**CSC435 – Ron Swanson - Notes for Assignment 1**

**Requirements from assignment:**

A two page paper providing technical documentation on the following prompts:

Your experience creating a simple node.js script to serve a HTML file on a local server

Your experience creating and deploying the application using Heroku

The importance of Platform as a Service in full stack development

**SETUP**

Created local structure to save files to learn what process I want to follow before setting up and using git projects, (and waiting for class as there might be more specific instruction there). Have to start somewhere and will refine over time.

*C:\mine\concordia\435notes\*

Ended up creating two git projects, if it works, I will create a gitproject for each week’s homework. It might be overkill, but we’ll see. Otherwise I think there is a place in Blackboard. I thought we would get more specific direction as how the instructor wanted it, so will see how it goes.

<https://github.com/reswanson/my-react-project>

<https://github.com/reswanson/assignment-deliverables>

I think I want to use eclipse, I used Intellij a bit at my last job as that is what most of the developers used. But I used eclipse at my job before that. So trying something different and *forcing* myself to use an IDE. Downloaded and installed git, eclipse and java from these links:

<https://github.com/git-for-windows/git/releases/download/v2.25.1.windows.1/Git-2.25.1-64-bit.exe>

<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2019-12/R/eclipse-inst-win64.exe>

<https://www.oracle.com/java/technologies/javase-jdk13-downloads.html>

Copied my first project into eclipse and committed to my newly created GIT hub project.

( I always have trouble importing new projects into eclipse, it should be easy, but I always fight with it )

I used these instructions for git/eclipse integration, which included adding eclipse egit plugin, creating ssh key pair and added public key to Github profile:

<https://stackoverflow.com/questions/21473308/integrating-eclipse-and-github>

**Your experience creating a simple node.js script to serve a HTML file on a local server**

**Use your favorite HTML editor (or**[**get Notepad++**](https://notepad-plus-plus.org/downloads/v7.7.1/)**) to create a HTML file that includes your name, the class, “Assignment 1”, and the date**

**Install node.js locally from**[**https://nodejs.org/en/download/**](https://nodejs.org/en/download/)**and write the appropriate code to load the .html file, created in the previous step.**

Created .html and .js files, installed node and executed node to display html.

Downloaded/installed node from here:

<https://nodejs.org/dist/v12.16.1/node-v12.16.1-x64.msi>

Took the defaults and let it install dependencies

Used this documentation to create .js file to load the .html file I created.

<https://www.w3schools.com/nodejs/nodejs_filesystem.asp>

Ran the following to start up node using my .js file

C:\mine\concordia\435notes>*node class1.js*

Hit localhost:8080 in browser and verified that it displays my .html file.

**Download and install the official MongoDB driver, open the Command Terminal and execute the following:------- npm install mongodbreference:-** [**https://www.w3schools.com/nodejs/nodejs\_mongodb.asp**](https://www.w3schools.com/nodejs/nodejs_mongodb.asp)

Ran npm install mondodb instead as I couldn’t figure out what mongodbreference was.

*C:\mine\concordia>npm install mongodb*

*npm WARN saveError ENOENT: no such file or directory, open 'C:\mine\concordia\package.json'*

*npm WARN enoent ENOENT: no such file or directory, open 'C:\mine\concordia\package.json'*

*npm WARN concordia No description*

*npm WARN concordia No repository field.*

*npm WARN concordia No README data*

*npm WARN concordia No license field.*

*+ mongodb@3.5.4*

*added 20 packages from 11 contributors and audited 22 packages in 3.291s*

*found 0 vulnerabilities*

**Install Expressjs, a web application framework for Node.js ---- npm install express --save**

*C:\mine\concordia\435notes>npm install express --save*

*npm WARN saveError ENOENT: no such file or directory, open 'C:\mine\concordia\package.json'*

*npm WARN enoent ENOENT: no such file or directory, open 'C:\mine\concordia\package.json'*

*npm WARN concordia No description*

*npm WARN concordia No repository field.*

*npm WARN concordia No README data*

*npm WARN concordia No license field.*

*+ express@4.17.1*

*added 51 packages from 37 contributors and audited 189 packages in 6.841s*

*found 0 vulnerabilities*

**To connect to the MongoDB database we are going to use a module called Mongoose. We will need to install mongoose module just like we did with express. Go to your terminal and enter the following command. ---- npm install mongoose –save**

*C:\mine\concordia\435notes> npm install mongoose --save*

*npm WARN saveError ENOENT: no such file or directory, open 'C:\mine\concordia\package.json'*

*npm WARN enoent ENOENT: no such file or directory, open 'C:\mine\concordia\package.json'*

*npm WARN concordia No description*

*npm WARN concordia No repository field.*

*npm WARN concordia No README data*

*npm WARN concordia No license field.*

*+ mongoose@5.9.4*

*added 13 packages from 9 contributors and audited 385 packages in 4.26s*

*1 package is looking for funding*

*run `npm fund` for details*

*found 0 vulnerabilities*

**Signup for a free Github for your student development pack at https://education.github.com/pack to obtain a free Heroku hobby account.**

Setup github account and got access to Heroku, fairly painless.

Ran these commands as outlineD in assignment.

npm i -g npx

npx create-react-app my\_react\_project

npm install redux

cd my\_react\_project

npm start

Googled a helloworld react app and found [this](https://reactjs.org/docs/hello-world.html) as sample to update the App.js file, and watched as the webpage updated after saving changes to file. Tried different edits to help understand how ReactDOM.render() worked with regards to the App() and displaying html.

**Your experience creating and deploying the application using Heroku**

Was cool to see how easy it was to setup. It was intuitive enough I didn’t need to reference the documentation to get started. After getting everything checked into my [GIT project](https://github.com/reswanson/my-react-project), I turned on automation to auto-deploy and tested a few times from eclipse and from Github to watch and see how the automation worked and to watch the build logs. I have spent the last 15 or so years working on build automation and I haven’t seen this tool before, so it’s always fun to see how things work and I want to look under the covers at things like where the build is actually running, how that is provisioned, etc…

**The importance of Platform as a Service in full stack development**

The first thing off the top of my head before researching, is a PAAS takes care of all the necessary tooling, plumbing and electrical to allow the developer to focus on the actual application instead of the myriad of other details that go into these activities. In my career, I have focused a great deal on builds and deploys. So I first notice those parts. But there are many other pieces a PAAS provides such as networking, routing, scaling, server provisioning, managing access, the list goes on… In my recent experience, corporations are running headlong into cloud providers and Development teams are HAVING to focus more and more on all the other details such as infrastructure-as-code, and dealing with networking in VPCs in cloud accounts, but using a PAAS manages and provides all the structure to allow developers to focus on their apps.