**БИН2003**

**Киселев Михаил**

Account.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

    <form action="" method="post">

        {% if full\_name %}

        <p>Hello, {{full\_name}}! </p>

        {% endif %}

        </p>

    </form>

</head>

<body>

</body>

</html>

Login.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Login</title>

</head>

<body>

    <form action="" method="post">

        <p>

     <label for="username">Username</label>

     <input type="text" name="username">

 </p>

 <p>

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

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

 </p>

 <p>

     <input type="submit" value="login" name="login">

  <input type="submit" value="registration" name="registration">

 </p>

 </form>

</body>

</html>

Registration.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Registration</title>

</head>

<body>

    <form action="" method="post">

      <p>

    <label for="name">Your name:</label>

    <input type="text" name="name">

     </p>

        <p>

        <label for="login">Login:</label>

        <input type="text" name="login">

         </p>

    <p>

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

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

    </p>

    <p>

        <input type="submit" value="Sign In">

    </p>

    </form>

</body>

</html>

App.py

import psycopg2

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

from config import DATABASE, USER, PASSWORD

app = Flask(\_\_name\_\_)

conn = psycopg2.connect(database=DATABASE,

                        user=USER,

                        password=PASSWORD,

                        host='localhost',

                        port='5432')

cursor = conn.cursor()

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

def index():

    return render\_template('login.html')

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

def login():

    if request.method == 'POST':

        if request.form.get("login"):

            username = request.form.get('username')

            password = request.form.get('password')

            cursor.execute("SELECT \* FROM users WHERE login=%s AND password=%s",

                            (str(username), str(password)))

            records = list(cursor.fetchall())

            if not (username and password):

                    return "Вы ничего не ввели"

            elif not password:

                return "Вы не ввели пароль"

            elif not login:

                return "Вы не ввели логин"

            else:

                if not records:

                    return ('Неверное имя пользователя или пароль')

                else:

                    return render\_template('account.html', full\_name=records[0][1])

        elif request.form.get("registration"):

            return redirect("/registration/")

    return render\_template('login.html')

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

def registration():

    if request.method == 'POST':

        name = request.form.get('name')

        if not name.replace(' ', '').isalpha():

            return 'В имени должны быть только буквы'

        login = request.form.get('login')

        password = request.form.get('password')

        if not password or not login:

            return 'Вы не ввели логин или пароль'

        cursor.execute("SELECT \* FROM users WHERE login=%s",

                        (str(login),))

        if not cursor.fetchone():

            cursor.execute('INSERT INTO users (fullname, login, password) VALUES (%s, %s, %s);',

                        (str(name), str(login), str(password)))

            conn.commit()

        else:

            return "Вы уже зарегистрированы"

        return redirect('/login/')

    return render\_template('registration.html')

Config.py

from starlette.config import Config

config = Config(".env")

DATABASE: str = config("DATABASE")

USER: str = config("USER")

PASSWORD: str = config("PASSWORD")

Результат работы:



