# *Web Programming III (420-C30-HR)*

# *Assignment 2 – node.js*

Date assigned: Wednesday, October 28, 2015

Date due: **Wednesday, November 18, 2015**

**Learning Objectives**

Upon successful completion of this assignment, the student will be able to:

* Create a functional web server using node.js
* Return static pages and data from a web server location depending on the request

To do:

**General idea of the assignment:**

Create a node.js web server. The server will return all kinds of data (html, images, css, xml, etc). The server will also return data from a file when requested from a form.

**Details:**

You are creating a web server using **native** node.js modules

1. The web server runs on port 9000
2. Static files are returned from the ./public folder or folders beyond that path.

You must do the following when the request is a GET:

1. Have the web server be able to return files of the following types with the following extensions:

|  |  |
| --- | --- |
| Extensions | Mime type |
| .html | text/html |
| .css | text/css |
| .js | application/javascript |
| .png | image/png |
| .jpg | image/jpeg |
| .gif | image/gif |
| .xml | text/xml |
| .txt | text/plain |

* 1. When no file name is specified, return the index.html file from the ./public folder.
  2. To do this, you need to create a JavaScript object of the extensions and the mime types.
  3. When the request is received, validate the extension (should be one command if you create the JavaScript object).
  4. Remember the browser will automatically request (GET) any linked files when the HTML page is returned.

1. If a file with a different extension is requested, return error 406 in a page.
2. If a non-existent file is requested, return error 404 in a page.
3. If an error is encountered in reading the file, return error 500 in a page.

You must do the following when the request is a POST:

1. Determine the filename of the file sent to server.
2. Create a user object and write the information as a comma separated file (CSV) to the file users.txt. The file is located in the ./data folder at the same level as the ./public folder.
3. Return the object saved in an appropriate (nicely formatted) HTML page. You can do this with native node.js or use a template such as mustache.
4. If an error is encountered writing the object, return the error page with error code 520, a message to the user and the error object.

The user object has the following attributes: firstName, lastName, username, emailAddress ad phoneNum.

**Other things**

1. The root folder MUST be called username\_C30\_A02 where username is your username.
2. You must create a user object as a node.js module. This must be in its own folder. Include the entire module and not just a file name within it.
3. I have provided a number of html files with embedded images, external CSS and JavaScript files. I have also provided a couple of text and XML files to test with as well.
4. There are marks for interface design. This only applies to the error pages and the returned page on the post.
5. Make the code efficient; there are marks for efficiency.
6. You must use HTML5 and (proper) CSS techniques for this assignment.

**To submit**

When you have completed the assignment zip the YourUserName\_C30\_A02 folder containing all the files for the assignment and copy it to the course page.