**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 №3**

по дисциплине «Введение в информационные технологии» на тему:

Подключение базы данных PostgreSQL к веб-приложению на Flask.

Выполнил: студент группы БВТ2108

Степанова Дарина Викторовна

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

Создадим новый проект в IDE PyCharm (рис.1):

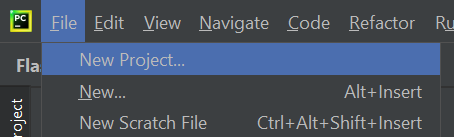


Рисунок 1

Затем создим файл app.py (рис.2):

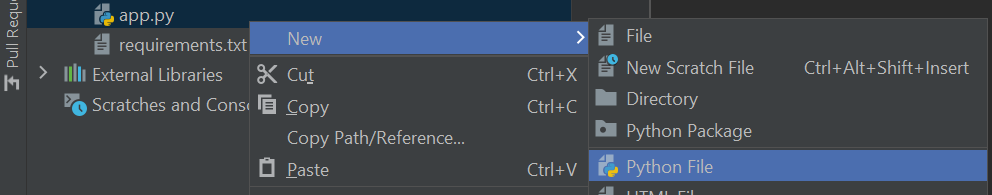


Рисунок 2

Добавляем в него следующий код:

import requests

from flask import Flask, render\_template, request

import psycopg2

app = Flask(\_\_name\_\_)

conn = psycopg2.connect(database="nana",

user="postgres",

password="123456",

host="localhost",

port="5432")

cursor = conn.cursor()

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

def index():

return render\_template('login.html')

@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())

# не введено имя пользователя

if not username:

not\_field = 'Заполните поле "Username".'

return render\_template('login.html', not\_field=not\_field)

# не введён пароль

if not password:

not\_field = 'Заполните поле "Password".'

return render\_template('login.html', not\_field=not\_field)

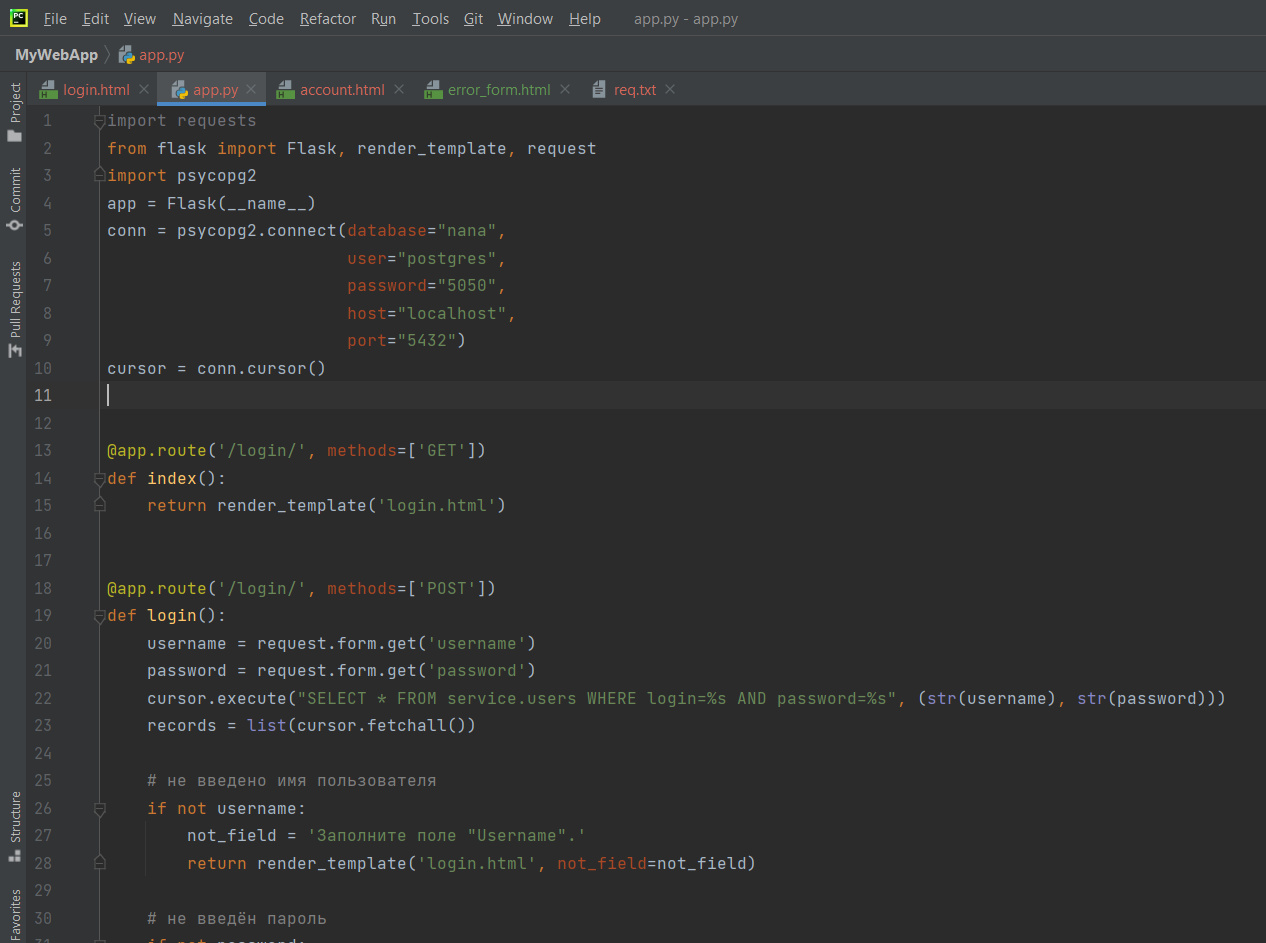
# пользователь не найден

if not records:

return render\_template('error\_form.html')

else:

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

Результат (рис.3)

*Рисунок 3*

Создадим директорию «templates» (рис.4):

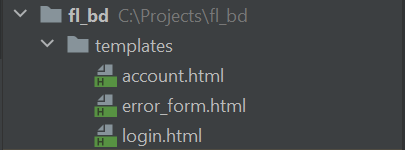


Рисунок 4

В account.html добавляем код:

<!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>

Ваш логин: {{login}}<br>

Ваш пароль: {{password}}

</p>

{% endif %}

</p>

</form>

</head>

<body>

</body>

</html>

В error\_form.html добавляем код:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>error</title>

</head>

<body>

<h1>Пользователь не найден</h1>

</body>

</html>

В login.html добавляем:

<!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>

Запускаем «PgAdmin 4», создаём базу данных «nana» (рис.5):

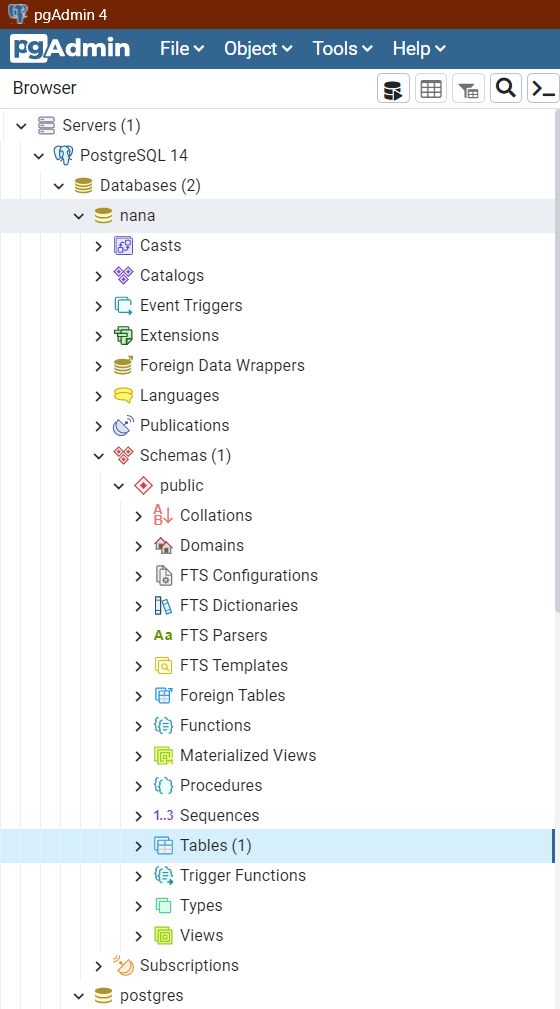


Рисунок 5

Запустим Query Editor (рис.6):

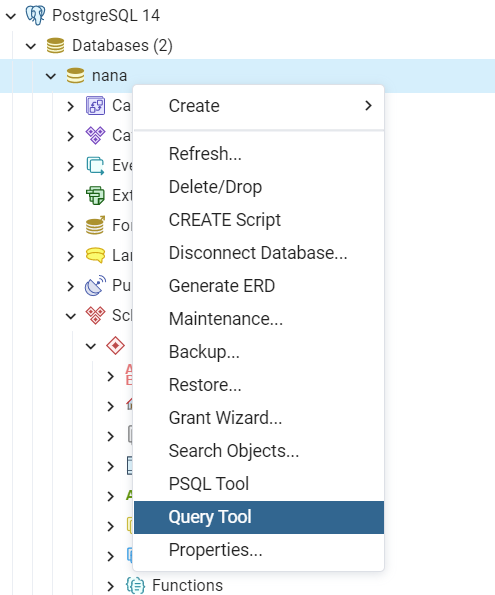


Рисунок 6

Cоздадим таблицу users с колонками id, full\_name, login, password (рис.7):

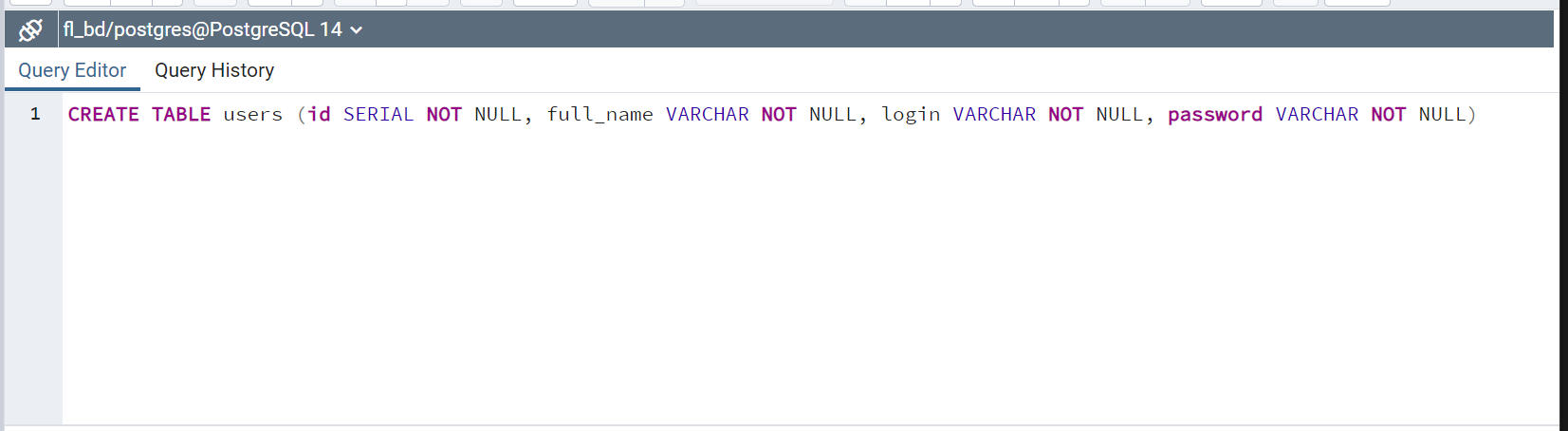


Рисунок 7

Затем добавим 10 пользователей. Пример добавления нескольких пользователей показан на рис.8:

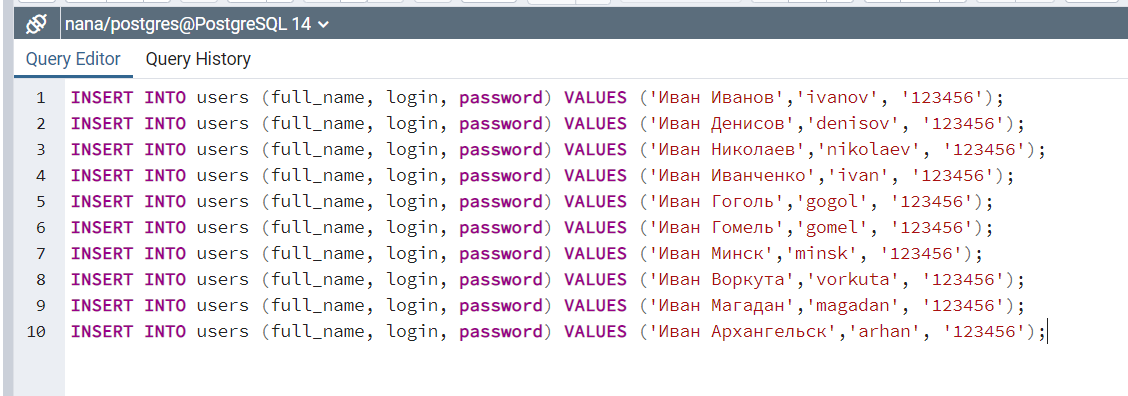


Рисунок 8

Вот так выглядит таблица users в итоге (рис.9):

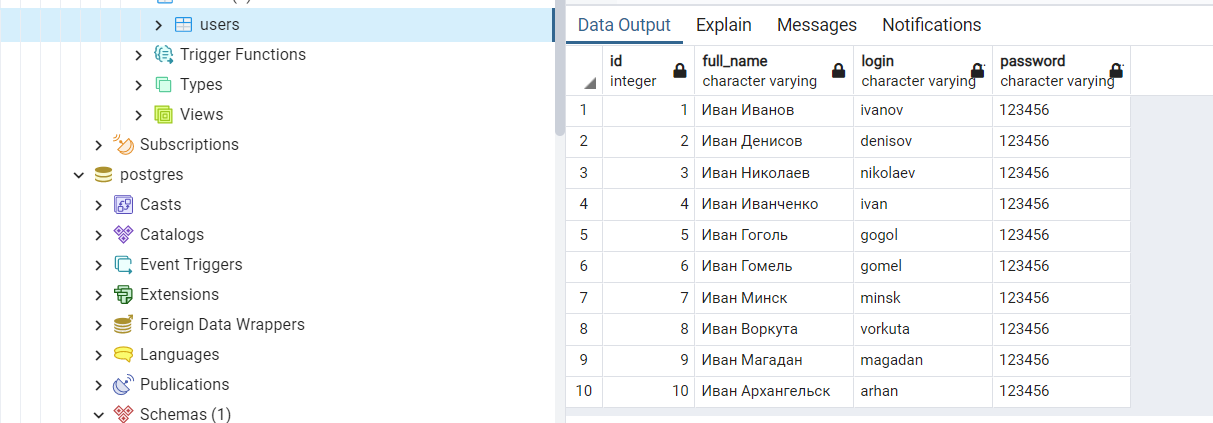


Рисунок 9

1. Результат

Что получилось в итоге:

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

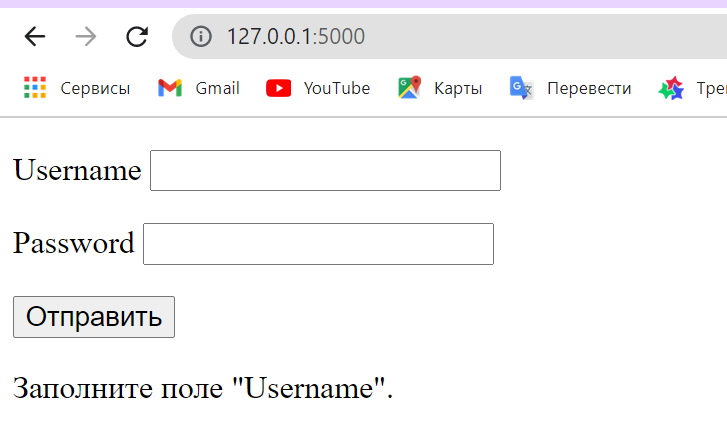


Рисунок 10

Если отправить пустое поле «Password», то сайт выдаст ошибку (рис. 11):

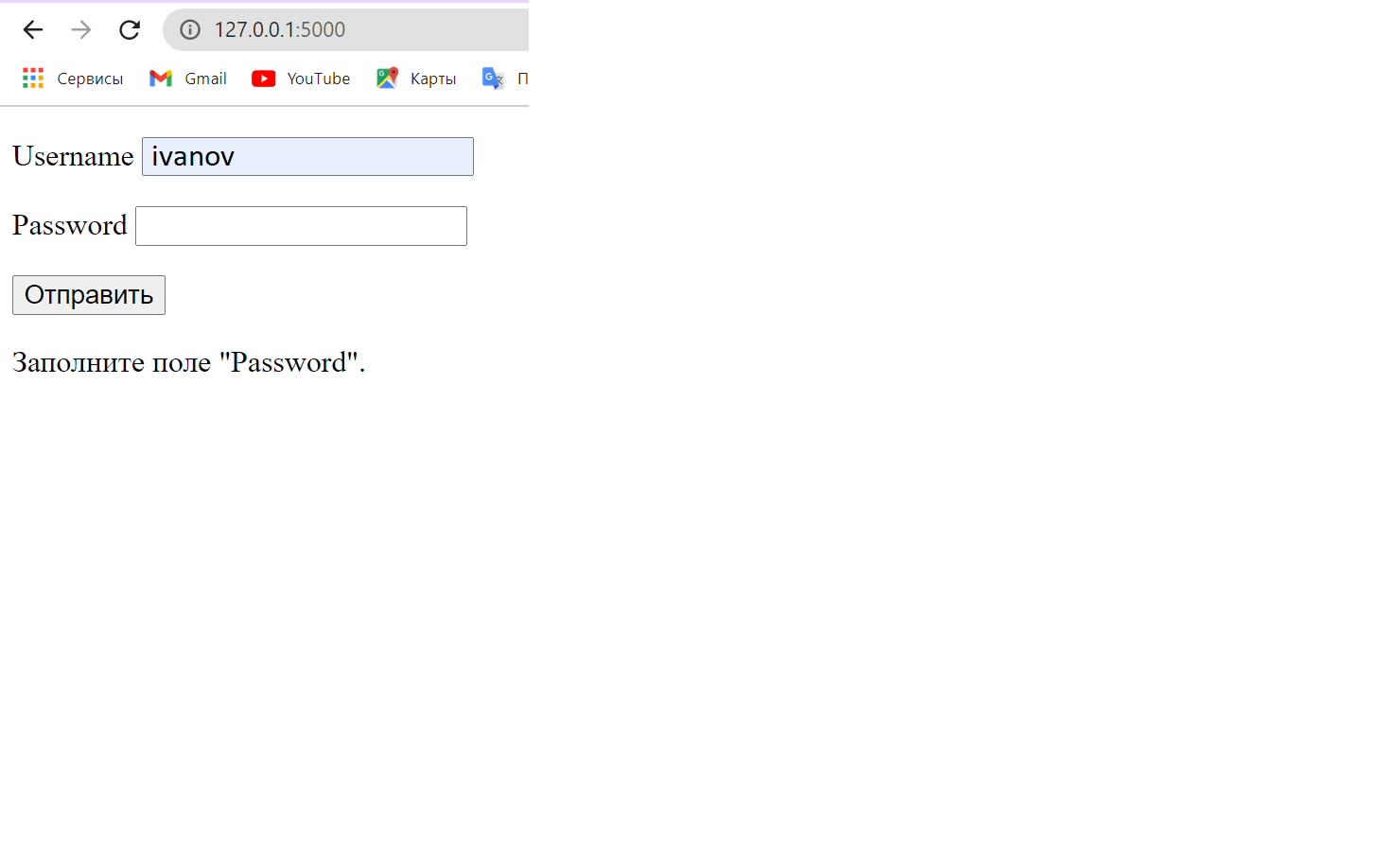


Рисунок 11

Попробуем ввести логин и пароль, которого нет в базе данных. Нам высветится следующее сообщение (рис.12):

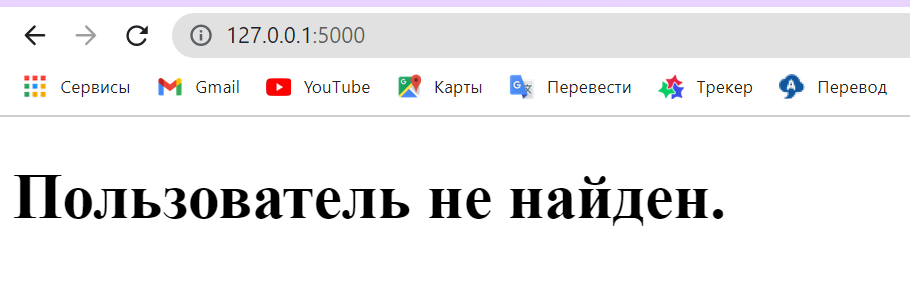


Рисунок 12

Попробуем ввести логин и пароль, который есть в базе данных. Нам высветится такое сообщение (рис.13):

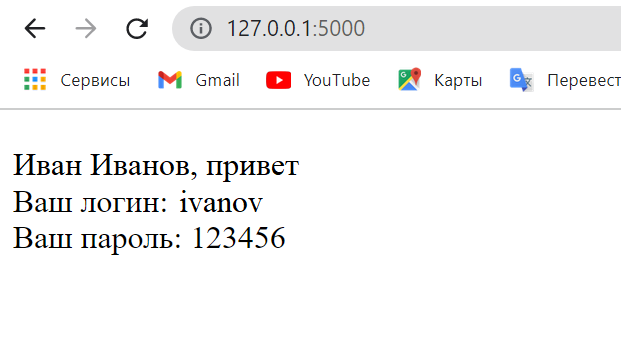


Рисунок 13

1. Вывод

Мы научились пользоваться Flask, подключать базу данных PostgreSQL к веб-приложению. Проверили работу формы авторизации пользователя.