

# RIDI 서점 사이트제작

First project

Design by. 백기림

# INDEX

01 홈페이지 선정 및 방향

02 페르소나

03 홈페이지 구현

HTML

CSS

JAVA Script

04 보완점

홈페이지 선정 및 방향

01

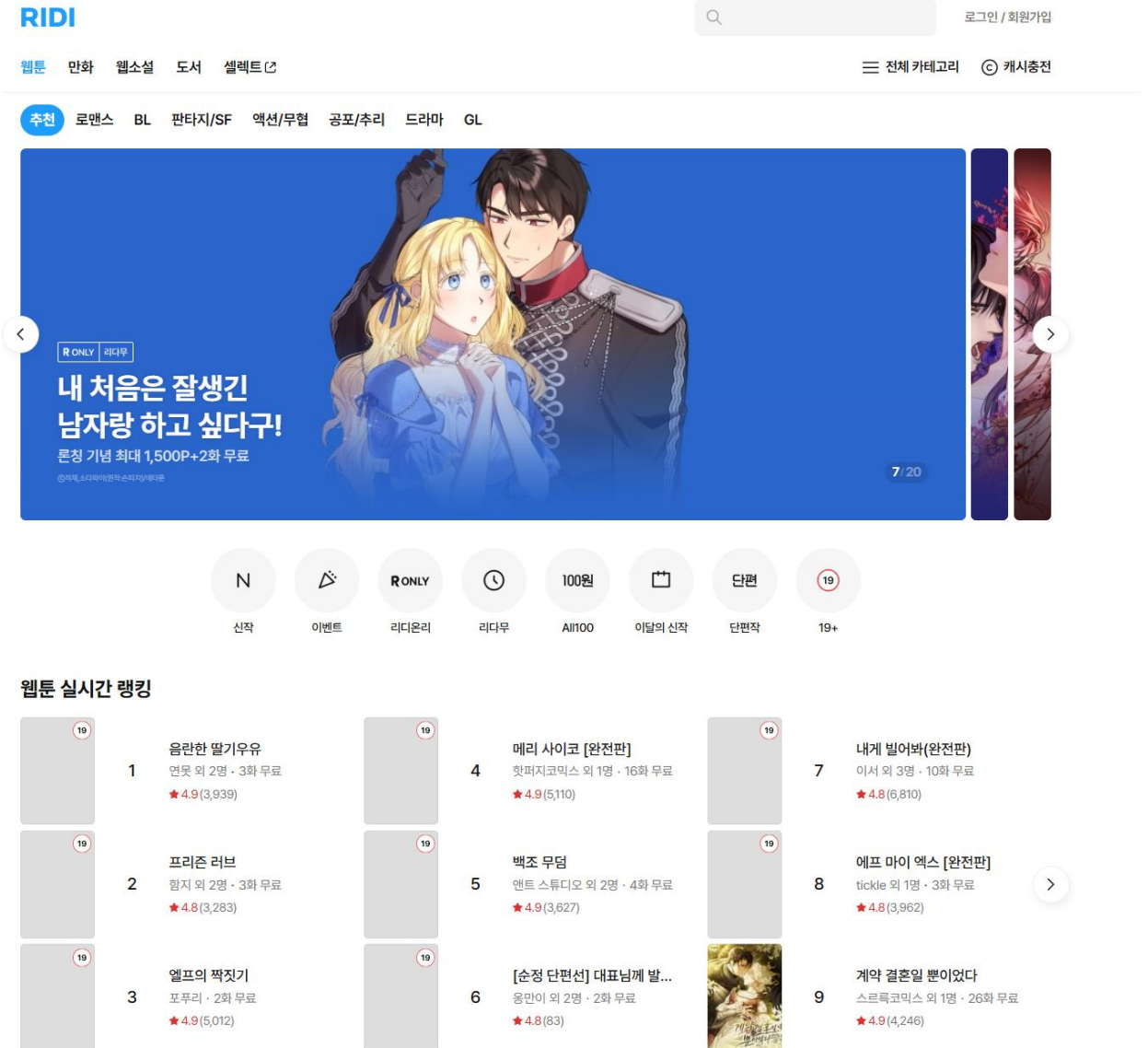
02

03

04

## 홈페이지 선정 및 방향

디자인 구성이 깔끔하고,  
애니메이션 구현이 다양하게  
구성되어 있어 선택



페르소나

01

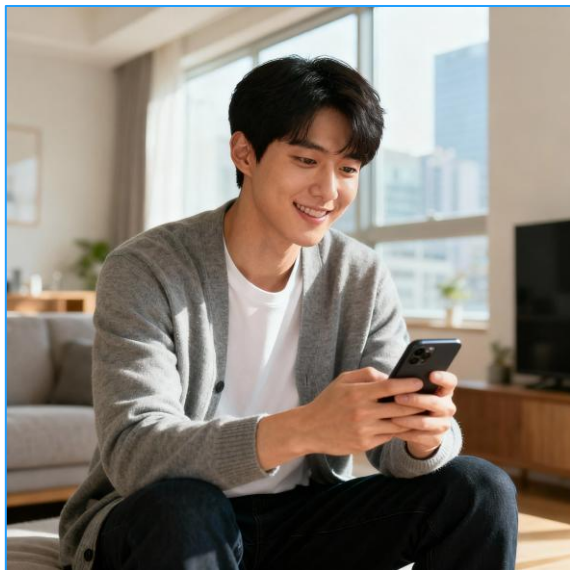
## 페르소나

02

03

04

## 최준수



나이 : 25세  
 직업 : 대학생  
 지역 : 울산  
 성격 : 여유있고, 탐구적인 소비지향

## 라이프 스타일

최준수 씨는 현재 대학생이고 쉴 땐 집에서 많은 시간을 보내며, 특별히 온라인 플랫폼 사용 경험이 풍부하다. 한 번 빠지면 깊이 파고드는 성격이라, 최근엔 리디 홈페이지를 우연히 보고 그 세련된 인터페이스와 편리함에 흠뻑 매료됐다. 평소 심플하고 고급스러운 디자인에 관심이 많고, 취미로 웹사이트 리뷰와 UI/UX 디자인에 대한 자기만의 평론을 남기곤 한다

## 목표

세련된 UI 경험을 찾고, 실제 일상에 잘 스며드는 플랫폼을 즐기고 싶다.  
 집에서의 여가를 가장 중요하게 생각하며, 책·영상·커뮤니티 등 다양한 온라인 콘텐츠를 탐색한다.  
 새로운 서비스의 기능과 디자인을 적극적으로 경험하고, 주변에 추천하며 의견을 나누고 싶다.  
 삶에 "편리함"과 "감각적인 변화"를 더하고, 언제 어디서든 최고의 경험을 누리하고자 한다.

## 동기부여

자극: 새로운 것에 대한 탐구욕과 호기심  
 가격: 경제적 여유가 있어, 가격보다는 '경험 가치'를 우선시함  
 성취: 세련된 취향과 남다른 경험을 쌓으며 자기 만족을 추구  
 성장: 지적 호기심을 채우고, 다양한 플랫폼과 기술을 배우고자 함

홈페이지 구현

01

02

03

04

## HTML 구현

```
<!DOCTYPE html>
<html lang="en">
<head>...
</head>
<body>
  <header>...
  </header>
  <hr>
  <section>...
  </section>

  <!--메인 section 공간 -->
  <main>
    <section>...
    </section>
    <section class="section-icon">...
    </section>
    <section class="label3 basic-box">...
    </section>
    <section class="lavel6 basic-box">...
    </section>
    <section class="lavel5 basic-box">...
    </section>
    <section class="event2 basic-box">...
    </section>
    <section class="label4 basic-box">...
    </section>
    <section class="event1">...
    </section>
  </main>
  <hr>
  <footer class="footer-main">...
  </footer>

  <script src="./js/rldi.js"></script>
</body>
</html>
```

각각의 구성들을 섹션으로 묶어서  
가독성을 높임

```
<section class="lavel5 basic-box">
  <div class="basic-title1">
    <div class="basic-title2">
      <h2 class="basic-h2">오늘, 리디의 발견</h2>
    </div>
  </div>
  <div class="basic-body">
    <div class="basic-body-div">
      <div id="scroll_start5" class="scrollldiv"></div>
      <div class="basic-body-div-d1 slider-container5">
        <ul class="basic-body-ul slider-wrapper5">
          <li class="s5-li slider-block5" id="li_start5"></li>
          <li class="s5-li slider-block5"></li>
          <li class="s5-li slider-block5"></li>
          <li class="s5-li slider-block5"></li>
          <li class="s5-li slider-block5"></li>
          <li class="s5-li slider-block5"></li>
          <li class="s5-li slider-block5"></li>
          <li class="s5-li slider-block5"></li>
          <li class="s5-li slider-block5"></li>
          <li class="s5-li slider-block5" id="li_end5"></li>
        </ul>
      </div>
      <div class="button-set2">
        <button id="prevBtn5" class="button-left"></button>
        <button id="nextBtn5"></button>
      </div>
      <div id="scroll_end5" class="scrollldiv"></div>
    </div>
  </div>
</section>
```

섹션안엔 리스트만 작성하여 애니메  
이션 구현에 도움되기 작성



## 01

## 03

[illegible][illegible]

## 초기 세팅 설정해주고 각 섹션마다 별개로 작동해야하는 CSS 묶어서 정리해주기

특정 기능에서 보여지는 부분이 적절하게 잘 나타날수 있도록 별개로 묶어서 정리해주기

01

02

03

04

# JAVA Script 구현

```

// =====
// 기능 : label5 속성 추가(배우자)
// 설명 : 자식태그가 누르면 되면서 자결해독이있어야 함
const items = document.querySelectorAll('.s5-11');

for(let i = 0; i < 12; i++){
  items[i].innerHTML = `<div class="s5-11-div"><div class="s5-11-div-div"><a href="#" class="img1"></a></div></div>
  <div class="s5-11-div"><div class="title-free"></div><div class="member-free"><p class="p_member"></p><p class="sbd-member-star"></p></div></div>
  <div class="star-free"></div></div></div>`;
}

// =====
// 기능 : label5 속성 추가(배우자)
// 설명 : 자식태그가 누르면 되면서 자결해독이있어야 함
for(let i=0; i<12; i++){
  let a=document.querySelectorAll('.s6-11');
  let b=document.createElement('div');b.classList.add('s5-11-div');
  let c=document.createElement('div');c.classList.add('s5-11-div-div');
  let d=document.createElement('div');d.classList.add('s5-11-div-div');
  let a3=document.createElement("a");a3.href="#";a3.classList.add("img1");

  let div13=document.createElement("div");div13.classList.add("title-free");
  let div12=document.createElement("div");div12.classList.add("member-free");
  let div13=document.createElement("div");div13.classList.add("star-free");

  let p21=document.createElement("p");p21.classList.add("p_member");
  let p22=document.createElement("p");p22.classList.add("sbd-member-star");
  p22.textContent=" ";
  let p23=document.createElement("p");p23.classList.add("p_free");
  p23.textContent=" ";
  let p33=document.createElement("p");p33.classList.add("p_star");

  b.appendChild(d);d.appendChild(a3);
  c.appendChild(div13);
  c.appendChild(div12);
  div12.appendChild(p21);
  div12.appendChild(p22);
  div12.appendChild(p23);
  c.appendChild(div13);
  div13.appendChild(p33);
  a[i].appendChild(b);
  a[i].appendChild(c);
}

// =====
// 기능 : label3 속성 추가(배우자)
// 설명 : 뒤 밭식 보다 좀더 간결하고 깔끔함
// 처음에 구조를 잘 짜주면 반복되는 아이를 막아서 처리
// innerHTML 잘 활용에 주의
const label3 = document.querySelectorAll('.label3_11');
label3.forEach(insert => {
  insert.innerHTML = `<div class="basic-bighox"><!-- 이미지 넣는 구간 --><div class="basic-lngbox"></div><!-- 인덱스넣는 구간 -->
  <div class="basic-indexbox"><div class="basic-index"></div></div><!-- 후속 넣는 구간 --><div class="basic-bighox-d4">
  <div class="basic-form"><div class="title-free"></div><div class="member-free"><p class="p_member"></p><p class="sbd-member-star">
  </p><p class="p_free"></p></div><div class="star-free"><p class="p_star"></p></div></div></div>`;
});

const label4 = document.querySelectorAll('.label4_11');
label4.forEach(insert => {
  insert.innerHTML = `<div class="basic-bighox"><!-- 이미지 넣는 구간 --><div class="basic-lngbox"></div><!-- 인덱스넣는 구간 -->
  <div class="basic-indexbox"><div class="basic-index"></div></div><!-- 후속 넣는 구간 --><div class="basic-bighox-d4">
  <div class="basic-form"><div class="title-free"></div><div class="member-free"><p class="p_member"></p><p class="sbd-member-star">
  </p><p class="p_free"></p></div><div class="star-free"><p class="p_star"></p></div></div></div>`;
});

```

- 반복되는 리스트를 HTML에 나타내기보다 추가되는 리스트에 맞춰서 알아서 코드가 삽입되도록 설정
- 두가지 방법으로 구현해보기

```

// =====
// 기능 : checkCollision(교점박스, 이동박스, 왼쪽바튼, 오른쪽바튼, 오른쪽 버튼)
// 기능 : 디자인 이동할수 없는 버튼은 안보이도록 하든 처리
function checkCollision(num) {
  const list1 = ['scroll_start', 'scroll_end', 'l_end', 'prevBtn', 'nextBtn'];
  const elements = list1.map(id => document.getElementById(id + num));
  const [box1, box2, box3, box4, btn1, btn2] = elements;

  const [rect1, rect2, rect3, rect4] = elements.slice(0, 4).map(e1 => e1.getBoundingClientRect());

  const rect44 = rect4.x + 1300;

  btn1.style.visibility = rect1.x > rect2.x ? 'visible' : 'hidden';
  btn2.style.visibility = rect3.x <= rect44 ? 'visible' : 'hidden';

  // =====
  // 함수명 : 버튼 숨김시키기
  // 기능 : 이미 왼쪽 오른쪽 다른 버튼 이미 어떻게 만들어졌는데 합칠수 있는 방법 생각해보기
  // =====
  const buttons = [
    { next: nextBtn3, prev: prevBtn3, num: 3 },
    { next: nextBtn4, prev: prevBtn4, num: 4 },
    { next: nextBtn5, prev: prevBtn5, num: 5 },
    { next: nextBtn6, prev: prevBtn6, num: 6 },
  ];

  buttons.forEach(( { next, prev, num } ) => {
    next.addEventListener('click', () => {
      setTimeout(() => checkCollision(num), 520);
    });
    prev.addEventListener('click', () => {
      setTimeout(() => checkCollision(num), 520);
    });
  });

  // =====
  // 함수명 : touchbar(교점박스,이동박스, 실제박스, 왼쪽바튼, 오른쪽바튼, 이동시합박스계수, 이벤트가, 한동안에 보이는 박스수 )
  // 기능 : 정한 횟수만큼의 박스가 버튼 누를때 이동하는 이벤트
  // =====
  function touchbar(a, b, c, d, e, f, g, h) {
    let wrapper = document.querySelector(a);
    let blocks = document.querySelectorAll(b);
    let container = document.querySelector(c);

    let currentIndex = 0;
    let blockWidth = blocks[0].offsetWidth + g; // 블록 너비 + 간격
    let visibleCount = h; // 한 화면에 보이는 블록 수
    let maxIndex = blocks.length - visibleCount; // 슬라이드 가능한 최대 인덱스

    wrapper.style.transition = 'transform 0.5s ease';

    function slideTo(index) {
      if (index < 0) index = 0;
      if (index > maxIndex) index = maxIndex;
      currentIndex = index;
      wrapper.style.transform = `translateX(${(blockWidth * currentIndex)px})`;
    }

    document.getElementById(d).addEventListener('click', () => slideTo(currentIndex - f));
    document.getElementById(e).addEventListener('click', () => slideTo(currentIndex + f));
  }

```

- 슬라이더 부분에서 버튼 속성을 지정 넘길 화면이 없을 땐 숨김 처리
- 몇 개의 리스트를 넘겨줄지 설정
- 비슷한 모든 버튼에 각각의 속성을 달리할 수 있도록 함수로 지정

01

02

03

04

# JAVA Script 구현

```

//=====
// 함수명 : fetchBooks(원하는 주제)
// 기능 : 특정미리미리를 검색하면 url 만들어주기
// 추가기능 : 검색결과에 있는 data 삭제하기
//=====
async function fetchBooks(query) {
  const params = new URLSearchParams({target: "title", query, size: 40});
  const url = "https://dapi.kakao.com/v3/search/book?${params}";

  const response = await fetch(url, {method: 'GET', headers: {
    Authorization: "KakaoAK 7b2300fc6315bb6503d1a3c7b49b161"
  }});
  if (!response.ok) { throw new Error("HTTP 오류: ${response.status}"); }
  return response.json();
}

//=====
// 함수명 : bookData(컨텐츠, 이미지를 박스, 제목명을 박스, 멤버명을 박스)
// 기능 : 필요한 부분은 태그를 직접 생성해서 그 태그안에 API자료 넣기
//=====
async function bookData(a,b,c,d,e) {
  const data = await fetchBooks(a);
  const label5_img = document.querySelectorAll(b);
  const label5_title = document.querySelectorAll(c);
  const label5_member = document.querySelectorAll(d);
  const label5_index = document.querySelectorAll(e);
  for (let i = 0; i < label5_title.length; i++) {
    //이미지추가
    console.log(i);
    const img = document.createElement("img");
    img.src = data.documents[i].thumbnail;
    label5_img[i].appendChild(img);
    //제목추가
    const title = document.createElement("a");
    title.classList.add("sbd-title");
    title.textContent = data.documents[i].title;
    label5_title[i].appendChild(title);
    //작성자추가
    const member = document.createElement("a");
    member.classList.add("sbd-member");
    member.textContent = data.documents[i].authors;
    label5_member[i].appendChild(member);
    //인덱스 추가
    label5_index[i].textContent = i+1
  }
}

```

Open API 를 활용하여 책데이터 가져오고 활용하기

```

//=====
// 함수 : tabtab(class name)
// 요약 : 클릭을 하면 다른 아이들의 active는 다 사라지고 클릭한 친구만 주기
//=====
function tabtab(class_name){
  const act = document.querySelectorAll(class_name);
  act.forEach(div => {
    div.addEventListener('click', () => {
      act.forEach(d => d.classList.remove('active'));
      div.classList.add('active');
    });
  });
}

tabtab('.56-days')
tabtab('.51-menu-ul li')

//=====
// 함수 : toggleImageClasses()
// 요약 : 끝수지만 총 6개의 사진의 클래스스가 돌아가면서 반복되는 패턴 만들어주기
//=====

const container = document.getElementById('container');
const images = document.getElementsByClassName('img2');

function toggleImageClasses() {
  for (let i = 0; i < images.length; i++) {
    if (images[i].classList.contains('slide-in')) {
      images[i].classList.remove('slide-in');
      images[i].classList.add('slide-in1');
    } else if (images[i].classList.contains('slide-in1')) {
      images[i].classList.remove('slide-in1');
      images[i].classList.add('slide-in2');
    } else if (images[i].classList.contains('slide-in2')) {
      images[i].classList.remove('slide-in2');
      images[i].classList.add('expanded');
    } else if (images[i].classList.contains('expanded')) {
      images[i].classList.remove('expanded');
      images[i].classList.add('slide-out');
    } else if (images[i].classList.contains('slide-out')) {
      images[i].classList.remove('slide-out');
      images[i].classList.add('slide-out-move');
    } else if (images[i].classList.contains('slide-out-move')) {
      images[i].classList.remove('slide-out-move');
      images[i].classList.add('slide-in');
    }
  }
}

container.addEventListener('click', toggleImageClasses); // 클릭 시 호출
setInterval(toggleImageClasses, 3000); // 3초 간격으로 자동 호출

```

탭속성을 구현해서 특정이벤트 발동 시 class 추가 될수 있도록 하기

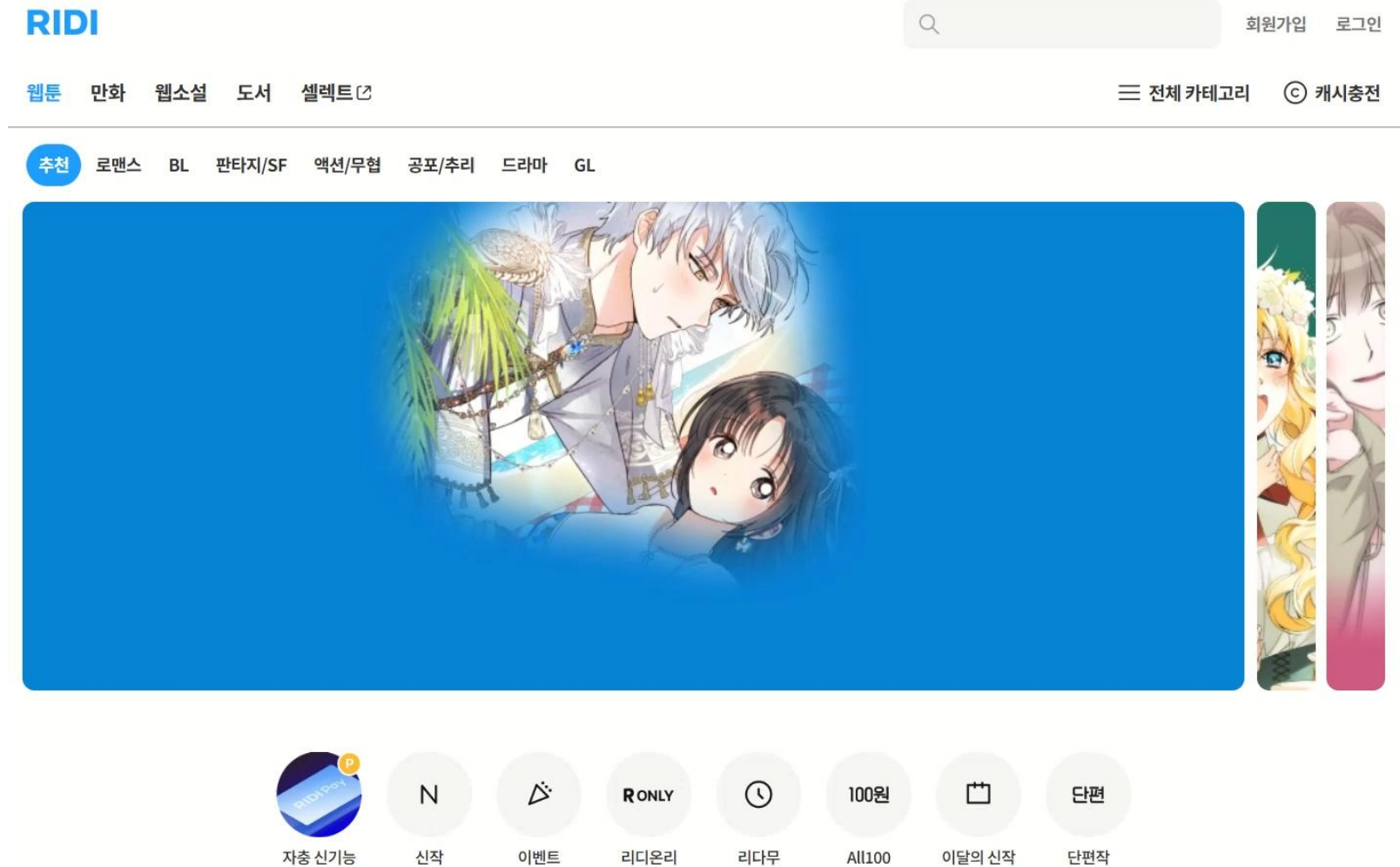
01

02

03

04

## 홈페이지 구현모습1-슬라이더



01










02

03

04

# 홈페이지 구현모습2-탭기능

## 웹툰 실시간 랭킹

	1	사랑해 사랑해 사랑해(출간... 버나뎃 로제티 슈스탁 *		4	사랑을 무게로 안 느끼게 박한서 *		7	오늘 밤, 세계에서 이 사랑... 이치조 미사키 *
		★ 4.8 (7791)			★ 4.5 (744)			★ 5 (6090)
	2	여수의 사랑 한강 *		5	돈, 뜨겁게 사랑하고 차갑... 앙드레 코스톨라니 *		8	사랑인 줄 알았는데 부정맥 전국유로실버타운협회 *
		★ 4.2 (6494)			★ 4.5 (8345)			★ 4.1 (4694)
	3	푸바오, 언제나 사랑해 *		6	사랑의 기술 에리히 프롬 *		9	나는 오래된 거리처럼 너를... 진은영 *
		★ 4.5 (6093)			★ 4.5 (7304)			★ 4.5 (229)

>

## 요일별 웹툰

[더보기](#)

월	화	수	목	금	토	일
						
월 300만 원 버는 주식 투... 진서빈 * 외 1명 ★ 4.2 (6198)	나는 미국 월배당 ETF로 4... 최영민 * 외 1명 ★ 4.8 (1662)	주식투자 최적의 타이밍을... 스탠 와인스타인 * 외 1명 ★ 4.5 (685)	투자에 대한 생각 하워드 막스 * 외 1명 ★ 4.2 (8207)	제시 리버모어의 주식투자... 제시 리버모어 * 외 1명 ★ 4.8 (7533)	월 50만 원으로 8억 만드... 평은,김지형 * 외 1명 ★ 4.6 (7649)	

01

## 번외 -상단바 반응형

02

03

04

RIDI



회원가입

로그인

웹툰

만화

웹소설

도서

선택트

≡ 전체 카테고리

© 캐시충전

추천

로맨스

BL

판타지/SF

액션/무협

공포/추리

드라마

GL

개선사항



01

## 개선사항

02

03

04

1.정돈되지 않은 HTML

2.통일되지 않은 클래스로 인한 중복 CSS

3.직접 JAVA Script를 만들다보니 상용화 되어있는 친구들 적용 및 수정 이해 x

4.좀더 단축 시킬수 있는 방법 모색

5.반응형까지 작업하면 더 완벽



감사합니다