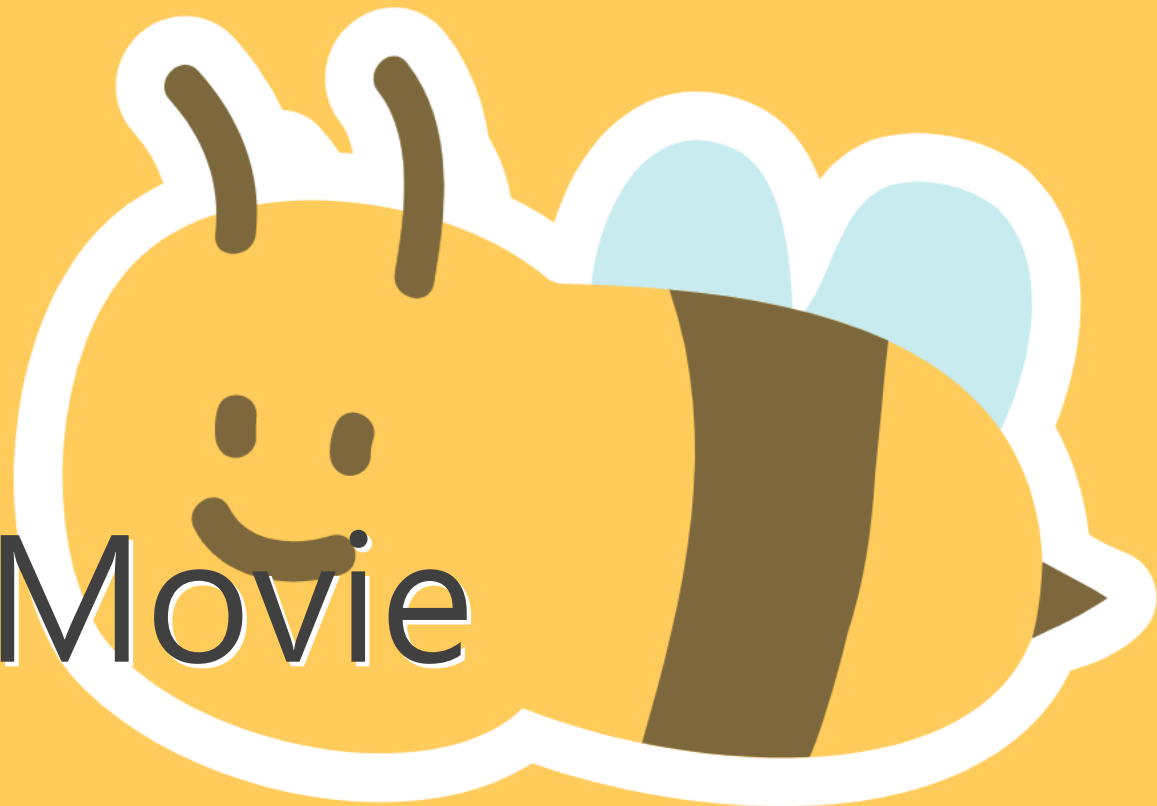


Beesy Movie



바쁜 벌꿀은 영화 볼 시간도 없다

목차

- Intro
- 데이터 전처리 및 기초 작업
- 핵심기능
 - Movie Recommendation
 - Movie Shorts
- 핵심 코드
 - Movie Shorts
- 부가 코드
 - SEO

INTRO



INTRO

바쁘다 바빠 현대사회

배달만 시켜놔. 영화는 골라줄게!



INTRO

영화 찾기 귀찮고

볼 시간도 없는

현대인들을 위한

Beesy의 영화 추천 서비스



데이터 전처리 및 기초 작업

The logo for Movielens, featuring the word "movielens" in a white, lowercase, sans-serif font on an orange rectangular background.

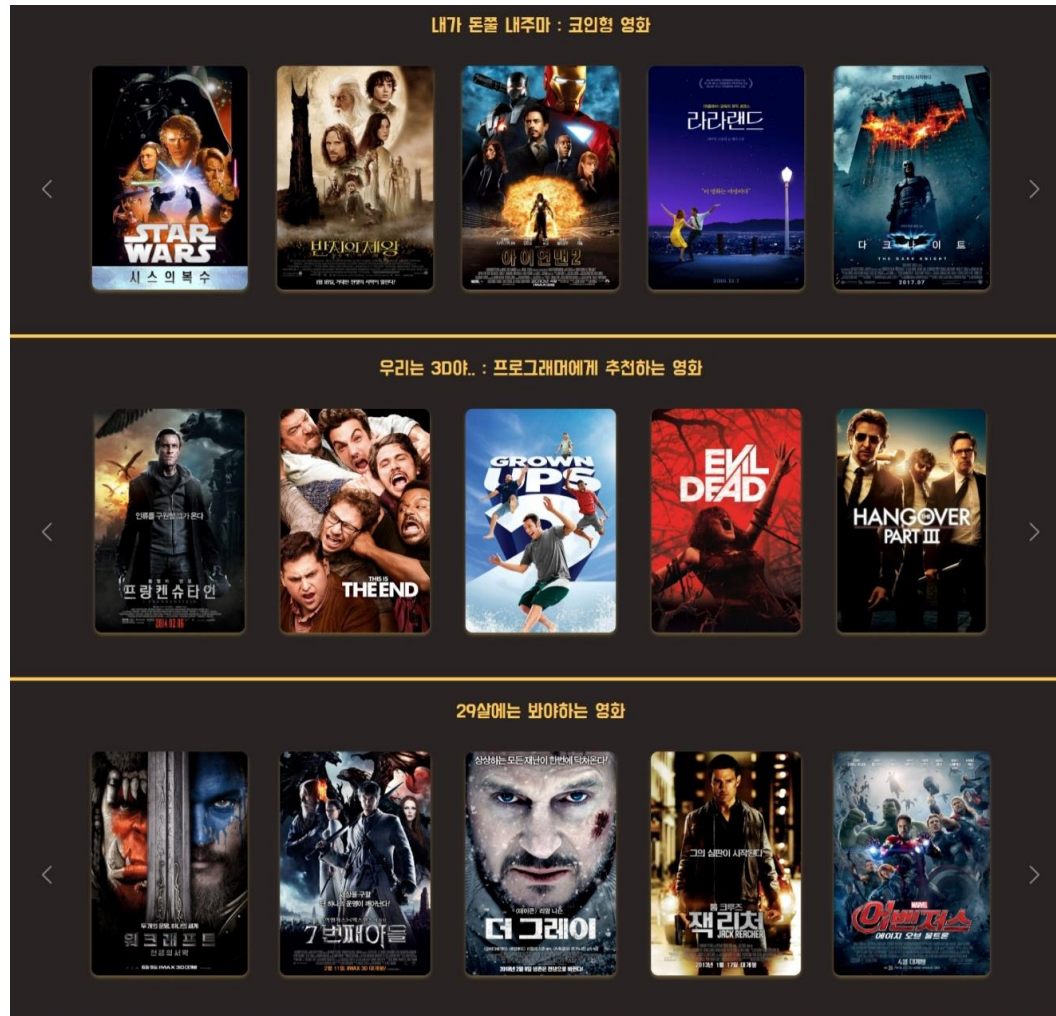
4,000 개의 영화 데이터
6,000 개의 고객 데이터
성별, 직업, 연령 별 평점
100만 개 데이터

YouTube API 활용해서

바쁜 현대인을 위해
영화 예고편 및 스포일러 영상 제공

The YouTube logo, consisting of the word "You" in black and "Tube" in white inside a red rounded rectangle.

핵심기능 : Movie Recommendation



Movie Lens - 1M Rate Data

TMDB API - SIMILAR

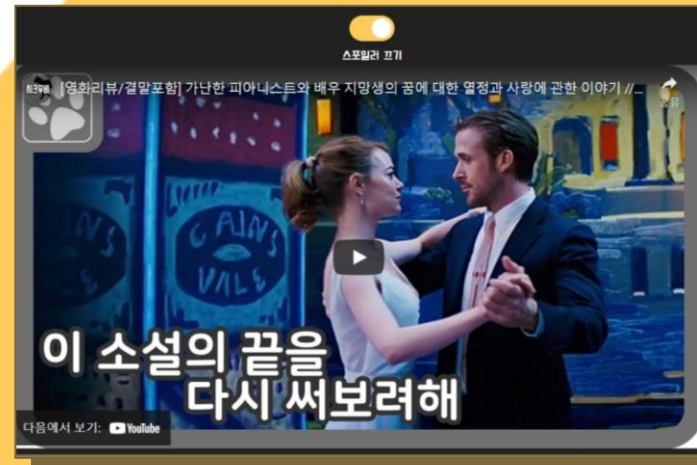
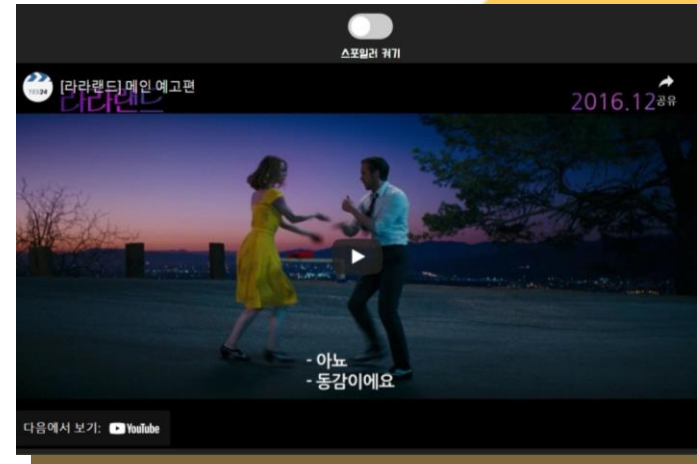
코인형 영화

직업 · 나이 · 성별 영화 추천



핵심기능 : Movie Shorts

Beesy가 골라준 영화를
클릭 한 번으로
예고편부터 정주행까지



핵심코드 : Movie Shorts

```
import { mapActions, mapGetters } from 'vuex'
import ReviewList from '@components/ReviewList.vue'
import VideoItem from '@components/VideoItem.vue'
import axios from 'axios'

const API_KEY = '--비밀--'
const API_URL = 'https://www.googleapis.com/youtube/v3/search'

export default {
  name: 'DetailView',
  data() {
    return {
      spoiler: false,
      videos1: [],
      selectedVideo1: null,
      videos2: [],
      selectedVideo2: null,
    }
  },
  components: {
    VideoItem
  },
  methods: {
    ...mapActions(['setDetailMovie']),
    spoiler_change() {
      this.spoiler = !this.spoiler
    },
    loadVideo1() {
      let params = {
        key: API_KEY,
        part: 'snippet',
        type: 'video',
        q: this.detailMovie.title + ' 예고편'
      }
      axios({
        method: 'GET',
        url: API_URL,
        params,
      })
        .then(res => {
          this.videos1 = res.data.items
          this.selectedVideo1 = this.videos1[0]
        })
        .catch(err => {
          console.log(err)
        })
    },
  },

```

```
    loadVideo2() {
      let params = {
        key: API_KEY,
        part: 'snippet',
        type: 'video',
        q: this.detailMovie.title + '영화 리뷰'
      }
      axios({
        method: 'GET',
        url: API_URL,
        params,
      })
        .then(res => {
          this.videos2 = res.data.items
          this.selectedVideo2 = this.videos2[0]
        })
        .catch(err => {
          console.log(err)
        })
    },
  },
  computed: {
    ...mapGetters(['detailMovie']),
    inlineStyle() {
      return {
        backgroundImage: `url(${this.detailMovie.backdrop_path})`
      }
    },
  },
  watch: {
    detailMovie: function() {
      this.loadVideo1()
      this.loadVideo2()
    },
  },
  created() {
    this.setDetailMovie(this.$route.query.id)
  },
}
```



핵심코드 : Movie Shorts

```
<div class="row" style="width : 100%; margin:0;">
  <div class="col-4" >
    <div class="img-box">
      
    </div>
  </div>
  <div class="col-8 content row align-items-center second">
    <div class="second-box">
      <div class="custom-control custom-switch">
        <label class="switch">
          <input type="checkbox" @click="spoiler_change()">
          <span class="slider-detail round"></span>
        </label>
        <div class="switch-info" v-if="!spoiler">스포일러 켜기</div>
        <div class="switch-info" v-if="spoiler">스포일러 끄기</div>
      </div>
      <div v-if="spoiler">
        <video-item :selected-video="selectedVideo2"></video-item>
      </div>
      <div v-if="!spoiler">
        <video-item :selected-video="selectedVideo1"></video-item>
      </div>
    </div>
  </div>
</div>
```

Check박스에 v-on:click을
적용한 후

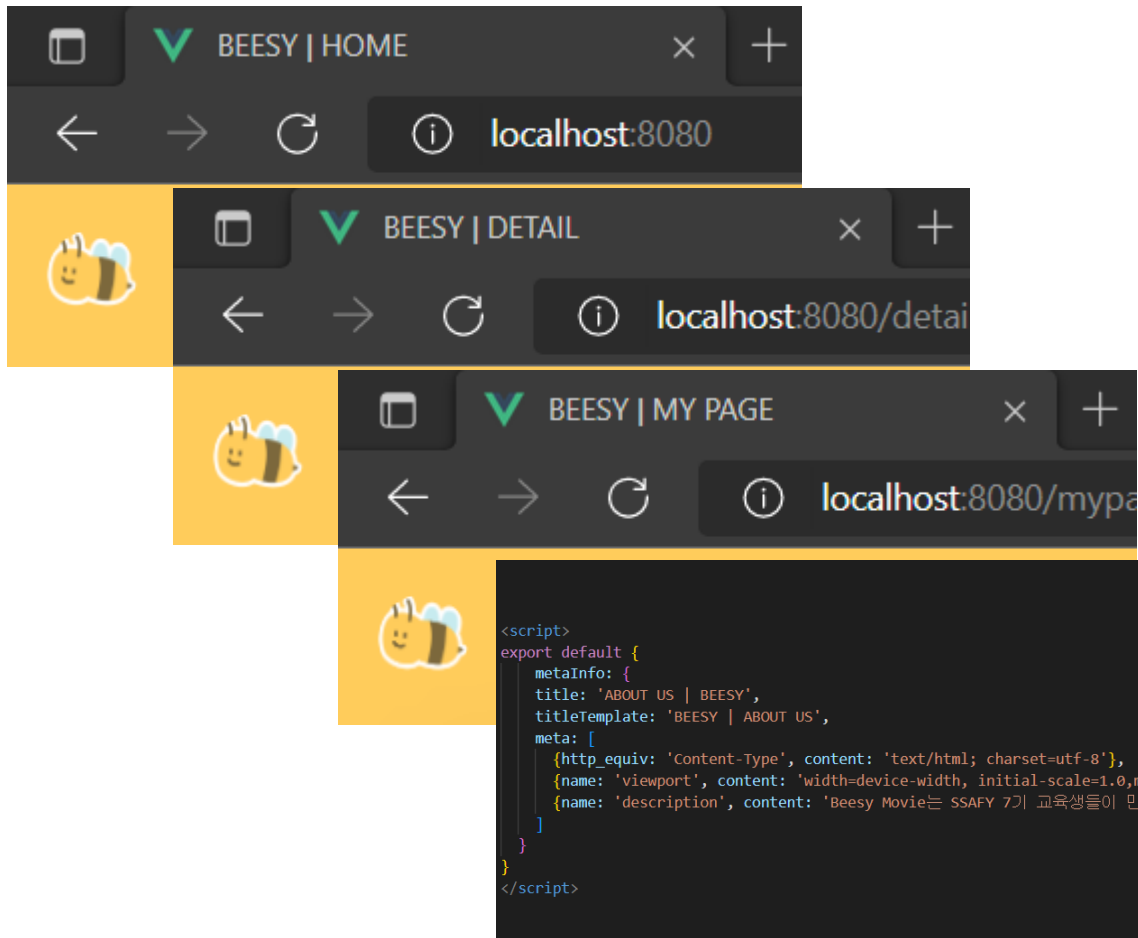
클릭으로 data의 spoiler의
참/거짓 값 변경하는

spoiler_change() 메서드 실행

이후 값에 따라 보여주는 영상
렌더링



부가코드 : SEO



vue-meta



Beesy 개발자들



이름 : 전재권
Front : .watch master
Back : allauth expert

한줄평 : "실력은 코드가 아니라 소통에서 나온다."



이름 : 문요성
Front : CSS 블루칼라
Back : json 파일 전처리 노가다

한줄평 : "응 실력은 코드야."



감사합니다!

홈페이지 시연으로 이어집니다.

