MOVIE TRAIN

NEW WEBPAGE PROJECT

새로운 Movie 추천 커뮤니티, 영화와 커뮤니티를 한 번에

SSAFY 서울 1반 8조
이성섭, 윤민영

MOVIETRAIN

Contents

Concept

Structure

Design

Review

MOVIETRAIN

Concept

컨셉

프로젝트명 : Movie Train

Target : 장시간 이동하며 영화를
감상할 사람들을 위한
영화추천 및
커뮤니티 서비스

파트분배 : 이성섭 (Backend, 기획)
윤민영 (Frontend, 디자인)

주요 기능

Login 회원가입 (아이디, 비밀번호, 이메일 정보수정)
회원탈퇴, 로그아웃,

영화검색 TMDB API 검색
Detail(제목, 평점, 설명, 포스터 이미지)

커뮤니티 게시글 소통(게시글 작성, 댓글)

프로필 기본정보(닉네임, 팔로워, 팔로잉 등)
활동 목록

MOVIETRAIN

Communication

아침: 목표 md 작성 후 공유,

점심: 회의 및 회의록 작성,

저녁: git upload 및 readme 공유

Movietrain_pjt / Day6_Front_dev /

↑ Top

Movietrain	Day6 upload	yesterday
img	Day6 upload	yesterday
readme.md	Day6 upload	yesterday

readme.md

[Movietrain] Day6_Frontend_YMY (0524)

오늘의 할일

1. DESIGN 관련 회의 (오후)

```
- [Layout] DetailView
- [Tone] LoginView / SignupView
```

2. Vue

```
1. 레이아웃(디자인)
    - MovieDetailView
      - ROW 1 : 해당 영화의 트레일러를 유튜브 API를 가져와서 재생
      - ROW 2 : 2개의 COLUMN으로 구성 (영화 포스터 / 정보 TEXT)
      - ROW 3 : 추천 영화 5개를 card-group 으로 구성

    - LoginView / SignupView
      - 컬러톤 변경, 배경 변경, 파스텔톤 그라데이션 적용

2. 라우팅 - 메인 화면에서 Movie Card 클릭시 MovieDetailView로 이동(/:id)
```

[Movietrain] Day2_Frontend_YMY (0518)

오늘의 할일

1. DESIGN Layout 초안 완성
   - (Main, Login, 프로필)
2. DESIGN Layout 관련 회의(오후)
3. 로고 / 슬로건 관련 회의(오후)
4. 컴포넌트 구성안 제작 및 수정
5. [Vue] 실제 컴포넌트 기초구조 구축

회의록

1. (시간 남으면) 이용가이드 페이지 만들기
2. (역시나 시간이 남으면) footer 구성 (얼단 구조상 컴포구축 필요)
3. detail page -> 게시글 댓글이 아닌 코멘트로 바꾸기
   - 별점 넣을 수 있는 시스템 구현 방법 찾기
   - search 완료 (PM 3:20)
     - https://mdbootstrap.com/docs/b4/vue/plugins/rating/
     - bootstrap 5라서 호환이 될지는 모르겠음
4. 로고 / 슬로건 관련 회의(오후)
   i. 여행이라는 컨셉에 맞게 좀더 green하고 라이트한 느낌의 로고 찾기

진척도

- ★★[Vue] Frontend 컴포넌트 기본구조 구축
  - [Fix] 레이아웃(Navbar, footer)이 필요한 컴포들과, 필요없는 단일 컴포들(ex- login) 분리가 필요하다고 판단,
    - 레이아웃이 포함된 DefaultLayout과, 라우터만 있는 EmptyLayout으로 쪼갠 후
    - 라우트 세팅에서 필요에 따라 페이지들을 분리해서 용도에 맞게 라우팅되도록 세팅했음
    - 즉, Navbar등의 레이아웃이 필요한 컴포넌트들은 DefaultLayout.vue를 통해 라우팅패스를 설정하고,
    - 단일 페이지만 보여줘도 되는 컴포넌트들은 EmptyLayout.vue를 통해 라우팅하도록 설정했음.

[Movietrain] Day6_Frontend_YMY (0524)

오늘의 할일

1. DESIGN 관련 회의 (오후)

```
- [Layout] DetailView
- [Tone] LoginView / SignupView
```

2. Vue

```
1. 레이아웃(디자인)
    - MovieDetailView
      - ROW 1 : 해당 영화의 트레일러를 유튜브 API를 가져와서 재생
      - ROW 2 : 2개의 COLUMN으로 구성 (영화 포스터 / 정보 TEXT)
      - ROW 3 : 추천 영화 5개를 card-group 으로 구성

    - LoginView / SignupView
      - 컬러톤 변경, 배경 변경, 파스텔톤 그라데이션 적용

2. 라우팅 - 메인 화면에서 Movie Card 클릭시 MovieDetailView로 이동(/:id)
```

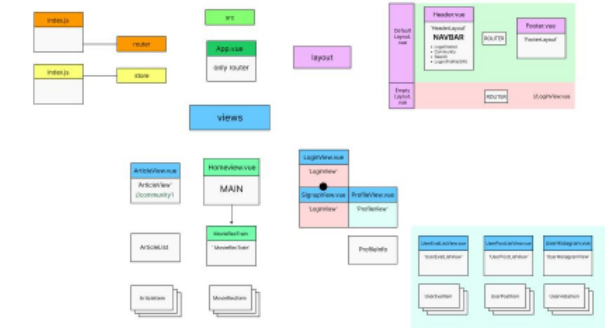
회의록

1. LoginView / SignUpView 톤 변경 - 로고색 반영
2. MovieDetailView 레이아웃 회의
3. Frontend + Backend 병합 작업 일정 : 05/25 (Backend Auth완성시)

진척도

★★[Vue] Frontend 진행상황

- 컴포넌트 구성 업데이트

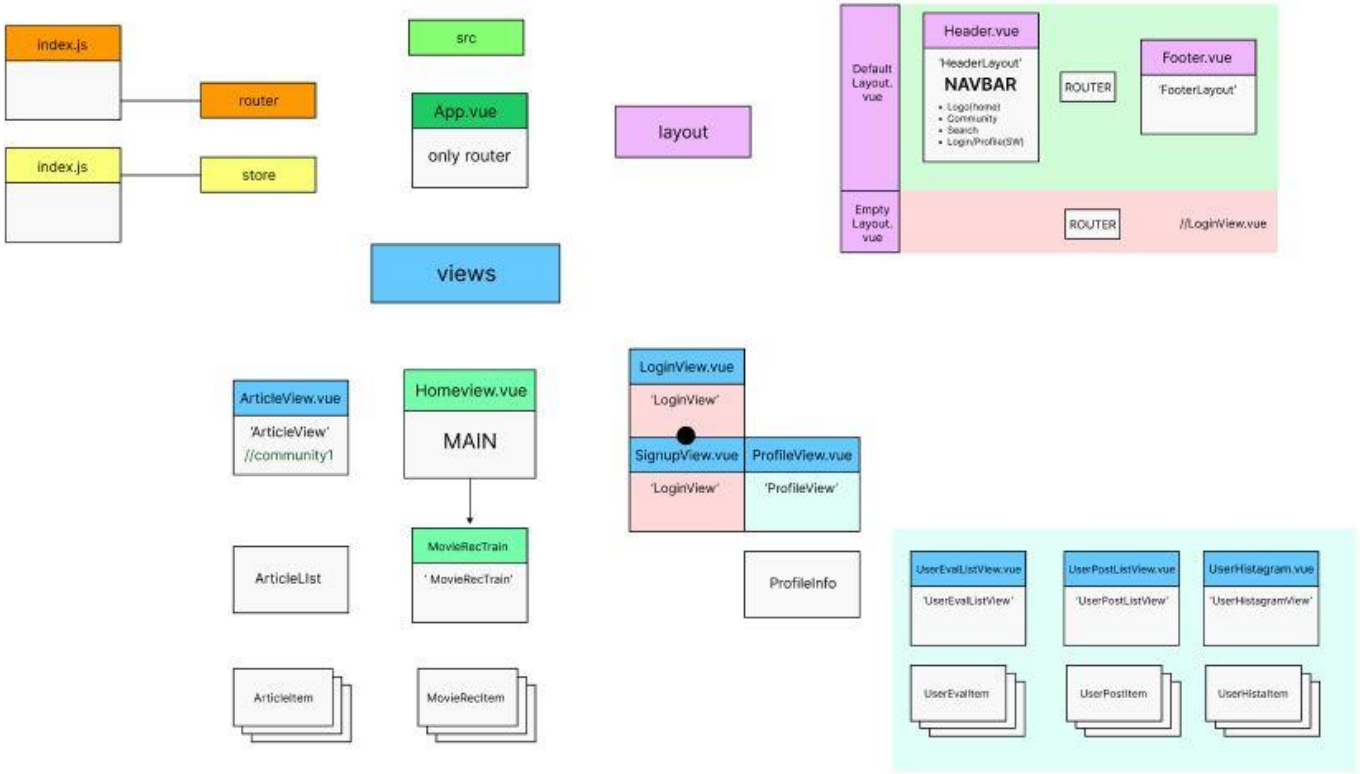


MovieDetailView

- xxl size, xl size

Structure

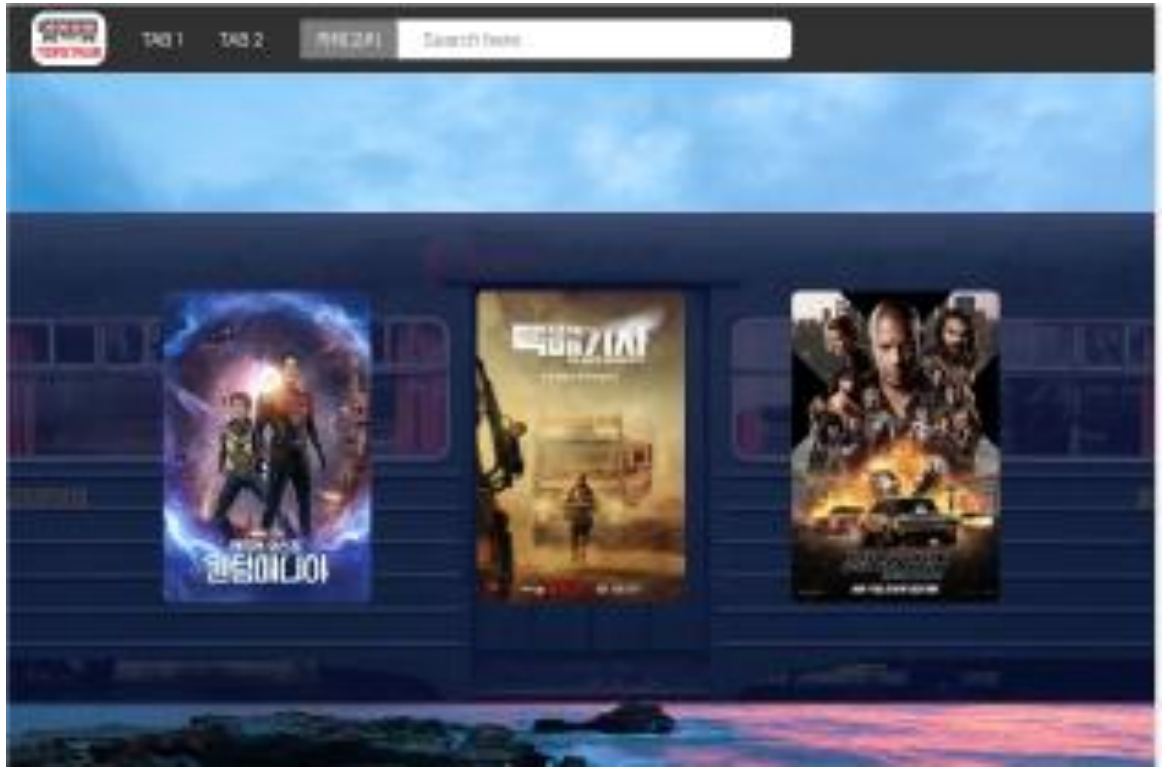
Components



ERD

Design

초기 main

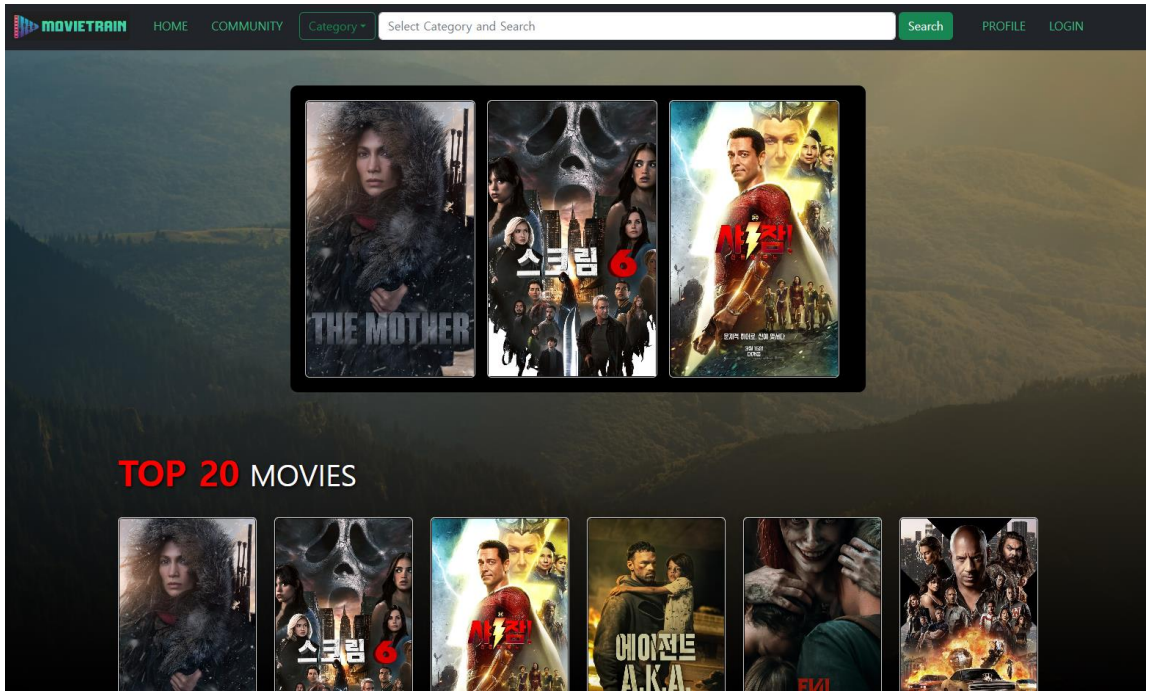


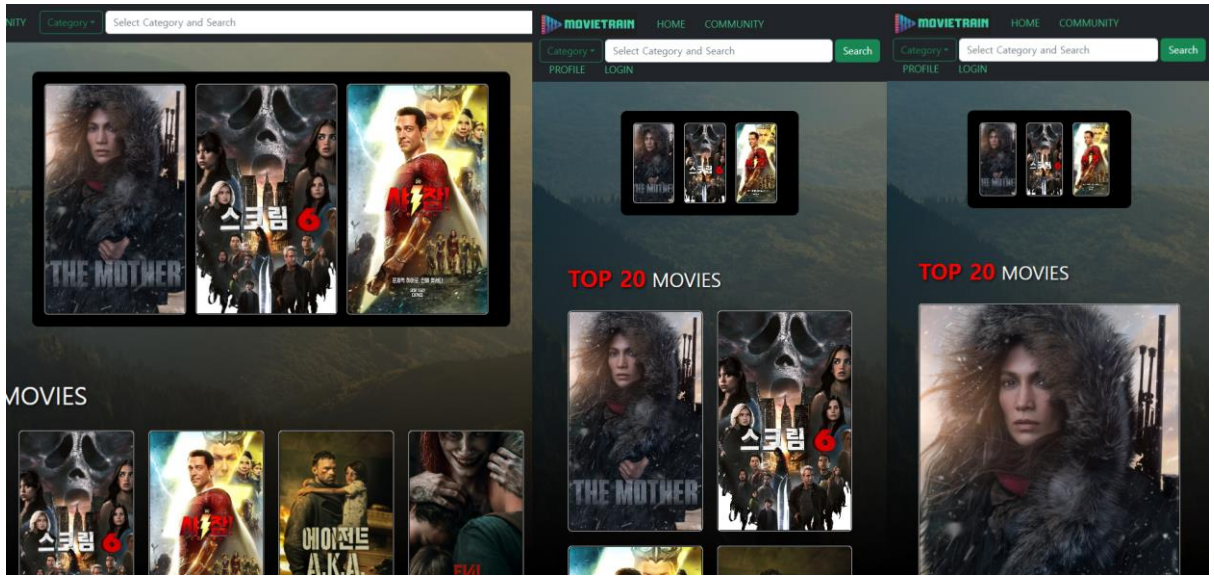
초기 profile

MOVIETRAIN

Design

최종 main



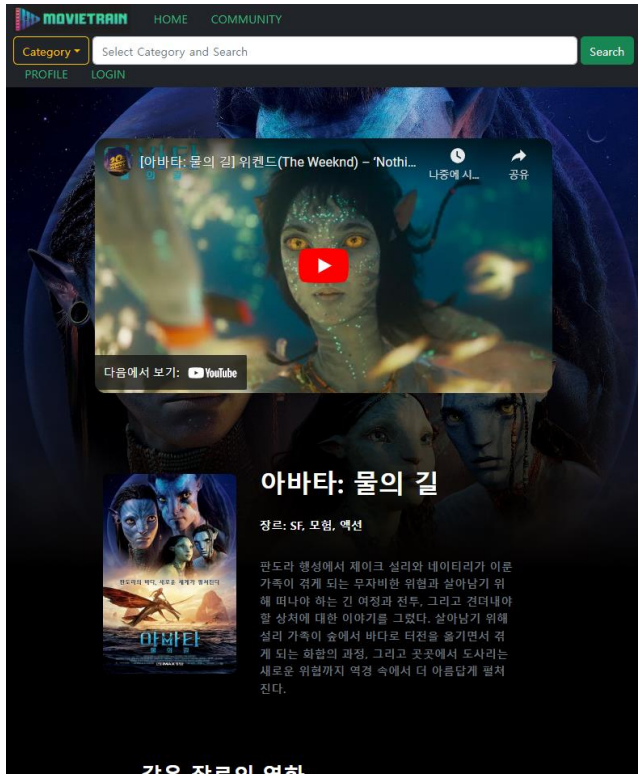


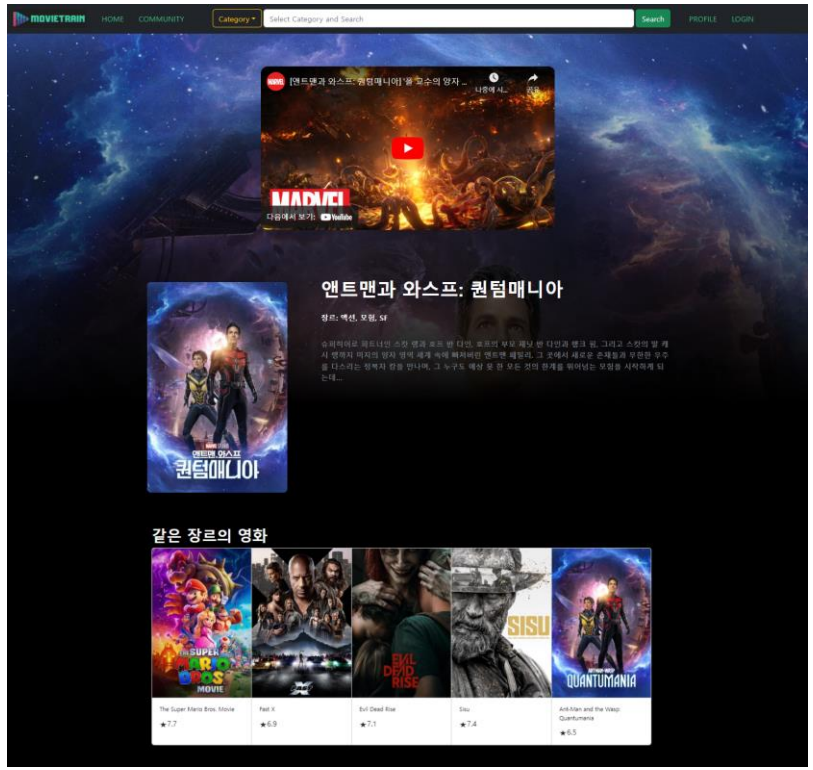
Display 크기에 따른 반응형 디자인 구현

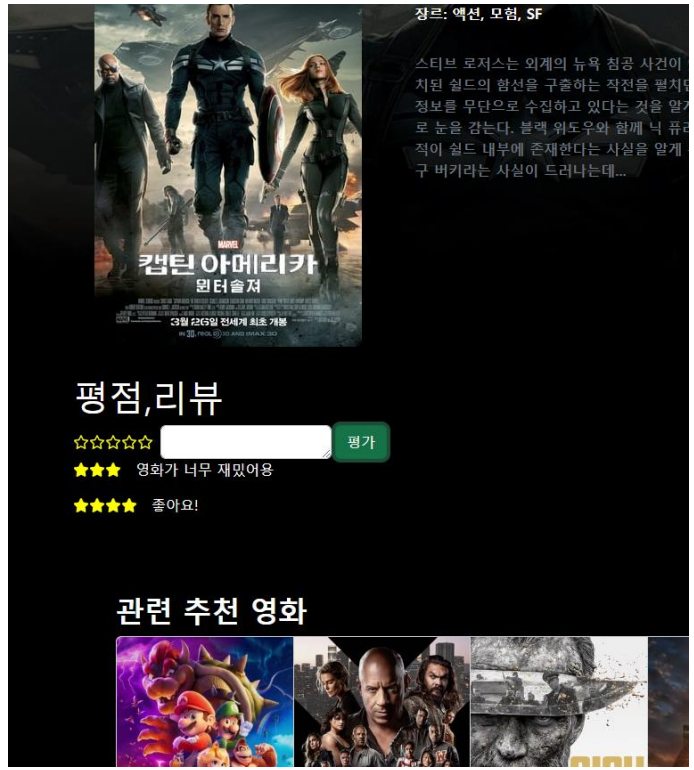
MOVIETRAIN

Design

최종 detail page



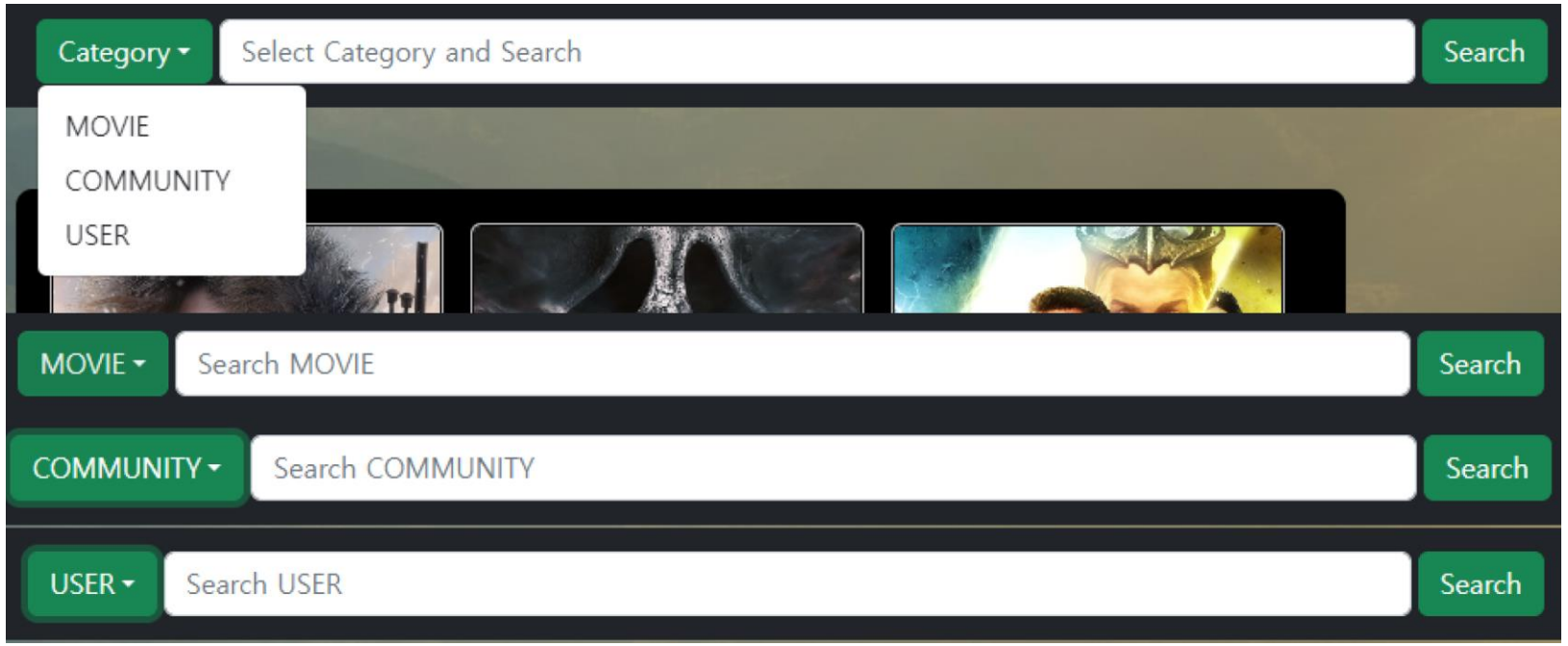




- Display 크기에 따른 반응형 디자인 구현
- 배경에 포스터 이미지 그라데이션 적용 / 평점 +한줄평 구현

Design

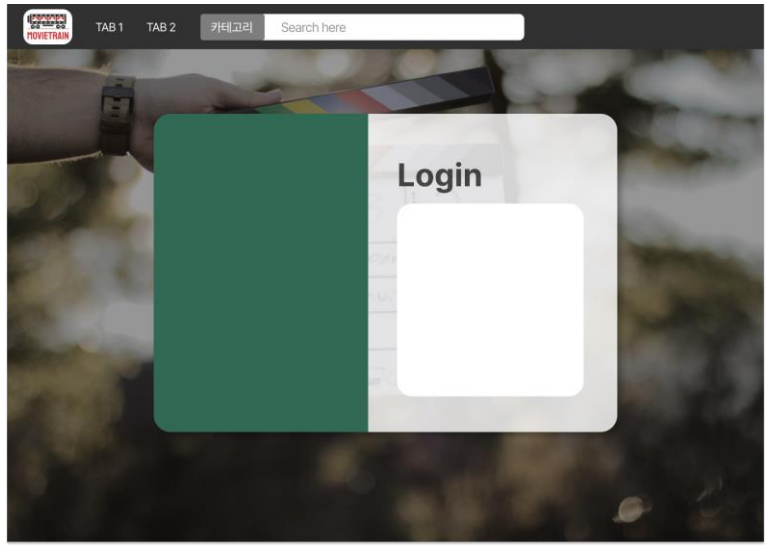
반응형 Navbar (Header)



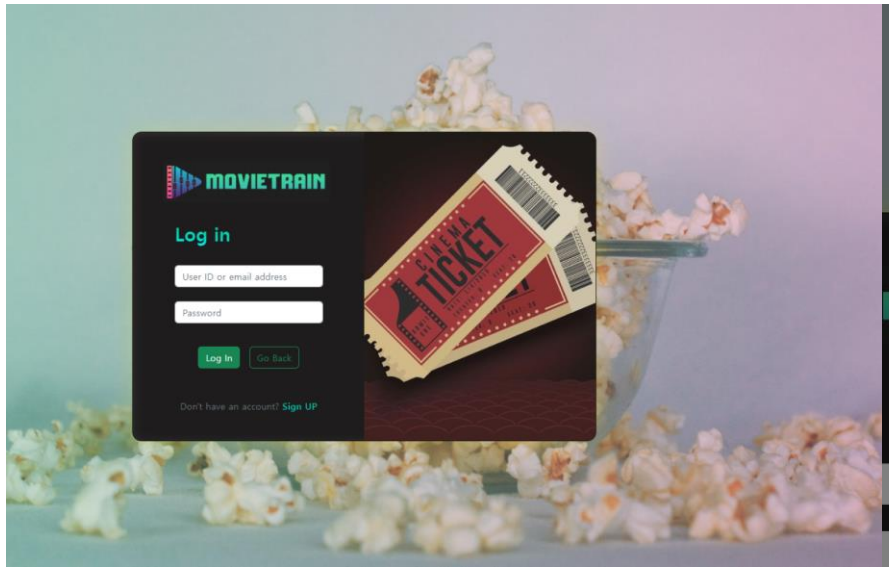
MOVIETRAIN

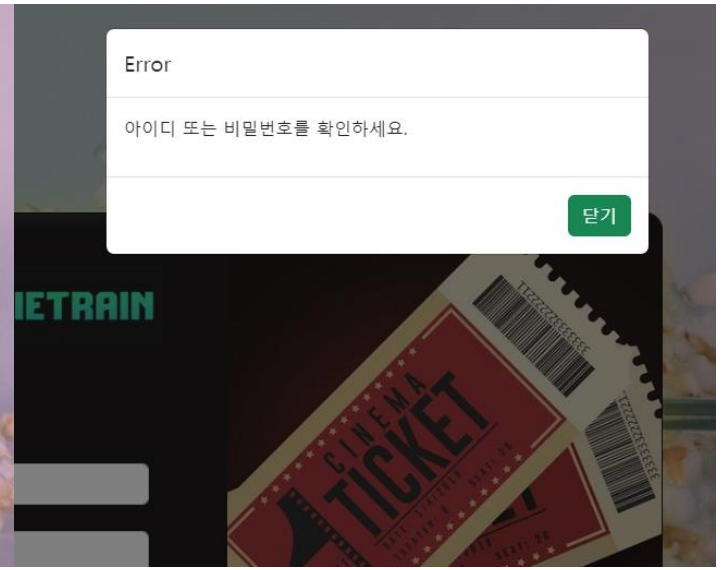
Design

초기 Login



최종 Login

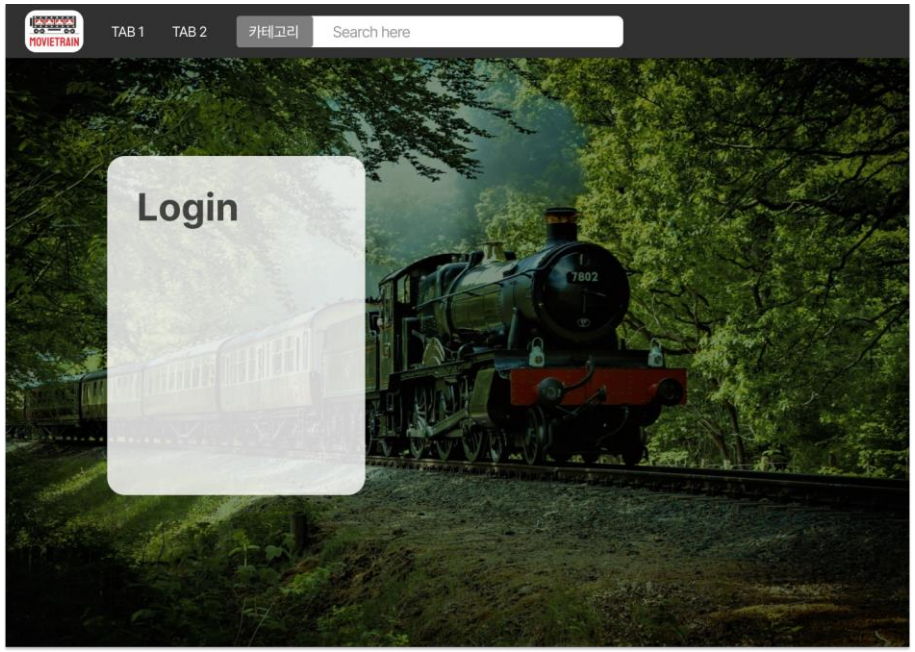




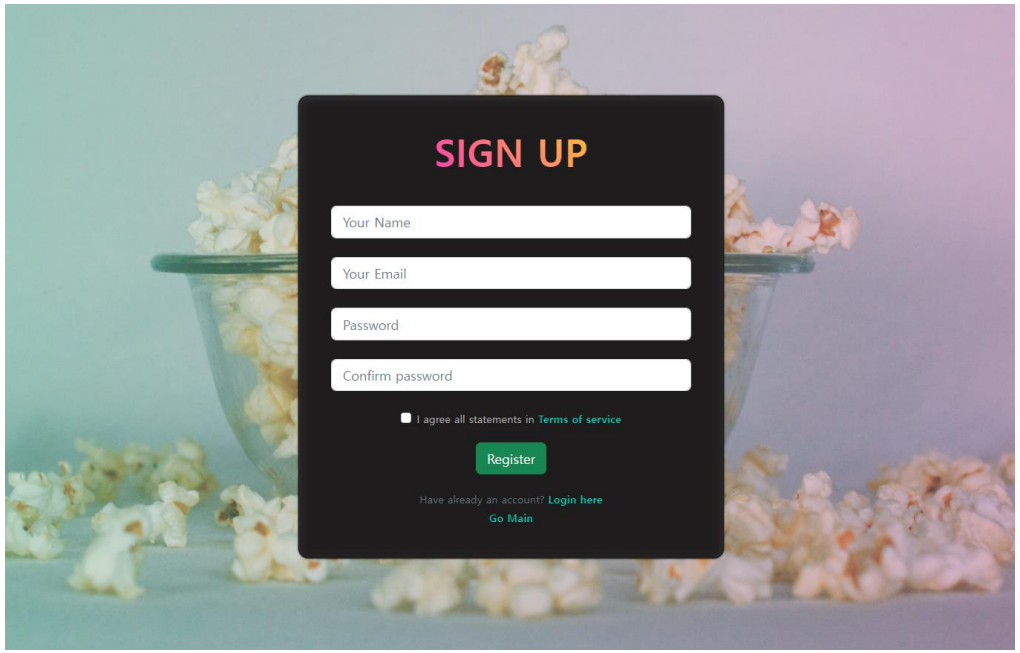
MOVIETRAIN

Design

초기 signup



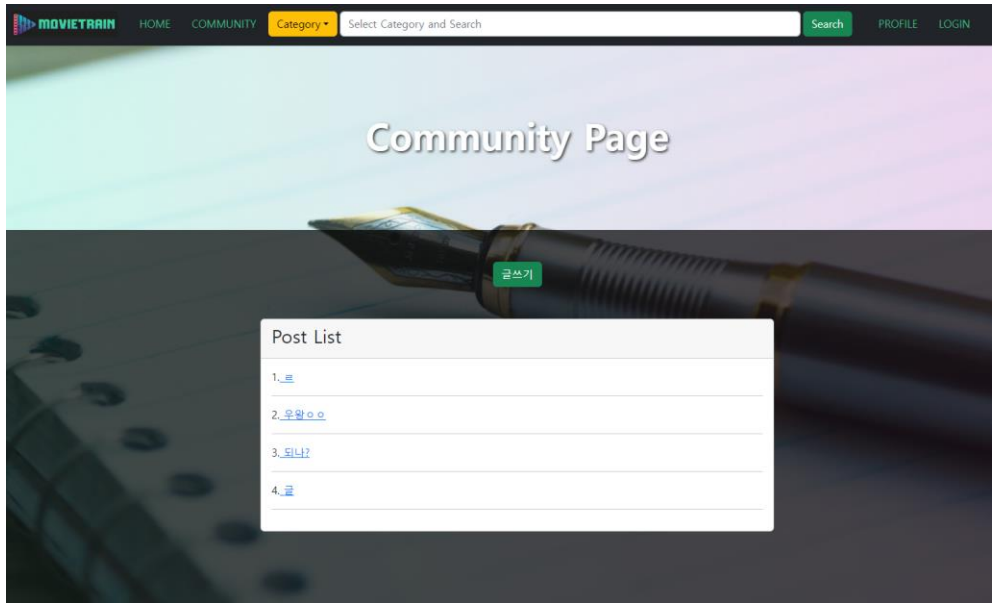
최종 signup



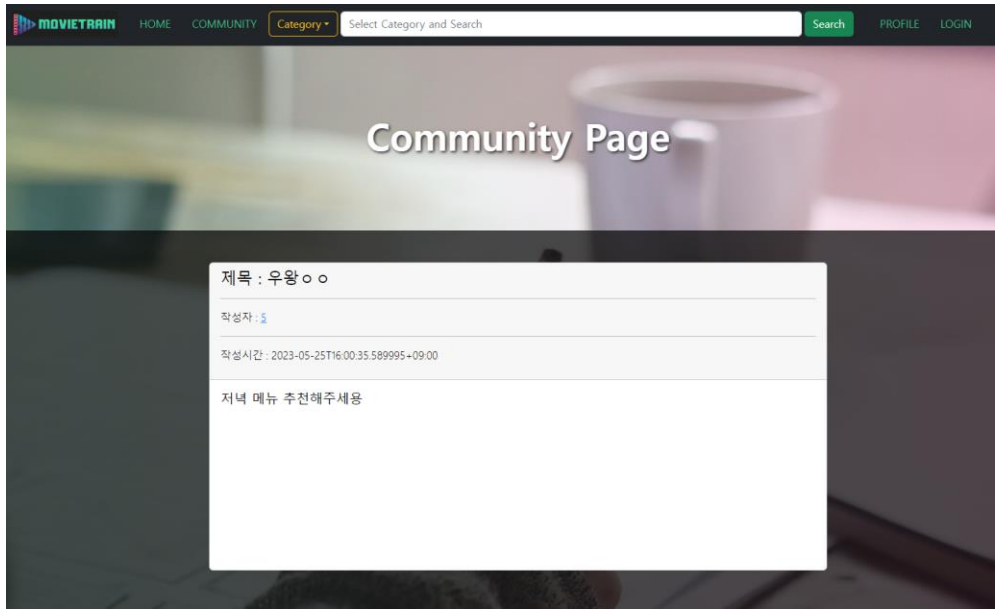
MOVIETRAIN

Design

ArticleList



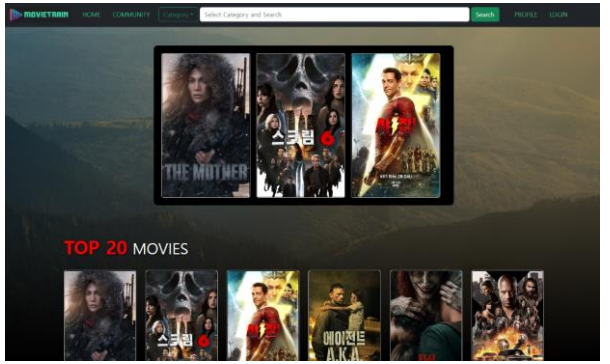
ArticleDetail



Design

필요에 따른 페이지 레이아웃 구현

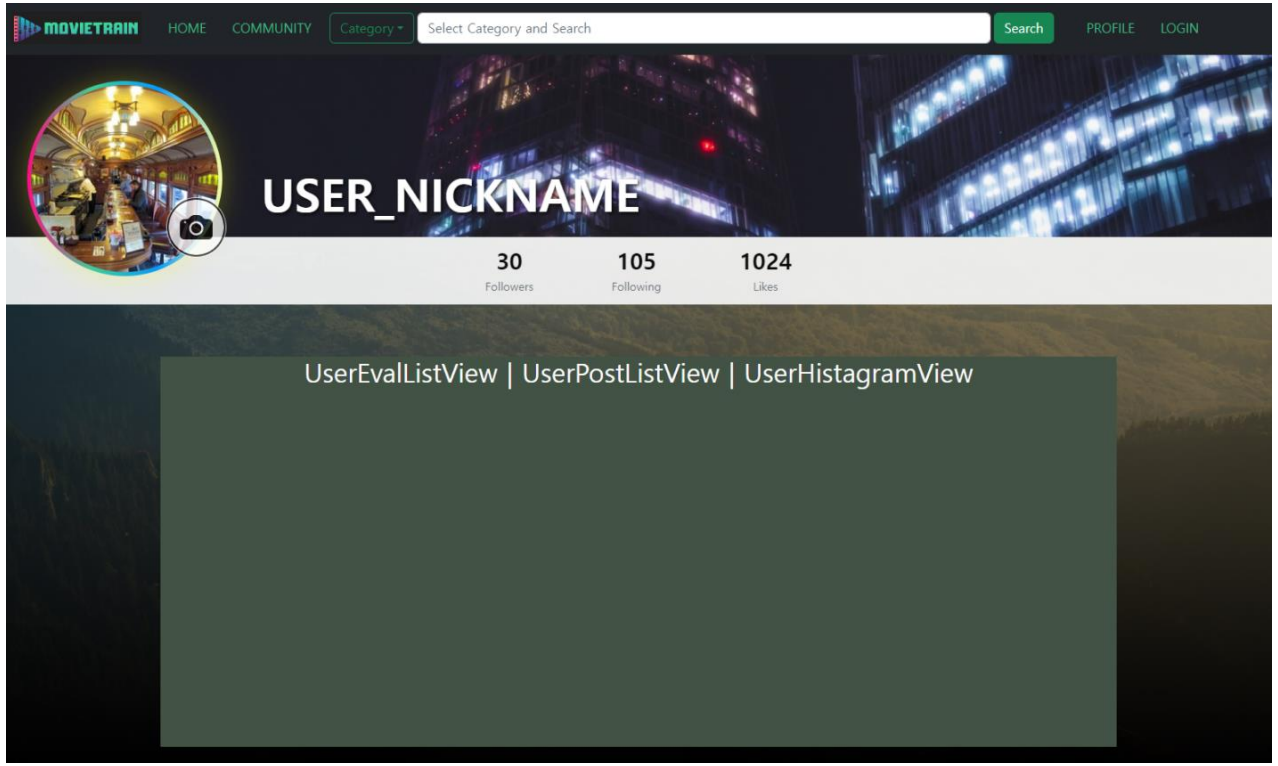
```
const router = new VueRouter({
  mode: 'history',
  routes : [
    {
      path:'/',
      name:'DefaultLayout',
      component: DefaultLayout,
      children:[
        {
          path: '/',
          name: 'Home',
          component: ()=>import('@/views/HomeView')
        },
        {
          path: '/profile',
          name: 'Profile',
          component:()=>import('@/views/ProfileView')
        }
      ]

    },
    {
      path:'/',
      name:'EmptyLayout',
      component:EmptyLayout,
      children:[
        {
          path: '/login',
          name: 'Login',
          component: ()=>import('@/views/LoginView')
        },
      ]

    }
```



Navbar 필요 여부를 따져 router-link로 component 연결 path 분할

Design

Profile page

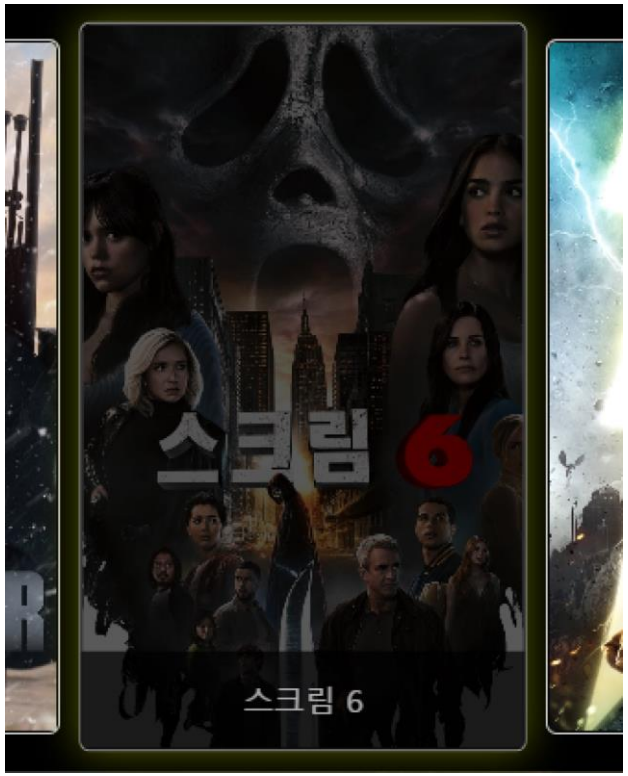


프로필 이미지를 바꿀 수 있도록 구현

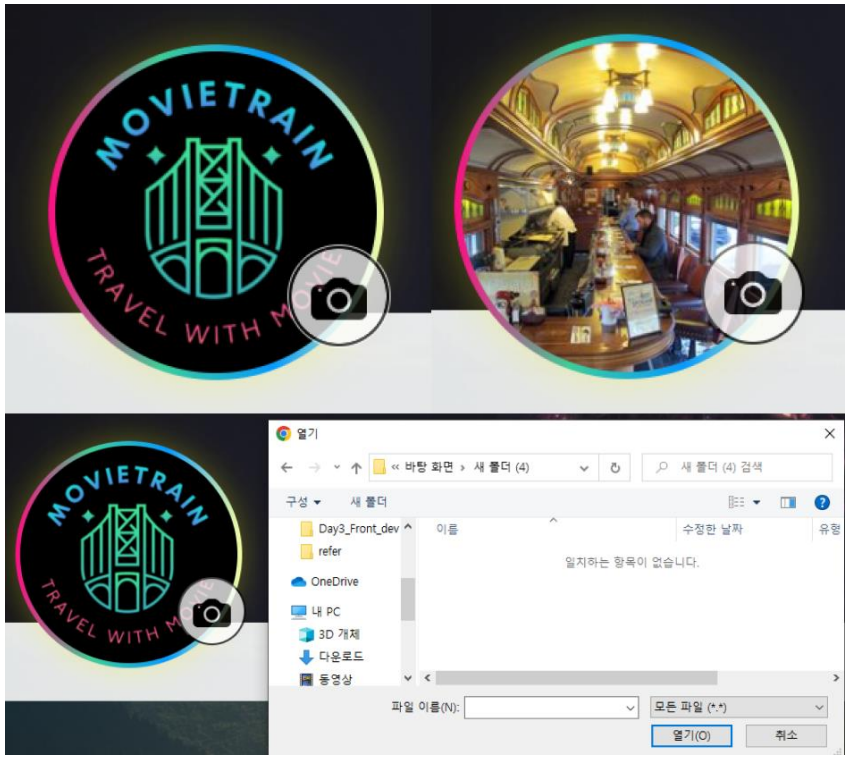
MOVIETRAIN

Design

기타 반응형 이펙트



마우스 Hover시
glow, ease in 효과



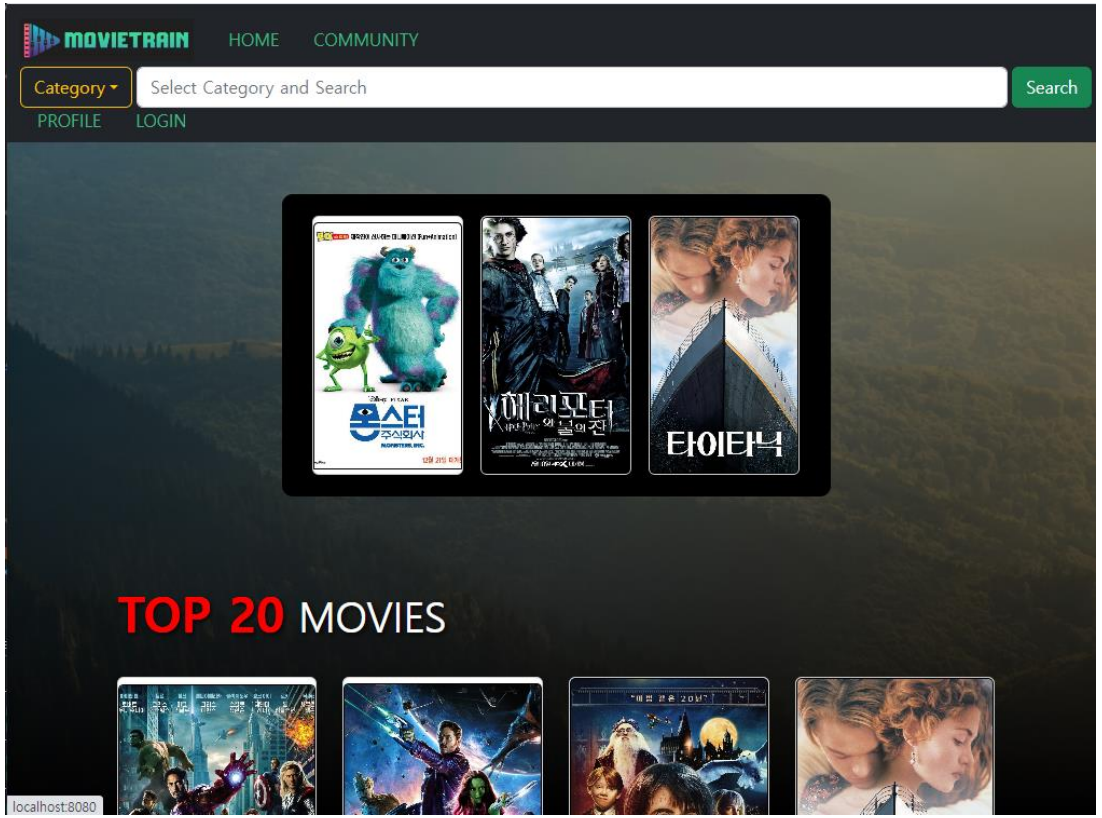
프로필 사진 변경

```
@keyframes shake {
  0% {
    transform: translate(0px, -2px);
  }
  50% {
    transform: translate(0px, 2px);
  }
  100% {
    transform: translate(0px, -2px);
  }
}

#Pbutton:hover .camera-icon {
  animation: shake 0.8s infinite;
  transform: translateY(-2px)
}

/* 클릭 시 hover 효과를 멈추고 ease-in-out 효과만 발생 */
#Pbutton:active .camera-icon,
#Pbutton.isClicked:hover .camera-icon {
  animation: none;
  transition: transform 0.1s ease-in-out;
  transform: scale(0.8);
}

</style>
```

Hover, Click 동시 구현
(icon)

Review

Page 시연



Scrum

어려웠던 점

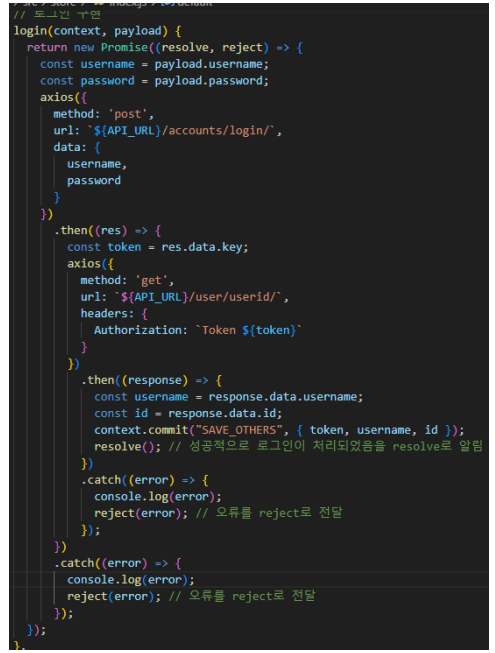
오류 오류가 오류를 낳는 사태…
알고 보니까 별 거 아닌 오류…

병합 Front와 Back을 병합시키는 데 많은 어려움
소통의 중요성 느낌

VUE VUE와 Bootstrap을 연동시,
미리 작성해 놓았던 CSS와
호환 및 혼합이 잘 안되어
원하는 대로 구현하는데 어려움을 느낌

CSS 이미지 + 그라데이션 등 다중 레이어를 쌓아가며 background를 작업할 때, ::before, ::after, 그리고 div간의 상하관계에 따라 css가 무시되어 z-index를 사용하는데 어려움을 느낌.

** 꿀팁 : 상하관계때문에 z-index가 적용이 안 될땐
position 세팅을 해서 상속이 아닌 고유의
z-index를 가지게 한다



```
// 로그인 구현
login(context, payload) {
  return new Promise((resolve, reject) => {
    const username = payload.username;
    const password = payload.password;
    axios({
      method: 'post',
      url: `${API_URL}/accounts/login/`,
      data: {
        username,
        password
      }
    })
      .then((res) => {
        const token = res.data.key;
        axios({
          method: 'get',
          url: `${API_URL}/user/userid/`,
          headers: {
            Authorization: `Token ${token}`
          }
        })
          .then((response) => {
            const username = response.data.username;
            const id = response.data.id;
            context.commit("SAVE_OTHERS", { token, username, id });
            resolve(); // 성공적으로 로그인이 처리되었음을 resolve로 알림
          })
          .catch((error) => {
            console.log(error);
            reject(error); // 오류를 reject로 전달
          });
      })
      .catch((error) => {
        console.log(error);
        reject(error); // 오류를 reject로 전달
      });
  });
},
```

Store, Compo 특정 컴포넌트에서 store.dispatch로 store의 action을 정의했을 때, Action에서 발생한 응답(정상/error)을 원래의 컴포넌트로 넘기는 방법을 알기 위해 꽤나 고생함.
Present, resolve와 reject 이용.

감사합니다.