



영화 예매 사이트

● BUSINESS PRESENTATION

CONTENT

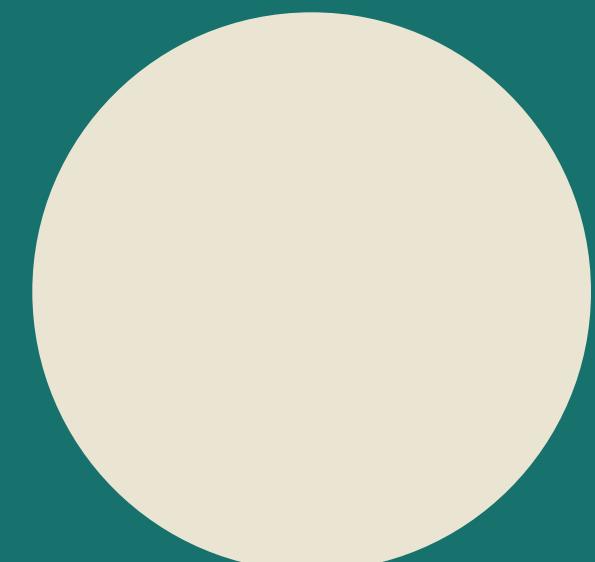
01 벤치마킹

02 기술스택

03 화면설계

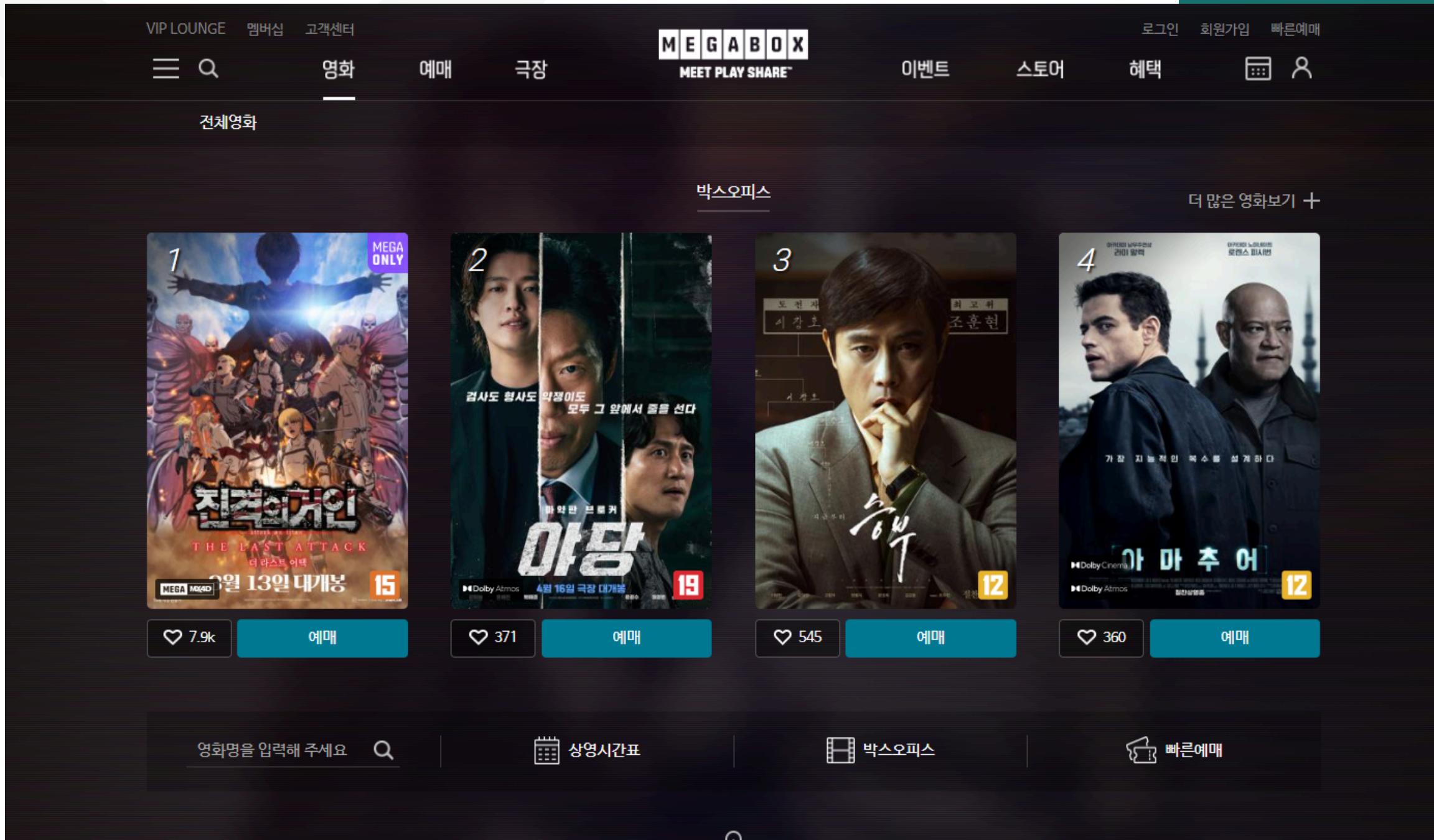
04 기능상세

05 후기



01. 벤치마킹

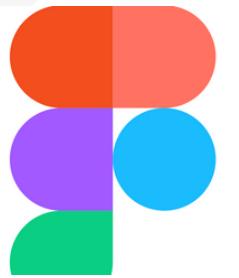
메가박스 사이트



02. 기술스택



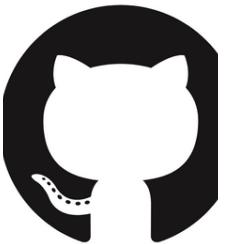
HTML/CSS/JS



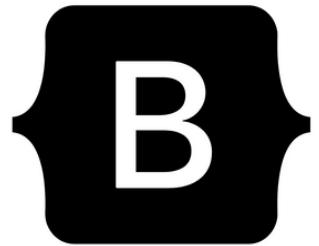
figma



jQuery



GitHub



Bootstrap5

03. 화면설계

• 메인페이지



03. 화면설계

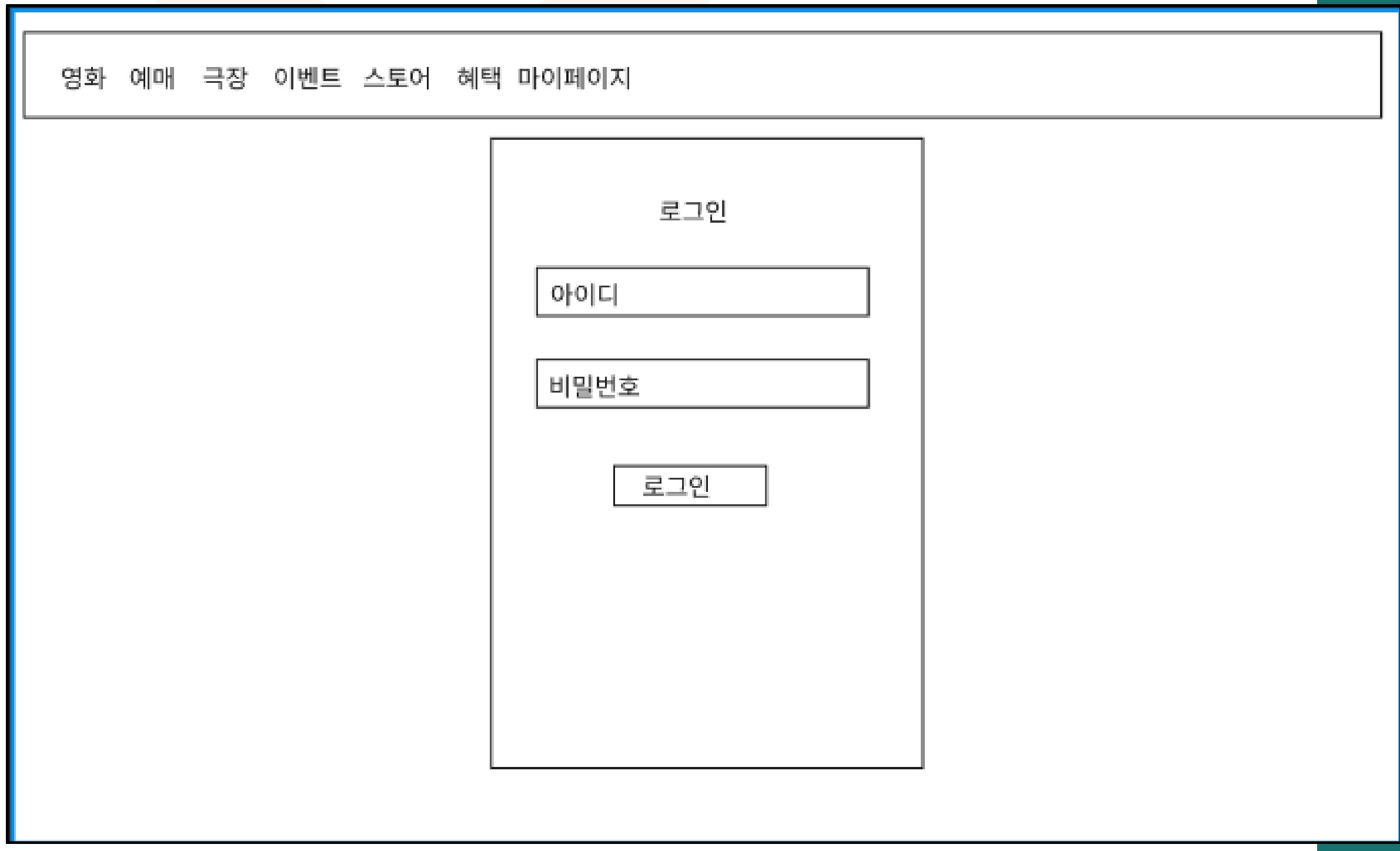
- 회원가입

The wireframe illustrates a member registration interface. At the top, there is a horizontal navigation bar containing links: 영화 예매 극장 이벤트 스토어 혜택 마이페이지. Below this, the main form area has a title '회원가입' (Member Registration) centered above four input fields. Each input field is labeled with its purpose: '아이디' (ID), '이메일' (Email), '비밀번호' (Password), and '비밀번호 확인' (Password Confirmation). A large, prominent '회원가입' button is located at the bottom center of the form.

항목	설명
아이디	ID
이메일	Email
비밀번호	Password
비밀번호 확인	Confirm Password
회원가입	

03. 화면설계

- 로그인



A wireframe diagram of a login screen. At the top, there is a horizontal navigation bar containing links: 영화, 예매, 극장, 이벤트, 스토어, 혜택, and 마이페이지. Below the navigation bar is a large rectangular input form. Inside this form, the word "로그인" (Login) is centered above two input fields: "아이디" (ID) and "비밀번호" (Password). Below these fields is another button labeled "로그인" (Login).

03. 화면설계

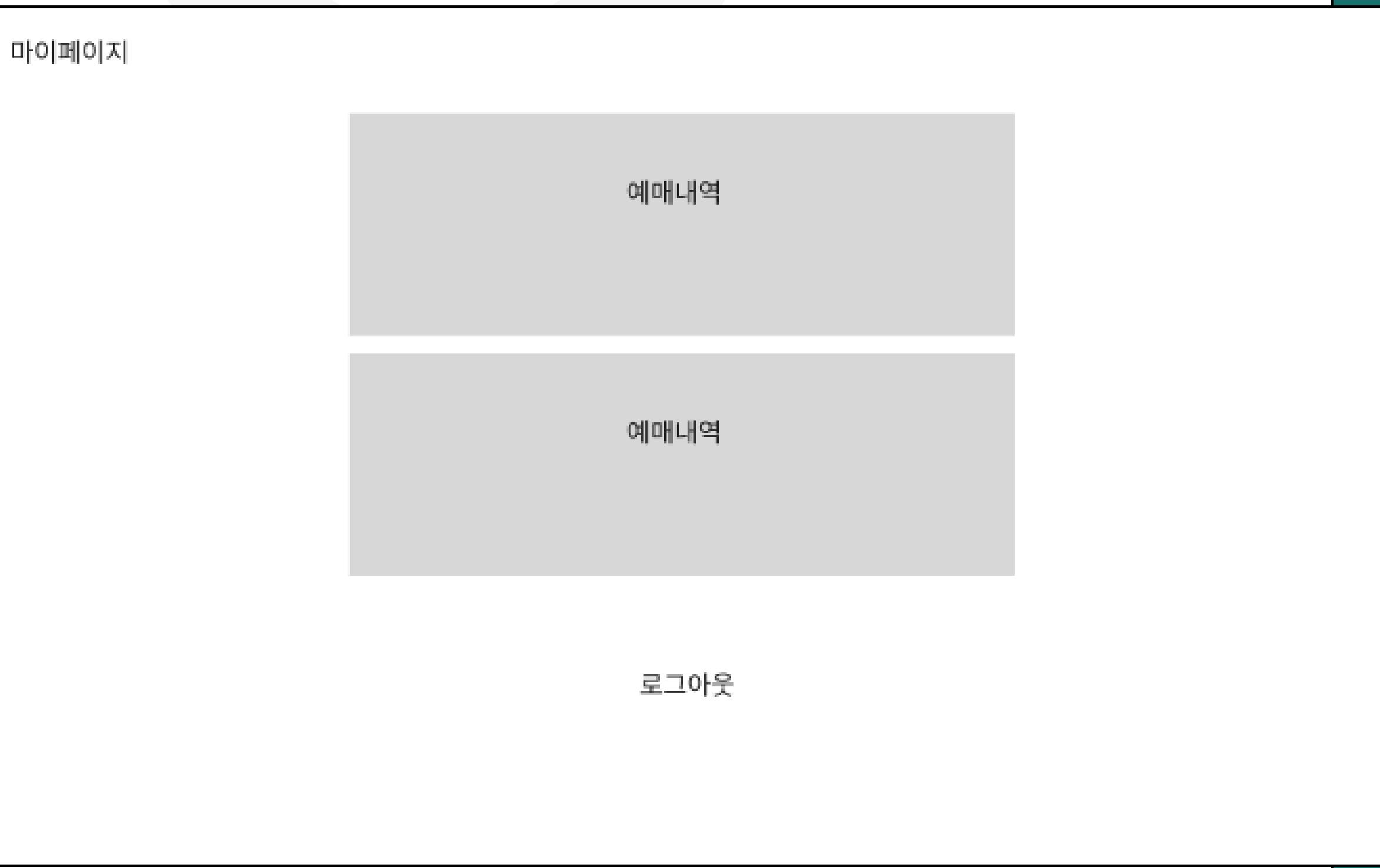
• 영화 검색

The wireframe illustrates a movie search interface. At the top, there is a navigation bar containing links for '영화', '예매', '극장', '이벤트', '스토어', '혜택', and '마이페이지'. To the right of the navigation bar is a button labeled '영화검색'. Below the navigation bar, there are three search results, each consisting of a thumbnail image labeled '영화 이미지', a title labeled '제목(년도)', and a subtitle labeled '제목(년도)'.

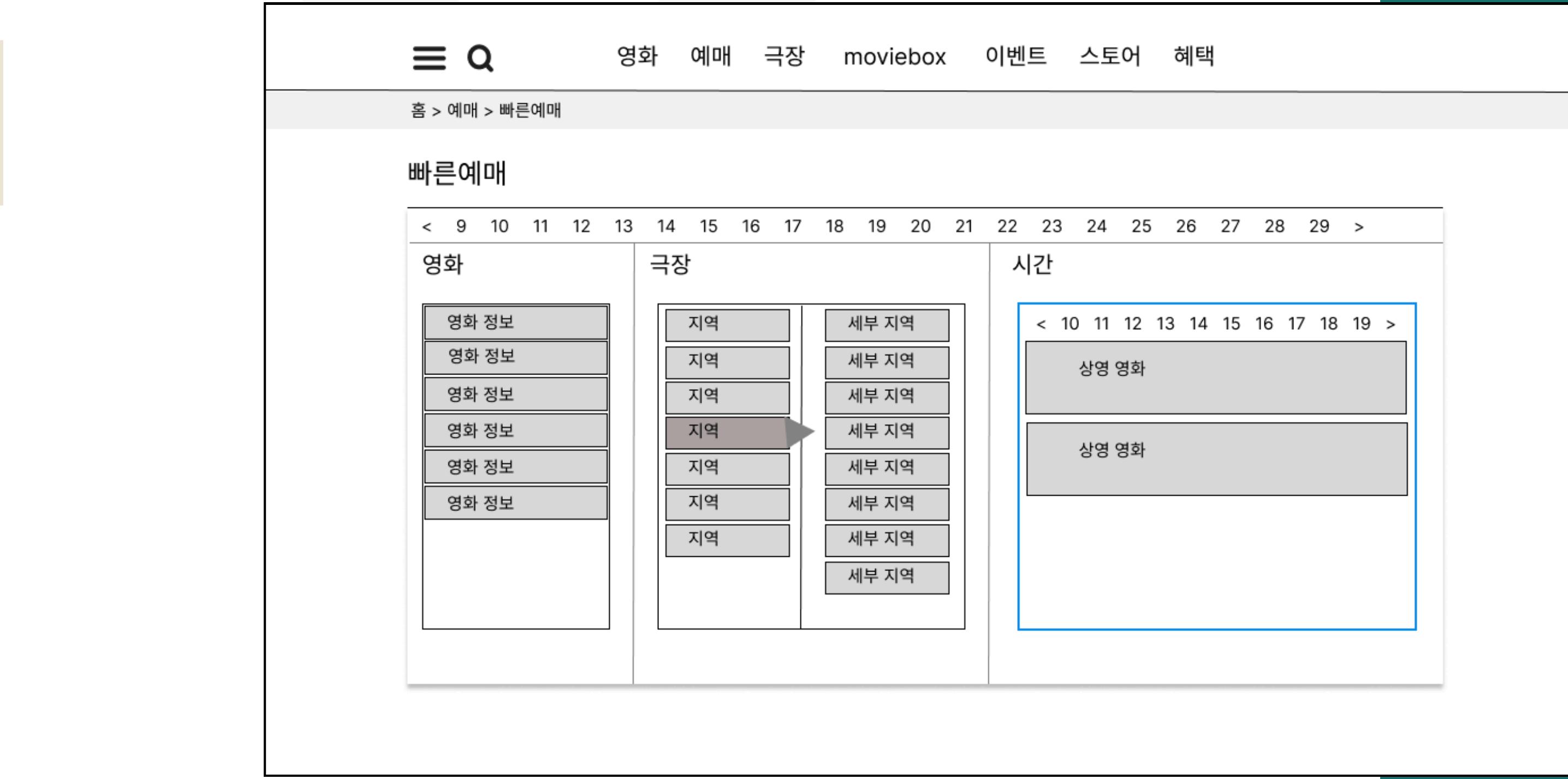
영화 이미지	제목(년도)	영화 이미지	제목(년도)	영화 이미지	제목(년도)
영화 이미지	제목(년도)	영화 이미지	제목(년도)	영화 이미지	제목(년도)

03. 화면설계

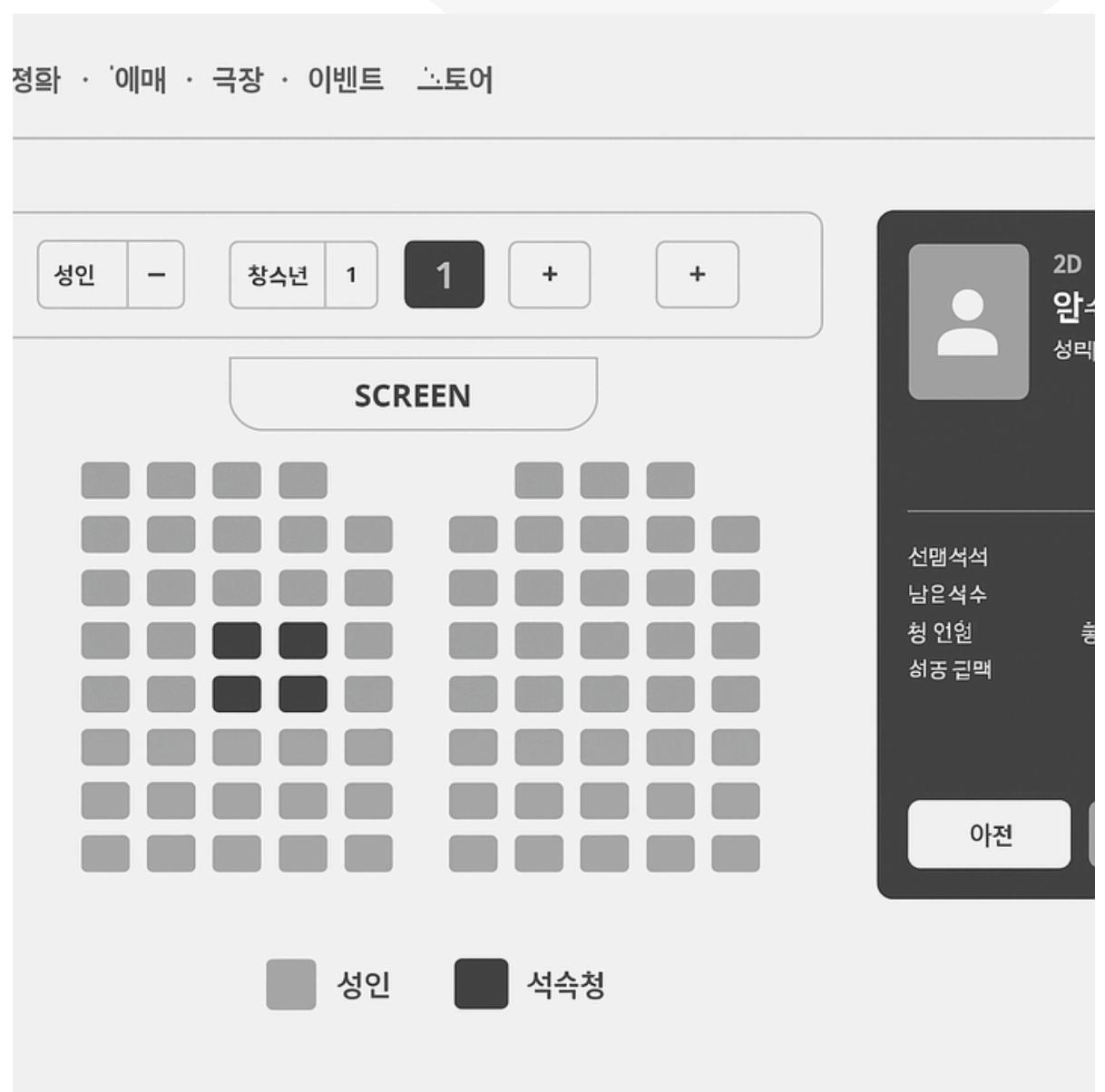
• 마이페이지



03. 화면설계



03. 화면설계



03. 화면설계

• 영화상세페이지

영화 예매 극장 MEGABOX 이벤트 스토어 혜택

영화제목
영어제목

액션 드라마

영화이미지

상영타입 : 2D, 2D ATMOS, 한글자막(ICC)
감독 : 흥병국 | 장르 : 드라마, 액션 | 122 분 | 등급 : 청소년관람불가 | 개봉일 : 2025.04.16
출연진 : 강하늘, 유해진, 박해준, 류경수, 채원빈

줄거리

영화줄거리

쪽쪽쪽....

더보기

영화제목에 대한 n개의 이야기가 있어요!

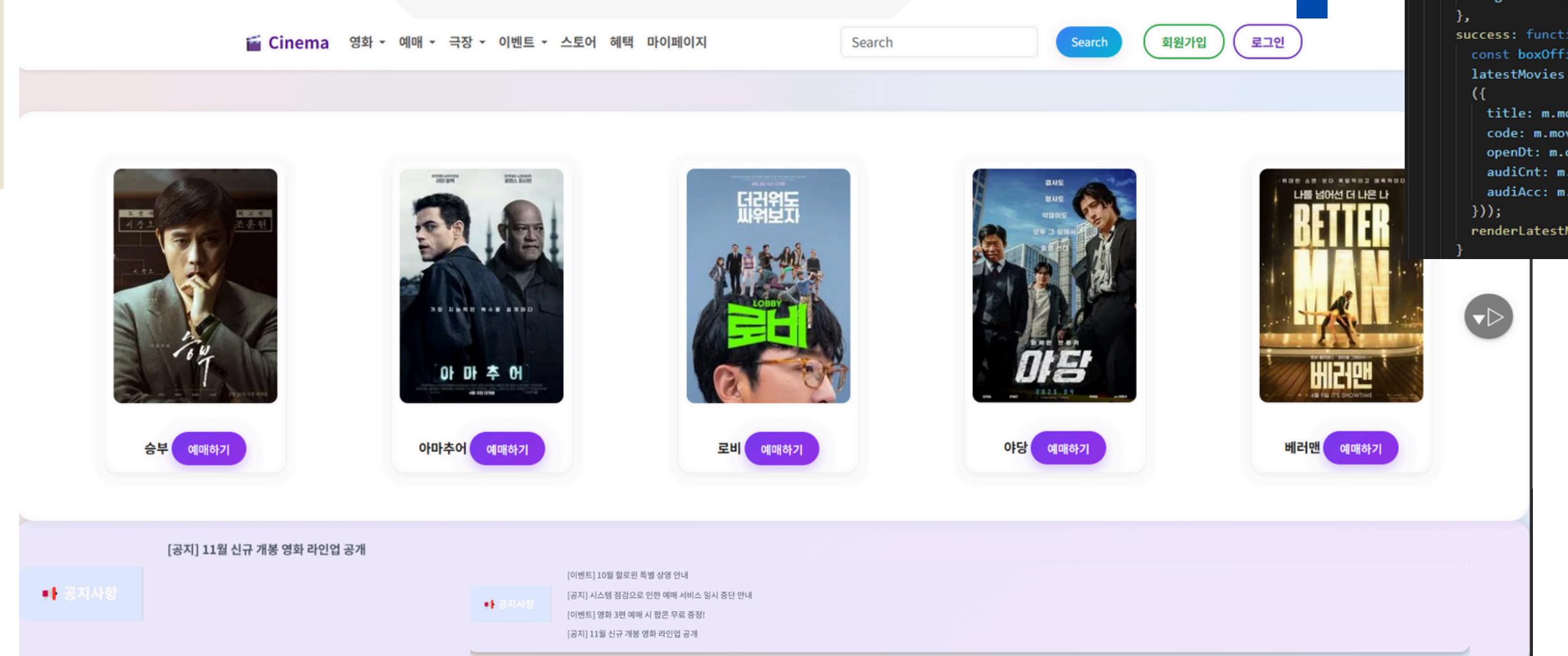
MEGABOX 관람평을 작성해주세요 관람평쓰기

프사 점수 스토리
아이디 멤버미
OST
배우
연출
댓글1

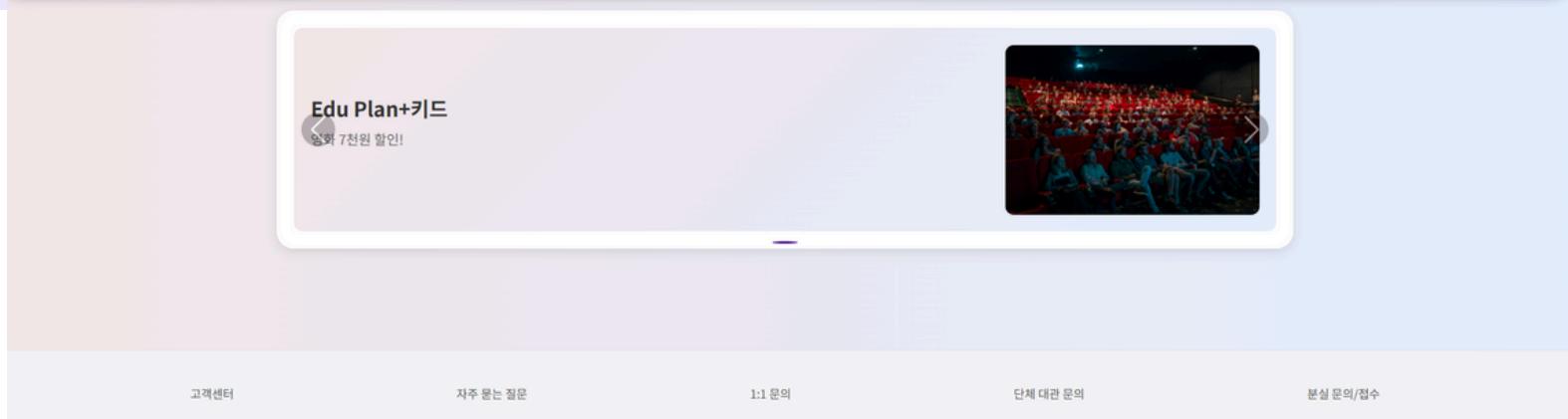
프사 점수 스토리
아이디 멤버미
OST
배우
연출
댓글2

04.기능상세

홈화면



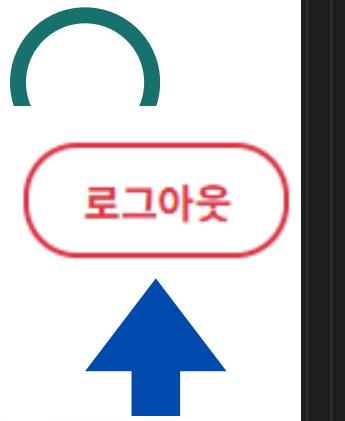
메인 페이지 아래 배너->



로그인 성공시 변경

환영합니다, kenneth93님!

로그아웃



```
// 영화 출력
let latestMovies = [];
let currentPage = 0;

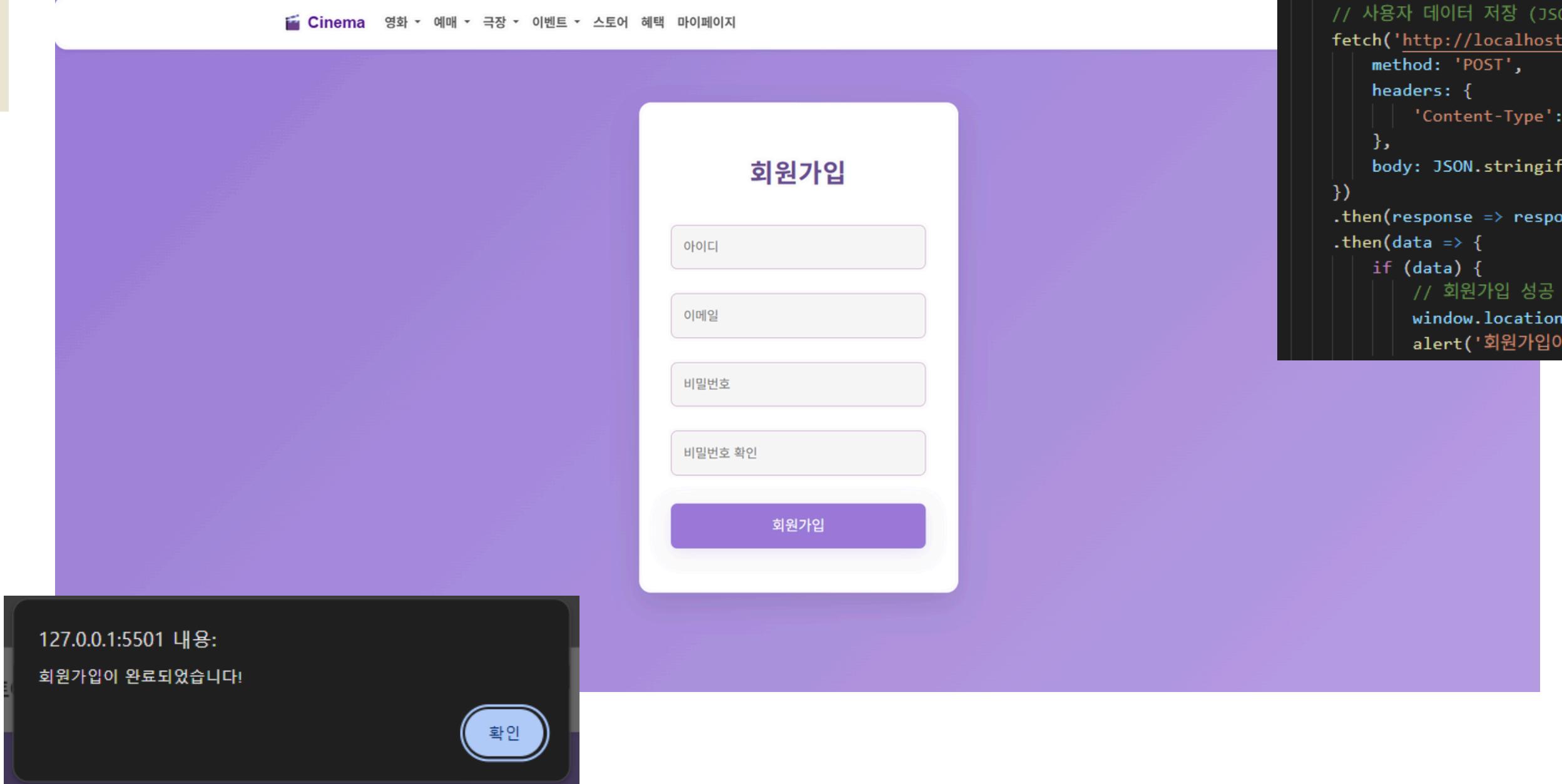
// 최신 영화 10개 불러오기 (박스오피스 TOP 10 기준)
$(document).ready(function () {
    const today = new Date();
    today.setDate(today.getDate() - 1); // 어제 날짜 기준
    const y = today.getFullYear();
    const m = String(today.getMonth() + 1).padStart(2, '0');
    const d = String(today.getDate()).padStart(2, '0');
    const targetDt = `${y}${m}${d}`;

    $.ajax({
        url: 'https://www.kobis.or.kr/kobisopenapi/webscr',
        type: 'GET',
        data: {
            key: '455015aa3f97520437568b039887193b',
            targetDt: targetDt
        },
        success: function (res) {
            const boxOfficeList = res.boxOfficeResult.dailyBoxOfficeList;
            latestMovies = boxOfficeList.map(m =>
            ({
                title: m.movieNm,
                code: m.movieCd,
                openDt: m.openDt,
                audiCnt: m.audiCnt,
                audiAcc: m.audiAcc
            }));
            renderLatestMovies();
        }
    });
});
```

KOBIS API로 최신영화 10개 불러옴
TMDB API로 포스터 불러옴
화살표 클릭하면 다음 포스터 5개를 보여줌

04. 기능상세

회원가입



```
// 회원가입 함수
function signup(event) {
  event.preventDefault(); // 기본 폼 제출 방지

  const username = document.getElementById('username').value;
  const email = document.getElementById('email').value;
  const password = document.getElementById('password').value;
  const confirmPassword = document.getElementById('confirm-password').value;

  // 오류 메시지 초기화
  document.getElementById('error-message').innerText = '';

  // 비밀번호 확인
  if (password !== confirmPassword) {
    document.getElementById('error-message').innerText = '비밀번호가 일치하지 않습니다.';
    return;
  }

  // 사용자 데이터 저장 (JSON Server에 보내는 코드 또는 로컬 저장)
  fetch('http://localhost:5500/users', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({ username, email, password })
  })
  .then(response => response.json())
  .then(data => {
    if (data) {
      // 회원가입 성공 시
      window.location.assign('index.html'); // 로그인 페이지로 리디렉션
      alert('회원가입이 완료되었습니다!');
    }
  })
}
```

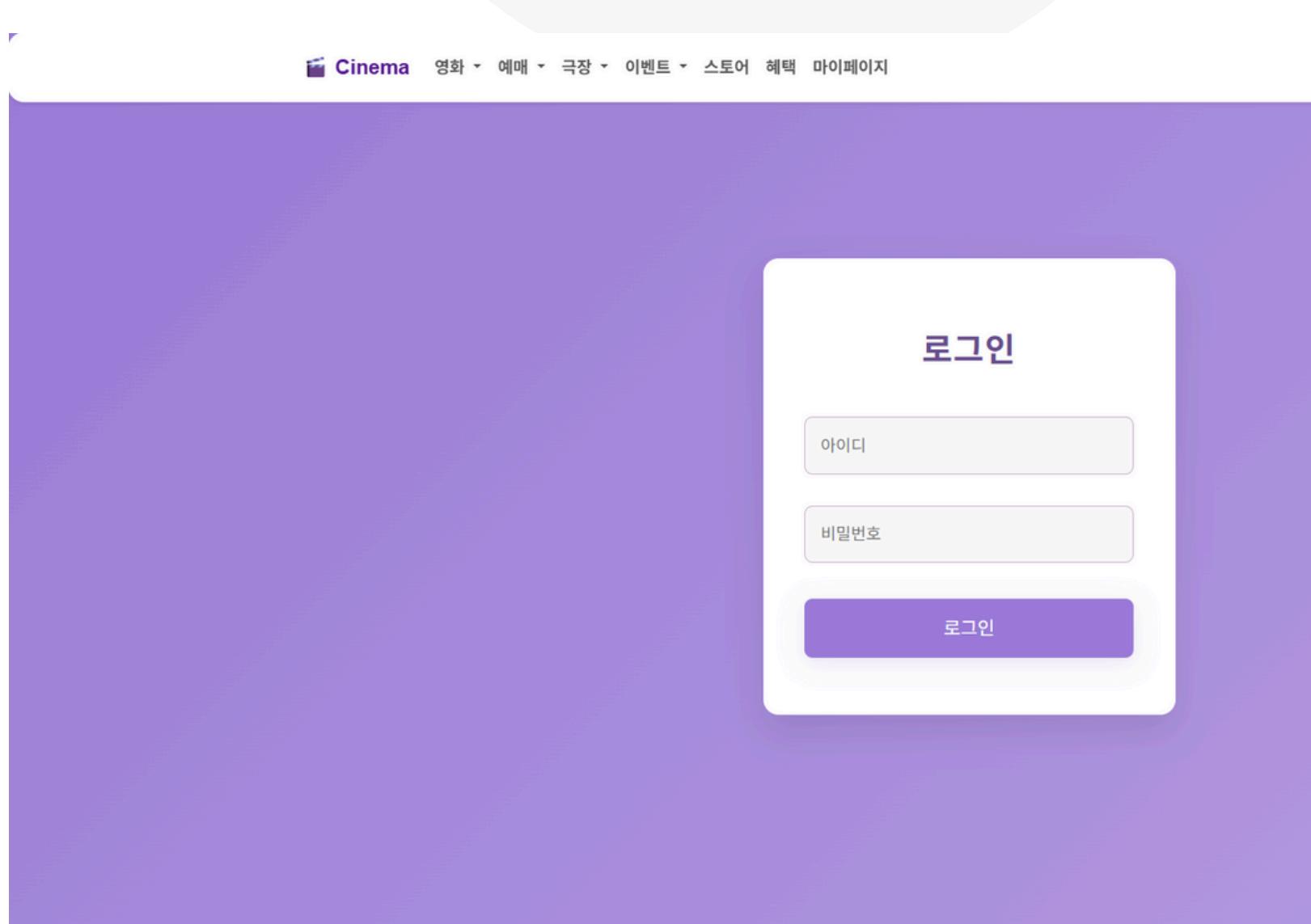
회원가입 폼을 제출하면 JSON Server에

```
"users": [
  {
    "username": "1111",
    "email": "1111@gmail.com",
    "password": "1234",
    "id": 1
}]
```

형식으로 사용자 정보를 저장

04. 기능상세

로그인



```
function login(event) {
  event.preventDefault(); // 기본 폼 제출 방지

  const username = document.getElementById('username').value;
  const password = document.getElementById('password').value;

  // 오류 메시지 초기화
  document.getElementById('error-message').innerText = '';

  // JSON Server에서 사용자 정보 가져오기
  fetch('http://localhost:5500/users')
    .then(response => response.json())
    .then(users => {
      const user = users.find(u => u.username === username && u.password === password);

      if (user) {
        // 로그인 성공 시
        localStorage.setItem('isLoggedIn', true);
        localStorage.setItem('username', username); // 로그인된 사용자 정보 저장

        // 1초 후 리디렉션을 시도하여 페이지 새로 고침 후 리디렉션
        setTimeout(() => {
          window.location.href = 'index.html'; // 홈 페이지로 리디렉션
        }, 1000); // 1초 딜레이 후 이동
      } else {
        document.getElementById('error-message').innerText = '아이디나 비밀번호가 틀렸습니다.';
      }
    })
    .catch(error => {
      console.error('로그인 오류:', error);
    });
}
```

로그인 성공하면 index.html로 이동
로그인 정보를 localStorage에 저장해 세션 유지에 활용

04. 기능상세

영화 검색

 Cinema 영화 ▾ 예매 ▾ 극장 ▾ 이벤트 ▾ 스토어 혜택 마이페이지



콘클라베
(2024)

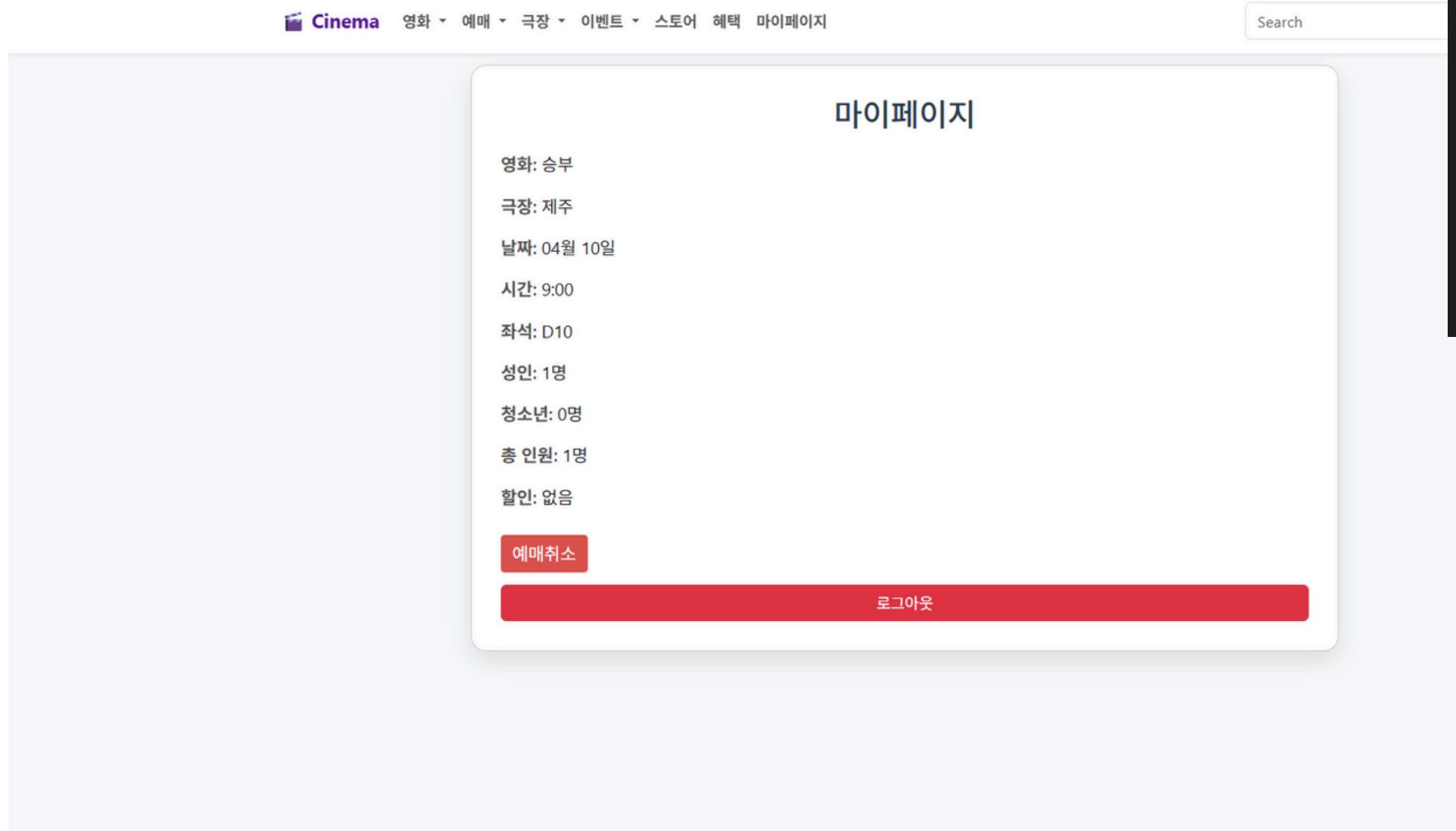
```
// URL에서 query 파라미터 가져오기
const params = new URLSearchParams(window.location.search);
const query = params.get('query');

if (query) {
    // KOBIS 영화 검색
    $.ajax({
        url: 'https://www.kobis.or.kr/kobisopenapi/webservice/rest/movie/searchMovieList.json',
        type: 'GET',
        data: {
            key: kobisKey,
            movieNm: query
        },
        success: function (data) {
            const list = data.movieListResult.movieList.slice(0, 10);
            if (list.length === 0) {
                $('#search-result').append('<li>검색 결과가 없습니다.</li>');
            } else {
                list.forEach(movie => {
                    fetchPoster(movie.movieNm, movie.prdtYear, function (posterUrl) {
                        const posterImg = posterUrl ? `` : '';
                        $('#search-result').append(`<li>${posterImg} <strong>${movie.movieNm}</strong> (${movie.prdtYear})</li>`);
                    });
                });
            }
        },
        error: err => console.error('KOBIS 에러:', err)
    });
}
```

주소창에 있는 검색어(query)로 KOBIS API에서 영화를 검색

04. 기능상세

마이페이지



```
const user = localStorage.getItem('username'); // 로그인한 사용자 확인

if (user) {
  fetchBookings(user); // 사용자 예매 내역 가져오기
} else {
  alert("로그인 후 마이페이지를 확인할 수 있습니다.");
  window.location.href = "login.html"; // 로그인 페이지로 리디렉션
}

function fetchBookings(username) {
  fetch(`http://localhost:5500/bookings?username=${username}`)
    // 응답받은 데이터 response를 json으로 파싱
    .then(response => response.json())
    // data : 위에서 파싱한 데이터. 즉, response.json()
    .then(data => {
      // 예매내역이 존재한다면, displayBooking 함수 실행
      if (data.length > 0) {
        displayBookings(data);
      } else {
        document.getElementById('bookings-list').innerHTML = "<p>예매 내역이 없습니다.</p>";
      }
    })
    .catch(error => {
      console.error("예매 내역을 가져오는 중 오류 발생:", error);
      alert("예매 내역을 가져오는 데 실패했습니다.");
    });
} // end fetchBookings
```

JSON Server를 활용해서
"bookings": [

```
{
  "username": "1111",
  "movie": "아마추어",
  "theater": "제주",
  "date": "04월 10일",
  "time": "10:00",
  "seats": "C12",
  "adult": "1",
  "teen": "0",
  "id": 1
}]
```

에서 username 비교해서 정보 가져옴

04. 기능상세

상세 페이지

index페이지로부터 get 방식으로 영화코드를 전달받음.
전달받은 영화코드를 KOBIS와 TMDB에 요청해서
영화정보와 포스터를 가져와서 출력함

The screenshot shows a movie detail page for the Korean film '로비' (Lobby). At the top, there's a navigation bar with 'Cinema' and various links like '영화', '예매', '극장', '이벤트', '스토어', '혜택', and 'マイ페이지'. Below the navigation is a search bar with 'Search' and 'Search' button. The main content area features a large movie poster with a man in glasses looking down. To the right of the poster is a smaller image showing a group of people standing on a roof. The movie title '로비' is displayed prominently in large letters. Below the poster, there's a synopsis in Korean: '연구밖에 모르는 스타트업 대표 창욱은 라이벌 회사 대표 광우의 뒷거래 때문에 기회도, 기술도 번번히 빼앗긴다. 그의 회사의 유일한 탈출구는 4조 원에 달하는 국책사업을 따내어, 한방에 자본을 확보하는 것! 하지만 로비에 있어선 한수 ...' There are also buttons for '예매하기' (Buy tickets) and '더보기' (More details).

```
// 현재 페이지의 쿼리 문자열을 가져옴
const queryString = window.location.search;
// 쿼리 파라미터를 분석
const urlParams = new URLSearchParams(queryString);
// movieCd 값을 가져와서 변수에 저장
const movieCd = urlParams.get('movieCd');
console.log(movieCd); // 👍 "20253093" 출력됨

/** 영화정보 받아오기 */
const KOBIS_KEY = '3b2ef121a16d8b50bc772dcb2b51e544';
// const movieCd = "20050110";

$.ajax({
  url: "https://www.kobis.or.kr/kobisopenapi/webservice/rest/moviedetail/getMovieDetailByMovieCd.json?&key=" + KOBIS_KEY + "&movieCd=" + movieCd,
  method: "GET",
  dataType: "json",
  data: {
    key: KOBIS_KEY,
    movieCd: movieCd
  },
  success: function(data) {
    console.log(data);
  }
});
```

04. 기능상세

관람평

관람평

로그인 되어있는 상태에선 작성자 아이디가,

작성자 : 로그아웃 상태에선 작성자를 임의로 작성할 수 있음.

 kenneth93 작성한 코멘트는 db에 저장되어 아래에 출력됨.

평점:



평점은 5점 만점이고 0.5점씩 선택할 수 있음

좋았던 점:

스토리 연출 영상미 배우 OST

재밌어요

등록



kenneth93

평점 : 5

스토리

체크한 좋았던 점은 코멘트 위 사각형에 보여짐

좋아요~!!

좋아요

추천수: 0

```
"comments": [
  {
    "movieCd": "20233008",
    "writer": "kenneth93",
    "rating": "4.5",
    "good_points": "<em class=\"tag\">배우</em> &nbsp;",
    "comment": "역시",
    "created_at": "2025-04-11T05:20:26.716Z",
    "like": 8,
    "id": 1
  },
  {
    "movieCd": "20233008",
    "writer": "kenneth93",
    "rating": "3.5",
    "good_points": "<em class=\"tag\">스토리</em> &nbsp;",
    "comment": "굳굳",
    "created_at": "2025-04-11T06:11:24.752Z",
    "like": 0,
    "id": 2
  }
]
```

04. 기능상세

관람평



kenneth93

평점 : 5

스토리

좋아요~!!

좋아요

추천수: 5

```
{  
    "movieCd": "20203863",  
    "writer": "kenneth93",  
    "rating": "5",  
    "good_points": "<em class=\"tag\">스토리</em> &nbsp;",  
    "comment": "좋아요~!!",  
    "created_at": "2025-04-11T06:30:40.880Z",  
    "like": 5,  
    "id": 3  
}
```

좋아요를 누르면 db에 저장된 코멘트의 like+=1
결과적으로 추천수가 1씩 늘어남.

04. 기능상세

예매

The screenshot shows a user interface for movie booking. At the top is a calendar for April 2025, with the 11th highlighted. Below it are three tabs: 영화 (Movie), 극장 (Theater), and 시간 (Time). The 영화 tab shows a dropdown menu with '승부' selected, and a list of movies: 아마추어, 극장판 진격의 거인 완결편 더 라스트 어택, 로비, and 애당. The 극장 tab shows a dropdown menu with '경기' selected, and a list of theaters: 서울 (서울, 수원, 고양), 경기 (인천, 대전/충청/세종, 부산/대구/경상, 광주/전라, 강원), and 충북 (아마추어). The 시간 tab shows a timeline from 9 to 23 hours. A purple bar highlights the time range from 16:00 to 18:15, labeled '16:00 ~ 18:15' and '2관 · 2D' with '잔여석 7/100'. Below this, another purple bar highlights the time range from 21:00 to 23:15, labeled '21:00 ~ 23:15' and '1관 · 3D' with '잔여석 75/100'. An arrow points upwards from the theater selection area towards the time selection area.

```
function updateSelectionSummary() {
  const summary = document.getElementById('selection-summary');
  if (selectedMovie && selectedTheater && selectedDate && selectedTime) {
    summary.textContent = `${selectedMovie} | ${selectedTheater} | ${selectedDate.slice(4, 6)}월 ${selectedDate.slice(6)}일 | ${selectedTime}`;
  } else {
    summary.textContent = '날짜, 영화, 극장, 시간을 선택하면 여기에 표시됩니다.';
  }
}
```

<랜덤 상영시간 생성 함수>
오전 9시부터 오후 11시까지 시간대 중 무작위로 상영시간을 생성하고,
관 번호, 포맷(2D/3D), 좌석 수(잔여/총 좌석)를 포함한 상영정보 객체를 배열로 반환.
상영정보는 기존 화면에 표시할 때 활용.

```
function generateRandomTimes() {
  const times = [];
  const screens = ['1관', '2관', '3관'];
  const formats = ['2D', '3D'];

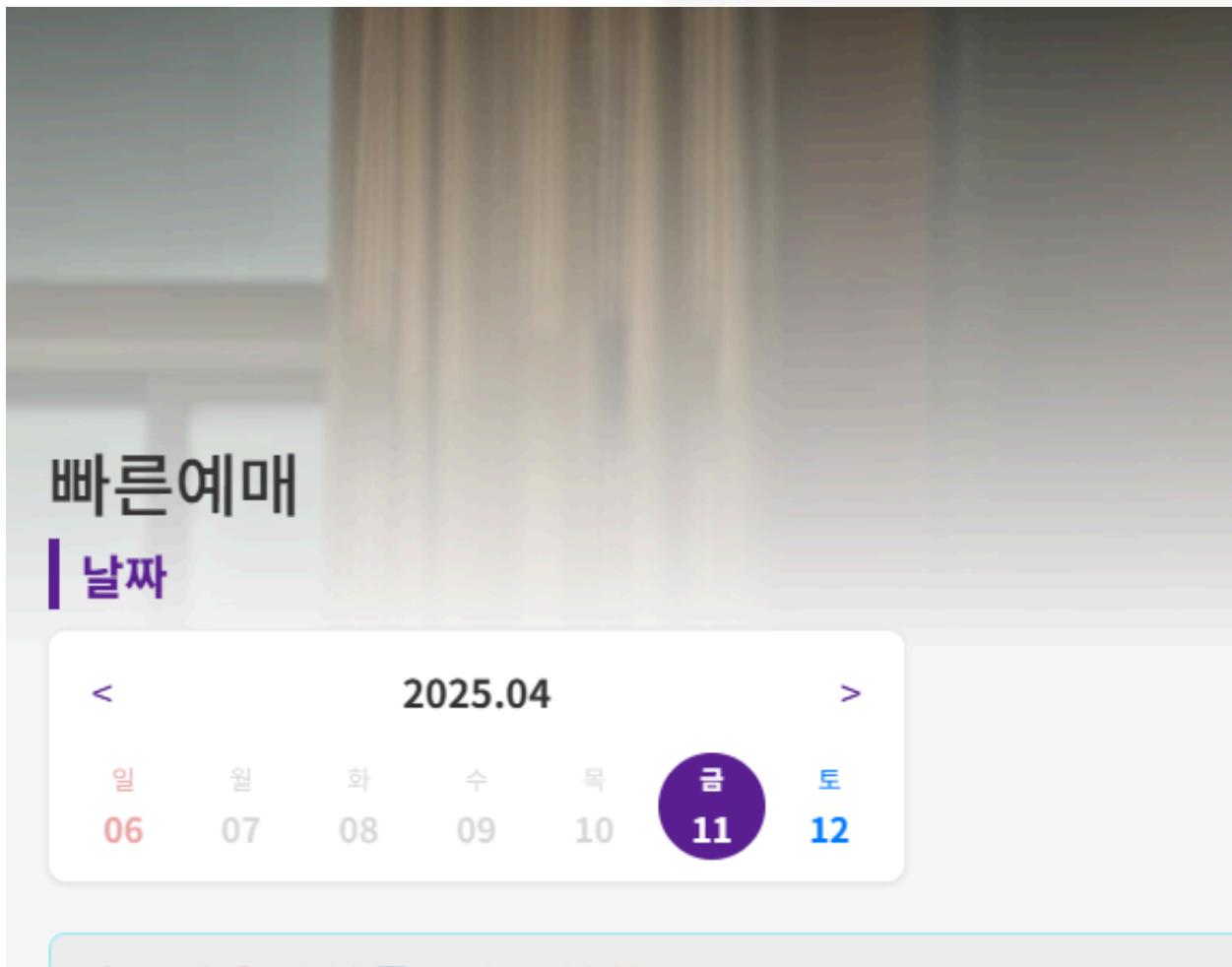
  for (let h = 9; h <= 23; h++) {
    if (Math.random() > 0.4) {
      const startHour = h;
      const endHour = h + 2;
      const screen = screens[Math.floor(Math.random() * screens.length)];
      const format = formats[Math.floor(Math.random() * formats.length)];
      const totalSeats = 100;
      const remaining = Math.floor(Math.random() * totalSeats);

      times.push({
        start: `${startHour}:00`,
        end: `${endHour}:15`,
        screen,
        format,
        remaining,
        total: totalSeats
      });
    }
  }
  return times;
}
```

선택된 영화, 극장, 날짜, 시간을 요약해서 화면에 표시.
모든 항목이 선택되었을 때만 요약 문구를 출력하고,
선택 전에는 안내 문구를 출력.
선택 상태를 실시간으로 반영하여 기존 요약 표시 영역
업데이트.

04. 기능상세

예매



<날짜 선택 UI 구성>

오늘 날짜는 기본 선택되며, 주말은 색상 강조(토요일: 파란색, 일요일: 빨간색).

과거 날짜는 선택 불가하도록 비활성화 처리.

날짜 클릭 시 선택된 날짜로 갱신되고, 상영시간 영역 업데이트.

```
// 오늘 날짜 선택
if (
  date.getDate() === today.getDate() &&
  date.getMonth() === today.getMonth() &&
  date.getFullYear() === today.getFullYear()
) {
  btn.classList.add('selected');
  selectedDate = date.toISOString().slice(0, 10).replace(/-/g, '');
}

// 주말 강조 (일요일: 빨간색, 토요일: 파란색)
const isSunday = date.getDay() === 0;
const isSaturday = date.getDay() === 6;
if (isSunday) btn.classList.add('sunday');
if (isSaturday) btn.classList.add('saturday');

// 과거 날짜 비활성화
if (date < today) {
  btn.disabled = true;
  btn.style.opacity = '0.5';
  btn.style.cursor = 'not-allowed';
} else {
  // 클릭 이벤트
  btn.addEventListener('click', () => {
    document.querySelectorAll('#date-selector .date-btn').forEach(b => b.classList.remove('selected'));
    btn.classList.add('selected');
    selectedDate = date.toISOString().slice(0, 10).replace(/-/g, '');
    updateTimeSection();
  });
}

container.appendChild(btn);
```

04. 기능상세

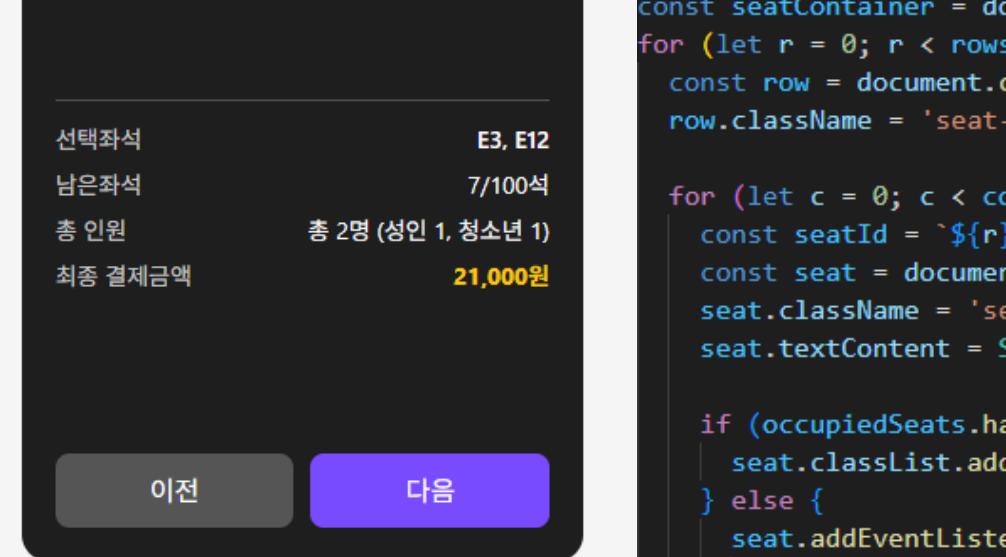
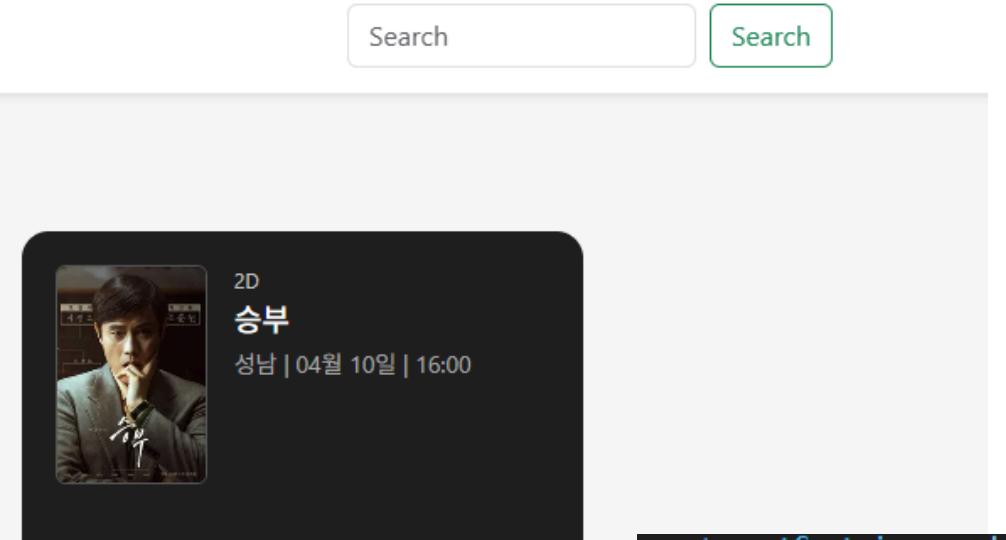
좌석 선택

The screenshot shows a movie ticketing website's seat selection page. At the top, there are buttons for selecting the number of adults (성인), children (청소년), and discounts (우대). Below this is a large grid labeled "SCREEN" containing 120 seats, numbered from A1 to H12. Seats E3 and E12 are highlighted in purple, indicating they are selected. To the right of the grid, there is a summary of the selection: 2D, 승부, 성남 | 04월 10일 | 16:00. The summary also shows the number of passengers (총 인원) as 2명 (성인 1, 청소년 1) and the total amount (최종 결제금액) as 21,000원.

좌석을 행×열로 동적으로 생성하고, 이미 예약된 좌석은 occupied 처리.

예약 가능한 좌석은 클릭 시 선택/해제 가능.

선택 좌석 수는 인원 수(counts)만큼 제한되며, 요약 카드 실시간 갱신.



선택한 영화 정보 불러오기

```
const params = new URLSearchParams(window.location.search);
const movie = params.get('movie');
const theater = params.get('theater');
const date = params.get('date');
const time = params.get('time');
const totalSeats = parseInt(params.get('total'));
const remainingSeats = parseInt(params.get('remaining'));
const occupiedCount = totalSeats - remainingSeats;
```

```
const seatContainer = document.getElementById('seat-map');
for (let r = 0; r < rows; r++) {
  const row = document.createElement('div');
  row.className = 'seat-row';

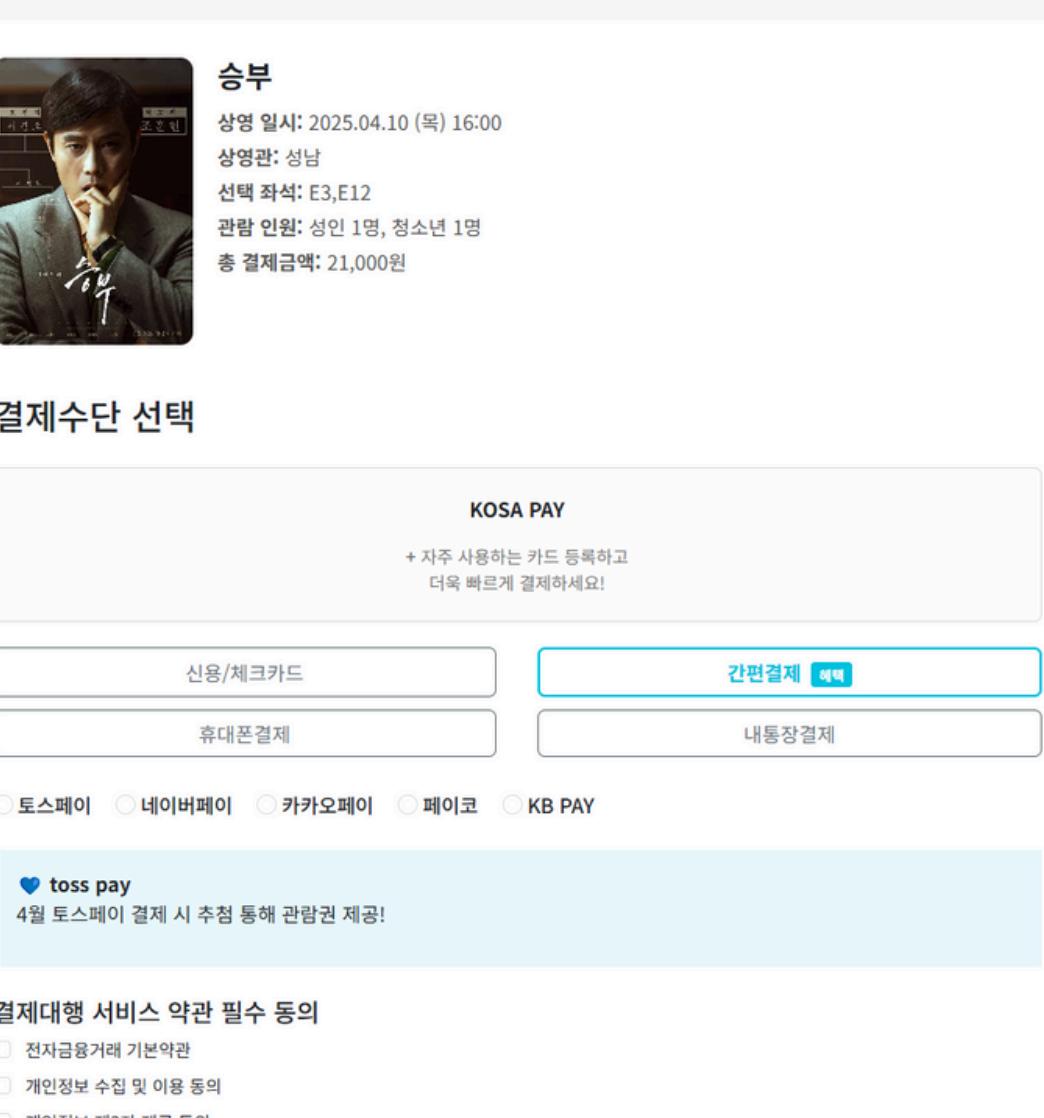
  for (let c = 0; c < cols; c++) {
    const seatId = `${r}-${c}`;
    const seat = document.createElement('div');
    seat.className = 'seat';
    seat.textContent = String.fromCharCode(65 + r) + (c + 1);

    if (occupiedSeats.has(seatId)) {
      seat.classList.add('occupied');
    } else {
      seat.addEventListener('click', () => {
        const maxSelectableSeats = counts.adult + counts.teen + counts.discount;

        if (seat.classList.contains('selected')) {
          seat.classList.remove('selected');
          selectedSeats.delete(seatId);
        } else {
          if (selectedSeats.size >= maxSelectableSeats) {
            alert(`${maxSelectableSeats}명까지만 선택할 수 있어요.`);
            return;
          }
          seat.classList.add('selected');
          selectedSeats.add(seatId);
        }
      });
    }
  }
}
```

04. 기능상세

결제



```
function selectMethod(button) {
  document.querySelectorAll('.payment-buttons .btn').forEach(btn => btn.classList.remove('selected'));
  button.classList.add('selected');

  const simplePayOptions = document.querySelector('.simple-pay-options');
  const promoBox = document.querySelector('.promotion');

  if (button.textContent.includes('간편결제')) {
    simplePayOptions.style.display = 'block';
    promoBox.style.display = 'block';
  } else {
    simplePayOptions.style.display = 'none';
    promoBox.style.display = 'none';
  }
}
```

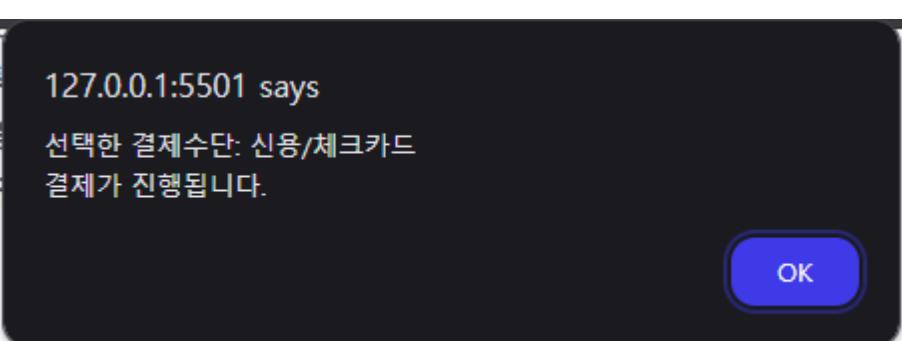
클릭된 버튼에 .selected 클래스를 적용

'간편결제'가 포함된 버튼이면 간편결제 옵션과 프로모션 박스 표시

다른 방식일 경우 해당 요소 숨김

약관 필수동의를 다 선택해야 결제 가능

결제 알림창



04. 기능상세

모바일 티켓



승부
2025.04.10 | 10:00 ~ 12:00

상영관: 강남
좌석: D7
관람 인원: 성인 1명



PDF 다운로드

홈으로

```
<div class="btn-box">
  <button class="download-btn" onclick="window.print()">
    PDF 다운로드
  </button>
  <button class="home-btn" onclick="window.location.href='mypage.html'">
    홈으로
  </button>
```

PDF 다운로드 버튼: 화면 프린트

홈버튼: 예매내역

이미지를 a태그 안에 넣음 → QR 클릭시 페이지 이동

```
<div class="qr">
  <a id="qrLink" href="#" target="_blank">
    <img id="qr" alt="QR Code" />
  </a>
</div>
```

QR코드를 스캔하면 해당 영화 검색 결과 페이지로 이동

```
const searchURL = `https://www.google.com/search?q=${encodeURIComponent(
  movie
)}+영화`;

const qrImg = document.getElementById("qr");
const qrLink = document.getElementById("qrLink");

qrImg.src = `https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=${encodeURIComponent(
  searchURL
)}`;
qrLink.href = searchURL;
```

04. 기능상세

모바일 티켓



승부

2025.04.10 | 10:00 ~ 12:00

상영관: 강남
좌석: D7
관람 인원: 성인 1명



PDF 다운로드

홈으로

```
const bookingInfo = {
  username: user,
  movie: movie,
  theater: theater,
  date: `${params.get("date").slice(4, 6)}월 ${params
    .get("date")
    .slice(6)}일`,
  time: time,
  seats: seats,
  adult: adult,
  teen: teen,
```

예매 정보 객체 생성

```
.then(response) => response.json()
.then(data) => {
  window.location.replace("mypage.html");
  alert("결제가 완료되었습니다! 즐거운 영화 관람 되세요 😊");
}
.catch(error) => {
  console.error("예매 정보 저장 중 오류 발생:", error);
  alert("예매 정보 저장에 실패했습니다.");
});
```

응답을 JSON으로 받음

성공하면 마이페이지 이동 → 알림 메시지

실패하면 콘솔에 로그 출력 → 실패 메시지

```
fetch("http://localhost:5500/bookings", [
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify(bookingInfo),
])
```

예매 정보 객체를 POST 방식으로 전송
JS객체를 JSON 문자열로 바꿔서 전송

04. 기능상세

영상





THANK YOU

● FOR YOUR NICE ATTENTION

