

ТЕХНИЧЕСКОЕ ЗАДАНИЕ ПО СОЗДАНИЮ САЙТА ОТЕЛЯ

Структура сайта

В разрабатываемом приложении предполагается создание двух независимых интерфейсов пользователей: интерфейс пользователей-посетителей и интерфейс пользователя-администратора сайта. Для интерфейса администратора сайта необходима организация авторизованного доступа. Интерфейс пользователя-администратора обеспечивает администратору сайта возможность добавлять, изменять и удалять информацию о номерах и отзывах. Информация, добавляемая администратором, сохраняется в базе данных и отображается в соответствующих разделах пользовательского интерфейса приложения.

Пользовательский интерфейс

Структура пользовательского меню

На страницах сайта должно располагаться меню. Пункты меню отображаются на каждой странице сайта в виде строки ссылок. Должна быть возможность переходить по страницам без необходимости указывать адрес вручную в адресной строке.

Ниже представлена структура пользовательского меню:

Перечень страниц пользовательского интерфейса сайта

Главная страница сайта

Страница с отзывами гостей

Страница со списком номеров

Страница бронирования номера

Главная страница сайта

Главная страница сайта содержит:

- меню навигации по сайту;
- фотографии номеров отеля;
- описание отеля.

Страница с отзывами гостей

Страница с отзывами гостей содержит:

- меню навигации по сайту;
- форму для добавления отзыва, содержащую следующие поля:
 - имя посетителя отеля;
 - электронный адрес клиента;
 - отзыв в виде текста.

- список добавленных отзывов от посетителей отеля.

Для добавления отзыва необходимо заполнить все поля. Данные, указанные в форме, сохраняются в базе данных для повторного отображения отзыва другим пользователям. Перед этим в интерфейсе администратора необходимо подтвердить отображение этого отзыва. Отзывы отображаются в нисходящем порядке от самого последнего добавленного.

Страница со списком номеров

Страница со списком номеров содержит:

- меню навигации по сайту;
- список номеров с кратким описанием каждого;
- ссылки для перехода к бронированию номера.

В списке номеров отображается одно из изображений каждого номера и его краткое описание. В дополнение к описанию указана стоимость проживания за сутки. Перейти на страницу конкретного номера можно нажав на кнопку “Забронировать” или на названием номера. Редактирование информации о номере осуществляется с помощью интерфейса администратора.

Страница бронирования номера

Страница бронирования номера содержит:

- меню навигации по сайту;
- фотографии номера;
- подробную информацию о номере;
- форму для ввода данных для бронирования.

Интерфейс администратора:

Ниже представлено описание страниц раздела администратора.

Страница администрирования номеров отеля.

Страница “Номера” интерфейса администратора предназначена для добавления, удаления и редактирования информации о номерах отеля. Страница содержит:

- таблицу с добавленными номерами.

Таблица содержит информацию о названии номера, даты создания и последнего обновления номера. Также в таблице располагаются кнопки для удаления и перехода на страницу редактирования номера.

Страница добавления/редактирования номера отеля.

Страница предназначена для добавления нового номера или редактирования информации о существующем номере. Страница содержит:

- форму для заполнения данных о номере.

Страница администрирования отзывов.

Страница “Отзывы” интерфейса администратора предназначена для подтверждения и удаления отзывов посетителей. Страница содержит:

- список всех отзывов.

Отзывы расположены по убыванию даты добавления гостями отеля. Администратор может удалить отзыв, либо подтвердить его корректность. После этого, отзыв будет виден остальным посетителям сайта на странице “Отзывы” интерфейса клиента.

Страница администрирования заявок на бронирование.

Страница “Список заявок на бронирование” интерфейса администратора предназначена для подтверждения и удаления заявок на бронирование от посетителей. Страница содержит:

- список всех заявок.

Заявки расположены по убыванию даты добавления гостями отеля. Администратор может удалить заявку, либо подтвердить ее актуальность.

ОБЩИЕ ТРЕБОВАНИЯ

К запрашиваемым ресурсам должны применяться ограничения по выдаче. Так, обычный пользователь не должен видеть непроверенные администратором отзывы на странице. Разделение областей, доступных пользователю и администратору, должно быть видимым: разные URL пути, соответственно, разные пути в конфигурации routes и разные контроллеры.

Для администратора должна быть предусмотрена аутентификация. Необходимо производить авторизацию при попытках изменять информацию. Для реализации аутентификации рекомендуется использовать гем Devise. Будет плюсом использование гема Pundit, для авторизации доступа, если у администратора может быть несколько ролей.

Хранить загруженные изображения можно на сервере с проектом, либо в облачном хранилище (актуально при загрузке приложения на Heroku). Для работы с загрузкой и сохранением изображений предлагается использовать решение Active Storage или гем CarrierWave.

Требуется реализовать возможность формировать CSV и, с помощью подручных гемов, Excel или PDF отчеты со статистикой о подтвержденных бронированиях. Генерация таких файлов должна происходить в background (фоновой) задаче, реализованной с помощью Active Job и гема Sidekiq. После этого, ссылка для скачивания файла должна быть доступна в интерфейсе администратора.

Необходима реализация Unit-тестов для создаваемых объектов. Рекомендуется использование RSpec гема для написания тестов, а также проверка уровня покрытия с помощью гема SimpleCov (рекомендуемый минимальный уровень покрытия — 85%).

Для поддержания “чистоты” кода рекомендуется использовать геммы Rubocop, Rubocop-Rails.

ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Для функционирования разрабатываемого сайта необходимо выполнение следующих обязательных требований:

- поддержка PostgreSQL 13 или выше;
- поддержка Ruby 3.0 или выше, Ruby On Rails 6 или выше;
- развертывание приложения на Heroku;
- реализация двух интерфейсов: для пользователя и администратора;
- добавление минимального количества стилей и JS кода на страницы;
- формирование файлов со статистикой о подтвержденных бронированиях;
- возможность загрузки файлов;
- наличие тестов.

Будет плюсом:

- добавление авторизации доступа и нескольких ролей администратора;
- добавление конфигурации для запуска Docker контейнеров;
- генерация файлов в нескольких форматах в фоновой задаче;
- добавление на страницу информации о погоде или курсе валют из доступной

API системы;

- выполнение минимального уровня покрытия тестами;
- реализация frontend части на одном из популярных фреймворков (React, Vue, Angular), и реализация ответов контроллеров в формате JSON (если имеются навыки работы с соответствующими фреймворками).