**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

Чугунова Татьяна Александровна

Проверил:

Мкртчян Грач Маратович

Москва

2021

1. Цель работы:

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

1. Ход работы

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

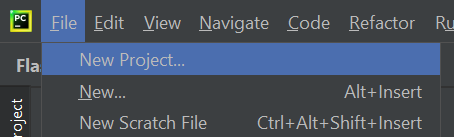


Рисунок 1 Создание проекта

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

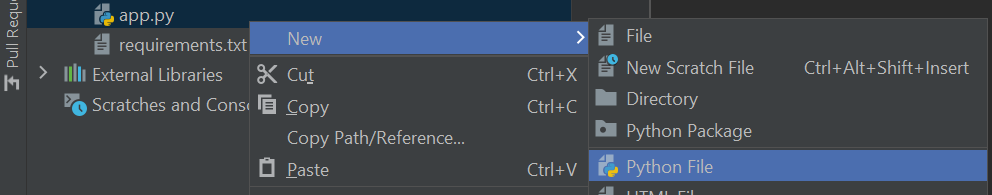


Рисунок 2 Создание файла

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="fl\_bd",

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, psswrd=password)

Создадим директорию «templates» (рис.3):

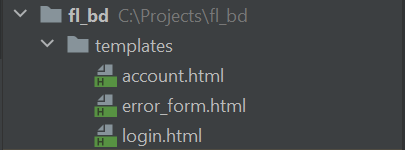


Рисунок 3 Создание директории

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}}, привет :)<br>

Твой логин: {{lgn}}<br>

Твой пароль: {{psswrd}}

</p>

{% endif %}

</p>

</form>

</head>

<body>

</body>

</html>

В error\_form.html добавим следующее:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>error</title>

</head>

<body>

<h1>Такого пользователя не существует.</h1>

</body>

</html>

В login.html добавляем следующее:

<!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">

</p>

</form>

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

{% if not\_field %}

<p>{{not\_field}} </p>

{% endif %}

</p>

</form>

</body>

</html>

Далее, запустим «PgAdmin 4», создадим базу данных «fl\_bd» (рис.4):

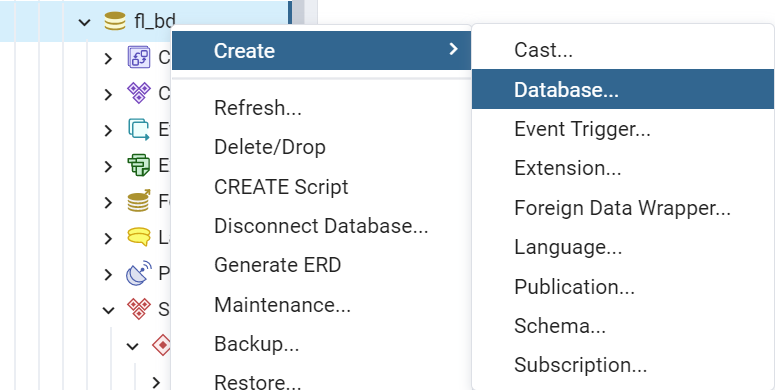


Рисунок 4 Создание базы данных

Запустим Query Editor (рис.5):

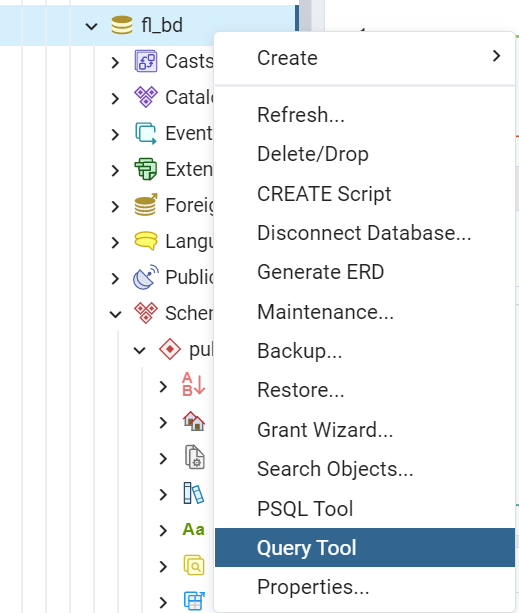


Рисунок 5 Query Editor

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

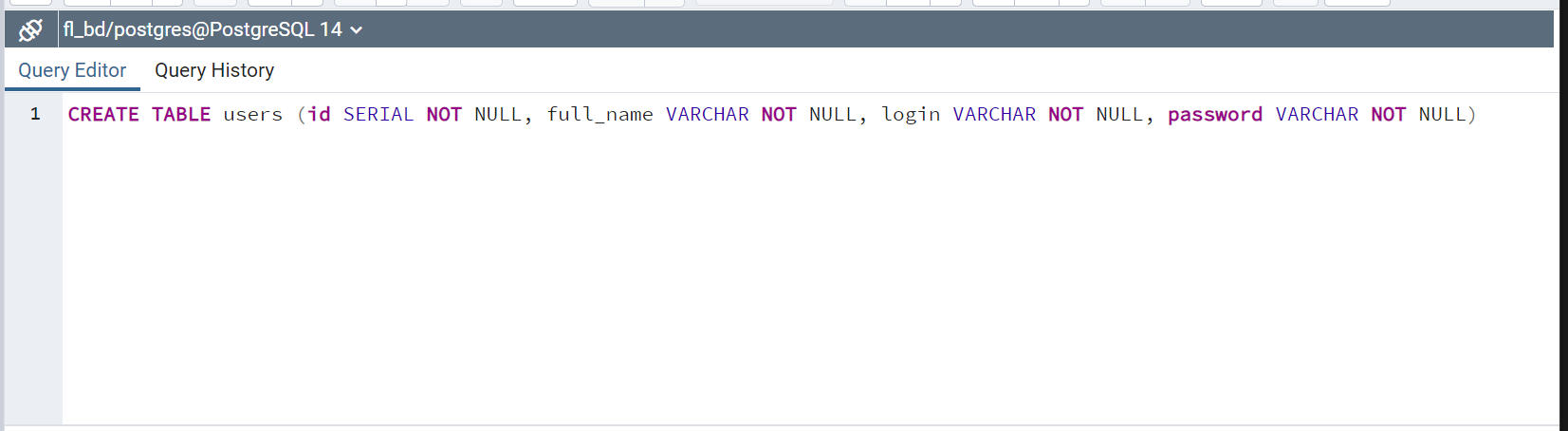


Рисунок 6 Создание таблицы users

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

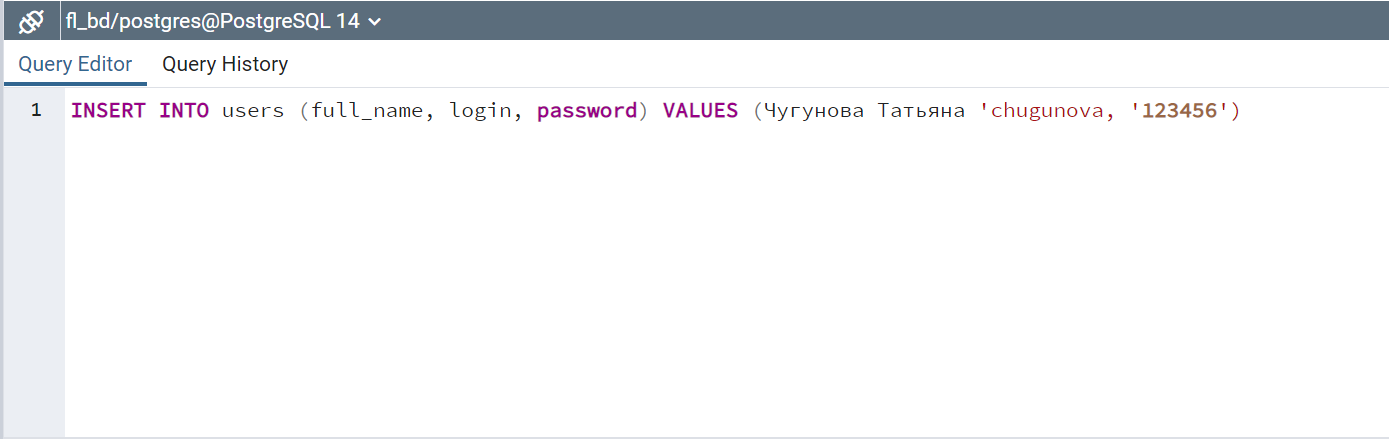


Рисунок 7 Пример добавления одного пользователя

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

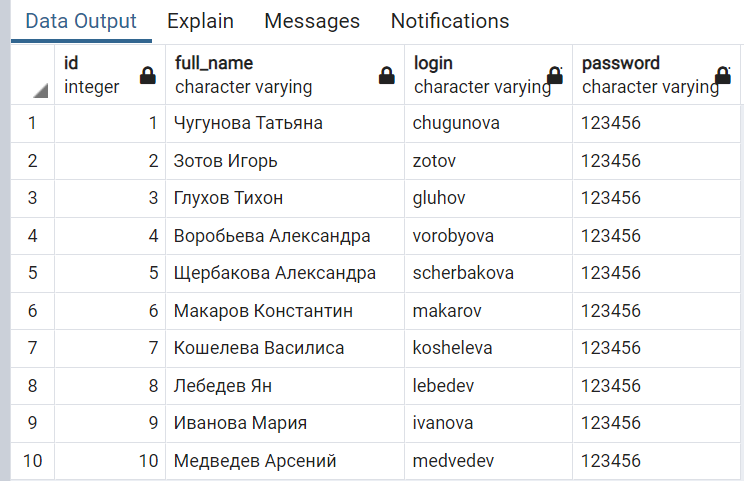


Рисунок 8 Таблица users

1. Результат

Что получилось в итоге:

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

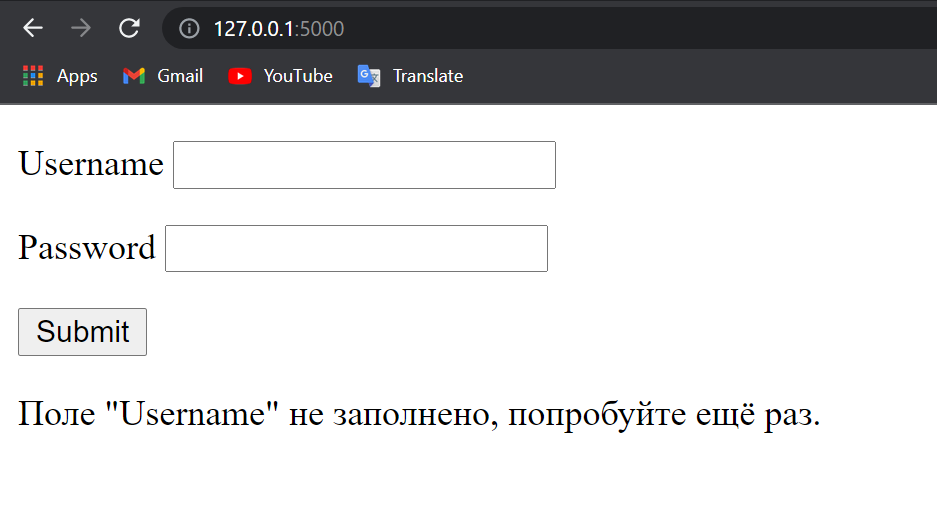


Рисунок 9 Пустое поле «Username

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

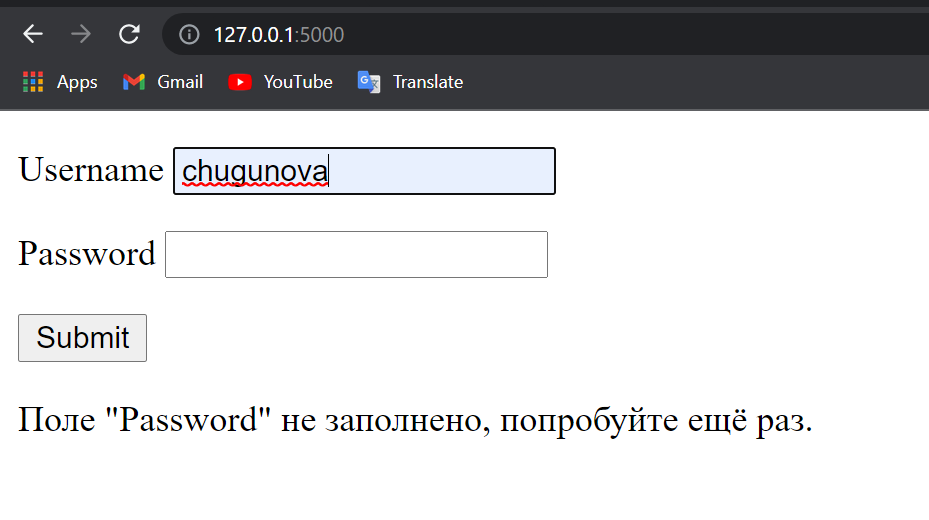


Рисунок 10 Пустое поле «Password»

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

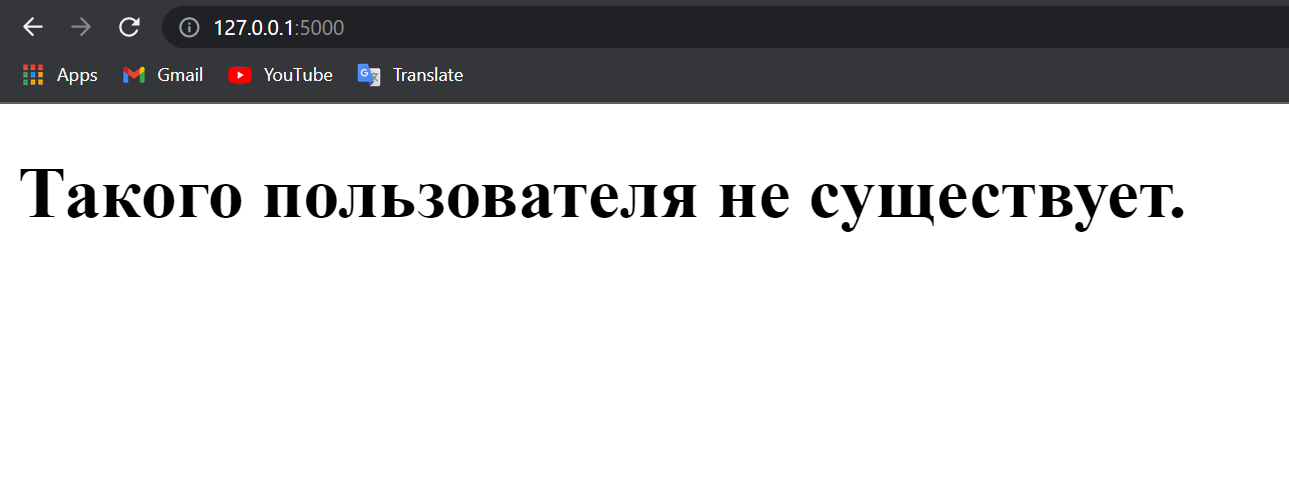


Рисунок 11 Логин и пароль, которого нет в базе данных

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

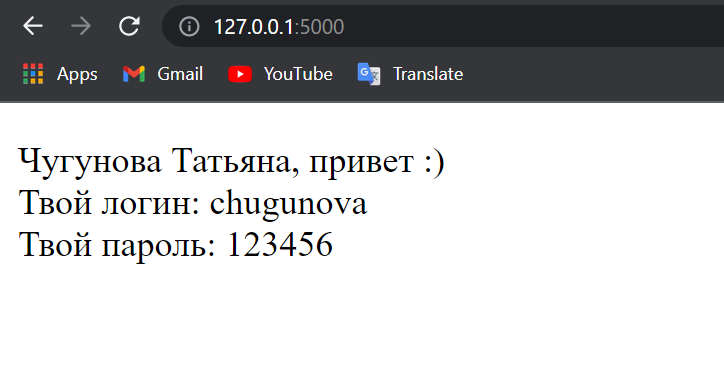


Рисунок 12 Логин и пароль, который есть в базе данных

1. Вывод

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