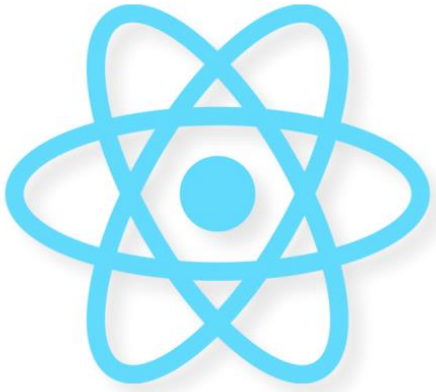
LEOPOLD

사이트
모바일 + 웹
가이드문서

작성자 안주현
작성일 2023.12.08

사용 기술

REACT



<HTML>
HTML

{CSS}
CSS

(JS)
JavaScript

제작 의도

1. 제품별 필터 기능 추가

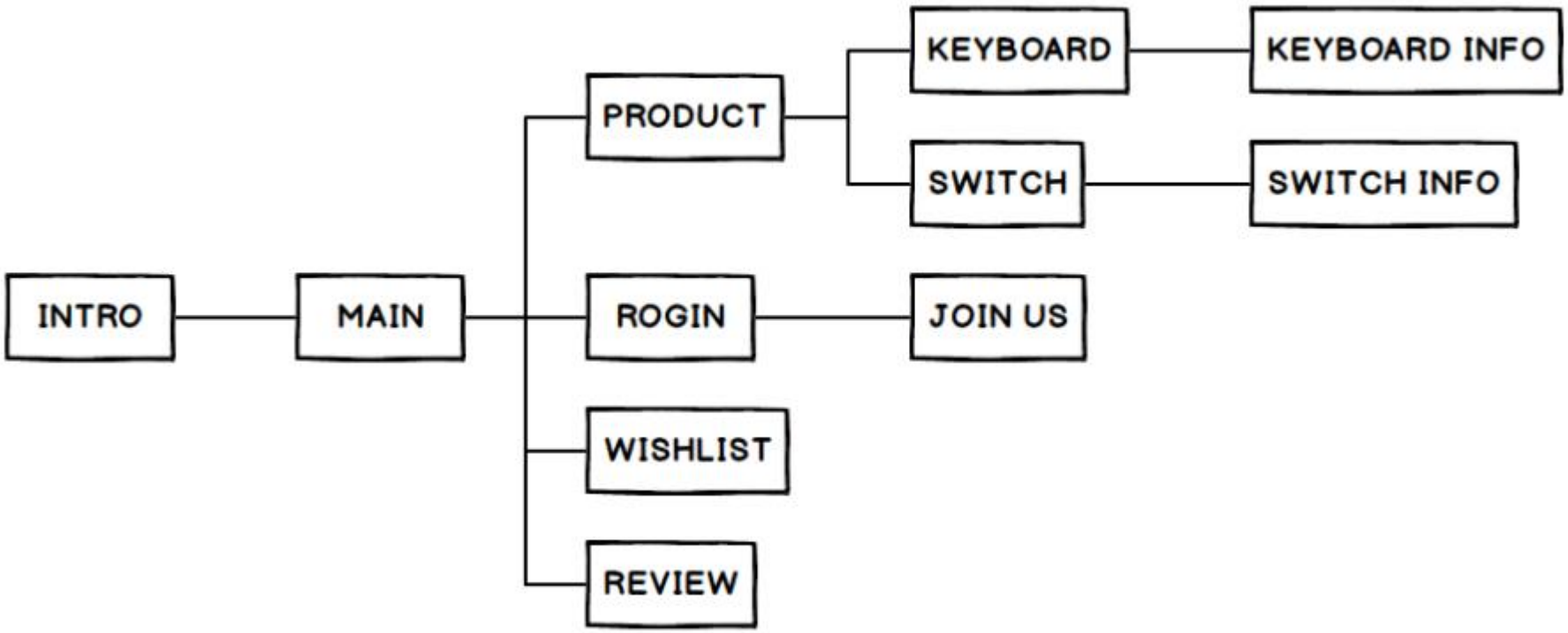
2. 키보드와 스위치 페이지를 별도로 구성

-> 고객에게 필요한 제품을 구매할 수 있도록 방향성 제시

3. 기존 사이트와 다른 구성

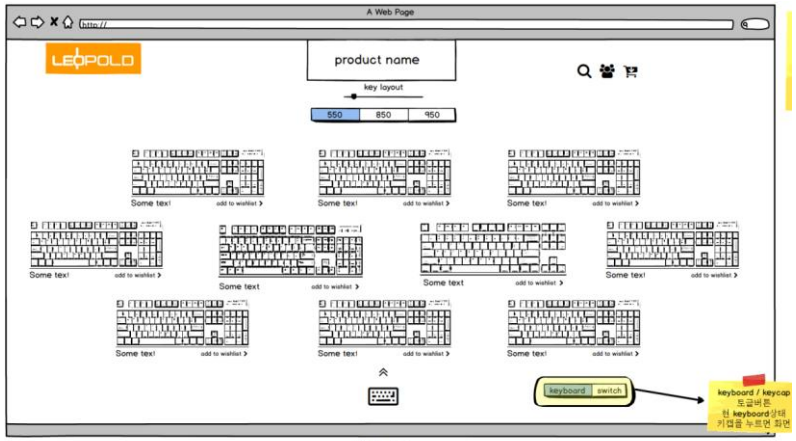
4. 메뉴 페이지에서 키보드 판매 사이트임을 나타내기 위해 디자인 추가

분석 설계 - 사이트 맵


INTRO
MAIN
PRODUCT
KEYBOARD
KEYBOARD INFO
SWITCH
SWITCH INFO
ROGIN
JOIN US
WISHLIST
REVIEW

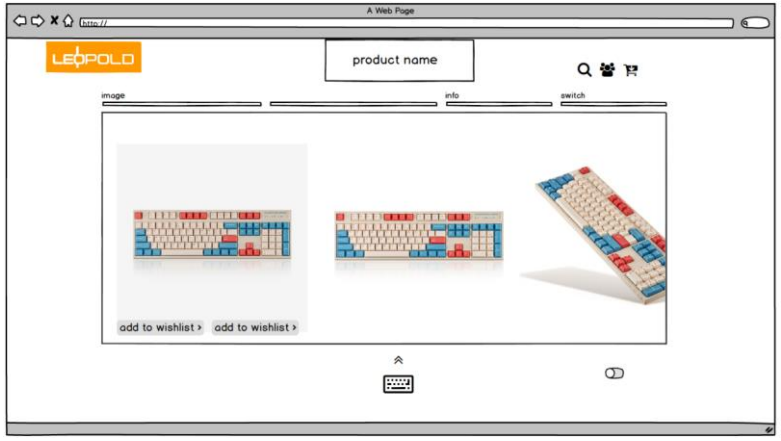
분석 설계 - 웹 페이지

인트로 페이지

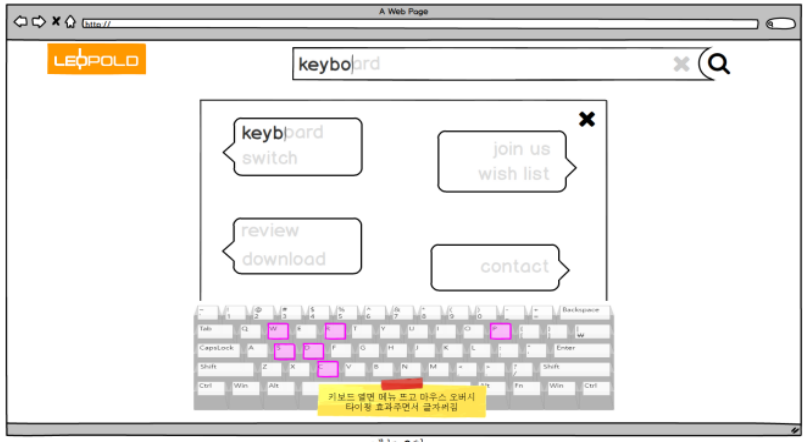


메인 - DT

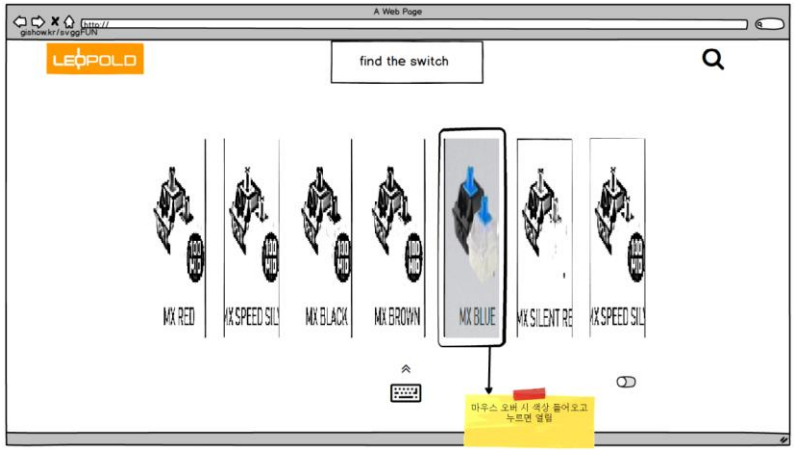
메인 페이지



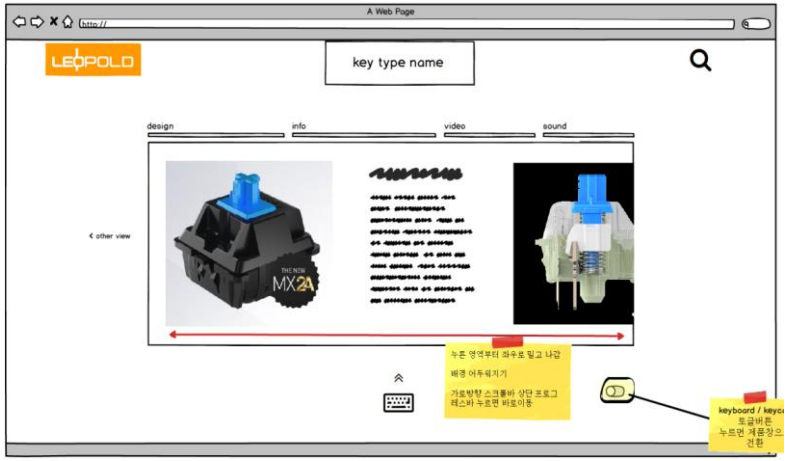
메인 상세 페이지



메뉴 페이지

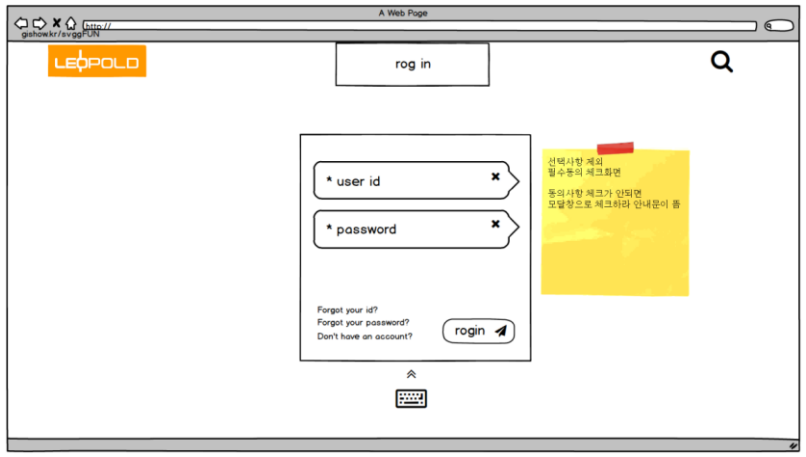


스위치 페이지

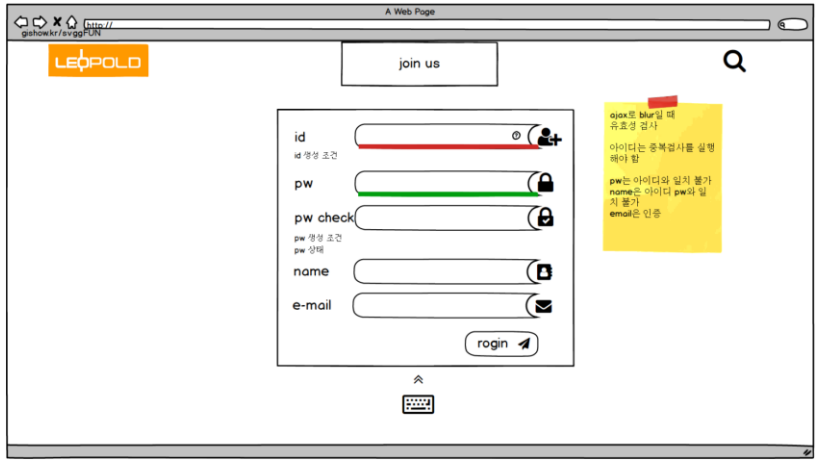


스위치 상세 페이지

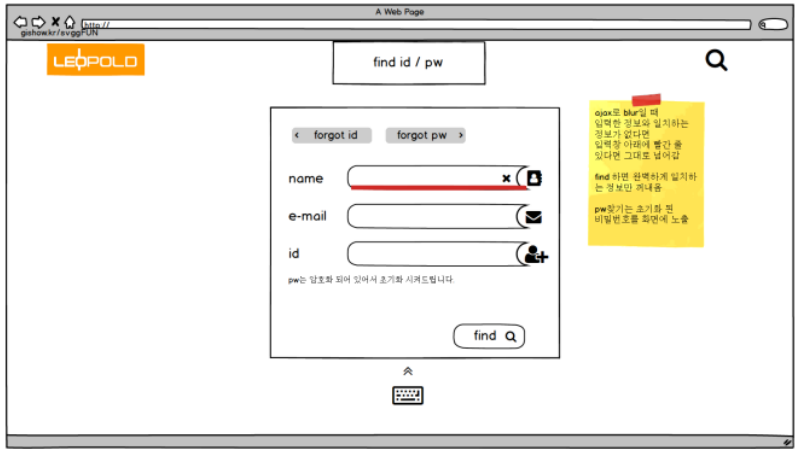
분석 설계 - 웹 페이지



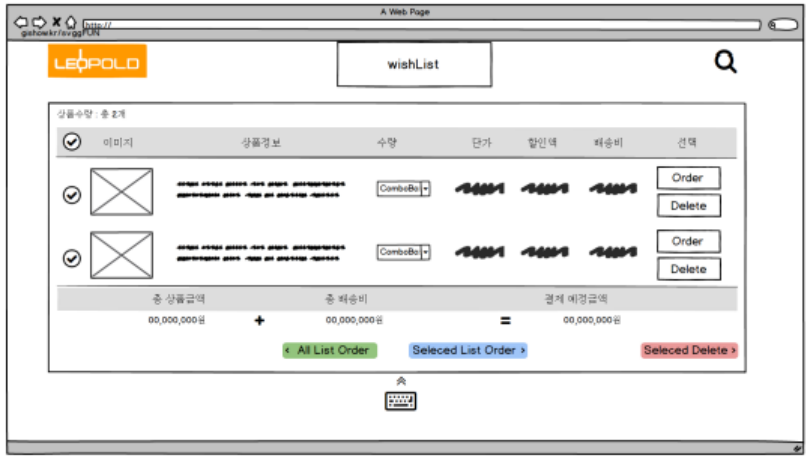
로그인 페이지



회원가입 페이지



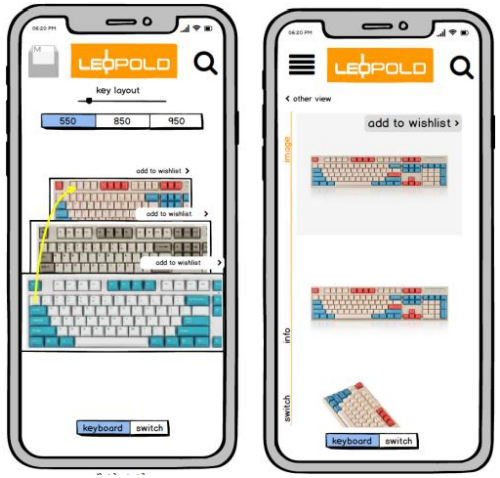
회원찾기 페이지



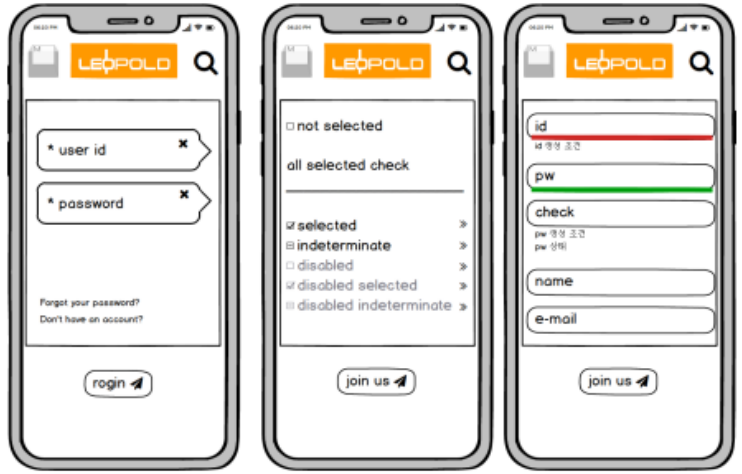
장바구니 페이지

분석 설계 – 모바일

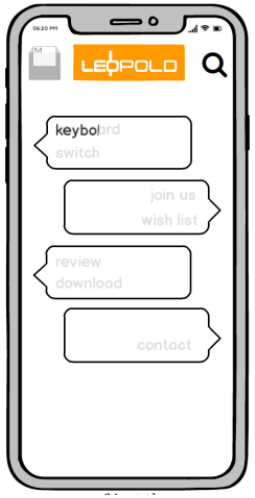
인트로 페이지



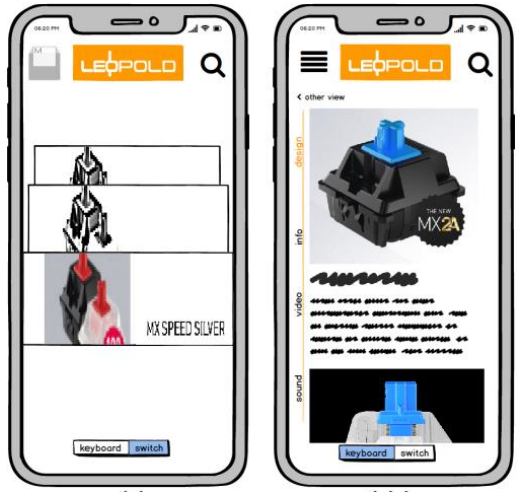
메인 페이지



로그인 페이지



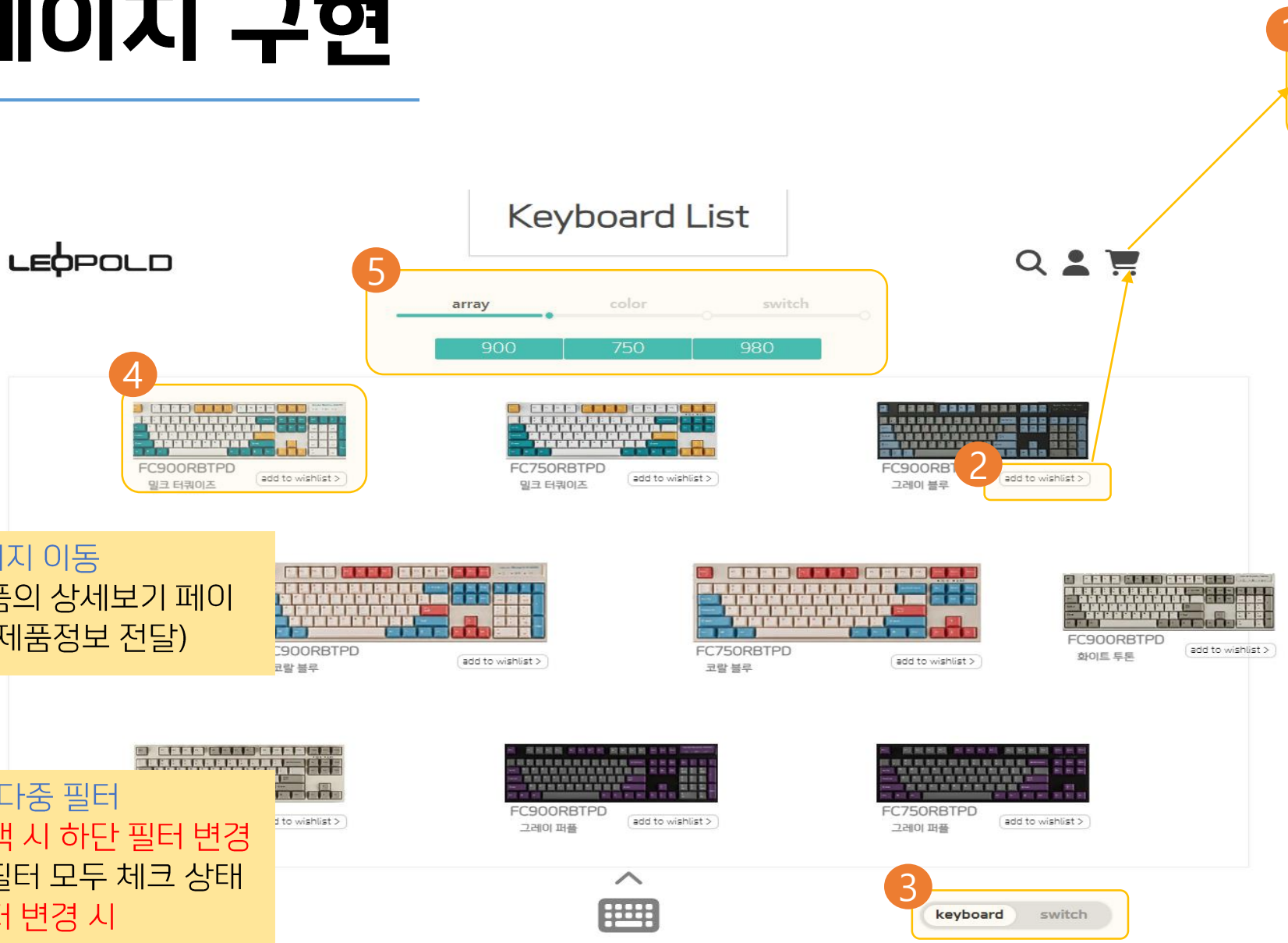
메뉴 페이지

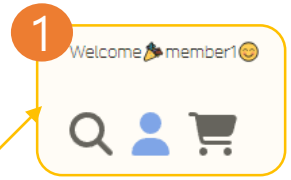


스위치 페이지

장바구니 페이지

페이지 구현





1 세션스토리지에 로그인 정보 저장

로그인 상태

- 아이콘 변경
- 상단에 로그인 아이디 정보
- 로그인페이지 / 회원가입 페이지 접근 불가

로그아웃 시

- 로그인 페이지 접근 가능

2 로컬스토리지에 위시리스트 저장

클릭 시 제품 수량 1개 추가 중복 상품 클릭 시 수량이 증가 장바구니정보는 카트 아이콘을 통해 접근 가능

3 토글 버튼을 통한 페이지 이동

상품을 보기 위한 keyboard와 스위치 정보를 보기 위한 switch페이지 연결

4 상세페이지 이동

클릭 상품의 상세보기 페이지 이동(제품정보 전달)

5 상/하단 다중 필터

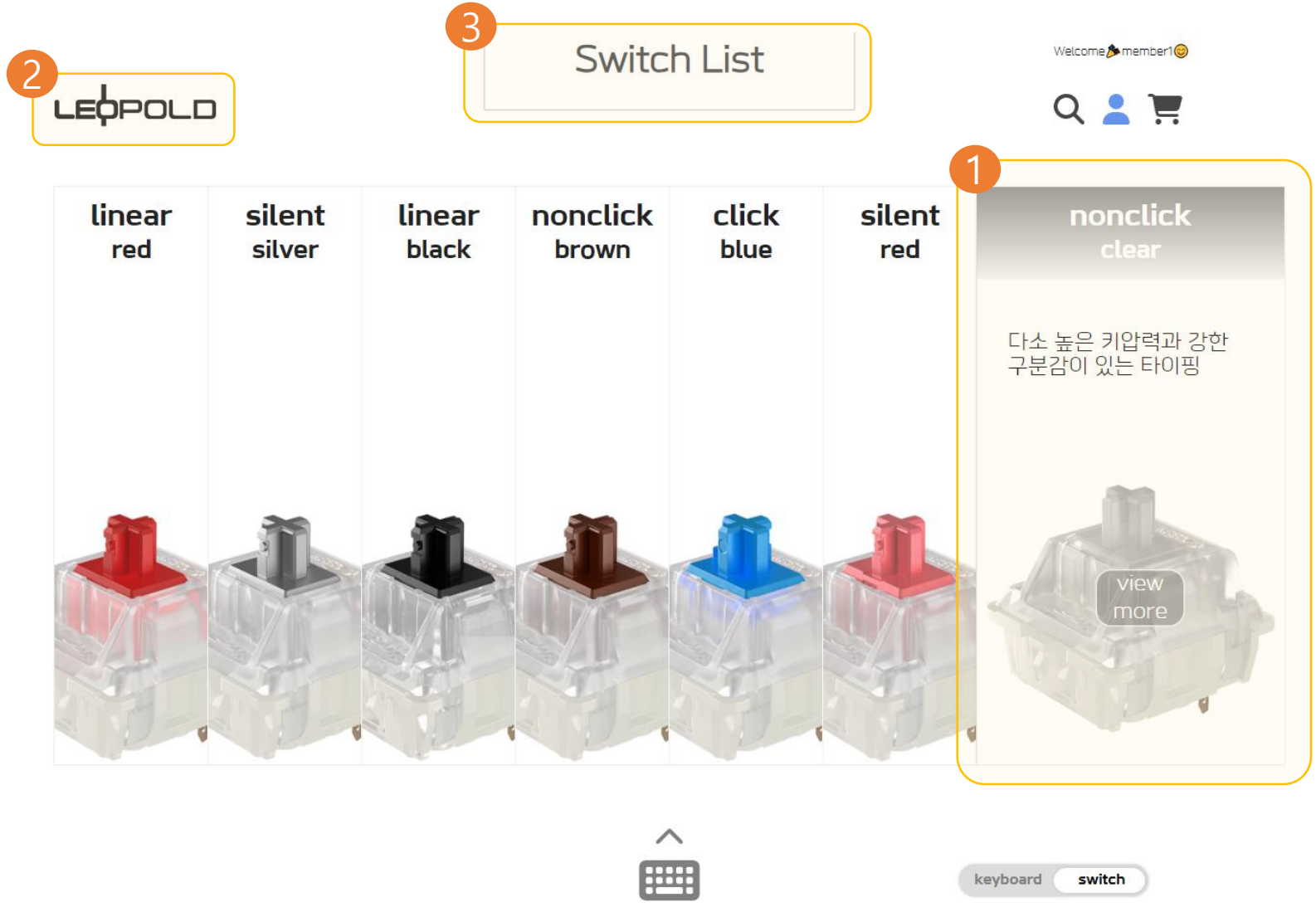
상단 선택 시 하단 필터 변경

- 하단 필터 모두 체크 상태

하단 필터 변경 시

제품 필터

페이지 구현



1

Flex display
마우스 오버 된 스위치를 크게 보여주고 간단한 정보 제공

- 클릭 시 상세페이지 이동
- 상세페이지에 클릭 정보 담아서 이동

2

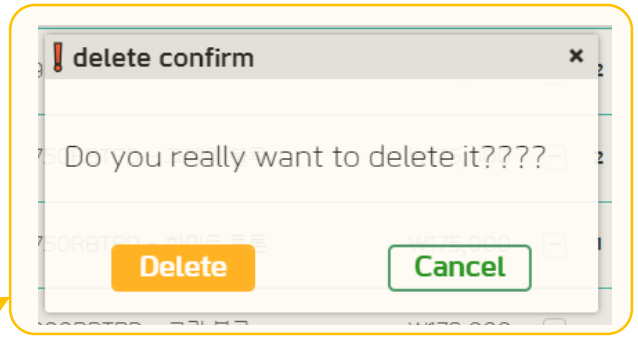
메인 페이지 (keyboard) 이동 버튼
토글 변경을 통한 진입도 가능

링크 정보에 따른 상단 타이틀 변경

페이지 구현

LEOPOLD

Wish List



WishList

1

SubTotal(4 items)

checked order

2

checked delete

	Image	Product	Price	Quantity	Sub Total	Select
☑		FC900RBTPD – 밀크 터쿼이즈	₩178,000	2	₩356,000	order delete
☑		FC750RBTPD – 그레이 블루	₩175,000	2	₩350,000	order delete
☑		FC750RBTPD – 화이트 투톤	₩175,000	1	₩175,000	order delete
☑		FC900RBTPD – 코랄 블루	₩178,000	1	₩178,000	order delete

3

₩1,059,000(Product Price) + ₩2,500(delivery fee) = Total ₩1,061,500

로컬스토리지에 위시리스트 정보 불러오기

1 제품별 수량 업데이트
- 1개 이하 item 2개 이상 items

2 체크 항목 지우기 / 개별 항목 지우기
- 제품 별 체크 된 항목만 삭제
- 개별 delete클릭 시 해당 제품만 삭제
- 체크박스는 전체 체크에만 상단 체크
- 삭제하면 팝업 창 띄우고 Delete 클릭 시 에만 해당제품 삭제
- 합계금액 업데이트

3 제품 수량 변경
- 로컬에 바로 적용
- 합계와 하단 총 금액 업데이트

페이지 구현

1

image1 image2 image3 must-read info1

메인 상세 페이지

white-image black-image infomation motion

스위치 상세 페이지

스위치 정보

성향	클릭감이 좋고 경쾌함
특성	촉각 및 가청 스위칭 특성
키압	60cN 작동력
이동	2.2mm 사전 이동
범위	총 이동거리 4.0mm
소음	High
용도	입문자/타이핑

show related product>

```
// 이미지 세로크기 저장 함수
const sizeCheck = (ele) => {
  // console.log(ele);
  imgWidSize.length = 0;
  // 이동거리 저장용 임시변수
  let xpos = 0;
  // 순번값
  let seq = 0;
  // 이미지 세로크기 배열에 넣기
  ele.each((i, v) => {
    // 개별 이미지 길이 저장
    imgHeiSize[i] = v.height;
    // 가로 이동거리 저장
    xpos += v.width;
    // 이미지 별 이동값 저장(네비용)
    imgWidSize[i] = xpos - limitW <= 0 ? 0 : xpos - limitW;
    if (v.height > v.width) {
      // 가로영역 이동한계값, 세로 한계값, 움직일 요소
      pos[seq] = [xpos - limitW, v.height - limitH, infoImg.eq(i)];
      seq++;
      // console.log(pos);
    }
  });
  // 전체 이동거리 업데이트
  all = xpos - limitW;
  // 위치이동 포인트 설정(이미지 가로길이보다 세로길이가 크면 세로스크롤)
}; //////// 사이즈 저장 함수 ///////////
```

네비게이션
- 클릭으로 내부 정보로 접근
- 하단 요소 크기에 따른 네비게이션 길이 동적 세팅
- 이미지 정보가 긴 경우 네비게이션을 이용하면 상단으로 바로 이동

메인 정보
- 가로 스크롤 이동+세로 스크롤 이동
- 내부 정보 크기에 따라서 이동할 방향을 자동으로 지정해 줌

페이지 구현

```
// 기능 : 가로세로 스크롤을 조합하여 박스를 이동시킨다.
// 파라미터 (이미지 전체박스)
const infoScroll = (delta,x)=>{
  // 휠 이벤트 시 네비설정
  for(let i=0; i < imgWidSize.length; i++){
    if(x <= -imgWidSize[i]){
      addOnNav(i)
    }
  }
  if (x >= 0 && delta > 0) x = 0;
  else if (x >= -pos[0][0]) x = horizonScroll(pos[0], delta);
  else if (x >= -pos[1][0]) x = horizonScroll(pos[1], delta);
  else if (x >= 0 && delta <= 0) x = 0;
  if (x <= -pos[1][0] && delta < 0) x = -pos[1][0];
  infoBox.css("left", x + "px");
}; //////// infoScroll 함수 ///////////////
```

```
// 기능 : 이미지의 세로높이를 받아와서 세로스크롤을 수행한다.
// 파라미터(이동대상, 이동값)
function verticalScroll(target, dir) {
  // target - 대상배열
  y += MOVE * dir;
  if (y > 0) y = 0;
  else if (y < -target[1]) y = -target[1];
  target[2].css("top", y + "px");
} ////////// 세로스크롤 함수 ///////////////
```

```
// 가로스크롤 함수
function horizonScroll(target, dir) {
  // target - 이동대상 dif - 방향
  x += MOVE * dir;
  if (dir === -1) {
    //아래방향 스크롤
    if (x > -target[0]) y = 0; //y고정 x이동
    else if (x <= -target[0] && target[2].position().top != -target[1]) {
      //x고정 y이동
      x = -target[0];
      verticalScroll(target, dir);
    }
  } else {
    //윗방향 스크롤
    if (target[2].position().top != 0) {
      // x고정 y이동
      x -= MOVE * dir; // x축 이동 초기화
      verticalScroll(target, dir);
    }
  }
  return x;
} ////////// 가로스크롤 함수 ///////////////
```

페이지 구현

KEYBOARD

SWITCH

LOGIN

REGISTER

WISHLIST

CONTACT

메뉴 페이지
마우스 오버 시 하단의 키보드 타이핑
마우스 오버 시 상단의 글자 타이핑
클릭으로 링크 이동

하단 키보드 모델링


```
keyData.forEach(ele=>{
    hcode += `
    <div class="key ${ele[0]}">
        <!-- 키 윗면 -->
        <span class="span1 key-part">
        <!-- 키 윗면 글자부분 -->
        <div class="key-top">
            ${insertTop(ele[1])}
        </div>
        </span>
        <!-- left -->
        <span class="span2"></span>
        <!-- right -->
        <span class="span3"></span>
        <!-- top -->
        <span class="span4"></span>
        <!-- bottom -->
        <span class="span5"></span>
        <!-- 키 맨윗면 -->
        <span class="span6"></span>
    </div>
    `;
}); //////keyData forEach문//////////
```

```
// ['키 길이','키보드 자판글자']
export const keyData = [
    ["" , ["~","`"]],
    ["" , ["!","1"]],
    ["" , ["@","2"]],
    ["" , ["#","3"]],
    ["" , ["$","4"]],
    ["" , ["%","5"]],
    ["" , ["^","6"]],
    ["" , ["&","7"]],
    ["" , ["*","8"]],
    ["" , ["(","9"]],
    ["" , [")","0"]],
    ["" , ["-","_"]],
    ["" , ["+","="]],
    ["size0" , "Backspace"],
    ["size1" , "Tab"],
    ["" , "Q"],
    ["" , "W"],
    ["" , "E"],
    ["" , "R"],
    ["" , "T"],
    ["" , "Y"],
    ["" , "U"],
    ["" , "I"],
    ["" , "O"],
    ["" , "P"],
```

개선방향

1. 검색기능 추가
2. 미디어 쿼리 – 제품상세페이지
3. 상세페이지에서 본 페이지로 이동 버튼 구현
4. 인트로 페이지 연결