ЧАСТНОЕ УЧРЕЖДЕНИЕ ПРОФЕССИОНАЛЬНАЯ

ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ

ТЕХНИКУМ «БИЗНЕС И ПРАВО»

|  |  |
| --- | --- |
| РЕКОМЕНДУЕТСЯ К ЗАЩИТЕ  Председатель цикловой комиссии технических дисциплин | РАБОТА ЗАЩИЩЕНА С ОЦЕНКОЙ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.А.Мартыненко | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись ответственного секретаря) |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г | «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |

ДИПЛОМНЫЙ ПРОЕКТ

Система бронирование столов в ресторане

|  |  |  |  |
| --- | --- | --- | --- |
| Специальность: | |  | |
| Выполнил (а): | Корнев В.В | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
|  |  | |  |
| Руководитель: |  | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

г. Белореченск

2024г.

Содержание

|  |  |
| --- | --- |
| ВВЕДЕНИЕ | 3 |
| 1 Общие сведения бронирование столов ресторана | 5 |
| 1.1 Анализ предметной области | 5 |
| 1.2 История развития автоматизированных информационных систем в сфере бронирования столов в ресторане | 5 |
| 1.3 Внедрение автоматизированных информационных систем в сферу бронирования столов в ресторанах | 7 |
| 1.4 Преимущества и недостатки автоматизированных информационных систем в сфере бронирования столов в ресторанах Традиционные алгоритмические языки | 8 |
| 1.5 Традиционные алгоритмические языки | 10 |
| 1.6 Инструментальные средства общего назначения | 15 |
| 2.Разработка автоматизированной информационной системы бронирования столов в ресторане | 21 |
| 2.1 Техническое задание | 21 |
| 2.2 Создание базы данных и подключение к проекту | 48 |
| 2.3 Руководство пользователя | 51 |
| 2.4 Руководство администратора | 54 |
| ЗАКЛЮЧЕНИЕ | 55 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 56 |
| ПРИЛОЖЕНИЕ | 58 |

Введение

В современном информационном обществе вопросы автоматизации и цифровизации бизнес-процессов становятся все более актуальными. Это касается и ресторанного бизнеса, где автоматизация бронирования столов играет ключевую роль в улучшении обслуживания клиентов и оптимизации работы ресторана. С развитием цифровых технологий и интернета бронирование столов через онлайн-платформы становится стандартом для многих заведений.

Проект по разработке автоматизированной системы бронирования столов для ресторана актуален в современных условиях. Он направлен на повышение эффективности управления, улучшение пользовательского опыта и снижение издержек. Автоматизация процесса бронирования позволяет не только обеспечить комфорт клиентам, но и оптимизировать работу персонала, снизить вероятность ошибок и увеличить общий уровень удовлетворенности посетителей.

Объектом исследования является процесс бронирования столов в ресторане. Предметом исследования является разработка автоматизированной системы бронирования столов, включающей клиентскую и серверную части, реализованные с использованием PHP и веб-технологий.

Целью данного проекта является разработка эффективной автоматизированной системы бронирования столов для ресторана, которая обеспечит пользователям возможность легко и быстро бронировать столики через интернет.

Для достижения поставленной цели необходимо решить следующие задачи:

Изучить современные подходы к автоматизации процесса бронирования столов.

Разработать веб-приложение на основе анализа аналогичных решений.

Провести тестирование разработанных компонентов.

Оценить эффективность системы на основе результатов тестирования.

Платформа и технологии

Разрабатываемая система будет представлять собой веб-приложение для ресторана "Lagoona". Приложение должно включать в себя интерфейс для бронирования столов, систему управления бронированиями для администратора и механизм уведомлений для пользователей.

Для разработки клиентской части будут использованы HTML (язык разметки), CSS (каскадные таблицы стилей), JavaScript (язык программирования) и библиотека Bootstrap для обеспечения адаптивного дизайна. Серверная часть будет реализована на языке PHP, обеспечивающем взаимодействие с базой данных.

Анализ предметной области: Исследование текущих решений и тенденций в автоматизации бронирования столов.

Разработка архитектуры системы: Построение логической модели системы, включая диаграммы потоков данных и структур.

Реализация клиентской части: Создание пользовательского интерфейса с использованием HTML, CSS, JavaScript и Bootstrap.

Реализация серверной части: Разработка серверной логики на PHP, включая взаимодействие с базой данных.

Тестирование: Проверка работоспособности системы, исправление ошибок и оптимизация производительности.

Оценка эффективности: Анализ результатов тестирования и сбор отзывов от пользователей для оценки качества и удобства системы.

1. Общие сведения об бронировании столов ресторана.

* 1. Анализ предметной области

Бронирование столов в ресторане - это процесс, который играет ключевую роль в обеспечении комфортного пребывания клиентов и эффективной организации работы заведения. Этот процесс позволяет гостям заранее забронировать место в ресторане на определенное время и дату, что обеспечивает им уверенность в том, что они смогут насладиться своим обедом, ужином или другими услугами в удобное для них время.

Бронирование столов может происходить посредством различных каналов связи. Традиционно клиенты могли бронировать столик, позвонив в ресторан, отправив электронное письмо или даже придя лично. С развитием технологий и появлением интернета в ресторанной отрасли стали активно использоваться онлайн-сервисы бронирования столов. Это позволяет клиентам забронировать столик в несколько кликов, выбрав удобное время и указав количество гостей.

Онлайн-сервисы бронирования столов в ресторане обладают рядом преимуществ. Во-первых, они обеспечивают удобство и доступность для клиентов, позволяя им бронировать столики в любое время суток, вне зависимости от рабочего графика ресторана. Во-вторых, такие сервисы часто предоставляют дополнительную информацию о ресторане, его меню, атмосфере и отзывах клиентов, что помогает принять более обоснованное решение о посещении.

Когда гости решают посетить ресторан, они часто имеют ясное представление о том, какой атмосферы и уровня сервиса они ожидают. Бронирование столика заранее дает возможность клиентам убедиться в наличии места и избежать неприятных сюрпризов при прибытии в ресторан. Кроме того, это позволяет ресторану лучше организовать рабочий процесс и гарантировать оптимальное распределение ресурсов.

Для ресторанов бронирование столов является важным инструментом управления загрузкой и планированием ресурсов. Правильное распределение гостей по времени и местам позволяет ресторану обеспечить оптимальное обслуживание, избегая перегрузок и длительных ожиданий для клиентов. Кроме того, бронирование столов позволяет ресторану оценить спрос на свои услуги и прогнозировать его изменения в будущем.

В целом, бронирование столов в ресторане - это важный элемент организации работы заведения, способствующий удовлетворению потребностей как клиентов, так и персонала. Онлайн-сервисы бронирования столов представляют собой удобный и эффективный способ для всех сторон взаимодействия, обеспечивая комфорт и уверенность в планировании своего визита в ресторан.

* 1. История развития автоматизированных информационных систем в сфере бронирования столов в ресторанах

История развития автоматизированных информационных систем в сфере бронирования столов в ресторанах представляет собой важный аспект, необходимый для понимания текущего состояния и перспектив развития данной области. Начиная с первых попыток создания систем для управления бронированием до современных технологических решений, этот аспект оказывает значительное влияние на организацию и предоставление услуг в ресторанной индустрии.

Этапы истории развития автоматизированных информационных систем в сфере бронирования столов в ресторанах включают:

Первые системы бронирования:

В начале развития ресторанной индустрии бронирование столов осуществлялось вручную через телефонные звонки или личные визиты. Первые системы бронирования представляли собой бумажные журналы и книги, где фиксировались заказы клиентов.

Внедрение компьютерных систем:

С появлением персональных компьютеров рестораны начали использовать простые программные приложения для управления бронированиями. Эти системы позволяли автоматизировать процесс записи и хранения информации о бронированиях, что существенно улучшило точность и удобство управления.

Появление интернет-сервисов:

С развитием интернета стали появляться первые онлайн-сервисы бронирования столов. Эти платформы позволяли клиентам бронировать столики через веб-сайты, что значительно повысило доступность и удобство для пользователей.

Развитие технологий и внедрение PHP:

С развитием веб-технологий и широким распространением языка программирования PHP стало возможным создание мощных и функциональных онлайн-систем бронирования. PHP позволял создавать динамические веб-сайты, которые могли взаимодействовать с базами данных и предоставлять пользователям актуальную информацию о доступности столиков.

Адаптивный дизайн и мобильные приложения:

В последние годы акцент сместился на создание адаптивных веб-сайтов и мобильных приложений, которые позволяют пользователям бронировать столики с любого устройства – мобильного телефона, планшета или компьютера. Адаптивный дизайн обеспечивает удобство использования сайта на различных экранах, что повышает удовлетворенность клиентов.

Интеграция с CRM и ERP системами:

Современные системы бронирования все чаще интегрируются с системами управления взаимоотношениями с клиентами (CRM) и системами планирования ресурсов предприятия (ERP). Это позволяет ресторанам более эффективно управлять своими ресурсами, анализировать поведение клиентов и предоставлять персонализированные услуги.

Использование искусственного интеллекта и машинного обучения:

Новейшие технологии, такие как искусственный интеллект и машинное обучение, начинают внедряться в системы бронирования для прогнозирования спроса, автоматического распределения столиков и персонализации клиентского опыта.

* 1. Внедрение автоматизированных информационных систем в сферу бронирования столов в ресторанах

Внедрение автоматизированных информационных систем в сферу бронирования столов в ресторанах является ключевым этапом в развитии данной отрасли. Этот процесс включает создание и внедрение технологических решений, направленных на оптимизацию управления бронированием, улучшение пользовательского опыта и повышение эффективности бизнес-процессов.

Основные этапы внедрения автоматизированных информационных систем в сферу бронирования столов в ресторанах включают:

Анализ и выбор подходящих технологических решений:

На этом этапе проводится анализ рынка и выбираются подходящие технологические платформы и инструменты для создания и внедрения системы бронирования столов. Важно учитывать специфические требования ресторана, такие как масштабируемость, интеграция с существующими системами управления и возможность адаптации под уникальные бизнес-процессы.

Разработка и тестирование системы:

После выбора технологических решений проводится разработка и тестирование системы. Этот процесс включает создание пользовательского интерфейса, интеграцию с внешними сервисами (например, системами онлайн-платежей или CRM), а также проверку работоспособности системы на предмет обнаружения и устранения ошибок.

Внедрение и настройка системы:

После успешного завершения тестирования система внедряется и настраивается на стороне клиента. Это включает установку на сервер, настройку параметров безопасности, а также обучение персонала работе с новой системой. Важно обеспечить, чтобы все сотрудники понимали функциональность системы и могли эффективно использовать ее в повседневной работе.

Поддержка и сопровождение системы:

После внедрения системы осуществляется ее поддержка и сопровождение. Это включает регулярные обновления для улучшения функциональности, исправление ошибок и добавление новых возможностей в соответствии с изменяющимися потребностями ресторана и ожиданиями клиентов.

Преимущества внедрения автоматизированных информационных систем в ресторанах

Повышение эффективности управления бронированиями:

Автоматизированные системы позволяют значительно упростить процесс бронирования столов, снижая нагрузку на персонал и уменьшая вероятность ошибок. Это обеспечивает более эффективное управление ресурсами и улучшает обслуживание клиентов.

Улучшение пользовательского опыта:

Системы онлайн-бронирования предоставляют клиентам возможность бронировать столики в любое время и с любого устройства. Это делает процесс бронирования более удобным и доступным, что повышает удовлетворенность клиентов.

Интеграция новых технологий:

Современные системы бронирования могут интегрироваться с различными технологиями, такими как мобильные приложения, системы лояльности и CRM. Это позволяет ресторанам лучше понимать потребности своих клиентов и предлагать им персонализированные услуги.

Развитие систем аналитики и отчетности:

Автоматизированные системы бронирования часто включают инструменты для аналитики и отчетности. Эти системы позволяют анализировать данные о бронированиях, предпочтениях клиентов и загруженности ресторана, что помогает принимать обоснованные управленческие решения и улучшать общий опыт клиентов.

Интеграция онлайн-платформ для бронирования столов

С развитием веб-технологий и широким доступом к интернету рестораны активно используют онлайн-платформы для бронирования столов. Эти платформы предоставляют клиентам удобный доступ к бронированию, а также возможность ознакомиться с меню, отзывами и фотографиями.

Интеграция таких платформ с системами управления рестораном позволяет автоматизировать и оптимизировать множество процессов, улучшая координацию работы и повышая общий уровень сервиса. Использование современных технологий, таких как адаптивный дизайн, обеспечивает доступность и удобство использования системы на различных устройствах, от смартфонов до настольных компьютеров.

Внедрение автоматизированных информационных систем в сферу бронирования столов в ресторанах представляет собой стратегически важный шаг, способствующий улучшению эффективности работы ресторанов, повышению удовлетворенности клиентов и созданию конкурентных преимуществ на рынке.

# Преимущества и недостатки автоматизированных информационных систем в сфере бронирования столов в ресторанах\

# некоторые ограничения и недостатки, которые следует учитывать при их разработке и использовании.

# Преимущества:

# Увеличение доступности и удобства:

# Автоматизированные системы позволяют клиентам бронировать столики в любое время и из любой точки мира, что делает процесс бронирования более гибким и удобным. Клиенты могут выбрать желаемую дату и время, а также указать особые предпочтения или требования.

# Повышение эффективности управления:

# Системы автоматизации упрощают управление бронированиями, снижая нагрузку на персонал. Это позволяет улучшить координацию работы и минимизировать ошибки при распределении мест и времени.

# Улучшение пользовательского опыта:

Автоматизированные системы могут предоставлять информацию о доступности столиков, меню, специальных предложениях и отзывах клиентов, что способствует повышению удовлетворенности клиентов. Кроме того,

# Автоматизированные информационные системы для бронирования столов в ресторанах обладают рядом преимуществ, однако они также имеют

# интеграция с системами лояльности может предложить персонализированные предложения.

# Повышение точности и аналитики:

# Такие системы могут собирать данные о бронированиях, предпочтениях клиентов и загруженности ресторана. Это помогает владельцам и менеджерам принимать обоснованные решения и прогнозировать спрос.

# Улучшение коммуникации и взаимодействия:

# Клиенты могут оставлять отзывы и комментарии, а рестораны могут оперативно отвечать на запросы и предоставлять актуальную информацию, что способствует более активному взаимодействию и улучшению сервиса.

# Недостатки:

# Зависимость от интернет-соединения:

# Система бронирования требует стабильного интернет-соединения для работы. Проблемы с сетью могут привести к затруднениям в процессе бронирования и неудобствам для клиентов.

# Высокие первоначальные затраты:

# Внедрение автоматизированных систем требует значительных инвестиций в разработку, установку и настройку программного обеспечения. Это может быть значительным барьером для небольших ресторанов с ограниченными бюджетами.

# Необходимость обучения и поддержки пользователей:

# Внедрение новых систем требует времени и ресурсов на обучение персонала и обеспечение технической поддержки. Это может вызвать временные трудности в работе ресторана.

# Проблемы конфиденциальности и безопасности данных:

# Автоматизированные системы обрабатывают и хранят конфиденциальные данные клиентов, что повышает риск утечек информации и киберугроз. Требуется строгий контроль и обеспечение безопасности информационных систем.

# Необходимость постоянного обновления и совершенствования технологий:

# Быстрое развитие технологий требует постоянного обновления и

# модернизации систем для поддержания их конкурентоспособности и соответствия ожиданиям клиентов. Это может включать интеграцию новых функций и улучшение пользовательского интерфейса.

# Ограничения в доступности для некоторых клиентов:

# Не все клиенты могут быть технически подкованы или иметь доступ к интернету, что может ограничить их возможности воспользоваться системой онлайн-бронирования.

# Возможные технические сбои:

# Автоматизированные системы могут столкнуться с техническими проблемами, такими как сбои в работе серверов или программные ошибки, что может повлиять на доступность и надежность системы.

# Внедрение автоматизированных информационных систем в сферу бронирования столов в ресторанах представляет собой стратегически важный шаг, который способствует повышению эффективности работы ресторанов, улучшению пользовательского опыта и созданию конкурентных преимуществ на рынке. Однако для достижения максимальной эффективности важно учитывать и своевременно устранять возможные недостатки и ограничения таких систем.

# 1.5 Традиционные алгоритмические языки

* Традиционные алгоритмические языки играют важную роль в создании систем бронирования столов в ресторане. Они позволяют разработчикам создавать гибкие и мощные приложения, которые могут быть адаптированы к специфическим потребностям ресторана. Характерные черты систем бронирования, созданных средствами прямого программирования, включают:
* Разнообразие стилей реализации
* Использование традиционных алгоритмических языков, таких как Python, JavaScript и PHP, позволяет разработчикам реализовывать различные стили и подходы при создании систем бронирования:
* Цветовая палитра: Разработчики могут свободно выбирать и изменять цветовые схемы, чтобы они соответствовали фирменному стилю ресторана, улучшали визуальное восприятие и создавали привлекательный интерфейс.
* Интерфейс: Возможность создания уникальных и интуитивно понятных интерфейсов, которые обеспечивают удобство для пользователей. Это включает в себя кастомные кнопки, формы бронирования, календарные виджеты и прочие элементы.
* Структура системы: Гибкость в организации архитектуры приложения, позволяя реализовывать различные функциональные модули и компоненты, такие как управление пользователями, аналитика бронирований и интеграция с внешними сервисами.
* Способ подачи информации: Возможность интеграции мультимедийных элементов, таких как изображения, видео и анимации, для улучшения пользовательского опыта и повышения информативности.
* Сложность модификации и сопровождения
* Создание систем бронирования с использованием традиционных алгоритмических языков может привести к ряду сложностей:
* Модификация: Изменение существующего кода и добавление новых функций требует значительных усилий, особенно если система была разработана без учета модульности и расширяемости.
* Сопровождение: Поддержка и обновление системы может быть сложным процессом, особенно если разработка велась без должного документирования и стандартизации кода. Это может увеличить время и затраты на сопровождение системы в долгосрочной перспективе.
* Большие затраты времени и трудоемкость
* Разработка системы бронирования с нуля с использованием традиционных алгоритмических языков требует значительных временных и трудовых ресурсов:
* Проектирование: Подробное планирование и проектирование архитектуры системы для обеспечения ее масштабируемости, надежности и безопасности.
* Разработка: Написание большого объема кода для реализации всех необходимых функций и обеспечения их корректной работы. Это включает интеграцию с базами данных, создание пользовательских интерфейсов и обработку запросов.
* Тестирование: Обширное тестирование системы для выявления и устранения ошибок, обеспечения совместимости с различными устройствами и браузерами.
* Отсутствие аппаратных ограничений
* Использование традиционных алгоритмических языков предоставляет возможность создания системы бронирования, которая может быть адаптирована под имеющуюся техническую базу:
* Гибкость: Возможность развертывания системы на различных платформах, включая серверы Windows, Linux и облачные сервисы. Это обеспечивает универсальность и адаптивность системы под разные условия эксплуатации.
* Оптимизация: Возможность оптимизации производительности системы для работы на различных устройствах, включая настольные компьютеры, планшеты и смартфоны, что обеспечивает доступность и удобство использования для широкого круга пользователей.

# 1.6 Инструментальные средства общего назначения

* Инструментальные средства общего назначения (ИСОН) предназначены для создания систем бронирования столов пользователями, которые не являются квалифицированными программистами. ИСОН, применяемые при проектировании таких систем, как правило, обеспечивают следующие возможности:
* Формирование структуры системы бронирования: Предоставляют инструменты для создания логической структуры системы, включая создание базы данных для хранения информации о бронированиях, столах и клиентах.
* Ввод, редактирование и форматирование текста: Включают текстовые редакторы для создания и редактирования текстовых элементов интерфейса, таких как описания столов, подтверждения бронирования и сообщения пользователям.
* Подготовка статической иллюстративной части: Содержат графические редакторы для создания визуальных элементов интерфейса, включая иконки, баннеры и фотографии ресторана.
* Подготовка динамической иллюстративной части: Обеспечивают инструменты для добавления динамических элементов, таких как анимации и аудиовизуальные материалы, чтобы улучшить пользовательский интерфейс и взаимодействие.
* Подключение исполняемых модулей: Позволяют интегрировать дополнительные модули и плагины, разработанные с применением других средств разработки, для расширения функциональности системы бронирования.
* К достоинствам инструментальных средств общего назначения следует отнести:
* Возможность создания системы бронирования лицами, которые не являются квалифицированными программистами: Инструменты предоставляют визуальные редакторы и простые интерфейсы, которые облегчают процесс создания и настройки системы.
* Существенное сокращение трудоемкости и сроков разработки системы: Предоставляют готовые шаблоны и компоненты, что значительно ускоряет процесс разработки и уменьшает затраты времени и усилий.
* Невысокие требования к компьютерам и программному обеспечению: Могут работать на стандартных офисных компьютерах и не требуют специализированного программного обеспечения или оборудования.
* Вместе с тем ИСОН имеют ряд недостатков, таких как:
* Не всегда дружественный интерфейс: Некоторые инструментальные средства могут быть сложны в освоении для пользователей без технического образования, что затрудняет их эффективное использование.
* Меньшие возможности по сравнению с специализированными системами: Ограничены в функциональности и гибкости по сравнению с мультимедийными и гипермедийными системами, что может ограничивать возможности кастомизации и расширения.
* Отсутствие возможности создания сложных программ управления и аналитики: ИСОН часто не предоставляют возможности для разработки продвинутых функций, таких как аналитика данных, управление клиентами и интеграция с внешними системами.
* В заключение, инструментальные средства общего назначения представляют собой полезный инструмент для разработки систем бронирования столов в ресторанах, особенно для пользователей без глубоких знаний в программировании. Однако, несмотря на их простоту и доступность, они имеют определенные ограничения, которые необходимо учитывать при выборе подходящего инструмента для конкретного проекта.

Среда разработки PhpStorm

Выбор среды разработки PhpStorm обоснован следующими причинами:

1. Интеграция с php:

PhpStorm обладает мощной интеграцией с php, что обеспечивает высокую производительность разработчика. Он предоставляет множество инструментов и функций, специально адаптированных для работы с php: автодополнение кода, быструю навигацию по проекту и автоматическую проверку синтаксиса.

2. Богатый набор инструментов:

PhpStorm предлагает широкий выбор инструментов и функций для удобной и эффективной разработки веб-приложений на PHP. Среди них: отладчик, система контроля версий, автоматическое дополнение кода, интеграция с базами данных, система управления зависимостями и многое другое.

3. Удобство использования:

PhpStorm отличается удобным и интуитивно понятным интерфейсом, что делает его привлекательным выбором для разработчиков всех уровней опыта. Он предоставляет множество настраиваемых параметров и возможностей для оптимизации рабочего процесса.

4. Поддержка различных технологий:

PhpStorm поддерживает не только PHP, но также широкий набор других технологий, используемых в веб-разработке, таких как HTML, CSS, JavaScript, TypeScript и другие. Это обеспечивает единый инструмент для работы с различными аспектами проекта.

5. Большое сообщество и поддержка:

PhpStorm имеет огромное сообщество пользователей и разработчиков, что обеспечивает доступ к богатой базе знаний, документации, форумам поддержки и расширениям. Это делает процесс разработки более прозрачным и эффективным, а также обеспечивает оперативное решение возникающих проблем.

В результате, использование среды разработки PhpStorm обеспечивает высокую производительность, удобство и эффективность при разработке на фреймворке Laravel.

* 1. Локальный сервер Xampp

XAMPP (от кросс-платформенный (X), Apache (A), MySQL (M), PHP (P), Perl (P)) - это полноценный пакет для создания локального сервера, который позволяет разработчикам легко создавать и тестировать веб-сайты и веб-приложения на своем собственном компьютере. Вот подробнее о его особенностях и преимуществах:

Комплексное решение: XAMPP предлагает все, что нужно для разработки веб-приложений, включая веб-сервер Apache, базу данных MySQL, интерпретатор PHP и Perl. Это позволяет начать работу над проектом сразу после установки, без необходимости установки и настройки каждого компонента отдельно.

Кросс-платформенность: XAMPP поддерживает все основные операционные системы, включая Windows, Linux и macOS. Это обеспечивает гибкость для разработчиков, позволяя им работать с XAMPP на своей предпочитаемой платформе.

Простота установки и использования: Установка XAMPP обычно занимает всего несколько кликов и не требует сложной настройки. После установки XAMPP можно легко запустить и начать использовать, что делает его идеальным для начинающих разработчиков.

Локальная разработка: XAMPP позволяет создавать и тестировать веб-приложения локально на вашем компьютере. Это удобно для разработчиков, так как они могут вносить изменения и проверять работу своих проектов без необходимости загрузки их на удаленный сервер.

Поддержка различных версий PHP и MySQL: XAMPP предлагает возможность выбора различных версий PHP и MySQL для разработки. Это позволяет разработчикам работать с различными версиями и проверять совместимость своих приложений.

Большое сообщество и поддержка: XAMPP имеет широкое сообщество пользователей и разработчиков, что обеспечивает доступ к большому объему документации, форумов поддержки и обмену опытом.

Бесплатность и открытый исходный код: XAMPP является свободно распространяемым и полностью бесплатным программным обеспечением с открытым исходным кодом. Это означает, что вы можете использовать его без ограничений и даже вносить свои изменения, если это необходимо.

Встроенные дополнительные инструменты: Помимо основных компонентов, XAMPP также включает в себя некоторые дополнительные инструменты, такие как phpMyAdmin (инструмент администрирования MySQL), Mercury Mail Server (SMTP-сервер для отправки почты), FileZilla FTP Server (FTP-сервер для передачи файлов) и другие. Эти инструменты могут быть полезны для различных задач разработки и тестирования.

Легкость конфигурации: XAMPP предоставляет простой и интуитивно понятный интерфейс для конфигурации и управления его компонентами. Пользователи могут легко настраивать параметры веб-сервера, базы данных и других компонентов с помощью встроенных инструментов управления.

Возможность установки на портативное устройство: XAMPP предлагает портативную версию, которая может быть установлена на USB-флеш-накопитель или другое портативное устройство. Это позволяет вам взять свой локальный сервер с собой и запускать его на любом компьютере без необходимости повторной установки.

Обновления и поддержка: Разработчики XAMPP регулярно выпускают обновления, в которых учитывают новые функции и исправляют ошибки безопасности. Кроме того, XAMPP имеет активное сообщество пользователей и разработчиков, которые могут предоставить поддержку и помощь при возникновении проблем.

Использование в образовательных целях: XAMPP часто используется в учебных заведениях для обучения студентов веб-разработке и базам данных. Его простота установки и использования делает его отличным выбором для практических занятий и лабораторных работ.

В целом, XAMPP - это мощное и удобное средство для разработки веб-приложений локально на вашем компьютере. Его простота установки и использования, а также обширный набор функций делают его популярным выбором среди разработчиков всех уровней опыта.

# РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ БРОНИРОВАНИЯ СТОЛОВ В РЕСТОРАНЕ



# 2.1 Техническое задание

Наименование работы: «Разработка автоматизированной информационной системы бронирования столов в ресторане»

Наименование программы: «BELHOTEL»

Сведения о заказчике

Заказчиков является общество с ограниченной ответственностью фирма ООО «Респекталь» Краснодарский г.Белореченск, ул.Карла Либкнехта, 99/1 Директор Акопян Алексан Стёпанович. Основной вид деятельности является: Ресторан

Назначение и область применения

Программа предназначена бронировании столов ресторана организации ООО “Респекталь”.

Требования к программе

Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

* Возможность добавления, редактирования и удаления
* запоминающийся приятный дизайн;
* удобный интерфейс;
* интуитивно понятная навигация;
* наличие ссылок;
* отображение в любом браузере и с помощью разных устройств;

Требования к надежности

Требования к обеспечению надежного функционирования программы:

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

* Доступ к хостингу должен быть предоставлен только администратору или ответственному лицу;
* Обработка .htaccess;
* Разрешен short\_open\_tag в PHP: Да.
* Значение memory\_limit в PHP (и реальное ограничение памяти): 32 Mб и выше.
* Отправка почты (функция mail()): Нет.
* Функции работы с сокетами: подключение к серверам на порт 80.
* Создание файлов и папок от процесса, выполняющего PHP, запуск созданных файлов.
* Загрузка файлов (картинки должны корректно отображаться).
* FTP и SSH доступ к хостингу.

Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) управляющей системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) управляющей системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы из-за некорректных действий пользователей системы

Отказы программы вследствие некорректных действий пользователя при взаимодействии с программой возможны только в случае удаления файлов системы.

Условия эксплуатации

Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

Для работы с программой необходим один (пользователь). В перечень задач, выполняемых системным администратором, должны входить:

* задача поддержания работоспособности технических средств;
* задачи установки (инсталляции) и поддержания работоспособности системных программных средств — операционной системы;
* задача установки (инсталляции) программы.
* задача создания резервных копий сайта.

Требования к составу и параметрам технических средств

В состав технических средств должен входить сервер под управлением ОС Linux или другой операционной системы с поддержкой возможности размещения сайта:

* Оперативная память: не менее 32 МБ;
* Свободное место на сервере: не менее 10 МБ;
* Установленное браузерное приложение для просмотра страниц сайта;

Требования к информационной и программной совместимости

Требования к информационным структурам и методам решения

Требования к информационным структурам и методам решения не предъявляются.

Требования к исходным кодам и языкам программирования

Дополнительные требования не предъявляются.

Требования к программным средствам, используемым программой

Требования к защите информации и программ

Требования к защите информации и программ не предъявляются.

Специальные требования

Специальные требования не предъявляются.

Требования к программной документации

Состав программной документации должен включать в себя:

* техническое задание;
* руководство пользователя;
* руководство программиста.

Стадии и этапы разработки

Стадии разработки

Разработка должна быть проведена в три стадии:

* разработка технического задания;
* рабочее проектирование;
* внедрение.

Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

* разработка программы;
* разработка программной документации;
* испытания программы.

На стадии внедрения должен быть выполнен этап разработки, подготовка и передача программы.

Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

* постановка задачи;
* определение и уточнение требований к техническим средствам;
* определение требований к программе;
* определение стадий, этапов и сроков разработки программы и документации на неё;
* согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

* разработка, согласование и утверждение и методики испытаний;
* проведение приемо-сдаточных испытаний;
* корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика.

Порядок контроля и приемки

Виды испытаний

Приемо-сдаточные испытания должны проводиться на объекте Заказчика в оговоренные сроки.

Приемо-сдаточные испытания программы должны проводиться согласно разработанной Исполнителем и согласованной Заказчиком Программы и методик испытаний.

Ход проведения приемо-сдаточных испытаний Заказчик и Исполнитель документируют в Протоколе проведения испытаний.

Общие требования к приемке работы

На основании Протокола проведения испытаний Исполнитель совместно с Заказчиком подписывает Акт приемки-сдачи программы в эксплуатацию.

Была изучена предметная область и разработана функциональная структура сайта, которая представлена на рисунке 1

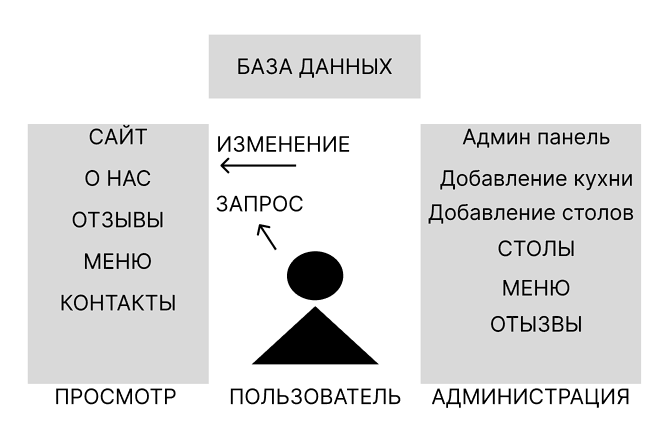


Рисунок 1 – Функциональная структура сайта

Разрабатываемый сайт ООО "РЕСПЕКТАЛЬ" (BELHOTEL) имеет следующие основные разделы:

* Раздел «О нас»:

Подробно описывает компанию деятельность и предоставляемые услуги.

* Раздел «Отзывы»:

Содержит форму для заполнения отзыва пользователя сайта.

* Раздел «Меню»:

Содержит в себе все блюда которые предоставляет ресторан.

* Раздел «Контакты»:

Список наших контактов

На сайте предусмотрена панель администратора, доступ к которой возможен после успешного перехода по ссылке <http://localhost/diplom/public/admin_panel.php>, доступным только сотрудникам ресторана. Панель администратора включает в себя следующие функции:

- Добавления стола : Возможность добавить новые столы на сайт.

- Удаление стола : Удаление существующих столов на сайте.

Основные задачи проекта

В процессе работы на сайтом должны быть выполнены следующие задачи:

- Проектирование сайта:

Разработка структуры и навигации.

- Разработка информационно-логической модели;

Создание логической структуры базы данных и взаимосвязей между сущностями.

- Оценка трудоемкости проекта:

Оценка времени и ресурсов, необходимых для выполнения проекта.

- Разработка программного обеспечения:

Реализация серверной и клиентской частей сайта.

- Разработка руководств:

Создание инструкций для пользователей и администраторов сайта.

Система бронирования столов в ресторане должна предоставлять исчерпывающую информацию о доступных столах, текущих бронированиях, а также возможности для персонала управлять бронированиями. Основные функциональные требования включают:

Информация о ресторане и услугах:

Полное описание ресторана, меню, график работы и контактная информация.

Информация о специальных предложениях и мероприятиях.

Функционал бронирования:

Возможность бронирования столов на определенную дату и время.

Проверка доступности столов в реальном времени.

Опция выбора предпочтительных мест (например, возле окна, в отдельном зале).

Управление бронированиями:

Просмотр, изменение и отмена бронирований как клиентами, так и администрацией.

Уведомления о подтверждении или изменении бронирования через электронную почту.

Визуальные требования

Визуальное оформление системы бронирования столов должно включать:

Фотографии и иллюстрации:

Визуальное сопровождение информации о ресторане для улучшения восприятия.

Фотографии интерьера, блюд и мероприятий.

Простой и красочный дизайн:

Дизайн интерфейса должен быть понятным и запоминающимся, соответствующим целевой аудитории.

Цветовая схема и шрифты должны создавать приятное впечатление и быть легко читаемыми.

Единый стиль оформления:

Все страницы системы должны быть выполнены в одном стиле для создания целостного впечатления.

Цельность в элементах дизайна и навигации.

Адаптивность

Система бронирования должна быть удобной для просмотра и использования на различных устройствах, таких как:

- Мобильные телефоны

- Планшеты

- Ноутбуки

- Мониторы ПК

- Взаимодействие с пользователями

- Для клиентов:

Просмотр информации о ресторане и текущих бронированиях.

Оформление новых бронирований через интуитивно понятный интерфейс.

Получение уведомлений о статусе бронирования.

Для администраторов:

Авторизация для доступа к панели управления.

Управление содержимым системы, бронированиями и пользователями.

Диаграммы

При проектировании системы бронирования столов были разработаны следующие диаграммы:

Контекстная диаграмма:

Представляет систему как набор иерархических действий. Высшее действие называется действием контекста и описывает систему на самом высоком уровне. Более низкие уровни представляют подпроцессы родительского действия.

Диаграмма IDEF0:

Модель для функционального анализа и проектирования системы. Помогает в определении функций системы и их взаимодействий.

Диаграмма вариантов использования:

Описывает различные сценарии взаимодействия пользователей с системой. Включает сценарии бронирования столов, управление бронированиями и авторизацию администраторов.

Диаграмма деятельности:

Визуализирует поток выполнения действий в системе. Помогает в понимании последовательности процессов бронирования и управления.

Диаграмма состояний:

Определяет состояния объектов системы и переходы между этими состояниями. Например, состояние столов (свободен, забронирован, занят) и переходы между этими состояниями в зависимости от действий пользователей.

Эти шаги обеспечивают эффективное проектирование и реализацию сайта, его корректную работу и удовлетворение потребностей пользователей.

Контекстная диаграмма представлена на рисунке 2.

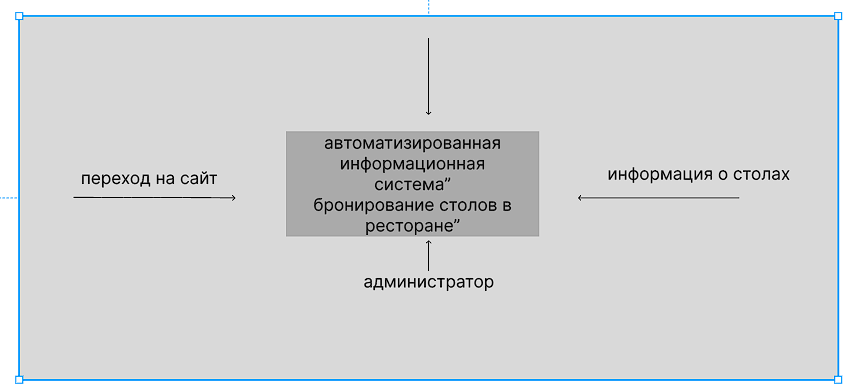


Рисунок 2 – Контекстная диаграмма

В таблице 1 представлено описание стрелок данной контекстной диаграммы.

Таблица 1 – Описание стрелок контекстной диаграммы

|  |  |
| --- | --- |
| Наименование стрелки | Описание |
| Пользователь | Входит на сайт |
| Администратор | Администратор сайта, |
| Информация о турах | Информация о выбранных столах |

Методология IDEF0 предписывает построение иерархической системы диаграмм – единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная деком – позиция – система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции).

Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности.

Каждая IDEF0 – диаграмма содержит блоки и дуги. Блоки изображают функции моделируемой системы. Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними.

Диаграмма IDEF0 представлена на рисунке 3.

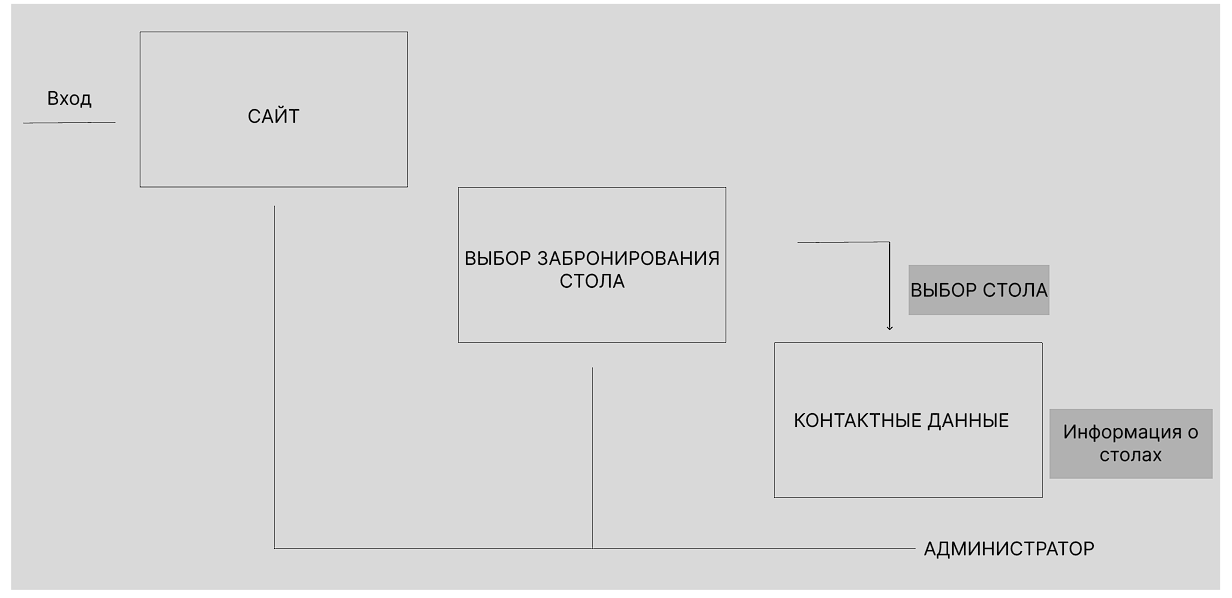


Рисунок 3 – Диаграмма IDEF0

Основные элементы модели IDEF0 описаны в таблице 2.

Таблица 2 – Основные элементы модели IDEF0

|  |  |
| --- | --- |
| Название проекта: Автоматизированная информационная система “Бронирования столов в ресторане” | |
| Цель проекта: Реализация структурной функциональной модели сайта | |
| Технология моделирования: метод функционального моделирования IDEF0 | |
| Инструментарий: программный продукт Figma | |
| Список данных | Перечень функций |
| Администратор  Информация по столам | 1. Автоматизированная информационная система “Бронирование столов в ресторане” |
| Сайт  Столы  Выбранный стол | 2. Вход  3. Выбор стола  4. Контактные данные |

Описание функциональных блоков диаграммы IDEF0 представлено в таблице 3.

Таблица 3 – Описание функциональных блоков IDEF0

|  |  |
| --- | --- |
| Наименование блока | Описание решаемых задач |
| 1. Вход | Осуществляется вход пользователя. |
| 2. Выбор стола | Пользователь осуществляет выбор интересующего стола. |
| 3. Контактные данные | После выбора интересующего пользователя стола имеются контактные данные для связи или позвонят. |

Пользователи всегда идут по сценарию на диаграмме представляет внешнюю сущность, которая взаимодействует с системой.

Диаграмма вариантов использования представлена на рисунке 4.

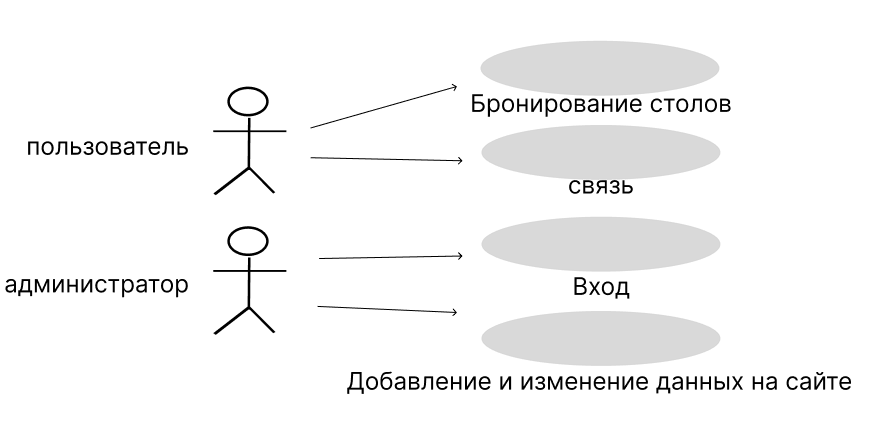


Рисунок 4 – Диаграмма вариантов использования

По рисунку выделим два актера: пользователь и администратор.

Пользователем является любой неавторизованный пользователь работающий с сайтом, который может осуществлять просмотр информации, а также бронировать столы, администратор может добавлять и изменять данные сайта.

Описание раздела «Входа» представлено в таблице 4

Таблица 4 – Раздел «Входа»

|  |  |
| --- | --- |
| Вариант использования | Вход |
| Актеры | Администратор |
| Цель | Получение доступа к панели администратора |
| Краткое описание | Пользователь вводит спец ссылку для входа в панель администратора |
| Тип | Базовый |

Описание раздела «Типичный ход событий» представлено в таблице 5.

Таблица 5 – Раздел «Типичный ход событий»

|  |  |
| --- | --- |
| Действия актера | Отклик системы |
| 1. Администратор вводит спец ссылку | 2Сайт переходит в админ панель |
| 3.Администратор попадает на админ панель | 4.Сайт отображает скрытые функции для администратора |

Таблица 6 – Требования к системе Бронирования столов ресторана

|  |  |  |
| --- | --- | --- |
| 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 |

Продолжение таблицы 6 – Требования к системе

|  |  |  |
| --- | --- | --- |
| 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 7 представлены требования к надежности.

Таблица 7 – 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 8 представлен перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.

Таблица 8 – 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;

– 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 9.

Таблица 9 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 меню представлены в таблице menu\_dishes 10.

Таблица 10 – menu\_dishes – Данные о меню.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 11.

Таблица 11 reviews – Данные об отзывах.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничения |
| Id | Int | 11 | индентификатор | Первичный |
| 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 12.

Таблица 12 tables – Данные о столах

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничения |
| id | INT | 255 | Идентификатор | Превичный |
| 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 |

Установка и разработка среды разработки PHPStorm:

Скачать PHPStorm:

1. Перейдите на официальный сайт PHPStorm по адресу: <https://www.jetbrains.com/phpstorm/download/> (рисунок 5.)
2. На странице загрузки выберите нужную версию (Community или Ultimate) и нажмите "Download".
3. Следуйте инструкциям на экране, чтобы скачать установочный файл для вашей операционной системы.

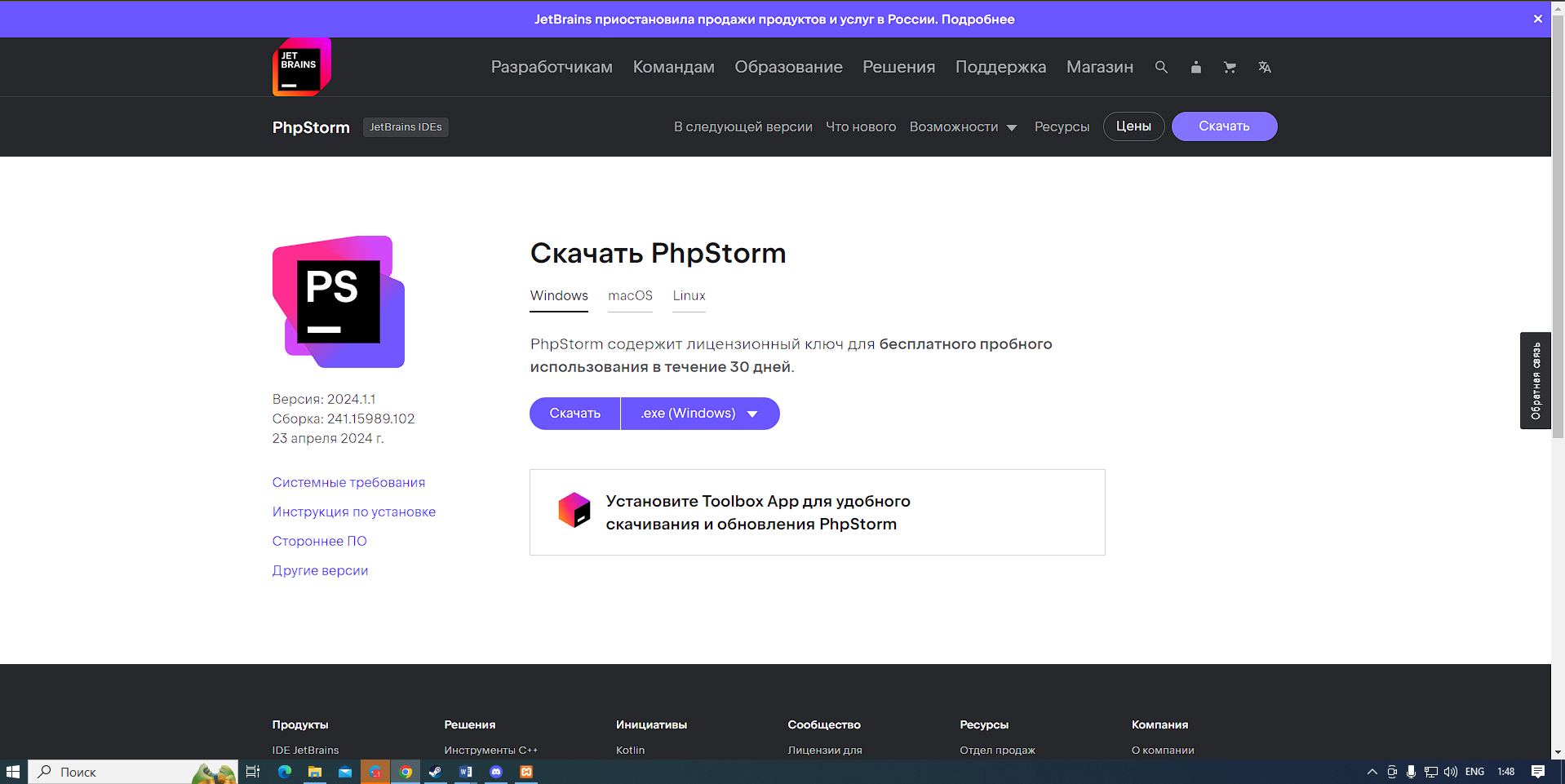


Рисунок 5. - Официальный сайт PHPStorm

Установить PHPStorm:

1. Запустите установочный файл, который вы скачали.
2. Следуйте инструкциям мастера установки, чтобы установить PHPStorm на ваш компьютер. (рисунок 6.)

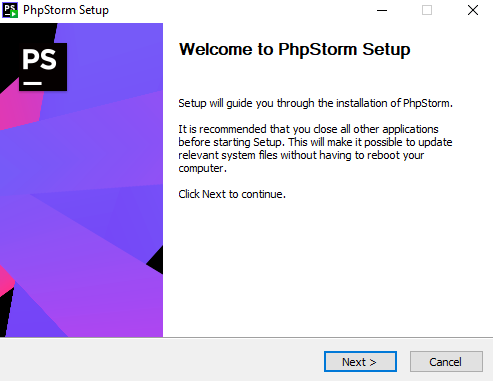


Рисунок 6. - Инструкция по установке PHPStorm

1. После завершения установки запустите PHPStorm.

Настройка PHPStorm:

1. При первом запуске вас попросят настроить некоторые основные параметры, такие как выбор темы оформления, раскладки клавиатуры и т. д. Следуйте указаниям мастера настройки. (Рисунок 7.)
2. Выберите путь к JDK (если требуется), если он не был найден автоматически.
3. Создайте или войдите в учетную запись JetBrains, чтобы активировать PHPStorm (или продолжить с пробной версией).

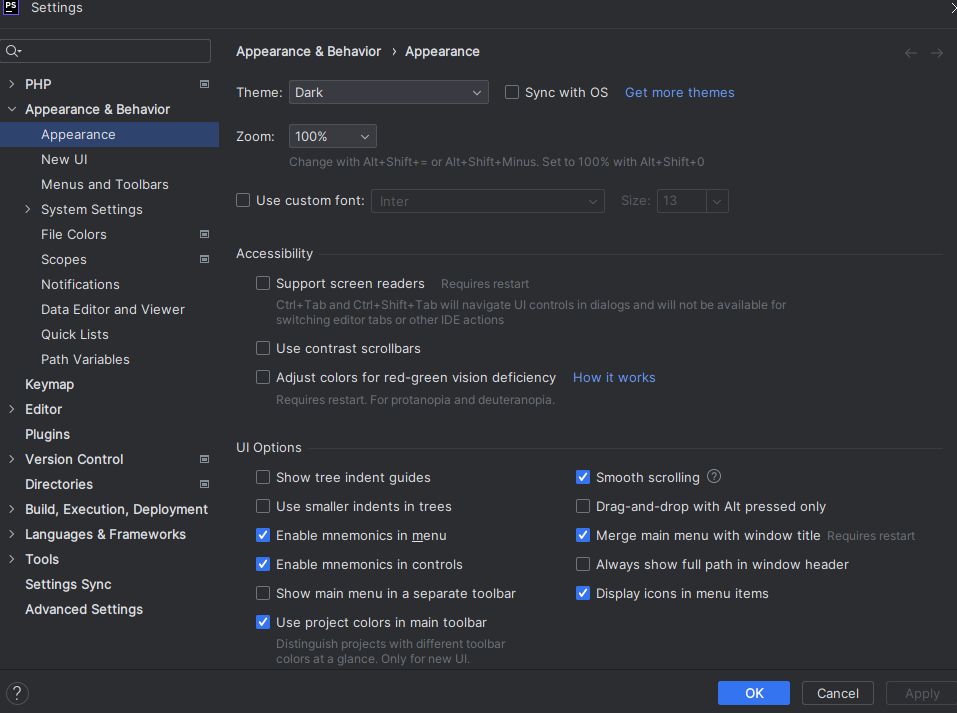


Рисунок 7. - Настройка PHPStorm

Настройка PHP и веб-сервера:

1. Убедитесь, что на вашем компьютере установлен PHP. Если нет, установите его с помощью инструкций для вашей операционной системы. Рекомендуется использовать XAMPP или WAMP для установки PHP и веб-сервера на Windows, либо MAMP на macOS.
2. В PHPStorm откройте настройки (Settings/Preferences) и найдите раздел "Languages & Frameworks" -> "PHP". Укажите путь к исполняемому файлу PHP на вашем компьютере.
3. Если вы используете веб-сервер для разработки (например, Apache), настройте его в PHPStorm, указав путь к директории, в которой находятся ваши проекты.

Создание проекта:

1. Нажмите на "Create New Project" при первом запуске или выберите "New Project" из меню File.
2. Укажите путь к папке, в которой будет храниться ваш проект.
3. Выберите тип проекта (например, PHP Empty Project, PHP Web Application) и нажмите "Create".

Начало разработки:

1. PHPStorm предлагает множество возможностей для увеличения производительности разработки, таких как автодополнение кода, отладка, интеграция с системами контроля версий и многое другое. Изучайте и используйте эти функции для повышения эффективности вашей работы.

Использование библиотек Bootstrap а так же их плюсы и минусы

1. <https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css> (ссылка на CSS-файл Bootstrap)

Описание: Это ссылка на CSS-файл Bootstrap, который предоставляет стили для веб-страниц, включая сетку, типографику, формы, кнопки и другие компоненты интерфейса.

Плюсы:

Предоставляет готовые стили для быстрой разработки пользовательского интерфейса.

Широко используется и имеет обширное сообщество поддержки.

Минусы:

Может быть избыточным, если используются только некоторые компоненты, так как размер файла CSS относительно большой.

Стили могут не всегда соответствовать уникальным дизайнам проекта.

1. <https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js> (JavaScript-файл Bootstrap)

Описание: Это ссылка на JavaScript-файл Bootstrap, который предоставляет функциональность для работы с компонентами интерфейса Bootstrap, такими как модальные окна, всплывающие подсказки, выпадающие списки и т. д.

Плюсы:

Предоставляет готовый JavaScript для быстрой настройки функциональности интерфейса.

Хорошо интегрируется с CSS-файлом Bootstrap.

Минусы:

Может добавить лишний объем JavaScript-кода к проекту, что может негативно сказаться на производительности.

1. <https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js> (JavaScript файл библиотеки Propper.js)

Описание: Это ссылка на JavaScript-файл библиотеки Popper.js, который используется для создания всплывающих подсказок и позиционирования элементов интерфейса.

Плюсы:

Popper.js обеспечивает гибкое и точное позиционирование элементов на веб-странице.

Поддерживает различные сценарии использования, такие как всплывающие подсказки и выпадающие списки.

Минусы:

Добавляет дополнительный объем JavaScript-кода к проекту.

Может потребоваться время на изучение и использование правильных параметров для настройки позиционирования.

1. <https://code.jquery.com/jquery-3.5.1.js> (JavaScript-файл библиотеки jQuery)

Описание: Это ссылка на JavaScript-файл библиотеки jQuery, которая предоставляет множество функций для облегчения работы с DOM-элементами, анимацией, AJAX-запросами и другими задачами.

Плюсы:

jQuery упрощает манипуляции с DOM и обработку событий, делая код более читаемым и поддерживаемым.

Имеет обширную документацию и широкое распространение.

Минусы:

Может быть избыточным в некоторых случаях, особенно если используются только небольшие части его функциональности.

Размер файла jQuery относительно большой, что может повлиять на время загрузки страницы.

1. <https://cdn.jsdelivr.net/npm/sweetalert2@11/dist/sweetalert2.all.min.js> (JavaScript-файл библиотеки SweetAlert2)

Описание: Это ссылка на JavaScript-файл библиотеки SweetAlert2, предназначенной для создания красивых и адаптивных модальных окон с предупреждениями, сообщениями об успехе и другими типами сообщений.

Плюсы:

SweetAlert2 предоставляет простой способ создания красивых и адаптивных модальных окон на веб-странице.

Имеет множество параметров настройки, позволяющих создавать различные стили и типы модальных окон.

Минусы:

Может добавить дополнительный объем JavaScript-кода к проекту.

Использование библиотеки может сделать код менее модульным, если она используется в избытке без необходимости.

1. <https://nosir.github.io/cleave.js/dist/cleave.min.js> (JavaScript-файл библиотеки Cleave.js)

Описание: Это ссылка на JavaScript-файл библиотеки Cleave.js, которая предназначена для форматирования и валидации вводимых пользователем данных в текстовых полях.

Плюсы:

Cleave.js обеспечивает удобный способ форматирования вводимых данных, таких как телефонные номера, даты и т. д.

Позволяет легко настроить правила форматирования и валидации для различных типов данных.

Минусы:

Дополнительная зависимость в проекте, что может увеличить размер файлов и время загрузки страницы.

Может потребоваться изучение документации для настройки и правильного использования.

1. <https://cdn.jsdelivr.net/npm/sweetalert2@11/dist/sweetalert2.all.min.js> (JavaScript-файл библиотеки SweetAlert2)

Описание: Это ссылка на JavaScript-файл библиотеки SweetAlert2, которая используется для создания красивых и адаптивных модальных окон с предупреждениями, сообщениями об успехе и другими типами сообщений.

Плюсы:

SweetAlert2 предоставляет простой способ создания красивых и адаптивных модальных окон на веб-странице.

Имеет множество параметров настройки, позволяющих создавать различные стили и типы модальных окон.

Минусы:

Может добавить дополнительный объем JavaScript-кода к проекту.

Использование библиотеки может сделать код менее модульным, если она используется в избытке без необходимости.

1. PDO (PHP Data Objects) - это расширение PHP, которое предоставляет унифицированный интерфейс для доступа к базам данных из PHP. (рисунок 8.)

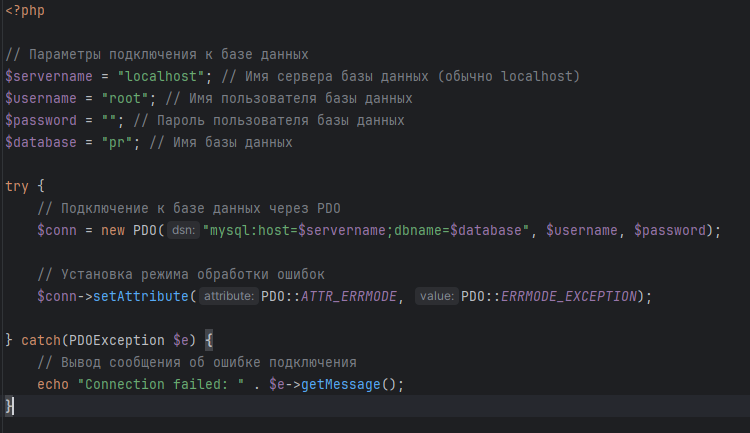


Рисунок 8. - Пример использования PDO

Структура разрабатываемого проекта показана на рисунке 9.

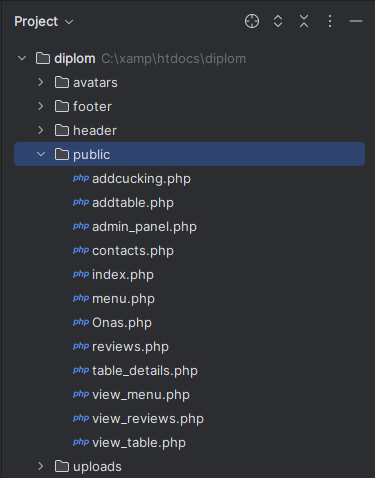


Рисунок 9. - структура проекта

avatars: Предположительно, содержит изображения аватаров пользователей или других графических элементов, связанных с профилями пользователей.

footer: содержит файлы или скрипты, связанные с подвалом (footer) веб-страниц.

header: Содержит файлы или скрипты, связанные с шапкой (header) веб-страниц.

public: Основная папка, в которой находятся все основные файлы веб-приложения, доступные для публичного доступа. Внутри этой папки содержится множество PHP-файлов, выполняющих различные функции веб-приложения.

uploads: эта папка используется для хранения загруженных файлов, таких как изображения, документы или другие медиафайлы, загружаемые пользователями.

PHP-файлы в папке public

adducking.php: файл для добавления управления данными о них (специфично для конкретного проекта).

addtable.php: используется для добавления новой таблицы данных или резервации столика.

admin\_panel.php: Файл, содержащий код для административной панели, где администратор может управлять различными аспектами веб-приложения.

contacts.php: Страница контактов, где пользователи могут найти контактную информацию или отправить сообщение.

index.php: Главная страница веб-приложения.

menu.php: Страница, отображающая меню (например, ресторанное меню).

Onas.php: Страница "О нас", предоставляющая информацию о компании или сайте.

reviews.php: Страница для отображения или добавления отзывов пользователей.

table\_details.php: содержит детали о таблицах данных или столиках.

view\_menu.php: Страница для просмотра меню.

view\_reviews.php: Страница для просмотра отзывов.

view\_table.php: Страница для просмотра информации о столиках или таблицах данных.

Установка локального сервера XAMPP

Установка локального сервера XAMPP включает в себя следующие шаги:

1. Скачивание установочного файла.

Перейдите на официальный сайт Apache Friends по адресу <https://www.apachefriends.org/ru/index.html> и выберите раздел загрузки (Download). Затем выберите версию XAMPP для вашей операционной системы и нажмите на кнопку «Скачать» (Download).

1. Запуск установки.

После загрузки файла запустите его и следуйте инструкциям на экране. Выберите язык установки, установите путь установки и выберите компоненты, которые вы хотите установить. По умолчанию устанавливаются все необходимые компоненты, такие как Apache, MySQL, PHP и т.д.

1. Настройка сервера.

После установки XAMPP, запустите контрольный панель и выберите необходимые сервисы (Apache, MySQL и т.д.). Затем нажмите на кнопку «Start» для запуска выбранных сервисов.

1. Тестирование работы сервера.

Откройте браузер и введите в адресной строке <http://localhost/>. Если вы видите страницу приветствия XAMPP, значит сервер успешно установлен и запущен.

1. Настройка базы данных.

Для создания базы данных можно использовать утилиту phpMyAdmin, которая входит в состав XAMPP. Откройте браузер и введите в адресной строке <http://localhost/phpmyadmin/>. Затем создайте новую базу данных и настройте её параметры под свои нужды.

1. Настройка веб-приложения.

Чтобы запустить ваше веб-приложение на локальном сервере XAMPP, скопируйте все необходимые файлы в папку «htdocs», которая находится в папке установки XAMPP. Затем откройте браузер и введите в адресной строке путь к вашему веб-приложению.

Это базовый набор шагов для установки локального сервера XAMPP. При необходимости могут быть добавлены дополнительные настройки и параметры в зависимости от конкретных потребностей проекта.

2.2 Создание базы данных и подключение к проекту

Для того чтобы подключить базу данных к нашему проекту ее для начала нужно создать.

Необходимо зайти в браузер, и в адресную строку ввести <http://localhost/phpmyadmin/>, мы попадаем на страницу phpMyAdmin где и будет создана наша база данных. На рисунке 5 продемонстрирован интерфейс phpMyAdmin.

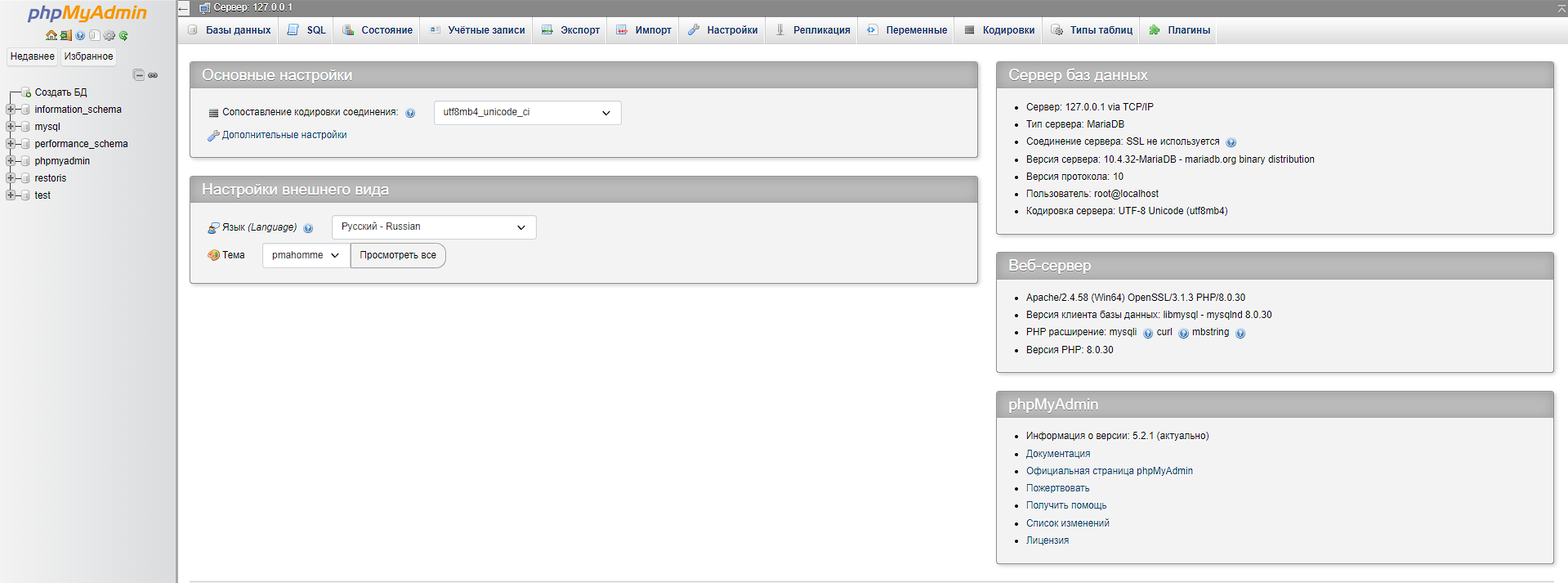


Рисунок 10 - Интерфейс phpMyAdmin.

Затем в левом верхнем углу нажимаем создать БД. На рисунке 11 продемонстрировано создание БД.

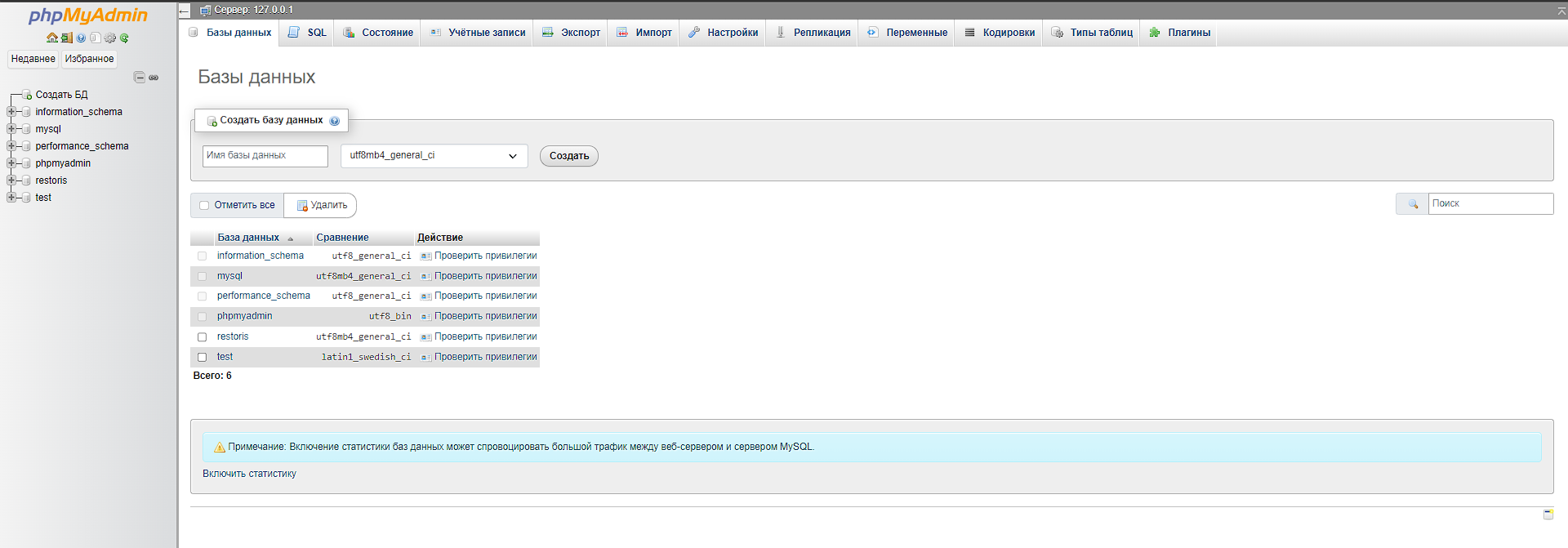


Рисунок 11 - Создание БД.

Необходимо ввести название БД sec и нажать кнопку создать.

На рисунке 12 продемонстрирована страница на которую меня перебросит после того как я нажму кнопку создать, так же наша БД sto появится справа в списке.

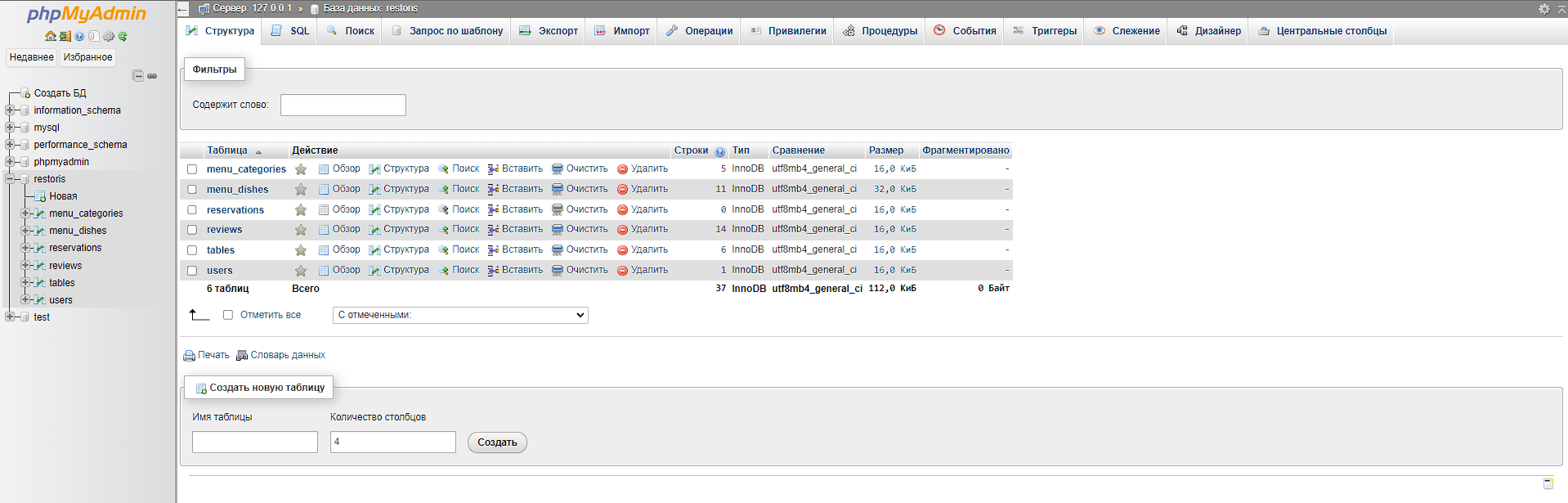


Рисунок 12 - База данных restoris.

Затем чтобы получить доступ к этой базе данных в приложении мы должны создать пользователя. Для этого нажимаю на кнопку “Привилегии”. На рисунке 13, 14, 15 продемонстрирована страница “Привилегии” phpMyAdmin.

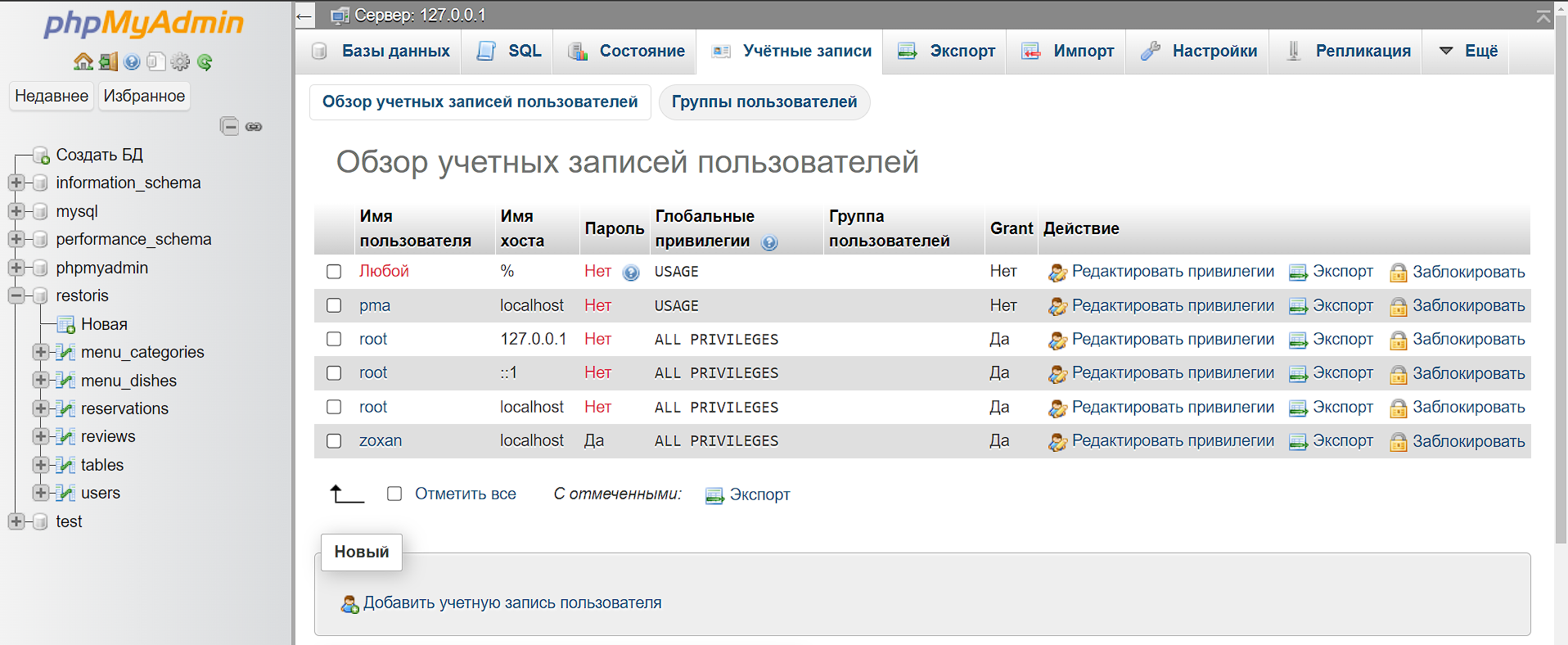


Рисунок 13 - Учетные записи пользователей PhpMyAdmin.

После мы выбираем нашу базу данных (sec).

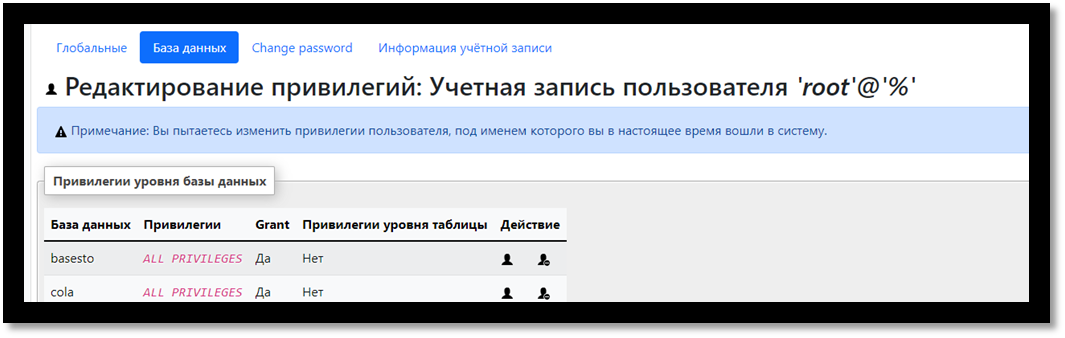


Рисунок 14 - Выбор базы данных.

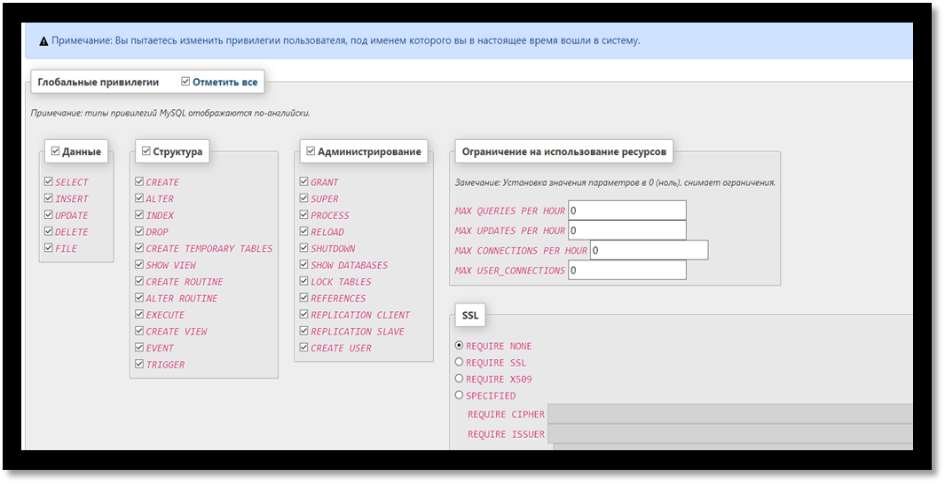


Рисунок 15 - Выбор привилегий пользователя для нашей БД.

После того создается БД и пользователя, необходимо подключить ее к приложению. На рисунке 8 продемонстрирована конфигурация bd\_connect для подключения к нашей БД.

2.3 Руководство пользователя

Мы попали на сайт, навигация сайта находиться сверху, где и логотип сайта, а также мы можем переходить на следующие страницы с помощью навигации сайта: О нас, Меню, Отзывы, Контакты и в добавок есть кнопка вернуться на прошлую страницу.

Навигация продемонстрирована на рисунке 16

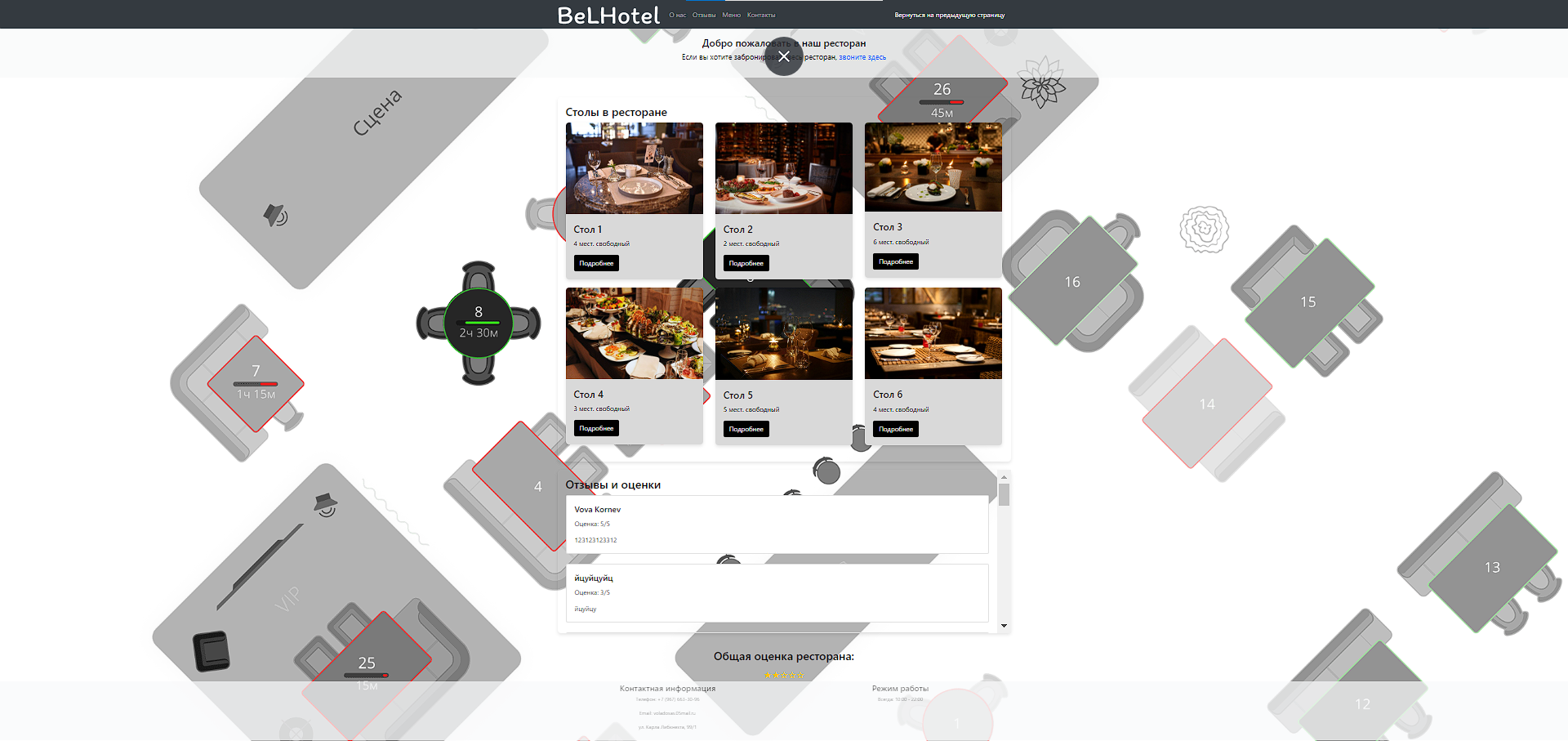


Рисунок 16 – Навигация

После того как мы нажмем на кнопку подробнее, мы попадем на страницу со столом какой пользователь выбрал

На рисунке 17 изображена страница стола который можно забронировать или посмотреть свободен он или нет а также ниже есть информация связи и отзывы.

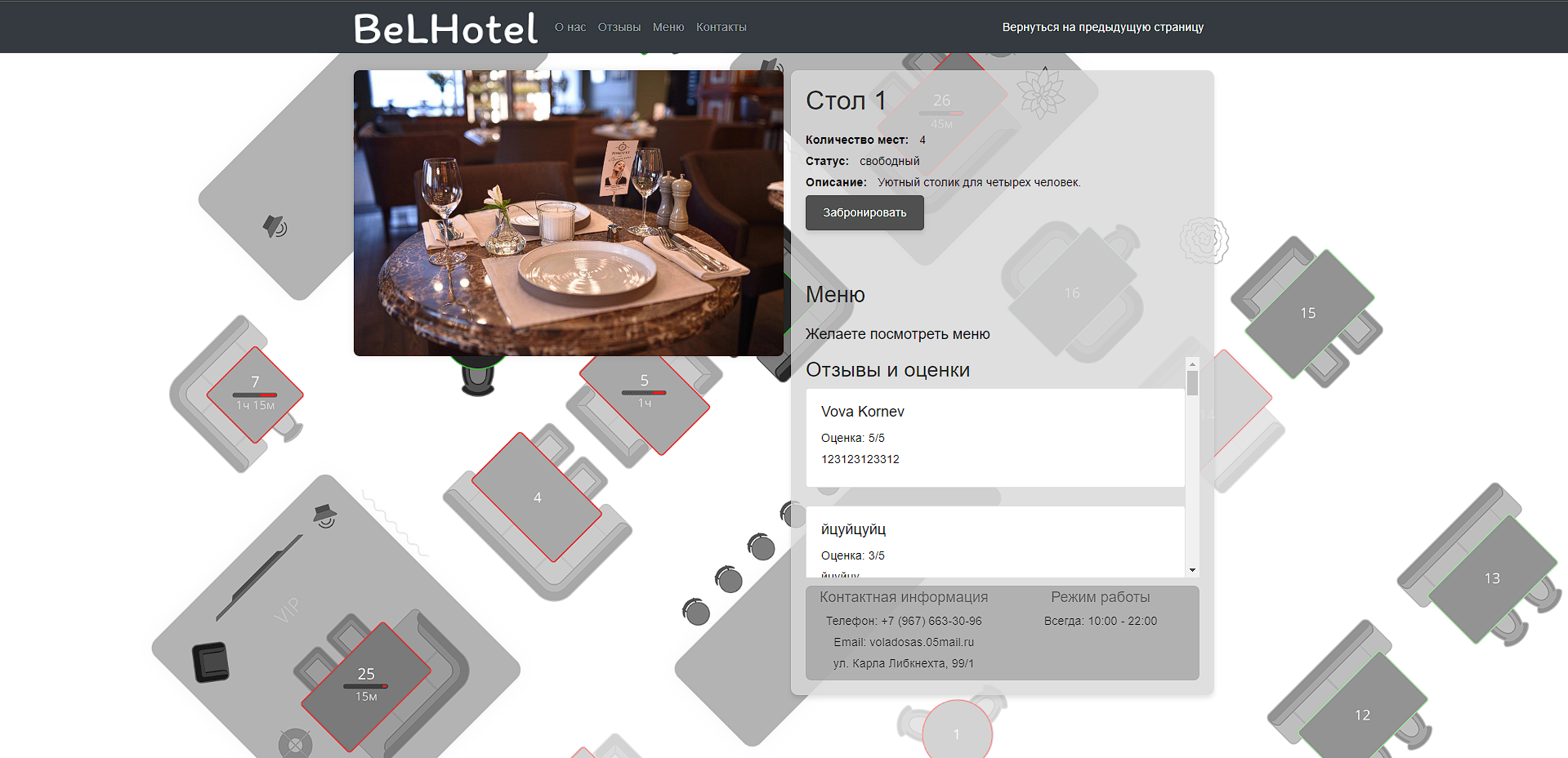


Рисунок 17 Страница стола

На рисунке 18 изображено Следующее это кнопка забронировать на странице который выбран, после нажатия кнопки забронировать всплывает модальное окно, где нужно заполнить форму для того чтобы забронировать столик на вечер или день по указанному времени соответственно.

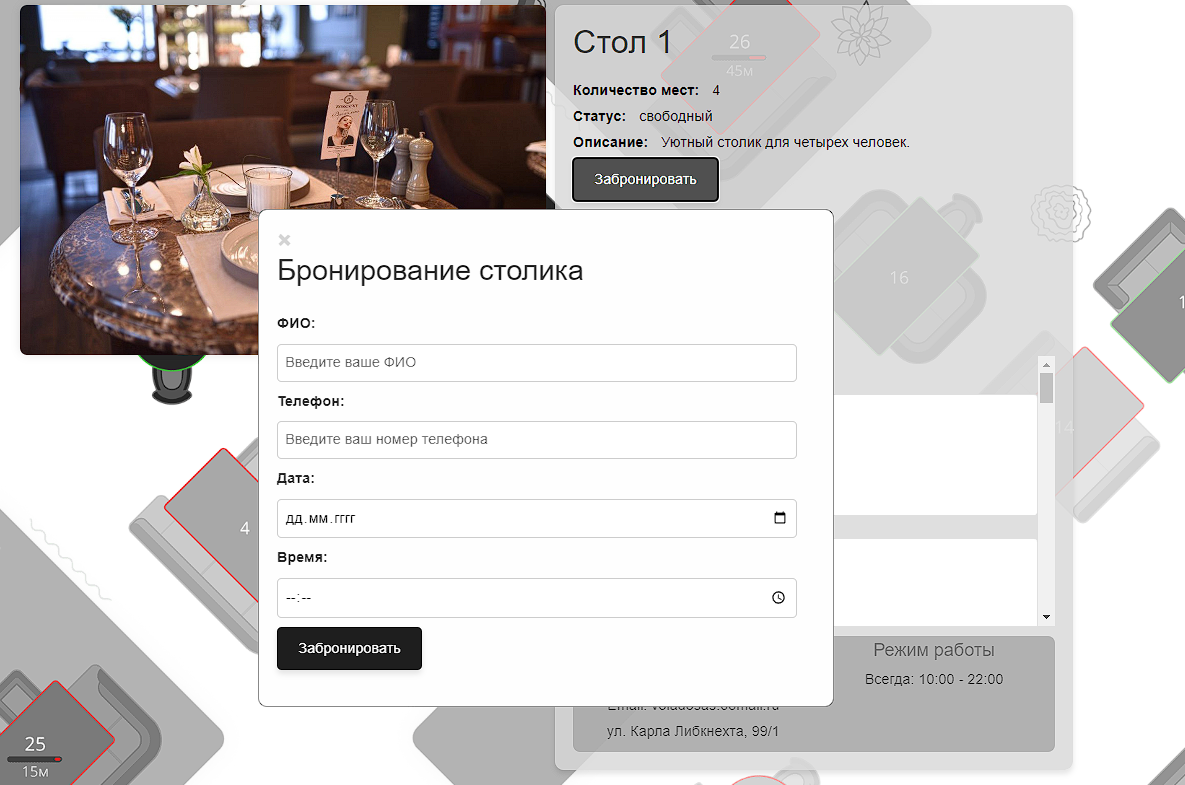


Рисунок 18 Форма заполнения для бронирования столика

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 отзывах и столах.

На рисунке 19 представлена страница, показывающая все столы, которые есть в базе данных



Рисунок 19. – Страница админ панели.

Обзор Отзывов:

На рисунке 20 представлена страница админ-панели, на которой отображается таблица со всеми отзывами, включая их данные.

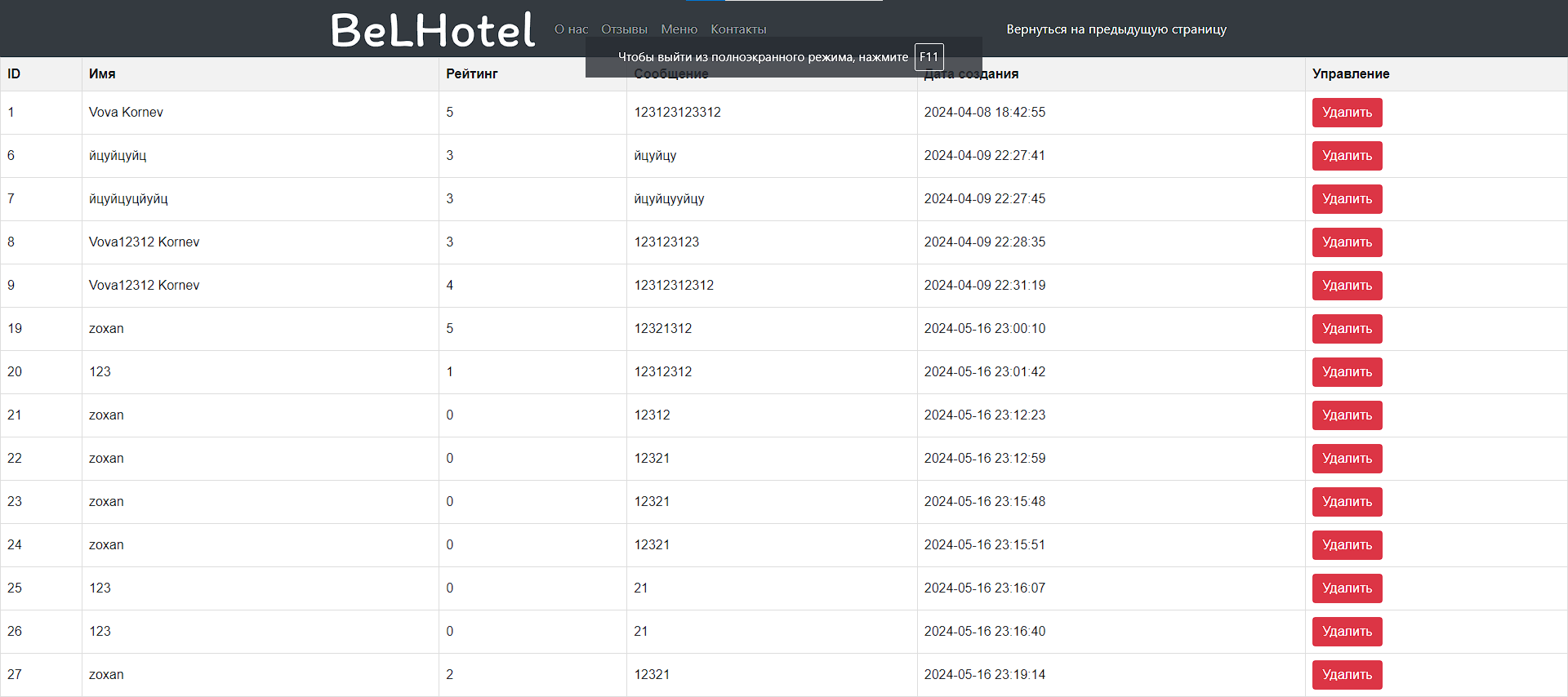


Рисунок 20. – Таблица отзывов.

На рисунке 21 представлена представлена страница админ-панели, на которой отображается добавления стола, включая его данные.

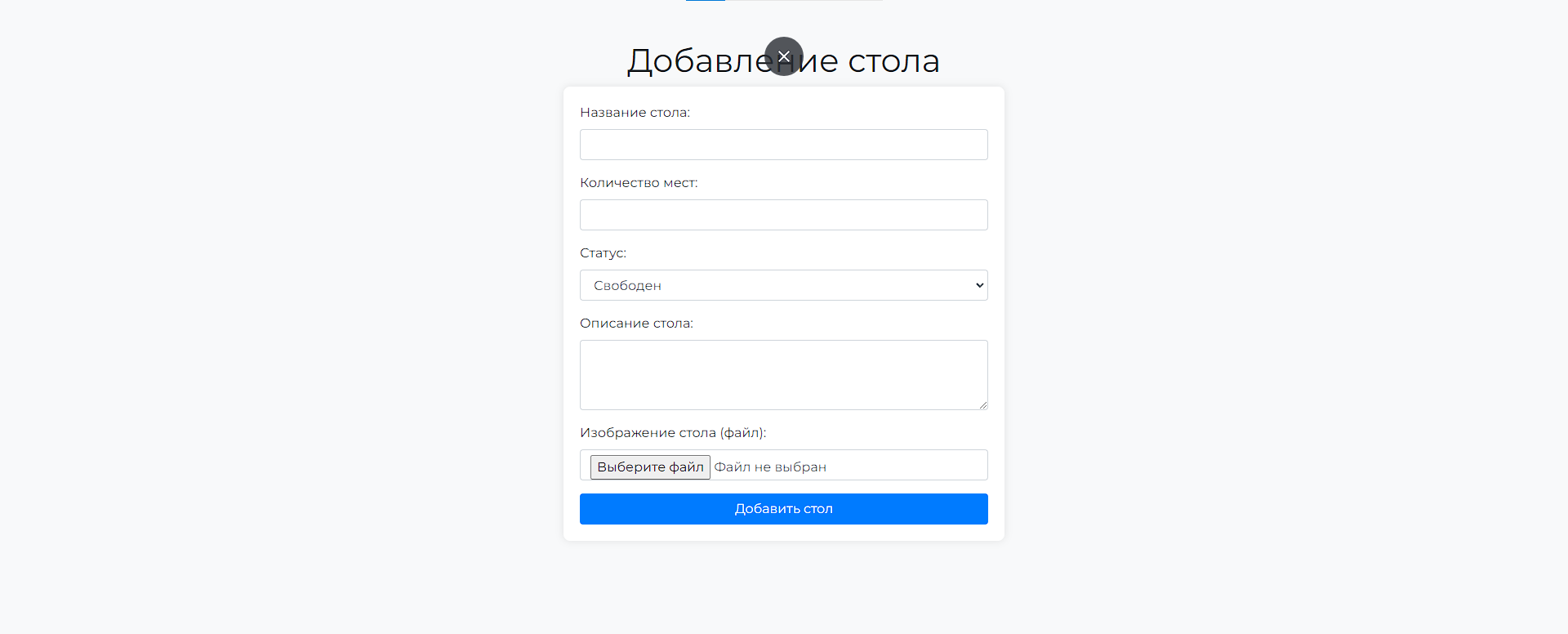


Рисунок 21. – Добавление стола.

ЗАКЛЮЧЕНИЕ

Современная степень развития информационных технологий открыла перед ресторанным бизнесом новые горизонты в управлении и организации работы, но при этом поставила и новые задачи. Бурное развитие информационных технологий, медленное, но неуклонное превращение компьютера и мобильных устройств в повседневную обыденность, появление Интернет и его широкое распространение – все это рано или поздно должно было затронуть и такую традиционно консервативную область, как ресторанный бизнес. В настоящее время, когда процесс создания таких систем уже вышел за рамки отдельных частных экспериментов, когда предпринимаются активные попытки внедрить их в бизнес-процессы ресторанов, и на этом пути уже накоплен некоторый опыт, можно, наконец, говорить о том, что определение самого термина "автоматизированная система бронирования столов" и его концепция, которую первопроходцы-энтузиасты нащупывали практически вслепую, начинает, наконец, проясняться.

В ходе работы были выполнены следующие задачи:

изучена предметная область и выявлены недостатки существующих методов бронирования столов в ресторане;

Была создана автоматизированная система бронирования столов, которая в дальнейшем будет использоваться рестораном для управления бронированиями и улучшения клиентского опыта. Система отвечает основным заявленным требованиям заказчика. Для разработки системы был выбран сервер под управлением ОС Linux, а программное обеспечение было написано с использованием таких технологий, как PHP, MySQL, HTML, CSS и JavaScript.

Основная цель выпускной квалификационной работы — разработка автоматизированной информационной системы бронирования выполнена.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Бейдсон, Кайл. JavaScript для профессиональных разработчиков. - М.: ДМК Пресс, 2021. - 800 с.
2. Вандюк, Джон К.. CMS Drupal. Руководство по разработке системы управления сайтом / Вандюк, Джон К., Мэтт Вестгейт. - М.: Вильямс, 2019. - 400 с.
3. Велтихов, А.Ю.. Node.js в действии. - М.: О’Рейли, 2022. - 512 с.
4. Вильямс, Робин. HTML и CSS: Учебное пособие для начинающих. - М.: Эксмо, 2020. - 320 с.
5. Гаевский, А.Ю.. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. - М.: Триумф, 2020. - 464 с.
6. Гейдж, Дэвид. Современные технологии веб-разработки. - М.: Мир, 2021. - 390 с.
7. Дакетт, Джон. HTML и CSS. Разработка и дизайн веб-сайтов / Джон Дакетт. - Москва: Наука, 2019. - 480 с.
8. Дакетт, Джон. JavaScript и jQuery. Интерактивная веб-разработка / Джон Дакетт. - М.: Наука, 2021. - 640 с.
9. Дронов, В.. JavaScript в Web-дизайне / В. Дронов. - М.: СПб: БХВ, 2019. - 880 с.
10. Дронов, В.. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов / В. Дронов. - М.: БХВ-Петербург, 2019. - 480 с.
11. Дронов, В.А.. PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов / В.А. Дронов. - М.: БХВ-Петербург, 2019. - 820 с.
12. Дронов, Владимир. JavaScript и AJAX в Web-дизайне / Владимир Дронов. - Москва: Высшая школа, 2019. - 736 с.
13. Дронов, Владимир. PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных Web-сайтов / Владимир Дронов. - М.: БХВ-Петербург, 2019. - 448 с.
14. Колисниченко, Д.. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д. Колисниченко. - М.: БХВ-Петербург, 2021. - 560 с.
15. Коршунов, А.В.. Vue.js в действии. - М.: ДМК Пресс, 2021. - 350 с.
16. Кэртер, Мэтью. Глубокое изучение HTML5. - М.: Эксмо, 2022. - 416 с.
17. Ли, Джейсон. Основы CSS Grid Layout. - М.: Эксмо, 2020. - 240 с.
18. Мейер, Эрик. CSS: Практическое руководство. - М.: О’Рейли, 2021. - 300 с.
19. Павлов, И.В.. Реактивное программирование с React. - М.: Наука, 2021. - 540 с.
20. Рейли, Майк. Основы разработки веб-приложений на Python и Flask. - М.: ДМК Пресс, 2022. - 288 с.
21. Робинсон, Майк. Полное руководство по PHP 7. - М.: Наука, 2020. - 480 с.
22. Смит, Билл. Разработка на JavaScript для новичков. - М.: ДМК Пресс, 2021. - 300 с.
23. Торнтон, Бен. Angular: Современные веб-приложения. - М.: Вильямс, 2022. - 360 с.
24. Фримен, Эрик. HTML5: Карманный справочник. - М.: О’Рейли, 2020. - 216 с.
25. Шмидт, Роберт. Разработка веб-сайтов с использованием PHP и MySQL. - М.: ДМК Пресс, 2021. - 512 с.

# 

# ПРИЛОЖЕНИЕ