IMY 220
Practical 8:
MongoDB

Due: Friday 21 October at 9:30.

The submission instructions are available on ClickUP. Any deviation from these instructions will cause a 10% deduction from your mark.

Instructions

- Use MongoDB and display events that fall in a certain criteria.
- Download *events.json* from ClickUP. This file contains the data you will need to upload to a MongoDB database and display on your webpage.
- Your package.json file needs to be correct according to the lecture notes and must list all the packages your code refers to. In other words, whoever marks your practical should be able to run **npm install** and **npm start**. The default **npm start** code inside the package.json file (if set up correctly) will run the code without error, so don't add unnecessary code to this.

Before you start: Note that due to a recent update with Mongo, when connecting to the MongoShell the option on their website listed as "Connect with The Mongo Shell" can causes issues sometimes so rather use the option "Connect Using MongoDB Compass" and use the same method (i.e., open it in the shell, and just paste the code that's given). This is purely for testing purposes, but it is something that you should take note of.

Note: for this practical you will need to include the connection details, including the password, for your MongoDB database. These details are the ones you created the database with and **not** your login details for *cloud.mongodb.com*. Regardless, you should probably delete this database when you are finished with this module.

Section 1 – Create MongoDB cluster, database, and collections

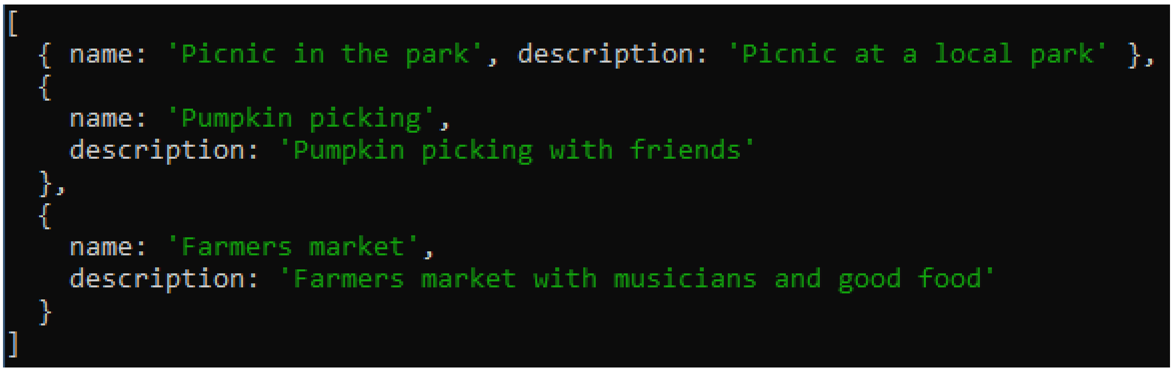
Create a MongoDB cluster, named whatever you want, and a database, named DBExample, as per the notes. Create a collection called events and populate it with the contents of *events.json*. Create a text file (*details.txt*) with the code for connecting to your MongoDB database within the MongoDB shell. The result should be that the marker is able to use the connection string inside *details.txt* to connect to your database and run **show dbs** to see the list of databases, i.e., *DBExample*, *admin*, and *local*.

Section 2 – Load events

Note that for this practical you **do not** need to create a node server. Inside your *server.js* file, connect to your MongoDB database and query the events collection to fetch only the entries that fall under the specific criteria:

- An event should take place in **Brooklyn**.
- An event should fall **between** the dates of the 8th of October and the 26th of October.

After fetching the correct data, display only the name and description of the event. If this has been done correctly, the following will be returned:



```
[
  { name: 'Picnic in the park', description: 'Picnic at a local park' },
  {
    name: 'Pumpkin picking',
    description: 'Pumpkin picking with friends'
  },
  {
    name: 'Farmers market',
    description: 'Farmers market with musicians and good food'
  }
]
```

HINT - Use the following webpages as a guide for using MongoDB with NodeJS and socket.io:

- https://docs.mongodb.com/drivers/node/current/quick-start/
- https://docs.mongodb.com/drivers/node/current/usage-examples/find/
- https://docs.mongodb.com/drivers/node/current/fundamentals/crud/read-operations/cursor/#std-label-cursor-methods

Additional Information

- Refer to the slides and online resources for help

Submit only the following file(s) according to the submission instructions.

All files (including details.txt) ***except*** *the node_modules folder*

(Including node_modules will result in a 10% penalty).