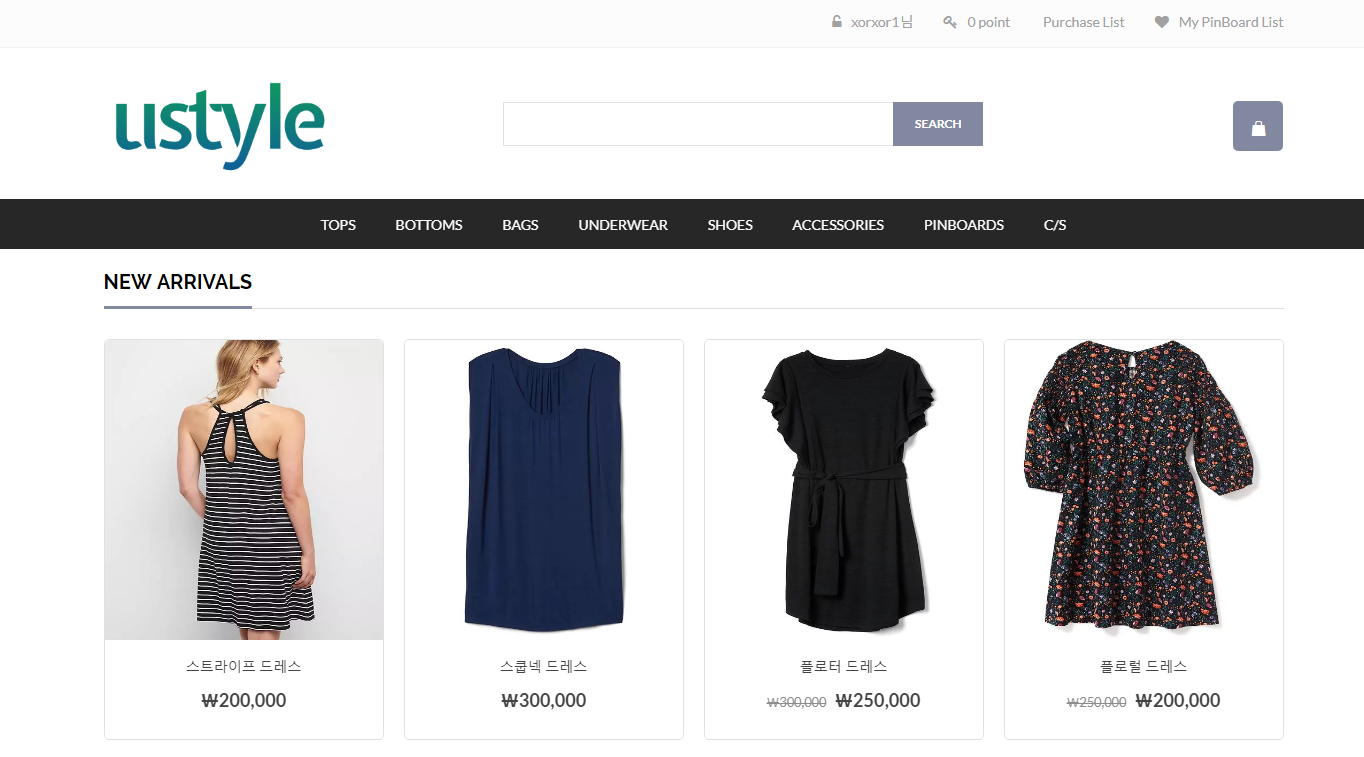
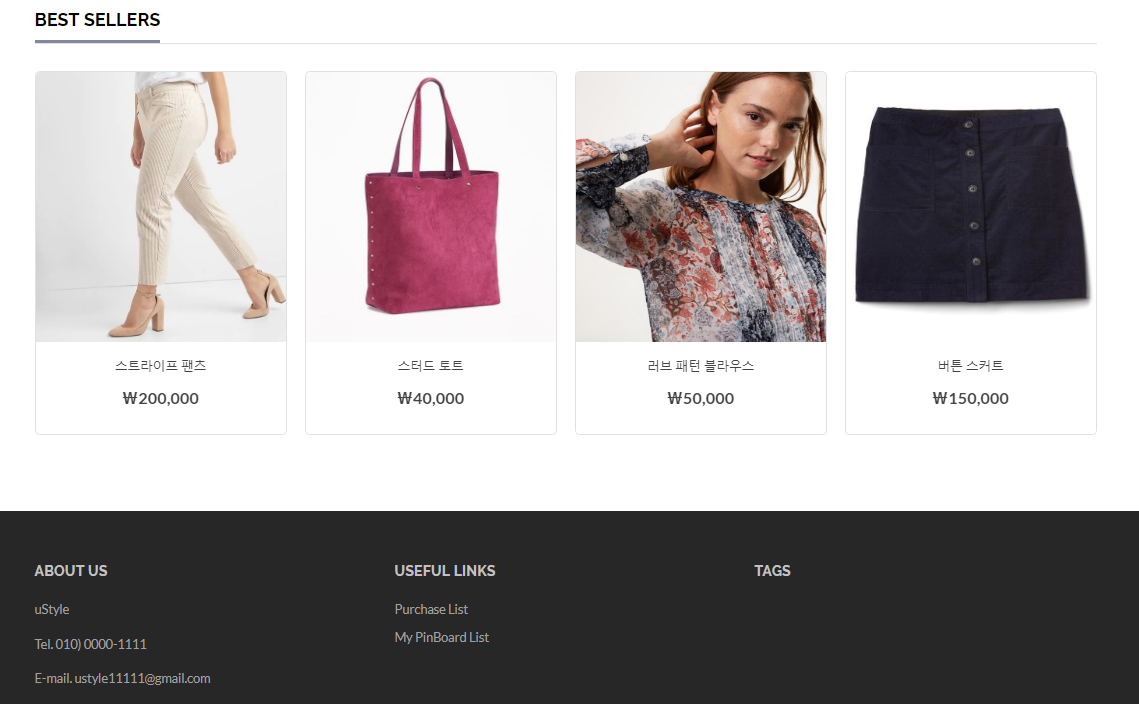
<uStyle Project>

* 구성원: 박선호, 최병일, 권택준
* 프로젝트 수행 기간: 2017.04 ~ 2017.09

<목 차>

1. 개요
2. 팀 구성 및 역할
3. 프로젝트에서 사용한 Framework와 구현 기술
4. 프로젝트 구조
5. 기능별 설명
6. 팀원별 프로젝트 Review와 소감
7. 개요
8. uStyle Project 소개





uStyle은 여성을 대상으로 한 쇼핑몰이다. 다양한 카테고리의 상품을 확인해볼 수 있으며, 상품의 색상, 사이즈를 선택하여 구매할 수 있는 작업이 이루어지도록 하였다. 그리고 원하는 상품을 등록해놓고, 등록된 상품을 다른 사용자에게 보여줌으로써, 그 상품의 구매를 유도할 수 있는 PinBoard 기능을 구현하였다.

기획, 디자인, 설계, 구현 작업까지 약 5개월의 시간을 가지고 작업을 진행하였다.

1. 개발 환경

|  |  |
| --- | --- |
| 구분 | 내용 |
| 운영체제 |  |
| 사용 언어와 개발 도구 | Java 1.8, Spring Tool Suite 3.8.3 |
| 사용된 라이브러리와 프레임워크 | Spring Framework 4.1.7, Log4j 1.2.17, Tiles 3.0.5,  Spring Security 4.0.4, MyBatis 3.2.8, jQuery 1.11.3, Handlebars.js 3.0.1 |
| DBMS | MySQL 5.7 |
| WAS | Apache Tomcat 8.0 |
| 버전관리 | GitHub(SourceTree 이용) |

1. 팀 구성 및 역할

박선호 - 개발 환경 Setting, 로그인/회원가입 기능 구현,

상품 리스트, FAQ / Q&A, 관리자 페이지 구현.

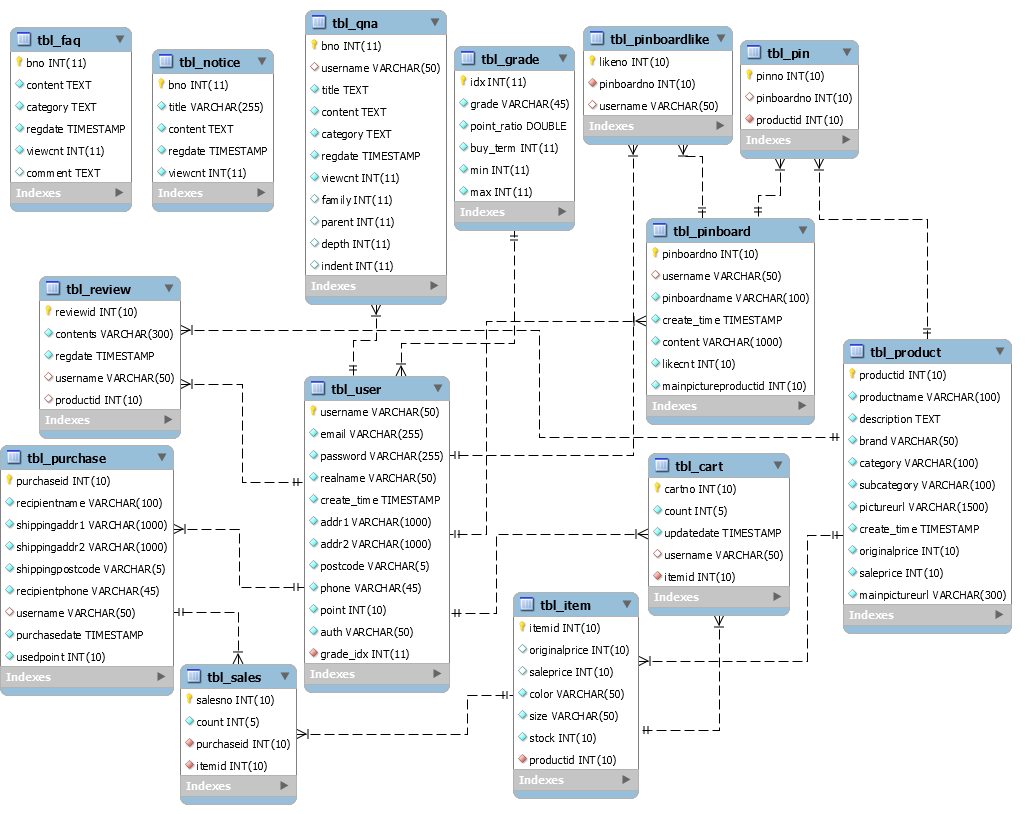
최병일 - 프로젝트 기획, 화면단 디자인 작업, 전체적인 설계 담당.

권택준 - 회원정보 수정/탈퇴 기능 구현, 상품 상세페이지, 장바구니, 결제, 관리자 페이지, PinBoard 기능 구현.

1. 프로젝트에서 사용한 Framework와 구현 기술

* Backend
  + github를 통한 소스공유(sourcetree 이용)
  + AWS EC2 이용 [http://52.78.169.40](http://52.78.169.40/) : 본 uStyle 홈페이지 주소
  + 3306포트 : mysql DB , 8080포트 : jenkins , 8089포트 : locust
  + docker 를 이용해 mysql DB 와 jenkins 를 각각 컨테이너로 세팅
  + jenkins 를 통해 CI/CD 구현. 빌드 및 배포 자동화. 지속적인 유지보수관리 가능.
  + github hook service와 jenkins내 플러그인 이용 소스 자동업데이트
  + locust 를 통해 서버 부하테스트
* Frontend
  + REST API 문서화 도구 Swagger 사용
  + 로그인이 필요한 페이지를 판별하기 위해 Interceptor 클래스 활용
  + tiles를 이용한 template framework
  + Ajax를 이용하여 아이디 중복체크, 파일 업로드, 장바구니, 상품 리뷰 작성 구현
  + Java mailsender를 이용하여 회원가입 이메일 인증, 비밀번호 찾기 기능 구현
  + Spring Security 를 통해 csrf token 적용
  + BCryptPasswordEncoder를 이용한 password 암호화
  + 만들어진 CSS template 이용(themeforest, AdminLTE)
  + bootstrap modal을 이용하여 pinboard 기능 구현

1. 프로젝트 구조
2. 데이터베이스 구조

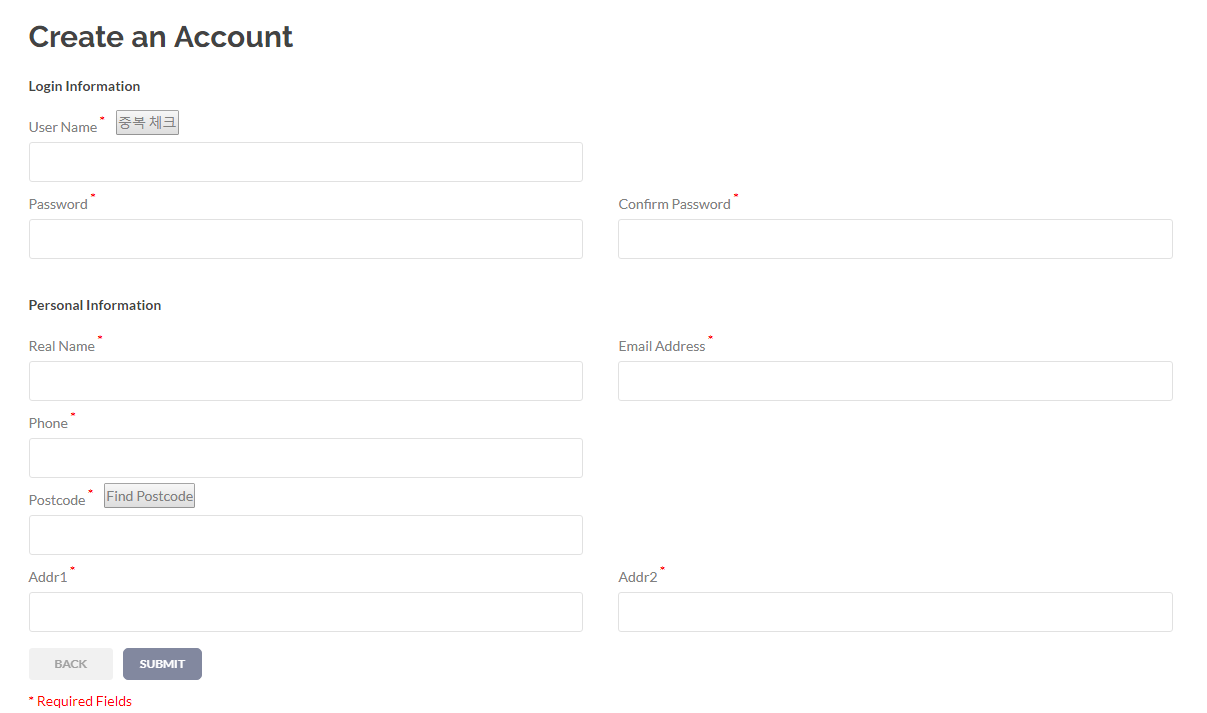


|  |  |
| --- | --- |
| 구분 | 테이블 |
| 사용자, 등급 | tbl\_user, tbl\_grade |
| 상품, 리뷰 | tbl\_product, tbl\_item, tbl\_review |
| 장바구니, 구매, 매출 | tbl\_cart, tbl\_checkout, tbl\_purchase, tbl\_sales |
| 게시판(공지사항, FAQ, Q&A) | tbl\_notice, tbl\_faq, tbl\_qna |
| Pin | tbl\_pinboard, tbl\_pin |

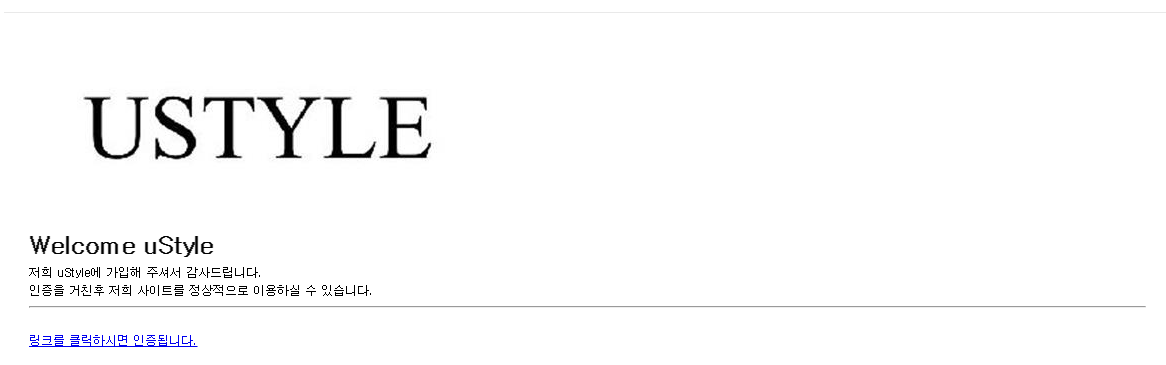
1. 디렉토리 구조

|  |  |
| --- | --- |
| 디렉토리 경로 | 설명 |
| src/main/java | 자바 소스 파일(컨트롤러, DAO, Service, Util 클래스) |
| src/main/resources | MyBatis 설정, 업로드 설정 파일, 매퍼 XML 파일 |
| src/main/webapp/WEB-INF | View단 JSP 파일, 프로젝트 구동 시 쓰이는 각종 설정 파일 |
| src/main/webapp/resources | 프로젝트에서 공통으로 쓰이는 각종 리소스 파일 |

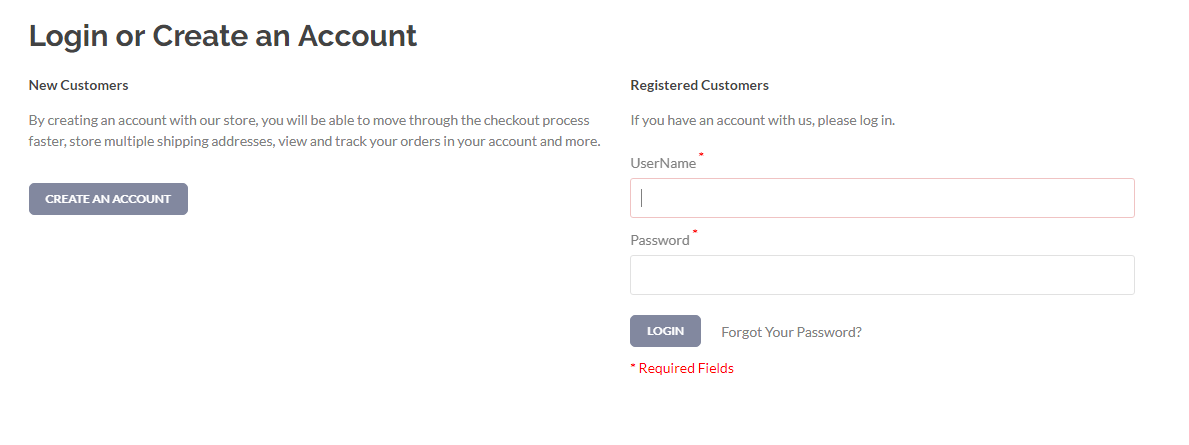
1. 기능별 설명
2. 회원 관련 프로세스 - 로그인, 회원가입, 회원정보 수정, 회원 탈퇴



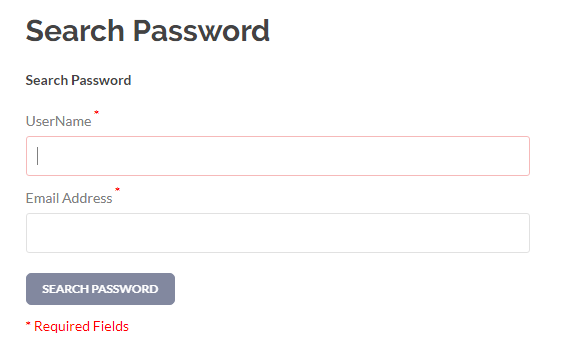
우측 상단의 Register 버튼을 클릭했을 때 나오는 회원가입 화면이다. 아이디 입력 후, 중복체크 버튼을 이용하여 회원가입하려는 아이디로 가입한 회원이 있는지 확인한다. 아이디는 영문 소문자로 시작하는 6~20자의 영문자 또는 숫자만 입력이 가능하도록 하였다. 비밀번호는 6~20자의 영문 대소문자, 숫자, 특수문자로 이루어져야 하며, 최소 한 개의 숫자 혹은 특수문자가 포함되도록 하였다. 이 비밀번호는 User 테이블에 암호화되어 저장된다. 비밀번호 암호화를 위해 Spring Security 암호화 클래스인 BCryptPasswordEncoder를 이용하였다. 모든 정보를 입력한 후, Register 버튼을 눌러서 입력한 이메일 주소로 회원가입 인증메일을 보내도록 하였다.



회원가입 인증메일이다. 메일에 첨부된 링크를 클릭하여 회원가입 인증을 진행한다. 이 과정을 거쳐야 정상적으로 로그인하여 서비스를 이용할 수 있다.



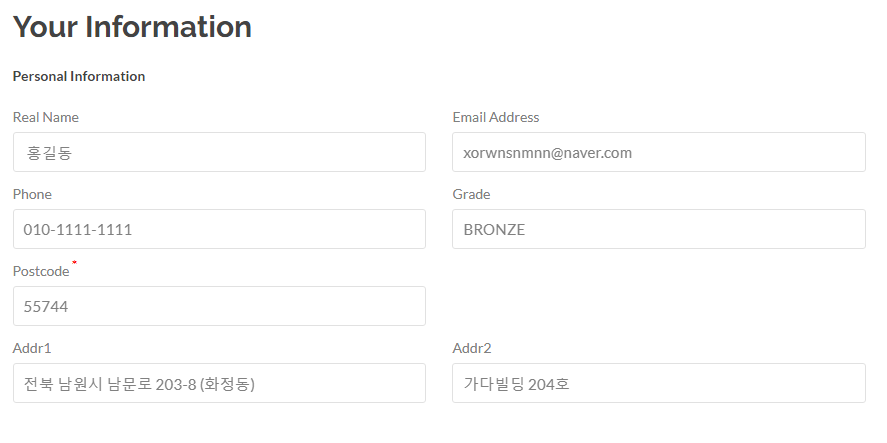
로그인 페이지이다. 가입할 때 입력했던 아이디와 비밀번호를 입력하여 로그인을 진행한다. 제대로 입력하지 않은 경우, 아이디와 비밀번호를 다시 확인하라는 alert 창이 나오게 된다.



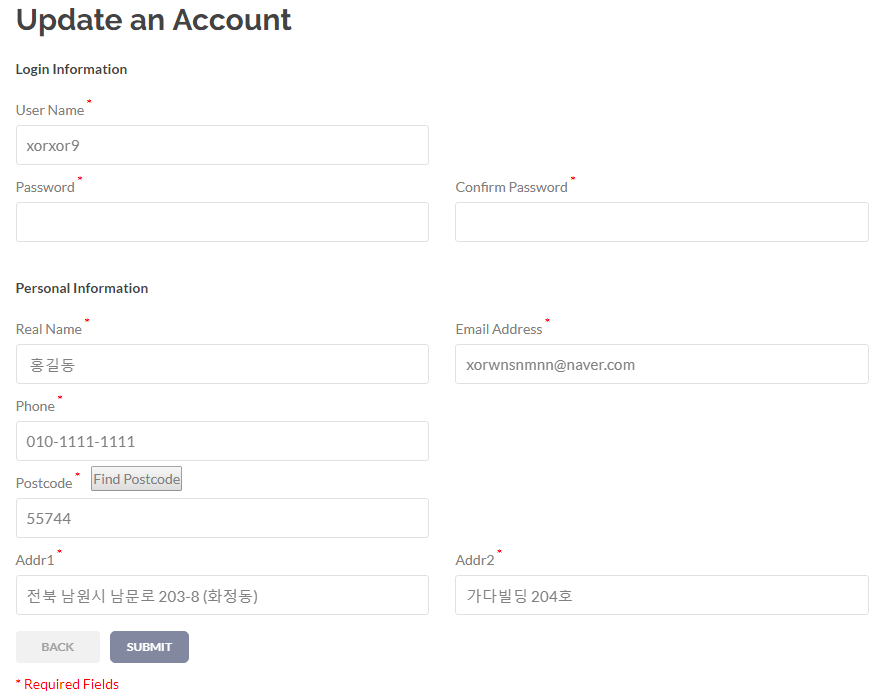


비밀번호를 분실했을 경우, 로그인 페이지의 Forgot Your Password 링크를 클릭하여 임시 비밀번호를 발급받을 수 있다. 아이디와 가입한 메일 주소를 입력하여 제출하면, 그 메일주소로 영문 소문자, 숫자 조합의 여덟 자리 임시 비밀번호가 발송된다. 그 비밀번호로 로그인이 가능하도록 구현하였다.

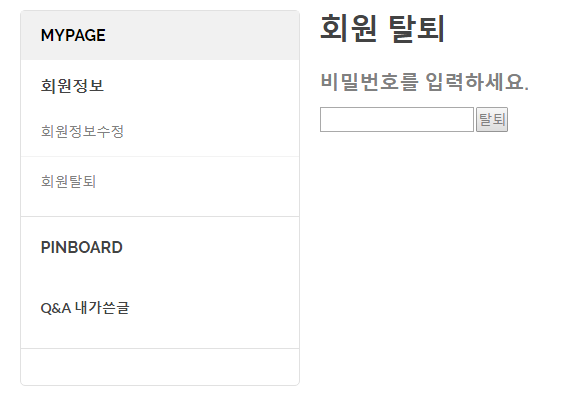
C:\Users\TaekJun\Desktop\제목 없음.jpg



로그인 후, 우측 상단에 있는 본인의 포인트 버튼을 클릭하여 회원의 정보를 확인할 수 있다. 회원의 메일주소, 실 주소, 연락처, 등급 등을 확인할 수 있다.

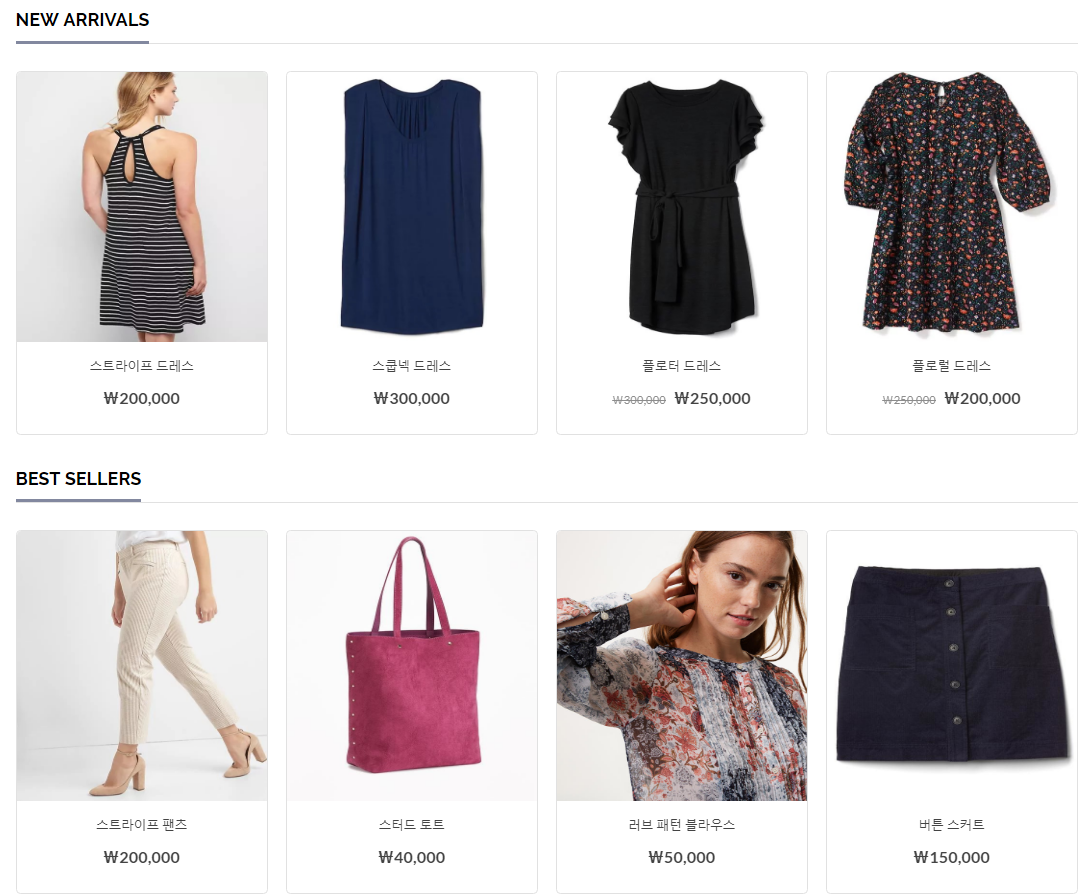


회원정보 수정 페이지이다. 이 페이지에서 비밀번호를 포함한 회원 정보를 수정할 수 있다.

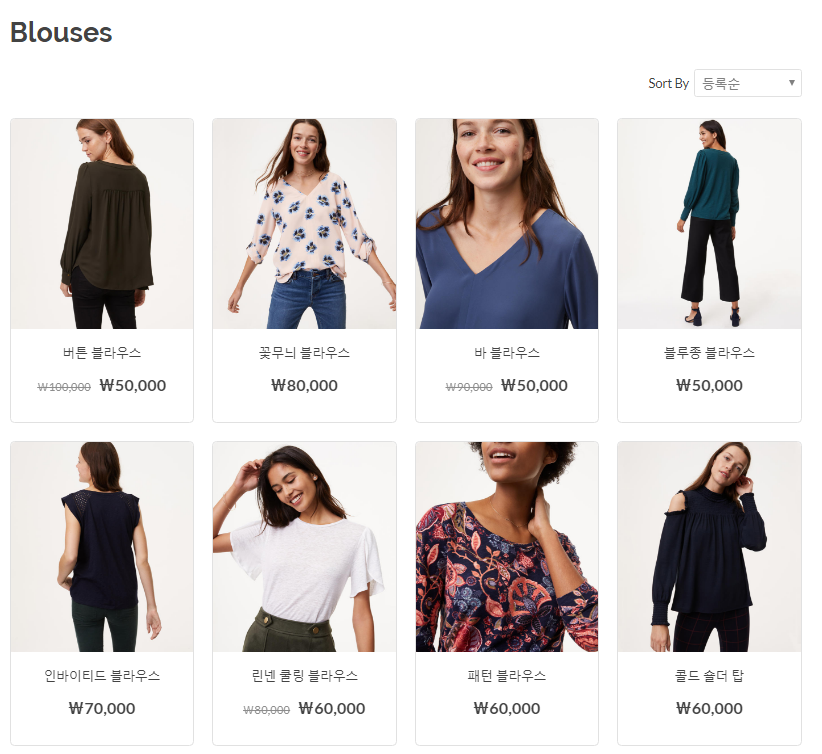


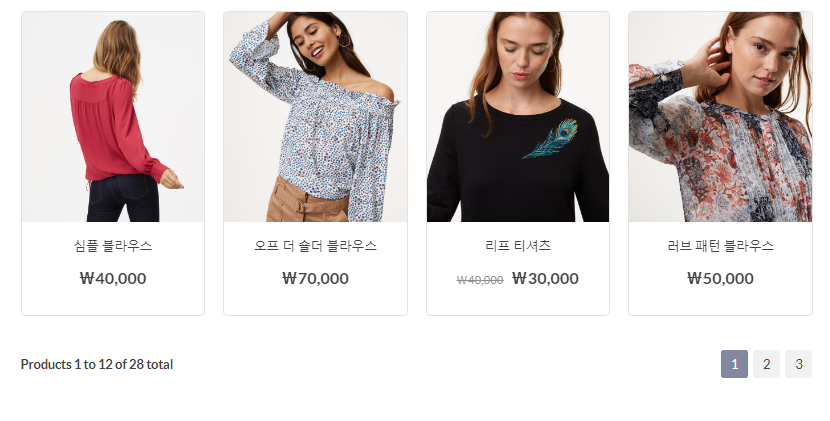
회원탈퇴 페이지이다. 비밀번호를 입력한 후, 탈퇴 버튼을 누르면 “탈퇴하시겠습니까?” 라는 confirm 창이 나오며, 여기서 확인 버튼을 누르면 탈퇴하도록 구현하였다. 탈퇴하게 되면, 탈퇴한 회원 정보가 user 테이블에서 모두 사라지며, 탈퇴한 회원이 작성한 상품 리뷰와 PinBoard, Q&A가 모두 삭제된다. 그리고 탈퇴한 회원의 구매 내역에 있는 아이디 값이 NULL로 바뀌도록 구현하였다.

1. 상품 나열 관련 프로세스 – 상품 리스트, 상세페이지



메인 페이지에 최근 추가된 상품과 인기 상품들을 각각 네 개씩 나열했다. 인기 상품의 기준은가장 많이 팔린 상품 순으로 하였다. 나열한 상품을 마우스로 클릭하면, 새로운 탭이 열리면서 그 상품의 상세페이지를 불러올 수 있도록 구현하였다.

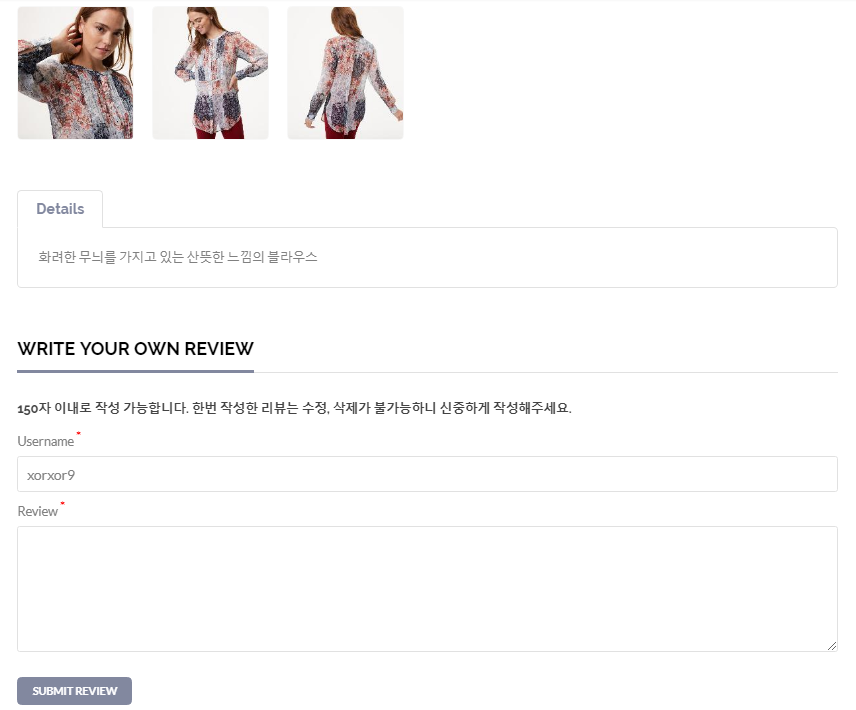


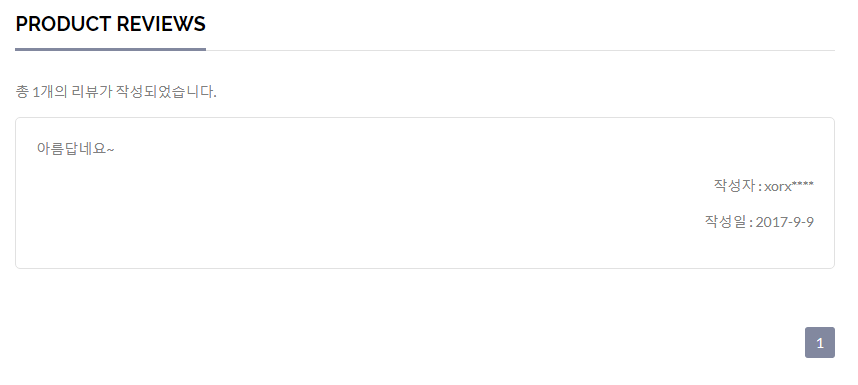


서브 카테고리별로 상품들을 열두 개씩 나열하였다. 우측 상단의 Select Box를 통해 최근 등록 순, 이름 순, 낮은 가격 순, 높은 가격 순으로 나열할 수 있도록 하였다. 나열한 상품을 클릭하면, 그 상품의 상세페이지를 불러올 수 있도록 구현하였다. 그리고 왼쪽에 나열된 서브카테고리를 클릭하면, 그 서브카테고리에 해당하는 상품 리스트 페이지로 이동하게 된다. 아래에 있는 브랜드는 그 서브카테고리 상품에 해당하는 브랜드를 리스트로 묶은 것이다. 각각의 브랜드를 클릭하면, 그 서브카테고리 상품 중에서 클릭한 브랜드에 해당하는 상품만 불러올 수 있다.

우측 하단에 PageMaker 클래스를 이용한 페이지네이션을 구현했으며, 페이지 개수가 최대 10개까지 나올 수 있도록 구현하였다.



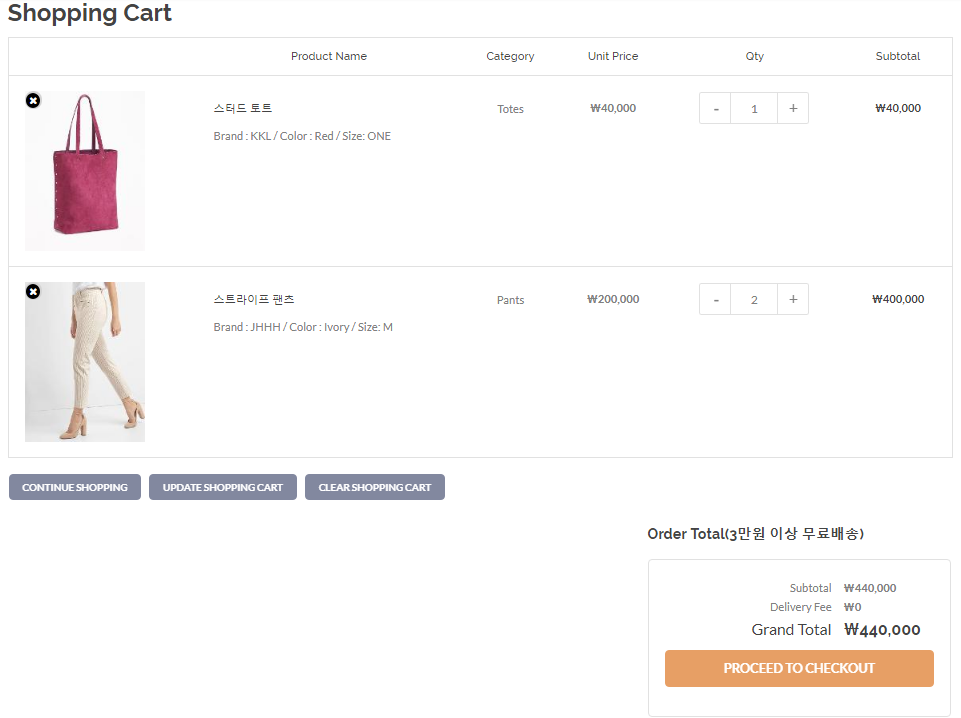




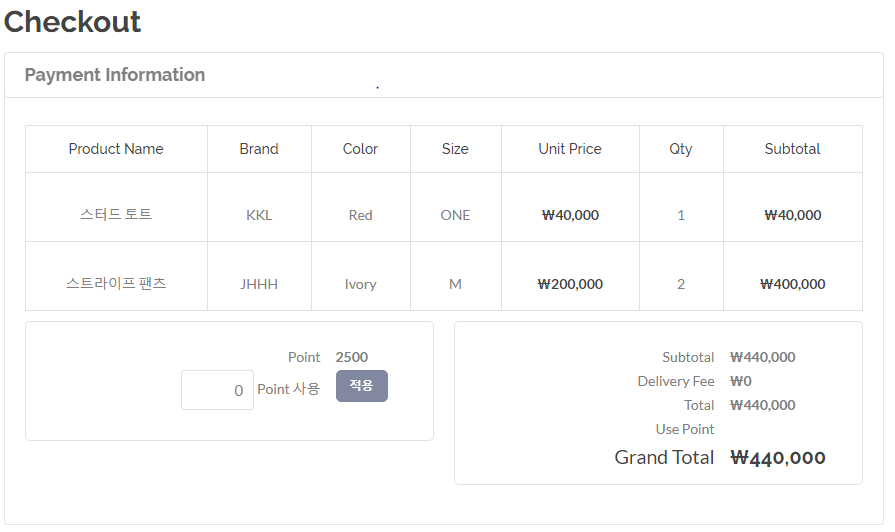
상품 상세페이지 화면이다. 좌측에는 상품의 메인 이미지와 추가 이미지가 나와있다. 아래에 있는 추가 이미지 썸네일을 클릭하면, 그 이미지를 크게 확대해서 볼 수 있다. 이미지 썸네일 아래쪽에는 상품 리뷰를 적을 수 있는 Form과 그 상품에 해당하는 리뷰들을 확인할 수 있다. 리뷰는 다섯 개씩 볼 수 있으며, 최대 10개까지 나올 수 있는 페이지네이션이 구현되어 있다. 그리고, 리뷰를 작성한 작성자의 아이디 뒷부분을 마스킹하여 작성자의 아이디가 완전히 노출되는 것을 막을 수 있도록 구현하였다.

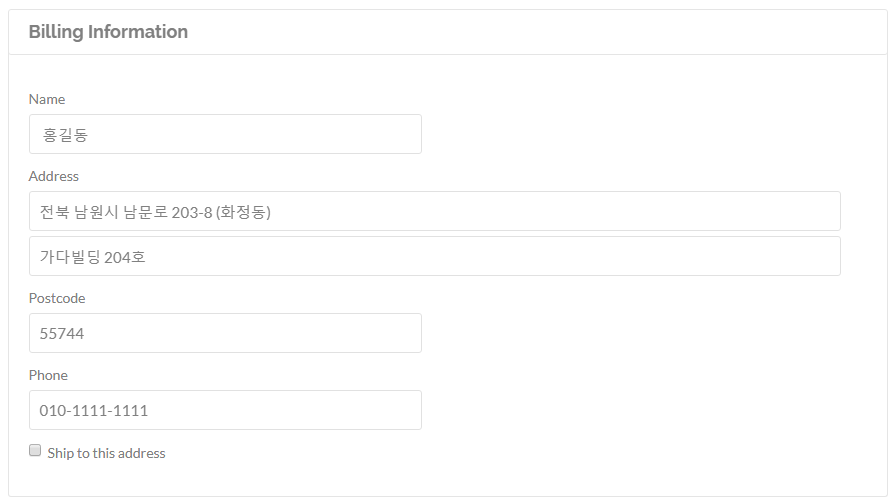
우측 상단에 상품명이 나와있으며, 아래에 상품 고유번호(자동 증가하는 기본 키), 상품 색상, 사이즈 목록이 나와있다. 색상을 선택하면, 그 색상에 맞는 사이즈가 Select Box에 나온다. 각 색상에 해당하는 사이즈를 표기할 때, Ajax를 이용한 비동기식 데이터 처리를 수행하였다. 매진된 사이즈의 경우, Out of Stock을 표기하여 옵션을 선택할 수 없도록 구현하였다. 그리고 아래의 두 버튼을 클릭하여 상품을 장바구니에 담거나 Pin으로 추가할 수 있도록 하였다.

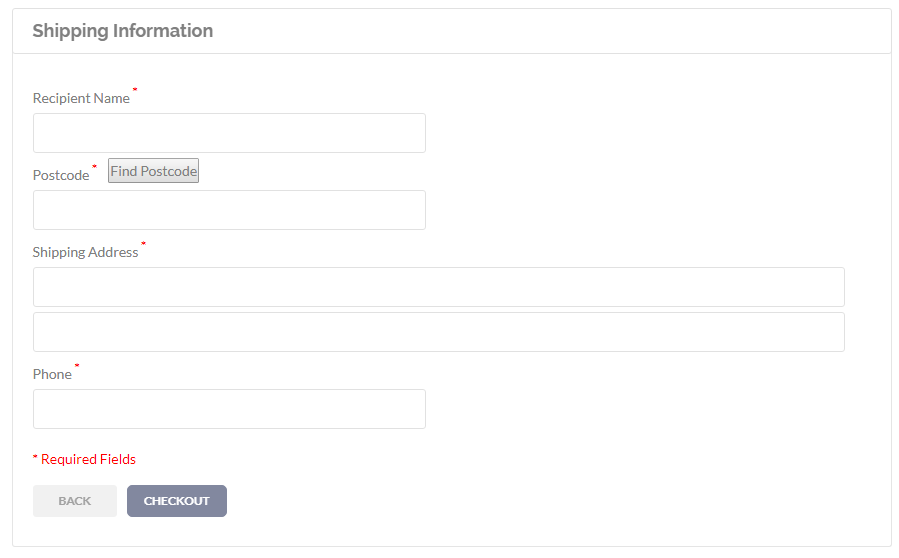
1. 상품 구매 관련 프로세스 – 장바구니 담기, 상품 구매하기, 구매내역 확인



상품의 상세페이지에서 색상과 사이즈를 선택한 후, ADD TO CART 버튼을 클릭하여 상품을 장바구니에 추가할 수 있다. 이때, 인터셉터를 이용하여 로그인이 되지 않았다면 로그인 창으로 이동할 수 있도록 하였다. 장바구니 페이지를 통해 상품의 개수를 늘리거나 줄일 수 있다. 또한, 상품 이미지 왼쪽에 있는 X 버튼을 이용하여 아예 상품을 뺄 수 있도록 구현하였다. 변경 사항을 적용하기 위해 좌측 하단에 있는 UPDATE 버튼을 이용하여 장바구니 정보를 변경할 수 있도록 하였다. 우측 하단을 통해 장바구니에 담긴 물건들의 총 가격을 확인할 수 있다. PROCEED TO CHECKOUT 버튼을 통해 구매 페이지로 이동한다.







상품을 구매하기 위해 회원 정보, 배송지 정보를 입력하는 Form이 나온다. 회원 정보와 배송지 정보가 동일할 경우, 동일하다는 체크박스를 클릭하여 사용자의 편의성을 높였다. 그리고 가지고 있는 포인트를 사용하여 결제 금액에서 차감할 수 있도록 하였다.

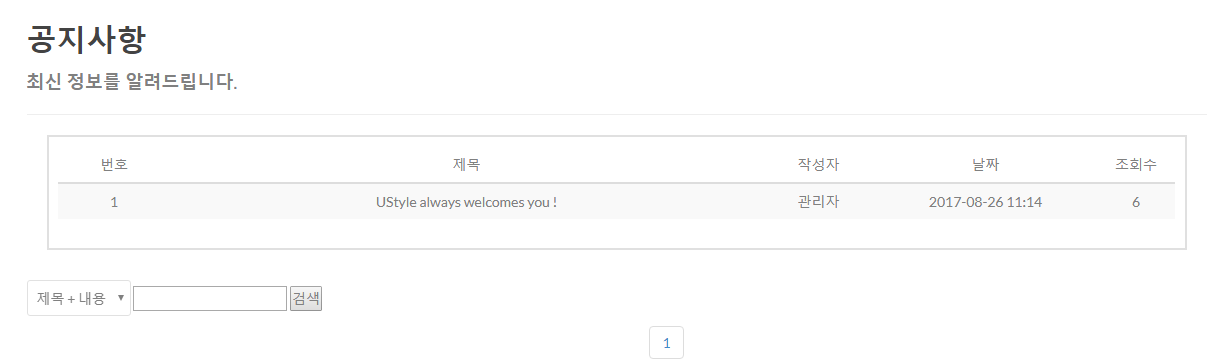


모든 정보가 올바르게 입력된 상태로 결제 버튼을 누르면, ‘결제 성공’ 이라는 알림창이 뜨고 메인화면으로 이동한다. 결제한 상품의 개수만큼 각각의 Item 재고수량에서 차감되며, 이와 동시에 이용자의 구매내역에 구매한 상품이 추가된다. 또한, 구매한 가격과 회원 등급에 따라 포인트가 적립된다. 이 포인트는 차후 uStyle 쇼핑몰에서 상품을 구매할 때 이용할 수 있도록 하였다.

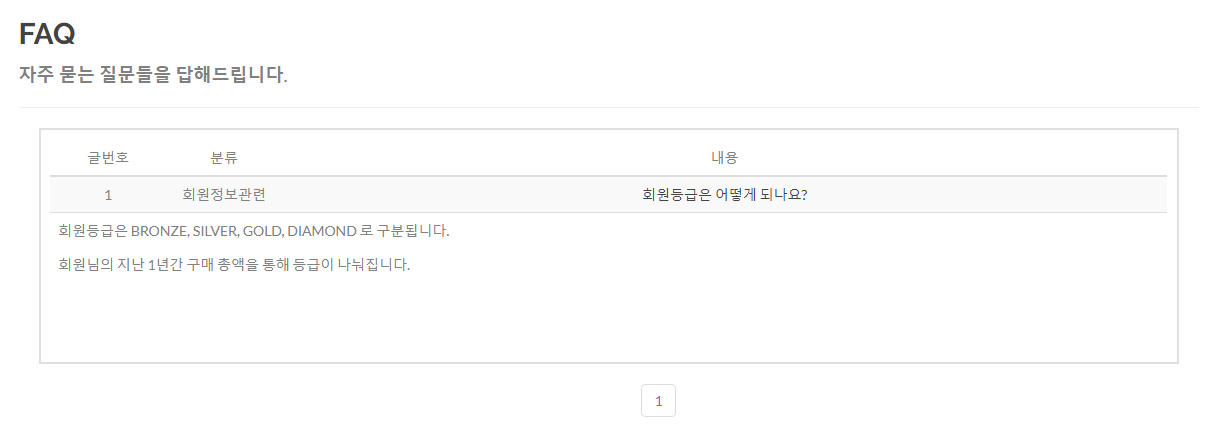


구매내역을 확인할 수 있는 페이지이다. 최근 1년동안 구매한 상품들을 하나하나 확인할 수 있다. 이 페이지에도 최대 10개까지 나올 수 있는 페이지네이션이 구현되어 있다.

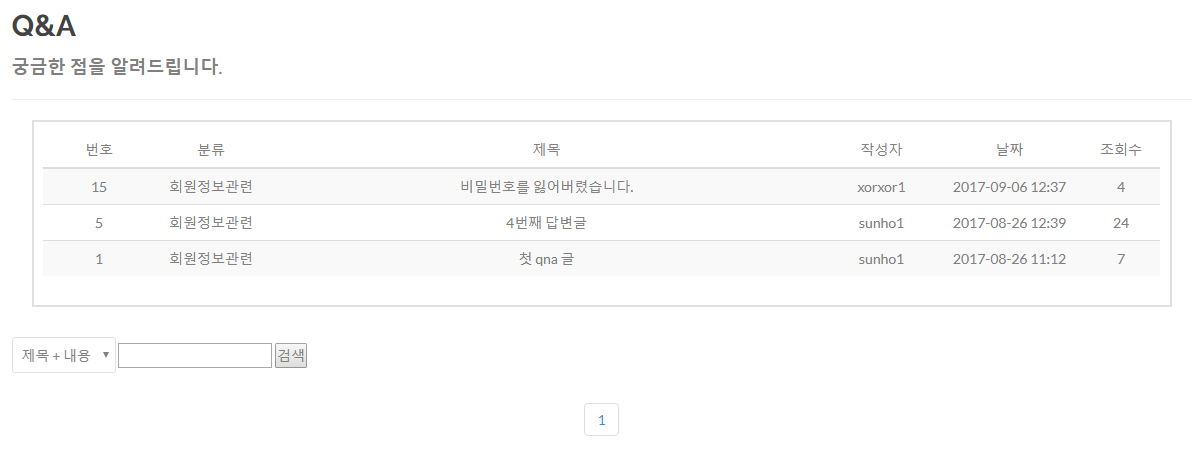
1. 게시글 관련 프로세스 – 공지사항, FAQ, Q&A



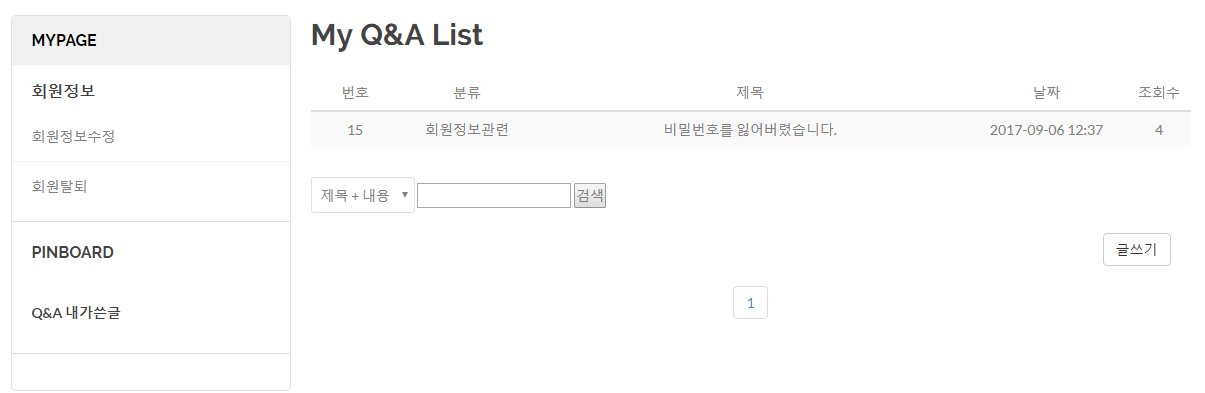
관리자가 추가해놓은 공지사항을 확인할 수 있는 페이지이다. 페이지네이션이 구현되어 있으며, 제목과 내용으로 공지사항 글을 검색해볼 수 있다.



자주 사용하는 질문인 FAQ를 모아놓은 페이지이다. 페이지네이션이 구현되어 있으며, 제목을 클릭하면, 페이지 이동 없이 내용을 바로 볼 수 있도록 구현하였다.

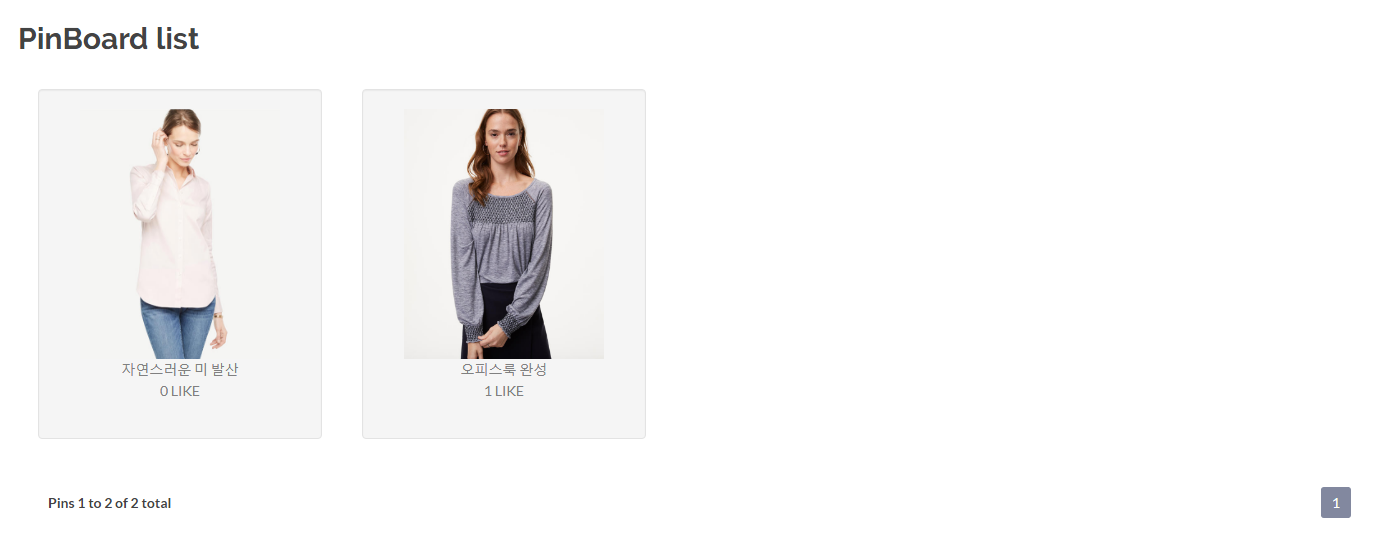


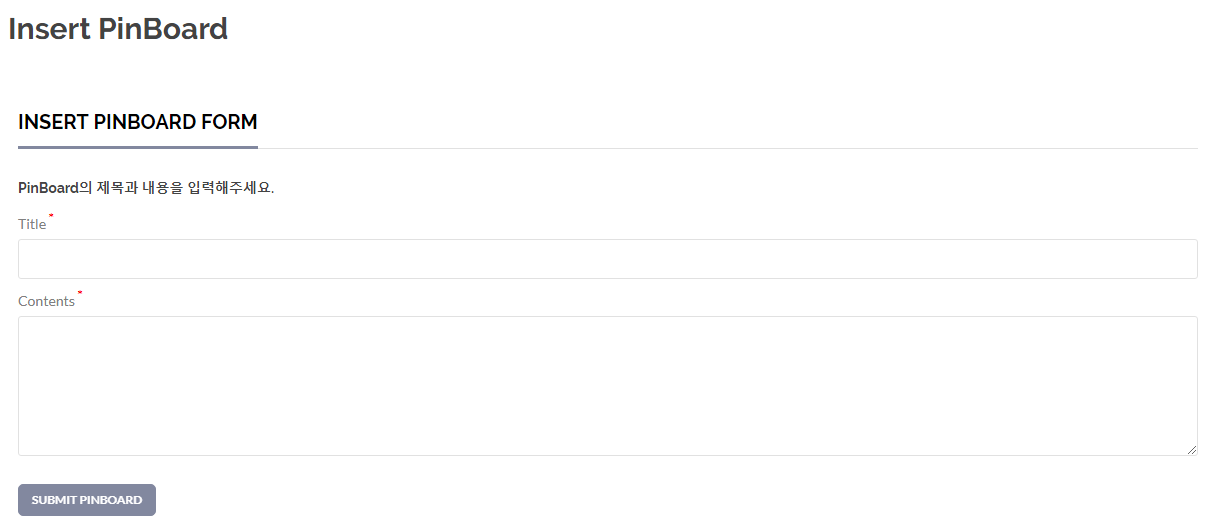
유저들이 질문을 올린 Q&A를 모아놓은 페이지이다. 페이지네이션이 구현되어 있으며, 제목과 내용으로 Q&A를 검색할 수 있다.



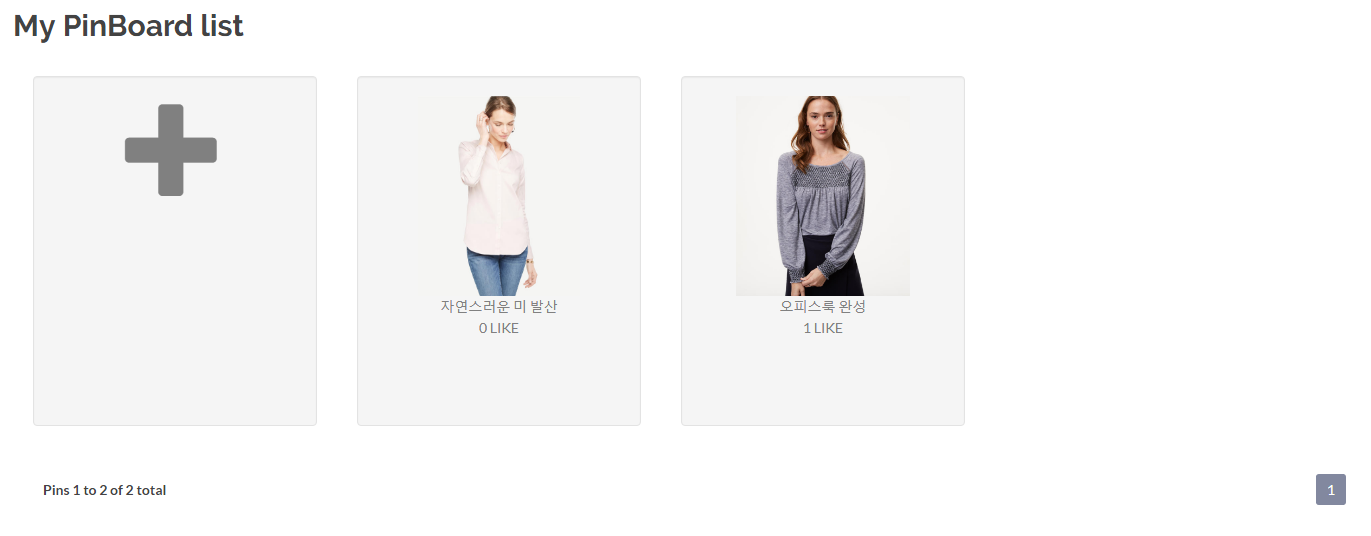
마이페이지 – Q&A 내가 쓴 글에서 로그인한 유저가 쓴 Q&A 목록을 확인할 수 있으며, 이 페이지에서 Q&A 쓰기, 수정, 삭제가 가능하다. 페이지네이션이 구현되어 있다.

1. Pin 관련 프로세스 – Pin, PinBoard

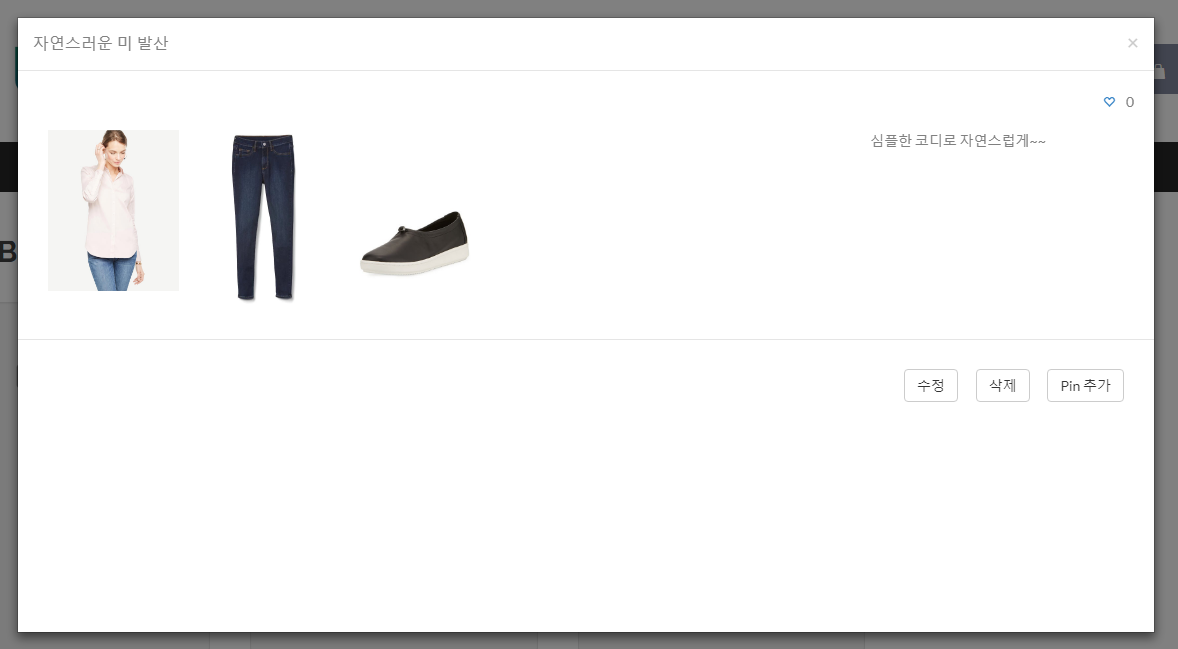




각각의 상품을 Pin으로 최대 4개까지 추가할 수 있는 PinBoard 리스트 페이지와 PinBoard 추가 Form이다. PinBoard 리스트 페이지에는 대표 이미지가 상품의 이미지로 명확하게 설정된 PinBoard만 나올 수 있도록 하였다. 페이지네이션이 구현되어 있으며, 한 페이지당 11개씩 나열되어 있다. 그리고 하나의 PinBoard에는 최대 4개의 상품을 Pin으로 넣을 수 있다. 좌측 상단의 PinBoard 추가 아이콘을 누르면, PinBoard를 추가하는 Form이 나오게 된다. 제목과 내용을 입력한 후, PinBoard를 추가할 수 있다.



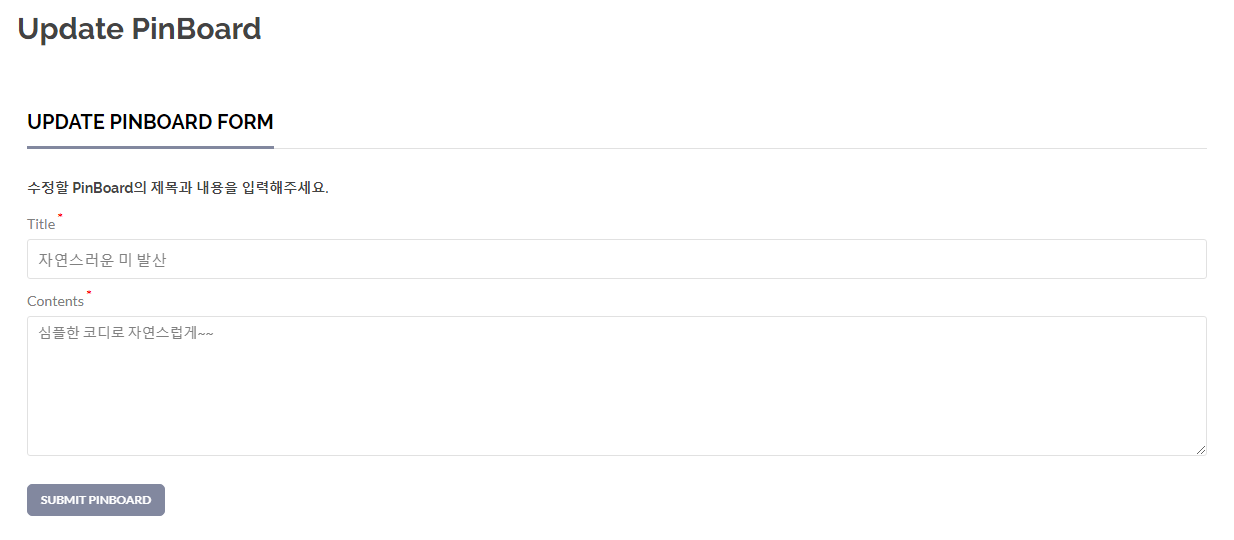
로그인한 유저 본인이 추가한 PinBoard만 따로 볼 수 있는 페이지를 제공하였다. 이 페이지는 로그인을 반드시 필요로 하는 페이지이기 때문에 인터셉터를 적용하여 로그인하지 않은 상태로 접근했을 때, 로그인을 유도하도록 하였다.

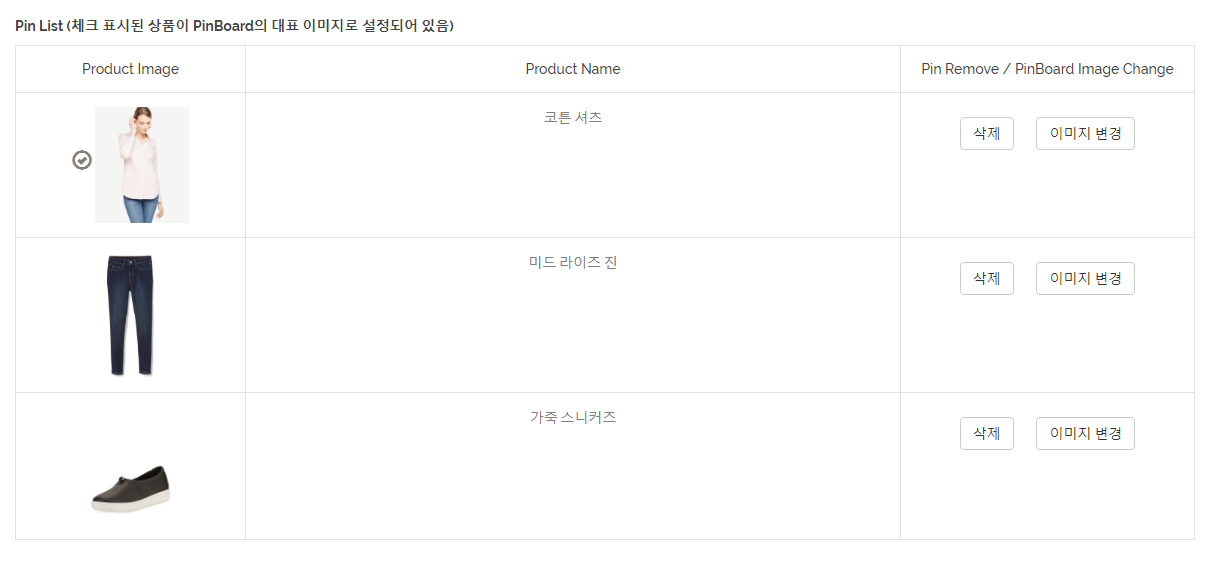


상품 상세페이지에 있는 ADD TO PIN 버튼을 클릭하여 원하는 PinBoard에 Pin을 추가할 수 있다. PinBoard에 가장 먼저 추가된 Pin이 그 PinBoard의 대표 이미지로 설정되며, 이 이미지는 PinBoard 수정 페이지에서 변경이 가능하다.

추가한 PinBoard를 리스트에서 클릭하면, Modal 창이 띄워지면서 PinBoard의 내용이 로딩된다. 좌측에 추가된 Pin이 나오고, 우측에 좋아요 수, PinBoard 내용, 수정 / 삭제 / Pin 추가 버튼이 있다. 여기서 수정 / 삭제 버튼은 로그인한 계정과 로딩된 PinBoard를 추가한 계정이 일치할 때만 나오도록 하였다. 삭제 버튼을 눌렀을 때, 삭제를 확인하는 confirm 창이 나오게 되며, 확인을 누르면 그 PinBoard의 내용과 좋아요 수, 그에 해당하는 Pin들이 모두 사라지게 된다. 그리고 Pin 추가 버튼은 상품 상세페이지의 ADD TO PIN 버튼을 클릭하여 PinBoard 페이지로 이동했을 때 나오는 버튼이다. 이 버튼을 통해 상품을 Pin으로 추가할 수 있다.

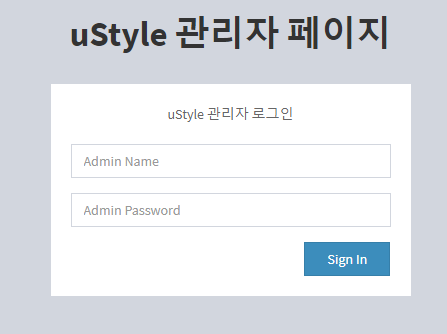
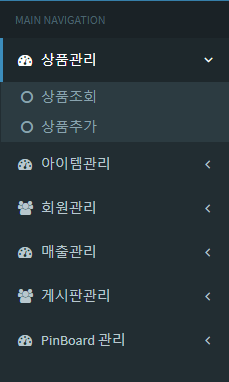
우측 상단에 있는 하트 아이콘을 클릭하여 좋아요 수를 1 증가시킬 수 있다. 하나의 회원이 하나의 게시물에 하나씩 적용시킬 수 있다. 좋아요 수가 많은 PinBoard를 통해 그 PinBoard에 있는 상품을 구매하도록 유도할 수 있다.





PinBoard 수정 버튼을 눌렀을 때의 화면이다. PinBoard의 제목과 내용을 바꿀 수 있으며, 추가된 Pin의 삭제와 PinBoard 대표 이미지 변경이 가능하다. 대표 이미지가 상품의 이미지로 명확하게 설정된 PinBoard만 리스트에 나올 수 있기 때문에 PinBoard 대표 이미지를 명확하게 설정해줘야 한다.

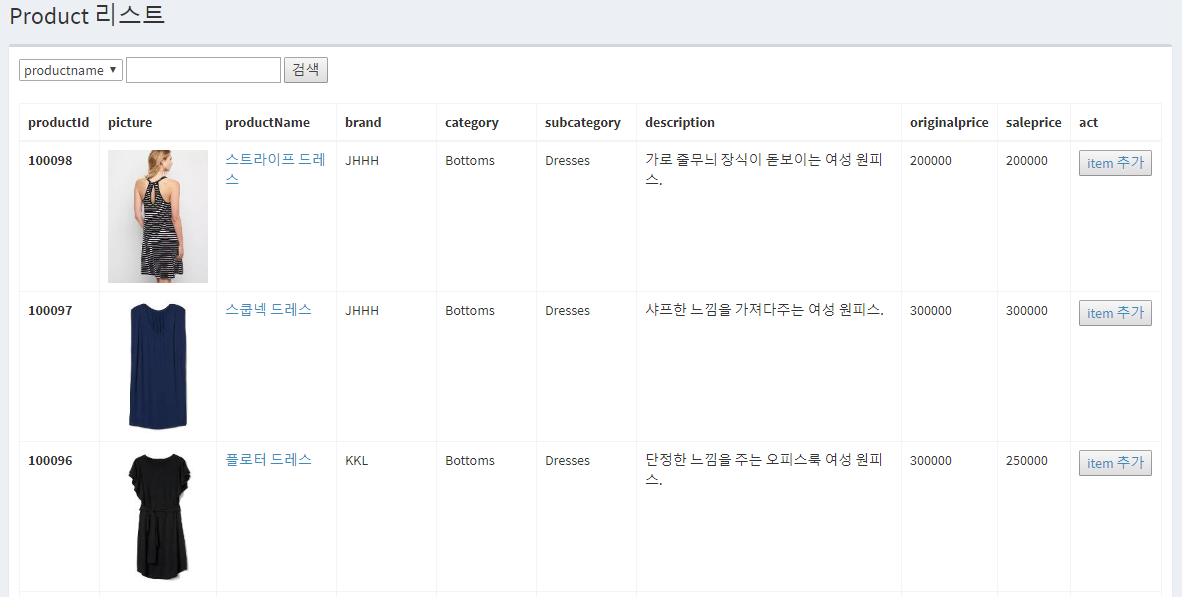
1. 관리자 프로세스 – 상품 관리, 아이템 관리, 매출 관리

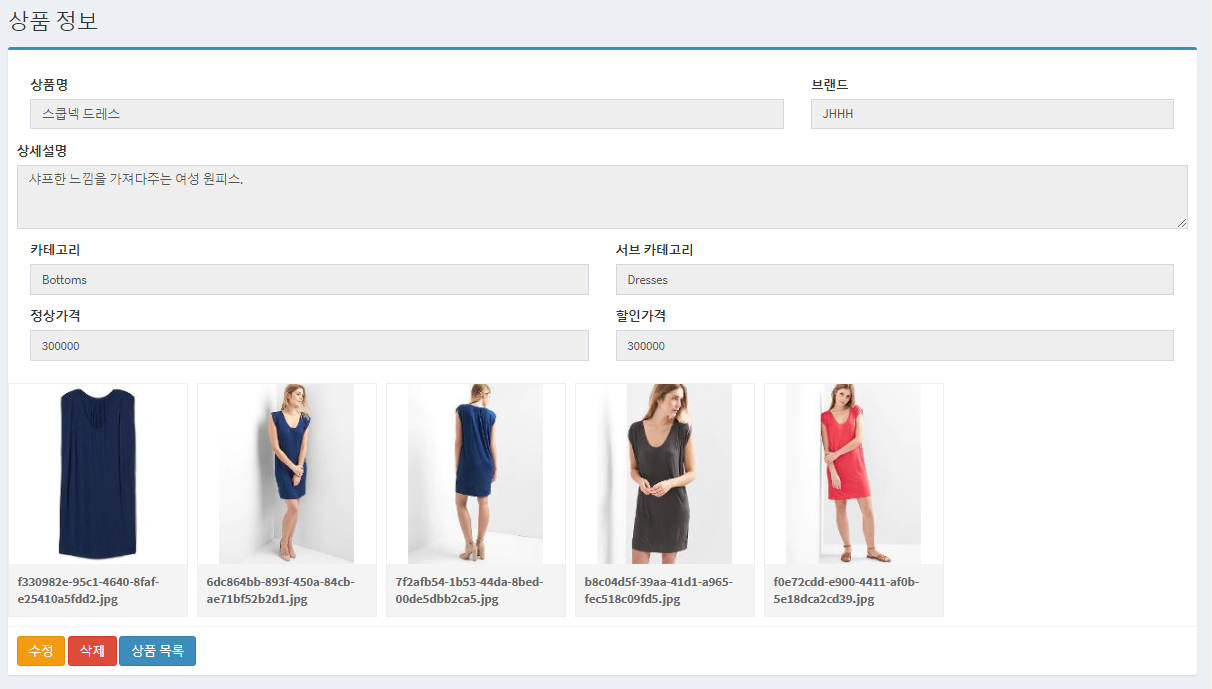
관리자 페이지(http://52.78.169.40/admin/main.do)에 접속하여 관리자 모드에서 작업을 진행하기 위해 우선 관리자 계정으로 로그인해야 한다. Spring Security를 이용하여 관리자 권한을 부여하였다.



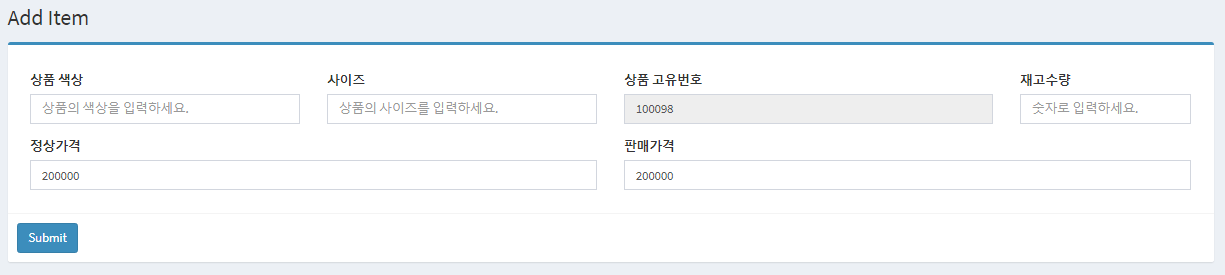
왼쪽 사이드바에 있는 상품관리 - 상품추가 항목을 클릭하면 위와 같은 상품 추가 Form이 나온다. 추가하려는 상품 정보를 각각의 속성에 맞춰서 입력한 후, Submit 버튼을 눌러서 추가하면 된다. 제대로 입력하지 않은 경우, alert창을 통해 경고메시지를 띄운다. 이미지 추가는 이미지 파일(PNG, JPG, GIF)을 회색으로 되어있는 공간에 한 장씩 Drag & Drop하여 최대 5장까지 추가할 수 있다.



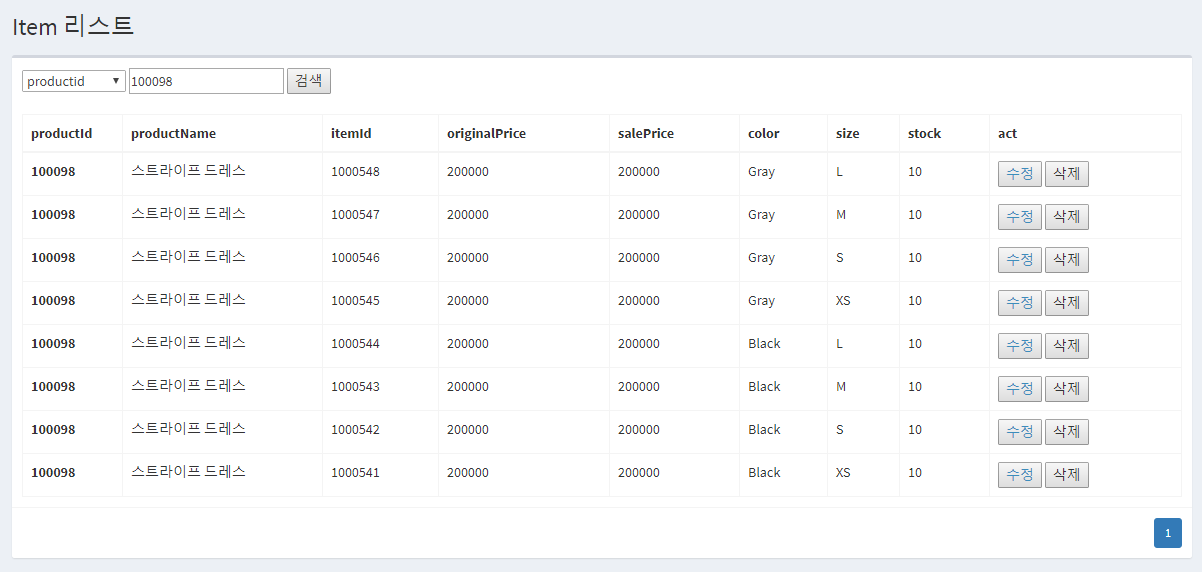
상품관리 – 상품조회 버튼을 클릭하면 추가된 상품을 10개씩 리스트로 볼 수 있는 페이지가 나온다. 페이지네이션이 적용되었으며, 상품 정보를 한눈에 볼 수 있도록 하였다. 그리고 좌측 상단의 Form을 이용하여 상품을 검색할 수 있다.



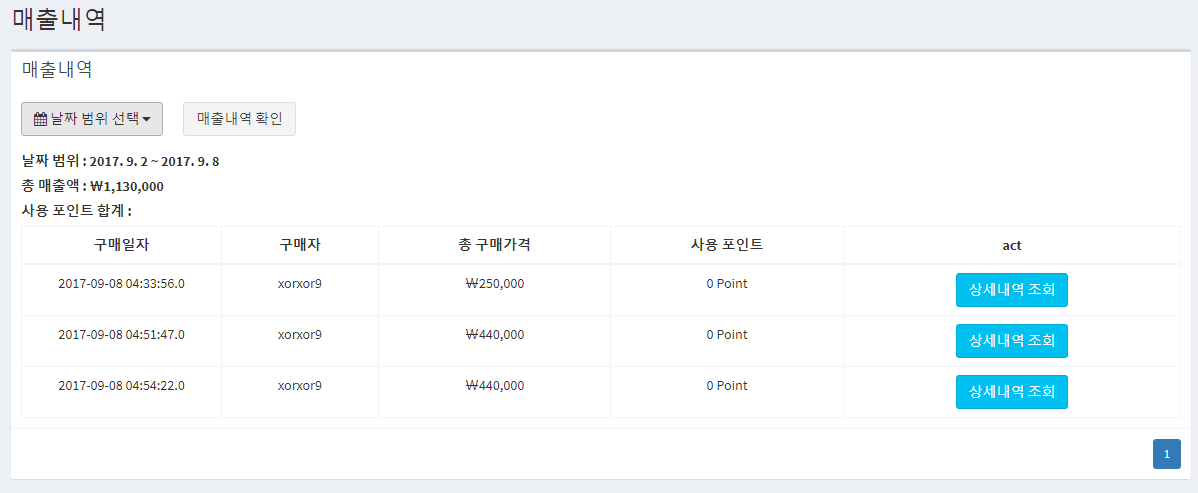
관리자 페이지의 상품 리스트에서 상품명을 클릭하면, 위와 같은 상품의 상세 정보를 확인할 수 있다. 추가 이미지까지 모두 조회가 가능하며, 상품 정보 수정, 상품 삭제를 할 수 있는 버튼이 하단에 존재한다. 수정 버튼을 누르면 상품 추가 Form과 같은 UI의 Form이 나온다. 그 Form을 이용하여 상품 정보를 수정할 수 있다. 삭제 버튼을 누르면 상품을 삭제할 수 있다. 그 상품에 해당하는 Item 정보까지 모두 삭제된다.



관리자 페이지의 상품 리스트 오른쪽에 있는 Item 추가를 클릭했을 때 나오는 Form이다. 상품에 해당하는 Item의 색상, 사이즈, 정상 가격, 할인 가격, 재고수량 등을 입력한 후, Submit 버튼을 눌러서 추가하면 된다. 추가한 Item은 관리자 페이지의 아이템조회에서 확인할 수 있다.



왼쪽 사이드바에 있는 아이템관리 - 아이템조회 항목을 클릭하면 Item 리스트 페이지가 나온다. 이곳에서 추가한 Item의 정보를 확인할 수 있다. 페이지네이션이 적용되었으며, 오른쪽의 수정 / 삭제 버튼을 통해 Item을 수정 혹은 삭제할 수 있다.

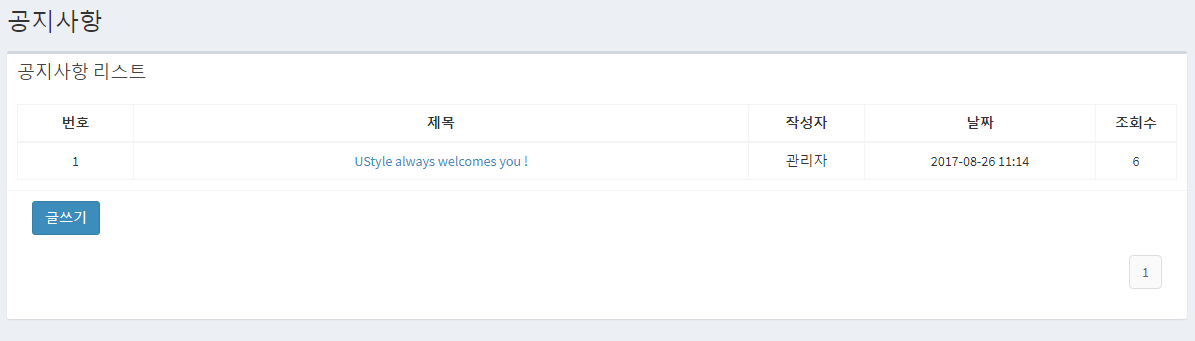


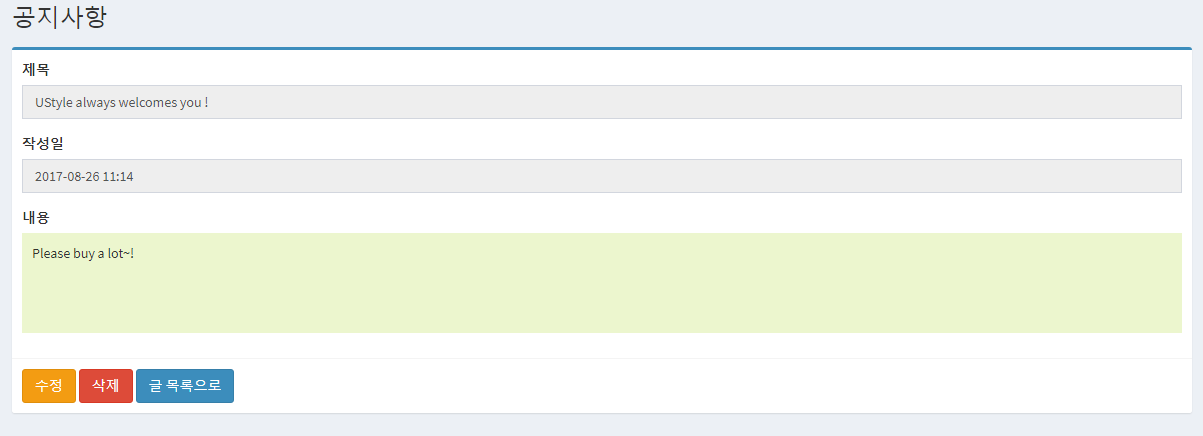
사이드바의 매출관리 – 매출조회 항목을 클릭하여 uStyle 쇼핑몰의 매출내역을 기간별로 확인할 수 있다. GET 방식을 통해 매출 범위가 지정된다. 선택한 기간에 맞춰 총 매출액, 사용포인트 합계를 확인할 수 있다. 실질적인 매출액은 총 매출액에서 사용한 포인트의 합계를 뺀 금액이 된다.



각 매출내역 우측에 있는 상세내역 조회 버튼을 클릭하여 매출 상세내역을 조회할 수 있다. 구매자 정보와 상세 구매 내역을 확인할 수 있다.

1. 관리자 프로세스 – 게시글 관리, 회원 관리, 등급 관리, PinBoard 관리

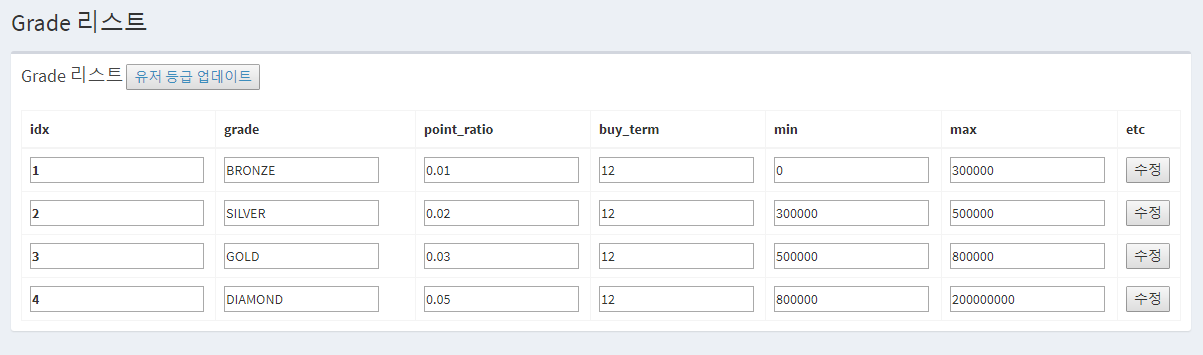




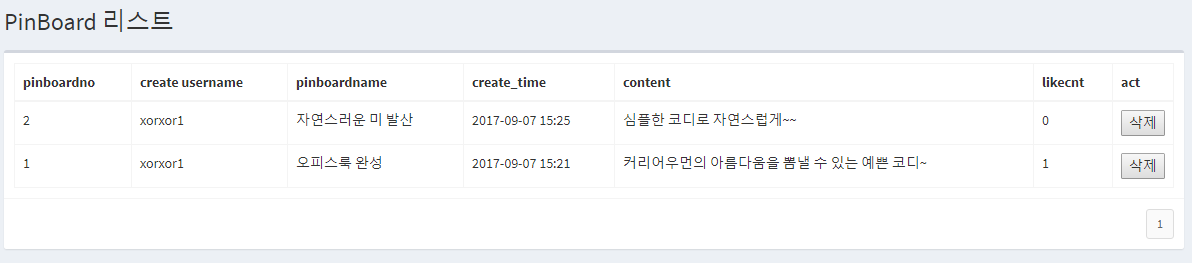
관리자 페이지에서 게시글들을 관리할 수 있다. 공지사항을 작성할 수 있고, 자주 묻는 질문(FAQ)을 작성할 수 있다. 그리고 유저가 올린 질문에 대한 답변을 작성할 수 있다. 또한, 작성한 게시글들을 하나하나 조회할 수 있다.



회원관리 – 회원정보조회를 통해 가입한 회원들을 한눈에 볼 수 있다. 페이지네이션이 적용되었으며, 각각의 회원 정보를 확인할 수 있다. 우측의 탈퇴 버튼을 누르면 해당 회원이 강제로 탈퇴된다. 그리고 0포인트 버튼을 누르면 해당 회원의 포인트가 0으로 초기화된다.



회원관리 – 등급관리에서 회원을 구분하는 등급의 기준을 조정할 수 있다. 포인트 적립금은 구매한 상품들의 액수와 회원의 등급에 따라 달라진다. 최근 1년간 회원이 구매한 상품들의 금액만큼 등급이 조정된다. 등급 기준을 수정한 후, 페이지 상단에 있는 유저 등급 업데이트 버튼을 눌러서 회원의 등급을 변경할 수 있다.



회원관리 - PinBoard 관리에서는 모든 유저들의 PinBoard 정보를 확인할 수 있다. 부적절한 PinBoard가 업로드되었을 때, 이를 삭제할 수 있도록 Ajax를 이용하여 PinBoard 삭제 기능을 구현하였다.

1. 팀원별 프로젝트 Review와 소감

* 박선호 –
* 최병일 –
* 권택준 – 8월 중순에 끝내는 것을 목표로 잡았었는데 예상보다 시간이 좀 오래 걸렸던 것 같다. 처음 접하는 기술도 많았고, 구현을 하면서 발생했던 각종 예외 때문에 적지 않은 시행착오를 겪었던 것 같다. 그래도 아이디어 구상, 설계, 구현에 이르기까지 실제로 활용되는 하나의 웹 프로그램을 나와 팀원의 힘으로 직접 구현했다는 것에 큰 의미를 두고 싶다. 그리고 이 프로젝트에서 사용한 Ajax를 이용한 비동기식 데이터 처리와 서비스단에서의 트랜잭션 구현, MVC 패턴 등등의 기술들이 실제 실무에서도 많이 쓰이는 기술이기 때문에 익숙해질 때까지 연마하는 작업을 꼭 해야 할 것 같다.