**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ МАССОВЫХ**

**КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Индивидуальный проект

по дисциплине: «Введение в информационные технологии»

«Создание системы авторизации в веб-приложении».

Выполнил студент группы БИН2308

Герасимов.И.Д

Проверил преподаватель:

Казиева Т.В

Москва

2023

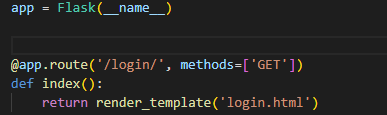
Цель работы : Создать веб-приложение, в котором пользователь будет иметь возможность авторизироваться, и так как наш код будет связан с базой данных, то пользователь будет либо успешно авторизироваться, либо будет получать сообщение с ошибкой.

Ход работы :

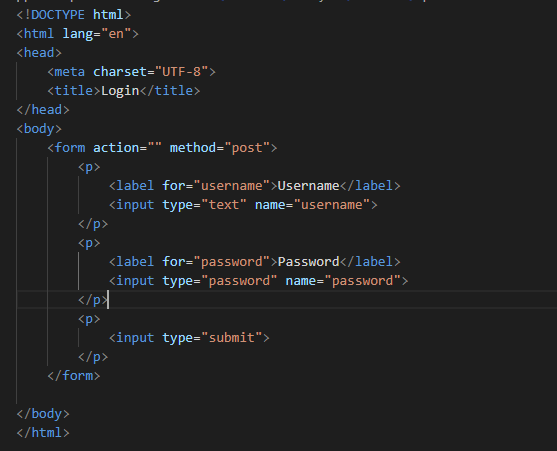
ШАГ 1. Импортируем все необходимые инструменты для создания приложения.



ШАГ 2. Создаём приложение через Flask



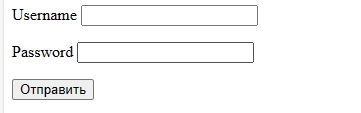
ШАГ 3. Создаём папку templates и внутри неё создаём файл login.html

ШАГ 4. После создания файла login.html прописываем в командной строке команду :

flask run и заходим на сайт.

ШАГ 5. Вот что мы должны увидеть.



ШАГ 6. Заходим в PostgreSQL и создаём базу данных, используя запрос :

CREATE DATABASE service\_db;

ШАГ 7. Создаём схему, используя запрос :

CREATE SCHEMA service;

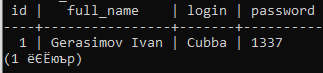
ШАГ 8. Создаём таблицу пользователей, используя запрос :

CREATE TABLE service.users(id SERIAL NOT NULL, full\_name VARCHAR NOT NULL, login VARCHAR NOT NULL, password VARCHAR NOT NULL);

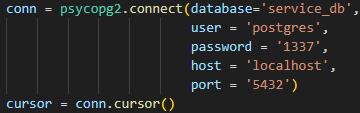
ШАГ 9. Добавим в таблицу одного пользователя, используя запрос :

INSERT INTO service.users(full\_name,login,password) VALUES (‘Gerasimov Ivan’,’Cubba’,’1337’);

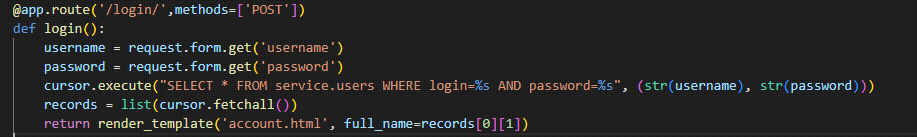
ШАГ 10. Получаем результат :



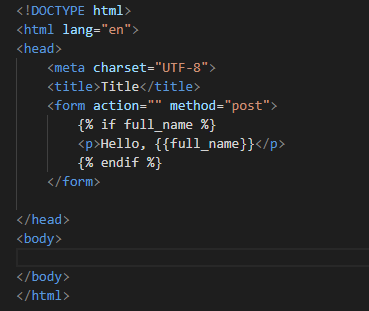
ШАГ 11. Подключаем наше приложение к базе данных.



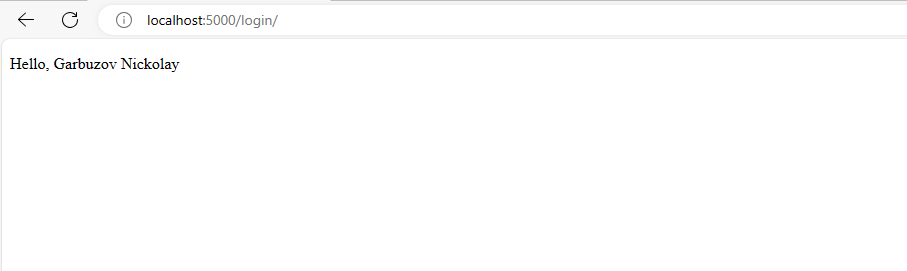
ШАГ 12. Создадим второй декоратор, который после получения никнейма и пароля будет перенаправлять нас на страницу аккаунта.



ШАГ 13. Создадим файл account.html, на котором будет страница нашего аккаунта. Перекидывать на него будет наш декоратор из предыдущего шага.



ШАГ 14. Вводим данные на сайте и получаем итоговый результат.



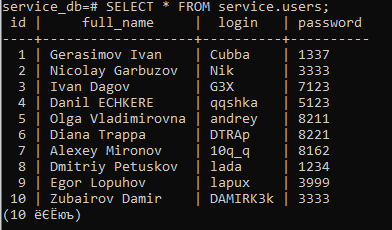
Домашняя работа.  
1. Дополнить таблицу users в базе данных до 10 пользователей.

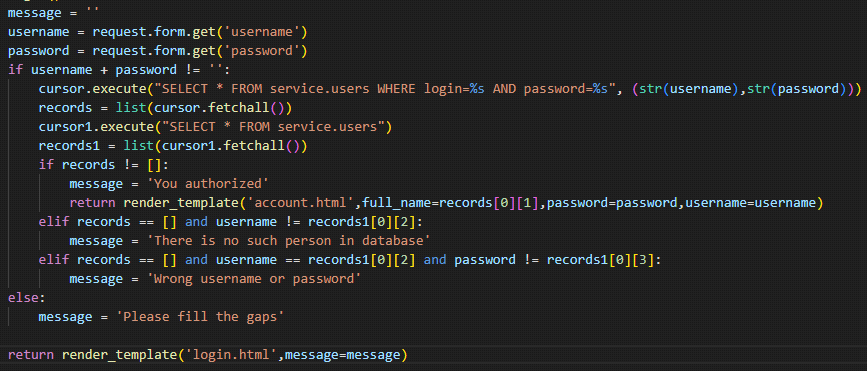
2. Сделать обработку исключения на ввод пустого логина и пароля.

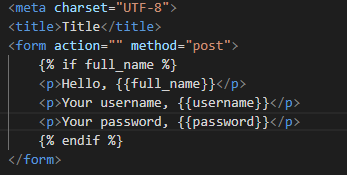
3. Сделать обработку исключения на отсутствие пользователя в базе данных.

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

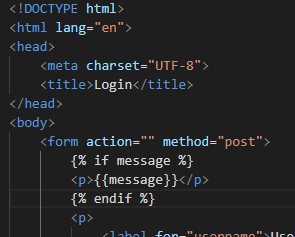
* Через функцию INSERT INTO, добавим ещё 9 пользователей в таблицу users. Получаем результат



* Проверим простым условием. Если никнейм + пароль составляют пустоту, то нужно вывести ошибку : Пожалуйста заполните поля. Если никнейм + пароль не составляют пустоту, то тогда программа продолжает работать.
* 
* На предыдущем скрине видно в условии elif, что я проверяю 2 раза условие records == [], так как при первом условии, пользователя попросту нет в базе данных (идёт проверка в индексах переменной records ), а при втором условии проверяется пароль при существующем пользователе. Если пользователь существует, но пароль неверный – вылезает одна ошибка. А если пользователя нет в базе данных – вылезает совсем другая ошибка.
* Нужно дополнить файл account.html



! ЧТОБЫ СООБЩЕНИЕ ОБ ОШИБКЕ ВЫЛЕЗАЛО, НУЖНО ОБЯЗАТЕЛЬНО ДОПОЛНИТЬ ФАЙЛ login.html



Вывод : В течение всей это работы мне понадобились навыки Python и SQL запросов. Я смог подключиться к своей созданной базе данных, и после запуска кода и ввода данных пользователем, программа начинала обрабатывать их. На основе этих данных пользователь либо успешно авторизировался, либо получал сообщение с соответствующей ошибкой.