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

from app import app, db

from app.models import Acao, Responsavel, Origem, Superintendencia, Usuario

from datetime import date, datetime

from werkzeug.security import generate\_password\_hash

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

def login():

    if request.method == 'POST':

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

    return render\_template('login.html')

@app.route('/')

def index():

    hoje = date.today()

    acoes = Acao.query.all()

    for acao in acoes:

        if acao.status != "Concluída" and acao.prazo < hoje and acao.status != "Atrasada":

            acao.status = "Atrasada"

            db.session.commit()

    filtro\_responsavel = request.args.get('responsavel')

    filtro\_superintendencia = request.args.get('superintendencia')

    filtro\_origem = request.args.get('origem')

    filtro\_status = request.args.get('status')

    data\_inicio = request.args.get('data\_inicio')

    data\_fim = request.args.get('data\_fim')

    query = Acao.query

    if filtro\_responsavel:

        query = query.filter\_by(responsavel\_id=filtro\_responsavel)

    if filtro\_superintendencia:

        query = query.join(Responsavel).filter(Responsavel.superintendencia\_id == filtro\_superintendencia)

    if filtro\_origem:

        query = query.filter\_by(origem\_id=filtro\_origem)

    if filtro\_status:

        query = query.filter\_by(status=filtro\_status)

    if data\_inicio:

        query = query.filter(Acao.prazo >= datetime.strptime(data\_inicio, '%Y-%m-%d').date())

    if data\_fim:

        query = query.filter(Acao.prazo <= datetime.strptime(data\_fim, '%Y-%m-%d').date())

    acoes\_filtradas = query.all()

    responsaveis = Responsavel.query.filter\_by(ativo=True).all()

    superintendencias = Superintendencia.query.filter\_by(ativo=True).all()

    origens = Origem.query.filter\_by(ativo=True).all()

    status\_opcoes = ["Não iniciada", "Em andamento", "Atrasada", "Concluída"]

    responsaveis\_info = [

        {"id": r.id, "nome": r.nome, "superintendencia\_id": r.superintendencia\_id}

        for r in responsaveis

    ]

    return render\_template(

        'index.html',

        acoes=acoes\_filtradas,

        responsaveis=responsaveis,

        superintendencias=superintendencias,

        origens=origens,

        status\_opcoes=status\_opcoes,

        responsaveis\_info=responsaveis\_info

    )

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

def cadastrar():

    if request.method == 'POST':

        origem\_id = request.form['origem']

        descricao = request.form['descricao']

        responsavel\_id = request.form['responsavel']

        prazo\_str = request.form['prazo']

        status = request.form['status']

        prazo = datetime.strptime(prazo\_str, '%Y-%m-%d').date()

        nova\_acao = Acao(

            origem\_id=origem\_id,

            descricao=descricao,

            responsavel\_id=responsavel\_id,

            prazo=prazo,

            status=status

        )

        db.session.add(nova\_acao)

        db.session.commit()

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

    responsaveis = Responsavel.query.filter\_by(ativo=True).all()

    origens = Origem.query.filter\_by(ativo=True).all()

    return render\_template('cadastrar.html', responsaveis=responsaveis, origens=origens)

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

def editar(id):

    acao = Acao.query.get\_or\_404(id)

    if request.method == 'POST':

        acao.origem\_id = request.form['origem']

        acao.descricao = request.form['descricao']

        acao.responsavel\_id = request.form['responsavel']

        acao.prazo = datetime.strptime(request.form['prazo'], '%Y-%m-%d').date()

        acao.status = request.form['status']

        if acao.status == 'Atrasada' and acao.prazo >= date.today():

            acao.status = 'Em andamento'

        db.session.commit()

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

    responsaveis = Responsavel.query.all()

    origens = Origem.query.all()

    return render\_template('editar.html', acao=acao, responsaveis=responsaveis, origens=origens)

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

def responsaveis():

    if request.method == 'POST':

        novo = Responsavel(

            nome=request.form['nome'],

            email=request.form['email'],

            telefone=request.form['telefone'],

            funcao=request.form['funcao'],

            superintendencia\_id=request.form['superintendencia\_id'],

            unidade=request.form['unidade'],

            tipo\_acesso=request.form['tipo\_acesso']

        )

        db.session.add(novo)

        db.session.commit()

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

    todos = Responsavel.query.all()

    superintendencias = Superintendencia.query.filter\_by(ativo=True).all()

    return render\_template('responsaveis.html', responsaveis=todos, superintendencias=superintendencias)

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

def editar\_responsavel(id):

    responsavel = Responsavel.query.get\_or\_404(id)

    superintendencias = Superintendencia.query.filter\_by(ativo=True).all()

    if request.method == 'POST':

        responsavel.nome = request.form['nome']

        responsavel.email = request.form['email']

        responsavel.telefone = request.form['telefone']

        responsavel.funcao = request.form['funcao']

        responsavel.superintendencia\_id = request.form['superintendencia\_id']

        responsavel.unidade = request.form['unidade']

        responsavel.tipo\_acesso = request.form['tipo\_acesso']

        db.session.commit()

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

    return render\_template('editar\_responsavel.html', responsavel=responsavel, superintendencias=superintendencias)

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

def origens():

    if request.method == 'POST':

        nova = Origem(nome=request.form['nome'])

        db.session.add(nova)

        db.session.commit()

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

    todas = Origem.query.all()

    return render\_template('origens.html', origens=todas)

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

def superintendencias():

    if request.method == 'POST':

        nova = Superintendencia(nome=request.form['nome'])

        db.session.add(nova)

        db.session.commit()

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

    todas = Superintendencia.query.all()

    return render\_template('superintendencias.html', superintendencias=todas)

@app.route('/responsavel/desativar/<int:id>', methods=['POST'])

def desativar\_responsavel(id):

    responsavel = Responsavel.query.get\_or\_404(id)

    responsavel.ativo = False

    db.session.commit()

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

@app.route('/responsaveis/ativar/<int:id>', methods=['POST'])

def ativar\_responsavel(id):

    responsavel = Responsavel.query.get\_or\_404(id)

    responsavel.ativo = True

    db.session.commit()

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

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

def editar\_origem(id):

    origem = Origem.query.get\_or\_404(id)

    if request.method == 'POST':

        origem.nome = request.form['nome']

        db.session.commit()

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

    return render\_template('editar\_origem.html', origem=origem)

@app.route('/origens/desativar/<int:id>', methods=['POST'])

def desativar\_origem(id):

    origem = Origem.query.get\_or\_404(id)

    origem.ativo = False

    db.session.commit()

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

@app.route('/origens/ativar/<int:id>', methods=['POST'])

def ativar\_origem(id):

    origem = Origem.query.get\_or\_404(id)

    origem.ativo = True

    db.session.commit()

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

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

def editar\_superintendencia(id):

    superintendencia = Superintendencia.query.get\_or\_404(id)

    if request.method == 'POST':

        superintendencia.nome = request.form['nome']

        db.session.commit()

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

    return render\_template('editar\_superintendencia.html', superintendencia=superintendencia)

@app.route('/desativar\_superintendencia/<int:id>', methods=['POST'])

def desativar\_superintendencia(id):

    superintendencia = Superintendencia.query.get\_or\_404(id)

    superintendencia.ativo = False

    db.session.commit()

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

@app.route('/ativar\_superintendencia/<int:id>', methods=['POST'])

def ativar\_superintendencia(id):

    superintendencia = Superintendencia.query.get\_or\_404(id)

    superintendencia.ativo = True

    db.session.commit()

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

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

def esqueci\_senha():

    if request.method == 'POST':

        email = request.form['email']

        # Aqui você pode adicionar lógica para envio de e-mail de recuperação, se desejar

        return render\_template('esqueci\_senha.html', mensagem="Se o e-mail estiver cadastrado, enviaremos as instruções.")

    return render\_template('esqueci\_senha.html')

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

def usuarios():

    if request.method == 'POST':

        email = request.form['email']

        if Usuario.query.filter\_by(email=email).first():

            flash('Este e-mail já está cadastrado. Escolha outro.', 'danger')

            usuarios = Usuario.query.all()

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

        novo = Usuario(

            nome=request.form['nome'],

            email=email,

            acesso\_responsaveis='acesso\_responsaveis' in request.form,

            editar\_responsaveis='editar\_responsaveis' in request.form,

            acesso\_origens='acesso\_origens' in request.form,

            editar\_origens='editar\_origens' in request.form,

            acesso\_superintendencias='acesso\_superintendencias' in request.form,

            editar\_superintendencias='editar\_superintendencias' in request.form,

            acesso\_acoes='acesso\_acoes' in request.form,

            editar\_acoes='editar\_acoes' in request.form,

        )

        novo.set\_senha(request.form['senha'])

        db.session.add(novo)

        db.session.commit()

        flash('Usuário cadastrado com sucesso!', 'success')

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

    usuarios = Usuario.query.all()

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

from app.models import Usuario  # certifique-se de ter importado

from flask import flash

from werkzeug.security import generate\_password\_hash

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

def editar\_usuario(id):

    usuario = Usuario.query.get\_or\_404(id)

    if request.method == 'POST':

        usuario.nome = request.form['nome']

        usuario.email = request.form['email']

        usuario.acesso\_responsaveis = 'acesso\_responsaveis' in request.form

        usuario.editar\_responsaveis = 'editar\_responsaveis' in request.form

        usuario.acesso\_origens = 'acesso\_origens' in request.form

        usuario.editar\_origens = 'editar\_origens' in request.form

        usuario.acesso\_superintendencias = 'acesso\_superintendencias' in request.form

        usuario.editar\_superintendencias = 'editar\_superintendencias' in request.form

        usuario.acesso\_acoes = 'acesso\_acoes' in request.form

        usuario.editar\_acoes = 'editar\_acoes' in request.form

        usuario.ativo = 'ativo' in request.form

        nova\_senha = request.form.get('nova\_senha')

        if nova\_senha:

            usuario.senha\_hash = generate\_password\_hash(nova\_senha)

        db.session.commit()

        flash('Usuário atualizado com sucesso!', 'success')

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

    return render\_template('editar\_usuario.html', usuario=usuario)