**Project Documentation**

Abstract: The code is designed to implement an interactive network for IMDB movie data set

Web-Technologies Used: HTML, CSS, D3.js, Json, xampp.

Compatibility: The created web pages work’s fine in Microsoft Edge, Mozilla fire fox (Quantum-69.0.1), Google chrome (77.0.3865.90)

Description: The code consists of a HTML page described as index.html, a cascading style sheet which provides style for the web page and a JavaScript code that provides functions for the webpage. Firstly, we created a json file from movie data set and in the index.html where we created the html code and links to json file and style sheets we created 2 sub networks in the same html file and executed similarity Matrix.

Q&A:-

1. The first phase of the project began with preprocessing the given dataset (IMDb 5000 Movies) wherein, it was analyzed, and data cleanup was carried out. The networks that the project aims to build movies from the year 2001-2016 in genres horror & comedy. Data cleanup involved dropping the rows (movies) which are not related to that subject and the columns (attributes) that didn’t contribute much to the analyses that the project intended to carry out. Consequently, a dataset containing only relevant information to build our network was obtained. We constructed the network using html, CSS and D3.js. we deleted unwanted columns using Macro’s and links are placed based on the indexing.
2. After making the network we made 2 subnetworks movies based on the years. The first subnetwork contains data from the year 2001 to 2010 and the second subnetwork had the data from the year 2011 to 2016. In the two-sub network’s we have the data of movies based on the main network by making a movie matrix and took 60 samples out of which 30 samples are for sub-network-1 and sub-network-2. Based on the content rating given to them we created the links between the nodes.
3. We used Jaccard similarity to find out the correlation between the nodes selected from each subnetwork. These subnetworks are based on nodes (movies) and their connections (similar content rating).
4. The interactive design to compute and visualize the similarity between two subnetworks was done by allowing the user to select nodes (which essentially symbolize the movies) from the two subnetworks individually and then click the ‘Similarity’ button in order to find the commonality. This results in production of a table that presents the movies from two subnetworks being compared along with the common attribute. Additionally, this also results in computation and display of the similarity score once the common attributes are identified.
5. The program is submitted along with the supporting JSON files from which the subnetworks were built. It is necessary to place these JSON files in the same directory from which the source code runs. It is required to host the source file on a local web server in order to see complete execution of the project. The application could be run at the localhost:8080/HCI/index.html on the user’s browser. The project submitted comes as a bundle comprising of the source Html, CSS, JavaScript files along with some other supporting files that include the two JSON files as well.