

## splflix

- This project is written in c++ followed by the principles of memory management.
- This program should be run on linux OS.
- You need to compile the project with the command “make” in the main folder(where is the “makefile” located).
- You need to run the project by entering to the terminal opened in the “bin” folder the command:./splflix config1.json.
- This project is given to me in the course “System programming language”.
- The program is the manner of “Netflix”, simulate managing users, watching series and movies and printing history of watching.

### instructions:

- There are three recommendation algorithms:
  - Rerun Recommender – This algorithm is intended for users who don't like new stuff. It will recommend content based on this user's watching history. Starting from the first watchable content, in a cyclic order. That is – the first watchable to be recommended is the first watchable watched by the user. Identified with 3 letter-rer
  - Similar Genre – This algorithm will recommend content based on the most popular tag in the user's watch history. If a set of tags has the same popularity, the algorithm will use lexicographic order to pick between them. It will recommend a content which is tagged by the most popular tag in the user's history, which wasn't already watched by this user. If no such content exists, it will try with the second most popular tag, and so on. (3-letter code – gen)
  - Length Recommender – This algorithm is based on the assumption that users prefer to watch content in a similar length. It will recommend the content whose length is closest to the average length of the content in the user's watch history, and which isn't in the user's watch history. (3-letter code – len)
- Actions:
  - Create User – Creates a new user in the current session. **Syntax:** createuser <user name> <algorithm code>.For example:”createuser Gil len”

- Change Active User– Changes the current active user. **Syntax:** changeuser <user name> If the user doesn't exist, this action should result in an error. Example: " changeuser Yossi" – Changes the current active user to be Yossi's.
- Delete User – Removes a given user. **Syntax:** deleteuser <user name>,Example: "deleteuser Yossi" – Deletes Yossi's user.
- Duplicate User – Creates a copy of the given user, with its watch history and recommendation algorithm, but with a new name.  
**Syntax:** dupuser<original user name> < new user name>.  
Example: "dupuser Yossi David" – Creates a copy of Yossi's user, with the name David.
- Print Content List – Prints all the watchable content available.  
**Syntax:** content.
- Print Watch History – Prints the watch history of the current active user. **Syntax:** watchhist .Example: "watchhist" will print: Watch history for Yossi if Yossi is the active user.
- Watch – Watches a content. **Syntax:** watch <content id>. Where is the id of the content in the content list that will be displayed.
- Exit – Return to the initial screen **Syntax:** exit