# CS374 – Intro to Database Management

# Database Project

# Rubric for Project Write-up

## Group Member: Salim Dhahri

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Requirements | Points | Awarded |
| Description of Application | * System requirements (e.g. hardware, DBMS, other software) * A detailed description of your application, including how each aspect of your application will satisfy your client’s needs   + There should be enough detail that others could implement your application * Are there design goals that were not met? What are they, and why won’t you fulfill them? | 15 |  |
| Class Concepts | * Discussion of concepts from class that were applied to your project * Discussion of concepts from class that did not apply to your project | 10 |  |
| Personal Contributions | * Contributions listed accurately and fairly * Team discussions that took place over design and implementation * Discussion of alternative designs and algorithms * Discussion about how/whether the project made concepts discussed in class more con * crete | 15 |  |
| Grammar | * Spelling and grammar * Follow rules from the Penguin handbook on writing * References as appropriate (e.g. if you are modeling your application after an existing application, make note of that) | 10 |  |

* **Description of Application**

Our program aims to allow users to make an account where they can store the movies they've watched, including the date they watched it and their rating. The program also gives users suggestions for movies. The database for this program holds data about different movies, directors, users, and the relationships within these schemas.

This is achieved through using MySQLWorkbench to manage the data. Primary modifications and implementation of the code is done through server.py, which uses the Flask framework of Python, to manage queries using embedded SQL. The index.js file uses the React library of Javascript and communicates with the server. index.css is then used for designing the webpage and tables.

The Movie Database is connected with a database on MySQLWorkbench. The .sql files must be imported to a schema named "Project" locally, in order to run the queries successfully.

* A .env file must be made in the tracker-server of this project, with your MySQLWorkbench password.
* Installation of the libraries in the requirements.txt file must be downloaded, which can be done using the following terminal command:
  + pip install -r requirements.txt
* Flask framework must also be installed using:
  + pip install flask
  + pip install flask-cors
* Lastly, the axios library of React must be installed:
  + npm install axios

When it comes to design goals that were not met, initially, we talked about how we wanted the program to recommend movies based on the user’s watch history, however, after storing all the movies that a specific user watched, it was a bit challenging generating movies that match in terms of genre or director. Consequently, we have a recommendation system based on the user input, so they choose their intended genre or director.

* **Class Concepts**

We extensively applied concepts such as database creation with MySQL Workbench, embedded SQL for querying, and schema conversion using createdb.py from txt/CSV to SQL.

While relational algebra wasn’t explicitly used, its principles were inherent in our approach to database management.

* **Personal Contributions**
* When it comes to contributions to this project:
* Created database schemas and converted txt/CSV files to SQL for table creation.
* Developed embedded SQL for the movie recommendation system in test.py.
* Defined app routes in server.py to correspond with embedded SQL in test.py.
* Worked on index.css for effective presentation of information tables.
* Organized the output on the web interface in index.js.
* Team discussions:
  + While working on the project, we were able to understand our main goals when we started communicating our intentions and the work done. It is important to update team members about any ideas or progress. After ensuring that we were communicating well, our workflow became better and we were able to achieve most of what we wanted in addition to troubleshooting any roadblocks together.
  + It was important to discuss the design improvements with each other, especially after meeting with Pete to figure out a specific solution for an issue that we faced, individually.
* Alternate designs and class concepts:
  + Working on a project that is connected to a local database on Whitgit/GitHub is hard to handle. While working on this, we ran into several issues when it comes to pulling/pushing and syncing the project. This is why I believe having the whole project on AWS would have made it easier to access and work with.
  + This project was definitely a great learning experience especially to solidify the concepts that I learned in class. I got more familiar with creating databases and accessing them using embedded SQL.
  + I worked with different coding languages to make this project happen which was a great opportunity for me to learn new things. For example, I never used Python or React before and my experience with JavaScript is so limited, so I enjoyed enhancing my skills with different languages and understanding the connection between them.
* Overall, this project was an invaluable learning experience that solidified class concepts and exposed me to diverse coding languages, reinforcing their interconnectedness. I really enjoyed working on this project with my team.