

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(**Н И У « Б е л Г У »**)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ

РАЗРАБОТКА WEB ПРИЛОЖЕНИЯ БРОНИРОВАНИЯ МЕСТ В КАФЕ

Выпускная квалификационная работа
обучающегося по направлению подготовки 02.03.02 Фундаментальная
информатика и информационные технологии
заочной формы обучения, группы 07001360
Рябинина Эдуарда Михайловича

Научный руководитель
к.т.н., доцент,
Михелев В.М.

БЕЛГОРОД 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
Глава 1. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЙ ОБЩЕСТВЕННОГО ПИТАНИЯ	6
1.1 Информационные технологии и автоматизация кафе	6
1.2 Анализ процесса онлайн бронирования мест в кафе	10
1.3 Анализ программных средств бронирования мест в кафе	12
1.4. Анализ требований к веб-приложению	16
ГЛАВА 2. ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ РАЛИЗАЦИИ ВЕБ- ПРИЛОЖЕНИЯ ДЛЯ БРОНИРОВАНИЯ МЕСТ В КАФЕ	21
2.1. Архитектура веб-приложения.....	21
2.2. Выбор технологий реализации веб-приложения	25
2.3. Выбор инструментов реализации веб-приложения.....	30
ГЛАВА 3. РАЗРАБОТКА И РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ БРОНИРОВАНИЯ МЕСТ В КАФЕ	37
3.1. Разработка базы данных веб-приложения.....	37
3.2. Порядок разработки веб-приложения.....	42
3.3. Тестирование веб-приложения	52
3.4. Внедрение веб-приложения в деятельность кафе	56
ЗАКЛЮЧЕНИЕ	60
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	62
ПРИЛОЖЕНИЕ А	65
ПРИЛОЖЕНИЕ Б.....	74
ПРИЛОЖЕНИЕ В	84

ВВЕДЕНИЕ

Информационные технологии плотно проникают в нашу жизнь. Уже сложно найти сферу человеческой деятельности, в которой не задействованы вычислительные системы.

Использование вычислительной техники и Интернет позволяет ускорить работу, связанную с документацией, сократить обязанности обслуживающего персонала, а значит увеличить производительность труда.

Сегодня, когда мы приходим в ресторан или в кафе, мы видим, что возле барной стойки стоит компьютер, на котором официанты вводят заказ, который впоследствии передается поварам. Такая система очень удобна и позволяет экономить рабочее время официантов и сократить простои поваров. Преобразившие нашу жизнь информационные технологии внесли новый креатив и инновации в ресторанный бизнес. Сегодня и туристы, и жители городов отдадут предпочтение ресторану, кафе или бару, в котором есть бесплатный Wi-Fi: в ожидании заказа можно поработать, проверить почту, почитать новости или пообщаться с друзьями и близкими.

Предприятия, занятые в сфере общественного питания (кафе, рестораны, бары, пиццерии и т.д.) зачастую значительную часть прибыли теряют ввиду «человеческого фактора», другими словами, из-за неоптимального распределения мест при бронировании или посадке посетителей.

В целом автоматизация задачи по бронированию мест выгодна для предприятия ввиду очевидного удобства пользователей как будущих посетителей. Другими словами, функция онлайн бронирования столиков полезна и для клиента, и для самого предприятия. Администрация увеличивает оперативность работы персонала, расширяет область деятельности, ограниченную площадью заведения, привлекает новых посетителей и повышает заполняемость, снижает вероятность возникновения

ситуации, когда свободных мест в заведении нет. В таком случае, клиенту для оформления заявки необходимо на сайте кафе заполнить бланк с такими полями как: имя и фамилия, чтобы по прибытию в ресторан с идентификацией не возникло никаких проблем; номер телефона; количество персон, чтобы подобрать подходящий столик. Все это благоприятно повлияет на имидж предприятия и уровень лояльности посетителей.

Автоматизация бронирования позволит достигнуть повышения эффективности работы предприятия в целом за счет увеличения количества обслуживаемых клиентов (как следствие выпуска продукции), а также позволит улучшить значения показателей качества хранения, передачи и обработки информации, т.е. сократить время обработки и получения оперативных данных для принятия управленческих решений.

Объект исследования: бронирование мест в кафе.

Предмет исследования: веб-программирование и бронировка мест в кафе.

Цель работы: разработка и реализация веб-приложения для бронирования столиков и мест в кафе, которое в дальнейшем может быть интегрировано в веб-сайт предприятия общественного питания как отдельная веб-страница.

В соответствии с целью выделены следующие задачи:

1. Анализ предметной области.
2. Проектирование веб-приложения для бронирования мест в кафе.
3. Разработка, тестирование и отладка веб-приложения для бронирования мест в кафе.

В ходе выполнения дипломной работы были использованы следующие методы исследования: анализ, обобщение, методы математической обработки, методы представления информации графическим способом, компьютерное моделирование и др. При этом использовалась специальная и техническая литература, справочные руководства, техническая

документация, электронные источники информации, а также данные, полученные с помощью прикладного программного обеспечения и т.д.

ГЛАВА 1. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЙ ОБЩЕСТВЕННОГО ПИТАНИЯ

1.1 Информационные технологии и автоматизация кафе

Кафе представляет собой заведение, предназначенное для общественного питания и отдыха, которое напоминает ресторан небольших размеров, но кафе отличает ограниченный и более простой в приготовлении ассортимент продукции наряду с возможностью самообслуживания. Сейчас четкой градации между рестораном и кафе практически не существует.

Кафе в рабочих районах, где в обеденный перерыв можно покушать, зачастую называют закусочными. По месторасположению кафе бывают стационарными и уличными, расположенными в отдельных зданиях или пристройках. Другим видом кафе являются придорожные, которые расположены в отдельных зданиях вдоль дорог. В зависимости от целевых посетителей кафе может быть молодежным, детским, арткафе и другими. В зависимости от ассортимента реализуемой продукции кафе делятся на кофейни, кафе-пекарни (кондитерские), кафе-бары, кафе-мороженое, кафе-гриль и Интернет-кафе. Среди специализированных видов кафе выделяются кофейня и чайная, в которых реализация продукции поставлена на отдельных категориях продукции – в данном случае кофе и кофейные напитки, а также чай. Широко распространены сезонные кафе, например, у морского побережья, работающие в теплый период.

Одной из отраслей сферы услуг, где необходимо применение новых технологий, является ресторанный бизнес. Предприятия, которые несколько лет используют в своем ресторанном бизнесе информационные технологии, получили конкурентные преимущества. В данном случае, речь идет о следующих информационных технологиях: веб-сайт или приложение, беспроводной доступ к Интернет, электронное меню, система бронирования, система веб-мониторинга и т.д.

Наличие веб-сайта (его мобильной версии) или отдельного приложения кафе в Интернет позволяет пользователям просмотреть информацию о предприятии, произвести заказ блюд и их оплату с помощью электронных платежных систем.

Использование бесплатных публичных точек беспроводного доступа типа Wi-Fi для подключения к сети Интернет позволяют клиенту воспользоваться возможностями сети Интернет во время ожидания заказа: проверить электронную почту, пообщаться в социальной сети, по Skype и т.д.

Интерактивное электронное меню (интегрированное с системой управления предприятием) позволяет вести прием заказов посредством мобильных устройств (например, планшета) или большого сенсорного стола. Клиент выбирает блюда, перетаскивая их из меню в поле заказа, просматривает калорийность блюд, цену, общую сумму. Если что-то не устраивает, можно легко удалить и выбрать что-то другое до подачи заказа. После подтверждения своего выбора, информация о заказе передается к месту приготовления. Имеется возможность поиграть в игры, посмотреть новости и т.д.

Система бронирования столиков и мест полезна клиенту и предприятию. Ранее до полного вступления в силу «антибачного» закона посетитель мог выбрать столик в зоне для курящих или некурящих. Система резервирования столов дает такие возможности как просмотр плана зала, управление очередью, отправка СМС-уведомлений, расчет продолжительности занятости столика и т.д. Учитывая ускоренный темп жизни, многие посетители остаются недовольны качеством оказываемых услуг, а порой и не оказываемых, в случае отсутствия свободных мест в кафе.

Система веб-мониторинга помогает регистрировать несоблюдение санитарных норм и правил безопасности, фиксация нарушений в работе сотрудника, например, когда персонал работает без перчаток или головных уборов при обработке пищи.

Использование QR-кодов (штрих-кода) открывает неограниченные

возможности для онлайн взаимодействия кафе и посетителя. Скорость распознавания кода высока, его можно размещать на чеках, меню, вывесках. С помощью кода кафе может оповещать клиентов об акциях, розыгрышах, активизировать программы лояльности, устраивать голосования, опросы и получать отзывы от клиентов. Посетитель, отсканировав код и внося в календарь телефона информацию о заинтересовавшем событии или встрече, уже никогда о них не забудет.

Помимо рассмотренных информационных технологий в деятельности кафе также выделяют: структурированные кабельные системы, телефонию, телевидение, учетные приложения (POS, ERP: калькуляционные, складские, бухгалтерские, финансовые), вычислительные средства (сервера, станции, терминалы), видеонаблюдение, эквайринг (прием к оплате банковских карт в т.ч. посредством Интернет).

Сегодня отсутствие информационных технологий и автоматизированных систем управления в работе кафе приводит к убыткам, поскольку уровень и качество обслуживания, оперативный учет продукции и контроль деятельности сотрудников получают низкую оценку.

Система автоматизации управления рестораном должна решать ряд задач: учет товаров и материалов; учет производство блюд и полуфабрикатов; расчет себестоимости производства; складской учет; розничное ценообразование; работа с меню; учет продаж; учет взаиморасчетов; бухгалтерский и налоговый учет. Сегодня простых функций автоматического учета и контроля недостаточно. Система должна содержать ряд готовых бизнес-процессов, необходимых для эффективного и результативного управления. Сегодня все чаще возникает необходимость в комплексном управлении всеми ресурсами кафе.

Аппаратное (такое как сенсорные POS-терминалы, фискальные регистраторы, принтеры сервис-печати, программируемые клавиатуры, сканеры штрих-кода, считыватели карт, электронные весы, авторизаторы безналичных платежей) и программное обеспечение (типа 1С:Ресторан,

ТРАКТИРЬ) для автоматизации кафе оказывают реальную помощь персоналу предприятия питания: технологам, поварам, заведующим производством, калькуляторам, кладовщикам, бухгалтерам, отвечающими за различные участки учета, кассирам и официантам.

Процесс обслуживания посетителей в автоматизированном кафе упрощенно выглядит следующим образом. Официант в зале принимает заказ клиента и набирает его на терминале. После заказа (блюда и фамилия официанта) отправляется на кухню, где распечатывается на принтере или появляется на мониторе. Работники готовят блюда, а после снимают заказ из очереди. Официант доставляет заказ и ожидает выбытия клиента, параллельно обслуживая других посетителей. Когда подходит момент платы по счету, официант распечатывает предварительный счет и передает его клиентам. Клиент расплачивается (наличными или картой). Официант проводит оплату по карте или вносит наличные в кассу, после чего закрывает заказ. Клиент получает кассовый чек. После закрытия заказа система учета автоматически списывает израсходованную продукцию.

Автоматизация различных направлений в работе кафе страхует от рисков, связанных с махинациями со стороны сотрудников, учетом складских запасов и т.д., сокращает время ожидания, контролирует процесс подачи блюд, снижает нагрузку на бухгалтерию. Автоматизация позволяет создавать технологические карты и калькуляции, подробные отчеты по разным параметрам (финансовые, складские, производственные). Руководство кафе может провести аналитику и оценить работу персонала и получаемую прибыль.

В текущих рыночных условиях уже невозможно представить себе предприятие питания без современного оборудования и специального отраслевого программного обеспечения, учитывающего все особенности ресторанного бизнеса.

1.2 Анализ процесса онлайн бронирования мест в кафе

Онлайн-бронирование – это процесс размещения и подтверждения заявки на услугу или ресурс в интерактивном режиме через Интернет. Термин применяется по отношению к прокату автомобилей, гостиничным номерам, билетам (транспортным, театральным и т.д.), столам, местам и т.д.

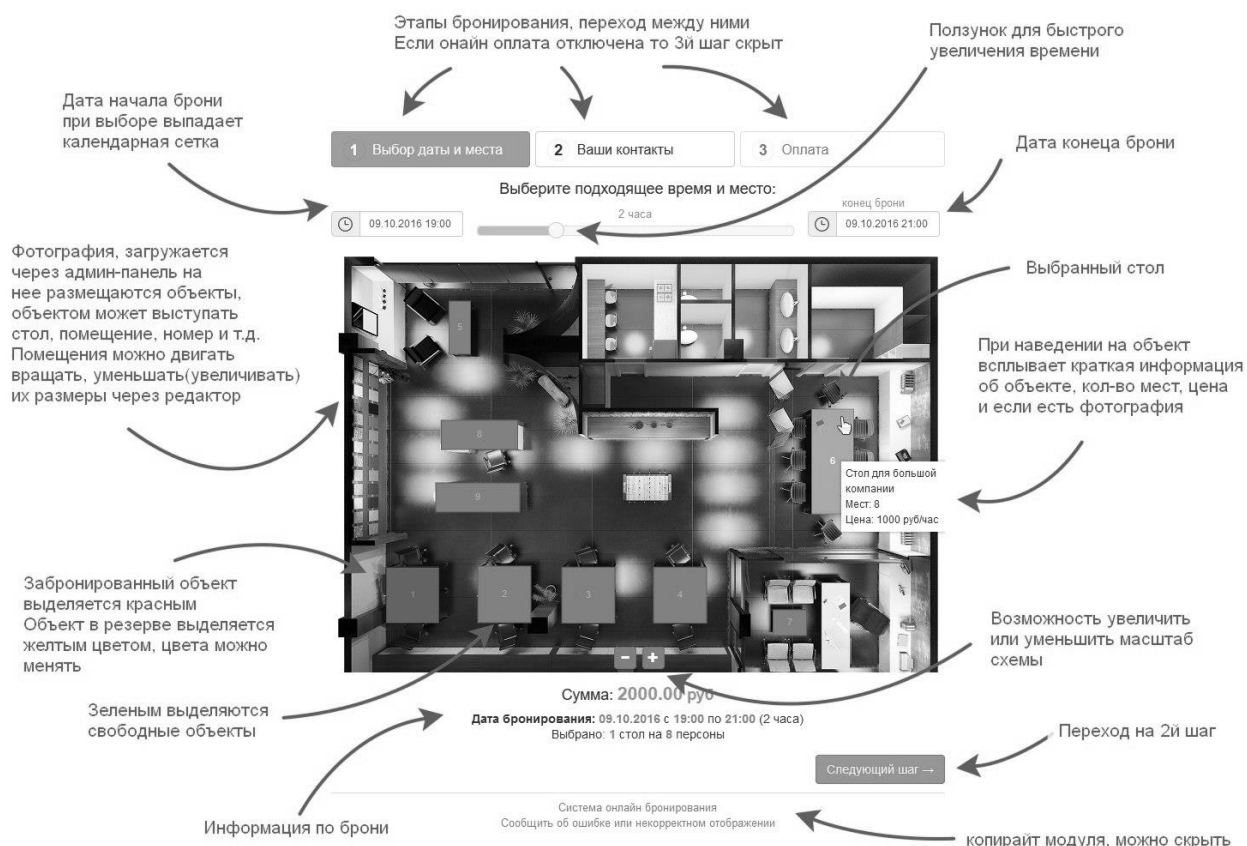


Рис. 1. 1. Графический режим модуля онлайн-бронирования столика со стороны пользователя (сервис m-bron.ru)

Общие принципы онлайн-бронирования:

1. Поиск доступных предложений. Определение пользователем онлайн параметров бронирования. Например, для бронирования столика характерны критерии: дата и время, продолжительность, расположение, количество мест.

2. Заполнение формы с контактной и платежной информацией: ФИО, телефон, эл. почта, способ оплаты (тип карты, номер, держатель, ключ).

3. Выполнение платежа. Обычно при платеже средства блокируются на карте пользователя, а при подтверждении правильности всех данных, списываются.

4. Получение подтверждения бронирования. Клиент получает документ или сообщение об успешном завершении бронирования, что гарантирует ему оказание услуги.

Система онлайн-бронирования столиков представляет веб-приложение, позволяющее пользователю увидеть текущее состояние забронированных и свободных столов, определиться с выбором стола, заказать блюда из меню ко времени прихода, рисунок 1.

Онлайн-бронирование призвано сэкономить время посетителей кафе до нескольких минут по сравнению с традиционным бронированием, когда требуется выполнить поиск телефона предприятия, дозвониться до администратора и переговорить с ним.

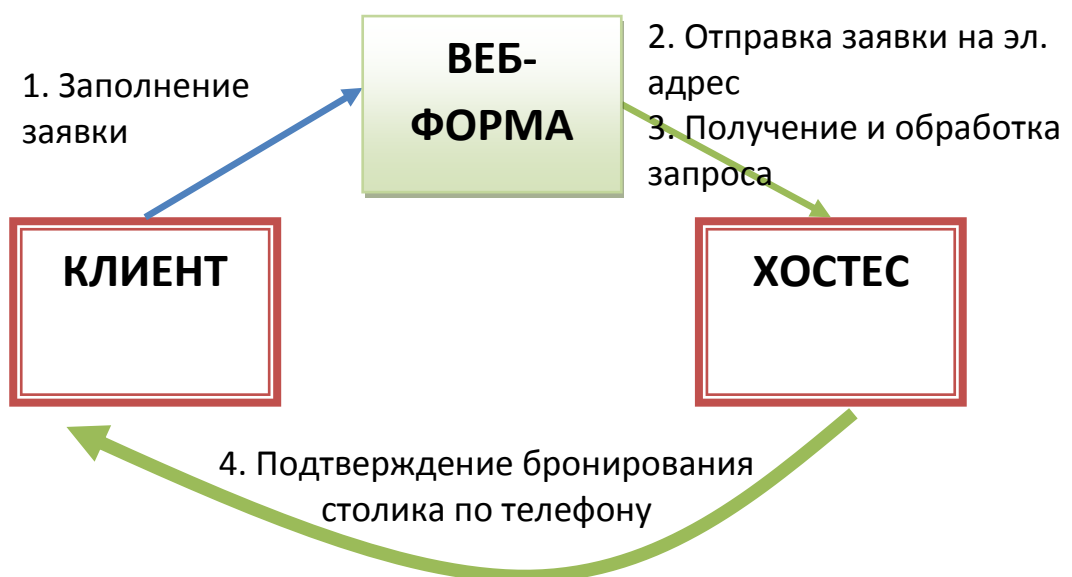


Рис. 1.2. Пример «бронирования» столика в кафе

В Интернет популярно два вида бронирования столиков:

1. Истинное онлайн-бронирование. Пользователь видит схему расположения столов в зале, выбирает понравившийся, выполняет заказ блюд, оплачивает и получает подтверждение брони. Все действия

совершаются на только сайте.

2. Псевдоонлайн-бронирование («заявка на бронь»), рисунок 2. Пользователь заполняет форму заявки на сайте и она отправляется на электронный адрес администратора ресторана.

Как правило, пользователь не уверен в том, что отправленная им заявка принята, обработана, и стол поставлен в резерв. По этой причине многие не пользуется второй формой бронирования столиков.

В данной работе требуется выполнить разработку веб-приложения максимально приближенному к первому типу бронирования – «истинному».

1.3 Анализ программных средств бронирования мест в кафе

В зависимости от решаемых задач система онлайн-бронирования может интегрироваться с существующим сайтом предприятия и взаимодействовать с автоматизированной системой управления, которая используется в заведении.

Кафе для внедрения в практику своей работы онлайн-бронирования может использовать один из трех вариантов: использовать веб-сервис, подключить готовый модуль на веб-сайт, заказать разработку собственной системы онлайн бронирования.

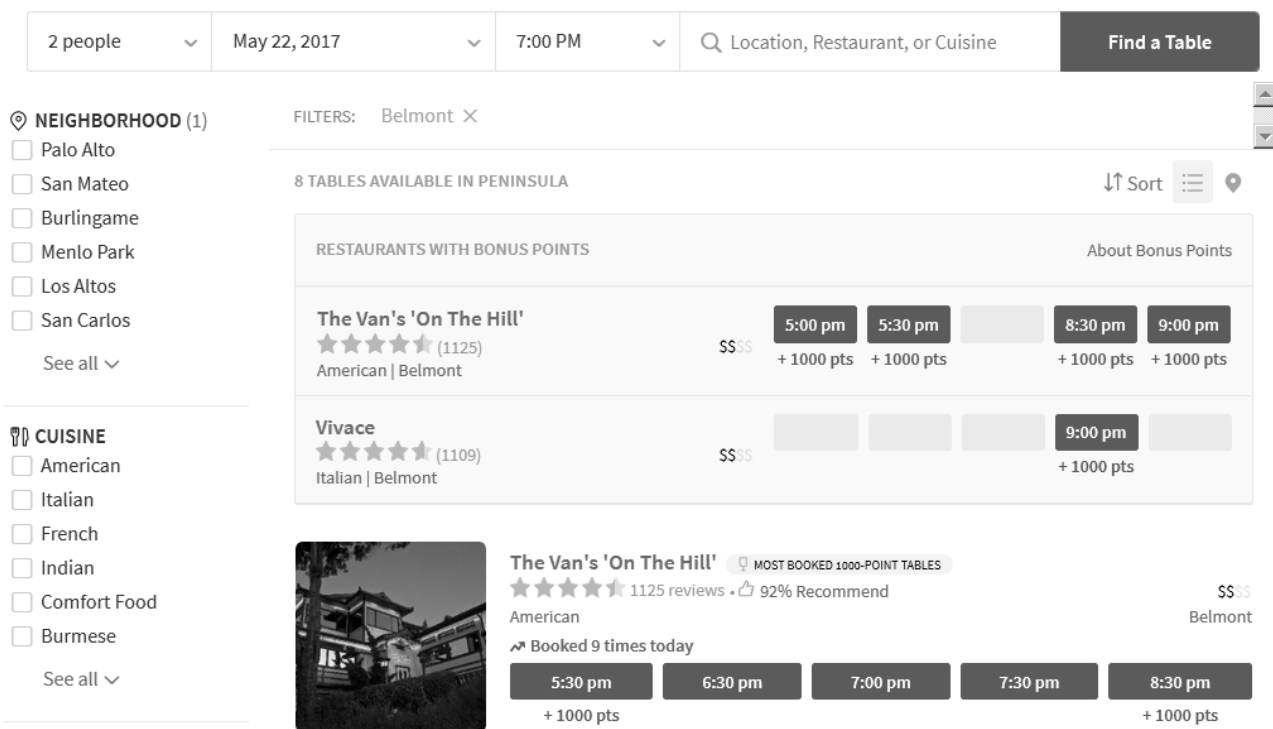


Рис. 1.3. Бронирование мест в ресторане по местоположению на opentable.com

Задача веб-сервиса – собирать пользователей, рассказывать им о заведениях в городе и предложить забронировать столик с их сайта. Сервис может просить у кафе абонентскую плату или скидку для гостей, которые пришли от них. Сервисы самостоятельно разрабатывают мобильные приложения, в которых клиентам удобно выбрать заведение и сделать бронь. Кафе получает уведомление о новой брони и работает с ним.

Известной международной системой онлайн-бронирования столов является OpenTable (более 28 тысяч заведений), рисунок 3. В России известны системы: Stoliki (более 35 тысяч заведений), Z-o-n (826 ресторанов), GetTable (407 ресторанов), LeClick (более 4000 ресторанов). У каждого сервиса свои преимущества, которые выделяют его на фоне конкурентов. Например, у OpenTable большое число заведений, информативность и наглядность.

Сервис gettable.ru, рисунок 4. В системе 813 заведений Москвы, 545 Петербурга, 13 в Екатеринбурге. Пользователю предлагают выбрать

заведение по названию или расположению и забронировать столик. Сервис помогает выбирать по большому числу фильтров: кухне, цене, популярности и т.д. Делает подборки новых и лучших заведений на свой выбор.

Закажите лучший столик

Сегодня, 22 мая

11:00 4 чел.

Как вас зовут?

+7 () - - -

Ваши пожелания к брони

ЗАБРОНИРОВАТЬ

- ★ Это бесплатно
- ★ Лучший столик
- ★ Без звонков за 2 минуты

8 499 649-78-49

Рис. 1.4. Форма бронирования мест в ресторане по местоположению на gettable.ru

Сервис stoliki.ru. У сервиса широкий охват, столики можно бронировать более чем в 100 городах России, крупных городах Казахстана и в Тбилиси. Работают с ресторанами по подписке. Есть личный кабинет администратора, которые собирает контакты пользователей в базу, анализирует заполняемость зала. Есть возможность принимать онлайн платежи.

Второй вариант – подключить готовый модуль онлайн-бронирования на собственный сайт. Предприятие покупает модуль у разработчика, самостоятельно привлекают аудиторию на форму бронирования, через поисковые системы и социальные сети. Предприятие самостоятельно решает, на каких условиях осуществлять бронирование, управляет бронью. Такие модули есть и у систем автоматизации ресторана: R-keeper, Jowi и др.

Stoliki от hesus.ru – это простой и удобный модуль онлайн-бронирования столиков для ресторанов и кафе. Наглядный интерфейс модуля: интерактивный календарь, который отражает сведения по занятым-свободным столикам на месяцы вперед. Простое бронирование столиков: выбрать свободный день, свободный столик и указать контактную информацию. Онлайн-оплата бронирования.

Плагин ReDi Restaurant Reservation – решение для сайтов на WordPress. Помогает управлять бронированием в заведениях. Показывает клиентам свободные места и время с учетом общего количества мест в зале. Хостесс может следить за количеством заявок. Можно разрешать или запрещать бронирование на конкретные дни или часы. Подтверждение брони отправляется клиентам на почту. Ежемесячная оплата за каждую установку.

Третий вариант – самостоятельно разработать модуль бронирования для сайта или мобильного приложения. Основное преимущество в том, что предприятие может реализовать любые потребности своего бизнеса. Недостатки – время реализации и цена.

При использовании подобных систем бронирования заведение получает следующие преимущества:

1. Администратор управляет системой: открывает-закрывает возможность бронирования, выбирает схему расположения столов в залах, определяет сумму депозита, список дополнительных услуг и другие параметры.

2. Система автоматически отправляет СМС-сообщение администратору с информацией о новой брони.

3. Система автоматически сохраняет всю информацию внесенную клиентами при бронировании, позволяет получать статистические данные и вести клиентскую базу.

4. Система постоянно работает и доступна для пользователей.

Системы автоматизации ресторанного бизнеса и бронирования мест станут еще более функциональными, а число использующих их заведений

будет расти. В них будет реализована и функциональность для решения связанных задач – например, бронирование столиков и сбор отзывов посетителей. Кроме того, появятся и решения по автоматизации и внедрение дополнительного контроля новых аспектов функционирования предприятия – например, скорости реагирования на нажатие кнопки вызова официанта.

1.4. Анализ требований к веб-приложению

Разработка программного обеспечения подчиняется определенному жизненному циклу. Существует множество подходов реализации жизненного цикла разработки (например, такие как SWOT, VCM, BPR, ISA и т.д.), но всех их объединяет наличие этапа определения требований к программной системе.

Недостаточный объем информации, поступающей от пользователей, требования, сформулированные не полностью, их кардинальное изменение после начала проектирования являются основными причинами, из-за которых нарушаются сроки разработки и рамки бюджета, чтобы предоставить пользователям полнофункциональный продукт.

Требования можно классифицировать по группам: требования к средствам обеспечения поддержки и функционирования (к платформе, аппаратным средствам, операционной системе, СУБД, безопасности), функциональные требования (к методам обработки данных, самим данным, подсистемам, совместимости с другими системами, к организации совместной работы), требования общего характера (к интерфейсу пользователя, к документации, затратам на обслуживание).

Выделены следующие ключевые требования к разрабатываемому программному обеспечению под названием «Table Reserver», предназначенному для бронирования столиков и мест на предприятии общественного питания (кафе).

Назначение программного обеспечения «Table Reserver» – выбор

интересующего свободного столика и бронирование необходимого числа мест из доступных за ним в режим онлайн, т.е. посредством веб-интерфейса (веб-страницы в браузере).

Требования к функциям программного обеспечения «Table Reserver» (системе бронирования столиков и мест в кафе):

1. Формирование администратором произвольного общего количества столиков в зале кафе.

2. Формирование администратором произвольного числа мест за каждым отдельным столиком в зале кафе.

3. Поддержка одновременной онлайн работы нескольких пользователей при бронировании столиков и мест за ними («блокирование» столика и мест за конкретным пользователем). «Блокирование» столика по умолчанию на три минуты, возможность изменения времени «блокирования» администратором.

4. Возможность выбора столика для бронирования пользователем.

5. Возможность выбора определенного числа мест за столиком для бронирования пользователем.

6. Просмотр схемы зала кафе пользователем.

7. Просмотр доступных столиков и занятых другими пользователями.

8. Просмотр доступных мест за каждым отдельным столиком.

Общий сценарий работы пользователя с программным обеспечением «Table Reserver»:

1. Пользователь заходит на веб-страницу.

2. Пользователь видит общее количество столиков доступных в кафе и уже занятые столики.

По каждому столику пользователь видит общее число свободных мест за столиком и цену бронирования места.

3. Пользователь выбирает интересующий его столик (кликает по нему).

Выполняется проверка доступности столика.

Приложение генерирует кнопки с числами, соответствующими

количеству бронируемых мест.

4. Пользователь выбирает число мест которое он бронирует (кликает по кнопке с числом бронируемых мест).

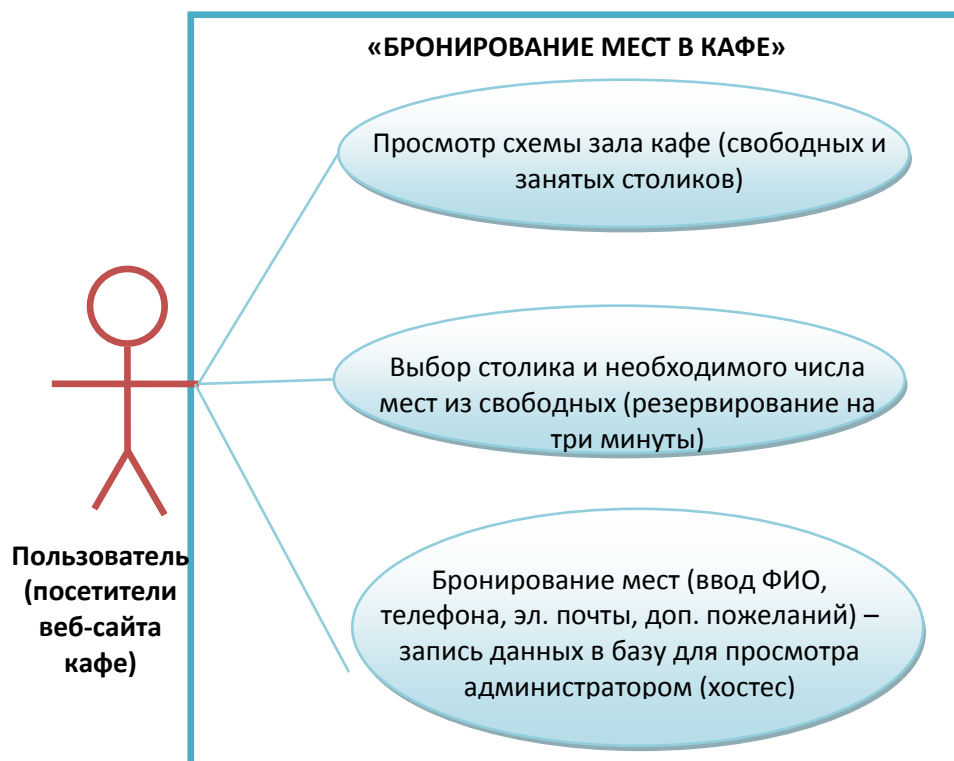


Рис. 1.5. Диаграмма прецедентов веб-приложения «Table Reserver»

Приложение «резервирует» за пользователем выбранный столик и количество указанных пользователем мест на три минуты. Другими словами, поддерживается одновременная работа множества пользователей с приложением.

Выполняется проверка доступности мест (блокировки другими пользователями).

5. Пользователь вводит свои данные и нажимает кнопку «Купить места» за выбранным столиком.

6. Администратор видит пользователя, забронированный им столик и места.

Администратор имеет возможность изменения времени резервирования мест за указанным пользователем столиком на другой временной период.

Администратор может вносить изменения в базу данных с целью

формирования определенного количества имеющихся в наличии столов, мест, стоимости.

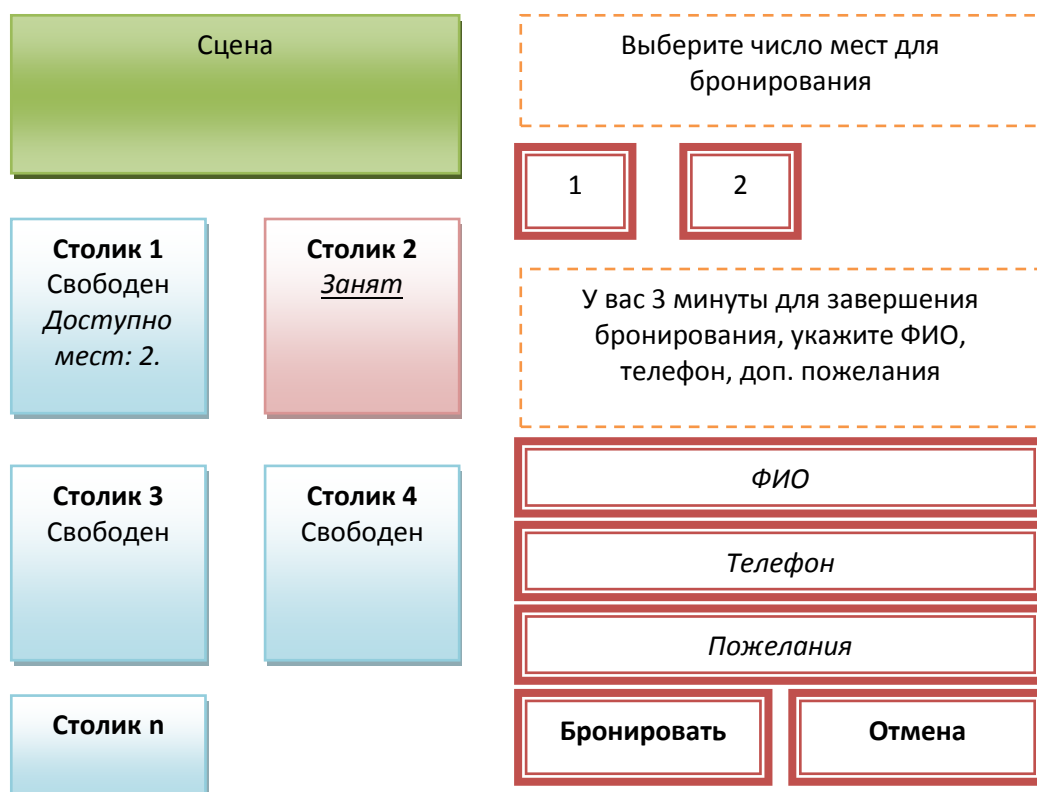


Рис. 1.6. Прототип пользовательского интерфейса веб-приложения «Table Reserver» (1 – сцена, 2 – необходимое число мест, 3 – регистрация)

Необходимо предусмотреть возможность интеграции программного обеспечения «Table Reserver» с веб-сайтом кафе и внесения изменений в оформление: изображений столиков, их позиций на веб-странице за счет файла стилей CSS и разделения кода на модули. Предусмотреть возможность подключения модуля внесения оплаты за бронирование типа МТС, QIWI, СБЕРБАНК, Яндекс.Деньги или Робокасса, защиты от спама при оформлении бронирования онлайн (поле защиты типа Google reCaptcha), внесения дополнительных пользовательских данных (полей типа email, телефон, WhatsApp, пожелания посетителя, дата и время прихода и т.д.). Также привлекательными стали бы возможности легкой смены ранее забронированного стола, изменения даты или отказа от брони, использования виртуального тура – 3D-панорамы для обзора зала с выбранного стола.

В будущем большим плюсом станет возможность интеграции функций бронирования с функциями предварительного оформления заказа блюд и напитков, согласно текущего меню кафе до прихода в заведение.

Таким образом, гость пройдет полный цикл бронирования и выбора дополнительных услуг в течение нескольких минут. Получит подтверждение о бронировании (на электронную почту, СМС-уведомление при интеграции с сервисом).

Исходя из требований к функционалу программного обеспечения «Table Reserver» можно сделать вывод, что приложение повысит удобство использования услуги бронирования стола в кафе, поскольку возможность бронирования стола и мест станет доступным в любое время суток, оптимизирует работу хостес (администратора), повысит привлекательность кафе со стороны клиентов, позволит создать и вести клиентскую базу кафе.

Программное обеспечение «Table Reserver» должно работать в современных популярных браузерах типа Mozilla FireFox, Google Chrome, Opera, Safari, Яндекс, Internet Explorer на настольных и мобильных аппаратных платформах под управлением операционных систем типа Windows, Linux, Apple, Android, iOS.

ГЛАВА 2. ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ РАЛИЗАЦИИ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ БРОНИРОВАНИЯ МЕСТ В КАФЕ

2.1. Архитектура веб-приложения

Исходя из проанализированных в предыдущем разделе требований (в т.ч. требования кроссплатформенности к работе приложения) можно заключить, что программное обеспечение «Table Reserver» будет представлять собой веб-приложение.

Веб-приложение – это клиент-серверное приложение, в котором клиентом выступает браузер, а сервером – веб-сервер. Клиентская часть реализует интерфейс, формирует запросы к серверу и обрабатывает ответы от него. Серверная часть получает запрос от клиента, выполняет вычисления, после формирует веб-страницу и отправляет её клиенту с использованием протокола HTTP. Таким образом, логика веб-приложения распределена между сервером и клиентом, хранение данных преимущественно осуществляется на сервере, обмен информацией происходит по Интернет.

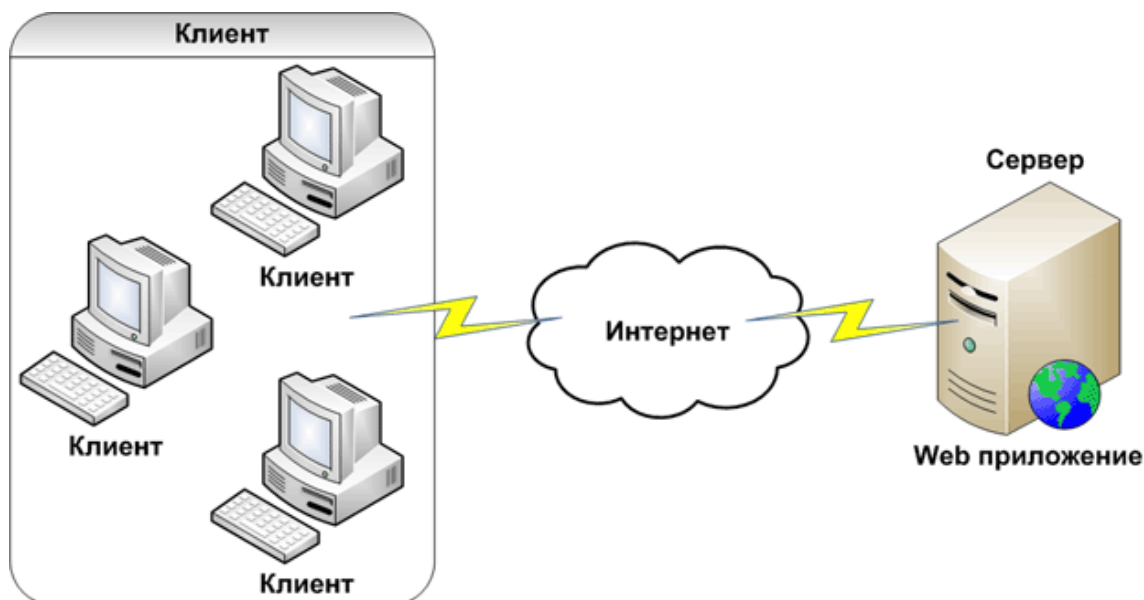


Рис. 2.7. Взаимодействие веб-приложения с пользователем

Другими словами файлы веб-приложения расположены и выполняются

на сервере, а клиент при этом получает только результат работы.

Основная функция любого браузера – отображение данных, полученных из Интернет в виде страницы, описанной на языке HTML, следовательно, результат, передаваемый сервером клиенту, должен быть представлен на этом языке. Ответ формируется на основе исполнения программного кода: веб-формы, страницы HTML, другого содержимого, включая графические файлы.

На стороне сервера веб-приложение выполняется специальным программным обеспечением – веб-сервером, который и принимает запросы клиентов, обрабатывает их, формирует HTML-ответ и передает его клиенту. Одним из веб-серверов является Internet Information Services компании Microsoft, единственный который поддерживает платформу ASP. NET, другим примером может служить Apache.

Получается, что результат работы веб-приложения идентичен результату запроса к традиционному веб-сайту, однако, в отличие от него, веб-приложение генерирует HTML-код в зависимости от запроса пользователя, а не просто передает его клиенту в том виде, в котором этот код хранится в файле на стороне сервера, т.е. ответ динамически формируется с помощью исполняемого кода (исполняемой части). За счет исполняемой части, веб-приложения способны выполнять практически те же операции, что и обычные «оконные» приложения.

К наиболее типичным операциям, выполняемым Web-приложениями, относятся: прием данных от пользователя и сохранение их на сервере; выполнение различных действий по запросу пользователя: извлечение данных из базы данных, добавление, удаление, изменение данных, проведение сложных вычислений; аутентификация пользователя и отображение интерфейса системы, соответствующего данному пользователю; отображение постоянно изменяющейся оперативной информации и т. д.

Существенное преимущество веб-приложения в том, что его функции выполняются независимо от операционной системы клиента (т.е. веб-

приложение является кроссплатформенным). Однако различная реализация HTML, CSS, DOM и других спецификаций в браузерах может вызвать проблемы при разработке и последующей поддержке веб-приложения. Кроме того, возможность пользователя настраивать такие параметры браузера как размер шрифта или отключение поддержки сценариев может препятствовать корректной работе веб-приложения. Другой менее универсальный подход заключается в использовании Adobe Flash, Silverlight или Java-апплетов для полной или частичной реализации пользовательского интерфейса.

На стороне клиента для реализации интерфейса используется: HTML, XHTML и CSS, а для формирования запросов, создания интерактивного и независимого от браузера интерфейса: ActiveX, Adobe Flash, Adobe Flex, Java, JavaScript и Silverlight.

Для создания веб-приложений на стороне сервера используются разнообразные технологии и любые языки программирования (например, PHP), способные осуществлять вывод в стандартную консоль.

В настоящее время набирает популярность новый подход к разработке веб-приложений, называемый AJAX. При использовании AJAX страницы веб-приложения не перезагружаются целиком, а лишь догружают необходимые данные с сервера, что делает их более интерактивными и производительными.

Формально в разрабатываемом веб-приложении «Table Reserver» можно выделить три взаимно независимые части:

1. Модуль, который исполняется веб-браузером. Приложение может быть написано на любом языке, который поддерживает браузер. В данном случае будет использован JavaScript как наиболее поддерживаемый и имеющий большую библиотечную поддержку, что важно и при экономии бюджета проекта.

2. Модуль, исполняемый на серверной стороне под управлением веб-сервера. Приложение может быть написано на любом языке (Java или PHP, который будет выбран в данном случае), интерпретацию которого

поддерживает выбранный разработчиком веб-сервер (OpenServer).

3. База данных. Здесь существует достаточно широкий выбор. Есть промышленные базы данных, такие как Oracle, DB2, PostgreSQL, но в данном случае будет достаточно «легкой» базы данных MySQL, выбранной на основании целей и области решаемых задач.

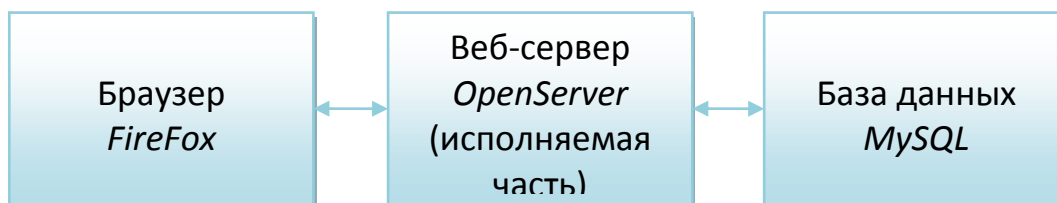


Рис. 2.8. Структурные единицы (модули) в архитектуре веб-приложения

Представленные структурные единицы порождают два вида связей: связь между браузером и серверной частью; связь между серверной частью и базой данных.

Для достижения максимальной независимости между структурными единицами, необходимо чтобы каждая единица оперировала только необходимым ей набором данных.

Поэтому для минимизации зависимостей между браузером и веб-сервером необходимо, чтобы язык разметки HTML был задействован только в браузере, а веб-сервер предоставлял интерфейс для получения необходимых данных для страницы. Данная модель достижима следующим образом: программа выполняемая браузером написана на JavaScript и общается с веб-сервером через AJAX, получая ответы в соответствии с определенным протоколом, т.е. достигается разделение программной части веб-сервера и браузера (это позволит вносить изменения в структурную единицу для браузера и не вызывать косвенных изменений в серверной части).

Взаимодействие базы данных и веб-сервера возможно организовать на основании двух принципиально разных сценариях: бизнес логика находится в базе данных; бизнес логика находится в коде веб-сервера.

В первом случае база данных хранит данные и предоставляет интерфейс доступа к данным: выборка данных – решается через представления; модификация данных – решается через хранимые процедуры. Программа для веб-сервера является драйвером для доступа к бизнес-логике, т.е. она просто связывает браузер с бизнес-логикой, которая реализована в базе данных.

Во втором случае база данных хранит данные, и предоставляет прямой доступ к данным. Бизнес-логика реализована в коде веб-сервера. В этом случае база данных предоставляет транзакции для проведения атомарных операций.

Для минимизации зависимостей между веб-сервером и базой данных, необходимо, чтобы бизнес-логика была определена только в одном месте. То либо в коде веб-сервера, либо в базе данных.

На основании вышеизложенного структура проекта примет следующий вид: модуль для браузера, для веб-сервера, для базы данных, протокол обмена между модулем браузер и веб-сервером, интерфейс взаимодействия между модулем браузера и веб-сервером, интерфейс взаимодействия между веб-сервером и базой данных.

2.2. Выбор технологий реализации веб-приложения

Исходный код программного обеспечения «Table Reserver» будет разрабатываться на веб-языках: HTML, CSS, PHP, MySQL, AJAX, JavaScript.

Язык HTML позволит выполнить создание структуры (макета) веб-страницы для бронирования столов.

Язык CSS позволит выполнить создание стилового оформления веб-страниц.

Язык PHP будет использоваться для написания основного кода, отвечающего в целом за работу приложения, обработку данных вводимых пользователем в форму, работу с сессиями, отправку запросов к базе данных,

вызов функций JavaScript и т.д.

Язык JavaScript и технология AJAX позволят выполнить загрузку данных в фоновом режиме для обеспечения работы приложения, обеспечит генерацию кнопок (свободных мест за столиком), обращение к элементам посредством объектной модели документа DOM, реализацию основных функций приложения

Язык MySQL будет использоваться для создания запросов на выборку, вставку, обновление, удаление данных из базы данных.

Для понимания тонкостей работы с каждым отдельным языком рекомендуется использовать удобные онлайн-справочники и среды разработки с готовыми шаблонами проектов (рисунок 7, 8).

Язык HTML (HyperText Markup Language) – стандартизированный язык разметки документов. Большинство веб-страниц содержат описание разметки на языке HTML. Язык интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране устройства. Язык HTML является приложением SGML и соответствует международному стандарту ISO 8879. Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием шифрования.

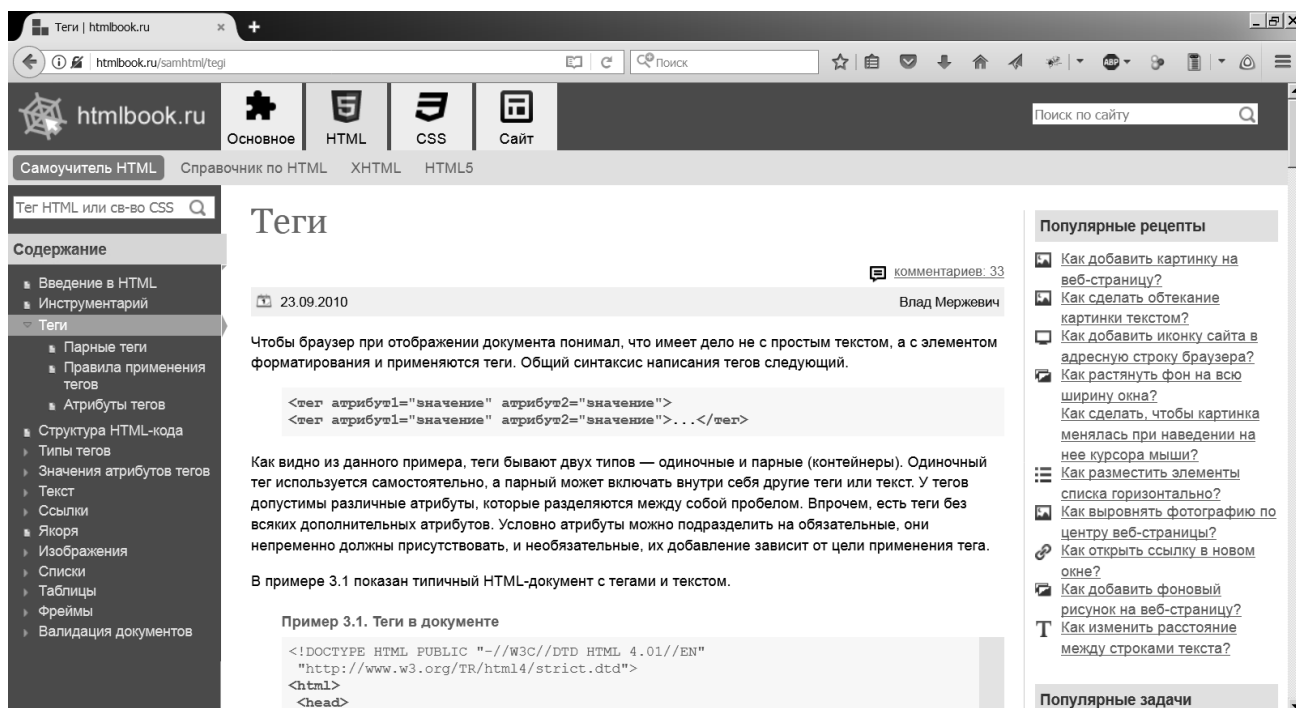


Рис. 2.9. Онлайн-справочник htmlbook по веб-языкам (HTML, CSS, JavaScript)

Язык CSS (Cascading Style Sheets) – формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG.

Язык CSS используется разработчиками веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц.

Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы от описания внешнего вида этой страницы. Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Язык позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное

представление, печатное представление, чтение голосом, или при выводе устройствами, использующими шрифт Брайля.

Язык PHP (PHP: Hypertext Preprocessor) – скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений.

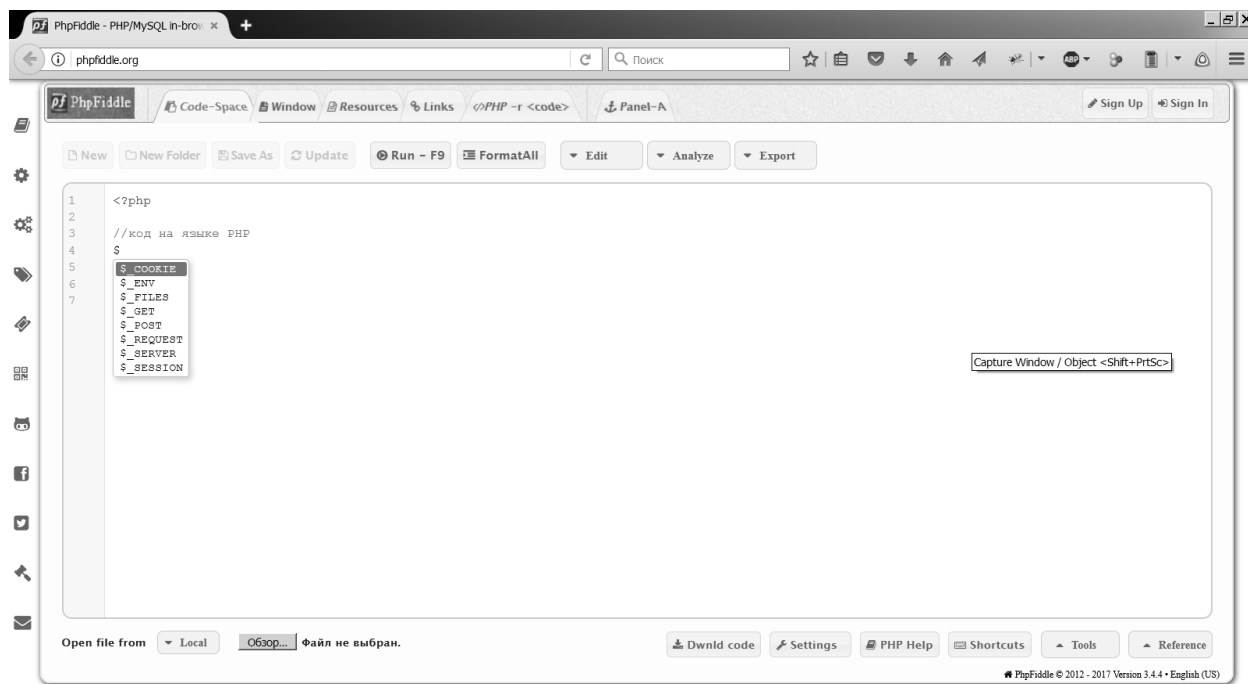


Рис. 2.10. Онлайн среда разработки (IDE) для работы с PHP

В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов. Язык PHP можно использовать и для создания GUI-приложений.

Синтаксис PHP подобен синтаксису языка Си. Язык PHP является языком программирования с динамической типизацией, не требующим указания типа при объявлении переменных, равно как и самого объявления переменных.

В PHP существуют суперглобальные массивы, имеющие глобальную область видимости без использования директивы `global`. Большая часть этих массивов содержит входные данные запроса пользователя (параметры GET-запроса, поля форм при отправке методом POST, cookie-файлы и т.п.). С пятой

версии PHP обладает полной поддержкой объектно-ориентированного программирования. Работа с классами оптимизирована и код работает достаточно быстро.

Язык JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript. Язык JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений.

Наиболее широкое применение JavaScriptë находят в браузерах как язык сценариев для придания интерактивности веб-страницам. Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса. Язык JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками.

Язык JavaScript используется в AJAX, популярном подходе к построению интерактивных пользовательских интерфейсов веб-приложений, заключающемся в «фоновом» асинхронном обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью и интерфейс веб-приложения становится быстрее, чем это происходит при традиционном подходе (без применения AJAX).

Язык AJAX (Asynchronous Javascript and XML) или асинхронный JavaScript и XML – это подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее. AJAX базируется на двух основных принципах: использование технологии динамического обращения к

серверу «на лету», без перезагрузки всей страницы полностью, например, с использованием XMLHttpRequest. Действия с интерфейсом преобразуются в операции с элементами DOM, с помощью которых обрабатываются данные, доступные пользователю, в результате чего представление их изменяется. Здесь же производится обработка перемещений и щелчков мышью, а также нажатий клавиш. Каскадные таблицы стилей CSS, обеспечивают согласованный внешний вид элементов приложения и упрощают обращение к DOM-объектам.

Система MySQL – свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle. MySQL является решением для малых и средних приложений. Входит в состав серверов OpenServer, WAMP и др.

Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удаленные клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость системы MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. MySQL имеет интерфейсы API для PHP, C++, Java, Python и др.

2.3. Выбор инструментов реализации веб-приложения

Для разработки системы бронирования выбраны следующие мощные и популярные приложения: PHPStorm, FireFox (с FireBug), OpenServer (с PhpMyAdmin), HeidiSQL, Adobe PhotoShop.

Рассмотрим возможности основных выбранных инструментов.

Мощная среда веб-разработки JetBrains PHPStorm для написания исходного кода приложения и работы с языками HTML, CSS, PHP, MySQL,

AJAX, JavaScript.

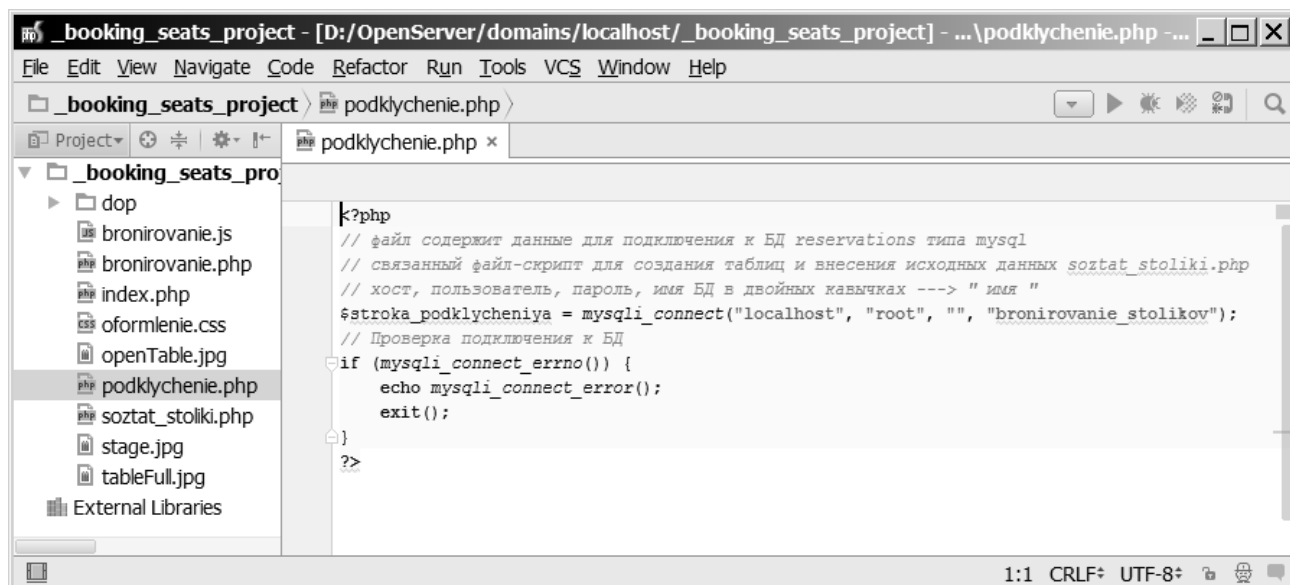


Рис. 2.11. Среда JetBrains PhpStorm (работа с проектом в IDE: файлом podklychenie.php для подключения к базе данных MySQL – bronirovanie_stolikov)

Среда PhpStorm компании JetBrains представляет коммерческую кросс-платформенную интегрированную среду разработки для PHP. Среда создана на основе платформы IntelliJ IDEA, написанной на Java. Пользователи могут расширить функциональность среды разработки за счет установки плагинов, разработанных для платформы IntelliJ, или написав собственные плагины. PhpStorm представляет собой интеллектуальный редактор для PHP, HTML и JavaScript с возможностями анализа кода на лету, предотвращения ошибок в коде и автоматизированными средствами рефакторинга для PHP и JavaScript. Автодополнение кода в PhpStorm поддерживает спецификацию PHP 5.3, 5.4, 5.5, 5.6, 7.0 и 7.1, включая генераторы, сопрограммы, пространства имен, замыкания, типаж и синтаксис коротких массивов. Имеется полноценный SQL-редактор с возможностью редактирования полученных результатов запросов.

Браузер Firefox с дополнением FireBug для тестирования работы приложения.

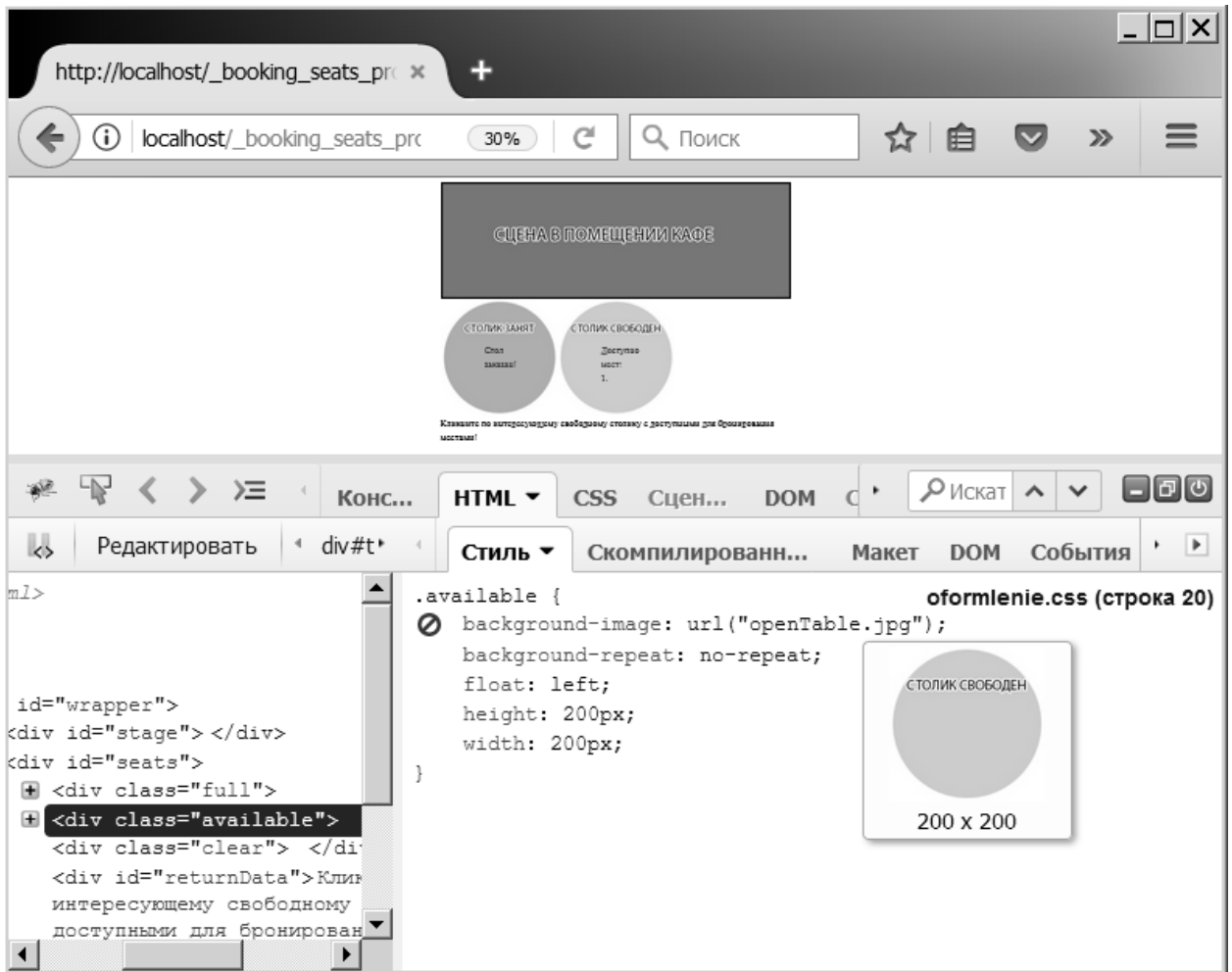


Рис. 2.12. Браузер FireFox с дополнением FireBug (инспектирование элементов веб-страницы – свойство background-image (фоновое изображение))

Браузер Firefox является свободным браузером на движке Gecko, разработкой и распространением которого занимается Mozilla Corporation. Firefox официально выпускается для Windows, macOS, GNU/Linux и Android. Доступны неофициальные сборки для FreeBSD, множества других UNIX-подобных операционных систем, а также BeOS. В браузере присутствует интерфейс со многими вкладками, проверка орфографии, поиск по мере набора, «живые закладки», менеджер загрузок, поле для обращения к поисковым системам.

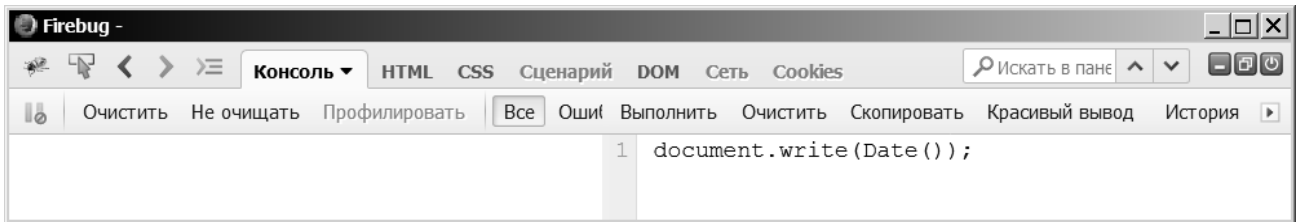


Рисунок 2.13. Дополнение FireBug, вкладка «Консоль» позволяет работать с кодом на JavaScript (вывод текущей даты и времени на странице)

Новые функции в браузер можно добавлять при помощи расширений, например, FireBug. Дополнение Firebug является отладчиком веб-приложений, используется как отдельное дополнение, являющееся консолью, отладчиком, и DOM-инспектором JavaScript, DHTML, CSS, XMLHttpRequest. Firebug показывает в консоли вызвавшую ошибку функцию, стек вызовов функций, вызвавших эту ошибку. Дополнение предупреждает, когда правило CSS или метод-свойство JavaScript не существует.

Локальный сервер OpenServer для запуска приложения на локальной машине и PHPMyAdmin для работы с базой данных MySQL.

Программная среда Open Server Panel является портативной серверной платформой, созданной для веб-разработчиков с учетом их рекомендаций и пожеланий. Программный комплекс имеет набор серверного программного обеспечения, многофункциональный интерфейс, обладает мощными возможностями по администрированию и настройке компонентов. Платформа используется с целью разработки, отладки и тестирования веб-проектов, а так же для предоставления веб-сервисов в локальных сетях.

В основе работы веб-сервера OpenServer лежит свободный веб-сервер Apache (HTTP-сервер) – кроссплатформенное программное обеспечение, поддерживающее операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS. Основными достоинствами Apache считаются надежность и гибкость конфигурации. Сервер позволяет подключать внешние модули для предоставления данных, использовать

системы управления базами данных для аутентификации пользователей, модифицировать сообщения об ошибках и т. д.

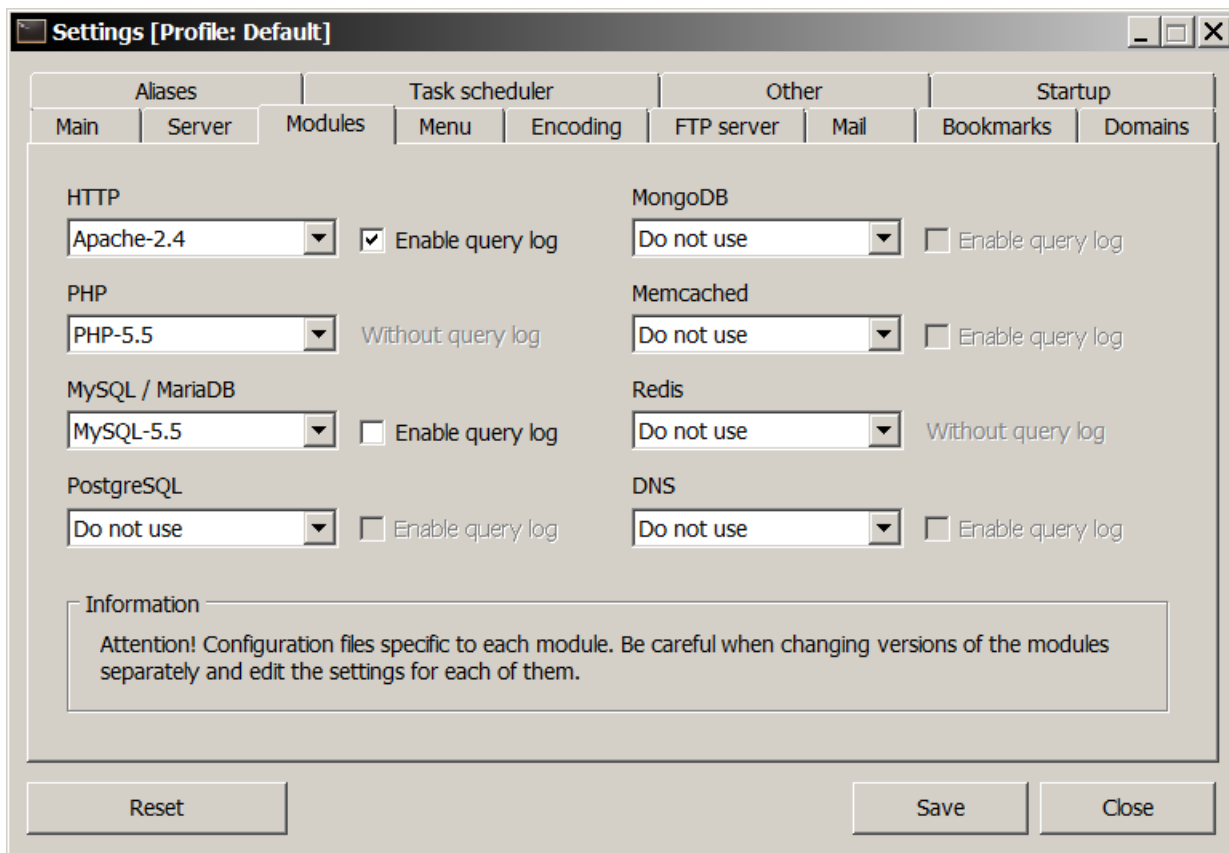


Рис. 2.14. Настройки локального веб-сервера OpenServer (указание используемой версии Apache, PHP и MySQL)

Порядок установки веб-сервера OpenServer в семействе систем Windows одинаков, разница заключается в дистрибутивах и разрядность релиза: скачать установочный дистрибутив с официального сайта; запустить установщик и выбрать директорию для установки (в данном случае D:\OpenServer); дождаться завершения установки и запустить веб-сервер; проверить правильность работы служб (иконка веб-сервера в трее должна загореться зеленым цветом) для этого в браузере FireFox набрать адрес localhost. На сервере можно получить информацию о версиях установленных служб, а также ссылку на получение информации о PHP (запуск функции phpinfo), ссылку на доступ в панель PhpMyAdmin и др.

В случае если веб-сервер не запускается, то необходимо устранить проблему сделав следующее: перезапустить поочередно веб-службы; отключить антивирус или брандмауэр если он установлен и запущен; отключить программу звонков типа Skype если она установлена и запущена; выполнить настройку файла hosts; установить дополнения типа Microsoft Visual C++ Redistributable; Отметим, что помимо локального веб-сервера OpenServer, также можно использовать программные аналоги типа WAMP, Denwer или XAMP.

Веб-приложение PHPMyAdmin с открытым кодом, написанное на языке PHP, представляет собой веб-интерфейс для администрирования системы управления базами данных MySQL без непосредственного ввода SQL команд. Приложение позволяет через браузер осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Подавляющее большинство российских провайдеров используют приложение в качестве панели управления для того, чтобы предоставить своим клиентам возможность администрирования выделенных им баз данных.

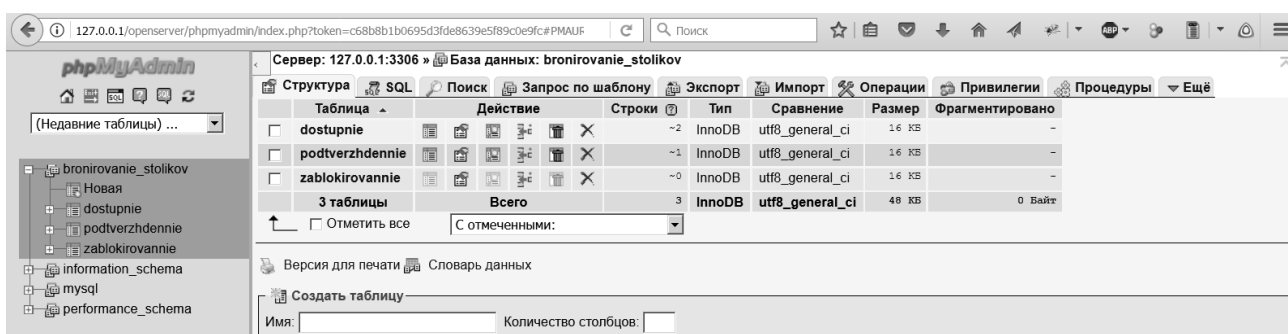


Рис. 2.15. Структура базы данных bronirovanie_stolikov в PHPMyAdmin (таблицы dostupnie, podtverzhdennie, zablokirovannie)

При необходимости разработанные таблицы в базе данных веб-приложения могут быть изменены, например, расширена таблица «podtverzhdennie» – добавлены поля типа: номер WhatsApp, время в которое

МОЖЕТ ЗВОНИТЬ ХОСТЕС И Т.Д.

ГЛАВА 3. РАЗРАБОТКА И РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ БРОНИРОВАНИЯ МЕСТ В КАФЕ

3.1 Разработка базы данных веб-приложения

В основе работы приложения «Table Reserver» и логики ее функционирования лежит база данных `bronirovanie_stolikov`.

Для реализации программного обеспечения «Table Reserver» база данных необходима для хранения исходных данных о столиках и местах в кафе, а также сведений о забронированных столиках и местах. Кроме этого в базу данных необходимо записывать сведения о пользователях, выполняющих бронирование мест, такие как фамилия, имя, телефон и т.д.

Для реализации требований к приложению «Table Reserver» необходимо разработать три таблицы:

1. Таблица `dostupnie` – доступные места имеет следующие поля: код, номер столика, число доступных мест, цена за место (поля: `id`, `nazvanie_stolika`, `dostupno_mest`, `cena_mesta`).

Таблица 3.1

Структура таблицы `dostupnie`

Название поля	Ключевое	Тип данных
<code>id</code>	Да	<code>INT(10)</code>
<code>nazvanie_stolika</code>	Нет	<code>VARCHAR(255)</code>
<code>dostupno_mest</code>	Нет	<code>INT(10)</code>
<code>cena_mesta</code>	Нет	<code>VARCHAR(50)</code>

После того как приложение генерирует кнопки с числами, соответствующими количеству бронируемых мест, соответствующее число

мест снимается из поля `dostupno_mest` и прибавляется к полю `zablokirovanno_mest` в таблице `zablokirovannie`.

Поле `dostupno_mest` автоматически увеличивается после удаления мест скриптом с истекшим сроком (3 минуты) на число временно заблокированных мест.

2. Таблица `zablokirovannie` – блокирование (резервирование).

Таблица 3.2

Структура таблицы `zablokirovannie`

Название поля	Ключевое	Тип данных
<code>id</code>	Да	INT(10)
<code>zablokirovannii_stolik</code>	Нет	VARCHAR(255)
<code>zablokirovanno_mest</code>	Нет	INT(10)
<code>vremya_blokirovaniya</code>	Нет	TIMESTAMP

Таблица является промежуточной служебной таблицей.

```
sozdat_stolki.php x
<?php include_once("podklychenie.php");
// файл для создания таблиц в базе данных бронирование столиков (бронирование мест в кафе)
// создание таблицы dostupnie (доступные места: код, номер столика, доступно мест, цена)
// id, название столика, avail, цена места
$stolik_dostupn =
"CREATE TABLE IF NOT EXISTS dostupnie ( id int(11) NOT NULL AUTO_INCREMENT, nazvanie_stolika varchar(50) NOT NULL, dostupno_mest int(11) NOT NULL, cena_mesta v
$zapros = mysql_query($stroka_podklycheniya, $stolik_dostupn);
if ($zapros == TRUE) {
    echo "<h3>Таблица dostupnie создана :</h3>";
} else {
    echo "<h3>Ошибка создания таблицы dostupnie :</h3>";
}
}
// создание таблицы podtverzhdennie (занятые места пользователями: код, номер столика, заказчик, эл. почта, телефон, сообщение)
// id, podtverzhdennii_stolik, podtverzhdeno_mest, klient, email, phone, soobshenie
$stolik_podtverzhdenn =
"CREATE TABLE IF NOT EXISTS podtverzhdennie ( id int(11) NOT NULL AUTO_INCREMENT, podtverzhdennii_stolik varchar(50) NOT NULL, podtverzhdeno_mest int(11) NOT NU
$zapros = mysql_query($stroka_podklycheniya, $stolik_podtverzhdenn);
if ($zapros == TRUE) {
    echo "<h3>Таблица podtverzhdennie создана :</h3>";
} else {
    echo "<h3>Ошибка создания таблицы podtverzhdennie :</h3>";
}
}
// создание таблицы zablokirovannii (бронирование: код, номер столика, количество мест, время)
// id, zablokirovannii_stolik, zablokirovanno_mest, vremya_blokirovaniya
$stolik_zablokirovan =
"CREATE TABLE IF NOT EXISTS zablokirovannii ( id int(11) NOT NULL AUTO_INCREMENT, zablokirovannii_stolik varchar(50) NOT NULL, zablokirovanno_mest int(11) NOT NU
$zapros = mysql_query($stroka_podklycheniya, $stolik_zablokirovan);
if ($zapros == TRUE) {
    echo "<h3>таблица zablokirovannii создана :</h3>";
} else {
    echo "<h3>Ошибка создания таблицы zablokirovannii :</h3>";
}
}
// запись исходных данных в таблицу dostupnie (доступные места: код, номер столика, доступно мест, цена)
// данные обязательно должны быть внесены в базу данных бронирование столиков
// пользователь сможет видеть на веб-странице свободные-занятые столики, свободные места за каждым столиком и цену бронирования за место
$stolik_zapisat = "INSERT INTO dostupnie (nazvanie_stolika,dostupno_mest,cena_mesta) VALUES ('Столик 1', 10, '500.00'), ('Столик 2', 8, '500.00'), ('VIP-столк 1',
$zapros = mysql_query($stroka_podklycheniya, $stolik_zapisat);
//создание таблицы dostupnie (доступные места)
if ($zapros == TRUE) {
    echo "<h3>Исходные данные о столиках в заведении записаны в БД :</h3>";
} else {
```

Рис. 3.16. Файл sozdat_stolki.php для автоматического создания таблиц и внесения исходных данных в базу данных бронирование_столиков

Таблица используется для блокирования выбранных мест за определенным столиком от других пользователей: код, номер столика, количество бронируемых мест, время (поля: id, zablokirovannii_stolik, zablokirovanno_mest, vremya_blokirovaniya).

Поле vremya_blokirovaniya с типом дата-время используется скриптом для снятия блокировки (удаления) заблокированных мест с истекшим временем (3 минуты). Т.е. если запись в поле vremya_blokirovaniya храниться более трех минут, то ее необходимо удалить и при этом увеличить количество доступных мест в поле dostupno_mest таблицы dostupnie.

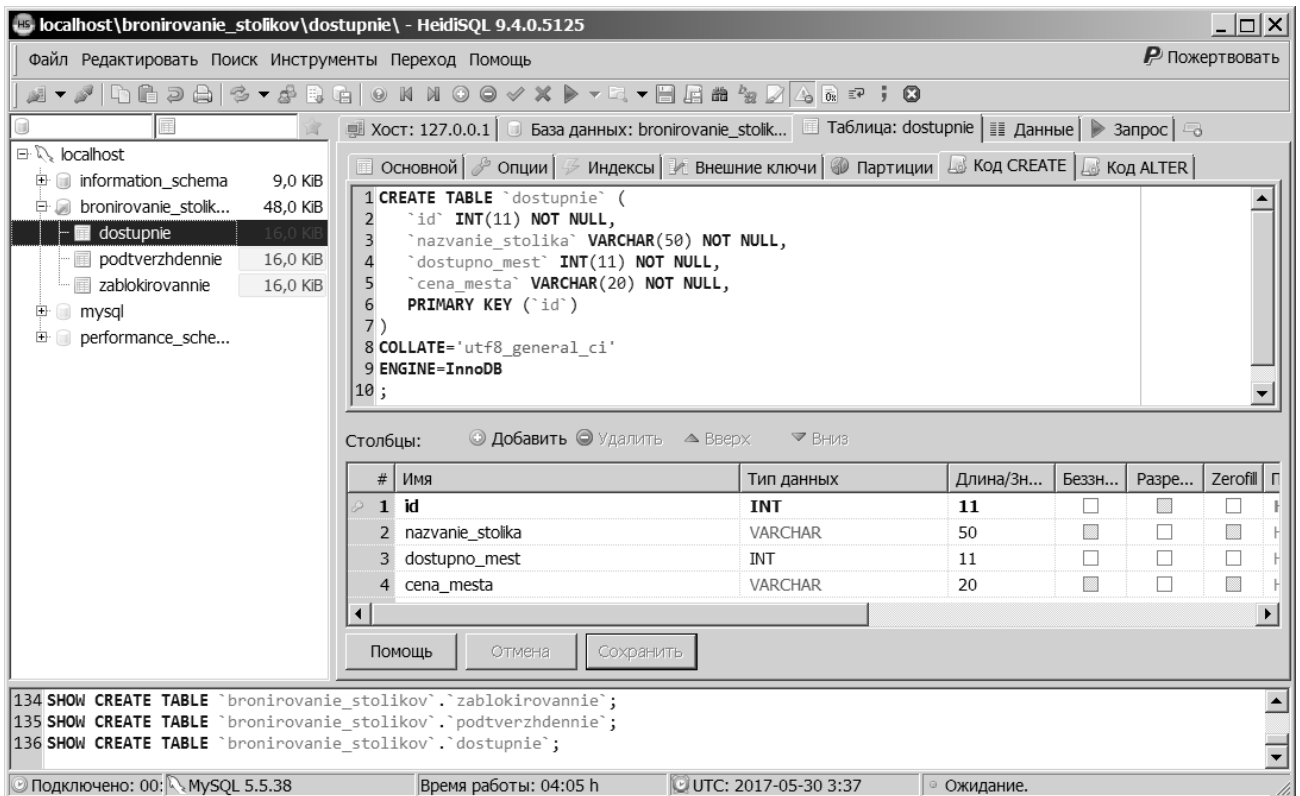


Рис. 3.17. Определение структуры таблицы reserves из базы данных reservations в HeidiSQL (полей, ключевого поля, типов данных, длины, сопоставлений и т.д.)

После того как пользователь ввел имя и свои данные, и нажал кнопку «Бронировать места», данные из таблицы zablokirovannie нужно отправить (синхронизировать) в таблицу confirms.

Если пользователь не стал вводить имя и свои данные и нажал кнопку «Отменить бронирование», т.е. освобождает места за столиком – необходимо вернуть (прибавить число) места в таблицу dostupnie.

3. Таблица podtverzhdennii – занятые места пользователями, т.е. заказ брони подтвержден: код, номер столика, заказчик, эл. почта, телефон, сообщение (поля: id, podtverzhdennii_stolik, podtverzhdeno_mest, klient, email, phone, soobshenie).

Для полей таблиц (столбцов) использованы следующие типы данных: целочисленный – int, символьный переменной длины – varchar, дата-время – datetime.

Структура таблицы podtverzhdennie

Название поля	Ключевое	Тип данных
id	Да	INT(10)
podtverzhdennii_stolik	Нет	VARCHAR(255)
podtverzhdeno_mest	Нет	INT(10)
klient	Нет	VARCHAR(50)
email	Нет	VARCHAR(50)
phone	Нет	VARCHAR(50)
soobshenie	Нет	VARCHAR(255)

После создания таблиц необходимо внести исходные данные по кафе, а именно тип столика (название), количество имеющихся столиков, количество мест за столиком, стоимость места. Для упрощения работы с веб-приложением «Table Reserver» создан файл soztat_stoliki.php, который позволяет автоматически создать необходимые таблицы и внести исходные данные в них.

Для упрощения работы с базой данных, разрабатываемого программного обеспечения «Table Reserver» использовалось приложение HeidiSQL (другим примером может служить Toad for MySQL или PhpMyAdmin).

Приложение HeidiSQL предназначено для управления базами данных MySQL и Microsoft SQL. С помощью HeidiSQL возможно просматривать и редактировать данные, создавать и редактировать таблицы, процедуры, триггеры и запланированные события. Структуру базы и данные можно экспортировать в SQL файлы, буфер обмена или на другие серверы.

3.2. Порядок разработки веб-приложения

В данной работе веб-разработка описывает процесс создания веб-приложения «Table Reserver».

В процессе веб-разработки выполняется четыре ключевых вида работ: дизайн, верстка, установка и настройка веб-сервера (возможно использовать бесплатный хостинг типа www.byet.host, однако процесс разработки и отладки в некотором смысле будет более трудоемким), разработка базы данных и программирование на клиентской и серверной стороне. Дизайн – в данном случае подразумевает создание «прототипа» интерфейса веб-страницы с которой будет работать пользователь, см. пункт 1.4. Верстка – написание кода на языках HTML и CSS. Конфигурирование веб-сервера – локального веб-сервера OpenServer версии 5.1.1, выбор версии Apache, интерпретатора PHP и системы управления базами данных MySQL. Разработка базы данных MySQL и программирование на стороне клиент-сервер с использованием языков PHP, AJAX, JavaScript, MySQL.

Основные этапы разработки веб-приложения «Table Reserver»:

1. Проектирование веб-приложения (планирование разработки, сбор и анализ требований, постановка цели и задач, решаемых приложением). Рассматриваются аналогичные системы бронирования мест, представленные на рынке программного обеспечения, определяются их сильные и слабые стороны. Составляется некое техническое задание с учетом возможностей разработчика (наличия аппаратно-программного обеспечения и т.д.). Этап проектирования веб-приложения зависит от многих факторов, таких как возможности и функциональность приложения. Основные требования к веб-приложению представлены в разделе 2.1.

2. Разработка логики работы веб-приложения. Общий сценарий работы пользователя с веб-приложением «Table Reserver» описан в разделе 2.1.

Исходя из требований к веб-приложению «Table Reserver» основная сложность связана с поддержкой одновременной работы заранее неизвестного количества пользователей. В этой связи в логику работы веб-приложения необходимо внести функцию «удаление (блокировка)

неподтвержденной пользователем брони столика и мест по истечении времени (3 минут)». Использовать программный таймер не рекомендуется, т.к. это приведет к дополнительной трате вычислительных ресурсов (например, пользователь может несколько раз нажимать кнопку браузера «Обновить страницу»). В данном случае, выбор сделан в пользу внесения записи с типом дата-время в базу данных, т.е. когда пользователь кликает по кнопке с конкретным числом бронируемых мест, выполняется скрипт, проверяющий бронь на предмет подтверждения пользователем в течение трех минут (иначе, бронь автоматически снимается (удаляется запись и базы данных)).

Часть логики работы веб-приложения «Table Reserver» описана в разделе 2.2 при разработке базы данных.

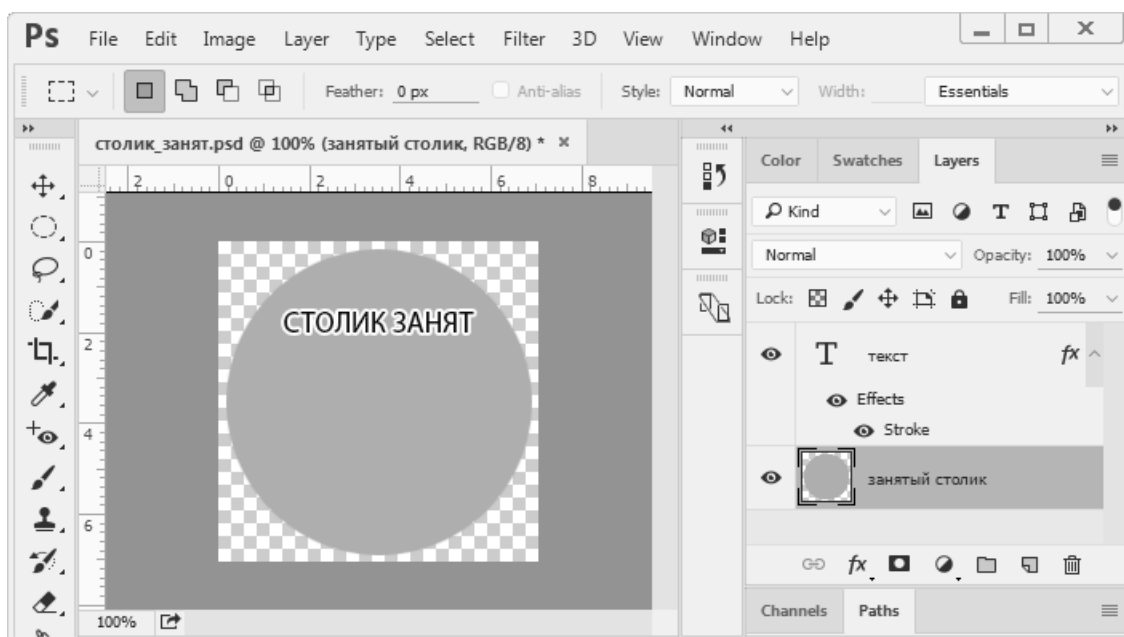


Рис. 3.18. Подготовка графического контента для веб-приложения «Table Reserver» в графическом редакторе Adobe PhotoShop

3. Проектирование интерфейса пользователя в соответствии с требованиями к веб-приложению. Прототип интерфейса пользователя веб-приложения для бронирования мест за столиком представлен в разделе 2.1.

4. Подготовка контента включала создание графических изображений в приложении Adobe Photoshop (исходный формат для внесения изменений –

*.psd, и экспортированный сжатый формат – *.jpg) и набор текстовых сообщений, выдаваемых веб-приложением.

5. Верстка и программирование. Конечный вариант интерфейса переводится в HTML-страницу, создается ее разметка для работы с веб-приложением в браузере. Далее выполняется программирование «с нуля». Завершающим этапом разработки веб-приложения является тестирование.

```
<!-- строим основную веб-страницу для пользователя -->
<!DOCTYPE html>
<html>
<head>
  <!-- параметры кодировки -->
  <meta charset="UTF-8">
  <!-- подключаем файл таблицы стилей на CSS -->
  <link rel="stylesheet" href="oformlenie.css">
  <!-- подключаем основной скрипт на JavaScript -->
  <script src="bronirovanie.js"></script>
</head>
<body>
  <!-- делаем обертку -->
  <div id="wrapper">
    <div id="stage"></div>
    <div id="seats"> <?php echo $interfeis; ?> </div>
    <div id="returnData">Кликните по интересующему свободному
столику с доступными для бронирования местами!</div>
  </div>
</body>
</html>
```

Рис. 3.19. Основной код HTML-разметки для формирования интерфейса веб-приложения «Table Reserver»

6. Этапы тестирования, внесения изменений, размещения в сети Интернет и внедрение веб-приложения описаны далее в работе.

Структура разработанного проекта в виде дерева файлов представлена на рисунке 15.

Все файлы проекта распложены в каталоге `_booking_seats_project`.

Основные файлы, отвечающие за работу веб-приложения «Table Reserver» находятся в корне каталога `_booking_seats_project`. В случае разработки веб-приложения на локальном веб-сервере OpenServer рабочий каталог находится по адресу: `D:\OpenServer\domains\localhost\`.

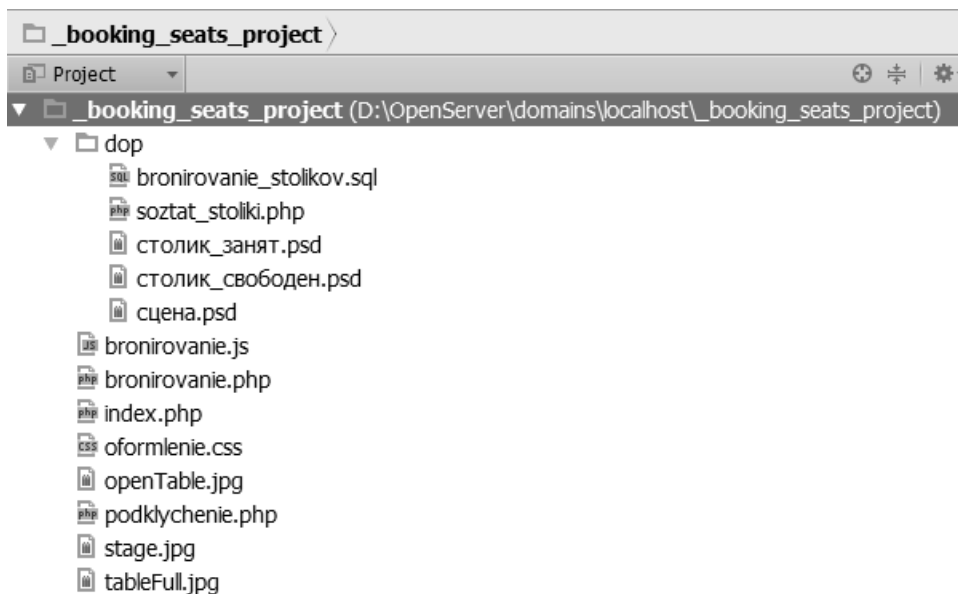


Рис. 3.20. Основные файлы проекта `_booking_seats_project`, отвечающие за работу веб-приложения «Table Reserver»

Файл `index.php` – является «стартовой точкой» веб-приложения. Файл содержит код на языках HTML, PHP и MySQL. Путем отправки запросов к базе данных происходит вывод сведений о свободных и занятых столиках.

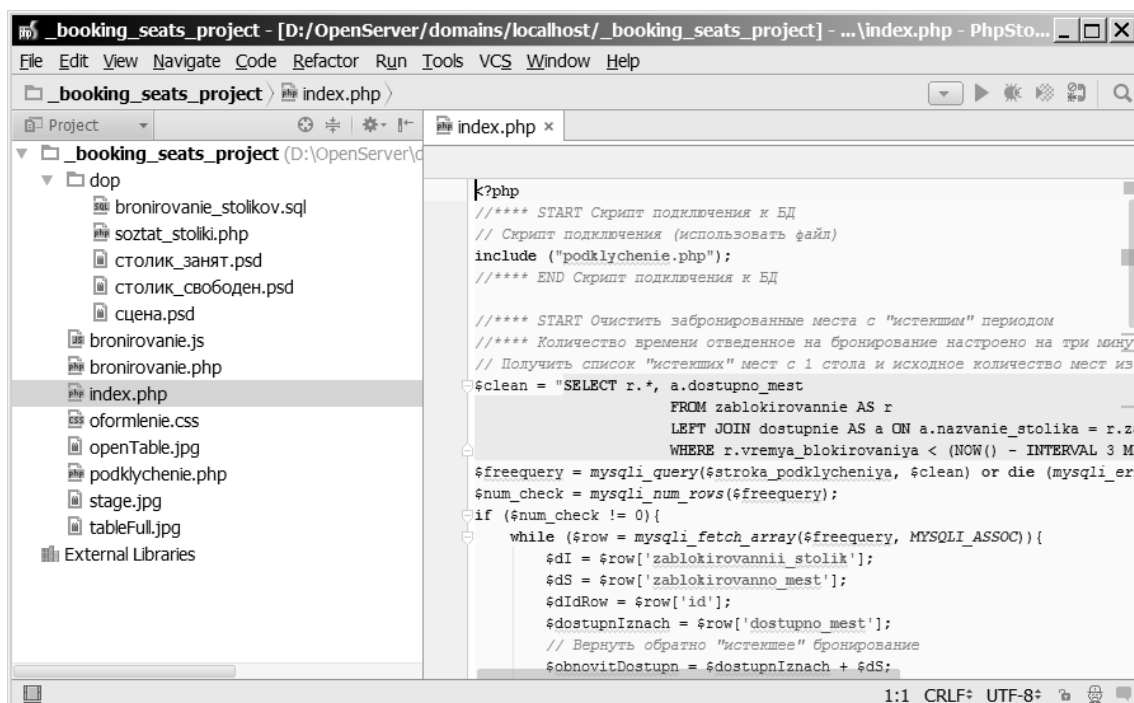


Рис. 3.21. Файл `index.php`

Построение основной пользовательской веб-страницы – веб-

интерфейса в файле index.php выполняется путем заключения кода PHP внутри HTML-тегов – конструкция «<div id="seats"> <?php echo \$interfeis; ?> </div>».

Файл index.php обнуляет интервал резервирования столиков в три минуты. Возвращает зарезервированные места в таблицу доступные. В основном логика работы в файле построена на циклах с предусловием «while» и логических операторах «if». Файл index.php связан с файлом podklychenie.php путем использования инструкции типа «include ("podklychenie.php");».

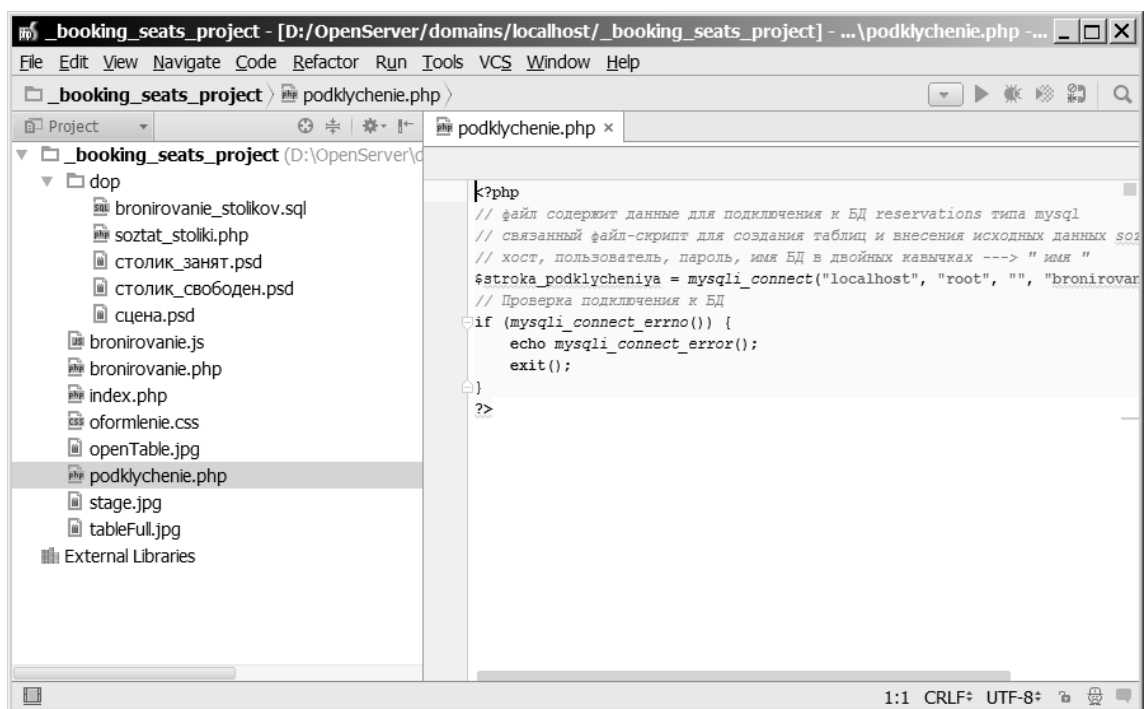
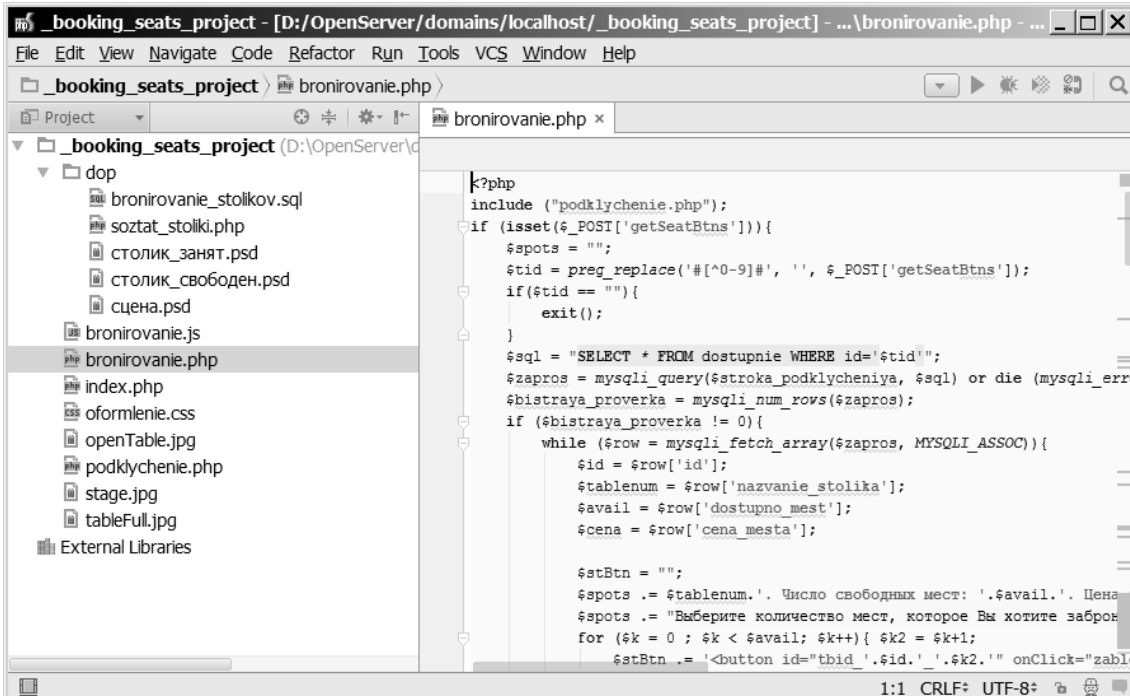


Рис. 3.22. Файл podklychenie.php

Файл podklychenie.php – файл полностью написан на языке PHP, отвечает за проверку корректности подключения и непосредственно подключение к базе данных MySQL. В файле содержатся такие параметры подключения как адрес веб-сервера (в данном случае localhost), имя администратора системы управления базами данных (в данном случае root), пароль администратора системы управления базами данных (в данном случае

отсутствует) и имя рабочей базы данных. В случае отсутствия базы данных, или неверного указания какого-либо параметра подключения пользователь увидит в браузере сообщение об ошибке (использован логический оператор «если» и инструкция «echo»).

Файл `bronirovanie.php` – файл на языке PHP отвечает за вывод пользователю сведений о доступных-занятых местах в ходе его работы с веб-приложением. Файл также формирует основную форму бронирования с идентификатором «confirmform» мест с полями типа: Ф.И.О., телефон, эл. почта, сообщение и выполняет отправку введенных пользователем данных на проверку скрипту JavaScript, который в случае корректного заполнения формы выполняет запись пользовательских данных в базу данных. Создание элементов формы выполняется с помощью стандартных HTML-тегов типа «input» и «button». Отметим, что форма подтверждения «confirmform» для хранения пользовательских данных содержит скрытые поля с атрибутом (свойством) «hidden». Файл связан с файлом `podklychenie.php`.



```

<?php
include ("podklychenie.php");
if (isset($_POST['getSeatBtns'])) {
    $spots = "";
    $tid = preg_replace('#^[^0-9]#', '', $_POST['getSeatBtns']);
    if ($tid == "") {
        exit();
    }
    $sql = "SELECT * FROM dostupnie WHERE id='$tid'";
    $zapros = mysql_query($stroka_podklycheniya, $sql) or die (mysql_errr
    $bistraya_proverka = mysql_num_rows($zapros);
    if ($bistraya_proverka != 0) {
        while ($row = mysql_fetch_array($zapros, MYSQLI_ASSOC)) {
            $id = $row['id'];
            $tablename = $row['nazvanie_stolika'];
            $avail = $row['dostupno_mest'];
            $cena = $row['cena_mesta'];

            $stBtn = "";
            $spots .= $tablename.'. Число свободных мест: '.$avail.'. Цена
            $spots .= "Выберите количество мест, которое Вы хотите заброн
            for ($k = 0 ; $k < $avail; $k++) { $k2 = $k+1;
                $stBtn .= "<button id='tbody_'. $id.' '. $k2.'" onClick="zablc

```

Рис. 3.23. Файл `bronirovanie.php`

Логику работы в файле `bronirovanie.php` обеспечивают конструкции

типа: «if – isset – \$_POST», например, «if (isset(\$_POST["имя-идентификатор-поля-ввода"])) {», т.е. проверка заполнения поля с указанным идентификатором, если выражение истинно, то выполняются инструкции из тела оператора «если». Другими словами выполняется доступ к данным из HTML-формы.

Файл `bronirovanie.js` – написан на языке JavaScript, для обращения к элементам веб-страницы используется объектная модель документа DOM (Document Object Model) – например, инструкции типа «получить элемент по его идентификатору» – «`document.getElementById("name").value;`», а также возможности технологии AJAX для «фоновой» передачи данных (инструкции типа «`var hr = new XMLHttpRequest();`»).

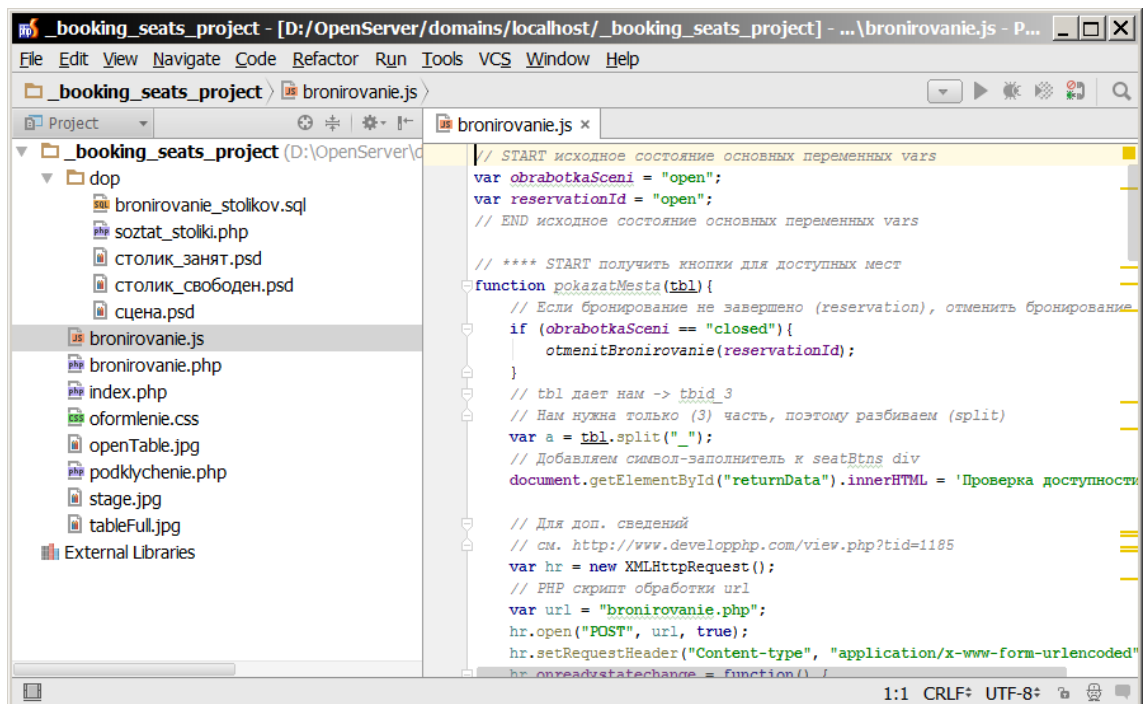


Рис. 3.24. Файл `bronirovanie.js`

Объект `XMLHttpRequest` (или кратко «XHR») дает возможность из JavaScript делать HTTP-запросы к серверу без перезагрузки страницы. Несмотря на слово «XML» в названии, `XMLHttpRequest` может работать с любыми данными, а не только с XML. В данном файле объект `XMLHttpRequest` используется для загрузки данных.

Файл `bronirovanie.js` содержит основные функции обеспечивающие

работу веб-приложения «Table Reserver»: pokazatMesta с входным параметром «столик», zablokirovatMesta с входным параметром «число резервируемых мест на три минуты», podtverditMesta (функция получает данные из пользовательской формы описанной в файле bronirovanie.php, выполняет проверку заполнения и запись данных в базу данных), otmenitBronirovanie с входным параметром «идентификатор блокирования (резервирования мест)».

Файл oformlenie.css – содержит исходный код на языке CSS, отвечает за оформление пользовательского веб-интерфейса. Другими словами, для дизайна веб-приложения используется внешний файл стилей, к которому обращается основной файл приложения index.php посредством тега «<link rel="stylesheet" href="oformlenie.css">» в заголовке веб-страницы (вложен в секцию head).

Код на языке CSS состоит из повторяющихся блоков следующего вида: «селектор { свойство1: значение; свойство2: значение; }». Такой блок называется «CSS-правило». Каждое CSS-правило содержит хотя бы один селектор и свойство. Простейшие селекторы – это селекторы по именам тегов. С их помощью можно задать стили для всех абзацев на странице, для всех ссылок, заголовков первого уровня и так далее.

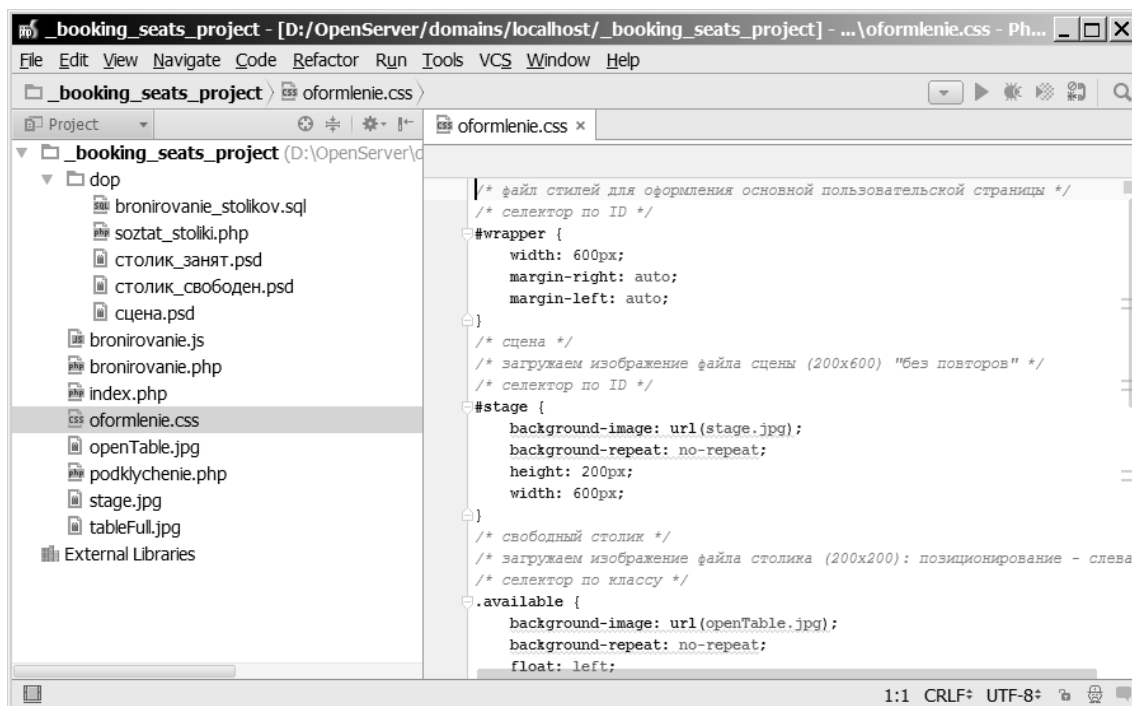


Рис. 3.25. Файл oformlenie.css

Для обращения к HTML-элементам веб-страницы в целях оформления используются селекторы по идентификатору «#» и по классу «.класс». Класс позволяет объединять разные элементы в смысловые группы и применять к ним одинаковое оформление. Например, можно создать класс «элементы с ошибкой» и задать ему красный цвет текста. Затем можно добавлять этот класс к любому HTML-тегу: абзацу, заголовку, элементу списка и так далее.

Графические файлы: openTable (свободный столик), tableFull (занятый столик), stage (сцена, бар или танцевальная площадка) в формате jpg используются файлом oformlenie.css, т.е. кодом на языке CSS, который в частности отвечает за вывод на веб-страницу графических файлов.

В основном каталоге имеется вложенная папка dop в которой хранятся исходные файлы изображений столиков в формате psd, а также экспортированная база данных bronirovanie_stolikov в формат SQL (со тремя таблицами и исходными данными о занятых-свободных столиках в кафе).

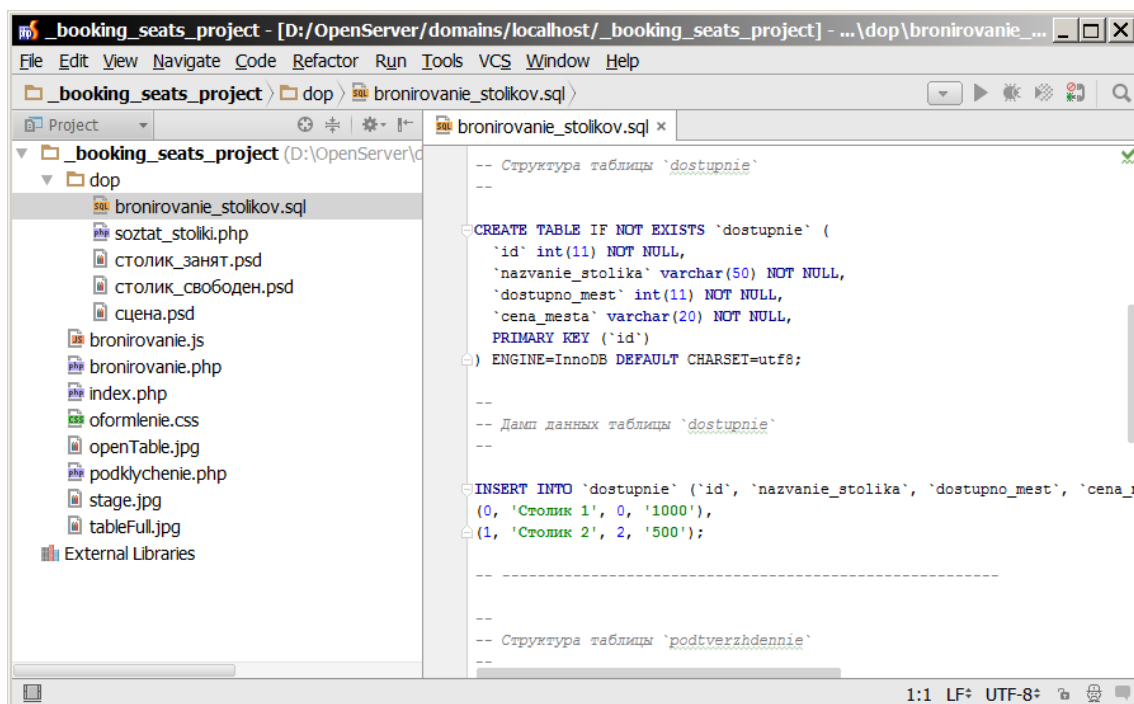


Рис. 3.26. Файл базы данных bronirovanie_stolikov.sql

В папке `dop` также имеется файл `soztat_stoliki.php` – специально разработан для автоматизации создания трех таблиц веб-приложения и внесения в базу данных исходных данных. Файл `soztat_stoliki.php` должен быть связан с файлом `podklychenie.php`, а именно с указанным именем базы данных в файле `podklychenie.php`.

Исходный код разработанного веб-приложения и его внешний вид представлены приложении к данной работе. Код сопровождается комментариями (конструкции `<<!--комментарий_в_PHP-->` и `<<!--комментарий_в_HTML_CSS-->`).

Следует отметить, что разработка веб-приложения «Table Reserver» представляет собой повторяющийся многократно процесс, т.е. основана на итеративном подходе к разработке программного обеспечения, а именно представляет собой выполнение работ параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы: планирование – реализация – проверка – оценка.

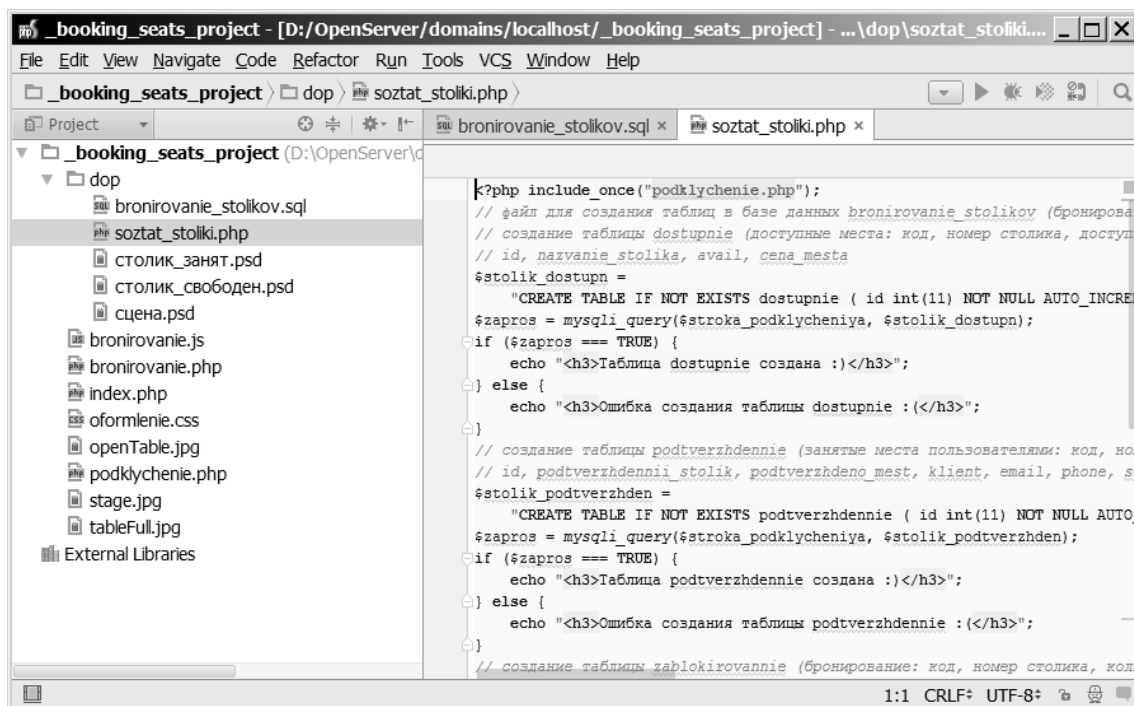


Рис. 3.27. Файл sozdat_stoliki.php

В случае исполнения и работы с веб-приложением «Table Reserver» на локальном веб-сервере OpenServer его запуск должен выполняться через браузер, в данном случае FireFox при вводе URL-адреса следующего вида: http://localhost/_booking_seats_project/index.php).

3.3. Тестирование веб-приложения

Процесс тестирования веб-приложения «Table Reserver» может проводиться под различные требования и включать в себя самые разнообразные проверки для оценки правильности его работы согласно изначальным требованиям: просмотр страницы веб-приложения с увеличенным масштабом, просмотр при изменении размеров окна браузера, нажатие кнопок «Обновить» или «Назад», ввод «некорректных» данных в поля ввода в процессе бронирования или вообще их игнорирование и многие другие.

В первую очередь у веб-приложения «Table Reserver» следует проверить работу системы навигации. Систему навигации также позволяют проверить инструменты веб-разработчика типа Яндекс, специализированное программное обеспечение, например, Semonitor или Xenu's Link Sleuth, а также онлайн-сервисы, например, www.validator.w3.org/checklink.



Рис. 3.28. Тестирование веб-приложения на работу с устройствами

Работу веб-приложения «Table Reserver» следует протестировать в разных браузерах и различных платформах (мобильных и настольных), т.к. пользователи могут заходить на сайт кафе для бронирования столиков с различных браузеров и устройств.

Следующий шаг проверки «Table Reserver» – проверка соответствия веб-стандартам с помощью валидаторов. Браузер может успешно отображать страницу не соответствующую стандартам, но проблемы могут возникнуть при работе веб-приложения и его индексации роботом поисковой системы. Достичь полной валидации всех страниц приложения достаточно трудно.

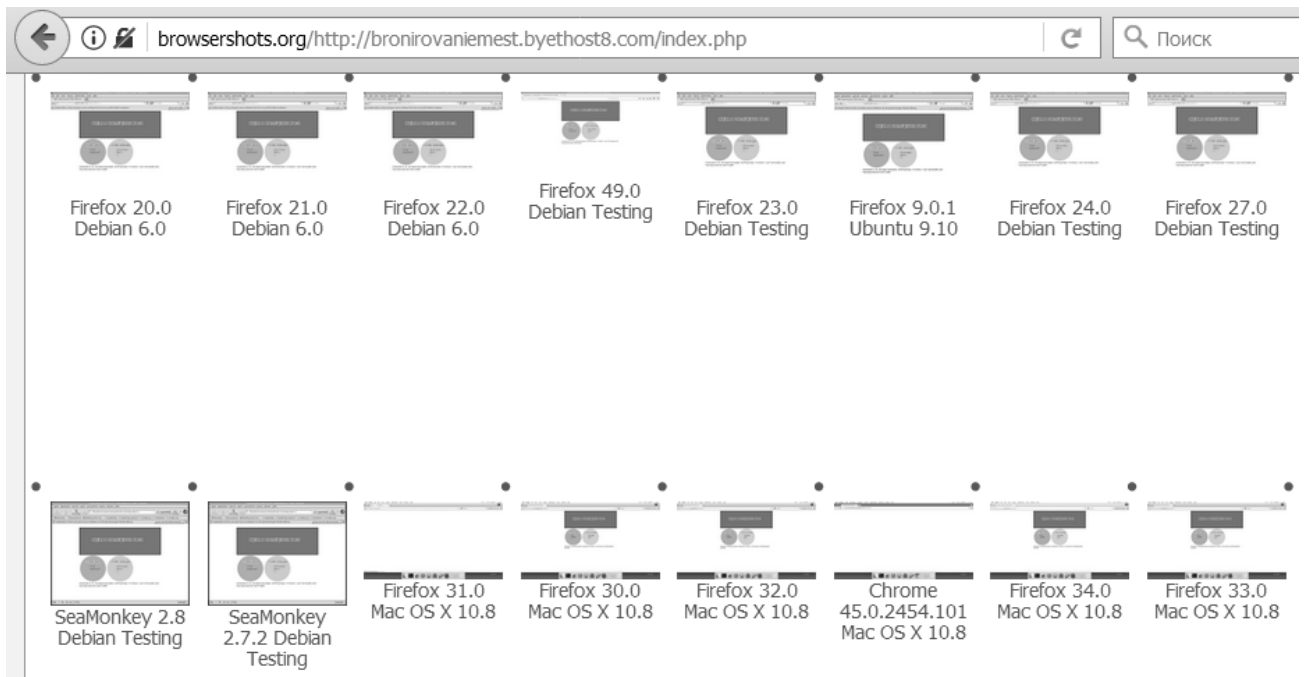


Рис. 3.29. Тестирование веб-приложения на работу с браузерами

Рекомендуется проведение тестирования скорости загрузки веб-приложения «Table Reserver», т.е. время того, как быстро пользователь получает доступ к приложению, что влияет на его высокую посещаемость. Для тестирования скорости открытия веб-страниц приложения существуют сервисы типа host-tracker.com.

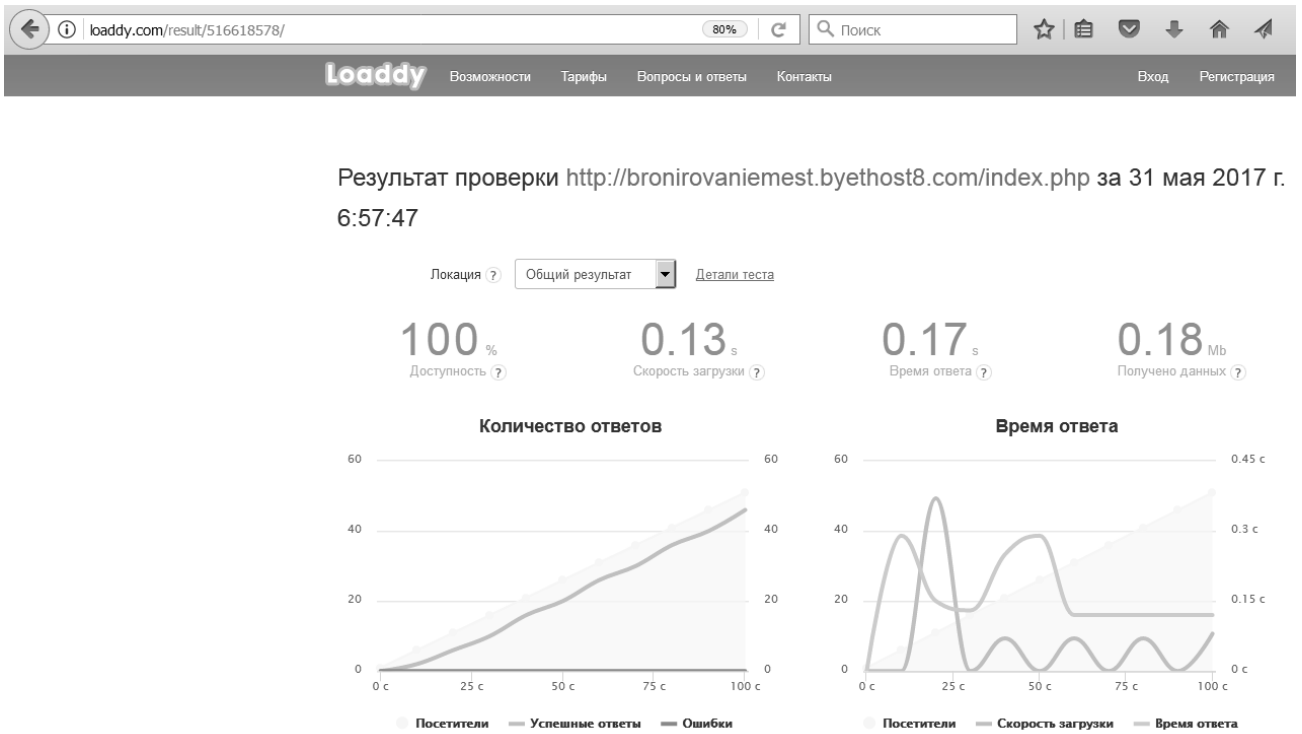


Рис. 3.30. Нагрузочное тестирование веб-приложения

Для проверки валидности, использовался сервис wave.webaim.org. В результате сервис указывает ошибки, отмечая места различными цветами.

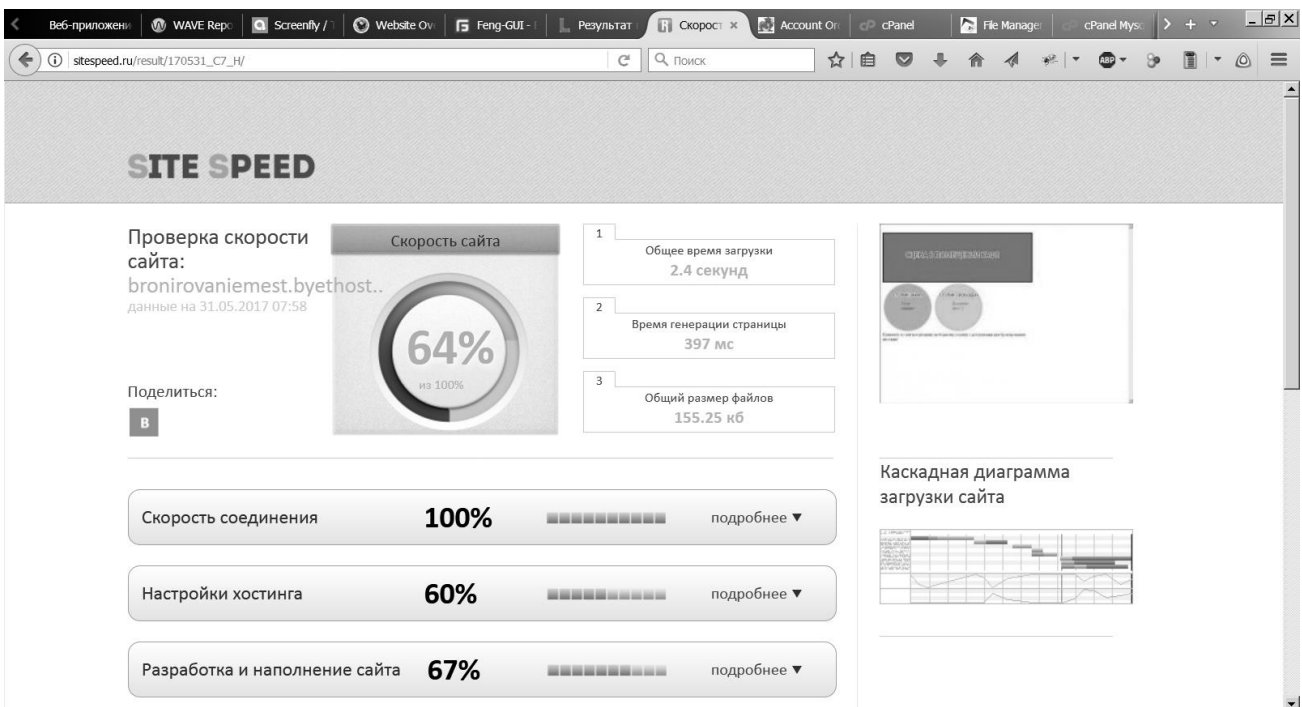


Рис. 3.31 Тестирование веб-приложения на скорость загрузки

Тестирование адаптивности веб-приложения к различным расширениям экрана выполнено при помощи сервиса quirktools.com, приложение хорошо отображается на экранах ноутбуков, планшетов и компьютеров.

Проверка на корректное отображение в различных браузерах выполнено при помощи browsershots.org, искажений не обнаружено.

Сервиса feng-gui.com позволил выявить направление взгляда пользователя на точках.

Нагрузочное тестирование выполнено сервисом loaddy.com, приложение показало стабильную работу. Время отклика функций в норме.

Скорость загрузки протестирована сервисом sitespeed.ru.

Тестирование является завершающим этапом разработки веб-приложения и играет важную роль в процессе создания качественного программного обеспечения.

В данном случае, тестирование веб-приложения «Table Reserver» с помощью онлайн сервисов проведено после его размещения на бесплатном хостинге byet.host, т.к. данные сервисы не выполняют поддержку локальных веб-серверов типа OpenServer на котором изначально разрабатывалось веб-приложение.

3.4. Внедрение веб-приложения в деятельность кафе

Внедрение веб-приложения в деятельность любого кафе представляет собой перенос разработанных и отлаженных на локальном веб-сервере OpenServer файлов и базы данных приложения на хостинг-площадку и открытие доступа к нему пользователям посредством приобретения доменного имени. Если у предприятия питания имеется собственный сайт, то приложение внедряется также путем переноса файлов и базы данных (или ее объединения с существующей базой данных, например, базой системы управления содержимым Joomla!), а также заданием ссылки на основной

файл приложения index.php.

При переносе веб-приложения «Table Reserver» на хостинг-площадку, например, www.sweb.ru и приобретении доменного имени, путь к веб-приложению должен быть изменен соответствующим образом: например, www.кафе.рф/бронирование-мест/index.php.

Хостинг-провайдер предоставляет услугу хостинга и выбора доменного имени, предлагает администратору панель управления, позволяющую настроить работу веб-приложения, управлять файлами и базами данных, доменами и поддоменами, а также восстановить файлы из резервной копии в случае каких-либо непредвиденных обстоятельств.

Пре выборе хостинга необходимо учитывать: способ и место регистрация домена; обеспечение технической поддержки; производительность используемых SSD-серверов; надежность дата-центра; стоимость услуг. В данном случае, выбор сделан в пользу «бесплатного» хостинга.

Процесс публикация веб-приложения в сети Интернет аналогичен публикации динамического веб-сайта и состоит из следующих этапов:

1. Выбор свободного доменного имени (адреса). В данном случае, поддомена, т.к. хостинг-площадка – «бесплатная».
2. Прохождение регистрации домена (пользователя).
3. Перенос базы данных MySQL на хостинг.

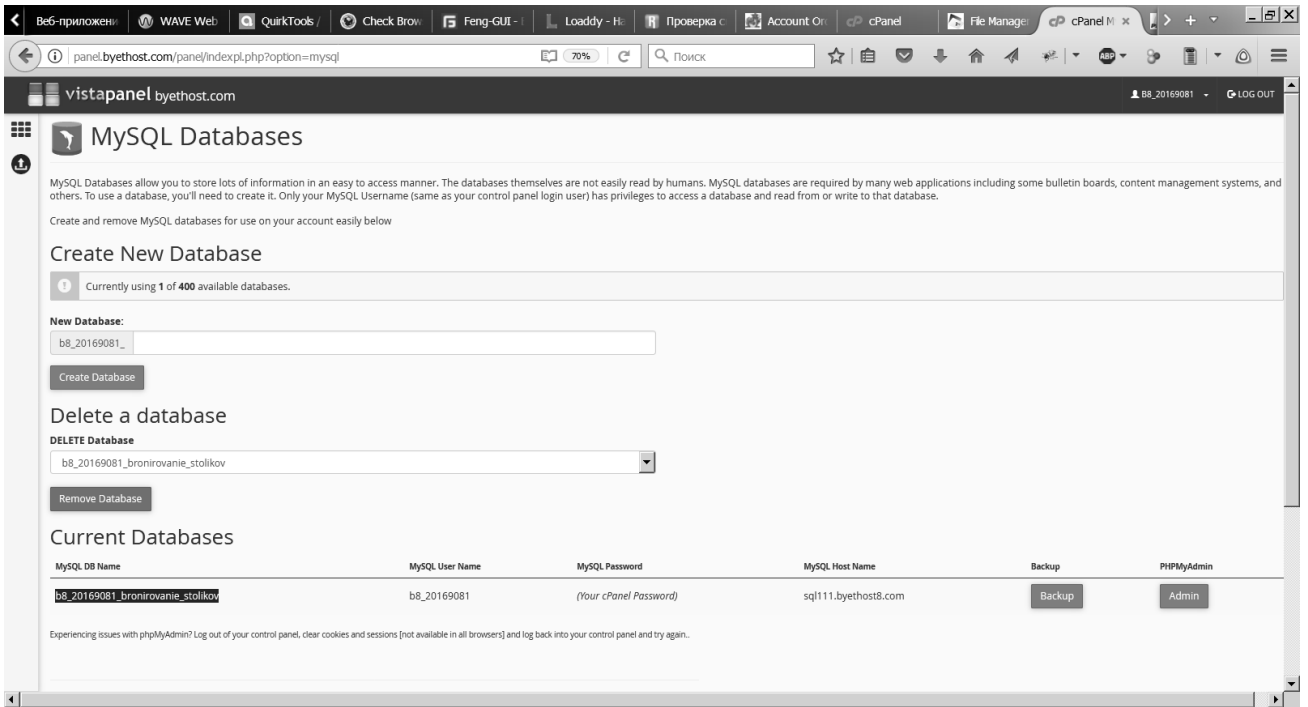


Рис. 3.32. Создание базы данных MySQL на хостинг-площадке для импорта разработанной на локальном хостинге базы

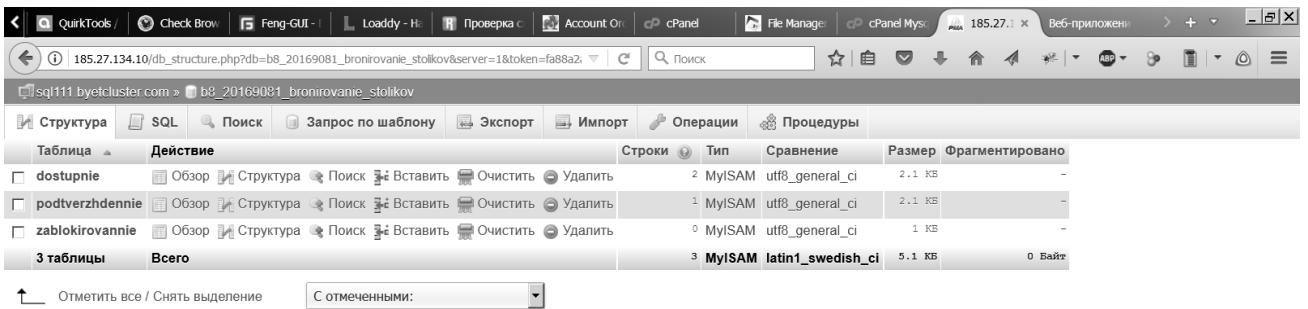


Рис. 3.33. Импортированная на сервер хостинг-площадки база данных MySQL

4. Перенос файлов веб-приложения через файловый менеджер на хостинг, настройка файла подключения к базе данных в соответствии с выданными регистрационными данными.

Файловый менеджер позволяет загрузить файлы и работать с ними на сервере хостинг-провайдера: переименовывать, создавать, удалять, перемещать, назначать права доступа и т.д.

All	Name	Type	Size	Owner	Group	Perms	Mod Time	Actions
	Up ..							
<input type="checkbox"/>	bronirovanie.js	JavaScript file	6099	20169081	20169081	rw-r--r--	May 30 23:52	Edit
<input type="checkbox"/>	bronirovanie.php	PHP script	7928	20169081	20169081	rw-r--r--	May 30 23:52	Edit
<input type="checkbox"/>	index.php	PHP script	4468	20169081	20169081	rw-r--r--	May 30 23:52	Edit
<input type="checkbox"/>	oformlenie.css	Cascading Style Sheet	1509	20169081	20169081	rw-r--r--	May 30 23:52	Edit
<input type="checkbox"/>	openTable.jpg	JPEG file	32454	20169081	20169081	rw-r--r--	May 30 23:52	Edit
<input type="checkbox"/>	podklychenie.php	PHP script	645	20169081	20169081	rw-r--r--	May 30 23:56	Edit
<input type="checkbox"/>	stage.jpg	JPEG file	56673	20169081	20169081	rw-r--r--	May 30 23:52	Edit
<input type="checkbox"/>	tableFull.jpg	JPEG file	32417	20169081	20169081	rw-r--r--	May 30 23:52	Edit

Рис. 3.34. Работа с файлами веб-приложения в файловом менеджере на хостинг-площадке

```

<?php
// файл содержит данные для подключения к БД reservations типа mysql
// связанный файл-скрипт для создания таблиц и внесения исходных данных sozdat_stoliki.php
// хост, пользователь, пароль, имя БД в двойных кавычках --> " имя "
$stroka_podklycheniya = mysqli_connect("sql111.byethost8.com", "b8_20169081", "256352256352", "b8_20169081_bronirovanie_stolikov");
// Проверка подключения к БД
if (mysqli_connect_errno()) {
    echo mysqli_connect_error();
    exit();
}
?>

```

Рис. 3.35. Настройка файла подключения к базе данных в соответствии с полученными регистрационными данными от хостинг-провайдера

5. Проверка работоспособности веб-приложения.

После внедрения веб-приложения «Table Reserver» на хостинг-площадку работа с ним стала доступной по адресу <http://bronirovaniemest.byethost8.com/index.php>. В будущем любому кафе также следует уделить внимание SEO-оптимизации ресурса и приложения, а также его дальнейшей доработки и сопровождению (см. пункт 1.4).

Когда конкретный заказчик или его доверенное лицо просмотрят готовый проект и в случае их удовлетворения подписываются документы о сдаче проекта – веб-приложения «Table Reserver».

ЗАКЛЮЧЕНИЕ

На сегодняшний день подавляющее множество кафе с бронированием столиков и мест получают большинство своих клиентов через Интернет, а конкретно через поисковые системы. Схема взаимодействия с клиентом выглядит примерно следующим образом: клиент находит подходящий ему сайт в Интернете, выбирает понравившееся ему кафе и заполняет заявку на бронирование на соответствующем сайте. Сотрудники кафе получают заявку (обычно в качестве электронного письма) и связываются с клиентом по телефону, чтобы уточнить детали и предложить ему более подходящий вариант.

Делаем вывод, что наиболее подходящей системой бронирования столиков и мест в кафе города является веб-приложение, которое было бы способно обеспечить удобный просмотр и редактирование информации о кафе, а также позволяло обрабатывать поступившие заявки.

Вопрос разработки системы бронирования порой решается разными способами и их комбинированием: прием сотрудника на ставку или открытие целого ИТ-отдела для поддержки сайта кафе (ресторана), разовые контакты с разработчиками, самостоятельное освоение веб-технологий или использование веб-сервисов.

В выбранной тематике дипломной работы выполнена разработка и реализация веб-приложения «Table Reserver» для бронирования столиков и мест в кафе, которое в дальнейшем может быть интегрировано в веб-сайт предприятия общественного питания как отдельная веб-страница. В качестве демонстрации работы веб-приложения «Table Reserver» был выполнен его перенос на бесплатную хостинг-площадку с доменным именем (адресом) bronirovaniemest.byethost8.com/index.php.

Бронирование стало неотъемлемой частью жизни во всех государствах мира. Даже в беднейших странах Чад и Сомали бронируют столики в кафе,

барах и ресторанах. В целом автоматизация задачи по бронированию мест выгодна ввиду удобства для клиентов и повышения оперативности работы персонала. Клиенту необходимо перейти на сайт (запустить приложение в браузере), выбрать столик, необходимое количество мест и заполнить такие поля как: фамилия; номер телефона; количество персон и дополнительные пожелания, что благоприятно повлияет на имидж предприятия и уровень лояльности клиентов.

Веб-приложение предусматривает возможность интеграции в веб-сайт любого кафе и внесения изменений в оформление; возможность подключения модуля оплаты бронирования; защиты от спама; внесения дополнительных данных.

В будущем привлекательными стали бы возможности легкой смены ранее забронированного стола, изменения даты или отказа от брони, использования виртуального тура – 3D-панорамы для обзора зала с выбранного стола, СМС-уведомлений. Для прохождения гостем полного цикла бронирования выбора дополнительных услуг в течение нескольких минут рекомендуется интеграция функций бронирования с функциями предварительного оформления заказа блюд и напитков, согласно текущего меню кафе до прихода в заведение.

Исходя из требований к «Table Reserver» можно сделать вывод, что приложение повысит удобство использования услуги бронирования стола в кафе, поскольку возможность бронирования стола и мест станет доступным в любое время суток, оптимизирует работу хостес, повысит привлекательность кафе со стороны клиентов, позволит создать и вести клиентскую базу кафе.

Исходя из вышеуказанного, поставленная цель дипломной работы – достигнута, а соответствующие задачи – выполнены. Проведен анализ предметной области и требований к приложению; выполнено проектирование веб-приложения для бронирования мест в кафе; проведена разработка, тестирование и отладка веб-приложения для бронирования мест в кафе; приложение размещено на бесплатном хостинге.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Автоматизация предприятий, www.soft-servis.ru, дата обращения: 05.03.2017.
2. Автоматизация предприятий индустрии развлечения, tillypad.ru, дата обращения: 05.03.2017.
3. Автоматизация предприятий общественного питания, <http://iiko.ru>, дата обращения: 05.03.2017.
4. Автоматизация ресторанов, <http://www.r-keeper.ru>, дата обращения: 05.03.2017.
5. Автоматизированная система управления, www.edelink.ru, дата обращения: 05.03.2017.
6. Бочкова Р.В., Киселев Г. М. Информационные технологии в экономике и управлении. – Дашков и Ко. 2009. – 272 с.
7. Гущин А. Н. Базы данных [Электронный ресурс] : учебник / А. Н. Гущин. – М.: Директ-Медиа, 2014. – 266 с.
8. Дакетт Д. Разработка и дизайн Web-сайтов / Д. Дакетт. – М.: Эксмо, 2013. – 389 с.
9. Евсеев Д. Web-дизайн в примерах и задачах / Д.А. Евсеев, В.В. Трофимов. – М.: КНОРУС, 2016. – 264 с.
10. Ефимов Е.Н. Информационные системы и технологии в экономике. – Издательский центр «МарТ», Феникс. 2010. – 288 с.
11. Илюшечкин В. М. Основы использования и проектирования баз данных: учеб. Пособие – М. : Юрайт : ИД Юрайт, 2011. – 213 с.
12. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2002. – 304 с.: ил.
13. Клименко Р. Веб-мастеринг на 100% / Р. Клименко. – СПб.: Питер, 2013. – 530 с.
14. Кренке Д.М. Теория и практика построения баз данных: Учебное

пособие. – СПб.: Питер, 2005. – 786 с.

15. Магия для ресторанов, magicsoft.ru, дата обращения: 05.03.2017.

16. Макнейл П. Веб-дизайн. Книга идей веб-разработчика / П. Макнейл. – СПб.: Питер, 2014. – 271 с.

17. Мак-Дональд М. Создание Web-сайта. Недостающее руководство / М. Мак-Дональд. – СПб.: БХВ-Петербург, 2013. – 420 с.

18. Немцова Т.И. Компьютерная графика и web-дизайн: учебное пособие для вузов / Т.И. Немцова, Т.В. Казанкова, А.В. Шнякин; под общ. ред. Л.Г. Гагариной. – М.: ФОРУМ: ИНФРА-М, 2014. – 219 с.

19. Никсон Р. Создаем динамические Web-сайты с помощью PHP, MySQL и JavaScript / Р. Никсон. – СПб.: Питер, 2013. – 206 с.

20. Полякова Л.Н. Основы SQL: Курс лекций. Учебное пособие. – М.: Интернет-Университет Информационных Технологий, 2004. – 368 с.

21. Прием заказа у потребителей, <http://restoraner.ru/priem-zakaza-u-potrebitelej>, дата обращения: 05.03.2017.

22. Программные продукты автоматизации, <http://www.prog-matik.ru/>, дата обращения: 05.03.2017.

23. Профессиональная программа автоматизации, www.averstech.ru, дата обращения: 05.03.2017.

24. Прохоренок Н.А. HTML, JavaScript, PHP и MySQL Джентльменский набор Web-мастера: учебник / Н.А. Прохоренок. – СПб.: БХВ-Петербург, 2015. – 631 с.

25. Седерхольм Д. Пуленепробиваемый веб-дизайн. / Д. Седерхольм, А Струсевич. – СПб.: Питер, 2012. – 304 с.

26. Седерхольм Д. CSS ручной работы. Библиотека специалиста. / Д. Седерхольм, И. Маркотт. – СПб.: Питер, 2013. – 240 с.

27. Советов, Б.Я. Базы данных: теория и практика [Текст] : учеб. для бакалавров : рек. УМО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. – 2-е изд. – М. : Юрайт, 2012. – 464 с.

28. Тестирование сайта, <http://website-ru.net/txt/instrument/testirovanie->

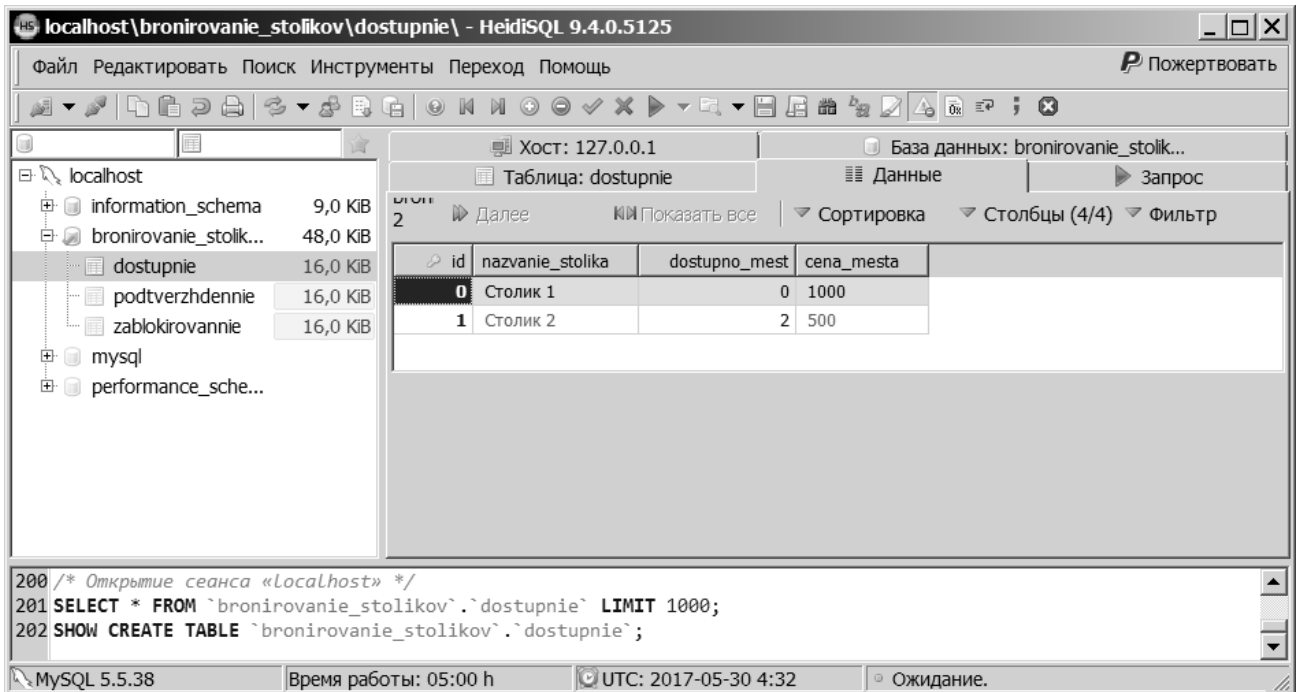
saita.shtml, дата обращения: 05.03. 2017.

29. Электронное меню для предприятий питания,
<http://www.busation.com/elektronnoe-menu>, дата обращения: 05.03.2014.

30. Электронное меню для ресторанов, <http://www.menu55.ru/>, дата
обращения: 05.03. 2017.

ПРИЛОЖЕНИЕ А

Снимки экрана при работе с веб-приложением «Table Reserver» (бронирование столиков и мест в кафе).



localhost\bronirovanie_stolikov\dostupnie - HeidiSQL 9.4.0.5125

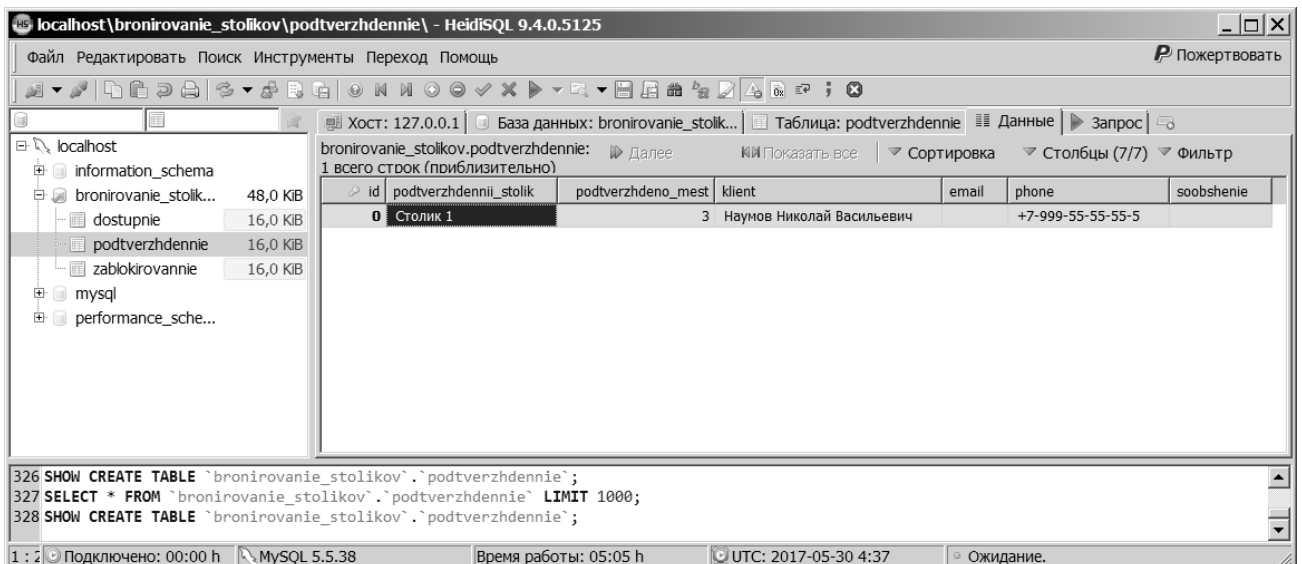
Хост: 127.0.0.1 База данных: bronirovanie_stolk... Таблица: dostupnie

id	nazvanie_stolika	dostupno_mest	cena_mesta
0	Столик 1	0	1000
1	Столик 2	2	500

```
200 /* Открытие сеанса «localhost» */
201 SELECT * FROM `bronirovanie_stolikov`.`dostupnie` LIMIT 1000;
202 SHOW CREATE TABLE `bronirovanie_stolikov`.`dostupnie`;
```

MySQL 5.5.38 Время работы: 05:00 h UTC: 2017-05-30 4:32 Ожидание.

Исходные данные о свободных местах за столиками.



localhost\bronirovanie_stolikov\podtverzhdennye - HeidiSQL 9.4.0.5125

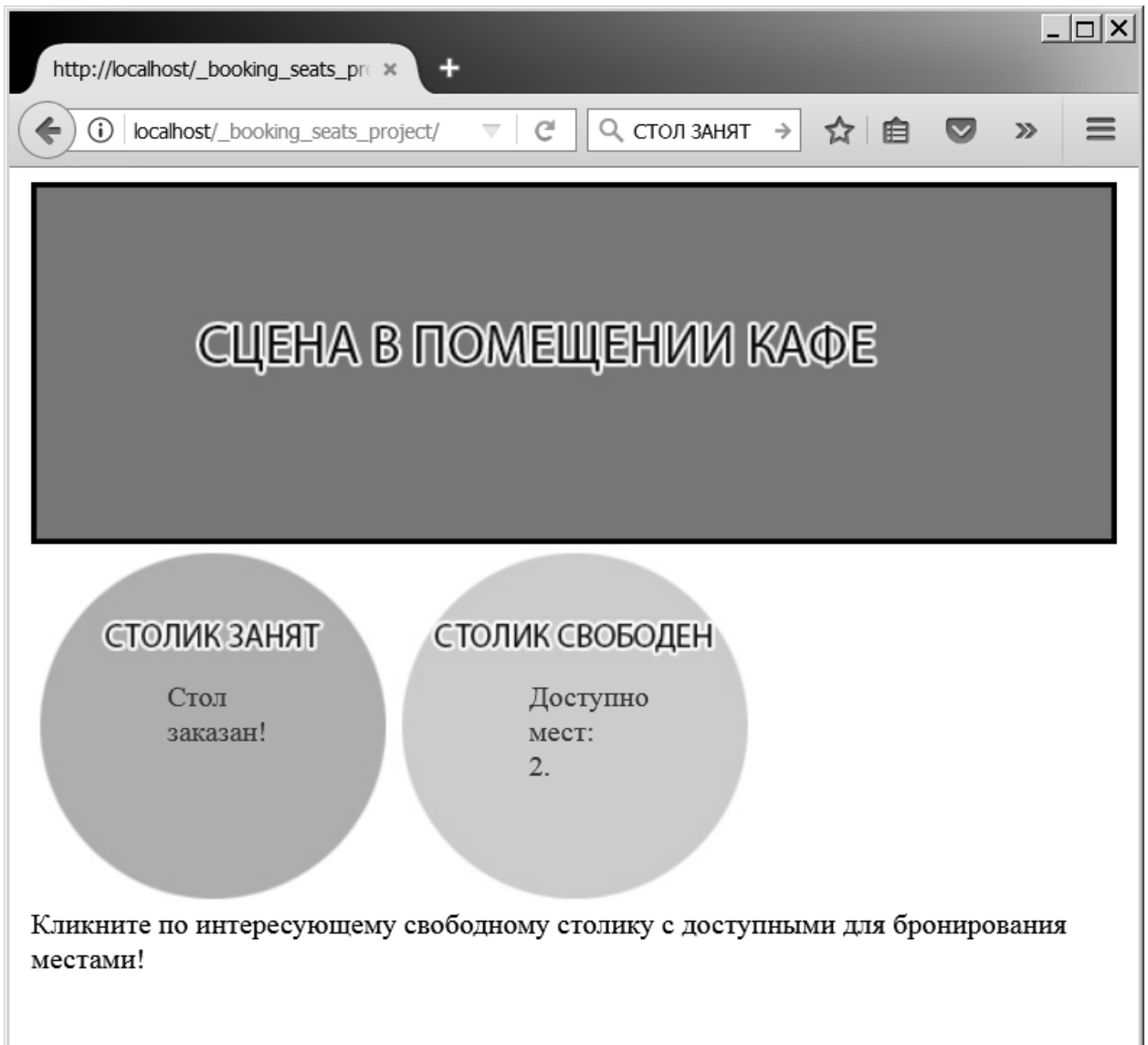
Хост: 127.0.0.1 База данных: bronirovanie_stolk... Таблица: podtverzhdennye

id	podtverzhdennii_stolik	podtverzhdeno_mest	Klient	email	phone	soobshenie
0	Столик 1	3	Наунов Николай Васильевич		+7-999-55-55-5	

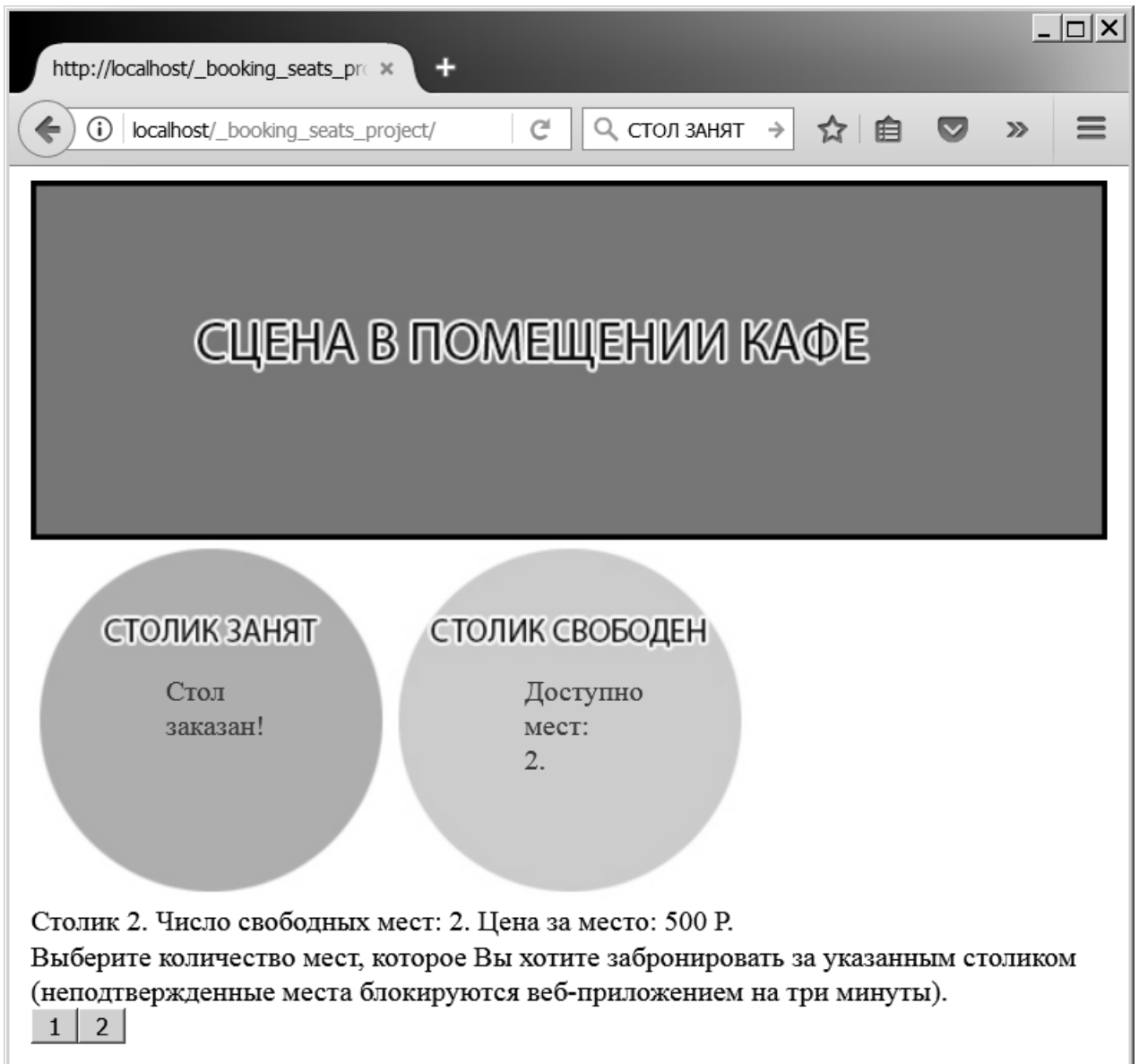
```
326 SHOW CREATE TABLE `bronirovanie_stolikov`.`podtverzhdennye`;
327 SELECT * FROM `bronirovanie_stolikov`.`podtverzhdennye` LIMIT 1000;
328 SHOW CREATE TABLE `bronirovanie_stolikov`.`podtverzhdennye`;
```

1 : 2 Подключено: 00:00 h MySQL 5.5.38 Время работы: 05:05 h UTC: 2017-05-30 4:37 Ожидание.

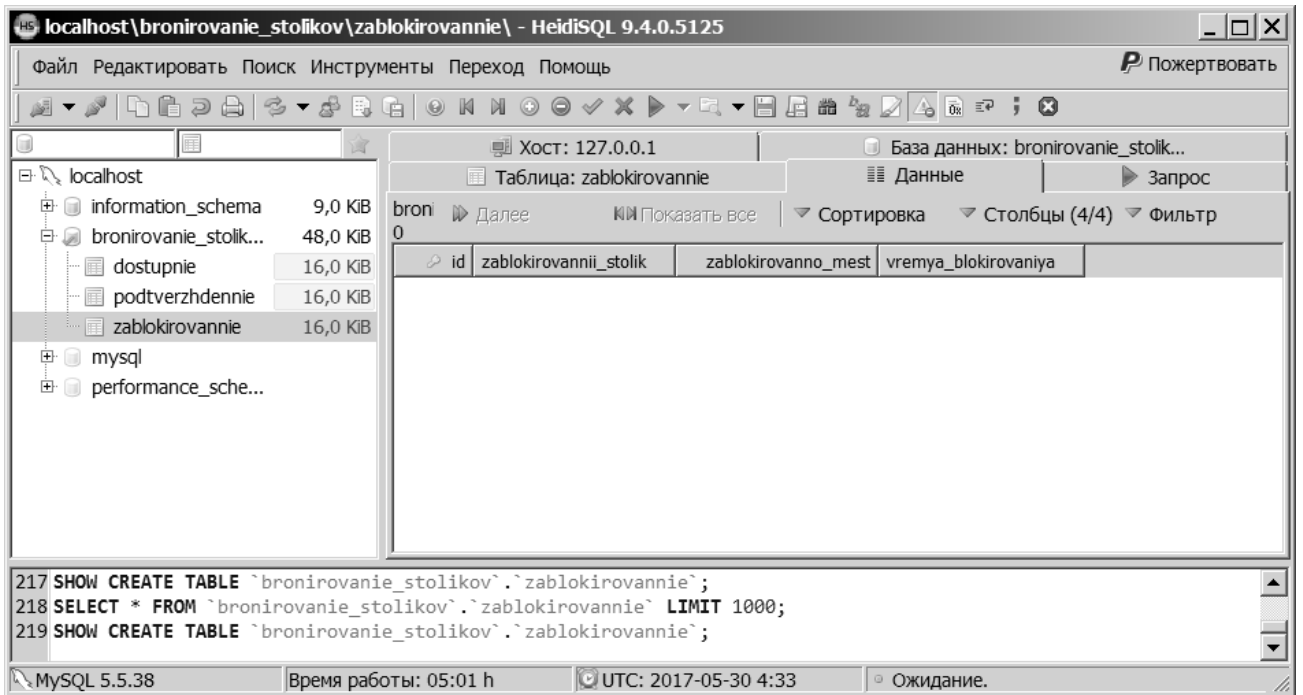
Исходные данные о забронированных местах за столиками.



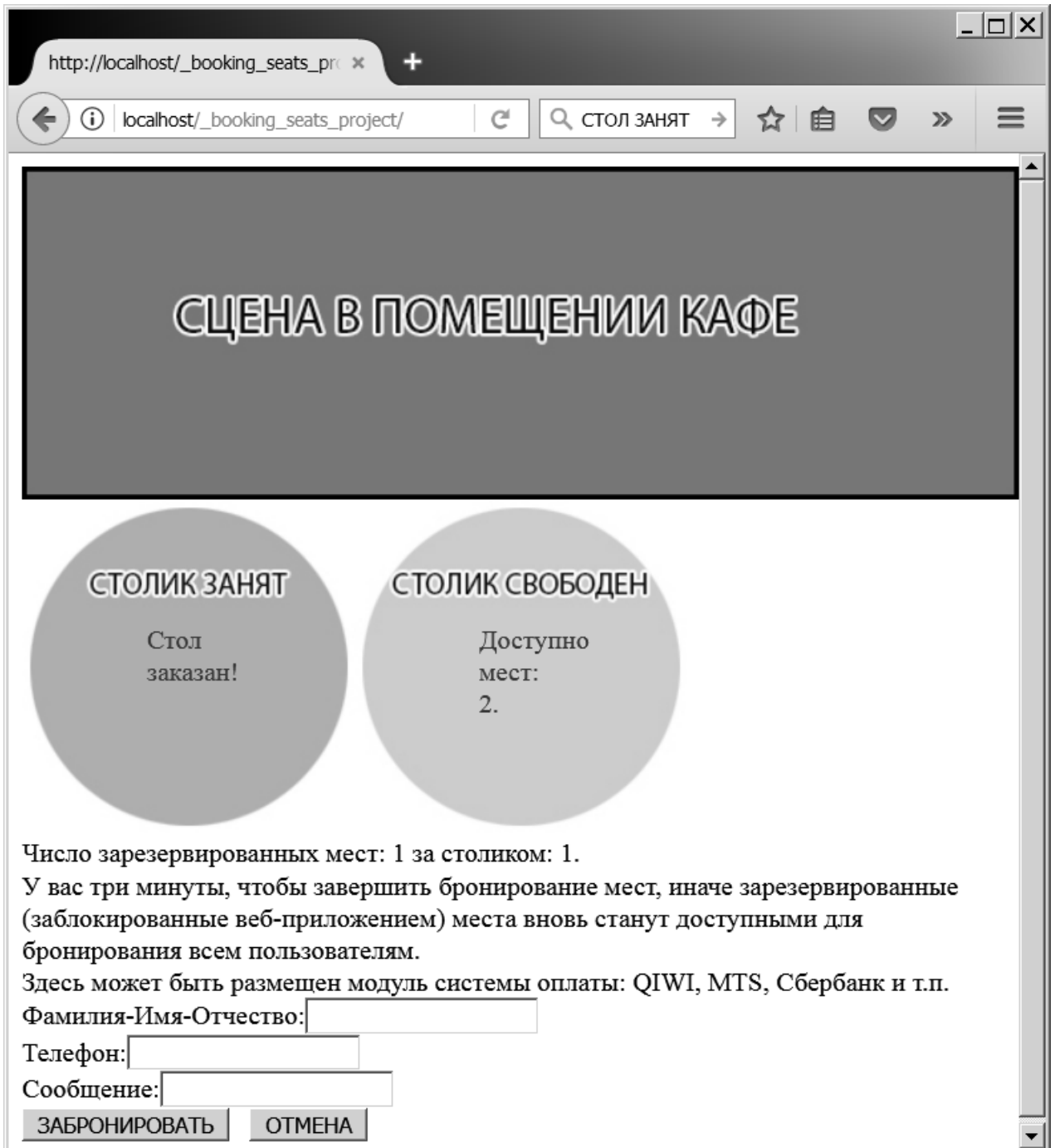
Внешний вид стартовой веб-страницы в соответствии с исходными данными, имеющимися в базе данных.



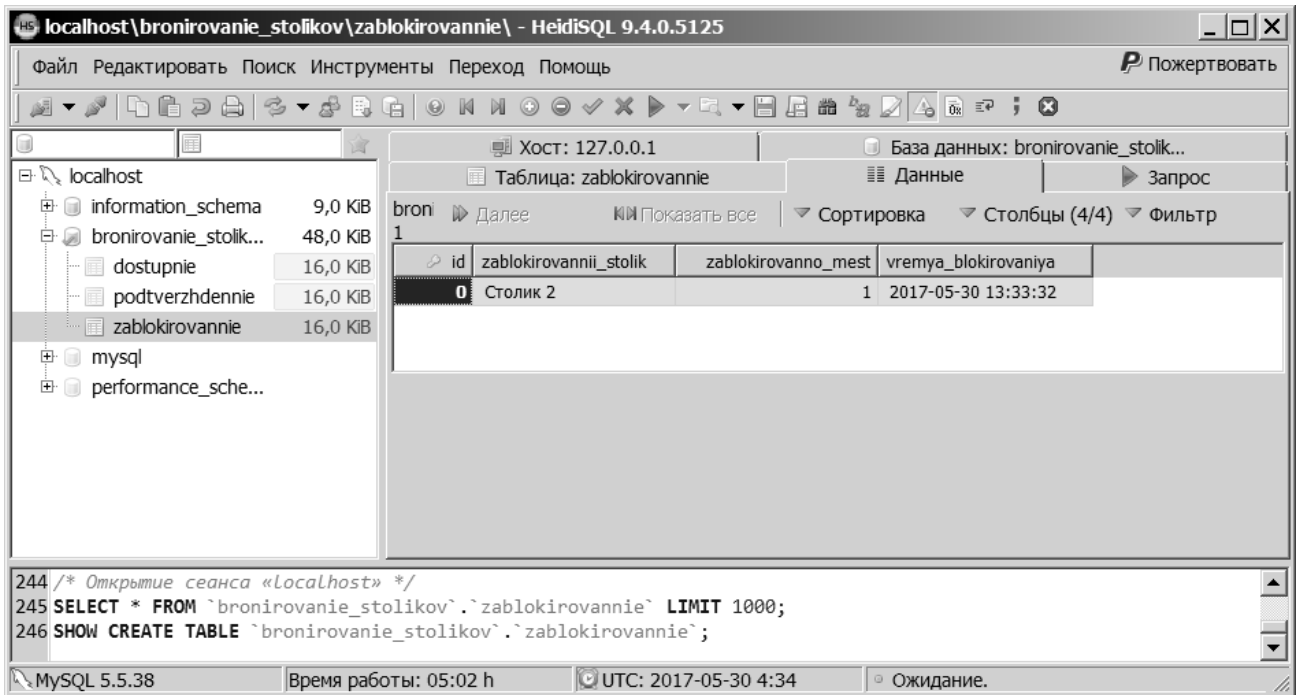
Внешний вид веб-страницы после выбора свободного столика.



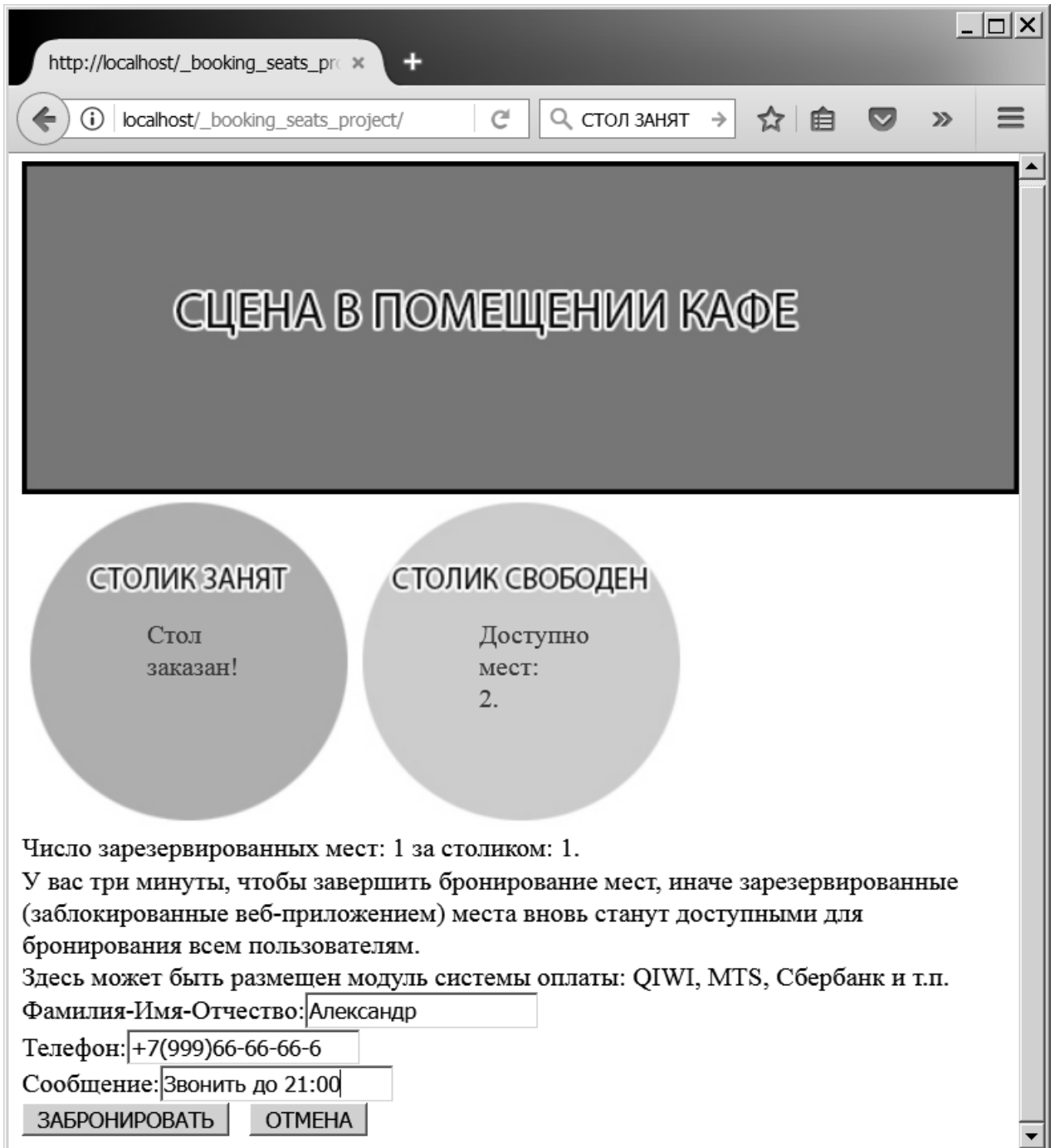
Данные о заблокированных на три минуты местах веб-приложением «Table Reserver» (пользователь еще не указал количество бронируемых мест).



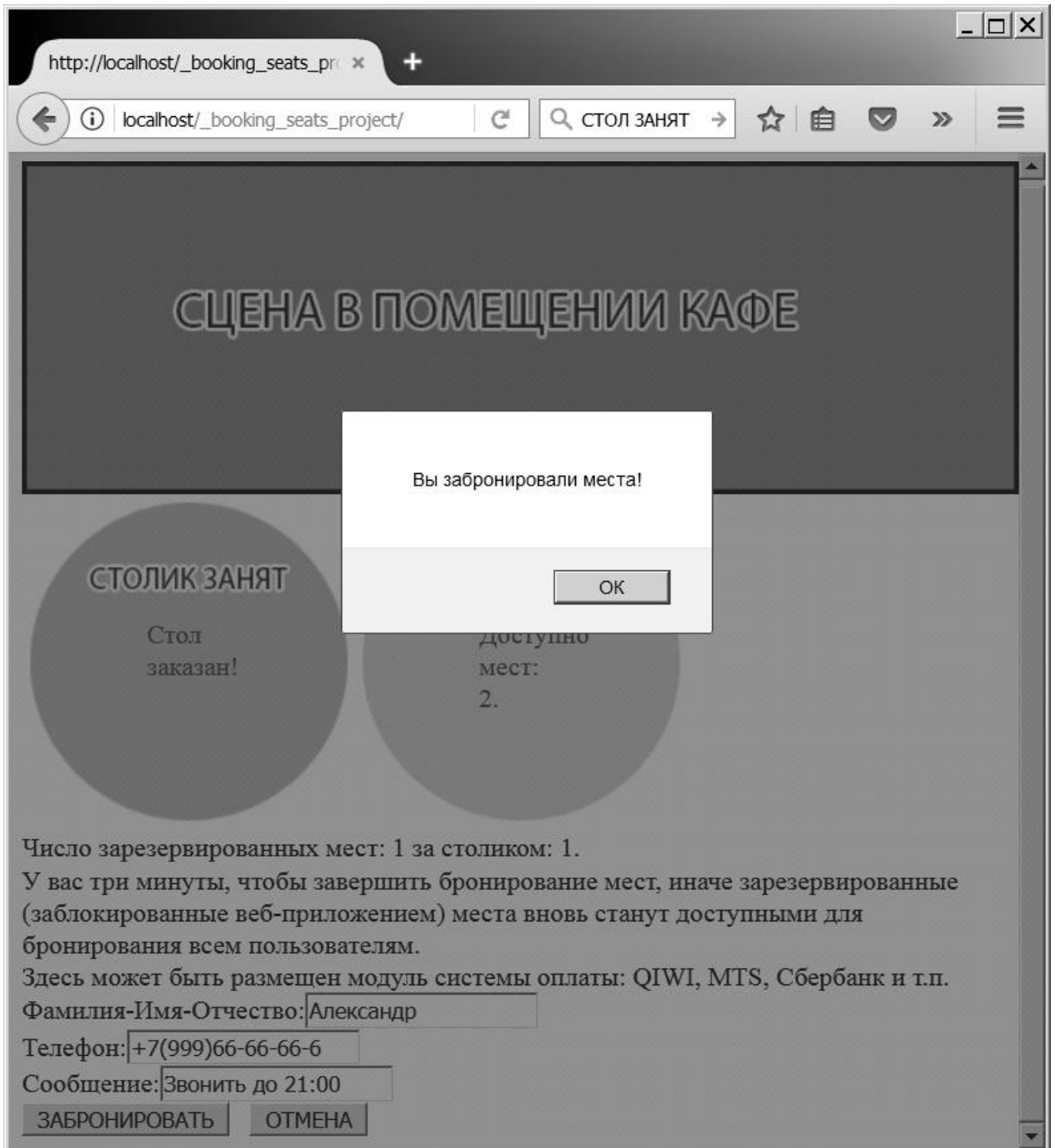
Внешний вид веб-страницы после указания числа бронируемых мест за выбранным столиком.



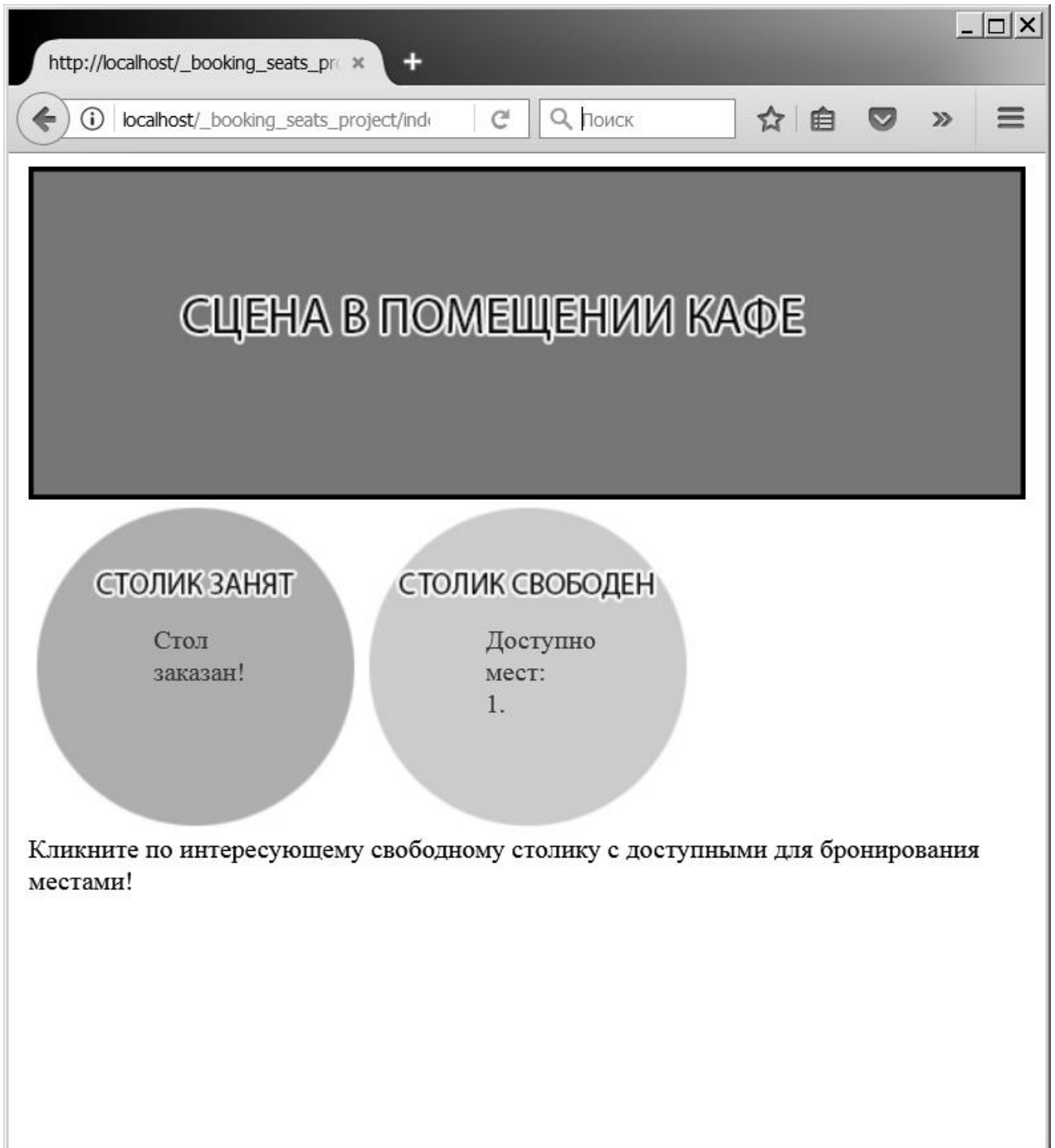
Данные о заблокированных на три минуты местах веб-приложением «Table Reserver» (пользователь указал количество бронируемых мест – одно).



Внешний вид веб-страницы после ввода данных клиентом.



Сообщение о подтверждении бронирования мест клиентом.



Внешний вид стартовой веб-страницы после того как пользователь забронировал одно место за свободным столиком.

ПРИЛОЖЕНИЕ Б

Исходный код веб-приложения «Table Reserver» с комментариями
(бронирование столиков и мест в кафе)

Файл index.php.

```
<?php
**** START Скрипт подключения к БД
// Скрипт подключения (использовать файл)
include ("podklychenie.php");
**** END Скрипт подключения к БД

**** START Очистить забронированные места с "истекшим" периодом
**** Количество времени отведенное на бронирование настроено на три минуты
// Получить список "истекших" мест с 1 стола и исходное количество мест из
другой таблицы
$clean = "SELECT r.*, a.dostupno_mest
FROM zablokirovannii AS r
LEFT JOIN dostupnie AS a ON a.nazvanie_stolika = r.zablokirovannii_stolik
WHERE r.vremya_blokirovaniya < (NOW() - INTERVAL 3 MINUTE)";
$freequery = mysqli_query($stroka_podklycheniya, $clean) or die
(mysqli_error($stroka_podklycheniya));
$num_check = mysqli_num_rows($freequery);
if ($num_check != 0){
while ($row = mysqli_fetch_array($freequery, MYSQLI_ASSOC)){
$dI = $row['zablokirovannii_stolik'];
$dS = $row['zablokirovanno_mest'];
$dIdRow = $row['id'];
$dostupnIznach = $row['dostupno_mest'];
// Вернуть обратно "истекшее" бронирование
$obnovitDostupn = $dostupnIznach + $dS;
// Удалить занятые места из таблицы zablokirovannii
$sql3 = "DELETE FROM zablokirovannii WHERE zablokirovannii_stolik='$dI' LIMIT
1";
$zapros3 = mysqli_query($stroka_podklycheniya, $sql3);
// Обновить БД новыми доступными местами
$sql3 = "UPDATE dostupnie SET dostupno_mest='$obnovitDostupn' WHERE
nazvanie_stolika='$dI' LIMIT 1";
$zapros3 = mysqli_query($stroka_podklycheniya, $sql3);
}
}
**** END Очистить забронированные места с "истекшим" периодом

**** START Получить исходное состояние столиков с местами после "очистки"
// Привести наш вывод к NULL
$chart = "";
// Запрос к столикам с доступными местами
```

```

$sql = "SELECT * FROM dostupnie";
$zapros = mysqli_query($stroka_podklycheniya, $sql) or die
(mysqli_error($stroka_podklycheniya));
// Зациклить и получить все данные
while ($row = mysqli_fetch_array($zapros, MYSQLI_ASSOC)){
// Назначить сведения о столиках к переменным
$id = $row['id'];
$tablenum = $row['nazvanie_stolika'];
$savail = $row['dostupno_mest'];
$scena = $row['cena_mesta'];
// Построить вывод выходных данных
// Вывод для столиков с недоступными местами
if ($savail == 0){
$chart .= '<div class="full">
<div class="numSeats">0 Мест Доступно </div></div>';
} else {
// Отобразить столики с доступными местами – clickable inner div
$chart .= '<div class="available"><div id="tbl_'.$id.'" class="numSeats"
onClick="pokazatMesta(this.id)">'.$savail.' мест доступно</div></div>';
}
} $chart .= '<div class="clear">';
//**** END Получить исходное состояние столиков с местами после "очистки"
?>

```

```

<!-- строим основную веб-страницу для пользователя -->
<!DOCTYPE html>
<html>
<head>
<!-- параметры кодировки -->
<meta charset="UTF-8">
<!-- подключаем файл таблицы стилей на CSS -->
<link rel="stylesheet" href="oformlenie.css">
<!-- подключаем основной скрипт на JavaScript -->
<script src="bronirovanie.js"></script>
</head>
<body>
<!-- делаем обертку -->
<div id="wrapper">
<div id="stage"></div>
<div id="seats"> <?php echo $chart; ?> </div>
<div id="returnData">Кликните по интересующему свободному столику с
доступными для бронирования местами!</div>
</div>
</body>
</html>

```

Файл podklychenie.php.

```

<?php
// файл содержит данные для подключения к БД reservations типа mysql
// связанный файл-скрипт для создания таблиц и внесения исходных данных
sozdat_stoliki.php
// хост, пользователь, пароль, имя БД в двойных кавычках ---> " имя "

```

```

$stroka_podklycheniya = mysqli_connect("localhost", "root", "",
"bronirovanie_stolikov");
// Проверка подключения к БД
if (mysqli_connect_errno()) {
    echo mysqli_connect_error();
    exit();
}
?>

```

Файл bronirovanie.php.

```

<?php
include ("podklychenie.php");
if (isset($_POST['getSeatBtns'])){
    $spots = "";
    $tid = preg_replace('#[^0-9]#', "", $_POST['getSeatBtns']);
    if($tid == ""){
        exit();
    }
    $sql = "SELECT * FROM dostupnie WHERE id='$tid'";
    $zapros = mysqli_query($stroka_podklycheniya, $sql) or die
(mysqli_error($stroka_podklycheniya));
    $bistraya_proverka = mysqli_num_rows($zapros);
    if ($bistraya_proverka != 0){
        while ($row = mysqli_fetch_array($zapros, MYSQLI_ASSOC)){
            $id = $row['id'];
            $stblenum = $row['nazvanie_stolika'];
            $savail = $row['dostupno_mest'];
            $scena = $row['cena_mesta'];

            $stBtn = "";
            $spots .= $stblenum.' Число свободных мест: '.$savail.'. Цена за место: '.$scena.' P.<br
/>';
            $spots .= "Выберите количество мест, которое Вы хотите забронировать за
указанным столиком (неподтвержденные места блокируются веб-приложением на
три минуты).<br />";
            for ($k = 0 ; $k < $savail; $k++){ $k2 = $k+1;
            $stBtn .= '<button id="tbid_'.$id.'_'.$k2.'"
onClick="zablokirovatMesta(this.id)">'.$k2.'</button>';
            }
            $spots .= "$stBtn<br />";
        }//while
    } else {
        $spots = "Извините, места больше недоступны за данным столиком.";
    }
    echo $spots;
    exit();
}
if (isset($_POST['reserve'])){
    $spots = "";
    $tid = preg_replace('#[^0-9]#', "", $_POST['reserve']);
    $num = preg_replace('#[^0-9]#', "", $_POST['num']);
    if($tid == "" || $num == ""){

```



```

$name = preg_replace('#[^a-z0-9 ]#i', '', $_POST['n']);
$tableNumber = preg_replace('#[^a-z0-9 ]#i', '', $_POST['tn']);
$numSeats = preg_replace('#[^0-9]#', '', $_POST['ns']);
if($rid == "" || $name == "" || $tableNumber == "" || $numSeats == ""){
    exit();
} //if
} //if
$sql = "SELECT id FROM zablokirovannie WHERE id='$rid' LIMIT 1";
$zapros = mysqli_query($stroka_podklycheniya, $sql) or die
(mysqli_error($stroka_podklycheniya));
$bistraya_proverka = mysqli_num_rows($zapros);
if ($bistraya_proverka != 1){
    $sql = "SELECT id, dostupno_mest FROM dostupnie WHERE
nazvanie_stolika='$tableNumber' AND dostupno_mest>='$numSeats' LIMIT 1";
    $zapros = mysqli_query($stroka_podklycheniya, $sql) or die
(mysqli_error($stroka_podklycheniya));
    $bistraya_proverka2 = mysqli_num_rows($zapros);
    if ($bistraya_proverka2 == 0){
        $confirmedStatus = "false";
        $otvet = "Бронирование истекло, обновите страницу";
        echo "$confirmedStatus|$otvet";
        exit();
    } else {
        while ($row = mysqli_fetch_array($zapros, MYSQLI_ASSOC)){
            $id = $row['id'];
            $savail = $row['dostupno_mest'];
        }
        $savailNow = $savail - $numSeats;
        $sql = "UPDATE dostupnie SET dostupno_mest='$savailNow' WHERE id='$id' LIMIT
1";
        $zapros = mysqli_query($stroka_podklycheniya, $sql);
        $sql = "INSERT INTO
podtverzhdennie(podtverzhdennii_stolik,podtverzhdeno_mest,klient) VALUES
('$tableNumber','$numSeats','$name)";
        $zapros = mysqli_query($stroka_podklycheniya, $sql);
        $confirmedStatus = "true";
        $otvet = 'необходимо в случае ошибки';
        echo "$confirmedStatus|$otvet";
        exit();
    }
} else {
    $sql = "INSERT INTO
podtverzhdennie(podtverzhdennii_stolik,podtverzhdeno_mest,klient) VALUES
('$tableNumber','$numSeats','$name)";
    $zapros = mysqli_query($stroka_podklycheniya, $sql);
    $sql = "DELETE FROM zablokirovannie WHERE id='$rid' LIMIT 1";
    $zapros = mysqli_query($stroka_podklycheniya, $sql);
    $otvet = 'необходимо в случае ошибки';
    $confirmedStatus = "true";
    echo "$confirmedStatus|$otvet";
    exit();
}

```

```

}
if (isset($_POST['clearRes'])) {
    $rid = preg_replace('#[^0-9]#', '', $_POST['clearRes']);
    $clean = "SELECT r.*, a.dostupno_mest FROM zablokirovannii AS r LEFT JOIN
dostupnie AS a ON a.nazvanie_stolika = r.zablokirovannii_stolik WHERE r.id = '$rid'";
    $freequery = mysqli_query($stroka_podklycheniya, $clean) or die
(mysqli_error($stroka_podklycheniya));
    $num_check = mysqli_num_rows($freequery);
    if ($num_check != 0) {
        while ($row = mysqli_fetch_array($freequery, MYSQLI_ASSOC)) {
            $dI = $row['zablokirovannii_stolik'];
            $dS = $row['zablokirovanno_mest'];
            $dIdRow = $row['id'];
            $dostupnIznach = $row['dostupno_mest'];
            $sobnovitDostupn = $dostupnIznach + $dS;
            $sql3 = "DELETE FROM zablokirovannii WHERE zablokirovannii_stolik='$dI' LIMIT
1"; $zapros3 = mysqli_query($stroka_podklycheniya, $sql3);
            $sql3 = "UPDATE dostupnie SET dostupno_mest='$sobnovitDostupn' WHERE
nazvanie_stolika='$dI' LIMIT 1";
            $zapros3 = mysqli_query($stroka_podklycheniya, $sql3);
        } //while
    } //if
} //if
?>

```

Файл бронирование.js.

```

// START исходное состояние основных переменных vars
var obrabotkaSceni = "open";
var reservationId = "open";
// END исходное состояние основных переменных vars

// **** START получить кнопки для доступных мест
function pokazatMesta(tbl) {
    // Если бронирование не завершено (reservation), отменить бронирование
    if (obrabotkaSceni == "closed") {
        otmenitBronirovanie(reservationId);
    }
    // tbl дает нам -> tblid_3
    // Нам нужна только (3) часть, поэтому разбиваем (split)
    var a = tbl.split("_");
    // Добавляем символ-заполнитель к seatBtns div
    document.getElementById("returnData").innerHTML = 'Проверка доступности ...';

    // Для доп. сведений
    // см. http://www.developphp.com/view.php?tid=1185
    var hr = new XMLHttpRequest();
    // PHP скрипт обработки url
    var url = "bronirovanie.php";
    hr.open("POST", url, true);
    hr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    hr.onreadystatechange = function() {
        if (hr.readyState == 4 && hr.status == 200) {

```



```

var return_data = hr.responseText;
// Отправить кнопки мест к div id seatBtns
document.getElementById("returnData").innerHTML = return_data;
}
}
// Отправить getSeatBtns=3 к PHP скрипту обработки
hr.send("getSeatBtns="+a[1]);
}
// **** END получить кнопки для доступных мест

// **** START зарезервировать (заблокировать) количество мест
function zablokirovatMesta(numseats){
    obrabotkaSceni = "closed";
    // numseats дает нам -> tbid_2_3
    // 2 это id для таблицы (id не имя таблицы)
    // 3 это число мест, которые хотят
    // Разбить наши данные (Split) в _
    var a = numseats.split("_");
    // Добавить текст пока запрос обрабатывается к buyNow div
    document.getElementById("returnData").innerHTML = 'Проверка доступности ...!';
    // Для доп. сведений
    // см. http://www.developphp.com/view.php?tid=1185
    var hr = new XMLHttpRequest();
    // PHP скрипт обработки url
    var url = "bronirovanie.php";
    hr.open("POST", url, true);
    hr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    hr.onreadystatechange = function() {
    if(hr.readyState == 4 && hr.status == 200) {
    var return_data = hr.responseText.split("|");
    reservationId = return_data[1];
    // Вернуть отправленные данные к buyNow div
    document.getElementById("returnData").innerHTML = return_data[0];
    }
    }
    // Отправить reserve=2&num=3 к скрипту обработки PHP
    hr.send("reserve="+a[1]+"&num="+a[2]);
}
// **** END START зарезервировать (заблокировать) количество мест

// **** START регистрация/бронирование мест
function podtverditMesta(){
    // Получить данные из формы
    var name = document.getElementById("name").value;
    var tnum = document.getElementById("tableNumber").value;
    var nseats = document.getElementById("numSeats").value;
    var rid = document.getElementById("reserveID").value;
    if(name == "" || tnum == "" || nseats == "" || rid == ""){
    return false;
    }
    // Компилировать данные из формы и отправить их к обработке
    var confData = "confirm="+rid+"&n="+name+"&tn="+tnum+"&ns="+nseats;

```

```

// Для доп. сведений
// см. http://www.developphp.com/view.php?tid=1185
var hr = new XMLHttpRequest();
// PHP скрипт обработки url
var url = "bronirovanie.php";
hr.open("POST", url, true);
hr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
hr.onreadystatechange = function() {
if(hr.readyState == 4 && hr.status == 200) {
var return_data = hr.responseText.split("|");
// Пользователь слишком долго ждал и кто-то другой забрал места
if (return_data[0] == "false"){
// Вывести сообщение
document.getElementById("returnData").innerHTML = return_data[1];
var obrabotkaSceni = "open";
var reservationId = "open";
} else {
// Места получили успешно
var obrabotkaSceni = "open";
var reservationId = "open";
// Выдать сообщение об успешном бронировании
alert ("Вы забронировали места");
// Перезагрузить страницу
window.location = 'index.php';
}
}
}
// Отправить наши откомпилированные данные
hr.send(confData);
}
// **** регистрация/бронирование мест

// START кнопка отмены бронирования
function otmenitBronirovanie(resId){
// Для доп. сведений
// см. http://www.developphp.com/view.php?tid=1185
var hr = new XMLHttpRequest();
// PHP скрипт обработки url
var url = "bronirovanie.php";
hr.open("POST", url, true);
hr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
// Отправить идентификатор бронирования
hr.send("clearRes="+resId);
window.location = 'index.php';
}
// END кнопка отмены бронирования

```

Файл oformlenie.css.

```

/* файл стилей для оформления основной пользовательской страницы */
/* селектор по ID */
#wrapper {

```

```

width: 600px;
margin-right: auto;
margin-left: auto;
}
/* сцена */
/* загружаем изображение файла сцены (200x600) "без повторов" */
/* селектор по ID */
#stage {
background-image: url(stage.jpg);
background-repeat: no-repeat;
height: 200px;
width: 600px;
}
/* свободный столик */
/* загружаем изображение файла столика (200x200): позиционирование – слева */
/* селектор по классу */
.available {
background-image: url(openTable.jpg);
background-repeat: no-repeat;
float: left;
height: 200px;
width: 200px;
}
/* столик занят полностью */
/* загружаем изображение файла столика (200x200): позиционирование – слева */
/* селектор по классу */
.full {
background-image: url(tableFull.jpg);
background-repeat: no-repeat;
float: left;
height: 200px;
width: 200px;
}
/* селектор по классу */
.clear {
clear: both;
}
/* количество мест: цвет, отсут, высота, ширина */
/* селектор по классу */
.numSeats {
color: #03F;
margin: 75px;
height: 50px;
width: 50px;
}

```

Файл sozdat_stoliki.php.

```

<?php include_once("podklychenie.php");
// файл для создания таблиц в базе данных bronirovanie_stolikov (бронирование мест
в кафе)
// создание таблицы dostupnie (доступные места: код, номер столика, доступно
мест, цена)

```

```

// id, nazvanie_stolika, avail, cena_mesta
$stolik_dostupn =
"CREATE TABLE IF NOT EXISTS dostupnie ( id int(11) NOT NULL
AUTO_INCREMENT, nazvanie_stolika varchar(50) NOT NULL, dostupno_mest int(11)
NOT NULL, cena_mesta varchar(20) NOT NULL, PRIMARY KEY (id) )";
$zapros = mysqli_query($stroka_podklycheniya, $stolik_dostupn);
if ($zapros === TRUE) {
echo "<h3>Таблица dostupnie создана :)</h3>";
} else {
echo "<h3>Ошибка создания таблицы dostupnie :( </h3>";
}
// создание таблицы podtverzhdennie (занятые места пользователями: код, номер
столика, заказчик, эл. почта, телефон, сообщение)
// id, podtverzhdennii_stolik, podtverzhdeno_mest, klient, email, phone, soobshenie
$stolik_podtverzhden =
"CREATE TABLE IF NOT EXISTS podtverzhdennie ( id int(11) NOT NULL
AUTO_INCREMENT, podtverzhdennii_stolik varchar(50) NOT NULL,
podtverzhdeno_mest int(11) NOT NULL, klient varchar(255) NOT NULL, email
varchar(255) NOT NULL, PRIMARY KEY (id) )";
$zapros = mysqli_query($stroka_podklycheniya, $stolik_podtverzhden);
if ($zapros === TRUE) {
echo "<h3>Таблица podtverzhdennie создана :)</h3>";
} else {
echo "<h3>Ошибка создания таблицы podtverzhdennie :( </h3>";
}
// создание таблицы zablokirovannie (бронирование: код, номер столика, количество
мест, время)
// id, zablokirovannii_stolik, zablokirovano_mest, vremya_blokirovaniya
$stolik_zablokirovan =
"CREATE TABLE IF NOT EXISTS zablokirovannie ( id int(11) NOT NULL
AUTO_INCREMENT, zablokirovannii_stolik varchar(50) NOT NULL,
zablokirovano_mest int(11) NOT NULL, vremya_blokirovaniya datetime NOT NULL,
PRIMARY KEY (id) )";
$zapros = mysqli_query($stroka_podklycheniya, $stolik_zablokirovan);
if ($zapros === TRUE) {
echo "<h3>таблица zablokirovannie создана :)</h3>";
} else {
echo "<h3>Ошибка создания таблицы zablokirovannie :( </h3>";
}
// запись исходных данных в таблицу dostupnie (доступные места: код, номер
столика, доступно мест, цена)
// данные обязательно должны быть внесены в базу данных bronirovanie_stolikov
// пользователь сможет видеть на веб-странице свободные-занятые столики,
свободные места за каждым столиком и цену бронирования за место
$stolik_zapisat = "INSERT INTO dostupnie
(nazvanie_stolika,dostupno_mest,cena_mesta) VALUES ('Столик 1', 10, '500.00'),
('Столик 2', 8, '500.00'), ('VIP-столлик 1', 5, '1200.00)";
$zapros = mysqli_query($stroka_podklycheniya, $stolik_zapisat);
//создание таблицы dostupnie (доступные места)
if ($zapros === TRUE) {
echo "<h3>Исходные данные о столиках в заведении записаны в БД :)</h3>";
} else {

```

```
echo "<h3>Ошибка записи исходных данных :(</h3>"; }  
?>
```

ПРИЛОЖЕНИЕ В

Данные от хостинг-провайдера <https://byet.host> и работа с хостингом.

Адрес (URL): <http://bronirovaniemest.byethost8.com>.

Адрес панели управления: cpanel.byethost8.com.

Логин в панель управления: b8_20169081.

Пароль в панель управления: *****.

Логин MySQL: b8_20169081.

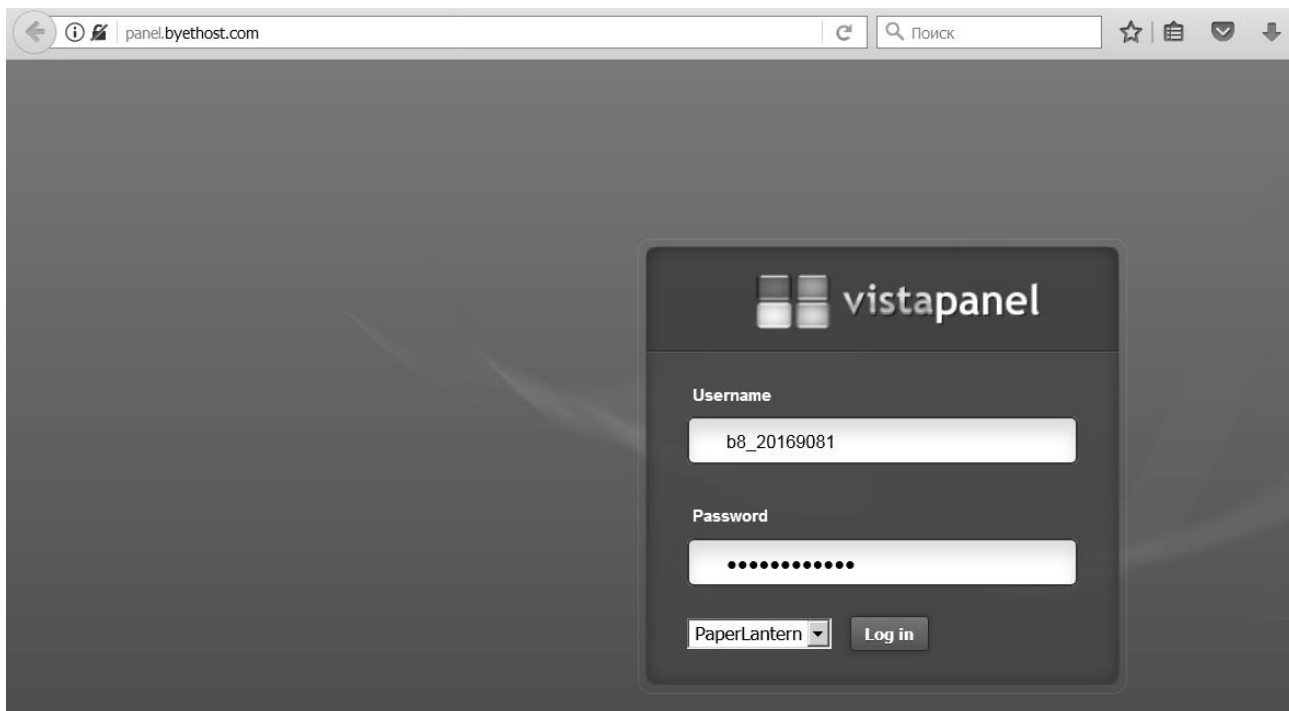
Пароль MySQL: *****.

Имя хоста MySQL: sql111.byethost8.com.

Логин FTP: b8_20169081.

Пароль FTP: *****.

Сервер FTP: [ftp.byethost8.com](ftp://ftp.byethost8.com).



Вход в панель управления хостингом.

The screenshot shows the Vistapanel web hosting control panel. The browser address bar displays `panel.byethost.com/panel/index.php?id=a8a3c51c9b1856a762d3b1af71ce13aa21e0ced7`. The page header includes the Vistapanel logo and the user ID `BB_20169081` with a `LOG OUT` link. A search bar is located at the top left with the placeholder text "Find functions quickly by typing here." The main content area is divided into three sections: **PREFERENCES**, **FILES**, and **DATABASES**. The **PREFERENCES** section includes links for "Change Password", "Change Language", "Update Contact Email", "Getting Started", "Account Upgrades", and "Account Settings". The **FILES** section includes "Online File Manager", "Directory Privacy", "Disk Usage", "FTP Accounts", "Free FTP Software", and "Backups". The **DATABASES** section includes "phpMyAdmin", "MySQL Databases", and "Remote MySQL". On the right side, there is a **STATISTICS** table showing various hosting metrics.

STATISTICS	
Plan:	Free Hosting
FTP accounts:	1 / 1
Sub-Domains:	0 / Unlimited
Add-on Domains:	0 / Unlimited
Parked Domains:	0 / Unlimited
MySQL Databases:	0 / Unlimited
Disk Quota:	Unlimited
Disk Space Used	0 MB
Disk Free	Unlimited
Inodes Used	0 % (0 of 30019)
Bandwidth:	Unlimited
Bandwidth used:	0 MB
Bandwidth remaining:	Unlimited
Daily Hits Used	0 % (0 of 50000)

Панель управления хостингом.

The screenshot shows a web browser with the address bar containing `bronirovaniemest.byethost8.com/index.php`. The browser interface includes standard navigation buttons (back, forward, home, search) and a search bar with the text "Поиск".

The screenshot shows a web application interface for table reservation. At the top, a large dark gray box contains the text "СЦЕНА В ПОМЕЩЕНИИ КАФЕ" (Cafe Room Scene). Below this, there are two circular buttons. The left button is labeled "СТОЛИК ЗАНЯТ" (Table Occupied) and contains the text "Стол заказан!" (Table ordered!). The right button is labeled "СТОЛИК СВОБОДЕН" (Table Available) and contains the text "Доступно мест: 1." (Available seats: 1.).

Веб-приложение «Table Reserver» в сети Интернет.