7장. 게시물 수정 & 삭제 기능



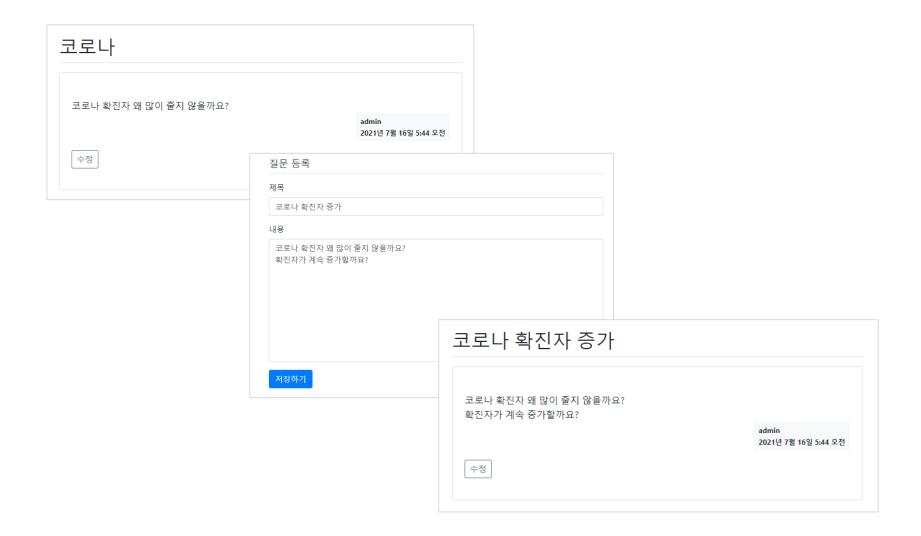
목 차

1

질문 수정 / 삭제

2

답변 수정 / 삭제



Qusetion. Answer 모델 수정

● Qusetion. Answer 모델에 modify_date 필드 추가

```
class Question(models.Model):
    ...
    modify_date = models.DateTimeField(null=True, blank=True)

class Answer(models.Model):
    ...
    modify_date = models.DateTimeField(null=True, blank=True)
```

null=True는 DB에서 modify_date 칼럼에 null을 허용한다는 의미 blank=True는 작성 폼에서 form.is_valid() 검사시 값이 없어도 된다는 의미

Qusetion. Answer 모델 수정

● Makemigrations, migrate 명령 수행하기

(mysite)C:\projects\mysite>python manage.py makemigrations (mysite)C:\projects\mysite>python manage.py migrate

● 질문 수정 기능 추가

1. 질문 수정 버튼 추가하기 – pybo/question_detail.html

```
<div class="d-flex justify-content-end">
  <div class="badge badge-light p-2 text-left">
    <div class="mb-2">{{ question.author.username }}</div>
    <div>{{ question.create_date }}</div>
  </div>
</div>
{% if request.user == question.author %}
<div class="my-3">
  <a href="{% url 'pybo:question_modify' question.id %}"
     class="btn btn-sm btn-outline-secondary">수정</a>
</div>
{% endif %}
```

2. 질문 수정 URL 매핑 추가하기 – pybo/urls.py

```
urlpatterns = [
    ....
    path('question/modify/<int:question_id>/', views.question_modify,
        name='question_modify'),
]
```

3. 질문 수정 함수 정의하기 – pybo/views.py

```
@login_required(login_url='common:login')
def question_modify(request, question_id):
   # 질문 수정
    question = qet_object_or_404(Question, pk=question_id)
    if request.user != question.author:
       messages.error(request, '수정 권한이 없습니다.')
       return redirect('pybo:detail', question_id=question.id)
    if request.method == "POST":
       form = QuestionForm(request.POST, instance=question)
        if form.is_valid():
            question = form.save(commit=False)
            question.author = request.user
            question.modify_date = timezone.now()
            question.save()
            return redirect('pybo:detail', question_id=question.id)
    else:
       form = QuestionForm(instance=question)
    context = {'form': form}
    return render(request, 'pybo/question_form.html', context)
```

- 3. 질문 수정 함수 정의하기 pybo/views.py
- messages 모듈은 장고가 제공하는 함수로 오류를 임의로 발생시키고 싶은 경우에 사용. 이때 폼 필드와 관련이 없으므로 넌필드 오류에 해당된다.
- instance 매개 변수에 question을 지정하면 기존 값을 폼에 채울 수 있다. 그렇지 않으면 오류 발생!!

IntegrityError at /pybo/question/modify/5

NOT NULL constraint failed: pybo_question.create_date

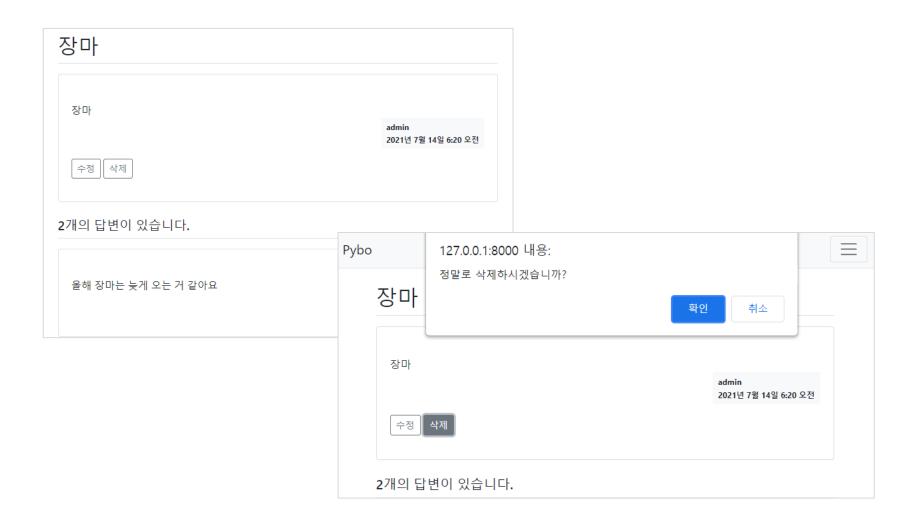
Request Method: POST

Request URL: http://127.0.0.1:8000/pybo/question/modify/5

Django Version: 3.2

Exception Type: IntegrityError

Exception Value: NOT NULL constraint failed: pybo_question.create_date



● 질문 삭제 기능 추가

1. 질문 삭제 버튼 추가하기 – pybo/question_detail.html

```
{% if request.user == question.author %}

<div class="my-3">
        <a href="{% url 'pybo:question_modify' question.id %}"
            class="btn btn-sm btn-outline-secondary">수정</a>

<a href="#" data-uri="{% url 'pybo:question_delete' question.id %}"
            class="delete btn btn-sm btn-outline-secondary">삭제</a>

</div>
{% endif %}
```

data-uri는 제이쿼리와 연동하는 부분 class의 delete를 선택함

2. 질문 삭제 URL 매핑 추가하기 – pybo/urls.py

```
urlpatterns = [
    ....
    path('question/delete/<int:question_id>/', views.question_delete,
        name='question_delete'),
]
```

3. 질문 삭제 함수 정의하기 – pybo/views.py

```
@login_required(login_url='common:login')

def question_delete(request, question_id):
    # 절문 삭제
    question = get_object_or_404(Question, pk=question_id)
    if request.user != question.author:
        messages.error(request, "삭제 권한이 없습니다.")
        return redirect('pybo:detail', question_id=question.id)

question.delete()
    return redirect('pybo:index')
```

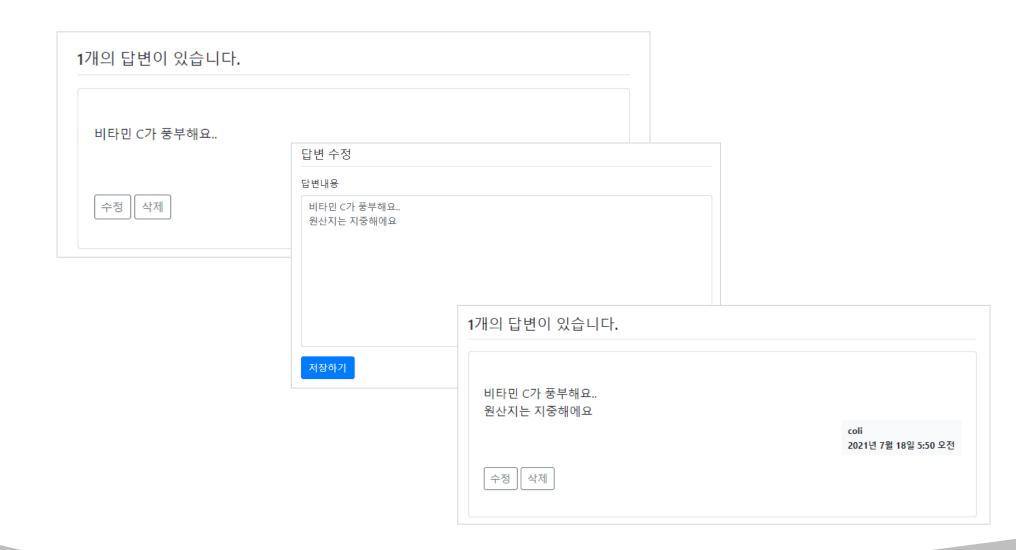
● 제이쿼리로 삭제 하기

1. base.html 변경하기 – pybo/base.html

```
<body>
{% include 'navbar.html' %}
<!--기본 템플릿 안에 삽입될 내용 Start-->
{% block content %}
{% endblock %}
<!--기본 탬플릿 안에 삽입될 내용 End-->
<script src="{% static 'jquery-3.6.0.js' %}"></script>
<script src="{% static 'bootstrap.js' %}"></script>
<!--자바 스크립트 Start-->
{% block script %}
{% endblock %}
<!--자바 스크립트 End-->
</body>
</html>
```

- 제이쿼리로 알림 창 만들기
 - 2. 질문 상세 페이지에 삭제 알림창 만들기 pybo/question_detail.html

```
{% endblock %}
{% block script %}
<script>
  $(function(){
    $(".delete").on('click', function(){
      if(confirm("정말로 삭제하시겠습니까?")){
       location.href=$(this).data('uri')
      }
   });
 });
</script>
{% endblock %}
```



- 답변 수정/삭제 기능 추가
 - 1. 답변 수정/삭제 버튼 추가하기 pybo/question_detail.html

```
{% if request.user == answer.author %}

<div class="my-3">

<a href="{% url 'pybo:answer_modify' answer.id %}"

        class="btn btn-sm btn-outline-secondary">수정</a>

<a href="#" data-uri="{% url 'pybo:answer_delete' answer.id %}"

        class="delete btn btn-sm btn-outline-secondary">삭제</a>

</div>
{% endif %}
```

2. 답변 수정/ 삭제 URL 매핑 추가하기 – pybo/urls.py

```
urlpatterns = [
    ....

path('answer/modify/<int:answer_id>/', views.answer_modify,
    name='answer_modify'),

path('answer/delete/<int:answer_id>/', views.answer_delete,
    name='answer_delete'),
]
```

3. 답변 수정 함수 정의하기 – pybo/views.py

```
@login_required(login_url='common:login')
def answer_modify(request, answer_id):
    # 답변 수정
    answer = get_object_or_404(Answer, pk=answer_id)
    if request.user != answer.author:
        messages.error(request, '수정 권한이 없습니다.')
        return redirect('pybo:detail', question_id=answer.question.id)
    if request.method == "POST":
        form = AnswerForm(request.POST, instance=answer)
        if form.is_valid():
            answer = form.save(commit=False)
            answer.author = request.user
            answer.modify_date = timezone.now()
            answer.save()
            return redirect('pybo:detail', question_id + answer.question.id)
    else:
        form = AnswerForm(instance=answer)
    context = {'form': form}
    return render(request, 'pybo/answer_form.html', context)
```

4. 답변 수정 폼(페이지) 작성하기 – pybo/answer_form.html

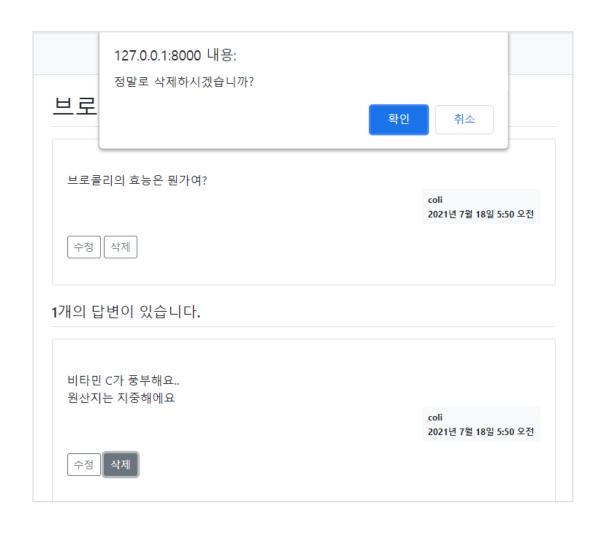
```
{% extends 'base.html' %}
{% block content %}
<div class="container">
  <h5 class="my-3 border-bottom pb-2">답변 수정</h5>
  <form method="post" class="post-form my-3">
   {% csrf token %}
   {% include 'form_errors.html' %}
   <div class="form-group">
      <label for="content">답변내용</label>
      <textarea class="form-control" name="content" id="content"
                rows="10">{{ form.content.value|default_if_none:''}}</textarea>
   </div>
    <button type="submit" class="btn btn-primary">저장하기</button>
  </form>
</div>
{% endblock %}
```

5. 답변 삭제 함수 정의하기 – pybo/views.py

```
@login_required(login_url='common:login')

def answer_delete(request, answer_id):
# 답변 삭제

answer = get_object_or_404(Answer, pk=answer_id)
if request.user != answer.author:
    messages.error(request, '삭제 권한이 없습니다.')
else:
    answer.delete()
return redirect('pybo:detail', question_id=answer.question.id)
```



● 수정일시 표시하기



● 수정일시(질문) 표시하기 – pybo/question_detail.py

```
<div class="card-text" style="white-space: pre-line;">
 {{ question.content }}
</div>
<div class="d-flex justify-content-end">
  {% if question.modify_date %}
  <div class="badge badge-light p-2 text-left mx-3">
    <div class="mb-2">modified at</div>
    <div>{{ question.modify_date }}</div>
  </div>
  {% endif %}
  <div class="badge badge-light p-2 text-left">
    <div class="mb-2">{{ question.author.username }}</div>
    <div>{{ question.create_date }}</div>
  </div>
```

● 수정일시(답변) 표시하기 – pybo/question_detail.py

```
<div class="card-body">
  <div class="card-text" style="white-space: pre-line">
    {{ answer.content }}
  </div>
  <div class="d-flex justify-content-end">
    {% if answer.modify_date %}
    <div class="badge badge-light p-2 text-left mx-3">
      <div class="mb-2">modified at</div>
      <div>{{ answer.modify_date }}</div>
    </div>
    {% endif %}
    <div class="badge badge-light p-2 text-left">
      <div class="mb-2">{{ answer.author.username }}</div>
      <div>{{ question.create_date }}</div>
    </div>
  </div>
```