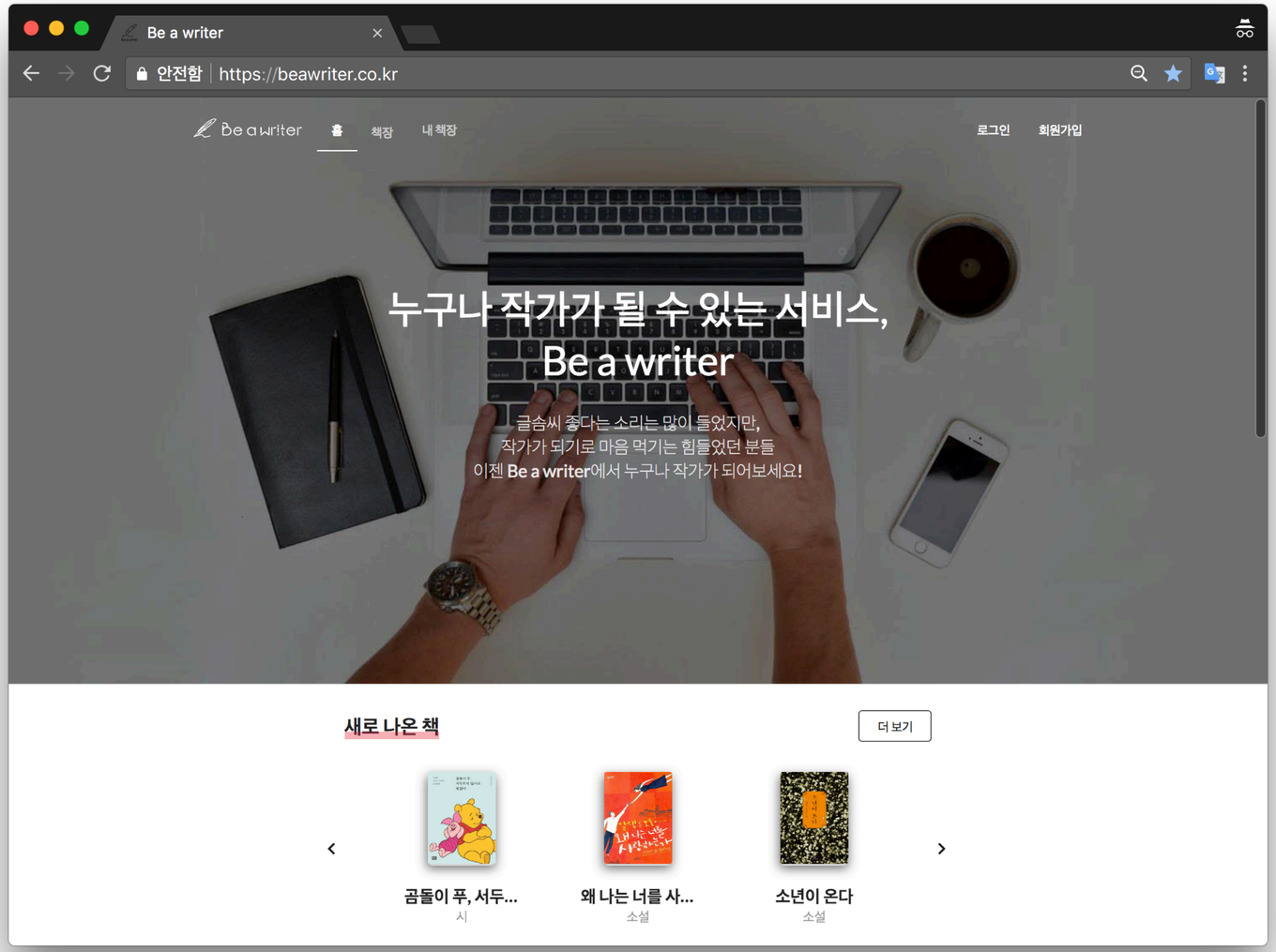


백엔드 개발 환경

- Python 3.6.4
- Django 2.0.6

프론트엔드 개발 환경

- Django template
- jQuery 3.3.1

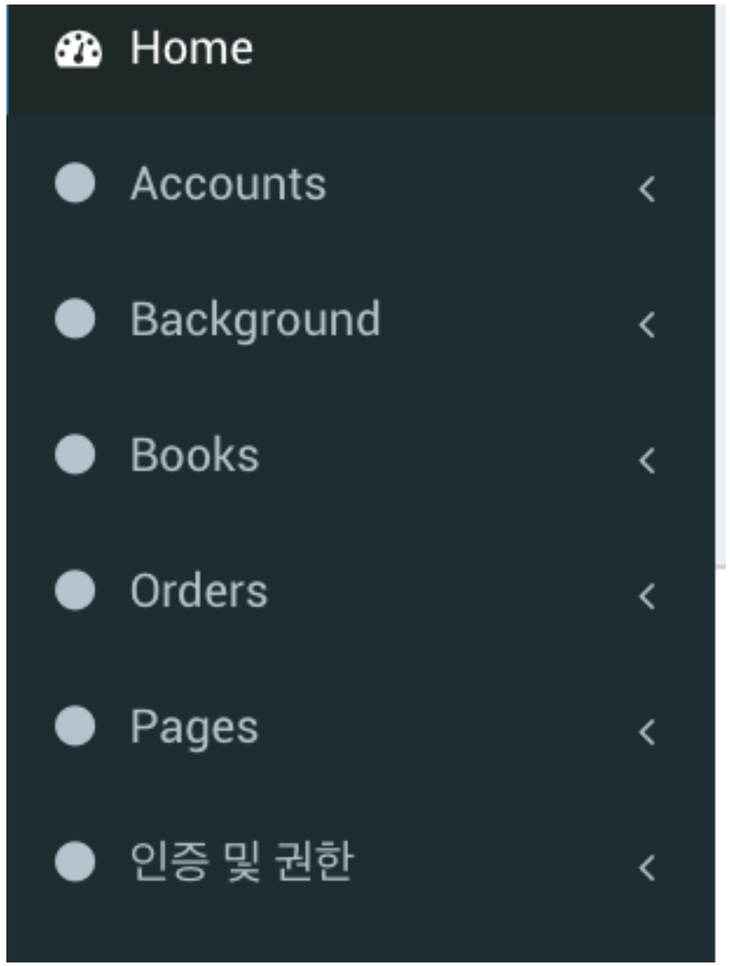
기타 라이브러리

- Semantic UI 2.3.1
- rater.js
- slick.js

배포 환경

- Docker
- AWS ElasticBeanstalk

rainsound128@gmail.com



Accounts
- 계정 관련 모델 및 뷰

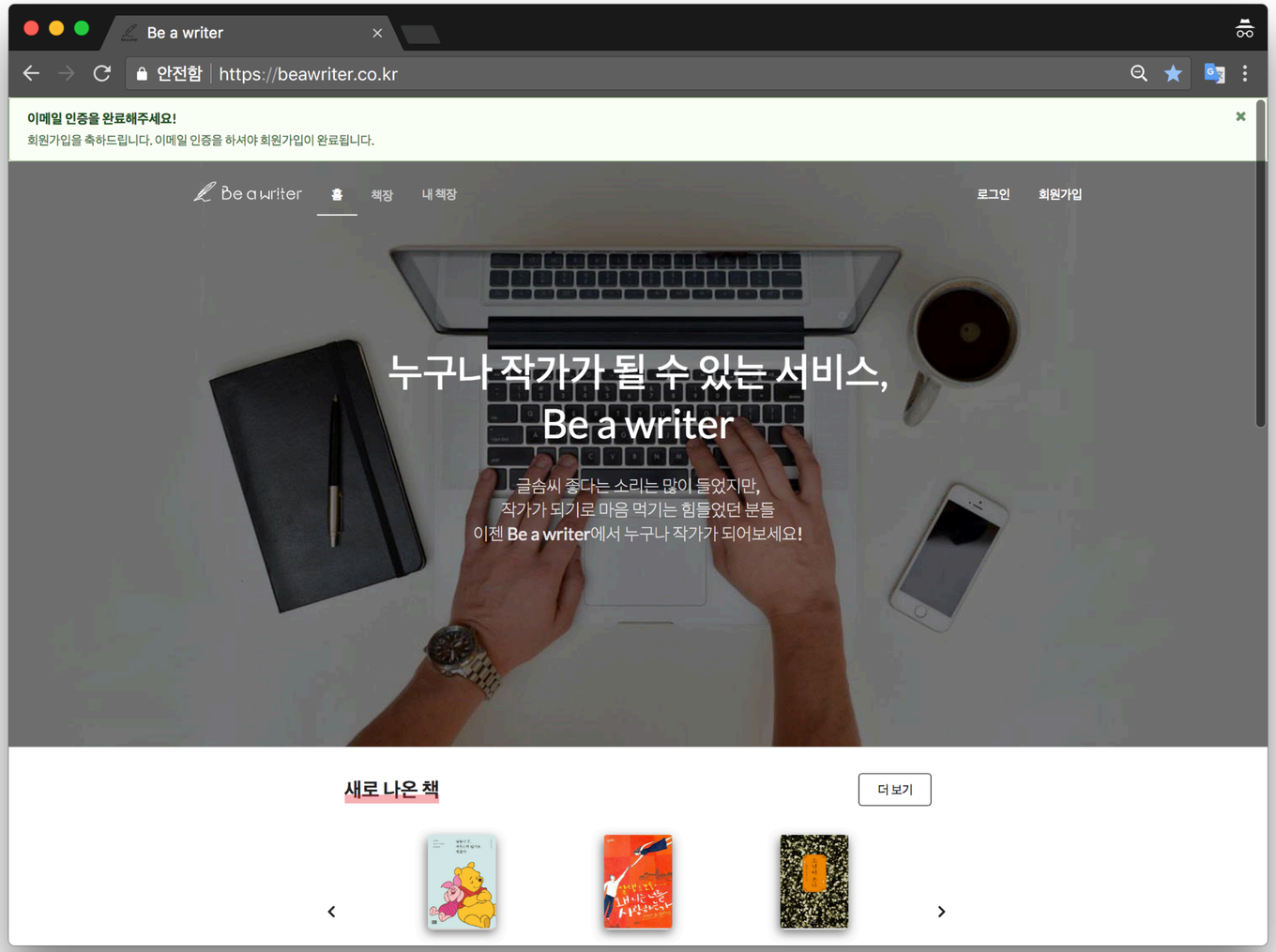
Background
- 메인 페이지 배경 이미지 관리 모델

Books
- 책 생성/수정/상세/삭제/목록/출판 관련 모델 및 뷰
- 책 리뷰 생성/삭제 관련 모델 및 뷰
- 책 좋아요 관련 모델 및 뷰

Orders
- 코인 주문 관리 모델 및 뷰
- 책 주문 관리 모델 및 뷰

Pages
- 페이지 생성/수정/상세/목록 관련 모델 및 뷰
- 페이지 댓글 관련 모델 및 뷰

accounts - 이메일 인증 회원 가입 기능(1/2)



Be a writer

회원 가입
이메일 인증 메일입니다.

안녕하세요, **rainsound** 님.
Be a writer 회원가입을 환영합니다!

아래 이메일 인증 버튼으로 회원가입을 완료할 수 있습니다.

인증 완료하기

RAYI 주식회사

사업자등록번호 : 000-00-0000　개인정보관리책임자 : 강성우
이메일 : rainsound128@gmail.com

기타 문의 사항은 위의 메일로 보내주시기 바랍니다.

이메일 인증 회원 가입 기능

accounts - 이메일 인증 회원 가입 기능(2/2)

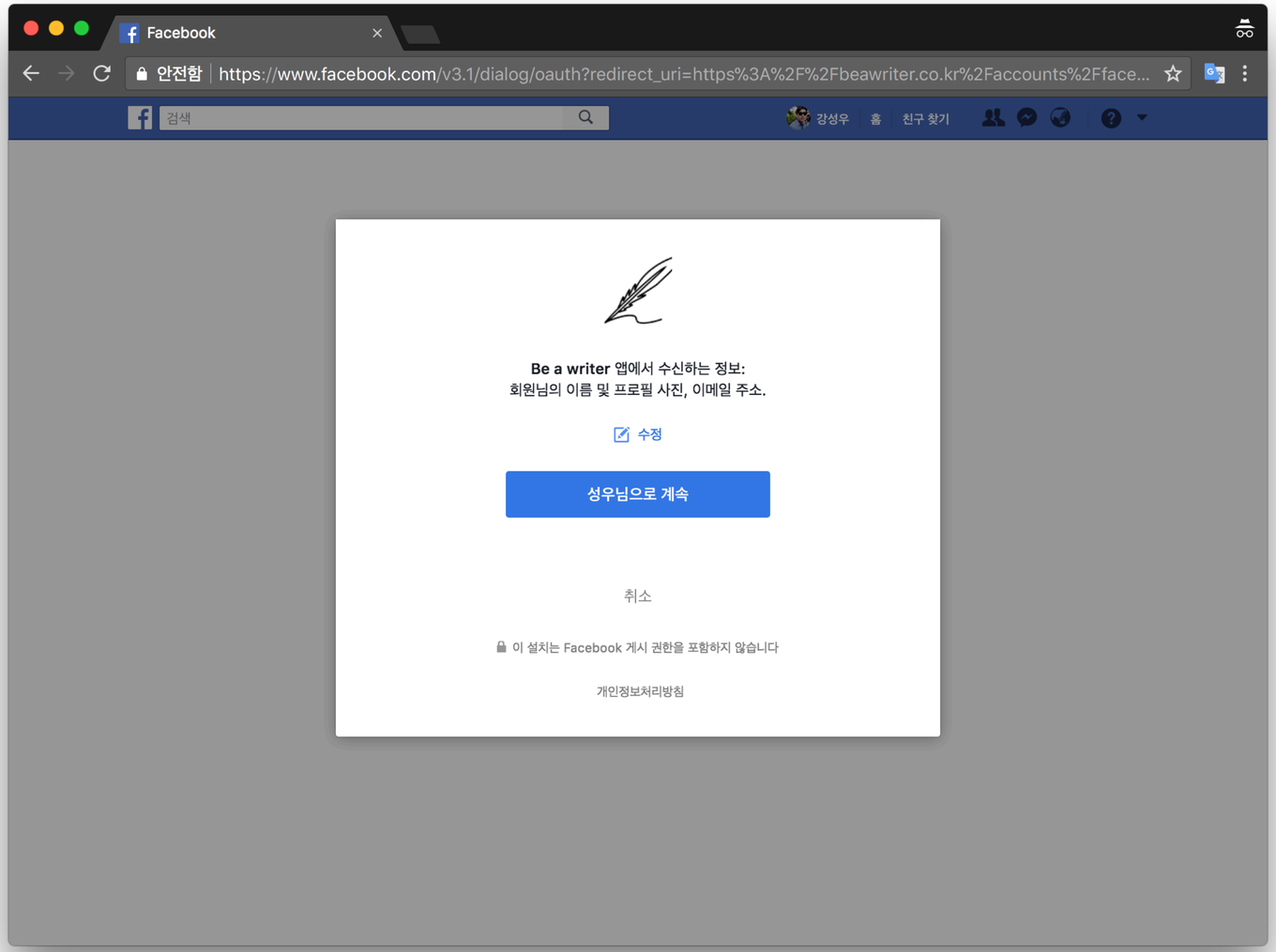
```python
def signup_view(request):
    ...

    if request.method == 'POST':
        form = SignUpForm(request.POST, request.FILES)
        if form.is_valid():
            new_user = form.save(commit=False)
            new_user.is_active = False
            new_user.save()
            current_site = get_current_site(request)
            mail_subject = '[Be a writer] 회원가입을 위한 이메일 인증입니다.'
            body = {
                'user': new_user,
                'domain': current_site.domain,
                'uid': urlsafe_base64_encode(force_bytes(new_user.pk)).decode(),
                'token': account_activation_token.make_token(new_user),
            }
            message = render_to_string('accounts/active-email-form.html', body)
            to_email = form.cleaned_data.get('email')
            email = EmailMessage(mail_subject, message, to=[to_email, ])
            email.content_subtype = 'html'
            email.send()

            return redirect('index')

    ...
```

Colored by Color Scripter

```python
class TokenGenerator(PasswordResetTokenGenerator):
    def _make_hash_value(self, user, timestamp):
        return six.text_type(user.pk) + six.text_type(timestamp) + six
.text_type(user.is_active)


account_activation_token = TokenGenerator()
```

Colored by Color Scripter

이메일 인증 회원 가입 기능 코드 일부



페이스북 회원가입/로그인 기능

rainsound128@gmail.com

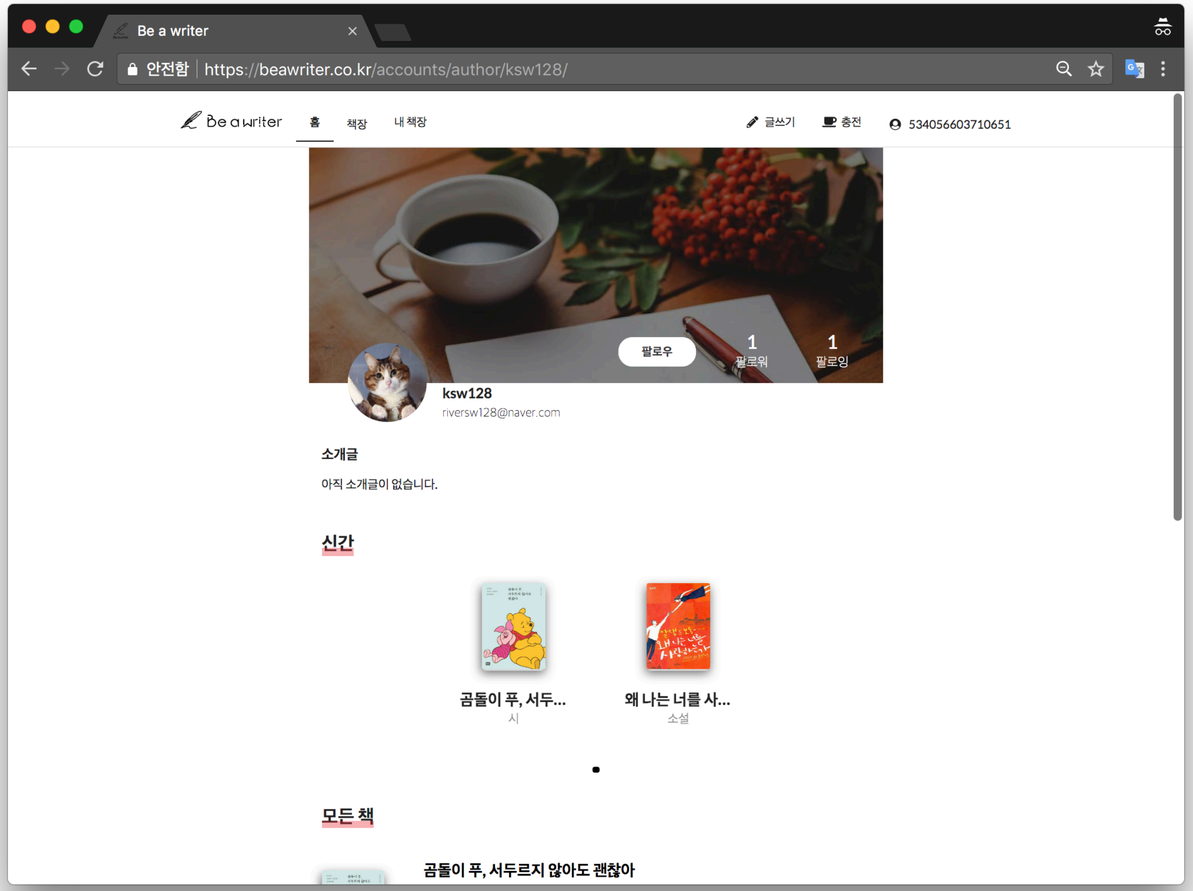
accounts - 페이스북 회원가입/로그인 기능(2/2)

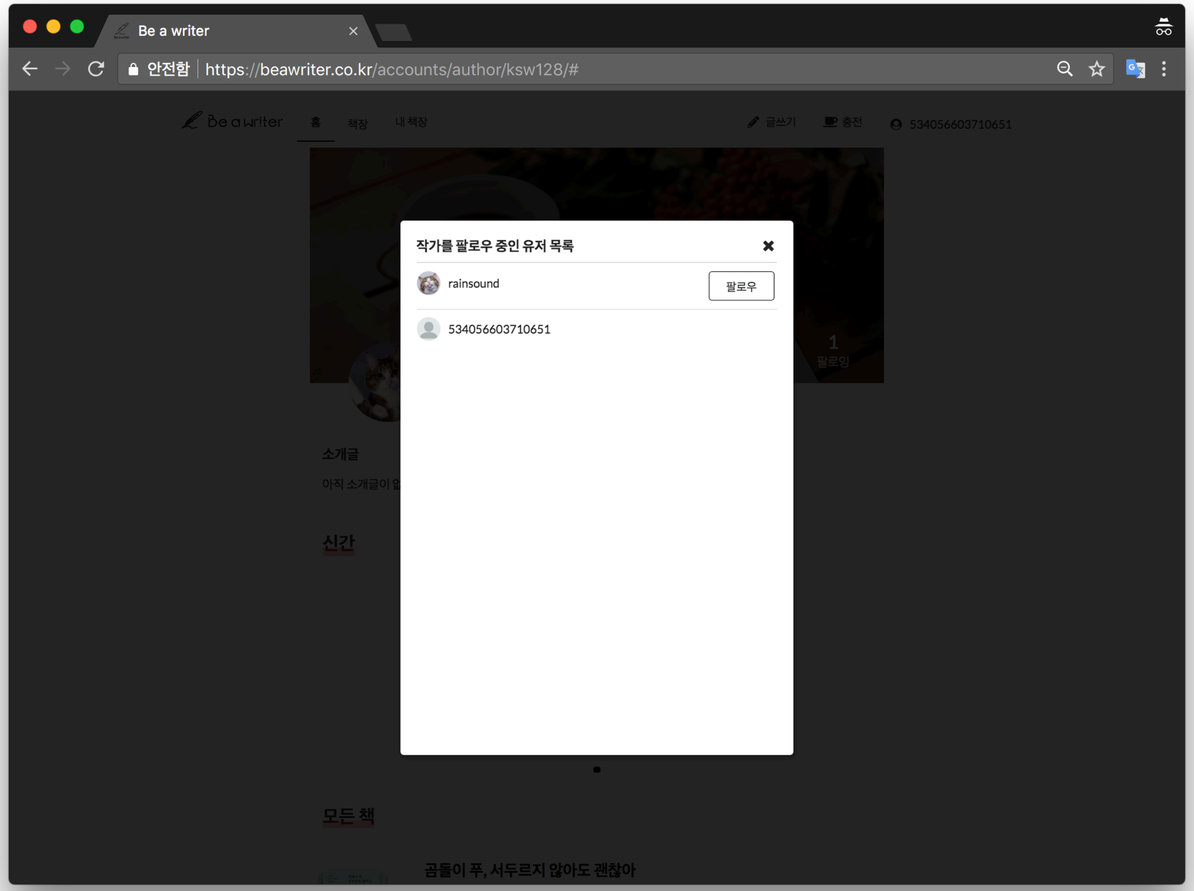
```python
class FacebookBackend:
    ...

    def authenticate(self, code):
        def get_access_token(auth_code):
            redirect_uri = 'https://beawriter.co.kr/accounts/facebook-login/'
            params_access_token = {
                'client_id': self.CLIENT_ID,
                'redirect_uri': redirect_uri,
                'client_secret': self.CLIENT_SECRET,
                'code': auth_code,
            }
            response = requests.get(self.URL_ACCESS_TOKEN, params_access_token)
            response_dict = response.json()
            return response_dict['access_token']

        def get_user_info(user_access_token):
            params = {
                'access_token': user_access_token,
                'fields': ','.join([
                    'id',
                    'email',
                ])
            }
            response = requests.get(self.URL_ME, params)
            response_dict = response.json()
            return response_dict

        access_token = get_access_token(code)
        user_info = get_user_info(access_token)

        facebook_id = user_info['id']
        facebook_email = user_info['email']

        try:
            user = User.objects.get(email=facebook_email)
        except User.DoesNotExist:
            user = User.objects.create_user(
                username=facebook_id,
                email=facebook_email,
            )
        return user

    ...
```

```python
def facebook_login(request):
    code = request.GET.get('code')
    user = authenticate(request, code=code)
    login(request, user)
    return redirect('index')
```

페이스북 회원가입/로그인 기능 코드 일부

accounts - 작가 팔로우 기능(1/2)





작가 팔로우 기능

rainsound128@gmail.com

accounts - 작가 팔로우 기능(2/2)

```python
def follow(request):
    from_user = request.user
    pk = request.POST.get('pk')
    to_user = get_object_or_404(User, pk=pk)
    relation, created = Relation.objects.get_or_create(from_user=from_user, to_us
er=to_user)

    if created:
        status = 1
    else:
        relation.delete()
        status = 0

    context = {
        'status': status,
    }
    return HttpResponse(json.dumps(context), content_type="application/json")
```

Colored by Color Scripter CS

```javascript
$(document).on('click', '#followAuthor.button', function(){
    var pk = $(this).attr('name');
    $.ajax({
      type: "POST",
      url: "{% url 'accounts:follow' %}",
      data: {
        'pk': pk,
        'csrfmiddlewaretoken': '{{ csrf_token }}',
      },
      dataType: "json",

      success: function(response){
        $(".follower-btn").load(window.location + " .follower-btn");
        if(response.status){
          $("#followAuthor").text("팔로잉");
          $("#followAuthor").toggleClass("white");
        }else {
          $("#followAuthor").text("팔로우");
          $("#followAuthor").toggleClass("white");
        }
      },
      error: function(request, status, error){
        ...
      }
    })
  });
```

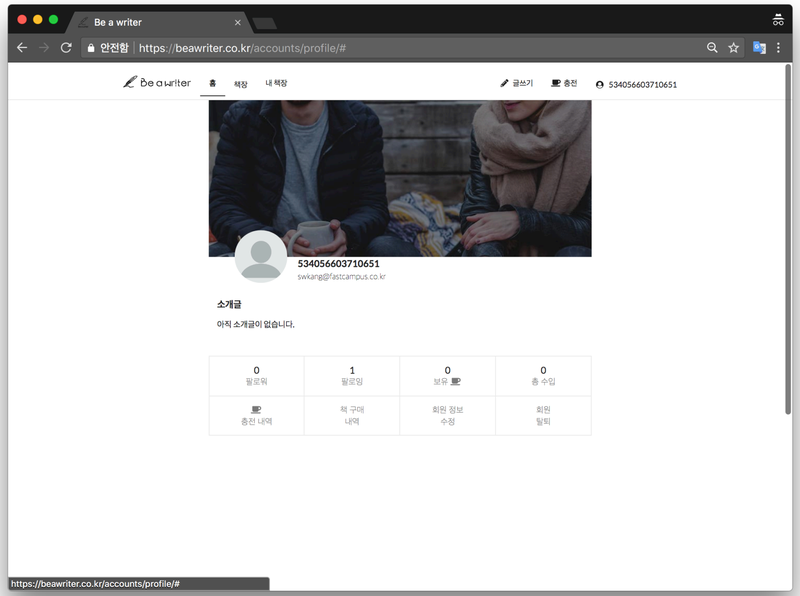
Colored by Color Scripter CS

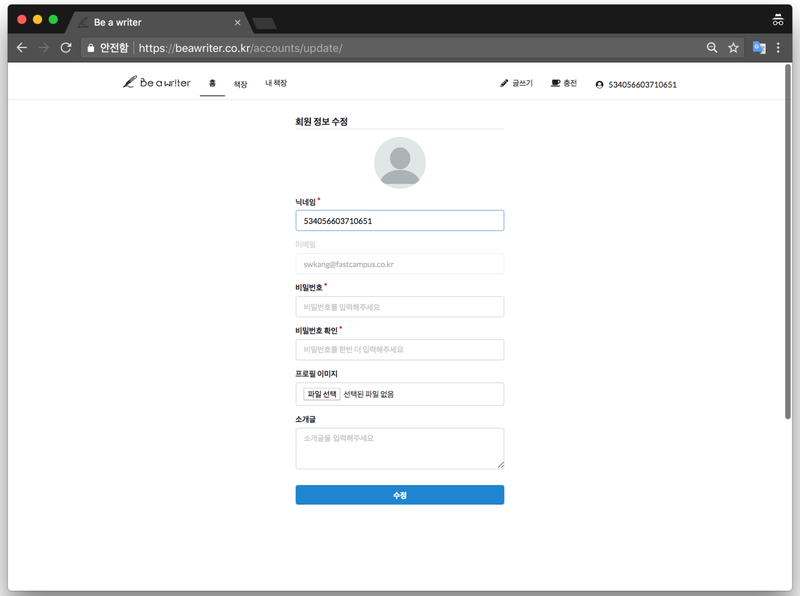
작가 팔로우 기능 코드 일부

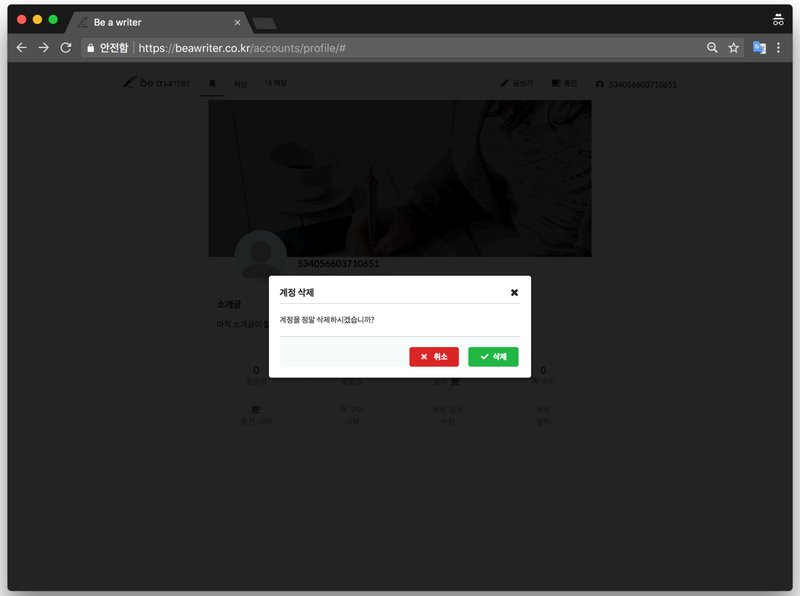
rainsound128@gmail.com

Be a writer

accounts - 프로필 상세/수정/회원 탈퇴 기능







프로필 상세보기 기능

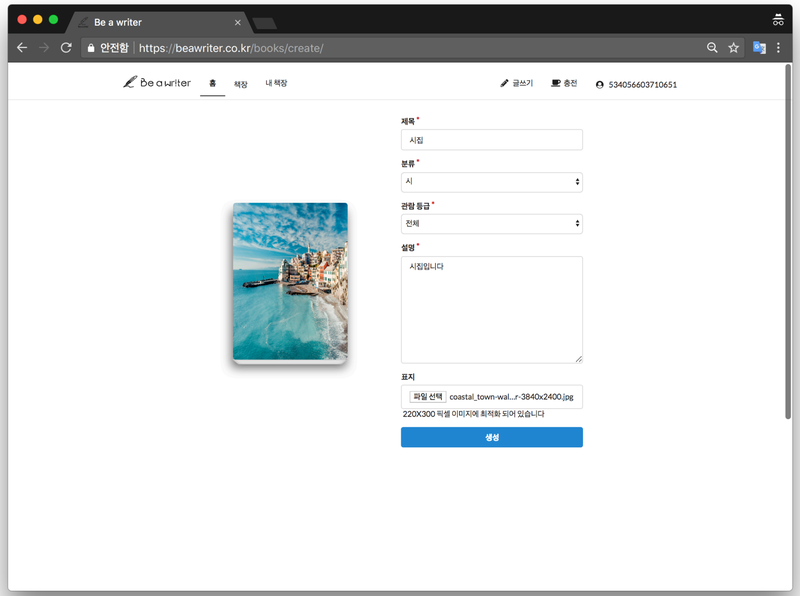
프로필 수정 기능

회원 탈퇴 기능

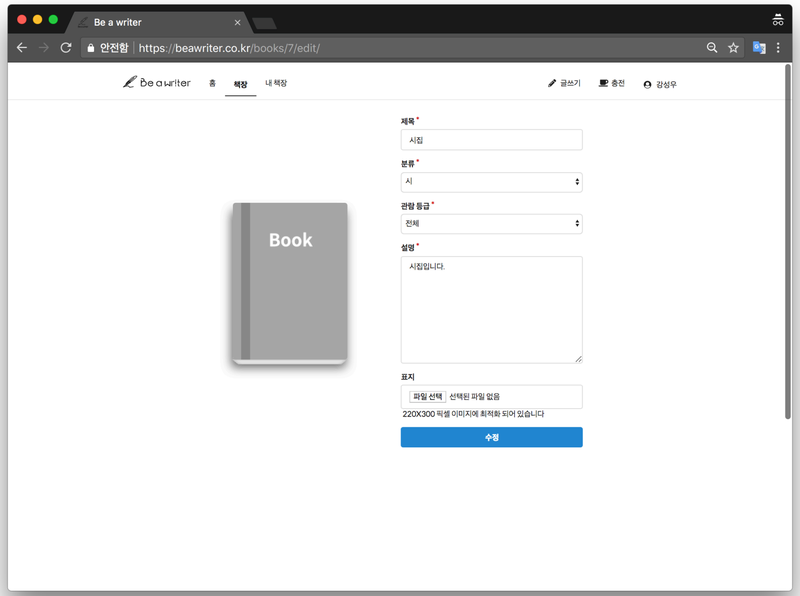
rainsound128@gmail.com

Be a writer

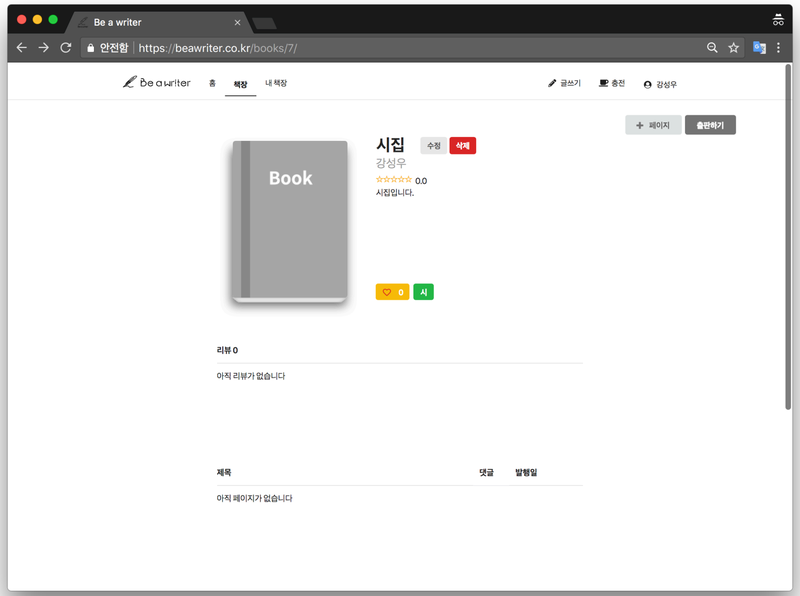
books - 책 생성/수정/삭제 기능



책 생성 기능



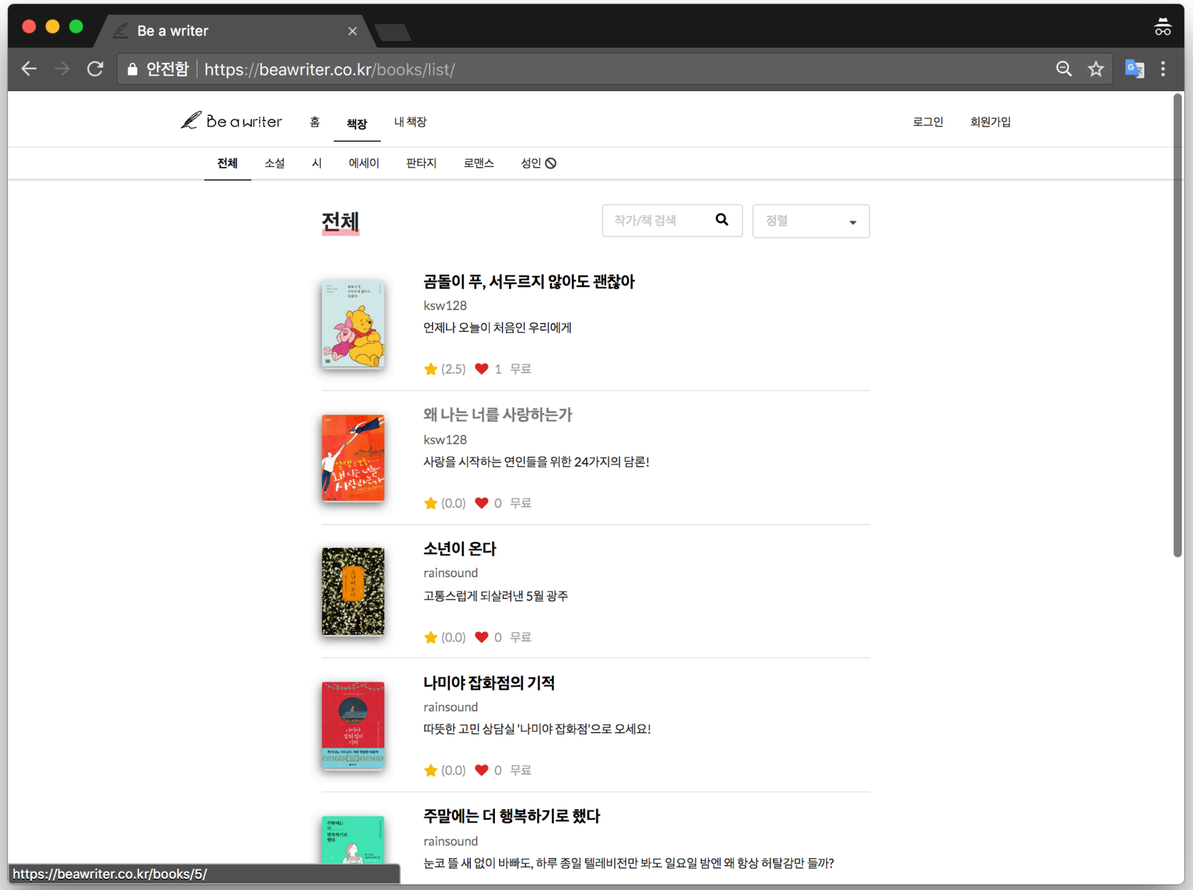
책 수정 기능

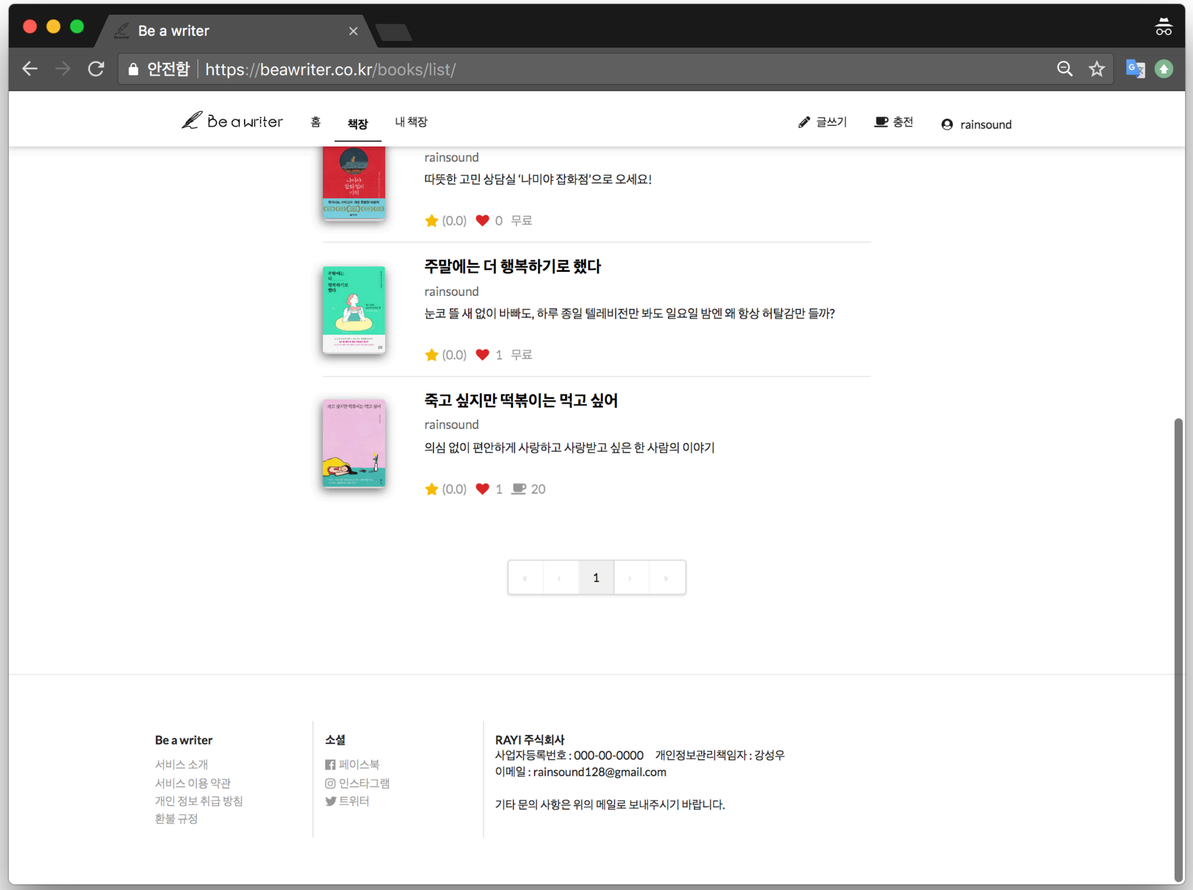


책 삭제 기능

rainsound128@gmail.com

books - 책 리스트 검색/필터/페이지네이션 기능(1/2)





책 리스트 검색/필터/페이지네이션 기능

rainsound128@gmail.com

books - 책 리스트 검색/필터/페이지네이션 기능(2/2)

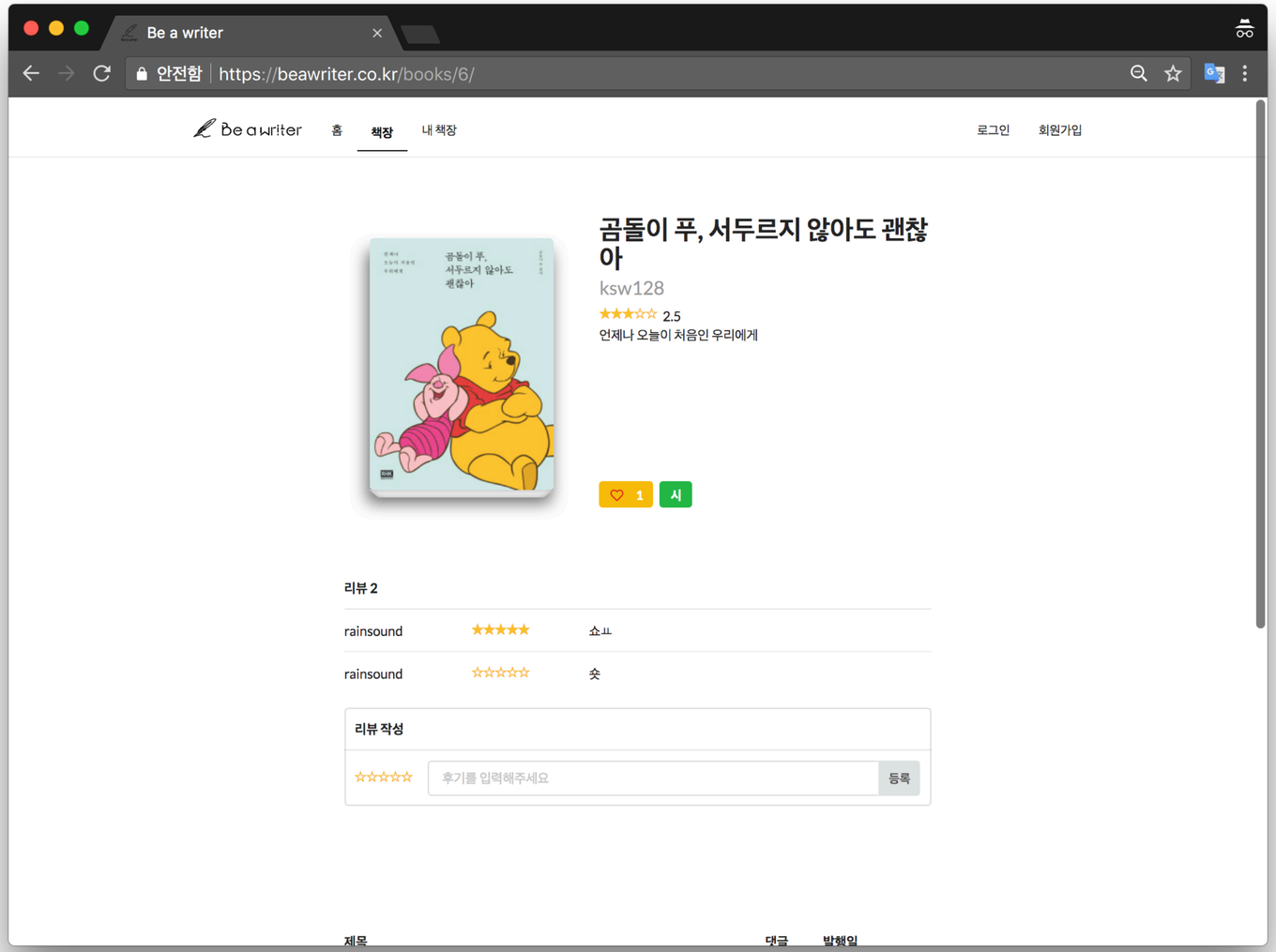
```python
def all_book_list(request):
    q = request.GET.get('q', '')
    if q:
        books_list = Book.objects.filter(
            Q(rating='g', is_published=True, author__username__contains=q) |
            Q(rating='g', is_published=True, title__contains=q)
        ).order_by('-created_time')
    else:
        books_list = Book.objects.filter(rating='g', is_published=True).order_by('-created_time')
    book_filter = BookFilter(request.GET, queryset=books_list)
    paginator = Paginator(book_filter.qs, 20)
    page_index = request.GET.get('page')
    books = paginator.get_page(page_index)

    context = {
        'book_filter': book_filter,
        'books': books,
        'q': q,
    }
    return render(request, 'books/book-list.html', context)
```



```python
class BookFilter(django_filters.FilterSet):
    ordering = django_filters.OrderingFilter(
        fields=(
            'published_date',
            'like_users_count',
            'average_star_rating',
            'amount',
        )
    )

    class Meta:
        model = Book
        fields = (
            'category',
        )
```


책 리스트 검색/필터/페이지네이션 기능 코드 일부

rainsound128@gmail.com



책 상세/좋아요/리뷰(별점) 기능

rainsound128@gmail.com

books - 책 상세/좋아요/리뷰(별점) 기능(2/2)

```python
def book_like_toggle(request):
    book_pk = request.POST.get('pk', None)
    book = get_object_or_404(Book, pk=book_pk)
    book_like, book_like_created = book.like_user_info_list.get_or_create(user
=request.user)

    if not book_like_created:
        book_like.delete()
        book.like_users_count = book.like_users.count()
        book.save()
        status = 0
    else:
        book.like_users_count = book.like_users.count()
        book.save()
        status = 1
    context = {
        'status': status,
    }
    return HttpResponse(json.dumps(context), content_type='application/json')
```



```python
def review_create(request, book_pk):
    pk = request.POST.get('pk', None)
    star_rating = request.POST.get('star_rating', None)
    content = request.POST.get('content', None)
    book = get_object_or_404(Book, pk=pk, is_published=True)
    if content:
        BookReview.objects.create(
            user=request.user,
            book=book,
            star_rating=star_rating,
            content=content,
        )

        # 평균 별점을 계산해서 Book의 average_star_rating 필드에 입력
        if book.reviews.aggregate(Avg('star_rating'))['star_rating__avg']:
            book.average_star_rating = round(book.reviews.aggregate(Avg('star_
rating'))['star_rating__avg'], 1)
            book.save()

        context = {
            'message': '리뷰가 등록됐습니다',
        }
    else:
        context = {
            'message': '내용을 모두 작성해주세요',
            'error': 1,
        }
    return HttpResponse(json.dumps(context), content_type='application/json')
```


책 상세/좋아요/리뷰(별점) 기능 코드 일부

rainsound128@gmail.com

```python
@register.simple_tag(takes_context=True)
def url_replace(context, **kwargs):
    query = context['request'].GET.copy()

    for kwarg in kwargs:
        try:
            query.pop(kwarg)
        except KeyError:
            pass

    query.update(kwargs)

    return mark_safe(query.urlencode())
```

카테고리 필터, 검색과 페이지 숫자를 동시에 get 인자로 넘기기 위해 사용

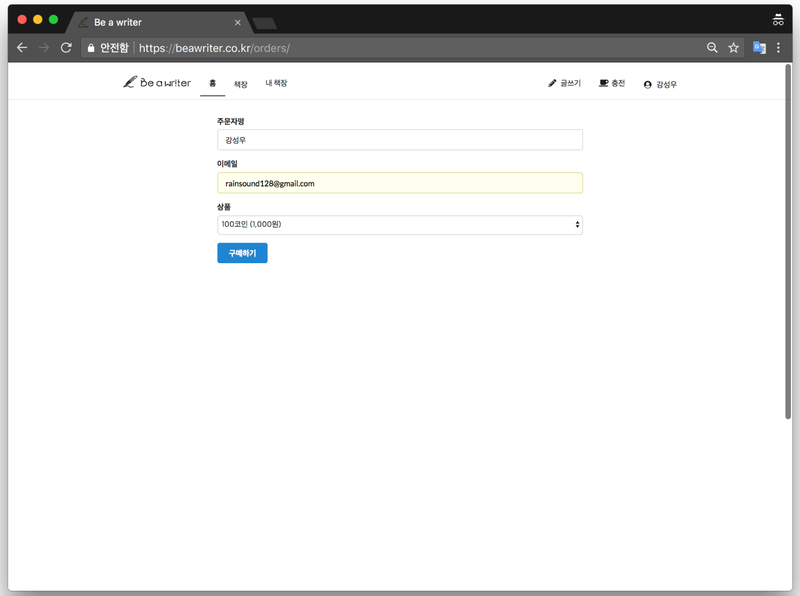
```python
@register.filter
def get_item(dictionary, key):
    return dictionary.get(key)
```

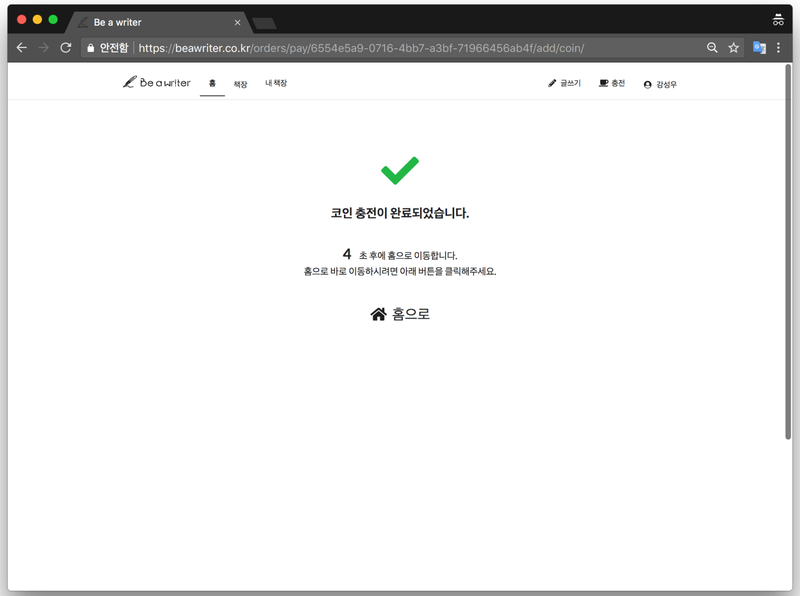
dictionary의 value 값을 key 값으로 얻기 위해 사용

책 커스텀 templatetags

rainsound128@gmail.com

orders - 코인 주문 기능(1/2)





코인 주문 기능

orders - 코인 주문 기능(2/2)

```python
def order_new(request):
    instance = Order(order_email=request.user.email)
    if request.method == 'POST':
        form = OrderForm(request.POST, instance=instance)
        amount = int(form.data['coin'])
        amount_dict = dict(Order.CHOICES_COIN)

        if form.is_valid():
            order = form.save(commit=False)
            order.user = request.user
            order.amount = amount
            order.buyer_name = form.data['order_username']
            order.buyer_email = form.data['order_email']
            order.name = amount_dict[form.data['coin']]
            order.save()
            return redirect('orders:order-pay', str(order.merchant_uid))

    ...
```

```python
def order_pay(request, merchant_uid):
    order = get_object_or_404(Order, user=request.user, merchant_uid=merchant_uid, status='ready')
    if request.method == 'POST':
        form = PayForm(request.POST, instance=order)
        if form.is_valid():
            form.save()
            return redirect('orders:coin-add', merchant_uid)

    ...
```



```python
def mobile_order_pay(request):
    imp_uid = request.GET.get('imp_uid', None)
    merchant_uid = request.GET.get('merchant_uid', None)
    order = get_object_or_404(Order, user=request.user, merchant_uid=
merchant_uid, status='ready')
    if request.method == 'POST':
        form = MobilePayForm(request.POST, instance=order)
        if form.is_valid():
            form.save()
            return redirect('orders:coin-add', merchant_uid)

    ...
```



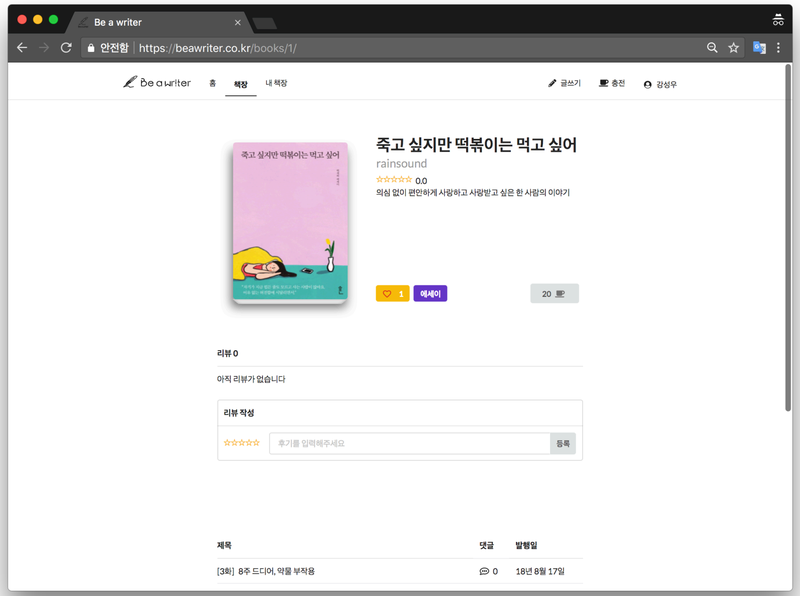
```python
def coin_add(request, merchant_uid):
    order = get_object_or_404(Order, user=request.user, merchant_uid=merchant_uid)
    user = request.user
    if order.is_paid_ok:
        if order.amount == 1000:
            user.coin += 100
        elif order.amount == 5000:
            user.coin += 500
        elif order.amount == 10000:
            user.coin += 1200
        elif order.amount == 50000:
            user.coin += 6500
        else:
            pass
        user.save()
        return render(request, 'orders/order-complete.html')
    ...
```

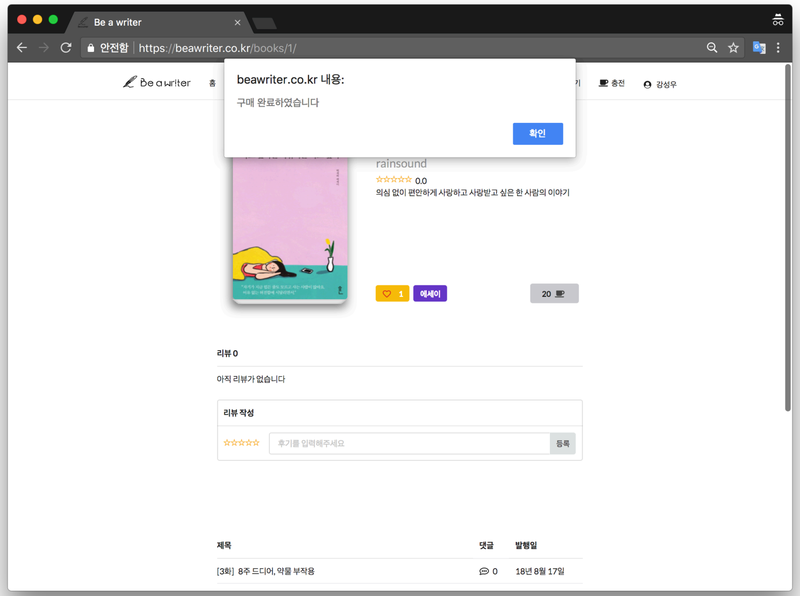
코인 주문 기능 코드 일부

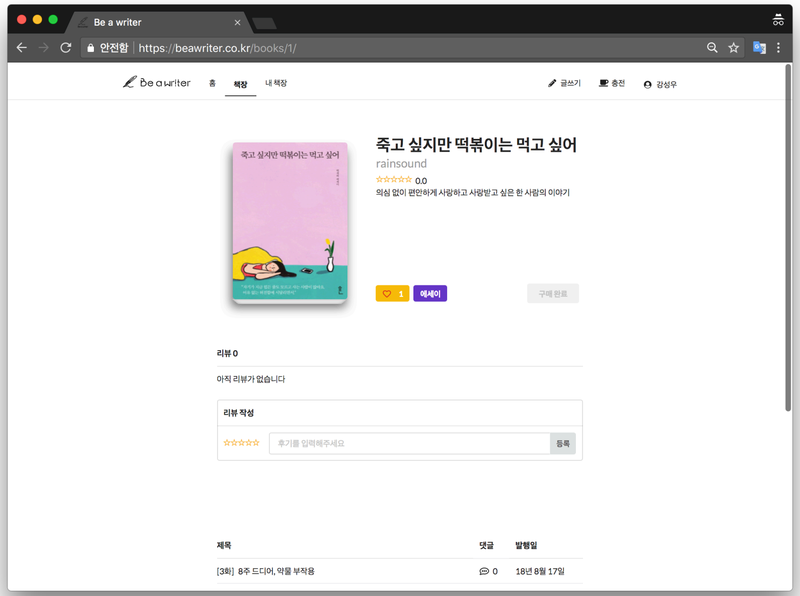
rainsound128@gmail.com

Be a writer

orders - 책 주문 기능(1/2)







책 주문 기능

rainsound128@gmail.com

```
def purchase_book(request, book_pk):
    book = get_object_or_404(Book, pk=book_pk)
    user = request.user

       ...

    elif user.coin >= book.amount:
        user.purchase_books.add(book)
        user.coin -= book.amount
        book.purchase_count += 1
        user.save()
        book.save()
        PurchaseBook.objects.create(
            user=user,
            book=book,
            book_amount=book.amount,
        )
        context = {
            'message': '구매 완료하였습니다',
        }

    ...

    return HttpResponse(json.dumps(context), content_type='application/json')
```

책 주문 기능 코드 일부