Final Project (~11/25)

유하랑 세진이랑

Aa 이름	날짜	텍스트
D3 VUE d3.selection version	@2022년 11월 22일	
profile pic DB	@2022년 11월 22일	
CSS	@2022년 11월 21일	
목적지 선택 화면 구현	@2022년 11월 21일	
사이트 이름 정하기	@2022년 11월 21일	
genre random 추천	@2022년 11월 20일	
Algorithm	@2022년 11월 20일	
like_movies/genre/lang	@2022년 11월 19일	
youtube 관련 영상	@2022년 11월 19일	
review - url 연결	@2022년 11월 18일	
API - YOUTUBE Trailer & Teaser	@2022년 11월 18일	
django-vue 기능 연동	@2022년 11월 18일	
vue 유저 부분 / 홈화면	@2022년 11월 17일	
ER 그램 마무리	@2022년 11월 17일	
Front-end	@2022년 11월 16일	
Back-end	@2022년 11월 16일	
Movie.json data dump & load	@2022년 11월 16일	
ER Diagram 작성	@2022년 11월 16일	
템플릿	@2022년 11월 15일	기획 의도 / 전체 목차 마인드 맵 / Vue 구조 , Django 구조
다국어	@2022년 11월 15일	첫 화면 : 지도 이미지 + 여러 갈래의 라우터를 이용해서 언어를 선택하게 = 영어, 중국어, 일본어, 한국어

사이트 이름 후보

- Fly-In-Cinema : FIC
- Fly-In-Theater
- Fly-In-Theatre
- AirBird

URL 정리

config (project - 세팅)	
path('admin/', admin.site.urls)	
path('movies/', include('movies.urls'))	
path('accounts/', include('dj_rest_auth.urls'))	
path('accounts/signup/', include('dj_rest_auth.registration.urls'))	
movies (application - 메인)	
# 영화 데이터 로드 # path('tmdb/', views_tmdb.tmdb_data),	

# 홈 페이지 path('', views.home),	
# 배우 path('actors/', views.actor_list), path('actors/int:actor_pk/', views.actor_detail),	
# 영화 path('movies/', views.movie_list), path('movies/int:movie_pk/', views.movie_detail),	
# 리뷰 수정 후 path('int:movie_pk/review_list_create/', views.review_list_create), path('review/int:review_pk/', views.review_update_delete),	# 리뷰 수정 전 (오류 : 영화별로 리뷰를 달려고 했는데, 각 영화에 모든 리뷰가 출력된다!) # path('reviews/', views.review_list), # path('reviews/int:review_pk/', views.review_detail), # path('movies/int:movie_pk/reviews/', views.review_create),
# 좋아요 _ 영화 선택 path('<int:my_pk>/<movie_title>/like/', views.like_movies), # 사용자가 좋아요한 영화 목록 넘겨주기 (프로필 표시용) path('yourmovie/', views.yourmovie, name='yourmovie'),	
# 영화 추천	좋아요 없으면 인기영화, 있으면 유사 장르
path('recommend/', views.recommend, name='recommend'),	홈화면에 띄움
accounts (application - 계정)	
# 모든 유저 조회 path('users/', views.users),	
# 회원가입 path('signup/', views.signup),	vue에서 가능하면 필요없을지도
# jwt # path('api-token-auth/', obtain_jwt_token),	jwt 토큰 안 쓰면 필요 없음
# 내 프로필 조회 path('myprofile/', views.my_profile),	
# 다른 user의 프로필 path('<username>/profile/', views.user_profile),]	
# 팔로우 path('follow/int:my_pk/int:user_pk/', views.follow),	
# 프로필 사진 관련 url	NEW!

다국어 서비스 제공

- HOME : 사용자를 모국어 사이트로 안내하는 전용 랜딩 페이지

알고리즘

1. 영어권 / 중화권 / 일어권 / 한국어권 관객이 선호하는 영화 추천
    a. 각 언어별 사이트의 영화 click 수 혹은 좋아요 click 수 데이터 수집
    b. 영화 추천 사이트 1 생성 (각 국가별 추천 영화 5~10 개 랭킹 뜨게 구현)
2. 여행 목적지행 비행기에서 볼 영화 추천
    a. 목적지 선택
    b. movie의 language 정보를 기반으로
    c. 영화 추천 사이트 2 생성
    d. 디자인 : 지도 + 비행기 + 비행기 창문 (넘어가는 영화 팸플릿)
3. 사용자가 좋아요 누른 영화 장르 기반 추천
4. 사용자가 가입시, 선호하는 영화를 선택하고 그에 기반하여 영화 추천

기획 의도

- 사용자 친화적 영화 추천
    - 상황별 / 사용자별 추천

- 언어에 따른 영화 추천 (현재 생활권 - NOW)
  - 언어 생활권에 따라 사용자의 정서가 다르기 때문에 같은 언어를 사용하는 관객들이 선호하는 영화에는 공통된
- 여행 장소에 따른 영화 추천 (새로운 경험 - NEW!)
  -

다른 나라 문화를 직접 체험하기 이전에 간접 체험 → 깊은 이해

→ 외국인 친구들 : 다른 나라 영화 → 홍보? 다국어?
→ 문화 컨텐츠에는 국경이 없다

: 영화 사이트 ? 이유

→ (사용자 친화적)

- 언어
- 목적
- 상황

코로나로 인해 2년여간 불가능했던 해외 여행
지금은 상황이 점차 완화되어
여행이 가능해짐

준비하는 것부터 여행이고
계획하는 과정도 여행이고

사실, 비행기를 타고 가는 시간동안 할 수 있는 것이 별로 없고
대부분의 사람들이 잠을 자거나 영화를 봄
(통계)

여행 예열 단계

과정까지 즐기기 위해

비행기 [모니터] ← 추천 알고리즘
목적지와 연관된 영화 추천

항공사 기내 영화 → 이런 알고리즘을 제공할 것이다.

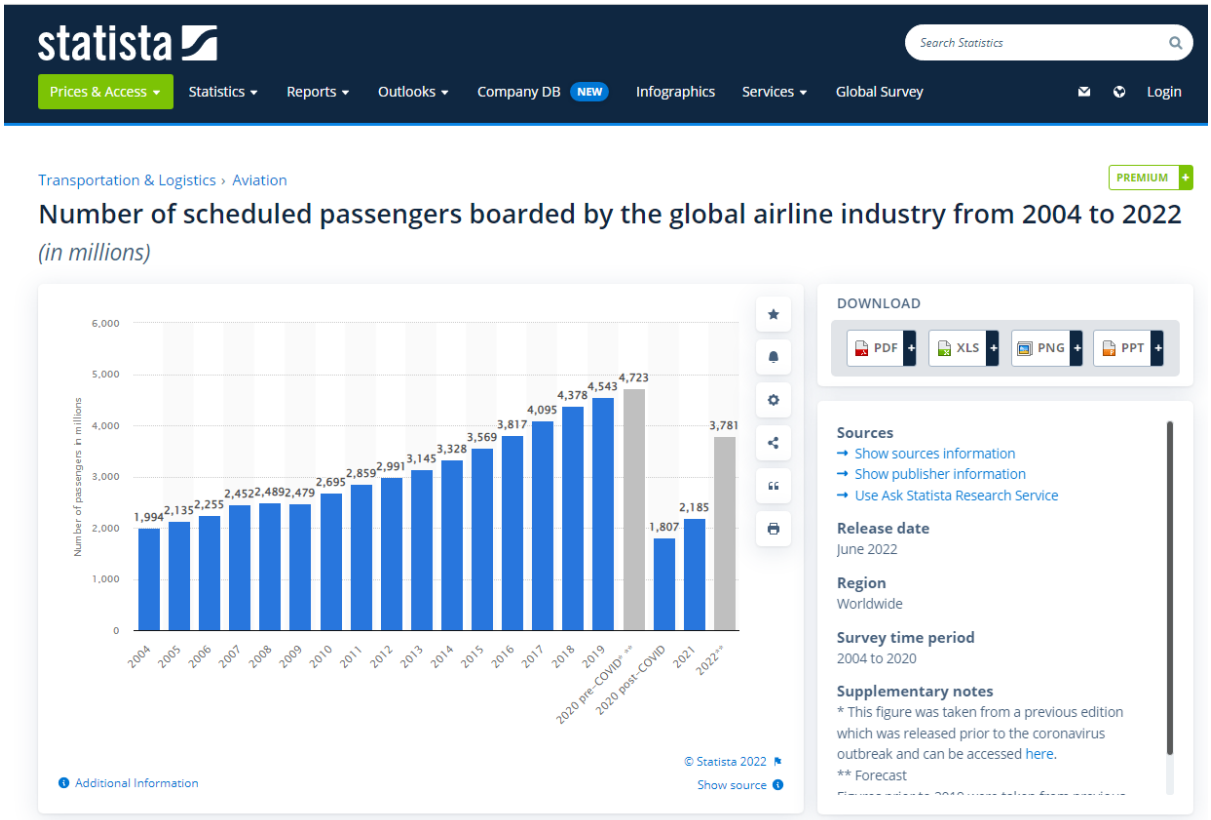
기내 영화 리스트

open api

여행 목적지

영화 판권 ——> 우리가 추천 (보는 사람 국적 혹은 목적지별 추천) ——> 항공사 적용

body > app-root > div > ke-sky-program > ke-basic-layout > div:nth-child(2) > div > div > ke-entertainment-result > div > ul > li:nth-child(1) > a > span.entertain-result__desc > span.entertain-result__info > span.entertain-result__rated.ng-star-inserted > span._hidden



항공 산업의 성장

https://www.statista.com/statistics/564717/airline-industry-passenger-traffic-globally/

1. 기업
    - 항공 산업의 성장
    - 코로나로 인한 성장세 약화 → 새로운 기획 필요, 사람을 끌어들일 섬세한 기획 ((2020 COVID_19 OUTBREAK))
    - 디지털 산업 성장
    - 항공 여행 산업과 디지털 미디어 산업의 결합
    - 기내에서는 개인 데이터 사용 불가능 → 항공사가 제공하는 디지털 서비스를 이용할 수밖에 없게 됨
    - 사용자의 시간을 가장 의미 있게 보내는 방법이 무엇일까?

Airbus and Boeing aircraft with in-flight entertainment and connectivity (IFEC) 2017

Number of in-service Airbus and Boeing narrowbodies worldwide in 2017, by IFEC offering

Global airline travelers - interest in digital screens interiors on airplanes 2019

Share of airline travelers worldwide who are interested in airplanes whose interior featured digital scr...

2. 사용자
    - 지루한/상투적인 기내 모니터 콘텐츠에 지침
    - 무슨 영화를 봐야할지 모름
        - 비긴어게인만 봄
        - 라라랜드
    - 여행지에 도착을하면
    - 어떤 나라는 아예 테마여행처럼 꾸며둔 곳이 있어서
    - 미리 그 영화를 봤으면 좋았을껄....
        - 오스트리아 - 짤츠부르크 - 사운드 오브 뮤직 (의 도시)
    - 사용자에서 의미있는 경험을 제공
    - 도움이 됨 (이해도 상승)

기사

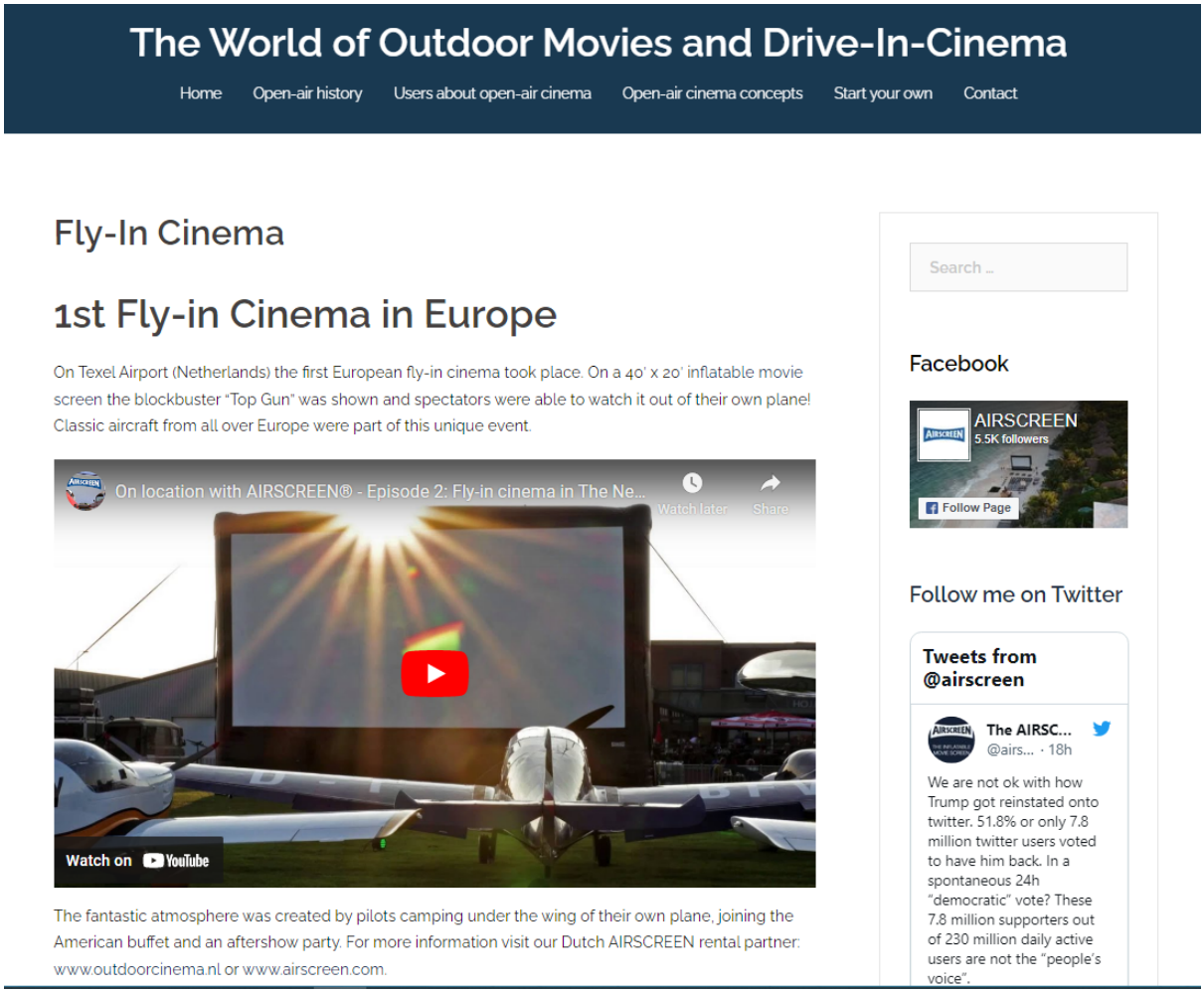
2016년 국제항공운송협회(IATA)가 전 세계 145개국 6920명의 승객을 대상으로 한 설문조사 결과를 보면, 취침(69%, 복수응답)보다 더 많은 77%의 응답자가 장거리 비행 때 영화감상을 하며 시간을 보낸다고 답했다. 대한항공 관계자는 “전체 탑승객의 99.9%가 비행하면서 짧게라도 기내 엔터테인먼트를 이용하고 있다”고 말했다.

In Flight Entertainment

https://www.hani.co.kr/arti/economy/marketing/910037.html

"기내 엔터테인먼트 시장은 매년 15% 이상 성장해 2020년까지 연간 7조원 규모에 달할 것으로 전망된다"고 분석했다. 한 항공업계 관계자는 "장거리 여행 수요가 늘어나고 다양한 항공사의 서비스를 접해본 승객들이 늘어나면서 항공사 간 기내 엔터테인먼트 경쟁도 거세지고 있다"며 "기내 엔터테인먼트가 그 항공사 이미지와 연결되기도 해, 기내 엔터테인먼트의 중요성은 더 커질 것으로 보인다"고 말했다.

자동차 극장처럼 경비행기 극장



https://openaircinema.blog/fly-in-cinema/

코로나 시기 목적지 착륙 없이 회항하는 비행 여행 상품

비행기 안서 대만 상품 제주관광 4분만에 완판

제주매일 | 입력 2020.09.14 | 댓글 0

코로나19로 국가간 해외여행이 제한된 상황에서 제주 상공을 선회한 후 회항하는 '가상 출국여행' 상품이 출시 4분만에 완판됐다.

한국관광공사에 따르면 공사 타이베이지사는 대만 중대형여행사 이지플라이(ezfly, 易飛網), 항공사 타이거에어(台灣虎航)와 공동으로 제주 상공을 여행하는 항공편 체험상품인 '제주 가상출국여행 얼리버드 프로모션' 상품을 11일 정오에 출시, 4분 만에 판매가 완료되는 인기를 보였다.

이 상품에는 대만관광객 120명이 참가한다. 19일 타이베이공항을 출발, 목적지인 제주공항에 착륙하지 않은 채 제주 상공을 선회한 뒤 대만으로 다시 회항하는 상품이다.

http://www.jejumaeil.net/news/articleView.html?idxno=211660

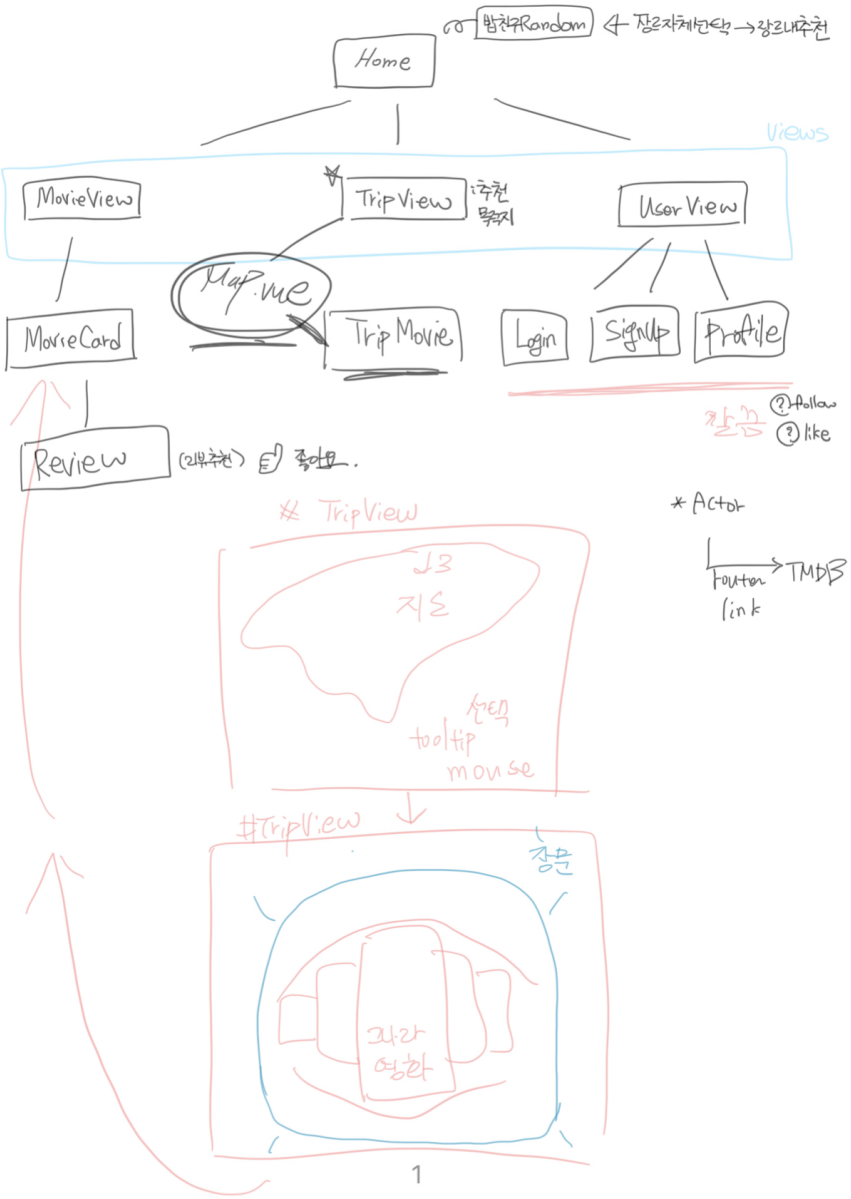
IDEA

- random lodash 밥 친구 영화
- d3 geo map
- 웹 디자인 (사진 / 동영상)
- detail.html → trailer/teaser 영상

1116

하세진 : 모델 구조화 Vue Movielist / Moviedetail / User / Login / Signup 뼈대 제작

장유하 : backend django 모델링, d3 geomap 시도



1117

- [ ] Vue Detail Home 부분 디자인
- [ ] User 팔로우 좋아요 정보
- [ ] django base.html이용해서 vue와 연결하는 방법 :
- [ ] d3 지도 그리기
  - [ ] 지도 위에 핀버튼 달기
- [ ] https://www.webucator.com/article/connecting-django-and-vue/
- [ ] Serializer 완성
- [ ] 구동 확인
- [ ] 프로필 사진 + 영상 css

- 목적지 기반 영화 추천

지도 클릭 → d3

마우스 클릭 over → 오류

지도 그림만 그리고

핀 뱃지 버튼

→ 해당 언어코드 국가의 영화를 추천

- API 가져오기 - youtube teaser and trailer

TMDB - actors 정보

선택 장르 내 추천 영화 (밥 친구)

en / ch / ja / kr / es / fr

router 번역 : 다음주

1118

- 100 개 영화 언어 종류 :
  - en - **English**
  - de - **German**
  - te - India
  - ja - Japan
  - es - Spain
  - fr - French
  - zh - Chinese
  - nl - Dutch
  - pt - Portuguese
  - uk - Ukrainian
- 라우터 페이지 번역 :
  - ch jp kr en
- url 공유 → 수정 시, 노션에 기록
- back-end 수정

```
# Django edit

1. movies/urls.py 에서
- review_detail url을 삭제
- review comment, community, comment도 삭제

2. views.py 에서
- jwt 관련 문장 주석 처리
- 리뷰 상세 조회 관련 함수 (review_detail) 삭제
- 리뷰 수정 및 삭제 관련 함수 (review_update_delete) 에서 serializer를 ReviewSerializer -> ReviewListSerializer 로 변경
- raise_exception=True 가 이미 있는 함수에서는 status HTTP 를 따로 써 주지 않아도 되니, 둘 중 하나만 씀
- 리뷰 댓글 / 커뮤니티 / 코멘트 삭제
- 리뷰 수정 삭제 시, 사용자 권한 관련 조건 삭제
- 리뷰 생성 시, 사용자 정보도 넘겨주기 request.user

3. serializers.py 에서
- ReviewListSerializers의 fields = '__all__' 로 변경해서, 리뷰 조회 시에 id 값이 뜨게 함 -> 수정 및 삭제 편의성을 위해
```

```
- 리뷰 관련 시리얼라이저에 movie는 read_only_fields 로 설정: 유효성 검사에서 제외시키기위해서

# POSTMAN
Headers
Key
Authorization
Value
Token 4848f50748f983e936f0034384d54b271f92910a
```

- 밥 친구 → 기내식 친구
  - 사육…

```
# Django revised

수정 부분
1. accounts/urls.py 수정 -> 노션 반영
- 장고 유알엘, 뷰 유알엘 둘다 살림
- 영화 유저 좋아요 부분 주석 처리
- 남의 프로필 유알엘 조금 수정
    # 다른 user 프로필
    path('<username>/profile/', views.user_profile),

2. accounts/views.py 수정
- 조회 POST 아니고 GET
- 함수 순서 위치 맥락 비슷하게 변경
- 영화 유저 좋아요 부분 주석처리
- jwt 토큰 주석처리
3. movies/views.py 수정
- 마지막 함수 ( 이 영화를 좋아하는 사람들) 주석처리
```

- geomap 그려낸 지도 위에 위도 경도 찍어서 나라 몇 군데 클릭 가능하게 만들기

```
# Vue edit

1. review movie 별로 나오게
2. review 생성
3. review 삭제
- 아직 삭제 후 갱신 되는건 X
4. v-on 에 데이터 담아서 보내기
  <ul>
        <li> 제목 : {{ review.title }} </li>
        <li> 작성자 : {{review.userName}} </li>
        <button @click="reviewDelete(review, $event)">delete</button>
  </ul>
5.vue url 다시 확인 (현재 django와 모두 동일)
```

1119

MovieDetail 에 Youtube 관련 컴포넌트 생성 / 관련 Youtube 리스트 (아직 스타일 적용X)
장르 누르면 장르가 같은 영화가 뜨도록 만듦 >> 내일 그 리스트중에서 랜덤으로 가져오는거 적용
좋아요한 영화 데이터 장고로 보내기까지 완료 >> 가져오는건 아직

- 좋아요 (좋아하는 영화 / 좋아하는 장르)
  - 초반에 만들어둔 '이런 영화를 좋아하는 유저들' 관련 Url은 일단 주석 처리

        - #path('int:my_pk/like/users/',views.likeusers)
    - 일단은, 내가 좋아하는 영화 / 장르에 집중

알고리즘 관련 고민

1. { 지도 페이지 } 지도에서 특정 나라/언어 선택하면 → 해당하는 original_language 의 영화들 중 (랜덤) 추천
    a. movies.json 표본
    - en - **English**
    - de - **German**
    - te - India
    - ja - Japan
    - es - Spain
    - fr - French
    - zh - Chinese
    - nl - Dutch
    - pt - Portuguese
    - uk - Ukrainian
2. { 영화 추천 페이지 } 기내식친구/비행친구
    a. 아예 랜덤 (vue lodash)
    b. user model의 like_movies 목록과 → 그걸 기반으로 한 like_genre 를 기반으로 해서 영화를 추천 (django views.py)
    c. 추천 페이지에서 원하는 장르를 선택하고 그 영화들 중에 랜덤 추천, 밥 시간 잔인한 영화 싫을수도.

accounts/models.py

```
from django.db import models
from django.contrib.auth.models import AbstractUser

# like => movie랑 genre
from movies.models import Movie, Genre

class User(AbstractUser):
    followers = models.ManyToManyField('self', symmetrical=False, related_name='followings')

    like_movies = models.ManyToManyField(Movie, related_name='user_movies') # 좋아하는 영화들
    like_genre = models.ManyToManyField(Genre, related_name='user_genre') # 좋아하는 장르
```

movies/urls.py

```
# 좋아요
    path('<int:my_pk>/<movie_title>/like/', views.likeornot), # 좋아하는 영화 선택하기
    path('<int:my_pk>/like/', views.Ilike), # 내가 좋아하는 영화들 데이터 넘겨줌
```

movies/views.py

```
# 좋아요 표시
# 1. like or not - checking
@api_view(['POST'])
# @authentication_classes([JSONWebTokenAuthentication])
@permission_classes([IsAuthenticated])
def likeornot(request, my_pk, movie_title):
  movie = get_object_or_404(Movie, title=movie_title)
  me = get_object_or_404(get_user_model(), pk=my_pk)
```

```
  if me.like_movies.filter(pk=movie.pk).exists():
      me.like_movies.remove(movie.pk)
      likeit = False
  else:
      me.like_movies.add(movie.pk)
      likeit = True
  return Response(likeit)

# 내가 좋아하는 영화들
# 2. I like these movies
@api_view(['POST'])
# @authentication_classes([JSONWebTokenAuthentication])
@permission_classes([IsAuthenticated])
def Ilike(request, my_pk):
  me = get_object_or_404(get_user_model(), pk=my_pk)
  data = []
  movies = request.data
  for movie_pk in movies:
    movie = get_object_or_404(Movie, pk=movie_pk)
    serializer = MovieSerializer(movie)
    data.append(serializer.data)
  return Response(data)
```

1120

vue - trip > 장르 누르면 랜덤으로 추천

django - like_movies 기반 영화 추천 알고리즘 JSON error 해결

1121

포스터 이미지가 없는 영화일때 오류 뜨는거 잡기

유저가 로그아웃 했을때 유저 정보 + 토큰 + 좋아요 목록 다 지우기

d3 marker 넣기

```
# movies/views.py _ yourmovie 함수 내용 변경

# 사용자 개인이 좋아하는 like_movies 정보 넘겨주기
@api_view(['GET'])
@permission_classes([IsAuthenticated])
def yourmovie(request):
  user = get_user_model()
  me = user.objects.get(username=request.user.username)
  likes = me.like_movies.all()
  # print(likes.count()) # 6번 사용자의 경우, 4편의 영화 나옴
  serializer = MovieListSerializer(likes, many=True)
  return Response(serializer.data, status=status.HTTP_200_OK)
```

- 홈 화면에 띄울 영화 추천 알고리즘 구현
  - 좋아요한 영화가 없는 유저의 경우, 인기 순 10개 영화 추천
  - 좋아요한 영화가 있는 유저의 경우, 유사 장르 10개 영화 추천

```
from django.shortcuts import render
from functools import partial
from django.shortcuts import render
from .models import Actor, Movie, Genre, Review
from .serializers import ActorListSerializer, ActorSerializer, MovieListSerializer, MovieSerializer, ReviewListSerializer, ReviewS
from django.shortcuts import get_list_or_404, get_object_or_404
from rest_framework.decorators import api_view
from rest_framework.response import Response
from rest_framework import status
from movies import serializers
from rest_framework.decorators import api_view, permission_classes, authentication_classes
from rest_framework.permissions import IsAuthenticated
from django.contrib.auth import get_user_model
```

```
# from rest_framework_jwt.authentication import JSONWebTokenAuthentication
from django.http import JsonResponse
from django.forms.models import model_to_dict

# 내가 좋아하는 영화 선택 ( 좋아요! 버튼 ) -> user model에 like_movies
# 1. 특정 영화에 좋아요를 하는 함수
@api_view(['POST'])
# @authentication_classes([JSONWebTokenAuthentication])
@permission_classes([IsAuthenticated])
def like_movies(request, my_pk, movie_title):
  movie = get_object_or_404(Movie, title=movie_title)
  me = get_object_or_404(get_user_model(), pk=my_pk)
  if me.like_movies.filter(pk=movie.pk).exists():
      me.like_movies.remove(movie.pk)
      likemovies = False
  else:
      me.like_movies.add(movie.pk) # 내가 좋아하는 영화 목록에 추가
      likemovies = True
  return Response(likemovies)

# 좋아요 한 영화들(like_movies)의 장르를 추출해서 좋아하는 장르 기반 영화 추천 def
# 2. 당신이 좋아할만한 영화 추천

@api_view(['GET'])
@permission_classes([IsAuthenticated])
def yourmovie(request):
  user = get_user_model()
  me = user.objects.get(username=request.user.username)
  likes = me.like_movies.all()
  # print(likes.count())
  serializer = MovieListSerializer(likes, many=True)
  return Response(serializer.data, status=status.HTTP_200_OK)

# 3. 추천
@api_view(['GET'])
@permission_classes([IsAuthenticated])
def recommend(request):
  user = get_user_model()
  movies = Movie.objects.all()  # 무비 모델에서 무비 정보들은 전부 가져옴 ( 쿼리 셋 )
  me = user.objects.get(username=request.user.username)
  like_movies_genre = {}
  likes = me.like_movies.all().values() # 좋아하는 영화 정보 받아오기

  # print(likes)

  # 좋아요한 영화 쿼리셋이 존재하지 않을 경우, 평점 높은 영화 10개 추천

  if not likes:

    popular_movies = Movie.objects.all().order_by('-vote_average')

    # print(popular_movies) # 평점 9.1 짜리 먼저 뜸

    # pop_dict = model_to_dict(popular_movies)

    # print(pop_dict)

    pops_num = 0
    pop_movies = []

    # pop_list = []

    for pop_movie in popular_movies:

      # 쿼리셋 -> dict 변환 필요

      pop_movie_data = model_to_dict(pop_movie)

      # pop_list = list(pop_movie_data)

      print(pop_movie_data)
      # print(pop_list)

      # for movie in pop_movie_data:

        # pop_list.append(movie)

        # print(pop_list)

        # print(movie[0]['id'])

        # movie는 하나야

        # print(movie)

      needs = { 'id' : pop_movie_data['id'], 'title' : pop_movie_data['title'], 'poster_path': pop_movie_data['poster_path'], 'ori
```

```
      # needs = { 'id' : 82007, 'title' : 'sth' }

      print(needs)

      pop_movies.append(needs)

      # pop_movies = [ { 영화 1의 id, title...  }, { 영화 2의 ...  } ]

      pops_num += 1

      # print(needs)

      if pops_num == 10:
        break

    # print(pop_movies)
    # TypeError: string indices must be integers

    context = {
      'movies':pop_movies,
    }

    return Response(context)

  # OR

  # 좋아요한 영화 쿼리셋이 존재할 경우, 알고리즘 기반 추천

  else:

    for movie in likes:
      movie = get_object_or_404(Movie, pk=movie['id'])
      movie_g = movie.genres.values()

      for genre in movie_g: # 장르들 중에 장르를 뽑았는데
        try:
          like_movies_genre[genre['id']] += 1
        except:
          like_movies_genre[genre['id']] = 1

    # print(movie_g)

    you_like = []

    for movie in movies:
      movie_data = model_to_dict(movie)

      if movie in likes:
        you_like.append((movie_data, 0))  # 이미 본 영화 제외해야하니까 0 주기 #

      else: # 모든 영화 중 안 본 영화들 대상으로
        whole = movie.genres.all() # 무비에 장르스 없음
        cnt = 0
        for genre in whole:
          if like_movies_genre.get(genre.pk, False):
            cnt += like_movies_genre.get(genre.pk)
        if cnt:
          you_like.append((movie_data, cnt))

    you_like.sort(key=lambda x: x[1], reverse=True) # 큰수부터 추천

    # print(you_like)

    recommend_movies_num = 0
    context_movie = []

    for movie in you_like:
      needs = { 'id' : movie[0]['id'], 'title' : movie[0]['title'], 'poster_path': movie[0]['poster_path'], 'original_language':mo
      context_movie.append(needs)

      # print(needs)

      recommend_movies_num += 1
      if recommend_movies_num == 10: # 10개까지
        break

    # print(context_movie)

    context = {
      'movies':context_movie,
    }

    return Response(context)
```

> 문제

- 좋아요한 영화에 존재하는 영화는 추천 리스트에서 제외하고 싶은데 pass 나 remove 를 해도 좋아요한 아바타가 추천 리스트에서 사라지지 않는다. query set 형태라서 그런걸까?

- 내일 할일 : 프로필 사진 db 저장 및 생성 & 수정