



# Final Project (~11/25)

## 유하랑 세진이랑

Aa 이름	📅 날짜	≡ 텍스트
<u>genre_random 추천</u>	@2022년 11월 20일	
<u>Algorithm</u>	@2022년 11월 20일	
<u>like_movies/genre/lang</u>	@2022년 11월 19일	
<u>youtube 관련 영상</u>	@2022년 11월 19일	
<u>review - url 연결</u>	@2022년 11월 18일	
<u>API - YOUTUBE Trailer &amp; Teaser</u>	@2022년 11월 18일	
<u>django-vue 기능 연동</u>	@2022년 11월 18일	
<u>vue 유저 부분 / 홈화면</u>	@2022년 11월 17일	
<u>ER 그램 마무리</u>	@2022년 11월 17일	
<u>Front-end</u>	@2022년 11월 16일	
<u>Back-end</u>	@2022년 11월 16일	
<u>Movie.json data dump &amp; load</u>	@2022년 11월 16일	
<u>ER Diagram 작성</u>	@2022년 11월 16일	
<u>템플릿</u>	@2022년 11월 15일	기획 의도 / 전체 목차 마인드 맵 / Vue 구조, Django 구조
<u>다국어</u>	@2022년 11월 15일	첫 화면 : 지도 이미지 + 여러 갈래의 라우터를 이용해서 언어를 선택하게 = 영어, 중국어, 일본어, 한국어

## 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 영화 에 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), # 좋아요 한 영화들의 장르를 추출해서 좋아하는 장르 기반 영화 추천 path('yourmovie/', views.yourmovie, name='yourmovie'),	
# 좋아하는 장르 선택 -> like_genre 에 해당 장르 영화들 저 장 -> vue 에서 해당 목록 내에서 lodash 랜덤 path('<int:my_pk>/<genre_name>/like/', views.like_genre),),	vue 에서 목록 받아서 랜덤 돌리는 게 안 되면, 그냥 장고에서 random 함수 써야함.
# 랜덤 일단 살려둬 path('yourgenre/', views.yourgenre, name='yourgenre'),	
# 선호하는 국가 언어 선택 -> like_lang 에 해당 언어 영화들 저장 -> vue 에서 해당 목록 내에서 lodash 랜덤 path('int:my_pk/<movie_original_language>/like/', views.like_lang),	
# 랜덤 일단 살려둬 path('yourlang/', views.yourlang, name='yourlang'),	
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),	

## 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 콘텐츠에는 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

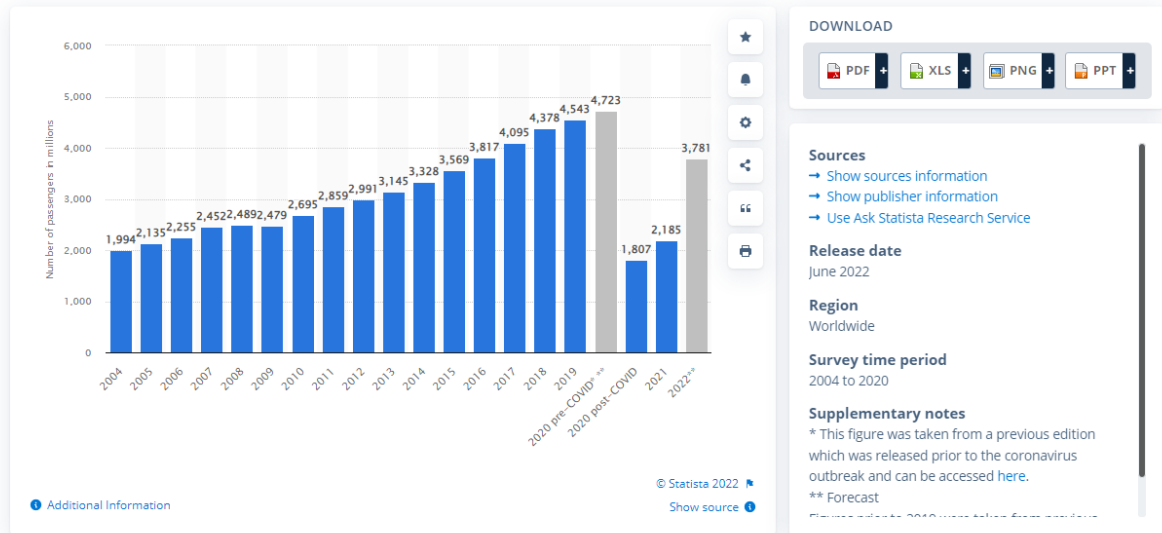
영화 판권 ——> 우리가 추천 (보는 사람 국적 혹은 목적지별 추천) ——> 항공사 적용

```
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
```

Transportation & Logistics > Aviation

PREMIUM +

## Number of scheduled passengers boarded by the global airline industry from 2004 to 2022 (in millions)



### 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 끌어들이는 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 좋았을걸....
  - 오스트리아 - 팔츠부르크 - 사운드 오브 뮤직 (의 도시)
- 사용자에서 의미있는 경험을 제공
- 도움이 됨 ( 이해도 상승 )

## 기사

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

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



“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.

## IDEA

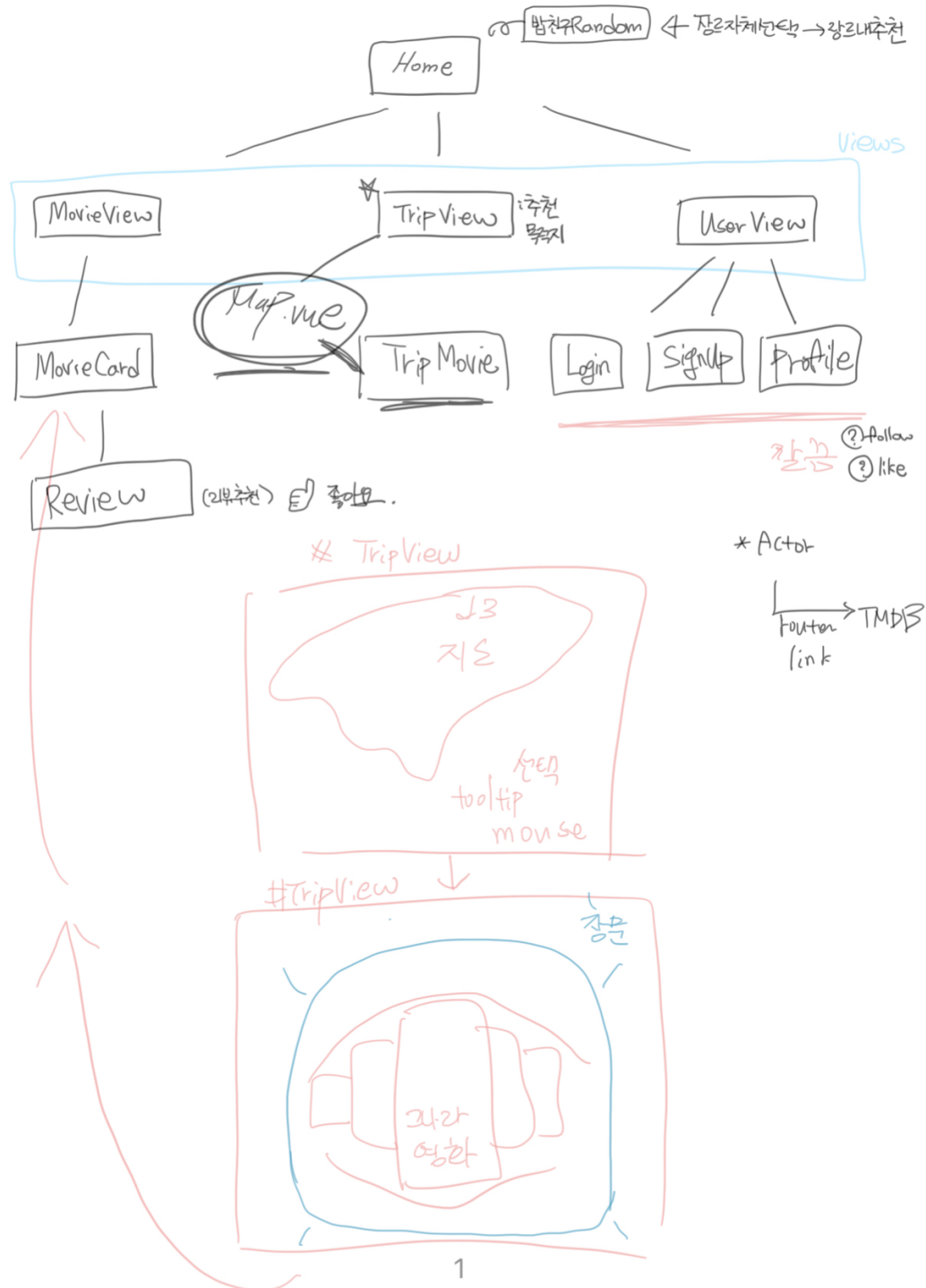
- random lodash 밥 친구 영화
- d3 geo map
- 웹 디자인 ( 사진 / 동영상 )
- detail.html → trailer/teaser 영상

---

## 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, 노선에 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 -> 노선 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 만들 >> 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

- 아예 랜덤 ( vue lodash )
- user model의 like\_movies 목록과 → 그걸 기반으로 한 like\_genre 를 기반으로 해서 영화를 추천 (django views.py)
- 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