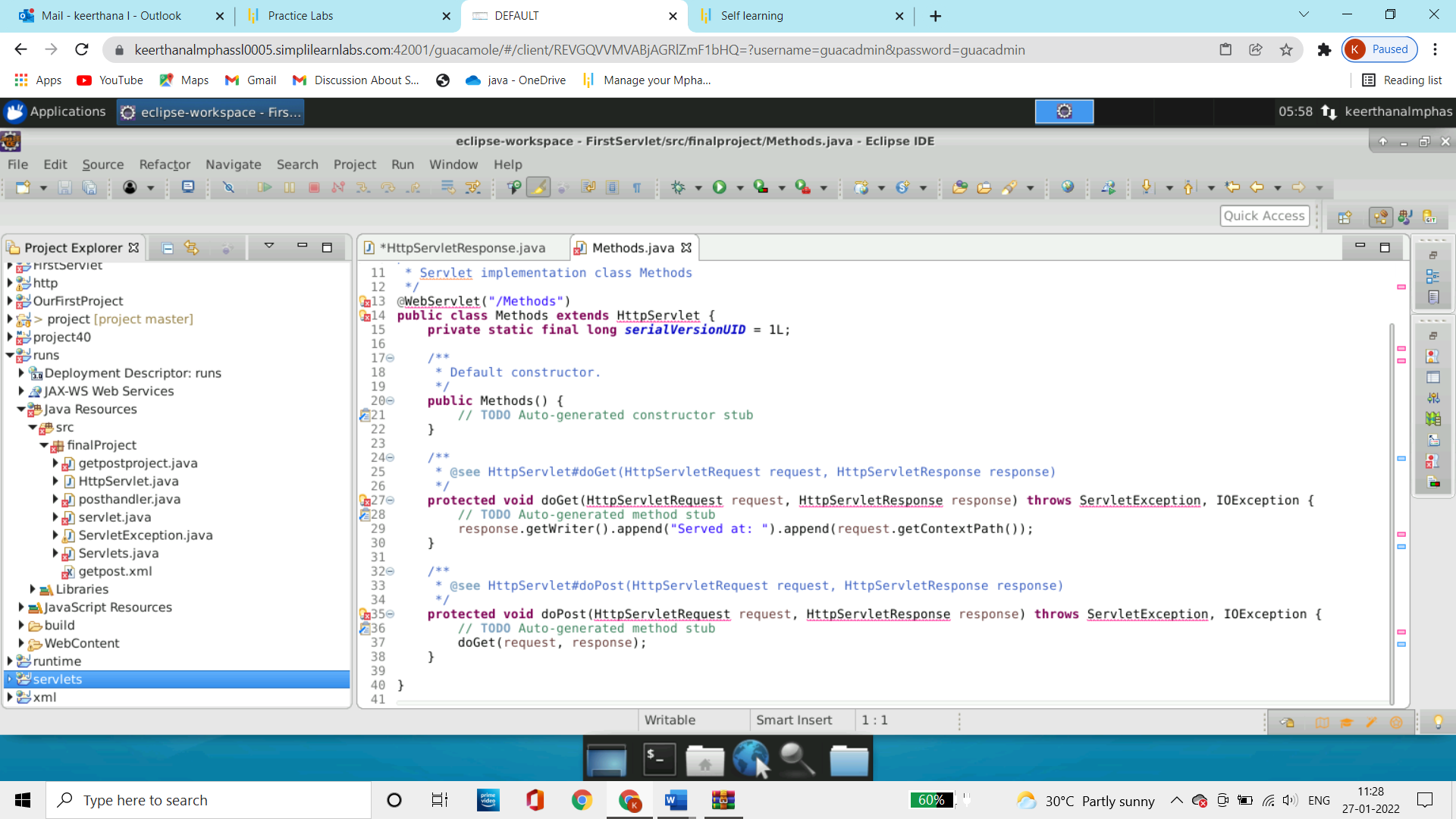












**3. GENERIC SERVLETS**

**Generic.html**

**package GenericServlets;**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>Name Servlet</title>**

**</head>**

**<body>**

**<center>**

**<form name=frmName method="GET" action="name">**

**<table width="50%" border="0">**

**<tr valign="top">**

**<td width="40%">Enter your firstname</td>**

**<td><input name="fname" id="fname" maxlength=30></td>**

**</tr>**

**<tr valign="top">**

**<td width="40%">Enter your lastname</td>**

**<td><input name="lname" id="lname" maxlength=30></td>**

**</tr>**

**<tr valign="top">**

**<td colspan=2 width="100%">**

**<button>Submit</button>**

**</td>**

**</tr>**

**</table>**

**</form>**

**</center>**

**</body>**

**</html>**

**GenericServlets.java**

**package GenericServlets;**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>Name Servlet</title>**

**</head>**

**<body>**

**<center>**

**<form name=frmName method="GET" action="name">**

**<table width="50%" border="0">**

**<tr valign="top">**

**<td width="40%">Enter your firstname</td>**

**<td><input name="fname" id="fname" maxlength=30></td>**

**</tr>**

**<tr valign="top">**

**<td width="40%">Enter your lastname</td>**

**<td><input name="lname" id="lname" maxlength=30></td>**

**</tr>**

**<tr valign="top">**

**<td colspan=2 width="100%">**

**<button>Submit</button>**

**</td>**

**</tr>**

**</table>**

**</form>**

**</center>**

**</body>**

**</html>**

**protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**// TODO Auto-generated method stub**

**doGet(request, response);**

**}**

**}**

**Generic.xml**

**<?xml version="1.0" encoding="UTF-8"?>**

**<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" id="WebApp\_ID" version="4.0">**

**<display-name>ServletDemo</display-name>**

**<welcome-file-list>**

**<welcome-file>index.html</welcome-file>**

**<welcome-file>index.htm</welcome-file>**

**<welcome-file>index.jsp</welcome-file>**

**<welcome-file>default.html</welcome-file>**

**<welcome-file>default.htm</welcome-file>**

**<welcome-file>default.jsp</welcome-file>**

**</welcome-file-list>**

**<servlet>**

**<servlet-name>NameServlet</servlet-name>**

**<servlet-class>NameServlet</servlet-class>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>NameServlet</servlet-name>**

**<url-pattern>/name</url-pattern>**

**</servlet-mapping>**

**</web-app>**

**4. CLASSES AND INTERFACES**

**Classesandinterfaces.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>Servlet</title>**

**</head>**

**<body>**

**<a href="interface">Show Servlet</a>**

**</body>**

**</html>**

**Interfaces.java**

**package ClassesandInterfaces;**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import javax.servlets.\*;**

**@WebServlet("/Interfaces")**

**public class Interfaces implements Servlet {**

**ServletConfig config=null;**

**public void init(ServletConfig config){**

**this.config=config;**

**System.out.println("Initialization complete");**

**}**

**public <ServletRequest> void service(ServletRequest req,ServletResponse res)**

**throws IOException,ServletException{**

**res.setContentType("text/html");**

**PrintWriter pwriter=res.getWriter();**

**pwriter.print("<html>");**

**pwriter.print("<body>");**

**pwriter.print("In the service() method<br>");**

**pwriter.print("</body>");**

**pwriter.print("</html>");**

**}**

**public void destroy(){**

**System.out.println("In destroy() method");**

**}**

**public ServletConfig getServletConfig(){**

**return config;**

**}**

**public String getServletInfo(){**

**return "This is a sample servlet info";**

**}**

**}**

**}**

**Classes.xml**

**<?xml version="1.0" encoding="UTF-8"?>**

**<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" id="WebApp\_ID" version="4.0">**

**<display-name>Servlet Interface</display-name>**

**<welcome-file-list>**

**<welcome-file>index.html</welcome-file>**

**<welcome-file>index.htm</welcome-file>**

**<welcome-file>index.jsp</welcome-file>**

**<welcome-file>default.html</welcome-file>**

**<welcome-file>default.htm</welcome-file>**

**<welcome-file>default.jsp</welcome-file>**

**</welcome-file-list>**

**<servlet>**

**<servlet-name>InterfaceDemo</servlet-name>**

**<servlet-class>InterfaceDemo</servlet-class>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>InterfaceDemo</servlet-name>**

**<url-pattern>/interface</url-pattern>**

**</servlet-mapping>**

**</web-app>**

**5. Servlet Filters**

**Filters.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>servlet filter</title>**

**</head>**

**<body>**

**<a href="servlet1">click here</a>**

**</body>**

**</html>**

**MyFilter.java**

**package ServerFilters;**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import java.nio.file.DirectoryStream.Filter;**

**import org.omg.CORBA.ServerRequest;**

**public abstract class MyFilter<FilterConfig> implements Filter{**

**public void init(FilterConfig arg0) throws ServletException {}**

**public void doFilter(ServerRequest req, ServletResponse resp,**

**Filter chain) throws IOException, ServletException {**

**PrintWriter out=resp.getWriter();**

**out.print("filter is invoked before");**

**((Object) chain).doFilter(req, resp);//sends request to next resource**

**out.print("filter is invoked after");**

**}**

**public void destroy() {}**

**}**

**Servletsh.java**

**package ServerFilters;**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import javax.servlet.ServletException;**

**public class Servletsh<HttpServletRequest, HttpServletResponse> {**

**public class HelloServlet extends HttpServlet {**

**public void doGet(HttpServletRequest request, HttpServletResponse response)**

**throws ServletException, IOException {**

**response.setContentType("text/html");**

**PrintWriter out = ((Object) response).getWriter();**

**out.print("<br>welcome to servlet<br>");**

**}**

**}**

**}**

**Filters.xml**

**<?xml version="1.0" encoding="UTF-8"?>**

**<web-app>**

**<servlet>**

**<servlet-name>s1</servlet-name>**

**<servlet-class>HelloServlet</servlet-class>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>s1</servlet-name>**

**<url-pattern>/servlet1</url-pattern>**

**</servlet-mapping>**

**<filter>**

**<filter-name>f1</filter-name>**

**<filter-class>MyFilter</filter-class>**

**</filter>**

**<filter-mapping>**

**<filter-name>f1</filter-name>**

**<url-pattern>/servlet1</url-pattern>**

**</filter-mapping>**

**</web-app>**

**6. Using Cookies**

**Cookies.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>Cookies</title>**

**</head>**

**<body>**

**<a href="login?userid=admin">Dashboard with cookie login</a><br>**

**<a href="dashboard">Dashboard without cookie login</a>**

**</body>**

**</html>**

**Cookies.java**

**package UsingCookies;**

**import java.io.PrintWriter;**

**import javax.servlet.\*;**

**import javax.servlet.http.\*;**

**public class Cookies extends HttpServlet {**

**public <HttpServletRequest, HttpServletResponse> void doPost(HttpServletRequest request, HttpServletResponse response){**

**try{**

**response.setContentType("text/html");**

**PrintWriter out = ((Object) response).getWriter();**

**String empName=request.getParameter("empName");**

**out.print("Hello "+empName);**

**Cookie c=new Cookie("employee",empName);**

**response.addCookie(c);**

**out.close();**

**}catch(Exception e){**

**System.out.println(e.getMessage());**

**}**

**}**

**}**

**Dashboard.java**

**package UsingCookies;**

**import java.io.IOException;**

**import java.io.\*;**

**import javax.servlet.\*;**

**import javax.servlet.annotation.\*;**

**import javax.servlet.http.\*;**

**@WebServlet("/Dashboard")**

**public class Dashboard<HttpServletResponse> extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**public Dashboard() {**

**super();**

**}**

**protected <HttpServletResponse> void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**// TODO Auto-generated method stub**

**Cookie cookie = null;**

**Cookie[] cookies = null;**

**boolean found = false;**

**PrintWriter out = ((Object) response).getWriter();**

**out.println("<html><body>");**

**cookies = request.getCookies();**

**if (cookies != null) {**

**for (int i = 0; i < cookies.length; i++) {**

**cookie = cookies[i];**

**if (cookie.getName().contentEquals("userid") && cookie.getValue() != null) {**

**out.println("UserId read from cookie:" + cookie.getValue() + "<br>");**

**found = true;**

**}**

**}**

**}**

**if (!found) {**

**out.println("No UserId was found in cookie.<br>");**

**}**

**out.println("</body></html>");**

**}**

**protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**// TODO Auto-generated method stub**

**doGet(request, response);**

**}**

**}**

**Cookies.xml**

**<?xml version="1.0" encoding="UTF-8"?>**

**<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd" id="WebApp\_ID" version="4.0">**

**<display-name>CookieTrackingDemo</display-name>**

**<welcome-file-list>**

**<welcome-file>index.html</welcome-file>**

**<welcome-file>index.htm</welcome-file>**

**<welcome-file>index.jsp</welcome-file>**

**<welcome-file>default.html</welcome-file>**

**<welcome-file>default.htm</welcome-file>**

**<welcome-file>default.jsp</welcome-file>**

**</welcome-file-list>**

**<servlet>**

**<servlet-name>LoginServlet</servlet-name>**

**<servlet-class>LoginServlet</servlet-class>**

**</servlet>**

**<servlet>**

**<servlet-name>Dashboard</servlet-name>**

**<servlet-class>Dashboard</servlet-class>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>Dashboard</servlet-name>**

**<url-pattern>/dashboard</url-pattern>**

**</servlet-mapping>**

**<servlet-mapping>**

**<servlet-name>LoginServlet</servlet-name>**

**<url-pattern>/login</url-pattern>**

**</servlet-mapping>**

**</web-app>**

**7. Session tracking using URL rewrite**

**URL.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>Tracking URL</title>**

**</head>**

**<body>**

**<form action="servlet1">**

**Name:<input type="text" name="userName"/><br/>**

**<input type="submit" value="go"/>**

**</form>**

**</body>**

**</html>**

**Servlet1.java**

**import java.io.\*;**

**import javax.servlet.\*;**

**import javax.servlet.http.\*;**

**public class FirstServlet extends HttpServlet {**

**public void doGet(HttpServletRequest request, HttpServletResponse response){**

**try{**

**response.setContentType("text/html");**

**PrintWriter out = response.getWriter();**

**String n=request.getParameter("userName");**

**out.print("Welcome "+n);**

**//appending the username in the query string**

**out.print("<a href='servlet2?uname="+n+"'>visit</a>");**

**out.close();**

**}catch(Exception e){System.out.println(e);}**

**}**

**Servlet2.java**

**package TrackingURL;**

**import java.io.IOException;**

**import java.io.\*;**

**import javax.servlet.\*;**

**import javax.servlet.http.\*;**

**@WebServlet("/Servlet2")**

**public class Servlet2 extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**public void doGet(HttpServletRequest request, HttpServletResponse response)**

**try{**

**response.setContentType("text/html");**

**PrintWriter out = response.getWriter();**

**//getting value from the query string**

**String n=request.getParameter("uname");**

**out.print("Hello "+n);**

**out.close();**

**}**

**catch(Exception e){System.out.println(e);}**

**}**

**}**

**url.xml**

**<?xml version="1.0" encoding="UTF-8"?>**

**<web-app>**

**<servlet>**

**<servlet-name>s1</servlet-name>**

**<servlet-class>FirstServlet</servlet-class>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>s1</servlet-name>**

**<url-pattern>/servlet1</url-pattern>**

**</servlet-mapping>**

**<servlet>**

**<servlet-name>s2</servlet-name>**

**<servlet-class>SecondServlet</servlet-class>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>s2</servlet-name>**

**<url-pattern>/servlet2</url-pattern>**

**</servlet-mapping>**

**</web-app>**

**8. Session tracking using Hidden Form Fields**

**Hidden.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title> Hidden Form Fields</title>**

**</head>**

**<body>**

**<form action="servlet1">**

**Name:<input type="text" name="userName"/><br/>**

**<input type="submit" value="go"/>**

**</form>**

**</body>**

**</html>**

**Servlet1.java**

**package** HiddenFormFields;

**import** java.io.IOException;

**import** java.net.http.HttpRequest;

**import** java.net.http.HttpResponse;

**import** java.io.\*;

**import** javax.servlet.\*;

**import** javax.servlet.http.\*;

**public** **class** Servlet1 **extends** HttpServlet {

**public** **void** doGet(HttpRequest request, HttpResponse response){

**try**{

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String n=request.getParameter("userName");

out.print("Welcome "+n);

//creating form that have invisible textfield

out.print("<form action='servlet2'>");

out.print("<input type='hidden' name='uname' value='"+n+"'>");

out.print("<input type='submit' value='go'>");

out.print("</form>");

out.close();

}**catch**(Exception e){System.***out***.println(e);}

}

}

**Servlet2**

**package** HiddenFormFields;

**import** java.io.IOException;

**import** java.net.http.HttpRequest;

**import** java.net.http.HttpResponse;

**import** java.io.\*;

**import** javax.servlet.\*;

**import** javax.servlet.http.\*;

**public** **class** Servlet2 **extends** HttpServlet {

**public** **void** doGet(HttpRequest request, HttpResponse response)

**try**{

response.setContentType("text/html");

PrintWriter out = response.getWriter();

//Getting the value from the hidden field

String n=request.getParameter("uname");

out.print("Hello "+n);

out.close();

}

**catch**(Exception e){

System.out.println(e);

}

**Hidden.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app>

<servlet>

<servlet-name>s1</servlet-name>

<servlet-class>FirstServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>s1</servlet-name>

<url-pattern>/servlet1</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>s2</servlet-name>

<servlet-class>SecondServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>s2</servlet-name>

<url-pattern>/servlet2</url-pattern>

</servlet-mapping>

</web-app>

9. Session using Http Session

HttpSession.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>HTTPSession Demo</title>

</head>

<body>

<a href=*"login?userid=admin"*>Dashboard with Session based login</a><br>

<a href=*"dashboard"*>Dashboard without Session based login</a>

</body>

</html>

Servlet1.java

**package** TrackingHttpSession;

**import** java.io.IOException;

**import** java.io.\*;

**import** javax.servlet.\*;

**import** javax.servlet.annotation.\*;

**import** javax.servlet.http.\*;

@WebServlet("/LoginServlet")

**public** **class** Servlet1 **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** Servlet1() {

**super**();

}

**protected** **void** doGet(HiddenFormFields.HttpServlet request, HiddenFormFields.HttpServlet response) **throws** ServletException, IOException {

// **TODO** Auto-generated method stub

String userId = request.getParameter("userid");

HttpSession session=request.getSession();

session.setAttribute("userid", userId);

response.sendRedirect("dashboard");

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request, response);

}

}

Servlet2.java

**package** TrackingHttpSession;

**import** java.io.IOException;

**import** java.io.\*;

**import** javax.servlet.\*;

**import** javax.servlet.annotation.\*;

**import** javax.servlet.http.\*;

@WebServlet("/Servlet2")

**public** **class** Servlet2<HttpSession> **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** Servlet2() {

**super**();

}

**protected** **void** doGet(HttpServlet request, HttpServlet response) **throws** ServletException, IOException {

PrintWriter out = response.getWriter();

out.println("<html><body>");

HttpSession session=request.getSession(**false**);

String userId = **null**;

**if** (session.getAttribute("userid") != **null**)

userId =(String)session.getAttribute("userid");

**if** (userId == **null** ) {

out.println("No UserId was found in session.<br>");

} **else** {

out.println("UserId obtained from session :" + userId + "<br>");

}

out.println("</body></html>");

}

**protected** <HttpServletRequest> **void** doPost(HttpServletRequest request, HttpServlet response) **throws** ServletException, IOException {

doGet(request, response);

}

}

Tracking.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>HTTPSessionDemo</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>Servlet1</servlet-name>

<servlet-class>Servlet1</servlet-class>

</servlet>

<servlet>

<servlet-name>Servlet2</servlet-name>

<servlet-class>Servlet2</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Servlet2</servlet-name>

<url-pattern>Servlet2</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Servlet1</servlet-name>

<url-pattern>/login</url-pattern>

</servlet-mapping>

</web-app>

</servlet-mapping>

<servlet-mapping>

<servlet-name>Servlet1</servlet-name>

<url-pattern>/login</url-pattern>

</servlet-mapping>

</web-app>

10.Session login and logout

Index.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Servlet Login Example</title>

</head>

<body>

<h1>Login App using HttpSession</h1>

<a href=*"login.html"*>Login</a>|

<a href=*"LogoutServlet"*>Logout</a>|

<a href=*"ProfileServlet"*>Profile</a>

</body>

</html>

Link.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Login Server</title>

</head>

<body>

<a href=*"login.html"*>Login</a> |

<a href=*"LogoutServlet"*>Logout</a> |

<a href=*"ProfileServlet"*>Profile</a>

</body>

</html>

Login.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<form action=*"LoginServlet"* method=*"post"*>

Name:<input type=*"text"* name=*"name"*><br>

Password:<input type=*"password"* name=*"password"*><br>

<input type=*"submit"* value=*"login"*>

</form>

</body>

</html>

LoginServlet.java

**package** LoginLogout;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

**public** **class** LoginServlet **extends** HttpServlet {

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

PrintWriter out=response.getWriter();

request.getRequestDispatcher("link.html").include(request, response);

String name=request.getParameter("name");

String password=request.getParameter("password");

**if**(password.equals("admin123")){

out.print("Welcome, "+name);

HttpSession session=request.getSession();

session.setAttribute("name",name);

}

**else**{

out.print("Sorry, username or password error!");

request.getRequestDispatcher("login.html").include(request, response);

}

out.close();

}

}

LogoutServlet.java

**package** LoginLogout;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** java.io.PrintWriter;

**public** **class** LogoutServlet **extends** HttpServlet {

**protected** <HttpServletRequest> **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

PrintWriter out=response.getWriter();

request.getRequestDispatcher("link.html").include(request, response);

HttpSession session=request.getSession();

session.invalidate();

out.print("You are successfully logged out!");

out.close();

}

}

ProfileServlet.java

**package** LoginLogout;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

**public** **class** ProfileServlet **extends** HttpServlet {

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

PrintWriter out=response.getWriter();

request.getRequestDispatcher("link.html").include(request, response);

HttpSession session=request.getSession(**false**);

**if**(session!=**null**){

String name=(String)session.getAttribute("name");

out.print("Hello, "+name+" Welcome to Profile");

}

**else**{

out.print("Please login first");

request.getRequestDispatcher("login.html").include(request, response);

}

out.close();

}

}

Web.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee*

*http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* id=*"WebApp\_ID"* version=*"2.5"*>

<servlet>

<description></description>

<display-name>LoginServlet</display-name>

<servlet-name>LoginServlet</servlet-name>

<servlet-class>LoginServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>LoginServlet</servlet-name>

<url-pattern>/LoginServlet</url-pattern>

</servlet-mapping>

<servlet>

<description></description>

<display-name>ProfileServlet</display-name>

<servlet-name>ProfileServlet</servlet-name>

<servlet-class>ProfileServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ProfileServlet</servlet-name>

<url-pattern>/ProfileServlet</url-pattern>

</servlet-mapping>

<servlet>

<description></description>

<display-name>LogoutServlet</display-name>

<servlet-name>LogoutServlet</servlet-name>

<servlet-class>LogoutServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>LogoutServlet</servlet-name>

<url-pattern>/LogoutServlet</url-pattern>

</servlet-mapping>

</web-app>

11.Validation of user login

Validate.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<form action=*"servlet1"* method=*"post"*>

Name:<input type=*"text"* name=*"username"*/><br/><br/>

Password:<input type=*"password"* name=*"userpass"*/><br/><br/>

<input type=*"submit"* value=*"login"*/>

</form>

</body>

</html>

ValidationUser

**package** Validation;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.RequestDispatcher;

**import** javax.servlet.ServletException;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**public** **class** ValidationUser{

**public** **class** FirstServlet **extends** HttpServlet {

**public** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String n=request.getParameter("username");

String p=request.getParameter("userpass");

**if**(LoginDao.validate(n, p)){

RequestDispatcher rd=request.getRequestDispatcher("servlet2");

rd.forward(request,response);

}

}

**else**{

out.print("Sorry username or password error");

RequestDispatcher rd=request.getRequestDispatcher("index.html");

rd.include(request,response);

}

out.close();

}

}

}

Logindemo.java

**package** Validation;

**import** java.io.IOException;

**import** java.sql.\*;

**public** **class** LoginDemo {

**public** **static** **boolean** validate(String name,String pass){

**boolean** status=**false**;

**try**{

Class.*forName*("oracle.jdbc.driver.OracleDriver");

Connection con=DriverManager.*getConnection*(

"jdbc:oracle:thin:@localhost:1521:xe","system","oracle");

PreparedStatement ps=con.prepareStatement(

"select \* from userreg where name=? and pass=?");

ps.setString(1,name);

ps.setString(2,pass);

ResultSet rs=ps.executeQuery();

status=rs.next();

}**catch**(Exception e){System.***out***.println(e);}

**return** status;

}

}

LoginServlet.java

**package** Validation;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**public** **class** LoginServlet **extends** HttpServlet {

**public** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String n=request.getParameter("username");

out.print("Welcome "+n);

out.close();

}

}

12. Setup JDBC environment

Jdbc.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>JDBC Setup</title>

</head>

<body>

<a href=*"init"*>Initialize JDBC</a><br>

</body>

</html>

DBconnection.html

package JDBCsetup;

package com.ecommerce;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.io.IOException;

public class DBconnection {

private Connection connection;

public DBconnection(String dbURL, String user, String pwd) throws ClassNotFoundException, SQLException{

Class.forName("com.mysql.jdbc.Driver");

this.connection = DriverManager.getConnection(dbURL, user, pwd);

}

public Connection getConnection(){

return this.connection;

}

public void closeConnection() throws SQLException {

if (this.connection != null)

this.connection.close();

}

}

DemoJDBC.java

**package** JDBCsetup;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.PrintWriter;

**import** java.math.BigDecimal;

**import** java.sql.CallableStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.Properties;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** com.ecommerce.DBConnection;

@WebServlet("/DemoJDBC")

**public** **class** DemoJDBC **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** DemoJDBC() {

**super**();

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

PrintWriter out = response.getWriter();

out.println("<html><body>");

InputStream in = getServletContext().getResourceAsStream("/WEB-INF/config.properties");

Properties props = **new** Properties();

props.load(in);

DBConnection conn = **new** DBConnection(props.getProperty("url"), props.getProperty("userid"), props.getProperty("password"));

out.println("DB Connection initialized.<br>");

conn.closeConnection();

out.println("DB Connection closed.<br>");

out.println("</body></html>");

conn.closeConnection();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request, response);

}

}

DemoJDBC.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>JDBCSetup</display-name>

<welcome-file-list>

<welcome-file>jdbc.html</welcome-file>

<welcome-file>jdbc.htm</welcome-file>

<welcome-file>jdbc.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>DemoJDBC</servlet-name>

<servlet-class>DemoJDBC</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>DemoJDBC</servlet-name>

<url-pattern>/init</url-pattern>

</servlet-mapping>

</web-app>

13. Setup JDBC environment

Jdbc.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>JDBC Setup</title>

</head>

<body>

<a href=*"init"*>Initialize JDBC</a><br>

</body>

</html>

DBconnection.html

package JDBCsetup;

package com.ecommerce;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.io.IOException;

public class DBconnection {

private Connection connection;

public DBconnection(String dbURL, String user, String pwd) throws ClassNotFoundException, SQLException{

Class.forName("com.mysql.jdbc.Driver");

this.connection = DriverManager.getConnection(dbURL, user, pwd);

}

public Connection getConnection(){

return this.connection;

}

public void closeConnection() throws SQLException {

if (this.connection != null)

this.connection.close();

}

}

DemoJDBC.java

**package** JDBCsetup;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.PrintWriter;

**import** java.math.BigDecimal;

**import** java.sql.CallableStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.Properties;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** com.ecommerce.DBConnection;

@WebServlet("/DemoJDBC")

**public** **class** DemoJDBC **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** DemoJDBC() {

**super**();

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

PrintWriter out = response.getWriter();

out.println("<html><body>");

InputStream in = getServletContext().getResourceAsStream("/WEB-INF/config.properties");

Properties props = **new** Properties();

props.load(in);

DBConnection conn = **new** DBConnection(props.getProperty("url"), props.getProperty("userid"), props.getProperty("password"));

out.println("DB Connection initialized.<br>");

conn.closeConnection();

out.println("DB Connection closed.<br>");

out.println("</body></html>");

conn.closeConnection();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request, response);

}

}

DemoJDBC.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>JDBCSetup</display-name>

<welcome-file-list>

<welcome-file>jdbc.html</welcome-file>

<welcome-file>jdbc.htm</welcome-file>

<welcome-file>jdbc.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>DemoJDBC</servlet-name>

<servlet-class>DemoJDBC</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>DemoJDBC</servlet-name>

<url-pattern>/init</url-pattern>

</servlet-mapping>

</web-app>

14. JDBC Connections, Statements, and Result Sets

Statements.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>JDBC Statements and Resultsets</title>

</head>

<body>

<a href=*"list"*>Product Info</a><br>

</body>

Statements.java

**package** JDBCstatements;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**public** **class** statements {

**private** Connection connection;

**public** statements(String dbURL, String user, String pwd) **throws** ClassNotFoundException, SQLException{

Class.*forName*("com.mysql.jdbc.Driver");

**this**.connection = DriverManager.*getConnection*(dbURL, user, pwd);

}

**public** Connection getConnection(){

**return** **this**.connection;

}

**public** **void** closeConnection() **throws** SQLException {

**if** (**this**.connection != **null**)

**this**.connection.close();

}

}

Product.java

**package** JDBCstatements;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.PrintWriter;

**import** java.math.BigDecimal;

**import** java.sql.CallableStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.Properties;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** com.ecommerce.DBConnection;

@WebServlet("/Product")

**public** **class** Product **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** Product() {

**super**();

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

PrintWriter out = response.getWriter();

out.println("<html><body>");

InputStream in = getServletContext().getResourceAsStream("/WEB-INF/config.properties");

Properties props = **new** Properties();

props.load(in);

DBConnection conn = **new** DBConnection(props.getProperty("url"), props.getProperty("userid"), props.getProperty("password"));

Statement stmt = conn.getConnection().createStatement(ResultSet.***TYPE\_SCROLL\_INSENSITIVE***, ResultSet.***CONCUR\_READ\_ONLY***);

stmt.executeUpdate("insert into eproduct (name, price, date\_added) values ('New Product', 17800.00, now())");

ResultSet rst = stmt.executeQuery("select \* from eproduct");

**while** (rst.next()) {

out.println(rst.getInt("ID") + ", " + rst.getString("name") + "<Br>");

}

stmt.close();

out.println("</body></html>");

conn.closeConnection();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request, response);

}

}

Product.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>JDBC Statements and Resultsets</display-name>

<welcome-file-list>

<welcome-file>statements.html</welcome-file>

<welcome-file>statements.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>Product</servlet-name>

<servlet-class>Product</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Product</servlet-name>

<url-pattern>/list</url-pattern>

</servlet-mapping>

</web-app>

15. Stored procedures with exceptions

Delimiter.sql

DELIMITER $$

CREATE PROCEDURE add\_product(IN pname varchar(100), IN pprice decimal(10,2))

INSERT INTO eproduct (name, price) VALUES (pname, pprice)

$$

DELIMITER ;

Excp.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>JDBC Stored Procedures</title>

</head>

<body>

<a href=*"list"*>Product Info</a><br>

</body>

</html>

Connection.java

**package** Exceptions;

**import** java.io.IOException;

**package** com.ecommerce;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**public** **class** Connection {

**private** Connection connection;

publicConnection(String dbURL, String user, String pwd) **throws** ClassNotFoundException, SQLException{

Class.*forName*("com.mysql.jdbc.Driver");

**this**.connection = DriverManager.*getConnection*(dbURL, user, pwd);

}

**public** Connection getConnection(){

**return** **this**.connection;

}

**public** **void** closeConnection() **throws** SQLException {

**if** (**this**.connection != **null**)

**this**.connection.close();

}

}

Conn.java

**package** Exceptions;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.PrintWriter;

**import** java.math.BigDecimal;

**import** java.sql.CallableStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.Properties;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** com.ecommerce.DBConnection;

@WebServlet("/Conn")

**public** **class** Conn **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** Conn() {

**super**();

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

PrintWriter out = response.getWriter();

out.println("<html><body>");

InputStream in = getServletContext().getResourceAsStream("/WEB-INF/config.properties");

Properties props = **new** Properties();

props.load(in);

DBConnection conn = **new** DBConnection(props.getProperty("url"), props.getProperty("userid"), props.getProperty("password"));

CallableStatement stmt = conn.getConnection().prepareCall("{call add\_product(?, ?)}");

stmt.setString(1, "new product");

stmt.setBigDecimal(2, **new** BigDecimal(1900.50));

stmt.executeUpdate();

out.println("Stored procedure has been executed.<Br>");

stmt.close();

out.println("</body></html>");

conn.closeConnection();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// **TODO** Auto-generated method stub

doGet(request, response);

}

}

Db.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>JDBC Stored Procedures</display-name>

<welcome-file-list>

<welcome-file>excp.html</welcome-file>

<welcome-file>excp.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>Connection</servlet-name>

<servlet-class>Conn</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Conn</servlet-name>

<url-pattern>/list</url-pattern>

</servlet-mapping>

</web-app>

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>JDBC Stored Procedures</display-name>

<welcome-file-list>

<welcome-file>excp.html</welcome-file>

<welcome-file>excp.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>Connection</servlet-name>

<servlet-class>Conn</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Conn</servlet-name>

<url-pattern>/list</url-pattern>

</servlet-mapping>

</web-app>

16. Create , Select and drop a database

Csd.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>JDBC Datbase Operations</title>

</head>

<body>

<a href=*"dboperations"*>Database Operations</a><br>

</body>

</html>

Csddatabase.java

**package** Database;

**package** com.ecommerce;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**public** **class** csddatabase {

**private** Connection connection;

**public** csddatabase(String dbURL, String user, String pwd) **throws** ClassNotFoundException, SQLException{

Class.*forName*("com.mysql.jdbc.Driver");

**this**.connection = DriverManager.*getConnection*(dbURL, user, pwd);

}

**public** Connection getConnection(){

**return** **this**.connection;

}

**public** **void** closeConnection() **throws** SQLException {

**if** (**this**.connection != **null**)

**this**.connection.close();

}

}

Operations.java

**package** Database;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.PrintWriter;

**import** java.math.BigDecimal;

**import** java.sql.CallableStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.Properties;

@WebServlet("/Operations")

**public** **class** Operations **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** Operations() {

**super**();

}

**protected** **void** doGet(HttpServletrequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

PrintWriter out = response.getWriter();

out.println("<html><body>");

InputStream in = getServletContext().getResourceAsStream("/WEB-INF/config.properties");

Properties props = **new** Properties();

props.load(in);

DBConnection conn = **new** DBConnection(props.getProperty("url"), props.getProperty("userid"), props.getProperty("password"));

Statement stmt = conn.getConnection().createStatement();

stmt.executeUpdate("create database mydatabase");

out.println("Created database: mydatabase<br>");

stmt.executeUpdate("use mydatabase");

out.println("Selected database: mydatabase<br>");

stmt.executeUpdate("drop database mydatabase");

stmt.close();

out.println("Dropped database: mydatabase<br>");

conn.closeConnection();

out.println("</body></html>");

conn.closeConnection();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request, response);

}

}

Setup.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>JDBC DB Operations</display-name>

<welcome-file-list>

<welcome-file>csd.html</welcome-file>

<welcome-file>csd.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>csddatabase</servlet-name>

<servlet-class>Operations</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>csddatabase</servlet-name>

<url-pattern>/Operations</url-pattern>

</servlet-mapping>

</web-app>

17. Create , select and drop a database

Csd.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>JDBC Datbase Operations</title>

</head>

<body>

<a href=*"dboperations"*>Database Operations</a><br>

</body>

</html>

Csddatabase.java

**package** Database;

**package** com.ecommerce;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**public** **class** csddatabase {

**private** Connection connection;

**public** csddatabase(String dbURL, String user, String pwd) **throws** ClassNotFoundException, SQLException{

Class.*forName*("com.mysql.jdbc.Driver");

**this**.connection = DriverManager.*getConnection*(dbURL, user, pwd);

}

**public** Connection getConnection(){

**return** **this**.connection;

}

**public** **void** closeConnection() **throws** SQLException {

**if** (**this**.connection != **null**)

**this**.connection.close();

}

}

Operations.java

**package** Database;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.PrintWriter;

**import** java.math.BigDecimal;

**import** java.sql.CallableStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.Properties;

@WebServlet("/Operations")

**public** **class** Operations **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** Operations() {

**super**();

}

**protected** **void** doGet(HttpServletrequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

PrintWriter out = response.getWriter();

out.println("<html><body>");

InputStream in = getServletContext().getResourceAsStream("/WEB-INF/config.properties");

Properties props = **new** Properties();

props.load(in);

DBConnection conn = **new** DBConnection(props.getProperty("url"), props.getProperty("userid"), props.getProperty("password"));

Statement stmt = conn.getConnection().createStatement();

stmt.executeUpdate("create database mydatabase");

out.println("Created database: mydatabase<br>");

stmt.executeUpdate("use mydatabase");

out.println("Selected database: mydatabase<br>");

stmt.executeUpdate("drop database mydatabase");

stmt.close();

out.println("Dropped database: mydatabase<br>");

conn.closeConnection();

out.println("</body></html>");

conn.closeConnection();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request, response);

}

}

Setup.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>JDBC DB Operations</display-name>

<welcome-file-list>

<welcome-file>csd.html</welcome-file>

<welcome-file>csd.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>csddatabase</servlet-name>

<servlet-class>Operations</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>csddatabase</servlet-name>

<url-pattern>/Operations</url-pattern>

</servlet-mapping>

</web-app>

18. Insertion, Updation and deletion of database records

Sql Table query

CREATE TABLE eproduct (ID bigint primary key auto\_increment, name varchar(100), price decimal(10,2), date\_added timestamp default now())

INSERT INTO eproduct(name, ‘HP Laptop ABC’, 12000)

INSERT INTO eproduct(name, ‘Acer Laptop ABC’, 14000)

INSERT INTO eproduct(name, ‘Lenovo Laptop ABC’, 12000)

SELECT \* from eproduct;

Update.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>JDBC Insert, Update, Delete</title>

</head>

<body>

<a href=*"list"*>Product Info</a><br>

</body>

</html>

DBConnection.java

**package** InsertUpdateDelete;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**public** **class** DBConnection {

**private** Connection connection;

**public** DBConnection(String dbURL, String user, String pwd) **throws** ClassNotFoundException, SQLException{

Class.*forName*("com.mysql.jdbc.Driver");

**this**.connection = DriverManager.*getConnection*(dbURL, user, pwd);

}

**public** Connection getConnection(){

**return** **this**.connection;

}

**public** **void** closeConnection() **throws** SQLException {

**if** (**this**.connection != **null**)

**this**.connection.close();

}

}

Productdetails.java

**package** InsertUpdateDelete;

**import** java.io.IOException;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.io.PrintWriter;

**import** java.math.BigDecimal;

**import** java.sql.CallableStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.Properties;

@WebServlet("/Productdetails")

**public** **class** Productdetails **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** Productdetails() {

**super**();

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

PrintWriter out = response.getWriter();

out.println("<html><body>");

InputStream in = getServletContext().getResourceAsStream("/WEB-INF/config.properties");

Properties props = **new** Properties();

props.load(in);

DBConnection conn = **new** DBConnection(props.getProperty("url"), props.getProperty("userid"), props.getProperty("password"));

Statement stmt = conn.getConnection().createStatement(ResultSet.***TYPE\_SCROLL\_INSENSITIVE***, ResultSet.***CONCUR\_READ\_ONLY***);

stmt.executeUpdate("insert into eproduct (name, price, date\_added) values ('New Product', 17800.00, now())");

out.println("Executed an insert operation<br>");

stmt.executeUpdate("update eproduct set price=2000 where name = 'New Product'");

out.println("Executed an update operation<br>");

stmt.executeUpdate("delete from eproduct where name = 'New Product'");

out.println("Executed a delete operation<br>");

stmt.close();

conn.closeConnection();

out.println("</body></html>");

conn.closeConnection();

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request, response);

}

}

Web.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>JDBC Insert, Update, Delete</display-name>

<welcome-file-list>

<welcome-file>web.html</welcome-file>

<welcome-file>web.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>Productdetails</servlet-name>

<servlet-class>Productdetails</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Productdetails</servlet-name>

<url-pattern>/list</url-pattern>

</servlet-mapping>

</web-app>

19. Retrieving the product details using product ID

ProductID.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Product ID</title>

</head>

<body>

<h1>Product ID</h1>

<hr>

<form action=*"login"*>

<input type=*"text"* name=*"p\_ID"* placeholder=*"Product ID"*><br>

<input type=*"submit"* value=*"SEARCH"*>

</form>

</body>

</html>

Login.java

**package** ProductID;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.Statement;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

@WebServlet("/login")

**public** **class** Login **extends** HttpServlet{

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

String url="jdbc:mysql://localhost:3306/student\_details";

String uname="root";

String pass="";

response.setContentType("text/html");

String pId = request.getParameter("p\_ID");

PrintWriter out = response.getWriter();

String query="select \* from product where p\_id=?";

out.print("<h1>Displaying the Product Details...</h1>");

out.print("<table border='1'><tr><th>Product Id</th><th>Product Name</th><th>Product Price</th></tr>");

**try** {

Class.*forName*("com.mysql.cj.jdbc.Driver");

Connection dbCon = DriverManager.*getConnection*(url, uname, pass);

PreparedStatement st= dbCon.prepareStatement(query);

st.setString(1, pId);

ResultSet rs =st.executeQuery();

**while**(rs.next()) {

out.print("<tr><td>");

out.println(rs.getInt(1));

out.print("</td>");

out.print("<td>");

out.print(rs.getString(2));

out.print("</td>");

out.print("<td>");

out.print(rs.getInt(3));

out.print("</td>");

out.print("</tr>");

}

}

**catch**(Exception e){

System.***out***.println("Some Issue : "+ e.getMessage());

}

out.print("</table>");

}

}

20. Configuration of Hibernate

Hibernate.xml

<?xml version=*'1.0'* encoding=*'utf-8'*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- Database connection settings -->

<property name=*"connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/ecommerce</property>

<property name=*"connection.username"*>mysql user id</property>

<property name=*"connection.password"*>mysql password</property>

</session-factory>

</hibernate-configuration>

Hibernate.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Hibernate Configuration Example</title>

</head>

<body>

<a href=*"init"*>Initialize Hibernate</a><br>

</body>

</html>

Hibernateconfiguration.java

**package** com.HibernateConfiguration;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.boot.Metadata;

**import** org.hibernate.boot.MetadataSources;

**import** org.hibernate.boot.registry.StandardServiceRegistry;

**import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;

**public** **class** Hibernateconfiguration {

**private** **static** **final** SessionFactory ***sessionFactory***;

**static** {

**try** {

StandardServiceRegistry standardRegistry = **new** StandardServiceRegistryBuilder()

.configure("hibernate.cfg.xml").build();

Metadata metaData = **new** MetadataSources(standardRegistry).getMetadataBuilder().build();

***sessionFactory*** = metaData.getSessionFactoryBuilder().build();

} **catch** (Throwable th) {

**throw** **new** ExceptionInInitializerError(th);

}

}

**public** **static** SessionFactory getSessionFactory() {

**return** ***sessionFactory***;

}

}

Hibernateconfig.java

**package** com.HibernateConfiguration;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletConfig;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.transaction.\*;

**import** javax.xml.bind.\*;

**import** java.io.Serializable;

**import** java.util.List;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**import** com.ecommerce.EProduct;

**import** com.ecommerce.HibernateUtil;

**public** **class** hibernateconfig {

@WebServlet("/hibernateconfig")

**public** **class** hibernateconfig **extends** HttpServlet {

**private** **static** **final** **long** serialVersionUID = 1L;

**public** hibernateconfig() {

**super**();

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

PrintWriter out = response.getWriter();

out.println("<html><body>");

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.openSession();

out.println("Hibernate Session opened.<br>");

session.close();

out.println("Hibernate Session closed.<br>");

out.println("</body></html>");

} **catch** (Exception ex) {

**throw** ex;

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// **TODO** Auto-generated method stub

doGet(request, response);

}

}

}

Hibernates.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>HibernateConfig</display-name>

<welcome-file-list>

<welcome-file>hibernate.html</welcome-file>

<welcome-file>hibernate.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>Hibernateconfigurtion</servlet-name>

<servlet-class>hibernateconfig</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Hibernateconfiguration</servlet-name>

<url-pattern>/init</url-pattern>

</servlet-mapping>

</web-app>

21. Hibernate configuration using xml in eclipse

Hib.sql

**CREATE** DATABASE ecommerce

USE ecommerce

**CREATE** **TABLE** eproduct (ID bigint **primary** **key** auto\_increment, name **varchar**(100), price **decimal**(10,2), date\_added **timestamp** **default** now())

**INSERT** **INTO** eproduct(name, ‘HP Laptop ABC’, 12000)

**INSERT** **INTO** eproduct(name, ‘Acer Laptop ABC’, 14000)

**INSERT** **INTO** eproduct(name, ‘Lenovo Laptop ABC’, 12000)

**SELECT** \* **from** eproduct

Hib.cfg.xml

<?xml version=*'1.0'* encoding=*'utf-8'*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- Database connection settings -->

<property name=*"connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/ecommerce</property>

<property name=*"connection.username"*>mysql user id</property>

<property name=*"connection.password"*>mysql password</property>

</session-factory>

</hibernate-configuration>

HibernateUtil.java

**package** hib;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.boot.Metadata;

**import** org.hibernate.boot.MetadataSources;

**import** org.hibernate.boot.registry.StandardServiceRegistry;

**import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;

**public** **class** HibernateUtil {

**private** **static** **final** SessionFactory ***sessionFactory***;

**static** {

**try** {

StandardServiceRegistry standardRegistry = **new** StandardServiceRegistryBuilder()

.configure("hibernate.cfg.xml").build();

Metadata metaData = **new** MetadataSources(standardRegistry).getMetadataBuilder().build();

***sessionFactory*** = metaData.getSessionFactoryBuilder().build();

} **catch** (Throwable th) {

**throw** **new** ExceptionInInitializerError(th);

}

}

**public** **static** SessionFactory getSessionFactory() {

**return** ***sessionFactory***;

}

}

Hiber.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Hibernate With XML</title>

</head>

<body>

<a href=*"list”*>List Products</a><br>

</body>

</html>

22. Hibernate configuration using annotations in eclipse

Annotations.sql

**CREATE** DATABASE ecommerce

USE ecommerce

**CREATE** **TABLE** eproduct (ID bigint **primary** **key** auto\_increment, name **varchar**(100), price **decimal**(10,2), date\_added **timestamp** **default** now())

**INSERT** **INTO** eproduct(name, ‘HP Laptop ABC’, 12000)

**INSERT** **INTO** eproduct(name, ‘Acer Laptop ABC’, 14000)

**INSERT** **INTO** eproduct(name, ‘Lenovo Laptop ABC’, 12000)

**SELECT** \* **from** eproduct

HibernateAnnotations.xml

<?xml version=*'1.0'* encoding=*'utf-8'*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- Database connection settings -->

<property name=*"connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/ecommerce</property>

<property name=*"connection.username"*>mysql user id</property>

<property name=*"connection.password"*>mysql password</property>

</session-factory>

</hibernate-configuration>

HiberAnno.java

**package** hibernateannotations;

**package** com.ecommerce;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.boot.Metadata;

**import** org.hibernate.boot.MetadataSources;

**import** org.hibernate.boot.registry.StandardServiceRegistry;

**import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;

**public** **class** Hiberanno {

**private** **static** **final** SessionFactory ***sessionFactory***;

**static** {

**try** {

StandardServiceRegistry standardRegistry = **new** StandardServiceRegistryBuilder()

.configure("hibernate.cfg.xml").build();

Metadata metaData = **new** MetadataSources(standardRegistry).getMetadataBuilder().build();

***sessionFactory*** = metaData.getSessionFactoryBuilder().build();

} **catch** (Throwable th) {

**throw** **new** ExceptionInInitializerError(th);

}

}

**public** **static** SessionFactory getSessionFactory() {

**return** ***sessionFactory***;

}

}

HiberAnno.java

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Hibernate Annotation Example</title>

</head>

<body>

<a href=*"list"*>List Products</a><br>

</body>

</html>

Eproduct.java

**package** com.ecommerce;

**import** java.math.BigDecimal;

**import** java.util.Date;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name= "eproduct")

**public** **class** EProduct {

@Id @GeneratedValue

@Column(name = "ID")

**private** **long** ID;

@Column(name = "name")

**private** String name;

@Column(name = "price")

**private** BigDecimal price;

@Column(name = "date\_added")

**private** Date dateAdded;

**public** **long** getID() {**return** **this**.ID; }

**public** String getName() { **return** **this**.name;}

**public** BigDecimal getPrice() { **return** **this**.price;}

**public** Date getDateAdded() { **return** **this**.dateAdded;}

**public** **void** setID(**long** id) { **this**.ID = id;}

**public** **void** setName(String name) { **this**.name = name;}

**public** **void** setPrice(BigDecimal price) { **this**.price = price;}

**public** **void** setDateAdded(Date date) { **this**.dateAdded = date;}

}

Listproducts.java

**package** hibernateannotations;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletConfig;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.transaction.\*;

**import** javax.xml.bind.\*;

**import** java.io.Serializable;

**import** java.util.List;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**import** com.ecommerce.EProduct;

**import** com.ecommerce.HiberAnno;

@WebServlet("/Listproducts")

**public** **class** Listproducts **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** Listproducts() {

**super**();

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.openSession();

List<EProduct> list = session.createQuery("from EProduct").list();

session.close();

PrintWriter out = response.getWriter();

out.println("<html><body>");

out.println("<b>Product Listing</b><br>");

**for**(EProduct p: list) {

out.println("ID: " + String.*valueOf*(p.getID()) + ", Name: " + p.getName() +

", Price: " + String.*valueOf*(p.getPrice()) + ", Date Added: " + p.getDateAdded().toString() + "<br>");

}

out.println("</body></html>");

} **catch** (Exception ex) {

**throw** ex;

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request, response);

}

}

Ha.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>HibernateAnnotations</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>Listproducts</servlet-name>

<servlet-class>Listproducts</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Listproducts</servlet-name>

<url-pattern>/list</url-pattern>

</servlet-mapping>

</web-app>

23. Hibernate logging by log4j

Hibernatelog4.xml

<?xml version=*'1.0'* encoding=*'utf-8'*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- Database connection settings -->

<property name=*"connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/ecommerce</property>

<property name=*"connection.username"*>mysql user id</property>

<property name=*"connection.password"*>mysql password</property>

</session-factory>

</hibernate-configuration>

Log4.java

**package** com.keerthana.hibern;

**public** **class** log4 {

log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=/home/oem/Downloads/hibernate.log

log4j.appender.file.MaxFileSize=1MB

log4j.appender.file.MaxBackupIndex=1

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

# Direct log messages to stdout

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.Target=System.out

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

# Root logger option

log4j.rootLogger=INFO, file, stdout

# Log everything. Good **for** troubleshooting

log4j.logger.org.hibernate=INFO

# Log all JDBC parameters

log4j.logger.org.hibernate.type=ALL

}

Hibernateex.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Hibernate Configuration Example</title>

</head>

<body>

<a href=*"init"*>Initialize Hibernate</a><br>

</body>

</html>

Hibernatelog4j.java

**package** com.keerthana.hibern;

**package** com.ecommerce;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.boot.Metadata;

**import** org.hibernate.boot.MetadataSources;

**import** org.hibernate.boot.registry.StandardServiceRegistry;

**import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;

**public** **class** hibernatelog4j {

**private** **static** **final** SessionFactory ***sessionFactory***;

**static** {

**try** {

StandardServiceRegistry standardRegistry = **new** StandardServiceRegistryBuilder()

.configure("hibernate.cfg.xml").build();

Metadata metaData = **new** MetadataSources(standardRegistry).getMetadataBuilder().build();

***sessionFactory*** = metaData.getSessionFactoryBuilder().build();

} **catch** (Throwable th) {

**throw** **new** ExceptionInInitializerError(th);

}

}

**public** **static** SessionFactory getSessionFactory() {

**return** ***sessionFactory***;

}

}

Hibernatelog4.java

**package** com.keerthana.hibern;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletConfig;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.transaction.\*;

**import** javax.xml.bind.\*;

**import** java.io.Serializable;

**import** java.util.List;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**import** com.ecommerce.hibernatelog4j;

**import** com.ecommerce.hibernatelog4;

@WebServlet("/hibernatelog4")

**public** **class** hibernatelog4 **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** hibernatelog4() {

**super**();

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

PrintWriter out = response.getWriter();

out.println("<html><body>");

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.openSession();

out.println("Hibernate Session opened.<br>");

session.close();

out.println("Hibernate Session closed.<br>");

out.println("</body></html>");

} **catch** (Exception ex) {

**throw** ex;

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request, response);

}

}

Hibernatelog4.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>HibernateLogging</display-name>

<welcome-file-list>

<welcome-file>hibernateex.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>hibernatelog4</servlet-name>

<servlet-class>hibernatelog4</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>hibernatelog4</servlet-name>

<url-pattern>/init</url-pattern>

</servlet-mapping>

</web-app>

24. Collection mapping in hibernate

Hibernmap.sql

**CREATE** **TABLE** `colors` (

`ID` bigint(20) **NOT** **NULL** AUTO\_INCREMENT,

`color\_name` **varchar**(40) **DEFAULT** **NULL**,

`idx` **int**(11) **DEFAULT** **NULL**,

`product\_id` bigint(20) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 **DEFAULT** CHARSET=latin1;

LOCK TABLES `colors` WRITE;

**INSERT** **INTO** `colors` **VALUES** (1,'Red',0,1),(2,'Silver',1,1),(3,'Gray',0,2),(4,'White',1,2),(5,'Maroon',0,3);

UNLOCK TABLES;

**DROP** **TABLE** IF **EXISTS** `eproduct`;

**CREATE** **TABLE** `eproduct` (

`ID` bigint(20) **NOT** **NULL** AUTO\_INCREMENT,

`name` **varchar**(100) **DEFAULT** **NULL**,

`price` **decimal**(10,2) **DEFAULT** **NULL**,

`date\_added` **timestamp** **NOT** **NULL** **DEFAULT** CURRENT\_TIMESTAMP,

**PRIMARY** **KEY** (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 **DEFAULT** CHARSET=latin;

LOCK TABLES `eproduct` WRITE;

**INSERT** **INTO** `eproduct` **VALUES** (1,'HP Laptop ABC',21900.00,'2019-06-04 07:18:57'),(2,'Acer Laptop ABC',23300.00,'2019-06-04 07:19:07'),(3,'Lenovo Laptop ABC',33322.00,'2019-06-04 07:19:19');

UNLOCK TABLES;

**DROP** **TABLE** IF **EXISTS** `finance`;

**CREATE** **TABLE** `finance` (

`ID` bigint(20) **NOT** **NULL** AUTO\_INCREMENT,

`ftype` **varchar**(10) **DEFAULT** **NULL**,

`name` **varchar**(30) **DEFAULT** **NULL**,

`product\_id` bigint(20) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 **DEFAULT** CHARSET=latin1;

LOCK TABLES `finance` WRITE;

**INSERT** **INTO** `finance` **VALUES** (1,'CREDITCARD','EMI on Citibank Card',1),(3,'BANK','40% finance from SBI',2),(4,'BANK','60% finance from ICICI',3),(5,'BANK','20% finance from ICICI',1);

UNLOCK TABLES;

**DROP** **TABLE** IF **EXISTS** `os`;

**CREATE** **TABLE** `os` (

`ID` bigint(20) **NOT** **NULL** AUTO\_INCREMENT,

`name` **varchar**(30) **DEFAULT** **NULL**,

`product\_id` bigint(20) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 **DEFAULT** CHARSET=latin1;

LOCK TABLES `os` WRITE;

**INSERT** **INTO** `os` **VALUES** (1,'Windows 10',1),(2,'Windows 10',2),(3,'FreeDOS',2),(4,'RedHat Linux',2),(5,'Windows 10',3);

UNLOCK TABLES;

**DROP** **TABLE** IF **EXISTS** `screensizes`;

**CREATE** **TABLE** `screensizes` (

`ID` bigint(20) **NOT** **NULL** AUTO\_INCREMENT,

`size` **varchar**(10) **DEFAULT** **NULL**,

`product\_id` bigint(20) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 **DEFAULT** CHARSET=latin1;

LOCK TABLES `screensizes` WRITE;

**INSERT** **INTO** `screensizes` **VALUES** (1,'12 in',1),(2,'14.5 in',2),(3,'14.9 in',2),(4,'15.5 in',3);

UNLOCK TABLES;

Color.java

**package** com.keerthana.hibernatemapping;

**package** com.ecommerce;

**public** **class** color {

**private** **long** COLORID;

**private** String name;

**public** Color() {

}

**public** Color(String name) {

**this**.COLORID = 0;

**this**.name = name;

}

**public** **long** getCOLORID() {**return** **this**.COLORID; }

**public** String getName() { **return** **this**.name;}

**public** **void** setCOLORID(**long** id) { **this**.COLORID = id;}

**public** **void** setName(String name) { **this**.name = name;}

}

eproduct.java

package hibernatemapping;

package com.ecommerce;

import java.math.BigDecimal;

import java.util.Collection;

import java.util.Date;

import java.util.List;

import java.util.Set;

import java.util.Map;

public class eProduct {

private long ID;

private String name;

private BigDecimal price;

private Date dateAdded;

private List<Color> colors;

private Collection<ScreenSizes> screenSizes;

private Set<OS> os;

private Map finance;

public eProduct() {

}

public long getID() {return this.ID; }

public String getName() { return this.name;}

public BigDecimal getPrice() { return this.price;}

public Date getDateAdded() { return this.dateAdded;}

public List<Color> getColors() { return this.colors;}

public Collection<ScreenSizes> getScreensizes() { return this.screenSizes;}

public Set<OS> getOs() { return this.os;}

public Map getFinance() { return this.finance;}

public void setID(long id) { this.ID = id;}

public void setName(String name) { this.name = name;}

public void setPrice(BigDecimal price) { this.price = price;}

public void setDateAdded(Date date) { this.dateAdded = date;}

public void setColors(List<Color> colors) { this.colors = colors;}

public void setScreensizes(Collection<ScreenSizes> sizes) { this.screenSizes = sizes;}

public void setOs(Set<OS> os) { this.os = os;}

public void setFinance(Map finance) { this.finance = finance;}

}

Hibernfinance.java

**package** hibernatemapping;

**package** com.ecommerce;

**public** **class** hibernfinance {

**private** **long** FINANCEID;

**private** String name;

**private** String ftype;

**public** hibernfinance() {

}

**public** hibernfinance(String name, String ftype) {

**this**.FINANCEID = 0;

**this**.name = name;

**this**.ftype = ftype;

}

**public** **long** getFINANCEID() {**return** **this**.FINANCEID; }

**public** String getName() { **return** **this**.name;}

**public** String getFtype() { **return** **this**.ftype;}

**public** **void** setFINANCEID(**long** id) { **this**.FINANCEID = id;}

**public** **void** setName(String name) { **this**.name = name;}

**public** **void** setFtype(String ftype) { **this**.ftype= ftype;}

}

OS. Java

**package** hibernatemapping;

**package** com.ecommerce;

**public** **class** OS {

**private** **long** OSID;

**private** String name;

**public** OS() {

}

**public** OS(String name) {

**this**.OSID = 0;

**this**.name = name;

}

**public** **long** getOSID() {**return** **this**.OSID; }

**public** String getName() { **return** **this**.name;}

**public** **void** setOSID(**long** id) { **this**.OSID = id;}

**public** **void** setName(String name) { **this**.name= name;}

}

Screensizes.java

**package** hibernatemapping;

**public** **class** screensizes {

**private** **long** SCREENID;

**private** String size;

**public** screensizes() {

}

**public** screensizes(String size) {

**this**.SCREENID = 0;

**this**.size = size;

}

**public** **long** getSCREENID() {**return** **this**.SCREENID; }

**public** String getSize() { **return** **this**.size;}

**public** **void** setSCREENID(**long** id) { **this**.SCREENID = id;}

**public** **void** setSize(String size) { **this**.size = size;}

}

Util.java

**package** hibernatemapping;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.boot.Metadata;

**import** org.hibernate.boot.MetadataSources;

**import** org.hibernate.boot.registry.StandardServiceRegistry;

**import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;

**public** **class** Util {

**private** **static** **final** SessionFactory ***sessionFactory***;

**static** {

**try** {

StandardServiceRegistry standardRegistry = **new** StandardServiceRegistryBuilder()

.configure("hibernate.cfg.xml").build();

Metadata metaData = **new** MetadataSources(standardRegistry).getMetadataBuilder().build();

***sessionFactory*** = metaData.getSessionFactoryBuilder().build();

} **catch** (Throwable th) {

**throw** **new** ExceptionInInitializerError(th);

}

}

**public** **static** SessionFactory getSessionFactory() {

**return** ***sessionFactory***;

}

}

Hm.xml

<?xml version=*"1.0"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package=*"hibernatemapping"*>

<class name=*"Color"* table=*"colors"*>

<id name=*"COLORID"* type=*"long"* column=*"ID"*>

<generator class=*"identity"*/>

</id>

<property name=*"name"* type=*"string"* column=*"COLOR\_NAME"*/>

</class>

</hibernate-mapping>

Hibernmapp.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package=*"hibernatemapping"*>

<class name=*"EProduct"* table=*"eproduct"*>

<id name=*"ID"* type=*"long"* column=*"ID"*>

<generator class=*"identity"*/>

</id>

<property name=*"name"* type=*"string"* column=*"NAME"*/>

<property name=*"price"* type=*"big\_decimal"* column=*"PRICE"*/>

<property name=*"dateAdded"* type=*"timestamp"* column=*"DATE\_ADDED"*/>

<list name=*"colors"* cascade=*"all"*>

<key column=*"product\_id"* />

<list-index column=*"idx"* />

<one-to-many class=*"hibernatemapping.Color"* />

</list>

<bag name=*"screensizes"* cascade=*"all"*>

<key column=*"product\_id"*></key>

<one-to-many class=*"hibernatemapping.ScreenSizes"*/>

</bag>

<set name = *"os"* cascade=*"all"*>

<key column = *"product\_id"*/>

<one-to-many class=*"OS"*/>

</set>

<map name = *"finance"* cascade=*"all"*>

<key column = *"product\_id"*/>

<index column = *"ftype"* type = *"string"*/>

<one-to-many class=*"hibernatemapping.Finance"*/>

</map>

</class>

</hibernate-mapping>

Finance.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package=*"hibernatemapping"*>

<class name=*"Finance"* table=*"finance"*>

<id name=*"FINANCEID"* type=*"long"* column=*"ID"*>

<generator class=*"identity"*/>

</id>

<property name=*"name"* type=*"string"* column=*"NAME"*/>

<property name=*"ftype"* type=*"string"* column=*"FTYPE"*/>

</class>

</hibernate-mapping>

OS.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package=*"hibernatemapping"*>

<class name=*"OS"* table=*"os"*>

<id name=*"OSID"* type=*"long"* column=*"ID"*>

<generator class=*"identity"*/>

</id>

<property name=*"name"* type=*"string"* column=*"NAME"*/>

</class>

</hibernate-mapping>

ScreenSizes.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package=*"hibernatemapping"*>

<class name=*"ScreenSizes"* table=*"screensizes"*>

<id name=*"SCREENID"* type=*"long"* column=*"ID"*>

<generator class=*"identity"*/>

</id>

<property name=*"size"* type=*"string"* column=*"SIZE"*/>

</class>

</hibernate-mapping>

Hibernate.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- Database connection settings -->

<property name=*"connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/ecommerce</property>

<property name=*"connection.username"*>root</property>

<property name=*"connection.password"*>master</property>

<mapping resource=*"com/hibernatemapping/EProduct.hbm.xml"*/>

<mapping resource=*"com/hibernatemapping/Color.hbm.xml"*/>

<mapping resource=*"com/hibernatemapping/ScreenSizes.hbm.xml"*/>

<mapping resource=*"com/hibernatemapping/Os.hbm.xml"*/>

<mapping resource=*"com/hibernmapping/Finance.hbm.xml"*/>

</session-factory>

</hibernate-configuration>

Collection.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Hibernate Collection Mapping</title>

</head>

<body>

<a href=*"details"*>Product Details</a><br>

</body>

</html>

ProductDetails.java

**package** hibernatemapping;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletConfig;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.transaction.\*;

**import** javax.xml.bind.\*;

**import** java.io.Serializable;

**import** java.math.BigDecimal;

**import** java.util.ArrayList;

**import** java.util.Calendar;

**import** java.util.Collection;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Set;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**import** com.ecommerce.Color;

**import** com.ecommerce.EProduct;

**import** com.ecommerce.Finance;

**import** com.ecommerce.HibernateUtil;

**import** com.ecommerce.OS;

**import** com.ecommerce.ScreenSizes;

@WebServlet("/ProductDetails")

**public** **class** ProductDetails **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** ProductDetails() {

**super**();

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.openSession();

List<EProduct> list = session.createQuery("from EProduct").list();

PrintWriter out = response.getWriter();

out.println("<html><body>");

out.println("<b>Product Listing</b><br>");

**for**(EProduct p: list) {

out.println("ID: " + String.*valueOf*(p.getID()) + ", Name: " + p.getName() +

", Price: " + String.*valueOf*(p.getPrice()) + ", Date Added: " + p.getDateAdded().toString());

List<Color> colors = p.getColors();

out.println("Colors: ");

**for**(Color c: colors) {

out.print(c.getName() + "&nbsp;");

}

Collection<ScreenSizes> sizes= p.getScreensizes();

out.println(", Screen Sizes: ");

**for**(ScreenSizes s: sizes) {

out.print(s.getSize() + "&nbsp;");

}

Set<OS> os= p.getOs();

out.println(", OS : ");

**for**(OS o: os) {

out.print(o.getName() + "&nbsp;");

}

Map finances = p.getFinance();

out.println(", Finance Options: ");

**if** (finances.get("CREDITCARD") != **null**) {

Finance f = (Finance) finances.get("CREDITCARD");

out.println(f.getName() + " &nbsp;");

}

**if** (finances.get("BANK") != **null**) {

Finance f = (Finance) finances.get("BANK");

out.println(f.getName() + " &nbsp;");

}

out.println("<hr>");

}

session.close();

out.println("</body></html>");

} **catch** (Exception ex) {

**throw** ex;

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request, response);

}

}

List.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>HibernateListMapping</display-name>

<welcome-file-list>

<welcome-file>collection.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>ProductDetails</servlet-name>

<servlet-class>ProductDetails</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ProductDetails</servlet-name>

<url-pattern>/details</url-pattern>

</servlet-mapping>

</web-app>

25.HIBERNATE LAZY COLLECTION

Hibernatelazy.sql

**DROP** **TABLE** IF **EXISTS** `colors`;

**CREATE** **TABLE** `colors` (

`ID` bigint(20) **NOT** **NULL** AUTO\_INCREMENT,

`color\_name` **varchar**(40) **DEFAULT** **NULL**,

`idx` **int**(11) **DEFAULT** **NULL**,

`product\_id` bigint(20) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 **DEFAULT** CHARSET=latin1;

LOCK TABLES `colors` WRITE;

**INSERT** **INTO** `colors` **VALUES** (1,'Red',0,1),(2,'Silver',1,1),(3,'Gray',0,2),(4,'White',1,2),(5,'Maroon',0,3);

UNLOCK TABLES;

**DROP** **TABLE** IF **EXISTS** `eproduct`;

**CREATE** **TABLE** `eproduct` (

`ID` bigint(20) **NOT** **NULL** AUTO\_INCREMENT,

`name` **varchar**(100) **DEFAULT** **NULL**,

`price` **decimal**(10,2) **DEFAULT** **NULL**,

`date\_added` **timestamp** **NOT** **NULL** **DEFAULT** CURRENT\_TIMESTAMP,

**PRIMARY** **KEY** (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 **DEFAULT** CHARSET=latin1;

LOCK TABLES `eproduct` WRITE;

**INSERT** **INTO** `eproduct` **VALUES** (1,'HP Laptop ABC',21900.00,'2019-06-04 07:18:57'),(2,'Acer Laptop ABC',23300.00,'2019-06-04 07:19:07'),(3,'Lenovo Laptop ABC',33322.00,'2019-06-04 07:19:19');

UNLOCK TABLES;

**DROP** **TABLE** IF **EXISTS** `finance`;

**CREATE** **TABLE** `finance` (

`ID` bigint(20) **NOT** **NULL** AUTO\_INCREMENT,

`ftype` **varchar**(10) **DEFAULT** **NULL**,

`name` **varchar**(30) **DEFAULT** **NULL**,

`product\_id` bigint(20) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 **DEFAULT** CHARSET=latin1;

LOCK TABLES `finance` WRITE;

**INSERT** **INTO** `finance` **VALUES** (1,'CREDITCARD','EMI on Citibank Card',1),(3,'BANK','40% finance from SBI',2),(4,'BANK','60% finance from ICICI',3),(5,'BANK','20% finance from ICICI',1);

UNLOCK TABLES;

**DROP** **TABLE** IF **EXISTS** `os`;

**CREATE** **TABLE** `os` (

`ID` bigint(20) **NOT** **NULL** AUTO\_INCREMENT,

`name` **varchar**(30) **DEFAULT** **NULL**,

`product\_id` bigint(20) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 **DEFAULT** CHARSET=latin1;

LOCK TABLES `os` WRITE;

**INSERT** **INTO** `os` **VALUES** (1,'Windows 10',1),(2,'Windows 10',2),(3,'FreeDOS',2),(4,'RedHat Linux',2),(5,'Windows 10',3);

UNLOCK TABLES;

**DROP** **TABLE** IF **EXISTS** `screensizes`;

**CREATE** **TABLE** `screensizes` (

`ID` bigint(20) **NOT** **NULL** AUTO\_INCREMENT,

`size` **varchar**(10) **DEFAULT** **NULL**,

`product\_id` bigint(20) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 **DEFAULT** CHARSET=latin1;

LOCK TABLES `screensizes` WRITE;

**INSERT** **INTO** `screensizes` **VALUES** (1,'12 in',1),(2,'14.5 in',2),(3,'14.9 in',2),(4,'15.5 in',3);

UNLOCK TABLES;

Color.java

**package** com.LazyCollection;

**public** **class** color {

**private** **long** COLORID;

**private** String name;

**public** color() {

}

**public** color(String name) {

**this**.COLORID = 0;

**this**.name = name;

}

**public** **long** getCOLORID() {**return** **this**.COLORID; }

**public** String getName() { **return** **this**.name;}

**public** **void** setCOLORID(**long** id) { **this**.COLORID = id;}

**public** **void** setName(String name) { **this**.name = name;}

}

Product.java

**package** com.LazyCollection;

**import** java.math.BigDecimal;

**import** java.util.Collection;

**import** java.util.Date;

**import** java.util.List;

**import** java.util.Set;

**import** java.util.Map;

**public** **class** Product {

**private** **long** ID;

**private** String name;

**private** BigDecimal price;

**private** Date dateAdded;

**private** List<Color> colors;

**private** Set<Finance> finance;

**private** PDescription pdescrip;

**public** Product() {

}

**public** **long** getID() {**return** **this**.ID; }

**public** String getName() { **return** **this**.name;}

**public** BigDecimal getPrice() { **return** **this**.price;}

**public** Date getDateAdded() { **return** **this**.dateAdded;}

**public** List<Color> getColors() { **return** **this**.colors;}

**public** Set<Finance> getFinance() { **return** **this**.finance;}

**public** PDescription getPdescrip() { **return** **this**.pdescrip;}

**public** **void** setID(**long** id) { **this**.ID = id;}

**public** **void** setName(String name) { **this**.name = name;}

**public** **void** setPrice(BigDecimal price) { **this**.price = price;}

**public** **void** setDateAdded(Date date) { **this**.dateAdded = date;}

**public** **void** setColors(List<Color> colors) { **this**.colors = colors;}

**public** **void** setFinance(Set<Finance> finance) { **this**.finance = finance;}

**public** **void** setPdescrip(PDescription pdescrip) { **this**.pdescrip = pdescrip;}

}

Lazyfinance.java

**package** com.LazyCollection;

**public** **class** Lazyfinance {

**private** **long** FINANCEID;

**private** String name;

**private** String ftype;

**public** Lazyfinance() {

}

**public** Lazyfinance(String name, String ftype) {

**this**.FINANCEID = 0;

**this**.name = name;

**this**.ftype = ftype;

}

**public** **long** getFINANCEID() {**return** **this**.FINANCEID; }

**public** String getName() { **return** **this**.name;}

**public** String getFtype() { **return** **this**.ftype;}

**public** **void** setFINANCEID(**long** id) { **this**.FINANCEID = id;}

**public** **void** setName(String name) { **this**.name = name;}

**public** **void** setFtype(String ftype) { **this**.ftype= ftype;}

}

OS.java

**package** com.LazyCollection;

**public** **class** OS {

**private** **long** OSID;

**private** String name;

**public** OS() {

}

**public** OS(String name) {

**this**.OSID = 0;

**this**.name = name;

}

**public** **long** getOSID() {**return** **this**.OSID; }

**public** String getName() { **return** **this**.name;}

**public** **void** setOSID(**long** id) { **this**.OSID = id;}

**public** **void** setName(String name) { **this**.name= name;}

}

ScreenSizes.java

**package** com.LazyCollection;

**public** **class** ScreenSizes {

**private** **long** SCREENID;

**private** String size;

**public** ScreenSizes() {

}

**public** ScreenSizes(String size) {

**this**.SCREENID = 0;

**this**.size = size;

}

**public** **long** getSCREENID() {**return** **this**.SCREENID; }

**public** String getSize() { **return** **this**.size;}

**public** **void** setSCREENID(**long** id) { **this**.SCREENID = id;}

**public** **void** setSize(String size) { **this**.size = size;}

}

HibernateUtil.java

**package** com.LazyCollection;

mport org.hibernate.SessionFactory;

**import** org.hibernate.boot.Metadata;

**import** org.hibernate.boot.MetadataSources;

**import** org.hibernate.boot.registry.StandardServiceRegistry;

**import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;

**public** **class** HibernateUtil {

**private** **static** **final** SessionFactory sessionFactory;

**static** {

**try** {

StandardServiceRegistry standardRegistry = **new** StandardServiceRegistryBuilder()

.configure("hibernate.cfg.xml").build();

Metadata metaData = **new** MetadataSources(standardRegistry).getMetadataBuilder().build();

sessionFactory = metaData.getSessionFactoryBuilder().build();

} **catch** (Throwable th) {

**throw** **new** ExceptionInInitializerError(th);

}

}

**public** **static** SessionFactory getSessionFactory() {

**return** sessionFactory;

}

}

Color.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package=*"Lazy"*>

<class name=*"Color"* table=*"colors"*>

<id name=*"COLORID"* type=*"long"* column=*"ID"*>

<generator class=*"identity"*/>

</id>

<property name=*"name"* type=*"string"* column=*"COLOR\_NAME"*/>

</class>

</hibernate-mapping>

e.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package=*"com.Lazy"*>

<class name=*"EProduct"* table=*"eproduct"*>

<id name=*"ID"* type=*"long"* column=*"ID"*>

<generator class=*"identity"*/>

</id>

<property name=*"name"* type=*"string"* column=*"NAME"*/>

<property name=*"price"* type=*"big\_decimal"* column=*"PRICE"*/>

<property name=*"dateAdded"* type=*"timestamp"* column=*"DATE\_ADDED"*/>

<list name=*"colors"* cascade=*"all"* lazy=*"true"*>

<key column=*"product\_id"* />

<list-index column=*"idx"* />

<one-to-many class=*"com.Lazy.Color"* />

</list>

<bag name=*"screensizes"* cascade=*"all"* lazy=*"true"*>

<key column=*"product\_id"*></key>

<one-to-many class=*"com.Lazy.ScreenSizes"*/>

</bag>

<set name = *"os"* cascade=*"all"* lazy=*”true”*>

<key column = *"product\_id"*/>

<one-to-many class=*"OS"*/>

</set>

<map name = *"finance"* cascade=*"all"*>

<key column = *"product\_id"*/>

<index column = *"ftype"* type = *"string"*/>

<one-to-many class=*"com.Lazy.Finance"*/>

</map>

</class>

</hibernate-mapping>

Finance.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package=*"com.Lazy"*>

<class name=*"Finance"* table=*"finance"*>

<id name=*"FINANCEID"* type=*"long"* column=*"ID"*>

<generator class=*"identity"*/>

</id>

<property name=*"name"* type=*"string"* column=*"NAME"*/>

<property name=*"ftype"* type=*"string"* column=*"FTYPE"*/>

</class>

</hibernate-mapping>

OS.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package=*"com.Lazy"*>

<class name=*"OS"* table=*"os"*>

<id name=*"OSID"* type=*"long"* column=*"ID"*>

<generator class=*"identity"*/>

</id>

<property name=*"name"* type=*"string"* column=*"NAME"*/>

</class>

</hibernate-mapping>

Hiberconfig.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package=*"com.Lazy"*>

<class name=*"ScreenSizes"* table=*"screensizes"*>

<id name=*"SCREENID"* type=*"long"* column=*"ID"*>

<generator class=*"identity"*/>

</id>

<property name=*"size"* type=*"string"* column=*"SIZE"*/>

</class>

</hibernate-mapping>

Hibernate.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- Database connection settings -->

<property name=*"connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/ecommerce</property>

<property name=*"connection.username"*>root</property>

<property name=*"connection.password"*>master</property>

<mapping resource=*"com/Lazy/EProduct.hbm.xml"*/>

<mapping resource=*"com/Lazy/Color.hbm.xml"*/>

<mapping resource=*"com/Lazy/ScreenSizes.hbm.xml"*/>

<mapping resource=*"com/Lazye/Os.hbm.xml"*/>

<mapping resource=*"com/Lazy/Finance.hbm.xml"*/>

</session-factory>

</hibernate-configuration>

Coll.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Hibernate Collection Mapping With XML</title>

</head>

<body>

<a href=*"details"*>Product Details</a><br>

</body>

</html>

h.java

**package** com.LazyCollection;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletConfig;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.transaction.\*;

**import** javax.xml.bind.\*;

**import** java.io.Serializable;

**import** java.math.BigDecimal;

**import** java.util.ArrayList;

**import** java.util.Calendar;

**import** java.util.Collection;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Set;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**import** Lazy.Color;

**import** Lazy.Product;

**import** Lazy.Finance;

**import** Lazy.HibernateUtil;

**import** Lazy.PDescription;

@WebServlet("/h")

**public** **class** h **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** h() {

**super**();

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

SessionFactory factory = HibernateUtil.*getSessionFactory*();

Session session = factory.openSession();

List<EProduct> list = session.createQuery("from EProduct").list();

PrintWriter out = response.getWriter();

out.println("<html><body>");

out.println("<b>One to One Mapping</b><br>");

**for**(EProduct p: list) {

out.println("ID: " + String.*valueOf*(p.getID()) + ", Name: " + p.getName() +

", Price: " + String.*valueOf*(p.getPrice()) + ", Date Added: " + p.getDateAdded().toString());

PDescription descrip = p.getPdescrip();

out.println("<br>Description:" + descrip.getDescrip());

out.println("<hr>");

}

out.println("<b>One to Many and Many to One Mapping</b><br>");

**for**(EProduct p: list) {

out.println("ID: " + String.*valueOf*(p.getID()) + ", Name: " + p.getName() +

", Price: " + String.*valueOf*(p.getPrice()) + ", Date Added: " + p.getDateAdded().toString());

List<Color> colors = p.getColors();

out.println("<br>Colors: <ul>");

**for**(Color c: colors) {

out.print("<li>" + c.getName() + "</li>");

}

out.println("</ul>");

out.println("<hr>");

}

out.println("<b>Many to Many Mapping</b><br>");

**for**(EProduct p: list) {

out.println("ID: " + String.*valueOf*(p.getID()) + ", Name: " + p.getName() +

", Price: " + String.*valueOf*(p.getPrice()) + ", Date Added: " + p.getDateAdded().toString());

Set<Finance> finances= p.getFinance();

out.println("<br>Finance Options : <ul>");

**for**(Finance f: finances) {

out.print("<li>" + f.getFtype() + "</li>");

}

out.println("</ul>");

out.println("<hr>");

}

session.close();

out.println("</body></html>");

} **catch** (Exception ex) {

**throw** ex;

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// **TODO** Auto-generated method stub

doGet(request, response);

}

}

Web.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>HibernateLazyCollections</display-name>

<welcome-file-list>

<welcome-file>Coll.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>ProductDetails</servlet-name>

<servlet-class>ProductDetails</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ProductDetails</servlet-name>

<url-pattern>/details</url-pattern>

</servlet-mapping>

</web-app>

26. COMPONENT MAPPING

Comp.sql

**DROP** **TABLE** IF **EXISTS** `eproduct`;

**CREATE** **TABLE** `eproduct` (

`ID` bigint(20) **NOT** **NULL** AUTO\_INCREMENT,

`name` **varchar**(100) **DEFAULT** **NULL**,

`price` **decimal**(10,2) **DEFAULT** **NULL**,

`date\_added` **timestamp** **NOT** **NULL** **DEFAULT** CURRENT\_TIMESTAMP,

`parts\_hdd` **varchar**(10) **DEFAULT** **NULL**,

`parts\_cpu` **varchar**(10) **DEFAULT** **NULL**,

`parts\_ram` **varchar**(10) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 **DEFAULT** CHARSET=latin1;

LOCK TABLES `eproduct` WRITE;

**INSERT** **INTO** `eproduct` **VALUES** (1,'HP Laptop ABC',21900.00,'2019-06-04 07:18:57','2 Gb HDD','AMD Phenom','4 Gb'),(2,'Acer Laptop ABC',23300.00,'2019-06-04 07:19:07','500 Gb HDD','Core-i7','4 Gb'),(3,'Lenovo Laptop ABC',33322.00,'2019-06-04 07:19:19','1 Tb HDD','Core-i7','8 Gb');

UNLOCK TABLES;

Component.java

**package** com.keerthana.ComponentMapping;

**import** java.math.BigDecimal;

**import** java.util.Collection;

**import** java.util.Date;

**import** java.util.List;

**import** java.util.Set;

**import** java.util.Map;

**public** **class** Component {

**private** **long** ID;

**private** String name;

**private** BigDecimal price;

**private** Date dateAdded;

**private** ProductParts parts;

**public** Component() {

}

**public** **long** getID() {**return** **this**.ID; }

**public** String getName() { **return** **this**.name;}

**public** BigDecimal getPrice() { **return** **this**.price;}

**public** Date getDateAdded() { **return** **this**.dateAdded;}

**public** ProductParts getParts() { **return** **this**.parts;}

**public** **void** setID(**long** id) { **this**.ID = id;}

**public** **void** setName(String name) { **this**.name = name;}

**public** **void** setPrice(BigDecimal price) { **this**.price = price;}

**public** **void** setDateAdded(Date date) { **this**.dateAdded = date;}

**public** **void** setParts(ProductParts parts) { **this**.parts = parts;}

}

ProductParts.java

**package** com.keerthana.ComponentMapping;

**public** **class** ProductParts {

**private** String hdd;

**private** String cpu;

**private** String ram;

**public** String getHdd() { **return** **this**.hdd;}

**public** String getCpu() { **return** **this**.cpu;}

**public** String getRam() { **return** **this**.ram;}

**public** **void** setHdd(String value) { **this**.hdd= value;}

**public** **void** setCpu(String value) { **this**.cpu= value;}

**public** **void** setRam(String value) { **this**.ram= value;}

}

HibernateComponent.java

**package** com.keerthana.ComponentMapping;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.boot.Metadata;

**import** org.hibernate.boot.MetadataSources;

**import** org.hibernate.boot.registry.StandardServiceRegistry;

**import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;

**public** **class** HibernateComponent {

**private** **static** **final** SessionFactory ***sessionFactory***;

**static** {

**try** {

StandardServiceRegistry standardRegistry = **new** StandardServiceRegistryBuilder()

.configure("hibernate.cfg.xml").build();

Metadata metaData = **new** MetadataSources(standardRegistry).getMetadataBuilder().build();

***sessionFactory*** = metaData.getSessionFactoryBuilder().build();

} **catch** (Throwable th) {

**throw** **new** ExceptionInInitializerError(th);

}

}

**public** **static** SessionFactory getSessionFactory() {

**return** ***sessionFactory***;

}

}

Product.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping package=*"com.ComponentMapping"*>

<class name=*"Product"* table=*"product"*>

<id name=*"ID"* type=*"long"* column=*"ID"*>

<generator class=*"identity"*/>

</id>

<property name=*"name"* type=*"string"* column=*"NAME"*/>

<property name=*"price"* type=*"big\_decimal"* column=*"PRICE"*/>

<property name=*"dateAdded"* type=*"timestamp"* column=*"DATE\_ADDED"*/>

<component name=*"parts"* class=*"com.ComponentMapping.ProductParts"*>

<property name=*"hdd"* column=*"parts\_hdd"* type=*"string"* />

<property name=*"cpu"* column=*"parts\_cpu"* type=*"string"* />

<property name=*"ram"* column=*"parts\_ram"* type=*"string"* />

</component>

</class>

</hibernate-mapping>

Cfg.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- Database connection settings -->

<property name=*"connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/ecommerce</property>

<property name=*"connection.username"*>root</property>

<property name=*"connection.password"*>master</property>

<mapping resource=*"com/ComponentMapping/EProduct.hbm.xml"*/>

</session-factory>

</hibernate-configuration>

Index.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Hibernate Component Mapping</title>

</head>

<body>

<a href=*"details"*>Product Details</a><br>

</body>

</html>

ProductDetails.java

**package** com.keerthana.ComponentMapping;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletConfig;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.transaction.\*;

**import** javax.xml.bind.\*;

**import** java.io.Serializable;

**import** java.math.BigDecimal;

**import** java.util.ArrayList;

**import** java.util.Calendar;

**import** java.util.Collection;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Set;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**import** com.ComponentMapping.EProduct;

**import** com.ComponentMapping.HibernateUtil;

**import** com.ComponentMapping.ProductParts;

@WebServlet("/ProductDetails")

**public** **class** ProductDetails **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** ProductDetails() {

**super**();

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

SessionFactory factory = HibernateUtil.getSessionFactory();

Session session = factory.openSession();

List<EProduct> list = session.createQuery("from EProduct").list();

PrintWriter out = response.getWriter();

out.println("<html><body>");

out.println("<b>Component Mapping</b><br>");

**for**(EProduct p: list) {

out.println("ID: " + String.*valueOf*(p.getID()) + ", Name: " + p.getName() +

", Price: " + String.*valueOf*(p.getPrice()) + ", Date Added: " + p.getDateAdded().toString());

ProductParts parts = p.getParts();

out.println("Parts =" + parts.getCpu() + ", " + parts.getHdd() + ", " + parts.getRam());

out.println("<hr>");

}

session.close();

out.println("</body></html>");

} **catch** (Exception ex) {

**throw** ex;

}

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request, response);

}

}

hi.xml

?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

<display-name>HibernateComponentMapping</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>ProductDetails</servlet-name>

<servlet-class>ProductDetails</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ProductDetails</servlet-name>

<url-pattern>/details</url-pattern>

</servlet-mapping>

</web-app>

27. Integration of hibernate with spring

Spring.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>SpringHibernateWeb</groupId>

<artifactId>SpringHibernateWeb</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>SpringHibernateWeb Maven Webapp</name>

<url>http://maven.apache.org</url>

<!-- JBoss repository for Hibernate -->

<repositories>

<repository>

<id>JBoss repository</id>

<url>http://repository.jboss.org/nexus/content/groups/public/</url>

</repository>

</repositories>

<properties>

<org.springframework.version>3.0.5.RELEASE</org.springframework.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.hibernate.javax.persistence</groupId>

<artifactId>hibernate-jpa-2.1-api</artifactId>

<version>1.0.0.Final</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-expression</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context-support</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.16</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>3.6.3.Final</version>

</dependency>

<dependency>

<groupId>javassist</groupId>

<artifactId>javassist</artifactId>

<version>3.12.1.GA</version>

</dependency>

<dependency>

<groupId>taglibs</groupId>

<artifactId>standard</artifactId>

<version>1.1.2</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.4</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.9</version>

</dependency>

</dependencies>

<build>

<finalName>SpringHibernateWeb</finalName>

</build>

</project>

Table.sql

**DROP** **TABLE** IF **EXISTS** `eproduct`;

**CREATE** **TABLE** `eproduct` (

`ID` bigint(20) **NOT** **NULL** AUTO\_INCREMENT,

`name` **varchar**(100) **DEFAULT** **NULL**,

`price` **decimal**(10,2) **DEFAULT** **NULL**,

`date\_added` **timestamp** **NOT** **NULL** **DEFAULT** CURRENT\_TIMESTAMP,

`parts\_hdd` **varchar**(10) **DEFAULT** **NULL**,

`parts\_cpu` **varchar**(10) **DEFAULT** **NULL**,

`parts\_ram` **varchar**(10) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 **DEFAULT** CHARSET=latin1;

LOCK TABLES `eproduct` WRITE;

**INSERT** **INTO** `eproduct` **VALUES** (1,'HP Laptop ABC',21900.00,'2019-06-04 07:18:57','2 Gb HDD','AMD Phenom','4 Gb'),(2,'Acer Laptop ABC',23300.00,'2019-06-04 07:19:07','500 Gb HDD','Core-i7','4 Gb'),(3,'Lenovo Laptop ABC',33322.00,'2019-06-04 07:19:19','1 Tb HDD','Core-i7','8 Gb');

UNLOCK TABLES;

Spring.java

**package** com.keerthana.HibernateSpring;

**import** java.math.BigDecimal;

**import** java.util.Date;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name= "Spring")

**public** **class** Spring {

@Id @GeneratedValue

@Column(name = "ID")

**private** **long** ID;

@Column(name = "name")

**private** String name;

@Column(name = "price")

**private** BigDecimal price;

@Column(name = "date\_added")

**private** Date dateAdded;

**public** **long** getID() {**return** **this**.ID; }

**public** String getName() { **return** **this**.name;}

**public** BigDecimal getPrice() { **return** **this**.price;}

**public** Date getDateAdded() { **return** **this**.dateAdded;}

**public** **void** setID(**long** id) { **this**.ID = id;}

**public** **void** setName(String name) { **this**.name = name;}

**public** **void** setPrice(BigDecimal price) { **this**.price = price;}

**public** **void** setDateAdded(Date date) { **this**.dateAdded = date;}

}

Web.java

**package** com.keerthana.HibernateSpring;

**import** java.util.List;

**import** org.hibernate.SessionFactory;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Repository;

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

@Repository

**public** **class** web {

@Autowired

**private** SessionFactory sessionFactory;

@SuppressWarnings("unchecked")

**public** List<EProductEntity> getAllProducts() {

**return** **this**.sessionFactory.getCurrentSession().createQuery("from EProducts").list();

}

}

Controller.java

**package** com.keerthana.HibernateSpring;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.ModelMap;

**import** org.springframework.validation.BindingResult;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.bind.annotation.RequestParam;

@Controller

**public** **class** Controller {

@Autowired

**private** EProductService eproductService;

@RequestMapping(value = "/productList", method = RequestMethod.GET)

**public** String listProducts(ModelMap map)

{

map.addAttribute("eproduct", **new** EProductEntity());

map.addAttribute("productList", eproductService.getAllProducts());

**return** "productList";

}

}

Beans.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:aop=*"http://www.springframework.org/schema/aop"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:jee=*"http://www.springframework.org/schema/jee"*

xmlns:lang=*"http://www.springframework.org/schema/lang"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:tx=*"http://www.springframework.org/schema/tx"*

xmlns:util=*"http://www.springframework.org/schema/util"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/aop/ http://www.springframework.org/schema/aop/spring-aop.xsd*

*http://www.springframework.org/schema/context/ http://www.springframework.org/schema/context/spring-context.xsd*

*http://www.springframework.org/schema/jee/ http://www.springframework.org/schema/jee/spring-jee.xsd*

*http://www.springframework.org/schema/lang/ http://www.springframework.org/schema/lang/spring-lang.xsd*

*http://www.springframework.org/schema/tx/ http://www.springframework.org/schema/tx/spring-tx.xsd*

*http://www.springframework.org/schema/util/ http://www.springframework.org/schema/util/spring-util.xsd"*>

<context:annotation-config />

<context:component-scan base-package=*"com.ecommerce.controller"* />

<bean id=*"jspViewResolver"*

class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"viewClass"*

value=*"org.springframework.web.servlet.view.JstlView"*></property>

<property name=*"prefix"* value=*"/WEB-INF/view/"*></property>

<property name=*"suffix"* value=*".jsp"*></property>

</bean>

<bean id=*"messageSource"*

class=*"org.springframework.context.support.ReloadableResourceBundleMessageSource"*>

<property name=*"basename"* value=*"classpath:messages"*></property>

<property name=*"defaultEncoding"* value=*"UTF-8"*></property>

</bean>

<bean id=*"propertyConfigurer"*

class=*"org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"*

p:location=*"/WEB-INF/jdbc.properties"*></bean>

<bean id=*"dataSource"*

class=*"org.apache.commons.dbcp.BasicDataSource"* destroy-method=*"close"*

p:driverClassName=*"${jdbc.driverClassName}"*

p:url=*"${jdbc.databaseurl}"* p:username=*"${jdbc.username}"*

p:password=*"${jdbc.password}"*></bean>

<bean id=*"sessionFactory"*

class=*"org.springframework.orm.hibernate3.LocalSessionFactoryBean"*>

<property name=*"dataSource"* ref=*"dataSource"*></property>

<property name=*"configLocation"*>

<value>classpath:hibernate.cfg.xml</value>

</property>

<property name=*"configurationClass"*>

<value>org.hibernate.cfg.AnnotationConfiguration</value>

</property>

<property name=*"hibernateProperties"*>

<props>

<prop key=*"hibernate.dialect"*>${jdbc.dialect}</prop>

<prop key=*"hibernate.show\_sql"*>true</prop>

</props>

</property>

</bean>

<bean id=*"eproductDAO"* class=*"com.ecommerce.dao.EProductDAO"*></bean>

<bean id=*"eproductService"* class=*"com.ecommerce.service.EProductService"*></bean>

<tx:annotation-driven />

<bean id=*"transactionManager"*

class=*"org.springframework.orm.hibernate3.HibernateTransactionManager"*>

<property name=*"sessionFactory"* ref=*"sessionFactory"*></property>

</bean>

</beans>

Hs.html

<html>

<head>

<title>Spring With Hibernate</title>

</head>

<body>

<h3>Product List </h3>

<c:if test=*"${!empty productList}"*>

<table class=*"data"*>

<tr>

<th>Name</th>

<th>Price</th>

<th>Date Added</th>

</tr>

<c:forEach items=*"${productList}"* var=*"product"*>

<tr>

<td>${product.name} </td>

<td>${product.price}</td>

<td>${product.date\_added}</td>

</tr>

</c:forEach>

</table>

</c:if>

</body>

</html>

Spring.html

<html>

<body>

<h2>Spring With Hibernate</h2>

<a href=*"/productList"*>Product List</a>

</body>

</html>

HibernateConfig.xml

<?xml version=*'1.0'* encoding=*'utf-8'*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<mapping class=*"HibernateConfig.entity.EProductEntity"*></mapping>

</session-factory>

</hibernate-configuration>

Web.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app>

<display-name>Archetype Created Web Application</display-name>

<welcome-file-list>

<welcome-file>/WEB-INF/view/index.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>eproduct</servlet-name>

<servlet-class>

org.springframework.web.servlet.DispatcherServlet

</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>eproduct</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/eproduct-servlet.xml</param-value>

</context-param>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

</web-app>

28. Adding a new product in the database

AddServlet.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<servlet-mapping>

<servlet-name>ProdMaintAppServlet</servlet-name>

<url-pattern>/add</url-pattern>

</servlet-mapping>

New.java

**package** com.keerthana.Add;

**public** **class** New {

**public** **static** **void** insertProduct(Product p) {

ConnectionPool pool = ConnectionPool.getInstance();

Connection connection = pool.getConnection();

PreparedStatement ps = **null**;

ResultSet rs = **null**;

String query=" INSERT INTO productsMaintenance VALUES( '"+ p.getCode()+"','"+ p.getDescription()+"','"+ p.getPriceCurrencyFormat()+"' ) ";

**try** {

ps = connection.prepareStatement(query);

rs = ps.executeQuery();

}

**catch** (SQLException e) {

System.***err***.println(e);

} **finally** {

DBUtil.closeResultSet(rs);

DBUtil.closePreparedStatement(ps);

pool.freeConnection(connection);

}

}

}

Query.java

**package** com.keerthana.Add;

**public** **class** Query {

**public** **static** **void** main(String args[]) {

}

String query = "INSERT INTO productsMaintenance (Code, Description, PriceCurrencyFormat) VALUES (?, ?, ?)";

ps = conn.prepareStatement(query);

ps.setString (1, p.getCode());

ps.setString (2, p.getDescription());

ps.setString (3, p.getPriceCurrencyFormat());

ps.execute();

}

}

29. Creating a simple JSP page

Web.xml

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Simple JSP</title>

</head>

<body>

<%

out.println("<h2>My first JSP page.</h2>");

%>

</body>

</html>

30. JSP Implicit Objects

ImplicitObjects.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>JSP Implicit Objects</title>

</head>

<body>

<%

String responseCheck = request.getParameter("office");

if (responseCheck != null ) {

response.setStatus(response.SC\_MOVED\_TEMPORARILY);

response.setHeader("Location", "response-redirect.jsp?office=" + responseCheck);

}

String errorCheck = request.getParameter("error");

if (errorCheck != null ) {

int x = 0;

if (x == 0)

throw new RuntimeException("Exception has been raised");

}

%>

<%

int serverPort = request.getServerPort() ;

out.println("The Server is running on port " + String.valueOf(serverPort) + "<br>");

out.println("Servlet Name is " + config.getServletName() + "<br>");

out.println("Server Info:" + application.getServerInfo() + "<br>");

String pageName = page.toString();

out.println("The name of the page is " + pageName + "<br>");

pageContext.setAttribute("userid", "John Doe");

out.println("userId attribute from pageContext: " + pageContext.getAttribute("userid") + "<br>");

%>

<a href=*"index.jsp?office=head\_office"*>Show usage of response object</a><br>

<a href=*"index.jsp?error=1"*>Show usage of error object</a><br>

<%

if (response.containsHeader("Office"))

out.println("Current location is " + response.getHeader("Office"));

%>

Web.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Get Header Example</title>

</head>

<body>

<%

String office = request.getParameter("office");

if (office != null)

out.println("value of Office obtained :" + office + "<br>");

else

out.println("No value of Office found<br>");

%>

</body>

</html>

Handling.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Error Handling page</title>

</head>

<body>

<% exception.printStackTrace(response.getWriter()); %>

<hr>

An exception was generated. Details are above:<br>

</body>

</html>

31. JSP Directives

JSP.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Error Handling page</title>

</head>

<body>

<% exception.printStackTrace(response.getWriter()); %>

<hr>

An exception was generated. Details are above:<br>

</body>

</html>

Directives.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<% out.println("This is content from included file.<br><br>"); %>

</body>

</html>

32. JSP Action Tags

Action.jsp

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>JSP Action Tags</title>

</head>

<body>

Usage of useBean tag<br>

<jsp:useBean id=*"productBean"* class=*"com.ecommerce.ProductBean"* scope=*"session"*></jsp:useBean>

<jsp:setProperty property=*"productId"* name=*"productBean"* value=*"18791"*/>

<jsp:setProperty property=*"productName"* name=*"productBean"* value=*"Optical Wireless Mouse"*/>

<jsp:setProperty property=*"price"* name=*"productBean"* value=*"600.00"*/>

<a href=*"showbean.jsp"*>Access bean properties from another page</a><br>

<a href=*"forward.jsp"*>Use Forward action to go to Google</a><br>

<hr>

<jsp:text>

<![CDATA[This is my content.<br/>This will show within a text action tag exactly as it has been entered]]>

</jsp:text>

</body>

</html>

Tags.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<jsp:forward page=*"forwarded.jsp"*></jsp:forward>

</body>

</html>

Forwarded.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Forward Test</title>

</head>

<body>

This page came from forward.jsp by using jsp:forward action tag

</body>

</html>

Show.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Display Bean properties</title>

</head>

<body>

<jsp:useBean id=*"productBean"* class=*"com.JSPAction.ProductBean"* scope=*"session"*></jsp:useBean>

Product Id: <jsp:getProperty name=*"productBean"* property=*"productId"* /> <br>

Product Name: <jsp:getProperty name=*"productBean"* property=*"productName"* /> <br>

Product Price: <jsp:getProperty name=*"productBean"* property=*"price"* /> <br>

</body>

</html>

Product.java

**package** JSPAction;

**import** java.io.Serializable;

**public** **class** Product **implements** Serializable{

**private** String productId;

**private** String productName;

**private** **double** price;

**public** **void** setProductId(String value) {

**this**.productId = value;

}

**public** **void** setProductName(String value) {

**this**.productName = value;

}

**public** **void** setPrice(**double** value) {

**this**.price = value;

}

**public** String getProductId() {

**return** **this**.productId;

}

**public** String getProductName() {

**return** **this**.productName;

}

**public** **double** getPrice() {

**return** **this**.price;

}

}

33. Session Handling

Handling.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>JSP Sessions</title>

</head>

<body>

<%

**if** (request.getParameter("error") != **null**)

out.println("<b>Your session has expired or is invalid.</b><br>");

%>

<form action=*"Handling.jsp"* metho=*"post"*>

Name <input name=*"name"* id=*"name"* maxlength=*40*><br>

Password <input type=*"password"* name=*"pwd"* id=*"pwd"* maxlength=*"10"*><br>

<button>Submit</button>

</form>

</body>

</html>

Login.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"*%>

<%

String name = request.getParameter("name");

String pwd = request.getParameter("pwd");

**if** (name == **null** || name.equals("") || pwd == **null** || pwd.equals("")) {

response.sendRedirect("index.jsp?error=1");

} **else** {

session.setAttribute("name", name);

response.sendRedirect("dashboard.jsp");

}

%>

</body>

</html>

Dashboard.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Dashboard</title>

</head>

<body>

<%

**if** (session.getAttribute("name") == **null**) {

response.sendRedirect("index.jsp?error=1");

}

%>

<b>Hello <%= session.getAttribute("name") %></b><br>

<a href=*"logout.jsp"*>Click here to logout</a>

</body>

</html>

Logout.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Session Logout</title>

</head>

<body>

<%

session.invalidate();

%>

<b>Your session has terminated.</b><br>

<a href=*"index.jsp"*>Login again</a>

</body>

</html>