

## Project 6

CRUD blog

Copy project of Project 4 website into Project 6

(1) templates → base.html

```
<div class = "new">
```

```
<h3> <a href = "{% url 'new-article' %}">
```

```
Add new Article </a> </h3>
```

```
</div>
```

python manage.py runserver ↵

(2) articles → views.py

```
from django.views.generic.edit import CreateView
```

```
class Article(CreateView(Create Base View)):
```

```
    model = article
```

```
    template_name = 'article_new.html'
```

```
    fields = '__all__'
```

(3) articles → urls.py

```
path('article/new/', views.ArticleCreateView.as_view(),  
      name = 'new_article')
```

↓  
base.html

(4) templates → article\_new.html (new file)

{% extends 'base.html' %}

{% block content %}

<h1>Enter your new Article </h1>

<form action="" method="POST">

{% csrf\_token %}

{{ form.as\_p }}

<input type="submit" value="Save">

</form>

{% endblock %}

(5) articles → models.py

```
from django.urls import reverse  
class ...  
!
```

```
def get_absolute_url(self):  
    return reverse('home')
```

→ name of url to redirect



- Update form database -

(1) templates → detail.html

to edit entire block

<div>

</div>

human: block

<h4><a href = "{% url 'article\_edit' %}" Edit this post

</a> </h4>

(2) create new template file article\_edit.html (template)

{% extends 'base.html' %}

{% block content %}

<h1> This is Edit page </h1>

<form action = "" method = "POST">

{% csrf\_token %}

{{ form.as\_p }}

<input type = "submit" value = "Update">

</form>

{% endblock %}

(3) articles → views.py

\_\_\_\_\_ (create view, UpdateView)

}

class ArticleUpdateView()

model = Article

template\_name = 'article\_edit.html'

fields = ['title', 'text']

form models.py

(4) articles → urls.py

path('article/<int:pk>/edit', views.ArticleUpdateView.as\_view(), name='article\_edit'),

form detail.html  
~~article\_edit.html~~

— Delete from database —

(1) templates → detail.html

<h4><a href="{% url 'article\_delete' pk %}">

Delete This POST </a></h4>

(2) create new file as article\_delete.html (templates)

{% extends 'base.html' %}

{% block content %}



<h2> This is an article delete page </h2>

<form action = "" method = "POST" >

{ % csrf\_token %}

<h3> WARNING !!! </h3>

<p> Are you sure that you want to delete ? {{ batman.title }}</p>

<input type = "submit" value = "Confirm" >

</form>

{ % endblock %}

(3) views.py

from django.urls import reverse\_lazy

class ArticleDeleteView ( DeleteView ):

model = Article

template\_name = 'delete.html'

context\_object\_name = 'batman'

success\_url = reverse\_lazy('home')

(4) article → vol-by

:

```
path ('article / <int:pk > / delete', view = ArticleDeleteView -  
as_view(), name = 'article_delete')
```