from flask import Flask, render\_template, flash, redirect, url\_for, session, request, logging

#from data import Articles

from flask\_mysqldb import MySQL

from wtforms import Form, StringField, TextAreaField, PasswordField, validators

from passlib.hash import sha256\_crypt

from functools import wraps

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

# Config MySQL

app.config['MYSQL\_HOST'] = '127.0.0.1'

app.config['MYSQL\_USER'] = 'root'

app.config['MYSQL\_DB'] = 'flaskmyproject'

app.config['MYSQL\_CURSORCLASS'] = 'DictCursor'

# init MYSQL

mysql = MySQL(app)

#Articles = Articles()

# Index

@app.route('/')

def index():

return render\_template('home.html')

# About

@app.route('/about')

def about():

return render\_template('about.html')

# Articles

@app.route('/articles')

def articles():

# Create cursor

cur = mysql.connection.cursor()

# Get articles

result = cur.execute("SELECT \* FROM articles")

articles = cur.fetchall()

if result > 0:

return render\_template('articles.html', articles=articles)

else:

msg = 'No Articles Found'

return render\_template('articles.html', msg=msg)

# Close connection

cur.close()

#Single Article

@app.route('/article/<string:id>/')

def article(id):

# Create cursor

cur = mysql.connection.cursor()

# Get article

result = cur.execute("SELECT \* FROM articles WHERE id = %s", [id])

article = cur.fetchone()

return render\_template('article.html', article=article)

# Register Form Class

class RegisterForm(Form):

name = StringField('Name', [validators.Length(min=1, max=50)])

username = StringField('Username', [validators.Length(min=4, max=25)])

email = StringField('Email',[validators.Email(message=None, granular\_message=False, check\_deliverability=False, allow\_smtputf8=True, allow\_empty\_local=False)])

password = PasswordField('Password', [

validators.DataRequired(),

validators.EqualTo('confirm', message='Passwords do not match')

])

confirm = PasswordField('Confirm Password')

# User Register

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

def register():

form = RegisterForm(request.form)

if request.method == 'POST' and form.validate():

name = form.name.data

email = form.email.data

username = form.username.data

password = sha256\_crypt.encrypt(str(form.password.data))

# Create cursor

cur = mysql.connection.cursor()

# Execute query

cur.execute("INSERT INTO users(name, email, username, password) VALUES(%s, %s, %s, %s)", (name, email, username, password))

# Commit to DB

mysql.connection.commit()

# Close connection

cur.close()

flash('You are now registered and can log in', 'success')

return redirect(url\_for('login'))

return render\_template('register.html', form=form)

# User login

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

def login():

if request.method == 'POST':

# Get Form Fields

username = request.form['username']

password\_candidate = request.form['password']

# Create cursor

cur = mysql.connection.cursor()

# Get user by username

result = cur.execute("SELECT \* FROM users WHERE username = %s", [username])

if result > 0:

# Get stored hash

data = cur.fetchone()

password = data['password']

# Compare Passwords

if sha256\_crypt.verify(password\_candidate, password):

# Passed

session['logged\_in'] = True

session['username'] = username

flash('You are now logged in', 'success')

return redirect(url\_for('dashboard'))

else:

error = 'Invalid login'

return render\_template('login.html', error=error)

# Close connection

cur.close()

else:

error = 'Username not found'

return render\_template('login.html', error=error)

return render\_template('login.html')

# Check if user logged in

def is\_logged\_in(f):

@wraps(f)

def wrap(\*args, \*\*kwargs):

if 'logged\_in' in session:

return f(\*args, \*\*kwargs)

else:

flash('Unauthorized, Please login', 'danger')

return redirect(url\_for('login'))

return wrap

# Logout

@app.route('/logout')

@is\_logged\_in

def logout():

session.clear()

flash('You are now logged out', 'success')

return redirect(url\_for('login'))

# Dashboard

@app.route('/dashboard')

@is\_logged\_in

def dashboard():

# Create cursor

cur = mysql.connection.cursor()

# Get articles

#result = cur.execute("SELECT \* FROM articles")

# Show articles only from the user logged in

result = cur.execute("SELECT \* FROM articles WHERE author = %s", [session['username']])

articles = cur.fetchall()

if result > 0:

return render\_template('dashboard.html', articles=articles)

else:

msg = 'No Articles Found'

return render\_template('dashboard.html', msg=msg)

# Close connection

cur.close()

# Article Form Class

class ArticleForm(Form):

title = StringField('Title', [validators.Length(min=1, max=200)])

body = TextAreaField('Body', [validators.Length(min=30)])

# Add Article

@app.route('/add\_article', methods=['GET', 'POST'])

@is\_logged\_in

def add\_article():

form = ArticleForm(request.form)

if request.method == 'POST' and form.validate():

title = form.title.data

body = form.body.data

# Create Cursor

cur = mysql.connection.cursor()

# Execute

cur.execute("INSERT INTO articles(title, body, author) VALUES(%s, %s, %s)",(title, body, session['username']))

# Commit to DB

mysql.connection.commit()

#Close connection

cur.close()

flash('Article Created', 'success')

return redirect(url\_for('dashboard'))

return render\_template('add\_article.html', form=form)

# Edit Article

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

@is\_logged\_in

def edit\_article(id):

# Create cursor

cur = mysql.connection.cursor()

# Get article by id

result = cur.execute("SELECT \* FROM articles WHERE id = %s", [id])

article = cur.fetchone()

cur.close()

# Get form

form = ArticleForm(request.form)

# Populate article form fields

form.title.data = article['title']

form.body.data = article['body']

if request.method == 'POST' and form.validate():

title = request.form['title']

body = request.form['body']

# Create Cursor

cur = mysql.connection.cursor()

app.logger.info(title)

# Execute

cur.execute ("UPDATE articles SET title=%s, body=%s WHERE id=%s",(title, body, id))

# Commit to DB

mysql.connection.commit()

#Close connection

cur.close()

flash('Article Updated', 'success')

return redirect(url\_for('dashboard'))

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

# Delete Article

@app.route('/delete\_article/<string:id>', methods=['POST'])

@is\_logged\_in

def delete\_article(id):

# Create cursor

cur = mysql.connection.cursor()

# Execute

cur.execute("DELETE FROM articles WHERE id = %s", [id])

# Commit to DB

mysql.connection.commit()

#Close connection

cur.close()

flash('Article Deleted', 'success')

return redirect(url\_for('dashboard'))

#comments

class CommentForm(Form):

body = TextAreaField('Body', [validators.Length(min=1)])

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

@is\_logged\_in

def comments():

form = CommentForm(request.form)

if request.method == 'POST' and form.validate():

body = form.body.data

# Create cursor

cur = mysql.connection.cursor()

# Execute

cur.execute("INSERT INTO comments(username, body) VALUES(%s, %s )",( body, session['username']))

# Commit to DB

mysql.connection.commit()

#Close connection

cur.close()

flash('Comment Added', 'success')

return redirect(url\_for('dashboard'))

return render\_template('comments.html', form=form)

if \_\_name\_\_ == '\_\_main\_\_':

app.secret\_key='secret123'

app.run(debug=True)

HTML FILES :

Navbar.html

<!-- <nav class="navbar navbar-default">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" href="#">MyFlaskApp</a>

</div>

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

<ul class="nav navbar-nav">

<li><a href="/">Home</a></li>

<li><a href="/about">About</a></li>

<li><a href="/articles">Articles</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

{% if session.logged\_in %}

<li><a href="/dashboard">Dashboard</a></li>

<li><a href="/logout">Logout</a></li>

{% else %}

<li><a href="/register">Register</a></li>

<li><a href="/login">Login</a></li>

{% endif %}

</ul>

</div>

</div>

</nav> -->

<header class="text-gray-600 body-font">

<div class="container mx-auto flex flex-wrap p-5 flex-col md:flex-row items-center">

<a class="flex title-font font-medium items-center text-gray-900 mb-4 md:mb-0">

<svg xmlns="http://www.w3.org/2000/svg" fill="none" stroke="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2" class="w-10 h-10 text-white p-2 bg-indigo-500 rounded-full" viewBox="0 0 24 24">

<path d="M12 2L2 7l10 5 10-5-10-5zM2 17l10 5 10-5M2 12l10 5 10-5"></path>

</svg>

<span class="ml-3 text-xl">BlogSpace</span>

</a>

<nav class="md:ml-auto md:mr-auto flex flex-wrap items-center text-base justify-center">

<a class="mr-5 hover:text-gray-900" href="/" ><button>Home</button></a>

<a class="mr-5 hover:text-gray-900" href="/articles" ><button>Blogs</button></a>

<a class="mr-5 hover:text-gray-900" href="/comments" ><button>Comments</button></a>

<a class="mr-5 hover:text-gray-900" href="/about" ><button>About</button></a>

</nav>

<ul class="nav navbar-nav navbar-right">

{% if session.logged\_in %}

<li><a href="/dashboard">Dashboard</a></li>

<li><a href="/logout">Logout</a></li>

{% else %}

<li><a href="/register">Register</a></li>

<li><a href="/login">Login</a></li>

{% endif %}

</ul>

<!-- <svg fill="none" stroke="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2" class="w-4 h-4 ml-1" viewBox="0 0 24 24">

<path d="M5 12h14M12 5l7 7-7 7"></path>

</svg> -->

</div>

</header>

Home.html

{% extends 'layout.html' %}

{% block body %}

<div class="jumbotron text-center">

<h1>Welcome To BlogSpace</h1>

<p class="lead">Hello nice to meet you you can start writing blogs after registration</p>

{% if session.logged\_in == NULL %}

<a href="/register" class="btn btn-primary btn-lg">Register</a>

<a href="/login" class="btn btn-success btn-lg">Login</a>

{% endif %}

</div>

{% endblock %}

Articles.html

{% extends 'layout.html' %}

{% block body %}

<h1 style="font-size: 1cm; padding-bottom: 1cm;">Articles</h1>

<ul class="list-group">

{% for article in articles %}

<li style="width: 50%" class="list-group-item"><a style="padding-right: 100px;" href="article/{{article.id}}">{{article.title}}</a></li>

{% endfor %}

</ul>

{% endblock %}