

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ**

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 №2

по дисциплине «Введение в информационные технологии» на тему:

Подключение базы данных PostgreSQL к веб-приложению на микрофреймворке Flask.

Выполнил: студент группы БВТ2108

Шимко Никита Сергеевич

Проверил:

Мкртчян Грач Маратович

Москва

2021

1. Цель работы:

Цель работы – подключение базы данных PostgreSQL к веб-приложению. Сделать проверку работы формы авторизации пользователя.

2. Ход работы

Для начала создадим новый проект в IDE PyCharm и назовем его «fl_bd» (рис.1):

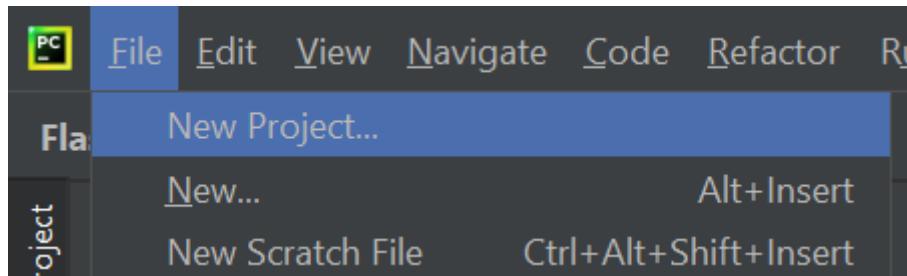


Рисунок 1 Создание проекта

Необходимо создать файл app.py (рис.2):

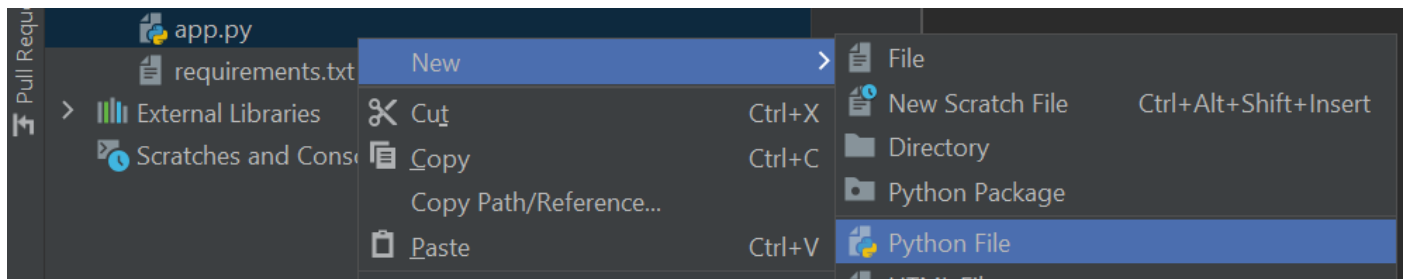


Рисунок 2 Создание файла

Добавляем в него следующий код:

```
import requests

from flask import Flask, render_template, request

import psycopg2

app = Flask(__name__)

@app.route('/', methods=['GET'])
def index():

    return render_template('login.html')

conn = psycopg2.connect(database="main_db",
                        user="postgres",
                        password="123456",
                        host="localhost",
                        port="5432")
```

```
cursor = conn.cursor()

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

    # print(username)
    # print('\n', password)
    # print(records)

    if not username:
        not_field = 'Поле "Username" не заполнено, попробуйте ещё раз.'
        return render_template('login.html', not_field=not_field)
    if not password:
        not_field = 'Поле "Password" не заполнено, попробуйте ещё раз.'
        return render_template('login.html', not_field=not_field)

    # пользователя не существует
    if not records:
        return render_template('error_form.html')
    else:
        return render_template('account.html', full_name=records[0][1], lgn=username,
psswrд=password)

#edit_1
```

Создадим директорию «templates» (рис.3):

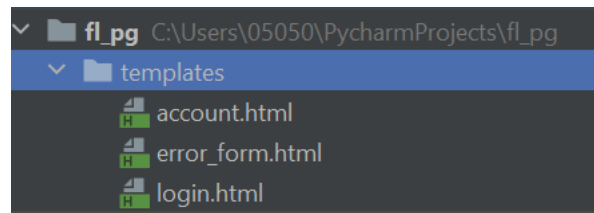


Рисунок 3 Создание директории

В account.html добавим следующее:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

    <form action="" method="post">

        {% if full_name %}

        <p>{{full_name}}, здарова!<br>

            Твой логин:  {{lgn}}<br>

            Твой пароль: {{psswrд}}

        </p>

        {% endif %}

    </p>

</form>

</head>

<body>

</body>

</html>
```

Далее, запустим «PgAdmin 4», создадим базу данных «fl_bd» (рис.4):

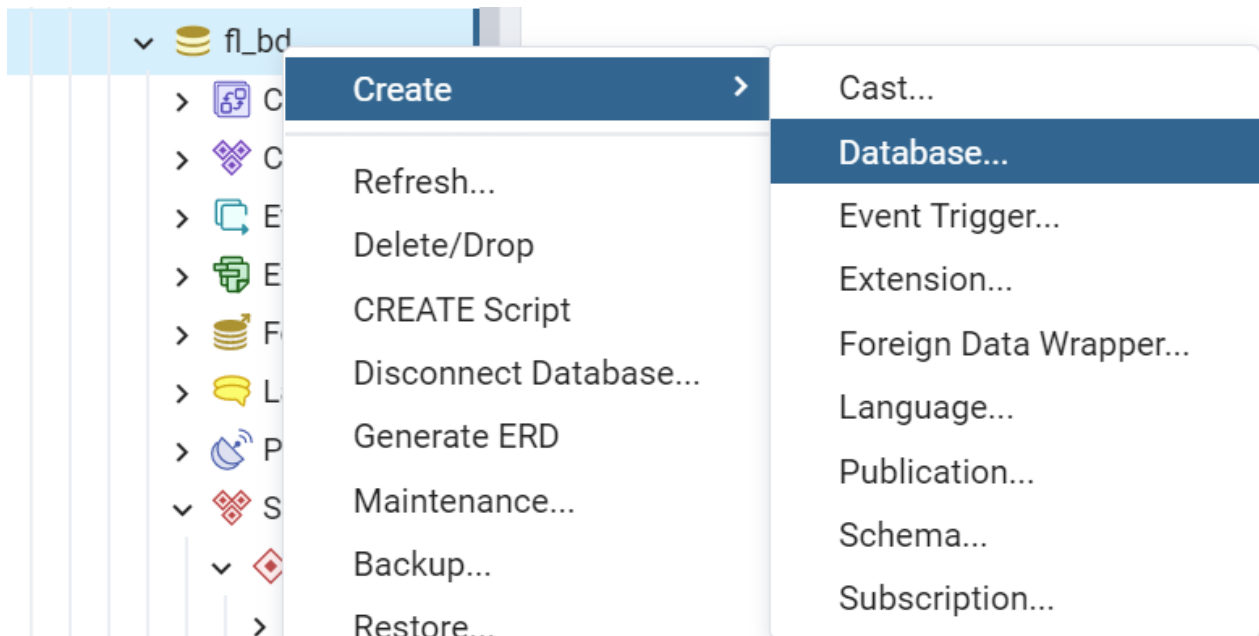


Рисунок 4 Создание базы данных

Запустим Query Editor (рис.5):

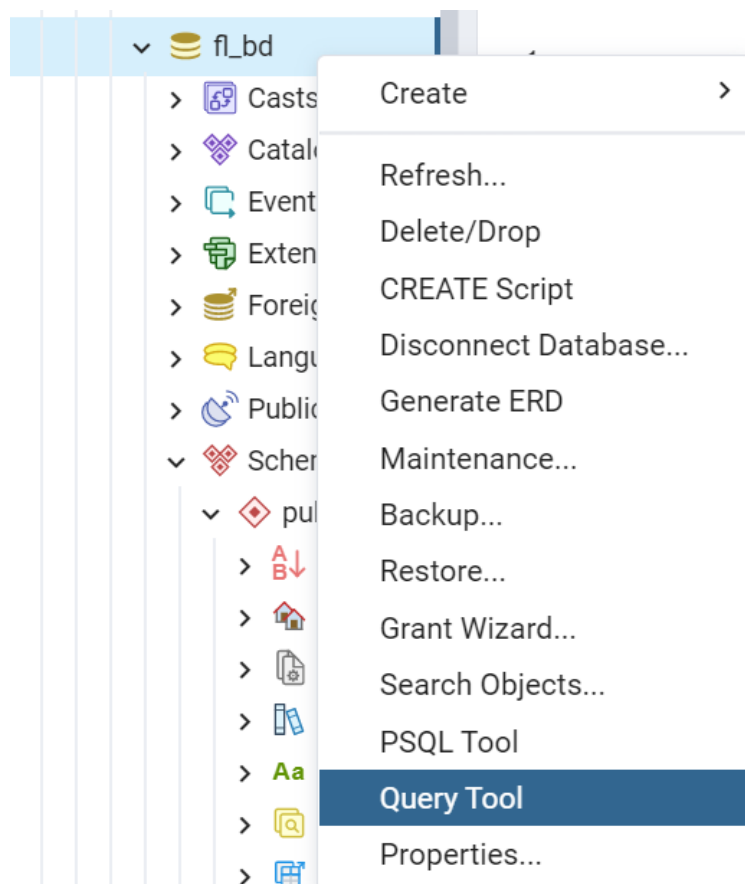


Рисунок 5 Query Editor

Сделаем следующие запросы к базе данных. Сначала создадим таблицу users с колонками id, full_name, login, password (рис.6):

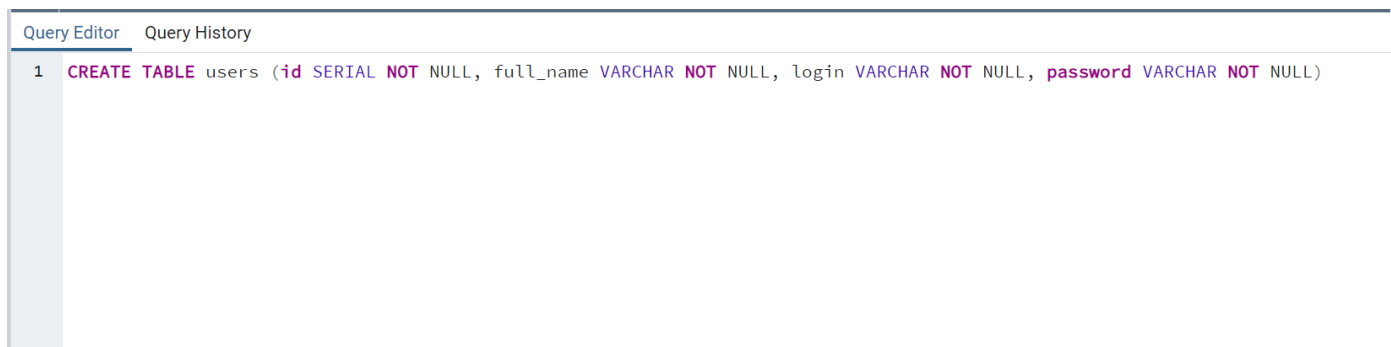


Рисунок 6 Создание таблицы users

Затем добавим 10 пользователей таким образом (рис.7):

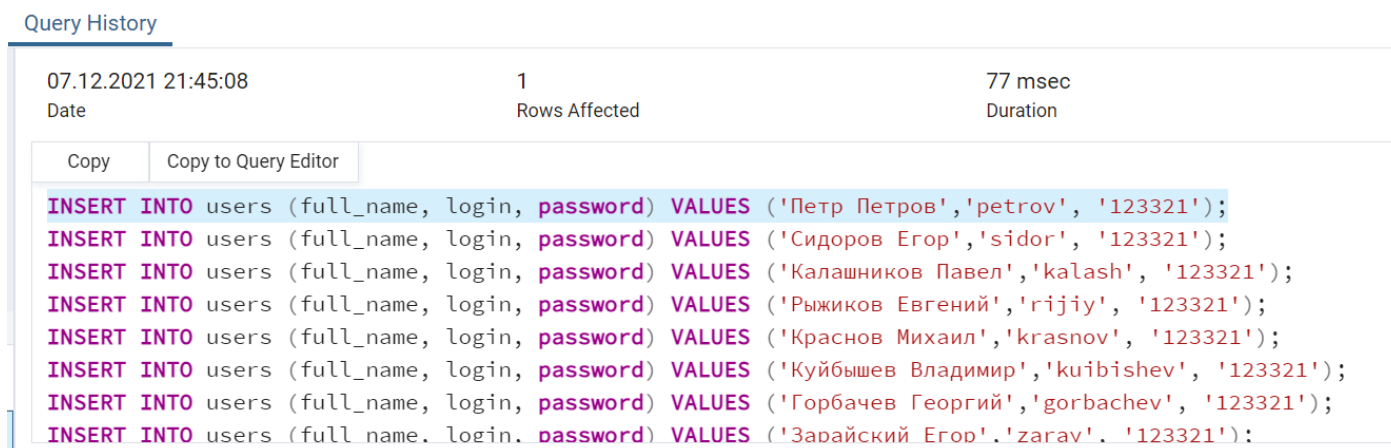


Рисунок 7 Пример добавления одного пользователя

Вот так выглядит таблица users в итоге (рис.8):

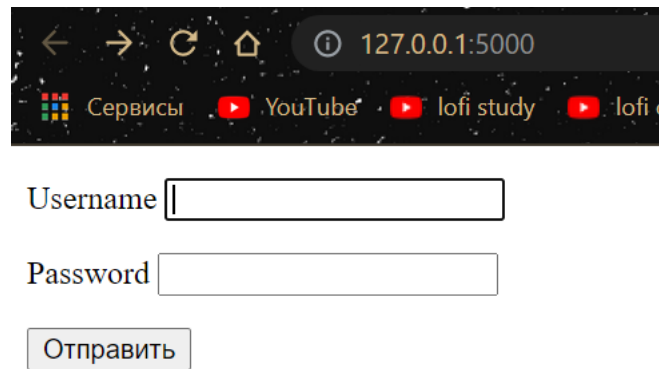
	id integer	full_name character varying	login character varying	password character varying
1	1	Петр Петров	petrov	123321
2	2	Сидоров Егор	sidor	123321
3	3	Калашников Павел	kalash	123321
4	4	Рыжиков Евгений	rijiy	123321
5	5	Краснов Михаил	krasnov	123321
6	6	Куйбышев Владимир	kuibishev	123321
7	7	Горбачев Георгий	gorbachev	123321
8	8	Зарайский Егор	zaray	123321
9	9	Шматко Тарас	shmatko	123321
10	10	Рыбка Семен	ribka	123321

Рисунок 8 Таблица users

3. Результат

Что получилось в итоге:

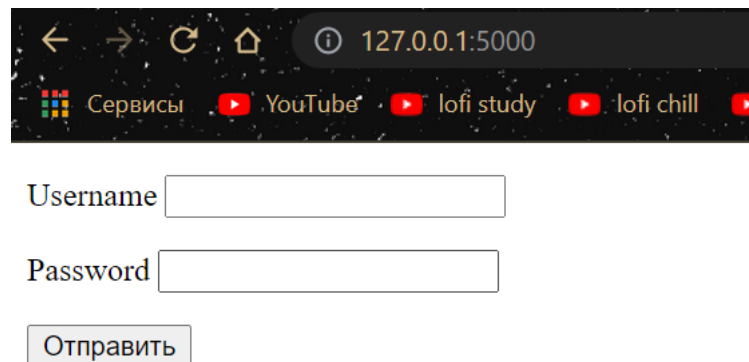
Если отправить пустое поле «Username», то сайт следующую выдаст ошибку (рис. 9):



A screenshot of a web browser window. The address bar shows '127.0.0.1:5000'. Below the address bar, there are several tabs: 'Сервисы', 'YouTube', 'lofi study', and 'lofi c'. The main content area displays a login form with two input fields: 'Username' and 'Password'. Both fields are empty. Below the 'Password' field is a button labeled 'Отправить'.

Рисунок 9 Пустое поле «Username»

Если отправить пустое поле «Password», то сайт следующую выдаст ошибку (рис. 10):

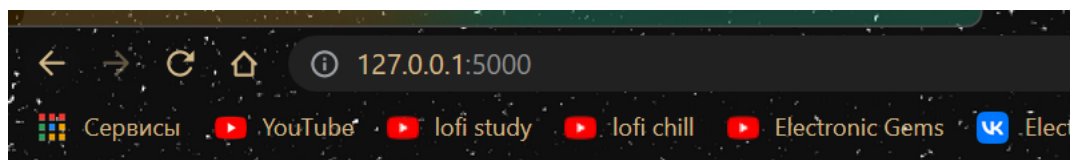


A screenshot of a web browser window. The address bar shows '127.0.0.1:5000'. Below the address bar, there are several tabs: 'Сервисы', 'YouTube', 'lofi study', 'lofi chill', and 'lofi c'. The main content area displays a login form with two input fields: 'Username' and 'Password'. Both fields are empty. Below the 'Password' field is a button labeled 'Отправить'.

Поле "Password" не заполнено, попробуйте ещё раз.

Рисунок 10 Пустое поле «Password»

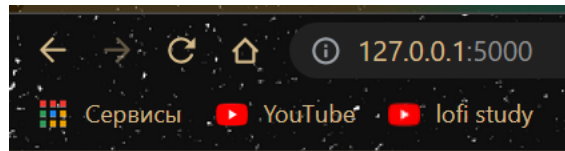
Попробуем ввести логин и пароль, которого нет в базе данных. Нам высветится такое сообщение (рис.11):



Такого пользователя не существует.

Рисунок 11 Логин и пароль, которого нет в базе данных

Попробуем ввести логин и пароль, который есть в базе данных. Нам высветится такое сообщение (рис.12):



Петр Петров, здарова!
Твой логин: petrov
Твой пароль: 123321

Рисунок 12 Логин и пароль, который есть в базе данных

4. Вывод

Мы научились пользоваться микрофрейворком Flask и базой данных PostgreSQL, создав на их основе веб-приложение с формой авторизации.