Lab 7: INFR 3120

Amirali Abari
Faculty of Business and IT
Ontario Tech University

1 Introduction

Welcome to the seventh lab! Remember that all lab activities are expected to be done individually or in group of 2. There are two roles for group members: *driver* and *passenger*. The driver is the person who is typing on the computer, and the passenger is participating in the development by sharing the ideas. If you are in group of 2, you need to frequently (every 15–20 minutes) change your roles.

Remember that when you have finished your lab activities, you need to show your final results to your TA. He will provide some feedback to you and record your participation mark.

The lab instructions are set as minimum expectations. You are always encouraged to go beyond the expectation. Challenge yourself, have fun, and learn more.

In this lab, we would like to practice our knowledge of Node.js to make our admin page communicate with our back-end engine.

2 Setting up

Please first go to the folder of your lab activities that we call "Labs 3120." Under this folder, create a sub folder and call it "Lab 7"; this is going to be our project directory for this lab. Copy all the content of Lab 6 to this directory (e.g., index.html, admin page, login page, css, JS, etc). If you have not completed your last week lab, you can start with the sample code provided on Blackboard. In this lab, we will start developing our back-end to make our web page a dynamic page.

3 Tasks Description

Our main goal is to make sure our admin page can send its form data to our back-end. To do so, you need to make sure the action attribute of your admin.html (or input.html in the code provided) is set to `http://localhost:8080/update/`. Also, make sure the Get method is used for sending the data. Now, You need to create a file called "myNode.js" in your project directory. In this file, you will write your node.js code. Your node.js code should do the following task:

- Constantly listening on the port 8080.
- If the forms data are sent by `http://localhost:8080/update/`, the server parses the data and store them in a txt file. Also a message will be shown in the browser indicating the data is saved.

- If someone visits `http://localhost:8080/data/`, the txt file should be read by the server and its content should be sent to the browser.

I have provided the skeleton of your code in below. But you need to fill up gaps.

```
var http = require ("http");



http.createServer(function(req, res){
var p_url = url.parse(req.url, true);
console.log(p_url.pathname)
switch (p_url.       ) {
  case "/update/":









   case "/data/":



    break;
}
}).listen("8080");
```

Hint 0: You can first watch this video, if you wish to know how to make your first "Hello World!" application in Node.js. `https://www.youtube.com/watch?v=4qGYwDxITiE`
Hint 1: You might need to use readFile() for reading txt files.
Hint 2: Familiarize yourself with this tutorial: `https://www.w3schools.com/nodejs/nodejs_url.asp`
Hint 3: You might need to remember what JSON.stringify() was doing.
Good luck and have fun!