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 «Информатика»

Реферат по теме

**«Примеры скриптов регистрации и авторизации на сайте с сохранением данных.»**

**по дисциплине**

**«Web-программирование»**

Выполнил: студент гр. БЭИ2202 Кулешов А. С.

Москва, 2022 г

**Вёрстка форм**

Для начала сделаем простой сайт для авторизации. Странички входа и регистрации соответственно. Исключая дизайн страницы нужно сделать формы. Смотри на рисунки ниже.

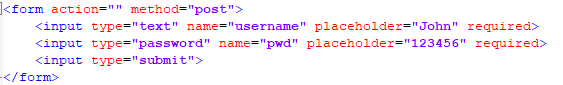


Рисунок 1 – html вёрстка формы регистрации

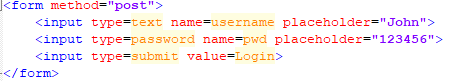


Рисунок 2 – html вёрстка формы логина

Добавим во всё это css и получим примерно следующие

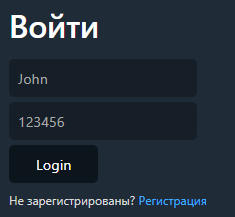


Рисунок 3 – примерный вид формы в браузере

**Backend\\Сервервная логика. Регистрация**

Теперь остаётся лишь самое сложное – выбрать фреймворк для бэкэнда сайта. Отличным вариантом для примера будет библиотека flask для Python. Простой для понимания, легко скачать, отличный вариант. В качестве базы данных использую модуль shelve.

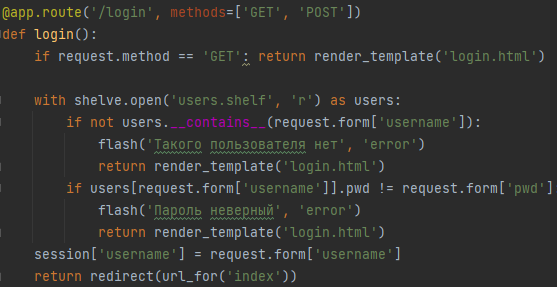


Рисунок 4 – backend логина

Примерно похожая логика и для регистрации



Рисунок 5 – backend регистрации

Остаётся лишь проверить функционал



Рисунок 6 – логин при отсутствующем пользователе



Рисунок 7 – успешная регистрация\\логин



Рисунок 8 – попытка логина под аккаунт с неверным паролем



Рисунок 9 – попытка создания аккаунта с имеющимся в базе данных именем

**Backend\\Сервервная логика. Функционал для авторизированных**

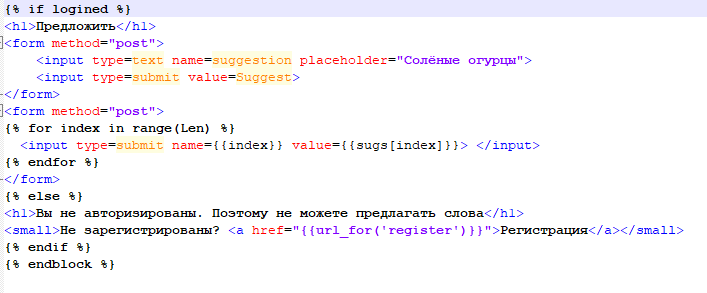


Рисунок 10 – html-заготовка для страницы с предложением слов

Я решил воссоздать игру «крокодил», в которой участники могут предлагать свои слова. Слова будут предлагаться самими пользователями, поэтому если пользователь не авторизирован, то тогда его оповестят о том, что нужно зарегистрироваться. Flask позволяет создавать сайты-заготовки, таким образом создавать сложную логику по заполнению html-документа со стороны сервера не надо. «Прототипирование» - главное преимущество flask.

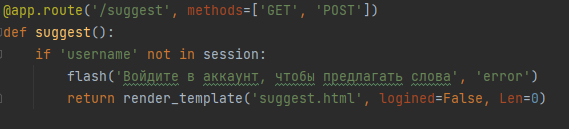


Рисунок 11 – python код предложений

Я опущу сложные детали добавления и проверки предложений, ведь сейчас речь идёт не об этом. С точки зрения авторизации - это весь код. Теперь проверим его.

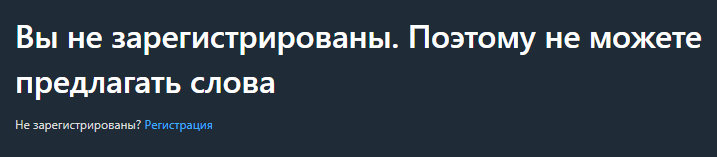


Рисунок 12 – попытка сделать предложение буду не зарегистрированным

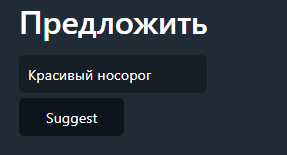


Рисунок 13 – форма для предложений доступна зарегистрированным пользователям