<팀 그릴그린 - 캡스톤 디자인 최종발표 대본>

|  |
| --- |
| 인사말 / 목차 / 팀 구조 |
| 팀 그릴그린의 팀장 오혜진입니다. 발표를 시작하겠습니다.  목차는 팀 구조, 프로젝트 설명, 산출물 소개, 결과물 및 테스팅 순서입니다.  그릴그린은 프로젝트별 조직이라고 가정합니다. 각자 맡은 역할은 오혜진 백엔드, 천세륜 프론트엔드, 오규진 빅데이터 분석 및 시각화입니다. |

|  |
| --- |
| CorpCollector란? / 프로젝트 기획 배경 / 프로젝트 기대 효과 / 사용한 HW, SW 목록 |
| CorpCollector(코프콜렉터)는 저희 팀이 기획한 프로젝트로, 소비자 관점에서 선한 기업 정보를 제공하는 웹 사이트입니다. 선한 기업의 예시로는 사회적 기업, 녹색 기업 등이 있습니다. 이런 기업 목록은 공공데이터 포털에서 파일 데이터, 또는 API로 제공합니다. 일반 사용자들이 보기에는 가독성이 떨어지는 부분을 고려해 웹으로 만들었습니다. 또한 데이터 분석 자료도 제공해서 자료의 질을 높였습니다.  이 서비스를 기획하게 된 배경을 설명하겠습니다. 특정 기업들의 논란으로 인해, 소비자들의 불매 행위가 불거지면서 해당 기업들의 매출이 하락하는 증세를 보입니다. 이는 부도덕한 행위를 한 기업 제품보다는 다른 기업 제품을 소비하자는 의견이 있었기 때문입니다. 따라서 선한 기업 목록을 CorpCollector에서 제공하여 다음과 같은 기대 효과를 얻고자 합니다.  웹서비스이기 때문에 사용자 접근성이 좋을 것입니다. 따라서 소비자가 쉽게 사회에 유익한 기업을 찾을 수 있습니다. 특정 소비자 층, 즉 자신의 가치관에 맞는 제품을 선택해 소비하는 ‘가치소비’자 층의 지출을 지향합니다. 이로써 기업의 사회 기여를 촉진하는 효과도 기대해볼 수 있습니다.  사용한 하드웨어는 개인용 개발 PC 3대, 소프트웨어는 깃허브, 태블로, DBMS, 이클립스 외 여러 툴이 있습니다. WAS(웹 애플리케이션 서버)는 아파치 톰캣을 사용했습니다. |

|  |
| --- |
| 퍼트 차트 & 간트 차트 / 사이트맵 구조 |
| 이어서 프로젝트 산출물을 소개하겠습니다.  전체 개발 주기를 표현한 퍼트 차트입니다. 사이트 맵 구성 이후에는 각 파트에 따라 설계 작업을 진행했습니다. 이후에는 약 1주일 단위로 개발/보완/수정 주기를 거치며 작업했습니다.  원래라면 4차 주기에서 개발이 끝나고 최종 발표 준비에 들어가야 하지만, 11월 초에 마감 기간이 늘어났기 때문에 유지보수와 추가 작업을 오래 진행했습니다. 먼저, 서비스 소개 페이지에 하위 메뉴별 기능 설명을 추가했습니다. 또한 개발 과정 페이지에 설계서 이미지를 올렸습니다. 관심 기업 페이지에는 전체 선택 말고도 목록을 초기화할 수 있는 기능을 추가했고, 현재 페이지 번호를 표시했습니다. 호스팅 적용 완료 후에는 업체에서 제공한 아웃룩 계정으로 이메일 인증 기능을 추가했습니다.  11월 초반 이후 일정까지 추가한 간트 차트는 이런 모습입니다. 4차 주기에서 백엔드 마무리 작업이 늦어졌지만, 이후 일정은 원활하게 진행되었습니다.  사이트 맵은 총 5가지 주메뉴로 이루어져 있습니다. 주 메뉴 카테고리로는 서비스 소개, 기업 찾기, 정보나눔, 커뮤니티, 마이페이지, 로그인 및 회원가입이 있습니다. 상세한 설명은 다음 챕터에서 설명하겠습니다. |

<프론트엔드>

|  |
| --- |
| 프론트엔드 |
| 이제 결과물 및 테스트 결과를 설명하겠습니다. 먼저, 프론트엔드 파트입니다. 프론트엔드 담당자 천세륜은 이클립스 개발 환경에서 jsp, js, css 등을 이용해 사이트를 구현했습니다. |

|  |
| --- |
| 화면 구성 소개 |
| 인덱스 페이지부터 간략하게 설명하겠습니다. 인덱스 페이지의 가운데에 ‘CorpCollector’ 로고와 검색바를 배치하여 사이트의 메인 기능인 검색 서비스를 빠르게 이용할 수 있도록 했습니다.  서비스 소개의 하위 메뉴인 ‘서비스 개요’에는 사이트 제작 배경, 각 메뉴 설명이 있고 ‘개발 과정’에는 각 개발 파트에 대한 간략한 설명과 설계도 등의 산출물이 있습니다.  다음으로 상위 메뉴인 ‘기업 찾기’에서는 기업 종류에 상관없는 통합 결과가 나타납니다. 사이드 바 하단의 기업 순위 정보를 참고하여 검색에 도움을 받을 수 있습니다.  기업 종류를 선택하면 선택한 기업 종류에 해당하는 리스트가 나타납니다. 예시로 ‘녹색 기업’을 클릭하면, 녹색 기업 리스트만 나타나고 녹색 기업 내의 검색 결과가 출력됩니다.  기업 리스트 중 하나를 선택하면 상세 정보 페이지로 이동하고 로그인 한 유저는 관심 기업 등록을 할 수 있습니다.  관심 기업으로 등록한 기업은 채워진 별 표시가 되고 목록의 별을 직접 클릭해서 관심 기업 등록, 삭제도 가능합니다.  정보 나눔 메뉴의 ‘기업 데이터 분석’에서는 기업 종류별로 데이터를 분석하여 그래프로 나타내 사용자에게 정제된 정보를 전달합니다.  다음 하위 메뉴인 ‘기업 기사 모음’은 기업들의 최근 기사를 리스트로 정리한 페이지로 링크를 클릭하면 기사 원문 사이트로 이동합니다.  ‘커뮤니티’ 메뉴를 설명하겠습니다. 공지사항 게시판에는 관리자만 글을 쓸 수 있으며, 그 외에는 고객후기 게시판과 같은 기능을 가집니다. 공개 글은 모든 유저가 볼 수 있고 비공개 글은 작성자와 관리자만 확인할 수 있습니다.  모든 글 작성 페이지는 로그인 한 사용자만 이용 가능하며 체크박스로 글과 작성자 공개/비공개를 설정할 수 있습니다.  마지막 메뉴인 ‘마이페이지’는 기업과 관련된 두 가지 하위 메뉴만 설명하겠습니다.  먼저, ‘관심 기업’ 페이지에서는 ‘기업 찾기’에서 관심 기업으로 등록한 기업 목록을 확인할 수 있습니다. 기업 종류별로 분류되어 나타나며 체크박스를 이용해 선택 기업을 관심 기업에서 삭제할 수 있습니다.  ‘최근 검색 기업’ 페이지는 로그인 한 사용자가 ‘기업 찾기’에서 상세 정보를 확인한 최근 50개 기업 목록을 보여줍니다. |

<백엔드>

|  |
| --- |
| 백엔드 |
| 백엔드 파트입니다. 백엔드 담당자 오혜진은 이클립스 개발 환경에서 웹 사이트의 기능 구성 및 구현을 진행했습니다. 해당 파트는 웹 사이트 구조, 클래스 다이어그램 등으로 이루어져 있습니다. |

|  |
| --- |
| 웹 구조 |
| 우선, CorpCollector의 웹 구조를 말씀드리겠습니다. 크게 클라이언트, 웹 애플리케이션 서버, 데이터베이스가 있습니다. 클라이언트, 브라우저가 컨텐츠를 요청하면 웹 서버가 정적인 컨텐츠를 제공하거나 요청을 웹 컨테이너에 보냅니다. 이 웹 컨테이너를 JSP, Servlet으로 사용했고, 웹 애플리케이션 서버는 아파치 톰캣을 사용했습니다. 웹 컨테이너는 데이터베이스에서 데이터를 가져와서 비즈니스 로직에 맞는 결과를 만들어 제공합니다. 여기에 사용된 데이터베이스는 MariaDB입니다. |

|  |
| --- |
| 디자인 패턴 |
| 백엔드 작업을 하면서 사용한 디자인 패턴은 MVC 패턴과 프론트 컨트롤러, 커맨드 패턴입니다. 모델, 뷰, 컨트롤러를 분리해 관리가 편하도록 설정했습니다. 모든 요청을 프론트 컨트롤러로 보내서 URI 매핑을 통해 어떤 커맨드를 실행할지 결정합니다. URI 매핑은 일반적으로 많이 쓰이는 \*.do(닷 두) 패턴을 썼습니다.  커맨드 패턴은 프론트 컨트롤러 서블릿이 직접 요청을 처리하지 않고, 커맨드 클래스가 처리하게 해줍니다. 이는 요청 개수가 많아질수록 코드가 복잡해지는 걸 방지할 수 있습니다. 요청에 해당하는 커맨드를 실행해서 DAO(Data Access Object)로 데이터베이스에 접근합니다. 이때, DTO(Data Transfer Object)를 생성해 프로세스 간에 데이터를 전달하기도 합니다. 이러한 MVC 모델을 사용하면 용도에 따라 코드를 분리할 수 있고, 따라서 중복코딩을 최소화할 수 있습니다. 또한, 유지보수성과 유연성이 향상하고, 프론트엔드 작업자도 뷰, 화면 인터페이스만 보고 작업할 수 있으므로 팀 작업에 적합하다고 판단했습니다. |

|  |
| --- |
| 클래스 다이어그램 |
| 관리 커맨드들은 MemberDAO를 호출합니다. 회원가입 시에 아이디 중복 방지를 위해 아이디 체크 커맨드를 사용합니다. 사용자가 로그인하고자 로그인 커맨드를 실행하면, 현재 사용자의 정보를 담은 MemberDTO가 하나 생성된 후, 세션에 저장됩니다. 비밀번호 찾기 이후에는 비밀번호를 수정하는 커맨드가 실행됩니다. 그 외에는 회원 관리에 필요한 기본적인 커맨드들입니다.  다음으로, 이메일 인증 기능을 보겠습니다. 회원가입에 기재한 이메일의 계정주만 개인 정보에 접근할 수 있게 해주는 기능입니다. 이메일 관련 커맨드는 ServerLogin 클래스를 통해 보낼 사람을 설정하고 인증 코드를 적재한 뒤, 이메일 체크 커맨드에서 사용자가 입력한 값과 비교합니다.  기업 찾기 클래스 다이어그램은 기업유형별 DAO, DTO를 다 넣으면 너무 길어져서 예시로 녹색 기업, 통합기업 관련 클래스만 다루겠습니다. 기업 유형만 다르고 기본적인 컨트롤 로직은 비슷하므로, FindCorpCommand라는 한 커맨드에서 기업유형별 검색을 수행할 수 있도록 구성했습니다. 기업 DAO와 DTO를 인터페이스로 작성한 이유입니다.  커뮤니티 클래스 다이어그램입니다. 각 커맨드들은 request로 받는 파라미터 값에 따라 PostDAO 인터페이스를 실체화한 DAO클래스가 달라집니다. 또한, 커뮤니티 게시판에서도 검색 서비스를 제공하기 있으므로, 기업 찾기처럼 특정 게시글 리스트를 가져오는 메소드가 존재합니다.  기업 기사 모음 페이지에서 기사 리스트를 불러올 때 정보 나눔 관련 커맨드가 사용됩니다. 기업 데이터 분석은 jsp파일에서 그래프 정보를 불러와서, 컨트롤러 방면에서 처리하지 않았습니다.  마이페이지에는 여러 테이블이 복합적으로 사용되므로 커맨드별 쓰이는 DAO가 다양합니다.  여태 설명이 적었던 관심 기업 DAO와 최근 검색 기업 DAO 및 연관 커맨드를 설명하겠습니다.  관심 기업 추가 커맨드에서는 먼저 로그인 여부를 따지고, 로그인했다면 관심 기업 등록 여부를 확인합니다. 등록이 안 되어 있으면 DB에 적재하고, 이미 기록이 있으면 삭제합니다. 기업 찾기 페이지에서 스크랩 버튼, 별 모양을 클릭하거나 상세 정보 페이지에서 관심 기업 등록 버튼을 토글 형식으로 사용할 수 있습니다.  최근 검색 기업 테이블에는 상세 정보를 본 기업 데이터가 추가됩니다. 이때, 한 아이디 당 과거 기록을 볼 수 있는 레코드 개수는 50개로 제한합니다. 개수 관리를 도와주는 프로시저는 화면과 같습니다. 현재 사용자 아이디를 IN 매개변수로 받아, 사용자가 최근 검색 후 상세 정보를 본 기업 개수를 변수에 저장합니다. 이후 50과 비교하여 그보다 이상이면 가장 오래된 날짜의 기록을 지웁니다. |

<빅데이터>

|  |
| --- |
| 빅데이터 |
| 마지막 부 챕터, 빅데이터 파트입니다. 빅데이터 파트 담당자 오규진은 데이터 분석 및 시각화, 기사 크롤링을 진행했습니다. 내용은 데이터 전처리 및 시각화 과정 등으로 구성되어 있습니다. |

|  |
| --- |
| 데이터 프로세스 구조 |
| 데이터 프로세스 구조를 설명하겠습니다.  기업 데이터는 공공데이터 포털에 있는 기업 데이터를 사용했습니다.  프로세스 순서는 다음과 같습니다. 파이썬으로 open api를 사용하여 공공데이터 포털에서 데이터들을 받습니다. 이때 데이터 형식은 CSV, XLS형식입니다.  데이터들의 raw데이터들을 sql에 적재할 수 있도록 데이터 전처리를 진행합니다. 그 후, 데이터들을 적재하기 위해 파이썬으로 pymysql(파이마이sql), sqlalchemy(sql알케미)를 사용하여 sql에 연결, 데이터를 적재하였습니다.  기업 기사는 크롤링을 사용하였습니다. 크롤링한 데이터들을 데이터베이스 설계의 형식에 맞게 전처리 후, 마찬가지로 파이썬으로 sql과 연결하여 적재하였습니다. |

|  |
| --- |
| 데이터 전처리, 적재 과정(코드) |
| API 요청과 크롤링 등으로 데이터를 수집하고 DB에 적재하기 위해선 전처리 과정이 필요합니다. 기업 데이터의 Raw 데이터는 Null값도 있었고, 업체의 중복 데이터, 지역 구분이 제대로 되지 않은 데이터도 있는 등, 바로 DB에 적재하기엔 부적합한 데이터가 많았습니다. Null값과 중복 데이터는 직접입력 하거나 삭제 등으로 결측치 처리하였고, 지역 구분을 정확하게 하기 위한 텍스트 전처리도 진행했습니다.  [데이터 적재 코드로 넘어가며]  데이터 적재는 아까 말씀드린 것처럼 파이썬 pymysql(파이마이sql), sqlalchemy(sql알케미) 라이브러리를 사용하여 SQL에 연결 후, 각 카테고리 기업 테이블에 데이터 적재하였습니다. |

|  |
| --- |
| 데이터 시각화-프로세스 |
| 데이터를 시각화하는 툴(Tool)은 Tableau(태블로)를 사용했습니다.  태블로를 선정한 이유는 총 3가지입니다. 첫 번째로 다양한 시각화를 사용할 수 있어서 편의성이 뛰어납니다. 두 번째로 시각화 화면끼리 상호작용할 수 있는 인터렉티브 시각화 기능을 가지고 있습니다. 마지막으로, 동적 그래프이기 때문에 데이터 표현성이 더 높다고 판단했습니다.  프로젝트에서는 태블로를 DB와 직접 연동하여 시각화했습니다. 직접 DB와 연결하여 데이터가 수정되거나 갱신되면 실시간 시각화 업데이트가 진행되기 때문에, 이 방법을 사용했습니다. |

|  |
| --- |
| 데이터 시각화-대시보드 |
| 데이터 시각화 대시보드 구성입니다.  태블로는 여러 그래프를 만들어 하나의 대시보드로 표현할 수 있습니다.  구성된 그래프를 설명하겠습니다. 첫 번째 그래프는 기업들의 업종 통계를 나타내는 그래프이고 트리 그래프로 표현하였습니다.  [다음 장면으로 넘어가며]  두 번째 그래프는 지역별로 기업의 업종 통계를 나타내는 그래프이고 누적 막대그래프로 표현하였습니다.  [다음 장면으로 넘어가며]  세 번째 그래프는 해당 카테고리에 선정된 기업들의 숫자를 나타낸 그래프이고 막대 그래프로 표현하였습니다.  네 번째 그래프는 기업의 지역 분포를 숫자로 표현하는 시각화이고 해당 그래프는 표로 표현하였습니다.  데이터는 기술 통계를 사용하여 데이터를 표현하였습니다. 데이터가 적어 추론 통계 등 다양한 통계를 사용하는데 제약이 있었습니다. |

|  |
| --- |
| 기사 모음(크롤링) 과정 |
| 기업 기사를 크롤링하는 과정입니다.  네이버 뉴스에서 기업 기사를 크롤링하여 DB에 적재했습니다. 크롤링하는 과정에서 파이썬 beautifulsoup(뷰티풀숲), request라이브러리를 사용하여 크롤링을 진행했습니다. 해당 카테고리의 기업을 검색어로 받고 뉴스의 시작, 끝 날짜를 받은 뒤 해당 뉴스들을 크롤링하도록 만들었습니다. |

|  |
| --- |
| 맺음말 |
| 이상으로 그릴그린의 웹 프로젝트 발표였습니다. 기타 자세한 프로젝트 작업물은 아래 첨부한 사이트로 들어가서 볼 수 있습니다. 감사합니다. |