

Toy Project

한승은, 오현지, 홍성빈

핀터레스트를 선택한 이유

<p>
외부 API를 활용하고 싶어
가장 잘 활용 할 수 있는 레이아웃이 핀터레스트였습니다.

그래서 핀터레스트를 선택 했습니다.

</p>

공동 목표

<div>

협업을 위한 코딩을 위해 중복된 값을 최대한 줄이고
쉽게 유추 가능한 id값을 사용해 보기쉽게 제작 하였고,

서로 만든 코드를 잘 이해할 수 있게
코드마다 설명을 주석 처리 했으며,

공통 관리 구역에 대한 변경이 있을 시
어떤 부분에 변경이 있었는지
서로에게 알려주어
소통을 매끄럽게 하려고 노력했습니다.

</div>

개인 목표

<div>

실제 로그인과 로그아웃
기능과 회원탈퇴 기능을
넣으려고 했고,

아직 배우진 않았지만
서버에 대한 개념을
이해 하고 직접 사용하는것에
목표를 두었습니다.

</div>

<div>

이전 토이프로젝트때 태그 영역을
정렬하는 것조차 익숙치 않아
많이 헤맸지만,

이번 프로젝트 때는
다양한 JS 라이브러리와
jQuery, html, css 코딩에 익숙해지고자
프론트엔드 작업을 맡았습니다.

</div>

<div>

누구든 이용 가능한 작은 웹사이트 하나
를 완성해 보는 것에 목표를 두었습니다.

실제 서비스중인 웹사이트들 처럼 가능한
한 대부분의 예외 상황들에 대응할 수 있
도록 로직을 짜보고자 했으며,

또한 사용자의 편의를 위한 ui와 그에 따
른 동작을 할당할 수 있도록 노력했습니
다.

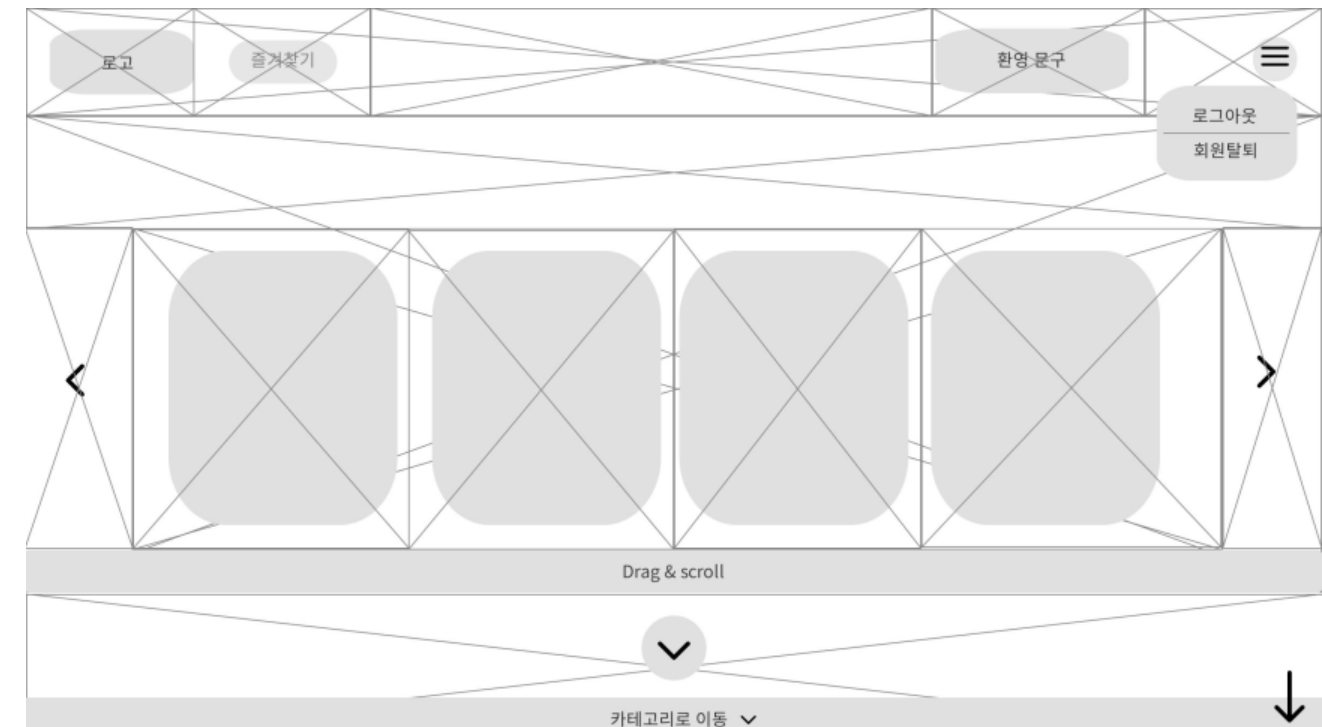
</div>

페이지 소개(와이어 프레임)

로그인 페이지

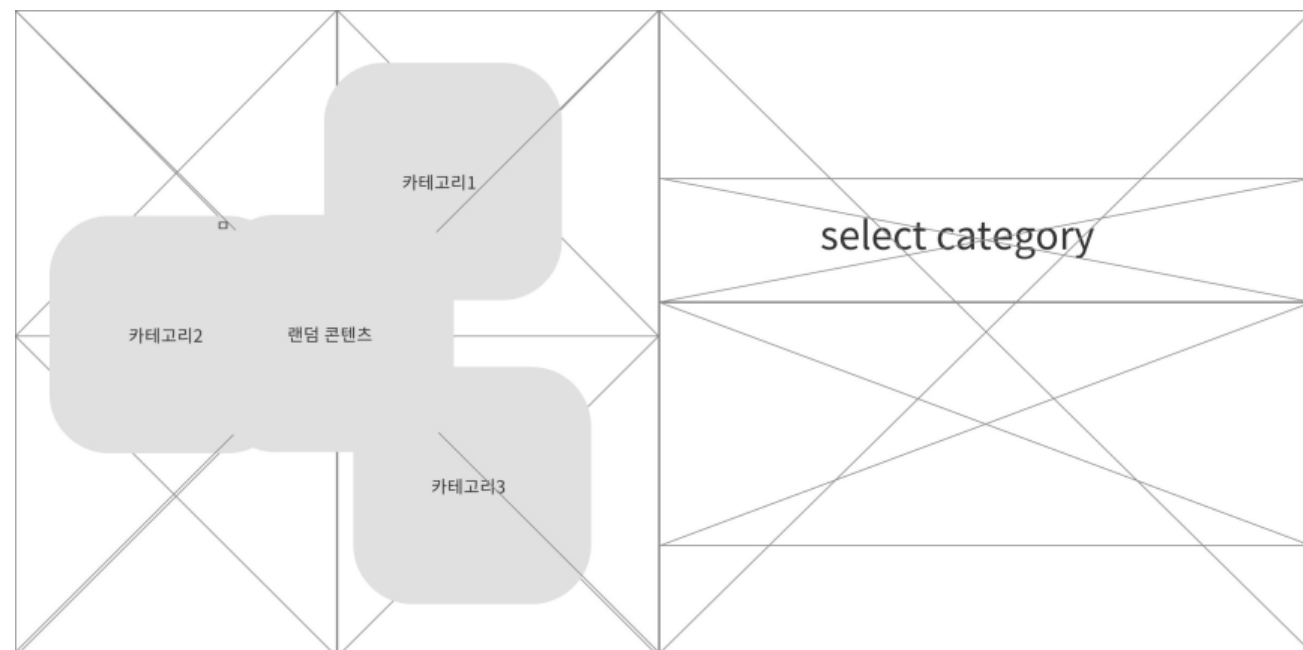


메인 페이지(slick_list1)

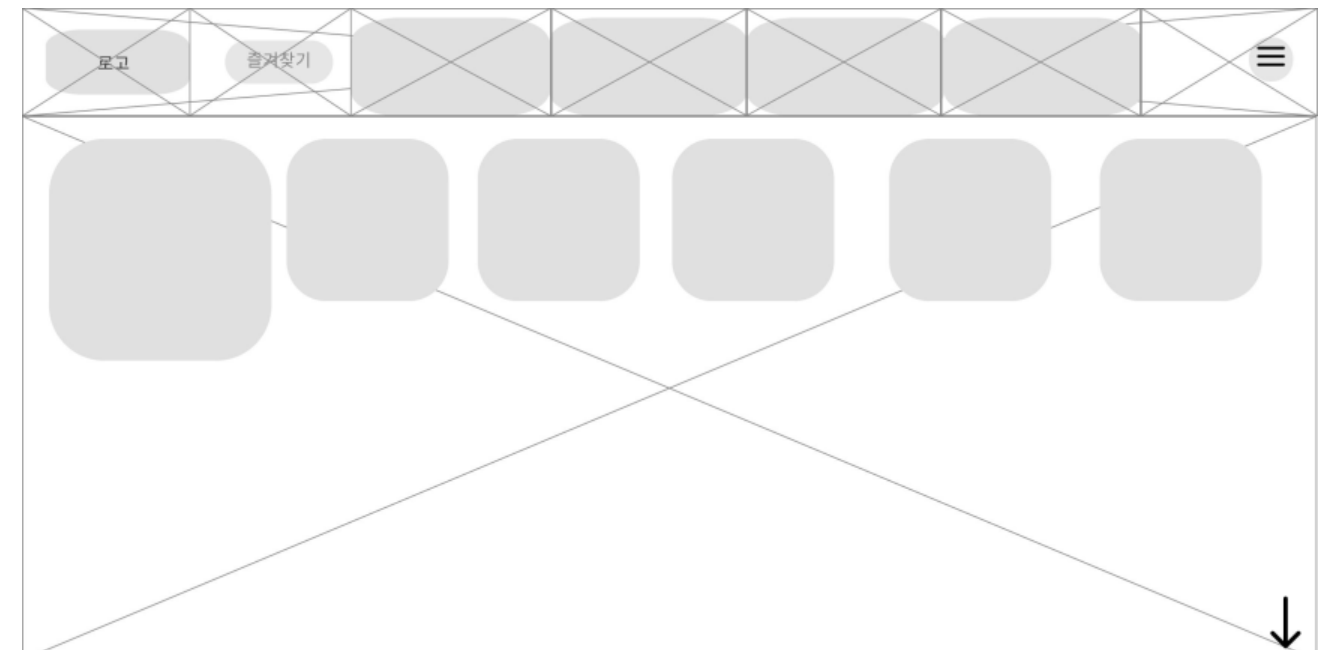


페이지 소개(와이어 프레임)

메인 페이지(slick_list2)

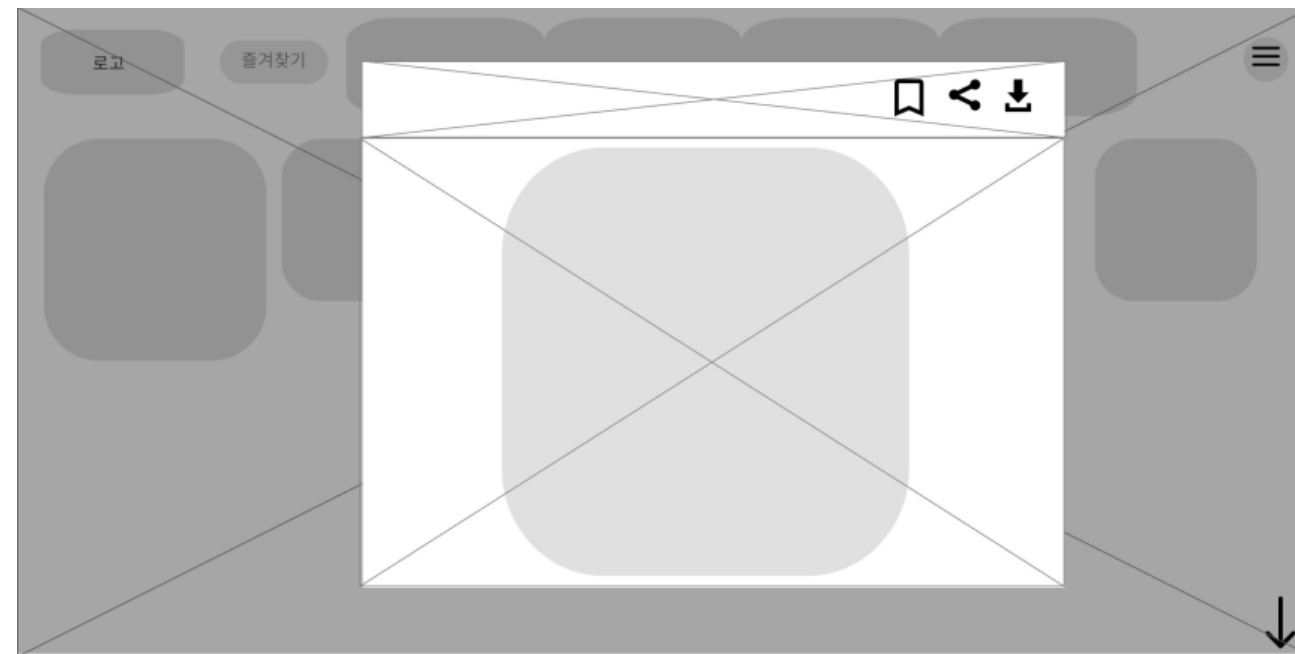


카테고리 페이지



페이지 소개(와이어 프레임)

상세 페이지(이미지 클릭 시)



페이지 소개

<https://hsy3373.github.io/kh-toyproject2/>

페이지 소개

```
$('.autoplay').slick({
  slidesToShow: 4,
  slidesToScroll: 1,
  autoplay: true,
  autoplaySpeed: 2700,
  infinite: true,
  dots: false,
  prevArrow: $('.prevArrow'),
  nextArrow: $('.nextArrow'),
  speed: 2200
  // cssEase: "linear",
  // ,
  // appendDots: $(".dots")
});
```

```
$('.slickSlide').slick({
  slidesToShow: 1,
  slidesToScroll: 1,
  draggable: false,
  autoplay: false,
  infinite: false,
  vertical: true,
  dots: false, // 테스트용 도트
  nextArrow: $('.goCt'),
  prevArrow: $('.prevNone'),
  // initialSlide: 0 => default
});
```

jquery

- 1).autoplay_slick
- 2).slickSlide_slick
- 3)wheel_event
- 4).categoryName_click

```
// 마우스 휠로 슬라이드 이동
$('.slickSlide').on('wheel', function (e) {
  e.preventDefault();

  if (e.originalEvent.deltaY < 0) {
    $(this).slick('slickPrev');
  } else {
    $(this).slick('slickNext');
  }
});
```

```
// 카테고리 선택될때마다 배경색 부여(회색)
$('.categoryName').click(function () {
  $(this).addClass('selectedCategory');
  var notClicked = $('.categoryName').not(this);
  notClicked.removeClass('selectedCategory');

  // 카테고리 정보도 변경해줌
  changeCategory($(this).attr('id'));
});
```

페이지 소개

```
$("#exit").click(function(){
  let ulist = JSON.parse(localStorage.getItem('userList'));
  let cUser = JSON.parse(localStorage.getItem('currentUser'));
  if(ulist && cUser){
    if(ulist[cUser]){
      let userPwval = document.getElementById("outPassword").value;
      let userPw = JSON.parse(localStorage.getItem("userPw"));
      var decrypt = CryptoJS.enc.Base64.parse(ulist[cUser].Pw);
      var hashData= decrypt.toString(CryptoJS.enc.Utf8);
      if(userPwval === hashData){
        delete ulist[cUser];
        alert("삭제 되었습니다.");
      } else {
        alert("비밀번호를 확인 해 주세요.");
        return;
      }
    }
  } else {
    alert("회원가입부터 해주세요.");
    return;
  }
  localStorage.setItem('userList', JSON.stringify(ulist));
});
```

로컬 스토리지에 암호화 돼 있는
비밀번호를 디버깅 해서
유효검사 후
키 , 밸류값을
삭제하는 구문(회원 탈퇴).

```
let ulist = JSON.parse(localStorage.getItem('userList'));
let user = document.getElementById("userEmail2").value;
if(ulist){
  ulist[user] = {
    email : user ,
    Pw : base64 ,
    favorite : {} ,
  };
} else {
  ulist = {
    [user] : {
      email : user ,
      Pw : base64 ,
      favorite : {} ,
    }
  }
}
localStorage.setItem('userList', JSON.stringify(ulist));
```

로컬 스토리지에 객체로
저장하는 구문.

```
var inputText = document.getElementById('userPw2').value;

console.log('< Base64 인코딩 및 디코딩 >');
var key = CryptoJS.enc.Utf8.parse(inputText);
var base64 = CryptoJS.enc.Base64.stringify(key);
console.log('인코딩된 값 : ' + base64);
var decrypt = CryptoJS.enc.Base64.parse(base64);
var hashData = decrypt.toString(CryptoJS.enc.Utf8);
console.log('디코딩된 값 : ' + hashData);
```

사용자가 입력한 값을
암호화 하는 구문.



다음

집안 꾸미기 아이디어를 찾아보세요

THANK YOU

