Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ПГТУ»)

Кафедра ИиСП

ОТЧЁТ ПО КУРСОВОМУ ПРОЕКТУ

На тему

«Разработка веб приложения для удаленной записи на прием к врачу»

По дисциплине «Базы данных»

Разработали: студенты ПС-32

Стариков Сергей Сергеевич

Проверил:

Лучинин З.С.-к.т.н. доцент кафедры ИиСП

Йошкар-Ола

2021 год

Оглавление

[Введение 3](#_Toc70291117)

[1. Техническое задание 4](#_Toc70291118)

[2. Разработка базы данных 5](#_Toc70291119)

[3. Разработка веб-приложения 6](#_Toc70291120)

[3.1 Структура backend проекта 7](#_Toc70291121)

[3.2 Структура frontend проекта 9](#_Toc70291122)

[3.3 Пользовательский интерфейс 10](#_Toc70291123)

[4. Пример SQL запроса 13](#_Toc70291124)

[Заключение 14](#_Toc70291125)

# **Введение**

С ростом потребителей интернета появляется требование пользователей к возможности удаленной на прием в различные структуры. Популярность сервисов электронной регистратуры объясняется их очевидными достоинствами, которые особенно заметны при рассмотрении продуктов, занимающих на рынке лидирующие позиции.

Ключевые преимущества онлайн-записи к медицинским специалистам являются: минимум затрачиваемого времени, круглосуточный режим работы, получение дополнительной информации о медицинских специалистах, отсутствие дополнительной платы или комиссии за услуги электронной регистратуры.

Основным минусом подобных интернет ресурсов выступают технические сбои, способные доставить пользователям немало проблем.

# **1. Техническое задание**

Цель: Разработка веб приложения для удаленной записи на прием к врачу.

Функционал:

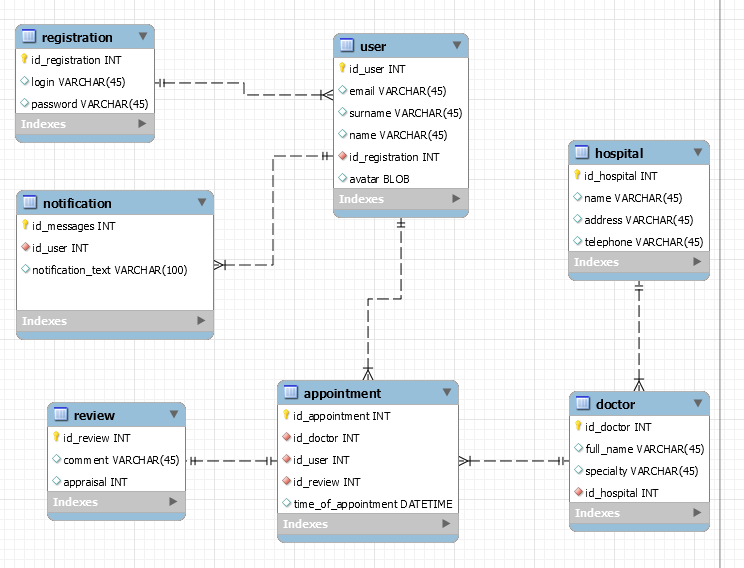
* Пользователю доступна регистрация/авторизация с сохранением сессии
* Все пользователи имеют возможность
  + Просматривать список врачей, их специальность и узнавать в каком учреждении они работают
* Авторизованные пользователи имеют возможность
  + Записываться на прием
  + Редактирование в профиле личных данных
  + Добавление аватара в профиле
  + Просматривать текущие записи на прием
  + Просматривать свои уведомления
  + Удалять свои уведомления
  + Удалять свои записи на прием
  + Оставлять отзыв и ставить оценку приему
  + Поиск записей по дате и имени доктора
  + Сортировка записей
    - По дате прохождения (убывание, возрастание)
    - По имени доктора (убывание, возрастание)

Средства разработки:

1. Клиентская часть: JavaScript, HTML, CSS
2. Серверная часть: Django (Python), SQLite

# **2. Разработка базы данных**

Для любого современного веб-приложения нужно хранилище данных. Место где будет осуществляться хранение, изменение и удаление данных. Для большинства приложений подходит реляционные базы данных. Ниже представлена структура базы данных приложения для удаленной записи на прием к врачу.



В базе данных присутствуют таблицы:

1. User – Пользователь
2. Appointment – Запить на прием
3. Doctor – Доктор
4. Hospital – Больница
5. Notification – Уведомление
6. Review – Отзыв о приеме у врача
7. Registration – Данные для авторизации

# **3. Разработка веб-приложения**

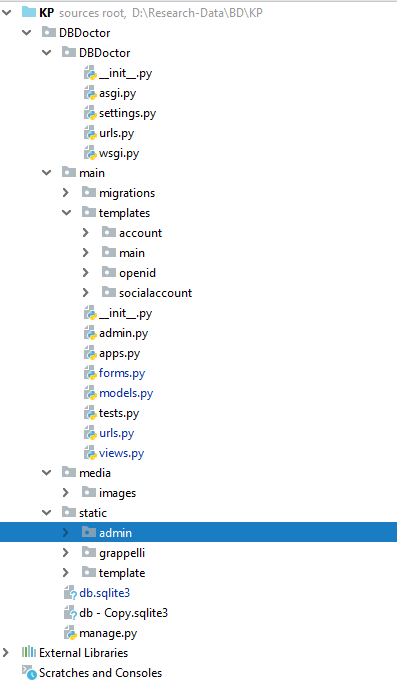
Веб-приложение - клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера. В основе клиентского приложения лежат следующие технологии:

* JavaScript – язык логики веб-страниц;
* CSS - каскадные таблицы стилей;
* HTML - гипертекстовый язык разметки.

В качестве серверной среды разработки выбраны:

* Python - серверный язык программирования;
* SQLite – компактная встраиваемая система управления базами данных;
* Django — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC.

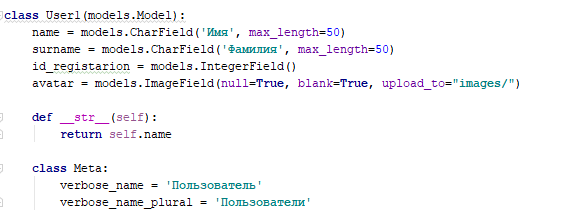
## **3.1 Структура backend проекта**



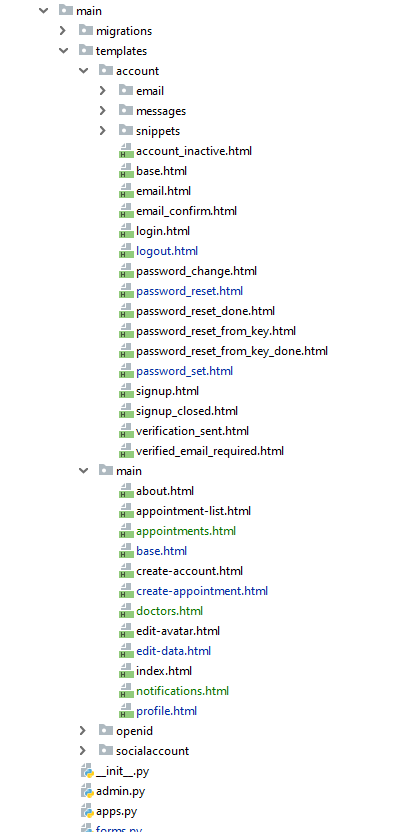
Django — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation.

Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка от некоторых других (например, Ruby on Rails). Один из основных принципов фреймворка — DRY (англ. Don't repeat yourself)

Класс User - таблица в базе данных, которая хранит всех сотрудников и их информацию. В таблицу за `писываются поля - имя(name), фамилия(surname), изображение(avatar) и id\_registration(id в таблице с паролями и логинами).



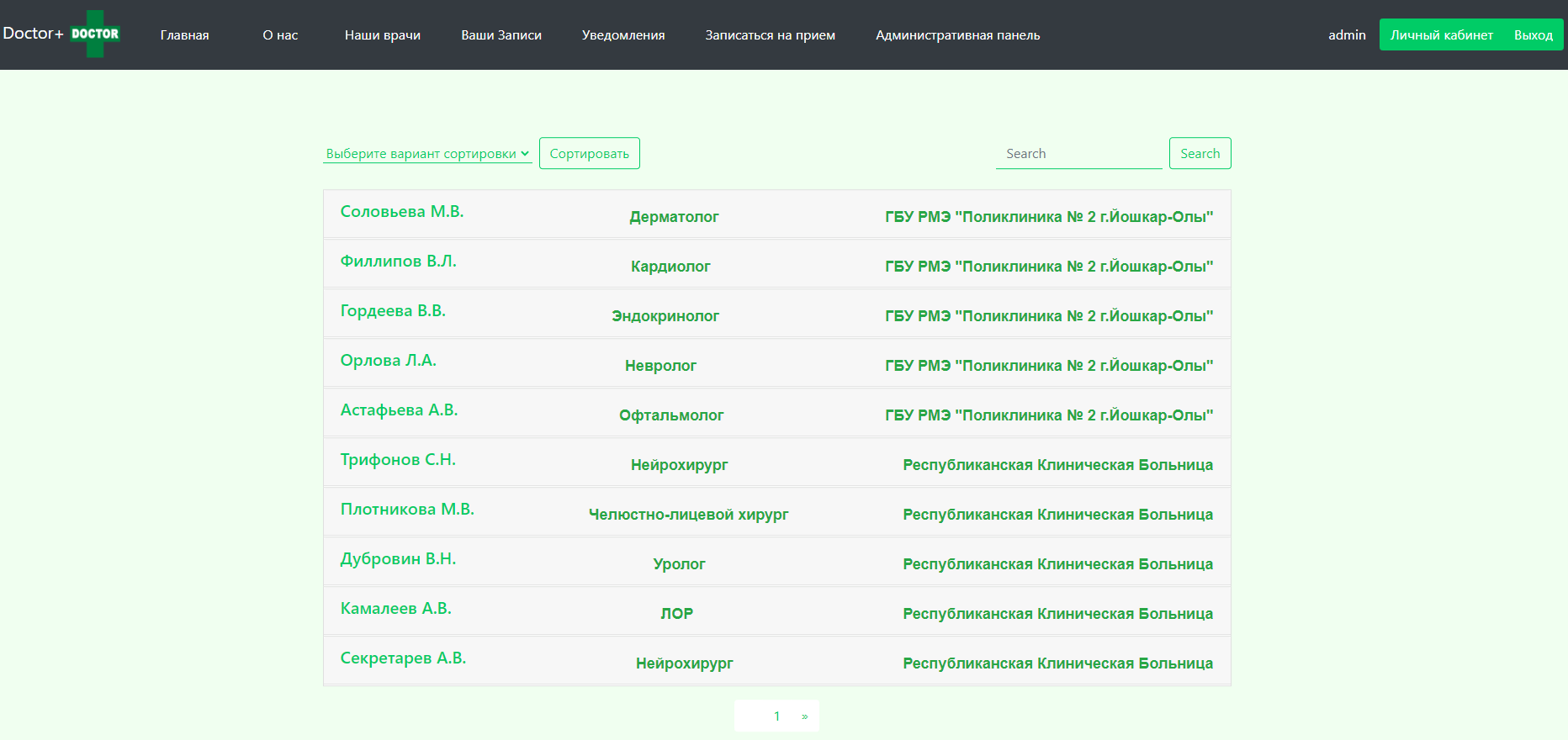
## **3.2 Структура frontend проекта**



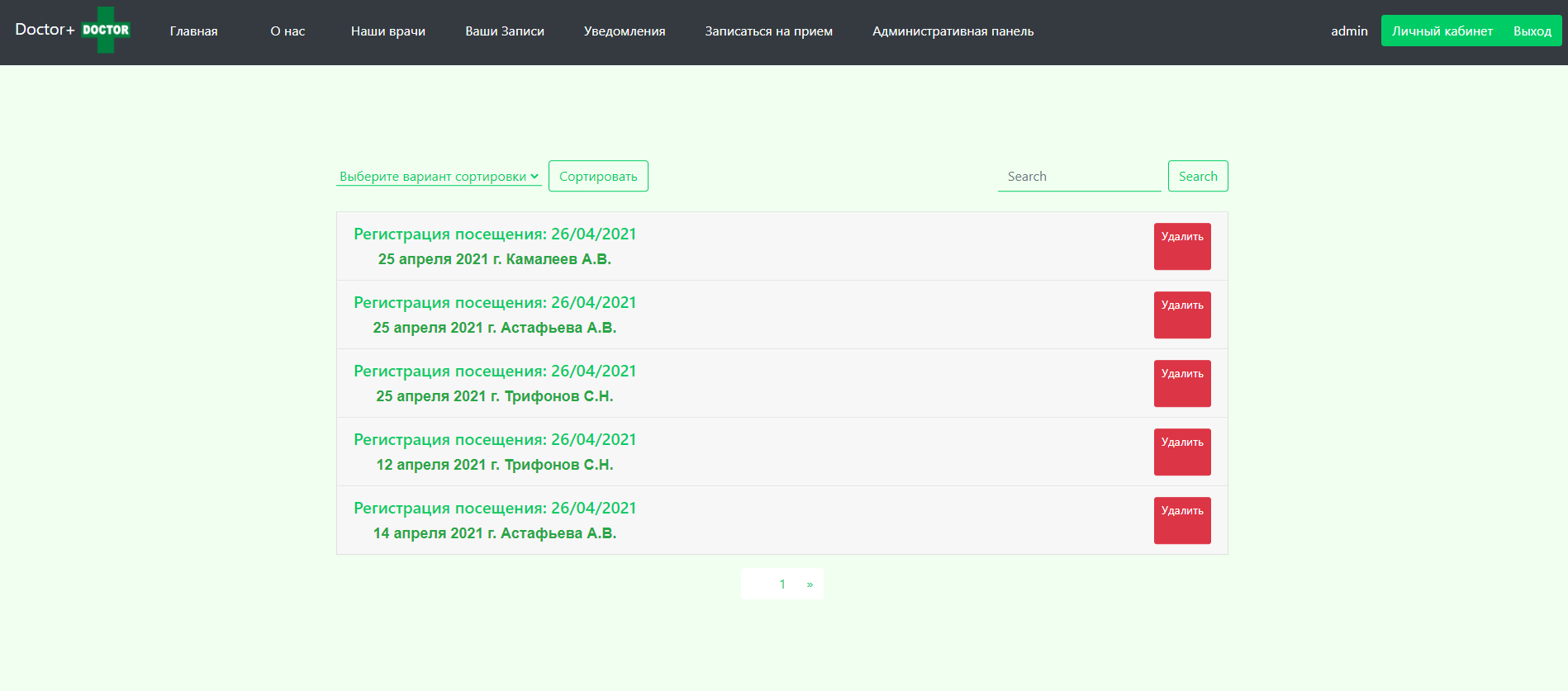
Клиентский код располагается в папке templates. В папке main хранится точка входа в приложение. В папке main представлены элементы, макеты и страницы приложения.

## **3.3 Пользовательский интерфейс**

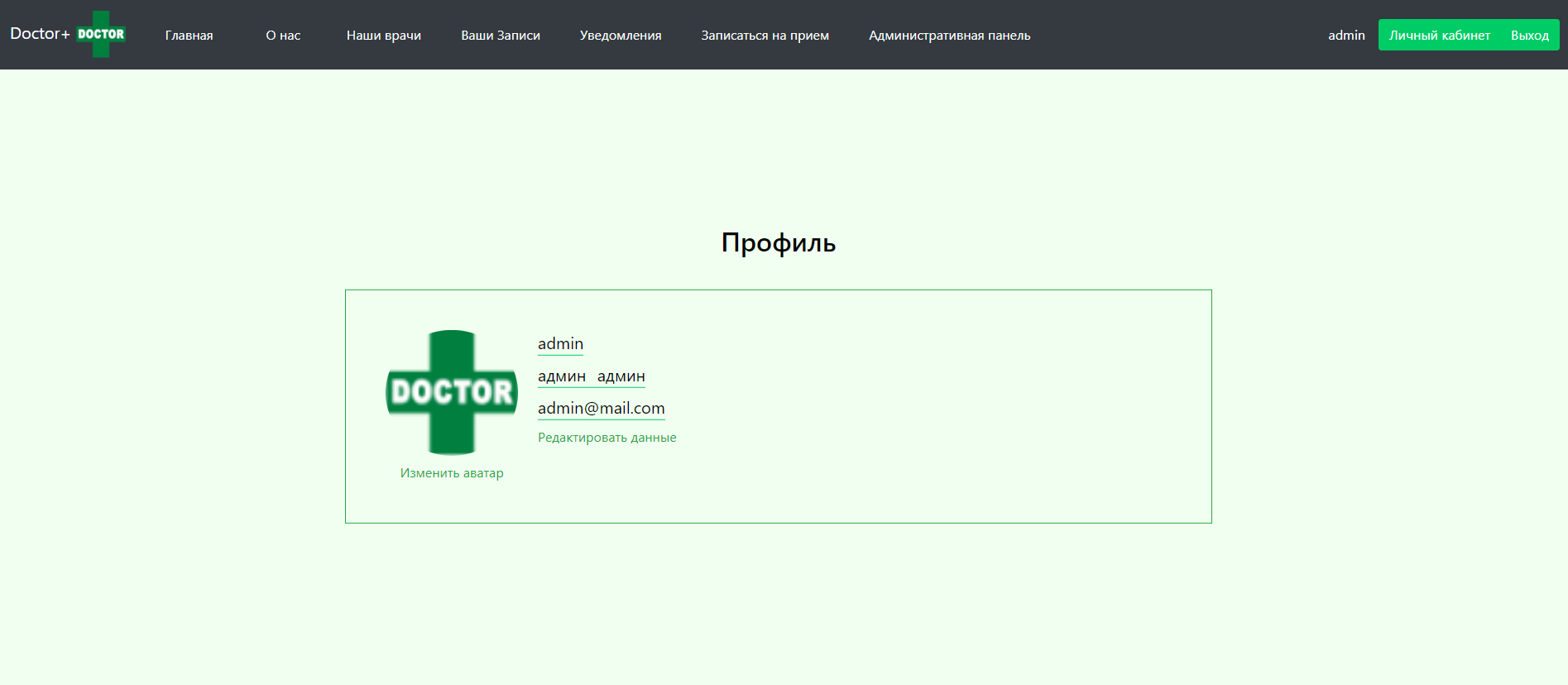
Список врачей



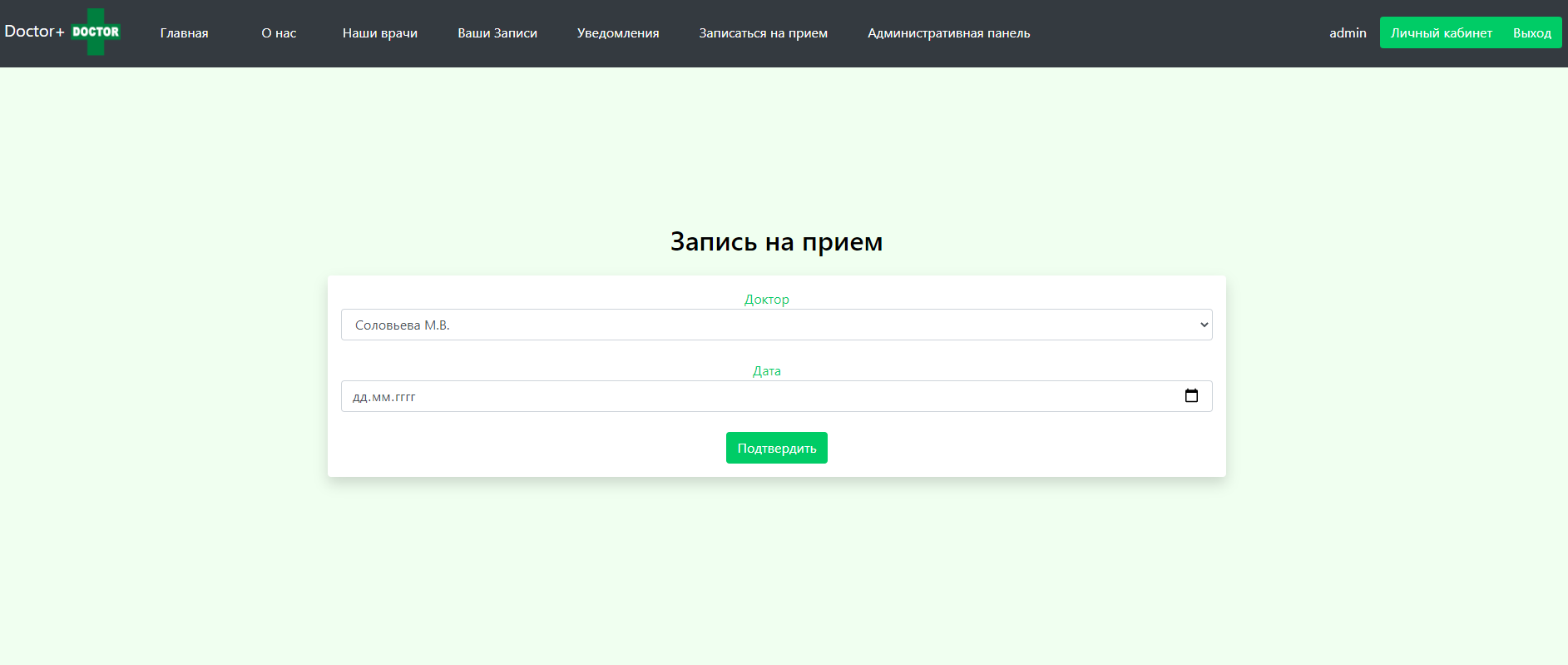
Список записей на прием



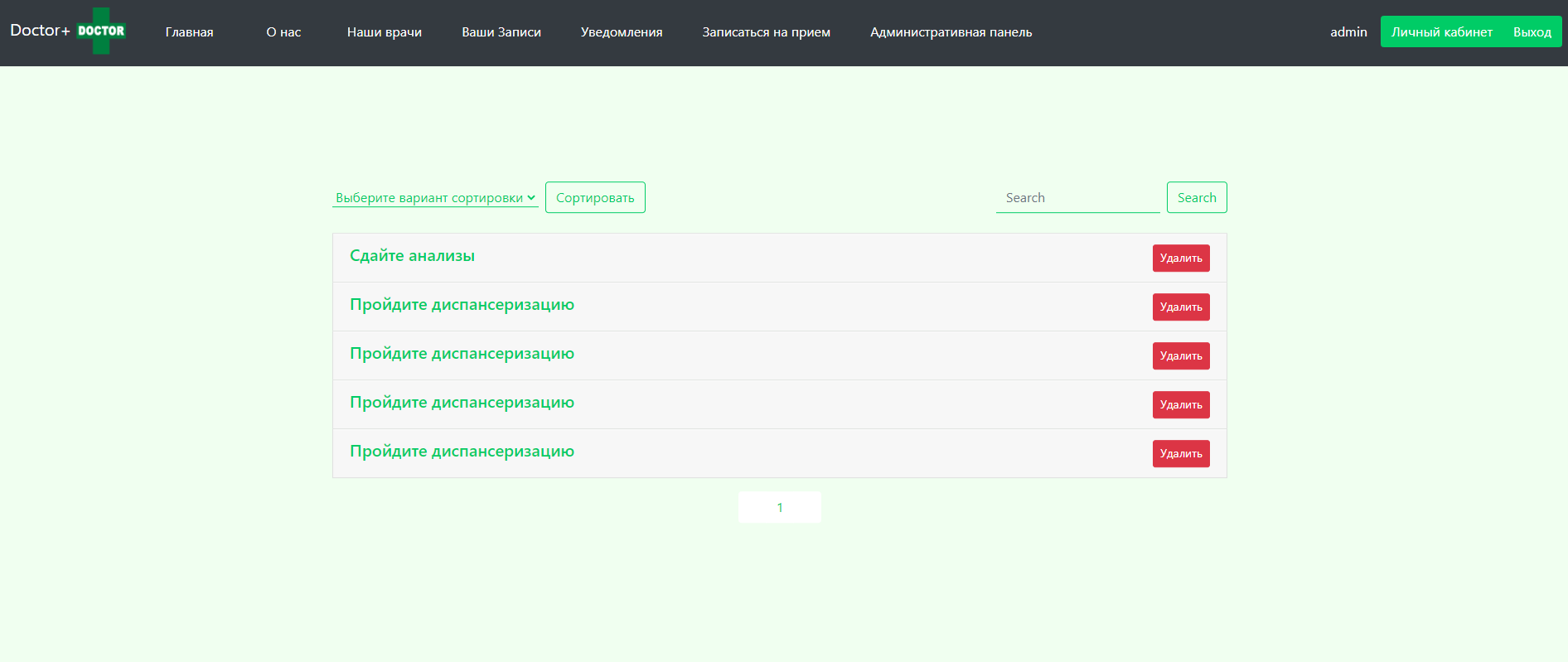
Профиль пользователя



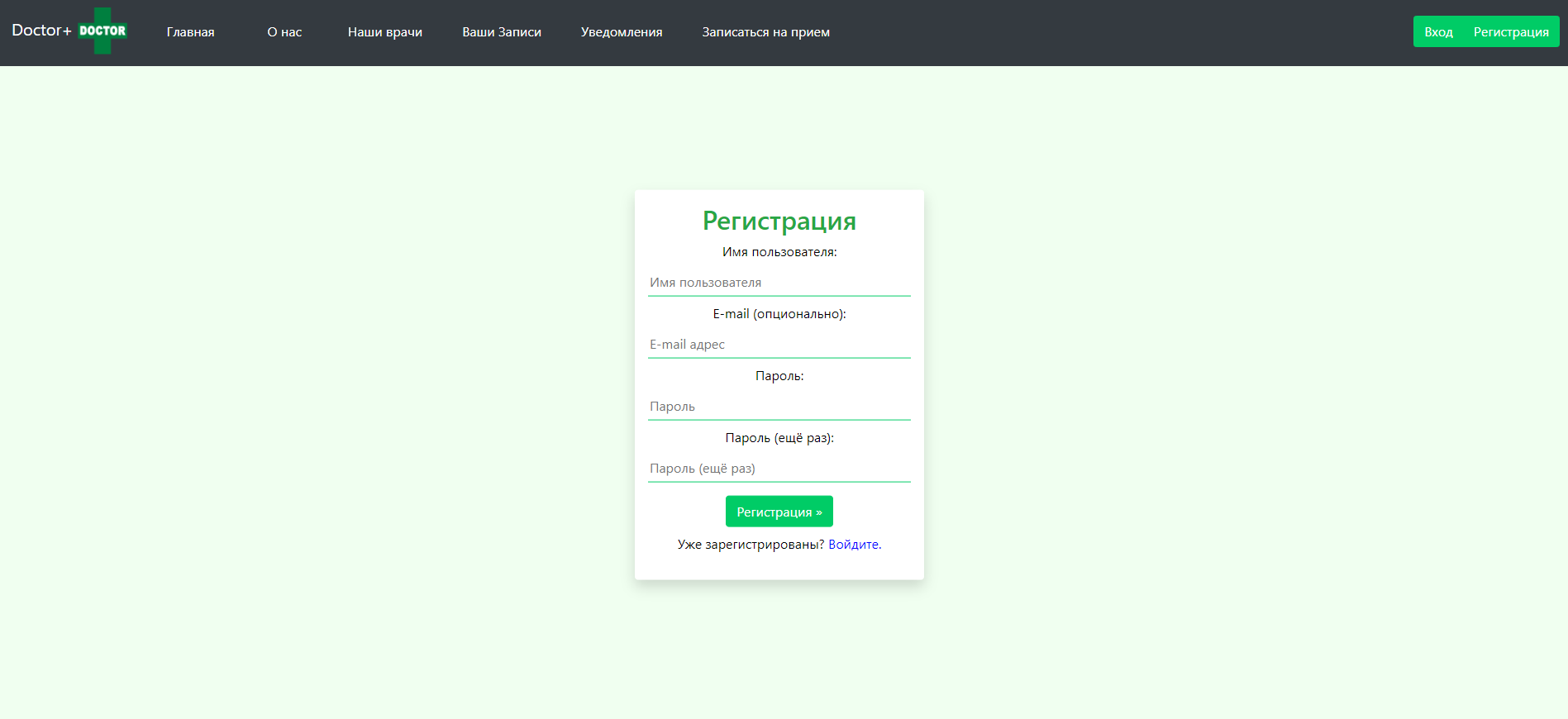
Запись на прием



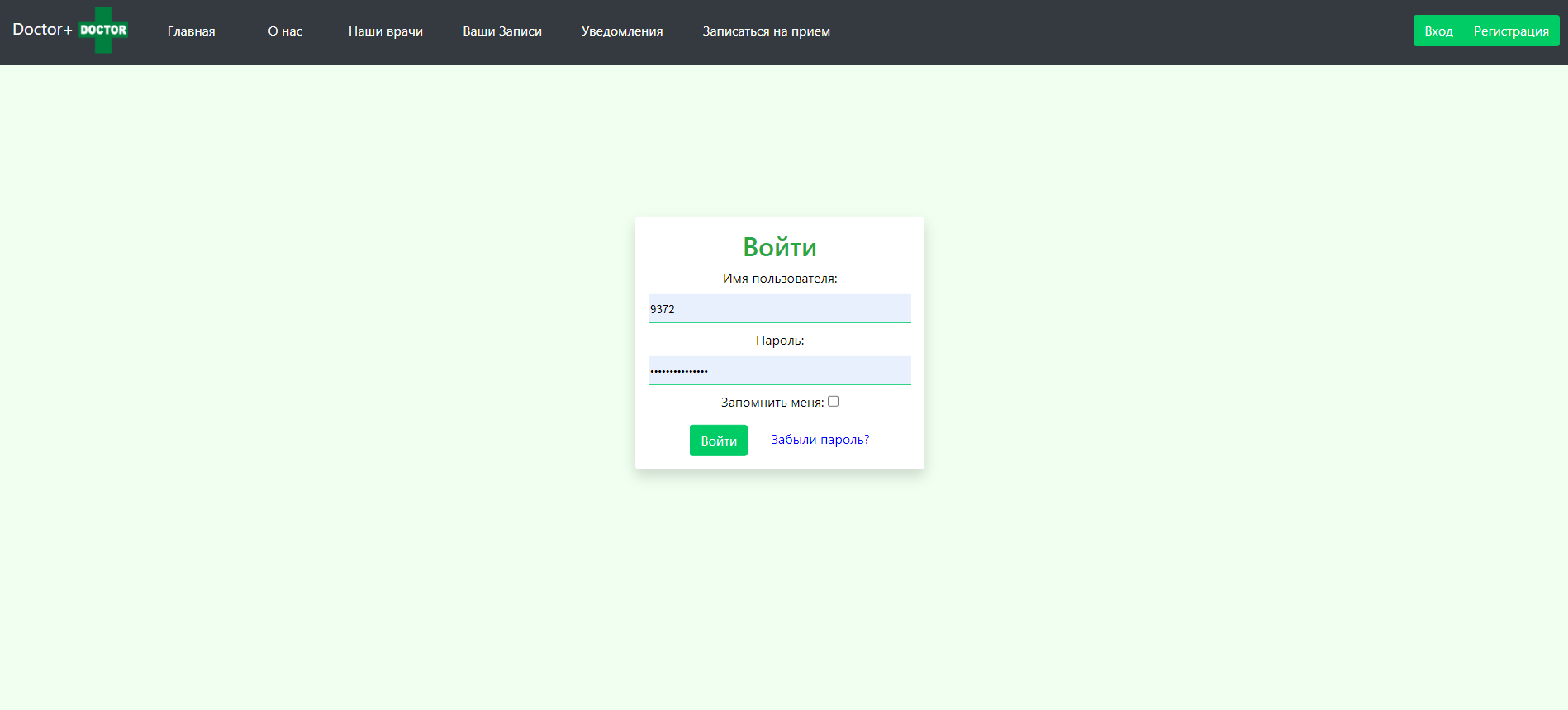
Список уведомлений



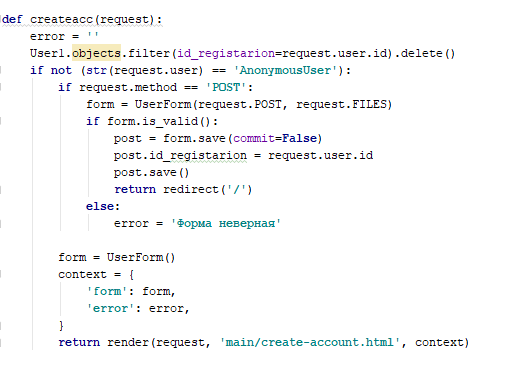
Регистрация



Вход



# **4. Пример SQL запроса**



Пример скрипта для заполнения аккаунта. Принимает аргумент - запрос. Команда objects.get вытаскивает из базы данных необходимую запись и на странице /create-account создает формы ввода данных.

# **Заключение**

При разработке приложения была спроектирована и использована база данных SQLite для обеспечения хранения информации приложения. Так же в процессе разработки приложения были изучены следующие понятия и технологии разработки: MVC, Django, Python, JavaScript, HTML, CSS. Были получены теоретические и практические навыки разработки современных клиент-серверных приложений.