**FLASK BLOG WITH DB MAIN**

post.html

<title>Posts</title>

<h2>Create New Blog Post:</h2>

<form action='/posts' method='POST'>

Title: <br>

<input type='text' name='title' id='title' class="form-control"> <br>

Author: <br>

<input type='text' name='author' id='author' class="form-control"> <br>

Post: <br>

<input type='text' name='content' id='content' class="form-control"> <br>

<input type='submit' value='Post' class="btn btn-success col-sm-3"> </form>

<hr>

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

<h3>By: {{ post.author }}</h3>

<h3>By: N/A</h3>

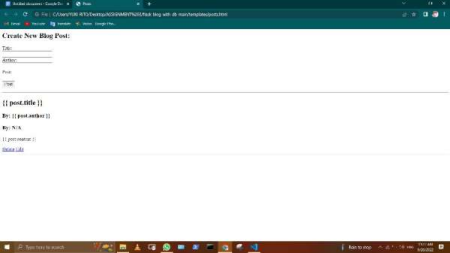
<p>{{ post.content }}</p>

<a href='/posts/delete/{{post.id}}'>Delete</a>

<a href='/posts/edit/{{post.id}}'>Edit</a>

<hr>

IMAGES:



● Index.html

<title>Home</title>

<h1>Home Page</h1>

<hr>

<h2>{YUKI RITO}</h2>

<h3>By: {YUKI RITO.author }</h3>

<h3>By: N/A</h3>

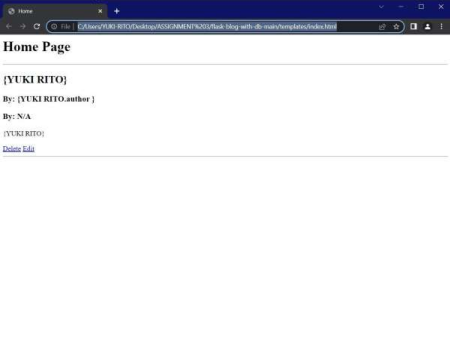
<p>{YUKI RITO}</p>

<a href='/posts/delete/{{post.id}}'>Delete</a>

<a href='/posts/edit/{{post.id}}'>Edit</a>

<hr>

IMAGE:



● EDIT.html

<title>Edit Post</title>

<h1>YUKI RITO</h1>

<hr>

<h2>Edit:</h2>

<form action='/posts/edit/{{post.id}}' method='POST'>

Title: <input type='text' name='title' id='title' value="{{post.title}}">

<br>

Author: <input type='text' name='author' id='author' value="{{post.author}}">

<br>

Post: <input type='text' name='content' id='content' value="{{post.content}}">

<br>

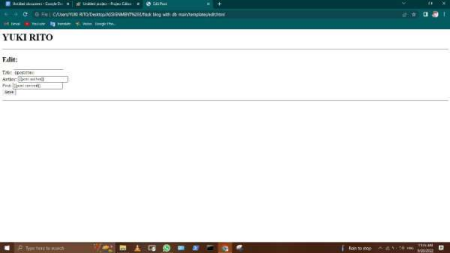
<input type='submit' value='Save'>

</form>

<hr>

IMAGE:

IMAGE:



● Base.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-

EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"

crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-

MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- <link rel="stylesheet" href="{{ url\_for('static', filename='css/main.css') }}"> --> </head>

<body>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="container-fluid">

<a class="navbar-brand" href="/">Blog</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="/">HOME</a>

</li>

<li class="nav-item">

<a class="nav-link" href="/posts">NEW POST</a>

</li>

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

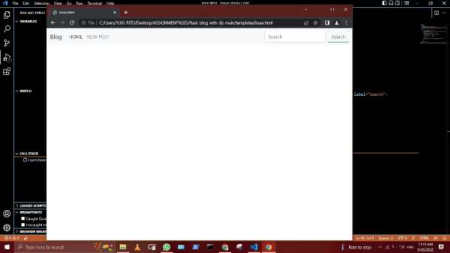
<div class="container py-5">

</div>

</body>

</html>

IMAGE:



● App.py

from flask import Flask, render\_template, request, redirect

from flask\_sqlalchemy import SQLAlchemy

from datetime import datetime

app = Flask(\_\_name\_\_)

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///posts.db'

db = SQLAlchemy(app)

class BlogPost(db.Model):

id = db.Column(db.Integer, primary\_key=True)

title = db.Column(db.String(100), nullable=False)

content = db.Column(db.Text, nullable=False)

author = db.Column(db.String(20), nullable=False, default='N/A')

date\_posted = db.Column(db.DateTime, nullable=False, default=datetime.utcnow)

def \_\_repr\_\_(self):

return 'Blog post ' + str(self.id)

@app.route('/')

def index():

all\_posts = BlogPost.query.order\_by(BlogPost.date\_posted).all()

return render\_template('index.html', posts=all\_posts)

@app.route('/posts', methods=['GET', 'POST'])

def posts():

if request.method == 'POST':

post\_title = request.form['title']

post\_content = request.form['content']

post\_author = request.form['author']

new\_post = BlogPost(title=post\_title, content=post\_content, author=post\_author) db.session.add(new\_post)

db.session.commit()

return redirect('/posts')

else:

all\_posts = BlogPost.query.order\_by(BlogPost.date\_posted).all()

return render\_template('posts.html', posts=all\_posts)

@app.route('/posts/delete/<int:id>')

def delete(id):

post = BlogPost.query.get\_or\_404(id)

db.session.delete(post)

db.session.commit()

return redirect('/posts')

@app.route('/posts/edit/<int:id>', methods=['GET', 'POST'])

def edit(id):

post = BlogPost.query.get\_or\_404(id)

if request.method == 'POST':

post.title = request.form['title']

post.author = request.form['author']

post.content = request.form['content']

db.session.commit()

return redirect('/posts')

else:

return render\_template('edit.html', post=post)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)