Нужно установить в проекте Django пагинатор. Вы можете это сделать в файле settings.py проекта, добавив в список установку следующую строку:

INSTALLED\_APPS = [

# ...

'django.core.paginator', # добавляем пагинатор

# ...

] # models.py

from django.db import models

class Post(models.Model):

title = models.CharField(max\_length=255)

content = models.TextField()

created\_at = models.DateTimeField(auto\_now\_add=True) # views.py

from django.shortcuts import render

from django.core.paginator import Paginator

from .models import Post

def post\_list(request):

# получаем все посты

posts = Post.objects.all()

# создаем пагинатор

paginator = Paginator(posts, 10) # 10 постов на странице

# получаем номер страницы, на которую переходит пользователь

page\_number = request.GET.get('page')

# получаем посты для текущей страницы

page\_posts = paginator.get\_page(page\_number)

# передаем контекст в шаблон

return render(request, 'post\_list.html', {'page\_posts': page\_posts})

<!-- post\_list.html -->

{% extends 'base.html' %}

{% block content %}

<h1>Список постов</h1>

<ul>

{% for post in page\_posts %}

<li>

{{ post.title }} ({{ post.created\_at }})

<p>{{ post.content }}</p>

</li>

{% empty %}

<p>Нет постов.</p>

{% endfor %}

</ul>

<!-- показываем пагинатор -->

<div class="pagination">

<p>Страницы:</p>

<ul>

{% if page\_posts.has\_previous %}

<li><a href="?page=1"><<</a></li>

<li><a href="?page={{ page\_posts.previous\_page\_number }}">Previous</a></li>

{% endif %}

<li>{{ page\_posts.number }}/{{ paginator.num\_pages }}</li>

{% if page\_posts.has\_next %}

<li><a href="?page={{ page\_posts.next\_page\_number }}">Next</a></li>

<li><a href="?page={{ paginator.num\_pages }}">>>></a></li>

{% endif %}

</ul>

</div>

{% endblock %}

Можно сделать обработку ошибок как дополнение. Например, если пользователь перешел на страницу, которая не существует, Django выдаст ошибку 404. Чтобы решить эту проблему, мы можем добавить проверку наличия страницы перед переходом на нее:

def post\_list(request):

# ...

page\_number = request.GET.get('page')

try:

page\_posts = paginator.get\_page(page\_number)

except PageNotAnInteger:

page\_posts = paginator.page(1)

except EmptyPage:

page\_posts = paginator.page(paginator.num\_pages)

return render(request, 'post\_list.html', {'page\_posts': page\_posts})