Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Национальный исследовательский Университет ИТМО»

Факультет безопасности информационных технологий

Дисциплина:

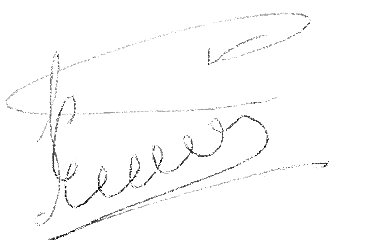
«Разработка систем аутентификации и криптографии»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2 Разработка системы парольной аутентификации в клиент-сервером приложении**

**«Вариант 2 - Реализация аутентификации по паролю с подтверждением по email»**

**Выполнили:**

Магистрант гр. N42514c Э.Р. Кочкаров



Санкт-Петербург

2020 г.

* **Цель работы (задача)** – реализация механизма аутентификации в клиент-серверном веб-приложении
* Требования к реализации программы:
* необходимо реализовать метод аутентификации в клиент-серверном приложении согласно варианту
* клиент должен представлять собой веб-страницу с формой авторизации пользователя
* сервер должен включать в себя две части:

1) таблица идентификаторов (данные о пользователях для аутентификации: логин/пароль/токен/и т. д. в зависимости от метода аутентификации);

2) процесс с реализованной логикой метода аутентификации.

1. **Описание выбранных средств реализации и обоснования выбора**

Для реализации алгоритма был выбран язык Python, так как к скорости выполнения программы не предъявлялось каких-либо особых требований.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Этот язык имеет довольно простой синтаксис, лёгок в изучении и имеет низкий порог вхождения. В то же время язык является очень мощным средством разработки. Такой выбор также обусловлен тем, что это достаточно распространённый язык в настоящее время.

Для реализации клиент-серверной архитектуры был выбран фреймворк Flask.

Flask — это упрощенная платформа Python для веб-приложений, которая обеспечивает основные возможности маршрутизации URL-адресов и визуализации страниц.

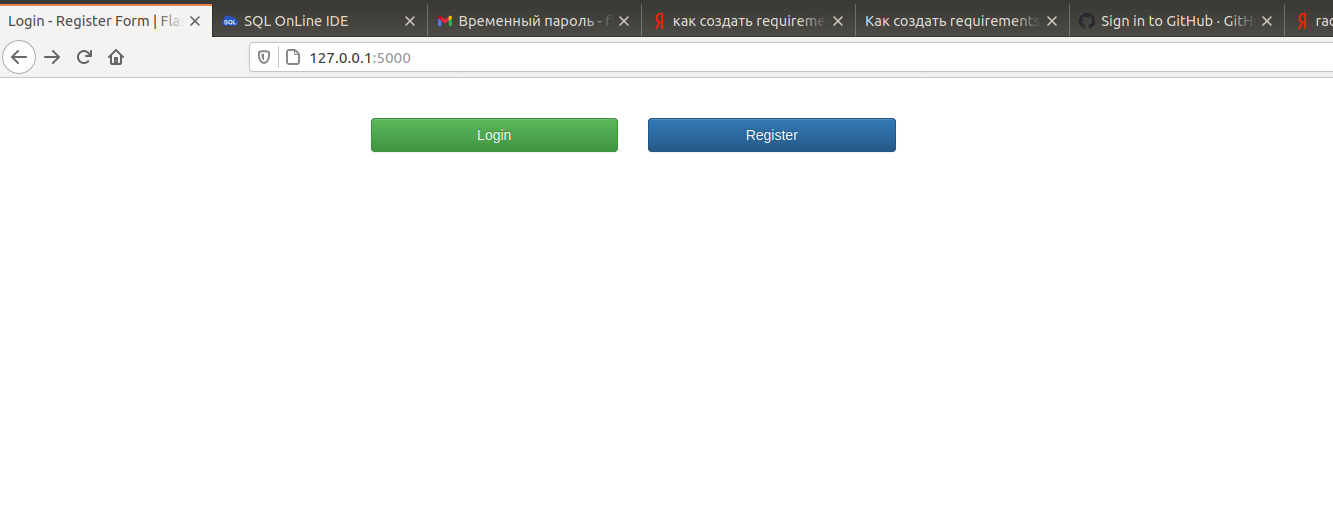
1. **Описание алгоритма.**

Реализовать аутентификацию по паролю с подтверждением по email. В таблице идентификаторов должны храниться: логин, email, пароль, временный код подтверждения. Таблица идентификаторов должна представлять собой таблицу в реляционной БД, данные должны передаваться через SQL-запросы. При аутентификации на сервере сравниваются пароли и на email пользователя отправляется сгенерированный на сервере временный код подтверждения. На клиенте после отправки данных с паролем должен произойти редирект на форму для ввода временного кода подтверждения. После отправки кода на сервере сравниваются пришедший код и код из БД. При совпадении кодов аутентификация считается успешной и происходит редирект на страницу-заглушку.

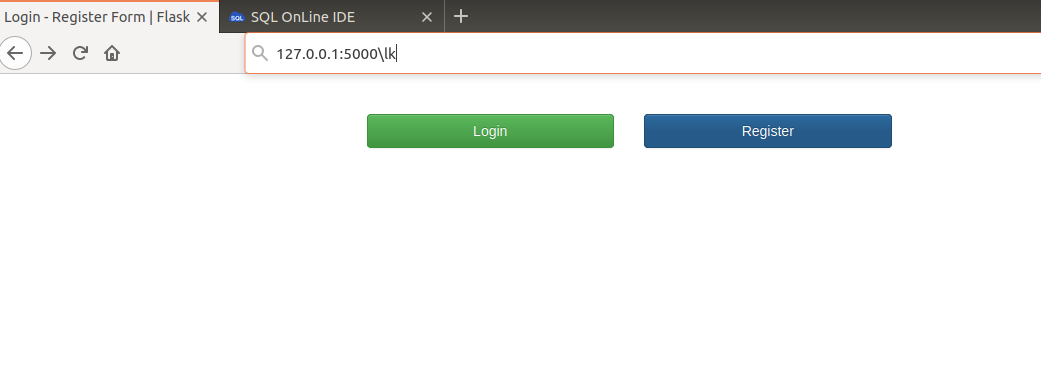
1. **Ссылка на исходный код.**

<https://github.com/elmurza/crypto/blob/main/task2/app.py>

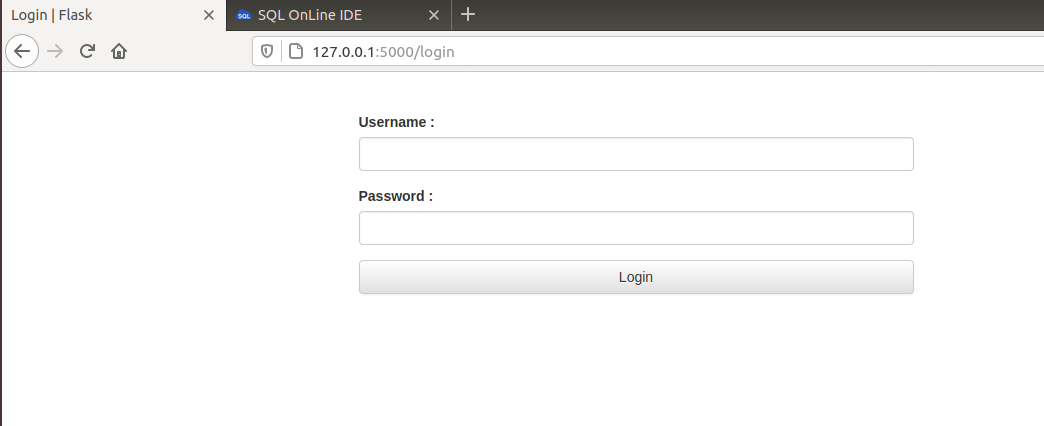
1. **Демонстрация работы**
2. Вход на главную страницу.



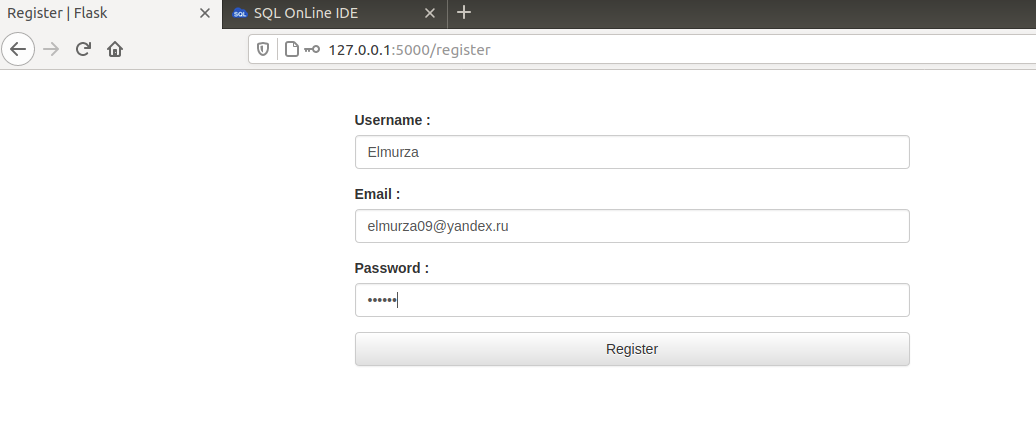
1. Попытка входа в личный кабинет без авторизации.



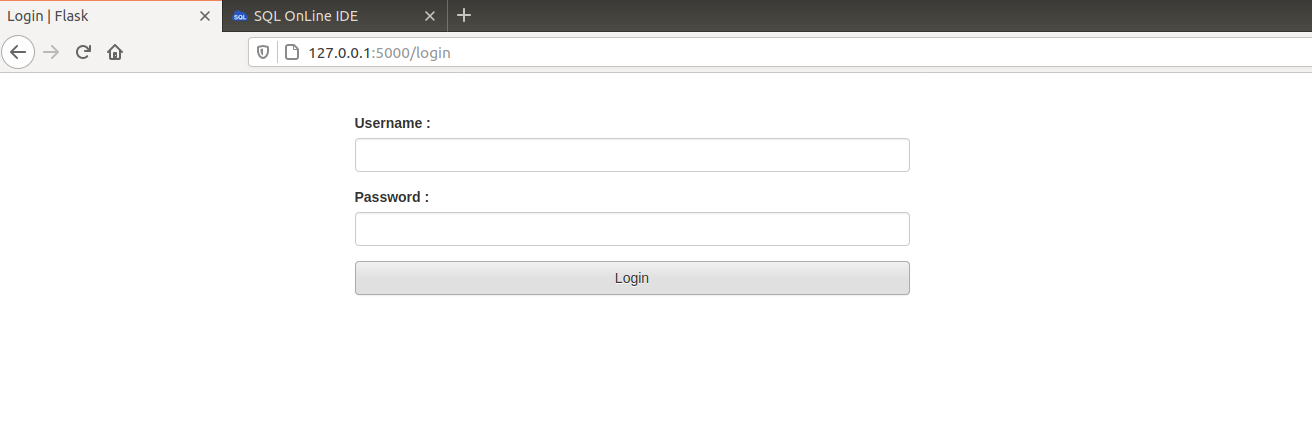
1. При попытке входа перенаправляет на страницу ввода логина и пароля.



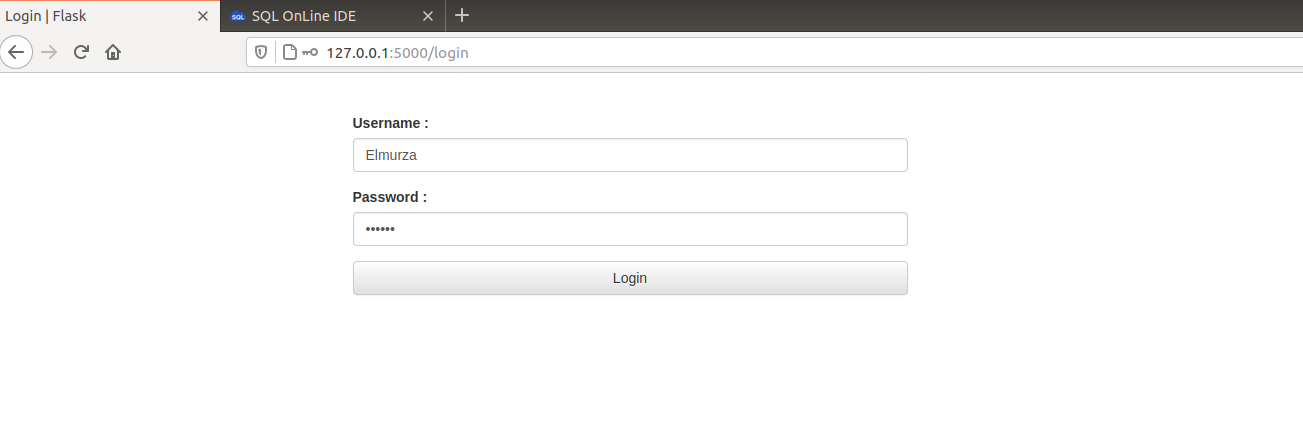
1. Форма регистрации



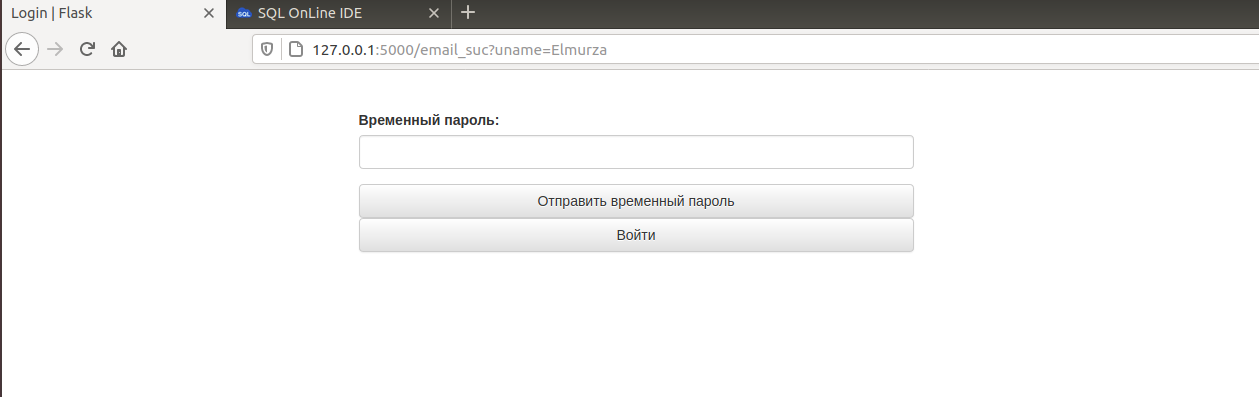
1. Форма ввода логина и пароля



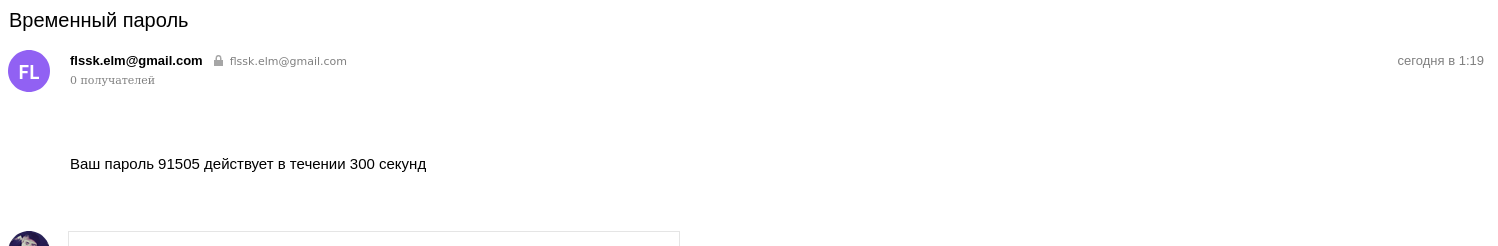
1. Регистрируем аккаунт и заполняем форму



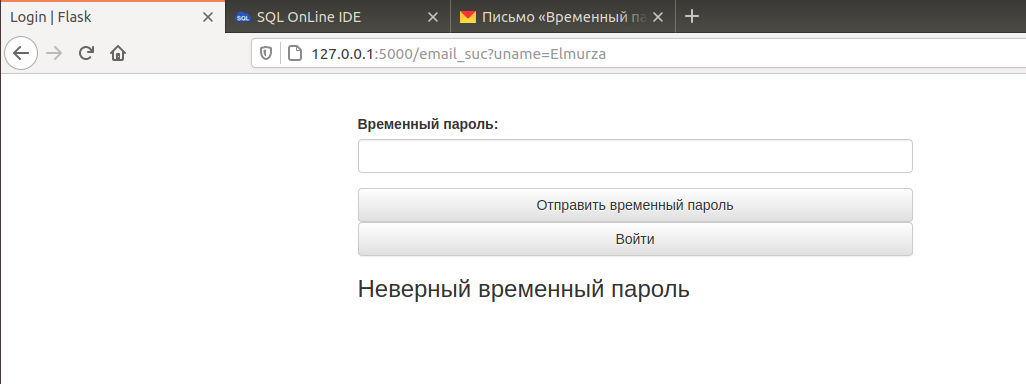
1. Идёт перенаправление на форму ввода временного пароля, нажимаем отправить



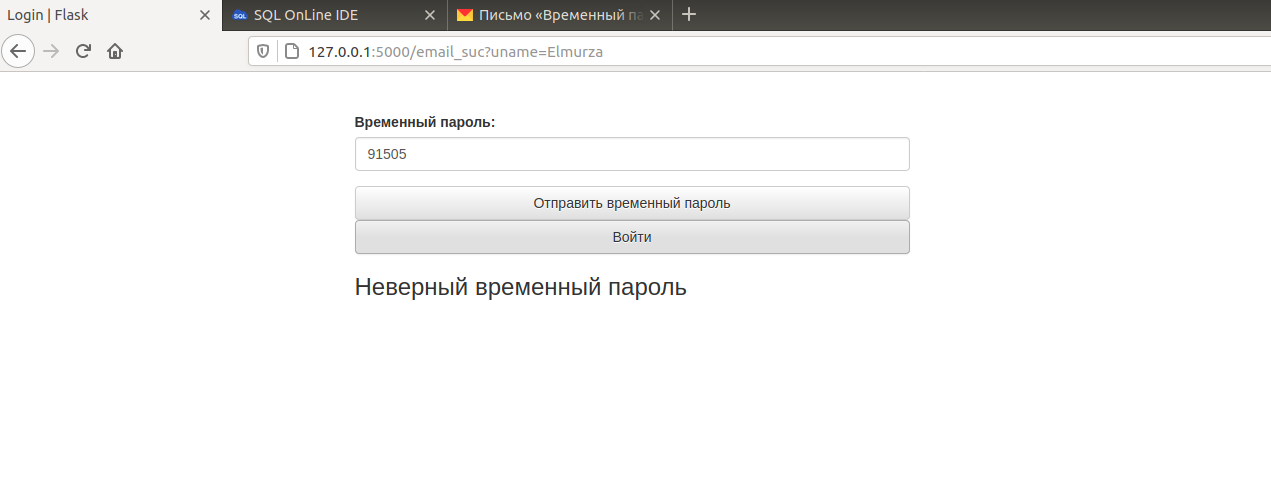
1. На почту приходит временный пароль



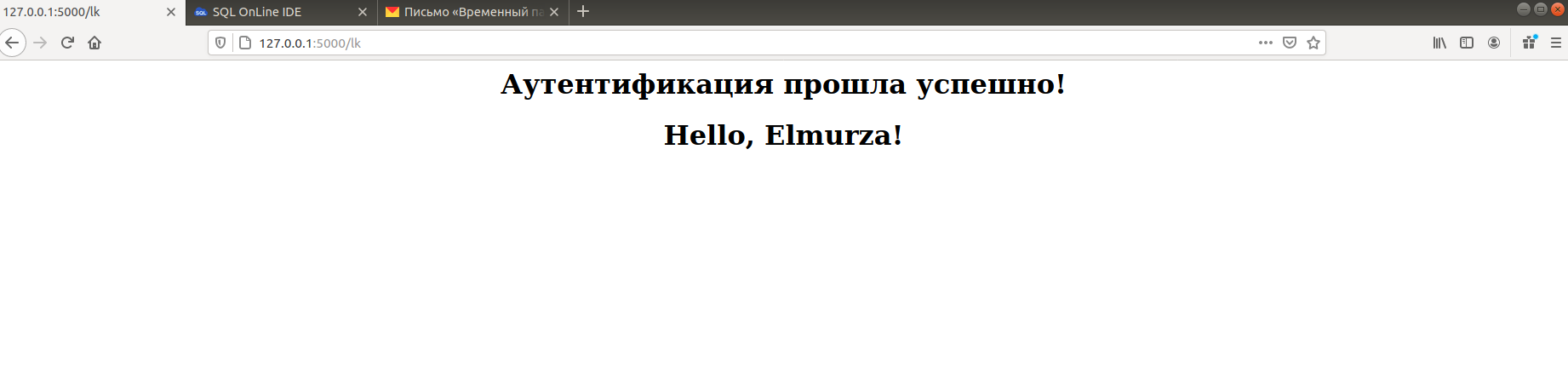
1. При попытке ввода неверного временного пароля отображается Alert



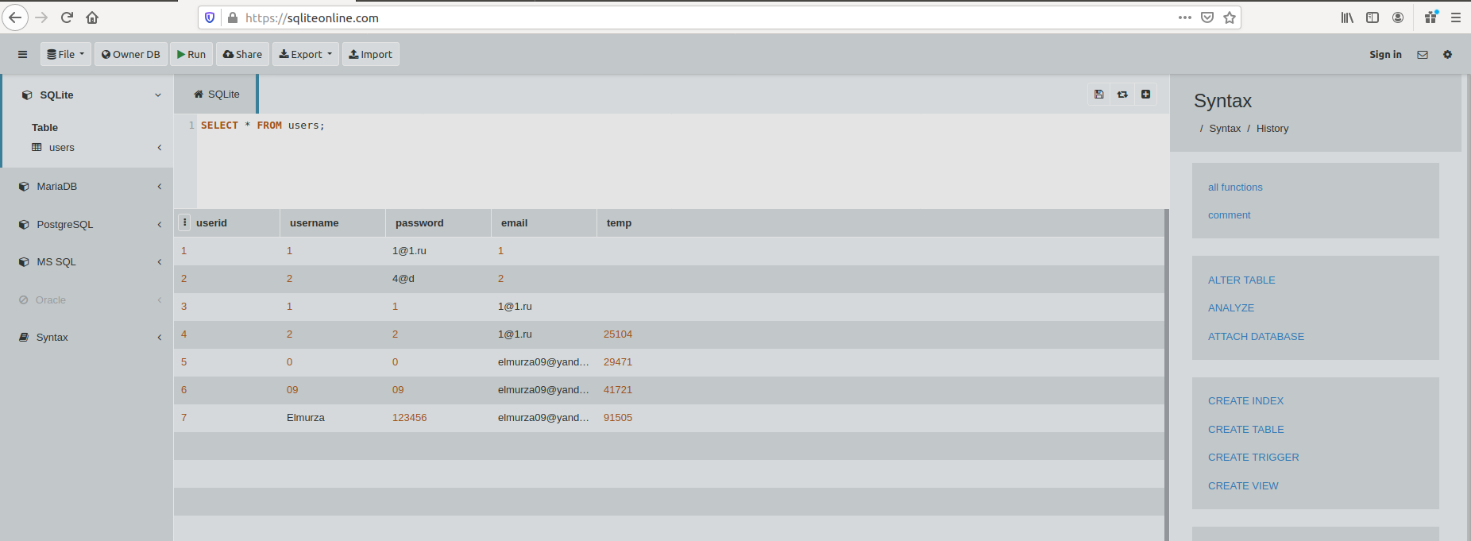
1. После ввода верного пароля нажимаем на кнопку Войти



1. Страница личного кабинета



1. База данных sql



1. **Выводы**

В ходе лабораторной работы разработана система парольной аутентификации с подтверждением по email в клиент-сервером приложении.