

SOEN 387

WEB-BASED ENTERPRISE

APPLICATIONS DESIGN

TUTORIAL – 2

Java Web & Servlets

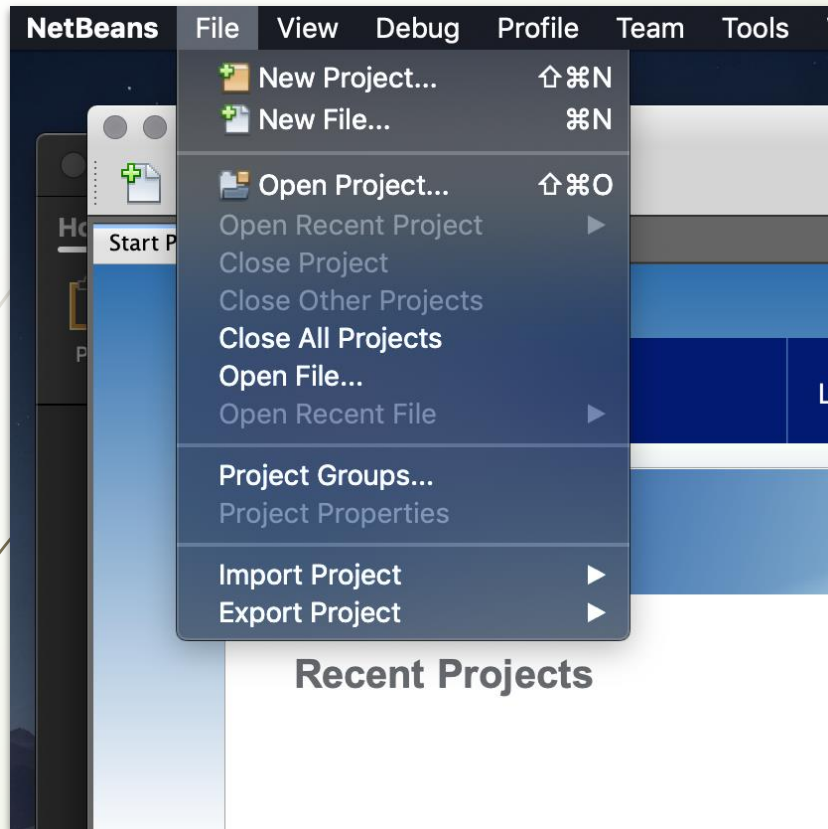
Agenda

- NetBeans IDE Prerequisites
- Setting up a Java project in NetBeans
- What is servlet
- Creating web project
- Creating servlet
- Servlet API
- Building and Running the Application
- Debugging a servlet file
- File Structure
- Exercise

NetBeans IDE Prerequisites

Software or Resource	Version Required
<u>NetBeans IDE</u>	version 7.2, 7.3, 7.4, or 8.0
<u>Java Development Kit (JDK)</u>	version 6, 7, or 8

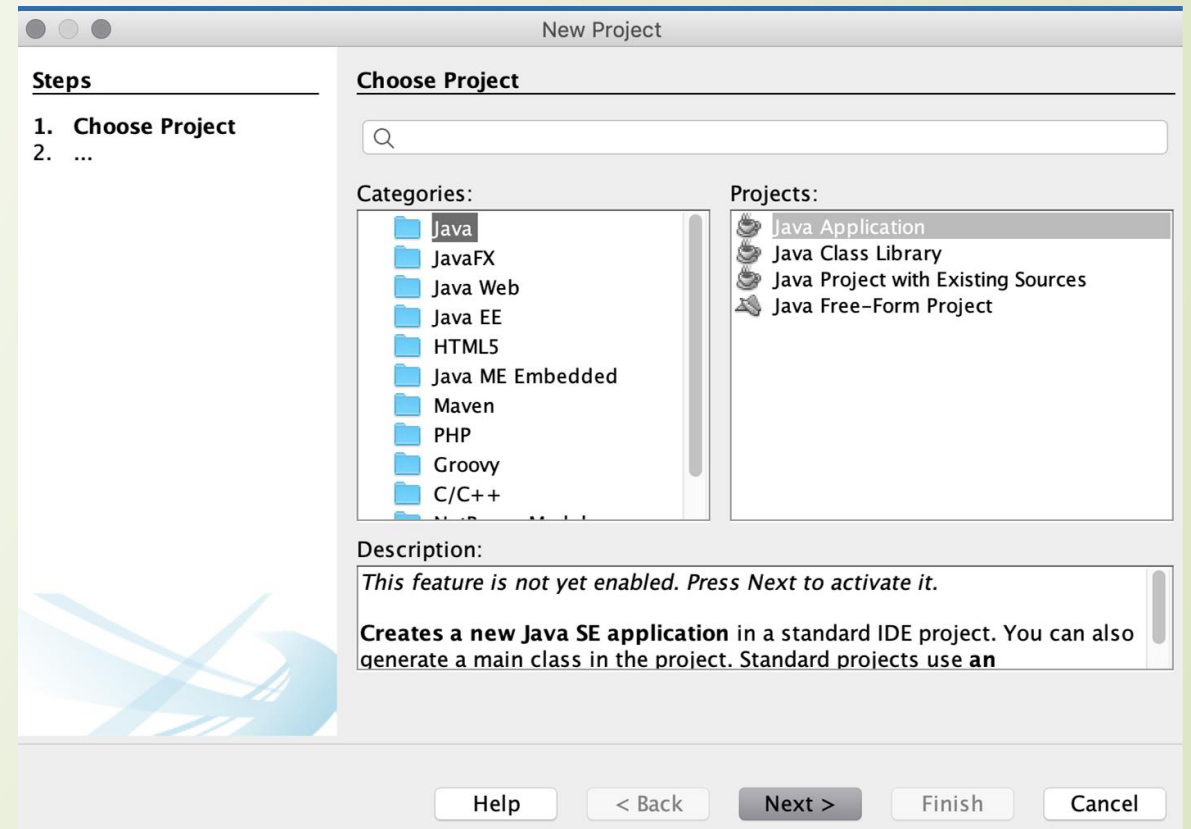
Setting up a Java project in NetBeans



4

In the IDE, choose File > New Project, as shown in the figure above.

SOEN 387 WEB-BASED ENTERPRISE APPLICATIONS DESIGN



In the New Project wizard, expand the Java category and select Java Application as shown in the figure below. Click Next.

Fall 2019

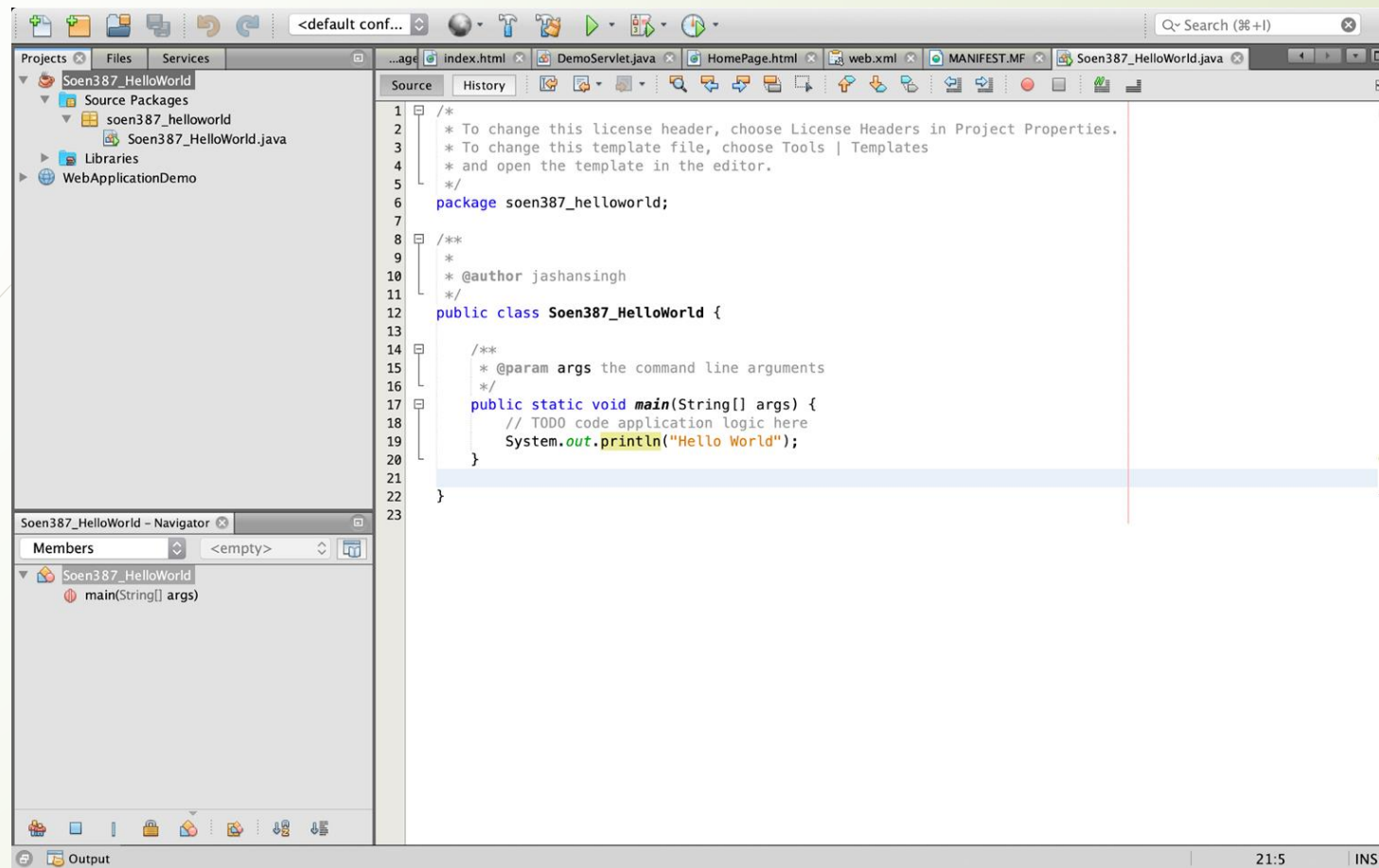
The screenshot shows the 'New Java Application' wizard in NetBeans, specifically the 'Name and Location' step. The 'Steps' pane on the left lists '1. Choose Project' and '2. Name and Location'. The main area contains the following fields and options:

- Project Name:** `Soen387_HelloWorld`
- Project Location:** `/Users/jashansingh/NetBeansProjects` with a **Browse...** button.
- Project Folder:** `gh/NetBeansProjects/Soen387_HelloWorld`
- ☐ **Use Dedicated Folder for Storing Libraries**
- Libraries Folder:** (empty) with a **Browse...** button.
- A note: "Different users and projects can share the same compilation libraries (see Help for details)."
- ☒ **Create Main Class** `soen387_helloworld.Soen387_HelloWorld`

At the bottom, there are buttons for **Help**, **< Back**, **Next >**, **Finish** (highlighted), and **Cancel**.

In the Name and Location page of the wizard, do the following:

- In the Project Name field, type `Soen387_HelloWorld`
- Leave the Use Dedicated Folder for Storing Libraries checkbox unselected.

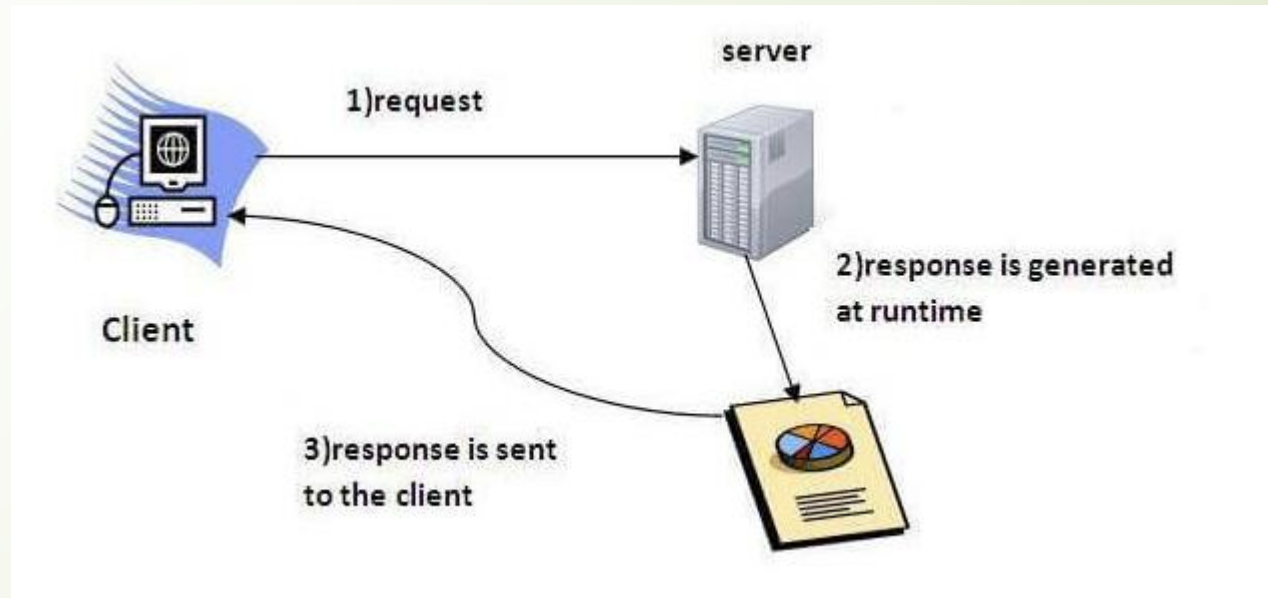


6

The project is created and opened in the IDE. You should see the following components:

- The Projects window, which contains a tree view of the components of the project, including source files, libraries that your code depends on..
- The Source Editor window with a file called Seon387_HelloWorld open.
- The Navigator window, which you can use to quickly navigate between elements within the selected class.
- To run the program:
- Choose Run > Run Project.

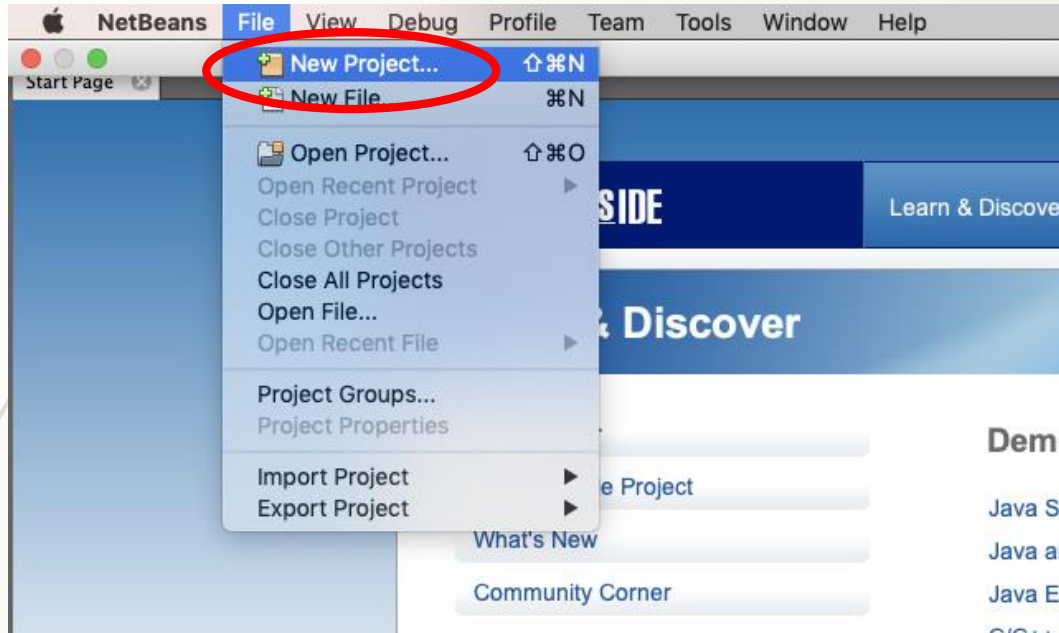
What is servlet



7

- Used to create a web application
- Resides at server side and generates a dynamic web page
- Client sends HTTP request
- Server receives request, servlets process it
- Results returned (HTML document, images, binary data)

Creating web project

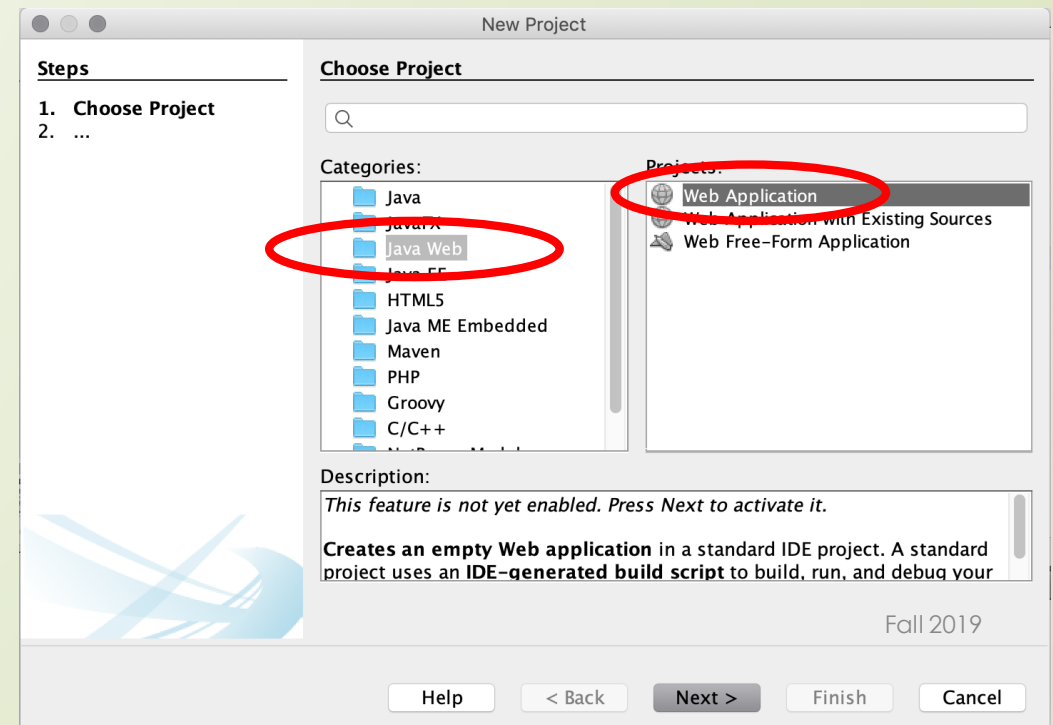


Open Netbeans IDE, Select File -> New Project

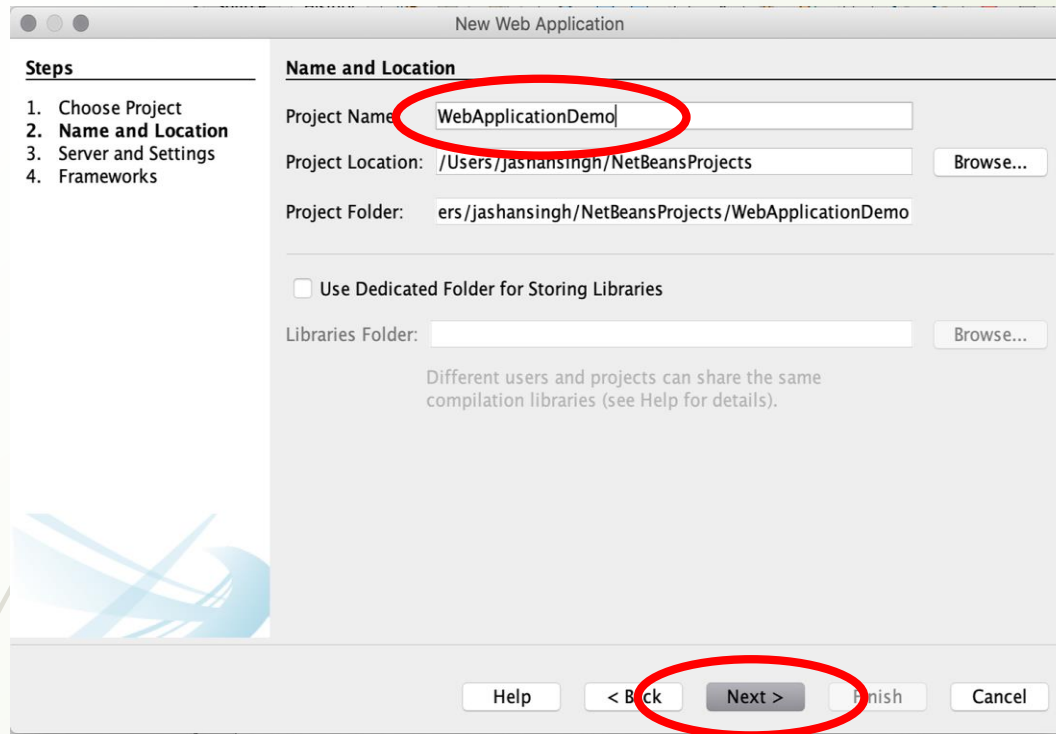
8

Select Java Web -> Web Application, then click on Next,

SOEN 387 WEB-BASED ENTERPRISE APPLICATIONS DESIGN



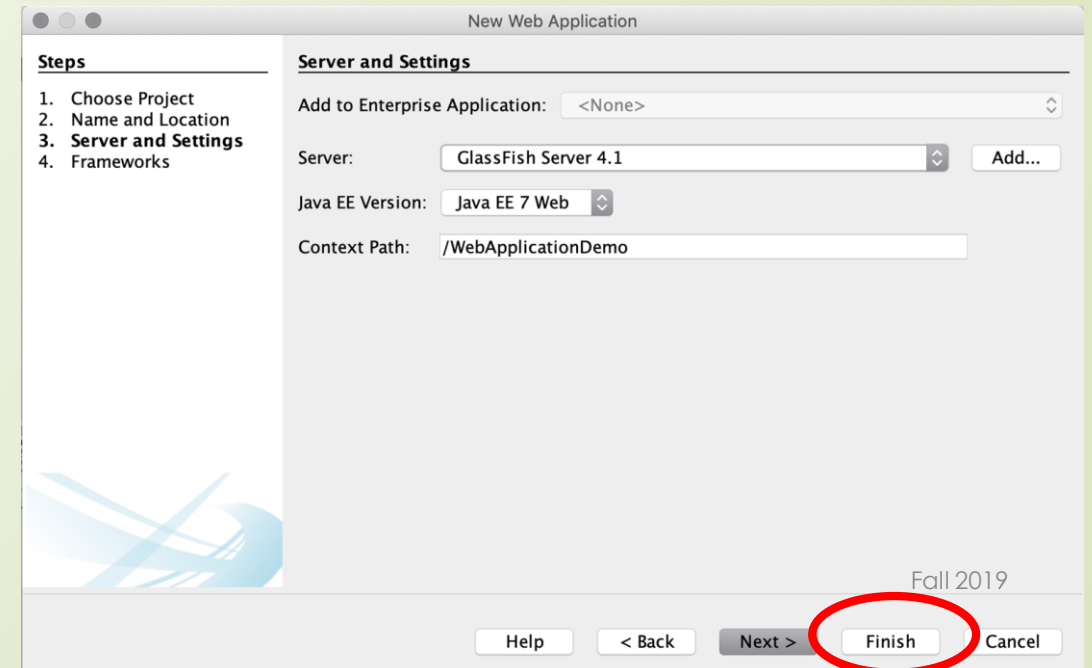
Fall 2019



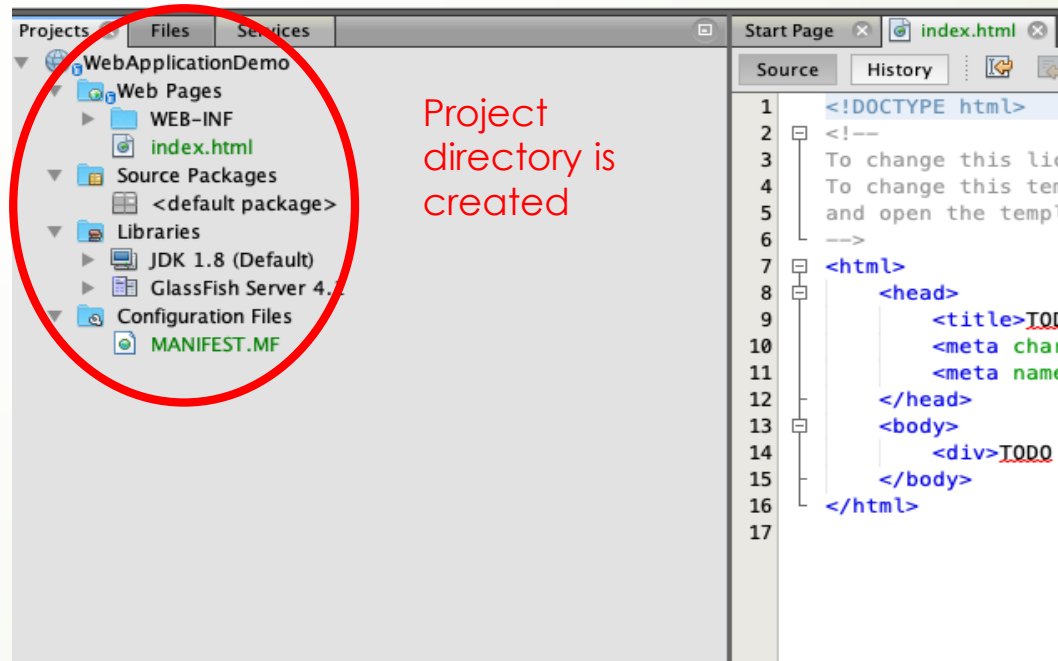
Give a name to your project and click on Next,

9

and then, Click Finish



Creating servlet

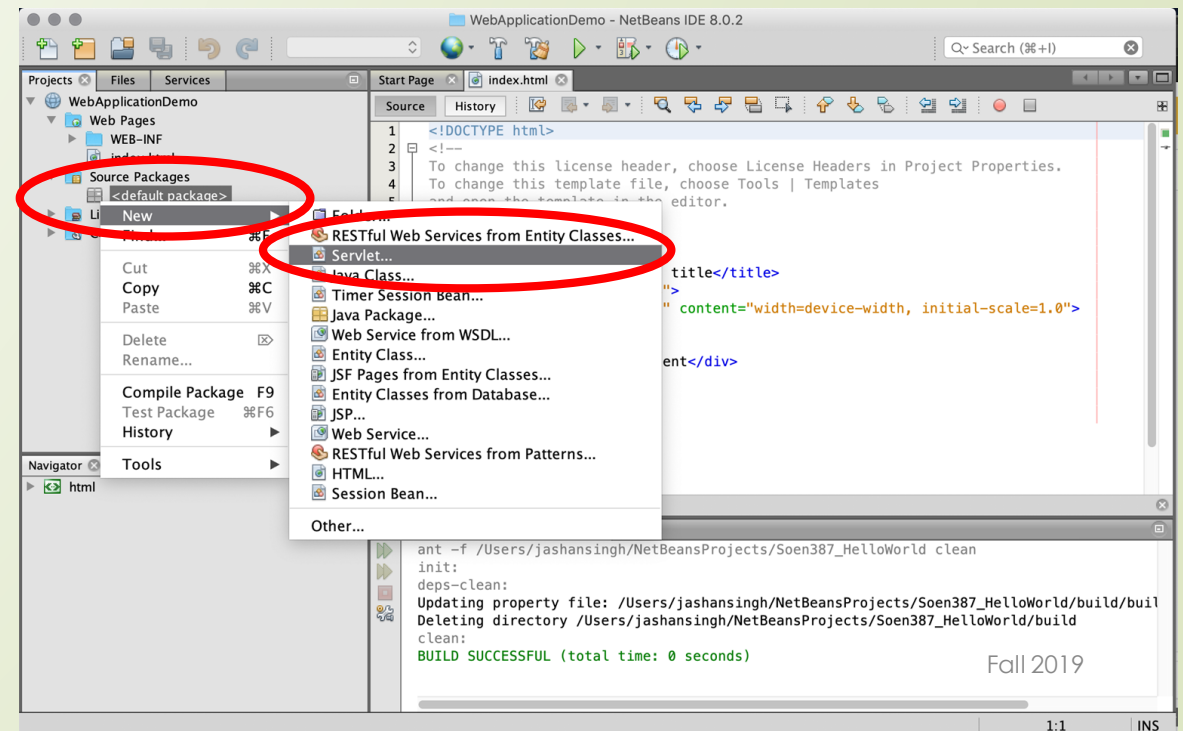


The complete directory structure required for the Servlet Application will be created automatically by the IDE.

10

To create a Servlet, open Source Package, right click on default packages -> New -> Servlet.

SOEN 387 WEB-BASED ENTERPRISE APPLICATIONS DESIGN



New Servlet

Steps

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

Name and Location

Class Name: DemoServlet

Project: WebApplicationDemo

Location: Source Packages

Package:

Created File: jashansingh/NetBeansProjects/WebApplicationDemo/src/java/DemoServlet.java

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

Help < Back **Next >** Finish Cancel

Give a Name to your Servlet class file

11

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: DemoServlet

Servlet Name: DemoServlet

URL Pattern(s): /DemoServlet

Initialization Parameters:

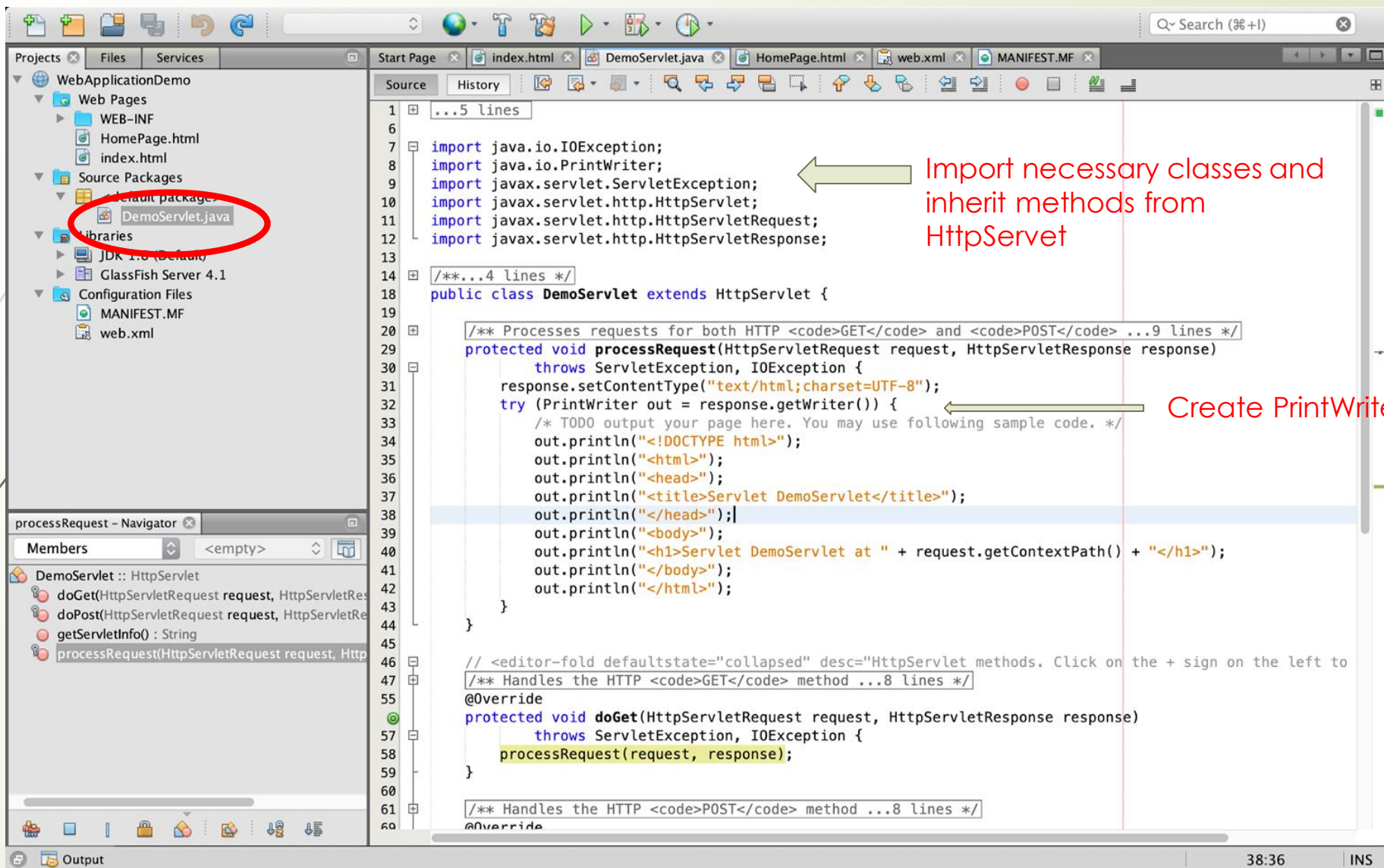
Name	Value
------	-------

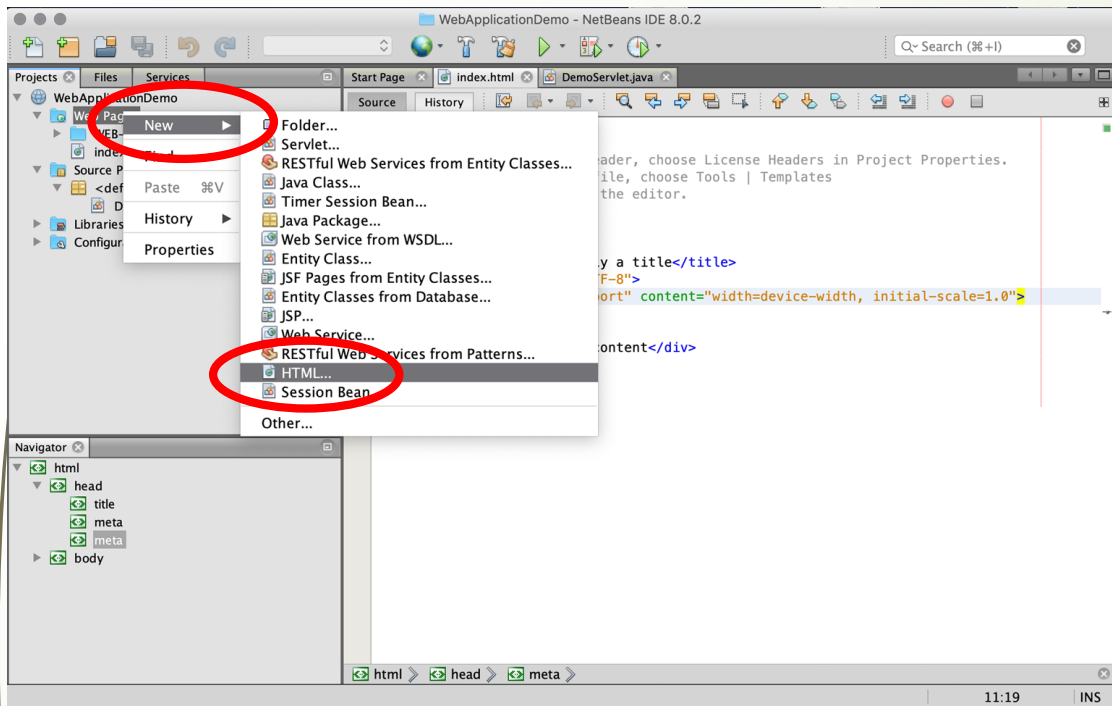
Edit... Delete

Help < Back Next **Finish** Cancel

This will add servlet information in web.xml

Change Servlet name and URL pattern from here

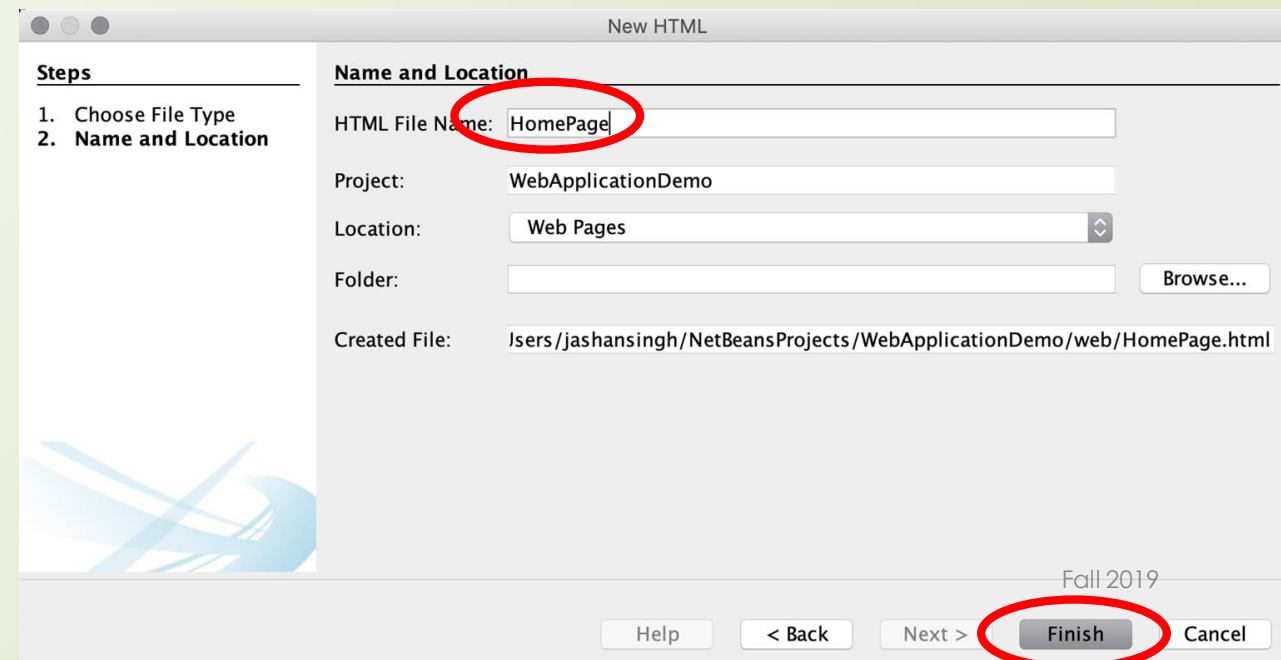




Create an HTML file, right click on Web Pages -> New -> HTML

13

Give it a name.



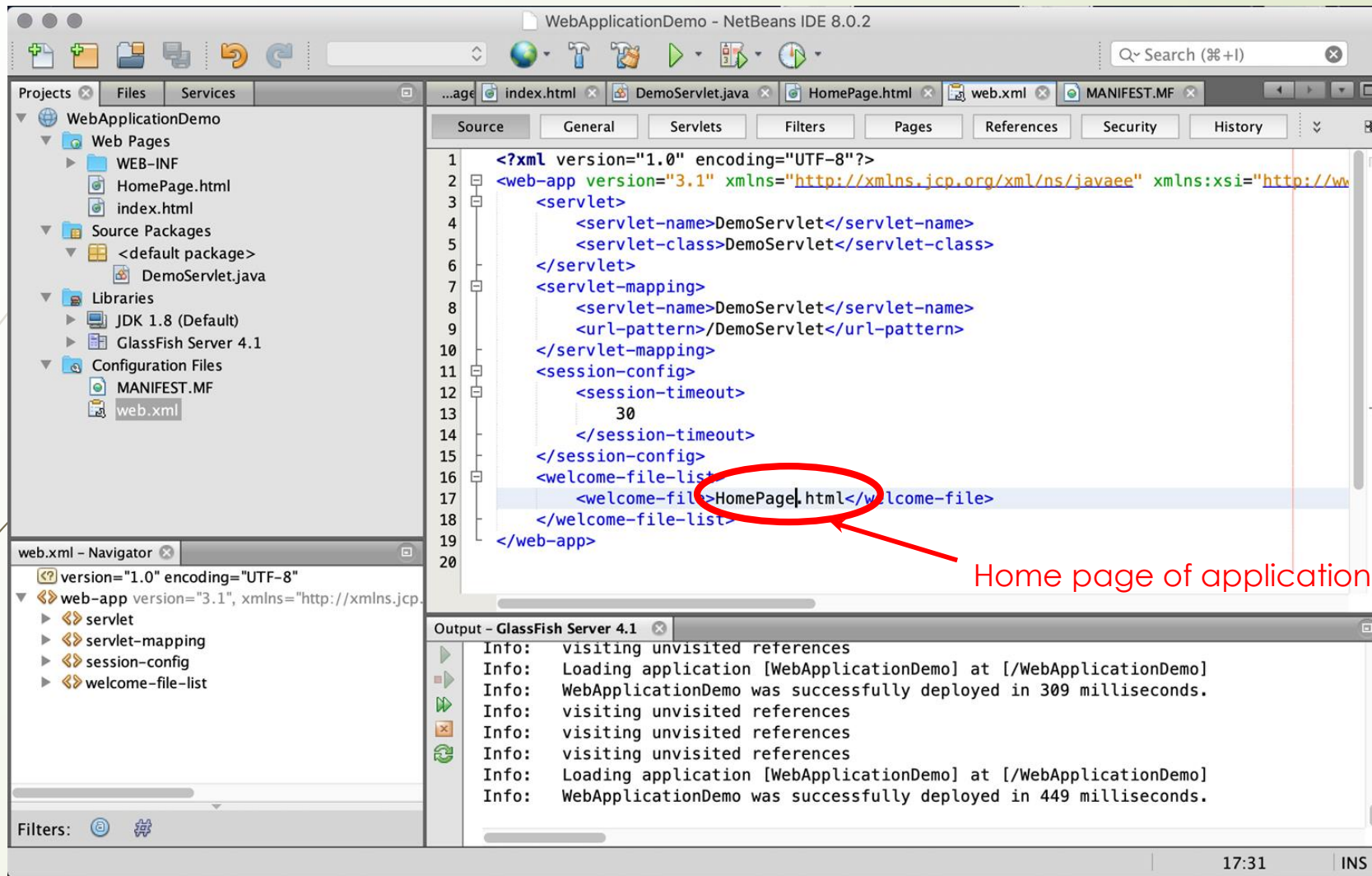
The screenshot shows an IDE with a project named 'WebApplicationDemo'. The 'Projects' pane on the left shows the project structure, including 'Web Pages', 'Source Packages', 'Libraries', and 'Configuration Files'. The 'index.html' file is selected in the 'Web Pages' folder. The 'Navigator' pane at the bottom left shows the DOM tree of the selected HTML file, with the 'a' tag highlighted. The main editor window shows the source code of 'index.html'. The code includes a DOCTYPE declaration, a license header, and an HTML document structure. A red arrow points to the 'href' attribute of the 'a' tag, which is labeled 'Servlet Name'.

```
1 <!DOCTYPE html>
2 <!--
3 To change this license header, choose License Headers in Project Properties.
4 To change this template file, choose Tools | Templates
5 and open the template in the editor.
6 -->
7 <html>
8 <head>
9 <title>Hello World</title>
10 <meta charset="UTF-8">
11 <meta name="viewport" content="width=device-width, initial-scale=1.0">
12 </head>
13 <body>
14 <div>Welcome to SOEN 387 Tutorial</div>
15 <h4>Click here to go to <a href = "DemoServlet"> DemoServlet Page </a></h4>
16 </body>
17 </html>
18
```

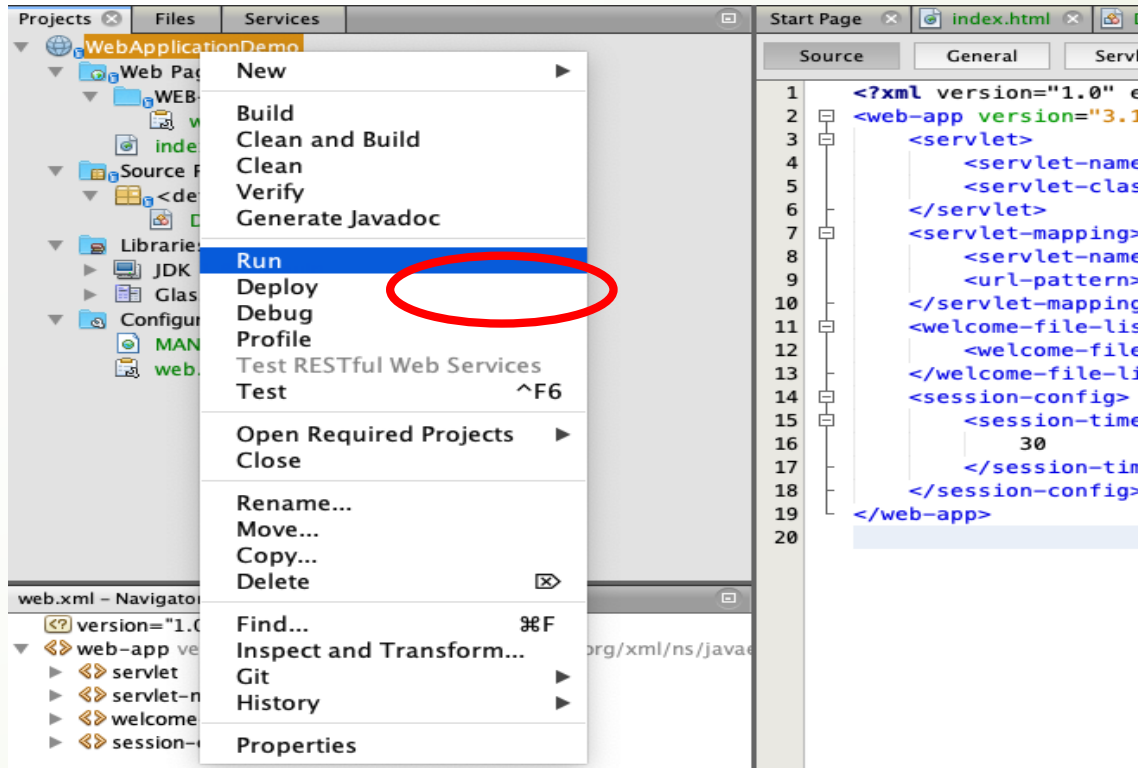
Servlet Name

14

Write some code inside your HTML file. We have created a hyperlink to our Servlet in our HTML file. Fall 2019
SOEN 387 WEB-BASED ENTERPRISE APPLICATIONS DESIGN



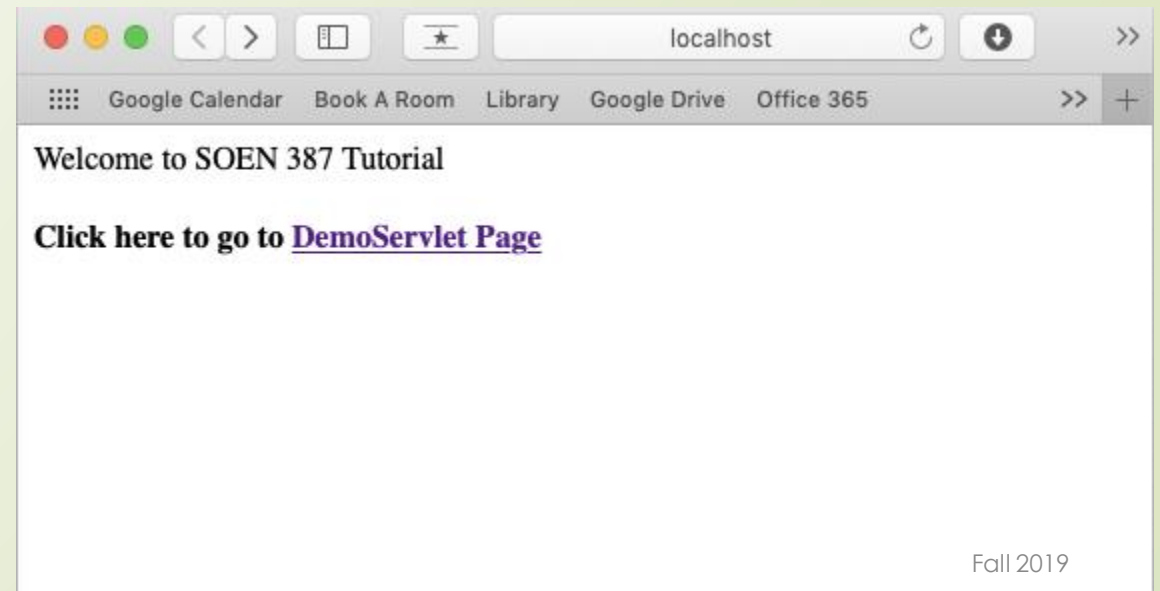
Edit web.xml file. In the web.xml file you can see, we have specified the url-pattern and the servlet-name, this means when DemoServlet url is accessed our Servlet file will be executed.

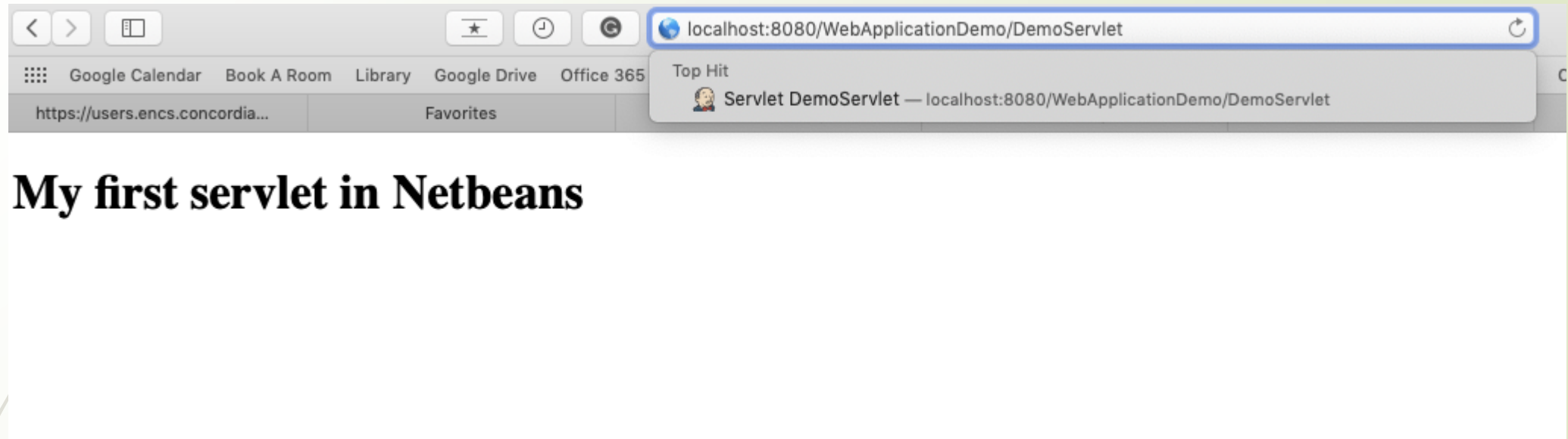


Run your application, right click on your Project and select Run

16

open localhost:
<http://localhost:8080/WebApplicationDemo/>





Hurray! Our First Servlet class is running.

Servlet API

- All servlets must import the following packages:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
```

- All servlets must extend the HttpServlet or Servlet class

```
public class DemoServlet extends HttpServlet
```

- Servlets must provide an implementation for doGet, doPost or both

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

- HTML can be written as text to the response
- Obtain the PrintWriter object from the response and write text to it

```
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 DemoServlet</title>");  
    out.println("</head>");  
    out.println("<body>");  
    out.println("<h1>Servlet DemoServlet at " +  
request.getContextPath() + "</h1>");  
    out.println("</body>");  
    out.println("</html>");  
}
```

Building and Running the Application

Once you have written and test run your application, you can use the Clean and Build command to build your application for deployment. When you use the Clean and Build command, the IDE runs a build script that performs the following tasks:

- Deletes any previously compiled files and other build outputs.
- Recompiles the application and builds a JAR file containing the compiled files.

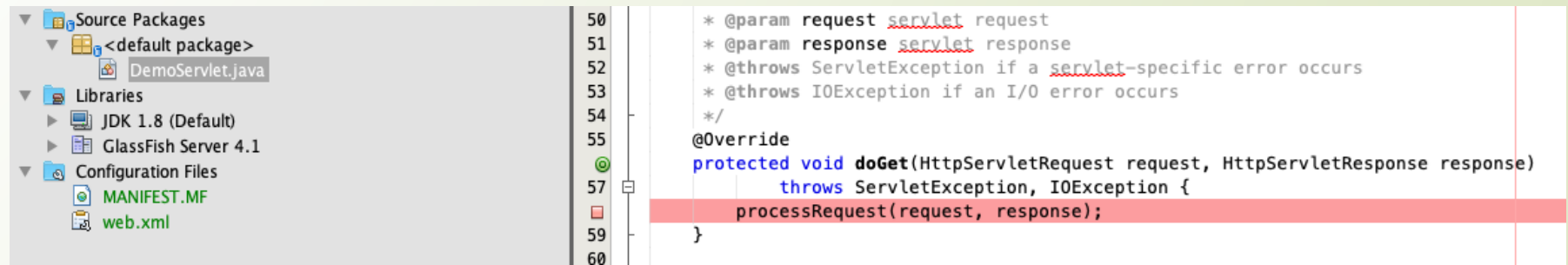
To build your application:

- Choose Run > Clean and Build Project. You can view the build outputs by opening the Files window and expanding the SOEN387_HelloWorld node.
- The compiled bytecode file SOEN387_HelloWorld.class is within the build/classes/SOEN387_HelloWorld subnode.

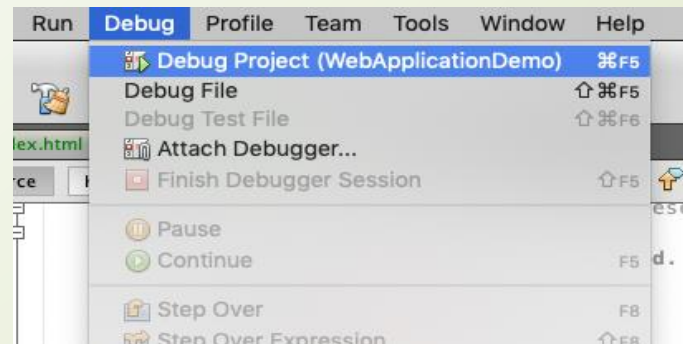
Debugging a servlet file

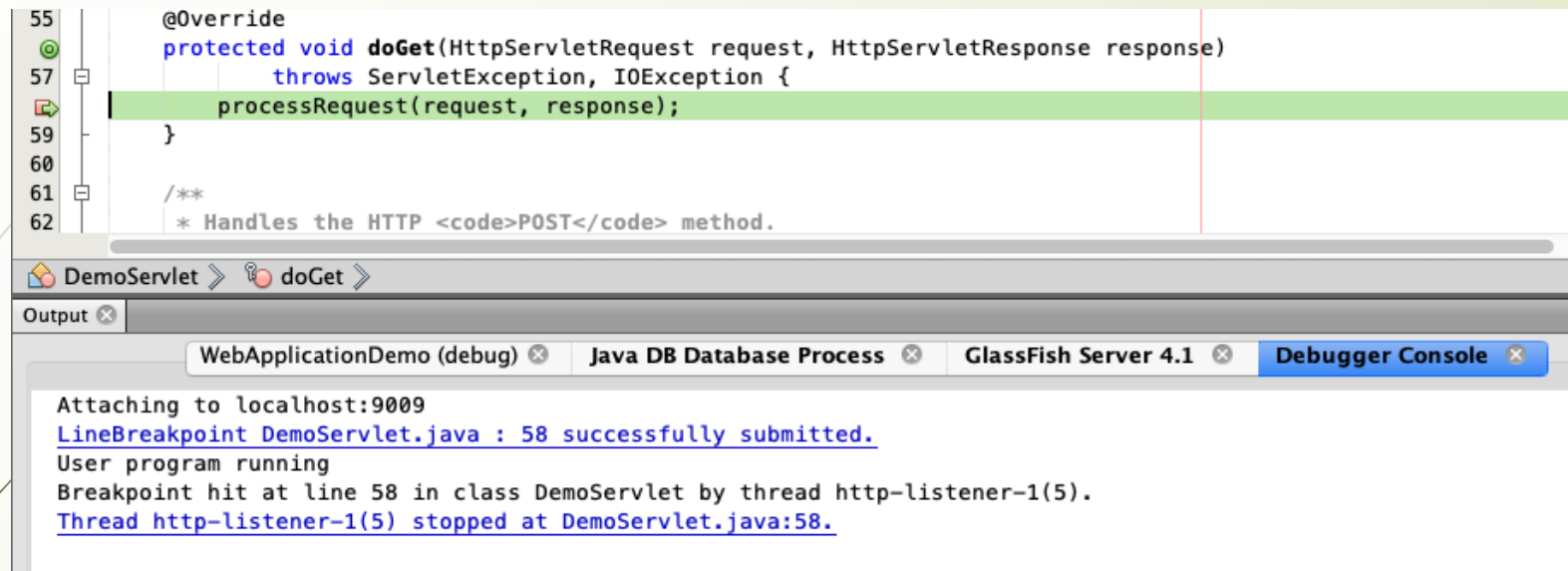
When you execute a servlet in NetBeans, it will try to start the internal Tomcat. Since Tomcat is running inside of NetBeans, you will be able to set breakpoints in the code and stop execution of the servlet.

- Place a breakpoint on a line inside of doGet



- From the Debug menu select Start Session and Run In Debugger

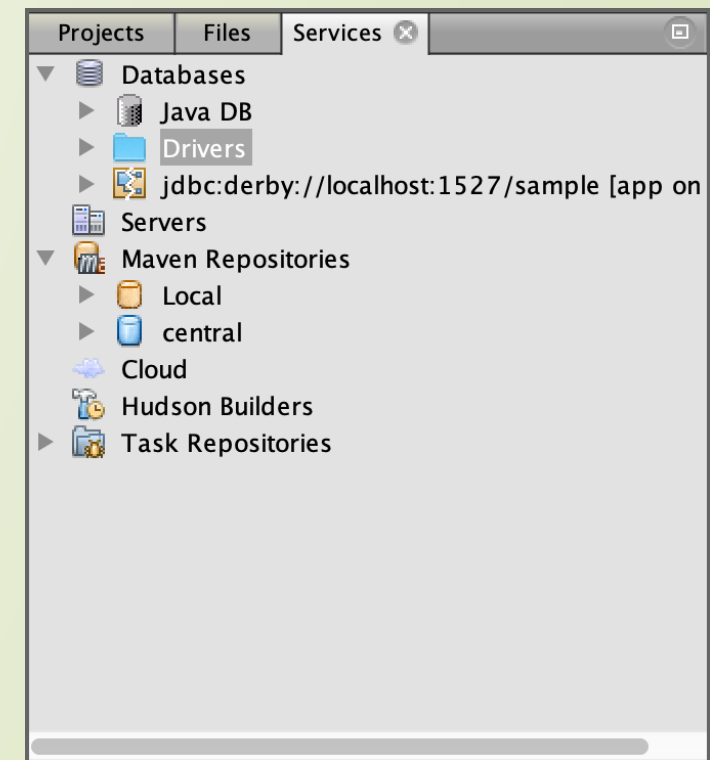
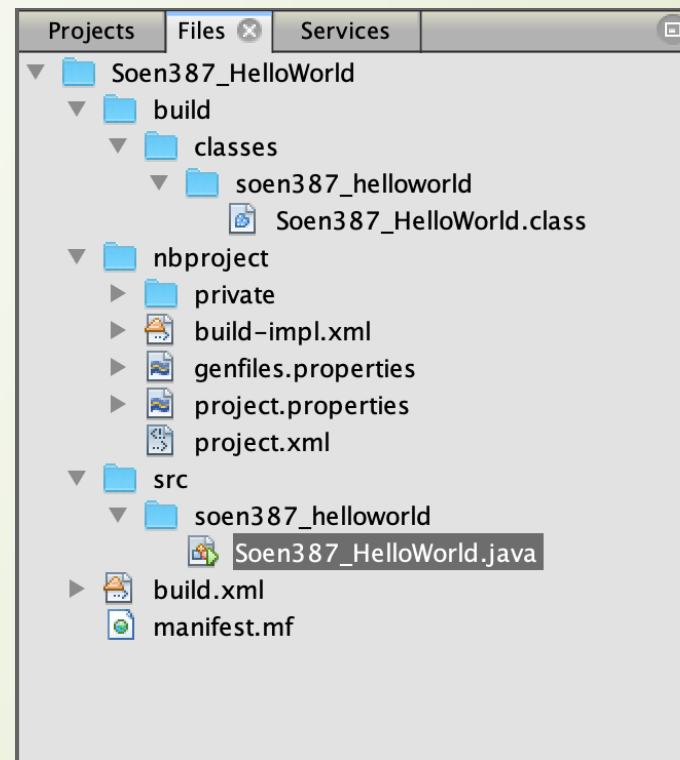
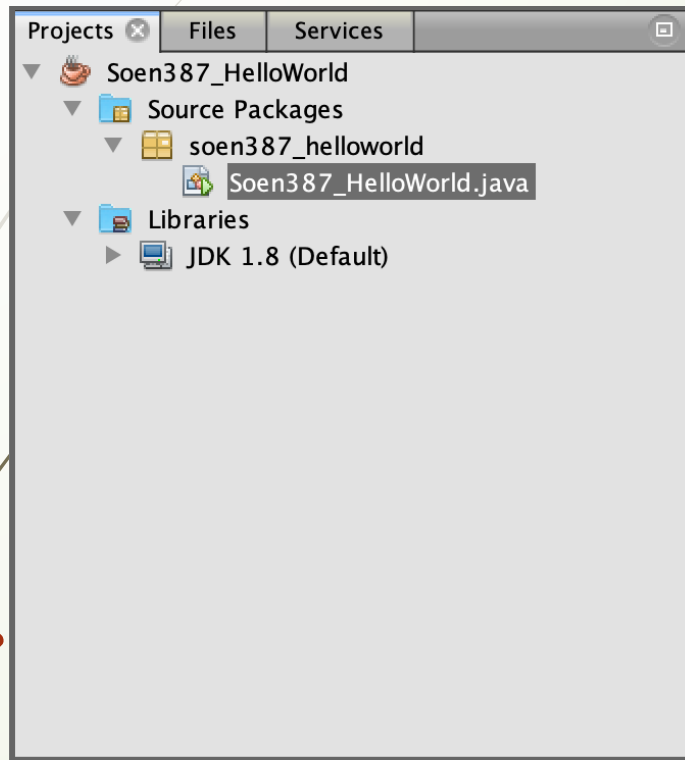




NetBeans will start the internal Tomcat and will display the servlet in the browser.

Control should pass to the breakpoint in NetBeans.

Project File Structure



Exercise

Setting the HTTP status code

It is sometimes necessary to set a status code. An HTTP status code— sometimes called a response code— effectively tells the client what the output represents. Or in other words, it indicates the status of the request: successful, page not found, access denied etc.

Write a servlet to send the appropriate HTTP status codes based on request parameter. For instance, you may look for a request parameter called 'document' and generate the following response codes:

1. status code 200 SC_OK, *when document equals 'document1'*
2. status code 301 SC_MOVED_PERMANENTLY , *when document equals 'document2'*
3. Status code 404 SC_NOT_FOUND , *when document equals 'document3'*
4. Status code 400 SC_BAD_REQUEST, for any other values