app\_imporved.py

|  |
| --- |
| #!/usr/bin/env python3  # Foundations of Python Network Programming, Third Edition  # https://github.com/brandon-rhodes/fopnp/blob/m/py3/chapter11/app\_improved.py  # A payments application with basic security improvements added.  import bank, uuid  from OpenSSL import SSL  #import ssl  from flask import (Flask, abort, flash, get\_flashed\_messages,  redirect, render\_template, request, session, url\_for)  app = Flask(\_\_name\_\_)  app.secret\_key = 'saiGeij8AiS2ahleahMo5dahveixuV3J'  #allaccount = [('brandon', 'atigdng'), ('sam', 'xyzzy')]  allaccount = [{'username':'brandon','password':'atigdng'}, {'username':'sam','password':'xyzzy'}, {'username':'root','password':'root'}]  @app.route('/root/<id>' , methods=['GET', 'POST'])  @app.route('/root' , methods=['GET', 'POST'])  def root(id=None):  if id:  print(id)  print(allaccount)  index = next(index for (index, d) in enumerate(allaccount) if d['username']== id)  del allaccount[index]  username = session.get('username')  if not username:  return redirect(url\_for('login'))  print(allaccount)  removeroot = allaccount  aa = next(index for (index, d) in enumerate(allaccount) if d['username']== 'root')  del removeroot[aa]  return render\_template('root.html', allaccount=allaccount)  @app.route('/login', methods=['GET', 'POST'])  def login():  username = request.form.get('username', '')  password = request.form.get('password', '')  if request.method == 'POST':  if {'username': username, 'password': password} in allaccount:  session['username'] = username  session['csrf\_token'] = uuid.uuid4().hex  return redirect(url\_for('index'))  return render\_template('login.html', username=username)  @app.route('/member' , methods=['GET', 'POST'])  def member():  username = str(request.form.get('username', '').strip())  password = str(request.form.get('password', '').strip())  complaint = None  if request.method == 'POST':  if [item for item in allaccount if item['username'] == username]:  complaint = 'member is exist'  elif not (password and username) :  complaint = 'fill all field'  else:  allaccount.append({'username':username,'password':password})  return redirect(url\_for('login'))  return render\_template('member2.html', complaint=complaint, username=username,  password=password)    @app.route('/logout')  def logout():  session.pop('username', None)  return redirect(url\_for('login'))  @app.route('/')  def index():  username = session.get('username')  if username == 'root':  root=True  else:  root=None  if not username:  return redirect(url\_for('login'))  payments = bank.get\_payments\_of(bank.open\_database(), username)  return render\_template('index.html',root=root, payments=payments, username=username,  flash\_messages=get\_flashed\_messages())  @app.route('/pay', methods=['GET', 'POST'])  def pay():  username = session.get('username')  if not username:  return redirect(url\_for('login'))  account = request.form.get('account', '').strip()  dollars = request.form.get('dollars', '').strip()  memo = request.form.get('memo', '').strip()  complaint = None  if request.method == 'POST':  person = [item for item in allaccount if item['username'] == account]  if request.form.get('csrf\_token') != session['csrf\_token']:  abort(403)  if account and dollars and dollars.isdigit() and memo and person:  db = bank.open\_database()  bank.add\_payment(db, username, account, dollars, memo)  db.commit()  flash('Payment successful')  return redirect(url\_for('index'))  #complaint = ('Dollars must be an integer' if not dollars.isdigit()  # else 'Please fill in all three fields')  if not person:  complaint = 'user is not exist'  elif dollars.isdigit():  complaint = 'Dollars must be an integer'  else:  complaint = 'Please fill in all three fields'  return render\_template('pay2.html', complaint=complaint, account=account,  dollars=dollars, memo=memo,  csrf\_token=session['csrf\_token'])  if \_\_name\_\_ == '\_\_main\_\_':  context = ('localhost.crt', 'localhost.key')  #app.debug = True  app.run('127.0.0.1', debug=True, port=8100, ssl\_context=context, threaded=True) |

bank.py

|  |
| --- |
| #!/usr/bin/env python3  # Foundations of Python Network Programming, Third Edition  # https://github.com/brandon-rhodes/fopnp/blob/m/py3/chapter11/bank.py  # A small library of database routines to power a payments application.  import os, pprint, sqlite3  from collections import namedtuple  def open\_database(path='bank.db'):  new = not os.path.exists(path)  db = sqlite3.connect(path)  if new:  c = db.cursor()  c.execute('CREATE TABLE payment (id INTEGER PRIMARY KEY,'  ' debit TEXT, credit TEXT, dollars INTEGER, memo TEXT)')  add\_payment(db, 'brandon', 'psf', 125, 'Registration for PyCon')  add\_payment(db, 'brandon', 'liz', 200, 'Payment for writing that code')  add\_payment(db, 'sam', 'brandon', 25, 'Gas money-thanks for the ride!')  db.commit()  return db  def add\_payment(db, debit, credit, dollars, memo):  db.cursor().execute('INSERT INTO payment (debit, credit, dollars, memo)'  ' VALUES (?, ?, ?, ?)', (debit, credit, dollars, memo))  def get\_payments\_of(db, account):  c = db.cursor()  c.execute('SELECT \* FROM payment WHERE credit = ? or debit = ?'  ' ORDER BY id', (account, account))  Row = namedtuple('Row', [tup[0] for tup in c.description])  return [Row(\*row) for row in c.fetchall()]  if \_\_name\_\_ == '\_\_main\_\_':  db = open\_database()  pprint.pprint(get\_payments\_of(db, 'brandon')) |

Template/base.html

|  |
| --- |
| <html>  <head>  <title>{% block title %}{% endblock %}</title>  <link rel="stylesheet" type="text/css" href="/static/style.css">  </head>  <body>  <h1>{{ self.title() }}</h1>  {% block body %}{% endblock %}  </body>  </html> |

Template/index.html

|  |
| --- |
| {% extends "base.html" %}  {% block title %}Welcome, {{ username }}{% endblock %}  {% block body %}  {% for message in flash\_messages %}  <div class="flash\_message">{{ message }}<a href="/">&times;</a></div>  {% endfor %}  <p>Your Payments</p>  <ul>  {% if root %}<a href='/root'>admin</a>{% endif %}  {% for p in payments %}  {% set prep = 'from' if (p.credit == username) else 'to' %}  {% set acct = p.debit if (p.credit == username) else p.credit %}  <li class="{{ prep }}">{{p.id}} ${{ p.dollars }} {{ prep }} <b>{{ acct }}</b>  for: <i>{{ p.memo }}</i></li>  {% endfor %}  </ul>  <a href="/pay">Make payment</a> | <a href="/logout">Log out</a>  {% endblock %} |

Template/login.html

|  |
| --- |
| {% extends "base.html" %}  {% block title %}Please log in{% endblock %}  {% block body %}  <form method="post">  <label>User: <input name="username" value="{{ username }}"></label>  <label>Password: <input name="password" type="password"></label>  <button type="submit">Log in</button><a href="/member">Register</a>  </form>  {% endblock %} |

Template/member.html

|  |
| --- |
| {% extends "base.html" %}  {% block title %}Register{% endblock %}  {% block body %}  <form method="post" action="/member">  {% if complaint %}<span class="complaint">{{ complaint }}</span>{% endif %}  <label>username: <input name="username" value="{{ username }}"></label>  <label>password: <input name="password" value="{{ password }}" type="password"></label>  <button type="submit">sign in</button> | <a href="/">Cancel</a>  </form>  {% endblock %} |

Template/member2.html

|  |
| --- |
| {% extends "base.html" %}  {% block title %}Register{% endblock %}  {% block body %}  <form method="post" action="/member">  {% if complaint %}<span class="complaint">{{ complaint }}</span>{% endif %}  <label>username: <input name="username" value="{{ username }}"></label>  <label>password: <input name="password" value="{{ password }}" type="password"></label>  <button type="submit">sign in</button> | <a href="/login">Cancel</a>  </form>  {% endblock %} |

Template/pay.html

|  |
| --- |
| {% extends "base.html" %}  {% block title %}Make a Payment{% endblock %}  {% block body %}  <form method="post" action="/pay">  {% if complaint %}<span class="complaint">{{ complaint }}</span>{% endif %}  <label>To account: <input name="account" value="{{ account }}"></label>  <label>Dollars: <input name="dollars" value="{{ dollars }}"></label>  <label>Memo: <input name="memo" value="{{ memo }}"></label>  <button type="submit">Send money</button> | <a href="/">Cancel</a>  </form>  {% endblock %} |

Template/pay2.html

|  |
| --- |
| {% extends "base.html" %}  {% block title %}Make a Payment{% endblock %}  {% block body %}  <form method="post" action="/pay">  {% if complaint %}<span class="complaint">{{ complaint }}</span>{% endif %}  <label>To account: <input name="account" value="{{ account }}"></label>  <label>Dollars: <input name="dollars" value="{{ dollars }}"></label>  <label>Memo: <input name="memo" value="{{ memo }}"></label>  <input name="csrf\_token" type="hidden" value="{{ csrf\_token }}">  <button type="submit">Send money</button> | <a href="/">Cancel</a>  </form>  {% endblock %} |

Template/root.html

|  |
| --- |
| {% extends "base.html" %}  {% block title %}Welcome, {{ username }}{% endblock %}  {% block body %}  <p>Member</p>  <ul>  {% for p in allaccount %}  <li>username: {{p.username}} password:{{p.password}}</li><a href="/root/{{ p.username }}">remove<a>  {% endfor %}  </ul>  <a href="/">Back</a>  {% endblock %} |

https://github.com/emilymemoryg/HW3