Full Stack Development for the First Time: A Rough Journal

I started the process by looking into Ruby as the back-end option. I’m fairly new to Ruby, but have done a couple projects in it lately, so it seemed like a good pick, especially since its popularity leads to many tutorials. After an afternoon of reading about Ruby and different database choices in the ‘Ruby Cookbook’, I decided to go with Ruby + MySQL. Unfortunately, I ran into a lot of installation errors and decided to take a break on that part.

So I worked on writing the html/css. I remembered a site I'd been on a few days previously, when learning about using selenium to test websites. The site has a page very similar to what was described in the challenge, so I had a look at its source code, and grabbed some pieces to modify. The page: <http://practicetestpage.kevinswallow.com/simple_form.html> I made a really basic search portal start page with the html, and then used a textbook ('HTML and CSS’ by Castro and Hyslop) I had in on my shelf to make it a little more attractive via css. A really simple design.

Now to learn about javascript. I started with some tutorials at <http://www.w3schools.com/js/js_validation.asp> Then, I began looking specifically for code to take the form submitted name and request the json. I discovered jQuery, a javascript library. Also Ajax. I then searched for javascript and JSON. It turns out JSON is JavaScriptObjectNotation. And javascript can easily switch formats from JSON data into javascript objects. <http://www.w3schools.com/js/js_json.asp>

Accessing the data was my next concern, so I found this <http://www.mkyong.com/javascript/how-to-access-json-object-in-javascript/> and this <http://stackoverflow.com/questions/14028259/json-response-parsing-in-javascript-to-get-key-value-pair> I got ahead of myself and went looking for a length function to use on the gpa calculation method. <http://stackoverflow.com/questions/5861536/json-getting-the-size-of-an-array>

At this point, I haven't debugged my JSON object processing yet, but I'm looking now for how to put those values into a webpage. I made a response webpage. It didn't load with form submit. I'm looking into form methods, especially get.

I realized I don't have enough of a wide-angle understanding of javascript, so I went looking for an overview, found eloquentjavascript.net. The 17th and 18th chapters were most useful. I’ve configured something wrong in the html, submit doesn’t bring up the response page, so I made a link to the response page, and that worked. Still not sure why the get url appending of ?first=john&last=smith is not occurring. (Later figured out that it was probably because I was intercepting it with javascript.)

Next, I used console.log to see if I was correctly getting the submitted form data, that worked correctly, so next to figure out how to use it properly. Still not going to response page. Tried the following. <http://stackoverflow.com/questions/16262797/html-form-action-and-onsubmit-issues> I’m looking for tutorials to put it all together.

Starting to think using JQuery might be a good idea, I saw it before but didn’t move in on it. <http://stackoverflow.com/questions/374644/how-do-i-capture-response-of-form-submit>

I went back to trying to get Ruby and MySQL set up. I’m still having issues installing gems, found this. <https://github.com/rubygems/rubygems/issues/977> - says I need to downgrade RubyGems.

The downgrade worked, but then I couldn't install the needed gem because the old RubyGems doesn't have correct paths, according to this: <https://gist.github.com/luislavena/f064211759ee0f806c88>

So, I got the correct certificate and the seg fault when installing error returned. Back to the front end.

I installed firebug, it may help to debug. (It did!) After more browsing of search results, it occurs to me, if I cannot find a tutorial to do what I want with javascript - perhaps it isn't supposed to be doing that thing. There are a lot of results with jQuery and AJAX. I think I'll look into these further. I had been putting it off, I guess the problem description made me think ‘pure’ javascript. It turns out there is no need to download and install jQuery, one can use a CDN <http://www.w3schools.com/jquery/jquery_get_started.asp>

I took a break and switched to working on the database. I ignored Ruby, installed MySQL, found a syntax reference, <https://gist.github.com/hofmannsven/9164408> and a textbook ( Database System Concepts, SIlberschatz) and made a source file to set up the database, once I planned out my tables with their keys, all based on the provided json file. I ran into a foreign key restraint; needed to change the order in which I created my tables. Then had to figure out how to run the file, (<http://www.avajava.com/tutorials/lessons/how-do-i-execute-an-sql-script-in-mysql.html>) I tested each of its major command types individually as I made it and then figured out the commands to get email, gpa, and class list with grades. That part was fun. (Alas, not what was ultimately needed – I didn’t figure out how to return JSON in the format given – it would have been so easy to manipulate!)

Full name: SELECT first, last FROM students WHERE first = ‘John’;

GPA: SELECT AVG(grade) FROM studentClasses WHERE email = ‘johnsmith@mailinator.com’;

Classes and grades: SELECT class\_name, grade FROM studentClasses JOIN classes ON studentCasses.id = classes.id WHERE email = ‘johnsmith@mailinator.com‘;

Now, it’s time to either get back to the front or look into the database -> json transactions. Not with Ruby at this point. So, time to google JVM languages and MySQL and web dev, JSON. I’m leaning toward Node.js for the back-end communication. It looks like a common choice, and I’d get to stick with javascript. It’s becoming more comfortable. <http://stackoverflow.com/questions/20532820/node-js-mysql-json-result-callback-issue-no-response-to-client>

Now I look for download tutorials for getting Node.js for windows. <http://blog.teamtreehouse.com/install-node-js-npm-windows> And then the MySQL driver for Node.js <http://stackoverflow.com/questions/8888078/what-mysql-driver-for-node-js-works-on-windows>

I installed everything successfully and started to try it out. Basic tutorials looked promising: <http://www.sitepoint.com/using-node-mysql-javascript-client/> <http://blog.josedacruz.com/2013/07/19/learning-node-js-using-mysql/#.VueCUEcfiFU>

But, I hit an error, ECONNREFUSED. Oddly my password wasn’t being accepted, even the SQL command line wouldn’t accept it. I opened up the installer, did the check root password, and re-checked start on startup and it worked. ( <http://stackoverflow.com/questions/2214843/mysql-password-not-working> ) The little node program established a connection.

The idea of a stored procedure might have merit, given that we know what we’ll be returning from the DB. But for now, looking into creating web server functionality with Node.js

I used a lot of code from tutorials, and got a little server set up. I think. I have to figure out how to actually query it. Also, what I have returns the response from a pre-set up query, and doesn’t use the actual request. This could still work if I returned essentially the json file that I began with, but it’s silly since no database returns its full contents in this manner. So, something else to work on.

I’m going to return to the front for a bit, I think I need to capture the form submit with the javascript to send the request to my server and handle the json response. This looks useful. <http://stackoverflow.com/questions/5384712/capture-a-form-submit-in-javascript>

I’m not sure whether to use a response page, or modify the display of the original page.

This (<http://www.javascript-coder.com/javascript-form/javascript-form-submit.phtml>) looks interesting. Submit as link to the response page maybe? No, on closer reading it’s just calling the javascript with a link instead of a submit button. Looks less professional. No.

I got the javascript to get the form data and put it in a json. Now to send it to the javascript server.

This looks helpful. <http://stackoverflow.com/questions/13342466/send-post-request-from-client-to-node-js>

I used a jQuery.ajax call in my javascript file, but now it is confused by the syntax. Then I looked closer and realized I didn’t want to take that path anyway – it wasn’t what I’m trying to do. Looking for communication between javascript front end to send request to Node backend. <https://www.sitepoint.com/community/t/how-to-send-requests-to-node-js-back-server/215148/6> Several useful looking links here. I tweaked the Node server a bit, but it’s not delivering the local webpage. <http://blog.modulus.io/build-your-first-http-server-in-nodejs> I used some things from here, but ultimately changed them later. Express won over HTTPDispatcher.

At this point, the search page data is being captured by the javascript and put into a json format. It needs to send that to the server so it can query the database and get the returned JSON string, and display the information. Starting here: <http://www.oranlooney.com/json-round-trip/>

My Node server doesn’t serve pages – I need to add that functionality, maybe with express. <http://ericsowell.com/blog/2014/6/17/enough-node-for-building-a-simple-website> This got me the ability to serve my pages with my Node server, but only with static pages. I think I need to make them dynamic. Either way, I need to make the request with the javascript and get the JSON file.

Maybe this will help. <http://stackoverflow.com/questions/19028982/connecting-to-ms-sql-db-with-node-js-functions> and <http://stackoverflow.com/questions/24207507/get-data-from-server-with-node-js>

I ended up finding that jQuery.ajax calls are a good way to send the get req that I could then strip params from. Getting the parameters was a bit tricky. I found req.params, but they turned out to be deprecated. It turned out that url.parse was the answer. <http://stackoverflow.com/questions/6912584/how-to-get-get-query-string-variables-in-express-js-on-node-js>

The next challenge was using the parameters to connect to the database. This was very, very helpful. <http://www.sitepoint.com/using-node-mysql-javascript-client/>

Then I wasn’t returning the json, but it turned out I needed a res.end call to send it. The results aren’t in the exact same format as the sample json, but I’ll be leaving that for the moment. Now I need to figure out how to access the response json. <http://stackoverflow.com/questions/28820738/how-do-i-receive-and-use-a-json-object-on-the-client-side-from-the-server-node> <http://stackoverflow.com/questions/9911481/accessing-ajax-response-data> It turns out it’s what the ‘success: ‘ portion of the ajax call returns. But now my malformed json (losing the structure that made it orderly is a mess) is rearing its head.

And I am out of time. I will have to finish later, past the deadline. As you can tell, I did a lot of floundering, trying to find the path, at first. I wasn’t sure who passed what message to whom, or what tools are common to do it. I wasted a bunch of time on dead ends. I ended up diagramming it, which helped to not get lost among the options. I didn’t complete the project, but I did manage to set up a small Node.js server that loaded the SearchHome page, let a user submit first last or both names, capture the ‘get’ with javascript, put the values into an ajax call to the server, use them to query the database, and return json to the javascript. The remainder is to get a better formed json response, probably through a better, or more than one database query, and either inject the information into the search page or the response page I made with it.