

**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»**

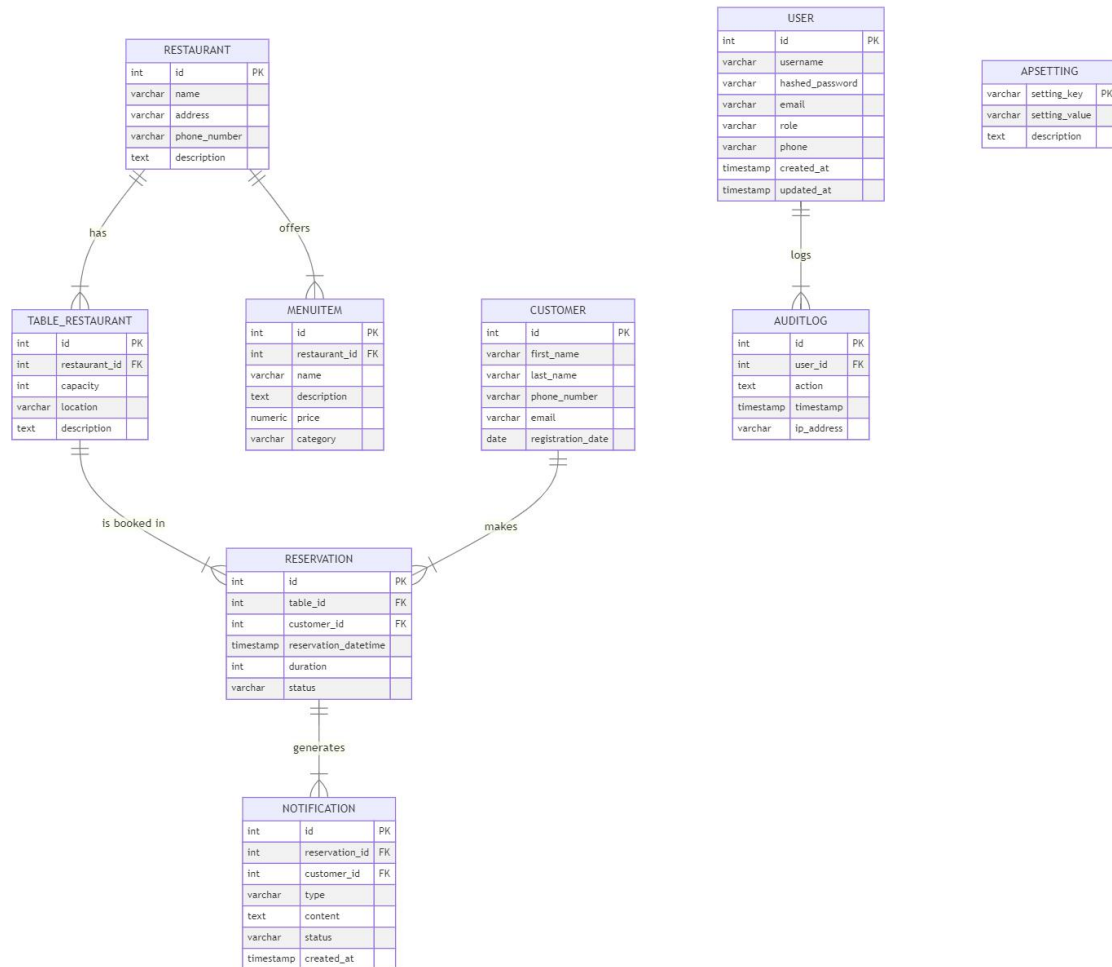
Система управления бронированием столиков в ресторане

**Выполнил: Н.С. Сарайкин
Группа: М8О-306Б-22
Преподаватель: А.В. Малахов**

Москва, 2024

Схема БД

Схема



Модели предметной области

Restaurant (Ресторан) - Хранит общую информацию о ресторане (название, адрес, контактные данные и т. п.).

Table_Restaurant (Столик) - Представляет столы в ресторане, которые могут быть забронированы.

Customer (Клиент) - Хранит данные клиентов, которые совершают бронирование.

Reservation (Бронирование) - Основная сущность для хранения информации о том, какой клиент, какой столик и на какое время забронировал.

MenuItem (Позиция меню) - Модель для описания блюд и напитков, которые могут быть предложены в рамках бронирования. Не всегда обязательно для бронирования, но в ряде систем позволяет клиентам изучить и частично заказать меню заранее.

Модели уровня инфраструктуры

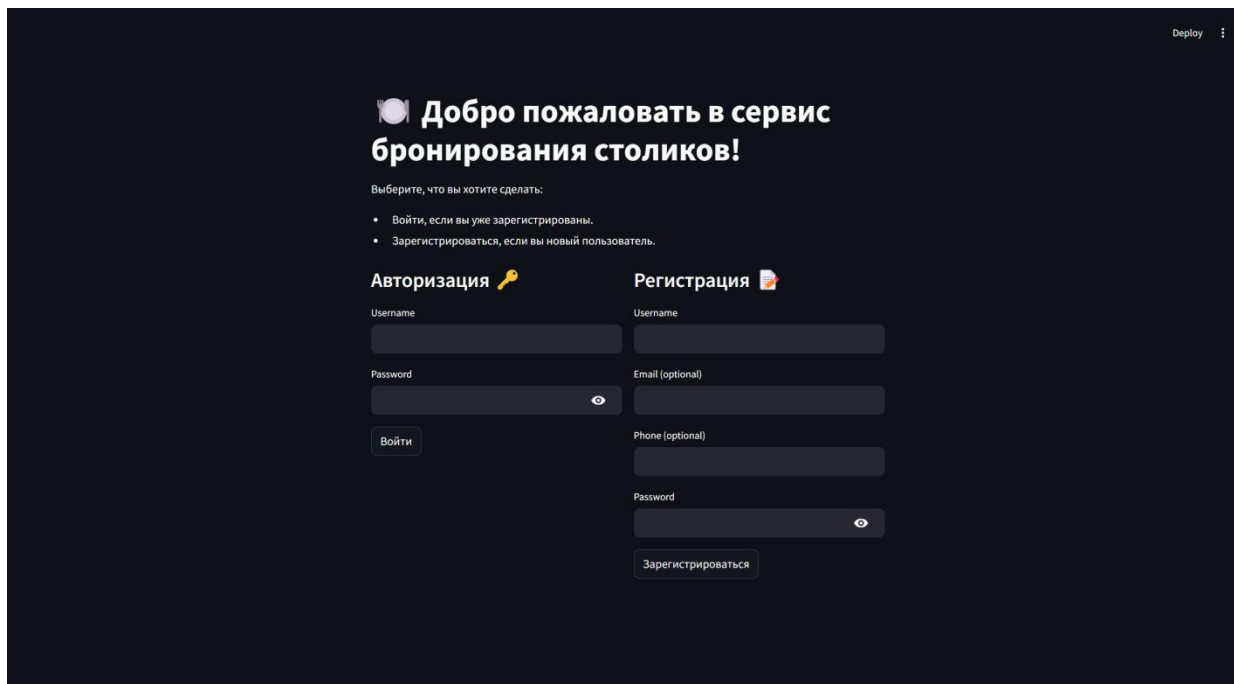
User (Пользователь системы) - Отвечает за аутентификацию и авторизацию пользователей (включая администраторов, редакторов и рядовых пользователей).

AuditLog (Логирование аудита) - Хранит логи действий, совершаемых пользователями (например, кто создал или отменил бронирование, поменял стол и т. д.).

Notification (Уведомление) - Модель для создания и отправки уведомлений (e-mail, SMS, push) при создании/отмене брони или других событиях. Динамически обновляет состояние брони.

AppSetting (Настройки приложения) - Хранит глобальные настройки для системы (часы работы ресторанов по умолчанию, максимальное время бронирования и т. д.).

Демонстрация работы приложения



The screenshot shows a dark-themed web interface for a table reservation service. At the top right, there is a 'Deploy' button with a dropdown arrow. The main heading is 'Добро пожаловать в сервис бронирования столиков!' (Welcome to the table reservation service!). Below this, a prompt 'Выберите, что вы хотите сделать:' (Choose what you want to do:) is followed by two bullet points: 'Войти, если вы уже зарегистрированы.' (Login if you are already registered.) and 'Зарегистрироваться, если вы новый пользователь.' (Register if you are a new user.). The interface is divided into two columns: 'Авторизация' (Login) on the left and 'Регистрация' (Registration) on the right. The login form includes 'Username' and 'Password' fields with a 'Войти' (Login) button. The registration form includes 'Username', 'Email (optional)', 'Phone (optional)', and 'Password' fields, with a 'Зарегистрироваться' (Register) button. Password fields have an eye icon to toggle visibility. A 'Deploy' button is located in the top right corner.

Рисунок 1 - Авторизация/Аутентификация

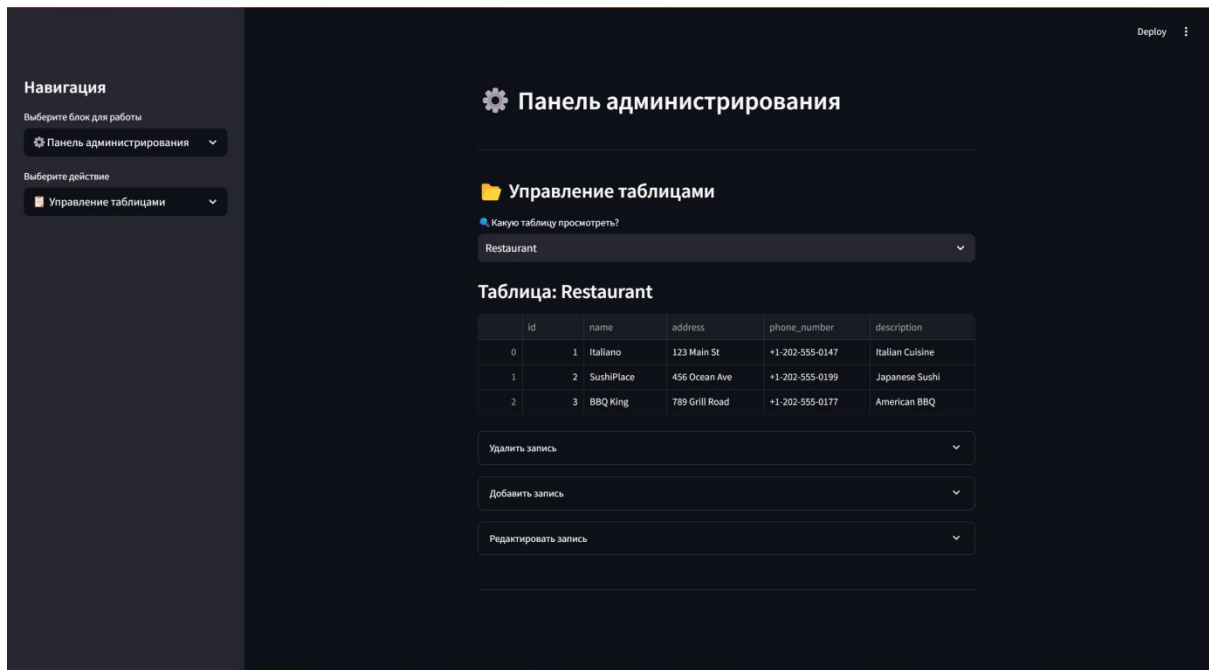


Рисунок 2 - Панель Администрирования

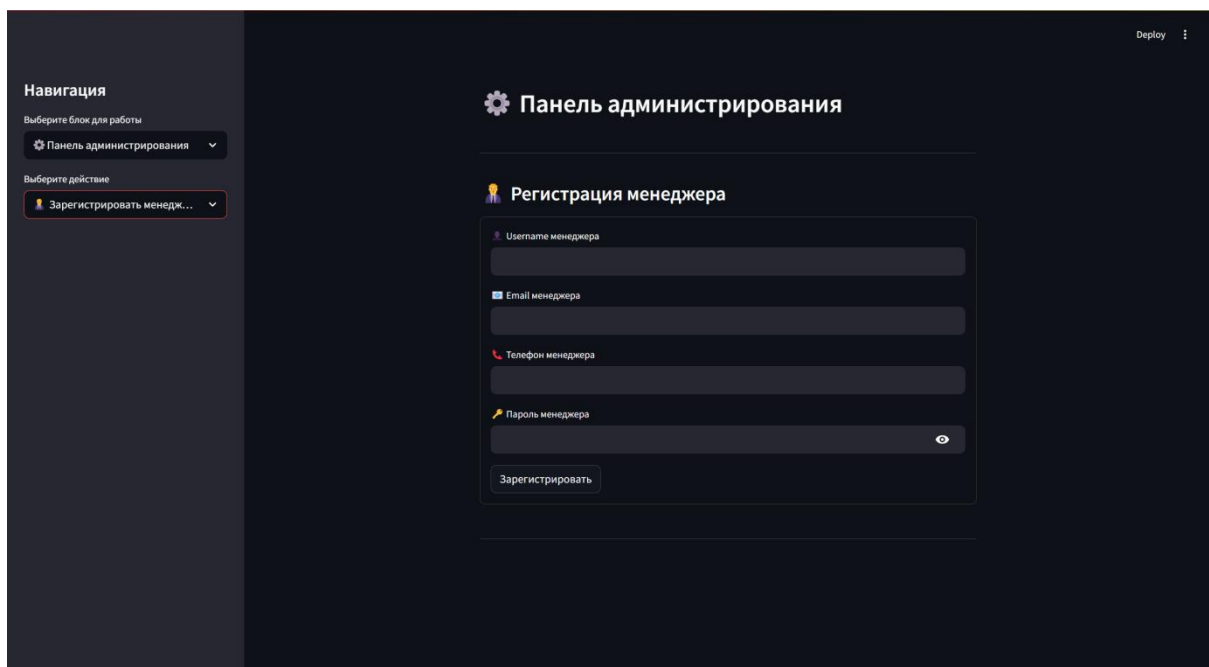


Рисунок 3 - Регистрация менеджера

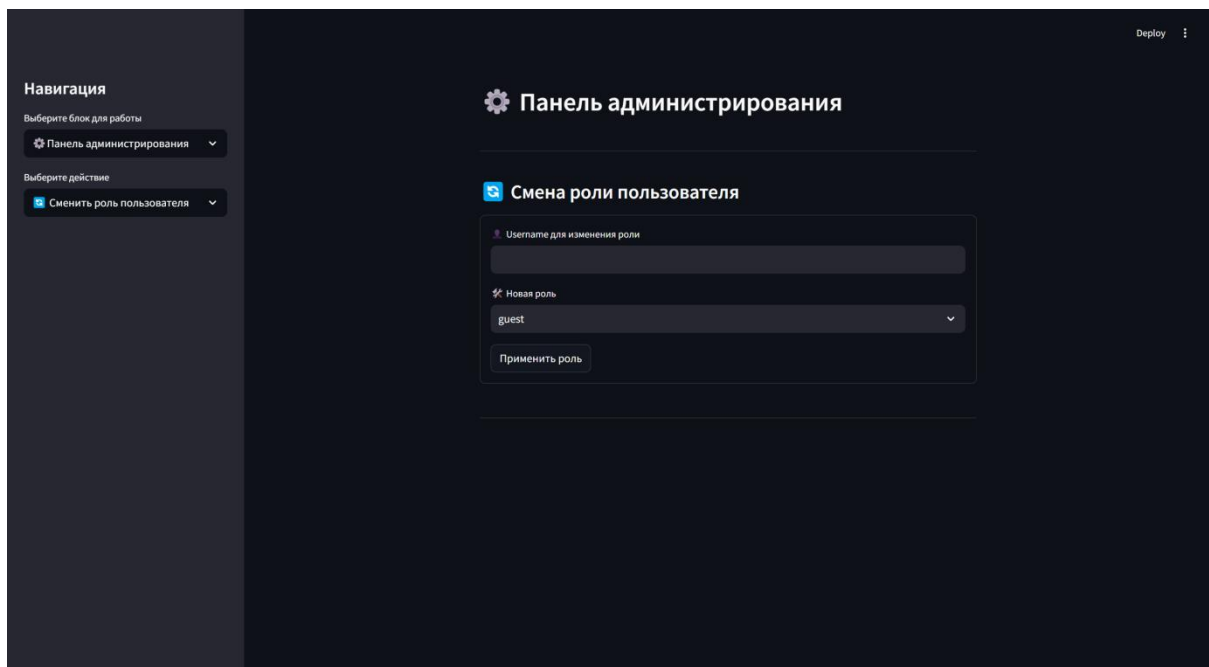


Рисунок 4 - Смена роли пользователя

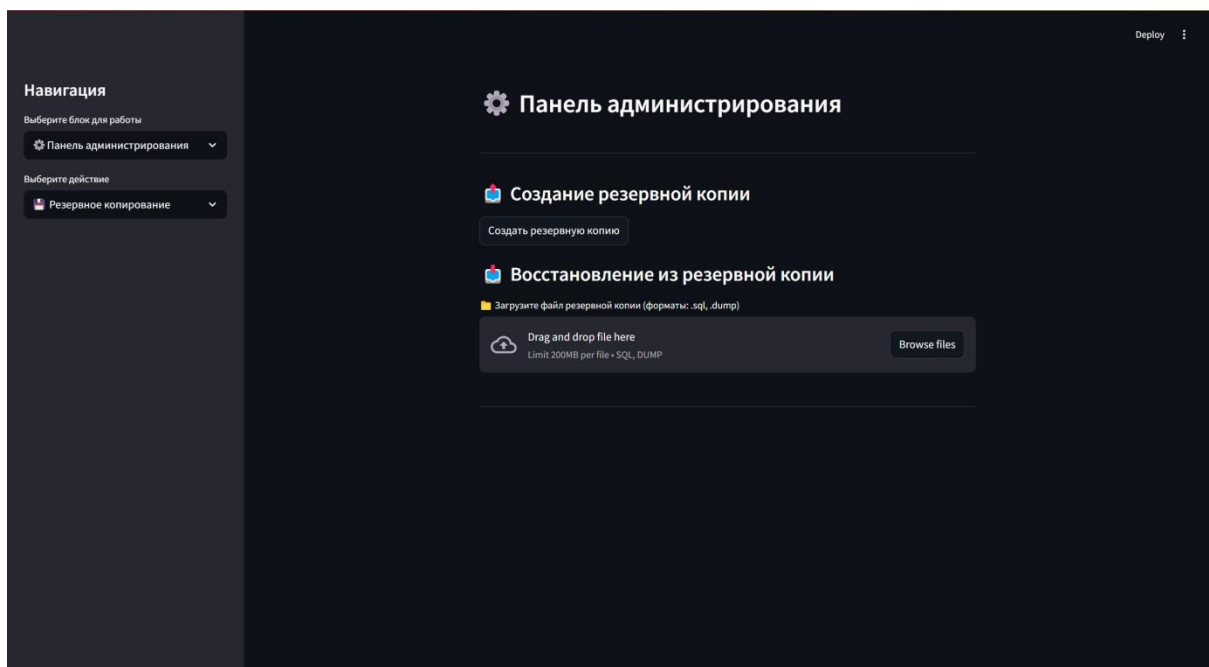


Рисунок 5 - Бэкап и восстановление

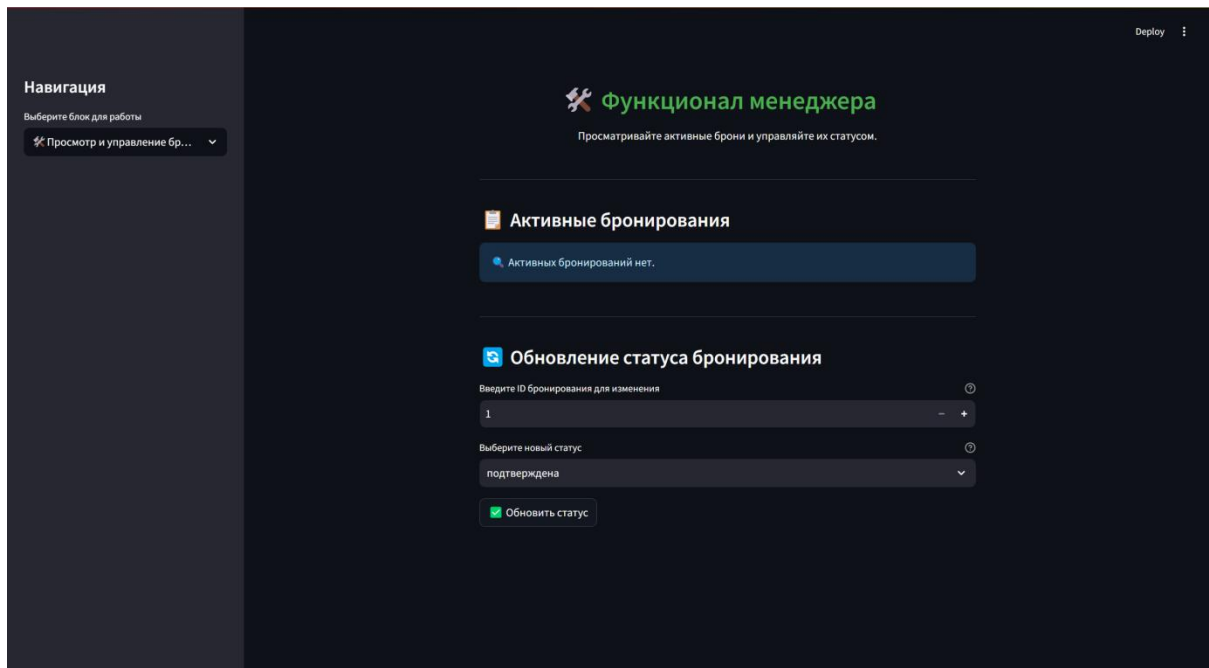


Рисунок 6 - Панель Менеджера

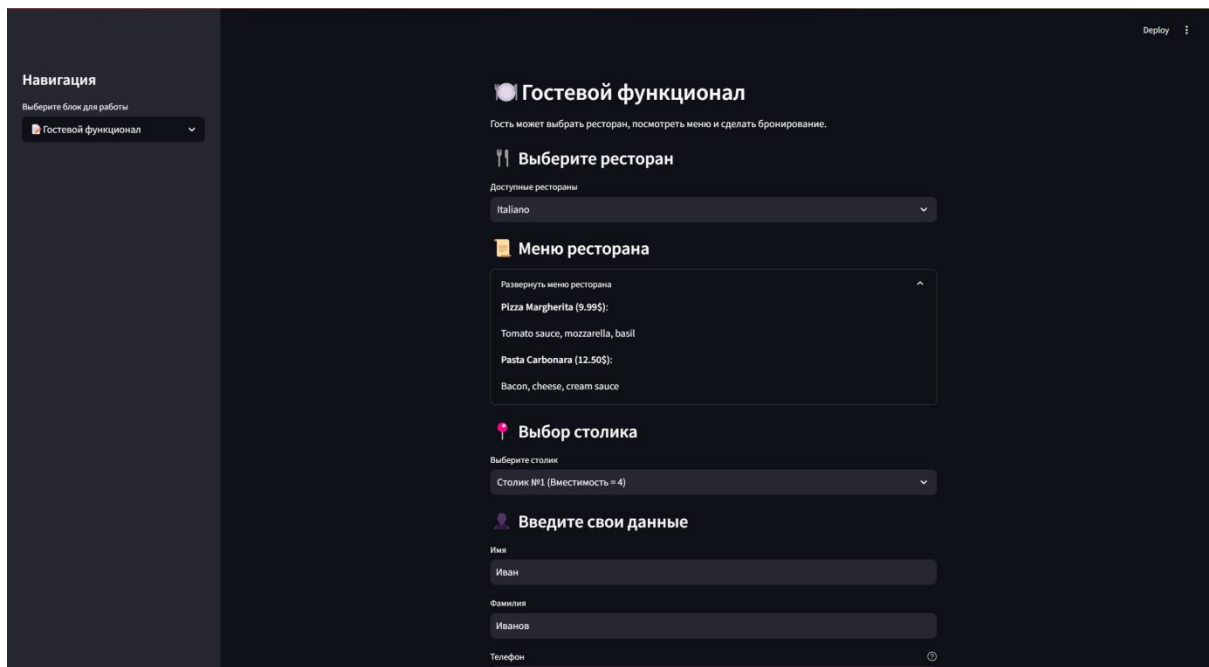


Рисунок 7 - Гостевой функционал №1

Навигация

Выберите блок для работы

Гостевой функционал

Выберите столлик

Столлик №1 (Вместимость = 4)

Введите свои данные

Имя

Иван

Фамилия

Иванов

Телефон

+7-900-000-0000

E-mail

ivanov@example.com

Данные бронирования

Дата бронирования

2024/12/21

Время бронирования

19:00

Продолжительность (мин.)

120

Забронировать

Рисунок 8 - Гостевой функционал №2

Описание проделанной работы

- 1. Разработана База Данных** - созданы таблицы инициализации `ddl.sql` и заполнения при инициализации `dml.sql`. Определены связи между таблицами для обеспечения целостности, также таблицы находятся в 3NF.
- 2. Реализовано Веб-приложение** на основе Streamlit, где логика обработки данных происходит на стороне серверного приложения, например, обработка `sql`-запросов. Реализована авторизация/аутентификация, панель администратора и бэкап с возможностью восстановления базы данных через резервную копию.
- 3. Разработан удобный и интуитивно понятный интерфейс** во front-end части с помощью Streamlit и Python.
- 4. Триггеры и представления (views)** активно используются в веб-приложении, причем их использование оправдано и осмыслено.
- 5. Безопасность и логгирование** - все действия по бронированию логируются для анализа изменений и политики безопасности.
- 6. Контейнеризация компонентов веб-приложения при помощи Docker** - веб-приложение легко развернуть на любом хостинге.

QR-код на репозиторий в GIT

