

PJT명	JS를 활용한 DOM 조작 프로젝트	
단계	[JavaScript 개발]	
진행일자	2025.11.21	
예상 구현 시간	필수기능	5H
	추가기능	2H
	심화기능	1H

1. 목표

- JavaScript를 이용하여 HTML의 내용을 조작할 수 있다.
- 사용자 동작에 따라 HTML의 내용을 변경할 수 있다.

2. 준비사항

1) 제공 파일

- 요구사항을 위한 UI 일부가 구현된 HTML 파일
 - index.html
- 데이터 제공을 위한 JS 파일
 - genres.js, movies.js
- 요구사항 구현을 위한 JS 파일
 - script.js

2) 개발언어 및 툴

- JavaScript
- Visual Studio Code
- Chrome Browser

3) 필수 라이브러리 / 오픈소스

- Bootstrap 5.3

3. 작업 순서

- 1) 팀원과 같이 요구사항을 확인하고, GitLab에 프로젝트를 생성한다.
 - 프로젝트 이름은 08-pjt로 지정한다.
 - 각 반 담당 강사님을 Maintainer로 설정한다.
- 2) 팀원과 합의하여 협업 방식을 결정하고, 요구사항을 구현한다.
- 3) 작성한 코드들을 정리하고, README를 작성한다.
 - .gitignore 파일을 활용하여 불필요한 파일 및 폴더는 제출하지 않는다.
- 4) README 작성이 완료되면 심화 학습을 진행한다.
- 5) 제출 기한에 맞춰 모든 산출물이 GitLab에 업로드 될 수 있도록 한다.

4. 요구사항

추천 알고리즘을 통한 영화 추천 커뮤니티 서비스를 구축하려고 한다. 다양한 스트리밍 플랫폼에서 제공되는 영화 정보를 수집 및 관리하고, 이를 기반으로 개인화된 영화 추천, 장르별 영화 탐색, 유사 영화 추천 등 다채로운 추천 기능을 설계 및 구현한다. 또한 영화에 대한 사용자 리뷰 및 감상평 공유 커뮤니티 기능을 제공하여, 사용자들이 활발하게 소통하고 정보를 교환할 수 있는 기능을 제공한다. 사용자는 자신이 본 영화를 평가하고, 다른 사용자의 리뷰를 참고하여 다음 영화를 선택하는데 도움을 받을 수 있다. 나아가, 관심 영화 목록을 맞춤형으로 구성하는 등 다양한 편의 기능을 제공한다. 팀원과 상의하여 아래 요구사항을 만족할 수 있도록 요구 사항 명세서를 작성 및 구현해보자.

사용자 경험 증대를 목표로 화면 개발을 하고자 한다. 영화 목록을 장르에 따라 필터링 하고, 목록에 데이터를 추가하는 기능을 화면 새로고침 없이 제공하기 위해 JavaScript를 도입해보자.

- 요구사항 예시(참고용)
 - 아래의 내용을 참고하여 추가적인 아이디어에 대해 요구사항을 추가 또는 수정하여 기능을 구현한다. 단, **필수 기능은 구현해야 하며, 수정할 수 없다.**

번호	분류	요구사항명	요구사항 상세	우선순위
기능적 요구사항				
F01	JavaScript	영화 목록 출력	주어진 영화 데이터에 따라 화면이 다르게 출력되도록 구현	필수
F02	JavaScript	장르별 필터링	특정 UI의 변경에 따라 출력되는 영화 데이터가 변경되도록 구현	필수
F03	JavaScript	영화 데이터 입력 양식	데이터를 입력할 수 있는 UI를 장르 데이터의 변화에 대응할 수 있는 형태로 구현	필수
F04	JavaScript	영화 데이터 추가	입력된 영화 데이터를 바탕으로 데이터 목록 갱신	필수
F05	JavaScript	영화 데이터 삭제	지정된 영화 데이터를 목록에서 제거	필수
F06	AI 활용	즐겨찾기 기능	AI를 활용하여 특정 영화 데이터를 쉽게 찾아볼 수 있는 기능 구현	도전
F07	AI 활용	Modal 구현	AI를 활용하여 영화의 상세정보를 확인할 수 있는 모달 창 구현	도전
...
비기능적 요구사항				
NF01	프로젝트 관리	Git 활용	개발자 간 작업 충돌이 일어나지 않게끔 Git을 활용하여 프로젝트 관리	필수
...

1) 프로젝트 관리 (필수)

Git의 기능을 적극적으로 활용하여, 여러 개발자가 동시에 개발할 수 있는 환경을 조성한다.

- 요구사항 번호: NF01
- 팀원과 합의하여 branch 생성 원칙 정리하기
 - Gitflow, GitHub Flow 등을 참고하여 원칙을 정할 수 있음
 - branch 운영 중 문제상황이 발생할 경우 원인에 대한 분석을 꼭 진행할 것
- 팀원과 합의하여 commit 내역 기록 원칙 정리하기
 - 기능 개발, 버그 수정 등 그 목적이 명확히 드러날 수 있도록 commit 메시지를 작성할 것
 - 어느 시점에 commit을 남길지에 대하여 충분한 합의를 이를 것
- 프로젝트 종료 시 초기에 세운 원칙이 잘 지켜졌는지 점검
 - README에 초기에 세운 원칙을 설명하여 정리할 것
 - 생성했던 branch 들에 대한 용도를 정리할 것
 - 작성된 commit들을 GitLab에서 확인하여 스크린샷으로 README에 포함시킬 것

2) JavaScript (필수)

주어진 영화, 장르 데이터를 바탕으로 사용자의 행동에 따라 동작하는 UI를 구현해보자.

- 요구사항에 따라 index.html에 이미 코드가 작성된 경우가 존재한다.
- 원한다면 CSS, Bootstrap 등을 활용하여 추가 스타일링이 가능하다.
 - 출력 예시에서 사용된 UI 컴포넌트가 아니더라도 사용할 수 있다.
- 주어진 genres.js, movies.js 또는 script.js에 미리 구현된 코드를 수정할 경우 정상적으로 동작하지 않을 수 있다.
 - 기본적으로 모든 요구사항은 index.html, script.js만 수정해도 구현할 수 있지만, 원한다면 JS파일을 추가할 수 있다.
- 요구사항 구현을 위한 코드가 전부 독립적일 필요는 없다.
 - 새로운 요구사항 구현을 위해, 이전에 작성한 코드를 수정하여도 무방하다.
- HTML 요소를 추가하기 위한 목적의 **Element.innerHTML의 사용을 금지**한다.
 - 특정 HTML 요소의 content를 제거하는 용도로는 사용 가능하다.

A. 영화 목록 출력

주어진 영화 데이터를 화면에 출력하고자 한다. 각 영화의 제목을 나열하여 출력해보자.

- 요구사항 번호: F01
- 'div#movie-list-section'의 내부에 목록으로 구현
 - script.js에 정의된 movies의 데이터 활용
 - ul 요소 사용
 - 각 영화 제목을 li 요소로 만든 후 ul 요소에 추가
- 출력 예시

Movies

Movie List

- 나의 히어로 아카데미아 더 무비: 히어로즈 라이징
- 반지의 제왕: 왕의 귀환
- 다크 나이트
- 스티븐 유니버스: 더 무비
- 7인의 사무라이
- 모탈 컴뱃 레전드: 스콜피온의 복수
- 저스티스 리그 다크: 아포콜립스 위
- 스파이더맨: 뉴 유니버스
- 스타워즈 에피소드 5: 제국의 역습
- 할복
- 반지의 제왕: 반지 원정대
- 신세기 에반게리온: 엔드 오브 에반게리온

B. 장르 선택 UI 및 필터링

영화 목록을 장르에 따라 필터링 하고 싶다. 장르를 선택할 수 있고, 선택된 장르의 영화만 출력하도록 하는 UI를 만들고 구현해보자.

- 요구사항 번호: F02
- 'div#movie-genre-filter' 내부에 구현
 - script.js에 정의된 genres의 데이터 활용
 - 장르 구분을 하지 않음을 의미하는 '전체'를 포함시킬 것
 - 장르를 선택하는 UI의 형식은 자유롭게 구성
- 영화의 장르 데이터의 경우 개별 영화의 'genres' 속성에 배열의 형태로 저장되어 있음
 - 해당 배열의 각 데이터는 각각 'genres' 배열의 동일한 값의 'id'를 가진 장르를 상징함
- 사용자가 장르를 선택할 경우 선택한 장르를 가진 영화만 목록에 출력되도록 구현
 - '전체'를 선택할 경우 장르와 상관없이 영화 목록 출력
 - 그 외 장르를 선택할 경우, 해당 장르를 가진 영화 목록을 출력

● 출력 예시

- 전체를 선택했을 때

Movies

Movie List

전체 ▾

- 가브리엘의 지옥 파트 2
- 가브리엘의 지옥
- Dedicada a mi ex
- 쇼생크 탈출
- 용감한 자가 신부를 데려가리
- 대부
- 쉰들러 리스트
- 나의 히어로 아카데미아 더 무비: 히어로즈 라이징
- 너의 이름은.
- 대부 2
- 센과 치히로의 행방불명

- 특정 장르를 선택했을 때

Movies

Movie List

액션 ▾

- 나의 히어로 아카데미아 더 무비: 히어로즈 라이징
- 반지의 제왕: 왕의 귀환
- 다크 나이트
- 스티븐 유니버스: 더 무비
- 7인의 사무라이
- 모탈 컴뱃 레전드: 스콜피온의 복수
- 저스티스 리그 다크: 아포콜립스 워
- 스파이더맨: 뉴 유니버스
- 스타워즈 에피소드 5: 제국의 역습
- 할복
- 반지의 제왕: 반지 원정대

C. 영화 생성 UI

영화를 목록에 추가하기 위한 UI를 만들어보자.

- 요구사항 번호: F03
- 'div#movie-form-section' 내부에 구현
 - 제목, 장르를 입력으로 받아야 함
 - 장르의 경우, 장르의 목록이 바뀔 수 있음을 염두에 두고 작성
 - 장르를 선택하는 UI는 자유롭게 구성
- 출력 예시

Movies

Movie List

전체 ▾

- 가브리엘의 지옥 파트 2
- 가브리엘의 지옥
- Dedicada a mi ex
- 쇼생크 탈출
- 용감한 자가 신부를 데려가리
- 대부
- 순들러 리스트
- 나의 히어로 아카데미아 더 무비: 히어로즈 라이징
- 너의 이름은.
- 대부 2

세 그림 카드로 이 채팅창 끝

Add Movie

제목

- 액션
- 모험
- 애니메이션
- 코미디
- 범죄
- 다큐멘터리
- 드라마
- 가족
- 판타지
- 음악
- 미스터리
- 로맨스
- SF
- TV 영화
- 스릴러
- 전쟁
- 서부

추가

D. 영화 데이터 추가

영화 추가를 위해 구현한 UI에 입력한 데이터를, 현재 영화 목록에 추가해보자.

- 요구사항 번호: F04
- 앞선 요구사항에서 구현한 UI의 기능 구현
 - 버튼 클릭이나 엔터 키 등을 통해 동작
 - 제목이 비었거나 장르가 선택되지 않은 경우 적절한 에러 메시지 표시
 - 추가가 되면 출력중인 영화 목록의 장르가 '전체'로 변경
 - 추가된 영화는 전체 영화 목록의 최상단에 보이도록 구현
- 추가된 영화는 브라우저 **새로고침 시 사라짐**에 유의

● 출력 예시

- 데이터 입력

Movies

Movie List Add Movie

전체 ▾

- 가브리엘의 지옥 파트 2
- 가브리엘의 지옥
- Dedicada a mi ex
- 쇼생크 탈출
- 용감한 자가 신부를 데려가리
- 대부
- 순들러 리스트
- 나의 히어로 아카데미아 더 무비: 히어로즈 라이징
- 너의 이름은.
- 대부 2
- 세고 카친코 이 채바브 며

범죄도시

액션
 모험
 애니메이션
 코미디
 범죄
 다큐멘터리
 드라마
 가족
 판타지

- 추가 결과

Movies

Movie List Add Movie

전체 ▾

- 범죄도시
- 가브리엘의 지옥 파트 2
- 가브리엘의 지옥
- Dedicada a mi ex
- 쇼생크 탈출
- 용감한 자가 신부를 데려가리
- 대부
- 순들러 리스트
- 나의 히어로 아카데미아 더 무비: 히어로즈 라이징
- 너의 이름은.
- 대부 2

제목

액션
 모험
 애니메이션
 코미디
 범죄
 다큐멘터리
 드라마
 가족
 판타지

- 장르 변경 - 해당 장르

Movies

Movie List

액션 ▾

- 범죄도시
- 나의 히어로 아카데미아 더 무비: 히어로즈 라이징
- 반지의 제왕: 왕의 귀환
- 다크 나이트
- 스티븐 유니버스: 더 무비
- 7인의 사무라이
- 모탈 컴뱃 레전드: 스콜피온의 복수
- 저스티스 리그 다크: 아포콜립스 워

스코리프트. 노이너스

Add Movie

제목

액션

모험

애니메이션

코미디

범죄

다큐멘터리

드라마

가족

판타지

- 장르 변경 - 다른 장르

Movies

Movie List

애니메이션 ▾

없음

- 나의 히어로 아카데미아 더 무비: 히어로즈 라이징
- 너의 이름은.
- 센과 치히로의 행방불명
- 스티븐 유니버스: 더 무비
- 모탈 컴뱃 레전드: 스콜피온의 복수
- 저스티스 리그 다크: 아포콜립스 워
- 스파이더맨: 뉴 유니버스
- 하울의 움직이는 성

바다를이아

Add Movie

제목

액션

모험

애니메이션

코미디

범죄

다큐멘터리

드라마

가족

판타지

E. 영화 삭제

영화 목록에서 영화를 삭제하는 기능을 구현해보자.

- 요구사항 번호: F05
- 영화 목록에 표시된 각 영화 옆에 구현
 - 디자인적 요소는 자유롭게 구성
- 삭제된 영화는 브라우저 **새로고침 시 다시 추가됨**에 유의
 - 단, 사용자가 추가한 영화는 다시 추가되지 않음

● 출력 예시

- 삭제 링크 추가

The screenshot shows a 'Movie List' section on the left and an 'Add Movie' section on the right. In the 'Movie List' section, there is a dropdown menu set to '전체'. Below it is a list of movie titles, each followed by a blue underlined '삭제' link. In the 'Add Movie' section, there is a '제목' input field and a list of genres with corresponding checkboxes.

제목	선택 여부	장르
가브리엘의 지옥 파트 2	X	액션
가브리엘의 지옥	X	모험
Dedicada a mi ex	X	애니메이션
쇼생크 탈출	X	코미디
용감한 자가 신부를 데려가리	X	범죄
대부	X	다큐멘터리
쉰들러 리스트	X	드라마
나의 히어로 아카데미아 더 무비: 히어로즈	X	가족
라이징	X	판타지
너의 이름은.	X	역사
대부 2	X	공포
센과 치히로의 행방불명	X	
기생충	X	
해밀턴	X	

- 삭제 이후

The screenshot shows the same movie list and add movie sections. The movie '가브리엘의 지옥' has been deleted, as indicated by the red box around its '삭제' link in the list.

제목	선택 여부	장르
가브리엘의 지옥 파트 2	X	액션
가브리엘의 지옥	X	모험
Dedicada a mi ex	X	애니메이션
쇼생크 탈출	X	코미디
용감한 자가 신부를 데려가리	X	범죄
대부	X	다큐멘터리
쉰들러 리스트	X	드라마
나의 히어로 아카데미아 더 무비: 히어로즈	X	가족
라이징	X	판타지
너의 이름은.	X	역사
대부 2	X	공포
센과 치히로의 행방불명	X	
기생충	X	
해밀턴	X	

The screenshot shows the movie list and add movie sections. The movie '가브리엘의 지옥' has been deleted, as indicated by the red box around its '삭제' link in the list. A red box also highlights the word '삭제됨' next to the deleted item in the list.

제목	선택 여부	장르
가브리엘의 지옥	X	액션
Dedicada a mi ex	X	모험
쇼생크 탈출	X	애니메이션
용감한 자가 신부를 데려가리	X	코미디
대부	X	범죄
쉰들러 리스트	X	다큐멘터리
나의 히어로 아카데미아 더 무비: 히어로즈	X	드라마
라이징	X	가족
너의 이름은.	X	판타지
대부 2	X	역사
센과 치히로의 행방불명	X	공포
기생충	X	
해밀턴	X	
우리는 그 투로 사랑해네	X	

3) 도전 과제

생성형 AI 도구를 활용하여 도전과제 요구사항을 해결해보자.

- 코드 생성, 아이디어 구상, 문제 해결 방법 탐색 등 다양한 방식으로 활용할 수 있다.
- 사용할 생성형 AI 서비스는 자유롭게 선택한다.
 - 제공되는 GPT API Key를 활용할 경우, 모델은 반드시 gpt-5-nano 모델을 사용한다.
- 최종 결과물은 AI 생성 내용을 바탕으로 직접 수정 및 개선하여 적용해본다.

보조 수단으로 활용하되, 능동적인 자세로 학습에 임할 것.

- 최종적인 이해와 적용은 자기 주도적 학습을 통해 이루어지며, 배운 내용을 스스로 기록하고 정리하며 학습 효과를 높일 것.

A. 즐겨찾기 기능 추가

영화 목록에서 원하는 영화를 즐겨찾기 하는 기능을 만들어보자.

- 요구사항 번호: F06
- 영화를 '즐겨찾기'에 추가할 UI 구현
 - 다른 UI와 어울리도록 자유롭게 구성
 - 사용자가 특정 영화가 '즐겨찾기'에 추가되었는지, 추가되지 않았는지 구분하기 쉽도록 구현
 - '즐겨찾기'에 추가된 영화의 경우 취소할 수 있도록 구현
- 장르 선택에 '즐겨찾기' 추가
 - 장르가 '즐겨찾기'로 변경될 경우, '즐겨찾기'에 추가된 영화로 영화 목록 출력
- '즐겨찾기'에 추가된 영화가 삭제될 시 '즐겨찾기'에서도 제거
- 장르가 '즐겨찾기'로 선택된 상태에서만, '즐겨찾기' 취소 시 현재 목록에서 영화 제거

● 출력 예시

- 즐겨찾기 UI

Movies

Movie List

전체

- 쇼생크 탈출
[삭제 즐겨찾기](#)
- 용감한 자가 신부를 데려가리
[삭제 즐겨찾기](#)
- 대부
[삭제 즐겨찾기](#)
- 친들러 리스트
[삭제 즐겨찾기](#)
- 나의 히어로 아카데미아 더 무비: 히어로즈
라이징
[삭제 즐겨찾기](#)
- 너의 이름은.
[삭제 즐겨찾기](#)
- 대부 2
[삭제 즐겨찾기](#)

Add Movie

제목

액션
 모험
 애니메이션
 코미디
 범죄
 다큐멘터리
 드라마
 가족
 판타지
 역사
 공포
 음악
 미스터리
 로맨스

- 추가된 경우

Movies

Movie List

전체

- 쇼생크 탈출
[삭제 취소](#)
- 용감한 자가 신부를 데려가리
[삭제 즐겨찾기](#)
- 대부
[삭제 취소](#)
- 친들러 리스트
[삭제 즐겨찾기](#)
- 나의 히어로 아카데미아 더 무비: 히어로즈
라이징
[삭제 즐겨찾기](#)
- 너의 이름은.
[삭제 취소](#)
- 대부 2
[삭제 즐겨찾기](#)

Add Movie

제목

액션
 모험
 애니메이션
 코미디
 범죄
 다큐멘터리
 드라마
 가족
 판타지
 역사
 공포
 음악
 미스터리
 로맨스

- 즐겨찾기 목록

Movies

Movie List

즐겨찾기

- 쇼생크 탈출
[삭제 취소](#)
- 대부
[삭제 취소](#)
- 너의 이름은.
[삭제 취소](#)

Add Movie

제목

액션
 모험
 애니메이션
 코미디
 범죄
 다큐멘터리
 드라마
 가족
 판타지
 역사
 공포
 음악
 미스터리
 로맨스

B. 상세보기 모달

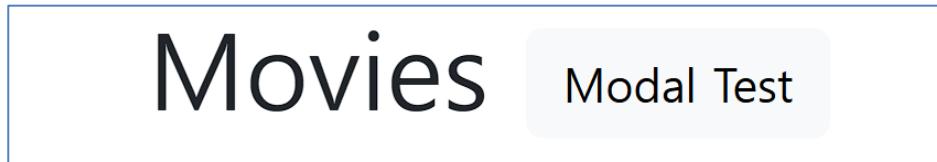
선택한 영화의 상세 정보를 확인할 수 있는 모달 창을 구현해보자.

- 요구사항 번호: F07
- 모달은 Bootstrap으로 구현되어 있음
- 주어진 index.html에 작성된 모달 및 모달 열기 버튼을 통해 모달의 모습과 동작 방식을 확인
 - 모달 열기 버튼의 경우 주석처리 되어 있음
- 출력된 영화 목록을 클릭할 시 모달 노출
 - 단, 영화 목록에 추가된 다른 기능을 위한 UI를 클릭할 경우 모달이 노출되지 않음
 - 이 때 클릭한 영화의 제목, 포스터 이미지, 줄거리가 모달에 채워져야 함
 - 단, 요구사항 3, 4로 인해 추가된 데이터의 경우 정상적으로 출력되지 않을 수 있음
- 모달의 Close 버튼이나 모달 바깥을 클릭할 시 모달 숨겨짐
- 그 외의 스타일 요소는 자유롭게 구성

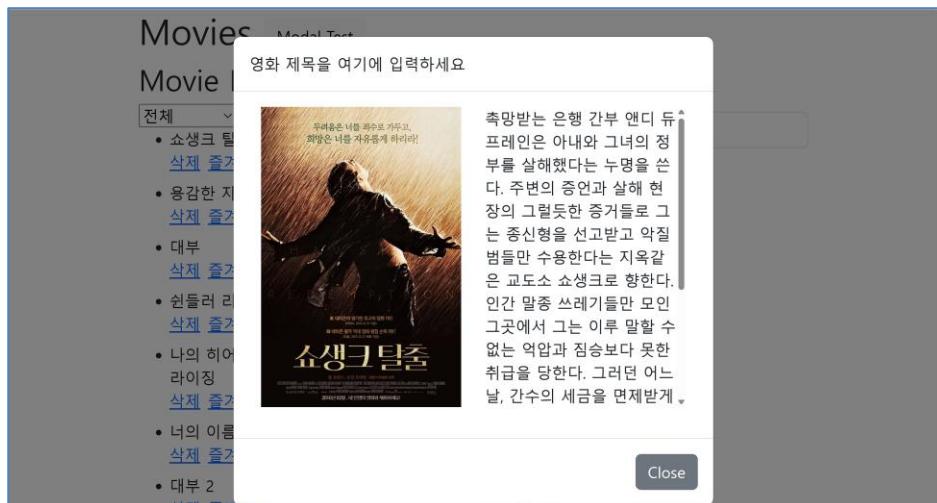
● 데모 모달 요소

- 모달 버튼

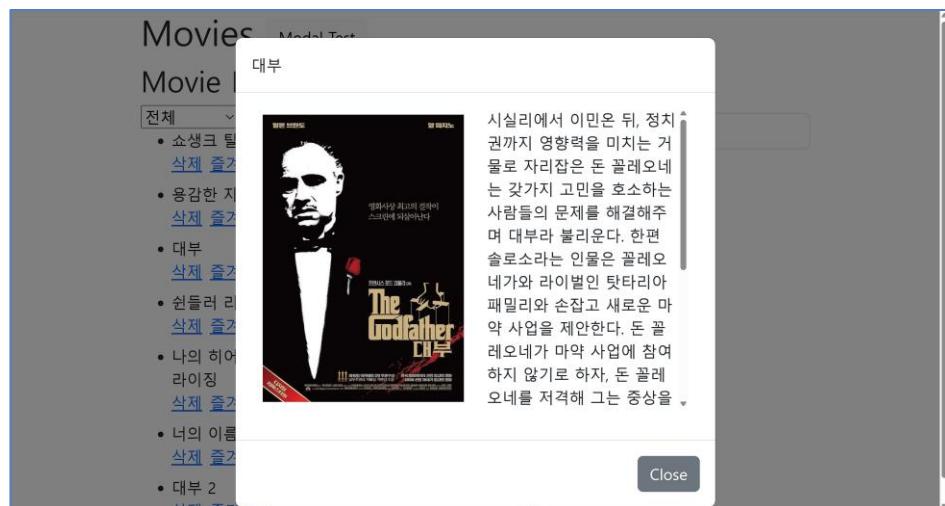
```
<div id="movie-container" class="container">
  <h1>Movies
    <!-- 모달 버튼 예시 -->
    <!-- <button id="show-modal-btn" class="class btn btn-light"
      data-bs-toggle="modal"
      data-bs-target="#movie-details-modal">Modal Test</button> -->
  </h1>
  <div class="row">
```



- 데모 모달 출력



● 출력 예시



5. 참고자료

- MDN
<https://developer.mozilla.org/en-US/>
- Bootstrap
<https://getbootstrap.com/>

6. 결과

제출 기한은 진행일 18시까지이므로 제출 기한을 지킬 수 있도록 한다. 제출은 GitLab을 통해서 이뤄진다.

- 산출물과 제출
 - 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 상세히 기록한 README.md
 - 완성된 각 문제 별 소스코드 및 실행 화면 캡쳐본
 - 프로젝트 이름은 08-pjt로 지정 및 각자 제출
 - 한 명의 GitLab 계정에 Git 저장소 Push 및 작업
 - 나머지 인원은 제출 시 해당 저장소를 Fork 하여 제출
 - 각 반 담당 강사님을 Maintainer로 설정

- 끝 -