CSCI354 Assignment 4

NOTE: Be sure to make a copy of your code before working on this assignment! It may require significant changes to the markup, styles and JavaScript files. Use your copy for working on the assignment and leave the original intact for using in class. This also applies to all later assignments.

Part A

In your **webserver** part of the Chattrbox project, take the code that does the file reading and error handling and move it into its own module.

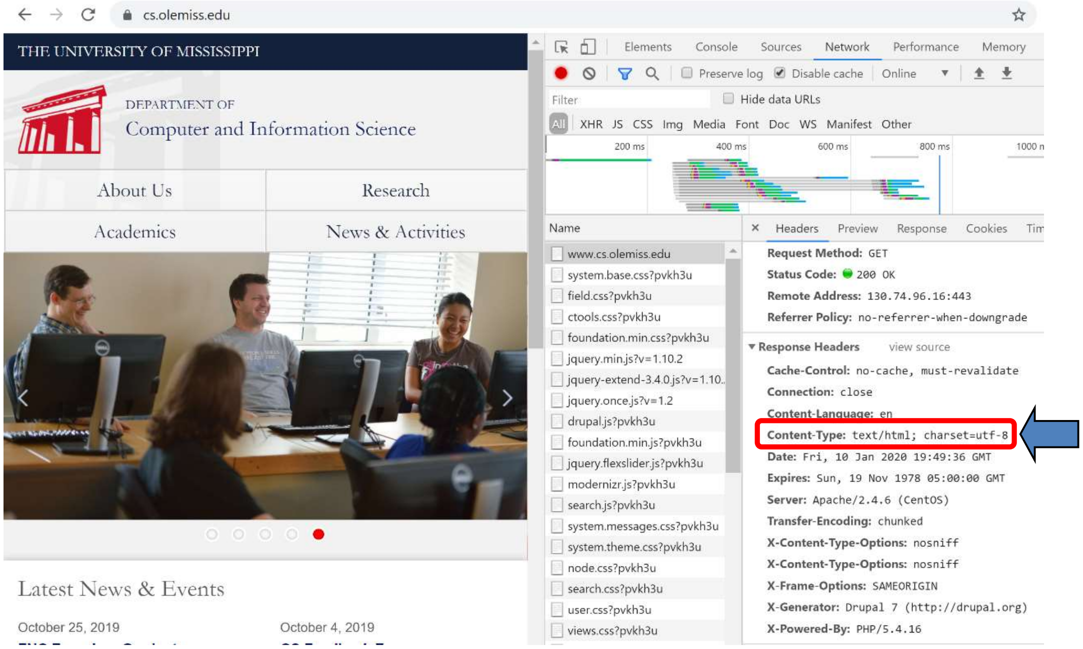
Also, make the module configurable so that when you import it you can specify what base folder the static HTML, CSS, and JavaScript are in.

Part B

You may have already notice that a computer knows to use a video player to open a movie file and use a document viewer to open a PDF. This is because your computer keeps a table of file types and the programs associated with those file types. It infers a file's type by looking at the file extension (e.g., .html or .pdf).

A browser needs those same associations so that it knows whether to render the response as HTML, use a plug-in to play music, or download a file to the hard drive. But HTTP responses do not have file extensions. Instead, the server must tell the browser what type of information is in the response.

It does this by specifying the MIME type or media type in the response's Content-Type header. For example, below shows what you would see in the network panel of the DevTools if you inspected the response for www.cs.olemiss.edu.



```
www.cs.olemiss.edu
system.base.css?pvkh3u
field.css?pvkh3u
ctools.css?pvkh3u
foundation.min.css?pvkh3u
jquery.min.js?v=1.10.2
jquery-extend-3.4.0.js?v=1.10..
jquery.once.js?v=1.2
drupal.js?pvkh3u
foundation.min.js?pvkh3u
jquery.flexslider.js?pvkh3u
modernizr.js?pvkh3u
search.js?pvkh3u
system.messages.css?pvkh3u
system.theme.css?pvkh3u
node.css?pvkh3u
search.css?pvkh3u
user.css?pvkh3u
views.css?pvkh3u
```

```
Request Method: GET
Status Code: ● 200 OK
Remote Address: 130.74.96.16:443
Referrer Policy: no-referrer-when-downgrade
▾ Response Headers
Cache-Control: no-cache, must-revalidate
Connection: close
Content-Language: en
Content-Type: text/html; charset=utf-8
Date: Fri, 10 Jan 2020 19:49:36 GMT
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Server: Apache/2.4.6 (CentOS)
Transfer-Encoding: chunked
X-Content-Type-Options: nosniff
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Generator: Drupal 7 (http://drupal.org)
X-Powered-By: PHP/5.4.16
```

The Content-Type header is set to text/html – the MIME type for HTML. You can set this header in the **webserver** part of the Chattrbox project, too, by using the following code:

```
var server = http.createServer(function (req, res) {
  console.log('Responding to a request.');
  var filePath = extract(req.url);
  fs.readFile(filePath, function (err, data) {
    if (err) {
      handleError(err, res);
      return;
    } else {
      res.setHeader('Content-Type', 'text/html');
      res.end(data);
    }
  });
});
```

This part asks you to again work on your **webserver** part of the Chattrbox project, dynamically provide a MIME type for your responses based on the file type. To help you with this task, install the mime module using npm. Information and documentation about the mime module is available at github.com/broofa/node-mime.

Add different files to your app folder, including plain text, PDFs, audio files, and movies. Make sure that your browser is displaying each type correctly.

Part C (Bonus Part)

Also in your **webserver** part of the Chattrbox project, when you go to a path for a file that does not exist on the browser, you currently may get a blank browse page and a 404 status code. For this part, create a special error page to display to the user instead of just returning the error as a status code.