

Introduction
Text Editors vs. Word Processors
GUI Text Editors
Terminal Text Editors
Integrated Development Environments (IDEs)
Lab 2 Preparatory Work
Install NetBeans
Lab 2 Classroom Work
Edit site.css Using vi
Edit an HTML File Using Emacs
Edit Your Site Using NetBeans
References

2 — Lab 2 — IDEs and Editors

2.1 Introduction

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

2.2 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 will not appear in the text file that was being edited.

2.3 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 files to your local computer, edit them, then upload the edited files to the remote server.

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

2.4 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, or change configuration a 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 Editor Wars

2.5 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 a developer's efficiency and code reliability.

At a minimum, for a language such as C++, modern IDEs include integrated(text) editor, debugger, and compiler. Some provide interpreters containing real-time feedback during coding,highlighting potential errors before source 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.

2.6 Lab 2 Preparatory Work

2.6.1 Install NetBeans

1. Go to NetBeans Download Page 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

2.7 Lab 2 Classroom Work

2.7.1 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 the 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 the AWS-Server (if you aren't already).
 2. Go to your `public_html` directory.
 3. Copy the file `~csc412/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 2.1: Editing site.css in Vi.

6. Vi opens and displays the file. There are fewer lines of text in the than there is space on the screen. Vi denotes empty lines at the end of a file as tildes (Figure 2.1).
 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 arrows keys don't work in command mode you can use the 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 the **i** after pressing **Esc**. When vi enters insert mode, you should see an **-- INSERT --** message appear at the bottom left of the window.

- Type the following text into site.css:

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

- To exit insert mode, press the **Esc** key. The **-- INSERT --** mess will disappear.

```
1 root {
2     display: block;
3 }
4
5 .centered {
6     text-align: center;
7 }
8
9 .sansSerif{
10    font-family: sans-serif;
11 }
```

"site.css" 11L, 110C written 8,0-1 All

Figure 2.2: Vi displaying information about the successfully written file.

- Now that you are back in command mode, type **:w Esc** to write the modifier file to the disk (Figure 2.2). Vi will tell you that it wrote the file, and how large it was.
- Type **:q Esc** and, provided you have not modified the file since the previous step, vi will quit and return you to the UNIX command prompt.

2.7.2 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**, "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.

Mac Users: Configure mac Terminal to work with Emacs

In order to use Emacs, special character sequence 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 **⌘ J**), and select the keyboard tab.

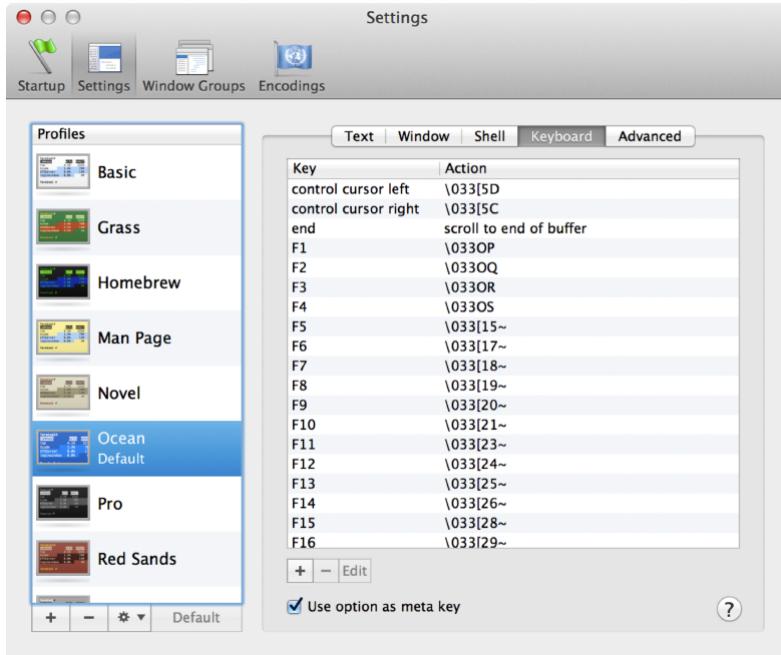


Figure 2.3: Configure Mac Terminal for use with Emacs

3. Make sure the "use option as a meta key" box is checked (Figure 2.3).
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 the AWS-Server and in your *public_html* directory.
2. Copy the *about_proto.html* file from *~csc412/* to your current directory, renaming it *about.html*. Do you know how to do this using only the **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://csc412.sfsu.com~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.

2.7.3 Edit Your Site Using NetBeans

Even though it might not seem like much, you already have an existing website on the AWS-Server. In this step you will download the source files of your website 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 AWS-Server, 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.

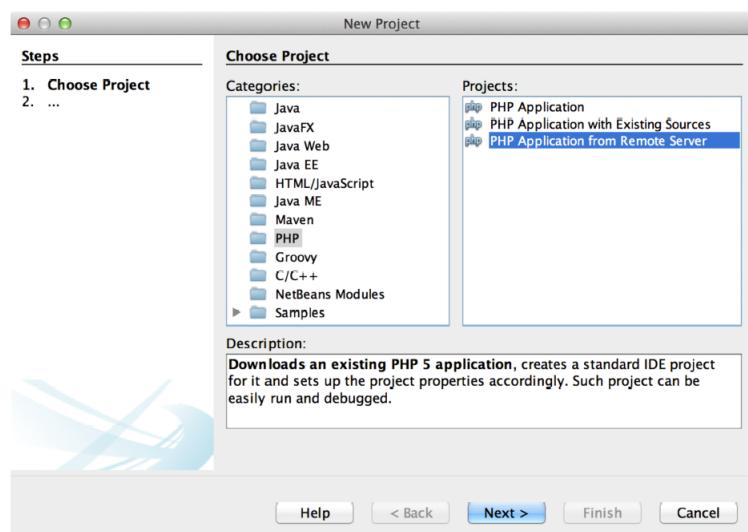


Figure 2.4: Create a new PHP project in NetBeans.

3. Create a new NetBeans project, by **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 2.4).

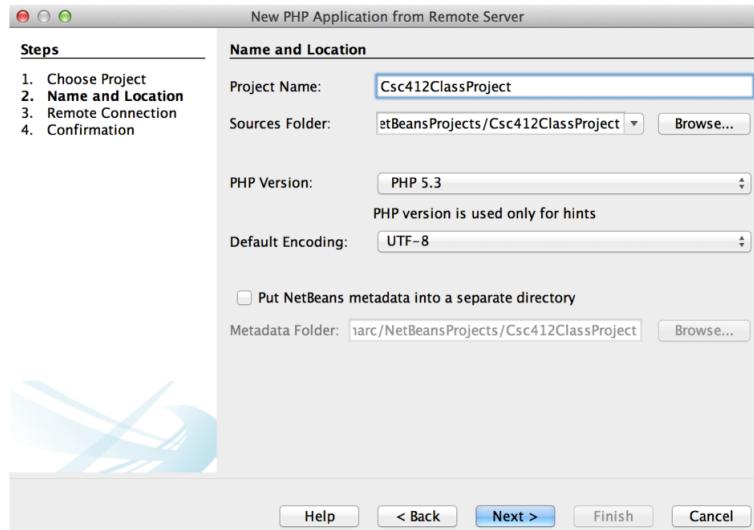


Figure 2.5: Name the Project.

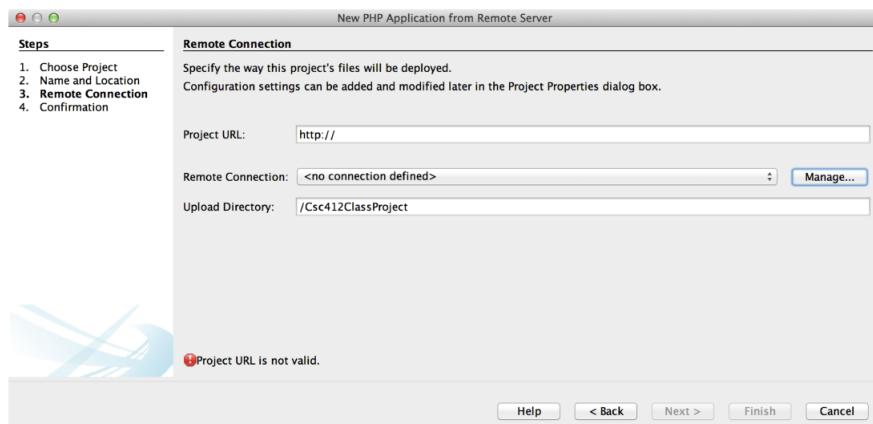


Figure 2.6: "Normal" error message when connecting to remote server.

4. Name the project "Csc412ClassProject" (Figure 2.5). Press next to continue.
5. The next dialog box ("new PHP Application from Remote Server") will indicate at the bottom that the "Project URL is not valid" (Figure 2.6). This is normal (even if it is poor UX design). To continue, you need to create a remote connection to your account on the AWS-Server. To do this click on the "Manage..." button to the right of the windows.

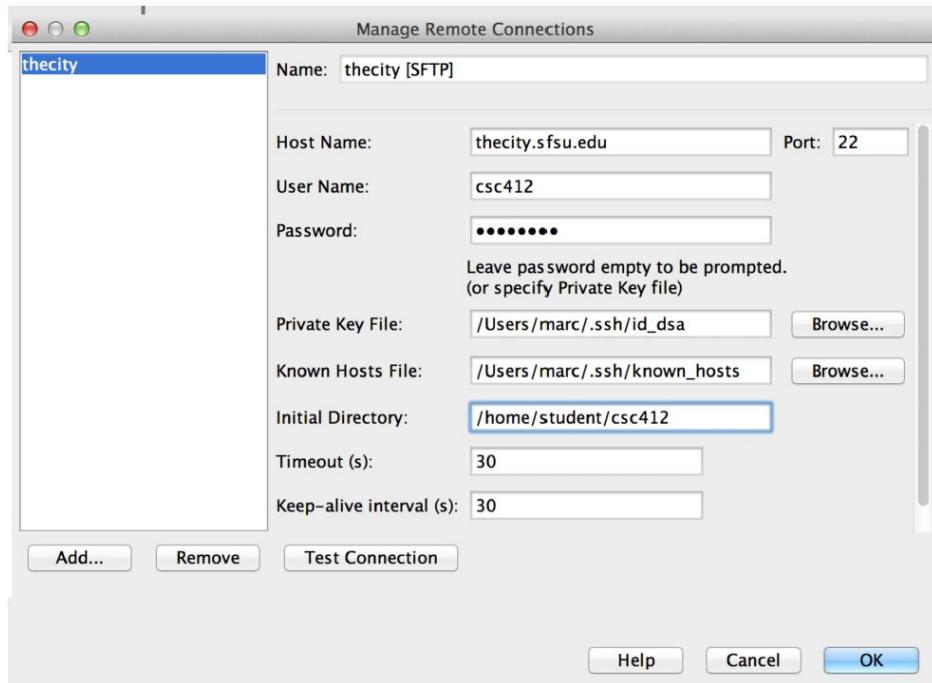


Figure 2.7: Creating a remote connection to the AWS-Server.

6. The "Manager Remote Connections" box will appear (Figure 2.7). Click the "Add.." button to the lower-left.

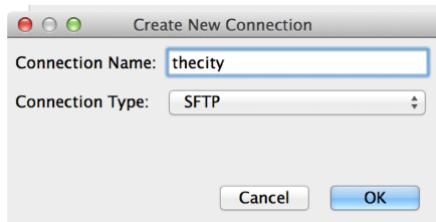


Figure 2.8: Create New Connection Box

7. When the "Create New Connection" box appears (Figure 2.8)
 - Select SFTP for connection type.
 - Name the connection name "aws-server"
 - Press OK to continue
8. The "Manage Remote Connections" box will appear again (Figure 2.7), this time with blank fields. Enter the following values:
 - for Host Name, type `csc412sfsu.com`
 - the Port should be 22
 - for User Name type your AWS-Server username
 - for Password enter your AWS-Server password
 - for Initial Directory, type `/home/directory`, where `directory` is your username on the AWS-Server.
 - 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 the AWS-Server. Dismiss these warnings

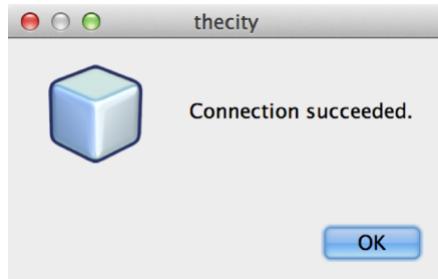


Figure 2.9: Successful connection

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 the OK in the Manage Remote Connection Dialog. This will return you to the previous box (Figure 2.6), but aws-server 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.

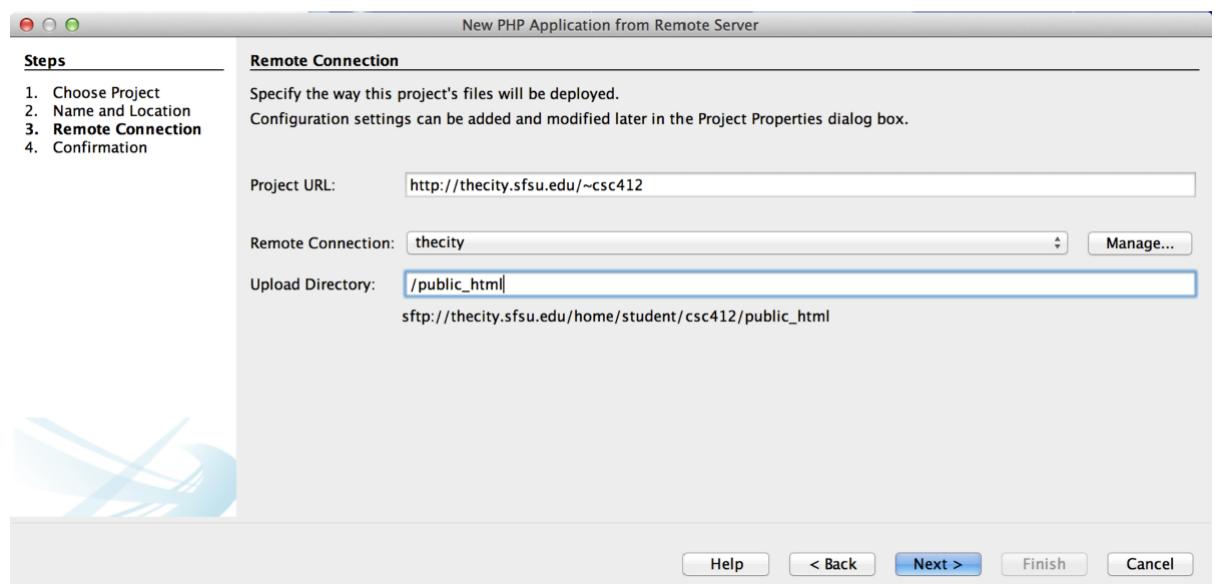


Figure 2.10: Completing the information for remote connection

12. Finish entering the information for a remote connection.
- For the project URL, type the URL of your web on the AWS-Server,
<http://csc412sfsu.com/~username> where `username` is your username on the AWS-Server.
 - For the upload directory, type `/public_html`
 - Press Next to continue.

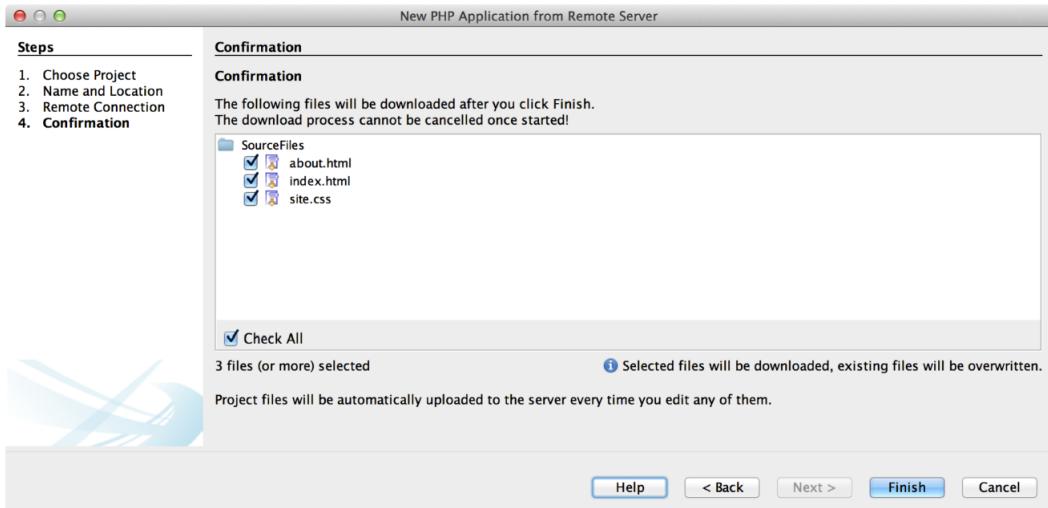


Figure 2.11: Confirmation box when importing a project

13. NetBeans will now download the existing files from your account, after asking you to confirm what is being downloaded (Figure 2.11). Press Finish to download your existing project. You may receive warnings about the authenticity of the AWS-Server, which you may dismiss.

Edit your site using NetBeans

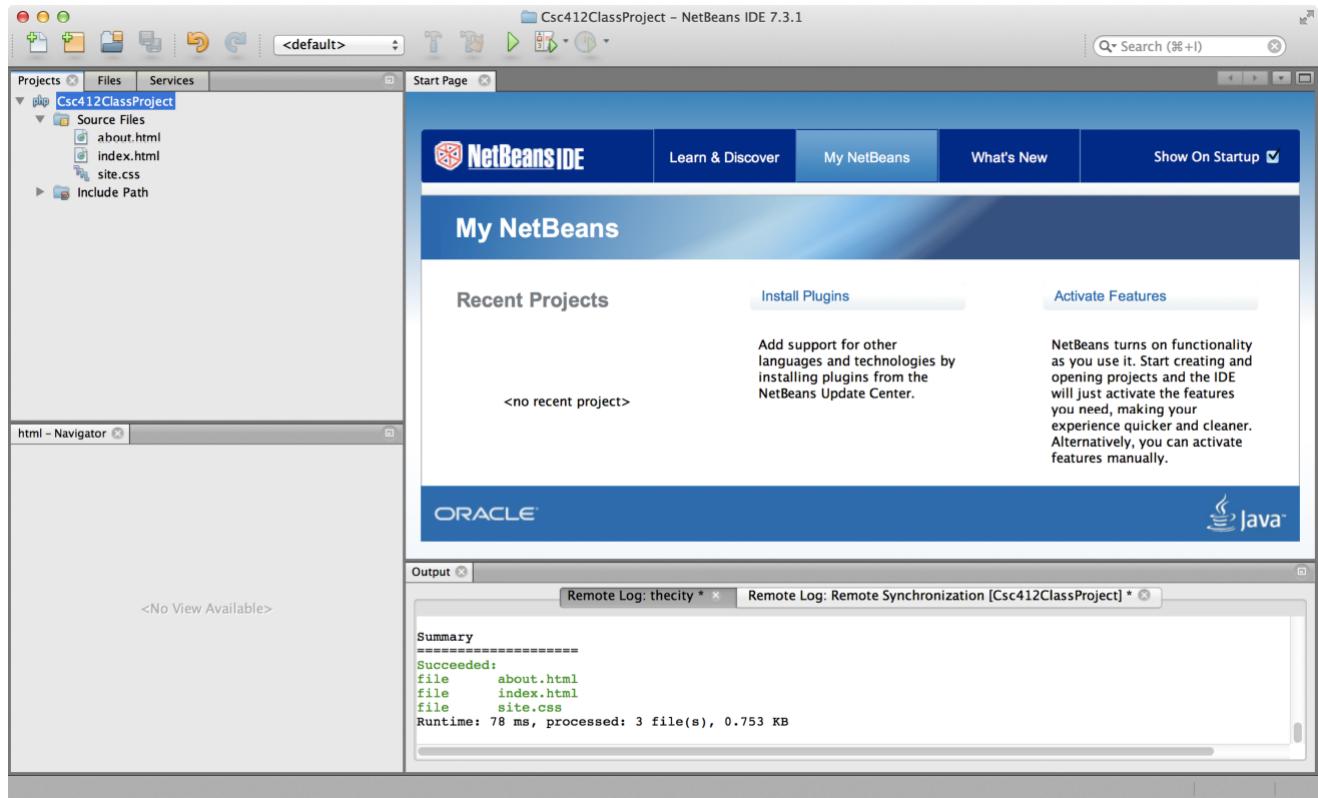


Figure 2.12: NetBeans after your site has been imported.

- When NetBeans is finished downloading the files from your site, the page will appear similar to (Figure 2.12). 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.
- Open the index.html file by double clicking on index.html in the upper-left panel.
- Start editing your index.html file. Insert a line at the top of index.html, and type

```
<!DOCTYPE html>
```

- Above the line that contain </head>, insert a line and type:

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

- 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.

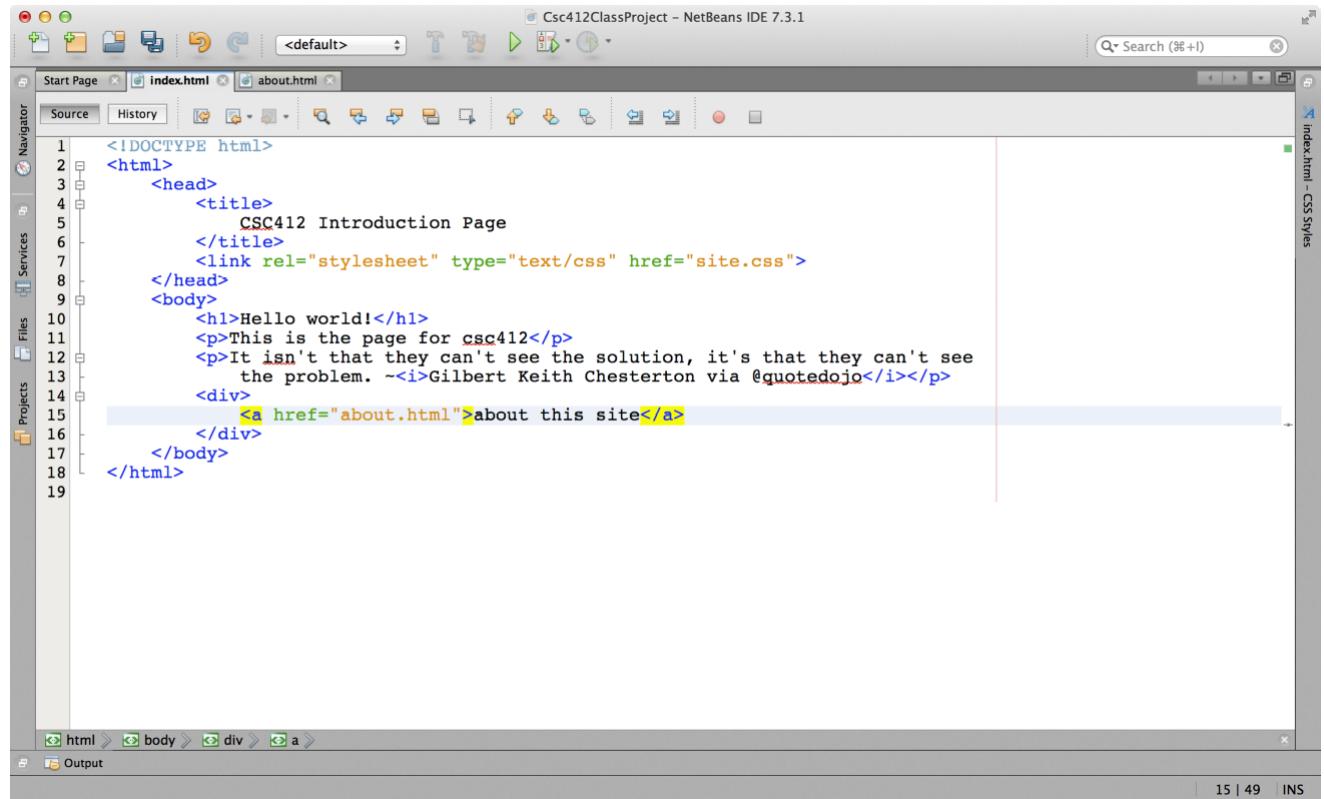


Figure 2.13: The index.html page after editing in NetBeans.

6. Above the </body> tag, type:

```

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

```

7. After you're done typing, reformat the page. With the source code window selected, on Mac type **[ctrl]+[↑]+[F]**, on Windows **[Alt]+[↑]+[F]**. Your page should appear similar to the one in Figure 2.13.
8. Save the file (**[⌘]+[S]** on Mac, **[ctrl]+[S]** in Windows).
9. Go to your browser, and browse to:
<http://csc412sfsu.com~username>,
where username is your username on the AWS-Server.
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>
```

so that it reads

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

13. Save the file.

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

2.8 References

- Comparison of test editors
- TextWrangler editor
- Sublime editor
- IDEs
- NetBeans
- Editor Wars
- vi
- vi cheat sheet
- emacs
- emacs cheat sheet