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» (МТУСИ)**

Кафедра «Системное программирование»

**ОТЧЕТ**

по лабораторной работе № 4 дисциплина **«Введение в информационные технологии»**

Выполнил: Студент гр.БВТ2207

Кузьмин И.А.

“ ” 2023г.

Проверил:

“ ” 2023г.

Москва, 2023

# Введение

# Цель работы:

Создание системы авторизации в веб приложении.

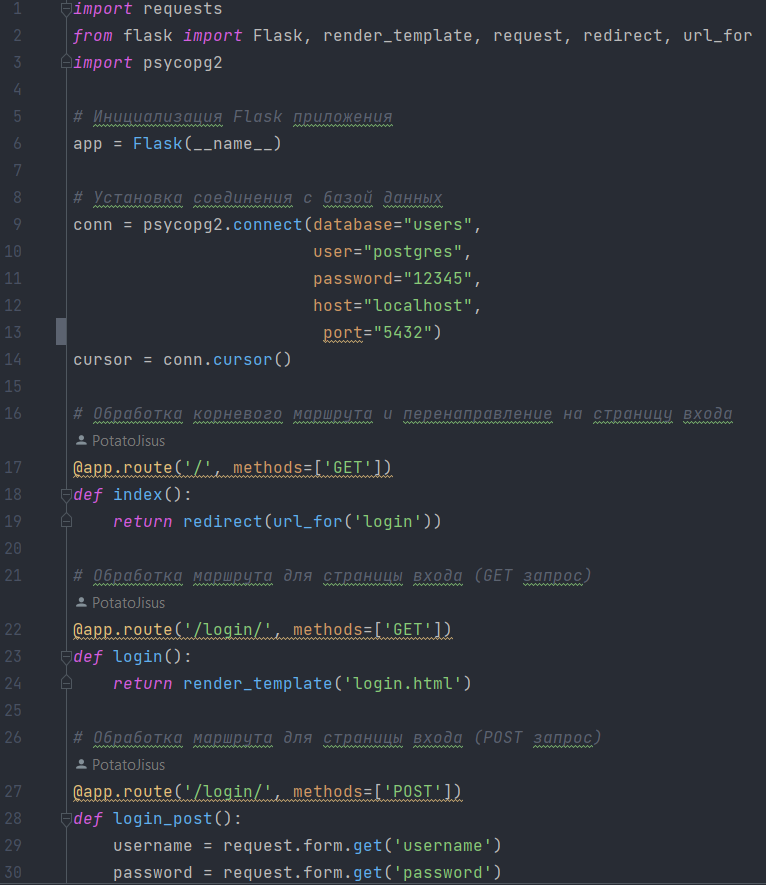
# Задание:

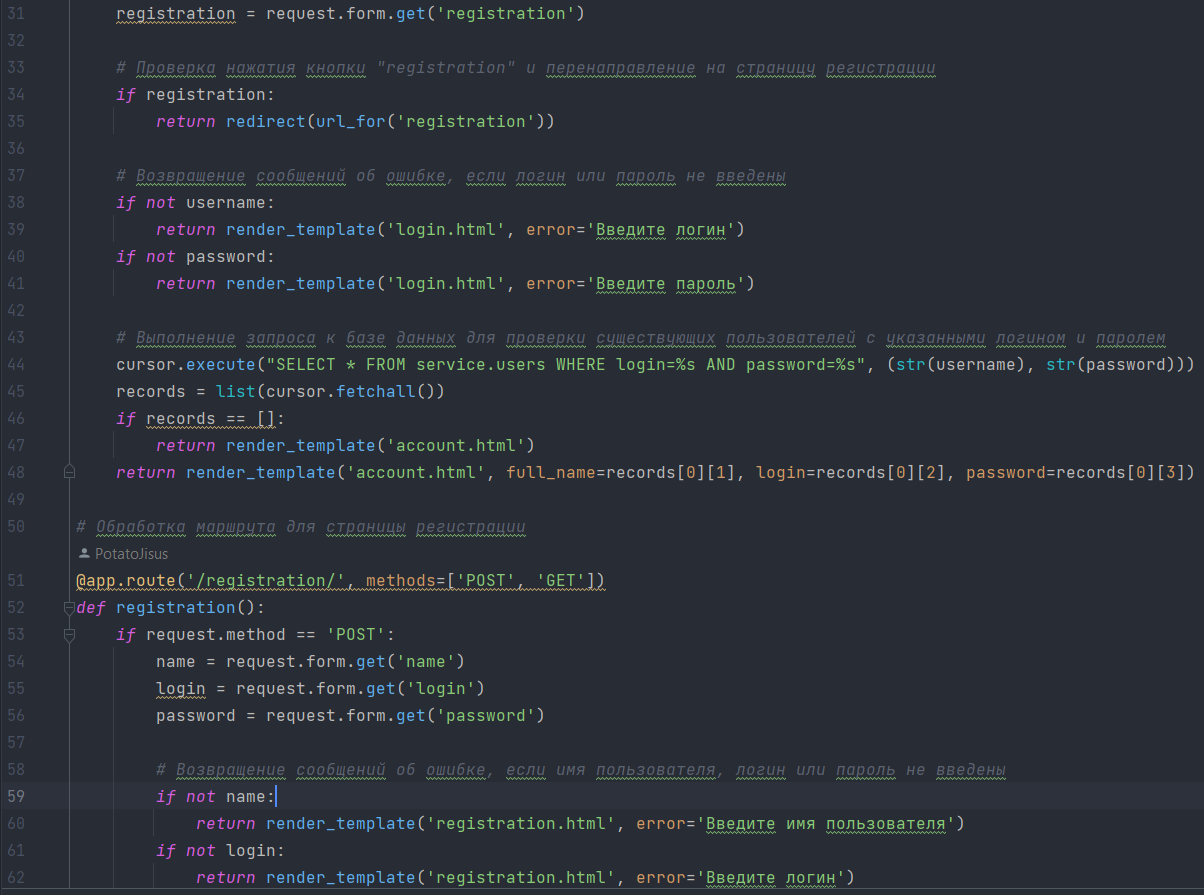
.

# Реализация

В процессе выполнения лабораторной работы использованы следующие информационные источники [1-2].

**Практическая реализация**





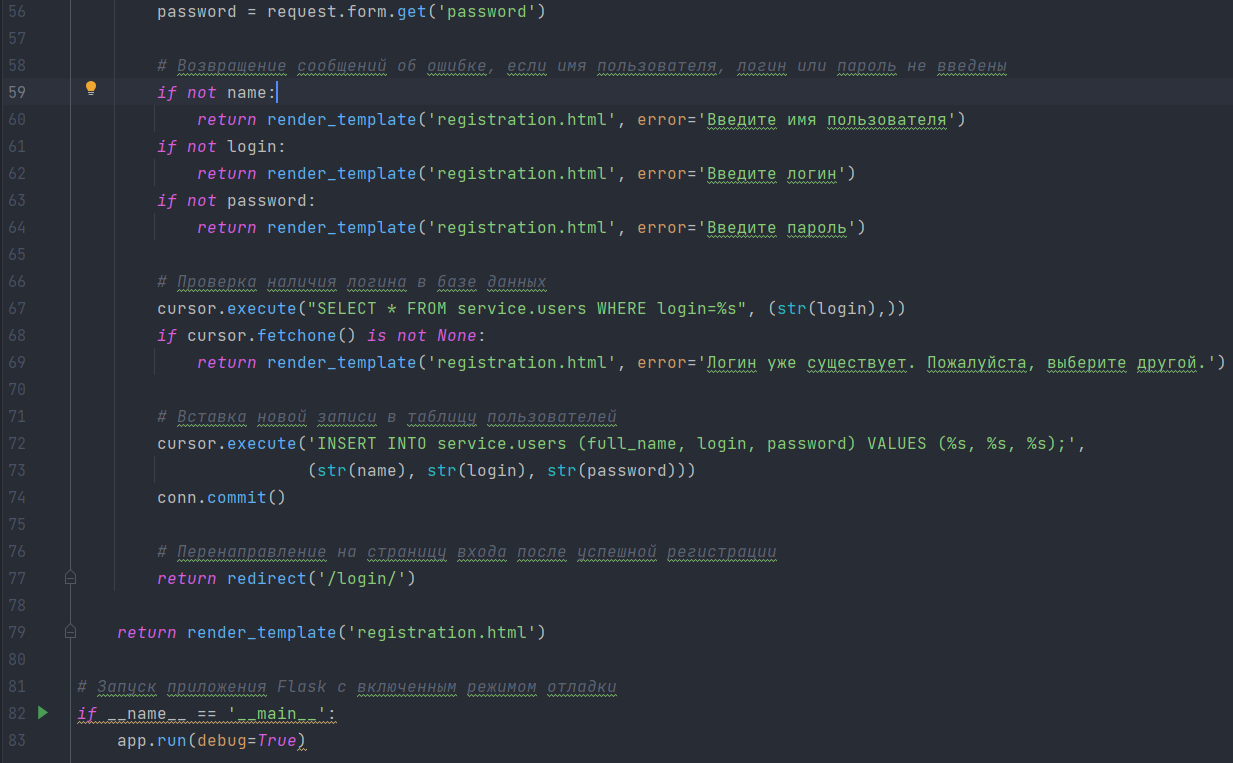


Рис. 1 – код приложения

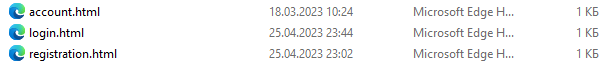
Так же у нас имеется папка templates, в которой содержится 3 файла:  


Рис. 2 – templates

Этот код представляет собой простое веб-приложение на основе Flask, которое обрабатывает запросы пользователя и взаимодействует с базой данных PostgreSQL.

Вот пошаговое объяснение кода:

1. Импортирование необходимых модулей:

- `requests` - модуль для выполнения HTTP-запросов (не используется в этом коде).

- `Flask` - класс, предоставляемый Flask для создания и настройки веб-приложений.

- `render\_template` - функция Flask для рендеринга HTML-шаблонов.

- `request` - объект Flask, содержащий информацию о текущем HTTP-запросе.

- `redirect` - функция Flask для перенаправления пользователя на другую страницу.

- `url\_for` - функция Flask для построения URL-адреса именованного маршрута.

- `psycopg2` - модуль для взаимодействия с базой данных PostgreSQL.

2. Инициализация Flask приложения:

- Создается экземпляр класса `Flask` и сохраняется в переменной `app`.

3. Установка соединения с базой данных:

- Используя модуль `psycopg2`, устанавливается соединение с базой данных PostgreSQL. Указываются параметры подключения, такие как имя базы данных, имя пользователя, пароль, хост и порт.

- Создается курсор для выполнения SQL-запросов на базе данных.

4. Обработка корневого маршрута и перенаправление на страницу входа:

- Создается маршрут `'/'` с методом GET.

- Функция `index()` перенаправляет пользователя на страницу входа с помощью функции `redirect()` и `url\_for()`.

5. Обработка маршрута для страницы входа (GET запрос):

- Создается маршрут `'/login/'` с методом GET.

- Функция `login()` возвращает шаблон `'login.html'`, который будет отображаться пользователю при запросе этого маршрута.

6. Обработка маршрута для страницы входа (POST запрос):

- Создается маршрут `'/login/'` с методом POST.

- Функция `login\_post()` извлекает данные из формы (логин, пароль и флаг регистрации).

- Если пользователь нажал кнопку "registration", он будет перенаправлен на страницу регистрации.

- Если логин или пароль не были введены, будет отображено сообщение об ошибке.

- Выполняется SQL-запрос к базе данных для проверки существующих пользователей с указанными логином и паролем.

- Если такой пользователь найден, информация о нем передается в шаблон `'account.html'`, иначе отображается шаб

лон `'account.html'` без информации о пользователе.

7. Обработка маршрута для страницы регистрации:

- Создается маршрут `'/registration/'` с методами POST и GET.

- Если метод запроса - POST, извлекаются данные из формы (имя пользователя, логин, пароль).

- Если не заполнены одно или несколько полей, отображается сообщение об ошибке.

- Выполняется SQL-запрос к базе данных для проверки наличия логина.

- Если логин уже существует, отображается сообщение об ошибке.

- Если все данные корректны, выполняется SQL-запрос для вставки новой записи в таблицу пользователей.

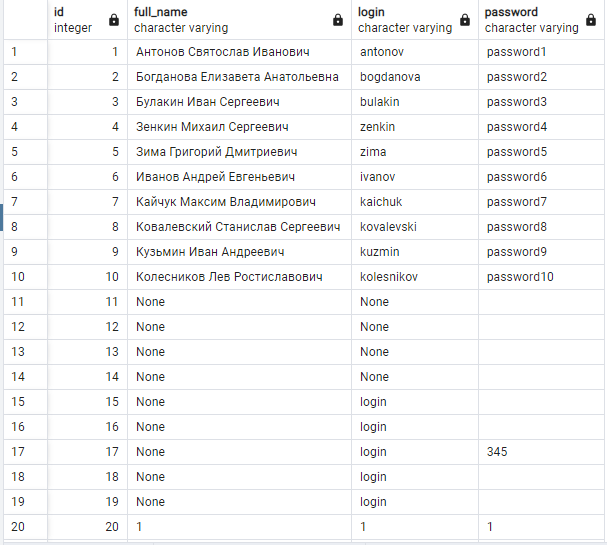
- После успешной регистрации пользователь перенаправляется на страницу входа.

8. Запуск приложения Flask с включенным режимом отладки:

- Если файл выполняется как самостоятельный скрипт, то запускается веб-сервер Flask для обслуживания приложения.

- Устанавливается режим отладки, чтобы выводить подробную информацию об ошибках при разработке.

Таким образом, код представляет собой простое веб-приложение для аутентификации и регистрации пользователей, использующее базу данных PostgreSQL для хранения и проверки информации о пользователях.

  
Рис. 2 – таблица users

# Заключение

# Список использованных источников