1. 안녕하세요, 3기 3팀 최종발표를 맡은 이유진 입니다.
2. 저희 팀 프로젝트의 이름은 책무비입니다. 발표 순서는 다음과 같이, 팀 프로젝트의 목표를 먼저 설명하고, 시연영상을 보여드린 뒤, 각 파트의 프로젝트 결과에 대해 설명하겠습니다.
3. 저희 팀의 프로젝트 이름인 책 무비의 서비스 유형 및 목적에 대해 말씀드리겠습니다.
4. 먼저, 저희 프로젝트는 사용자가 선택한 영화의 취향을 바탕으로 책을 추천해주는 서비스를 만드는 것이 목표입니다. 책을 읽는 것이 부담되는 분들을 위해, 자신의 영화 취향으로 책을 추천해주는 서비스입니다.
5. 이제 본격적으로 각 파트의 프로젝트 진행 결과에 대해 발표하겠습니다. 가장 먼저, 디자인 팀입니다. 디자인팀은 김하늘님, 손다현님 두 분이 진행하셨습니다.
6. **먼저 저희 웹사이트의 메인 화면입니다. 메인화면에는 웹 페이지에 대한 전체적인 설명 뿐 만 아니라, 각종 통계로 갈 수 있는 바로가기가 있습니다.** 페이지 상단에 위치한 남색 바를 보시면, 3줄짜리 bar가 있는데, 이를 누르면 다른 페이지로 이동할 수 있는 바로가기를 디자인했습니다. 또한 중하단에 슬라이드를 만들어 신규영화들을 보여줄 수 있게 만들었습니다. 페이지 하단 footer에는 프로젝트에 대한 전체적인 기능을 한 눈에 볼 수 있게 했습니다. 이 footer는 모든 페이지에 있습니다.
7. 또한 책무비가 무엇인지 설명하는 페이지를 따로 추가하여 사용자가 웹페이지의 기획의도와 서비스에 대해 알아볼 수 있게 구현하였으며, 신규 영화들을 스와이프 형식으로 볼 수 있게 만들었습니다.
8. **책무비 서비스의 가장 중요한 부분인 영화 선택장면과 결과로 나온 책을 보여주는 페이지 입니다**. 영화 선택은 국내, 해외로 영화를 나누어서 사용자의 취향에 맞게 체크 버튼을 누르면 되고, 영화는 스와이프 형식으로 넘어갑니다. **선호하는 영화를 선택하고 나면 책과 함께 책의 줄거리가 결과화면으로 제시됩니다.**
9. 또한 책무비에는 개인별 맞춤 추천 시스템화면과 검색화면, 책 구매 사이트 연동 화면도 있습니다.
10. 각각의 화면을 자세히 설명하면 다음과 같습니다.
11. 저희 프로젝트는 사용자의 선호 영화와 책의 기록을 따로 저장하여 추천받았던 결과를 계속 볼 수 있게 했습니다. 책의 목록은 스와이프로 넘어가게 기획했습니다.
12. 사용자가 책무비에서 선택한 영화들을 모두 볼 수 있으며, 책무비 추천을 받은 책 들 중에서도 선호하는 책들을 추가로 볼 수 있습니다.
13. 저희는 통계 페이지도 기획했는데요, 크게 연령별 통계, 성별 통계, 실시간 검색어 순위로 나뉩니다.
14. 자세한 페이지 설명은 다음과 같습니다. 특히 연령별 통계와 성별 통계에서는 물음표 버튼을 누르면 화면에 대한 부가적인 설명을 확인할 수 있도록 설계하였습니다.
15. 로그인페이지와 회원가입 페이지도 만들어, 사용자가 선택한 영화와 추천받은 책 목록을 계속 볼 수 있도록 하였습니다.
16. 마이 페이지에서는 사용자의 전체적인 개인정보와 개인정보를 수정하는 부분을 만들었습니다.
17. 또한 마이페이지에서는 사용자가 최근 한 달 동안 책무비 홈페이지를 이용한 활동들을 기록한 공간을 만들었고, sns활동을 위한 페이지도 구현하였습니다.
18. 이로써, 디자인 파트의 발표를 마치고 ai파트로 넘어가겠습니다. 저희 ai파트는 박초은님, 신우진님, 이유진님, 장효원님이 맡았습니다.
19. **저희 ai파트의 구현과정은 다음과 같습니다. 먼저 네이버에서 영화 리뷰를, 알라딘에서 도서리뷰를 크롤링합니다. 그 뒤, 한글 형태소 패키지를 활용하여 크롤링한 리뷰 데이터들을 토큰화 한 뒤, 불용어를 제거하여 csv파일로 저장합니다. 이를 바탕으로 sql api를 구축하고, 영화와 책의 벡터 평균값을 구하고, 그 값과 가장 가까운 도서를 추천해주는 모델을 개발했습니다.**
20. Ai 파트에서는 다음과 같은 기술을 사용했습니다.
21. Ai 파트 구현 정도는 이와 같은데요, 데이터 크롤링은 90%정도, api구축은 75%정도 구현이 완료되었습니다.
22. 간단하게 ai파트의 알고리즘과 코드를 리뷰하겠습니다. 먼저 영화 리뷰를 크롤링하는 코드와 영화 리뷰를 토큰화 시키는 코드입니다. 영화 리뷰는 beautifulSoup를 사용하여 크롤링했습니다.
23. 마찬가지도 책 리뷰도 크롤링하고, 토큰화 했습니다. 알라딘 서재 페이지의 특징상, selenium을 사용하여 크롤링 하였습니다.
24. 유사도 검색에 용이하도록 토큰화된 데이터에서 한국어 불용어를 제거하였습니다.
25. 다음 코드는 api를 이용하여 영화 데이터와 도서 데이터를 데이터베이스에 저장하는 코드입니다. Flask를 사용했습니다.
26. Api를 이용하여 영화를 선택하면 책을 추천해주는 모델 또한 구현하였습니다. 영화들의 평균 벡터값을 구하고, 그 값과 가장 가까운 도서를 추천 해 주는 방식입니다.
27. 이로서 ai파트 결과를 마치고, 프론트엔드 파트에 대해 발표하겠습니다. 프론트엔드는 김도연님, 이유진님이 담당하였습니다.
28. 프론트엔드 구현과정은 다음과 같습니다. 먼저 디자인팀에서 보내주신 figma파일을 figma의 플러그인인 anima를 사용하여 html코드로 바꿔줍니다. 그 뒤, anima로 받아 온 코드의 오류를 고치고, javascript 등을 이용하여 반응요소들을 추가합니다. 스와이퍼 데모를 수정하여 필요한 스와이퍼를 구현하고 추가 한 뒤, 사이트 내 이동을 구현했습니다.
29. **프론트엔드 파트 발표내용의 변화는 다음과 같습니다.** 1,2차 발표에서는 html,css,javascript,swiper데모,jQuery와 비주얼 스튜디오를 사용하려고 했지만, 구현할 때에는 jQuery를 사용하지 않았고, 비주얼 스튜디오 코드를 사용했습니다. 또한 추가적으로 anima와 디스코드, 깃허브를 이용하였습니다.
30. 프론트엔드파트 구현 정도는 다음과 같습니다. 웹페이지는 95%정도 구현하였으며, 스와이퍼는 처음에 계획한 것과 달리 약간의 변화가 있었기에 75%정도 구현하였습니다.
31. **다음은 페이지 간 이동을 설명하는 코드들 입니다. 시간관계상 간단히 설명하고 넘어가도록 하겠습니다. 먼저, main화면의 코드들입니다.**먼저 main화면의 html파일은 01\_main\_origin.html파일로, 이 창에서 about checkmovie를 클릭하면 책무비를 설명하는 창으로 이동하게 됩니다.
32. **상단바에 있는 책무비, recommend, mybook, shop에 대한 코드들 입니다.** 상단바에 있는 책무비, recommend, mybook, shop에 대해 설명하겠습니다. 먼저 책무비를 클릭하게 되면 국내 영화와 해외 영화를 선택하여 추천을 받을 수 있습니다.
33. **사용자가 책을 추천받는 페이지는 다음과 같이 구성되어 있습니다**. recommend를 누르면 사용자의 추천 리스트를 사진, 제목, 설명과 함께 확인할 수 있습니다.
34. **mybook버튼을 누르면 왼쪽에 있는 버튼들의 내용이 달라지게 됩니다. 추천된 책들과 선택한 영화들, 사용자가 선호한 작품들을 확인할 수 있습니다**.
35. shop버튼을 누르면 세 가지의 온라인 북스토어에 접속할 수 있도록 구성하였습니다.
36. 다음 코드는 통계와 랭킹들을 확인 할 수 있는 코드들입니다.
37. 성별에 따른 인기순위와 실시간 검색어 순위를 확인하는 페이지의 코드는 다음과 같습니다.
38. main페이지에서 more movies를 클릭하게 되면 국내와 해외의 최근 영화들이 나오게 됩니다.
39. 다음은 my page에 대한 코드로, 사용자의 개인정보를 확인하고 수정할 수 있는 페이지로 연결됩니다.
40. 최근의 책무비 이용 내역을 저장하고 에스엔에스로 연결하는 페이지의 코드는 다음과 같습니다.
41. 마지막으로 로그인버튼과 회원가입을 할 수 있는 부분도 구현하였습니다.
42. 저희가 사용한 스와이퍼에 대한 간단한 설명을 하도록 하겠습니다. 먼저, 책무비의 가장 중심이 되는 영화를 선택하는 스와이프 입니다. 기존의 스와이퍼 데모에서 체크 표시 등을 추가하여 구현하였습니다. 원래 목표는 왼쪽으로 넘기면 불호, 오른쪽으로 넘기면 선호로 선택할 수 있는 스와이프를 구현하려고 했으나, 구현 상 한계로 체크표시를 추가하였습니다.
43. 영화 및 책을 보여주는 스와이프 입니다. 선택된 영화 및 책을 보여주는 스와이프는 기존의 스와이퍼 데모에서 옆의 코드를 추가하여 구성하였습니다. 기본 스와이퍼는 swiper라는 클래스이름을 가지고, 가운데에 오는 스와이퍼는 swiper-slide-active라는 이름으로 클래스 이름이 변화하는 것을 이용하였습니다.
44. 이로써 프론트앤드 파트 발표를 마치고, 마지막으로 백엔드 파트의 개발과정을 설명하도록 하겠습니다. 백엔드는 주현정님과 최주리님이 담당하셨습니다.
45. 백엔드에서는 스프링부트와 mySQL을 사용하였고, 프로젝트 구조는 다음과 같이 이루어져있습니다.
46. 데이터베이스 테이블은 다음과 같이 책, 책무비, 영화로 이루어져 있습니다.
47. 먼저 entity는 테이블과 1:1로 매칭하였습니다. 추천될 책, 체크한 영화, 전체 영화로 나누었습니다. 책과 영화 테이블에는 id, title, img등이 저장되어 있습니다.
48. DTO또한 앞서 말한 것과 같이 추천될 책, 체크한 영화, 전체 영화로 나누었습니다. Entity로 보내주는 toEntity함수도 추가되어 있습니다.
49. Repositiory는 entity를 통해 데이터를 데이터베이스에 저장하는 곳 입니다. Repositiory도 마찬가지의 3가지로 나누어져 있습니다.
50. Service는 savebook 함수를 통해 book의 entity내용을 book repository로 저장하고, savemovie함수를 통해 체크한 영화가 checkmovie로 저장되도록 구현하였습니다.
51. Controller는 main 컨트롤러와 무비 컨트롤러로 나누었습니다. Main controller는 가장 main 화면으로 요청하였으며, 전체 영화의 데이터베이스를 리스트를 사용하여 모델로 넘겨줍니다.
52. 무비 컨트롤러는 여러 부분으로 나뉘어져있습니다. 그 중 결과인 추천 부분입니다. ai모델을 올려서 나온 추천 결과의 id를 넘겨주고 이를 통해 title과 책의 이미지를 얻어내도록 구현하였습니다.
53. 자신의 기록을 저장하는 mybook부분은 다음과 같습니다. Repository에 있는 내용을 전부 찾아서 list로 만든 후에 이 list로 넘겨주도록 구현하였습니다.
54. 영화를 체크하는 부분은 다음과 같습니다. 저희가 가지고 있는 전체 영화를 repositiory에서 찾아 list로 넘겨준 후, 체크한 영화를 savemovie로 넘겨서 checkmovie repository에 저장되게 구현하였습니다.
55. **시연영상을 시청하기 전에, 아직 user와 관련된 부분과 시각화 부분의 구현이 되지 않았습니다. 발표가 끝나고, 추후에 구현 예정입니다. 감안하고 시연영상을 봐 주시길 바랍니다. <시작하자마자>**먼저 main부분의 모습과 영화를 선택하면 책을 추천하는 부분의 모습입니다. <책 결과 나오면> 선호하는 책을 선택할 수 있게 되어있으며, user부분은 추후 구현 예정입니다.
56. 저희의 발표 내용은 여기까지 입니다. 경청해주셔서 감사드립니다. 혹시 질문이 있으시다면 질문부탁드립니다.