LabSheet 6a

**Hibernate Registration Form Example with JSP, Servlet, MySQL using Annotation**

Problem Statement : User registration processing using Hibernate Annotation

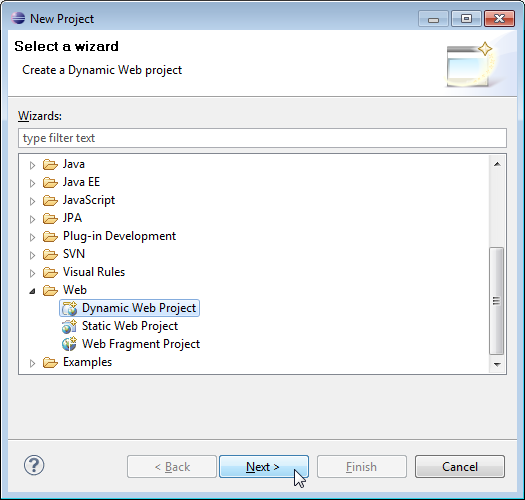
1. Creating a User Registration form using JSP  
2. Submit the User Registration form with a POST request  
3. After form submission the corresponding servlet will get called  
4. UserController class handles all the request parameters and sends a request to the UserDao class to save this data to the database.

# 1. Create a Dynamic Web Project in Eclipse IDE

To create a new dynamic Web project in Eclipse:

1. On the main menu select **File > New > Project...**

2. In the upcoming wizard choose **Web > Dynamic Web Project.**

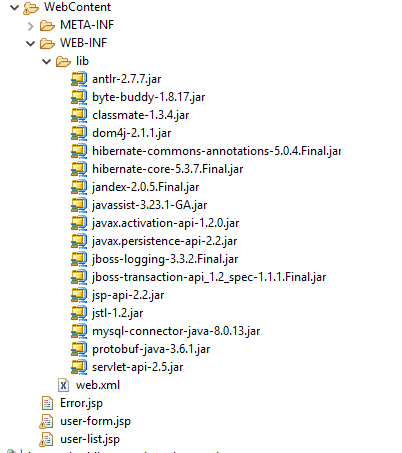


3. Click **Next**.

4. Enter project name as "hb1";  
5. Make sure that the target runtime is set to Apache Tomcat with the currently supported version.

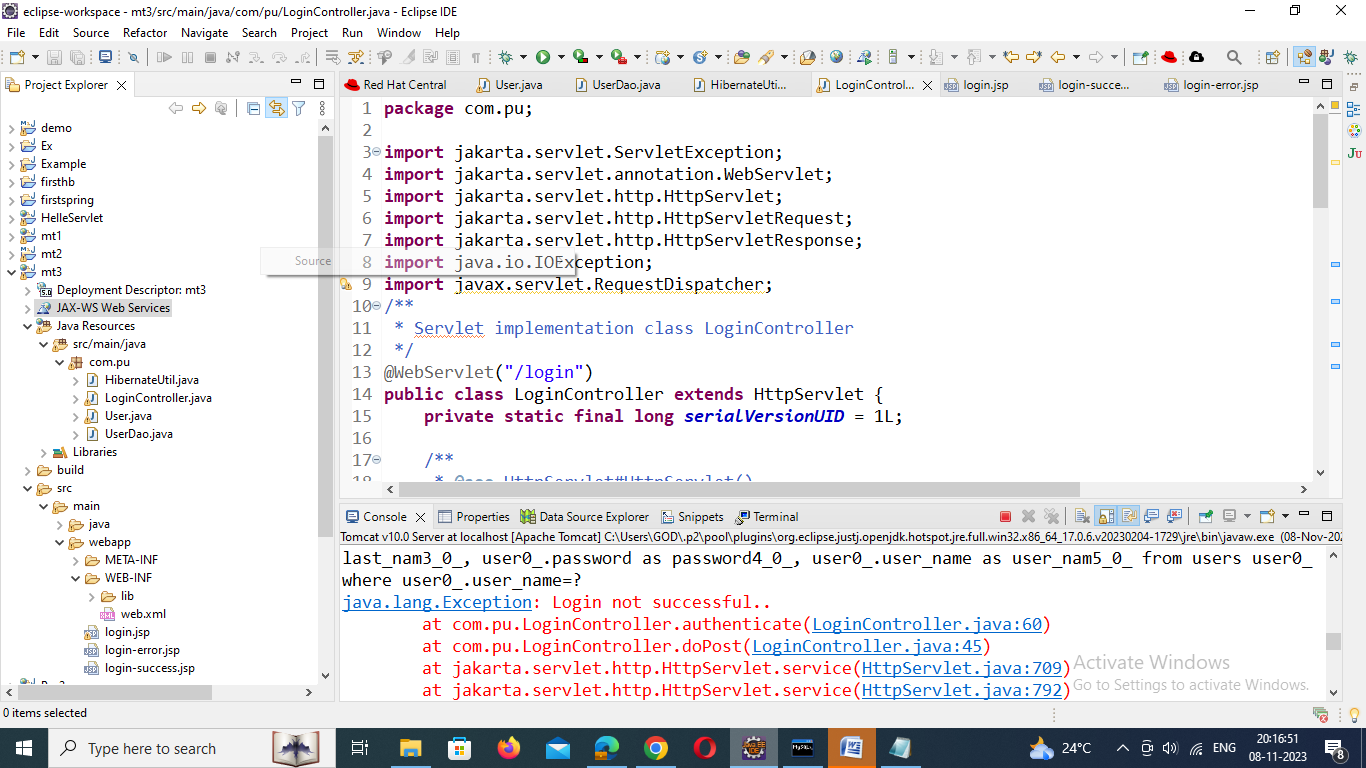
# 2. Add Jar Dependencies

You can get these jar dependencies from the GitHub repository (the link given at the end of this tutorial). Add the latest release of the below jar files to the lib folder:



# 3. Project Structure

Create project structure or packaging structure as per the below screenshot:

****

# 4. MySQL Database Setup

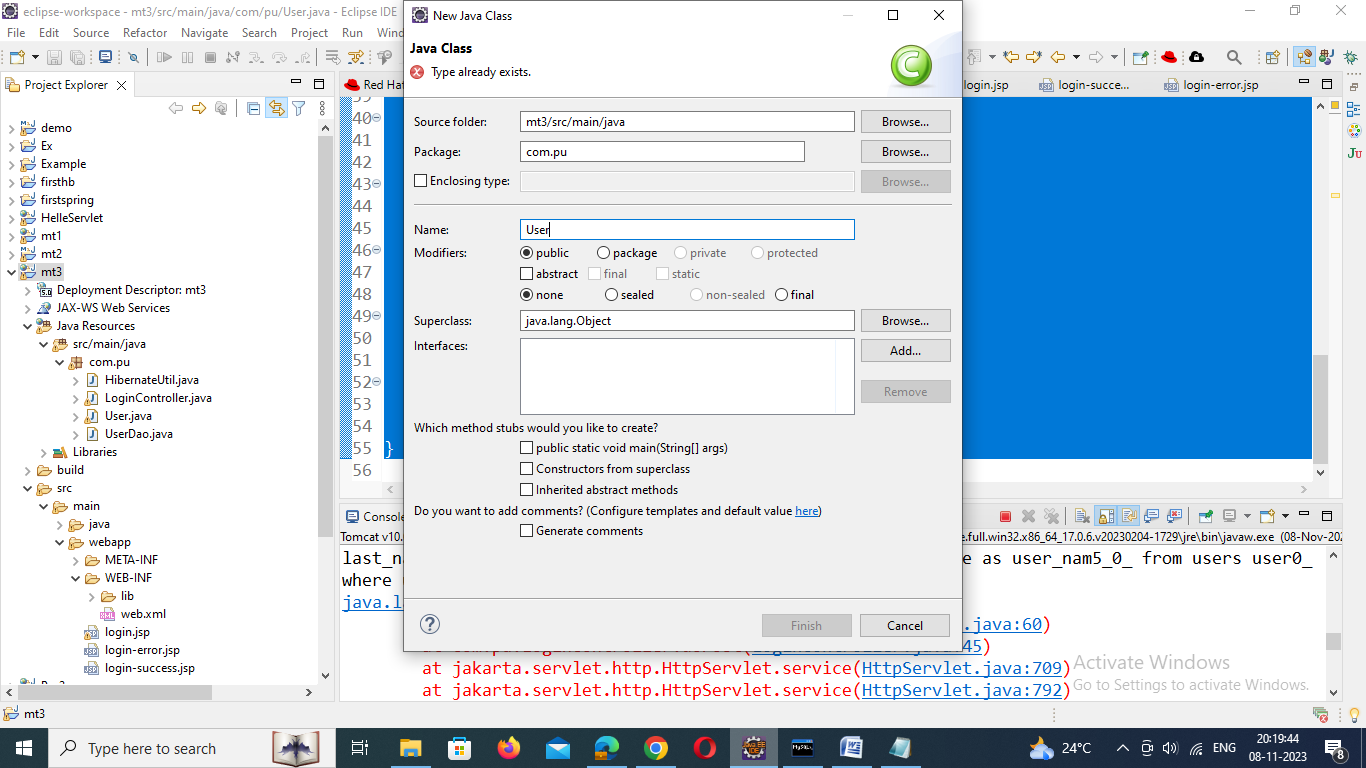
Let's create a database named "demo" in MySQL:

CREATE DATABASE 'demo';

**Note:**Hibernate will create **users**tables automatically. registered users so we will use the same **users**table to authenticate a User in this example.

# 5. Create a JPA Entity - User.java

Next, create User JPA Entity class and add the following content to it:



**User.java**

**package** com.pu;

**import** java.io.Serializable;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name = "users")

**public** **class** User **implements** Serializable {

@Id

@GeneratedValue(strategy = GenerationType.***AUTO***)

**private** **int** id;

@Column(name = "first\_name")

**private** String firstName;

@Column(name = "last\_name")

**private** String lastName;

@Column(name = "user\_name")

**private** String username;

@Column(name = "password")

**private** String password;

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

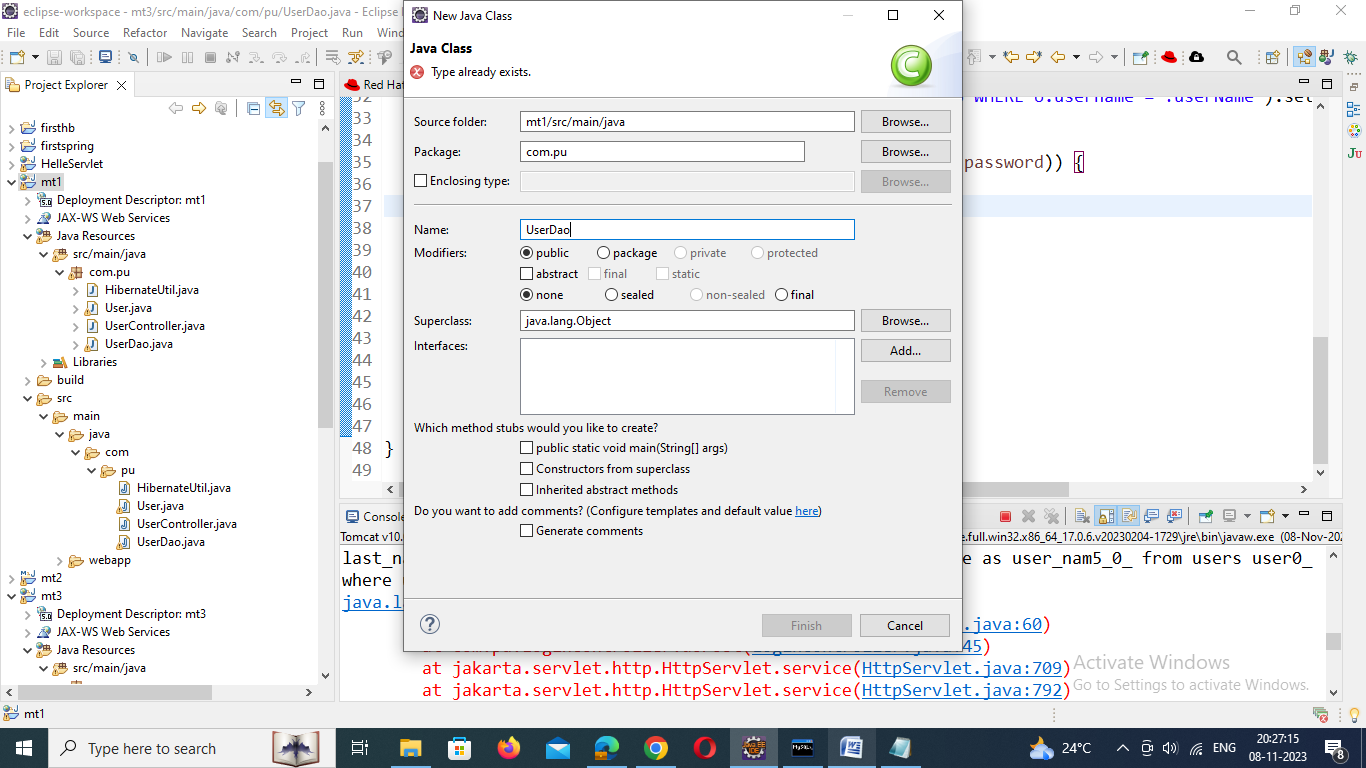
**this**.password = password;

}

}

# 6. Create UserDao to Save Registered User into Database

Let's create a *UserDao* class and add the *saveUser()* method to save Users into the *users* table in a database using Hibernate.



UserDao.java

**package com.pu;**

**import org.hibernate.Session;**

**import org.hibernate.Transaction;**

**public class UserDao {**

**public void saveUser(User user) {**

**Transaction transaction = null;**

**try (Session session = HibernateUtil.getSessionFactory().openSession()) {**

**// start a transaction**

**transaction = session.beginTransaction();**

**// save the student object**

**session.save(user);**

**// commit transaction**

**transaction.commit();**

**} catch (Exception e) {**

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

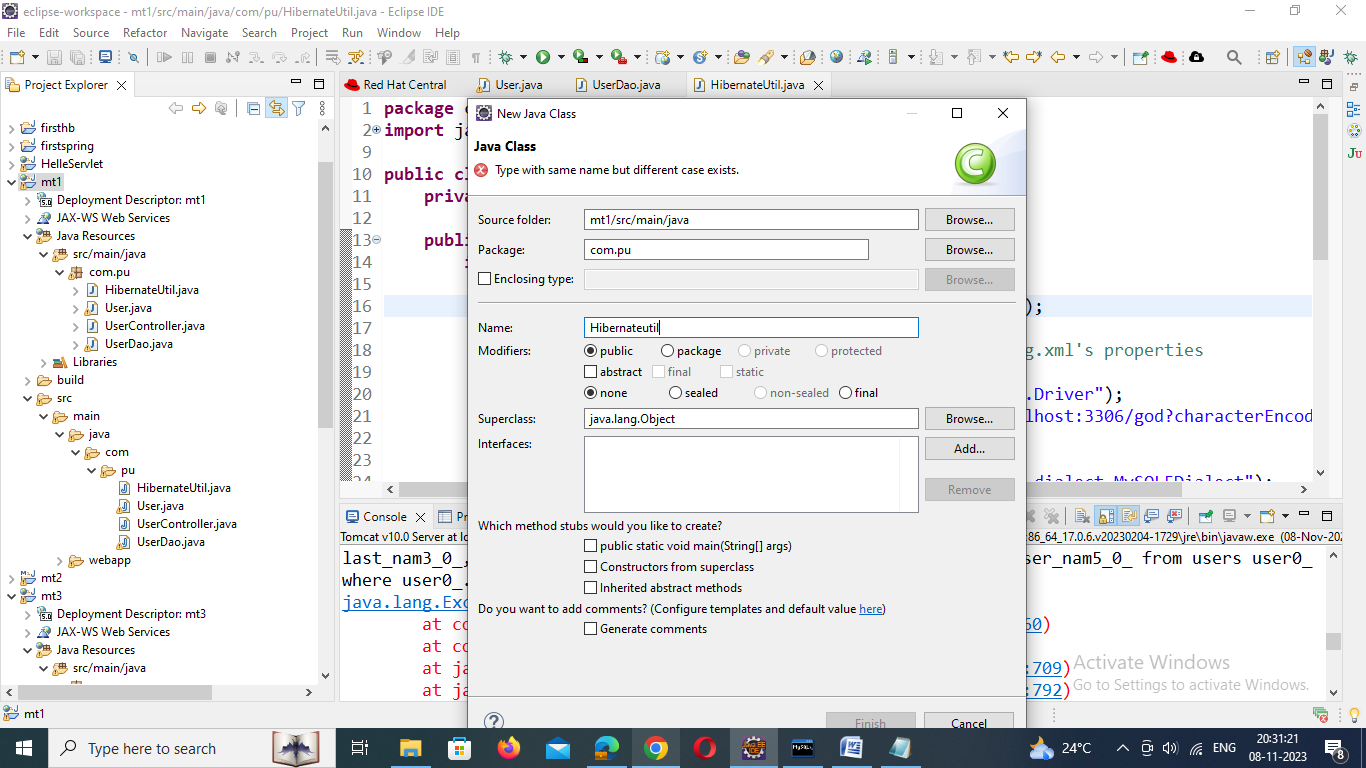
**transaction.rollback();**

**}**

**e.printStackTrace();**

**}**

**}**

****

# 7. Hibernate Java-Based Configuration

**HibernateUtil.java**

**package** com.pu;

**import** java.util.Properties;

**import** org.hibernate.SessionFactory;

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

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

**import** org.hibernate.cfg.Environment;

**import** org.hibernate.service.ServiceRegistry;

**public** **class** HibernateUtil {

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

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

**if** (*sessionFactory* == **null**) {

**try** {

Configuration configuration = **new** Configuration();

// Hibernate settings equivalent to hibernate.cfg.xml's properties

Properties settings = **new** Properties();

settings.put(Environment.***DRIVER***, "com.mysql.jdbc.Driver");

settings.put(Environment.***URL***, "jdbc:mysql://localhost:3306/god?characterEncoding=latin1");

settings.put(Environment.***USER***, "root");

settings.put(Environment.***PASS***, "admin123");

settings.put(Environment.***DIALECT***, "org.hibernate.dialect.MySQL5Dialect");

settings.put(Environment.***SHOW\_SQL***, "true");

settings.put(Environment.***CURRENT\_SESSION\_CONTEXT\_CLASS***, "thread");

settings.put(Environment.***HBM2DDL\_AUTO***, "create-drop");

configuration.setProperties(settings);

configuration.addAnnotatedClass(User.**class**);

ServiceRegistry serviceRegistry = **new** StandardServiceRegistryBuilder()

.applySettings(configuration.getProperties()).build();

System.***out***.println("Hibernate Java Config serviceRegistry created");

*sessionFactory* = configuration.buildSessionFactory(serviceRegistry);

**return** *sessionFactory*;

} **catch** (Exception e) {

e.printStackTrace();

}

}

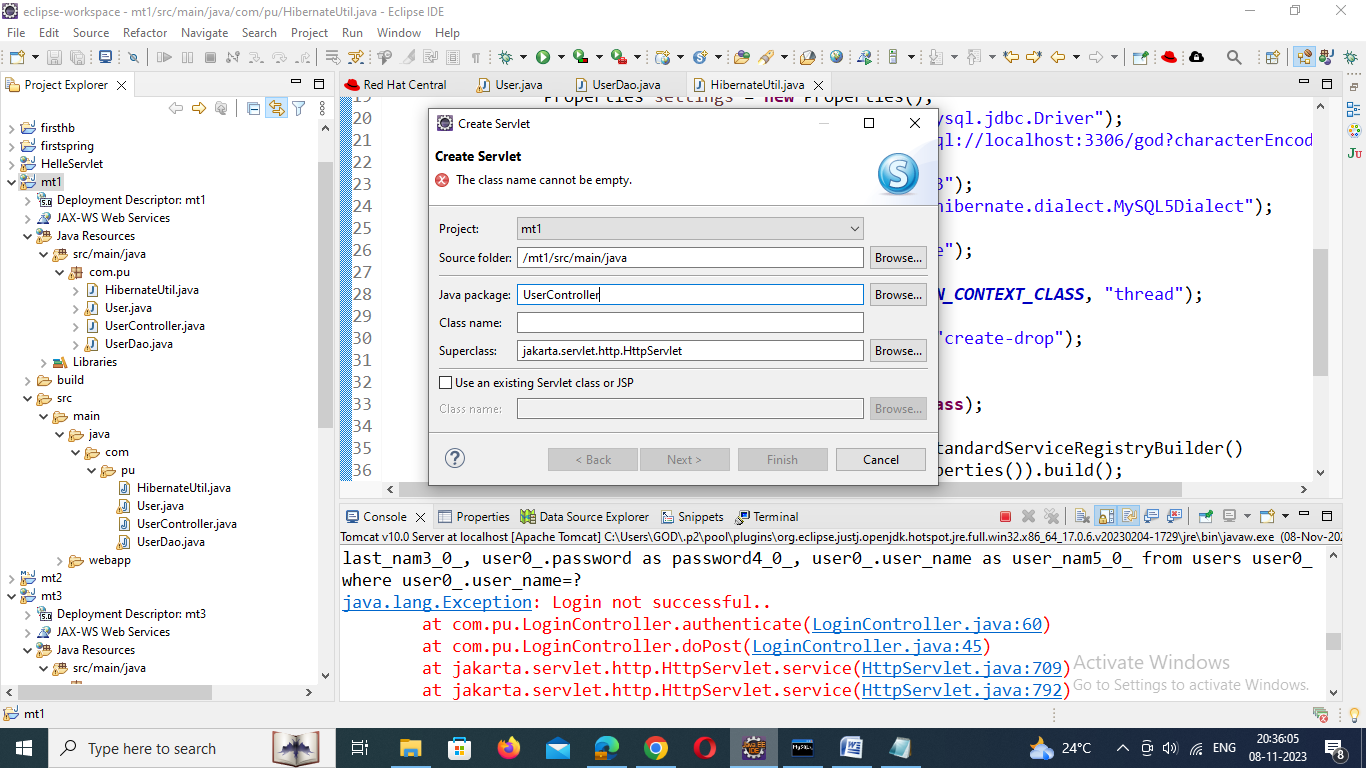
**return** *sessionFactory*;

}

}

# 8. Create a UserController.java

Now, let's create UserController (servlet) that acts as a page controller to handle all requests from the client:



**package** com.pu;

**import** jakarta.servlet.ServletException;

**import** jakarta.servlet.annotation.WebServlet;

**import** jakarta.servlet.http.HttpServlet;

**import** jakarta.servlet.http.HttpServletRequest;

**import** jakarta.servlet.http.HttpServletResponse;

**import** jakarta.servlet.RequestDispatcher;

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

/\*\*

\* Servlet implementation class UserController

\*/

@WebServlet("/register")

**public** **class** UserController **extends** HttpServlet {

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

**private** UserDao userDao;

/\*\*

\* **@see** HttpServlet#HttpServlet()

\*/

**public** UserController() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** **void** init() {

userDao = **new** UserDao();

}

/\*\*

\* **@see** HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

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

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

response.sendRedirect("register.jsp");

}

/\*\*

\* **@see** HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

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

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

register(request, response);

}

**private** **void** register(HttpServletRequest request, HttpServletResponse response) **throws** IOException, ServletException {

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

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

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

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

User user = **new** User();

user.setFirstName(firstName);

user.setLastName(lastName);

user.setUsername(username);

user.setPassword(password);

userDao.saveUser(user);

RequestDispatcher dispatcher = request.getRequestDispatcher("register-succes.jsp");

dispatcher.forward(request, response);

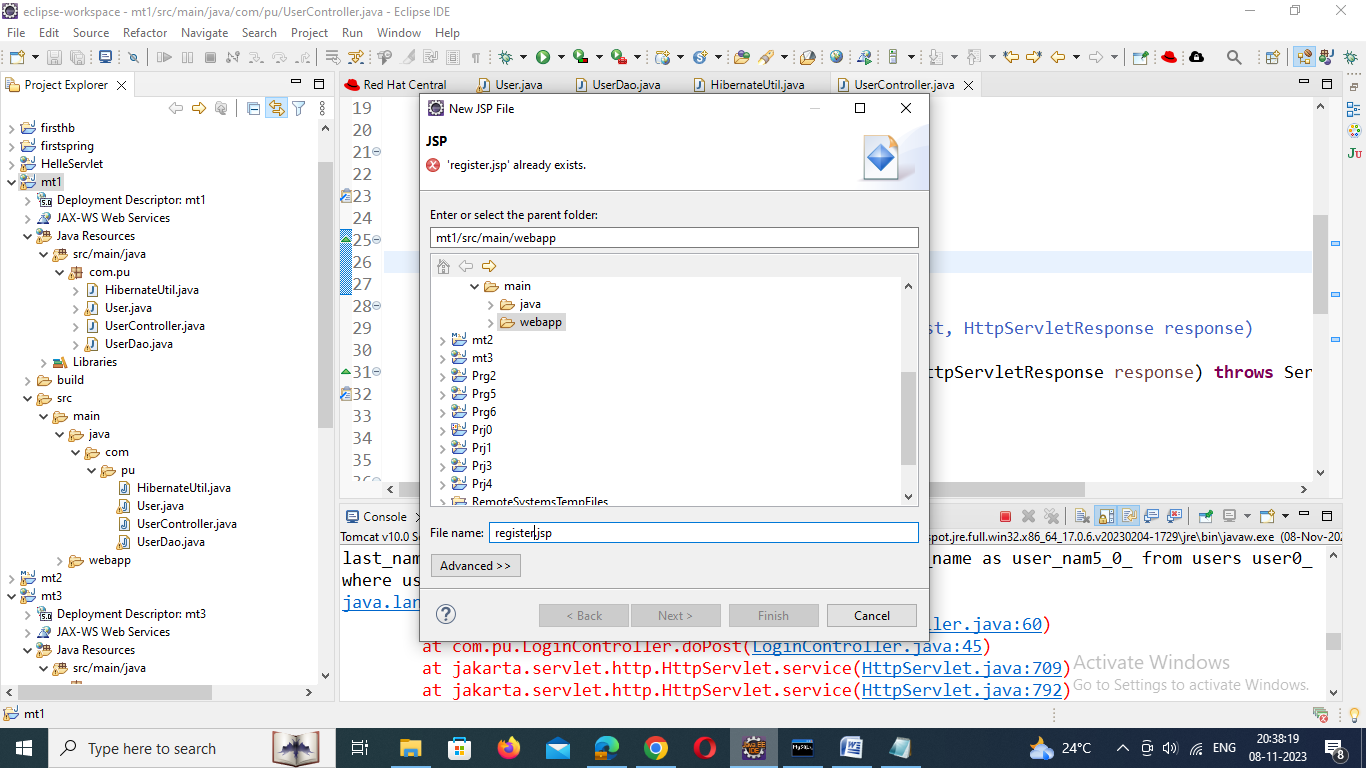
}

}

# 9. Create a View - register.jsp

Let's design a user registration HTML form with the following fields:

* firstName
* lastName
* username
* password



<%@ 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>

<link rel=*"stylesheet"*

href=*"https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"*

integrity=*"sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"*

crossorigin=*"anonymous"*>

</head>

<body>

<div class=*"container"*>

<div class=*"row text-center"* style="color: *tomato*;">

<h2>User Registration with JSP, Servlet and Hibernate</h2>

</div>

<hr>

<div class=*"row col-md-10 col-md-offset-3"*>

<div class=*"card card-body"*>

<h2>User Register Form</h2>

<div class=*"col-md-8 col-md-offset-3"*>

<form action=*"*<%=request.getContextPath()%>*/register"* method=*"post"*>

<div class=*"form-group"*>

<label for=*"uname"*>First Name:</label> <input type=*"text"*

class=*"form-control"* id=*"uname"* placeholder=*"First Name"*

name=*"firstName"* required>

</div>

<div class=*"form-group"*>

<label for=*"uname"*>Last Name:</label> <input type=*"text"*

class=*"form-control"* id=*"uname"* placeholder=*"last Name"*

name=*"lastName"* required>

</div>

<div class=*"form-group"*>

<label for=*"uname"*>User Name:</label> <input type=*"text"*

class=*"form-control"* id=*"username"* placeholder=*"User Name"*

name=*"username"* required>

</div>

<div class=*"form-group"*>

<label for=*"uname"*>Password:</label> <input type=*"password"*

class=*"form-control"* id=*"password"* placeholder=*"Password"*

name=*"password"* required>

</div>

<button type=*"submit"* class=*"btn btn-primary"*>Submit</button>

</form>

</div>

</div>

</div>

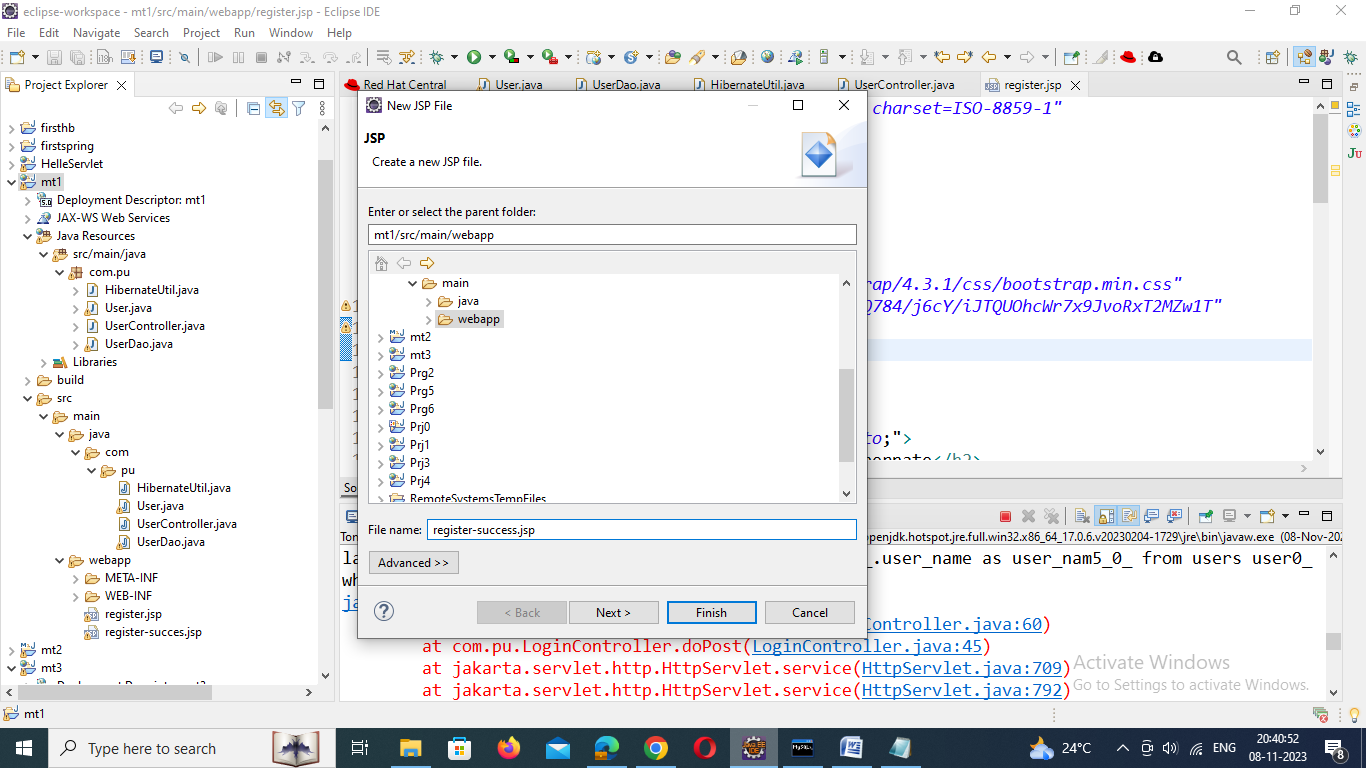
</div>

</body>

</html>

# 10. Create a View - register-success.jsp

Let's create a *register-success.jsp* page and add the following code to it:

****

<%@ 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>

<link rel=*"stylesheet"*

href=*"https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"*

integrity=*"sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"*

crossorigin=*"anonymous"*>

</head>

<body>

<div class=*"container"*>

<div class=*"row col-md-10 col-md-offset-3"*>

<div class=*"card card-body"*>

<h1>User successfully registered!</h1>

</div>

</div>

</div>

</body>

</html>