**pcto.py**

from app import app, db

from app.models import User, Post

@app.shell\_context\_processor

def make\_shell\_context():

return {'db': db, 'User': User, 'Post': Post}

- - -

**config.py**

import os

basedir = os.path.abspath(os.path.dirname(\_\_file\_\_))

class Config(object):

SECRET\_KEY = 'you-will-never-guess'

SQLALCHEMY\_DATABASE\_URI = 'sqlite:///' + os.path.join(basedir, 'app.db')

SQLALCHEMY\_DATABASE\_URI = 'postgresql://pcto:password@192.168.0.202:5432/pcto'

SQLALCHEMY\_TRACK\_MODIFICATIONS = False

DBMS\_USERNAME = 'postgres'

DBMS\_PASSWORD = 'password'

DBMS\_HOST = '192.168.0.202'

DBMS\_DB = 'pcto'

- - -

**app/\_\_init.py\_\_**

from flask import Flask

from flask\_sqlalchemy import SQLAlchemy

from flask\_migrate import Migrate

from flask\_login import LoginManager

from config import Config

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

app.config.from\_object(Config)

db = SQLAlchemy(app)

migrate = Migrate(app, db)

login = LoginManager(app)

login.login\_view = 'login'

from app import routes, models

- - -

**app/forms.py**

from flask\_wtf import FlaskForm

from wtforms import StringField, PasswordField, BooleanField, SubmitField, \

TextAreaField

from wtforms.validators import ValidationError, DataRequired, Email, EqualTo, \

Length

from app.models import User

**class LoginForm(FlaskForm):**

username = StringField('Username', validators=[DataRequired()])

password = PasswordField('Password', validators=[DataRequired()])

remember\_me = BooleanField('Remember Me')

submit = SubmitField('Sign In')

**class RegistrationForm(FlaskForm):**

username = StringField('Username', validators=[DataRequired()])

email = StringField('Email', validators=[DataRequired(), Email()])

password = PasswordField('Password', validators=[DataRequired()])

password2 = PasswordField(

'Repeat Password', validators=[DataRequired(), EqualTo('password')])

submit = SubmitField('Register')

def validate\_username(self, username):

user = User.query.filter\_by(username=username.data).first()

if user is not None:

raise ValidationError('Please use a different username.')

def validate\_email(self, email):

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

if user is not None:

raise ValidationError('Please use a different email address.')

**class EditProfileForm(FlaskForm):**

username = StringField('Username', validators=[DataRequired()])

about\_me = TextAreaField('About me', validators=[Length(min=0, max=140)])

submit = SubmitField('Submit')

- - -

**app/models.py**

from datetime import datetime

from hashlib import md5

from app import db, login

from flask\_login import UserMixin

from werkzeug.security import generate\_password\_hash, check\_password\_hash

**class User(UserMixin, db.Model):**

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

username = db.Column(db.String(64), index=True, unique=True)

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

password\_hash = db.Column(db.String(128))

posts = db.relationship('Post', backref='author', lazy='dynamic')

about\_me = db.Column(db.String(140))

last\_seen = db.Column(db.DateTime, default=datetime.utcnow)

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

return '<User {}>'.format(self.username)

def set\_password(self, password):

self.password\_hash = generate\_password\_hash(password)

def check\_password(self, password):

return check\_password\_hash(self.password\_hash, password)

def avatar(self, size):

digest = md5(self.email.lower().encode('utf-8')).hexdigest()

return 'https://www.gravatar.com/avatar/{}?d=identicon&s={}'.format(

digest, size)

**@login.user\_loader**

**def load\_user(id):**

return User.query.get(int(id))

**class Post(db.Model):**

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

body = db.Column(db.String(140))

timestamp = db.Column(db.DateTime, index=True, default=datetime.utcnow)

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

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

return '<Post {}>'.format(self.body)

- - -

**app/routes.py**

from datetime import datetime

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

from flask\_login import login\_user, logout\_user, current\_user, login\_required

from werkzeug.urls import url\_parse

from app import app, db

from app.forms import LoginForm, RegistrationForm, EditProfileForm

from app.models import User

**@app.before\_request**

**def before\_request():**

if current\_user.is\_authenticated:

current\_user.last\_seen = datetime.utcnow()

db.session.commit()

**@app.route('/')**

**@app.route('/index')**

**@login\_required**

**def index():**

posts = [

{

'author': {'username': 'John'},

'body': 'Beautiful day in Portland!'

},

{

'author': {'username': 'Susan'},

'body': 'The Avengers movie was so cool!'

}

]

return render\_template('index.html', title='Home', posts=posts)

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

**def login():**

if current\_user.is\_authenticated:

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

form = LoginForm()

if form.validate\_on\_submit():

user = User.query.filter\_by(username=form.username.data).first()

if user is None or not user.check\_password(form.password.data):

flash('Invalid username or password')

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

login\_user(user, remember=form.remember\_me.data)

next\_page = request.args.get('next')

if not next\_page or url\_parse(next\_page).netloc != '':

next\_page = url\_for('index')

return redirect(next\_page)

return render\_template('login.html', title='Sign In', form=form)

**@app.route('/logout')**

**def logout():**

logout\_user()

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

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

**def register():**

if current\_user.is\_authenticated:

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

form = RegistrationForm()

if form.validate\_on\_submit():

user = User(username=form.username.data, email=form.email.data)

user.set\_password(form.password.data)

db.session.add(user)

db.session.commit()

flash('Congratulations, you are now a registered user!')

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

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

**@app.route('/user/<username>')**

**@login\_required**

**def user(username):**

user = User.query.filter\_by(username=username).first\_or\_404()

posts = [

{'author': user, 'body': 'Test post #1'},

{'author': user, 'body': 'Test post #2'}

]

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

**@app.route('/edit\_profile', methods=['GET', 'POST'])**

**@login\_required**

**def edit\_profile():**

form = EditProfileForm()

if form.validate\_on\_submit():

current\_user.username = form.username.data

current\_user.about\_me = form.about\_me.data

db.session.commit()

flash('Your changes have been saved.')

return redirect(url\_for('edit\_profile'))

elif request.method == 'GET':

form.username.data = current\_user.username

form.about\_me.data = current\_user.about\_me

return render\_template('edit\_profile.html', title='Edit Profile',

form=form)

- - -

**apps/templates/base.html**

<!doctype html>

<html>

<head>

{% if title %}

<title>{{ title }} - Microblog</title>

{% else %}

<title>Welcome to Microblog</title>

{% endif %}

</head>

<body>

<div>

Microblog:

<a href="{{ url\_for('index') }}">Home</a>

{% if current\_user.is\_anonymous %}

<a href="{{ url\_for('login') }}">Login</a>

{% else %}

<a href="{{ url\_for('user', username=current\_user.username) }}">Profile</a>

<a href="{{ url\_for('logout') }}">Logout</a>

{% endif %}

</div>

<hr>

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

{% if messages %}

<ul>

{% for message in messages %}

<li>{{ message }}</li>

{% endfor %}

</ul>

{% endif %}

{% endwith %}

{% block content %}{% endblock %}

</body>

</html>

- - -

**apps/templates/edit\_profile.html**

{% extends "base.html" %}

{% block content %}

<h1>Edit Profile</h1>

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

{{ form.hidden\_tag() }}

<p>

{{ form.username.label }}<br>

{{ form.username(size=32) }}<br>

{% for error in form.username.errors %}

<span style="color: red;">[{{ error }}]</span>

{% endfor %}

</p>

<p>

{{ form.about\_me.label }}<br>

{{ form.about\_me(cols=50, rows=4) }}<br>

{% for error in form.about\_me.errors %}

<span style="color: red;">[{{ error }}]</span>

{% endfor %}

</p>

<p>{{ form.submit() }}</p>

</form>

{% endblock %}

- - -

**apps/templates/index.html**

{% extends "base.html" %}

{% block content %}

<h1>Hi, {{ current\_user.username }}!</h1>

{% for post in posts %}

<div><p>{{ post.author.username }} says: <b>{{ post.body }}</b></p></div>

{% endfor %}

{% endblock %}

- - -

**apps/templates/login.html**

{% extends "base.html" %}

{% block content %}

<h1>Sign In</h1>

<form action="" method="post" novalidate>

{{ form.hidden\_tag() }}

<p>

{{ form.username.label }}<br>

{{ form.username(size=32) }}<br>

{% for error in form.username.errors %}

<span style="color: red;">[{{ error }}]</span>

{% endfor %}

</p>

<p>

{{ form.password.label }}<br>

{{ form.password(size=32) }}<br>

{% for error in form.password.errors %}

<span style="color: red;">[{{ error }}]</span>

{% endfor %}

</p>

<p>{{ form.remember\_me() }} {{ form.remember\_me.label }}</p>

<p>{{ form.submit() }}</p>

</form>

<p>New User? <a href="{{ url\_for('register') }}">Click to Register!</a></p>

{% endblock %}

- - -

**apps/templates/\_post.html**

<table>

<tr valign="top">

<td><img src="{{ post.author.avatar(36) }}"></td>

<td>{{ post.author.username }} says:<br>{{ post.body }}</td>

</tr>

</table>

- - -

**apps/templates/register.html**

{% extends "base.html" %}

{% block content %}

<h1>Register</h1>

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

{{ form.hidden\_tag() }}

<p>

{{ form.username.label }}<br>

{{ form.username(size=32) }}<br>

{% for error in form.username.errors %}

<span style="color: red;">[{{ error }}]</span>

{% endfor %}

</p>

<p>

{{ form.email.label }}<br>

{{ form.email(size=64) }}<br>

{% for error in form.email.errors %}

<span style="color: red;">[{{ error }}]</span>

{% endfor %}

</p>

<p>

{{ form.password.label }}<br>

{{ form.password(size=32) }}<br>

{% for error in form.password.errors %}

<span style="color: red;">[{{ error }}]</span>

{% endfor %}

</p>

<p>

{{ form.password2.label }}<br>

{{ form.password2(size=32) }}<br>

{% for error in form.password2.errors %}

<span style="color: red;">[{{ error }}]</span>

{% endfor %}

</p>

<p>{{ form.submit() }}</p>

</form>

{% endblock %}

- - -

**apps/templates/user.html**

{% extends "base.html" %}

{% block content %}

<table>

<tr valign="top">

<td><img src="{{ user.avatar(128) }}"></td>

<td>

<h1>User: {{ user.username }}</h1>

{% if user.about\_me %}<p>{{ user.about\_me }}</p>{% endif %}

{% if user.last\_seen %}<p>Last seen on: {{ user.last\_seen }}</p>{% endif %}

{% if user == current\_user %}

<p><a href="{{ url\_for('edit\_profile') }}">Edit your profile</a></p>

{% endif %}

</td>

</tr>

</table>

<hr>

{% for post in posts %}

{% include '\_post.html' %}

{% endfor %}

{% endblock %}