Fandango Project

## MongoDB

**Read all the requirements before you start!**

**Create a folder called *yourLastName* to contain your files. When completed, the folder will contain your .js file.**

**Zip the folder and submit the *yourLastName.zip* file to the myCourses Assignment folder. (Note: you must submit a .zip file only. A .7z, .rar or .tar file will not be accepted).**

# Part 1:

For this part, you will need to examine a document in the Mongo Project folder under   
Content 🡪 08 – MongoDB: Aggregation Framework, Compass, MongoDB Project 🡪 MongoDB Project on myCourses:

* Movie Times in Rochester NY 20240829.pdf - this was printed from the Fandango web site, showing where and when movies will be playing in the Rochester area as of August 29, 2024

This document should look familiar, since it was (or should have been) used in your XML Project! Using this document, determine what elements and attributes you will need to put into MongoDB documents to create a database that captures the information on movie times for your selected theaters.

# Part 2:

Develop a javascript file that will build the documents in the MongoDB database that you will be creating. Be sure that the two theatres share at least one of the same movies.

Guidelines:

* capture the same 3 movies each from two theatres .
* The database must be named Fandango
* The collection must be named *yourLastName* (e.g. for me it would be Patric). If you have multiple collections, append a number to your name (e.g. Patric1, Patric2, etc.)
* The script should be named *yourLastName.js*.

Requirements:

* I must be able to run your script to load your data
* It should produce a nicely formatted listing for each of your theatres, listing theatre and movie information that a user would like to see to make a decision about what movie to see

# Part 3:

Answer the following questions on this document, and submit it with your script.

1. Describe in a few sentences how you approached the modeling.

I modeled the data by creating two documents, one for each theater, with each theater document containing an array of movies. Each movie includes attributes like title, genre, rating, duration, showtimes, and features to ensure that it provides all the relevant information a user would need to decide which movie to watch.

1. What design guideline did you use?

I used an embedded document model, where each theater contains an array of movies as embedded documents. This allows the data to be closely related and easy to retrieve in one query.

1. Why did you pick this particular guideline?

I chose this design because theaters and their movies are closely related and naturally fit into a single document structure. Embedding the movie details within each theater provides a simple, efficient way to store and retrieve the data with minimal queries.

1. What changes (if any) would you need to make if you had to create every theater and movie on the document?  
   If I had to create every theater and movie, I would still use the same embedded document model but would likely implement indexing on fields like ‘theaterName’ and ‘movieTitle’ to improve query performance. If the dataset grows very large, I might consider normalizing by separating theaters and movies into separate collections, using references to link them.

**Zip the folder containing the file(s) you created above with this document and submit the .zip folder to the myCourses Assignment folder by the due date specified.**