

React Project

API 비동기 데이터를 SPA로 관리하는 플랫폼

개요

1. 프로젝트 소개
2. 개발 도구 및 기술
3. 기능 소개
4. 문제 해결

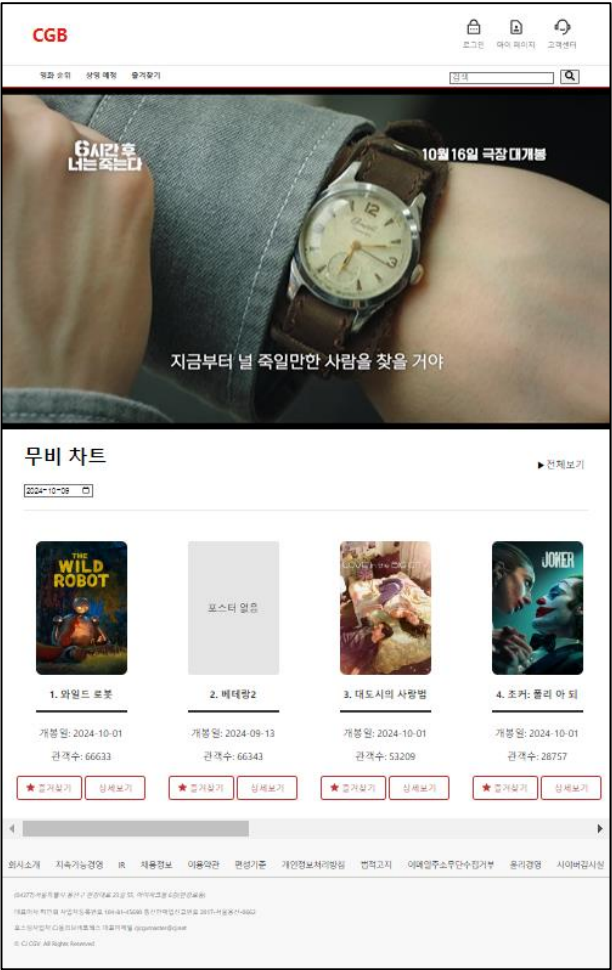
팀원

- 이종모 : harun107@naver.com
- 이경민 : cnrgk388@naver.com
- 황수림 : akdlgosvhs@naver.com
- 신승훈 : syh2612@naver.com

프로젝트 Github 링크 :

https://github.com/autumn-trip/React_Project

Project 소개



CGB Main Page

팀원	4명
프로젝트 기간	2024.09.30 ~ 10.11 (12일)
제목	CGB
성향	커뮤니티
내용	영화 순위, 상영 예정작 등 영화 정보 공유, 댓글 기능을 통하여 유저간 의견을 공유하는 커뮤니티
목적	- 영화 커뮤니티로 유명한 "무코" 활성화 빈약. - 대중적으로 활용하는 CGV / 롯데시네마 등 사이트에 커뮤니티 기능을 추가하여 유저들간의 활발한 소통 유도 - 온라인으로 편리한 소통을 통해 영화 업계쪽의 활력 증진 기대
주요 기능	로그인, 영화 일간/주간 순위 매일 업데이트, 즐거찾기, 영화 리뷰, 고객센터

Project WBS

[illegible]

개발 도구 및 기술



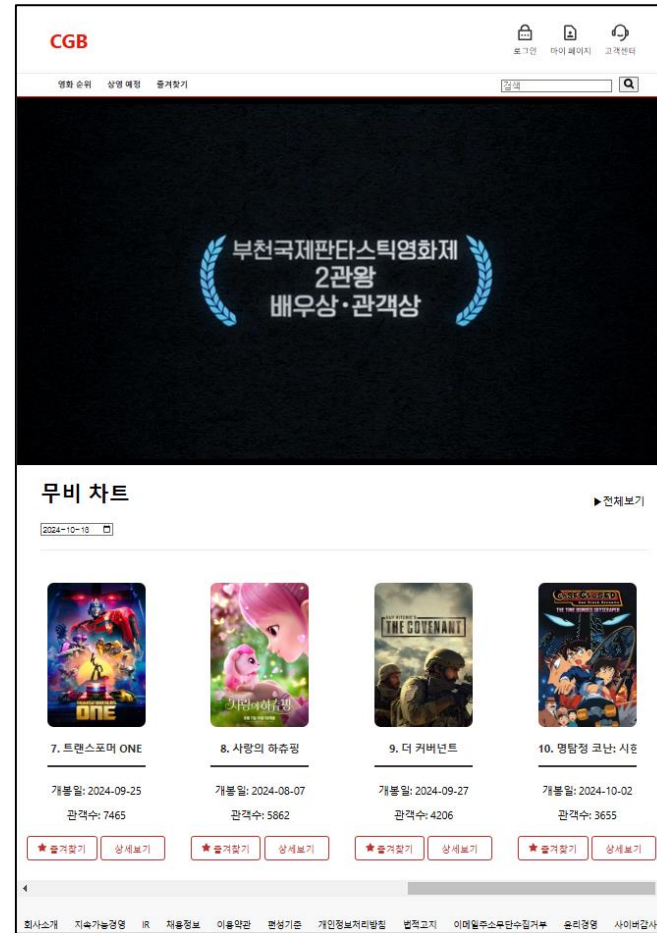
Tool	Visual Studio Code
Language	HTML5, CSS3, Javascript, npm, React
Library	<ul style="list-style-type: none">• react-icons• react-router-dom• react-virtualized• styled-component<ul style="list-style-type: none">• axios• Prettier
Hook	useState, useEffect, useContext, useReducer, useCallback, useMemo, useRef
API	영화진흥위원회, TMDB

개발 도구 및 기술 : BrowserRouter를 활용한 싱글페이지 구성

React

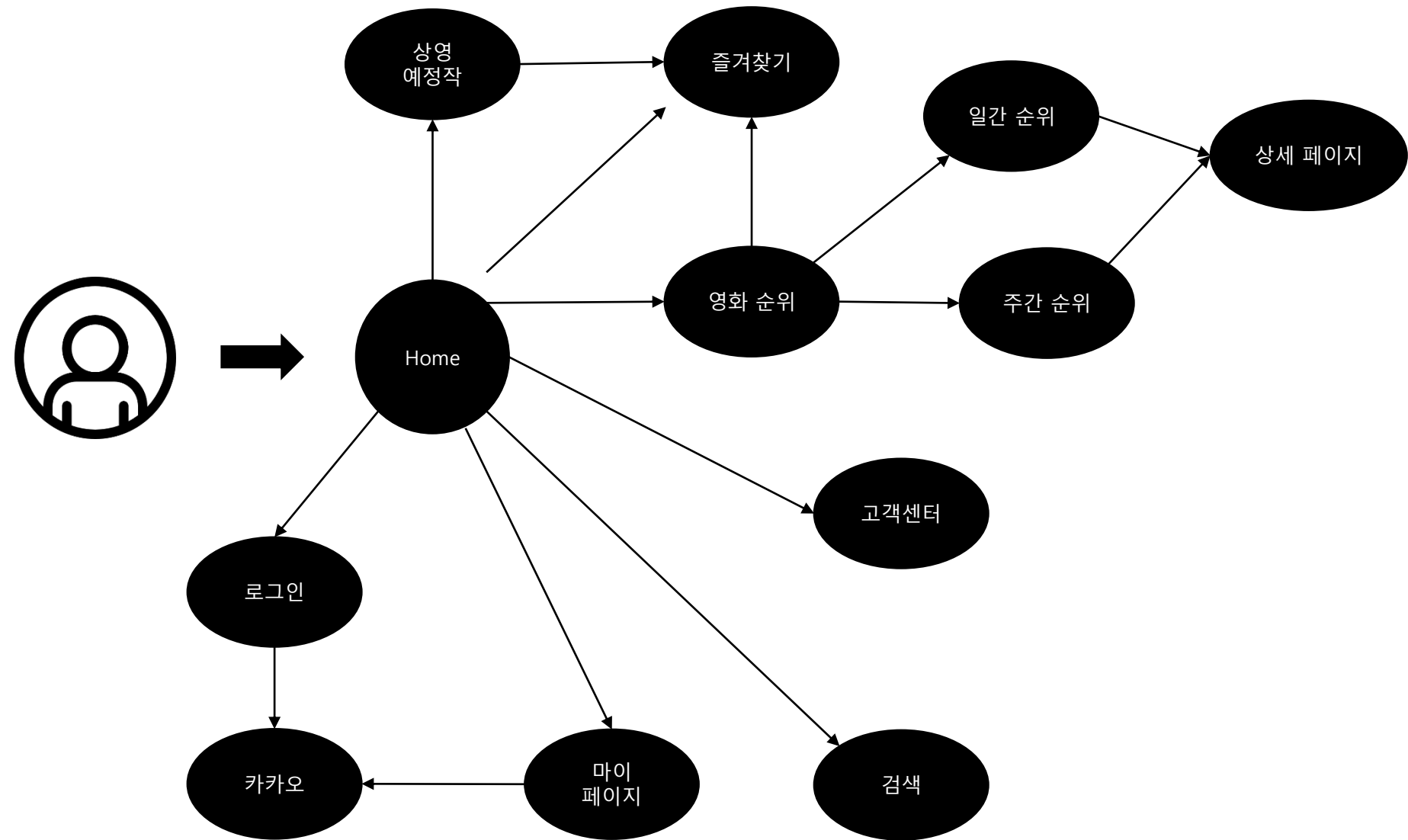
```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>
);
```

```
const App = () => {
  return (
    <FavoritesProvider>
      <Routes>
        <Route element={<Layout />} />
        <Route path="/" element={<Home />} />
        <Route path="/newmovie" element={<NewMovie />} />
        <Route path="/bookmark" element={<FavoriteMovie />} />
        <Route path="/service" element={<CustomerService />} />
        <Route path="/movie" element={<MovieList />} />
        <Route path="/weekly" element={<WeekMovie />} />
        <Route path="/login" element={<Login />} />
        <Route path="/mypage" element={<MyPage />} />
        <Route path="/search" element={<SearchResults />} />
        <Route path="/movie/:movieCd" element={<MovieDetail />} />
      </Routes>
    </FavoritesProvider>
  );
};
```



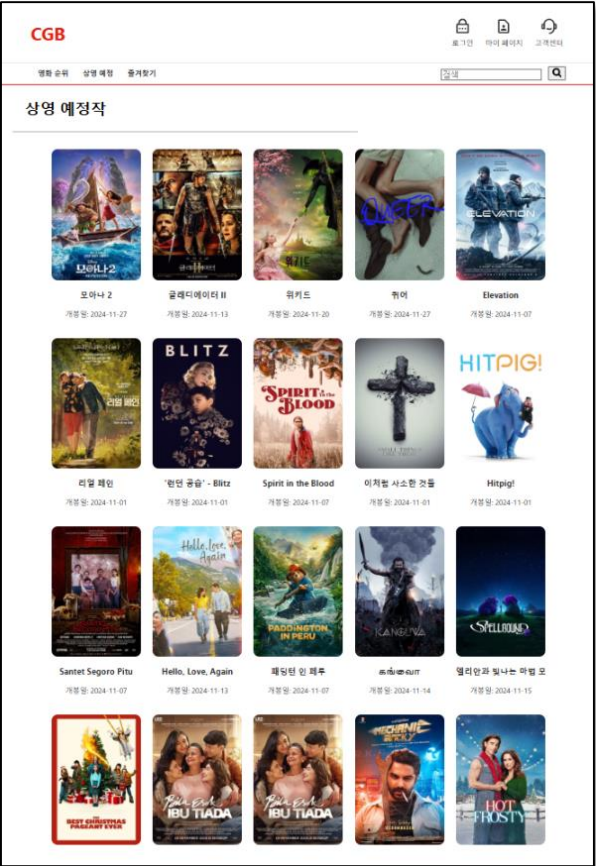
BrowserRouter를 활용한 싱글페이지 구성

기능 구조

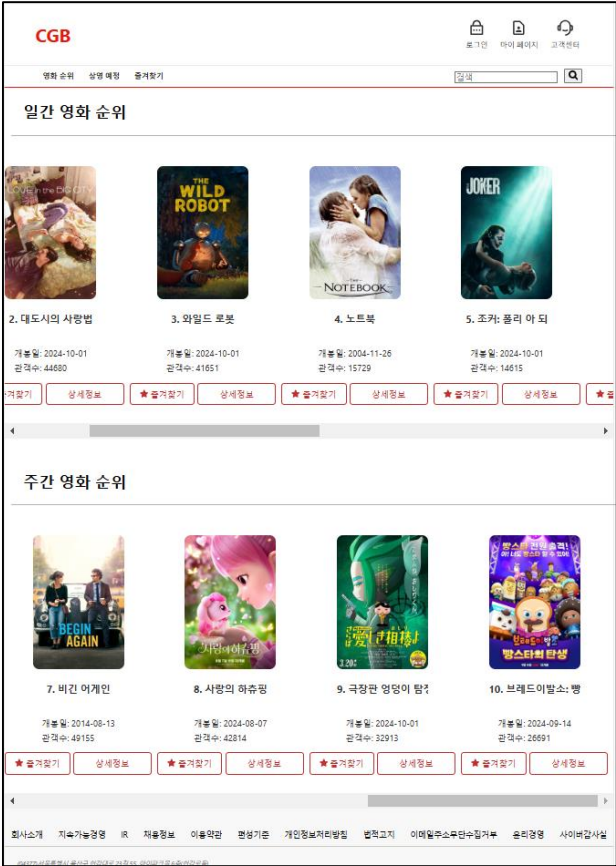


기능 소개 : 상영 예정작 & 일간/주간 영화 순위 & 상세페이지

- 사용자가 매일/매주 업데이트된 영화 순위와 정보를 확인할 수 있는 기능
- 영화 클릭 시 상세 페이지로 넘어가며 상세 정보 확인 가능



상영 예정작



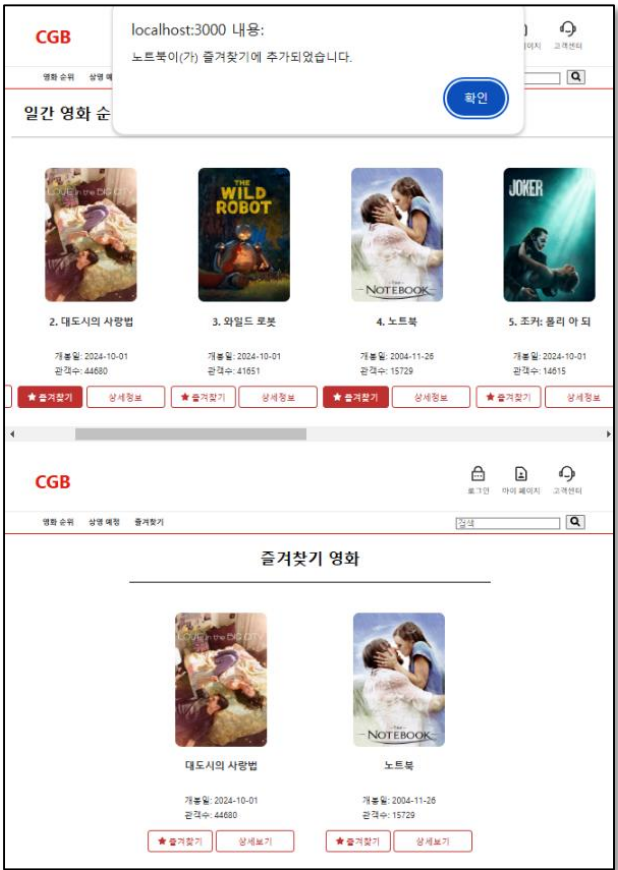
일간/주간 영화 순위



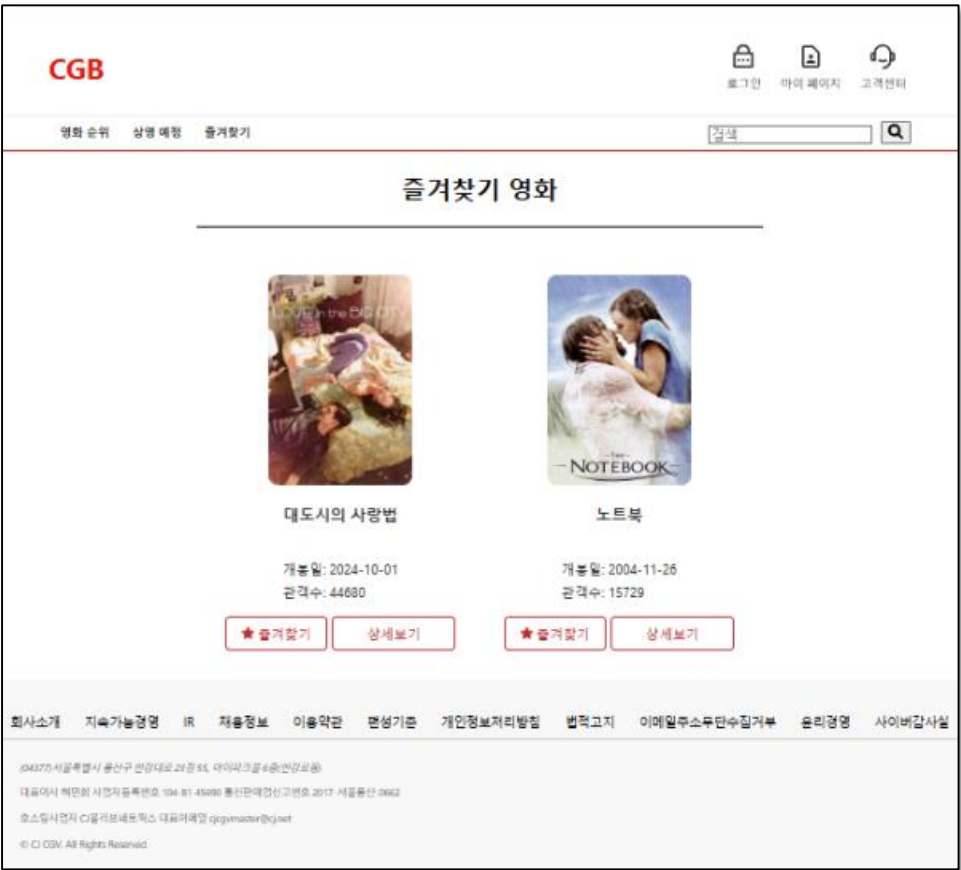
상세 페이지

기능 소개 : 즐겨찾기

- 사용자가 좋아하는 영화를 즐겨찾기로 저장하고 관리하는 기능



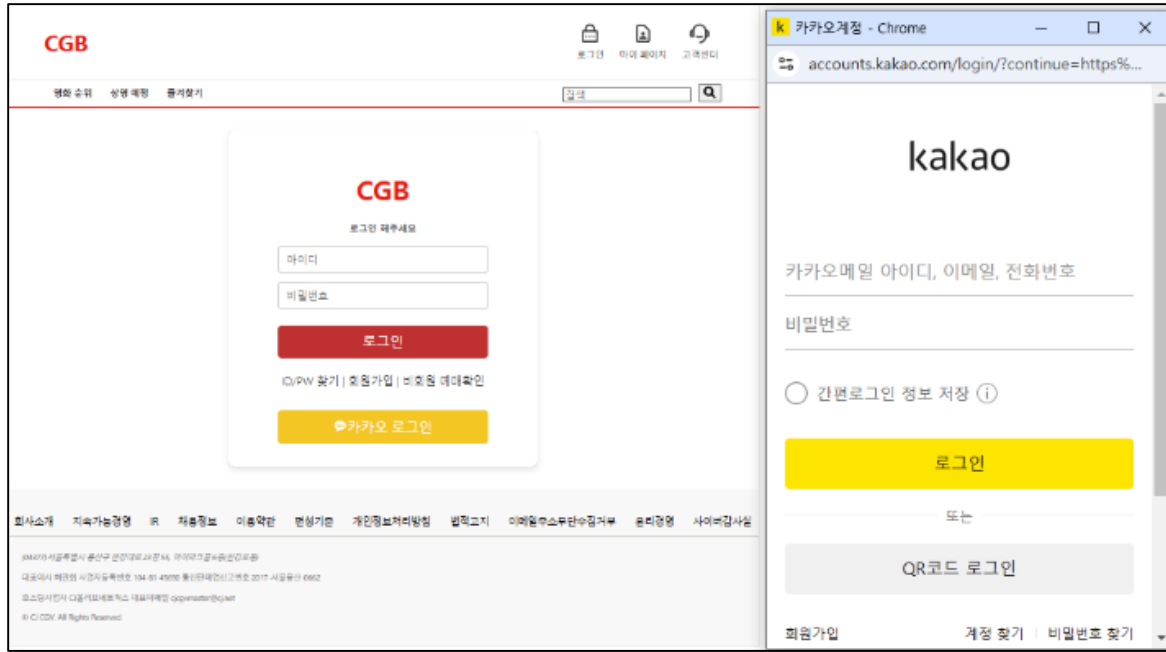
즐거찾기 추가



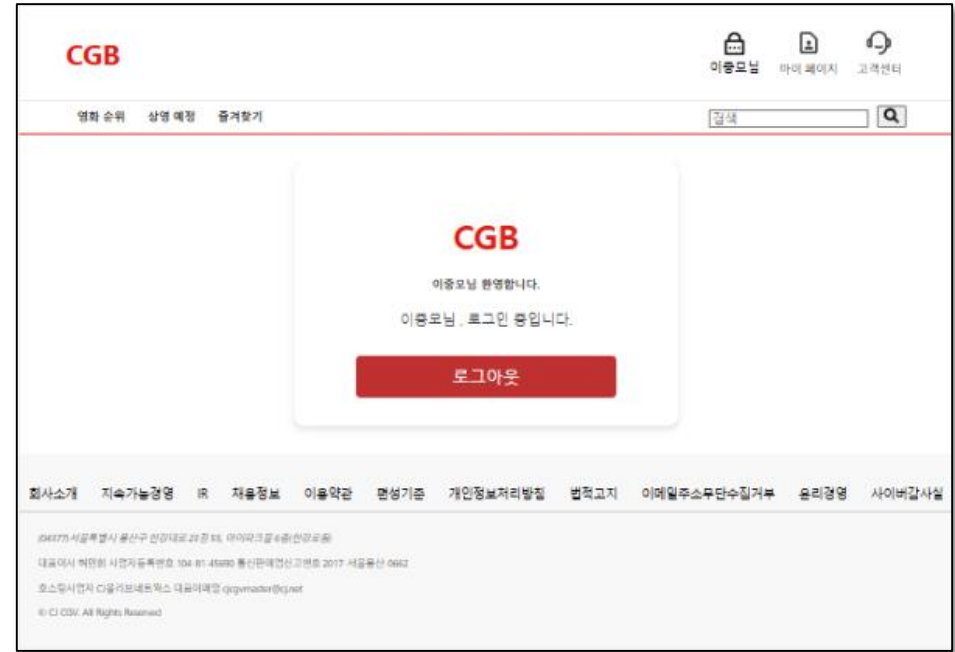
즐거찾기 페이지

기능 소개 : 로그인 / 로그아웃

- 카카오 로그인 API 연동을 통한 로그인 페이지 구현



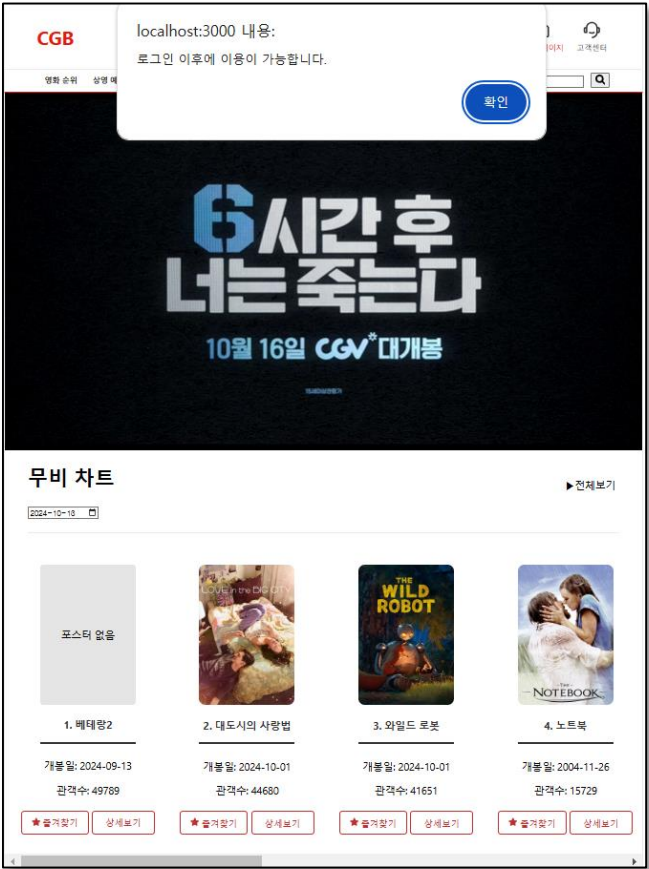
로그인 페이지 + 카카오 로그인



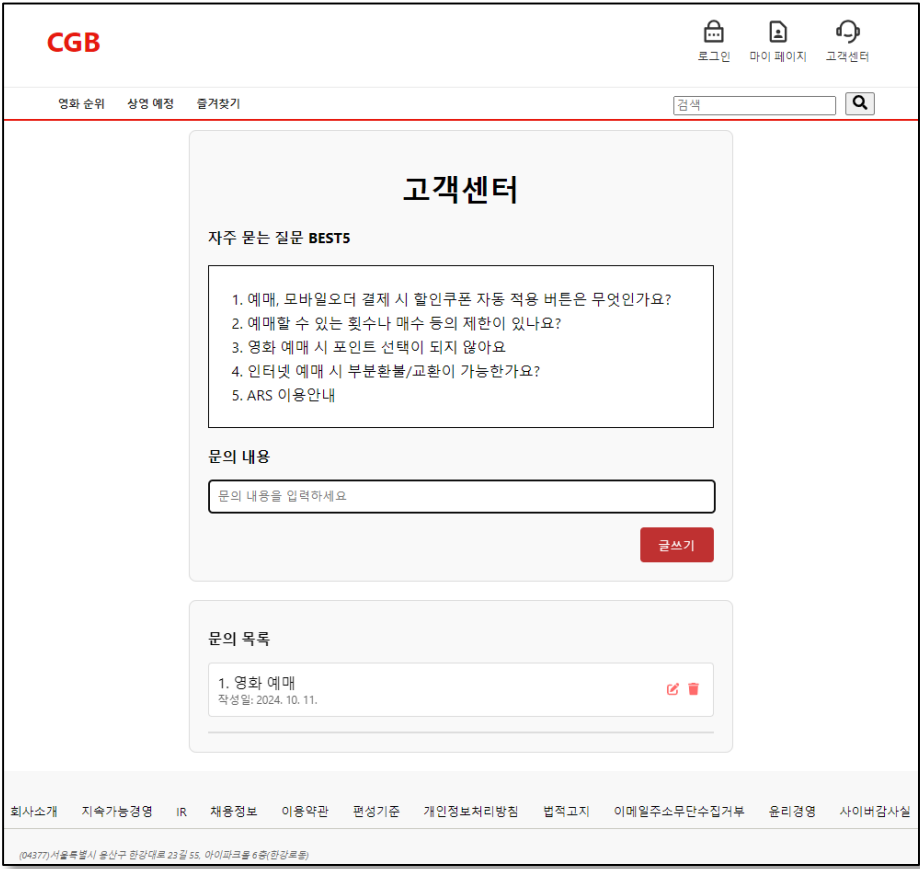
로그인 후

기능 소개 : 마이페이지 / 고객센터

- 마이페이지 클릭 시 로그인 화면으로 이동 기능
- 고객센터 페이지를 통한 문의 등록, 삭제, 수정 가능



마이페이지



고객센터

문제 해결_1

Error

```
const MovieList = () => {
  const [movies, setMovies] = useState([]);
  const [weeklyMovies, setWeeklyMovies] = useState([]);
  const [loading, setLoading] = useState(true);
  const { addFavorite, isFavorite } = useFavorites();
  const API_KEY_KOFIC = '303756f624abaa433a616a1f01bf62c2'; // KOFIC API 키
  const date = '20241009'; // 일간 영화 순위를 가져올 날짜

  useEffect(() => {
    const fetchMovies = async () => {
      try {
        const response = await axios.get(
          `http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.json?key=${API_KEY_KOFIC}&targetDt=${date}`
        );
        const movieList = response.data.boxOfficeResult.dailyBoxOfficeList;
      } catch {
        // Error handling
      }
    };
    fetchMovies();
  });
}
```

// KOFIC API를 통한 최신 영화 불러오기

시도:

-> 이미지 불러오기 실패

-> 크롤링을 통한 이미지 값 불러오기 실패

Complete

```
const MovieList = () => {
  const [movies, setMovies] = useState([]);
  const [weeklyMovies, setWeeklyMovies] = useState([]);
  const [loading, setLoading] = useState(true);
  const { addFavorite, isFavorite } = useFavorites();
  const API_KEY_KOFIC = '303756f624abaa433a616a1f01bf62c2'; // KOFIC API 키
  const API_KEY_TMDB = 'c46c5e36b91e0c3b8e75ebe785d68935'; // TMDB API 키
  const date = '20241009'; // 일간 영화 순위를 가져올 날짜
  const weeklyDate = '20241006'; // 주간 영화 순위를 가져올 날짜

  useEffect(() => {
    const fetchMovies = async () => {
      try {
        const response = await axios.get(
          `http://www.kobis.or.kr/kobisopenapi/webservice/rest/boxoffice/searchDailyBoxOfficeList.json?key=${API_KEY_KOFIC}&targetDt=${date}`
        );
        const movieList = response.data.boxOfficeResult.dailyBoxOfficeList;

        const moviesWithPosters = await Promise.all(
          movieList.map(async (movie) => {
            const tmdbResponse = await axios.get(
              `https://api.themoviedb.org/3/search/movie?api_key=${API_KEY_TMDB}&query=${encodeURIComponent(
                movie.movieNm
              )}`
            );
            const posterPath =
              tmdbResponse.data.results.length > 0
                ? tmdbResponse.data.results[0].poster_path
                : null;
          })
        );

        setMovies(movieList);
        setWeeklyMovies(moviesWithPosters);
        setLoading(false);
      } catch {
        // Error handling
      }
    };
    fetchMovies();
  });
}
```

TMDB API 이미지 값 확인

-> KOFIC API Key 값과 TMDB API Key 값 통일화

-> Key 값 호출 시 두 개의 API 불러오기를 통한 문제 해결

문제 해결_2

```
const handleFavoriteClick = (movie) => {  
  if (isFavorite(movie)) {  
    window.alert(`${movie.movieNm}은(는) 이미 즐겨찾기로 등록되어 있습니다.`);  
  } else {  
    addFavorite(movie);  
    window.alert(`${movie.movieNm}이(가) 즐겨찾기에 추가되었습니다.`);  
  }  
};
```

```
const addFavorite = (movie) => {  
  setFavorites((prevFavorites) => {  
    const isFavorite = prevFavorites.some(  
      (favMovie) => favMovie.title === movie.movieNm  
    );  
  });  
};
```

```
// 즐겨찾기에서 제거  
const removeFavorite = (movie) => {  
  setFavorites((prevFavorites) =>  
    prevFavorites.filter((favMovie) => favMovie.movieNm !== movie.movieNm)  
  );  
};  
  
// 즐겨찾기에 있는지 확인  
const isFavorite = (movie) => {  
  return favorites.some((favMovie) => favMovie.movieNm === movie.movieNm);  
};
```

즐거찾기에 영화 한편만 추가되는 오류 발생

시도:

- > some 함수를 map으로 변경
- > if문 안의 isFavorite에 "!" 기능 추가

해결:

- > Context와 모든 파일들의 받아오는 Key 값 통일화 진행

Thank you