

Project 4

RIDI BOOKS

유혜림



CONTENTS

- 기획
 - 주제선정
 - 페르소나
 - 일정표
- 화면 설계
 - 와이어 프레임
 - 프로토 타입
- 코드 구현
 - Main page
 - Sub page
- 평가/소감



RIDIBOOKS

리디북스를 선정한 이유

- 매력적인 **메인 슬라이드**

책이 책장에 꽂혀있다가 다음 슬라이드로 변경되며 마치 책이 펼쳐지는 듯한 효과를 주는 슬라이드를 보니 어떻게 구현한 것인지 흥미가 생겼습니다.

- 깔끔하고 세심한 **UI**

부분 포인트를 넣어 세련되며, 다양한 효과가 적용되어 있습니다. 또한 사용자가 정보를 알기 쉽도록 배치되어 있으며 전체적으로 깔끔하게 구성되어 있습니다.

- 친숙한 브랜드

최근, 종이 책보다 많은 사람들이 항상 들고 다니는 핸드폰으로 쉽게 볼 수 있는 e-book이 선호도가 더 높습니다. 그 중 친숙한 브랜드인 리디북스를 주제로 선정하였습니다.



이름 : 박나라

나이 : 34세

직업: 식품 관련 대기업 홍보팀 대리

사는 곳 : 수원시 영통구

PERSONA

라이프 스타일

서울에 있는 식품업 회사의 홍보팀 대리로 근무하고 있다. 미혼 여성으로 가족과 함께 수원에 거주하고 있으며, 밤낮으로 서울로 지하철을 타고 출퇴근을 한다. 지금껏 지루한 출퇴근 시간을 즐거나 유튜브 영상을 보며 보냈지만, 새 해 계획인 ‘일 년에 10권 책 읽기’를 도전하며 자기 개발을 하려 한다. 출퇴근 시간이 길기 때문에 그 시간 동안 핸드폰으로 책을 읽을 생각이다.

바라는 점

- 움직이는 지하철에서 작은 화면인 핸드폰으로 책을 읽어야 하는 만큼 글씨체의 크기와 화면 밝기 등을 자유롭게 조절해서 이용할 수 있으면 좋겠다.
- 가끔은 눈을 감고 들을 수 있도록 오디오 북이 있으면 좋겠다.
- 키워드들을 선택해서 업무와 관련된 책들이 자동 추천되는 기능이 있으면 좋겠다.



이름 : 주우현

나이 : 32세

직업: 요리사

사는 곳 : 서울시 영등포구

PERSONA

라이프 스타일

유럽에서 요리 학교를 졸업하고, 서울에 돌아와 조그만 원 테이블 레스토랑을 운영하며 요리도 하고 있다. 가게 내부에 메뉴판은 따로 없이 ‘그날 요리사 마음대로’라고 쓰여있다. 매일을 다른 요리를 하고 그걸 맛있게 먹는 손님을 보는 게 우현의 목표이자 꿈이기에, 쉬는 날마다 항상 새로운 음식을 해보려고 책을 보면 공부 중이다.

바라는 점

- 요리를 하며 확인할 수 있도록 책을 읽어주거나 자동으로 페이지를 넘겨주는 기능이 있으면 좋겠다.
- 다양한 요리 책 중에서도 나라 별로 카테고리가 나뉘어 있어 원하는 요리를 찾기 편리했으면 좋겠다.
- 책 속의 요리를 알 수 있도록 사진을 간략하게 보여주는 페이지가 있길 원한다.

TIMELINE

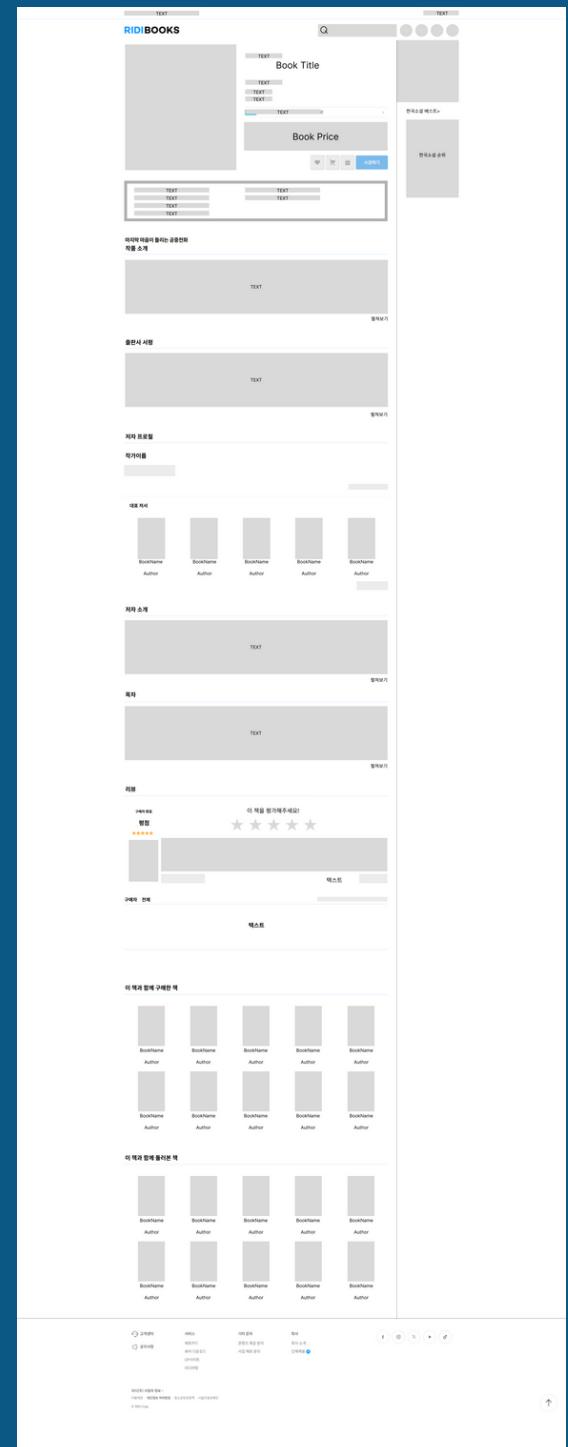


Project 4

WIREFRAME



MAIN PAGE



SUB PAGE

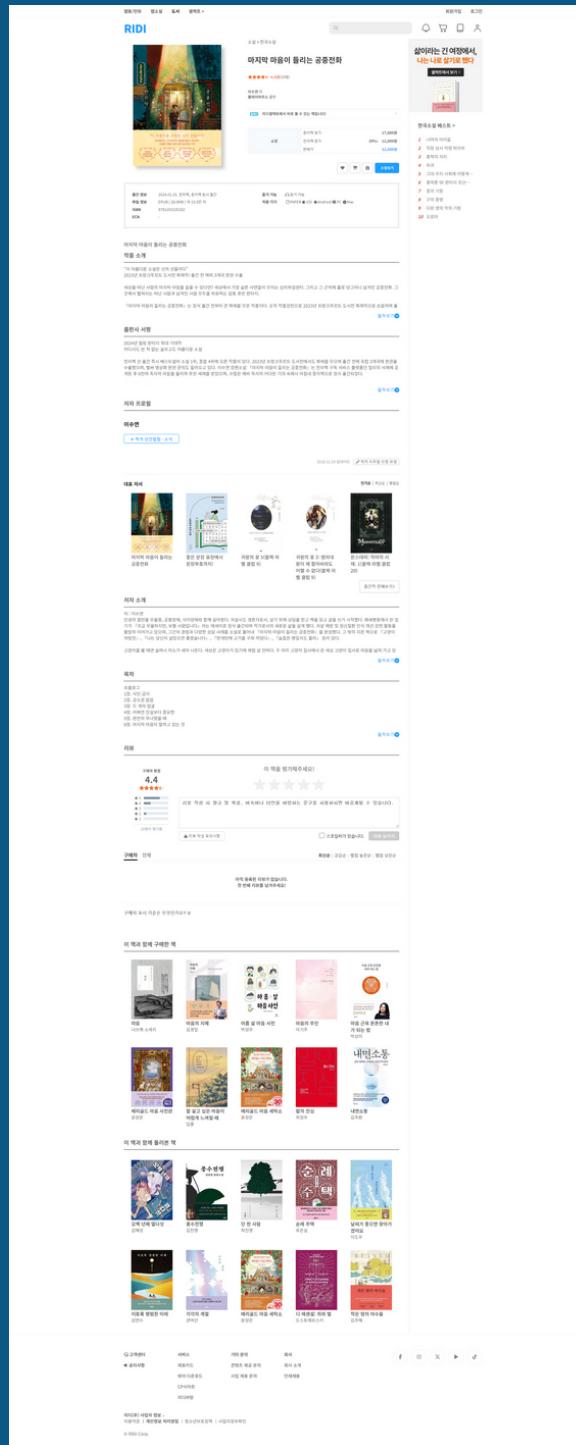


Project 4

PROTOTYPE



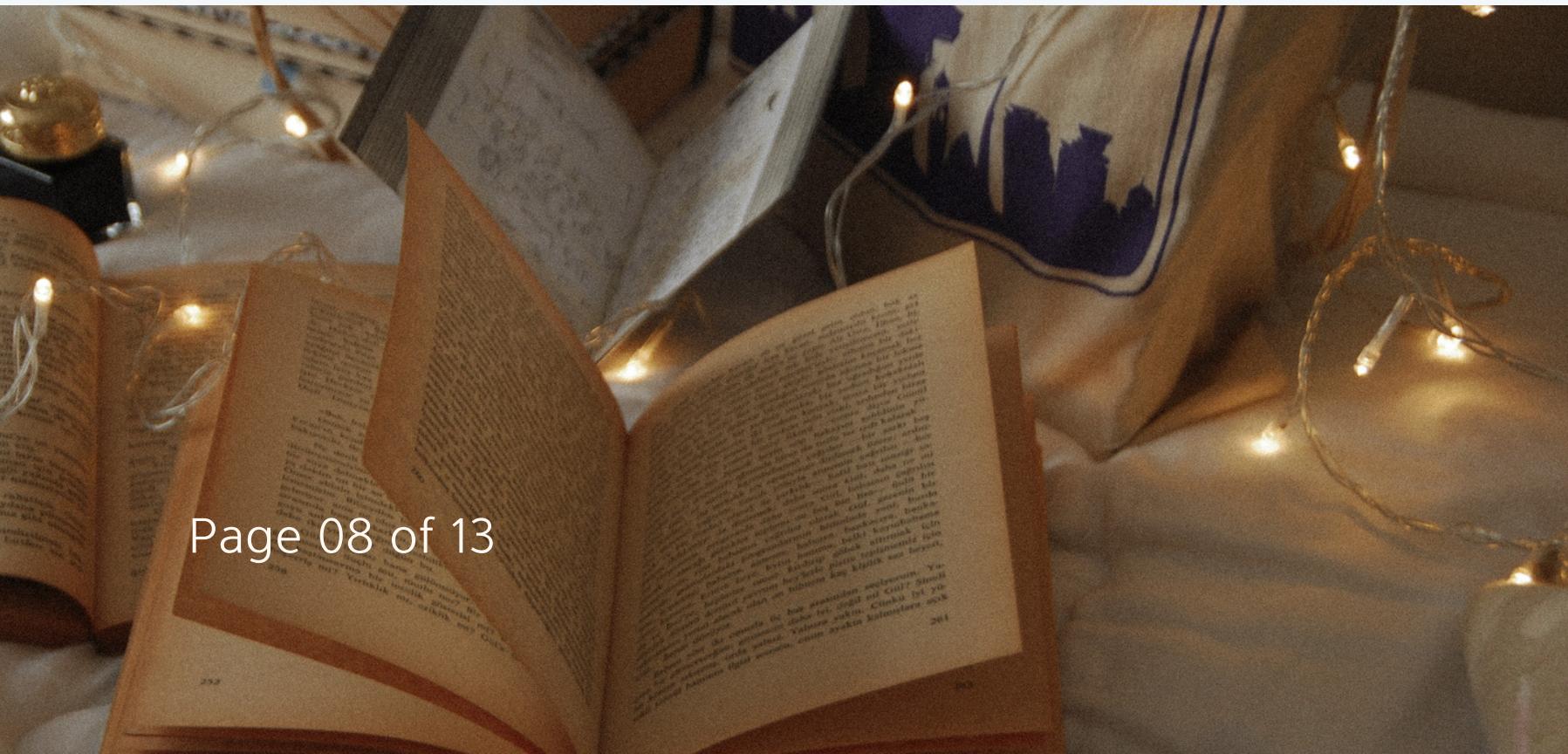
MAIN PAGE



SUB PAGE

Project 4

CODE



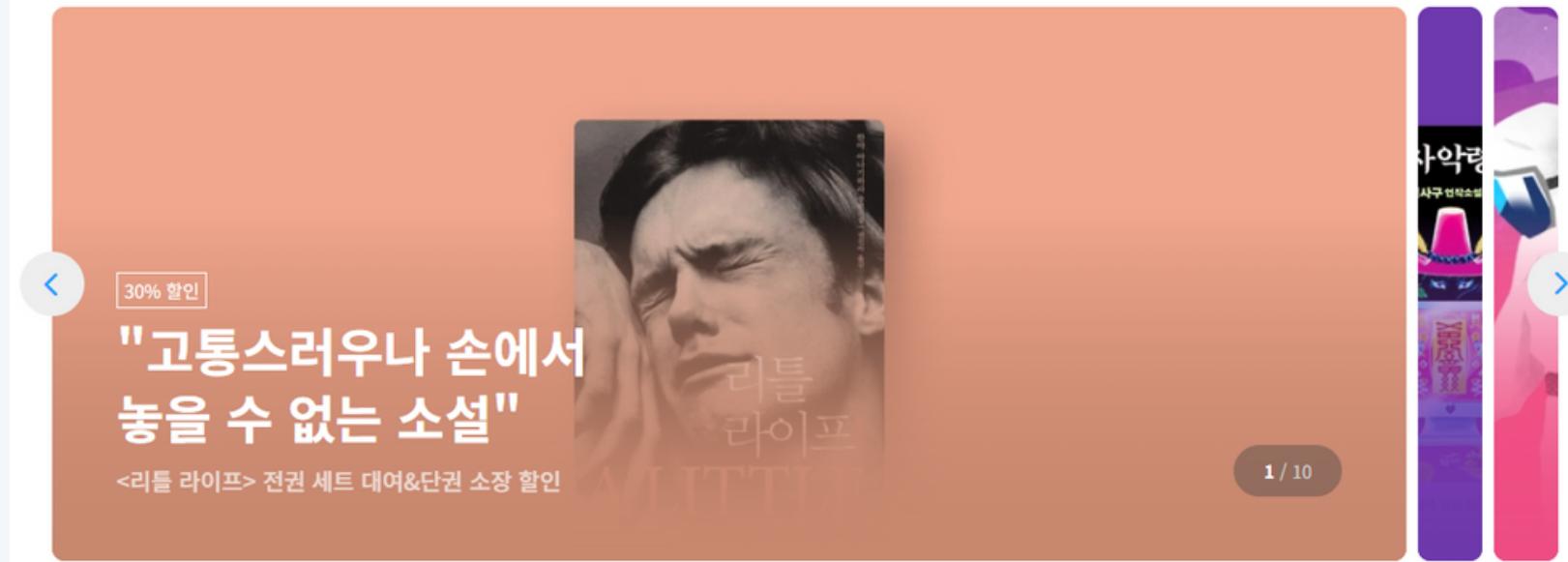
Page 08 of 13



- MAIN PAGE
- SUB PAGE



RIDIBOOKS



페이지 로드 시 자동 슬라이드와 좌, 우 버튼을 클릭하면 슬라이드 이전, 이후 페이지가 보일 수 있게 구현하였습니다. flexGrow와 flexBasis를 사용하여 마치 책이 접혀있다가 펼쳐지는 것 같은 효과를 주었습니다.

```

$('.slide li').eq(0).find('.num').show();

// 자동 슬라이드
function slide() {
  let lis = $('.slide li')

  $(lis).eq(0).find('.num').fadeOut()

  $(lis).eq(0).stop().animate({ flexGrow: 0, flexBasis: 0 }, 800)
  $(lis).eq(1).stop().animate({ flexGrow: 1, flexBasis: 1051 }, 800, function () {
    $(this).find('.num').fadeIn();
    $('.slide li:first').appendTo('.slide');
    $('.slide li:last').stop().animate({ flexGrow: 0, flexBasis: 50 }, 400, function ()
    () {
      $(this).find('.num').hide();
    });
  })
}

let autoslide = setInterval(slide, 5000)

// 버튼 누르면 슬라이드 이전, 이후
$('.slideWrap .btn .prev').click(function () {
  clearInterval(autoslide)
  let lis = $('.slide li')
  $(lis).eq(0).find('.num').fadeOut()

  $(lis).eq(0).stop().animate({ flexGrow: 0, flexBasis: 50 }, 800, function () {
  })
  $(lis).eq(-1).stop().animate({ flexGrow: 1, flexBasis: 1051 }, 800, function () {
    $(this).find('.num').fadeIn();
  })
  $('.slide li:last').prependTo('.slide');
  autoslide = setInterval(slide, 5000)
})

$('.slideWrap .btn .next').click(function () {
  clearInterval(autoslide)
  slide()
  autoslide = setInterval(slide, 5000)
})

```



Ajax를 이용하여 책 정보를 불러오고,
버튼을 클릭하면 다음 책이 좌측 박스로
옮겨가서 썸네일이 크게 보이고 콘텐츠 내용이 더 자세히 보이도록 구현하였습니다.

```
// new boxR
var swiper2 = new Swiper(".new .boxR.mySwiper", {
  slidesPerView: 3,
  spaceBetween: 10,
  loop : true,
  loopAdditionalSlides : 1,
  allowTouchMove : false,
  navigation: {
    nextEl: ".new .swiper-button-next",
    prevEl: ".new .swiper-button-prev",
  },
  on: {
    init: function () {
      $('.new .btn .prev').hide();
    },
    slideChange: function(){
      let idx = this.activeIndex;
      if (idx != 0){
        $('.new .btn .prev').fadeIn();
      }
      if(idx != 3){
        $('.new .btn .next').fadeIn();
      }
    },
  }
});
```

`// new boxR`

```
var swiper2 = new Swiper(".new .boxR.mySwiper", {
  slidesPerView: 3,
  spaceBetween: 10,
  loop : true,
  loopAdditionalSlides : 1,
  allowTouchMove : false,
```

`navigation: {`

```
  nextEl: ".new .swiper-button-next",
  prevEl: ".new .swiper-button-prev",
},
```

`on: {`

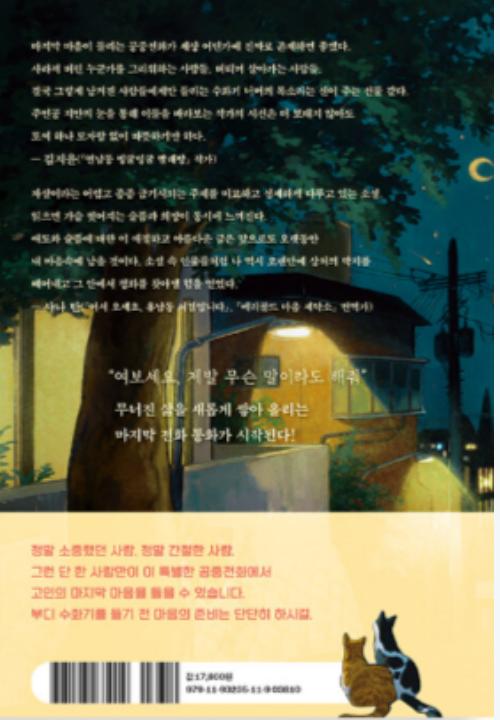
```
  init: function () {
    $('.new .btn .prev').hide();
  },
  slideChange: function(){
    let idx = this.activeIndex;
    if (idx != 0){
      $('.new .btn .prev').fadeIn();
    }
    if(idx != 3){
      $('.new .btn .next').fadeIn();
    }
  },
});
```

SUB PAGE



책의 실제 모습과 비슷하게 입체적으로 보이는 효과를 주기 위해, 페이지 로드 시에는 책의 앞 표지가 보이다가 호버하면 책의 사이드 부분이 보여지고, 이후 클릭 시 rotate되며 표지 뒷 부분이 보여지도록 구현하였습니다.

▼ 뒤 표지

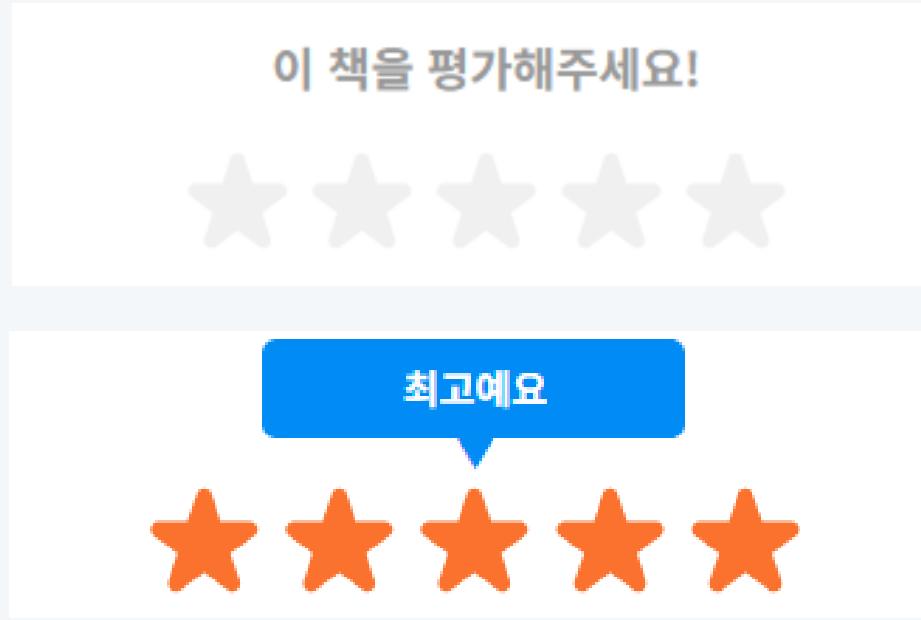


▲ 사이드



```
#contents .intro #bookCover {  
    width: 40%;  
    height: 440px;  
    perspective: 1000px;  
    transform-style: preserve-3d;  
    cursor: pointer;  
}  
#contents .intro #bookCover .turn_img {  
    width: 80%;  
    height: 90%;  
    position: relative;  
    transition: 0.5s;  
    transform-style: preserve-3d;  
    box-shadow: 10px 10px 10px rgba(0, 0, 0, 0.2);  
    margin: 20px 0 0 30px;  
    cursor: url(..../img/sub/ico_flip.svg), pointer !important;  
}  
#contents .intro #bookCover .turn_img:hover {  
    transform: rotateY(35deg);  
}  
#contents .intro #bookCover .turn_img.rotate {  
    transform: rotateY(180deg);  
}  
#contents .intro #bookCover .turn_img img {  
    width: 100%;  
    height: 100%;  
    position: absolute;  
    top: 0;  
    left: 0;  
}  
#contents .intro #bookCover .turn_img::before {  
    content: "";  
    width: 30px;  
    height: 100%;  
    position: absolute;  
    background-image: url(..../img/sub/cover_s_fix.jpg);  
    background-repeat: no-repeat;  
    background-size: cover;  
    background-position: center;  
    transform-origin: left;  
    transform: rotateY(90deg);  
    left: 1px;  
}  
#contents .intro #bookCover .turn_img::after {  
    content: "";  
    width: 100%;  
    height: 100%;  
    position: absolute;  
    background-size: cover;  
    background-position: center;  
    transform-origin: center;  
    transform: rotateY(180deg) translateZ(30px);  
    width: 100%;  
    height: 100%;  
    left: 2px;  
}
```

SUB PAGE



호버하면 별이 다섯 단계로 나뉘어 차례로 채워지고, 별점 위 부분에 각 설명이 나타나게 구현하였습니다. 이후 클릭하면 별점이 고정되도록 하되, 호버하면 설명만 바뀌어 별점 수정이 가능하도록 하였습니다. 호버하여 점수에 아무런 변화가 없을 경우 원래 클릭한 별점이 유지되도록 하였습니다.

```
<div class="reviewwrite">
  <div class="userReview">
    <h4>이 책을 평가해주세요!</h4>
    <div class="starRatebox">
      <p><span>▼</span>별로예요</p>
      <p><span>▼</span>그저 그래요</p>
      <p><span>▼</span>보통이에요</p>
      <p><span>▼</span>좋아요</p>
      <p><span>▼</span>최고예요</p>
    </div>
  </div>
  <div class="starRate">
    <div class="emptybox"></div>
    <input type="radio" name="rating" id="rate1" value="5">
    <label for="rate1">★</label>
    <input type="radio" name="rating" id="rate2" value="4">
    <label for="rate2">★</label>
    <input type="radio" name="rating" id="rate3" value="3">
    <label for="rate3">★</label>
    <input type="radio" name="rating" id="rate4" value="2">
    <label for="rate4">★</label>
    <input type="radio" name="rating" id="rate5" value="1">
    <label for="rate5">★</label>
    <div class="emptybox"></div>
  </div>
```

```
.review .reviewbox .reviewwrite .starRate {
  width: 100%;
  margin-bottom: 20px;
  display: flex;
  justify-content: center;
  flex-direction: row-reverse;
}
.review .reviewbox .reviewwrite .starRate .emptybox {
  width: 50%;
}
.review .reviewbox .reviewwrite .starRate label {
  font-size: 35px;
  color: transparent;
  text-shadow: 0 0 0 ■#f0f0f0;
  cursor: pointer;
}
.review .reviewbox .reviewwrite .starRate label:hover {
  text-shadow: 0 0 0 ■#FA722E;
}
.review .reviewbox .reviewwrite .starRate label:hover ~ label {
  text-shadow: 0 0 0 ■#FA722E;
}
.review .reviewbox .reviewwrite .starRate input[type=radio] {
  display: none;
}
.review .reviewbox .reviewwrite .starRate input[type=radio]:checked ~ label {
  text-shadow: 0 0 0 ■#FA722E;
}
```

```
// 책 평가 별점
$('.emptybox').click(function(){
  $('input[name="rating"]').prop('checked', false);
  $('.starRatebox').hide();
  $('.userReview>h4').css('opacity', 1);
})

function handleStarRateChange() {
  // 호버와 체인지 이벤트 처리
  $('.starRate label').hover(function () {
    let i = $(this).index()
    let j = $('.starRate').children().eq(i - 1).val()

    $('.starRatebox').css('display', 'block');
    $('.starRatebox p').eq(j - 1).css('display', 'block').siblings().css('display', 'none');
    $('.userReview>h4').css('opacity', 0);
  }, function () {
    // 체인지 이벤트에서 체크된 경우 처리하지 않도록 조건 추가
    var selectedRating = $('input[name="rating"]:checked').val();

    if($('input[name="rating"]:checked').length === 0) {
      $('.starRatebox').hide();
      $('.userReview>h4').css('opacity', 1);
    }

    $('.starRatebox p').eq(selectedRating - 1).css('display', 'block').siblings().css('display', 'none');
  });

  // 체인지 이벤트 처리
  $('input[name="rating"]').change(function () {
    var selectedRating = $('input[name="rating"]:checked').val();

    // 선택된 값이 있는지 확인
    if (selectedRating !== undefined) {
      console.log('선택된 별점: ' + selectedRating);

      $('.userReview>h4').css('opacity', 0)
      $('.starRatebox').css('display', 'block');
      $('.starRatebox p').eq(selectedRating - 1).css('display', 'block').siblings().css('display', 'none');

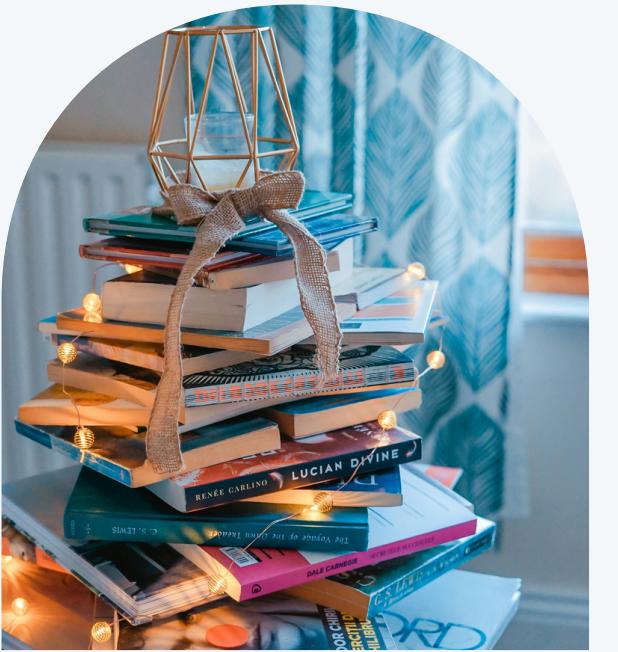
    } else {
      // 체크가 해제된 경우 처리
      $('.starRatebox').hide();
      $('.userReview>h4').css('opacity', 1);
    }
  });
}

$(document).ready(function () {
  handleStarRateChange();
});
```

REVIEW

SCSS와 Open Api를 이용하여 현재 서비스 중인 서점 사이트를 클론 코딩을 해보았습니다. 평소 익숙하게 이용했던 사이트의 구조를 가진 만큼 구현이 어렵지 않을 것이라고 생각했으나 메인 페이지의 슬라이드와 서브 페이지의 리뷰 별점을 구현하는 데에 생각보다 시간이 더 필요했었습니다. 하지만 지금껏 해보지 않았던 기능들을 이번 프로젝트에서 구현해봄으로써 조금 더 자신감을 가질 수 있게 되었고, 또한 Swiper와 같은 라이브러리를 이용하여 시간 관리에 효율적인 방법을 많이 알 수 있던 프로젝트였습니다.





THANK YOU! *