**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 «Математической кибернетики и информационных технологий»

**Отчет по лабораторной работе №4**

по дисциплине «Введение в информационные технологии» на тему:

Создание системы авторизации в веб-приложении

Выполнил:

Сидорук Данил Вадимович, БФИ-2202

Проверил:

Камиль Раисович Харрасов

Москва

2022

**1 Постановка задачи**

Цель: получить навыки работы с библиотекой flask, языком разметки html, языком разметки шаблонизатора jinja2

**Задание**

1. Создать директорию проекта

2. Открыть директорию в PyCharm

3. Установить необходимые для дальнейшей работы инструменты

4. Создать приложение

5. Установить PostgreSQL

6. Создать базу данных

7. Модернизировать приложение

8. Проверить работу приложения

**Домашнее задание**

1. Дополнить таблицу users в базе данных до 10 пользователей

2. Сделать обработку исключения на ввод пустого логина и пароля

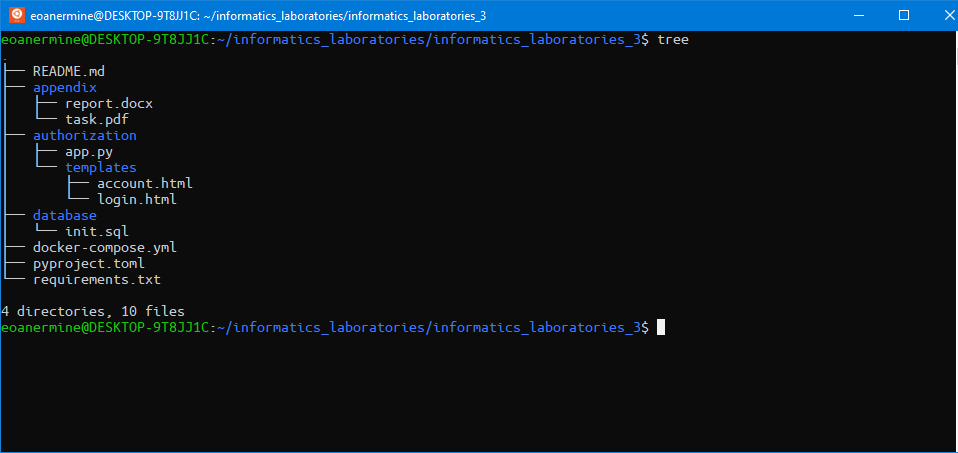
3. Сделать обработку исключения на отсутствие пользователя в базе данных

4. Вывести на странице аккаунта помимо имени пользователя его логин и пароль

**2 Результат выполненной работы**

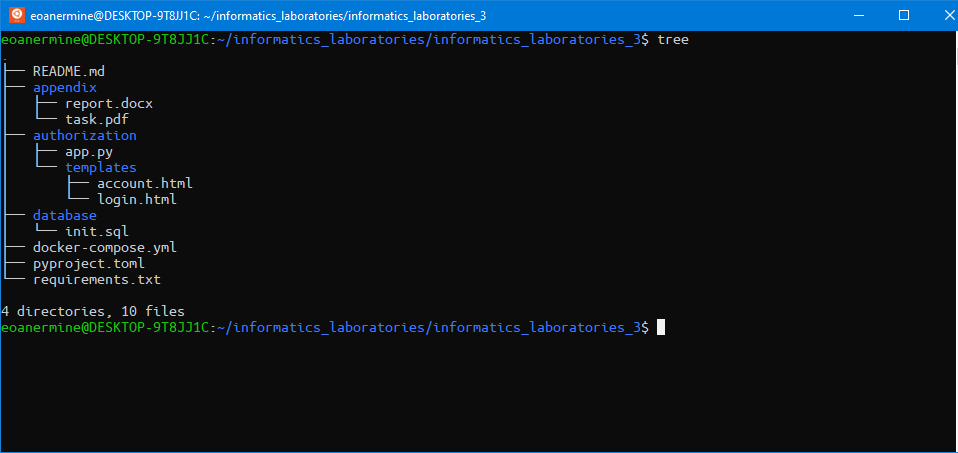
**Задание 1**

Новая директория была создана с помощью команды mkdir, на рисунке ниже приведена ее иерархия:

Рисунок 1.1 — иерархия проекта

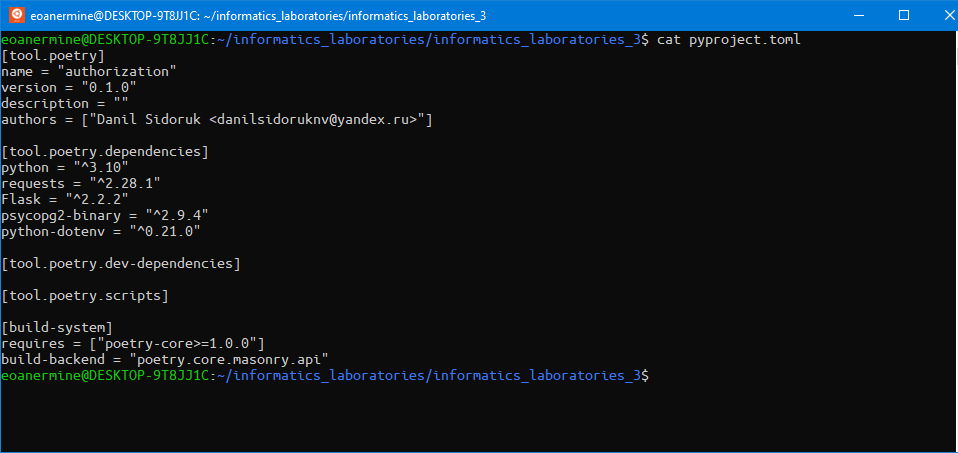
**Задание 2**

Директория была открыта с помощью команды cd, на рисунке ниже приведена ее структура:

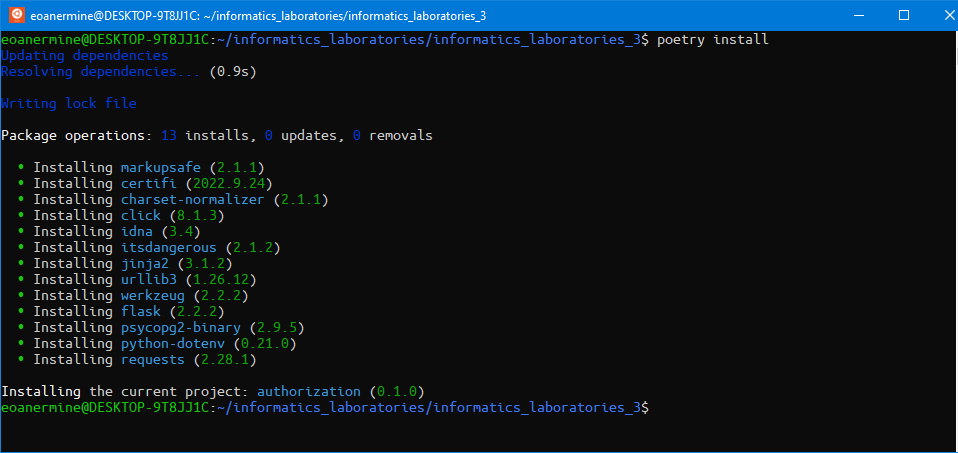
Рисунок 2.1 — иерархия директории проекта

**Задание 3**

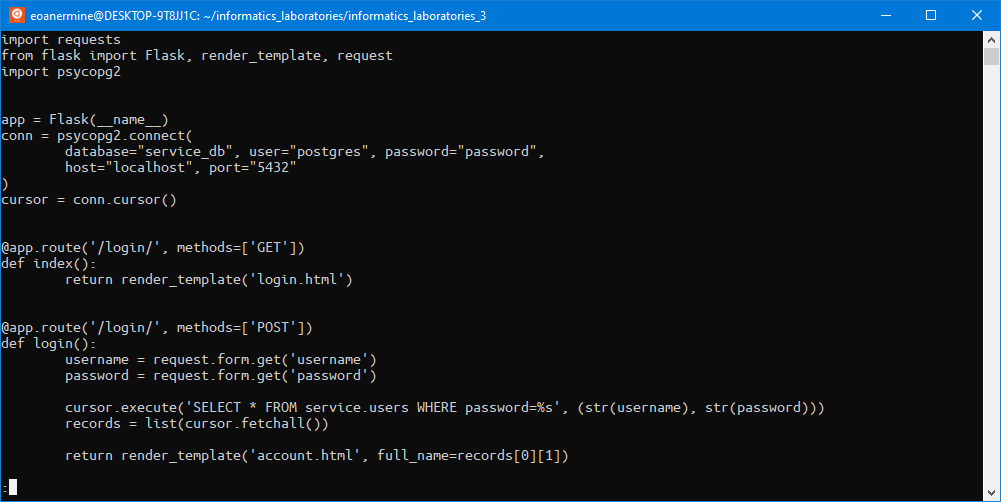
Все зависимости были зафиксированы в pyproject.toml:

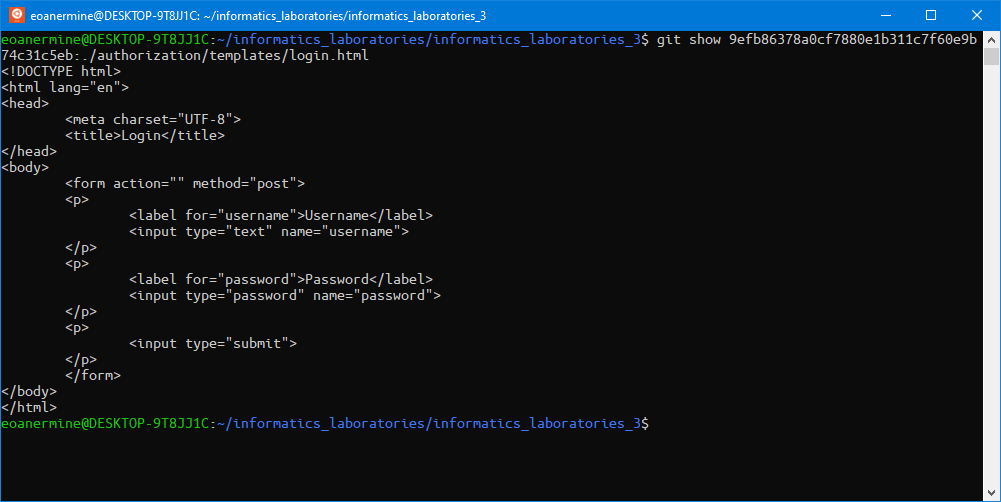
Рисунок 3.1 — фиксация зависимостей в pyproject.toml

И установлены с помощью команды poetry install:

Рисунок 3.2 — установка зависимостей

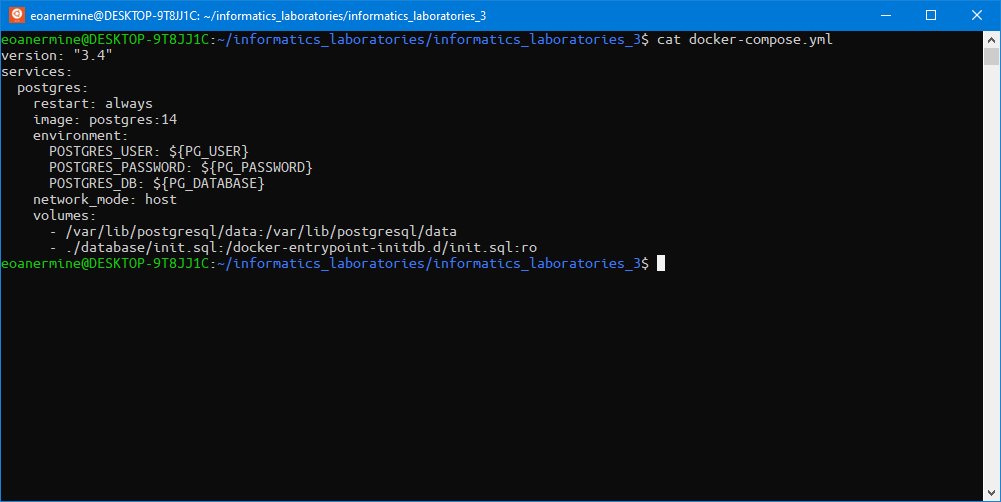
**Задание 4**

Рисунок 4.1 — листинг app.py

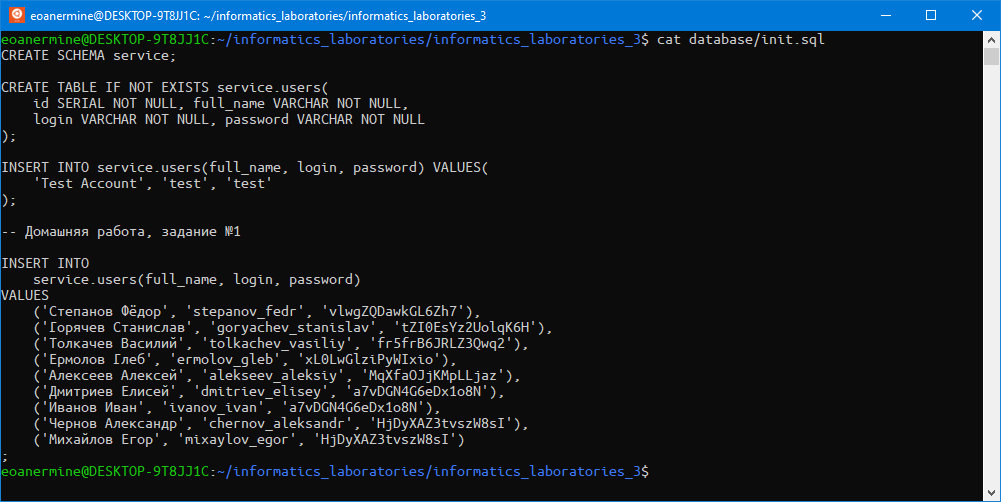
Рисунок 4.2 — листинг login.html

**Задание 5**

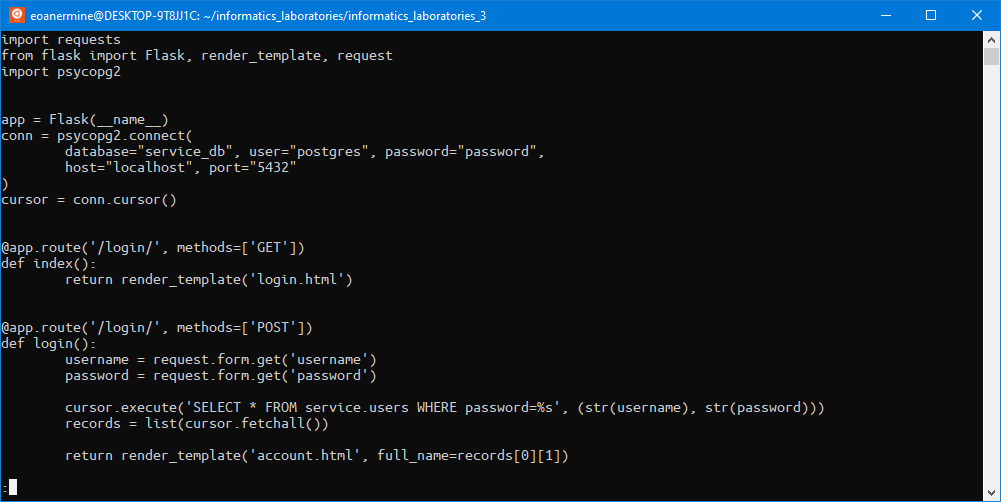
Развертывание сервера PostgreSQL было организовано с помощью docker compose:

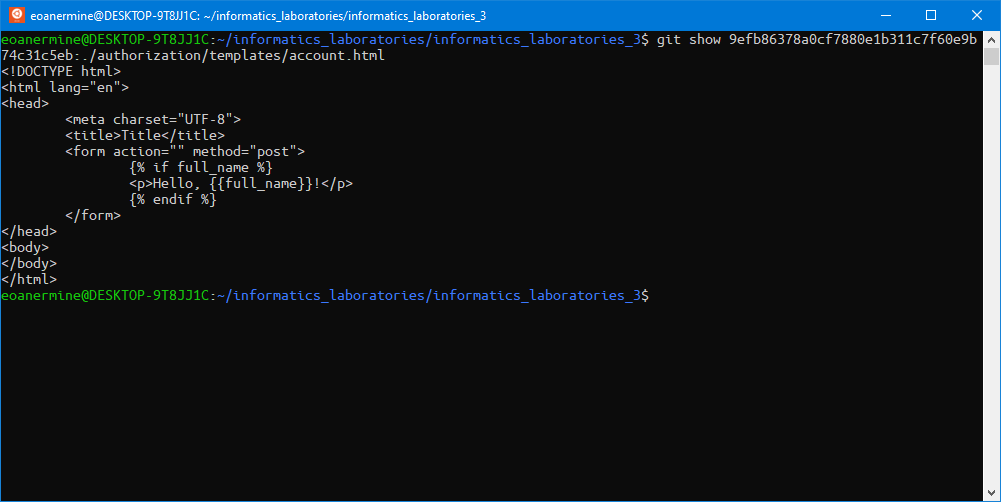
Рисунок 5.1 — листинг docker-compose.yml

**Задание 6**

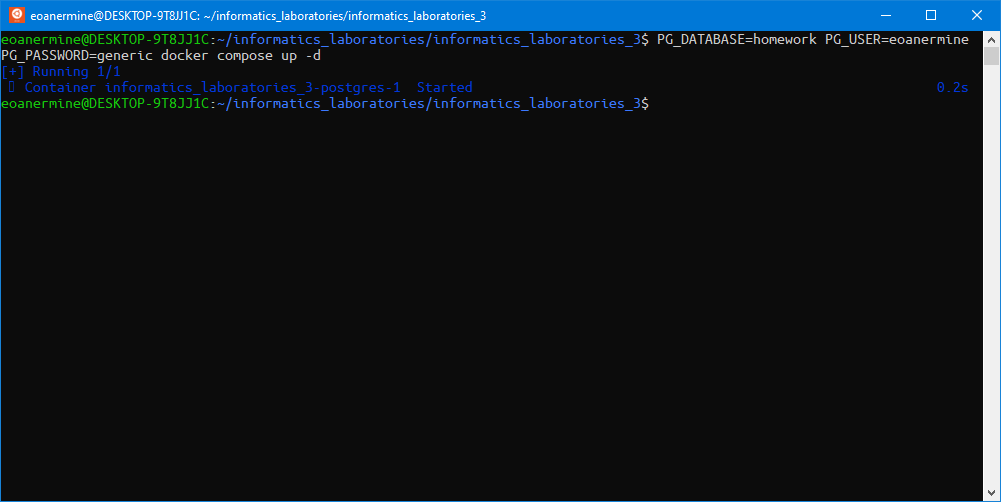
Рисунок 6.1 — создание базы данных

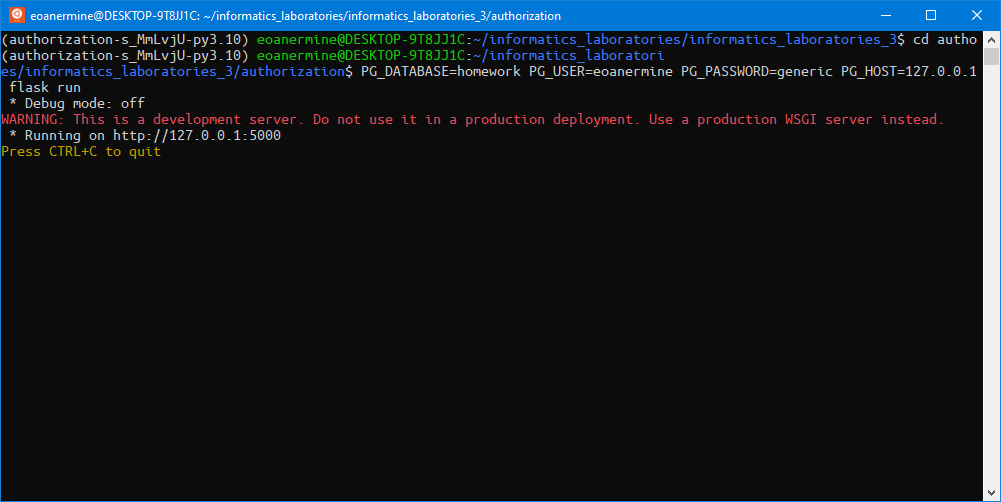
**Задание 7**

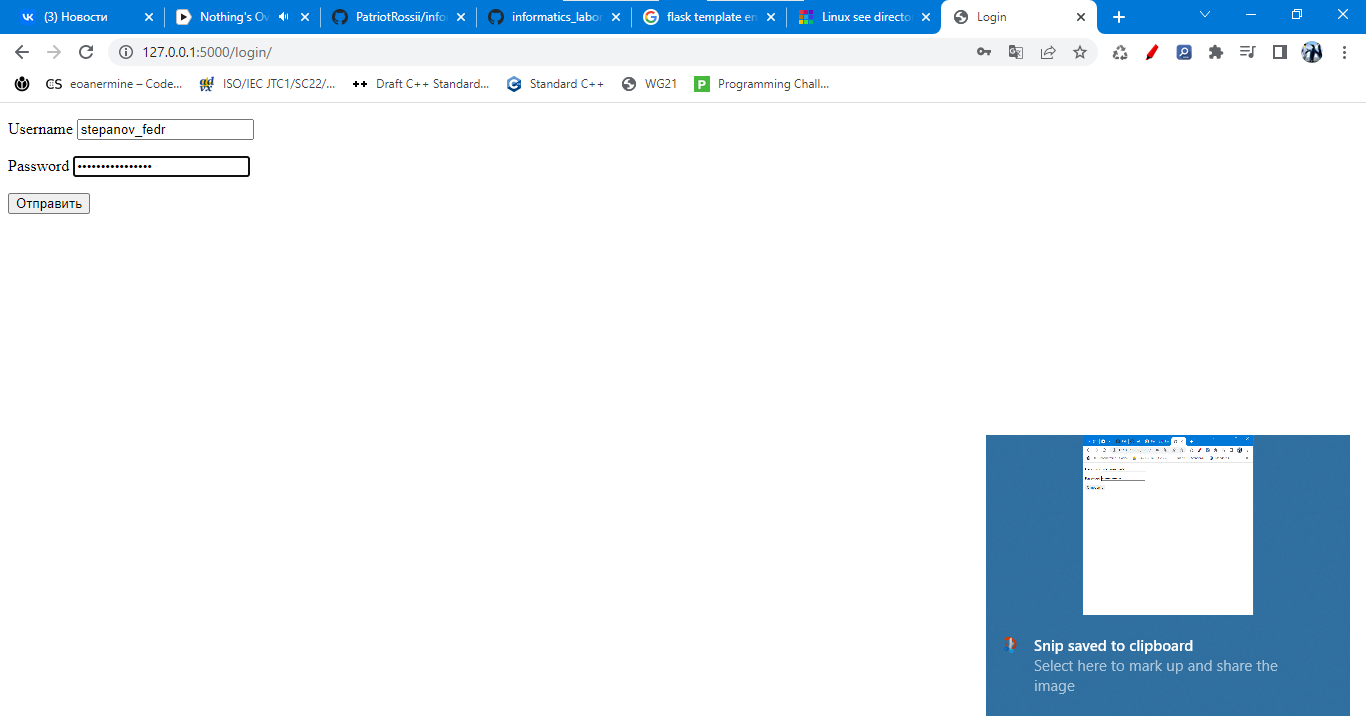
Рисунок 7.1 — листинг app.py

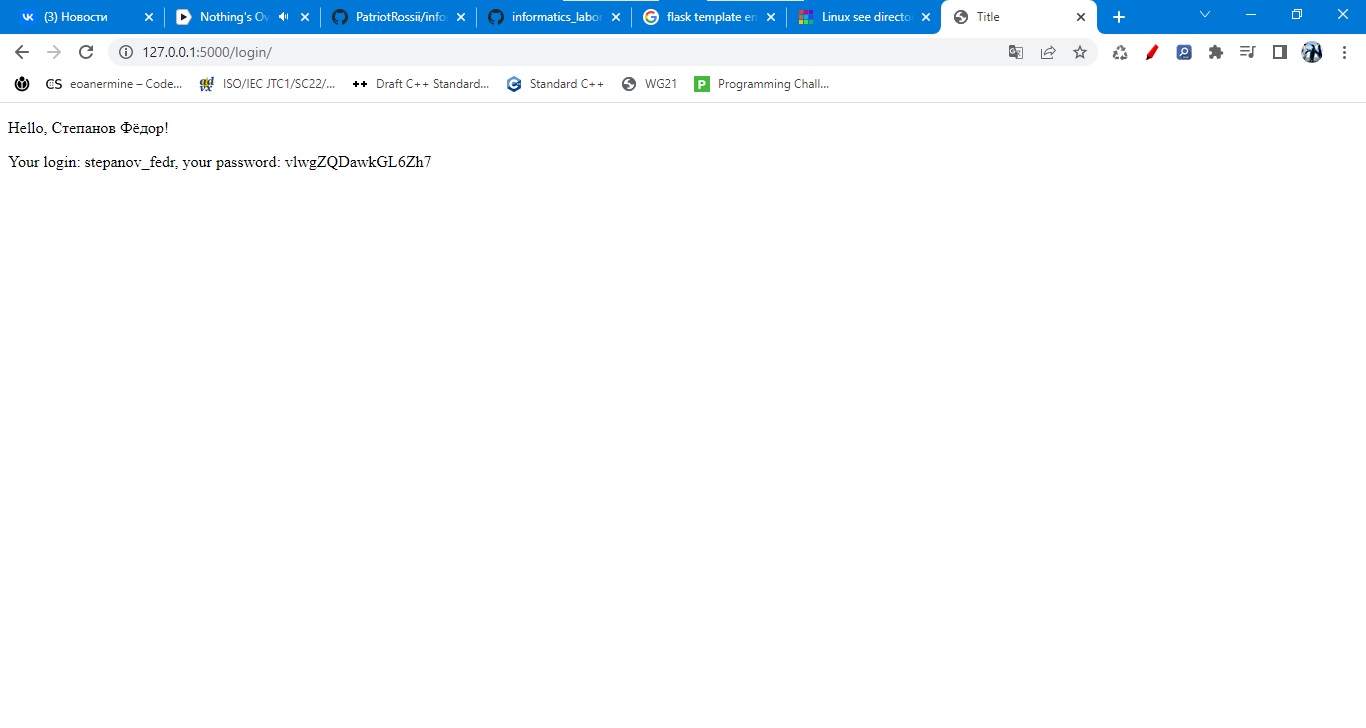
Рисунок 7.2 — листинг account.html

**Задание 8**

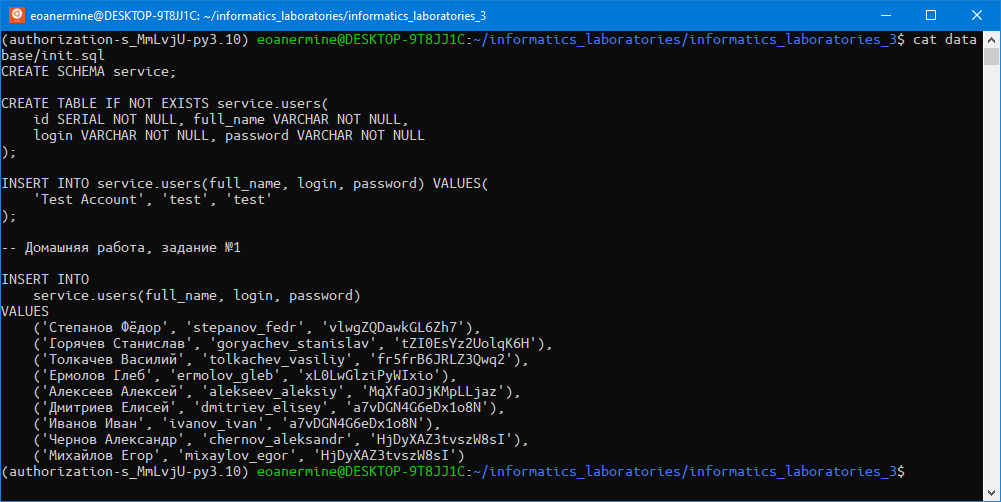
Рисунок 8.1 — поднятие базы данных

Рисунок 8.2 — запуск приложения

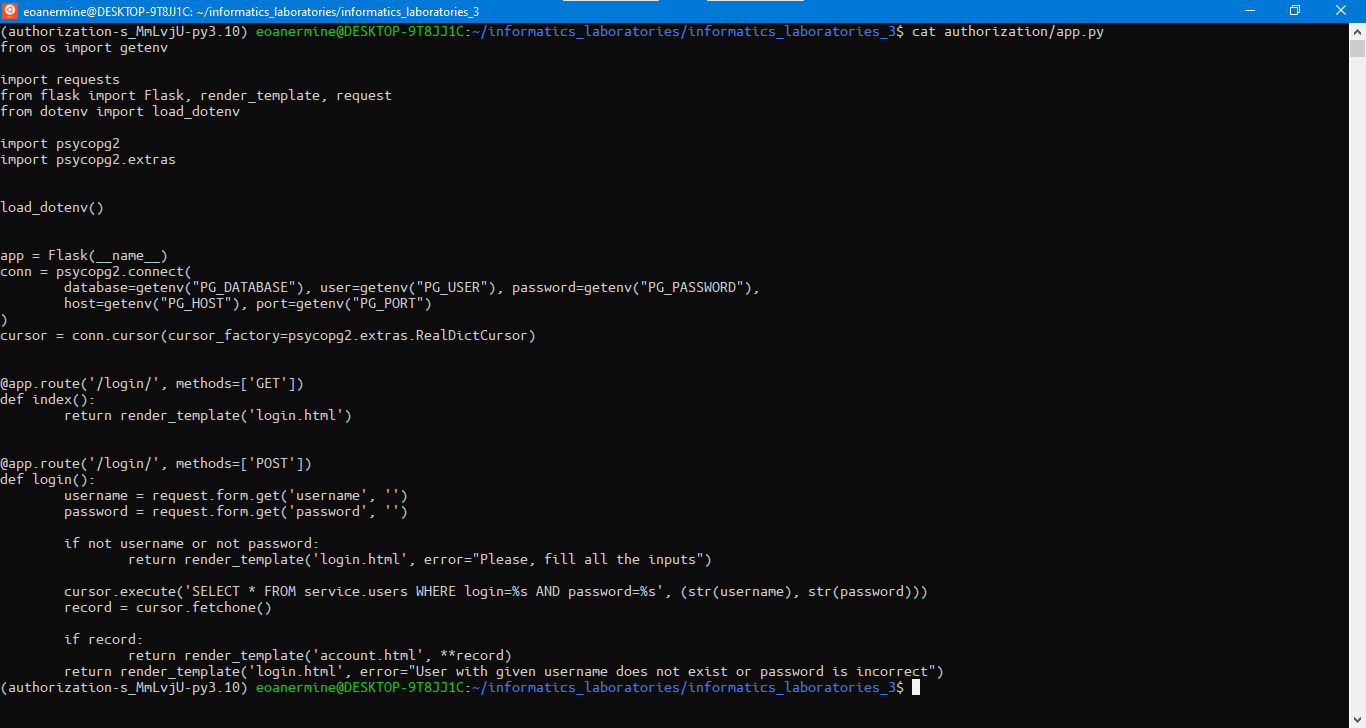
Рисунок 8.3 — ввод данных

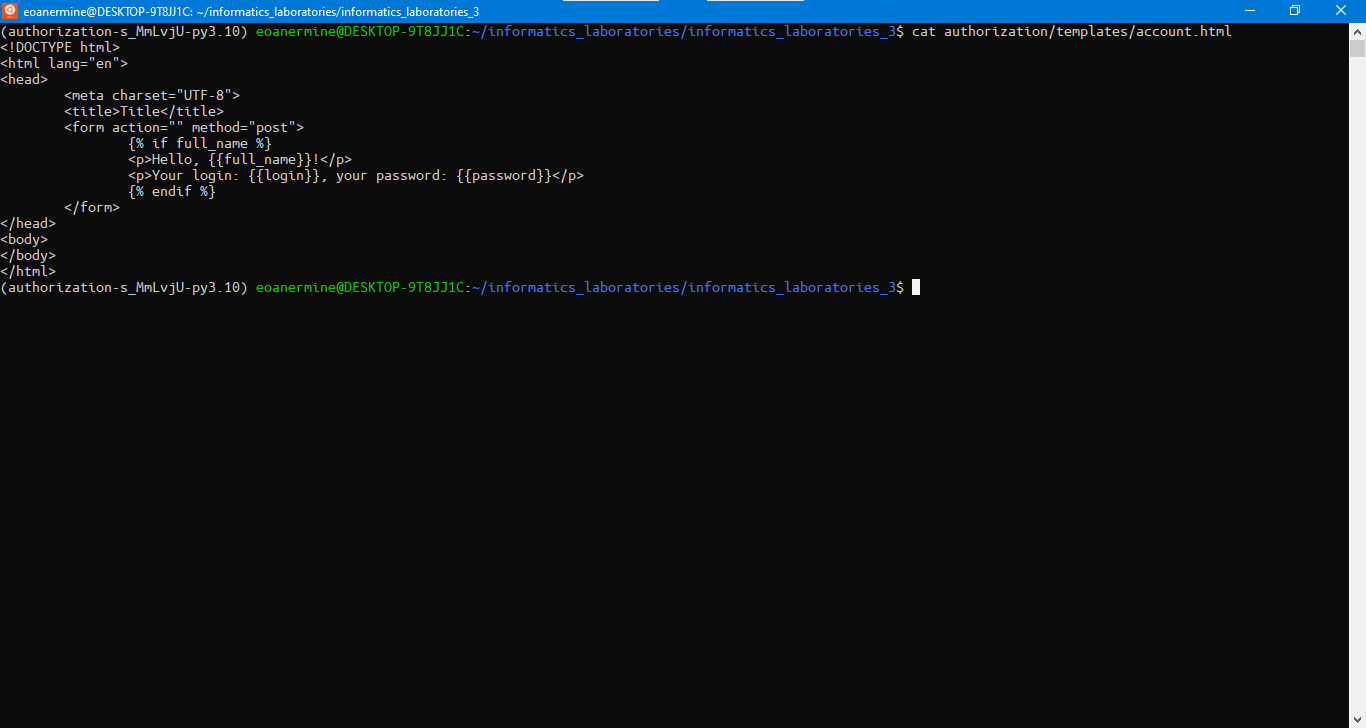
Рисунок 8.4 — результат авторизации

**Домашнее задание 1**

Рисунок 9.1 — создание тестовых пользователей

**Домашнее задание 2, 3**

Рисунок 10.1 — код обработки исключений

Рисунок 10.2 — вывод логина и пароля на странице аккаунта

**3 Вывод**

В результате проделанной работы мы получили навыки работы с библиотекой flask, языком разметки html, языком разметки шаблонизатора jinja2