**King Fahd University of Petroleum and Minerals**
**ICS 571 – Client Server Programming – Fall 2016**

**Assignment #4 (carries double weight)**

---

**Objective:** Learn ***http*** module (from Node.js) and use *createServer*() and *request*() methods.

[***Note*****:** *You must use the core module "http.request"* (*part of "http"*); *use streams where possible and do not follow redirect*].

Develop a proxy-like Node.js web server (***mySeerver.js***) to process the route **/*fetch*/** followed by a URL (without "*http://*"). The reply will be ***one of two formats*** depending on the *content-type* in the reply that comes when you issue the HTTP request as follows:

**[Format 1]** If content-type is undefined or it starts with "*text*" then reply to user with **200, text/plain** and send (as a body of HTTP reply) the following as separate lines:

**Status Code: <status code of reply from request>**
**Received headers**
**<Blank line>**
**Body <if present>**

Otherwise **[Format 2** – *used for fetching images***]**, reply to user with **200, *content-type*: <that of reply from request>** and send (as a body of your HTTP reply) the data (body) from reply. (Image should appear in browser – possibly you need to include *content-length* header.)

You will have to do something like this:

```
var contType =  res_req.headers["content-type"];  // res_req is response from the http.request()
 if ( (contType == undefined) || (contType.toLowerCase().indexOf("text")==0))
       { do Format 1  }
 else   { do Format 2 }
```

**Important Notes:**

1. If the HTTP status code is not 200 then reply with a 200-plain-text response (Format 1) containing status code and received headers; if the response is failing (timeout, invalid domain.. etc.) then pass the corresponding Node.js error message to the user (again as a plain-text reply).

2. **It is a must that you server cannot be easily crashed; catch errors and report them to the user.**

**TEST CASES (various kinds of URL that can be fetched):**

< ***site***>*/fetch/www.kfupm.edu.sa/* -- Returns a text reply (Format 1)

< ***site***>*/fetch/www.kfupm.edu.sa/main_web/images1/logo.png* -- Returns an image (Format 2)

< ***site***>*/fetch/www.dell.com/* -- It works when the path is a "/"

< ***site***>*/fetch/www.dell.com* -- It works without "/path" (treat it as a "/")

< ***site***>*/fetch/www.dell2.com/* -- User sees a node.js error message

As a help, the proper setting for opts for the *http.request*:

```
var opts = { method: 'GET' ,
        host:  <host>,
        port: 80,
        path: <path>,
       followRedirect: false
  };
```

Note that "followRedirect" is set to false; <host> and <path> are to be extracted from the user's URL.

**Submission:** Send your script file as a mail attachment by Sunday Nov. 6, 2016. In your mail state clearly any shortcomings in your solution.