SC561 NoSQL Databases
Programming Assignment #2

Using the same CouchDB database restaurants on your VM from Lab 6 Part 1, implement a restaurant NodeJS web front-end accordingly. For example, https://csc570e.uis.edu:9443/

You will need to implement the restaurants routes in server.js as well as their respective views (index.ejs and edit.ejs in the views folder). For this assignment, you will need to implement the ability to add/edit/delete every field for restaurants (name, food_type, phonenumber, website). Name and website are required fields and validation should be implemented. Although food_type is an array, you can consider it a single string field for this assignment. You may use the uniqid() method in the documentation or snake case to generate an _id when adding a new restaurant.

The NodeJS package that we will use for this assignment is node-couchdb. Refer to its documentation by going to: https://www.npmjs.com/package/node-couchdb

Video Tutorials

If you need a review, the video lectures from Programming Assignment 1 will be helpful for this assignment as well:

Learning Node.js
https://www.linkedin.com/learning-login/share?forceAccount=false&redirect=https%3A%2F%2Fwww.linkedin.com%2Flearning%2Flearning-node-js-2%3Ftrk%3Dshare_ent_url&account=43607124 (You may need to click the Sign In button in the upper right corner, and then log in with your UIS NetID and password.)

Access from Programming Language (Node.js Mongoose ORM)
[68 min 05 sec]:
https://cdnapisec.kaltura.com/index.php/extwidget/preview/partner_id/1371761/uiconf_id/13362791/entry_id/1_b88z1mj2/embed/dynamic

Use this table to determine which container is yours. You will log into the share with .\NetID for the username (.\tllos1 for example) and your UIN for the password.

Netid	Windows share	Url of the PHP application
agang2	\\10.64.3.56\agang2	https://csc570e.uis.edu:9444
bbala5	\\10.64.3.56\bbala5	https://csc570e.uis.edu:9445
bguti6	\\10.64.3.56\bguti6	https://csc570e.uis.edu:9446
brodr22	\\10.64.3.56\brodr22	https://csc570e.uis.edu:9447
chick7	\\10.64.3.56\chick7	https://csc570e.uis.edu:9448
eunsik2	\\10.64.3.56\eunsik2	https://csc570e.uis.edu:9449
jlund6	\\10.64.3.56\jlund6	https://csc570e.uis.edu:9450
jshei3	\\10.64.3.56\jshei3	https://csc570e.uis.edu:9451
mpavl3	\\10.64.3.56\mpavl3	https://csc570e.uis.edu:9452

rsayy2	\\10.64.3.56\rsayy2	https://csc570e.uis.edu:9453
sarya7	\\10.64.3.56\sarya7	https://csc570e.uis.edu:9454
skoch7	\\10.64.3.56\skoch7	https://csc570e.uis.edu:9455
szhen6	\\10.64.3.56\szhen6	https://csc570e.uis.edu:9456
zwold2	\\10.64.3.56\zwold2	https://csc570e.uis.edu:9457
smeka6	\\10.64.3.56\smeka6	https://csc570e.uis.edu:9458
pshet7	\\ 10.64.3.56\pshet7	https://csc570e.uis.edu:9459

In order to remotely access your CouchDB VM instance you will need to make an edit to your CouchDB initialization file to listen on all IP addresses (0.0.0.0) instead of just localhost. To do so, login to the shell of your CouchDB VM and edit the file /usr/local/etc/couchdb/default.ini with the vi or nano editor and modify the bind_address parameter to:

```
 bind_address = 0.0.0.0
```

Then reboot your Couchdb VM

You will also have to create a MapReduce view on your CouchDB VM to extract the data from your database documents. You may use something similar to the view provided on your Github lab6a.sh file (restaurants/_design/docs). This view is referenced in the server.js file under:

```
const viewUrl =  '_design/docs/_view/all';
```

You will need to update the server.js file with the IP address of your CouchDB VM. Ex:

```
const couchExternal = new NodeCouchDb({

host: '10.92.130.128',
```

This should be the first step you do. After doing so you should be able to see a listing of the restaurants stored in your database.

- For extra credit you may choose to implement an additional field (_attachments) for uploading a restaurant review on the edit form.

You will not need to submit anything to GitHub. I will grade your assignment by checking the URLs.