МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет прикладной математики, информатики и механики

Кафедра вычислительной математики и информационных прикладных технологий

Направление 01.03.02 Прикладная математика и информатика

Отчет по курсу

«Разработка приложений баз данных»

Тема: «Веб-сервис для создания и поиска мероприятий»

Обучающийся _____ Сафонов Н. С., 4 курс, 3 группа

Преподаватель _____ Крыжановская Ю. А.

Воронеж 2022

Содержание

Введение 3
Схема БД 4
Реализация приложения 5
Структура backend части приложения 5
Реализация frontend части приложения 6
Реализация взаимодействия backend и frontend частей 10
Заключение 11

Введение

В качестве темы для разрабатываемого приложения был выбран веб-сервис для создания и поиска мероприятий.

Основной задачей данного проекта является реализация сервиса, в котором пользователь будет иметь возможность создавать мероприятия, а также искать мероприятие, организованные другими пользователями и регистрироваться на их посещение.

Для реализации приложения был выбран следующий стек технологий:

- Java + Thymeleaf
- Spring Boot
- PostgreSQL

Схема БД

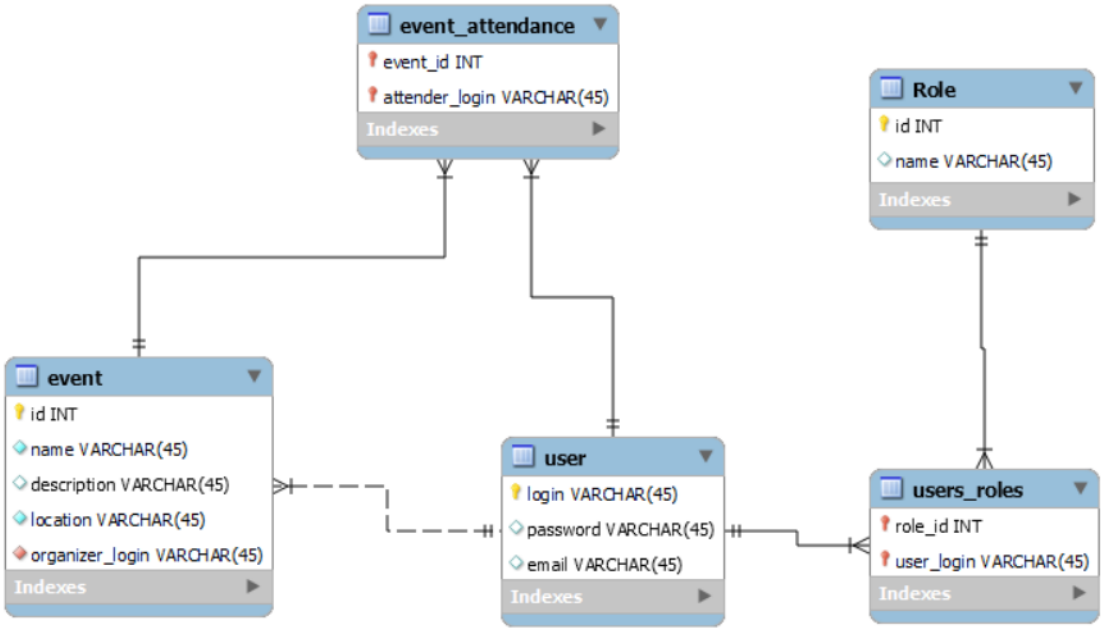

Рис. 1. Схема базы данных приложения

Основные таблицы базы данных:

1. «user» – таблица, хранящая данные о пользователе (логин, пароль, email);
2. «event» – таблица, которая содержит основную информацию о мероприятии (идентификатор, название, описание, место проведения, логин организатора);
3. «event_attendance» – таблица для хранения пары «мероприятие-посетитель»;
4. «role» – таблица, которая хранит роли пользователей (ADMIN/USER);
5. «users_roles» – таблица, содержащая информацию о ролях конкретного пользователя, хранит пары «роль-пользователь».

Реализация приложения

Структура backend части приложения

Рассмотрим более детально внутреннее устройство backend части приложения. Для этого необходимо построить и изучить диаграмму вложенности пакетов приложения (рис. 2).

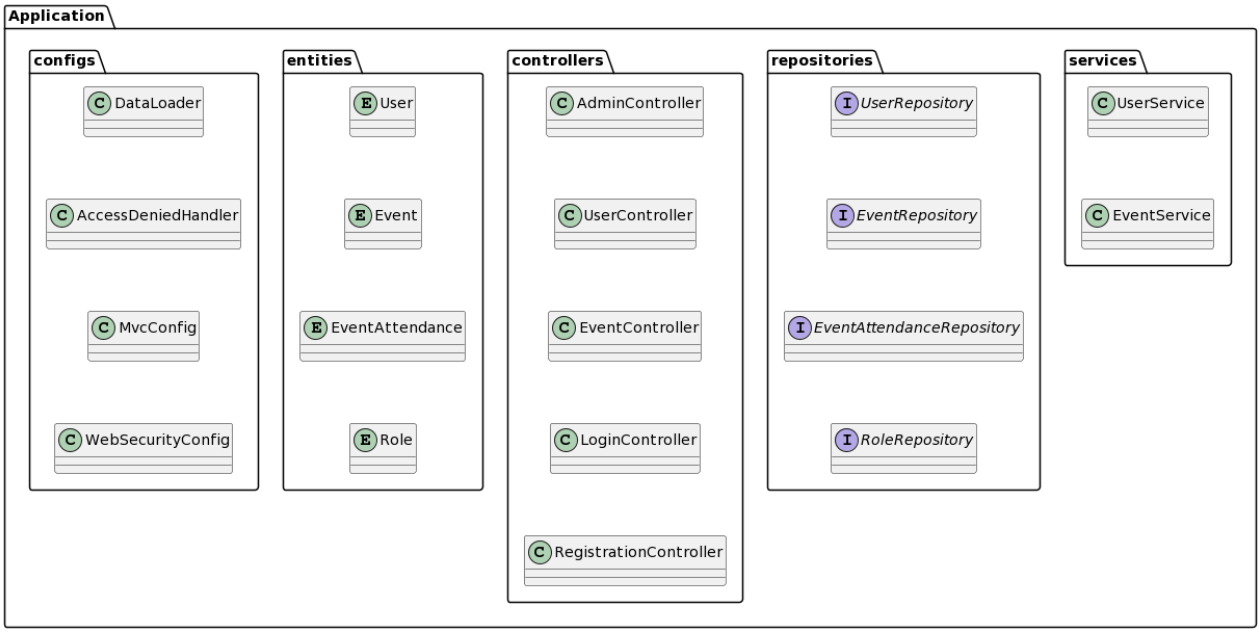

Рис. 2. Диаграмма вложенности пакетов приложения

- В пакете «configs» содержатся классы-конфигураторы, отвечающие за настройки работы веб-приложения;
- Пакет «entities» содержит классы-сущности, которые отображают структуру базы данных;
- Пакет «controllers» содержит классы-контроллеры, которые принимают запросы со стороны frontend;
- Пакет «repositories» содержит интерфейсы, которые унаследованы от JpaRepository и отвечают за взаимодействие между базой данных и сущностями в программе;

- В пакете «services» хранятся сервисы, в которых прописана бизнес логика.

Реализация frontend части приложения

Для реализации пользовательского интерфейса была применена такая технология, как Thymeleaf – механизм HTML-шаблонов, встроенный в Java. Данный механизм имеет обширную интеграцию с backend фреймворком Spring, поэтому был выбран именно он.

Рассмотрим элементы пользовательского интерфейса приложения (рис. 3-9):

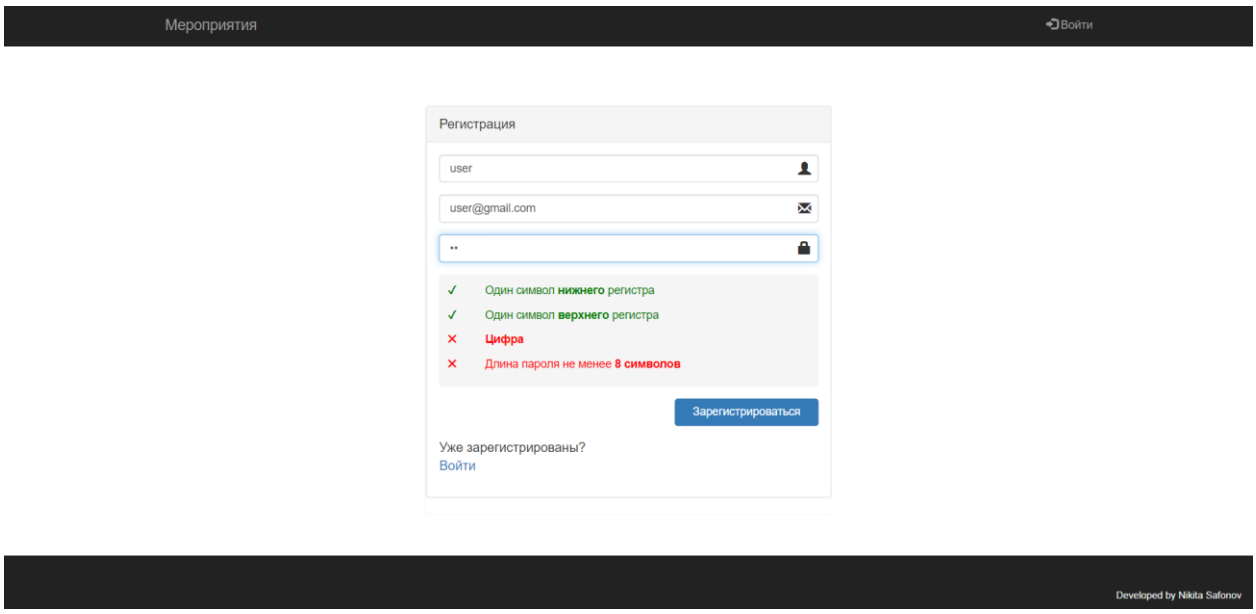

Рис. 3. Страница регистрации пользователя

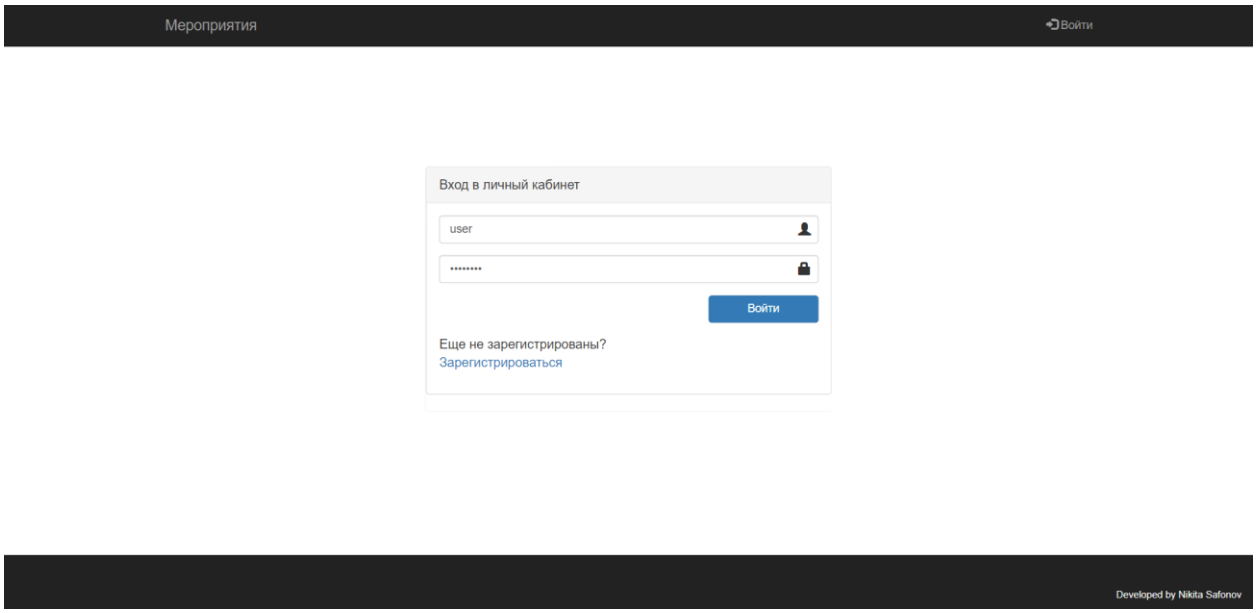

Рис. 4. Страница авторизации пользователя

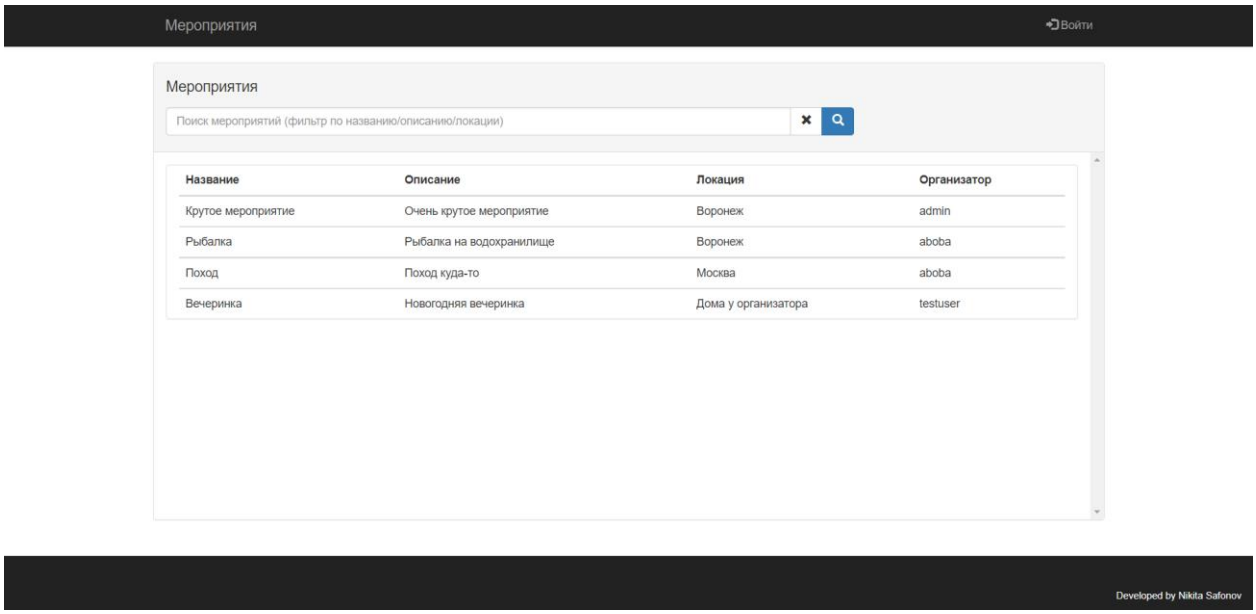

Рис. 5. Страница с мероприятиями (гость)

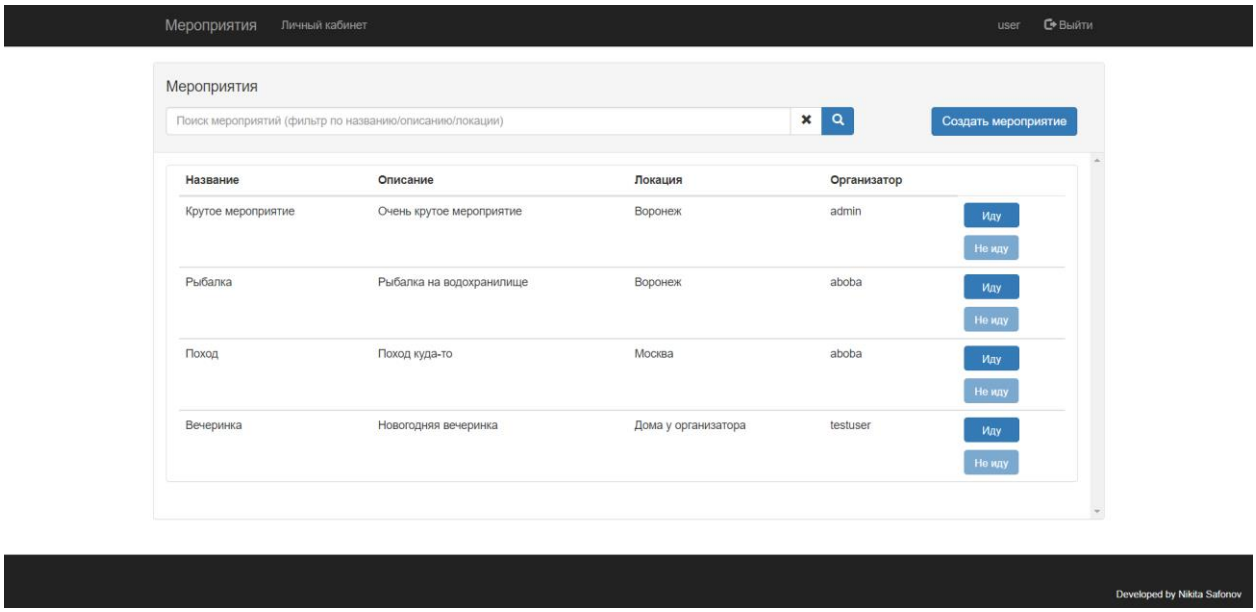

Рис. 6. Страница с мероприятиями (пользователь)

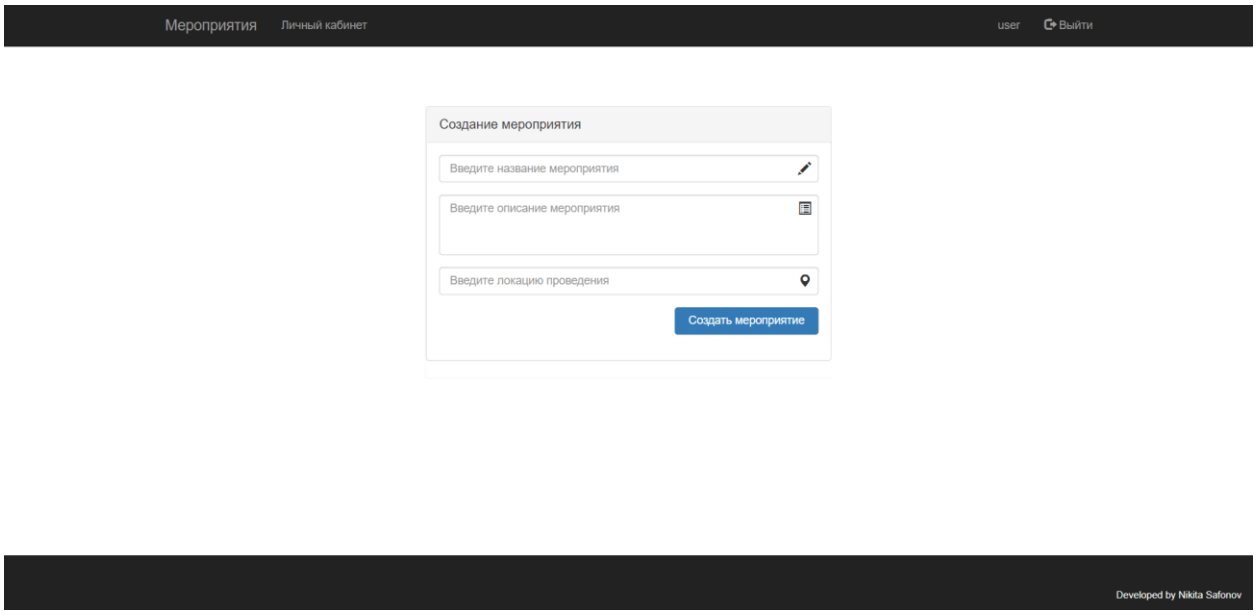

Рис. 7. Страница создания мероприятия

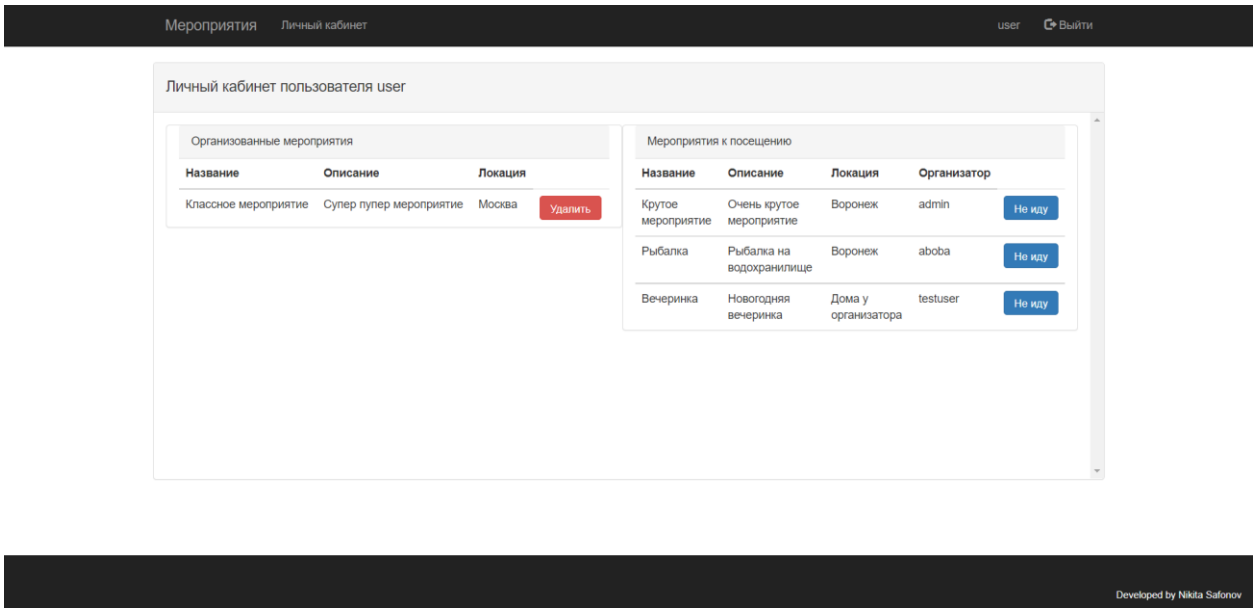

Рис. 8. Страница личного кабинета пользователя

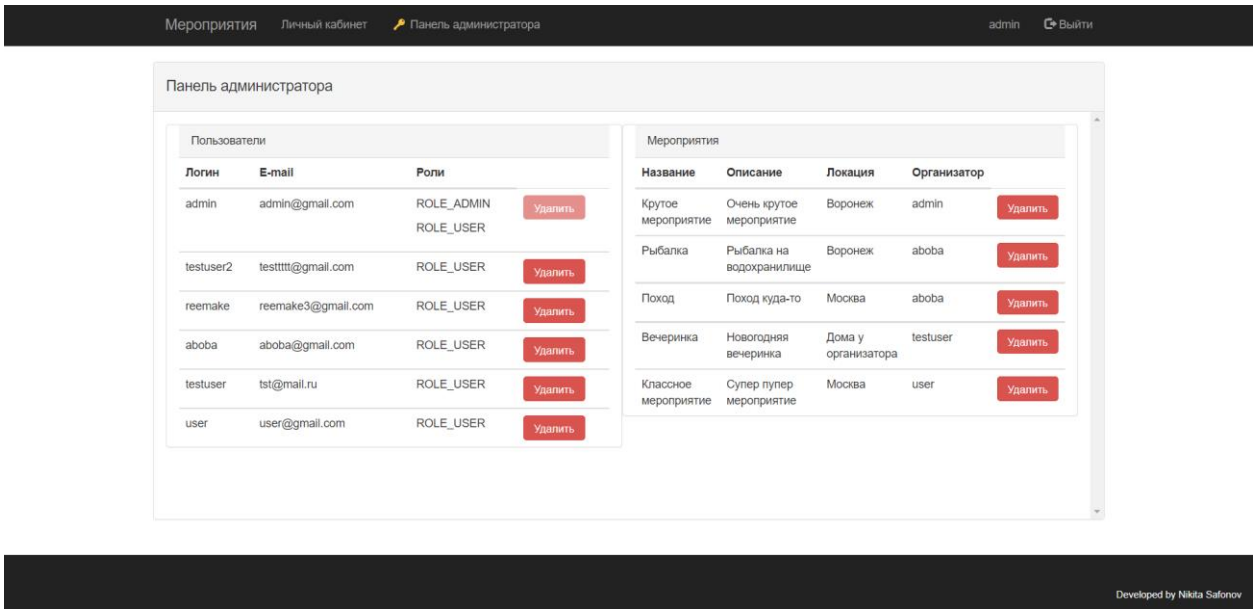

Рис. 9. Страница панели администратора

Реализация взаимодействия backend и frontend частей

Разработанное веб-приложение соответствует архитектурному шаблону передачи состояния представления REST. Это означает, что совершается обращение к ресурсам, которые представляются сущностями предметной области, с использованием уникального URI.

Используются следующие аннотации:

- @Autowired – отмечает конструктор, поле или метод как требующий автозаполнения инъекцией зависимости Spring;
- @GET, @POST – отображают HTTP методы;
- @RestController – используется для традиционных контроллеров Spring;
- @Repository – метка для классов, выполняющих роль хранилища;
- @Service – помечает класс, который выполняет бизнес-логику или вызов внешних API;
- @RequestMapping - используется для сопоставления веб-запросов с методами Spring Controller.

Заключение

Основной целью курса являлась разработка приложения, взаимодействующего с базой данных.

Реализации цели сопутствовало решение следующих задач:

- Изучение необходимого стека технологий;
- Проектирование и реализация базы данных приложения;
- Реализация компонентов приложения.

В ходе работы все задачи были успешно выполнены.