**Частное учреждение профессиональная образовательная организация**

**ТЕХНИКУМ «БИЗНЕС И ПРАВО»**

**ОТЧЕТ О ПРЕДДИПЛОМНОЙ ПРАКТИКЕ ПО**

**ПМ 09. ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА И ОПТИМИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЙ**

*практика по профилю специальности*

* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*наименование организации*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

в период с «\_\_» \_\_\_\_\_\_\_\_\_ 202\_\_ г. по «\_\_» \_\_\_\_\_\_\_\_\_ 202\_\_ г.

|  |  |
| --- | --- |
|  | обучающегося (йся)\_\_\_\_ курса очной формы обучения, группы \_\_\_\_\_\_\_\_\_\_\_\_  специальности 09.02.07 Информационные системы и программирование  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *фамилия, имя, отчество*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Руководитель практики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *фамилия, имя, отчество*  Дата\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

г.Белореченск

Содержание

[ВВЕДЕНИЕ 3](#_Toc166809696)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 5](#_Toc166809697)

[1.1 Анализ существующих решений 5](#_Toc166809698)

[1.2 Проектирование базы данных 6](#_Toc166809699)

[1.3 Цели и задачи выполняемой системы 6](#_Toc166809700)

[1.4 Метод реализации процесса проектирования 6](#_Toc166809701)

[1.5 Обоснование проектных решений 7](#_Toc166809702)

[2. ПРОЕКТНАЯ ЧАСТЬ 15](#_Toc166809703)

[2.1 Техническое задание 15](#_Toc166809704)

[2.2 Программное обеспечение информационной системы 15](#_Toc166809705)

[2.3 Руководство пользователя 16](#_Toc166809706)

[2.4 Руководство администратора 18](#_Toc166809707)

[ЗАКЛЮЧЕНИЕ 20](#_Toc166809708)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 21](#_Toc166809709)

[ПРИЛОЖЕНИЕ 22](#_Toc166809710)

# ВВЕДЕНИЕ

В современном мире технологий и информатизации ресторанный бизнес не остается в стороне и активно внедряет автоматизированные информационные системы для управления бизнес-процессами. Одной из таких систем является система бронирования столов в ресторане, которая обеспечивает удобство как для посетителей, так и для персонала.

Развитие информационных технологий и распространение мобильных устройств позволяет создавать эффективные и удобные решения для ресторанного бизнеса. Автоматизированные системы бронирования столов становятся неотъемлемой частью современных ресторанов, обеспечивая более гладкую и эффективную работу заведения. Эти системы позволяют клиентам быстро и удобно забронировать столик, не тратя время на звонки и ожидание, а также обеспечивают персонал точной и актуальной информацией о бронированиях.

Целью данного отчета является разработка автоматизированной информационной системы бронирования столов в ресторане. В рамках данного проекта будет рассмотрена архитектура системы, ее основные функциональные возможности, преимущества и перспективы внедрения.

Отчет включает в себя анализ требований к системе, проектирование архитектуры, разработку и реализацию программного обеспечения, а также тестирование и оценку эффективности системы. Особое внимание будет уделено пользовательскому интерфейсу, который должен быть интуитивно понятным и удобным как для клиентов, так и для сотрудников ресторана.

Разработка автоматизированной информационной системы бронирования столов в ресторане имеет большое практическое значение, поскольку позволяет улучшить качество обслуживания клиентов, оптимизировать рабочие процессы и повысить конкурентоспособность ресторанного бизнеса. Внедрение такой системы способствует сокращению числа ошибок при бронировании, улучшению управления потоком клиентов и повышению общей эффективности работы ресторана.

Кроме того, автоматизированная система бронирования предоставляет возможность анализа данных и получения ценных инсайтов о предпочтениях клиентов, загруженности ресторана и эффективности работы персонала. Это позволяет руководству ресторана принимать обоснованные решения, направленные на улучшение сервиса и увеличение прибыли.

В отчете также будут рассмотрены потенциальные трудности и риски, связанные с внедрением системы, а также предложены пути их преодоления. В частности, будет анализироваться необходимость обеспечения безопасности данных, защита от несанкционированного доступа и возможность масштабирования системы для поддержки растущих объемов данных и пользователей.

Таким образом, данный проект направлен на создание инновационного инструмента, который не только удовлетворяет текущие потребности ресторана, но и обладает потенциалом для адаптации к будущим изменениям и требованиям рынка.

# 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

## Анализ существующих решений

* Перед разработкой новой системы бронирования столов в ресторане необходимо провести анализ существующих решений на рынке. Этот этап позволит выявить особенности существующих систем, их преимущества и недостатки, а также определить требования и ожидания пользователей.
* На рынке существует ряд коммерческих и бесплатных решений для бронирования столов в ресторанах. Среди них можно выделить следующие основные типы систем:
* Веб-платформы и приложения для бронирования столов. Это онлайн-сервисы, которые позволяют пользователям совершать бронирование столов через интернет. Примерами таких платформ являются OpenTable, Bookatable, Resy и др.
* Системы управления рестораном с функциональностью бронирования столов. Это комплексные информационные системы, которые включают в себя модуль управления бронированием столов вместе с другими функциями, такими как учет посетителей, управление меню, финансовое управление и т.д. Примерами таких систем являются Micros (Oracle Hospitality), Toast, Lightspeed и др.
* Локальные системы бронирования столов. Это программное обеспечение, устанавливаемое на серверы ресторана и предназначенное для внутреннего использования. Такие системы могут быть разработаны на заказ или приобретены у специализированных поставщиков.
* Анализ существующих решений поможет определить лучшие практики и функциональные возможности, которые следует включить.

## 1.2 Проектирование базы данных

В предоставленной работе приведена спроектированная база данных для положения. Это включает в себя определение структуры базы данных, выбор подходящего типа данных и структуры данных, а также проектирование отношений между таблицами.

## 1.3 Цели и задачи выполняемой системы

Основные цели системы:

* Разработать удобное веб-приложение для бронирования столов в ресторане
* Реализовать функциональность пользователей.

Задачами являются:

* Достичь поставленных целей;
* Реализовать функционал бронирование столов.

## 1.4 Метод реализации процесса проектирования

При проектировании веб-приложения, которое использует базу данных для вывода данных, можно использовать следующий подход:

1) Анализ требований:

* Определение какие данные необходимо получить из базы данных.
* Определить как эти данные должны быть представлены в пользовательском интерфейсе.

2) Интеграция с базой данных:

* Использование PHP для создания запросов к базе данных и получения данных.
* Обработка ответов от базы данных.

3) Отображение данных:

* Используется HTML, CSS и Bootstrap для создания использования пользовательского интерфейса.
* Выделение полученных данных на страницу с использованием PHP.

## 1.5 Обоснование проектных решений

Использование определенного стека технологий – можно обосновать использование PHP, Bootstrap, CSS и HTML для разработки веб-приложения. это обеспечивает кросс-браузерную совместимость, легкость в изучении и использовании, а также широкую допустимость библиотек и фреймворков.

Использование определённой архитектуры – MVC(Model-View-Controller) для организации кода и его масштабируемости.

Система MVC архитектуры, представленных на рисунке 1:

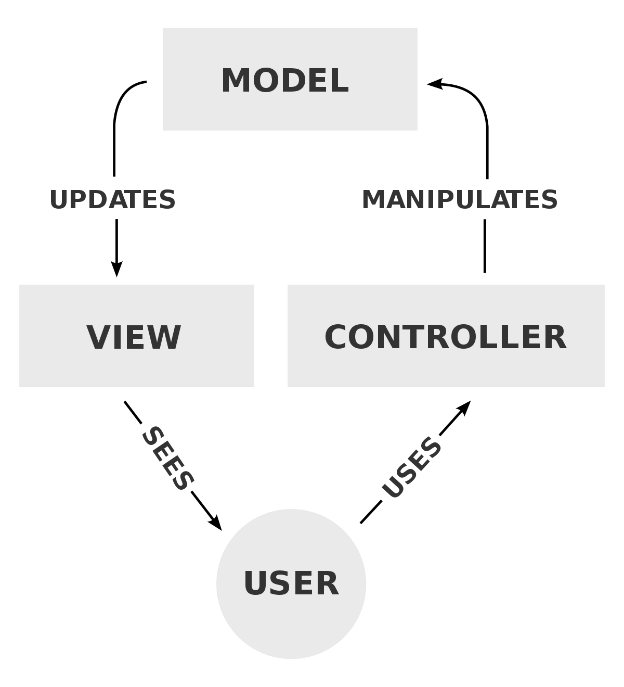


Рисунок 1 – MVC архитектура

Выбор определенного способа хранения данных – MySQL, для хранения информации о пользователях, товарах и заказов.

Требования к системе представлены в таблице 1.1.

Таблица 1.1 – Требования к системе учебного курса

|  |  |  |
| --- | --- | --- |
| Атрибуты | Метки | Требования |
| 1 | 2 | 3 |
| Удобство | | |
| Простота использования | Среднее время, необходимое  пользователю, чтобы найти  необходимый элемент в программе | Поиск отдельного элемента интерфейса не должен занимать более 1 секунды |
| Привлекательность | Соответствие интерфейса  требованиям | 99% элементов должны быть работоспособны для  пользователя |
| Обучаемость | Показатель, затрачиваемый  пользователями на обучение | Пользователь должен изучить работу ПО за первые 3-5 минут |
| Производительность | | |
| Временная эффективность | Время выполнения компонента программы | - Приложение должно  обслуживать любую функцию не дольше 1 сек  (не включая задержки в сети); |
| Эффективность использования ресурсов | Объемы ресурсов требуемых для выполнения задач | - Постоянная память,  используемая программой – не более 10МБ;  - Приложение должно потреблять не более 3МБ  памяти на каждый неактивный сеанс с  пользователем; |
| Удобство установки | Легкость установки | Необходима установка:  apache, mysql, admin php |
| Способность к сосуществованию | Способность ПО сосуществовать с другими  программами в общем окружении | Программа не должна  занимать большие объемы памяти устройства (максимум 3МБ) |
| Удобство замены другого ПО данным | Совместимость структур данных | - Должна поддерживать  внедрение новых версий;  - Система должна быть применима вместо других  программных систем для решения тех же задач в определенном окружении. |

Требования к численности и квалификации персонала системы и режиму его работы:

В системе существуют 2 группы пользователей: Пользователь и Администратор. Пользователи выполняют функцию: просмотра столов, и меню.

Администратор имеет возможность использовать полный функционал CRUD (Создание\Просмотра\Удаление) различных данных из БД

Пользователи системы должны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows на уровне пользователя.

В таблице 1.2 представлены требования к надежности.

Таблица 1.2 – Требования к надёжности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибуты | Метрики | | Требования | |
| 1 | | 2 | | 3 |
| Надёжность | | | | |
| Устойчивость к отказу | Коэффициент аварийных | | - Не менее 90% ошибок | |
|  | Отказов | | в программе должны  обрабатываться без  экстренного завершения;  - Обеспечение бесперебойного питание  активного сетевого  оборудования; | |
| Способность к восстановлению | Отношение количества числа успешного восстановление работы системы к ее аварийному завершению за определённый промежуток времени | | - Восстановление  работоспособности  системы не должно превышать 5 минут; | |
| Уровень зрелости | Среднее время работы без сбоев | | - Система должна  работать 24 часа в сутки;  - Система должна оповещать  - Система должна оповещать пользователя при некорректных действиях в системе; | |

Средняя доступность приложения должна составлять не менее 99%

Среднее время между сбоями — это среднее время, за которое компонент или модуль может выполнять свои функции без перерыва.

Измеряется от начала работы до момента следующего сбоя.

Среднее, временя работы без сбоев должно составлять не менее 1500 часов.

В таблице 1.3 представлен перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.

Таблица 1.3 – Перечень и критерии отказов для каждой функции

|  |  |
| --- | --- |
| Функции | Критерии отказа |
| 1 | 2 |
| Отображение данных | Отсутствие соединения с базой данных |
| Добавление | Отсутствие соединения с базой данных |
| Редактирование | Отсутствие соединения с базой данных |
| Удаление | Отсутствие соединения с базой данных |

Требования к функциям web-приложения, выполняемым программным обеспечением

Имя: Вход пользователя

Описание: Пользователь заходит на главную страницу

Действующий субъект: Пользователь

Предусловие: пользователь зашел на сайт для бронирования стола Сценарий:

* Пользователь на сайте
* Зашел на главную страницу
* Нашел нужный свободный стол

Перед вводом системы в действие необходимо:

– интернет-соединение;

– обеспечить каждого специалиста доступом к web-приложению;

– иметь данные для доступа к админ-панели;

– база данных MySQL.

Разрабатываемая система требует в своей архитектуре наличие БД-сервера со свободной реляционной системой управления – MySQL.

База данных информационной системы должна состоять из следующих связанных таблиц:

– menu\_categories– таблица, содержащая данные о категории еды.

– menu\_dishes– таблица, содержащая информацию о еде.

* reviews– таблица содержащая информацию о отзывах
* tables– таблица, содержащая информацию о столах.

Данные о категориях представлены в таблице menu\_categories 1.4.

Таблица 1.4 menu\_categories – Данные о категориях

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | int | 11 | Уникальный идентификатор | Первичный ключ |
| name | text | 11 | Название | NOT NULL |

Данные о еде представлены в таблице menu\_dishes 1.5.

Таблица 1.5 – menu\_dishes – Данные о еде.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| id | Int | 11 | Идентификатор | Первичный ключ |
| name | Varchar(255) | 255 | название | NOT NULL |
| description | text |  | описание | NOT NULL |
| price | int |  | Общая цена | NOT NULL |
| Category\_id | Int(11) |  | категория | NOT NULL |
| img | Varchar(255) |  | изображение |  |

Данные об отзывах представлены в таблице reviews 1.6.

Таблица 1.6 reviews – Данные об отзывах.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничения |
| Id | Int | 11 | Уникальный индентификатор | Первичный ключ |
| name | VARCHAR | 255 | ФИО | NOT NULL |
| rating | VARCHAR | 255 | оценка | NOT NULL |
| message | text |  | сообщение | NOT NULL |
| Created\_at | timestamp |  | Дата и время | NOT NULL |

Данные о коментариях должны размещаться в таблице tables 1.7.

Таблица 1.7 tables – Данные о столах

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничения |
| id | INT | 255 | Идентификатор | Превичный ключ |
| Name\_table | text | 255 | Имя стола | NOT NULL |
| quantity | text | 255 | количество | NOT NULL |
| status | text | 255 | статус | NOT NULL |
| img | text | 255 | изображение | NOT NULL |
| description | text |  | описание | NOT NULL |

На рисунке 1 представлено окно бронирование стола.

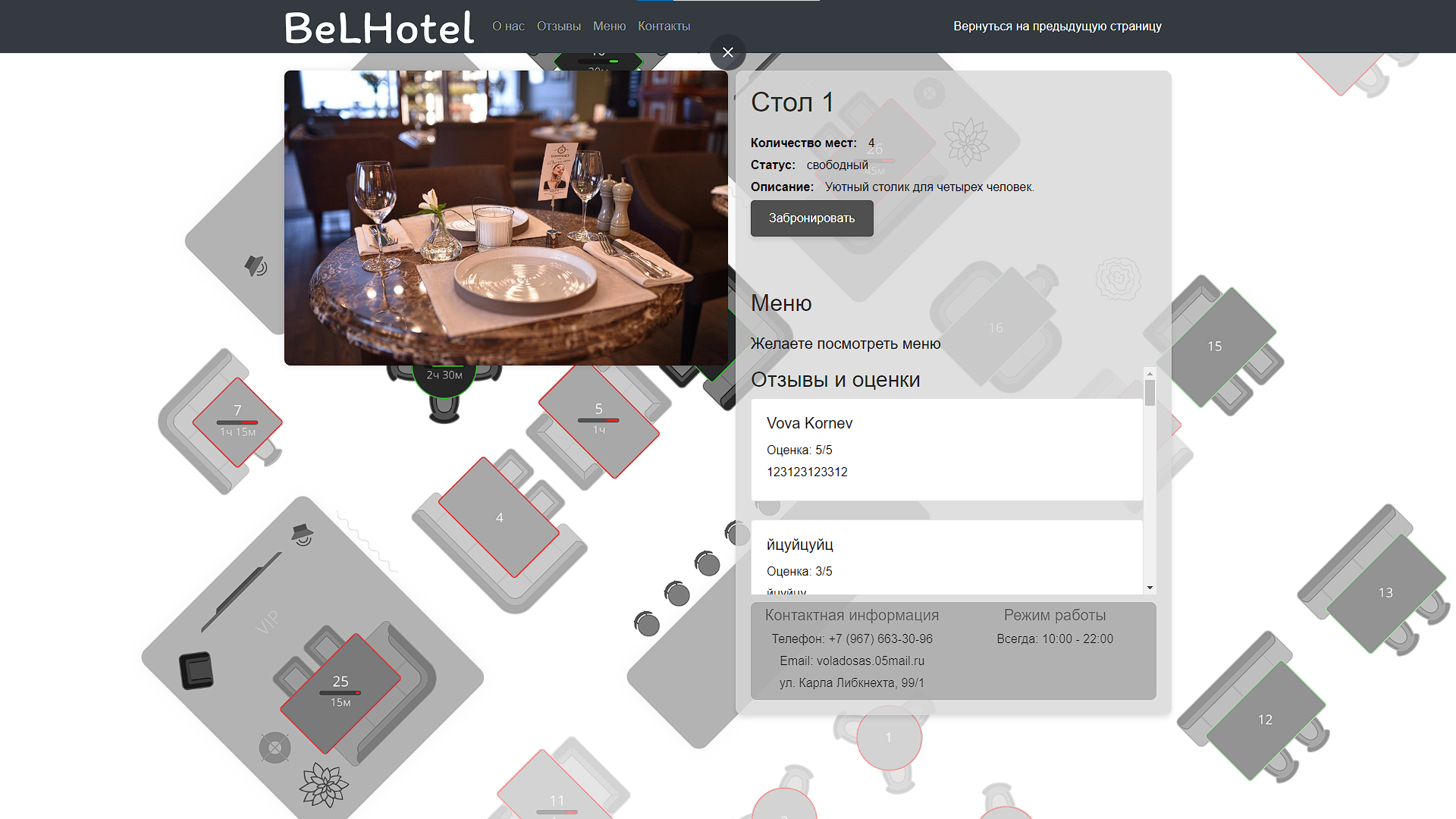


Рисунок 1. – окно стола.

На рисунке 2 представлено за бронирование стола.

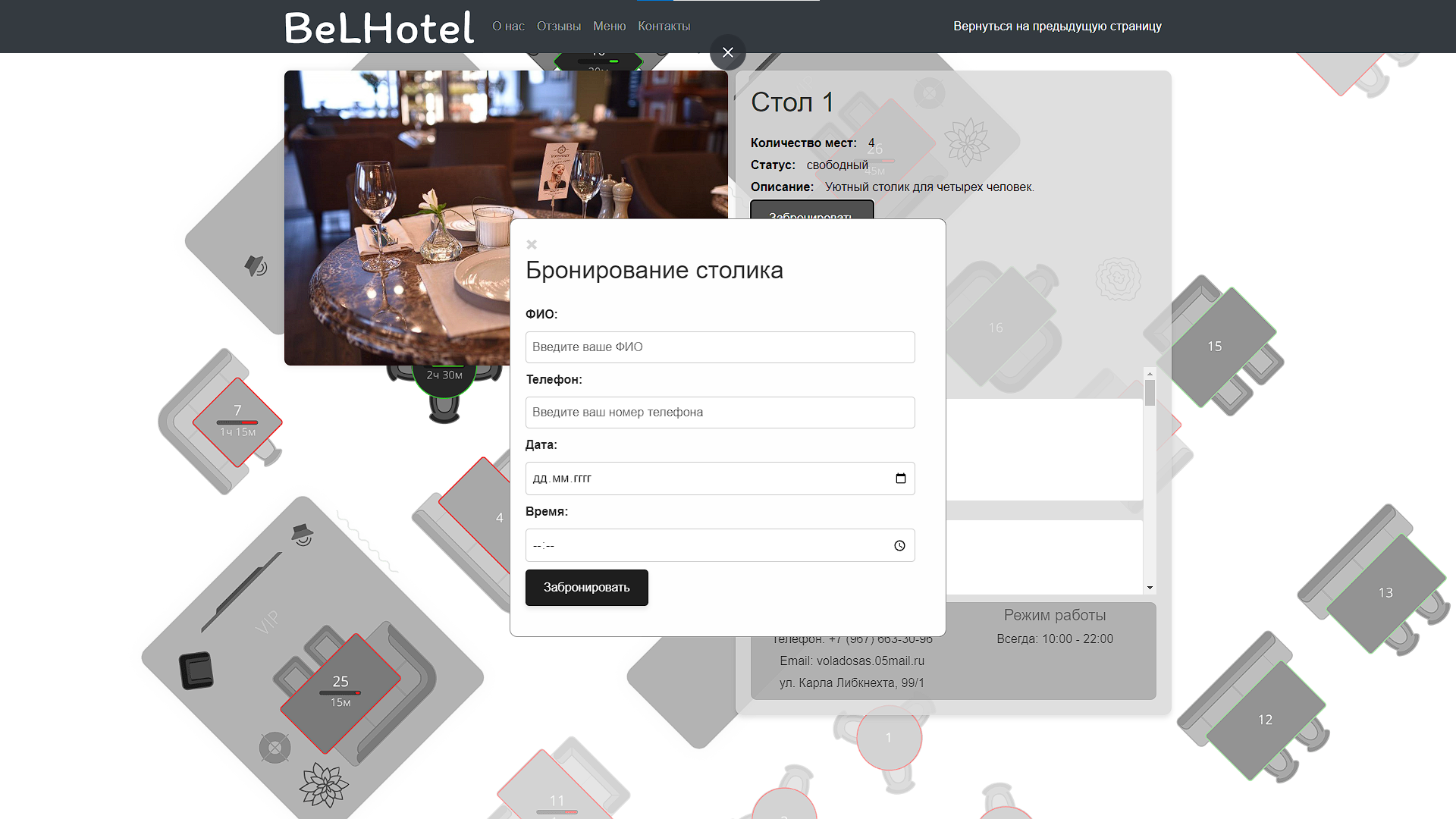


Рисунок 2. – окно забронировать стол.

На рисунке 3 – представлена страница просмотра столов.

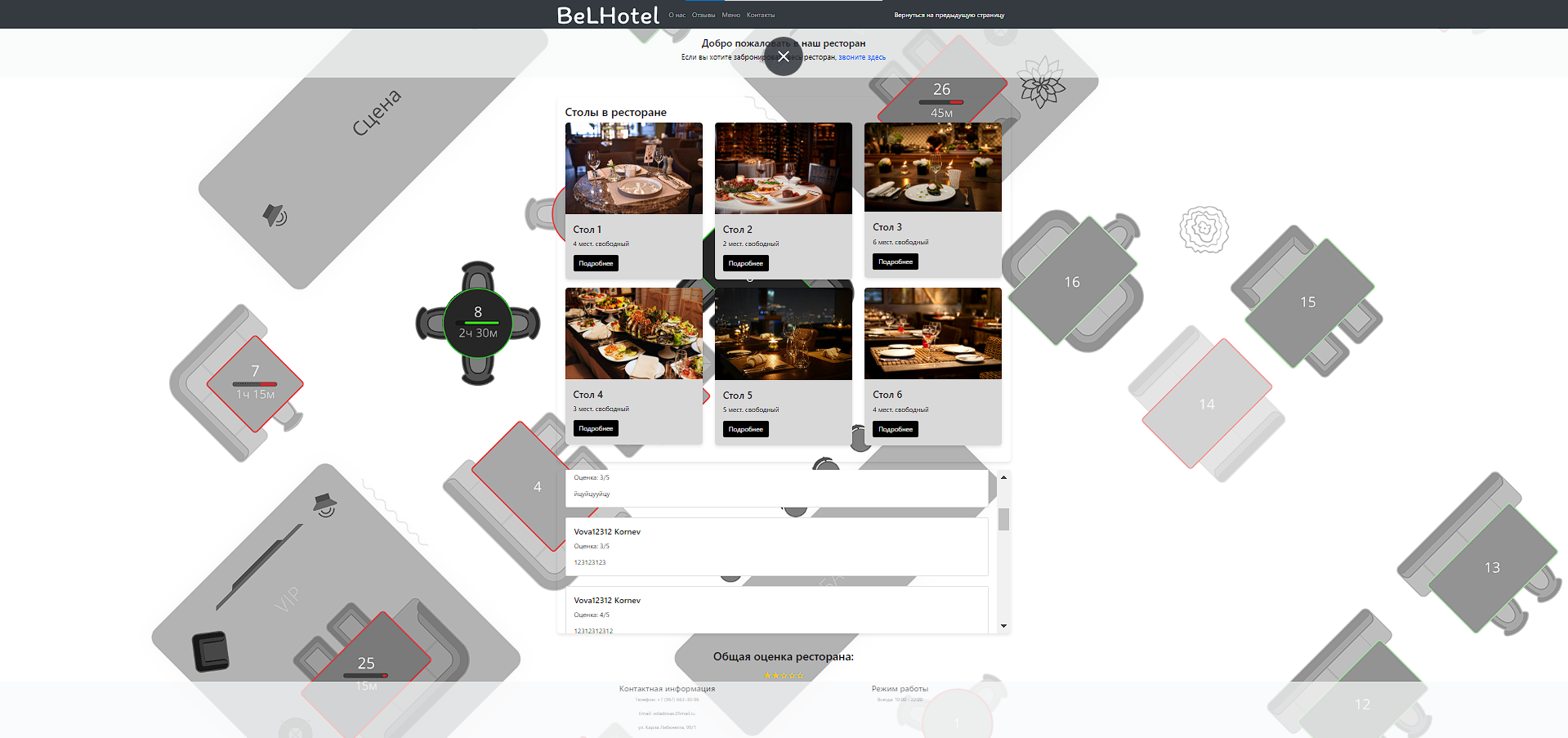


Рисунок 3. – окно столов.

На рисунке 4 – представлена страница создание отзыва.

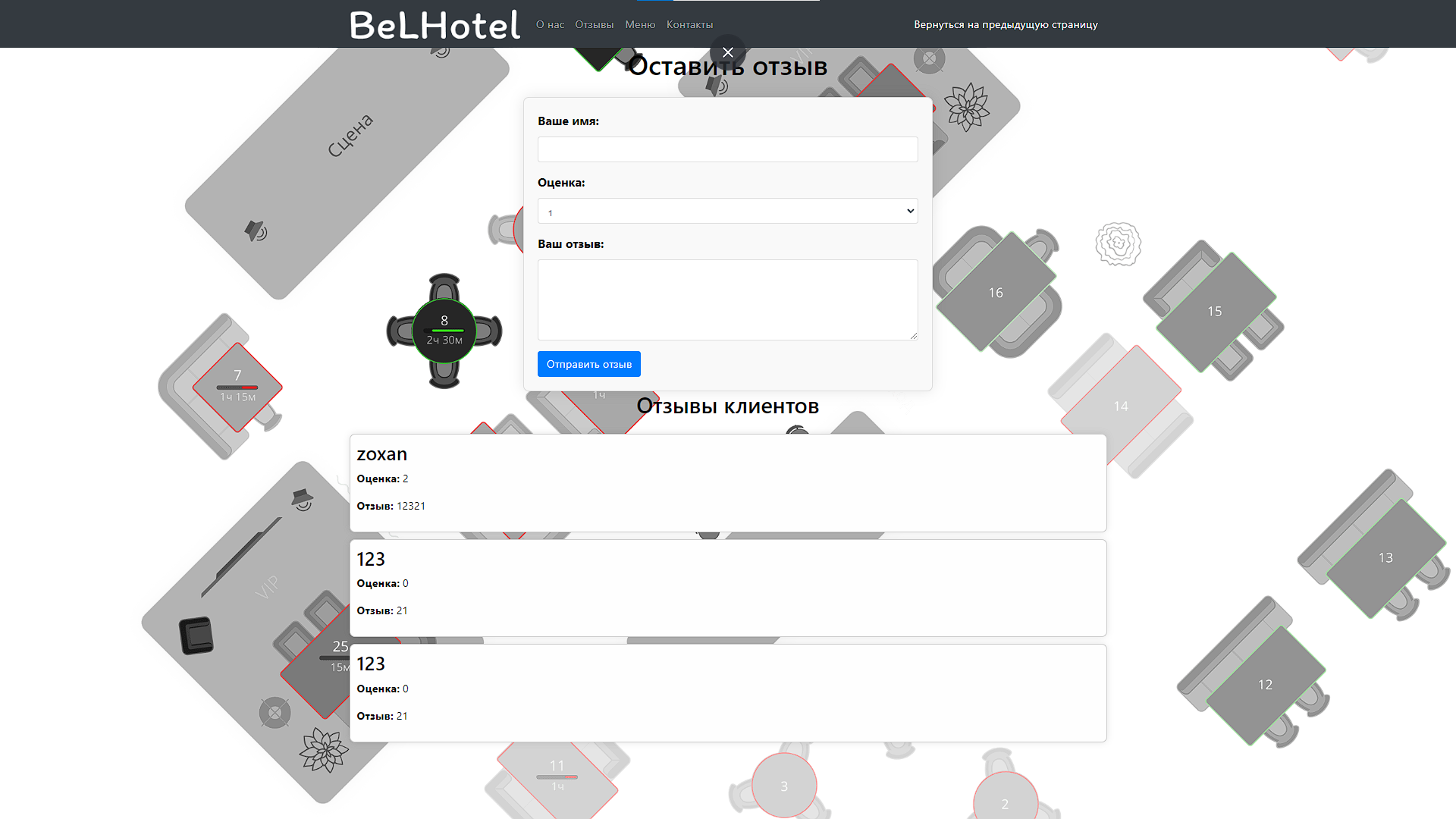
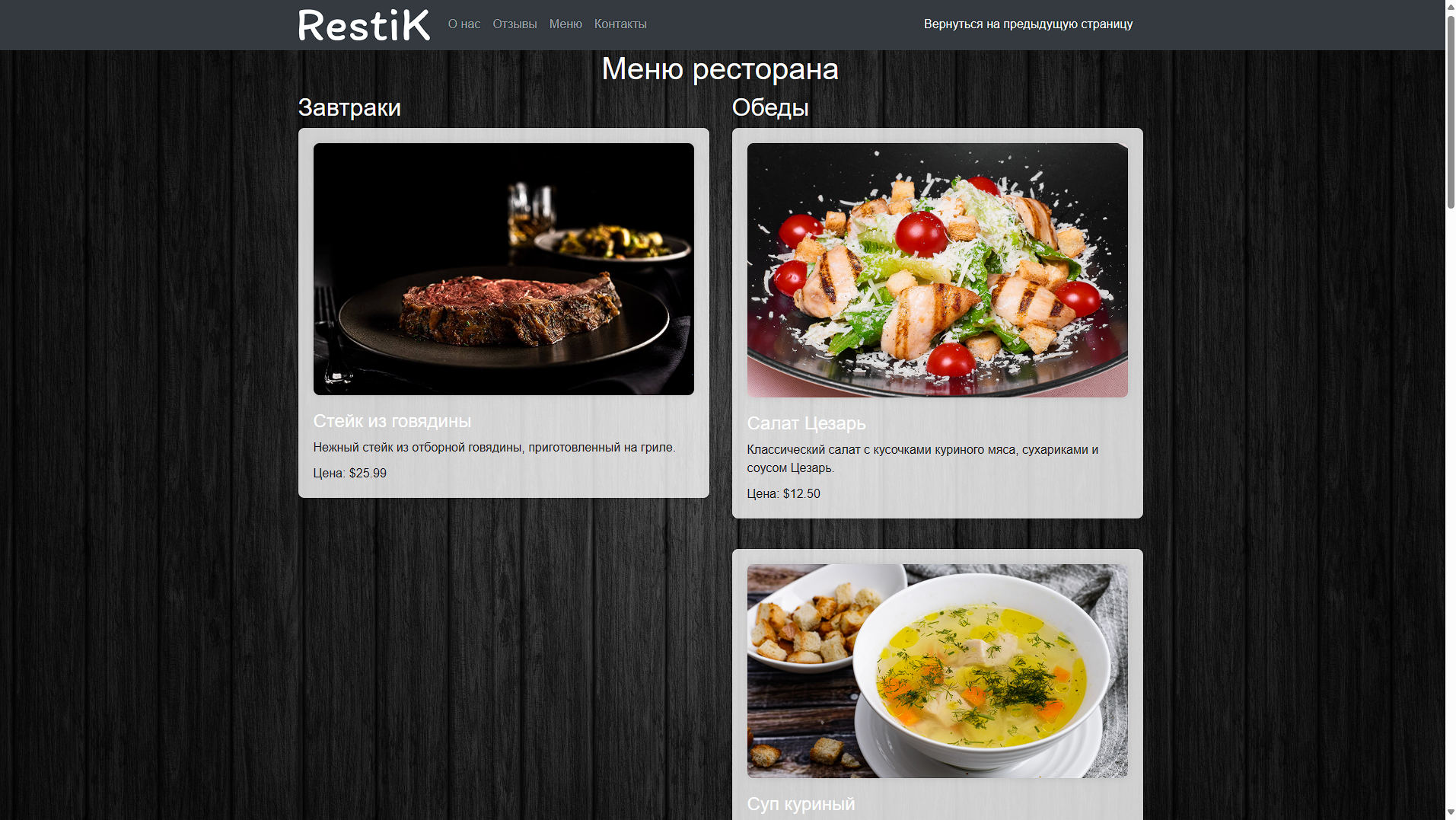


Рисунок 4 – страница отзыва.

На рисунке 5 – представлена страница меню.



На рисунке 5 – страница меню.

# 2. ПРОЕКТНАЯ ЧАСТЬ

## 2.1 Техническое задание

Административная часть сайта должна предоставлять возможность добавления, редактирования и удаления данных из базы данных

Веб-сайт должен соответствовать требованиям:

* единый стиль оформления;
* интуитивно понятное назначение элементов интерфейса;
* отображение на экране только тех возможностей, которые доступны конкретному пользователю;
* отображение на экране только необходимой для решения текущей прикладной задачи информации;

HTML— это язык гипертекстовой разметки текста. Он нужен, чтобы размещать на веб-странице элементы: текст, картинки, таблицы и видео.

Для реализации интерактивных элементов сервероной части должны использоваться языки php, javascript.

Объем flash-заставки не должен превышать 300 Kb.

CSS — язык описания внешнего вида HTML-документа. Это одна из базовых технологий в современном интернете. Практически ни один сайт не обходится без CSS, поэтому HTML и CSS действуют в единой связке.

Каскадные таблицы стилей работают с HTML, но это совершенно другой язык. HTML структурирует документ и упорядочивает информацию, а CSS взаимодействует с браузером, чтобы придать документу оформление.

## 2.2 Программное обеспечение информационной системы

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире. Данная СУБД подходит для самых различных проектов: от небольших приложений до больших высоконагруженных проектов.

Необходимое программное обеспечение для персонального компьютера:

- Операционная система: Windows/Linux/macOS

Сайт должен быть доступен для полнофункционального просмотра с помощью следующих браузеров:

– Opera 6.0 и выше;

– Google;

– Yandex.

JavaScript - предназначен для написания сценариев для активных HTML-страниц. Язык JavaScript не имеет никакого отношения к языку Java. Java разработан фирмой SUN. JavaScript - фирмой Netscape Communication Corporation. Первоначальное название - LiveScript. После завоевания языком Java всемирной известности LiveScript из коммерческих соображений переименовали в JavaScript.

2.3 Руководство пользователя

Авторизация и Регистрация: не необходима для это сайта, для за бронирования стола, все посетители сайта могут забронировать стол и созвониться с ресепшеном, вы спросите, а как мне связаться с ресепшеном?

Пользователю просто нужно будет перейти с главной страницы на стол по его вкусу и там он уже увидит забронирован стол или еще нет, если еще нет, то просто нажимаем на кнопку забронировать вводим данные и свой номер телефона и вам перезвонят через некоторое время или.

Вы же можете просто зайти на странице контактов, где будет написан номер ресепшена и все обсудить уже по телефонному разговору.

На рисунке 6 представлено окно бронирования стола

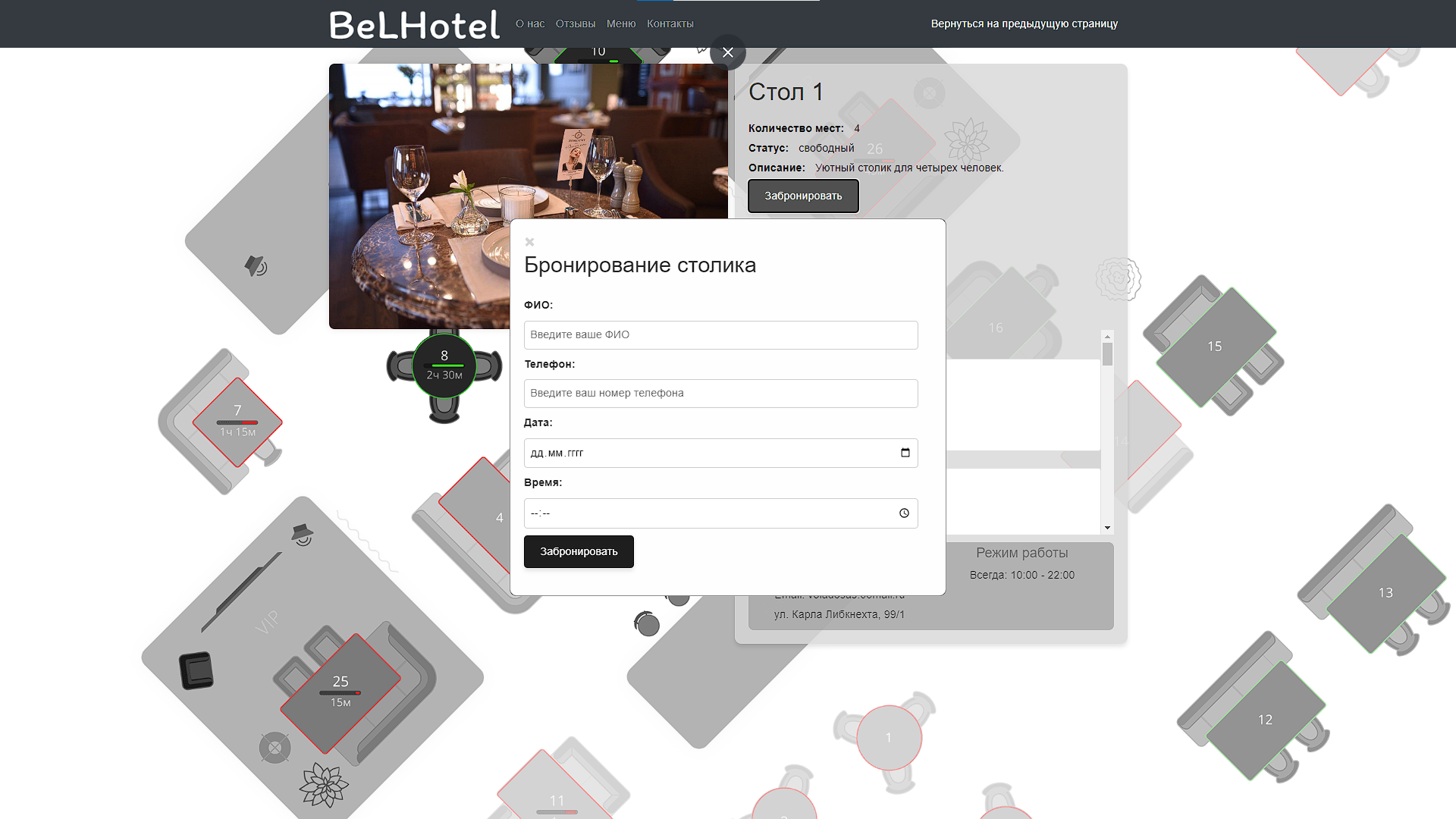


Рисунок 6. – окно бронирования.

После перехода на сайт пользователь попадет на главную страницу, где он сможет смотреть информацию о меню и столах, заходить на страницу стола, бронировать столы и переходить на страницу бронирования стола.

На рисунке 7 представлена главная страница.

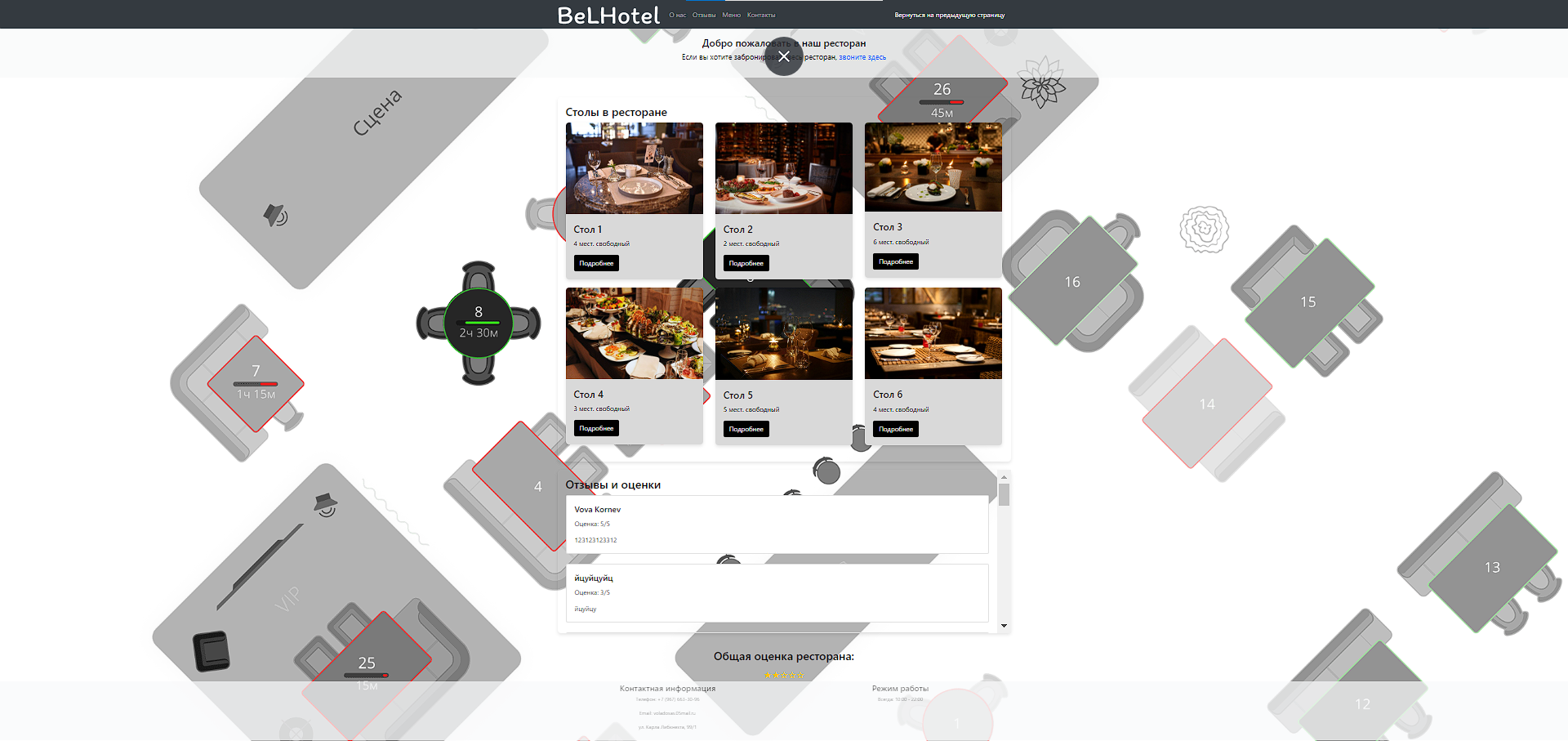


Рисунок 7. – Главная страница.

2.4 Руководство администратора

Авторизация:

Чтобы зайти как за администратора вы переходите посылке http://voladosas0.temp.swtest.ru/admin\_panel.php

Роль администратора определена в базе данных как «admin».

Доступ к Админ-Панели:

После успешного перехода на страницу под Администратором у вас появляется доступ к “Админ-панели”, там выводится вся информация об отзыва, столах и

На рисунке 8 представлена страница, показывающая все столы, которые есть в базе данных



Рисунок 8. – Страница админ панели.

Обзор Отзывов:

На рисунке 9 представлена страница админ-панели, на которой отображается таблица со всеми отзывами, включая их данные.

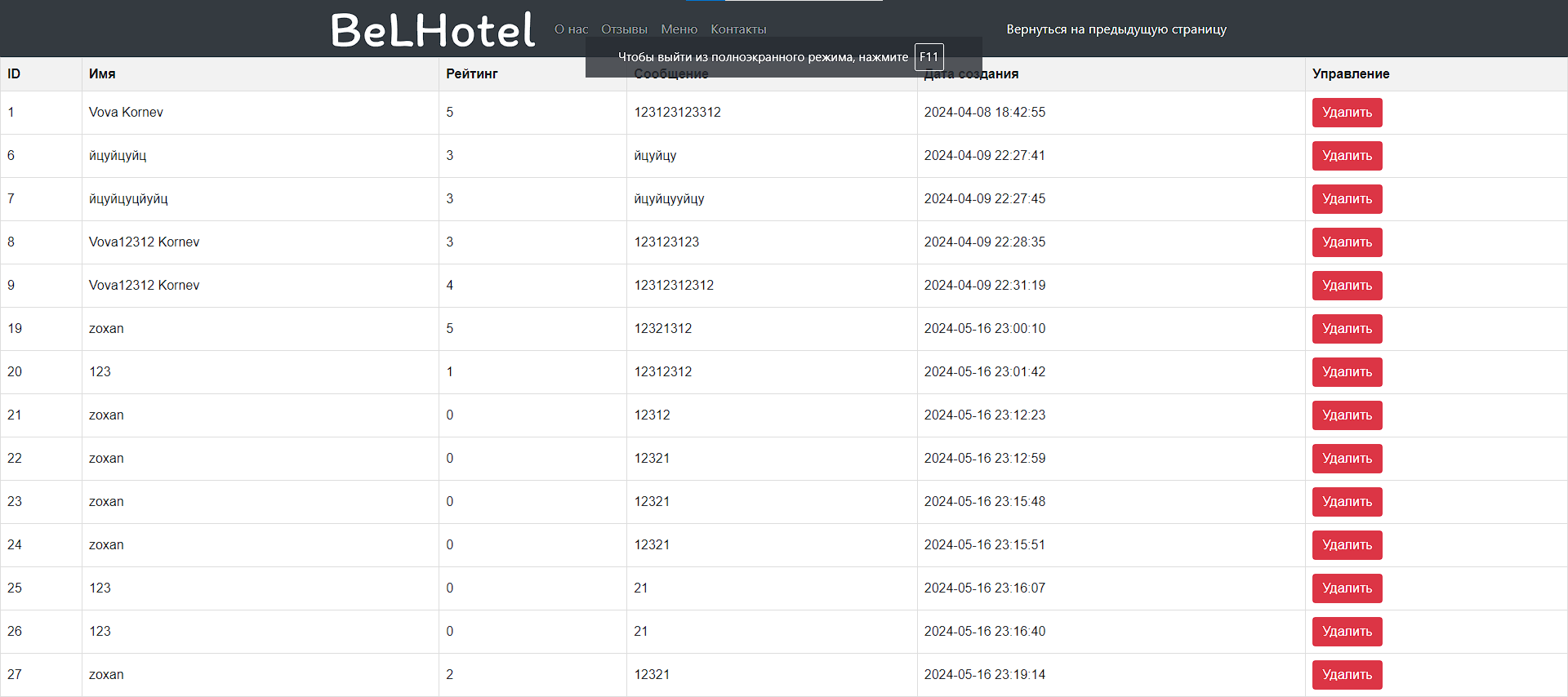


Рисунок 9. – Таблица отзывов.

Каждая строка содержит кнопку «Удалить»,

На рисунке 10 представлена представлена страница админ-панели, на которой отображается добавления стола, включая его данные.

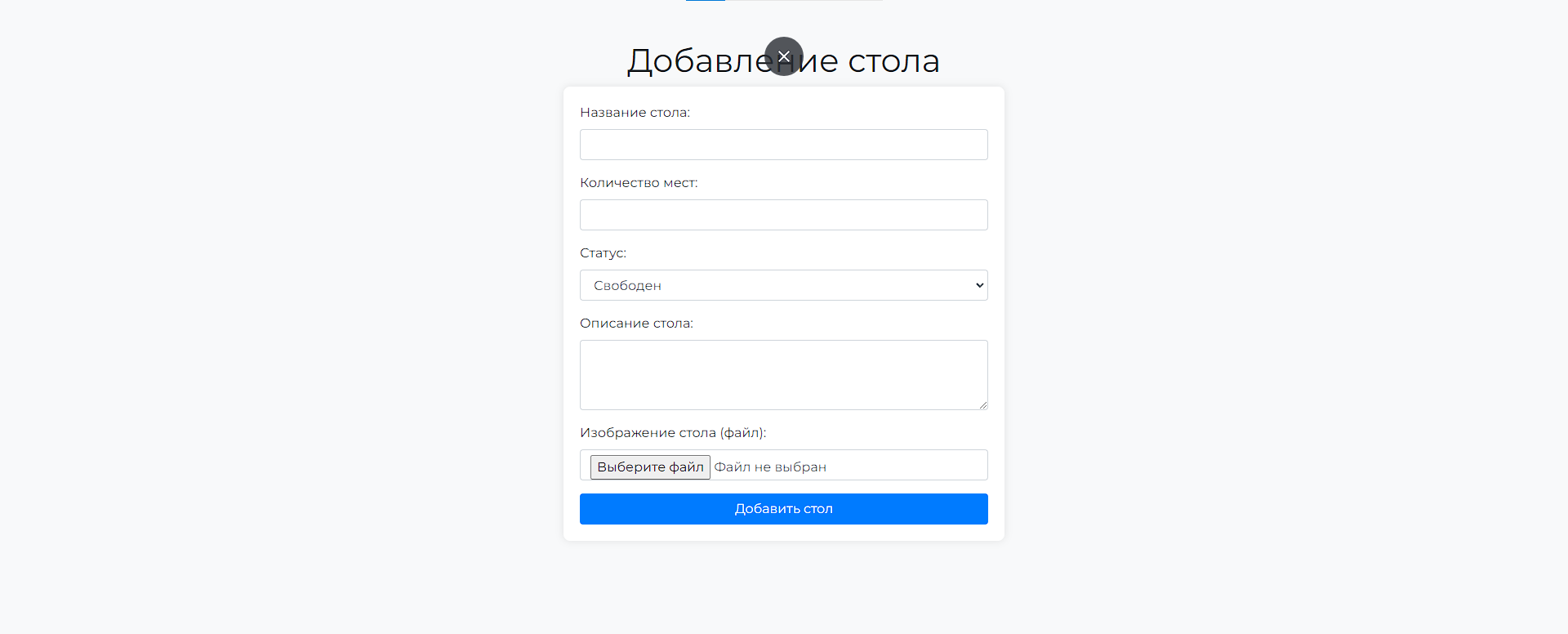


Рисунок 10. – Добавление стола.

# ЗАКЛЮЧЕНИЕ

При разработке приложения были достигнуты цели преддипломной практики, а именно: разработан удобный интерфейс для сайта по выбору бронирования стола в ресторане, который включает в себя все необходимые элементы для успешного функционирования онлайн-ресторана. Проект был разработан с учетом современных тенденций в дизайне, а также с учетом потребностей всех аудитории - людей, которые ищут качественный и стильный столик в ресторане.

В процессе разработки сайта были использованы современные технологии и инструменты, такие как HTML, CSS, JavaScript. Это позволило создать функциональный и удобный сайт, который будет привлекать клиентов и способствовать росту посещения людей.

В целом, разработка сайта бронирования столов ресторане была успешной и позволила получить ценный опыт в области веб-дизайна и разработки. Этот опыт будет полезен в дальнейшей карьере и поможет успешно решать различные задачи в области IT.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Элизабет Робсон. JavaScript, Веб-дизайн, Веб-программирование, Зарубежная компьютерная литература, Интернет Программирование, Создание сайтов. - 200c.
2. Гаевский, А.Ю., Романовский, В.А. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript. - М.: Триумф, 2020. - 464 c.
3. Дакетт, Джон. HTML и CSS. Разработка и дизайн веб-сайтов. - Москва: Наука, 2019. - 480 c.
4. Дронов, В. JavaScript в Web-дизайне. - М.: СПб: БХВ, 2019. - 880 c.
5. Дронов, В. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов. - М.: БХВ-Петербург, 2019. - 480 c.
6. Дронов, В.А. PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов. - М.: БХВ-Петербург, 2019. - 820 c.
7. Дронов, Владимир. JavaScript и AJAX в Web-дизайне. - Москва: Высшая школа, 2019. - 736 c.
8. Дронов, Владимир. PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных Web-сайтов. - М.: "БХВ-Петербург", 2019. - 448 c.
9. Колисниченко, Д. PHP 5/6 и MySQL 6. Разработка Web-приложений. - М.: БХВ-Петербург, 2021. - 560 c.
10. Колисниченко, Д.Н. PHP 5/6 и MySQL 6. Разработка Web-приложений. - М.: БХВ-Петербург, 2021. - 364 c.
11. Флэнаган, Дэвид. JavaScript: Подробное руководство. - М.: Вильямс, 2018. - 1088 c.
12. Никсон, Робин. PHP, MySQL, JavaScript и CSS. Создание динамических веб-сайтов. - М.: ДМК Пресс, 2020. - 816 c.
13. Уэйс, Бен. Полное руководство по JavaScript. - СПб: Питер, 2021. - 944 c.
14. МакФарланд, Дэвид Сойер. JavaScript и jQuery. Подробное руководство. - М.: Вильямс, 2019. - 672 c.
15. Ульман, Ларри. PHP и MySQL. Профессиональное программирование. - М.: Символ-Плюс, 2021. - 800 c.

# ПРИЛОЖЕНИЕ

Приложение A

index.php

<?php include \_\_DIR\_\_ . "/header/header.php"; ?>

<?php

$servername = "localhost";

$username = "zoxan";

$password = "123";

$dbname = "restoris";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$sql = "SELECT \* FROM tables";

$result = $conn->query($sql);

$sql\_reviews = "SELECT \* FROM reviews";

$result\_reviews = $conn->query($sql\_reviews);

$sql\_average\_rating = "SELECT AVG(rating) AS avg\_rating FROM reviews";

$result\_average\_rating = $conn->query($sql\_average\_rating);

$average\_rating = 0;

if ($result\_average\_rating->num\_rows > 0) {

$row = $result\_average\_rating->fetch\_assoc();

$average\_rating = $row["avg\_rating"];

}

?>

<style>

.restaurant-container {

border-radius: 8px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

padding: 20px;

margin-bottom: 20px;

}

.reviews-container {

border-radius: 8px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

padding: 20px;

margin-bottom: 20px;

max-height: 400px;

overflow-y: auto;

}

.restaurant-card {

background-color: #d9d9d9;

border-radius: 8px;

overflow: hidden;

margin-bottom: 20px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

transition: all 0.3s ease-in-out;

}

.restaurant-card:hover {

box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);

}

.restaurant-card img {

width: 100%;

height: auto;

border-bottom: 1px solid #ddd;

}

.restaurant-card-body {

padding: 20px;

}

h5 {

color: black;

}

.restaurant-card-title {

font-size: 24px;

margin-bottom: 10px;

}

.restaurant-card-description {

text-color: black;

font-size: 16px;

color: #000000;

}

body {

background-image: url('uploads/shema-zala-restorana.png');

background-size: cover;

background-color: #ffffff;

background-position: center;

background-repeat: no-repeat;

color: #000000;

}

.btn-primary {

background-color: #000000;

border-color: #000000;

color: #ffffff;

}

.btn-primary:hover {

background-color: #595959;

border-color: #ffffff;

}

.welcome-banner {

background-color: rgba(248, 249, 250, 0.7);

padding: 20px 0;

text-align: center;

}

.welcome-text {

color: #000;

font-size: 24px;

}

.overall-rating {

margin-top: 20px;

text-align: center;

}

.overall-rating span {

font-size: 24px;

color: #ffc107;

}

</style>

<body>

<div class="welcome-banner">

<h2 class="welcome-text">Добро пожаловать в наш ресторан</h2>

</div>

<div class="container mt-5">

<div class="row">

<div class="col-md-12">

<div class="restaurant-container">

<h3>Столы в ресторане</h3>

<div class="row">

<?php

if ($result->num\_rows > 0) {

while($row = $result->fetch\_assoc()) {

?>

<div class="col-md-4">

<div class="restaurant-card">

<img src="../<?php echo $row["img"]; ?>" alt="<?php echo $row["name\_table"]; ?>">

<div class="restaurant-card-body">

<h5 class="restaurant-card-title"><?php echo $row["name\_table"]; ?></h5>

<p class="restaurant-card-description"><?php echo $row["quantity"]; ?> мест. <?php echo $row["status"]; ?></p>

<a href="public/table\_details.php?id=<?php echo $row['id']; ?>" class="btn btn-primary">Подробнее</a>

</div>

</div>

</div>

<?php

}

} else {

echo "0 результатов";

}

?>

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12">

<div class="reviews-container">

<h3>Отзывы и оценки</h3>

<?php

if ($result\_reviews->num\_rows > 0) {

while($row = $result\_reviews->fetch\_assoc()) {

?>

<div class="card">

<div class="card-body">

<h5 class="card-title"><?php echo $row["name"]; ?></h5>

<p class="card-text">Оценка: <?php echo $row["rating"]; ?>/5</p>

<p class="card-text"><?php echo $row["message"]; ?></p>

</div>

</div>

<br>

<?php

}

} else {

echo "Пока нет отзывов.";

}

?>

<!-- Кнопка "Оставить отзыв" -->

<a href="public/reviews.php" class="btn btn-primary">Оставить отзыв</a>

</div>

</div>

</div>

</div>

<div class="overall-rating">

<h3>Общая оценка ресторана:</h3>

<?php

// Округляем среднюю оценку ресторана до ближайшего целого числа

$rounded\_average\_rating = round($average\_rating);

// Выводим звездочки в зависимости от округленной средней оценки

for ($i = 1; $i <= 5; $i++) {

if ($i <= $rounded\_average\_rating) {

echo "<span>&#9733;</span>"; // Звезда

} else {

echo "<span>&#9734;</span>"; // Пустая звезда

}

}

?>

</div>

<?php include \_\_DIR\_\_ . "/footer/footer.php"; ?>

</body>

Приложение B

Header.php

<?php const test = "C:/xampp/htdocs/diplom/";?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Бронирование столов в ресторане</title>

<!-- Подключение стилей Bootstrap -->

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

<!-- Подключение шрифта Montserrat из Google Fonts -->

<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap" rel="stylesheet">

<style>

/\* Основные стили оставляем без изменений \*/

</style>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container">

<!-- Ваш логотип -->

<a class="navbar-brand" href="../index.php"><img src="<?php echo "uploads/RestiK.png"?> " alt="Рестик" class="logo"></a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav mr-auto">

<li class="nav-item">

<a class="nav-link" href=" /diplom/public/Onas.php">О нас</a>

</li>

<li class="nav-item">

<a class="nav-link" href="/diplom/public/reviews.php">Отзывы</a>

</li>

<li class="nav-item">

<a class="nav-link" href="/diplom/public/menu.php">Меню</a>

</li>

<li class="nav-item">

<a class="nav-link" href="/diplom/public/contacts.php">Контакты</a>

</li>

</ul>

<button class="btn btn-dark" onclick="goBack()">Вернуться на предыдущую страницу</button>

</div>

</div>

</nav>

<!-- Подключение скриптов Bootstrap -->

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<!-- JavaScript для перехода на предыдущую страницу -->

<script>

function goBack() {

window.history.back();

}

</script>

</body>

</html>

Приложение C

Footer.php

<style>

/\* Стили для футера \*/

footer {

position: relative;

bottom: 0; /\* Размещение внизу страницы \*/

width: 100%; /\* Ширина 100% \*/

background-color: #f8f9fa; /\* Цвет фона \*/

padding: 5px 0; /\* Уменьшенные внутренние отступы \*/

text-align: center; /\* Выравнивание текста по центру \*/

opacity: 0.7; /\* Прозрачность \*/

font-size: 12px; /\* Уменьшенный размер шрифта \*/

}

/\* Стили для текста внутри футера \*/

footer p, footer h5 {

color: rgba(0, 0, 0, 0.7); /\* Цвет текста с полупрозрачностью \*/

}

</style>

<footer>

<div class="container">

<div class="row">

<div class="col-md-6">

<h5>Контактная информация</h5>

<p>Телефон: +7 (967) 663-30-96</p>

<p>Email: voladosas.05mail.ru</p>

<p>ул. Карла Либкнехта, 99/1</p>

</div>

<div class="col-md-6">

<h5>Режим работы</h5>

<p>Всегда: 10:00 - 22:00</p>

</div>

</div>

</div>

</footer>

Приложение D

View\_table.php

<?php include "../header/header.php"; ?>

<style>

table {

font-family: Arial, sans-serif;

border-collapse: collapse;

width: 100%;

background-color: #ffffff;

}

th, td {

border: 1px solid #dddddd;

text-align: left;

padding: 10px;

}

th {

background-color: #f2f2f2;

}

.table-img {

max-width: 100px;

height: auto;

transition: transform 0.3s ease-in-out; /\* Добавляем анимацию для изображений \*/

}

.table-img:hover {

transform: scale(1.1); /\* Увеличиваем изображение при наведении \*/

}

table tr:nth-child(even) {

background-color: #f2f2f2; /\* Цвет фона для четных строк \*/

}

table tr:hover {

background-color: #f8f8f8; /\* Цвет фона при наведении на строку \*/

}

.btn {

background-color: #007bff;

color: #fff;

border: none;

padding: 8px 16px;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s ease;

margin-right: 5px;

}

.btn-delete {

background-color: #dc3545;

}

.btn-delete:hover {

background-color: #c82333; /\* Изменяем цвет при наведении \*/

}

.btn-add {

background-color: #28a745;

}

.btn-add:hover {

background-color: #218838; /\* Изменяем цвет при наведении \*/

}

.btn:hover {

background-color: #0056b3;

}

</style>

<body>

<table>

<tr>

<th>ID</th>

<th>Название стола</th>

<th>Количество мест</th>

<th>Статус</th>

<th>Изображение</th>

<th>Описание</th>

<th>Управление</th>

</tr>

<?php

// Подключение к базе данных

$servername = "localhost";

$username = "zoxan";

$password = "123";

$dbname = "restoris";

// Создание подключения

$conn = new mysqli($servername, $username, $password, $dbname);

// Проверка подключения

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Удаление записи, если передан параметр delete\_id через GET запрос

if (isset($\_GET['delete\_id'])) {

$delete\_id = $\_GET['delete\_id'];

$sql\_delete = "DELETE FROM tables WHERE id = $delete\_id";

if ($conn->query($sql\_delete) === TRUE) {

echo "Record deleted successfully";

} else {

echo "Error deleting record: " . $conn->error;

}

}

// SQL запрос для выбора всех данных из таблицы

$sql = "SELECT id, name\_table, quantity, status, img, description FROM tables";

$result = $conn->query($sql);

// Проверка наличия данных

if ($result->num\_rows > 0) {

// Вывод данных каждой строки

while($row = $result->fetch\_assoc()) {

echo "<tr>";

echo "<td>" . $row["id"]. "</td>";

echo "<td>" . $row["name\_table"]. "</td>";

echo "<td>" . $row["quantity"]. "</td>";

echo "<td>" . $row["status"]. "</td>";

echo "<td><img src='../" . $row["img"]. "' alt='" . $row["name\_table"]. "' class='table-img'></td>";

echo "<td>" . $row["description"]. "</td>";

echo "<td><a href='?delete\_id=" . $row["id"] . "' class='btn btn-delete'>Удалить</a> </td>";

echo "</tr>";

}

} else {

echo "<tr><td colspan='7'>0 results</td></tr>";

}

// Закрытие соединения с базой данных

$conn->close();

?>

</table>

</body>

Приложение E

View\_reviews.php

<?php include "../header/header.php"; ?>

<style>

table {

font-family: Arial, sans-serif;

border-collapse: collapse;

width: 100%;

}

th, td {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

th {

background-color: #f2f2f2;

}

.btn-delete {

background-color: #dc3545;

color: #fff;

border: none;

padding: 6px 12px;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s ease;

}

.btn-delete:hover {

background-color: #c82333;

}

</style>

<body>

<table>

<tr>

<th>ID</th>

<th>Имя</th>

<th>Рейтинг</th>

<th>Сообщение</th>

<th>Дата создания</th>

<th>Управление</th>

</tr>

<?php

// Подключение к базе данных

$servername = "localhost";

$username = "zoxan";

$password = "123";

$dbname = "restoris";

// Создание подключения

$conn = new mysqli($servername, $username, $password, $dbname);

// Проверка подключения

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Проверяем, был ли отправлен запрос на удаление

if(isset($\_POST['delete\_id'])) {

$delete\_id = $\_POST['delete\_id'];

// SQL запрос для удаления записи с указанным ID из таблицы reviews

$delete\_sql = "DELETE FROM reviews WHERE id = $delete\_id";

// Выполняем запрос

if ($conn->query($delete\_sql) === TRUE) {

echo "<p>Отзыв успешно удален.</p>";

} else {

echo "Ошибка при удалении отзыва: " . $conn->error;

}

}

// SQL запрос для выбора всех данных из таблицы reviews

$sql = "SELECT id, name, rating, message, created\_at FROM reviews";

$result = $conn->query($sql);

// Проверка наличия данных

if ($result->num\_rows > 0) {

// Вывод данных каждой строки

while($row = $result->fetch\_assoc()) {

echo "<tr>";

echo "<td>" . $row["id"]. "</td>";

echo "<td>" . $row["name"]. "</td>";

echo "<td>" . $row["rating"]. "</td>";

echo "<td>" . $row["message"]. "</td>";

echo "<td>" . $row["created\_at"]. "</td>";

// Добавляем кнопку удаления с формой для отправки ID записи

echo "<td><form method='post'><input type='hidden' name='delete\_id' value='" . $row["id"] . "'><button type='submit' class='btn-delete'>Удалить</button></form></td>";

echo "</tr>";

}

} else {

echo "<tr><td colspan='6'>0 results</td></tr>";

}

// Закрытие соединения с базой данных

$conn->close();

?>

</table>

</body>

Приложение F

View\_menu.php

<?php include "../header/header.php"; ?>

<style>

table {

font-family: Arial, sans-serif;

border-collapse: collapse;

width: 100%;

}

th, td {

border: 1px solid #dddddd;

text-align: left;

padding: 8px;

}

th {

background-color: #f2f2f2;

}

.table-img {

max-width: 100px;

height: auto;

}

.btn-delete {

background-color: #dc3545;

color: #fff;

border: none;

padding: 6px 12px;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s ease;

}

.btn-delete:hover {

background-color: #c82333;

}

</style>

<body>

<table>

<tr>

<th>ID</th>

<th>Название</th>

<th>Описание</th>

<th>Цена</th>

<th>Изображение</th>

<th>Управление</th>

</tr>

<?php

// Подключение к базе данных

$servername = "localhost";

$username = "zoxan";

$password = "123";

$dbname = "restoris";

// Создание подключения

$conn = new mysqli($servername, $username, $password, $dbname);

// Проверка подключения

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Проверяем, был ли отправлен запрос на удаление

if(isset($\_POST['delete\_id'])) {

$delete\_id = $\_POST['delete\_id'];

// SQL запрос для удаления записи с указанным ID из таблицы menu\_dishes

$delete\_sql = "DELETE FROM menu\_dishes WHERE id = $delete\_id";

// Выполняем запрос

if ($conn->query($delete\_sql) === TRUE) {

echo "<p>Запись успешно удалена.</p>";

} else {

echo "Ошибка при удалении записи: " . $conn->error;

}

}

// SQL запрос для выбора всех данных из таблицы menu\_dishes

$sql = "SELECT id, name, description, price, img FROM menu\_dishes";

$result = $conn->query($sql);

// Проверка наличия данных

if ($result->num\_rows > 0) {

// Вывод данных каждой строки

while($row = $result->fetch\_assoc()) {

echo "<tr>";

echo "<td>" . $row["id"]. "</td>";

echo "<td>" . $row["name"]. "</td>";

echo "<td>" . $row["description"]. "</td>";

echo "<td>" . $row["price"]. "</td>";

echo "<td><img src='../" . $row["img"]. "' alt='" . $row["name"]. "' class='table-img'></td>";

// Добавляем кнопку удаления с формой для отправки ID записи

echo "<td><form method='post'><input type='hidden' name='delete\_id' value='" . $row["id"] . "'><button type='submit' class='btn-delete'>Удалить</button></form></td>";

echo "</tr>";

}

} else {

echo "<tr><td colspan='6'>0 results</td></tr>";

} $conn->close();

?>

</table>

</body>