**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**

**Отчет по лабораторной работе № 4**

**“Сайт с входом/регистрацией”**

**по дисциплине**

**"Введение в ИТ"**

Выполнил: студент гр. БИН2001 Казаков А.А.

Проверил: Аршинов Е.А.

Москва, 2021 г.

1. #Импортируем необходимые библиотеки
2. import psycopg2
3. import requests
4. from flask import Flask, render\_template, request, redirect
6. #Подключаемся к базе данных
7. conn = psycopg2.connect(database="service\_db",
8. user="postgres",
9. password="0000",
10. host="localhost",
11. port="5432")
13. #Обработка POST и GET запросов
14. @app.route('/login/', methods=['POST', 'GET'])
15. def login():
16. if request.method == 'POST':
17. if request.form.get("login"):
18. #Проверка на правильность введённых логина и пароля
19. username = request.form.get('username')
20. password = request.form.get('password')
21. if len(username) == 0 or len(password) == 0:
22. return render\_template('error.html', err\_text='Логин или пароль не могут быть пустыми')
23. cursor.execute("SELECT \* FROM service.users WHERE login=%s AND password=%s", (str(username),
24. 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')
34. @app.route('/registration/', methods=['POST', 'GET'])
35. def registration():
36. if request.method == 'POST':
37. #Проверка на существование пользователя с введёнными данными и вывод ошибки о занятости логина или
38. #вывод сообщения о успешной регистрации
39. name = request.form.get('name')
40. login = request.form.get('login')
41. password = request.form.get('password')
42. if len(name) == 0 or len(login) == 0 or len(password) == 0:
43. return render\_template('error.html', err\_text='Пустые поля недопустимы')
44. cursor.execute("SELECT login FROM service.users WHERE login=%s", (str(login),))
45. records = list(cursor.fetchall())
46. if len(records) != 0:
47. return render\_template('error.html', err\_text='Имя ' + str(login) + ' занято')
48. else:
49. cursor.execute('INSERT INTO service.users (full\_name, login, password) VALUES (%s, %s, %s);',
50. (str(name), str(login), str(password)))
51. conn.commit()
52. return redirect('/login/')
53. return render\_template('registration.html')

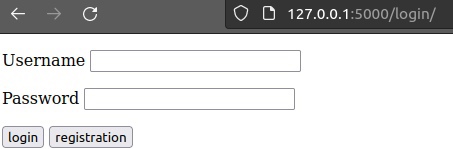


Рисунок 1 - Страница входа



Рисунок 2 - Страница при успешном входе

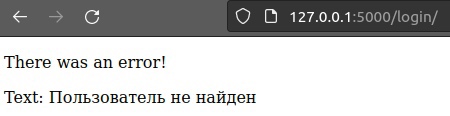


Рисунок 3 - Страница при вводе на странице входа неправильных данных

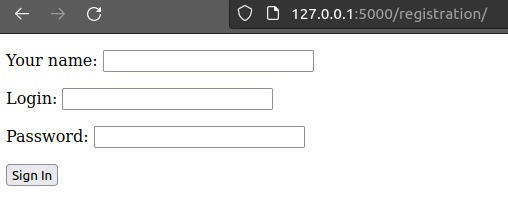


Рисунок 4 - Страница регистрации

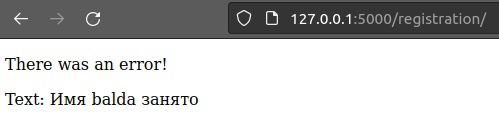


Рисунок 5 - Ошибка при попытке зарегистрировать существующий логин