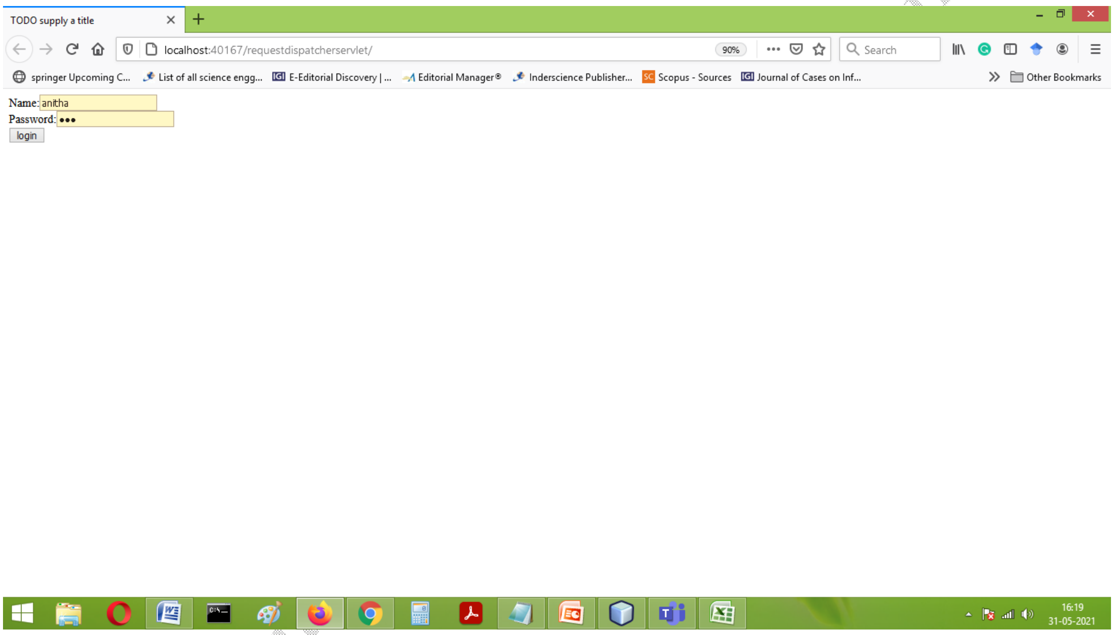
Manual for JAVA Servlet program

Steps for creating a Simple servlet program.

Let us explain the process of creating servlet program by creating a index page(homepage) for the design of the page. The index page is created as login page which takes the input from the user such as username and password. If the password is "aaa", then login page will be forwarded to the next page called welcomeservlet page which displays welcome message as welcome <username>. if the else the password is not correct, then it me be asking the user to enter the correct password.

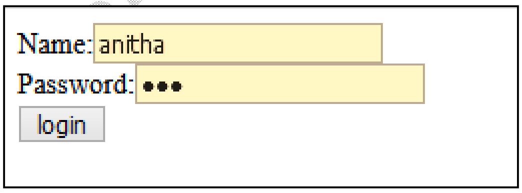
The output looks likes this:



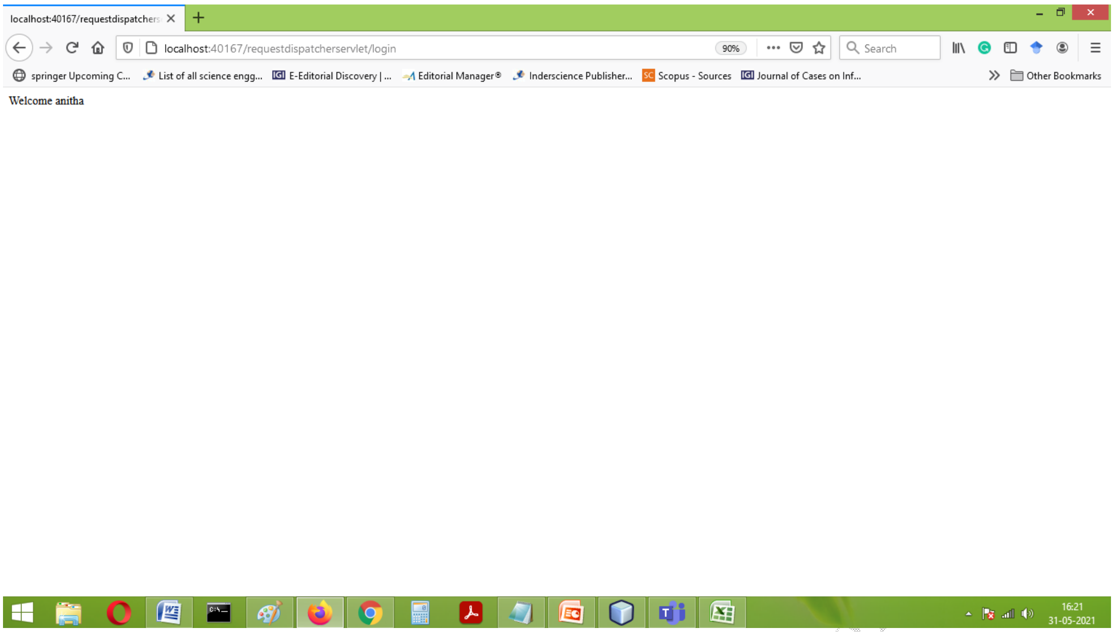
localhost:40167/requestdispatcherservlet/

Name: anitha
Password: ●●●
login

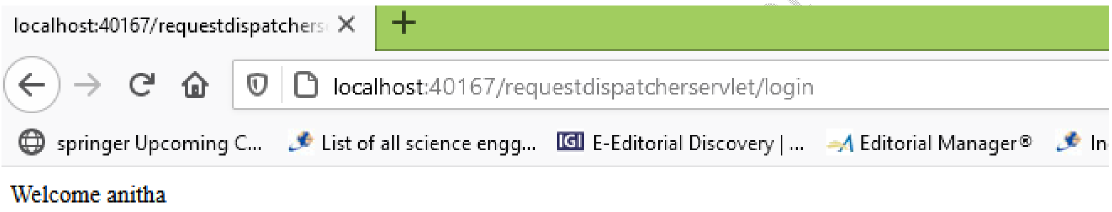
The output will be opened in the browser (Any browser you can select in the Properties -> Run - > browser



Name: anitha
Password: ●●●
login

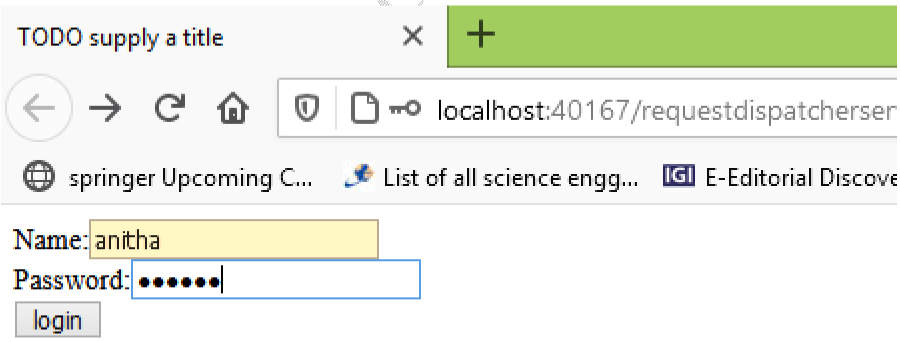

localhost:40167/requestdispatcher
localhost:40167/requestdispatcherservlet/login
springer Upcoming C...
List of all science engg...
E-Editorial Discovery | ...
Editorial Manager®
Inderscience Publisher...
Scopus - Sources
Journal of Cases on Inf...
Other Bookmarks
16:21
31-05-2021

Welcome anitha


localhost:40167/requestdispatchers
localhost:40167/requestdispatcherservlet/login
springer Upcoming C...
List of all science engg...
E-Editorial Discovery | ...
Editorial Manager®

Welcome anitha

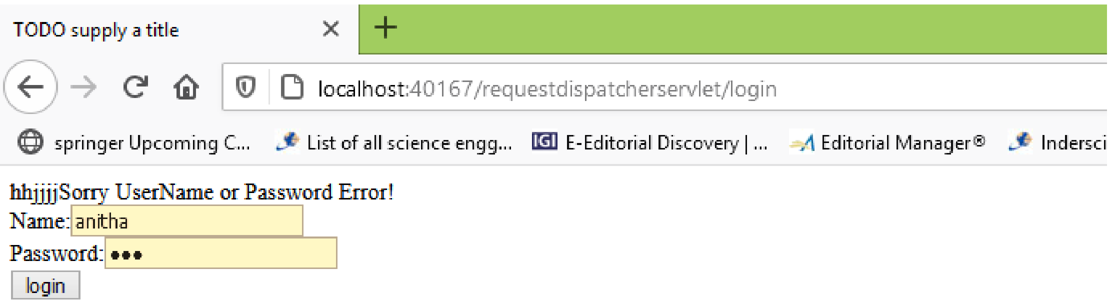
if the password is correct, the above screen will be moved else the homepage(index page) will be restored.


TODO supply a title
localhost:40167/requestdispatcherser
springer Upcoming C...
List of all science engg...
E-Editorial Discove

Name: anitha

Password: ••••••

login


TODO supply a title
localhost:40167/requestdispatcherservlet/login
springer Upcoming C...
List of all science engg...
E-Editorial Discovery | ...
Editorial Manager®
Indersci

hhjjjjSorry UserName or Password Error!

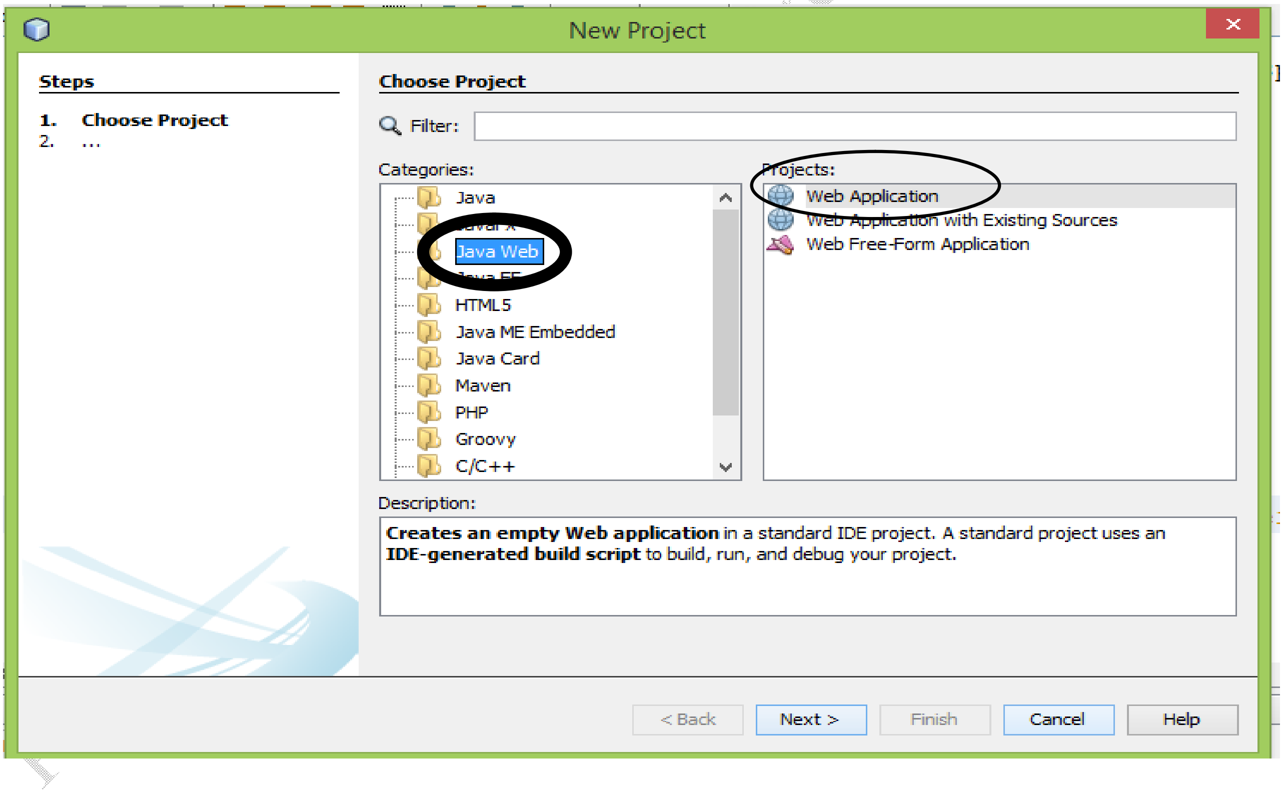
Name: anitha

Password: ●●●

login

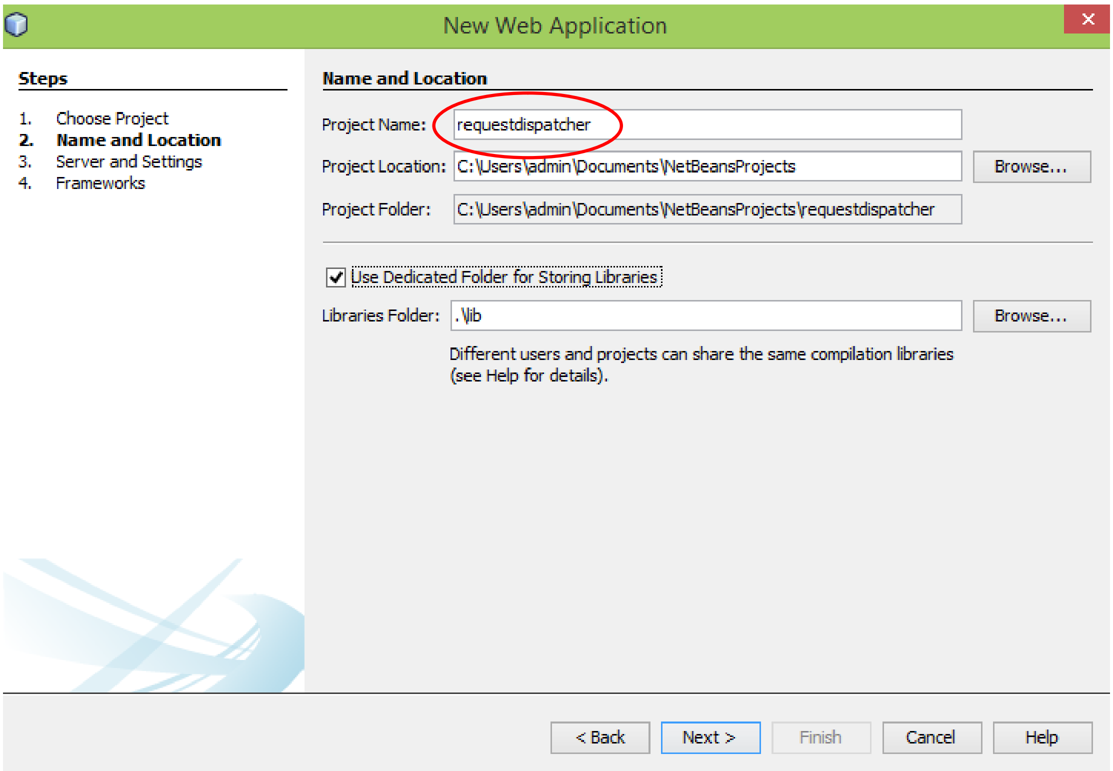
Students!!, Hope you might have gone through the output, let us see how to create this simple program

1. Go to netbeans
2. File - > new project



Creates an empty Web application in a standard IDE project. A standard project uses an **IDE-generated build script** to build, run, and debug your project.

2. Now click Next button



New Web Application

Steps

1. Choose Project
2. **Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name: requestdispatcher

Project Location: C:\Users\admin\Documents\NetBeansProjects

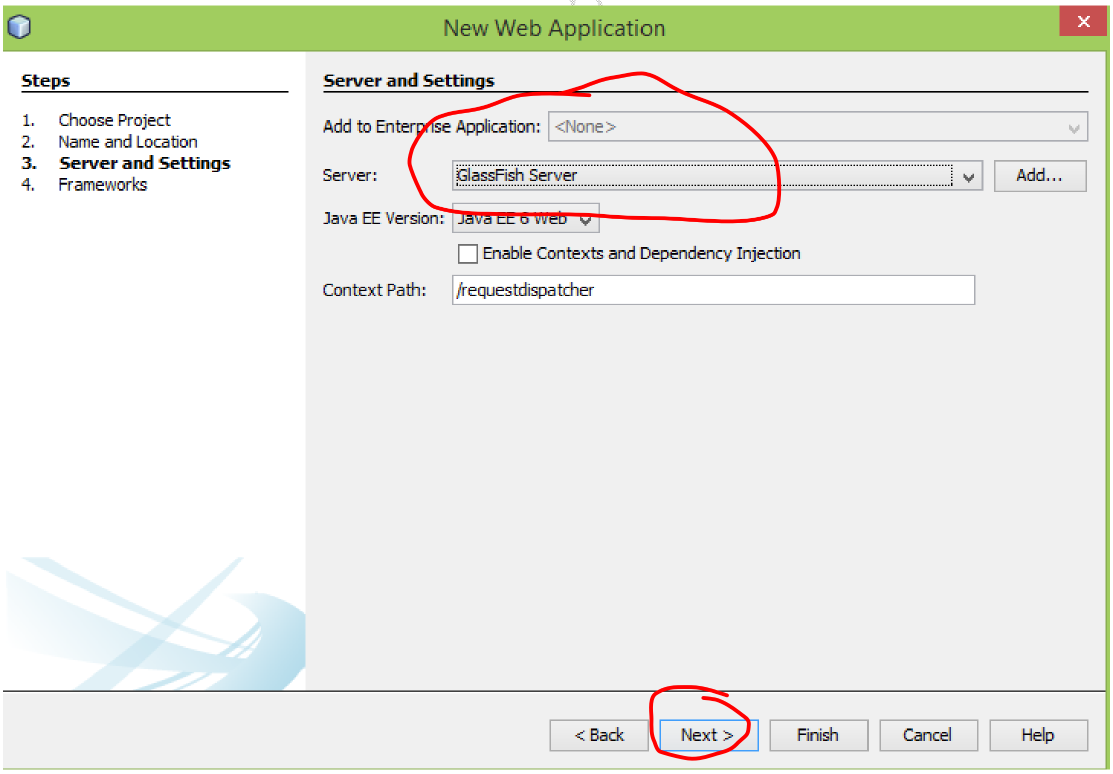
Project Folder: C:\Users\admin\Documents\NetBeansProjects\requestdispatcher

☑ Use Dedicated Folder for Storing Libraries

Libraries Folder: .\lib

Different users and projects can share the same compilation libraries
(see Help for details).

3. Now click next button



New Web Application

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**
4. Frameworks

Server and Settings

Add to Enterprise Application: <None>

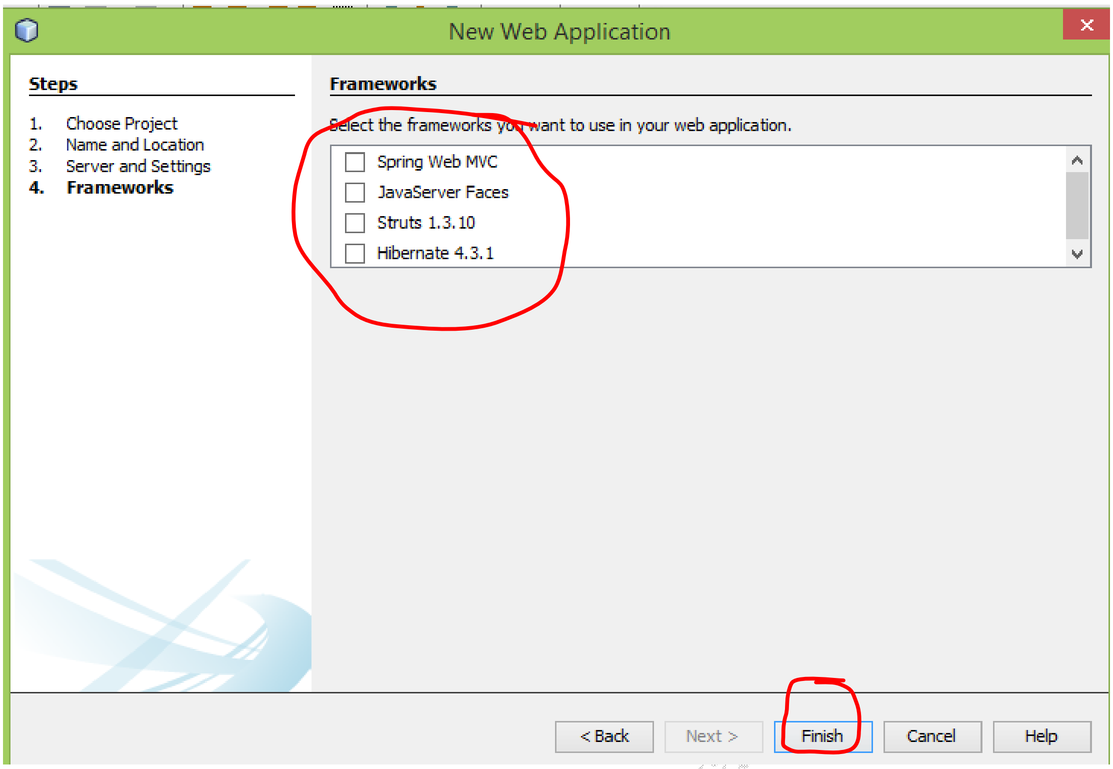
Server: GlassFish Server

Java EE Version: Java EE 6 Web

□ Enable Contexts and Dependency Injection

Context Path: /requestdispatcher

4. now click next..


New Web Application

Steps

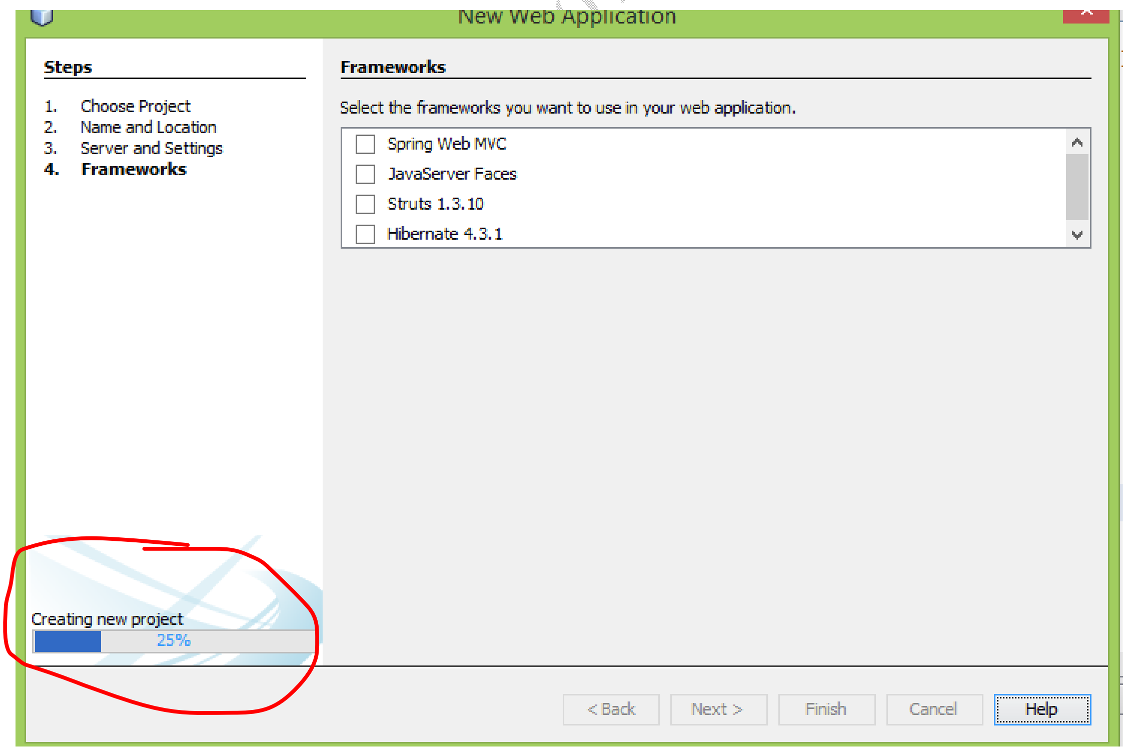
1. Choose Project
2. Name and Location
3. Server and Settings
4. **Frameworks**

Frameworks

Select the frameworks you want to use in your web application.

□ Spring Web MVC
□ JavaServer Faces
□ Struts 1.3.10
□ Hibernate 4.3.1

5. once the finish button pressed, we can see the project gets created


New Web Application

Steps

1. Choose Project
2. Name and Location
3. Server and Settings
4. **Frameworks**

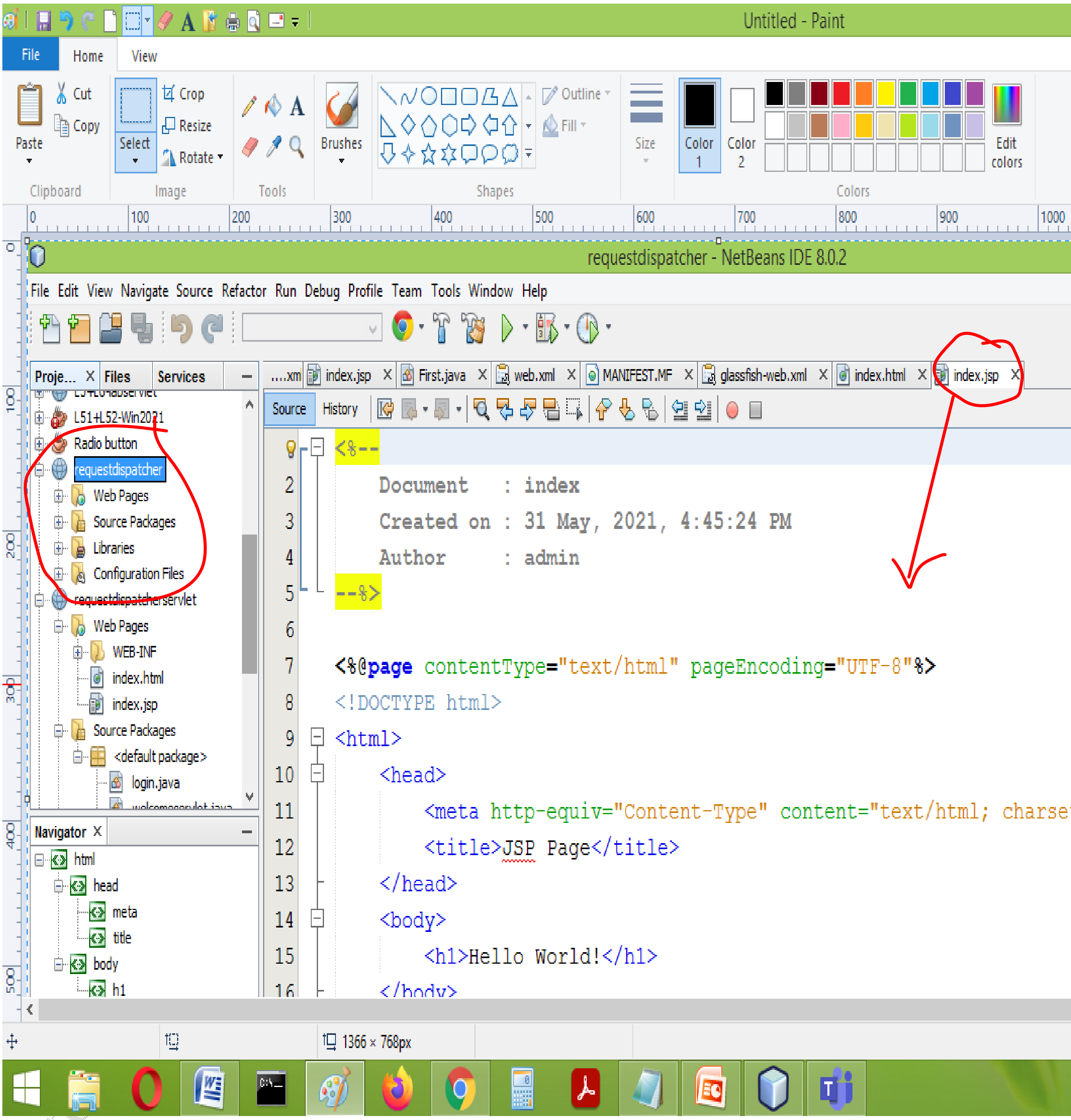
Frameworks

Select the frameworks you want to use in your web application.

□ Spring Web MVC
□ JavaServer Faces
□ Struts 1.3.10
□ Hibernate 4.3.1

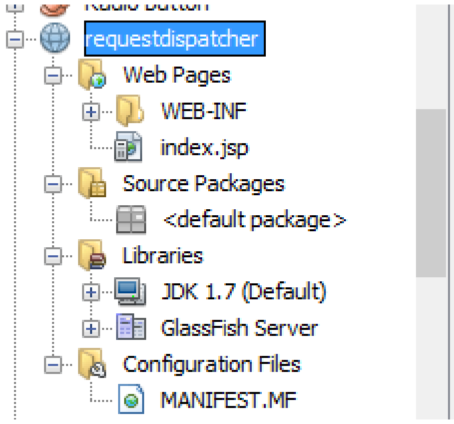
Creating new project
25%

6. we can see the project created with the name as requestdispatcher



```
<%--
    Document   : index
    Created on : 31 May, 2021, 4:45:24 PM
    Author     : admin
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charse
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Hello World!</h1>
    </body>
```

When the folders are expanded we can see like this



requestdispatcher
Web Pages
WEB-INF
index.jsp
Source Packages
<default package>
Libraries
JDK 1.7 (Default)
GlassFish Server
Configuration Files
MANIFEST.MF

for doing our program we have to create three files
1. index .html - this is html file act as homepage
2. login.java - this is servlet file
3.welcome.java - this is again the servlet file

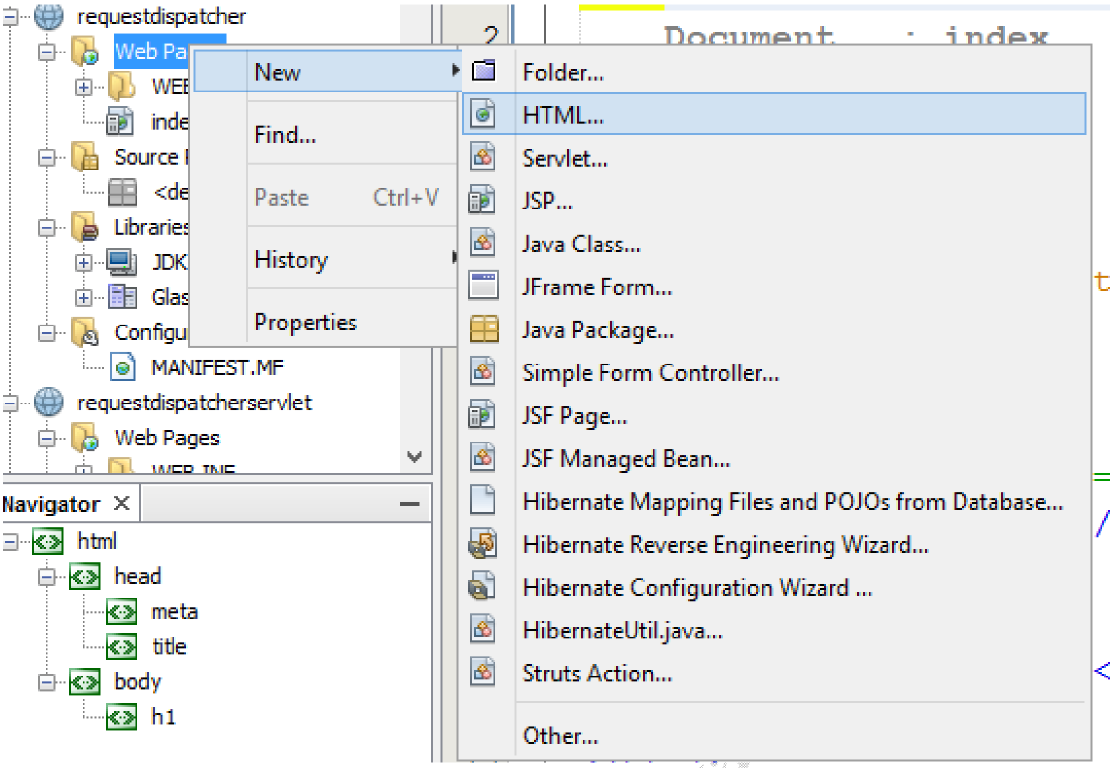
this movement of data is from

index.html---> login.java--->welcome.java (if password matches)

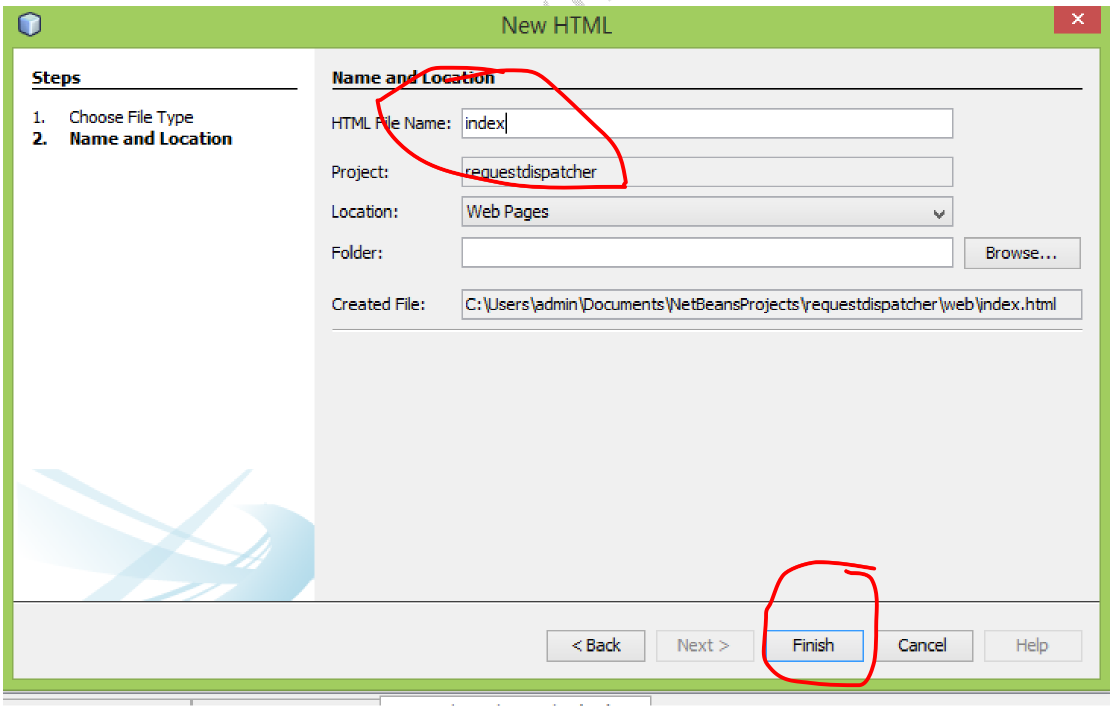
else

index.htmk --->login.java --->index.html(if passward doesnot matches, requesting user to give the correct password).

7. To create index.html, follow the below steps



we can see the below screen..


New HTML

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

HTML File Name: index

Project: requestdispatcher

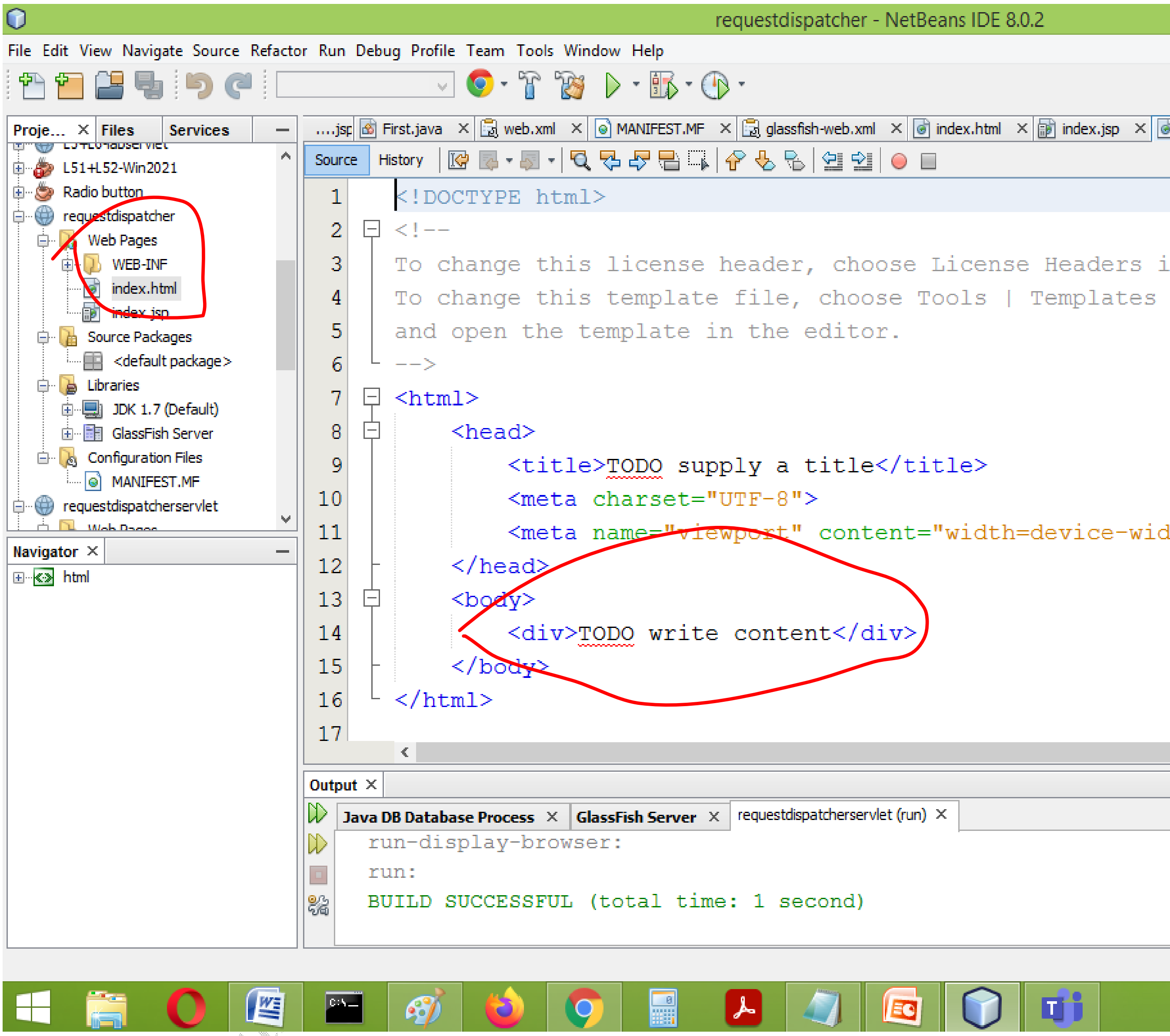
Location: Web Pages

Folder:

Created File: C:\Users\admin\Documents\NetBeansProjects\requestdispatcher\web\index.html

in this screen newhtml (default filename) should be changed as index, then click finish button

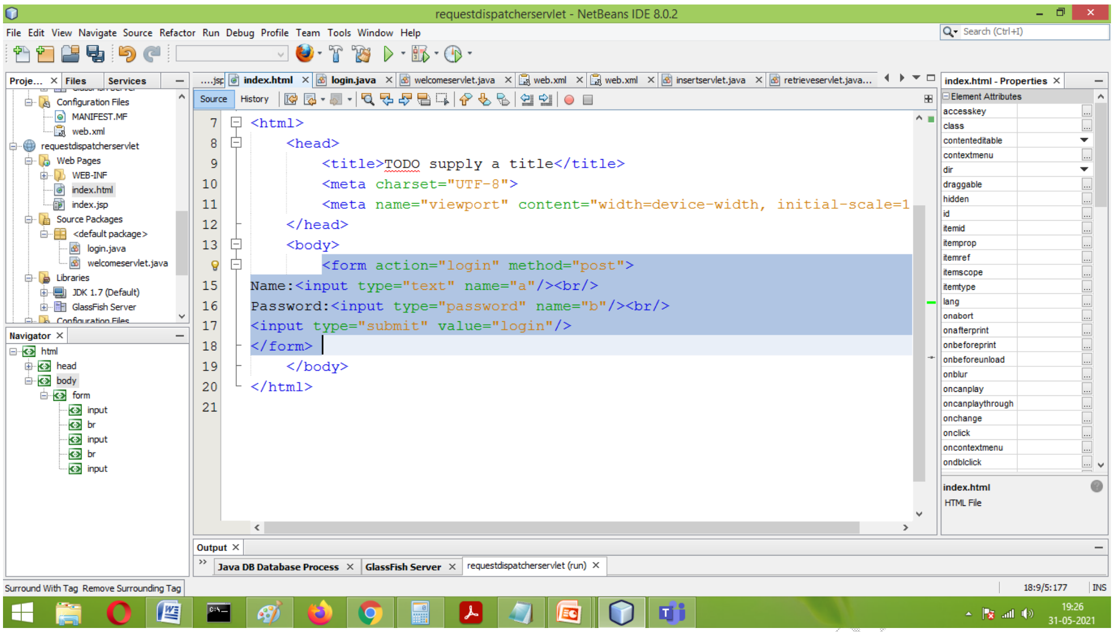
we can see the screen as below



```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers i
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-wid
    </head>
    <body>
        <div>TODO write content</div>
    </body>
</html>
```

```
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)
```

now in place of <div> delete it an replace the html coding of form creation

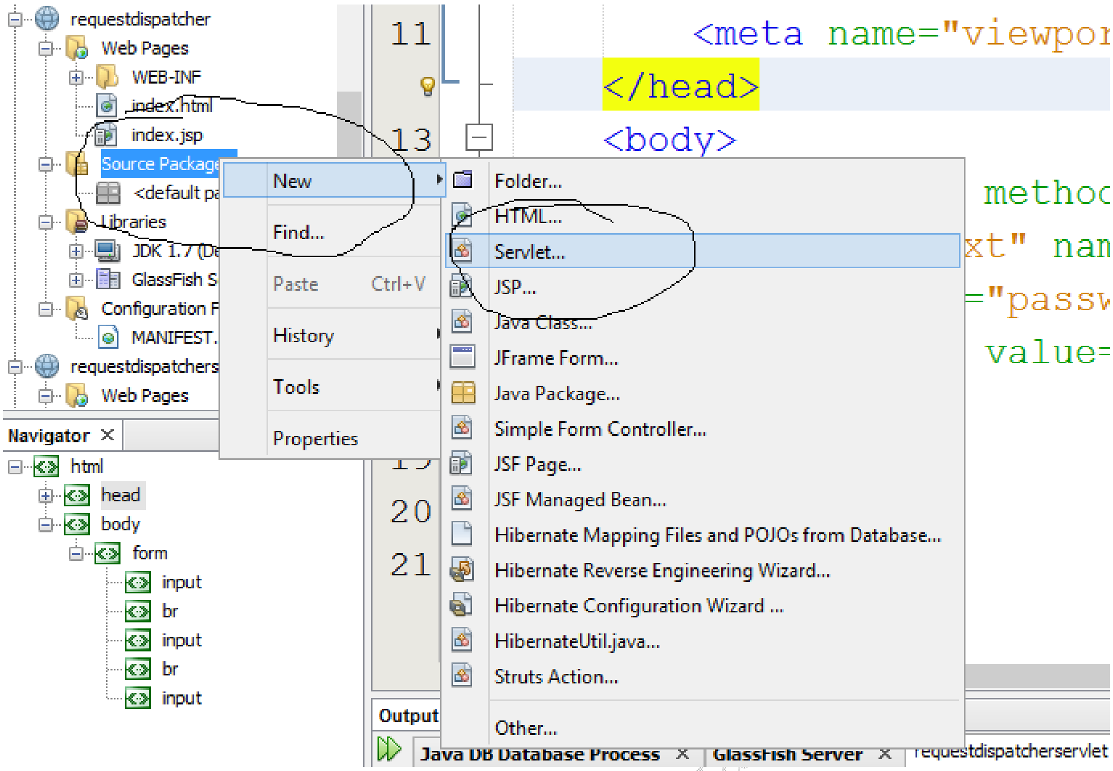


```
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1
    </head>
    <body>
        <form action="login" method="post">
Name:<input type="text" name="a"/><br/>
Password:<input type="password" name="b"/><br/>
<input type="submit" value="login"/>
</form>
    </body>
</html>
```

index.html

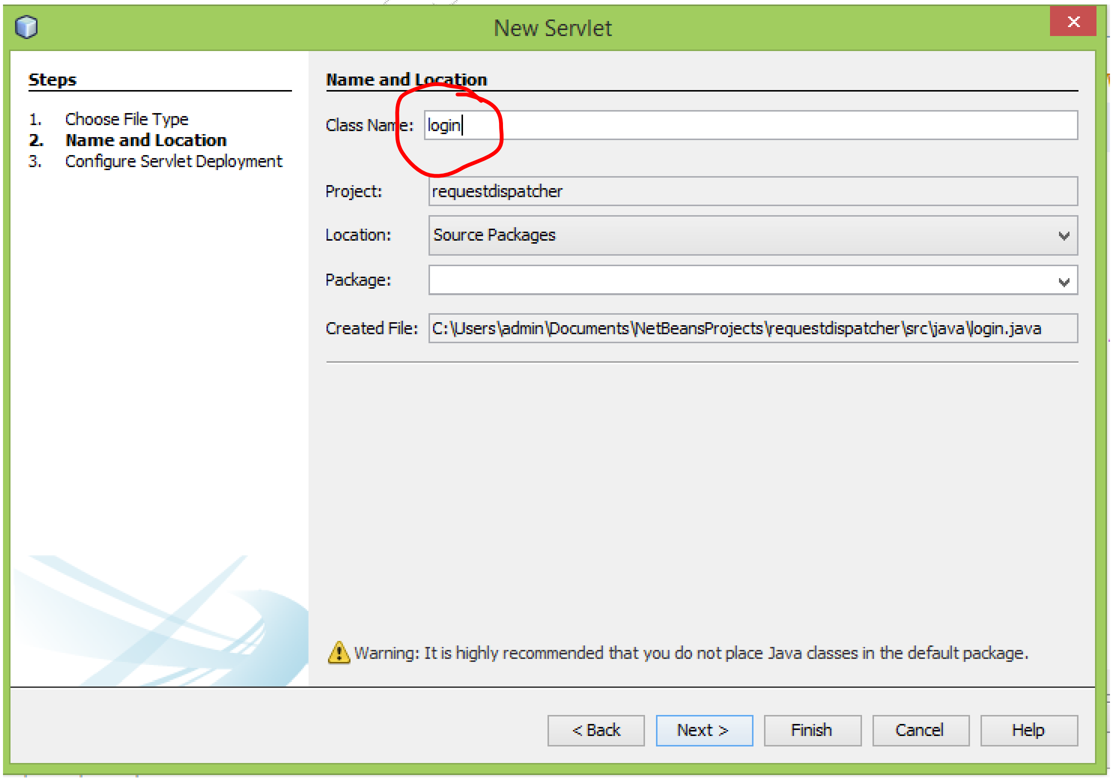
```
<form action="login" method="post">
Name:<input type="text" name="a"/><br/>
Password:<input type="password" name="b"/><br/>
<input type="submit" value="login"/>
</form>
```

8. Now it is time to create servelt - login



```
11      <meta name="viewpor
        </head>
13      <body>
                                  methoc
                                  xt" nan
                                  ="passw
                                  value=
20
21
```

Now we can see the screen as below , rename the file name from newservlet to login... and click finish button..the servlet file will be created under the source packages folder.



New Servlet

Steps

1. Choose File Type
2. **Name and Location**
3. Configure Servlet Deployment

Name and Location

Class Name: login

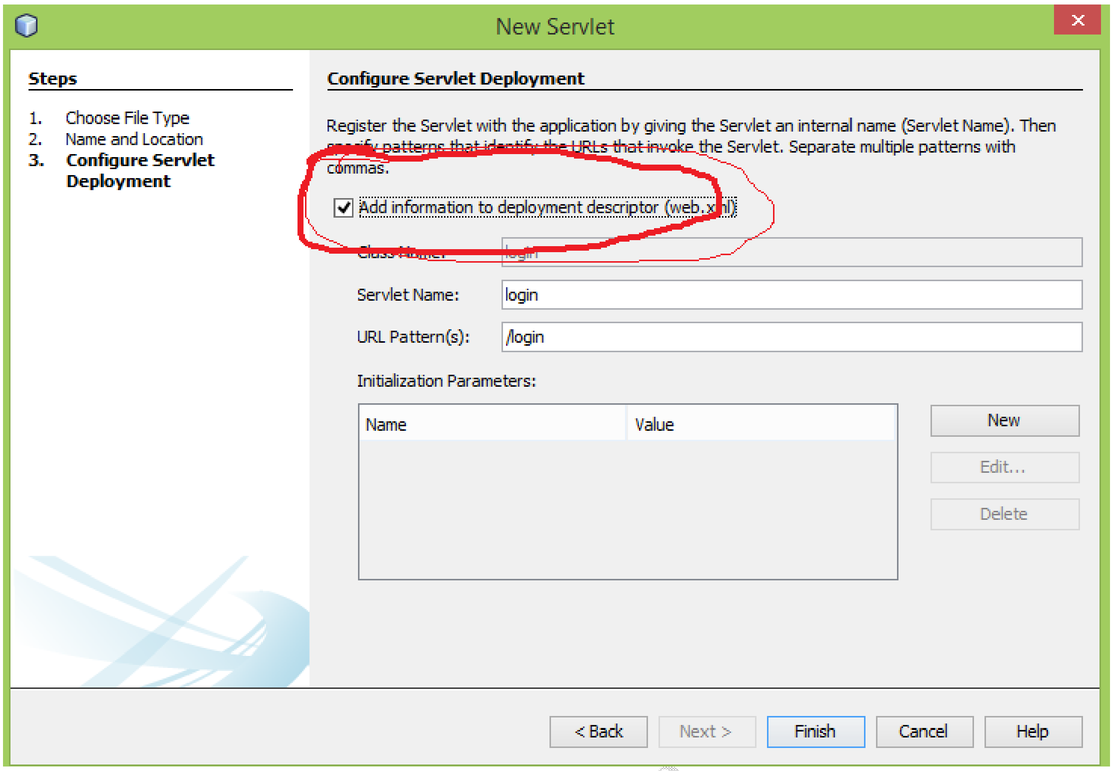
Project: requestdispatcher

Location: Source Packages

Package:

Created File: C:\Users\admin\Documents\NetBeansProjects\requestdispatcher\src\java\login.java

⚠ Warning: It is highly recommended that you do not place Java classes in the default package.



New Servlet

Steps

1. Choose File Type
2. Name and Location
3. **Configure Servlet Deployment**

Configure Servlet Deployment

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☑ Add information to deployment descriptor (web.xml)

Class Name: login

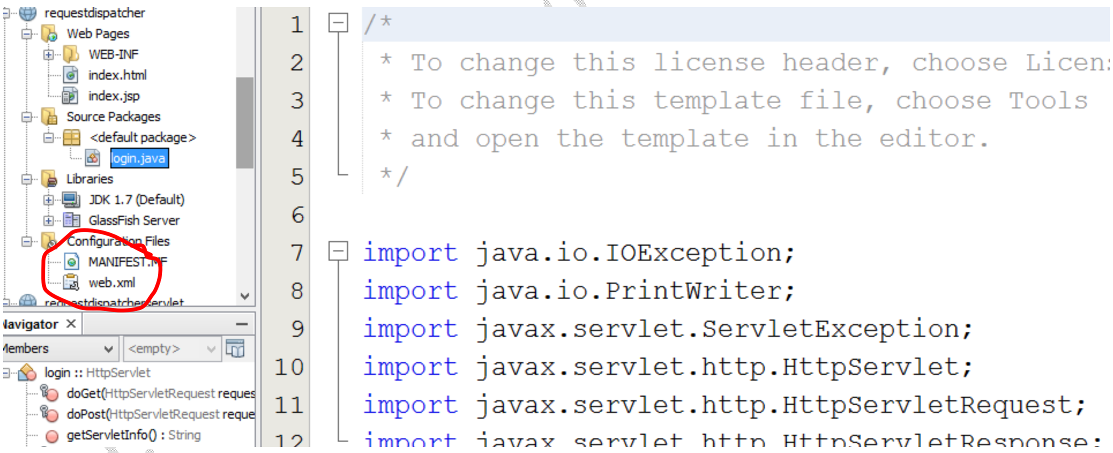
Servlet Name: login

URL Pattern(s): /login

Initialization Parameters:

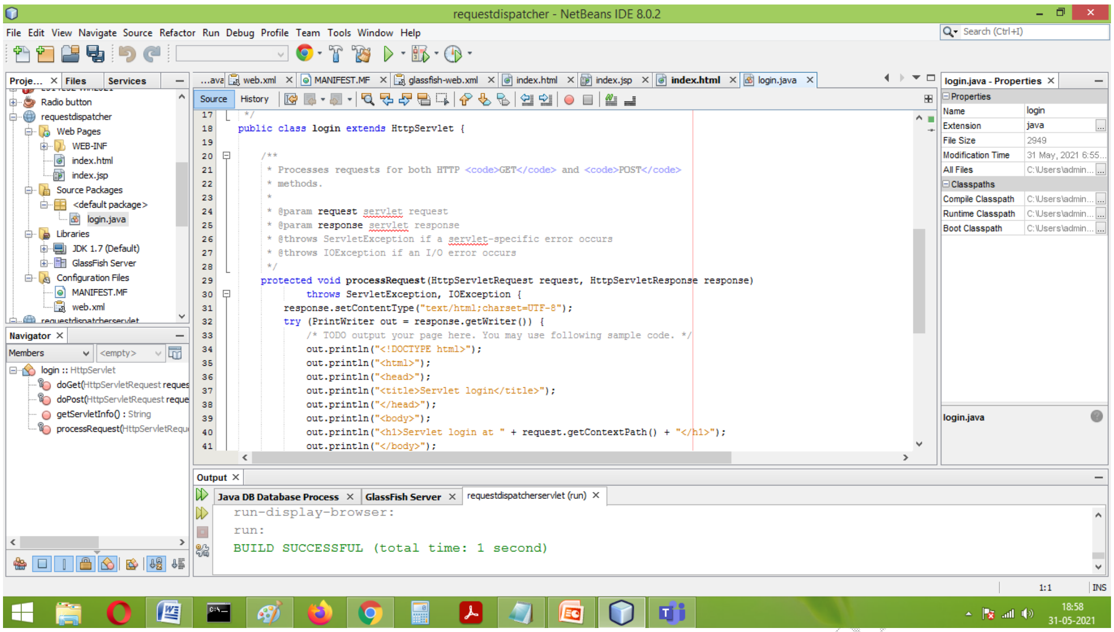
Name	Value

now clikc finish..now we can see the web.xml file created in the config package



```
/*
 * To change this license header, choose Licen
 * To change this template file, choose Tools
 * and open the template in the editor.
 */

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;
```

Also the login.java (servlet file) is created..
login.java will have the coding as given below



```
 */
public class login extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet login</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet login at " + request.getContextPath() + "</h1>");
            out.println("</body>");
```

Properties	
Name	login
Extension	java
File Size	2949
Modification Time	31 May, 2021 6:55...
All Files	C:\Users\admin...
Classpaths	
Compile Classpath	C:\Users\admin...
Runtime Classpath	C:\Users\admin...
Boot Classpath	C:\Users\admin...

login.java

```
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

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;

/**
 *
 * @author admin
 */
public class login extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and
<code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
```

Netbeans default uses the HTTPServlet as it class to extend, so that to use secure request and response operations

These are the request and response methods which helps for the data movement

```
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet login</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Servlet login at " + request.getContextPath() +
"</h1>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on
the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpSer
response)
            throws ServletException, IOException {
        processRequest(request, response);
```

if we want to use the get method then the coding have to be written here

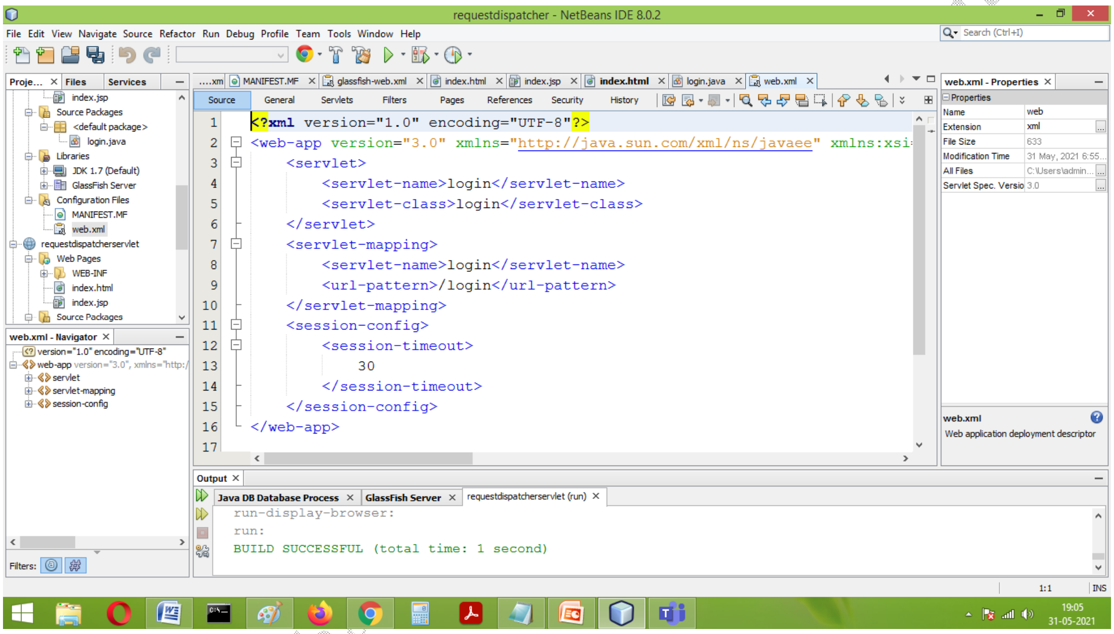
```
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        processRequest(request, response);
```

if we want to use the post method then the coding have to be written here

```
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

Now let us have the look on web.xml

web.xml



```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi
    <servlet>
        <servlet-name>login</servlet-name>
        <servlet-class>login</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>login</servlet-name>
        <url-pattern>/login</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
</web-app>
```

Properties	
Name	web
Extension	xml
File Size	633
Modification Time	31 May, 2021 6:55...
All Files	C:\Users\admin...
Servlet Spec. Versio	3.0

web.xml
Web application deployment descriptor

```
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)
```

web.xml repeated here

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">

    <servlet>
        <servlet-name>login</servlet-name>
        <servlet-class>login</servlet-class>
    </servlet>
```

This is the servlet which we created with name login and the url pattern also identifed as login (default), if want we can change, but should be represented in <form action="___" properly.

```
   <servlet-mapping>
        <servlet-name>login</servlet-name>
        <url-pattern>/login</url-pattern>
    </servlet-mapping>
```

Servlet mapping specifies the web container of which java servlet should be invoked for a url given by client.

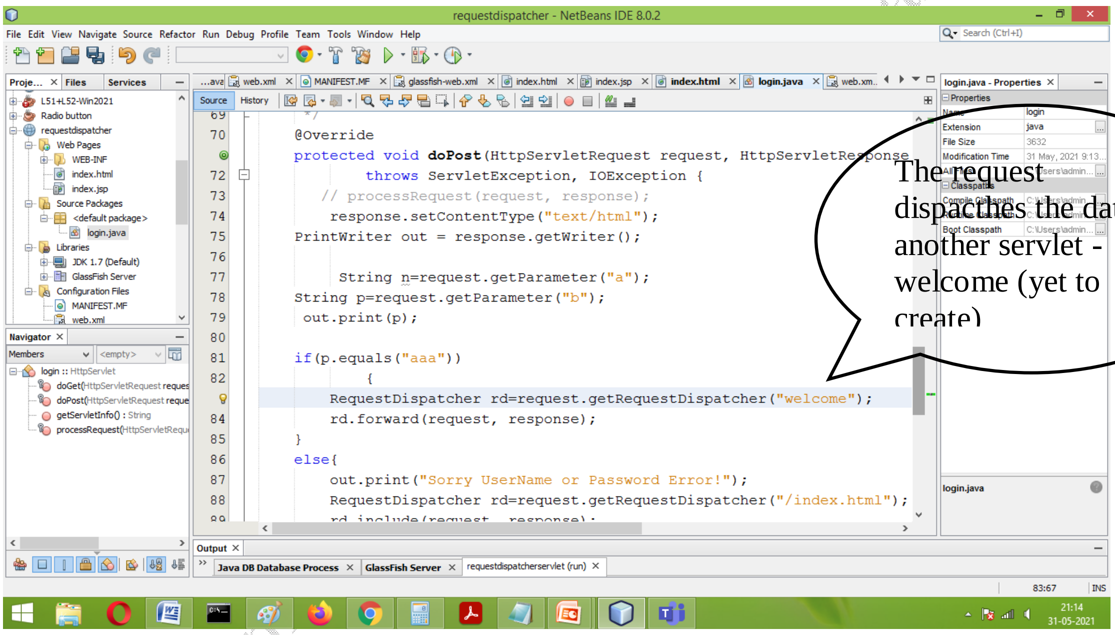
This object can be used to get initialization parameter for a particular web page

```
<session-config>
    <session-timeout>
        30
    </session-timeout>
</session-config>
</web-app>
```

The **session-timeout** tells us the timeout interval for all the sessions, default it will be 30 minutes, we can change as per out requirement

Now let us see the coding for login.java

since in the index.html we have given the method as post then we have to write our coding under dopost method as given below:



```
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
            throws ServletException, IOException {
       // processRequest(request, response);
        response.setContentType("text/html");
    PrintWriter out = response.getWriter();

         String n=request.getParameter("a");
    String p=request.getParameter("b");
     out.print(p);

    if(p.equals("aaa"))
            {
        RequestDispatcher rd=request.getRequestDispatcher("welcome");
        rd.forward(request, response);
    }
    else{
        out.print("Sorry UserName or Password Error!");
        RequestDispatcher rd=request.getRequestDispatcher("/index.html");
        rd.include(request, response);
```

The request dispaches the data to another servlet - welcome (yet to create)

the coding is as follows:

response.setContentType("text/html"); // content type either as html or msword or //jpg etc

PrintWriter out = response.getWriter(); // to write the script on the webpage using //out object

String n=request.getParameter("a"); //method to get the variable value //from the index.html (variable - //username

String p=request.getParameter("b"); // the variable b is used to get the password //from index.html

//out.print(p);

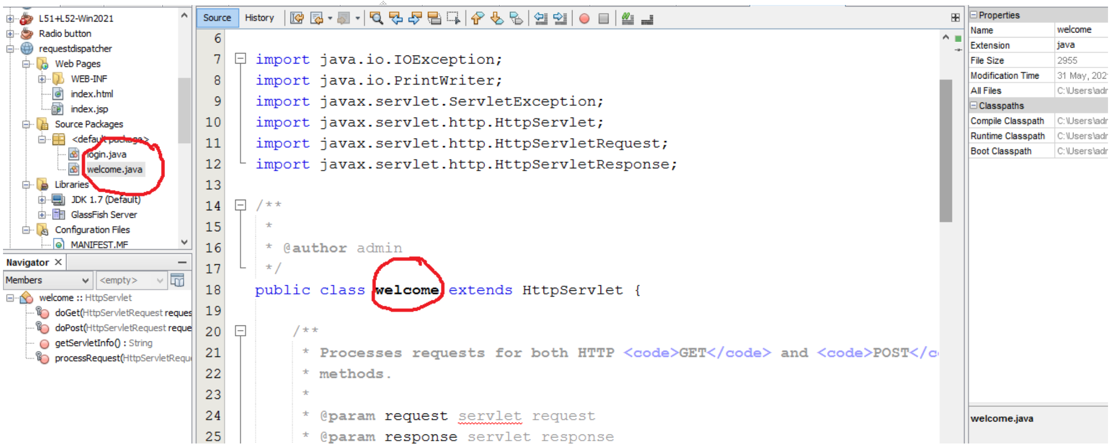
if(p.equals("aaa")) // this is condition checking line

{

```
        RequestDispatcher rd=request.getRequestDispatcher("welcome");

  // requestdispatcher is an interface which helps to dispatch the request to the another
page ( page may be servlet/jsp/html). in our case it is dispatched to a servelet called
welcome--which we are going to create

//**** important at this time if you get error on RequestDispatcher, right click on the
//screen and fix import, then the error disappears**********

// RequestDispatcher comes with two methods
//1.forward()  and 2. include()
// include() method includes the content of a resource in the response, the resource
//could be another Servlet, JSP or HTML file. While forward() method is used to
//forward the request to another resource.

        rd.forward(request, response);
    }
    else{
        out.print("Sorry UserName or Password Error!");
        RequestDispatcher rd=request.getRequestDispatcher("/index.html");
        rd.include(request, response);

        }
```

9. Now it is time to create the welcome servlet

similar to login.java create the welcome servlet

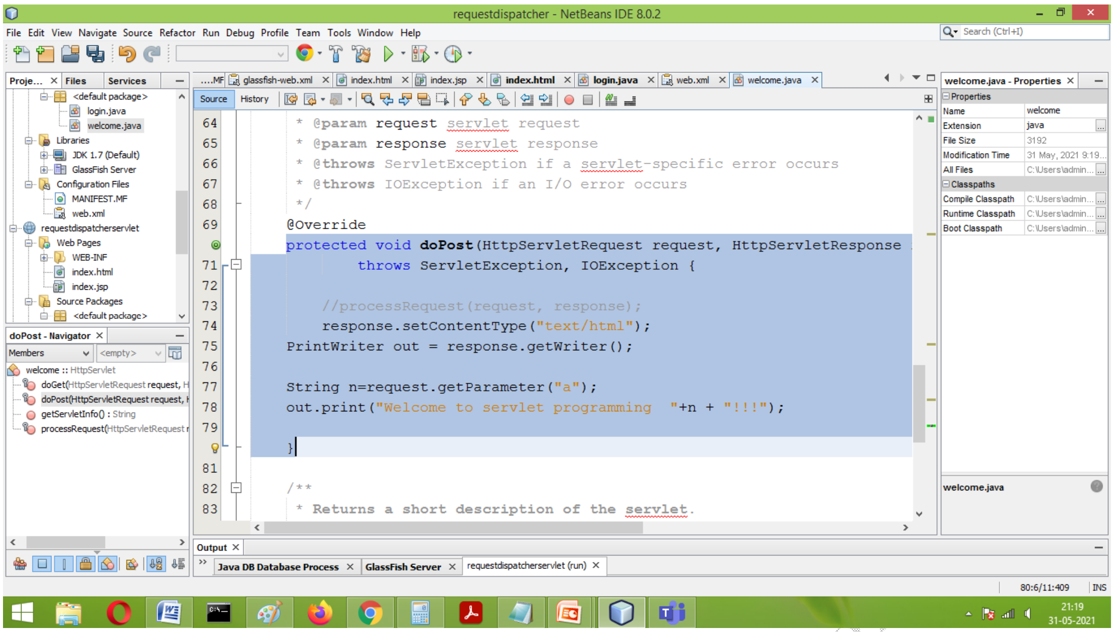


```

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;

/**
 *
 * @author admin
 */
public class welcome extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</c
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
```

Properties	
Name	welcome
Extension	java
File Size	2955
Modification Time	31 May, 202
All Files	C:\Users\adr
Classpaths	
Compile Classpath	C:\Users\adr
Runtime Classpath	C:\Users\adr
Boot Classpath	C:\Users\adr

welcome.java

request dispactcher will forward the request to welcome servlet, so have to write under doPost method of welcome.java file



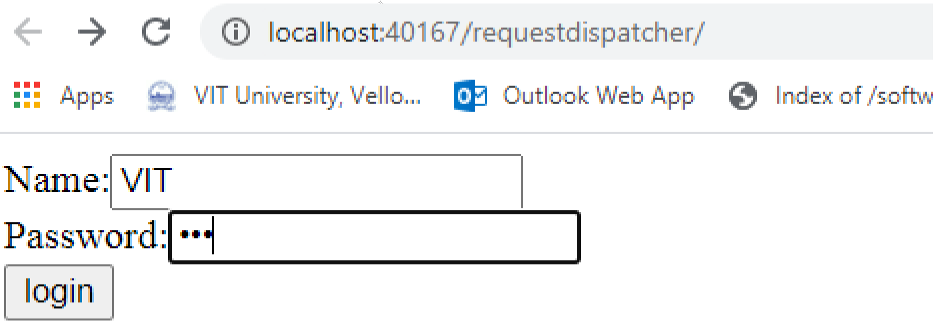
requestdispatcher - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
glassfish-web.xml
index.html
index.jsp
index.html
login.java
web.xml
welcome.java
Source
History
welcome.java - Properties
Output
Java DB Database Process
GlassFish Server
requestdispatcherservlet (run)

```
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
            throws ServletException, IOException {

        //processRequest(request, response);
        response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    String n=request.getParameter("a");
    out.print("Welcome to servlet programming  "+n + "!!!");

    }

    /**
     * Returns a short description of the servlet.
```

Write the following code in the doPost () under welcome.java

```
response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    String n=request.getParameter("a");
    out.print("Welcome to servlet programming  "+n + "!!!");
```

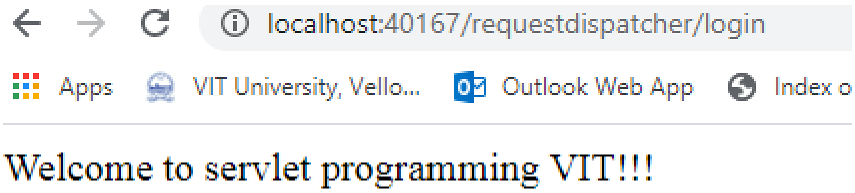
9. After saving the project, run file you can get the output as we have initial of this document



Name: VIT

Password: •••

login



Welcome to servlet programming VIT!!!

******** TRY THIS PROGRAM AND CHECK YOUR ANSWER**********