

Lab 8
CSC 2053 - Platform Based Computing
Grading: 30 points
Due Date: Nov 30th, 2018

Description:

In this lab we will utilize our Raspberry Pi as a custom webserver using the NodeJS interpreter, Express module, and reinforce some of the command line methods necessary to navigate through a Ubuntu system.

Part 1 -

```
ssh -l pi youripaddress
```

Once logged in, issue the following command to download nodejs, npm, and vim onto your system. This will take 2-3 minutes to complete.

```
sudo apt-get install nodejs npm vim
```

Part 2 - Create a new directory where you will store your web related files. From within this directory, issue the command,

```
npm init
```

This process will ask you a few questions about your project. Feel free to hit “enter” to every question and leave it blank. The result will be a package.json file that exists in this directory. Then execute,

```
npm install express --save
```

This will use npm (the node package manager) to download the express module.

Part 3 - Create a new file called index.js. Enter the following code into this file.

```
const express = require('express'); //require is like java import
const app = express(); //initialize express
const port = 5000; // we will use port 5000 for our webserver
app.get('/', (req,res) => {
  res.send("hello there");
}
app.listen(port, ()=> console.log(`Listening on port ${port}`)); //use of arrow function
and backticks
```

Then run your webserver using the following command,

```
nodejs index.js
```

Navigate to your webpage using your favorite web browser. In the URL put in your raspberry pi's ip address and port number something like, http://10.138.1.255:5000

Part 4 - You can stop the nodejs server by pressing CTRL-C. Serve up the webpage that you created for lab 7. Edit part 3 by changing index.js to have an additional line,

```
const express = require('express'); //require is like java import
const app = express(); //initialize express
const port = 5000; // we will use port 5000 for our webserver
app.use(express.static('public')); // added the public static route

app.get('/', (req,res) => {
  res.send("hello there");
}
app.listen(port, ()=> console.log(`Listening on port ${port}`)); //use of arrow function
and backticks
```

Then create a new directory called “public”. Upload your HTML and CSS from lab 7 into this directory using scp.

Then run your webserver again using the following command,

```
nodejs index.js
```

Finally, navigate to your webpage using a browser and the html file name
<http://youripaddress:5000/lab6.html>

Deliverables: Upload screenshots of your website showing the URL with IP address and port.