개요

오픈소스 프로그래밍 프로젝트에서 진행된 개발 과정에 대한 내용과, 프로젝트 실행 방법에 대한 설명을 작성했습니다.

프로젝트 설명

해당 프로젝트의 주제는 "날씨에 따른 영화 추천서비스"로 기존 넷플릭스와, 웨이브 등의 OTT서비스처럼 사용자의 시청기록에 대해서만 추천을 하는것과는 달리, 현재 날씨와 사용자의 장르 선호도에 따른 추천을 해주는 웹서비스제작이다.

시스템구조

이 프로젝트에서는 Flask인 웹 프레임워크를 베이스로 프론트에서 Bootstrap을 사용해 개발을 진행했다.

시스템의 큰 구조는 MVT(Model View Template)패턴으로 즉 코드상에서는, `model.py, views, templates`에 해당한다.

다음으로 각각 해당하는 코드들을 살펴보겠다.

Model(model.py)

```
# User 모델 정의
class User(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(10), nullable=False)
    password = db.Column(db.String(200), nullable=False)
    age = db.Column(db.Integer, nullable=False)
    set_weight = db.Column(db.Boolean, default=False)

# Movie 모델 정의
class Movie(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(255), nullable=False)
    release_date = db.Column(db.String(20), nullable=True)
    popularity = db.Column(db.Float, nullable=True)
    vote_count = db.Column(db.Integer, nullable=True)
    vote_average = db.Column(db.Float, nullable=True)
    overview = db.Column(db.Text, nullable=True)
    poster_path = db.Column(db.String(255), nullable=True)
    genres = db.relationship('Genre', secondary=movie_genre_association,
back_populates='movies')
    adult = db.Column(db.Boolean, nullable=True)

# Genre 모델 정의
class Genre(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(50), nullable=False)
    movies = db.relationship('Movie', secondary=movie_genre_association,
```

```
    back_populates='genres')

    class Genre_weight(db.Model):
        id = db.Column(db.Integer, primary_key=True)
        user_id = db.Column(db.Integer, db.ForeignKey('user.id'))
        genre_id = db.Column(db.Integer, db.ForeignKey('genre.id'))
        genre_weight = db.Column(db.Integer, nullable=False)


    class Weather(db.Model):
        id = db.Column(db.Integer, primary_key=True)
        user_id = db.Column(db.Integer, db.ForeignKey('user.id'))

        sunny1 = db.Column(db.String(10))
        sunny2 = db.Column(db.String(10))
        sunny3 = db.Column(db.String(10))
        sunny4 = db.Column(db.String(10))

        cloudy1 = db.Column(db.String(10))
        cloudy2 = db.Column(db.String(10))
        cloudy3 = db.Column(db.String(10))
        cloudy4 = db.Column(db.String(10))

        snowy1 = db.Column(db.String(10))
        snowy2 = db.Column(db.String(10))

        raining1 = db.Column(db.String(10))
        raining2 = db.Column(db.String(10))
```

위는 모델부분의 일부분이다. 코드와같이 user에 대한 정보를 담는 user테이블, API에서 가져온 영화들의 장르에 대한 Genre테이블, 사용자가 영화 장르 테스트를 마치고난뒤 각 사용자마다 장르 가중치를 저장하기위한 Genre_weight테이블, 각 사용자마다 날씨에 따른 관심장르에 대한 Weather테이블을 만들어주었다.

View(views)

view는 auth_views.py, main_views.py, question_views.py총 3개로 나눠서 개발을 해줬다. auth_views에서는 model에서 User테이블을 이용해 회원가입과, 로그인, 로그아웃기능을 만들어주었다. question_views.py에서는 질문을 하는것에 대한 기능을 구현해주었다. main_views.py에서는 TMDB라는 영화API를 통해 영화데이터를 가져와 json형식으로 저장하고, db에 에 저장하는 기능과, 사용자의 위치를 가져와 날씨API를 통해 날씨를 받아오고, 이를 이용해 사용자에게 영화를 추천하는 게능을 만들어줬다.

Template(templates)

템플릿 에서는 사용자가 실제로 눈에 보이는 영역이고 해당 코드는 현재 단순 디자인이기에 생략을 하겠다.

사용법

```
# git
git clone https://github.com/Jimin0605/OSP_TeamProject.git
```

```
# 가상환경 설정
pyhton -m venv venv
source venv/Scripts/activate
pip install -r requrements.txt
```

다음 스크립트를 작성해 루트폴더에 .env파일생성

```
OPEN_WEATHER_KEY = "cb1c6732992322afc6d1f746070fe99b"
SECRET_KEY = "antimony"
```

```
# flask 실행
source export.sh
flask run

# http://localhost:5000
# http://127.0.0.1:5000 접속
```