DUNWOODY
COLLEGE OF TECHNOLOGY

Lab 3

Node application utilizing RESTful API service

Due: Friday May 11th, 2018 @ 11:59pm

10 Points

Resources needed for this lab:

- Text Editor
- RESTful API data source that you can connect (API Must have good documentation)
- Github Repository
- Postman desktop app
- Language – NodeJs
- NPM modules
    - Nodemon
    - Hapijs - https://hapijs.com/
    - handlebars.js
    - vision.js

Overview:

For this lab, you will continue to build on your knowledge gained from labs 1 and 2. This lab will require you to select a RESTful API data source and utilize the data to bring about meaningful content to the end-user of your application. You will build an application that consist of three pages. You will have a home page (index.html), and a content page (content.html) that utilizes the data requested via API calls. Your application will need to be responsive, aesthetically pleasing and presents the end-user with meaningful information. You are free to add you own content to this application, but data from the API call must also be present.

Requirements:

Part A

1) Download the project folder titled **"lab3"** from the Learning Management system (Canvas).
2) Version control project and initialize Git by running the command **git init**
3) Open the project folder in your preferred editor. The folder should look similar to the image below:

4) Run the **npm init** command to create package.json and the package-lock.json file
5) Install nodemon third-party module – **npm install nodemon –-save**
6) Install hapi.js third-party module – **npm install hapi –save**

7) Open the index.js file and type in the code listed below
   a. Type in the code listed in the picture below(the code listed below is the code from https://hapijs.com/ feel free to copy code but make sure to make modifications based on what is depicted below:

```
'use strict';

const Hapi=require('hapi');

// Create a server with a host and port
const server=Hapi.server({
    host:'localhost',
    port:8000
});

// Add the index.html route
server.route({
    method:'GET',
    path:'/',
    handler:function (request,h) {

        return'<h1> You have reached the homepage</h1>';
    }
});

// Add the content.html route
server.route({
    method:'GET',
    path:'/content.html',
    handler:function (request,h) {

        return'<h1> You have reached the content page</h1>';
    }
});

// Start the server
async function start() {

    try {
        await server.start();
    }
    catch (err) {
        console.log(err);
        process.exit(1);
    }

    console.log('Server running at:', server.info.uri);
};

start();
```

8) Save the code listed in server.js and run the command **nodemon start**
***Note: Make sure that your package.json file has index.js listed as the value for the "main" key value pair.**

9) Open your browser and navigate to pages listed below:
   a. Localhost:8000/ → You have reached the homepage
   b. Localhost:8000/content.html → You have reached the content page
***Note: to kill server press the control key + c key**

At this point, your program should properly route and provide relevant content based on intended route. Prior to starting step 10, make sure to you select an API that will allow you to request data. The example below will walk you through New York Cities OpenData for open jobs:

https://data.cityofnewyork.us/City-Government/NYC-Jobs/kpav-sd4t
https://dev.socrata.com/foundry/data.cityofnewyork.us/swhp-yxa4

Part B – Coming soon!

Submission of Lab

You will submit this lab via Github. The url to your public repository will be due on the date listed above.