

## 5기? 6기? 기획서

### 1. 기획 및 역할 분배

#### 기획

🖼️! [image-20210527204309050] (README.assets/image-20210527204309050.png)

코로나로 인해 대면하지 않는 삶이 일상이 되었습니다. SSAFY 5기 면접부터 비대면으로 진행되었고, 우리 모두 다같이 모인 적은 아직 한 번도 없죠.

사람과 만남이 적어지며, 개인이 점점 외로워지고 있는 문제를 다루고 싶었기에 GUVIE 서비스를 기획하게 되었습니다.

GUVIE는 기본적인 영화 정보와 함께 영화를 볼 사람들을 찾아가는 **살롱**, 내게 맞는 영화를 찾을 수 있는 **맞춤형 추천 서비스**를 메인 기능으로 하고 있습니다.

#### 역할 분배

서비스는 프론트엔드(Vue)와 백엔드(Django)로 나누어 작업하였으며, 프론트엔드는 제가 담당하였고 백엔드는 여래님께서 담당하셨습니다.

### 2. 백엔드

### 3. 프론트엔드

#### 기술 스택

```
- Vue
  - VueRouter
  - Vuex

- Module
  - tailwindCSS (Styling)
  - fuse.js (검색)
```

#### 기술 스택

```
- Vue
  - VueRouter
  - Vuex

- Module
  - tailwindCSS (Styling)
  - fuse.js (검색)
  - moment (Date Formatter)
  - swiper (Carousel)
  - v-calendar (DateTime Picker)
```

select a language

#### 페이지 구성

🖼️! [image-20210527153615147] (README.assets/image-20210527153615147.png)

```
- (/) Home
  - (/intro) 서비스 소개
  - (/User) 이용자
    - (/login) 로그인
    - (/signup) 회원가입
    - (/:nickname) 프로필
  - (/Movies) 영화
    - (/) 영화 리스트
    - (/:movie_id) 영화 상세정보
  - (/Salons) 살롱
```

- (/create) 상품 생성
- (/salon\_id) 상품 상세정보

select a language

## Home

Home은 영화 섹션과 영화관 섹션으로 구성되어 있습니다.

- 영화 섹션

여러분께서 작업하신 협력 필터링 알고리즘에 의해 추천되는 "추천 영화" (12개)

🖼️! [image-20210527160352580] (README.assets/image-20210527160352580.png)

현재 상영중인 영화 (12개)

🖼️! [image-20210527160444031] (README.assets/image-20210527160444031.png)

각 영화 아이템은 앞면은 포스터, 뒷면은 영화 정보로 구성되어 있습니다.

🖼️! [image-20210527160519628] (README.assets/image-20210527160519628.png)

- 영화관 섹션

CGV와 MEGABOX에서 이벤트 정보를 가져와 보여주며, 클릭을 통해 해당 사이트로 이동할 수 있습니다.

## Salon

- 상품 리스트

다른 이용자들이 만든 상품들을 확인할 수 있는 페이지입니다.

참여자가 많은 순 / 적은 순 / 전체 상품 별로 구분하여 작성되었습니다.

swiper 라이브러리를 활용하여, 목록을 옆으로 넘길 수 있도록 하였습니다.

상품 생성과 상품 상세정보 페이지로 이동할 수 있습니다.

🖼️! [image-20210527162126347] (README.assets/image-20210527162126347.png)

I

- 상품 생성하기

상품을 생성할 수 있는 페이지로 로그인 되어야 접근할 수 있는 페이지입니다.

링크 버튼을 통해 접근하는 방법과 직접 주소를 입력하여 접근하는 방법이 있을 수 있다고 생각하여, 두 방법에 대해 다음과 같이 대책을 세웠습니다.

1. 링크 버튼 클릭 시

로그인 여부를 확인 후 로그인하지 않은 사용자라면, 로그인 요청 모달을 띄움

🖼️! [image-20210527164148026] (README.assets/image-20210527164148026.png)

2. 직접 주소를 입력한 경우

vue router의 네비게이션 가드를 활용하여, 로그인 페이지로 리다이렉트

```
<script>
export default {
  // ...
  beforeRouteEnter(to, from, next) {
    next((vm) => {
      console.log(vm)
      if (vm.$store.state.user.id === -1) {
        next({ name: "login" })
      }
    })
  },
  // ...
}
</script>
```

## Profile

유저의 기본 정보를 확인할 수 있으며, GUVIE에서 활동한 정보를 확인할 수 있습니다.

! [image-20210527173636435] (README.assets/image-20210527173636435.png)

프로필 수정 페이지에서는 회원 정보 중 프로필 이미지, 이메일, SNS 주소를 추가 혹은 변경할 수 있습니다.

SNS의 경우 올바른 주소형태여야 합니다.

변경 전 / 후 이미지를 확인할 수 있습니다.

! [image-20210527173908979] (README.assets/image-20210527173908979.png)

## • Login

아이디와 비밀번호를 통해 로그인할 수 있으며, SNS 로그인 기능을 기획하였습니다.

! [image-20210527173959513] (README.assets/image-20210527173959513.png)

## • Signup

가입 시 이메일은 선택적으로 입력할 수 있으며, 프로필 이미지는 프로필 수정 페이지에서 변경할 수 있습니다.

! [image-20210527174048185] (README.assets/image-20210527174048185.png)

## • Intro

GUVE 제작자 설명과 안내 영상을 확인할 수 있습니다.

! [image-20210527174152650] (README.assets/image-20210527174152650.png)

! [image-20210527174236651] (README.assets/image-20210527174234954.png)

## 2. 프로젝트

### 1. 협업 툴

#### • Notion

문서 정리 및 API 정리를 위해 사용하였습니다.

### Project

1. 프론트엔드 / 백엔드 나누어 작업을 하는 동안, 프론트엔드는 TMDB를 사용하여 더미데이터처럼 사용했었다. 전달받는 데이터의 형태는 백엔드와 유사하기 때문에 백엔드와 연결할 때 큰 어려움은 없었는데, 그래도 생각보다 시간이 많이 걸렸었다. TMDB와 같은 대체할 수 있는 더미데이터가 있는 경우는 많지 않을 것 같은데, 어떻게 하면 원활하게 진행할 수 있을까?

답은 딱히 없다.

### Vue

1. v-model로 Input과 binding을 했을 때, 영어는 글자 하나하나가 바로 state에 반영이 되지만 한글은 자음, 모음이 완성된 뒤에 다른 문자를 입력했을 때 반영이 된다.

@change 이벤트를 사용하여 event.target.value로 state를 직접 업데이트

2. 로그인을 해야 접근할 수 있는 페이지를 막으려면 어떻게 해야 할까?

Vue-router 네비게이션 가드를 사용한다.

beforeRouteEnter는 Vue Instance가 생성되기 전에 호출되기 때문에 this에 접근할 수 없습니다.

next 인자에 생성 될(아직 생성은 안되었으니) 인스턴스를 전달하여, 접근할 수 있습니다. 네비게이션이 확인되어 뷰 컴포넌트 인스턴스가 생성되어 콜백 인자로 전달되는 것입니다.

```
<script>
export default {
  // ...
  beforeRouteEnter(to, from, next) {
    next((vm) => {
      console.log(vm)
      if (vm.$store.state.user.id === -1) {
        next({ name: "login" })
      }
    })
  },
  // ...
}
</script>
```

3. 동일한 컴포넌트 내에서 동적 라우팅으로 이동하는 경우가 생겼을 때, 두 페이지가 다음을 나타내려면 어떻게 해야 할까?

## MoVS - VS 기반 영화 평가, 추천 커뮤니티

SSAFY 4기 final PJT 서울 3반 구본혁(qhsgur0126@gmail.com) 조단원(de.chloeee@gmail.com)

목차

### MoVS - VS 기반 영화 평가, 추천 커뮤니티

#### Introduction

1. 프로젝트 제목 :
2. 프로젝트 목적:
3. 배포 서버 URL
4. Installation

#### TEAM MoVS

1. 팀원 소개
- 구본혁
  - 조단원

#### 2. 협업 환경

GitLap (<https://lab.ssfy.com/ssafy4>)  
Discord  
Slack  
Google Spreadsheet (<https://docs.google.com/spreadsheets/d/1VZCIdi66qfj5fawNWYi7C5fWRllq2SIEYAPnllZOLk/edit?ts=5fb5f298#gid=795179377>)

#### Project MoVS

1. 개발환경
  - A. 언어
  - B. 도구
  - C. 아키텍처
2. 서비스 구현 목표 및 달성 정도
  - A. 초기 서비스 구성 목표
  - B. 실제 구현 정도
3. 핵심 기능
  - A. VS 기반 영화 평가
  - B. 날짜 기반 영화 평가 / 기본 태그 기록
  - C. 타임라인 형태로 지난 기록 조회
  - D. 영화 평가에 댓글을 남기는 커뮤니티 기능
  - E. 내 영화 평가 기반 영화 추천
4. 데이터베이스 모델링
5. 추가적인 구현 목표
  - A. 커뮤니티 기능 강화
  - B. 뷰포트 및 모바일 반응성 고도화
  - C. 영화 VS 기능 고도화

#### Notes & History

### Introduction

#### 1. 프로젝트 제목 :

- MoVS - Record your moments

#### 2. 프로젝트 목적:

- 게이미피케이션을 적용한 영화 평가 서비스 구성
- 평가 기반 영화 추천 서비스 구성
- 커뮤니티 서비스 구성

#### 3. 배포 서버 URL

- .

#### 4. Installation


- final-pjt-front 를 pull 받아 서버를 실행시킵니다.
- final-pjt-back 을 pull 받습니다.
  - 가상 환경을 설정하고 requirements를 install 합니다.
  - DB를 구성합니다.
    - `python manage.py loaddata accounts/fixtures/users.json`
    - `python manage.py loaddata movies/fixtures/movies.json`
    - `python manage.py loaddata movies/fixtures/db.json`
  - 서버를 실행시킵니다.
- TEST용 계정
  - ID : awe1 ~ awe200 까지 무작위로 사용 가능합니다.

## TEAM MoVS

팀원 소개, 업무분담내용

### 1. 팀원 소개


#### 구분혁

  />

무뚝뚝해 보이는 첫 인상과 달리 순둥한 코딩천재. 이게 어때요? 저거 넣을까요? 물어보면 잠시 후 "다 됐습니다." 하고 들고 오는 천재 백엔드면서, 화면도 구성 할 줄 아는 만능 플스택 개발자. 든든한 페어. 그러나 그와의 페어 프로젝트를 경험해보면, 그의 실력보다는 시려깊은 인성에 더 감탄하게 될 것입니다.

- 기획: 서비스 기획 및 홈페이지 목업 구성
- BE:
  - Django REST API 서버 구성
  - 기본 DB 모델링 및 구축
- FE:
  - request 및 response 기획 구성

#### 조단원

  />

첫 인상은 밝고 쾌활해서 과연 내가 따라갈 수 있을까? 라는 의문이 들었습니다. 같이 작업을 하보니 역시 밝게 열정 넘치게 이것저것 제안해주셨습니다. 걱정과 달리 잘 따라갈 수 있게끔 뛰어난 리더쉽을 발휘해 주셨습니다. 위의 성격 테스트는 상당히 정확했습니다. 페어 프로그래밍이 지루할 틈이 없도록 분위기도 좋게 만들어 주시고 기능적인 측면에서도 뛰어난 개발자입니다.

- 기획:
  - 서비스 기획 및 홈페이지 목업 구성
  - 브랜딩 컨셉 기획
  - VS 기반 영화 평가 알고리즘 및 추천 알고리즘 구현
- FE:
  - 사용자 친화 UX/UI 기획, 구성

### C. 아키텍처

- Django REST API 서버 & Vue.js & Vuex

### 2. 서비스 구현 목표 및 달성 정도

#### A. 초기 서비스 구성 목표

- 32개 영화를 대상으로 VS 시스템을 활용하여 영화 평가하기
- 해당 평가를 기반으로 자동으로 별점 매기기
- 평가 기반 커뮤니티 기능 만들기
- 모바일 대응을 위한 반응형 웹 서비스 설계하기
- JS를 적극적으로 활용하여 UX, UI 측면 고도화 하기 - 이용자에게 WOW를 주는 서비스 경험!

#### B. 실제 구현 정도

- 핵심 기능 빠대는 대부분 구현했다. 90% 정도!
- 32개 후보 영화 선정 방법을 고도화하지 못한 것이 아쉽다.
- 모바일 반응형 부분을 충분히 고려하지 못한 점도 아쉽다.

### 3. 핵심 기능

#### A. VS 기반 영화 평가

- TMDB API 활용, popularity 상위 1,000 개 영화 DB 확보
- DB 중 무작위로 32개 영화 선정
- 주어진 32개 영화를 순차적으로 2개씩 노출하여, 더 좋아하는 영화를 고르는 방식으로 평가
- 평가 결과는 자동으로 별점에 반영되어 DB에 기록되며, 변경 불가

순위	별점	영화 갯수
1위	★★★★★	1
2위	★★★★	1
3 - 4위	★★★	2
5 - 8위	★★	4

9 - 16위	★	8
이하	없음	16

## B. 날짜 기반 영화 평가 / 기본 태그 기록

- 평가 결과는 해당 날짜를 기반으로 1 건의 게시물로 기록
- 1위 영화에 대한 코멘트, 그 날의 기본 태그를 필수적으로 포함하여 기록
  - 기본 태그: 기배요, 술파요, 썰레요, 외로워요, 그리워요

## C. 타임라인 형태로 지난 기록 조회

My Moments

- 최근 기록부터 역순으로 조회 가능
- 타임라인 첫 화면에는[기록 날짜, 1위 영화 제목, 영화에 대한 코멘트, 기본 태그 노출
- 해당 기록 클릭시 1-4위까지 영화 평가 세부사항 조회 가능

## D. 영화 평가에 댓글을 남기는 커뮤니티 기능

- 영화 평가 세부 사항 조회시 댓글 남기기 가능

## E. 내 영화 평가 기반 영화 추천

Recommendation

- 내 가장 최근 영화 평가 기록을 기반으로 다른 사람의 영화 평가 게시물 추천 노출
- 내 최근 영화 평가의 기본 태그 기반:
  - 동일한 기본 태그를 가진 평가 게시글을 무작위로 1건 선정하여 1, 2위 영화 노출
- 내 최근 영화 평가의 상위 영화 기반:
  - 내가 평가한 영화 중 2위 내의 영화를 기준으로, 해당 영화가 4위 내에 기록된 평가 게시글을 무작위로 1건 선정하여 1, 2위 영화 노출
- 그 외 모든 영화 평가 게시글을 대상으로, 무작위로 3건 선정하여 1, 2위 영화 노출

## 4. 데이터베이스 모델링

## 5. 추가적인 구현 목표

### A. 커뮤니티 기능 강화

- User 검색 기능, Follow-Follower 기능, 이에 기반한 다른 사람의 moment 조회하는 Feed view 생성

### B. 뷰포트 및 모바일 반응성 고도화

- 보다 깔끔한 UI 구축

### C. 영화 vs 기능 고도화

- 현재는 1,000개의 DB를 기반으로 32개 영화를 후보군으로 선정하다 보니 사용자가 보지 않은 영화도 함께 선정되는 경우가 많음. 따라서 평가 결과의 정확도를 낮추게 됨.
- 따라서, 사용자가 이미 본 영화 DB를 생성하여 활용하거나, 혹은 봤을 법한 영화를 중심으로 32개 후보군 구성하는 기획 필요
  - "보지 않았어요" 버튼을 추가하는 방식 도입 고려중

## Notes & History

### 본격 Side

- 기획 단계에서 철저히 많은 것을 해야한다는 것을 확실히 느꼈습니다. 페어와 잘 맞아서 중간중간 의견을 공유하며 수정하는 것도 별 달 없이 진행했지만 Git이 나 번수명 등 미리 정해놔야하는 것들에서 통일성의 부재로 문제가 조금 있었습니다. 기획에 시간을 조금 더 할애했다면 코드를 짜는 시간이 줄었을거라 생각됩니다.
- 답은 공식문서에 있습니다. 기능 구현에서 막히는 부분을 검색하면 많은 블로그들이 있지만 개발 환경, 버전 별로 상황이 다르기때문에 결국 문제를 해결해주는 것은 공식 문서에 나와있는 것들이었습니다.
- Django를 공부하면서 가장 어려웠던 부분은 Serializer와 ManyToMany였습니다. 저희 프로젝트를 구성하면서 해당 요소들이 필요한 경우가 있었습니다. 다시 복습하기에는 시간이 부족하다고 여겨져 결국 다른 방법으로 우회하여 구성했습니다. 지식의 부재는 결국 시간 소모로 연결됐습니다.
- 아직도 Git을 잘 모르겠습니다... 프로젝트가 무너질 정도로 꼬이진 않았지만 여러개의 문제가 발생했습니다. Git을 조금 더 공부할 필요가 있네요.

백엔드의 기본은 결국 데이터를 어떻게 저장하고 어떻게 보내주는 것이라고 생각합니다.

- Models : 데이터베이스를 구성함에 있어서 Django의 Models는 상당히 편리했습니다. 추가 사항이 있으면 추가하고, 불필요한 요소는 중간에도 삭제해서 migrate로 관리하는 것은 저희가 작업을 진행하면서 필요하다고 느껴지는 것들을 잘 조정할 수 있게끔 해줍니다.
- 나쁜 프론트 인기는 하면서 보드였지만 Vue는 렌더링되는 막이빙에 막다 데이터를 소켓하는 것이 조금 까나트웠습니나, created를 누르 삭제하긴 했지만 그 외 다른 기능도 써봐야겠습니다.
- Vuex를 제대로 활용한다면 더 잘 구현할 수도 있었겠지만 역시나 제 지식의 부재로 인해 한 번 적용해봤습니다. 시간이 조금 더 있었다면 Vuex를 더 활용했을 겁니다.

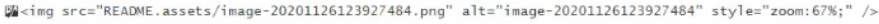
### 단원 Side

#### 프로젝트를 돌아보며:

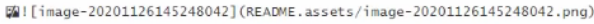
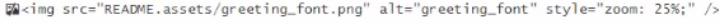
- 한 학기를 돌아보며 복습하는 것이 목표였다. 내 손으로 코드를 짜며 직접 주석을 달아보는 드라이버 역할을 해 보고 싶다는 욕심이 있었다. 그러나 페어가 너무나 나도 강했다... 내가 개입할 틈도 없이, 툭툭툭 모든 것을 쉽게 만들어버리는 페어 탓에(?) 서버 쪽에는 손도 대지 못했다. 페어를 따라갈 수 있는 실력을 길러야겠다는 다짐을 새롭게 했다.
- 현업 환경 구성은 물론이고 서비스 기획을 위한 다양한 통계를 사용해 보고 싶다는 욕심이 있었다. 어느 정도는 효과적으로 사용한 것 같아서 기쁘다. 새로운 도



## 프로젝트를 통해 배웠던 것들:

- Bootstrap 및 Flex 복습
  - HTML 및 CSS에 대한 제대로 된 이해 없이 덤벼들었을 때, 초기 디자인은 이런 모양이었다:  


```

```
  - 단순히 "예뻐보이기" 위해서가 아니라, 사용자가 우리 서비스의 핵심을 명확하게 이해하고 더 자주 사용할 수 있게 하기 위해서는 디자인 요소에 공을 많이 들여야 한다고 생각했다. 이를 위해 Bootstrap을 적절하게 활용하면서도, 사용자 경험의 핵심적인 부분은 브랜딩 컨셉에 맞춰 스타일링 하기 위해 노력했다.
- CSS Grid 학습
  - 몇 년 전부터, 디자인 트렌드로 Grid가 떠오르고 있다! (는 사실 CSS Grid와는 관련 없다)
  - Flex Box로는 원하는 모양새를 충분히 낼 수 없어서, CSS Grid를 도입했다. 확실히 가로, 세로 구성을 원하는 비율로, 원하는 배치대로 연출할 수 있다는 점이 좋았다.
- 브랜딩의 중요성
  - 초기, 우리 서비스는 "로맨스 영화"만을 대상으로 했다. 그래서 전체적으로 감성적이고 몽글몽글한 느낌을 주기 위해 노력했다. 이를 위해서
    - 브랜드 컬러칩을 미리 구성해 이를 기반으로 전체 디자인을 구성했다:  

    - 영어, 한글 모두 손글씨 느낌의 폰트를 사용했다:  

    - 메인 로고는 Majesty
    - 네비게이션 바를 비롯한 주요 영어 표기는 Permanent Marker
    - 한글 표기는 Yeon Sung 과 Nanum Gothic 사용

## References

프로젝트를 진행하며 참고했던 사이트들

- <https://docs.djangoproject.com/ko/3.1/intro/>
- <https://kr.vuejs.org/v2/guide/index.html>
- <https://router.vuejs.org/kr/>
- <https://pilot376.tistory.com/58>
- <https://fonts.google.com/>
- <https://studymeat.com/archives/532>
- <https://www.freecodecamp.org/news/how-to-use-the-position-property-in-css-to-align-elements-d8f49c403a26/>

## Credits :

### 본격

- 일단 저를 이끄느라 고생하신 **페어님**께 감사드립니다. 디자인으로도, 기능으로도, 분위기메이킹도, 모든 부분에서 저보다 뛰어나서 배울 점이 많았습니다. 프로젝트하는데 너무 고생해서 한편꼴론 죄송하기도 하네요...
- 둘째로 적극적인 피드백을 주신 **교수님**께 감사드립니다. 원래 저는 프로그래밍할 때 문제 해결은 구글님께서 해결해주시니 혼자 해보려고 했지만 페어님의 적극적인 질문과 교수님의 적극적인 답변으로 지금의 프로젝트가 만들어졌습니다.
- 세번째로 **서윤 3반** 사람들에게 감사드립니다. 서로 알려주고 배우면서 같이 성장하는 재미를 느꼈습니다.
- 마지막으로 그동안의 수업 자료를 공유해주신 **온라인 강사님**들께도 감사의 말씀 올립니다. 강사님들의 코드를 약간 변형시키며 이것저것 테스트해보며 프로젝트를 진행했습니다.

### 단원

- 누구보다도 가장 고생 많았던 **페어** 본혁님께 커다란 감사와 애정의 마음을 보냅니다. 저는 저만의 페이스가 확고한 사람인데, 저의 페이스에 잘 맞춰주시려고 고생 많으셨을 것을 알아요. 프로젝트 내에서도 원하는 기능을 적절히 구현해주셨던 것은 물론, 제가 이해하지 못하는 코드들이 있을 때는 잘 이해할 수 있도록 친절하게 설명해 주셨어요. 또 제 중심의 작업 공간이었는데도 불평 없이 어울려주셔서 다시 한 번 감사드립니다. 좋은 페어를 만나서 하고싶은 것을 잔뜩 할 수 있었어요!
- 한 학기 내내, 저의 개발자로서의 첫 여정을 함께 해주신 **ggu** 교수님께, 부끄러운 마음으로 감사 인사 드립니다. 늘 더 많은 것을 가르쳐주시려고 하셨는데, 다 배워가지 못해 아쉬움이 남습니다. 짧은 질문 하나에 질문자보다도 더 성의있는 답변을 돌려주시려는 마음에 늘 감사했습니다.