CPRG352 - Web Application Programming
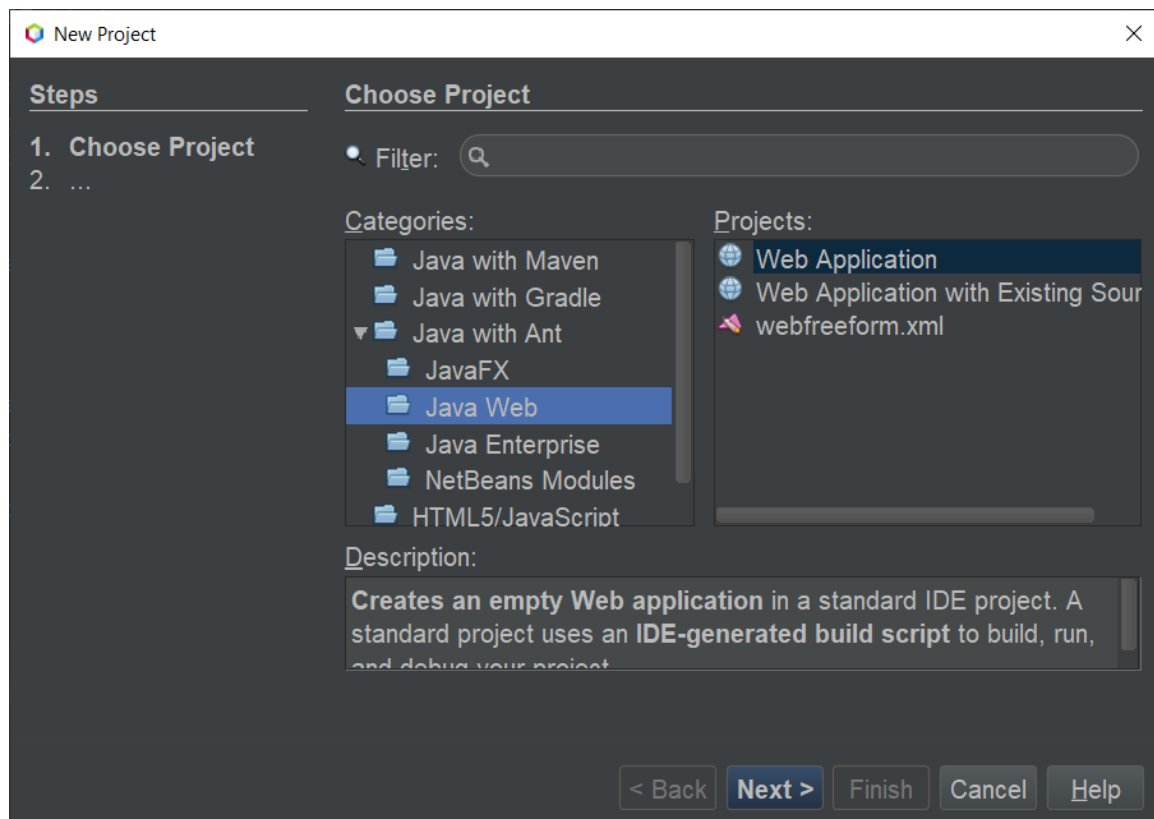
Intro Lab #2

Problem: Develop and deploy a simple Java web application

In this section of the lab you will use NetBeans to create a new project containing a Java web application. This application will, when run, simply display a message to the user in a web page.

- Go to NetBeans
- Select the *Projects* tab at the top-left of the NetBeans window
- Initially the list below the tab title will be empty because you have not created any projects yet. As you create new projects this list will fill in
- Select the *File>>New Project...* menu option

A *New Project* wizard will appear that will ask you questions about options to set for the new project.

- Expand the *Java with Ant* category and then select *Java Web* for the subcategory. Select *Web Application* under *Projects*:
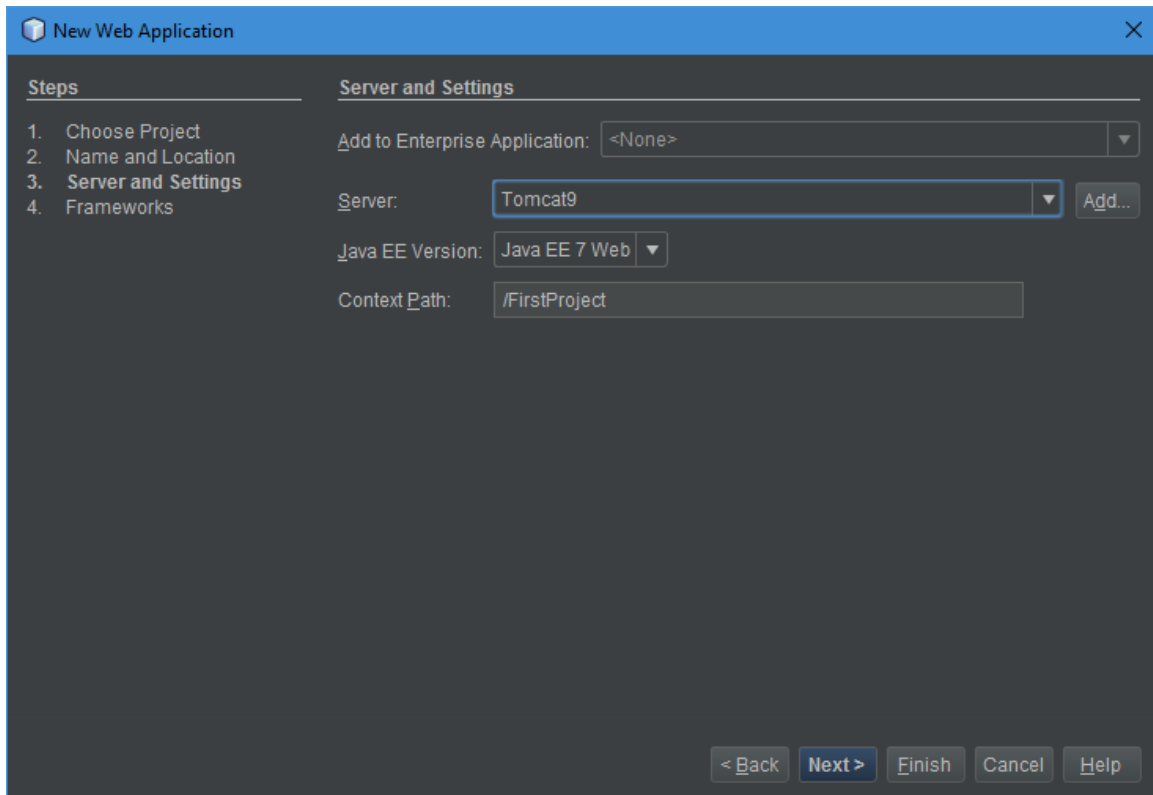
- Click on *Next*
- In the next page, change the Project Name to **FirstProject** (no spaces)

New Web Application ✕

| Steps | Name and Location |

Steps

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

Name and Location

Project Name: FirstProject

Project Location: C:\Users\joloughl\Documents\NetBeansProjects   Browse...

Project Folder: C:\Users\joloughl\Documents\NetBeansProjects\FirstProject

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: _____   Browse...

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

< Back   Next >   Finish   Cancel   Help
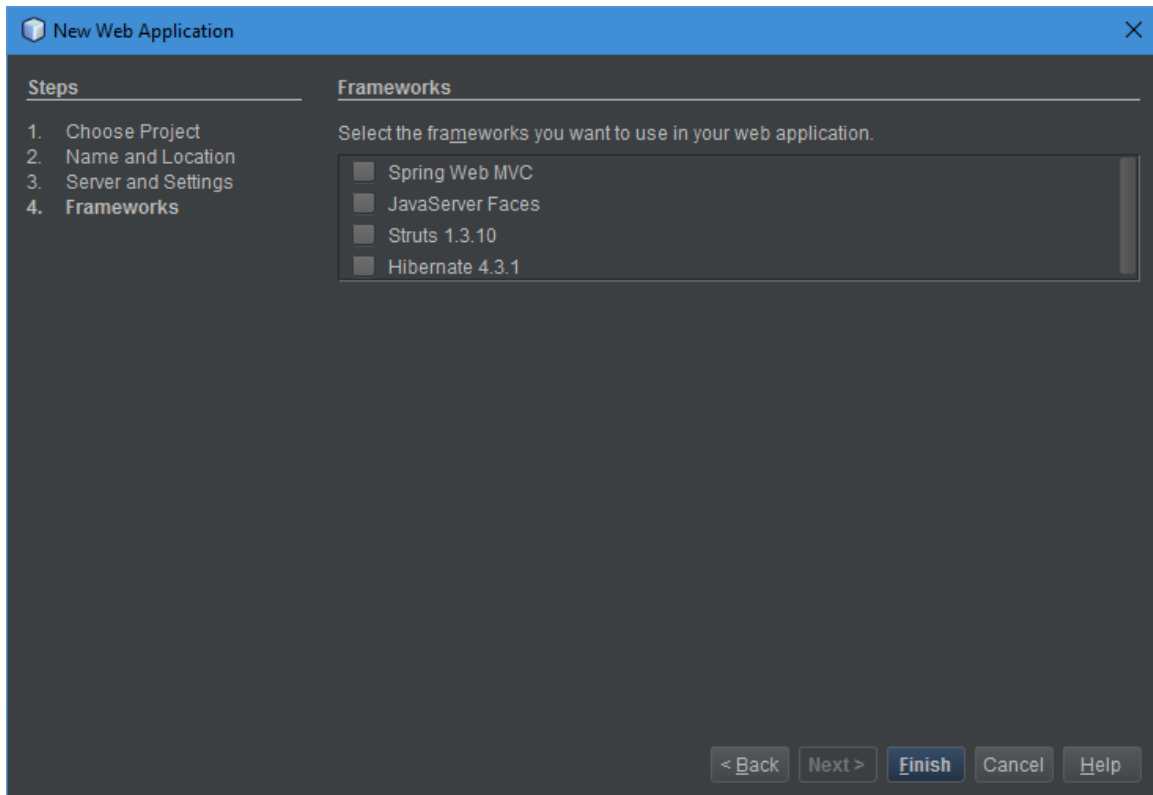
- Click on *Next*

The next page shows a number of options (see below), including: what application server to we want this web application to deploy and execute on; the version of the JEE framework to use; and the *context path* for the application (the extra part of the URL after the machine name and port you specify in a browser to tell the application server which application that you want to use. In this case we are saying that to run this application we will provide the URL: *http://localhost:8080/FirstProject*

A single application server could host many web applications at the same time, and the application server knows which particular application a user wants to use by looking at the context path s/he gives in the URL.

- We do not want to change any of these settings, so click on *Next*

The final page of the wizard asks you if you want to use a framework to help with the operation of the site, e.g. to construct fancy user interfaces in HTML, etc. In this case we will not need any of these.
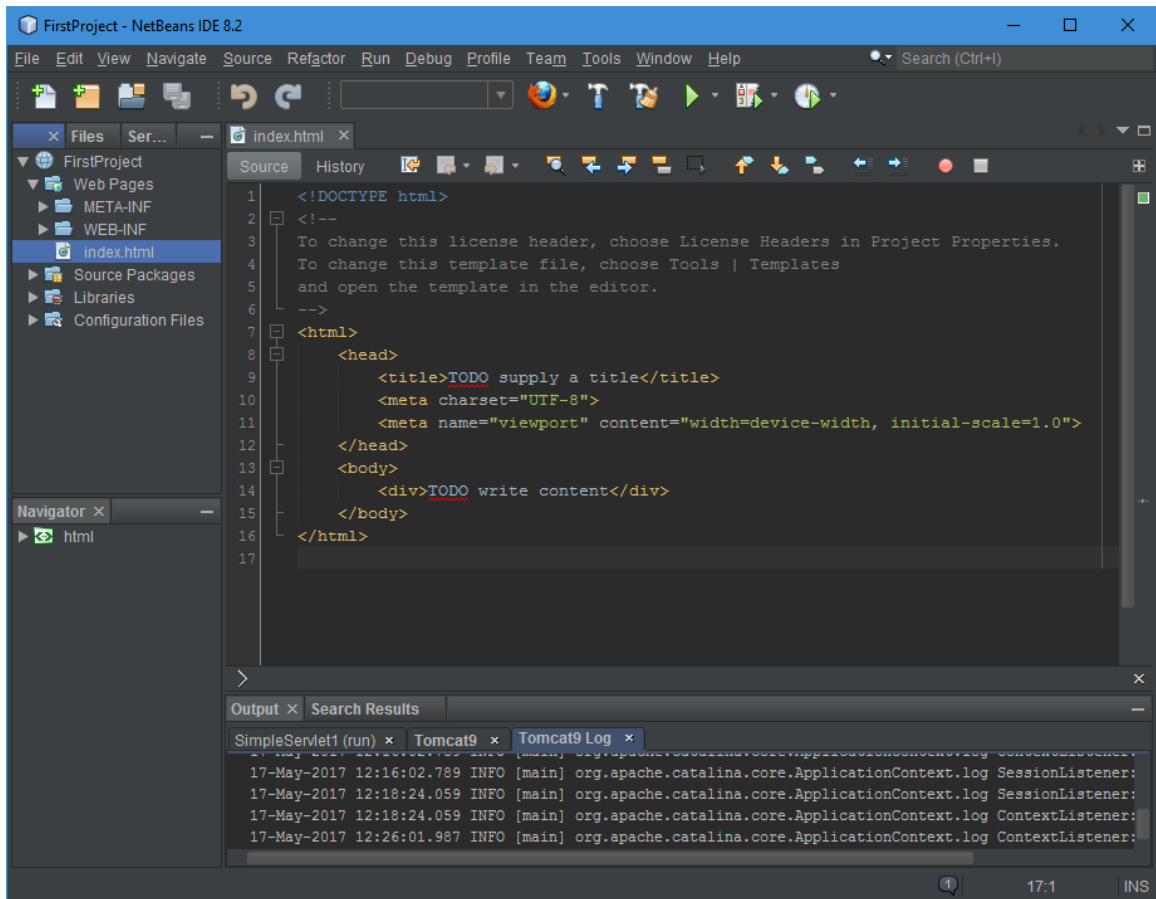
- Click on *Finish*

NetBeans will now use the settings specified in the wizard to create a new project. In our VM you can find the directory for the project at *C:\Users*<username>*\Documents\NetBeansProjects\FirstProject*.

NetBeans should now appear as shown below. The ***Project*** panel shows the project structure and contents. The panel to the right shows the code for the currently-selected source code file (***index.html*** in this case), and the ***Output*** panel at the bottom shows any output messages you might be interested in, e.g. errors, warnings, etc. Note that the Output panel can itself contain multiple panes, each showing messages for different parts of the overall system.

To use this new web application you need to:

1. Build it (compile all the files and package up the results).
2. Deploy the application to the server. NetBeans does this for us, which simplifies development and debugging
3. Run the application in a browser. Again, NetBeans will automatically load the application in a browser window (running the browser program if necessary)

To build, deploy and execute the new web application:

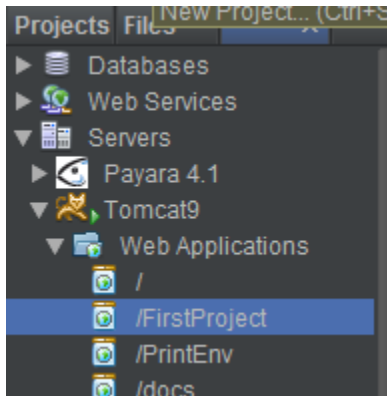- Click on the green "go" button in the NetBeans toolbar:

You should see the following in your browser:

TODO write content

This is the output of the new web application. *index.html* is the default page shown by the application if it is accessed without a specific page or resource being referenced in the URL, e.g. *localhost:8080/FirstProject/* versus *localhost:8080/FirstProject/mypage.html*

You can confirm that the web application is running from Tomcat by going to the **Services** tan in NetBeans and expanding the "**Web Applications**" option under the Tomcat9 server, e.g.:
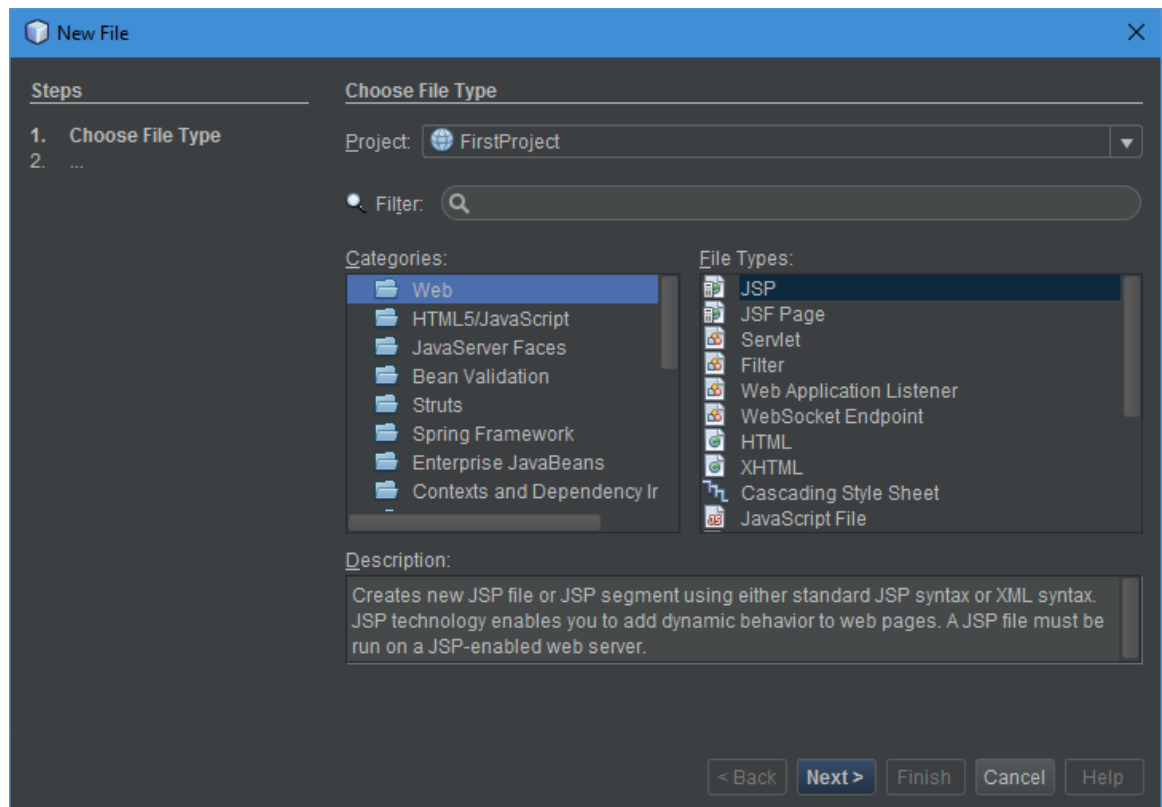

Once the application has been deployed to Tomcat it can be run without having to use Netbeans (although in our VM closing Netbeans will automatically stop Tomcat also, so do not do that yet).

To make the development process easier for the developer Netbeans will automatically redeploy an application from Netbeans to Tomcat if the developer makes a change to the program code and just saves the change.
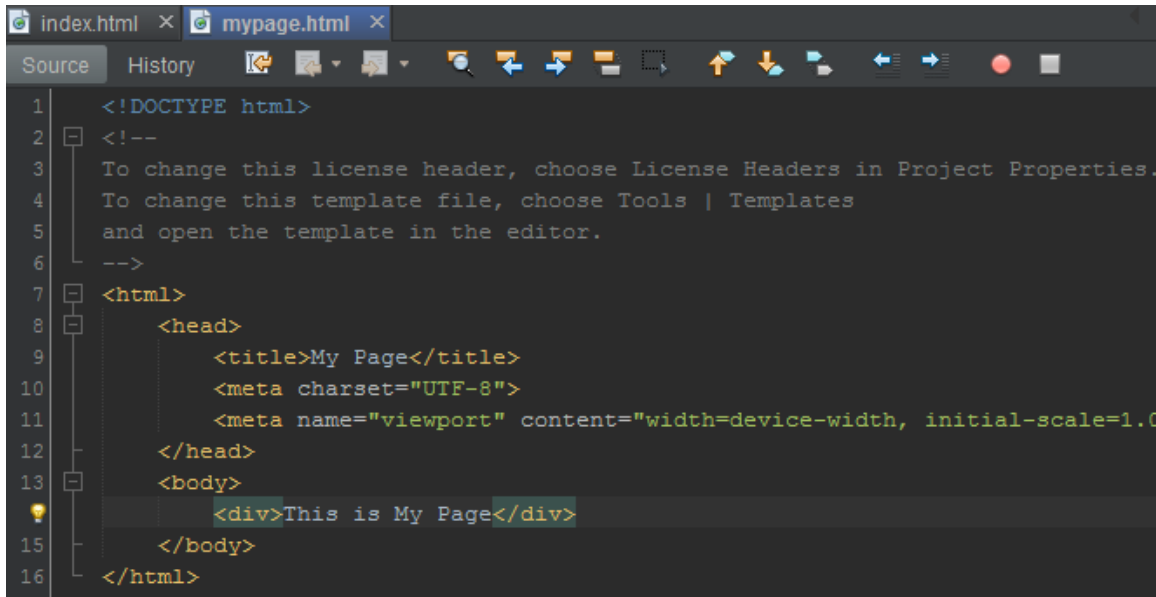
Adding a second page to the application

To add another web page to the application:

- Select the **File/New File…** menu option. You will see a dialog box showing all of the different kinds of files that can be added to the project:

- Under the *Categories* list select *Web*
- Under the "*File Type*s" list select *HTML*
- Click on *Next*
- Give a file name of "*mypage*". Do **not** add a *.html* extension on to the file name! Netbeans will do this for you automatically. If you enter a file name of "*mypage.html*" you will actually get a file called "*mypage.html.html*", which will be confusing (and incorrect)
- Click on *Finish*

Change the <title> text of the page to "*My Page*", and the text in the <div> in the body of the page to "*This is My Page*", e.g.

```
index.html  ×    mypage.html  ×
Source    History

  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>My Page</title>
 10              <meta charset="UTF-8">
 11              <meta name="viewport" content="width=device-width, initial-scale=1.0
 12          </head>
 13          <body>
                <div>This is My Page</div>
 15          </body>
 16      </html>
```
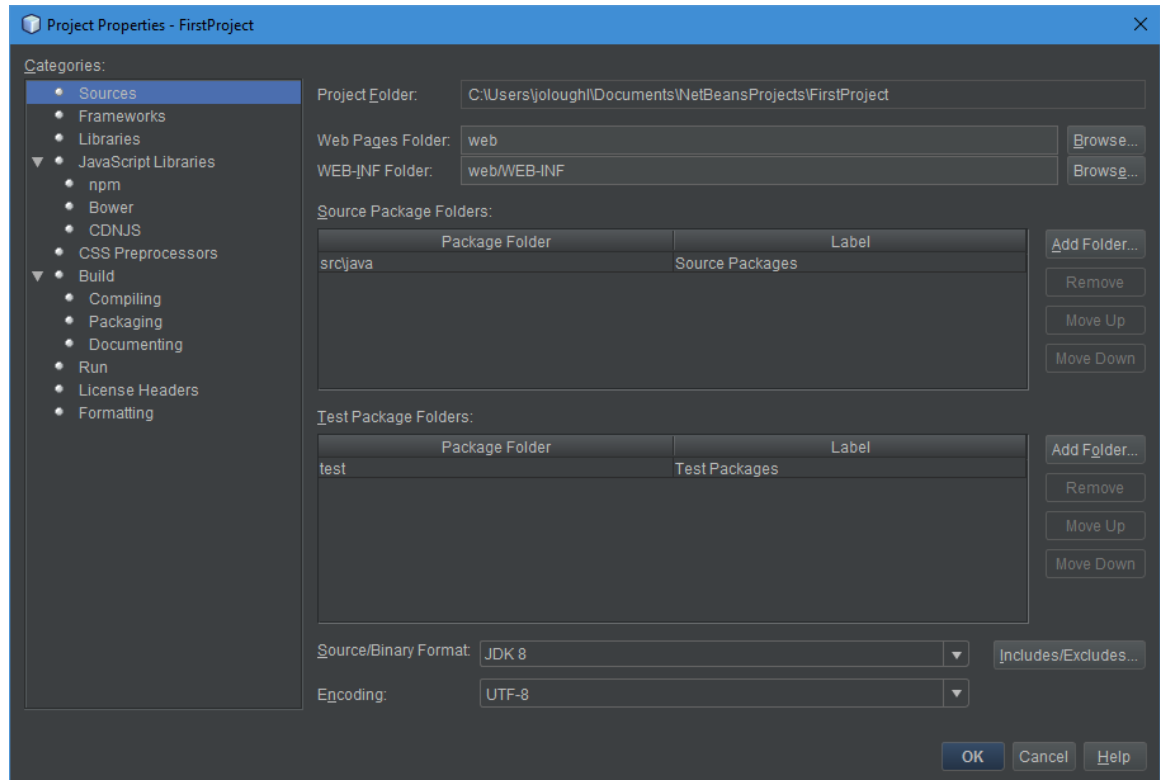
- Save the changes. Netbeans will automatically rebuild and redeploy the project for you. You don't have to click on the green **Run** button in the toolbar to do this

- In the browser's URL bar enter *localhost:8080/FirstProject/mypage.html* and press the return key. You will see the contents of **mypage.html** displayed in the browser. If you re-run the application you will instead see the contents of index.html (the default page shown for the application)

Changing the default page for the application

If you would prefer to see **mypage.html** as the default page of the application, make the following change to the project:

- Right-click on the "*FirstProject*" entry in the Project list in Netbeans and select "*Properties*" from the bottom of the popup menu. You will see the Project Properties dialog box:

- Select "**_Run_**" in the Categories list
- In the "**_Relative URL_**" textbox on the right enter "**_/mypage.html_**" and click on OK
- Re-run the application. This time you will see the contents of mypage.html displayed as the default content for the application

Lab Task

Create a new Java web application called **MyFirstApp**. This application will have three web pages contained within it:

- **index.html**: initial page seen by users of the application. Shows a standard header containing an icon and a title. The page contents is a welcome message and links to the other two pages
- **photos.html**: As well as the standard header this page shows a 3x3 table where each cell in the table shows a photograph. Each photo must be the same size on-screen as the others, even if the original images were different sizes. The page must also contain a link back to the **_index.html_** page
- **quotes.html**: a page with the standard header and content consisting of inspiring quotes that you like (about anything at all). This page also needs a link back to the **_index.html_** page

All the pages must be similarly styled via an external CSS style sheet stored in a file called **_myFirstAppStyle.css_**.