

09 – Pagination

Blog posts တွေ များပြားလာတာနဲ့ အမျှ မိမိရဲ့ website ဟာ ကြီးထွားလာသလို၊ မိမိရဲ့ home profile page မှာ posts list အပြည့်အစုံ ဖော်ပြဖို့ ခက်ခဲနေေးကွေးလာပါလိမ့်မယ်။ ကြီးထွားလာတဲ့ pages တွေကို generate လုပ်ဖို့၊ download လုပ်ဖို့၊ browser မှာ render လုပ်ဖို့ အချိန်ကြာမြင့်ပြီး user experience quality ကိုလည်း လျော့ကြစေပါတယ်။ အဲဒါတွေအတွက် ဖြေရှင်းနည်းကတော့ data ကို paginate လုပ်ပြီး၊ အပိုင်းလိုက် render လုပ်စေခြင်းပဲ ဖြစ်ပါတယ်။ Blog Posts တွေကို paginate လုပ်ကြည့်ရအောင်။

json format နဲ့ user account နှစ်ခုအတွက် blog post (25) ခု ရေးထားပါ။ json file ကိုဖတ်ပြီး database ထဲကို ထည့်ထားပေးဖို့ pag.py file နဲ့ အလုပ်လုပ်ပေးထားပါ။ ပြီးရင် paginate လုပ်ကြည့်ပါမယ်။ Python shell ကို သွားပါ။ Post model ကို import လုပ်ပါ။

```
>>> from flaskblog.models import Post
```

```
>>> posts = Post.query.all()
```

posts အားလုံးကို သုံးနေကျ query.all() method နဲ့ posts အားလုံးကို ကြည့်ပါ။ နောက် page တစ်ခုချင်း records ကို load လုပ်ရန် paginate() method ကို သုံးကြည့်ပါ။

```
>>> posts = Post.query.paginate()
```

```
>>> posts
```

```
>>> dir(posts)
```

paginate() method မှာ required argument အနေနဲ့ page number ကို ယူပေးတဲ့ page argument သာမဖြစ်မနေ လိုပါမယ်။ အခြား ရွေးချယ်အသုံးပြုနိုင်တဲ့ per_page argument နဲ့ page တစ်ခုမှာ ပြပေး ရမယ့် items အရေအတွက်ကို သတ်မှတ်နိုင်ပါတယ်။ default အနေနဲ့ page တစ်ခုမှာ items က အခု (20) ပါ ။

page တစ်ခုထဲမှာ ပါတဲ့ post အရေအတွက်ကို ကြည့်ချင်ရင်

```
>>> posts.per_page
```

လက်ရှိ page နံပါတ်ကို ကြည့်ချင်ရင်

```
>>> posts.page
```

ကြည့်ချင်တဲ့ page နံပါတ် နဲ့ page တစ်ခုမှာပါမယ့် items တွေသတ်မှတ်ပြီး ကြည့်ပါမယ်။

```
>>> posts = Post.query.paginate(page=2)
```

```
>>> for post in posts.items:
```

```
    print(post)
```

```
>>> posts = Post.query.paginate(per_page=5)
```

```
>>> posts.page
```

```
>>> for post in posts.items:
```

```
    print(post)
```

```
>>> posts = Post.query.paginate(per_page=5, page=2)
```

```
>>> posts.page
```

```
>>> for post in posts.items:
```

```
    print(post)
```

```
>>> posts.total
```

paginate() method ကို နားလည်ပြီးဆိုရင် လိုအပ်တဲ့ route တွေပြင်ပါမယ်။ "/" route ကိုသွားပါ။

query.all() နေရာမှာ paginate() အစားထိုးပါမယ်။ ပြထားတဲ့ code ကို ပြင်ရေးပါ။

```
page = request.args.get('page', 1, type=int)
```

```
posts = Post.query.paginate(page=page, per_page=5)
```

နောက် posts တွေပြပေးမယ့် home.html ရဲ့ for loop မှာ posts အစား posts.items ပြင်ပေးရပါမယ်။

ပြီးရင်တော့ run ကြည့်ပါမယ်။ ပထမ page ကို post (၅)ခု နဲ့ ပြပေးပါမယ်။ page နံပါတ်ကို query string ကနေယူထားတဲ့အတွက် url မှာ /page=2 အစရှိသဖြင့် ရိုက်ထည့်ပြီး နောက် page တွေကို ကြည့်မယ်ဆိုလည်းရပါမယ်။ မရှိတဲ့ page no ကို ရိုက်ထည့်ရင်တော့ error ပြပါမယ်။ ဒါဆို page

ဘယ်နှစ်ခုရှိမလဲ။ တစ်ခုနဲ့ တစ်ခု အလွယ်တကူ သွားနိုင်ဖို့ page links တွေထည့်ပါမယ်။ အဲဒီအတွက် pagination object ရဲ့ methods တွေထဲက iter_pages method ကို သုံးပါမယ်။

per_page မှာ post (2) ခု သတ်မှတ်ပေးပြီး၊ python shell ကို ပြန်သွားပြီး iter_pages ကို စမ်းကြည့်ပါမယ်။

```
>>> posts = Post.query.paginate(page=6, per_page=2)
>>> for page in posts.iter_pages():
    print(page)
```

ရှိတဲ့ post အရေအတွက်က (25) ခု page တစ်ခုမှာ post (2)ခုနဲ့ လက်ရှိရွေးထားတဲ့ page no က (6) ဆိုပါတော့...

iter_pages က ဘယ်လိုပြမလဲဆိုရင် 1,2, None, 4,5,6,7,8,9,10, None, 12,13 ဆိုပြီးပြပါမယ်။

1,2 => left_edge

4,5 => left_current

6 => current page

7,8,9,10 => right_current

12,13 => right_edge

None => gap in the sequence of pages တို့ ဖြစ်ပါတယ်။

ဒါဆို page links ထည့်ကြည့်ရအောင်။ main.html မှာ အောက်ပါ code ကို ရေးပြီး run ကြည့်ပါ။

```
{% for page_num in posts.iter_pages() %}
    {% if page_num %}
        <a class="btn btn-outline-info mb-4" href="{{ url_for('home',
page=page_num) }}">{{ page_num }} </a>
    {% else %}
        ...
    {% endif %}
{% endfor %}
```

ရွေးထားတဲ့ page နဲ့ left နဲ့ right မှာ ရှိတဲ့ page links တွေကို highlight လုပ်ကြည့်ပါမယ်။

ပေးထားတဲ့အတိုင်း code ကို ပြင်ပေးပါ။ ပြီးရင် run ကြည့်ပါ။

```
{% for page_num in posts.iter_pages(left_edge=1, right_edge=1, left_current=1,
right_current=2) %}
    {% if page_num %}
        {% if posts.page == page_num %}
            <a class="btn btn-info mb-4" href="{% url_for('home', page=page_num) %}">
{{ page_num }}</a>
        {% else %}
            <a class="btn btn-outline-info mb-4" href="{% url_for('home', page=page_num) %}">
{{ page_num }}</a>
        {% endif %}
    {% else %}
        ...
    {% endif %}
{% endfor %}
```

page link အနေနဲ့ အဆင်ပြေသွားပေမယ့် post ကို ဖော်ပြတဲ့အခါ ပထမဆုံး post မှ စပြနေပါမယ်။ တစ်ကယ်တော့ နောက်ဆုံးရေးလိုက်တဲ့ post က ထိပ်မှာ ပြပေးဖို့ လိုအပ်ပါတယ်။ အဲဒီအတွက် posts တွေကို order ပြောင်းပြန် စီပေးရပါမယ်။ "/" route မှာ ပြင်ပေးပါမယ်။

```
# posts = Post.query.paginate(page=page, per_page=5)
posts = Post.query.order_by(Post.date_posted.desc()).paginate(page=page,
per_page=5)
```

နောက်ဆုံးတင်တဲ့ post ကနေစပြဖို့ အဆင်ပြေသွားရင် user တစ်ဦးချင်းရဲ့ posts ကို ရွေးထုတ်ကြည့်ပါမယ်။ query.filter_by() နဲ့ user / post ကို ရွေးထုတ်မှာပါ။ ပြီးရင် user_posts.html

template file ကို render လုပ်ဖို့ (/user/<string:username>) route ရေးပါ။ home.html file ထဲက code ကို copy ကူးပြီးနောက်၊ လိုအပ်တဲ့ နေရာမှာ ပြင်ပေးလိုက်ပြီး run ကြည့်ပါမယ်။ code files တွေကို ကြည့်ပါ။

routes.py

```
import os, secrets

from PIL import Image

from flask import render_template, url_for, flash, redirect, request, abort
from flaskblog import app, db, bcrypt
from flaskblog.forms import RegistrationForm, LoginForm, UpdateAccountForm, PostForm
from flaskblog.models import User, Post
from flask_login import login_user, current_user, logout_user, login_required

@app.route("/")
@app.route("/home")
def home():
    page = request.args.get('page', 1, type=int)
    # posts = Post.query.paginate(page=page, per_page=5)
    posts = Post.query.order_by(Post.date_posted.desc()).paginate(page=page, per_page=5)
    return render_template('home.html', posts=posts)

@app.route("/user/<string:username>")
def user_posts(username):
    page = request.args.get('page', 1, type=int)
    user = User.query.filter_by(username=username).first_or_404()
    posts = Post.query.filter_by(author=user)\
```

```
.order_by(Post.date_posted.desc())\
.paginate(page=page, per_page=5)
return render_template('user_posts.html', posts=posts, user=user)
```

pag.py

```
from flaskblog import db
from flaskblog.models import Post
import json
```

```
with open('posts.json') as f:
    posts_json = json.load(f)
```

```
for post in posts_json:
    post = Post(title=post['title'], content=post['content'], user_id=post['user_id'])
    db.session.add(post)
    db.session.commit()
from PIL import Image
```

home.html

```
{% extends "layout.html" %}
{% block content %}
    {% for post in posts.items %}
        <article class="media content-section">
            
```

```

<div class="media-body">
    <div class="article-metadata">
        <a class="mr-2" href="{{ url_for('user_posts', username=post.author.username) }}">
{{ post.author.username }}</a>
        <small class="text-muted">{{ post.date_posted.strftime('%Y-%m-%d') }}</small>
    </div>
    <h2><a class="article-title" href="{{ url_for('post', post_id=post.id) }}">
{{ post.title }}</a></h2> <p class="article-content">{{ post.content }}</p>
    </div>
</article>
{% endfor %}

{% for page_num in posts.iter_pages(left_edge=1, right_edge=1, left_current=1,
right_current=2) %}
    {% if page_num %}
        {% if posts.page == page_num %}
            <a class="btn btn-info mb-4" href="{{ url_for('home', page=page_num) }}">
{{ page_num }}</a>
        {% else %}
            <a class="btn btn-outline-info mb-4" href="{{ url_for('home', page=page_num) }}">
{{ page_num }}</a>
        {% endif %}
    {% else %}
        ...
    {% endif %}
{% endfor %}
{% endblock content %}

```

user_posts.html

```
{% extends "layout.html" %}
```

```
{% block content %}
```

```
    <h1 class="mb-3">Posts by {{ user.username }} ({{ posts.total }})</h1>
```

```
    {% for post in posts.items %}
```

```
        <article class="media content-section">
```

```
            
```

```
            <div class="media-body">
```

```
                <div class="article-metadata">
```

```
                    <a class="mr-2" href="{{ url_for('user_posts', username=post.author.username) }}">
```

```
                    {{ post.author.username }}</a>
```

```
                    <small class="text-muted">{{ post.date_posted.strftime('%Y-%m-%d') }}</small>
```

```
                </div>
```

```
                <h2><a class="article-title" href="{{ url_for('post', post_id=post.id) }}">
```

```
                {{ post.title }}</a></h2>
```

```
                <p class="article-content">{{ post.content }}</p>
```

```
            </div>
```

```
        </article>
```



```

{% endfor %}

{% for page_num in posts.iter_pages(left_edge=1, right_edge=1, left_current=1,
    right_current=2) %}
    {% if page_num %}
        {% if posts.page == page_num %}
            <a class="btn btn-info mb-4" href="{{ url_for('user_posts', username=user.username,
                page=page_num) }}">{{ page_num }}</a>
        {% else %}
            <a class="btn btn-outline-info mb-4" href="{{ url_for('user_posts',
                username=user.username, page=page_num) }}">{{ page_num }}</a>
        {% endif %}
    {% else %}
        ...
    {% endif %}
{% endfor %}

{% endblock content %}

```