

**Министерство цифрового развития, связи и массовых коммуникаций
Российской Федерации**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное
образовательное учреждение высшего образования**

«Московский технический университет связи и информатики»

Кафедра математическая кибернетика и информационные технологии

Отчёт по лабораторной работе №4

«»

Выполнил: студент группы
БИН2001 Фадеев А. М.
Проверил: Аршинов Е. А.

Создадим необходимое окружение в БД Postgres согласно схеме (sources/schema.sql)

```
1 DROP TABLE IF EXISTS posts;
2
3 CREATE TABLE users (
4     id SERIAL NOT NULL,
5     full_name VARCHAR NOT NULL,
6     login VARCHAR NOT NULL,
7     password VARCHAR NOT NULL
8 );|
```

Настроим виртуальное окружение. Подключим необходимые модули.

```
1 import psycopg2
2 import requests
3 from flask import Flask, render_template, request, redirect
```

Объявим подключение к базе данных. В данном случае используется удалённая БД на личном сервере.

```
5 conn = psycopg2.connect(database="flask_app",
6                           user=" ",
7                           password=" ",
8                           host=" ",
9                           port="5432")
```

Обрабатываем GET/POST запросы на различные пути веб приложения

```

16 @app.route('/login/', methods=['POST', 'GET'])
17 def login():
18     if request.method == 'POST':
19         if request.form.get("login"):
20             username = request.form.get('username')
21             password = request.form.get('password')
22             if len(username) == 0 or len(password) == 0:
23                 return render_template('error.html', err_text='Логин или пароль не могут быть пустыми')
24             cursor.execute("SELECT * FROM users WHERE login=%s AND password=%s", (str(username), str(password)))
25             records = list(cursor.fetchall())
26             if len(records) == 0:
27                 return render_template('error.html', err_text='Пользователь не найден')
28             else:
29                 return render_template('account.html', full_name=records[0][1])
30         elif request.form.get("registration"):
31             return redirect("/registration/")
32     return render_template('login.html')
33
34 @app.route('/registration/', methods=['POST', 'GET'])
35 def registration():
36     if request.method == 'POST':
37         name = request.form.get('name')
38         login = request.form.get('login')
39         password = request.form.get('password')
40         if len(name) == 0 or len(login) == 0 or len(password) == 0:
41             return render_template('error.html', err_text='Пустые поля недопустимы')
42         cursor.execute("SELECT login FROM users WHERE login=%s", (str(login),))
43         records = list(cursor.fetchall())
44         if len(records) != 0:
45             return render_template('error.html', err_text='Имя ' + str(login) + ' занято')
46         else:
47             cursor.execute('INSERT INTO users (full_name, login, password) VALUES (%s, %s, %s);',
48                             (str(name), str(login), str(password)))
49             conn.commit()
50             return redirect('/login/')
51     return render_template('registration.html')

```

Проверка на правильность введенной пары логин/пароль

```

username = request.form.get('username')
password = request.form.get('password')
if len(username) == 0 or len(password) == 0:
    return render_template('error.html', err_text='Логин или пароль не могут быть пустыми')
cursor.execute("SELECT * FROM users WHERE login=%s AND password=%s", (str(username), str(password)))
records = list(cursor.fetchall())
if len(records) == 0:
    return render_template('error.html', err_text='Пользователь не найден')
else:
    return render_template('account.html', full_name=records[0][1])

```

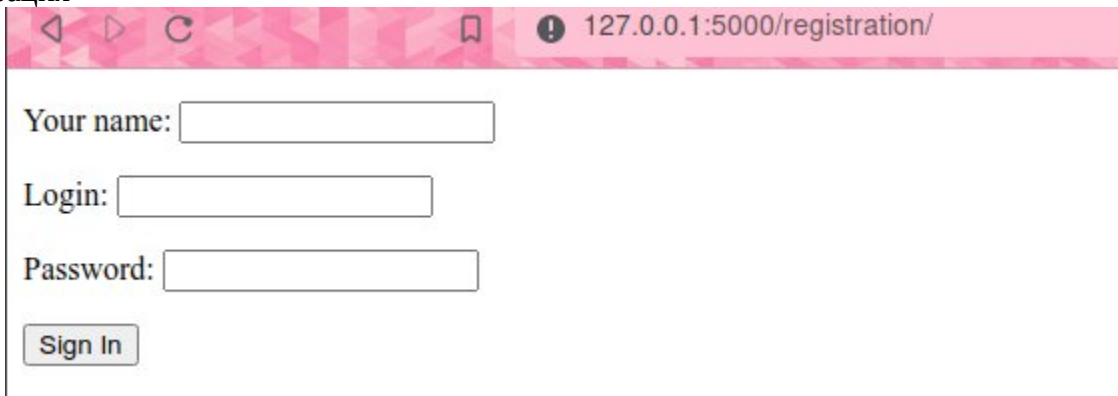
Проверка на существование пользователя в системе и вывод ошибки или информации об успешной регистрации

```

name = request.form.get('name')
login = request.form.get('login')
password = request.form.get('password')
if len(name) == 0 or len(login) == 0 or len(password) == 0:
    return render_template('error.html', err_text='Пустые поля недопустимы')
cursor.execute("SELECT login FROM users WHERE login=%s", (str(login),))
records = list(cursor.fetchall())
if len(records) != 0:
    return render_template('error.html', err_text='Имя ' + str(login) + ' занято')
else:
    cursor.execute('INSERT INTO users (full_name, login, password) VALUES (%s, %s, %s);',
                    (str(name), str(login), str(password)))
    conn.commit()
    return redirect('/login/')

```

Результат работы
Регистрация



A screenshot of a web browser window with a pink header bar. The address bar shows "127.0.0.1:5000/registration/". The page contains a registration form with three input fields: "Your name:", "Login:", and "Password:". Below the fields is a "Sign In" button.

Your name:

Login:

Password:

После ввода данных и их подтверждения произошла регистрация и переадресация на форму входа

Username

Password

Username

Password

Hello, abobus!

После входа получаем имя пользователя.
При попытке произвести регистрацию на занятый логин

Your name:

Login:

Password:

There was an error!

Text: Имя abobus123 занято

Получаем верную ошибку

При попытке войти в аккаунт несуществующего пользователя или ввода неверного пароля получаем ошибку

Username

Password

There was an error!

Text: Пользователь не найден

Файлы проекта в директории sources