08 – Create, Update, and Delete Posts

လေ့လာခဲ့ပြီးသော chapter တွေမှာ home page မှဖော်ပြပေးသော Blog post တွေအတွက် dummy data တွေကို သုံးခဲ့ပါတယ်။ ယခု အသုံးပြုသူ user များကို post အသစ်များ ရေးသားခွင့်၊ ပြင်ဆင်ခွင့်၊ ပေးရန် အတွက် code ရေးပါမယ်။ post/new route ထပ်ထည့်ပါမယ်။ post/new လို့ request တောင်းရင် Post ရေးဖို့ Post Form တစ်ခုချပေးရမယ်။ အဲဒီအတွက် forms.py မှာ PostForm class တစ်ခုရေးပါမယ်။ form ရဲ့ content field က TextAreaField သုံးထားတဲ့အတွက် TextAreaField ကို import လုပ်ရပါမယ်။ ပြီးရင်တော့ ပြထားတဲ့ code ကိုရေးပါ။

```
class PostForm(FlaskForm):
    title = StringField('Title', validators=[DataRequired()])
    content = TextAreaField('Content', validators=[DataRequired()])
    submit = SubmitField('Post')
```

နောက် create_post.html template file ဖန်တီးပါ။ (code ကို file မှာကြည့်ပါ)။ ပြီးရင် route ရေးပါမယ်။

```
@app.route("/post/new", methods=['GET', 'POST'])
@login_required
def new_post():
    form = PostForm()
        if form.validate_on_submit():
        flash('Your post has been created!', 'success')
        return redirect(url_for('home'))
        return render_template('create_post.html', title='New Post', form=form)
```

application ရဲ့ nav-bar ၊ right-side မှာ New Post ဆိုတဲ့ link item တစ်ခုထည့်ပါမယ်။

```
<a class="nav-item nav-link" href="{{ url_for('new_post') }}">New Post</a>
```

ပြီးရင် run ကြည့်ပါမယ်။ post ထည့်လို့ရပါပြီ။ ဒါပေမယ့် home page မှာ မပြပါဘူး။ ဘာကြောင့်လဲ ဆိုတော့ dummy data ပဲ သုံးထားဆဲမို့ပါ။ dummy data ကိုဖျက်၊ ရေးလိုက်တဲ့ post ကို သိမ်းပြီး၊ home page ပြဖို့ လိုအပ်တာရေးပါမယ်။ post/new route မှာ ပေးထားတဲ့ code ကို ထပ်ဖြည့်ပါ။

```
post = Post(title=form.title.data, content=form.content.data,


                        author=current_user)
db.session.add(post)
db.session.commit()
```

home route မှာတော့ database မှ post ကို query လုပ်ပေးရပါမယ်။

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

run ကြည့်လိုက်ရင် post ကို ပြပေးပါပြီ။ ဒါပေမယ့် လိုချင်တဲ့ပုံစံထပ်ပြင်ပါအုန်းမယ်။ post ရဲ့ authur name မှာ username ပဲပြပေးပြီး၊ date ကိုလည်း ပုံစံပြောင်းပေးပါမယ်။ နောက် picture ထည့်ပါမယ်။ လိုတာအားလုံး ကို home.html မှာ ပြင်ပေးပါ။ result ကို ကြည့်ပါ ။

နောက် post တွေကို id နဲ့ ရွေးထုတ်ကြည့်ချင်ပါတယ်။ အဲဒီအတွက် post/post_id route ရေးပါမယ်။

```
@app.route("/post/<int:post_id>")
def post(post_id):
    post = Post.query.get_or_404(post_id)
    return render_template('post.html', title=post.title, post=post)
```

post ကို id နဲ့ ဆွဲထုတ်တဲ့အခါ id ရှိရင် ပြပေးပြီး၊ မရှိတဲ့ id တောင်းမိရင် error ဖော်ပြရအောင် query.get_or_404 method ကို သုံးထားတာပါ။ post.html ကို render လုပ်ထားတဲ့အတွက် post.html ကို ရေးပါမယ်။ home.html အတိုင်းဖော်ပြမှာ ဖြစ်တဲ့အတွက် code ကို copy ကူးထည့်ပါ ။ ဒါပေမယ့် home က post အားလုံးကို ပြပြီး၊ id က ရွေးထုတ်ပြရမှာမို့ loop ကို ဖျက်ပေးရပါမယ်။ နောက် home မှာ ဖော်ပြထား တဲ့ post တွေရဲ့ post title link တွေကို post.html နဲ့ ပြန်ချိတ်ပေးပါ။ လိုအပ်တာတွေ ပြန်ပြင်ပြီးရင်တော့ run ကြည့်လို့ရပါပြီ။

post ကို ဖျက်ဖို့ ၊ ပြင်ဖို့ ရေးကြပါမယ်။ post/<post_id>/update route ကို ရေးပါမယ်။ post ကို ရေး၊ပြင်၊ဖျက် အတွက် login လုပ်ပြီးသော current_user ကို သာ အခွင့်ပေးမှာပါ။ အကယ်၍ current_user မဟုတ်ခဲ့ရင် error ကို status code (403)နဲ့ abort လုပ်ပါမယ်။ current_user ဖြစ်ခဲ့ရင် render လုပ်ပေးမဲ့ page က create_post.html template နဲ့ အတူတူပါဘဲ။ ဒါပေမယ့် create_post.html ကို post/new မှာလည်းသုံးထားတဲ့အတွက် legend field ကို ပြင်ပေးပါ။

update route မှာ GET request အတွက် post အဟောင်းကို ခေါ်ထည့်ပေးထားပြီး၊ POST request အတွက် ပြင်လိုက်သော post ကို post.html နဲ့ ပြပေးဖို့ ရေးပေးပါမယ်။

```
@app.route("/post/<int:post_id>/update", methods=['GET', 'POST'])
@login_required
def update_post(post_id):
    post = Post.query.get_or_404(post_id)
    if post.author != current_user:
            abort(403)
    form = PostForm()
    if form.validate_on_submit():
            post.title = form.title.data
            post.content = form.content.data
            db.session.commit()
            flash('Your post has been updated!', 'success')
            return redirect(url_for('post', post_id=post.id))
    elif request.method == 'GET':
            form.title.data = post.title
            form.content.data = post.content
    return render_template('create_post.html', title='Update Post',
                                                    form=form, legend='Update Post')
```

update / delete လုပ်ဖို့ Post.html မှာ update button နဲ့ delete button ထည့်ပါမယ်။ update ကိုရွေးရင် update form ပေးပြီး၊ delete ကို ရွေးရင်တော့ model တစ်ခုချပေးဖို့ ရေးပေးပါမယ်။ model အတွက် code ကို bootstrap မှ demo code ကို copy ကူးထည့်ပြီး လိုအပ်သလိုပြင်ပေးပါ။ အဓိက form element နဲ့ submit button ထည့်ပါ။ model box မှ delete ကို နှိပ်လျှင် delete_post လုပ်ပေးရမှာဖြစ်တဲ့အတွက် post/<post_id>/delete route ကို ရေးပါမယ်။

```
@app.route("/post/<int:post_id>/delete", methods=['POST'])
@login_required
def delete_post(post_id):
    post = Post.query.get_or_404(post_id)
        if post.author != current_user:
            abort(403)
            db.session.delete(post)
            db.session.commit()
            flash('Your post has been deleted!', 'success')
            return redirect(url_for('home'))
```

application ကို run ပြီး၊ post များကို ထည့်၊ ပြင်၊ ဖျက်ကြည့်ပါ။

routes.py

```
import os
import 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("/post/new", methods=['GET', 'POST'])
@login_required
def new_post():
    form = PostForm()
    if form.validate_on_submit():
        post = Post(title=form.title.data, content=form.content.data, author=current_user)
        db.session.add(post)
        db.session.commit()
        flash('Your post has been created!', 'success')
        return redirect(url_for('home'))
    return render_template('create_post.html', title='New Post', form=form, legend='New
Post')

@app.route("/post/<int:post_id>")
def post(post_id):
    post = Post.query.get_or_404(post_id)
    return render_template('post.html', title=post.title, post=post)

@app.route("/post/<int:post_id>/update", methods=['GET', 'POST'])
@login_required
def update_post(post_id):
    post = Post.query.get_or_404(post_id)
    if post.author != current_user:
```

```
        abort(403)
    form = PostForm()
    if form.validate_on_submit():
        post.title = form.title.data
        post.content = form.content.data
        db.session.commit()
        flash('Your post has been updated!', 'success')
        return redirect(url_for('post', post_id=post.id))
    elif request.method == 'GET':
        form.title.data = post.title
        form.content.data = post.content
    return render_template('create_post.html', title='Update Post',
form=form, legend='Update Post')

@app.route("/post/<int:post_id>/delete", methods=['POST'])
@login_required
def delete_post(post_id):
    post = Post.query.get_or_404(post_id)
    if post.author != current_user:
        abort(403)
    db.session.delete(post)
    db.session.commit()
    flash('Your post has been deleted!', 'success')
    return redirect(url_for('home'))
```

forms.py

```
from flask_wtf import FlaskForm
from flask_wtf.file import FileField, FileAllowed
from flask_login import current_user
from wtforms import StringField, PasswordField, SubmitField, BooleanField, TextAreaField
from wtforms.validators import DataRequired, Length, Email, EqualTo, ValidationError
from flaskblog.models import User

class RegistrationForm(FlaskForm):
...



class LoginForm(FlaskForm):
...

class UpdateAccountForm(FlaskForm):
...

class PostForm(FlaskForm):
    title = StringField('Title', validators=[DataRequired()])
    content = TextAreaField('Content', validators=[DataRequired()])
    submit = SubmitField('Post')
```

layout.html

```
<!DOCTYPE html>
<html>
<head> …
</head>
<body>
   <header class="site-header">
         …
          <!-- Navbar Right Side -->
          <div class="navbar-nav">
            {% if current_user.is_authenticated %}
              <a class="nav-item nav-link" href="{{ url_for('new_post') }}">New Post</a>
              …
   </header>
   <main role="main" class="container"> …
   </main>

   <!-- Optional JavaScript -->
   …
</body>
</html>
```

home.html

```
{% extends "layout.html" %}
{% block content %}
    {% for post in posts %}
        <article class="media content-section">
          <img class="rounded-circle article-img" src="{{ url_for('static',
filename='profile_pics/' + post.author.image_file) }}">
          <div class="media-body">
            <div class="article-metadata">
              <a class="mr-2" href="#">{{ 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 %}
{% endblock content %}
```

create_post.html

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

```
<div class="content-section">
    <form method="POST" action="">
        {{ form.hidden_tag() }}
        <fieldset class="form-group">
            <legend class="border-bottom mb-4">{{ legend }}</legend>
            <div class="form-group">
                {{ form.title.label(class="form-control-label") }}
                {% if form.title.errors %}
                    {{ form.title(class="form-control form-control-lg is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.title.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.title(class="form-control form-control-lg") }}
                {% endif %}
            </div>

            <div class="form-group">
                {{ form.content.label(class="form-control-label") }}
                {% if form.content.errors %}
                    {{ form.content(class="form-control form-control-lg is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.content.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
```

```
                </div>
            {% else %}
                {{ form.content(class="form-control form-control-lg") }}
            {% endif %}
        </div>
    </fieldset>

    <div class="form-group">
        {{ form.submit(class="btn btn-outline-info") }}
    </div>
  </form>
</div>
{% endblock content %}
```

post.html

```
{% extends "layout.html" %}
{% block content %}
  <article class="media content-section">
    <img class="rounded-circle article-img" src="{{ url_for('static', filename='profile_pics/' +
post.author.image_file) }}">
    <div class="media-body">
      <div class="article-metadata">
        <a class="mr-2" href="#">{{ post.author.username }}</a>
        <small class="text-muted">{{ post.date_posted.strftime('%Y-%m-%d') }}</small>
        {% if post.author == current_user %}
```

```
          <div>
            <a class="btn btn-secondary btn-sm mt-1 mb-1" href="{{ url_for('update_post',
post_id=post.id) }}">Update</a>
            <button type="button" class="btn btn-danger btn-sm m-1" data-toggle="modal"
data-target="#deleteModal">Delete</button>
          </div>
        {% endif %}
      </div>
      <h2 class="article-title">{{ post.title }}</h2>
      <p class="article-content">{{ post.content }}</p>
    </div>
  </article>

  <!-- Modal -->
  <div class="modal fade" id="deleteModal" tabindex="-1" role="dialog"
aria-labelledby="deleteModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title" id="deleteModalLabel">Delete Post?</h5>
          <button type="button" class="close" data-dismiss="modal" aria-label="Close">
            <span aria-hidden="true">&times;</span>
          </button>
        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
```

```
          <form action="{{ url_for('delete_post', post_id=post.id) }}" method="POST">
            <input class="btn btn-danger" type="submit" value="Delete">
          </form>
        </div>
      </div>
    </div>
  </div>
{% endblock content %}
```