

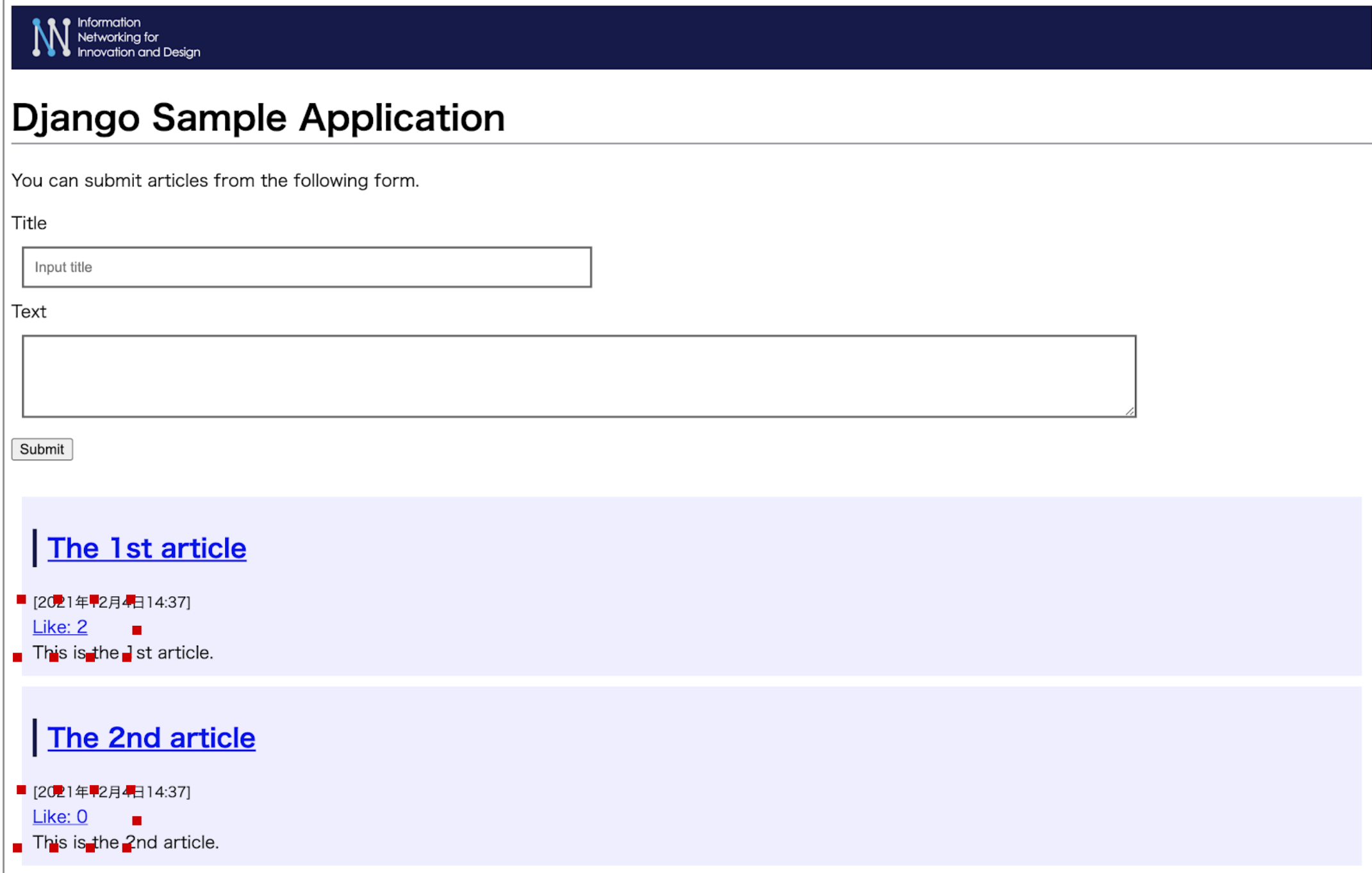
5

いいね機能の追加

「いいね」機能を追加してみましょう。

「いいね」を実装する

- ここでは「いいね機能」を追加することを考えてみます
- 一覧画面に、以下の機能を追加する
 - クリックすると数字を1増やし、詳細ページに遷移する



The screenshot displays the Django Sample Application interface. At the top, there is a dark blue header with the INIAD logo and the text 'Information Networking for Innovation and Design'. Below the header, the title 'Django Sample Application' is centered. A message states 'You can submit articles from the following form.' followed by a form with 'Title' and 'Text' input fields and a 'Submit' button. Below the form, there is a list of articles. The first article, 'The 1st article', has a date of '[2021年2月4日14:37]', a 'Like: 2' count, and the text 'This is the 1st article.' The second article, 'The 2nd article', has the same date, a 'Like: 0' count, and the text 'This is the 2nd article.'

手順

1. URLディスパッチャの追加
→ `blog/urls.py`
2. Controllerにおけるリクエストの処理の追加
→ `blog/views.py`
3. テンプレートの編集
→ `blog/templates/blog/index.html`

1. URLディスパッチャの追加：`blog/urls.py`

- URLディスパッチャが、以下のURLを受け付けるようにします
 - `http://127.0.0.1:8000/123/like`
→ `views.like` を呼び出す
- そのために、`urlpatterns`リストの最後に、次の内容を追加する
 - `path('<int:article_id>/like', views.like, name='like')`

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name='index'),
    ...
    path('<int:article_id>/update', views.update, name='update'),
    path('<int:article_id>/like', views.like, name='like'),
]
```

2. Controllerへの処理の追加 : `blog/views.py`

- `blog/views.py` の最後に、`views.like` 関数を定義する
 1. Article モデルから与えられたIDに対応したオブジェクトを取り出す
 - 見つからなかった場合は、404 Not Foundを返す
 2. オブジェクトの変数likeを変更し、保存する
 3. `views.detail` にリダイレクトする

```
...  
  
def like(request, article_id):  
    try:  
        article = Article.objects.get(pk=article_id)  
        article.like += 1  
        article.save()  
    except Article.DoesNotExist:  
        raise Http404("Article does not exist")  
  
    return redirect(detail, article_id)
```

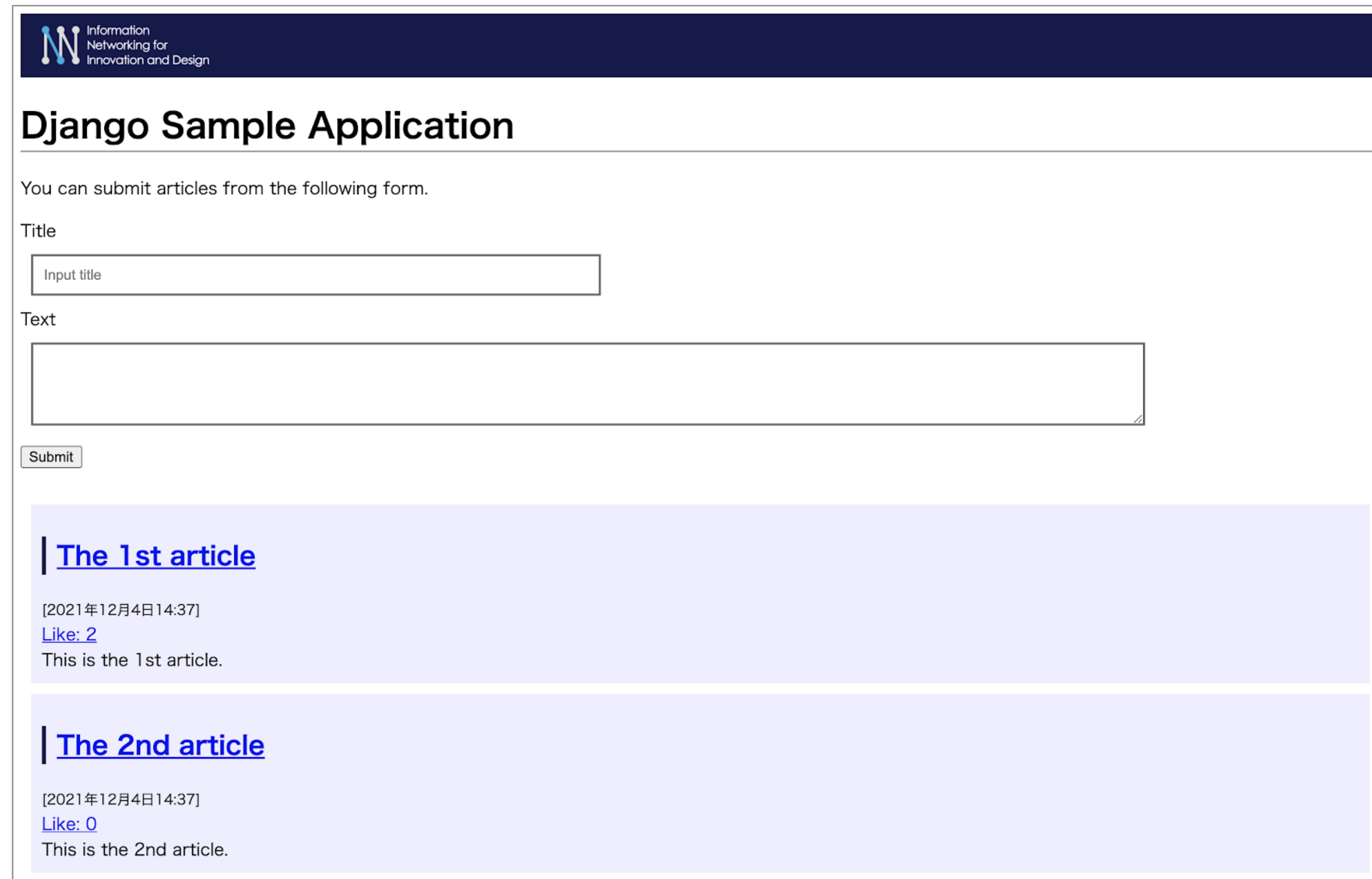
3. テンプレートの編集

- `blog/templates/blog/index.html` を編集し、「いいね」のリンクを追加する

```
...
{% for article in articles %}
<div class="text">
  <h2><a href="{% url 'detail' article.id %}">{{ article.title }}</a></h2>
  <small>[{{ article.posted_at }}]</small>
  <div>
    <a href="{% url 'like' article.id %}">
      Like: {{ article.like }}
    </a>
  </div>
  <div>
    {{ article.body |linebreaksbr }}
  </div>
</div>
{% endfor %}
...
```

動作確認

- Djangoサーバを起動し、「いいね」が動作することを確認しましょう



The screenshot displays the Django Sample Application interface. At the top, there is a dark blue header with the INIAD logo and the text "Information Networking for Innovation and Design". Below the header, the title "Django Sample Application" is centered. The main content area contains a form for submitting articles, with fields for "Title" and "Text", and a "Submit" button. Below the form, there is a list of existing articles. The first article is titled "The 1st article" and was posted on [2021年12月4日14:37] with 2 likes. The second article is titled "The 2nd article" and was also posted on [2021年12月4日14:37] with 0 likes.

Information Networking for Innovation and Design

Django Sample Application

You can submit articles from the following form.

Title

Text

Submit

[The 1st article](#)

[2021年12月4日14:37]
[Like: 2](#)
This is the 1st article.

[The 2nd article](#)

[2021年12月4日14:37]
[Like: 0](#)
This is the 2nd article.

<http://127.0.0.1:8000/>