3. Secure Shell (SSH): http://en.wikipedia.org/wiki/Secure_Shell
4. How to spell "unix": http://www.greens.org/about/unix.html
5. Unix Introduction:
   http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html
6. Unix commands: http://www.cs.bu.edu/teaching/unix/reference/
7. Bash Shell: http://www.gnu.org/software/bash/manual/bashref.html
8. Pico, nano http://en.wikipedia.org/wiki/Pico_(text_editor)
9. Bash shell: http://www.hypexr.org/bash_tutorial.php

Lab 2: IDEs and Editors

Introduction

In this lab you will be using text editors and IDEs, discovering their strengths and weaknesses for a given application, and hopefully finding some that fit how you work. You also begin to explore how you can use an IDE to manage documents on remote servers, or edit documents on remote and synchronize them with your individual computer.

Text Editors vs. Word Processors

A text editor is not the same thing as a word processor. In general, a word processor is focused more on the final printed *appearance* of the information it contains, while a text editor is designed to allow the user to understand the exact character-level contents of the file they are editing.

Word processors allow extensive formatting of fonts, color, layouts, inline photos, etc.; word processors generally do not save into text-only files, but may insert special characters into the file that are not seen while editing the document.

Text editors focus on showing the user exactly what characters are in the file. Some text editors may color the text in the display for *syntax highlighting* source code while editing, but that color does not appear in the text file that was being edited.

GUI Text Editors

On a local machine, many people prefer to use a text editor with a graphical user interface (GUI). Examples of this type of editor include notepad++ (Windows), TextWrangler/BBEdit (Mac), and Sublime (Linux). These types of editors may allow the use of the mouse in addition to the keyboard, and may employ multiple windows.

These editors allow the editing of text files on the local computer, and may allow editing of files on a remote computer through the use of ftp or sftp protocols.

For larger files, or files that require a lot of editing, it is probably best to load the file to your local computer, edit them, then upload them to the remote server.

The exploration of this type of editor is left to you.

Terminal Text Editors

Terminal text editors are text editors that run in a text-based terminal window, and are executed on the remote system, the system to which the terminal is connected. In the previous lab, you used the pico terminal text editor (see Lab 1 Classroom Work). On some systems, pico may not be available and nano, the GNU clone of pico, might be available. Using an editor in a text-based terminal means that the keyboard is used as the exclusive

input device. An editor like this generally does not allow a mouse or other pointers to be used for input. Output from a text editor must be completely text-based, with no pixel-level graphics included in the output file.

Terminal text editors can help you with many types of server oriented tasks. When you need to make small changes to a file on a remote system, look in logs, change configuration file, it is most likely going to be easier and faster to use a terminal text editor.

Two famous terminal text editors, Emacs and Vi (or Vim), are usually installed as standard editors on most unix systems. As a computer science major it is strongly recommended that you learn one or the other of these editors, which can be a daunting task. The rivalry between Emacs and Vi is legendary, dubbed the "Editor Wars." After you do the lab, it is worth going to http://en.wikipedia.org/wiki/Editor_war and learning a little history.

Integrated Development Environments (IDEs)

IDEs attempt to provide software developers a single software application that provides all tools necessary to develop software. This sort of arrangement can significantly enhance developer efficiency and code reliability.

At a minimum, for a language such as C++, modern IDEs include an integrated (text) editor, debugger, and compiler. Some provide interpreters that provide real-time feedback during coding, highlighting potential errors before the code is compiled. Other useful IDE tools can include object browsers, interfaces to databases, and collaboration and debugging tools.

In this lab you will be exploring a popular and easy to use IDE, NetBeans.

Lab 2 Preparatory Work

Install NetBeans

1. Go to https://netbeans.org/downloads/ and download the full version of NetBeans. You may also need to install the most current version of the Java Development Kit (JDK) first.

   Check with your instructor, who may have downloaded these files to a USB drive for quicker access.

2. Install NetBeans.

Lab 2 Classroom Work

Edit site.css using Vi

Before starting, a few words about Vi. Vi has two primary "modes" of operation that are accessed using keystrokes, the command mode and the text insert mode. When Vi starts, you are in command mode. In command mode, you can do all the things that you would expect to be able to do when you're editing a document, such as deleting, cutting and pasting, search and replace, etc.

Insert mode is the mode that allows you to enter text into a document. To enter insert mode from command mode, you type `i` (you can think that `i` is the command to go into insert mode). When you are in insert mode, what you type on the keyboard is inserted into your document. This is also the mode that you would use to change text in your file. To exit insert mode and return to command mode, press the `esc` key.

1. Log into your account on thecity (if you aren't already).

2. Go to your public_html directory.

3. Copy the file `~csc412/412/site_proto.css` to your `public_html` directory and rename it `site.css`.

4. Before opening the contents of site.css in vi, examine the contents of the file by printing it to the system console. Type `cat site.css↵`. You can see there are about seven lines. Once cat has output the file to the console, you should again see the unix command prompt.

5. Now, open site.css in vi by typing `vi site.css↵`

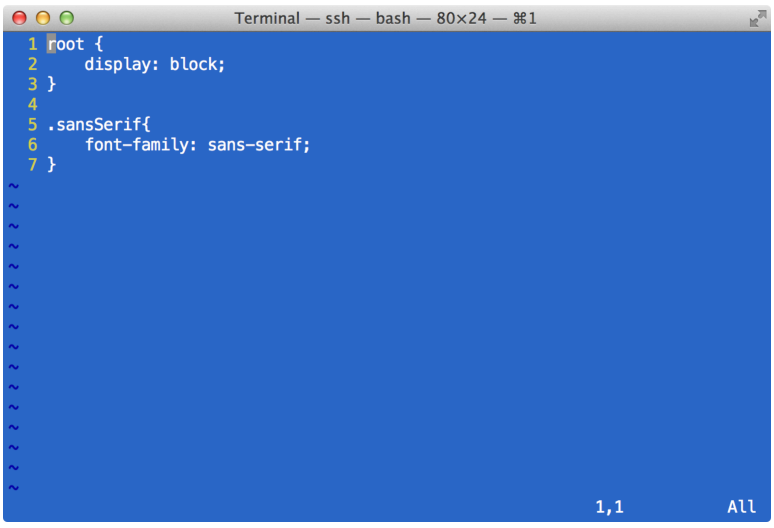

Figure 14. Editing site.css in Vi.

6. Vi opens and displays the file. There are fewer lines of text in the file than there is space on the screen. Vi denotes empty lines at the end of a file as tildes (Figure 14).

7. To show line numbers, type the command `:set number↵`. Your terminal screen should now appear as Figure 14.

8. Move the cursor down to line 4 by using the arrow keys on your keyboard. If the arrow keys don't work, in command mode you can use h key to move left, j to move up, k to move down, and l to move right.

9. When the cursor is on line 4, go into insert mode by pressing the `i` key (Note: To enter insert mode, do not press ↵ after pressing `i`). When vi enters insert mode, you should see an `-- INSERT --` message appear at the bottom left of the window.

10. Type the following text:

```
↵
.centered {↵
    text-align: center; ↵
}↵
```

11. To exit insert mode, press the `esc` key. The `-- INSERT --` message will disappear.

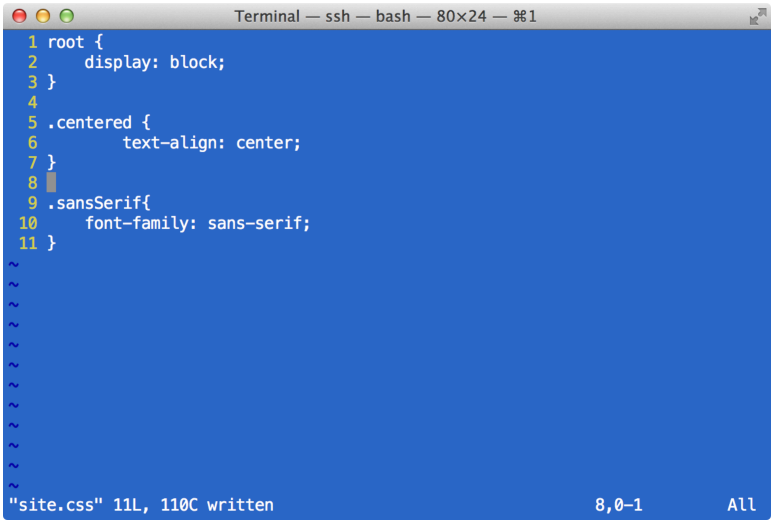

Figure 15. Vi displaying information about the successfully written file.

12. Now that you're back in command mode, type `:w↵` to write the modified file to the disk (Figure 15). Vi will tell you that it wrote the file, and how large it was.

13. Type `:q↵` and, provided you haven't modified the file since the previous step, vi will quit and return you to the unix command prompt.

Edit an html file using Emacs

The Emacs editor has a long history in the unix world. Though Emacs is a daunting task to learn, the initial investment in time pays off. Emacs remains one of the most powerful text editors, and it can be used for much, much more than just text editing.

Emacs uses programming-like commands that are "bound" to keystrokes, sometimes very complex combinations of `ctrl`, `esc`, and `meta` keys. The "meta" key corresponds to the `alt` key in putty on Windows, but the ⌘ key must be configured to be the meta key on Macs.

Windows users may skip ahead to the "Edit the about.html file using Emacs" section.

Mac Users: Configuring Mac Terminal to work with Emacs

In order to use Emacs, special character sequences involving the `meta` key need to be sent. The Terminal app needs special configuration to use the ⌘ key as the `meta` key.

1. If it isn't already open, open the Terminal app.

2. Open Terminal preferences, select Terminal > Preferences (or type ⌘,), and select the keyboard tab.

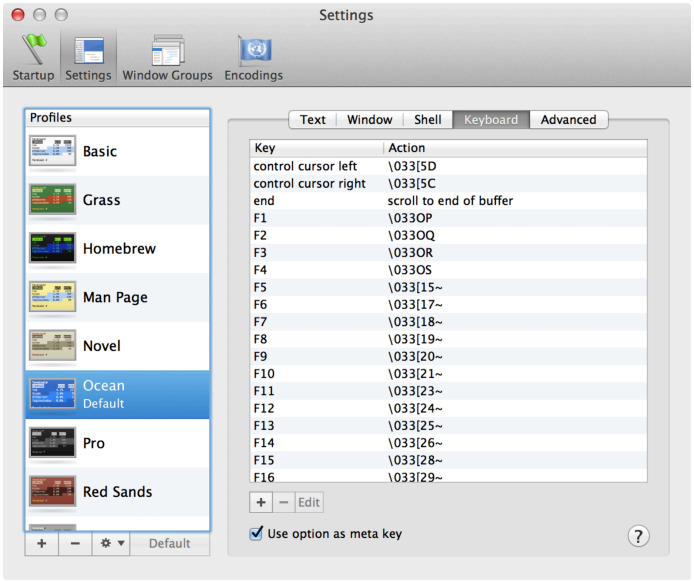

Figure 16. Configuring Mac Terminal for use with Emacs.

3. Make sure the "use option as a meta key" box is checked (Figure 16).

4. Quit Terminal, select Terminal > Quit (or type ⌘Q).

5. Restart Terminal.

Edit the about.html file using Emacs

1. Make sure you are logged into thecity and in your public_html directory.

2. Copy the `about_proto.html` file from `~csc412/412` to your current directory, renaming it `about.html`. Do you know how to do this using only one `cp` command?

3. Using a browser, look at how your `about.html` page currently appears. You'll need to specify the file that you want to see, so the url should be `http://thecity.sfsu.edu/~`*`username`*`/about.html`, where *username* is your username.

4. Open `about.html` in Emacs by typing `emacs about.html`↵.

5. Press ↵when you see the first screen, which will take you to your document.

6. Using the arrow keys, move to the line containing `<h1>About</h1>`.

7. Edit the line so it appears as:

```
<h1 class="centered">About</h1>
```

8. Save the file by typing `ctrl-x ctrl-c`, and answering yes to saving the file.

9. When Emacs exits and you return to the command line, see if Emacs has created a backup file in your `public_html` directory (how can you see if a file exists?). This file will have the name `about.html~`. Remove the file by (carefully) typing `rm about.html~`↵

10. Return to your browser and see how your `about.html` now appears.

Edit your site using NetBeans

Even though it might not seem like much, you already have an existing website on thecity. In this step you will download the files from the remote server onto your local computer. With an IDE such as NetBeans, you download the remote files onto your computer, edit them locally, then upload any changed files. One of the greatest pitfalls of this method, especially when you are working with other people on the same website, is issues of file synchronization; more on this later.

Create a NetBeans project for your site

1. Log into your account on thecity, and change to the public_html directory. List the files in this directory to remind yourself what is currently there.

2. If it is not already started, start NetBeans.

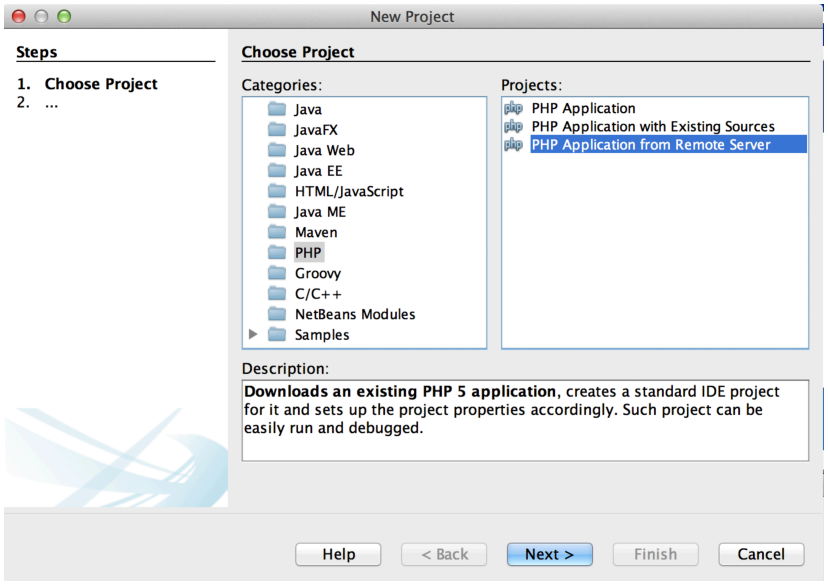

Figure 17. Create a new PHP project in NetBeans.

3. Create a new NetBeans project, by selecting File > New Project (`ctrl`-⇧-N in windows, ⇧⌘N on mac), then selecting PHP > PHP Application from Remote Server. Click on the next button to continue (Figure 17).

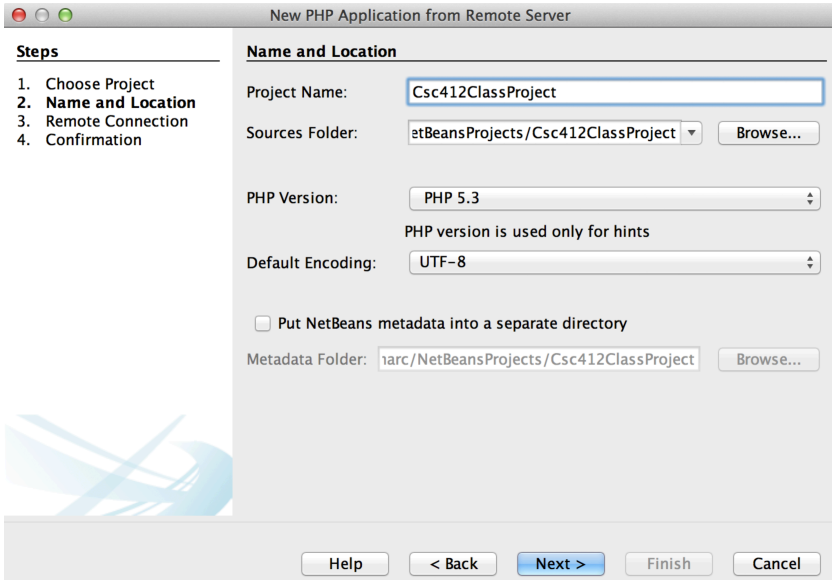

Figure 18. Name the project.

4. Name the project "Csc412ClassProject" (Figure 18). Press Next to continue.

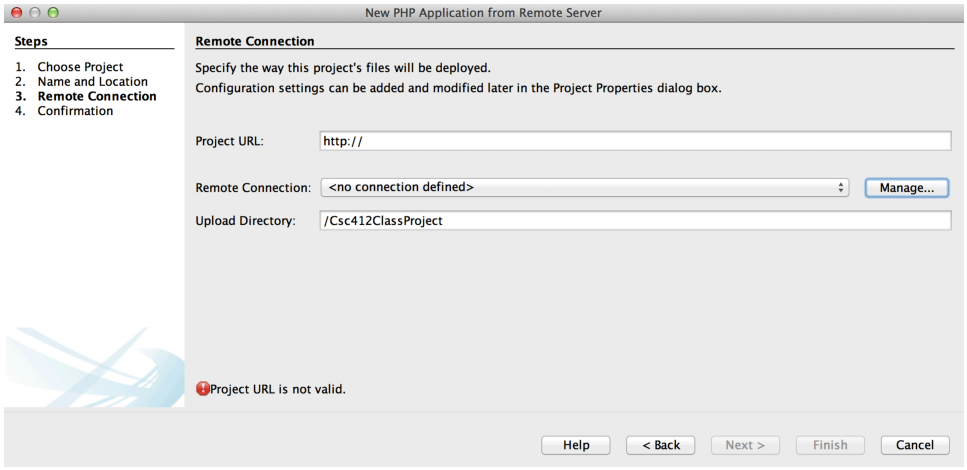

Figure 19. "Normal" error message when connecting to remote server.

5. The next dialog box ("New PHP Application from Remote Server") will indicate at the bottom that the "Project URL is not valid" (Figure 19). This is normal (even if it is poor UX design). To continue, you need to create a remote connection to your account on thecity; to do this click on the "Manage…" button to the right of the window.

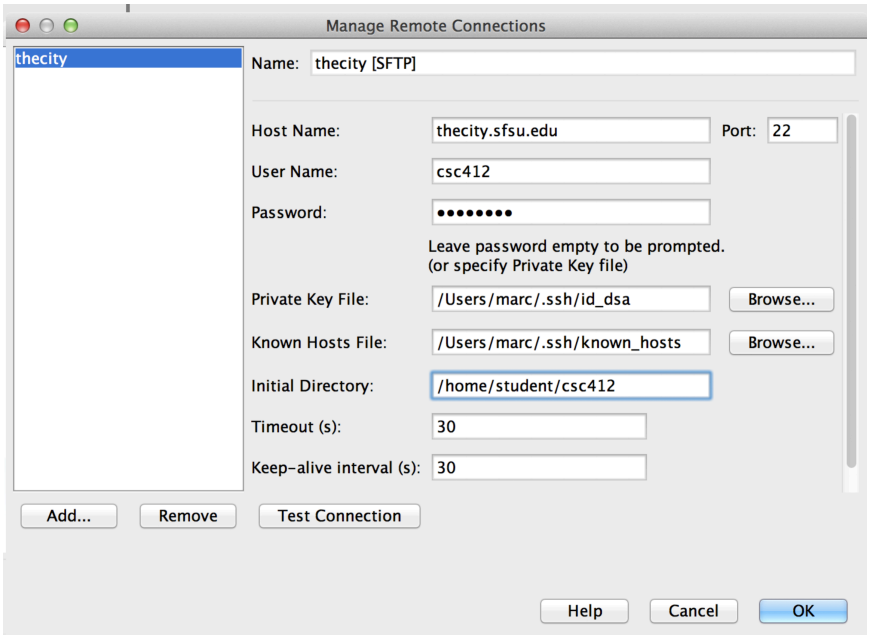

Figure 20. Creating a remote connection to thecity.

6. The "Manage Remote Connections" box will appear (Figure 20). Click the "Add…" button in the lower-left.

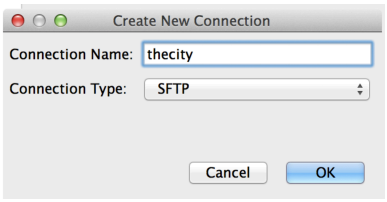

Figure 21. Create New Connection box.

7. When the "Create New Connection" box appears (Figure 21)
    a. Select SFTP for connection type
    b. Name the connection name "thecity"
    c. Press OK to continue
8. The "Manage Remote Connections" box will again appear (Figure 20), this time with blank fields. Enter the following values:
    a. for Host Name, type `thecity.sfsu.edu`,
    b. the Port should be 22,
    c. for User Name type your thecity username,
    d. for Password enter your thecity password.
    e. for Initial Directory, type `/home/student/`*`directory`*, where *`directory`* is your username on thecity.
    f. Be sure to leave the Private Key File and Known Hosts File settings as they are.
9. When you have entered this information, press the Test Connection Button.
10. You may receive warnings about the authenticity of thecity.sfsu.edu. Dismiss these warnings.

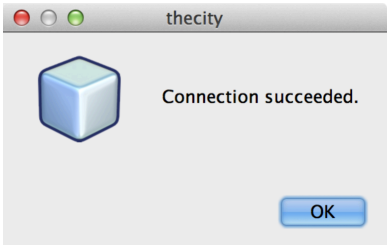

Figure 22. Successful connection.

11. You should receive a message indicating a successful test.

a. If you receive a successful test message, dismiss the message and then click on OK in the Manage Remote Connection Dialog. This will return you to the previous dialog box (Figure 19), but thecity should be selected as your remote connection. DO NOT PRESS NEXT, but rather continue at the next numbered step.

b. If you do not receive a successful test message, check your settings from steps 7 and 8 again.

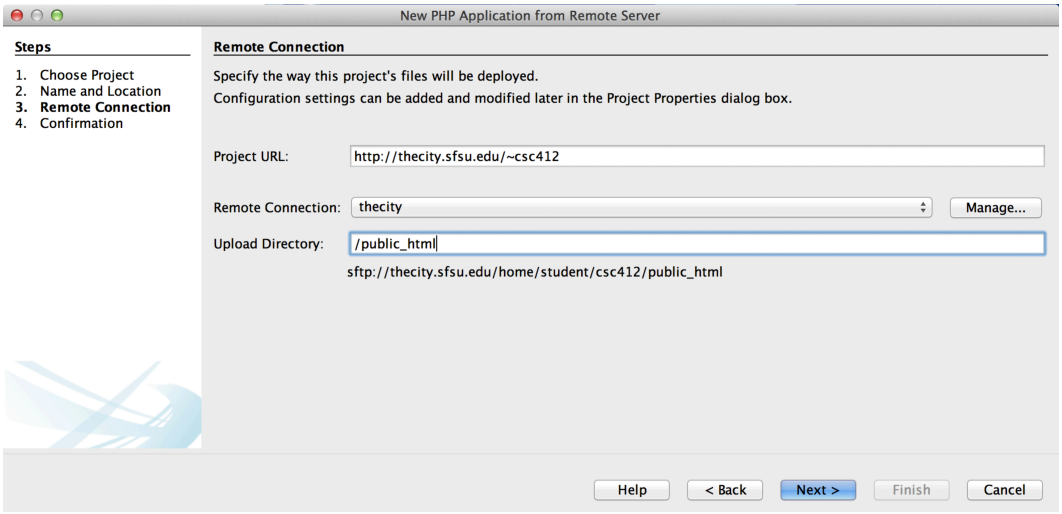

Figure 23. Completing the information for a remote connection.

12. Finish entering the information for the remote connection (Figure 23):

a. For the project URL, type the URL of your web on thecity, `http://thecity.sfsu.edu/~username`, where *`username`* is your username on thecity.

b. For the upload directory, use type `/public_html`

c. Press Next to continue.

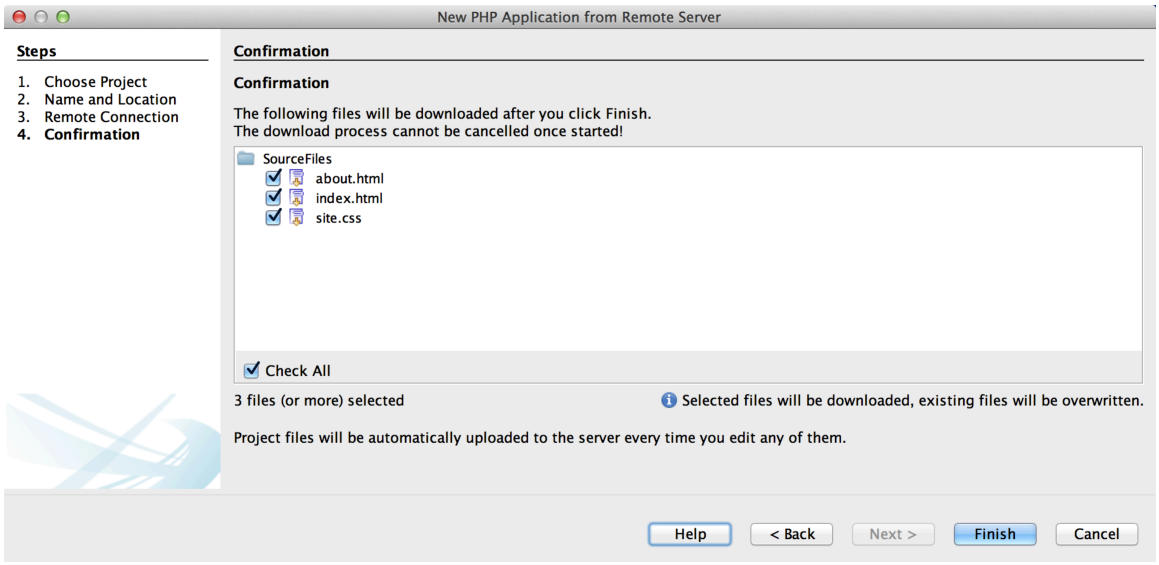

Figure 24. Confirmation box when importing a project.

13. NetBeans will now download the existing files from your account, after asking you to confirm what is being downloaded (Figure 24). Press Finish to download your existing project.

    You may receive warnings about the authenticity of thecity, which you may dismiss.

Edit your site using NetBeans

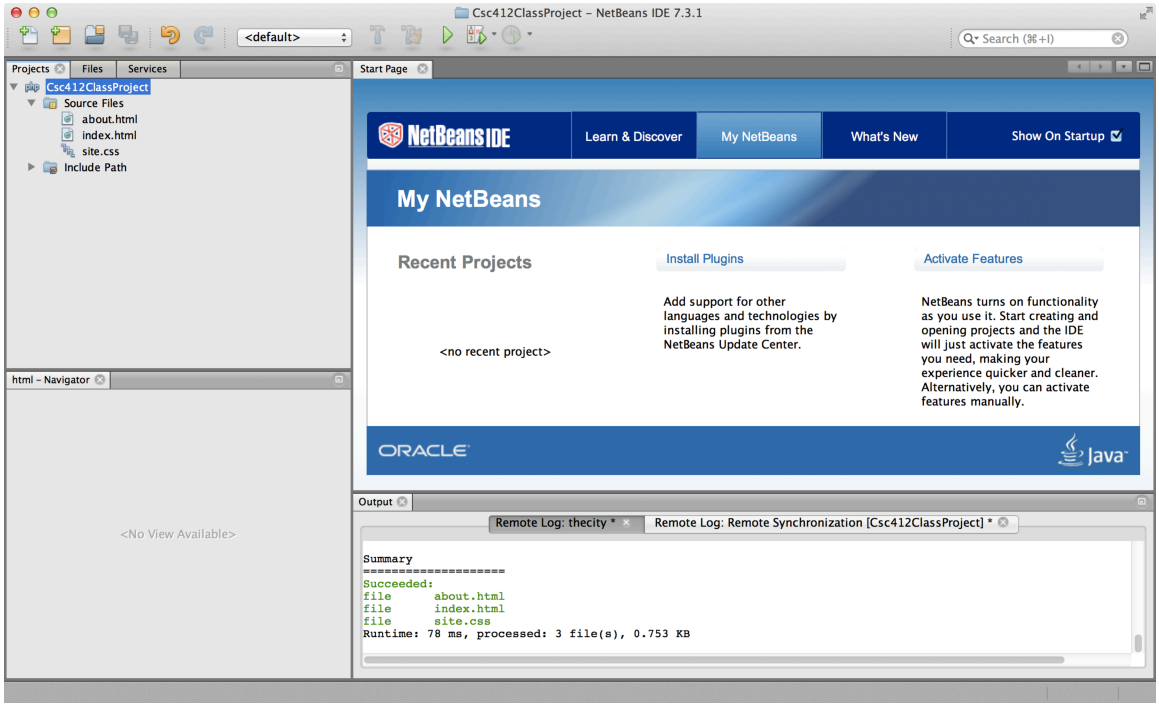

Figure 25. NetBeans after your site has been imported.

1. When NetBeans is finished downloading the files from your site, the page will appear similar to Figure 25. The upper-left panel is the project browser, which shows the files in your project. The upper-right panel is where you will edit your pages.

2. Open the index.html file by double clicking on index.html in the upper-left panel.

3. Start editing your index.html file. Insert a line at the top of index.html, and type `<!DOCTYPE html>`.

4. Above the line that contains </head>, insert a line and type

```
<link rel="stylesheet" type="text/css" href="site.css">
```

5. Edit the file, making sure that the rest of your page content between </head> and </html> is contained in a pair of <body> </body> tags.

6. Above the </body> tag, type

```
<div>
    <a href="about.html">about this site</a>
</div>
```

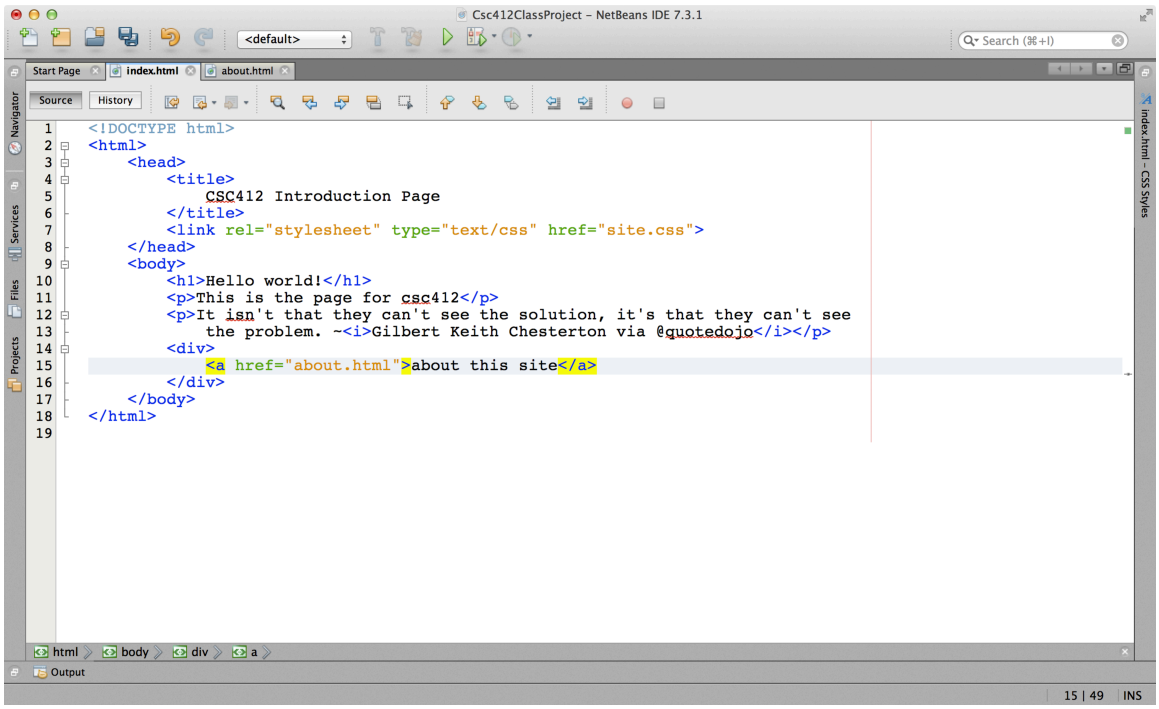

Figure 26. The index.html page after editing in NetBeans.

7. After you're done typing, reformat the page. With the source code window selected, on Mac type ctrl⇧F, on Windows, type alt-⇧-F. Your page should appear similar to the one in Figure 26.

8. Save the file (⌘S on Mac, ctrl-S on Windows).

9. Go to your browser, and browse to `http://thecity.sfsu.edu/~username`, where username is your *username* on thecity.

10. Click on the "about this page" link toward the bottom of the page.

11. Return to NetBeans, and edit the about.html file.

12. Edit the line

```
<h1 class="centered">About</h1>
```

    to read

```
<h1 class="centered sansSerif">About</h1>
```

13. Save the file.

14. Return to your browser, and press the refresh button.

Lab 2 References

Comparison of text editors:

http://en.wikipedia.org/wiki/Comparison_of_text_editors

TextWrangler editor: http://www.barebones.com/products/textwrangler/

Sublime editor: http://www.sublimetext.com/

IDEs: http://en.wikipedia.org/wiki/Integrated_development_environment

NetBeans: http://netbeans.org

Editor wars: http://en.wikipedia.org/wiki/Editor_war

vi: http://www.unix-manuals.com/tutorials/vi/vi-in-10-1.html

vi cheat sheet:

http://www.math.ntu.edu.tw/~wwang/mtxcomp2010/cssd2011/download/vi_cheat%20sheet.pdf

emacs: http://www2.lib.uchicago.edu/keith/tcl-course/emacs-tutorial.html

emacs cheat sheet: http://www.rgrjr.com/emacs/emacs_cheat.html