

Web Page Foundations

Overview

This lab walks you through creating and deploying a simple web page. The web page you create in this lab will have no functionality yet. It just contains many of the html elements you will see on most web pages today. We will turn this web page into a working web application next week. A text editor will be used to create the web page. You are welcome to use an html editor or Integrated Development Environment (IDE) to help you generate the web pages if you like. Please be sure you have read the “Creating Web Pages” competencies prior to completing this Lab. The online textbook has many html code examples that will help you become comfortable with the most popular html tags.

Learning Outcomes:

At the completion of the lab you should be able to:

1. Create a web page comprised of formatted text, images, lists, tables, hyperlinks and forms.
2. Review and analyze Apache Web server logs notating http access, http methods and http error codes

Lab Submission Requirements:

After completing this lab, you will submit a word (or PDF) document that meets all of the requirements in the description at the end of this document. In addition several html and image files along with the Apache2 access.log file will be submitted. You can submit all files in a zip file.

Virtual Machine Account Information

Your Virtual Machine has been preconfigured with all of the software you will need for this class. The default username and password are:

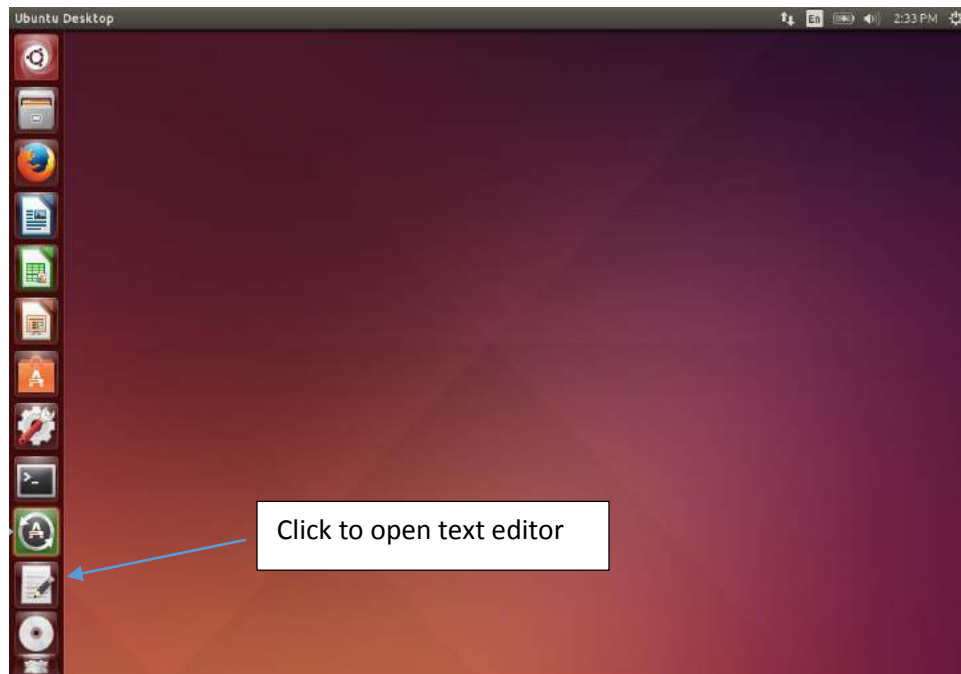
Username : umucsdev

Password: umuc\$d8v

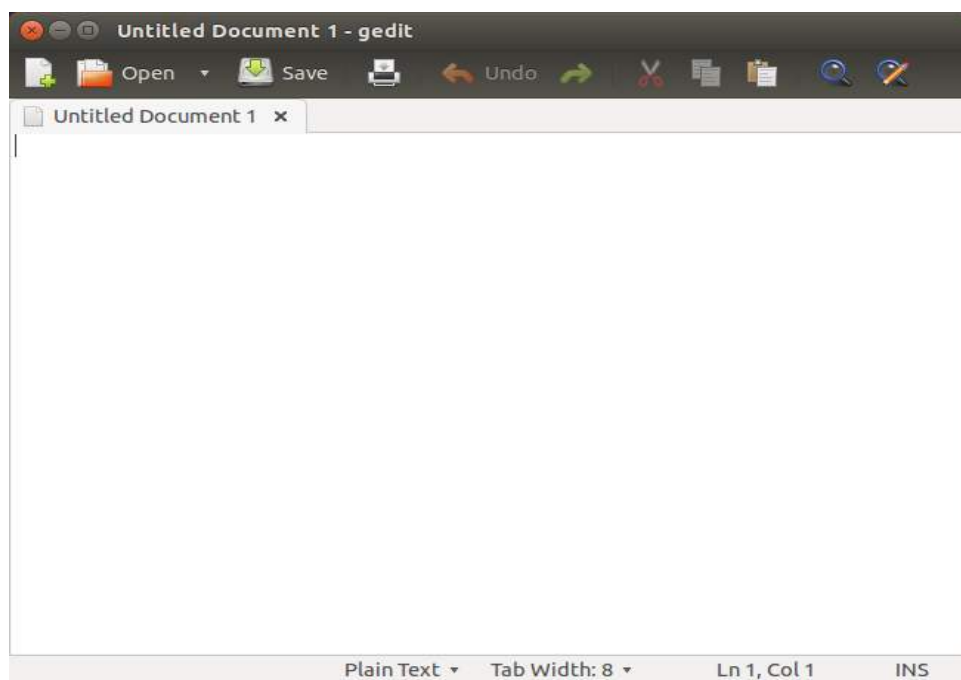
Part 1 – Create a Web page

We will use the gedit text editor to create the web page. The web page will resemble a company home page with an introduction, some formatted text, links to other web pages, images and a form designed to gather customer information.

1. Assuming you have already launched and logged into your SDEV32Bit Virtual Machine (VM) from the Oracle VirtualBox, click on the gedit icon found on the left side of the screen of your VM.



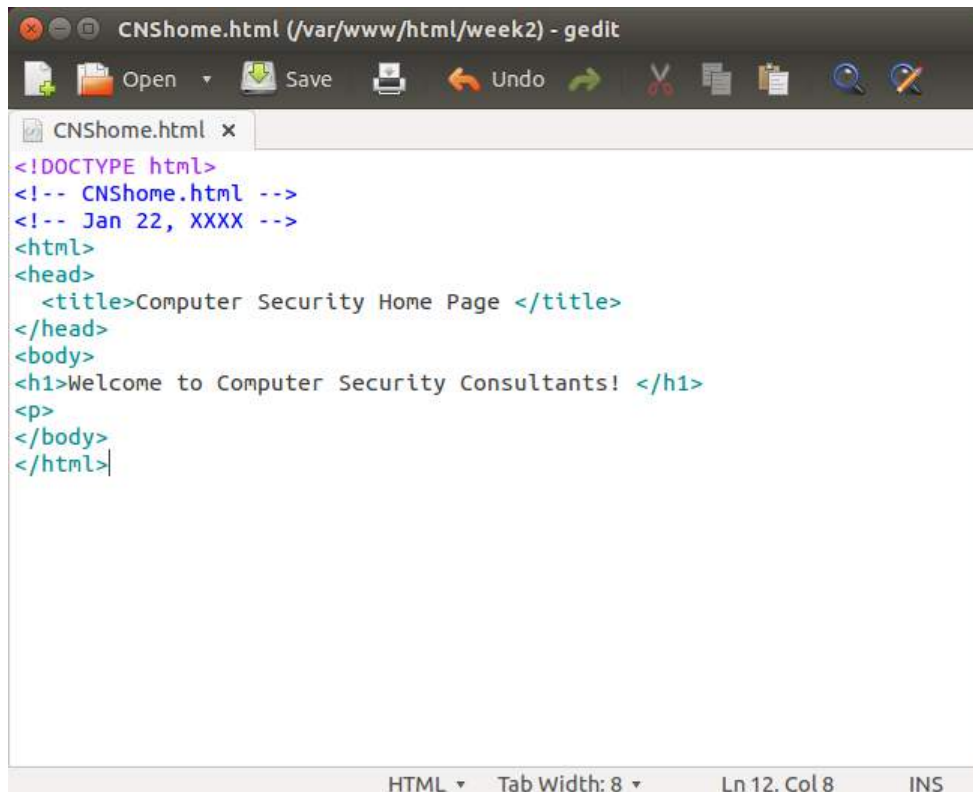
2. After clicking the terminal icon a terminal will appear



3. To create a new document just begin typing or copying and pasting the html code from the examples. We will create the web page in several steps adding a few paragraphs and sections at time. Viewing the web page between each step will help minimize errors in the html code. To add the first section of the html web page copy and paste the following html code into the gedit editor:

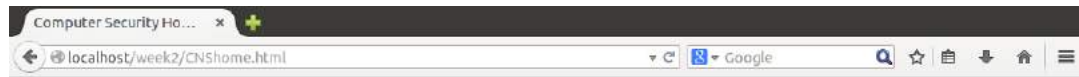
```
<!DOCTYPE html>
<!-- CNShome.html -->
<!-- Jan 22, XXXX -->
<html>
<head>
  <title>Computer Security Home Page </title>
</head>
<body>
<h1>Welcome to Computer Security Consultants! </h1>
<p>
</body>
</html>
```

Save the file in the /var/www/html/week2 folder in a file named CNShome.html. Note, you may need to create a folder named week2. Recall the /var/www/html is the location of the Apache2 web server html files. Creating separate folders for each week or application will help organize the server.



```
CNShome.html (/var/www/html/week2) - gedit
Open Save Undo
CNShome.html x
<!DOCTYPE html>
<!-- CNShome.html -->
<!-- Jan 22, XXXX -->
<html>
<head>
  <title>Computer Security Home Page </title>
</head>
<body>
<h1>Welcome to Computer Security Consultants! </h1>
<p>
</body>
</html>
HTML Tab Width: 8 Ln 12, Col 8 INS
```

Launch the Firefox browser and run your home page by entering the following URL:
localhost/CNShome.html



4. Continue to add more html code to the html file. The following code will add a table of hyperlinks to popular vendor websites:

```
<!-- Add Table of Hyperlinks -->
<p>
Click on any link in the table below to see some of our current customers:
</p>
<table border = "1">
<tr><td>Site</td><td>Web Address</td></tr>
<tr><td>UMUC</td><td><a href="http://umuc.edu">UMUC</a></td></tr>
<tr><td>Oracle</td><td><a href="http://oracle.com">Oracle</a></td></tr>
<tr><td>Microsoft</td><td><a href="http://www.
microsoft.com">Microsoft</a></td></tr>
<tr><td>Twitter</td><td><a href="http://www.
twitter.com">Twitter</a></td></tr>
</table>
```

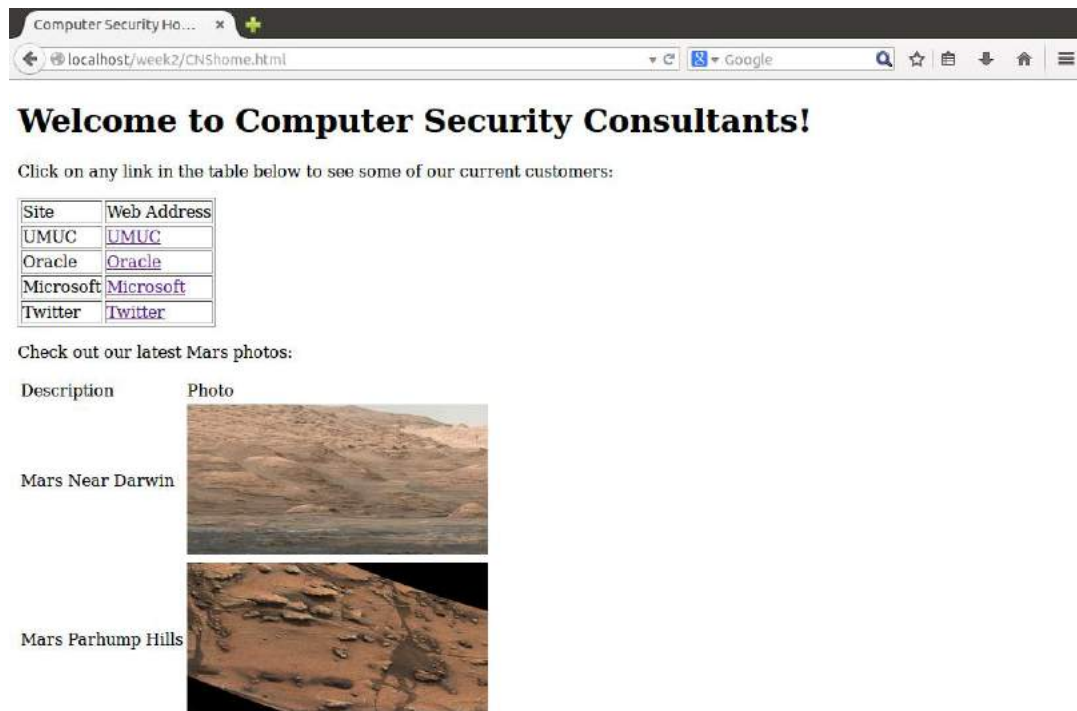
Be sure you place this code before the end `</body>` tag. When you add this, save the file and refresh the browser, the web page will take this appearance. Test the hyperlinks to verify they reach the expected websites.



5. Adding the following html code will provide images formatted in a table.

```
<!-- Add some images in a table -->
<p>
Check out our latest Mars photos:
</p>
<table>
<tr><td>Description</td><td>Photo</td></tr>
<tr><td>Mars Near Darwin</td><td></td></tr>
<tr><td>Mars Parhump Hills</td><td></td></tr>
</table>
```

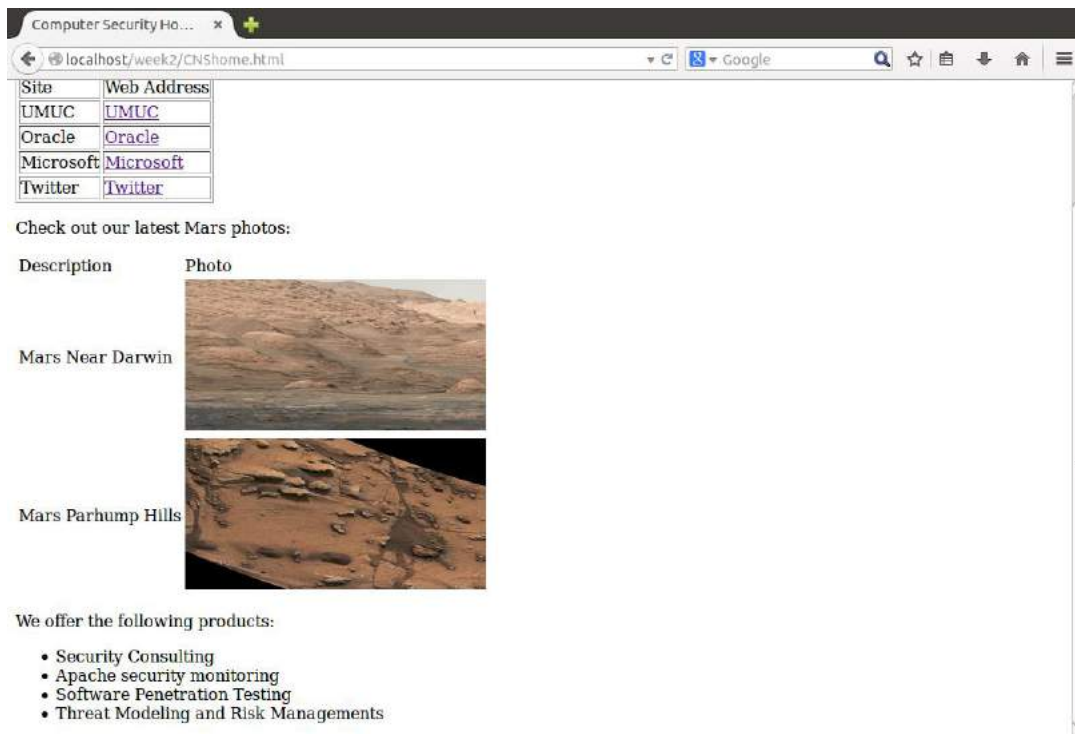
Saving and refreshing the file results in the following web page:



6. Unordered lists can be added with the following code:

```
<p>
We offer the following products:
<ul>
  <li>Security Consulting </li>
  <li>Apache security monitoring</li>
  <li>Software Penetration Testing</li>
  <li>Threat Modeling and Risk Managements </li>
</ul>
</p>
```

Saving and refreshing the file results in the following web page:



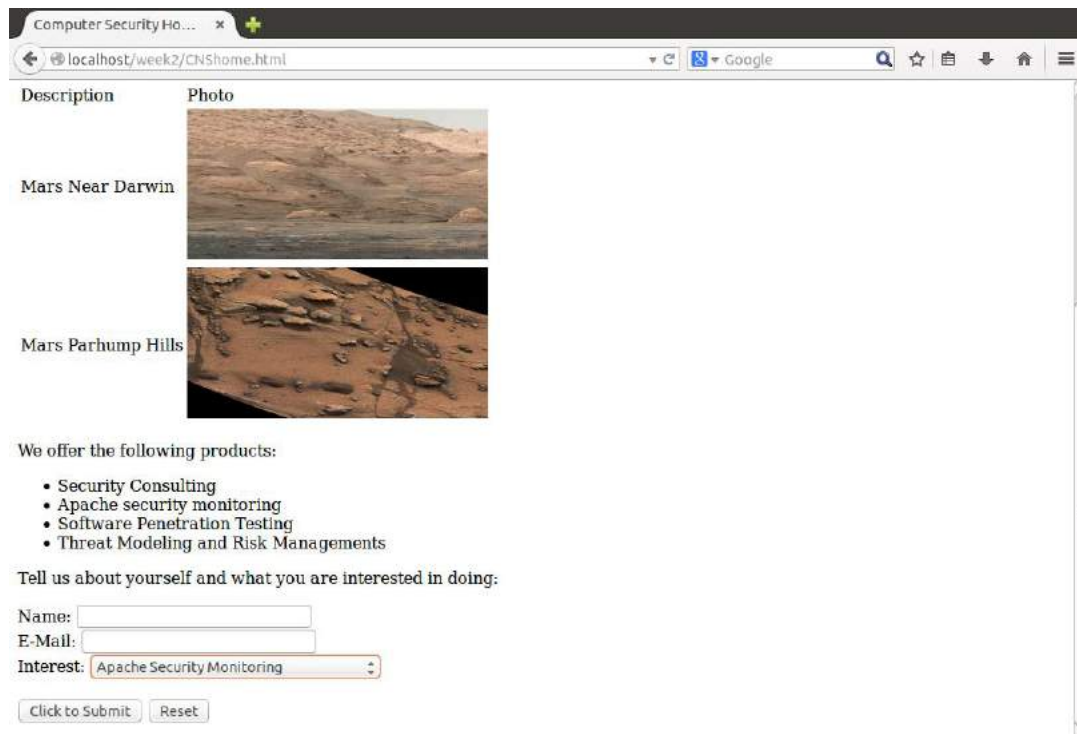
- Finally, we can add a form to collect information about the visitors to site. Keep in mind, we haven't implemented any functionality for saving or processing the form data yet. We will do that next week.

```
<!-- Add a Form -->
<p> Tell us about yourself and what you are interested in doing:
<form action="" method="post">
Name: <input type="text" name="username"></br>
E-Mail: <input type="text" name="e-mail"></br>
Interest: <select name="sport">
<option>Apache Security Monitoring</option>
<option>Security Consulting</option>
<option>Software Penetration Testing</option>
<option>Threat Modeling and Risk Management</option>
</select>
<br/><br/>

<input type="submit" value="Click to Submit"/>
<input type="reset" value="Reset"/>
</form>
</p>
```

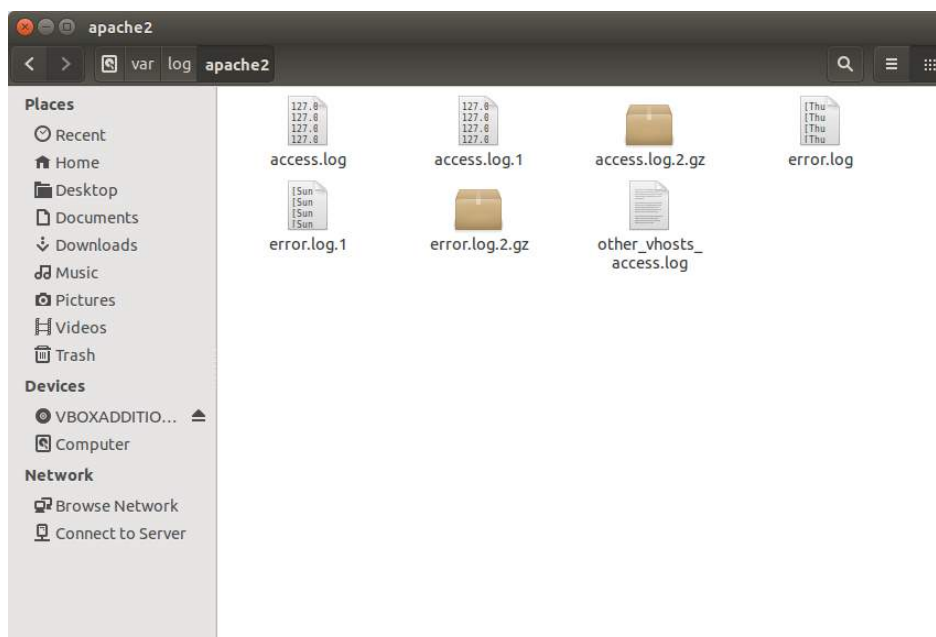
Saving and refreshing the file results in the following web page:

Note: You may need to scroll down on the web page to see the html form.



Part 2 Review and Analyze Apache2 Web Server Logs

By default, the Apache Web Server will log http traffic to a file. The file is saved in the `/var/log/apache2` folder. The log file with access information is named `access.log`. Additional files for error logging and back-ups versions of the log files may also be present in the directory. For this class, we will look primarily at the `access.log` file.



Opening the access.log file in the texteditor shows a log of all transaction along with time stamps and additional http details.

A typical line in the file may look like this:

```
127.0.0.1 - - [22/Jan/2015:17:04:56 -0500] "GET /week2/CNShome.html HTTP/1.1"
200 845 "-" "Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101
Firefox/31.0"
```

The line contains the IP address of the machine that accessed the server and the time. The IP address in this case is 127.0.0.1 which is the loopback or localhost for your machine. The next part of the line provides the http request verb get, the actual URL accessed, the http code of 200 and the number of bytes returned (845). Additional informal provides the browser type, operating system and other browser details of the machine accessing the server.

Log analysis is a critical part of web application security. In this exercise you should open the file and review each line paying careful attention to the http codes.

Extracting the lines of code related to running the CNShome.html file shows how the size of the bytes grew each iteration. This makes sense because we were building the web page and testing after each step.

```
127.0.0.1 - - [22/Jan/2015:20:07:56 -0500] "GET /week2/CNShome.html HTTP/1.1"
200 496 "-" "Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101
Firefox/31.0"
127.0.0.1 - - [22/Jan/2015:20:07:56 -0500] "GET /favicon.ico HTTP/1.1" 404
498 "-" "Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101
Firefox/31.0"
127.0.0.1 - - [22/Jan/2015:20:07:56 -0500] "GET /favicon.ico HTTP/1.1" 404
498 "-" "Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101
Firefox/31.0"
127.0.0.1 - - [22/Jan/2015:20:13:20 -0500] "GET /week2/CNShome.html HTTP/1.1"
200 698 "-" "Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101
Firefox/31.0"
127.0.0.1 - - [22/Jan/2015:20:16:14 -0500] "GET /week2/CNShome.html HTTP/1.1"
200 698 "-" "Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101
Firefox/31.0"
127.0.0.1 - - [22/Jan/2015:20:16:45 -0500] "GET /week2/CNShome.html HTTP/1.1"
200 895 "-" "Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101
Firefox/31.0"
127.0.0.1 - - [22/Jan/2015:20:19:43 -0500] "GET /week2/CNShome.html HTTP/1.1"
200 996 "-" "Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101
Firefox/31.0"
127.0.0.1 - - [22/Jan/2015:20:24:11 -0500] "GET /week2/CNShome.html HTTP/1.1"
200 1200 "-" "Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101
Firefox/31.0"
```

As you review the log file, you may see some unusual traffic. For example, the favicon.ico with the 404 error code. This is a browser search for a vendor specific icon that is placed in the URL and in the tab of the browser window. If no favicon.ico is found in the root directory of the web server, the 404 error will appear. If the favicon.ico is found it will be displayed.

If you attempt to access a page that doesn't exist, you will most likely see a 404 error code in the log file.

```
127.0.0.1 - - [22/Jan/2015:20:53:38 -0500] "GET /testing.html HTTP/1.1" 404
500 "-" "Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101
Firefox/31.0"
```

The user will see a message similar to the following screen by default. Redirection to a more user-friendly error pages is possible.



Lab submission details:

As part of the submission for this Lab, you will create your own Web page that uses most of the functionality discussed in this week including formatted text, hyperlinks, images, tables, lists and forms. The forms do not need to be functional for this lab.

Specifically, you need to create a well-organized Web page that provides a brief overview of a fictitious e-Commerce company that sells a themed product line of your choice. The web page should list the products along with their images, descriptions, and price. You should provide at least 12 products on your Web page. Each product should have a hyperlink that when clicked provides more details about the product selected. An order form should be included to provide a simple interface to select the quantity of each of the products. The form should also allow the customer to fill out their firstname, lastname, street address, city, state, zipcode, credit card type (Mastercard, Visa, American Express), credit card number and expiration date. The form should include a field where the customer can provide up to 300 words of comments. The form should be user-friendly taking advantage of drop down list components, radio buttons, and check boxes as appropriate.

As you are developing your Web page, be sure to build in small steps and test as you go along. This will ensure your access.log file will provide meaningful and accurate documentation of your test and build process. Review your access.log file and provide a brief summary to include analysis of http codes, request methods and timeline for testing.

Create screen shots showing the successful running of your e-Commerce site.

For your deliverables, you should submit a zip file containing your word document with screen shots and access.log analysis, html files, image files and access.log file. Note, you will most likely have multiple html pages and images for your submission.

Include your full name, class number and section and date in the document. You should name your document yournameSDEV300Lab3.docx (or yournameSDEC300Lab3.pdf) and submit the document to the Lab1 assignments area no later than the due date.

Hint: If you aren't sure what an e-Commerce site looks like, check out amazon.com. The level of detail found at amazon.com is not required (nor possible, with the knowledge shared to date) but the site may give you some design ideas and format considerations.