МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»

Кафедра №806 «Вычислительная математика и программирование»

**Курсовая работа**

**по курсу «Базы Данных»**

**Платформа для управления общественным транспортом**

Выполнил: Иванов И. М.

Группа: М8О-307Б-22

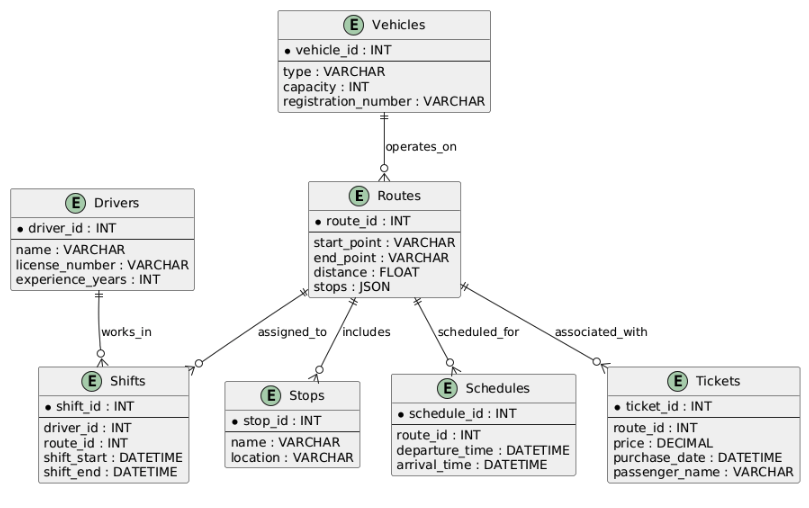
Преподаватель: Малахов А. В.

Москва, 2024

## **Схема базы данных**

Таблицы:

1. **routes** - содержит информацию о маршрутах, включая начальную и конечную точки, длину маршрута, данные об остановках (в формате JSON) и идентификатор транспортного средства, назначенного на маршрут,
2. **drivers** - содержит информацию о водителях, включая имя, номер водительской лицензии и количество лет опыта.,
3. **vehicles** - содержит информацию о транспортных средствах, включая тип, вместимость и регистрационный номер
4. **stops** - cодержит информацию об остановках, включая название, местоположение и идентификатор маршрута, к которому относится остановка,
5. **schedules** - содержит информацию о расписании движения по маршрутам, включая идентификатор маршрута, время отправления и время прибытия,
6. **tickets** - содержит информацию о билетах, включая идентификатор маршрута, цену, дату покупки и имя пассажира, на которого оформлен билет,
7. **shifts** - содержит информацию о сменах водителей, включая идентификатор водителя, маршрут, начало и окончание смены.



**Схема приложения. Модели**

Приложение состоит из следующих компонентов:

1. **Frontend (Streamlit)**  
   Пользовательский интерфейс реализован с использованием библиотеки Streamlit.  
   Приложение предоставляет следующие страницы:
   * **Маршруты**: позволяет получить расписании с водителями;
   * **Купленные билеты**: отображает информацию о купленных билетах
   * **Добавить покупку (только для администраторов)**: позволяет добавлять информацию о купленных билетах с именами пассажиров;
   * **Назначить водителя**: автоматически назначает водителей на маршруты на основе их опыта и времени смен.

Каждая страница имеет отдельный файл для логики и взаимодействия с пользователем.

1. **Backend:**  
   Взаимодействие с базой данных происходит через слой репозиториев. Репозитории содержат функции для выполнения SQL-запросов, таких как выборка данных, обновление остатков товаров, изменение статусов заказов и пополнение баланса.  
   Backend структурирован следующим образом:
   * **repositories**: модуль для работы с БД, включает файлы для управления продуктами, заказами, пользователями.
   * **services**: уровень бизнес-логики, где выполняются вычисления и операции с данными перед передачей в репозиторий.
2. **База данных**  
   База данных разработана в соответствии с нормализацией. Основные таблицы включают users, routes, drivers, vehicles, stops, schedules, tickets, shifts.

**Описание основных моделей:**

В приложении используются следующие модели данных:

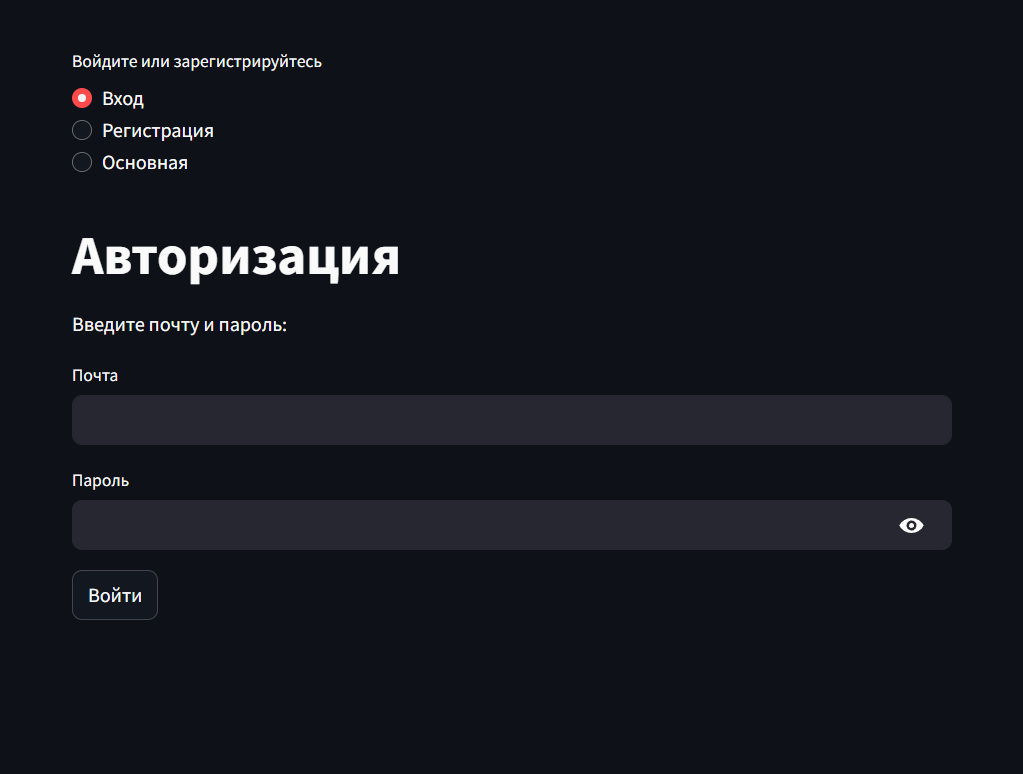
* **Пользователь (User)**: описывает данные о пользователе, включая электронную почту, баланс и роль (администратор/пользователь).
* **Маршрут (Route):** хранение информации о маршрутах общественного транспорта
* **Водитель (Driver)**: Хранение данных о водителях, работающих на маршрутах.
* **Транспортное средство (Vehicle):** Хранение данных о транспорте, используемом для перевозок.

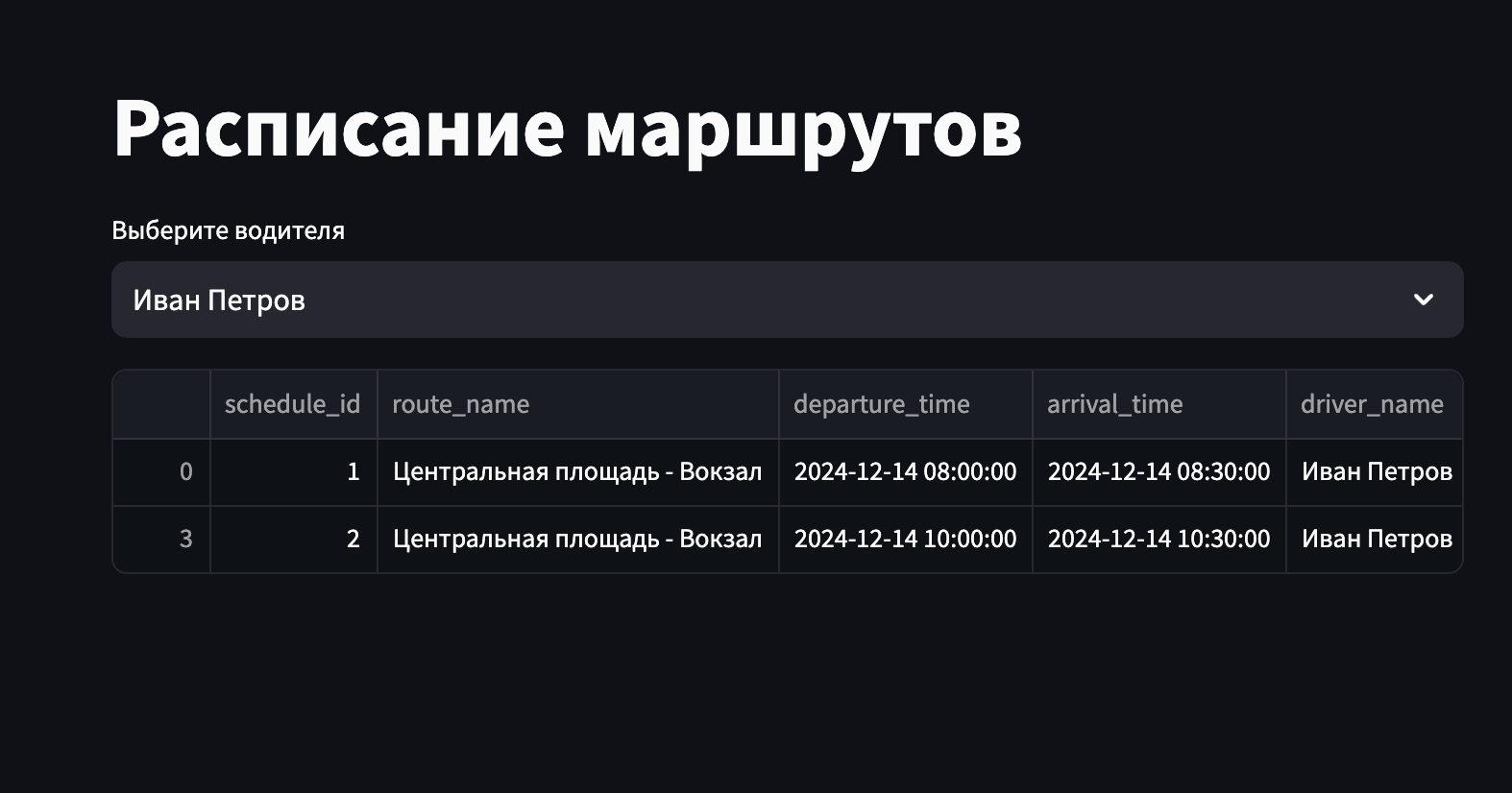
**Краткое описание работы приложения:**

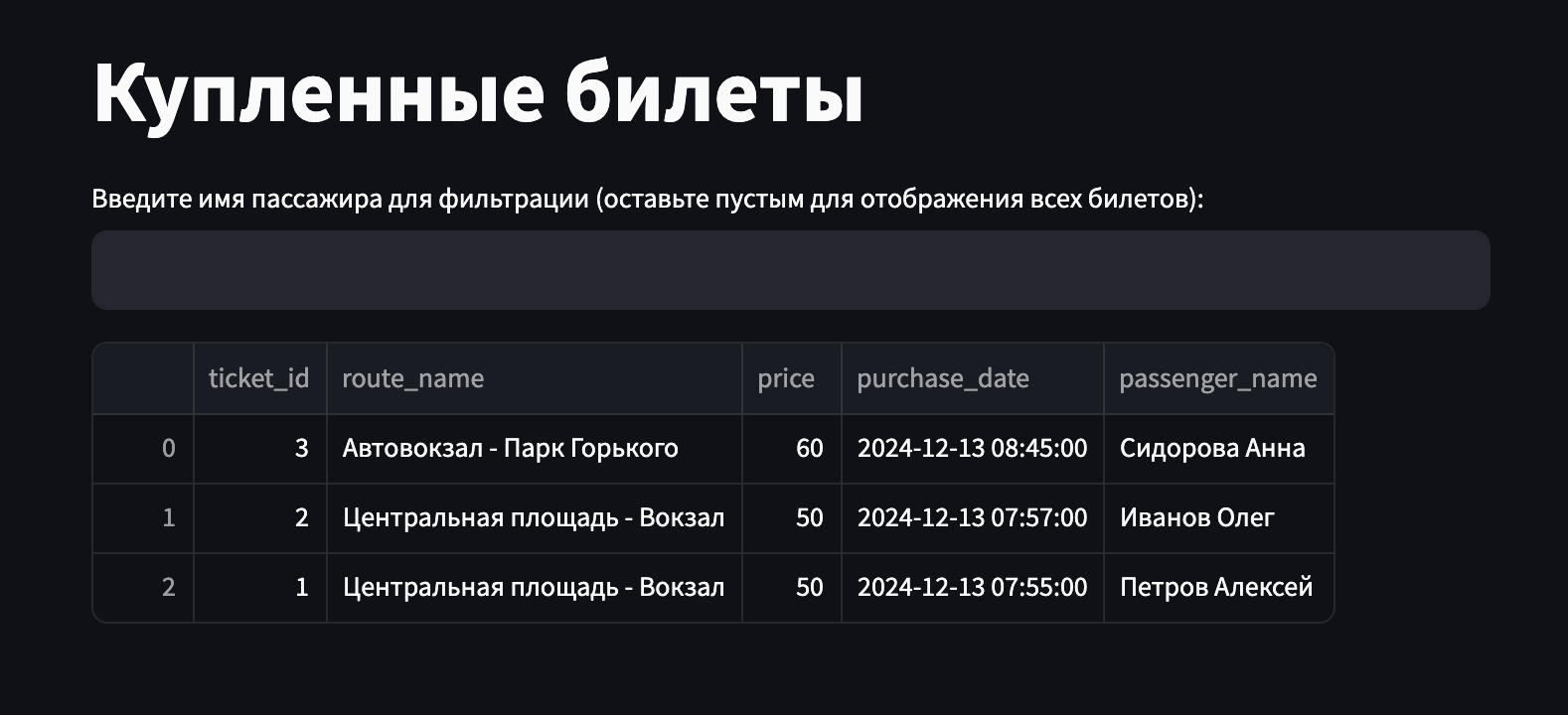
1. **Авторизация:**  
   Пользователь вводит email, который проверяется в базе данных. Если email найден и пароль верен, приложение предоставляет доступ к функционалу пользователя или администратора в зависимости от роли.
2. **Функционал пользователя:**
   * Просмотр маршрутов с остановками, водителями.
   * Просмотр купленных билетов: цена, дата, кто купил.
   * Назначение водителя на подходящие маршруты.
3. **Функционал администратора:**
   * Добавление купленных билетов: цена, количество, кто купил.

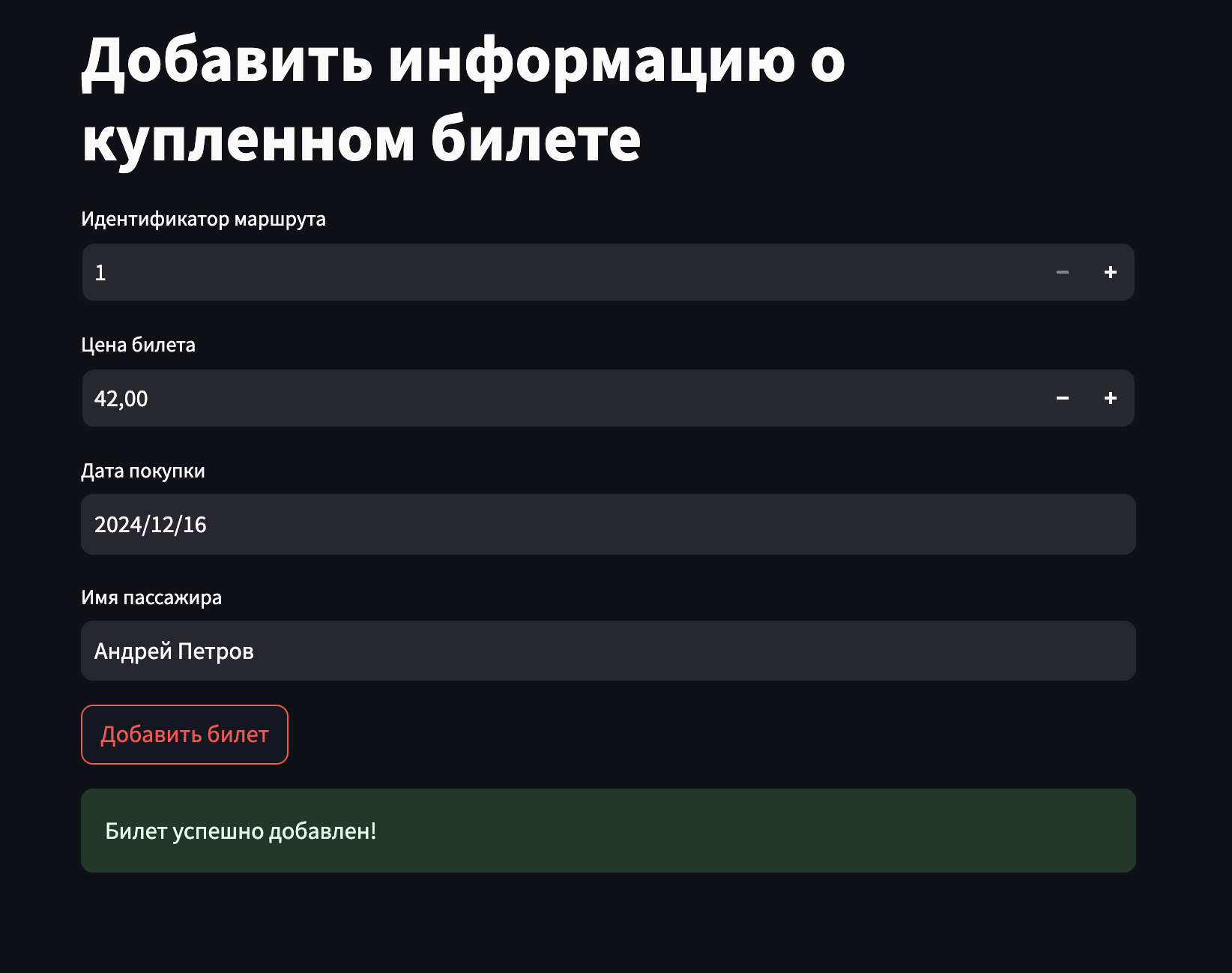
**Скриншоты из приложения**

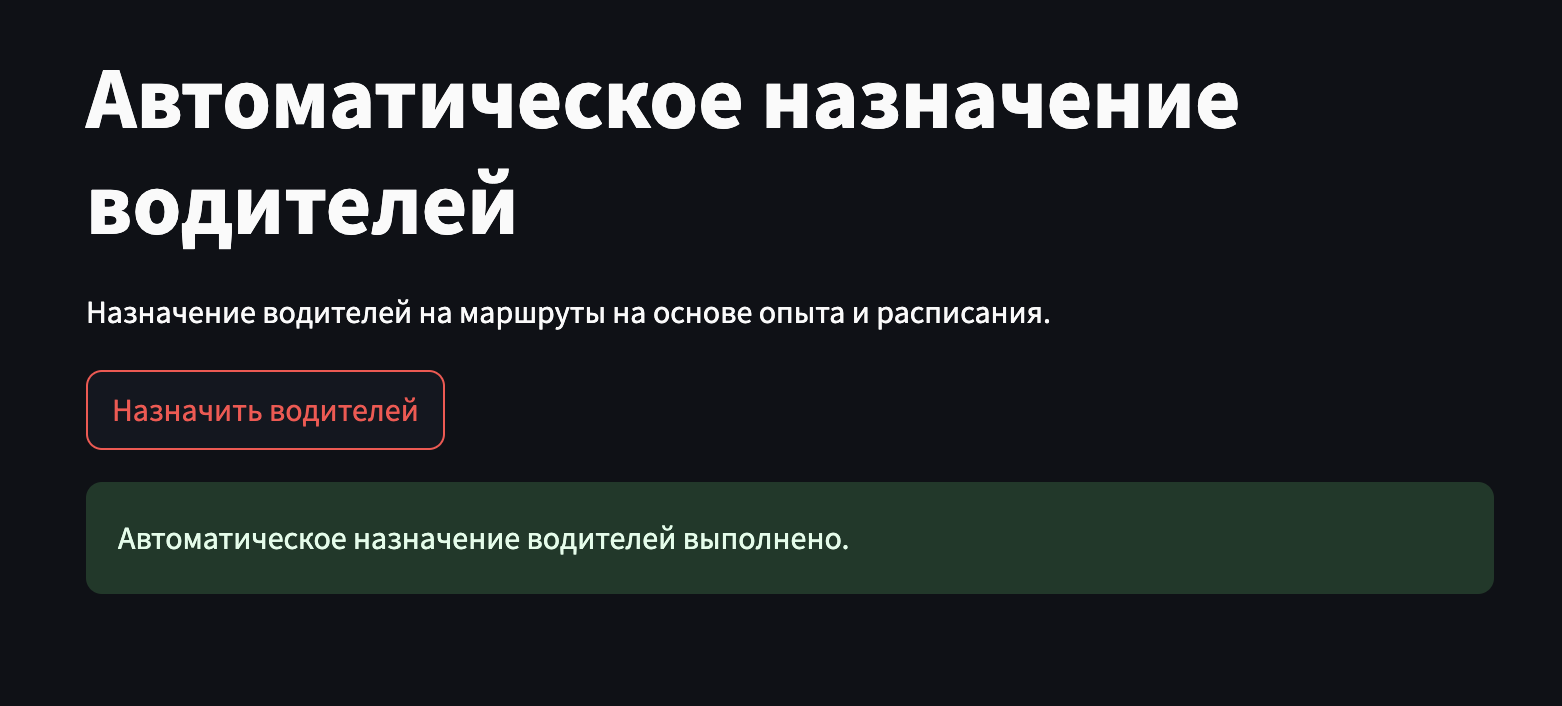
Окно авторизации



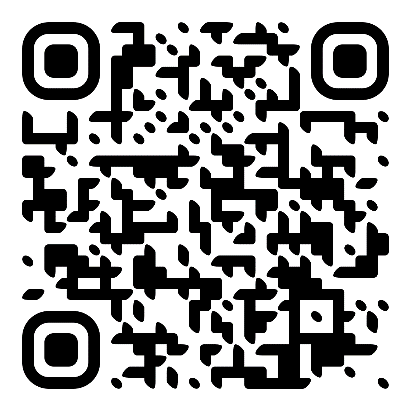
Страница с расписанием маршрутов  


Купленные билеты  


Добавление покупок билетов (только для администратора)  


Назначение водителей  


**Github QR**

****

**Вывод**

В ходе работы над курсовым проектом я познакомился с библиотекой streamlit и создал с помощью нее базовый дизайн сайта. Полученные на лекциях по базам данных знания, я применил для создания 7 таблиц и их заполнения. С помощью python, я обращался к БД для получения необходимой информации. В результате я смог реализовать платформу для управления общественным транспортом.