Государственное бюджетное общеобразовательное учреждение города Москвы "Марьинская школа № 1566 памяти Героев Сталинградской битвы"

**Проект**

**“Веб-приложение для взаимодействия с Яндекс диском”**

**Выполнил: ученик 10Ф класса Толоконцев Илья**

**Руководитель: Буданцев Артём Владиславович, преподаватель программы “Яндекс.Лицей”**

**Москва, 2021-22**

**Оглавление**

1. Введение 3
   1. Актуальность 3
   2. Цель и задачи работы 3
2. Описание проекта 4
   1. Техническое задание 4
   2. Проектирование архитектуры приложения 5
   3. Настройка Яндекс приложения 5
   4. Главный скрипт 6
   5. База данных 8
   6. Html-страницы 10
   7. API 11
   8. Прочие моменты 12
3. Результат и выводы 13
4. Список использованных источников 15

**Введение**

**Актуальность работы**

В настоящее время всё популярнее становятся облачные хранилища, позволяющие хранить данные не на собственном физическом накопителе, а использовать специально отведённые для этого дата-центры.

Существует множество таких хранилищ, предоставляющих возможность взаимодействия с файлами, не используя при этом память на своих дисках.

Есть смыл применения этой технологии в различных веб-приложениях, что позволяет значительно расширить их функциональность, не прибегая к затратам по памяти со стороны клиента.

**Цель работы**

Создать каркас браузерного приложения, в котором пользователь сможет взаимодействовать с Яндекс диском через интерфейс данного приложения.

**Задачи работы.**

* Создать веб-приложение на Flask (Python)
* Реализовать автоматическое получение токена пользователя для работы с API диска
* Создать HTML-шаблон и унаследованные от него страницы, сверстать их с помощью Bootstrap (CSS, JS)
* Получать доступ к файлам пользователя и отображать их на странице
* Реализовать интерфейс приложения (регистрация, вход, авторизация в Яндекс диске и взаимодействие с файлами на нём)
* Организовать взаимодействие с диском через интерфейс страницы: создание каталогов, удаление файлов, загрузка файлов, открытие файла на диске, получение ссылки на скачивание файла
* Хранение данных о пользователях производить в БД SQLite
* Добавить API**Описание проекта**

**Техническое задание**

На всех страницах приложения будет отображаться навигационная панель, включающая в себя: логотип и название приложения (ссылка на базовую страницу), ссылку на домашнюю страницу (“Главная”), ссылку для авторизации в Яндекс диске, кнопки регистрации и входа или имя авторизованного пользователя (нажав на него, можно будет отозвать токен для API диска или выйти из системы).

На базовой странице отображаются кнопки с выпадающим контентом, отвечающие на FAQ.

На домашней странице расположены функции: создание раздела (ввод названия раздела), добавление файла (ввод названия раздела и выбор файла с компьютера), удаление файла (ввод пути до файла), открытие файла на диске (ввод пути до файла), получение ссылки на скачивание файла (ввод пути до файла). Также отображается содержимое Яндекс диска, а под навигационной панелью отображаются сообщения об ошибках.

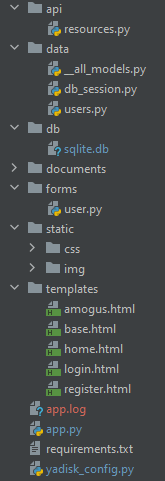
На странице регистрации поля для ввода: почта, пароль, повторение пароля, имя пользователя.

На странице входа поля для ввода: почта, пароль, запомнить меня (флаг)

В БД будут хранится id, почта, имя, хешированный пароль, дата регистрации и токен пользователей.

API приложения позволяет получить информацию об одном или нескольких пользователях, удалить или добавить пользователя.

**Проектирование архитектуры приложения**

Основной скрипт – app.py В нём реализованы обработчики http запросов, обработчики ошибок, вспомогательные функции, логирование в файл app.log, настройка и запуск приложения.

В файле yadisk.config.py находится информация о Яндекс приложении.

В папке templates находятся шаблоны html страниц.

В папке data находятся скрипты: db\_session.py для работы с сессией БД, users.py, в котором определена модель пользователя для БД, \_\_all\_models.py для быстрого импорта всех моделей.

В папке db хранится файл базы данных sqlite.db

В папке forms находится скрипт user.py, описывающий формы для flask-wtf

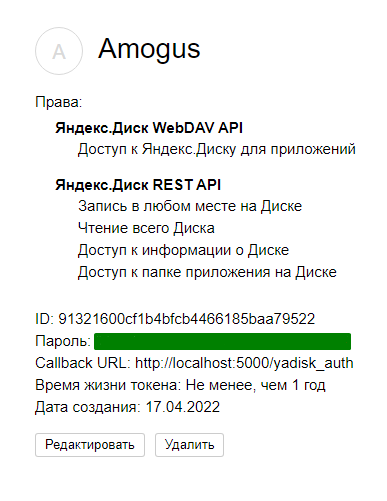
В папке static находятся папка с изображениями и папка с CSS файлом.

В папке api находится скрипт resource.py, в котором описаны ресурсы API

**Настройка Яндекс приложения**

Переходим на страницу регистрации приложения <https://oauth.yandex.ru/client/new> и заполняем нужные данные

По итогу получаем ID и пароль приложения:

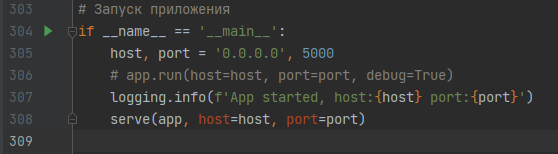


Callback URL – адрес страницы, на которую вернётся пользователь после аутентификации в Яндексе.

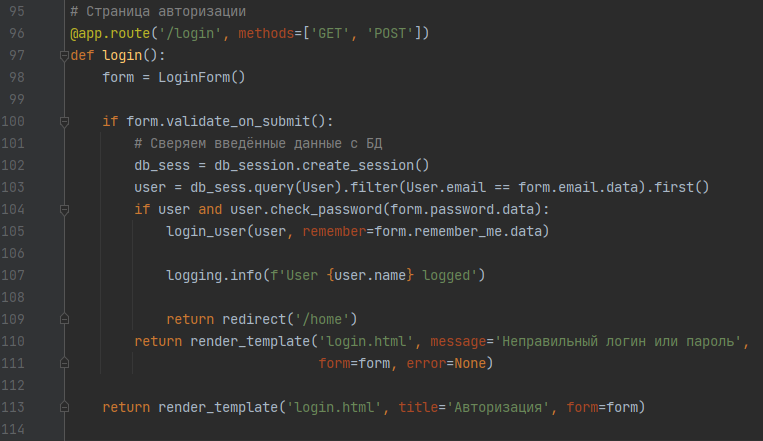
**Главный скрипт**

В файле app.py создаём Flask-приложение, работающее по адресу <http://localhost:5000>. Здесь же прописаны: обработчики страниц и ошибок, алгоритмы регистрации/входа в приложение и авторизации в системе Яндекс, логирование, выгрузка данных о пользователе из БД, подключение ресурсов API, функции взаимодействия с Яндекс диском, вспомогательные функции.

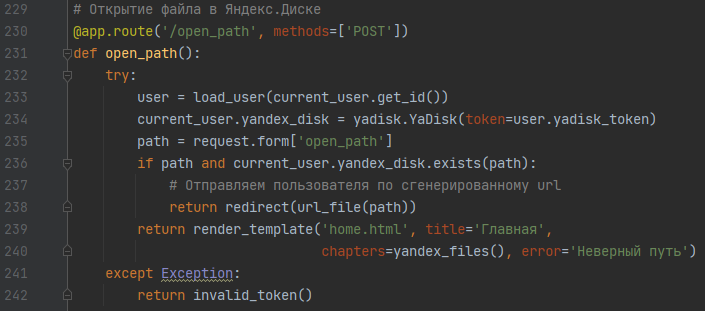
Несколько скриншотов кода:



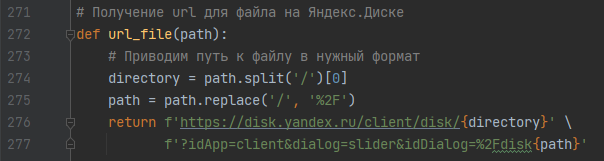
Пример обработчика страницы авторизации:



Пример функции для взаимодействия с файлом на Яндекс диске:



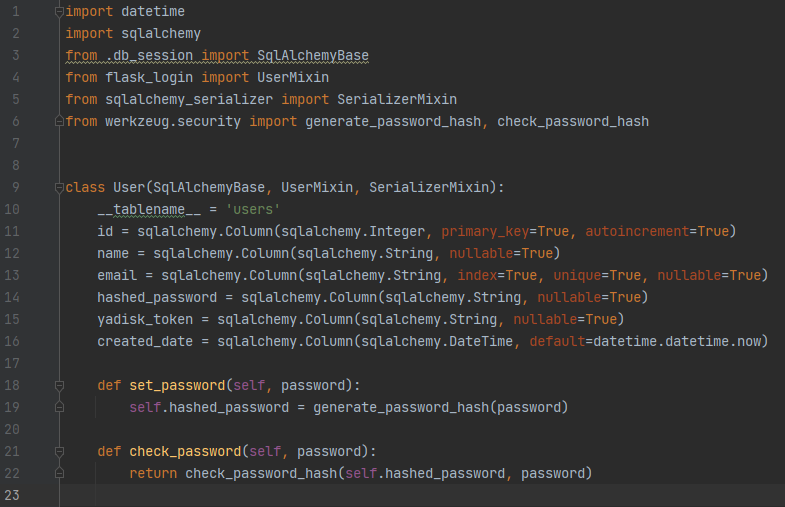
Одна из вспомогательных функций:



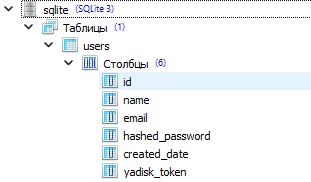
**База данных**

Для взаимодействия с БД используется технология ORM (Object-Relational Mapping). Это позволяет связать БД с концепцией объектно-ориентированного программирования.

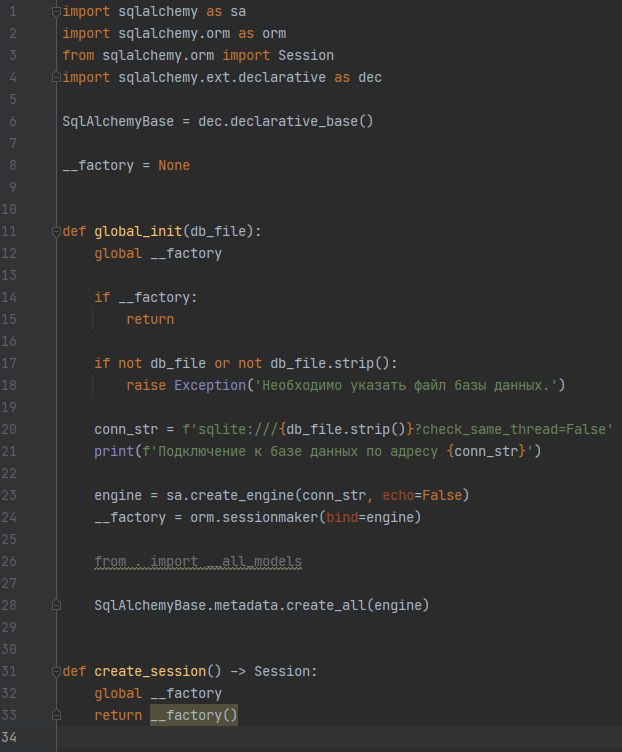
В первом файле пропишем класс пользователя с необходимыми полями:



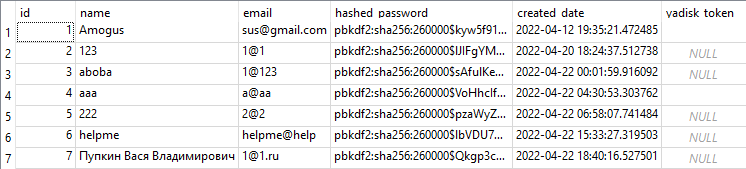
Для шифрования паролей используется модуль werkzeug.security, который позволяет создавать хэши паролей, а также организовать их проверку.

 Структура

Во втором файле – подключение к БД и создание сессий:



Пример данных в таблице:

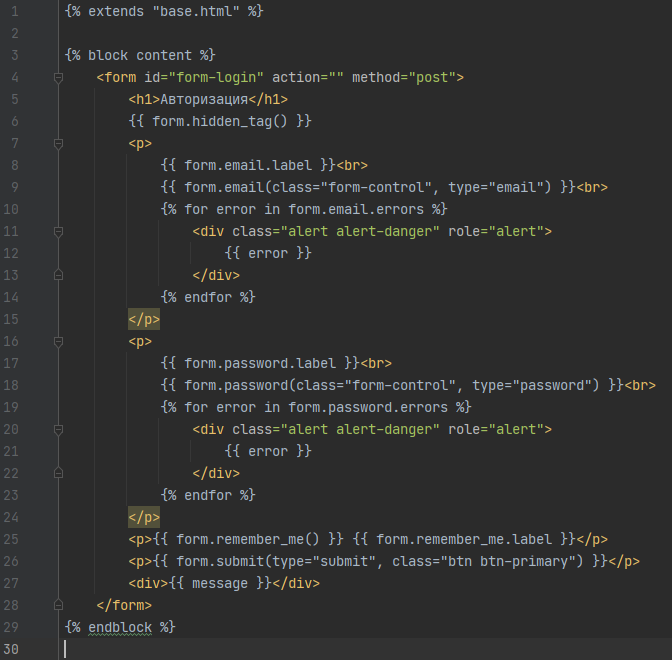


**Html-страницы**

Создадим шаблон, от которого унаследуем все остальные страницы. Также подключаем Bootstrap и собственный файл CSS.

Bootstrap - свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

Пример унаследованной от шаблона base.html страницы авторизации login.html:



Все расширения подставляются в блок в шаблоне:



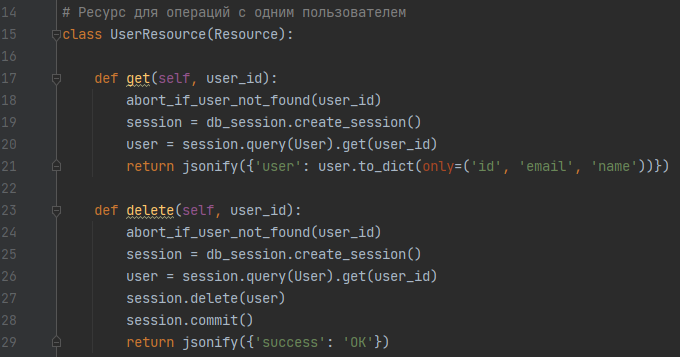
В папке static разместим файл style.css и каталог со статичными изображениями.

**API**

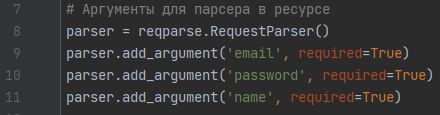
**API** (**A**pplication **P**rogramming **I**nterface - программный интерфейс приложения) – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.

Для реализации API также используется объектно-ориентированный метод. Создадим классы и методы ресурсов, унаследованных от базового из модуля flask-restful.

Пример одного из классов, имеющего методы получения информации о пользователе и удаление пользователя:



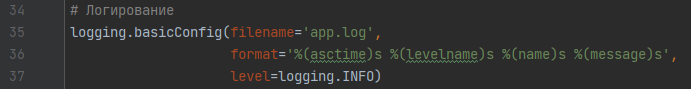
Необходимые аргументы:

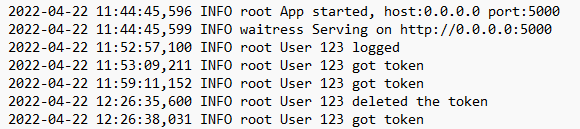


**Прочие моменты**

В ходе работы приложения происходит логирование основных действий в файл app.log, чтобы предоставлять информацию в случае сбоев или для расследования интересных случаев.

Пример логирования:



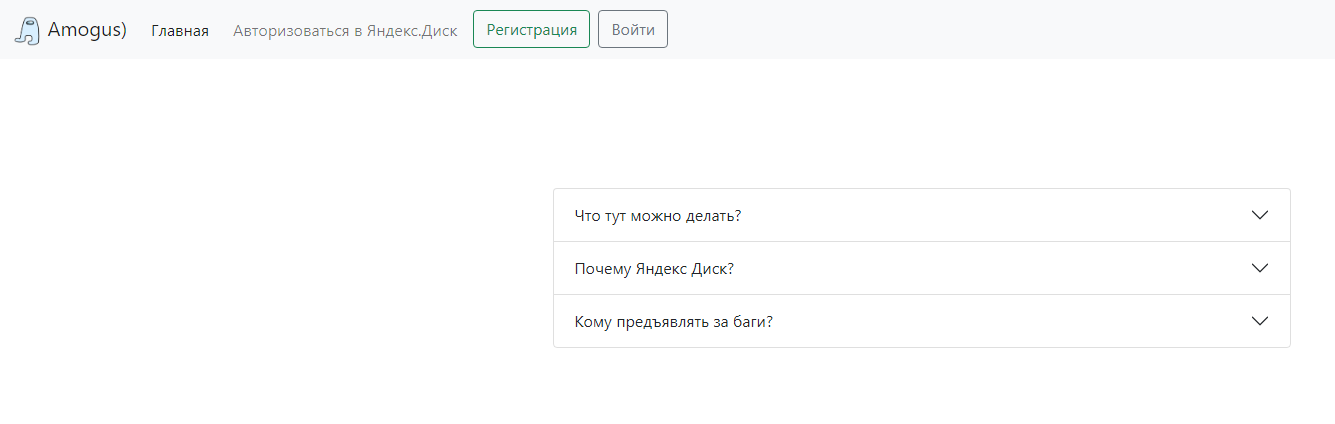


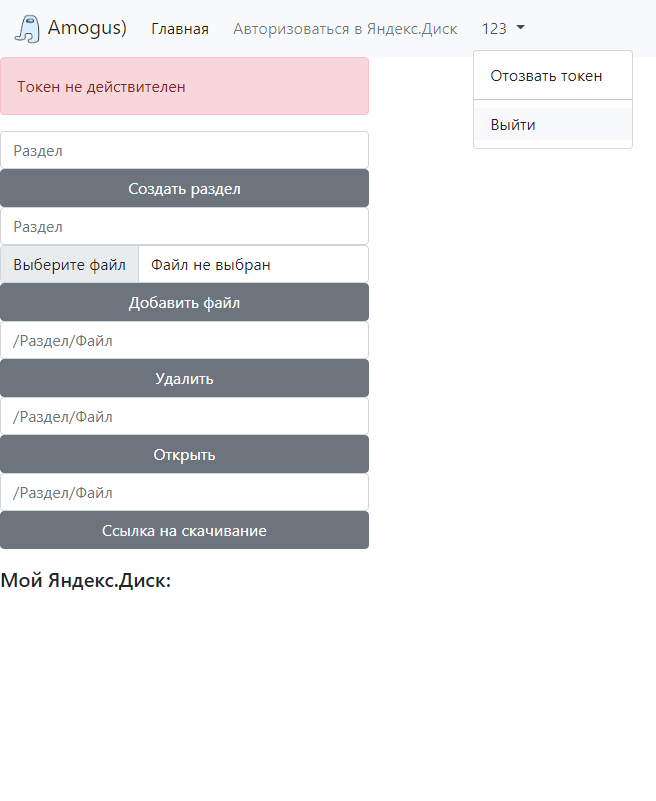
Все необходимые для работы приложения библиотеки и модули указаны в файле requirements.txt

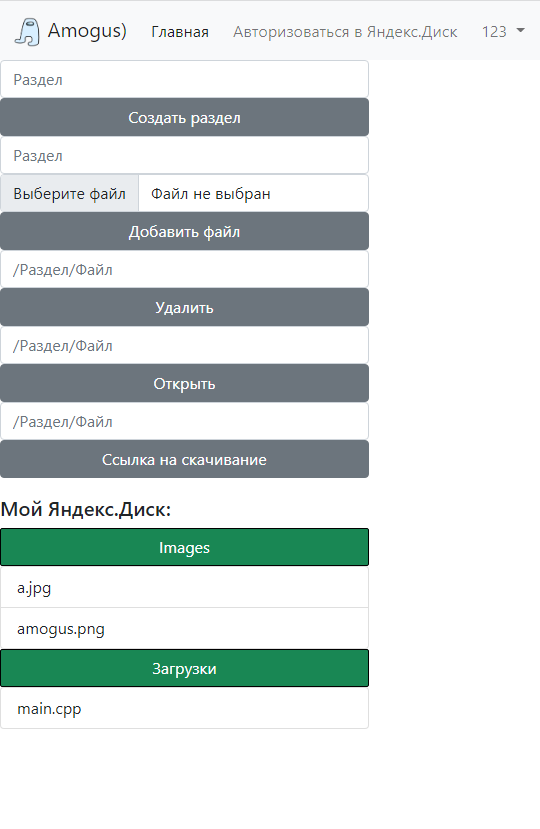
В файле yadisk\_config.py хранятся данные о Яндекс приложении – ID и пароль.

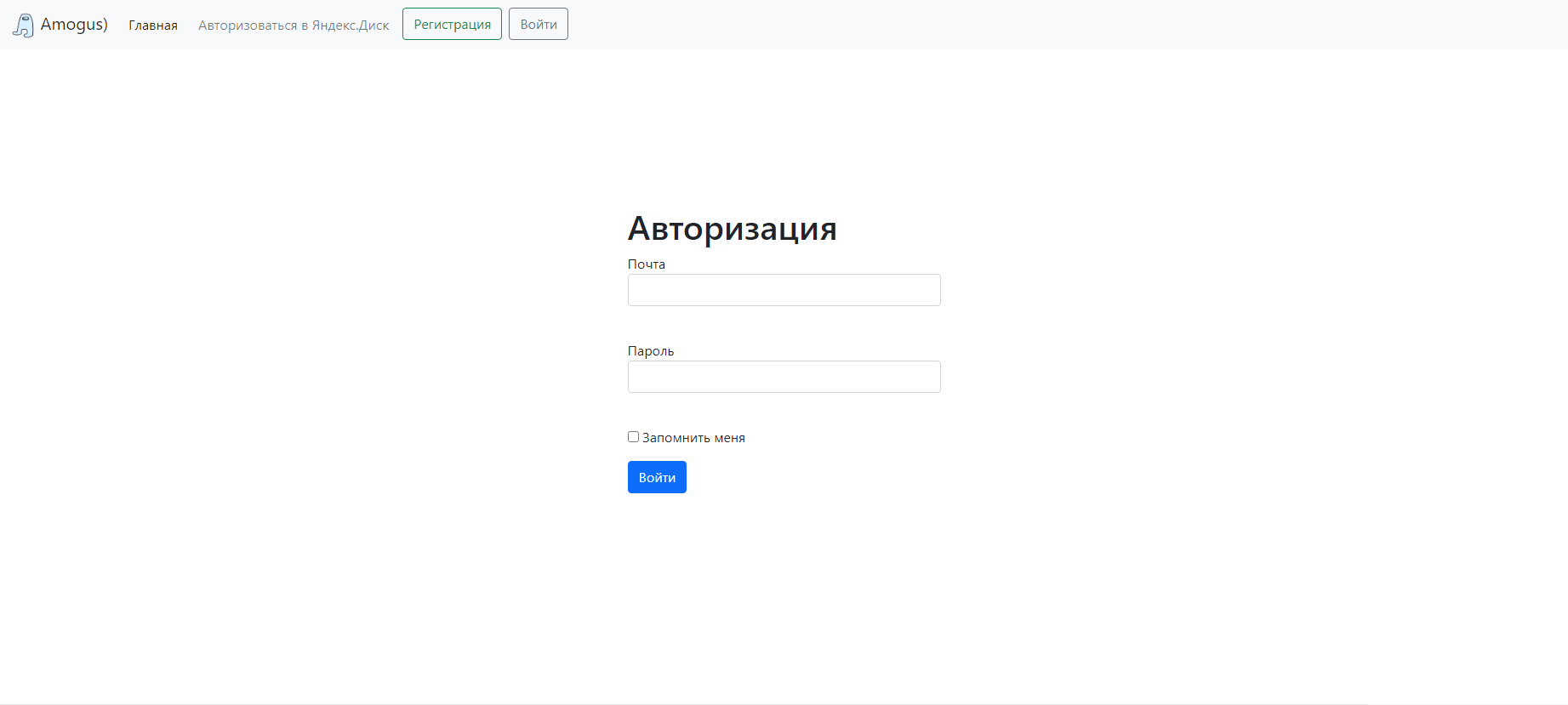
**Результат и выводы**

На выходе получился рабочий каркас веб-приложения, в котором реализованы поставленные задачи проекта. На его основе веб-разработчики могут быстро реализовать взаимодействие пользователя с файлами на Яндекс диске, уже имея при этом фундаментальный функционал любого приложения.









**Список использованных источников**

1. <https://yandex.ru/dev/direct/doc/examples-v5/python3-requests-token.html>
2. <https://bootstrap-4.ru/docs/5.1/getting-started/introduction/>
3. https://yadisk.readthedocs.io/ru/latest/intro.html