**포트폴리오**

**신입 웹프로그래머 이선제**

**Github URL -** <https://github.com/LeeSeonJe>

**세미 프로젝트**

COSMEDIC (화장품 정보 제공 및 리뷰, 피부과 정보 제공 및 리뷰, 피부과 예약 서비스 사이트)

|  |  |  |
| --- | --- | --- |
| **개발 목표** | 웹 크롤링을 이용해 많은 화장품과 브랜드 정보를 저장하는 데이터베이스를 활용하여 화장품 별 사용자 리뷰를 통해 정보를 공유하고 병원 예약 시스템까지 활용하여 피부에 대한 고민이 많은 고객층을 대상으로 한 사이트 구축 | |
| **개발 기간** | 2020년 03월 13일 ~ 2020년 04월 28일 | |
| **개발 환경** | |  |  | | --- | --- | | 운영체제 | Windows OS | | 개발도구 | Eclipse | | DBMS | Oracle DB – Sqldeveloper | | Servers | Apache Tomcat 8.5 | | Language | Java, HTML5, CSS3, JavaScript, JQuery, Servlet | | Front-Framework | Bootstrap | | API, Library | Naver Login API, kakao Map API, FullCalendar API, javax-mail API, Postcodify API, jsoup Library | | 형상 관리 시스템 | Git | | |
| **개발 인원** | 6명 | |
| **구현 기능** | **-** 메인, 화장품, 브랜드, 병원, 고민 탭을 통하여 페이지 접근 가능  **-** 화장품 카테고리 별 기본 랭킹 순, 필터를 통해 좋아요 순, 평점 순, 피부타입, 연령대 등 많은 필터 적용 가능  **-** 화장품과 브랜드는 웹 크롤링을 통해 다량의 데이터 삽입  **-** 브랜드 검색 필터를 통해 자음 별 검색 가능  **-** 병원 기본 랭킹 순, 리뷰 많은 순, 지역별 검색 필터 등 다양한 필터 적용 가능  **-** 화장품, 병원 랭킹 순은 자체적으로 제작한 랭킹 알고리즘 (해당 제품 평가 수 / 전체 제품 중 가장 많은 평가 수 \* 해당 제품 평점) 적용  **-** 병원 페이지에 가입시 등록한 이미지 자동 슬라이드를 통해 노출 및 지도API를 이용하여 지도 노출 및 카카오 로드 뷰 기능 제공  **-** 병원 마이페이지에서 캘린더API를 통해 예약자 현황 확인 가능  **-** 고민 탭에 전체 게시글 및 베스트 고민 (조회순) 확인 가능  **-** 게시글 별 댓글 작성 가능  **-** 사용자 및 관리자와 실시간 채팅 기능 지원  **-** 관리자는 모든 회원과 병원 정보를 확인, 수정, 탈퇴 시킬 수 있으며 병원 회원은 기본 가입 시 관리자의 승인이 있어야 지만 회원가입 가능  **-** 관리자는 리뷰 관리, 삭제, 화장품 등록 요청 확인 및 화장품 정보 등록, 게시판 관리, 1대1 문의 관리를 관리자페이지를 통해 가능  **-** 네이버 로그인 API를 통해 간편 가입 및 로그인 가능 | |
| **DB 설계** |  | |
| **담당 역할** | 전체적인 레이아웃 및 기능 관리,  화장품 랭킹페이지, 화장품 상세 페이지, 마이페이지 | |
| **구현 화면 및 설명** | | |
|  | | **마이페이지 – 내정보관리**   * 프로필을 등록, 수정 * 개인 정보 수정 * 비밀 번호 변경 * 회원 탈퇴 |
|  | | **마이페이지 – 내 리뷰**   * 회원이 작성한 리뷰 조회 * 리뷰 제목 클릭 시 리뷰 상세 페이지로 이동, 수정, 삭제 가능 |
|  | | **마이페이지 – 내 게시글**   * 회원이 작성한 게시글 조회 * 게시글 제목 클릭 시 게시글 상세 페이지로 이동, 수정, 삭제 가능 |
|  | | **마이페이지 – 내 예약**   * 피부과 예약 조회 * 피부과에서 예약을 거절하기 전 예약 수정 가능 |
|  | | **마이페이지 – 1대1 문의**   * 1대1 문의 조회 * 문의하기를 통해서 문의 가능 * 답변이 왔을 시 답변 보기로 답변 확인 |
|  | | **화장품 페이지 – 랭킹**   * 자체적인 랭킹 계산법을 통한 화장품 카테고리별 랭킹 표시 * 필터 설정 시 리뷰에 따른 랭킹 재설정 |
|  | | **화장품 상세 페이지**   * 화장품 상세 내용 * 화장품에 따른 리뷰 제공 * 리뷰 필터 적용 시 필터에 대한 리뷰 표시 * 리뷰 키워드 검색 가능 |
|  | | **브랜드 페이지**   * 브랜드 목록 * Ajax를 통한 무한 스크롤 * 자음에 따른 브랜드 필터 * 브랜드 상세로 이동할 경우 브랜드의 화장품 랭킹 표시 |
|  | | **브랜드 상세 페이지**   * 브랜드의 화장품 랭킹 표시 * 필터 설정 시 리뷰에 따른 랭킹 재설정 |

|  |  |
| --- | --- |
| **주요 코드 리뷰** | |
| **Jsoup Library** | |
|  | **Jsoup library를 통한 웹 크롤링**  크롤링 하기 위한 페이지에서 개발자 도구를 통해 값이 들어 있는 최상위 Elements의 클래스명을 가져옵니다.  가져온 후 최상위 안에 자식들의 클래스 명을 Iterator을 통해서 값을 하나씩 추출한 후 cosmetic 객체에 값을 넣어줍니다.  bufferedSave() 메소드에 cosmetic 객체를 넘겨줍니다.  파일 입출력을 통해서 SQL파일을 생성 후 안에 데이터 베이스에 저장하기 위한 INSERT문을 만들어 바로 저장시킵니다. |

|  |  |  |
| --- | --- | --- |
| **화장품 랭킹 동적 SQL 쿼리문** | | |
|  | | **필터에 따른 동적 SQL 쿼리**  화장품 페이지의 필터  브랜드 상세 페이지의 필터  적용 시 필터의 조건은 크게 3분류로 나뉘며 각 15개의 조건이 존재하게 됩니다.  리뷰를 등록한 사용자의 정보와 별점에 따라 랭킹이 설정되며 랭킹을 설정하기 위한 계산식은 ( 해당 화장품 평가 수 / 해장 화장품 카테고리 중 가장 많은 평가수 \* 해당 화장품 평점 ) 으로 계산됩니다.  파라미터 값을 받아 이와 같은 동적 쿼리 작성 부분을 거치게 되어 SQL 쿼리문을 만들어 주고 이를 통해서 결과 값을 얻어 랭킹을 보여줍니다. |
|  | | **필터에 따라 만들어진 SQL 구문** |
| **Kakao Map API** | | |
|  | **해**  **지도API - 병원 위치 표시하기**  카카오맵 API를 사용하기 위해 CDN방식으로 연결을 해줍니다  카카오 디벨로퍼에서 제공하는 자신만의 코드를 사용하여 dev 코드와 필요한 service lib 코드를 CDN 주소 안에 넣어줍니다.  병원 회원 가입 시 병원의 주소를 도로명으로 입력 받게 됩니다.  addressSearch 객체를 생성 후 addressSearch() 메소드를 통해서병원 주소를 파라미터 값으로 넘겨주면 카카오맵에 해당 위치의 마커가 찍히게 됩니다. | |
| **소감** | | |
| Jsp/servlet을 통한 프로젝트 진행하게 되었는데 짧은 기간 안에 배우면서 화면도 만들고 기능도 넣어야 했던 부분이 많았는데 팀원들이 잘 도와줘서 좋게 마무리 할 수 있던 것 같습니다.  전공자이지만 친구들과 프로젝트 했던 것과는 다른 느낌을 많이 받았습니다.  좀 더 체계적으로 기획, 설계, 구현을 한 것 같습니다.  이번 프로젝트는 기본적인 CRUD보단 새로운 것에 대해 알아보고 입혀보고 싶은 마음이 컸습니다.  파이널 때는 좀 더 기본기에 충실한 페이지를 구현해 보고 싶습니다. | | |

**파이널 프로젝트**

Bucket Story(버킷리스트 공유사이트)

|  |  |  |
| --- | --- | --- |
| **개발 목표** | 자신의 버킷리스트를 작성, 공유하고 버킷리스트를 이뤄가는 과정의 이야기를 쓰는 버킷리스트 관리사이트 | |
| **개발 기간** | 2020년 04월 29일 ~ 2020년 06월 30일 | |
| **개발 환경** | |  |  | | --- | --- | | 운영체제 | Windows OS | | 개발도구 | Eclipse | | DBMS | Oracle DB – Sqldeveloper | | Servers | Apache Tomcat 8.5 | | Language | Java, HTML5, CSS3, JavaScript, JQuery, Servlet | | API, Library | Kakao Map API, 도로명 주소 API, javax-mail API, web socket, JqueryUI – Autocomplete(search Autocomplete), jsoup Library,  NAVER smartEditor2.0 | | 형상 관리 시스템 | Git | | |
| **개발 인원** | 6명 | |
| **구현 기능** | **로그인, 회원가입**  - 이메일 인증, 아이디닉네임 중복확인, 비밀번호 암호화, 아이디비번찾기. 회원가입, 기업회원가입, 회원로그인, 기업로그인, 로그아웃  **메인페이지, 인포페이지**  - 최신순, 인기순, 추천 버킷리스트 뷰, 가입기업, 좋아요, 위시등록, 공유하기, 버킷디테일, 자동완성 검색창, 내정보변경, 웹소켓 실시간알람  **회원 마이페이지**  - 프로필 사진 변경, 나의 버킷리스트, 위시리스트, 버킷스토리 보기, 블로그식 작성, 회원팔로우, 스토리 좋아요, 댓글답글 작성, 게시글댓글답글 신고  **기업 마이페이지**  - 기업소개변경, 기업정보변경, 지원버킷추가, 버킷행사추가, 버킷견적서요청, 버킷견적서전송, 카카오페이 포인트충전, 포인트내역조회  **관리자 페이지**  - 축제 작성, 카카오맵API, 주소API, 기업승인, 포인트결제내역, 신고게시판, 경고주기, 강제탈퇴시키기, QnA 질문하기와 답변하기 | |
| **DB 설계** |  | |
| **담당 역할** | 회원 블로그페이지, 회원가입 이메일 인증, NAVER smartEditor 2.0를 적용한 게시판 페이지 | |
| **구현 화면 및 설명** | | |
|  | | **블로그페이지 – 버킷리스트**  블로그 페이지에 들어오게 되면  해당 유저가 본인인지 아닌지 판별하게 되고 본인일 경우 버킷리스트를 추가할 수 있는 버튼이 생깁니다.  팔로우와 팔로잉 신청한 사람을 볼 수 있으며 블로그 주인이 좋아요 한 게시글 리스트를 볼 수 있습니다. |
|  | | **버킷리스트 상세 UI**  버킷리스트를 클릭하게 되면 해당 버킷리스트의 상세 모달창이 활성화 되고 태그와 상세내용 버킷리스트를 함께 하고 있는 유저를 볼 수 있게 됩니다.  하트를 누를 경우 위시리스트로 등록되며 별을 누를 경우 버킷리스트로 등록됩니다. |

|  |  |
| --- | --- |
|  | **버킷 리스트 작성 페이지**  **버킷 리스트 수정 페이지**  회원은 블로그로 들어가 버킷을 작성할 수 있습니다. 버킷을 작성 할 때는 무조건 하나의 사진이 들어가야 하는 조건이 있어 유효성 검사가 실시되며 총 5개의 태그를 작성할 수 있습니다.  자신의 블로그일 경우 버킷 리스트 수정이 가능합니다. |
|  | **스토리 작성 페이지**  스토리 작성 페이지로 들어오게 되면 NAVER smartEditor 2.0이 적용되어 있고 사진과 글을 함께 쓸 수 있습니다.  Editor을 통해서 여러가지 서식을 사용할 수 있으며 사진 첨부 시 팝업이 뜨고 이미지를 올려 놓게 되면 이미지가 등록이 됩니다. |
|  | **스토리 수정 페이지**  스토리 수정 또한 NAVER smartEditor2.0이 적용 되어 있습니다. |
| …  C:\Users\JE\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1231231241.png | **블로그 페이지**  블로그에 들어가게 되면   1. 버킷리스트 내용 2. 게시글 이동 리모컨 3. 회원이 작성한 스토리 4. 스토리 댓글 5. 댓글의 답글 6. 게시글 리스트   이렇게 총 5가지의 내용이 보여지게 됩니다.  1 – 회원의 버킷리스트가 보이고 태그를 눌렀을 경우 검색 페이지로 해당 태그 값을 가지고 검색이 됩니다.  2 – 게시글 이동 리모컨이 있어서 스토리를 편리하게 이동할 수 있게 넣은 기능입니다.  JqueryUI의 Draggable을 사용하여 구현하였습니다.  3 – 회원이 작성한 스토리는 회원이 이 버킷리스트를 완료하기 위한 과정들이 적히게 되며 작성 순으로 밑으로 계속해서 볼 수 있습니다.  좋아요와 신고하기 기능이 존재합니다. Ajax를 통해서 회원당 좋아요와 취소가 가능하고 신고하기는 한번만 신고가 가능합니다.  4 – 스토리 댓글은 해당 스토리에 댓글을 달 수 있습니다. 비밀글로 달게 되면 블로그 주인과 본인만 볼 수 있습니다. 댓글은 신고기능이 존재하여 신고가 가능합니다. Ajax를 통해 신고는 한번만 가능합니다.  5 – 댓글의 답글은 해당 댓글에 답글을 적을 수 있어 커뮤니케이션을 할 수 있게 구현 하였습니다. 댓글과 답글이 작성되었을 시 알림이 가게 되어 확인 가능합니다.  6 - 게시글 리스트는 해당 유저의 버킷리스트 게시글의 리스트로 버킷리스트를 이동할 경우 자동으로 작성한 스토리가 아래로 쭉 보여집니다. |

|  |  |
| --- | --- |
| **주요 코드 리뷰** | |
| **Controller - BucketBlog()** | |
| C:\Users\JE\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2352fw.png | **Controller - BucketBlog()**  myBlog.me 주소 매핑을 통해 BucketBlog()가 실행됩니다.  유저가 블로그 탭을 통해 들어오게 되면 처음의 버킷리스트와 스토리가 화면에 보여지게 됩니다.  또한 블로그는 해당 유저만 쓰는게 아니고 모든 유저들이 들어 올 수 있습니다.   1. 블로그 탭 2. 버킷리스트 상세보기를 통한 블로그 이동 3. 좋아요한 게시글 리스트를 통한 블로그 이동   3가지로 접근이 가능합니다.  1 – 블로그 탭을 이용하게 되면 @RequestParam의 required 옵션을 통해서 URL: 매핑 값만 넘어오게 되면 첫번째 버킷리스트와 스토리가 보여집니다  2, 3- 버킷리스트 상세보기와 좋아요한 게시글 리스트를 통한 블로그 이동은 버킷리스트 번호와 스토리 번호를 같이 받게 되고 페이징처리가 되어있기 때문에 페이지를 찾기 위해 while문을 통해서 페이지 번호를 가져오고 안에 스토리 번호가 있는지 확인하게 됩니다. |

|  |  |
| --- | --- |
| **Mybatis를 통한 블로그 내용 가져오기** | |
|  | **Mybatis 활용하기**  Mybatis의 association과  collection을 통해 하나의 버킷리스트의 여러 개의 스토리와 스토리에 달린 댓글, 댓글에 달린 답글을 한번에 가져왔습니다.  Association은 하나의 객체에 대한 매핑을 해주는 옵션이고  Collection은 left join을 통해 나온 값들의 List를 가져올 수 있습니다.  association으로 댓글을 단 회원의 정보와 답글을 단 회원의 정보를 가져오게 됩니다  collection으로 스토리에 달린 댓글의 리스트, 댓글에 달린 답글 리스트를 가져오게 됩니다. |
| |  | | --- | | **소감** | | Jsp/springMVC 를 통해 파이널 프로젝트를 구현하였습니다.  코로나로 인해서 휴강과 온라인 강의 등 여러 가지 힘든 일정에도 불구하고 끝까지 해준 팀원들에게 너무 고맙다고 생각합니다.  이번 파이널 프로젝트는 게시판 CRUD 쪽에 많은 비중을 두었으며  JAVASCRIPT, AJAX를 통한 유효성 검사와 NAVER smartEditor2.0을 통한 게시판 구현,  기본 적인 게시판 등록, 수정, 삭제, 댓글-대댓글 관련해서 구현 하였습니다.  API나 Library를 통해서 여러 기능을 사용하는 것도 좋지만 이번 Spring을 통해서는 동작 구조나 흐름을 완벽하게 익히기 위해 프로젝트를 진행하였습니다.  또한 mybatis의 여러 기능을 사용하여 데이터를 가져왔지만 효율성에 대해 고려하지 못한 부분에 대해 공부가 필요할 것 같습니다. | | |