



## DATABASE PROJECT

### MOVIE RECOMMENDATION SYSTEM

Eslam Ahmed Mohamed	202000039
Esraa Negm Sayed	202000799
Mariam Amr Barakat	202000210
Mohamed Abdel-Maged Essawey	202000440
Rawan Mohammed Elframawy	202001762

Course: Database

Professor: Dr. Mona Arafa

## A. Project Idea

Our project is a website for movie recommendation. It will be a platform where a user can search for movies, filter movies according to his desired features, add movies to the favorites, saved or to watch later. Furthermore, the user has his account and recommendations will be made according. We will use content filtration for our recommendation system, this means that the user will be suggested movies like what he had searched. The website also has the functionality to extract a movies detail and makes the viewer see it in an organized manner.

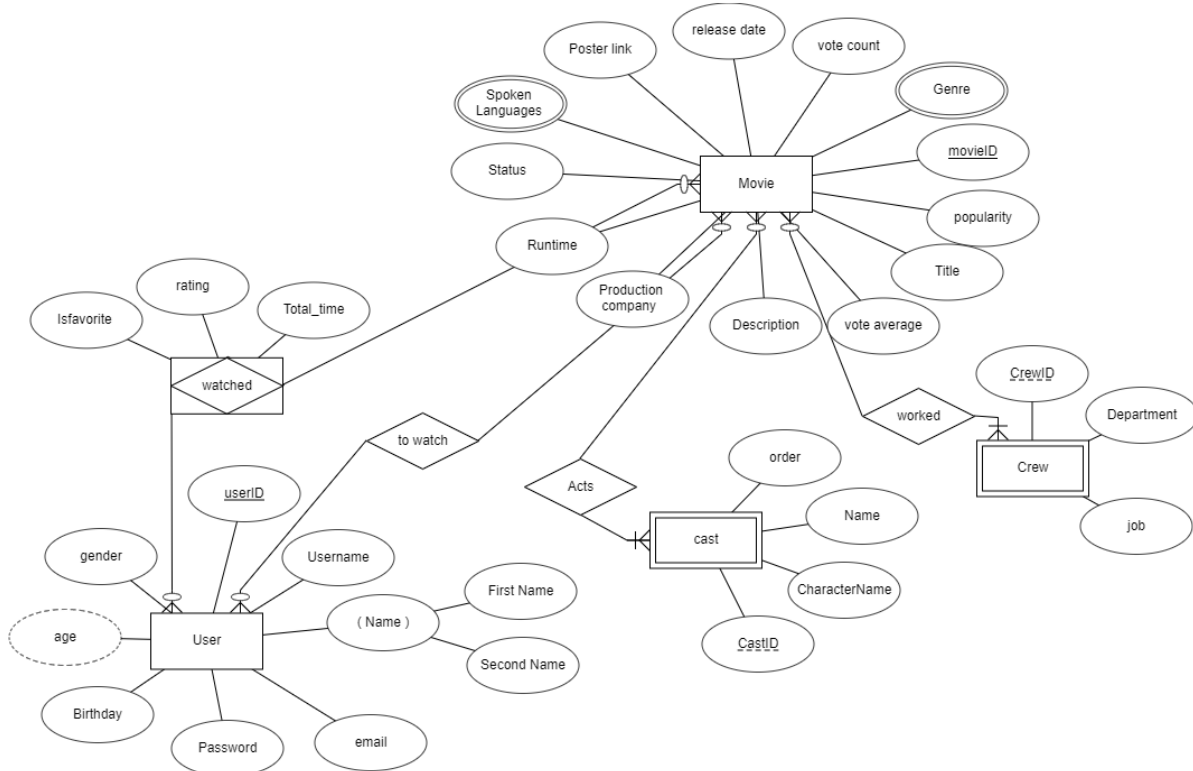
## B. Project Objective

Our main objective is to create a system that:

- Filter and predict only those movies that a corresponding user is most likely to want to watch.
- Keep track of films you've seen in the past.
- Indicate your favorite movies on your account.
- Rate the films on the platform.

Moreover, we are aiming to have a user-friendly environment with beautiful designs.

## C. Business rules and ERD



### ***D. Entity Description***

User	User who uses the website and enjoy its features.
Movie	The movie that user searches or wants to watch it.
Cast	Lead and supporting actors.
Crew	Director, Editor, Composer, Writer etc.

### ***E. Attribute Description***

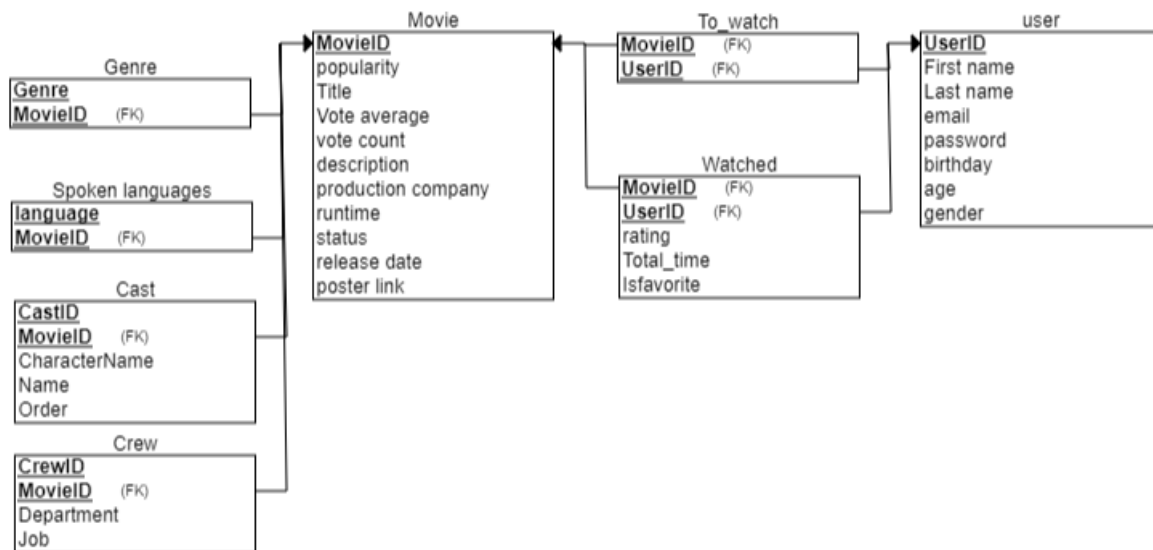
<b><u>User</u></b>	
UserID	Unique identifier for each user. This field is required by the DBMS <b>Primary key</b> Type: Number
Username	Username of the user. Type: Text
Name	Name of the user. Type: Text
Email	Email of the user. Type: Text
Password	Password of the user. Type: Text
Birthday	Birthday date of user. Type: Date
Age	Age of user. Type: Number
Gender	User's gender. It takes 2 values (Male or Female) Type: Text
<b><u>Watched associative entity</u></b>	
Isfavourite	If the film is favorite for the user or not.
Rating	The rating user give to the movie.
Total_time	The time user watches the move in minutes.
<b><u>Movies</u></b>	
MovieID	Unique identifier for each film. This field is required by the DBMS <b>Primary key</b> Type: Number
Popularity	A numeric quantity specifying the movie popularity. Type: Number
Title	Title of the movie. Type: Text
Vote average	Average ratings the movie received. Type: Number
Description	A brief about the movie that have explanation of the film's storyline. Type: Text
Production company	The production house of the movie. Type: Text
Runtime	The running time of the movie in minutes. Type: Number
Status	It takes 2 values ("Released" or "Rumored"). Type: Text

Spoken Languages	The language in which the movie was made. It is multivalued as a movie may have more than one spoken language. Type: Text
Poster link	Link of the poster of the movie. Type: Text
Release date	The date on which movie was released. Type: Date
Vote count	The count of votes received. Type: Number
Genre	The genre of the movie, Action, Comedy, Thriller etc. It is a multivalued attribute as a movie may identify with more than one genre. Type: Text
<b><u>Cast</u></b>	
CastID	Unique Identifier for each person in the cast. <u>Part of Primary key</u> (weak Entity) Type: Number
Name	Name of each person in the cast. Type: Text
Character_name	The name of the character the actor act in the movie. Type: Text
Order	Order of the actor in the movie, it indicates the importance of the actor role. Type: Number
<b><u>Crew</u></b>	
CrewID	Unique Identifier for each person in the crew. <u>Part of Primary key</u> (weak Entity) Type: Text
Department	Department of each person in the crew such as Art, Sound, Visual Effects, Editing .... Type: Text
Job	Job of each person in the crew such as Assistant Director, Construction Coordinator, Dialogue Editor.... Type: Text

### ***F. Relationship Description***

Watched	Relation between users and movies they watched. [Many to Many]
To_watch	Relation between users and movies they want to watch. [Many to Many]
Acts	Relation between movies and actors act in the movies. [Many to Many]
Worked	Relation between movies and crew work in the movie. [Many to Many]

## G. Relational database schema



## H. Samples of SQL instructions

Creation of tables sample

```
USE netflex;
create table IF NOT EXISTS users(
    userid INT NOT NULL AUTO_INCREMENT,
    username VARCHAR (40),
    first_name VARCHAR(40) NOT NULL,
    last_name VARCHAR(40) NOT NULL,
    email VARCHAR(100) NOT NULL,
    password VARCHAR(100) NOT NULL,
    birthday DATE,
    gender ENUM('m','f'),
    PRIMARY KEY ( userid ),
    UNIQUE (username),
    UNIQUE (email)
);

INSERT INTO users(username, first_name, last_name, email, password, birthday)
VALUES('MyriamB', 'Mariam', 'Barakat', 'ma.barakat@nu.edu.eg', '123', '2002-03-06');
```

## Login in execution in SQL

```
@app.route("/login", methods=["GET", "POST"])
def login():
    message=''
    if request.method == 'POST':
        username = request.form['Username']
        password = request.form['Password']
        cursor.execute('SELECT * FROM users WHERE username=%s AND password=%s',
(username, password))
        record = cursor.fetchone()
        if record != None:
            #create a session
            #session['loggedin'] = True
            #session['username'] = record['username']
            message = "Login a Success!"
            return redirect(url_for('home'))
        else:
            message = "Incorrect username or password. Try again!"
            print(message)

    return render_template('login.html', msg= message)
```

### *I. List of tools*

#### **Software tools:**

1. **MySQL server:** For storing all information of database.
2. **GitHub Command Line:** using GitHub from your computer's command line.
3. **Visual Studio Code:** For coding user interface of the system.

## J. SQL codes

```
> CREATE TABLE `casting` (  
  `movieid` int NOT NULL,  
  `castid` int NOT NULL,  
  `character_name` varchar(287) NOT NULL,  
  `actor_name` varchar(28) NOT NULL,  
  `priority` int NOT NULL,  
  PRIMARY KEY (`movieid`,`castid`),  
  CONSTRAINT `casting_ibfk_1` FOREIGN KEY (`movieid`) REFERENCES `movies` (`movieid`) ON DELETE CASCADE
```

```
55 • CREATE TABLE `crew` (  
56   `movieid` int NOT NULL,  
57   `crewid` varchar(24) NOT NULL,  
58   `name` varchar(33) NOT NULL,  
59   `department` varchar(17) NOT NULL,  
60   `job` varchar(38) NOT NULL,  
61   PRIMARY KEY (`movieid`,`crewid`),  
62   CONSTRAINT `crew_ibfk_1` FOREIGN KEY (`movieid`) REFERENCES `movies` (`movieid`) ON DELETE CASCADE
```

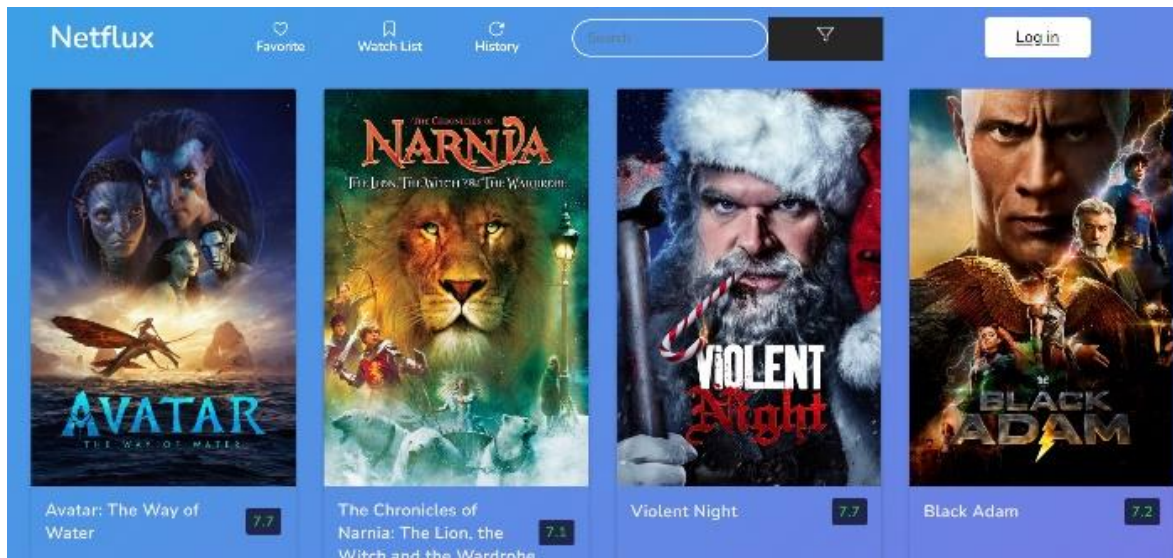
```
171 • CREATE TABLE `persons` (  
172   `userid` varchar(40) NOT NULL,  
173   `first_name` varchar(40) NOT NULL,  
174   `last_name` varchar(40) NOT NULL,  
175   `email` varchar(100) NOT NULL,  
176   `password` varchar(100) NOT NULL,  
177   `birthday` date DEFAULT NULL,  
178   `person_type` enum('user','admin') DEFAULT 'user',  
179   `gender` enum('M','F') DEFAULT NULL,  
180   `phone_number` varchar(13) NOT NULL,  
181   `country` varchar(45) DEFAULT NULL,
```

```
8  
9 connection = mysql.connector.connect(user="SWE", password="123456789000", host="localhost", database="netflux")  
10  
11 cursor = connection.cursor()  
12 cursor.execute("SELECT title FROM movies")
```

You, 1 second ago • Uncommitted changes

```
# Hash a password for the first time, with a randomly-generated salt
hashed_password = bcrypt.hashpw(password.encode(), bcrypt.gensalt(rounds=12))
cursor.execute(
    "INSERT INTO persons (userid, first_name, last_name, email, password, gender, phone_number, birthday) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)"
    (
        username,
        firstname,
        lastname,
        email,
        hashed_password,
        gender,
        phone_number,
        birthday,
    ),
    You, 5 days ago + Revert "Revert "Merge branch 'main' of https://gi...
)
connection.commit()
msg = "You have successfully registered !"
```

## GUI screenshots



### Registration

Full Name

Username

Email

Phone Number

Password

Confirm Password

Gender

☐ Male
☐ Female

Register

I have an account

Username

Password

Log in

Create an account



Account Setting

11

User Name

jana\_mohammed

First Name

Jana

Last Name

Mohammed

Email Address

jana.mohammed@gmail.com

Phone Number

+2010048808

Birth Date

mm/dd/yyyy

Gender

☒ Male ☐ Female ☐ Prefer not to say

Address

Egypt

Monofia

Bio

About you

Old Password

New Password

Update Password

Update Profile

