Get-it-done -- main.py

|  |
| --- |
| from flask import Flask, request, redirect, render\_template |
|  | from flask\_sqlalchemy import SQLAlchemy |
|  |  |
|  | app = Flask(\_\_name\_\_) |
|  | app.config['DEBUG'] = True |
|  | app.config['SQLALCHEMY\_DATABASE\_URI'] = 'mysql+pymysql://get-it-done:beproductive@localhost:8889/get-it-done' |
|  | app.config['SQLALCHEMY\_ECHO'] = True |
|  | db = SQLAlchemy(app) |
|  |  |
|  |  |
|  | class Task(db.Model): |
|  |  |
|  | id = db.Column(db.Integer, primary\_key=True) |
|  | name = db.Column(db.String(120)) |
|  | completed = db.Column(db.Boolean) |
|  |  |
|  | def \_\_init\_\_(self, name): |
|  | self.name = name |
|  | self.completed = False |
|  |  |
|  |  |
|  | @app.route('/', methods=['POST', 'GET']) |
|  | def index(): |
|  |  |
|  | if request.method == 'POST': |
|  | task\_name = request.form['task'] |
|  | new\_task = Task(task\_name) |
|  | db.session.add(new\_task) |
|  | db.session.commit() |
|  |  |
|  | tasks = Task.query.filter\_by(completed=False).all() |
|  | completed\_tasks = Task.query.filter\_by(completed=True).all() |
|  | return render\_template('todos.html',title="Get It Done!", |
|  | tasks=tasks, completed\_tasks=completed\_tasks) |
|  |  |
|  |  |
|  | @app.route('/delete-task', methods=['POST']) |
|  | def delete\_task(): |
|  |  |
|  | task\_id = int(request.form['task-id']) |
|  | task = Task.query.get(task\_id) |
|  | task.completed = True |
|  | db.session.add(task) |
|  | db.session.commit() |
|  |  |
|  | return redirect('/') |
|  |  |
|  |  |
|  | if \_\_name\_\_ == '\_\_main\_\_': |
|  | app.run() |

Main.py(alysson’s)

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

from flask\_sqlalchemy import SQLAlchemy

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

app.config['DEBUG'] = True

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'mysql+pymysql://get-it-done:5ch91msgrl@localhost:8889/get-it-done'

app.config['SQLALCHEMY\_ECHO'] = True

db = SQLAlchemy(app)

app.secret\_key = 'y337kGcys&zP3B'

# Note: the connection string after :// contains the following info:

# user:password@server:portNumber/databaseName

class Task(db.Model):

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

name = db.Column(db.String(120))

completed = db.Column(db.Boolean)

owner\_id = db.Column(db.Integer, db.ForeignKey('user.id'))

def \_\_init\_\_(self, name, owner):

self.name = name

self.completed = False

self.owner = owner

class User(db.Model):

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

email = db.Column(db.String(120), unique=True)

password=db.Column(db.String(120))

tasks = db.relationship('Task', backref='owner')

def \_\_init\_\_(self, email, password):

self.email = email

self.password = password

@app.before\_request

def require\_login():

allowed\_routes = ['login', 'register']

if request.endpoint not in allowed\_routes and 'email' not in session:

return redirect('/login')

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

def login():

if request.method == 'POST':

email = request.form['email']

password = request.form['password']

user = User.query.filter\_by(email=email).first()

if user and user.password == password:

session['email'] = email

flash("Logged in")

return redirect ('/')

else:

flash("User password incorrect, or user does not exist", 'error')

return render\_template('login.html')

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

def register():

if request.method == 'POST':

email = request.form['email']

password = request.form['password']

verify = request.form['verify']

existing\_user = User.query.filter\_by(email=email).first()

if not existing\_user:

new\_user = User(email, password)

db.session.add(new\_user)

db.session.commit()

session['email'] = email

return redirect('/')

else:

#TODO - user better response messaging

return '<h1> Duplicate user</h1>'

return render\_template('register.html')

@app.route('/logout')

def logout():

del session['email']

return redirect('/')

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

def index():

owner = User.query.filter\_by(email=session['email']).first()

if request.method == 'POST':

task\_name = request.form['task']

new\_task = Task(task\_name, owner)

db.session.add(new\_task)

db.session.commit()

tasks = Task.query.filter\_by(completed=False, owner=owner).all()

completed\_tasks = Task.query.filter\_by(completed=True, owner=owner).all()

return render\_template('todos.html', title='Get It Done!',

tasks=tasks, completed\_tasks=completed\_tasks)

@app.route("/delete-task", methods=['POST'])

def delete\_task():

task\_id = int(request.form['task-id'])

task = Task.query.get(task\_id)

task.completed = True

db.session.add(task)

db.session.commit()

return redirect('/')

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

app.run()

Base.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>{{title}}</title>

<style>

.error {

color: red;

}

</style>

</head>

<body>

<div>

<a href='/logout'>log out</a>

</div>

{% with messages = get\_flashed\_messages(with\_categories=True) %}

<ul>

{% for category, message in messages %}

<li class="{{category}}">{{message}}</li>

{% endfor %}

</ul>

{% endwith %}

{% block content %}

{% endblock %}

</body>

</html>

Login.html

{% extends "base.html" %}

{% block content %}

<h1>Log in to Get It Done!</h1>

<p>If you don't have an account, you may <a href="/register">register</a> for one</p>

<form action="/login" method="post">

<table>

<tr>

<td><label for="email">Email</label></td>

<td>

<input name="email" type="email" value="{{ email }}">

</td>

</tr>

<tr>

<td><label for="password">Password</label></td>

<td>

<input name="password" type="password">

</td>

</tr>

</table>

<input type="submit" value="Login">

</form>

{% endblock %}

Register.html

{% extends "base.html" %}

{% block content %}

<h1>Register to Get It Done!</h1>

<form action="/register" method="post">

<table>

<tr>

<td><label for="email">Email</label></td>

<td>

<input name="email" type="email" value="{{ email }}">

</td>

</tr>

<tr>

<td><label for="password">Password</label></td>

<td>

<input name="password" type="password">

</td>

</tr>

<tr>

<td><label for="verify">Verify Password</label></td>

<td>

<input name="verify" type="password">

</td>

</tr>

</table>

<input type="submit" value="Register">

</form>

{% endblock %}

Todos.html

{% extends "base.html" %}

{% block content %}

<h1>Get It Done!</h1>

<form method="post">

<label>

New Task:

<input type="text" name="task" />

</label>

<input type="submit" value="Add Task" />

</form>

<hr />

{% if tasks|length == 0 %}

<p>No tasks yet</p>

{% else %}

<ul>

{% for task in tasks %}

<li>{{task.name}}

<form method="POST" action="/delete-task" style="display:inline-block;">

<input type="hidden" name="task-id" value="{{task.id}}" />

<input type="submit" value="Done!" />

</form>

</li>

{% endfor %}

</ul>

{% endif %}

<hr />

<h2>

Completed

</h2>

<ul>

{% for task in completed\_tasks %}

<li>

{{task.name}}

</li>

{% endfor %}

</ul>

{% endblock %}