CMPUT 291 Mini-Project 2 Report

Overview

This program uses Python language. The main purpose of this project was to learn the concept of working with data stored in files and NoSQL databases. We used MongoDb inside python in this project. This program interacts with the user and displays the required data.

The whole project is divided into two phases, in phase 1 we are given tsv files and are asked to convert these files into json files and load our database with these files. We have done the above by writing two python programs which tells us the time taken to convert these files.

In phase 2 of the project, we divided it into 5 parts namely.

- searchTitles
- searchGenre
- searchCast
- addMovie
- addCast

We have made a different file for each function, and we call all of them into a main function which executes the whole project, which then ensures that every function works as intended in the Project.

Detailed Guide

The different functions we have used in this project are:

- def main( ): This is the main function of the project. Inside this function we call every other file and functions that we have made. We import all the other files and then we ask user to input which option he/she wants to execute and then call the respective function from the files.
- def searchTitles(db): In this function we prompt the user to enter one or more keywords. Then we print all the movies with matching keywords on the screen using AND semantics. The user then can select any title to see the movie's rating, votes and the characters of the respective movie.
- Def searchGenre(db): In this function we ask the user to enter a genre, after checking if the genre is valid or not we then ask for the minimum number of votes from the user. After getting these details we display the required movies with their titles and ratings on the screen.
- def searchCast(db): In this function we ask the user to enter a valid cast/crew name, after getting the name we display all the cast/crew members with matching name and their movie appearances if any on the screen.
- def addMovie(db): This function adds a movie into the given collection. We ask the user to enter a valid userId, if the userId is wrong we prompt until the user enters a valid userId. After this we ask for the different details about a movie and then add those details into the collection.
- def addCast(db): This function adds a new cast/crew member into the respective collection. We first ask the user to enter a castId which already exists in the collection, if the entry is not valid we prompt until a valid castId is entered, we do the same for the titleId. Then we insert the row into the collection according to the specification mentioned in the project details form eclass.

Testing Strategy

We employed two primary testing strategies. Firstly, we did manual testing of UI and its functionality. Each requirement was tested by simulating the actions described on the rubric. Correctness was checked by using the mongo terminal and ensuring that expected results were obtained.
Secondly, to check if the queries are being executed in proper timings, we created shell scripts. This enabled us to improve the time performance of our code specifically in phase1. Then we tested the queries in phase2 such that they were being executed almost instantly.
From our end we have made sure that the program is completely error free and runs as intended by checking if the code works properly on the lab machines.

Breakdown of work

Every team member contributed equally to this project. We all worked together on this by meeting in person and using the feature "Code with me" on PyCharm to collaborate and see each other's work. We discussed the progress of the project every day and discussed the strategies after each meeting. The detailed description of the work done by each member is given below:

Charvi Dhamija:

- Time spent - Approximately 9 hours
- Progress - Implemented the searchGenre and addMovie functions of the code and helped in some portions of phase1.

Dhruvraj Singh:

- Time spent - Approximately 13 hours
- Progress - Provided most of the groundwork and organization for the project. Implemented half portion of the phase1, searchCast and addCast .

Mobashhir Khan:

- Time spent - Approximately 9 hours
- Progress – Designed the main screen and implementation of the first query (searchTitles) and half portion of phase1.