CART 451 FALL 2023

DELIVERABLE FOR TECHNICAL EXERCISE I

DUE: OCTOBER 17th :: 5:45PM ::

WHAT: A PDF Uploaded to your Git Repository + the node project (no node modules needed) and provide a link to the relevant Moodle assignment link.

The purpose of this report is to use mongo db + node + some data … with the addition of express (your http server) + some server/client requests.

PART A: ACCESS AND IMPORT THE DATA:

1. Go to https://www.kaggle.com and download a dataset either in JSON or CSV format -> you CAN choose the dataset that you will be doing your presentation on … but it is NOT necessary.
2. Import the dataset into a new database in your atlas cloud: you now have two choices:

    a. using the terminal and installing the mongoshell/ atlas commandline interface (Atlas CLI)+ mongoimport tool (part of the MongoDB Database tools)
    https://www.mongodb.com/docs/atlas/cli/stable/install-atlas-cli/
    https://www.mongodb.com/docs/database-tools/mongoimport/
    Note: mac users should use homebrew and Windows users should use choclatey ** these are very useful package managers.

    b. If you do not feel comfortable working in the terminal/cmd prompt, then you can use Mongodb compass (their UI interactive tool)
    https://www.mongodb.com/products/compass

    c. Provide a paragraph in the written report, discussing the dataset, why you chose it, any issues etc…

PART B: WRITE SOME MONGO DB QUERIES:

1. Make and run at least **5** custom queries in mongo and run the queries using a node.js project + mongodb node.js driver **with the purpose of gathering insight to your data** set. These queries cannot **be insert/update/delete queries** … rather they should be statements where you are trying to analyse find out info … using find()/ aggregate()….

2. Provide a WRITTEN summary in the report on what your queries were and one paragraph per query on your findings … and what you found out about your dataset after having run the queries…

PART C: EXPRESS + MONGO:

1. Ok … So create a local webserver using Express - have a server running and implement some static pages (html, css, js..) - create an interface whereby the user can search for a given criteria.
2. Implement a get request – whereby the user's search criteria is sent to the webserver, you perform at least one of your mongo queries (from B) and return the results to the user.
3. Create a simple data visualization (not just a graph / chart ☺) client side for the data returned. Provide a series of screen shots depicting a client making a search request and the resulting output for at least 3 different search criteria. Put the screen shots+ explanations (what is the data visualization- why/how etc.…) + a description of the query used in the written report.