

**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»**

---

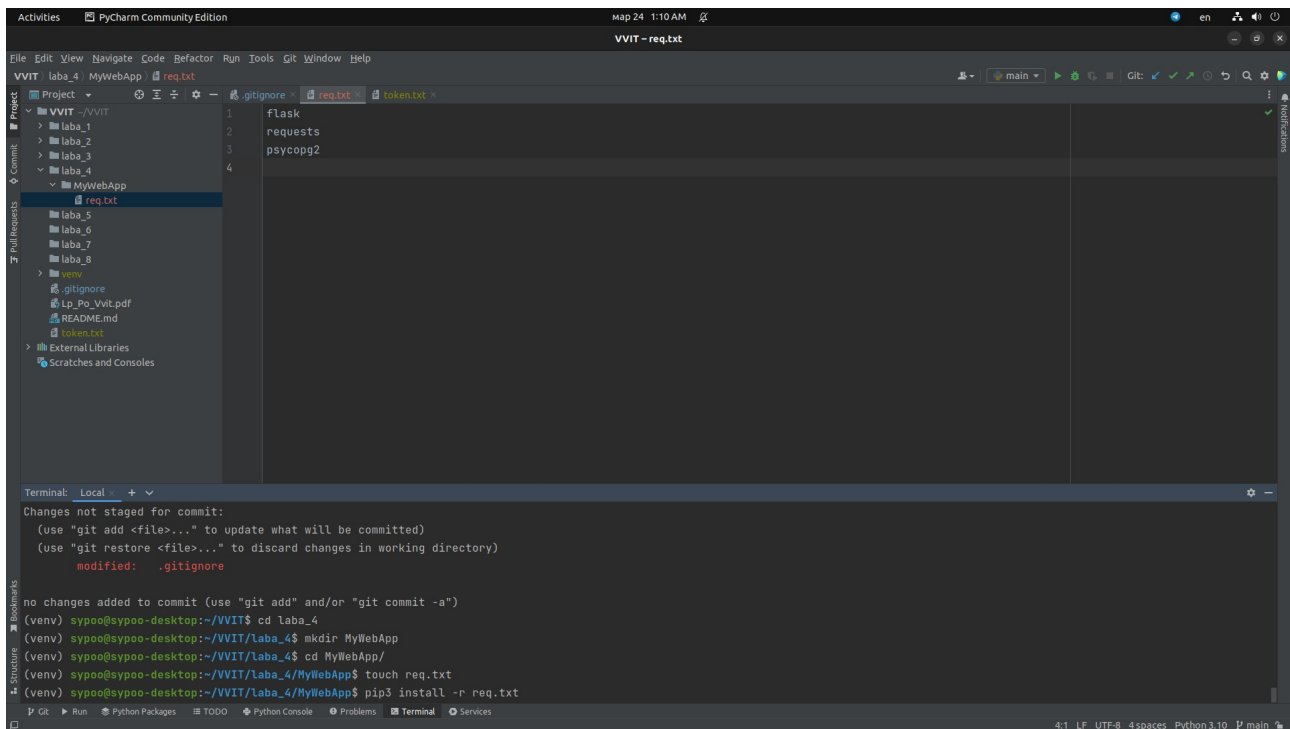
**Кафедра «Математическая кибернетика и информационные  
технологии»**

**Лабораторная работа № 4  
Создание системы авторизации в  
веб-приложении  
по дисциплине  
«Введение в информационные технологии»**

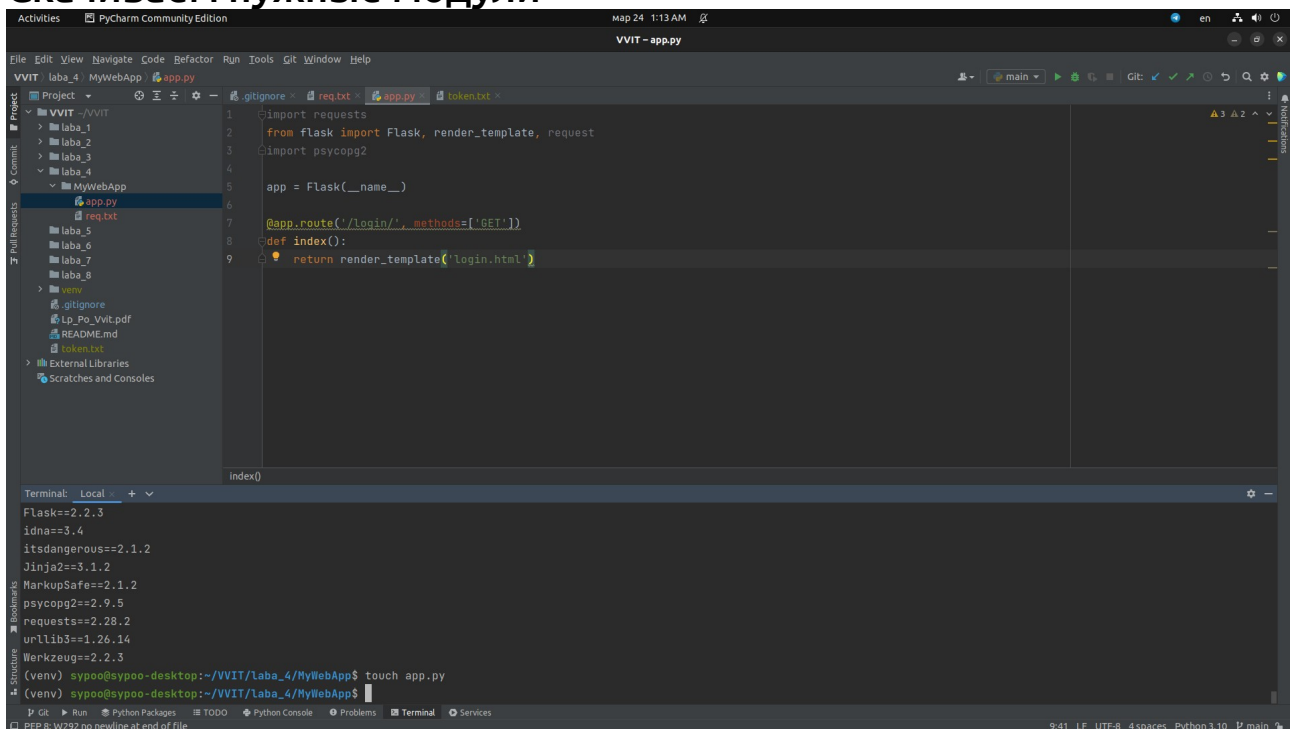
**Выполнил: студент гр. БВТ2201  
Шамсутдинов Р.Ф.**

**Проверил:**

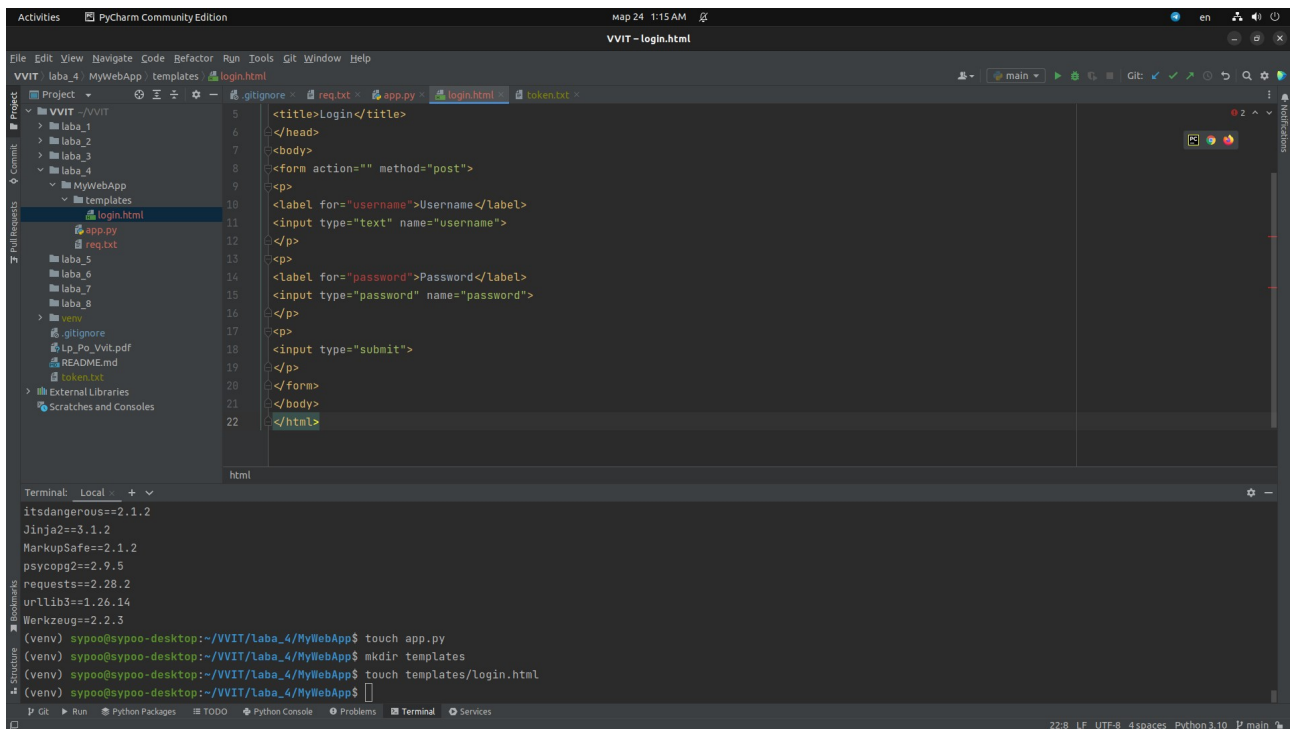
**Москва, 2023 г**



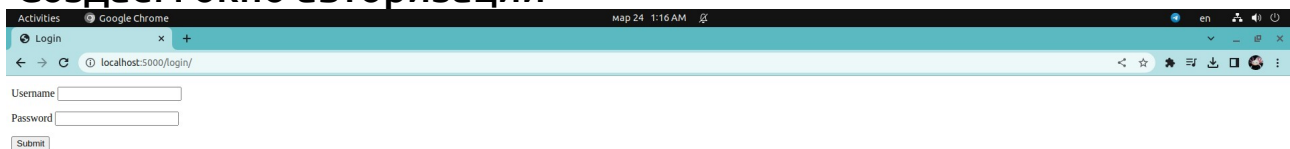
## Скачиваем нужные модули



## Создаем веб приложение



## Создаем окно авторизации



проверяем, все работает

```
Activities Terminal map 24 1:26 AM sypoo@sypoo-desktop: -
sypoo@sypoo-desktop:~$ sudo -u postgres psql
could not change directory to "/home/sypoo": Permission denied
psql (14.7 (Ubuntu 14.7-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# CREATE DATABASE service_db;
CREATE DATABASE
postgres=# \c service_db
You are now connected to database "service_db" as user "postgres".
service_db=# CREATE SCHEMA service;
CREATE SCHEMA
service_db=# CREATE TABLE service.users (id SERIAL NOT NULL, full_name VARCHAR NOT NULL, login VARCHAR NOT NULL, password VARCHAR NOT NULL);
CREATE TABLE
service_db=# \d
Did not find any relations.
service_db=# \l
service_db=# INSERT INTO service.users (full_name, login, password) VALUES ('Ivanov Ivan', 'ivanov', 123456);
INSERT 0 1
service_db=# SELECT * FROM service.users;
 id | full_name | login | password
-----+-----+-----+-----
  1 | Ivanov Ivan | ivanov | 123456
(1 row)

service_db=#
```

## Создаем базу данных, и вносим туда данные

```
Activities PyCharm Community Edition map 24 1:40 AM VVIT - app.py
File Edit View Navigate Code Refactor Run Tools Git Window Help
VVIT lab4_4 MyWebApp app.py
Project
  VVIT -VVIT
    lab4_1
    lab4_2
    lab4_3
    lab4_4
      MyWebApp
        templates
          account.html
          login.html
          app.py
          conf.py
          req.txt
        lab4_5
        lab4_6
        lab4_7
        lab4_8
        venv
        .gitignore
        Lp_Po_Vvit.pdf
        README.md
        token.txt
    External Libraries
    Scratches and Consoles
Terminal
  Local
  (venv) sypoo@sypoo-desktop:~/VVIT/Laba_4/MyWebApp$ flask run
  * Debug mode: off
  WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
  * Running on http://127.0.0.1:5000
  Press CTRL+C to quit

app = Flask(__name__)

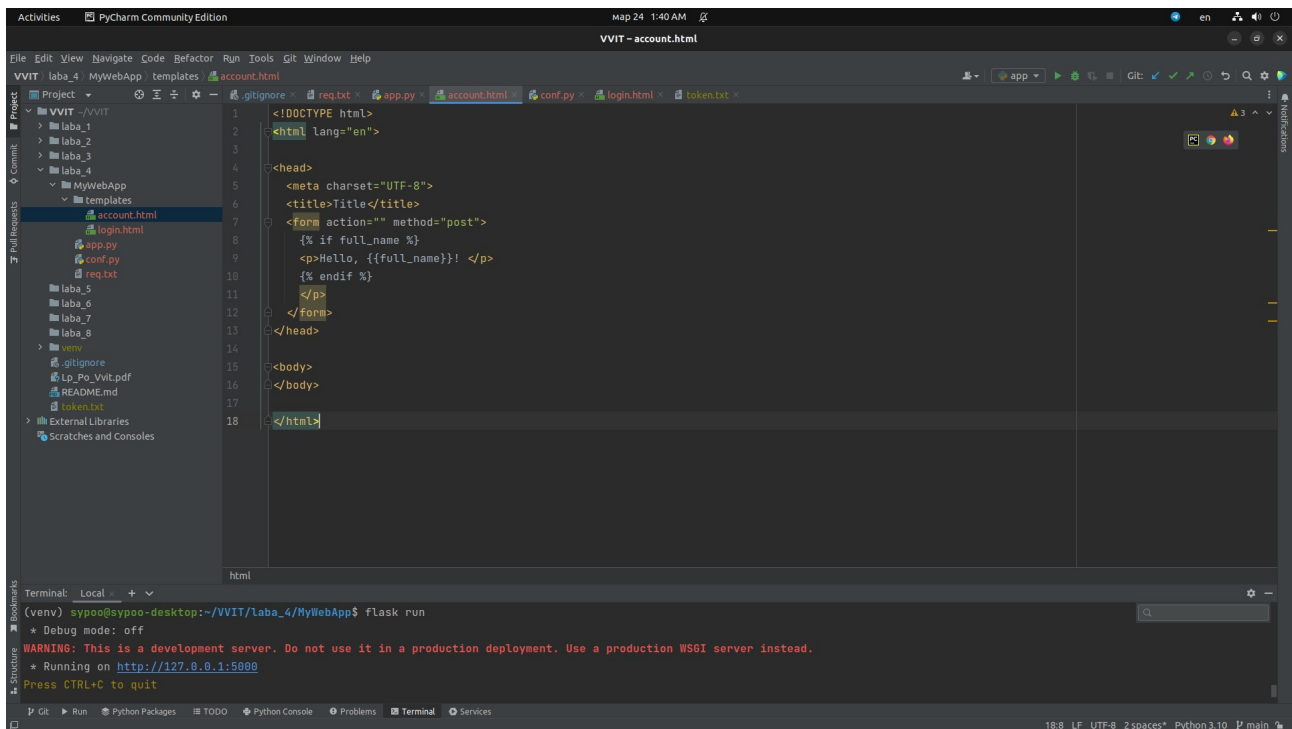
conn = psycopg2.connect(database="service_db",
                        user="postgres",
                        password=password,
                        host="localhost",
                        port="5432")

cursor = conn.cursor()

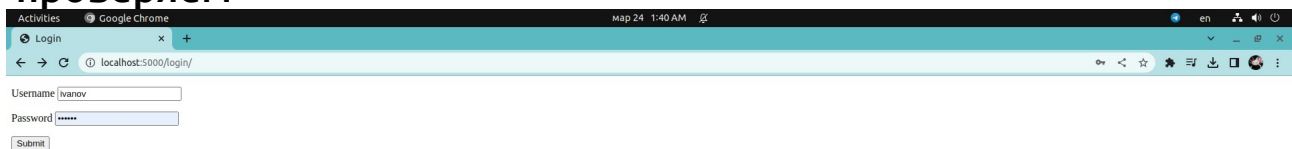
@app.route('/login/', methods=['GET'])
def index():
    return render_template('login.html')

@app.route('/login/', methods=['POST'])
def login():
    username = request.form.get('username')
    password = request.form.get('password')
    cursor.execute("SELECT * FROM service.users WHERE login=%s AND password=%s", (str(username), str(password)))
    records = list(cursor.fetchall())
    return render_template('account.html', full_name=records[0][1])
```

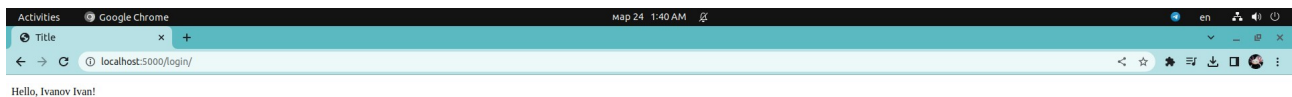
Подключаемся к базе данных с помощью питон кода, создаем страничку при успешной авторизации



проверяем



все работает



## Заполняем БД новыми пользователями

```
Activities Terminal map 24 2:29 AM sypoo@sypoo-desktop: -
LINE 1: SELECT * FROM service.users WHERE login=ivanov;
service_db=# SELECT * FROM service.users WHERE login='ivanov';
id | full_name | login | password
-----
1 | Ivanov Ivan | ivanov | 123456
(1 row)

service_db=# SELECT * FROM service.users WHERE login='ivanov2';
id | full_name | login | password
-----
(0 rows)

service_db=# SELECT * FROM service.users WHERE login='ivanov2' AND password='123';
id | full_name | login | password
-----
(0 rows)

service_db=# INSERT INTO service.users (full_name, login, password) VALUES ('Ivanov Oleg', 'IvOleg', 0123456);
INSERT 0 1
service_db=# INSERT INTO service.users (full_name, login, password) VALUES ('Mlnov Alex', 'mlnA', 1123456);
INSERT 0 1
service_db=# INSERT INTO service.users (full_name, login, password) VALUES ('Sldorov Dmtriy', 'sldr', 2123456);
INSERT 0 1
service_db=# INSERT INTO service.users (full_name, login, password) VALUES ('Sonin Michael', 'son', 3123456);
INSERT 0 1
service_db=# INSERT INTO service.users (full_name, login, password) VALUES ('Dubnikov Igor', 'dub', 4123456);
INSERT 0 1
service_db=# INSERT INTO service.users (full_name, login, password) VALUES ('Mavrln Ivan', 'Mava', 5123456);
INSERT 0 1
service_db=# INSERT INTO service.users (full_name, login, password) VALUES ('Popkov Nikita', 'nekit', 6123456);
INSERT 0 1
service_db=# INSERT INTO service.users (full_name, login, password) VALUES ('Zubkov Vitaliy', 'zub', 7123456);
INSERT 0 1
service_db=# INSERT INTO service.users (full_name, login, password) VALUES ('Kovalev Evgeniy', 'koval', 8123456);
INSERT 0 1
service_db=# INSERT INTO service.users (full_name, login, password) VALUES ('Kostln Pavel', 'Pasha', 9123456);
INSERT 0 1
service_db=# SELECT * FROM service.users;
id | full_name | login | password
-----
1 | Ivanov Ivan | ivanov | 123456
2 | Ivanov Oleg | IvOleg | 123456
3 | Mlnov Alex | mlnA | 1123456
4 | Sldorov Dmtriy | sldr | 2123456
5 | Sonin Michael | son | 3123456
6 | Dubnikov Igor | dub | 4123456
7 | Mavrln Ivan | Mava | 5123456
8 | Popkov Nikita | nekit | 6123456
9 | Zubkov Vitaliy | zub | 7123456
10 | Kovalev Evgeniy | koval | 8123456
11 | Kostln Pavel | Pasha | 9123456
(11 rows)

service_db=#
```

Добавляем в питон код исключения на пустые логин и пароль, а также на логины и пароли, которых нет в базе данных, также создаем странички ответа при выполнении этих исключений

Activities PyCharm Community Edition Map 24 3:16 AM en

VVIT - app.py

File Edit View Navigate Code Refactor Run Tools Git Window Help

lab\_4 MyWebApp app.py

Project

- lab\_4
- lab\_1
- lab\_2
- lab\_3
- lab\_4
- MyWebApp
  - templates
    - account.html
    - blank.html
    - login.html
    - stranger.html
  - app.py
  - conf.py
  - req.txt
- lab\_5
- lab\_6
- lab\_7
- lab\_8
- venv
  - .gitignore
  - Lp\_Po\_Vvit.pdf
  - README.md
  - token.txt
- External Libraries
- Scratches and Consoles

```
18 def index():
19     return render_template('login.html')
20
21 @app.route('/login/', methods=['POST'])
22 def login():
23     username = request.form.get('username')
24     password = request.form.get('password')
25
26     if username == '' or password == '':
27         return render_template('blank.html')
28
29     cursor.execute("SELECT * FROM service.users WHERE login=%s AND password=%s", (str(username), str(password)))
30     records = list(cursor.fetchall())
31
32     if len(records) == 0:
33         return render_template('stranger.html')
34
35     return render_template('account.html', full_name=records[0][1], login=records[0][2], password=records[0][3])
36
```

Terminal: Local +

```
^C(venv) sypoo@sypoo-desktop:~/VVIT/lab_4/MyWebApp$ flask run
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [24/Mar/2023 03:15:38] "GET /login/ HTTP/1.1" 200 -
127.0.0.1 - - [24/Mar/2023 03:15:39] "POST /login/ HTTP/1.1" 200 -
(venv) sypoo@sypoo-desktop:~/VVIT/lab_4/MyWebApp$
```

36:1 LF UTF-8 4spaces Python 3.10 P main

Activities PyCharm Community Edition Map 24 3:16 AM en

VVIT - stranger.html

File Edit View Navigate Code Refactor Run Tools Git Window Help

lab\_4 MyWebApp templates stranger.html

Project

- lab\_4
- lab\_1
- lab\_2
- lab\_3
- lab\_4
- MyWebApp
  - templates
    - account.html
    - blank.html
    - login.html
    - stranger.html
  - app.py
  - conf.py
  - req.txt
- lab\_5
- lab\_6
- lab\_7
- lab\_8
- venv
  - .gitignore
  - Lp\_Po\_Vvit.pdf
  - README.md
  - token.txt
- External Libraries
- Scratches and Consoles

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <title>Title</title>
7     <form action="" method="post">
8         <p> Wrong login or password! </p>
9     </form>
10 </head>
11
12 <body>
13 </body>
14
15 </html>
```

html | head | form | p

Terminal: Local +

```
^C(venv) sypoo@sypoo-desktop:~/VVIT/lab_4/MyWebApp$ flask run
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [24/Mar/2023 03:15:38] "GET /login/ HTTP/1.1" 200 -
127.0.0.1 - - [24/Mar/2023 03:15:39] "POST /login/ HTTP/1.1" 200 -
(venv) sypoo@sypoo-desktop:~/VVIT/lab_4/MyWebApp$
```

0:33 LF UTF-8 2spaces Python 3.10 P main

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Title</title>
7   <form action="" method="post">
8     <p> Login and password must be not blank! </p>
9   </form>
10 </head>
11
12 <body>
13 </body>
14
15 </html>
```

```
^C(venv) sypoo@sypoo-desktop:~/VVIT/Laba_4/MyWebApp$ flask run
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [24/Mar/2023 03:15:38] "GET /login/ HTTP/1.1" 200 -
127.0.0.1 - - [24/Mar/2023 03:15:39] "POST /login/ HTTP/1.1" 200 -
(venv) sypoo@sypoo-desktop:~/VVIT/Laba_4/MyWebApp$
```

Activities Google Chrome

Login x +

← → ↻ ⓘ localhost:5000/login/

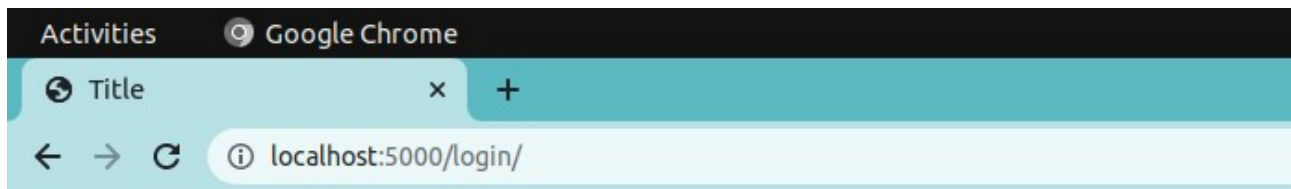
Username

Password

Проверяем работу веб приложения

Правильные данные нового пользователя:



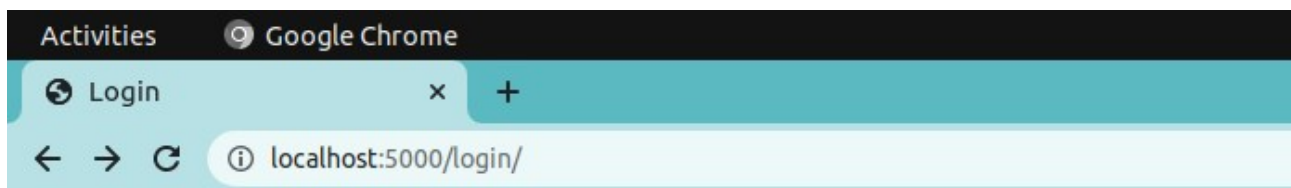


Hello, Dubnikov Igor!

Your login: dub

Your password: 4123456

## Пустые логин и пароль:



Username

Password

Activities Google Chrome

Login x +

localhost:5000/login/

Username

Password

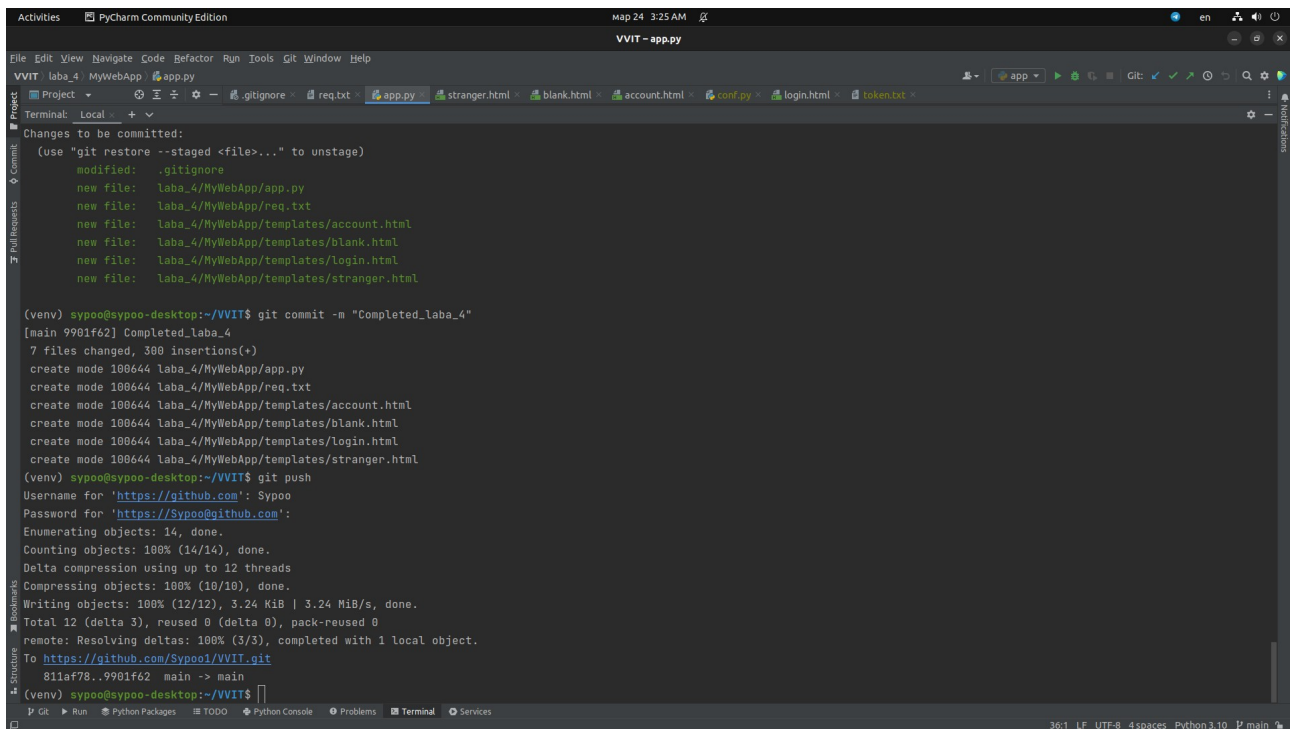
## Неправильные логин и пароль:

Activities Google Chrome

Title x +

localhost:5000/login/

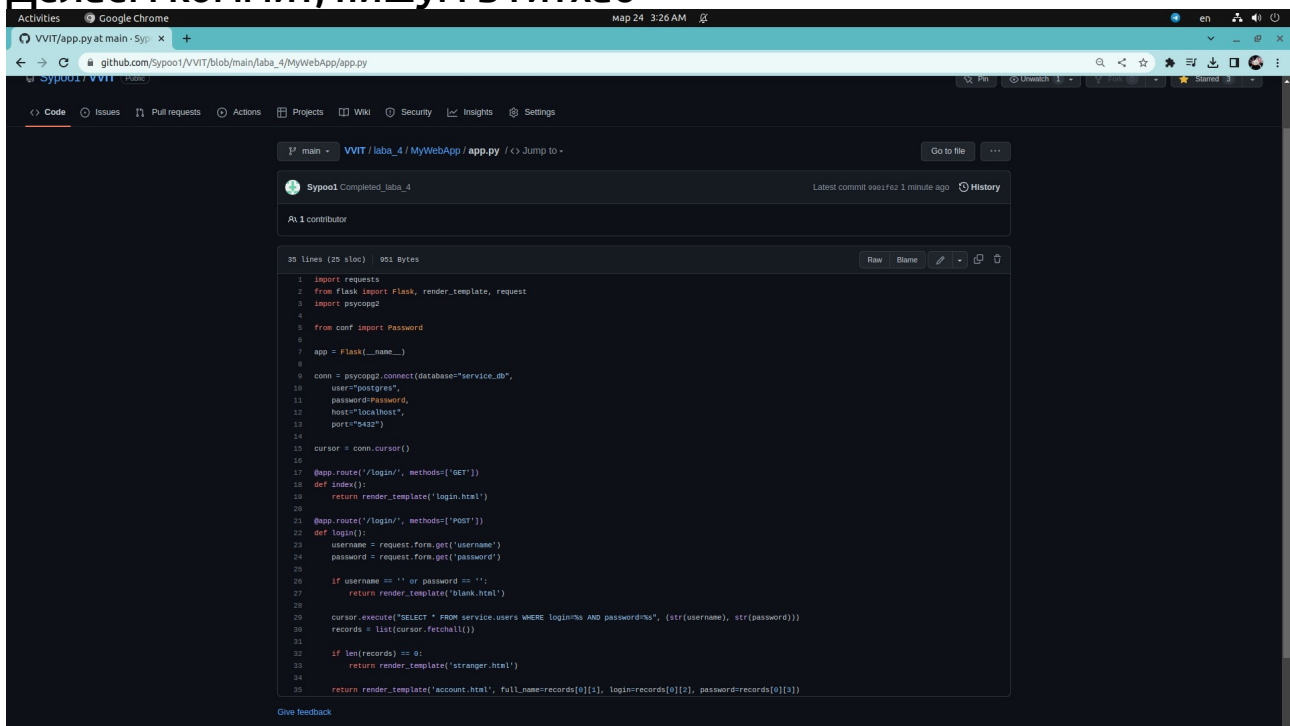
Wrong login or password!



```
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitignore
        new file:   laba_4/MyWebApp/app.py
        new file:   laba_4/MyWebApp/req.txt
        new file:   laba_4/MyWebApp/templates/account.html
        new file:   laba_4/MyWebApp/templates/blank.html
        new file:   laba_4/MyWebApp/templates/login.html
        new file:   laba_4/MyWebApp/templates/stranger.html

(venv) sypoo@sypoo-desktop:~/VVIT$ git commit -m "Completed_laba_4"
[main 9901f62] Completed_laba_4
 7 files changed, 380 insertions(+)
 create mode 100644 laba_4/MyWebApp/app.py
 create mode 100644 laba_4/MyWebApp/req.txt
 create mode 100644 laba_4/MyWebApp/templates/account.html
 create mode 100644 laba_4/MyWebApp/templates/blank.html
 create mode 100644 laba_4/MyWebApp/templates/login.html
 create mode 100644 laba_4/MyWebApp/templates/stranger.html
(venv) sypoo@sypoo-desktop:~/VVIT$ git push
Username for 'https://github.com': Sypoo
Password for 'https://Sypoo@github.com':
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (12/12), 3.24 KiB | 3.24 MiB/s, done.
Total 12 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/Sypoo1/VVIT.git
   811af78..9901f62  main -> main
(venv) sypoo@sypoo-desktop:~/VVIT$
```

## Делаем коммит, пишу в гитхаб



```
1 import requests
2 from flask import Flask, render_template, request
3 import psycopg2
4
5 from conf import Password
6
7 app = Flask(__name__)
8
9 conn = psycopg2.connect(database="service_db",
10 user="postgres",
11 password=Password,
12 host="localhost",
13 port="5432")
14
15 cursor = conn.cursor()
16
17 @app.route('/login/', methods=['GET'])
18 def index():
19     return render_template('login.html')
20
21 @app.route('/login/', methods=['POST'])
22 def login():
23     username = request.form.get('username')
24     password = request.form.get('password')
25
26     if username == '' or password == '':
27         return render_template('blank.html')
28
29     cursor.execute("SELECT * FROM service.users WHERE login%3A AND password%3A", (str(username), str(password)))
30     records = list(cursor.fetchall())
31
32     if len(records) == 0:
33         return render_template('stranger.html')
34
35     return render_template('account.html', full_name=records[0][1], login=records[0][2], password=records[0][3])
```

[https://github.com/Sypoo1/VVIT/tree/main/laba\\_4/MyWebApp](https://github.com/Sypoo1/VVIT/tree/main/laba_4/MyWebApp)

**Вывод: Прodelав данную работу, я научился создавать авторизацию на вебсайте, подключаться к базе данных и создавать html странички**