

Содержание

ВВЕДЕНИЕ	2
1 ПРИМЕНЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В РЕСТОРАННОМ БИЗНЕСЕ	4
1.1 Современное состояние ресторанного бизнеса	4
1.2 Место и роль сайта в деятельности компании	8
1.3 Актуальность использования информационных технологий в ресторанах.....	155
2 РЕАЛИЗАЦИЯ ПРОЕКТА САЙТА РЕСТОРАНА	222
2.1 Выбор информационной системы для организации процесса создания, редактирования и управления содержимым сайта.....	222
2.2 Выбор средств разработки интернет-сайта ресторана	300
2.2.1 Выбор хостинга	300
2.2.2 Обоснование проектных решений по программному обеспечению.....	33
2.3 Разработка карты сайта и шаблона дизайна.....	433
3 ОПИСАНИЕ СОЗДАННОГО САЙТА РЕСТОРАНА	511
3.1 Программная реализация основных разделов сайта	51
3.2 Описание порядка реализации виртуальной экскурсии	62
ЗАКЛЮЧЕНИЕ	65
СПИСОК ЛИТЕРАТУРЫ.....	67
Приложение 1. Исходный код файла «panorama_custom.js»:.....	711
Приложение 2. Исходный код файла «index.html» (подключение панорамы в HTML-коде страницы):.....	788
Приложение 3. Список условных обозначений.....	79

ВВЕДЕНИЕ

В сложившейся ситуации возрастающей конкуренции, которая с каждым годом всё возрастает на рынке ресторанного бизнеса, а также увеличения взыскательности клиентов к качеству предоставляемых услуг, особое внимание следует уделять мероприятиям по привлечению и удержанию клиентов предприятий общественного питания. Продвижение услуг ресторана рассматривается как комплекс мер по доведению информации до конечного потребителя о ресторане, его достоинствах и преимуществах. Необходимо привлечь внимание клиента, убедить его воспользоваться предлагаемыми услугами.

В настоящий момент работа ресторана связана с наличием большого количества информации о меню, личных данных сотрудников, счетах и так далее. Кроме того, предприятиям общественного питания необходимо решать такие важные задачи, как увеличение прибыли, управление изменениями в ресторане, вплоть до смены концептуальной направленности, к примеру, на основании результатов опроса клиентов ресторана.

Актуальность работы продиктована тем, что подавляющее большинство предприятий общественного питания в качестве информационной поддержки своей деятельности использует web-сайты, основной задачей которых является предоставление информации о концепции предприятия, о меню, о действующих специальных предложениях и прочем. Одновременно с этим администрация ресторана сможет влиять на степень лояльности клиентов, внедряя на сайт дополнительные сервисы. Отсюда вывод, что наличие своей визитной карточки в сети Интернет позволит обеспечить качественную информационную поддержку любого предприятия общественного питания.

Объектом данной выпускной квалификационной работы является ресторан «Бирхоф», который в настоящее время не имеет своего сайта.

Предмет исследования: информационные технологические процессы, которые используются в ресторанной деятельности.

Цель выпускной квалификационной работы: разработка сайта для ресторана «Бирхоф».

Задачи выпускной квалификационной работы:

1. Анализ текущей ситуации в направлении развития ресторанного бизнеса.
2. Анализ применяемых информационных технологий в предприятиях общественного питания.
3. Анализ роли сайтов в заведениях общественного питания.
4. Обоснование необходимости разработки сайта для ресторана «Бирхоф».
5. Проектирование и разработка сайта для ресторана «Бирхоф».

Практическая значимость заключается в возможности использования спроектированного сайта для информационной поддержки деятельности ресторана «Бирхоф», в первую очередь, для завоевания лояльности посетителей ресторана.

Данная работа состоит из трех глав. В первой описываются теоретические аспекты ресторанной отрасли, во второй – информационные технологии, которые используются для автоматизации предприятий общественного питания, а третья глава посвящена описанию разработки сайта ресторана.

1 ПРИМЕНЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В РЕСТОРАННОМ БИЗНЕСЕ

1.1 Современное состояние ресторанного бизнеса

Ресторанный бизнес – это сфера предпринимательской деятельности, основанная на оказании услуг, целью которых является удовлетворение потребностей в питании на сервисной основе и как следствие - получение прибыли[4].

Присутствие активной конкурентной борьбы на рынке ресторанных услуг вынуждает менеджмент предприятий выполнять определенные требования, например, такие, как необходимость увеличения производительности и усиление конкурентной позиции, что неминуемо сопряжено с внедрением в повседневную жизнь современных форм обслуживания. В основе таких форм - анимационные программы, которые помогают увеличить пропускную способность залов предприятий общественного питания.

Общественное питание – это сфера оказания услуг, а значит главная задача данного сектора - сделать людей счастливыми на то время, что они находятся в ресторане (кафе, баре или другом типе предприятия) [9].

Предпринимательство в ресторане успешно, когда главной целью является удовлетворение потребностей посетителей. Владелец ресторана и коллектив заинтересованы в формировании благоприятных условий в своем заведении для повышения его популярности. Ведь при большой конверсии посещаемости значительно увеличится сумма среднего чека на одного посетителя, что, в свою очередь, повлияет на увеличение объема продаж.

Деятельность ресторана можно представить в виде системы, состоящей из двух подразделений: производство и сбыт продукции. Главные цели этой системы - удовлетворить нужды конкретных потребителей и получить прибыль на основе исследования и прогнозирования рынка. Для достижения поставленных целей изучается внутренняя и внешняя среды предприятия, разрабатывается стратегия и тактика поведения на рынке при помощи

маркетинговых техник. В основе данных техник лежит определенный комплекс мер по улучшению качества продукции, по расширению ассортимента услуг, по изучению потребителей и конкурентов, по обеспечению ценовой политики, по формированию, оптимизации, рекламе, каналов товародвижения и организации сбыта [1].

Производственно-сбытовая деятельность ресторана начинается с сегментирования рынка, то есть с разбивки его на определенные группы потребителей, для каждой из которых должны быть предложены различные виды продукции и услуг. От того, насколько правильно выбраны сегменты рынка, во многом зависит успех ресторана в конкурентной борьбе. Заведению сферы ресторанного бизнеса следует решить, сколько сегментов рынка оно в состоянии охватить, и сосредоточиться на самых выгодных [13].

На сегодняшний день индустрия общественного питания в России представляет собой большое количество предприятий с разным уровнем сервиса, качеством продукции и разнообразием используемого оборудования.

До 30-х годов XX века отрасль не получала такого отклика, так как внимание страны было направлено на индустриализацию и организацию колхозного хозяйства. В послевоенные годы сеть предприятий общественного питания начала стремительно расширяться: увеличился ассортимент блюд, улучшилось обеспечение предприятий технологическим и холодильным оборудованием, повысилась культура обслуживания населения. Определенной особенностью советской системы общественного питания являлось одномоментное обслуживание внушительных масс населения в промышленности и сельском хозяйстве. Отрицательным моментом данной системы была вынужденная ограниченность в ассортименте. В этот период предприятия общественного питания активно занимались выездной (вывозной) реализацией готовой продукции, что также создавало определенные трудности [2].

В конце 80-х годов XX века многие предприятия общественного питания начали трудиться на основе хозяйственного расчета, т.е. иметь самостоятельный баланс. Стали открываться кооперативные предприятия общественного питания, хотя большая часть из них еще оставалась в составе государственных торговых организаций.

В период перехода на рыночные отношения в начале 90-х годов XX века стали формироваться небольшие, самокупаемые компании с высоким уровнем сервиса и качества приготовления пищи.

В современных условиях хозяйствования, согласно ГОСТ Р 50647-94 «Общественное питание. Термины и определения», утвержденного Постановлением Госстандарта России от 21.02.1994 № 35, общественное питание представляет собой совокупность предприятий различных организационно-правовых форм и граждан-предпринимателей, занимающихся производством, реализацией и организацией сбыта кулинарной продукции [2].

Кулинарной продукцией признается совокупность блюд, кулинарных изделий и кулинарных полуфабрикатов. В свою очередь, кулинарное изделие заданной формы из теста, в основном с фаршем, считается мучным кулинарным изделием (пироги, чебуреки, беляши, курники). Поэтому об оказании услуг общественного питания речь идет в том случае, если кулинарная продукция (в том числе пироги, лепешки, блины) была изготовлена самим предприятием общественного питания.

Отличительной характерной чертой предприятия общественного питания является сочетание в нем функций производства и потребления. Если основное количество предприятий смежных отраслей ограничивает себя выполнением лишь одной или двух функций, то предприятия общественного питания выполняют три взаимосвязанные функции: производство кулинарной продукции, реализация кулинарной продукции, организация ее потребления.

Исходной считается функция производства, затраты труда на которую составляют 70-90 % всех затрат труда в отрасли. В процессе производства создаётся новый продукт. Продукция собственного производства поступает в реализацию с новыми потребительскими качествами, характеристиками и возросшей стоимостью. Питание - это форма потребления [3].

В основе многообразия предприятий общественного питания лежит их классификация, в базу которой заложены следующие признаки: ассортимент реализуемого продукта и виды услуг для населения, объем и разновидность услуг, степень удобства и качества обслуживания, формы организации производства, цикличность функционирования в течение 12 месяцев, уровень мобильности (место дислокации), способ размещения предприятий в зданиях.

Чаще всего предприятия общественного питания различаются по характеру производства, по ассортименту выпускаемой продукции, по объему и виду предоставляемых услуг.

Наиболее распространёнными предприятиями общественного питания считаются рестораны, кафе, бары, столовые, заведения быстрого обслуживания. Они могут работать на сырье или полуфабрикатах, быть в системе управления структурного образования или независимыми, с различной формой собственности. Предприятиям такого типа предъявляют довольно жёсткие требования. Они касаются внешнего вида предприятия, оформления залов и помещений для потребителей, наличия сцены и танцевальной площадки, банкетного зала или закрытых (отдельных) залов, микроклимата, мебели, столовой посуды и приборов, столового белья, меню и ассортимента собственной продукции и покупных товаров, методов обслуживания потребителей, одежды и обуви обслуживающего персонала, музыкального сопровождения.

1.2 Место и роль сайта в деятельности компании

В нашей и зарубежной литературе можно наткнуться на самые разные мнения о роли web-сайта в деятельности компании. Любое современное коммерческое предприятие, стремящееся к достижению получения максимальной финансовой прибыли, должно иметь собственный web-сайт, который является эффективным инструментом не только для увеличения объемов продаж, но и с целью рекламных изучений в потребительской среде, а также организации обратной связи с покупателями. Это нематериальный актив компании, отличающий современный бизнес [1].

Компания без сайта – это то же самое, что деловой человек без визитной карточки. Чтобы пользоваться возможностями web-сайта с максимальной эффективностью, необходимо четко понимать, какие задачи решаются с помощью сайта [2].

В основе web-сайта должна лежать определенная бизнес-модель, которая является моделью поддержки существующего бизнеса, моделью создания нового бизнеса в сети Интернет.

Web-сайт можно сравнить с традиционными выставками. Его можно представить, как довольно большой международный выставочный зал, где у потенциальных покупателей есть шанс вступить в непосредственный контакт с потенциальными продавцами и посетить презентованную онлайн-экспозицию. Сайт выполняет роль не только рекламного инструмента, но и инструмента личных продаж. Помимо этого, web-сайт – это нечто среднее между прямыми продажами (поддержание диалога с потребителем) и рекламой (повышение информированности, демонстрация продукта и предоставление информации без интерактивного участия) [15].

Однако организовать увеличение числа посетителей сайта могут и офлайн-инструменты коммуникации, осуществляемые с помощью традиционных СМИ, сувенирной продукции или прямого контакта с

потребителем. Отсюда следует, что модель маркетинговых коммуникаций с использованием веб-сайта компании имеет вид, представленный на рис. 1.1.

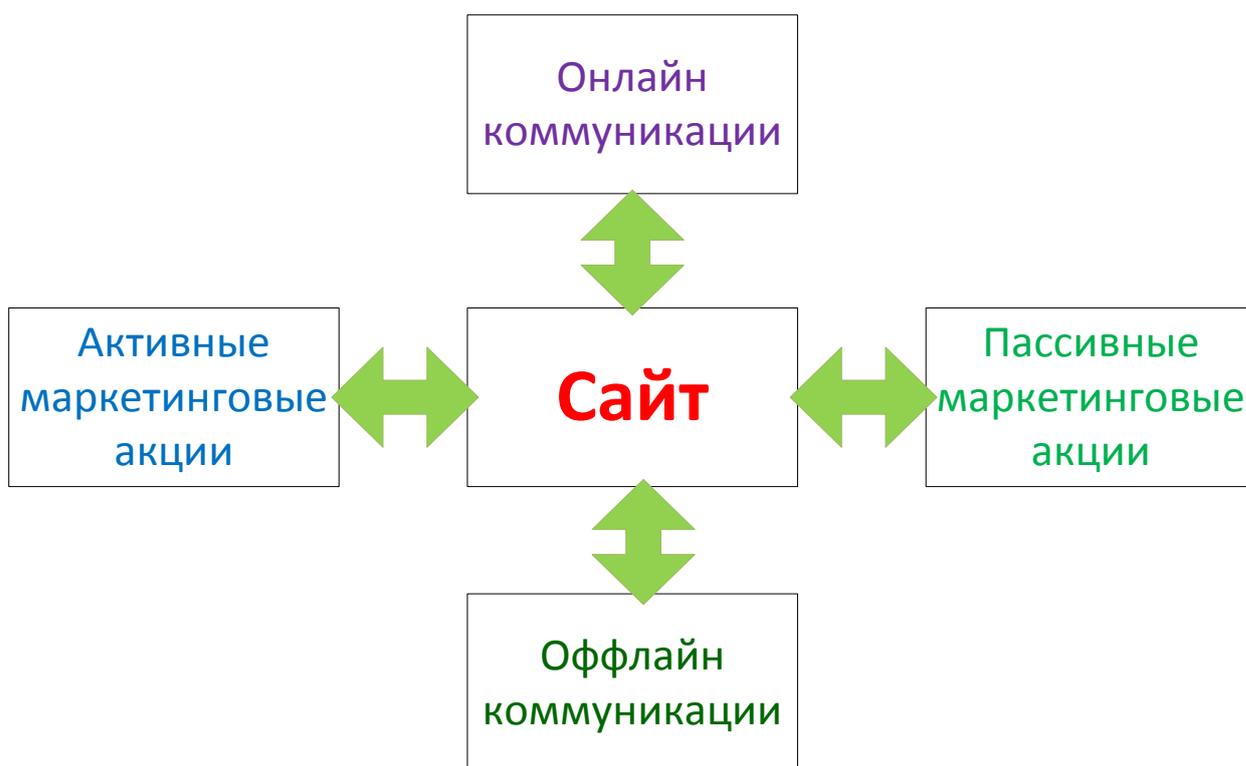


Рисунок 1.1 – Модель маркетинговых коммуникаций с использованием веб-сайта компании

Таким образом, рекламные коммуникации, которые создают поток посетителей на сайт, должны не только содержать в себе активные и пассивные коммуникации, но и привлекать онлайн- и офлайн-инструменты. Ведь при наличии собственного сайта возрастает эффективность проводимых в сети рекламных мероприятий. Информация о поведении посетителей на сайте может сразу попадать в информационную систему компании и служить для более качественного их обслуживания. Стоит сделать вывод, что сайт является основой системы коммуникаций в Интернете.

В целом перечисленные маркетинговые задачи, выполняемые сайтом, сводятся к следующим задачам: осуществление и повышение уровня продаж, формирование и поддержание имиджа, предоставление различного рода информации, формирование канала обратной связи с потребителем.

Сайт компании может также эффективно решать вопросы, связанные с послепродажным обслуживанием клиентов (рис. 1.2).



Рисунок 1.2 – Функции сайта коммерческой компании

На рис. 1.2 представлены основные функции сайта коммерческой компании. Сайт компании – один из главных коммуникативных рекламных инструментов в сети Интернет. Потребители компании являются главной целевой аудиторией сайта, но не единственной. С помощью сайта компания имеет возможность взаимодействовать с партнерами, акционерами, СМИ и другими контактными аудиториями.

Официальный сайт представляет собой мощный механизм развития и поддержания имиджа компании и ее товаров, способствует росту популярности бренда. Сайт обязан содержать в себе всю нужную информацию о компании, о сферах ее деятельности, о производимых товарах и оказываемых услугах. Представляя собой наилучшую маркетинговую площадку, грамотно разработанный корпоративный сайт непременно будет работать над созданием у потребителей и партнеров образа проверенной, идущей в ногу со временем, динамично развивающейся компании. Он также

может стать основной площадкой для обеспечения обратной связи с потребителями компании.

Главные характеристики сайта можно охарактеризовать моделью «7С», которая содержит следующие элементы [20]:

1. Контекст (Context): баланс между эстетическим и функциональным содержанием сайта, его оформлением и дизайном. Сайт может быть:

- функционально ориентированным, если основную роль играют функции, на выполнение которых он направлен;
- эстетически ориентированным, когда эстетический вид web-страницы стоит в приоритете;
- гибридным, если корпоративный сайт эффективен эстетически и функционально.

2. Содержание (Content): тексты, фотографии, изображения, музыка и видео.

3. Степень объединения (Community): способы, которые позволяют всем пользователям сайта поддерживать связь друг с другом. По способности корпоративных сайтов обеспечить поддержание такой связи их можно условно разделить на три вида:

- сайты, которые не дают пользователям возможность взаимодействовать друг с другом;
- ограниченные сайты, которые предлагают такие возможности, как чтение и размещение отзывов о товарах и услугах компании;
- сайты, которые предлагают такие интерактивные функции, как чаты и доски объявлений.

4. Коммуникации (Communication): способность сайта осуществлять двухстороннюю связь между представителем компании и потребителем.

5. Связь (Connection): степень взаимосвязанности конкретного сайта с другими сайтами с помощью размещения ссылок, партнерских программ, поисковой оптимизации.

6. Коммерция (Commerce): возможности корпоративного сайта по осуществлению коммерческих сделок. Так, сайты компаний, которые осуществляют электронную торговлю, имеют возможность зарегистрировать клиентов, хранить информацию о предпочтениях, контактах заказчика и параметрах доставки сделанного заказа.

7. Кастомизация (Customization): способность сайта предоставлять информацию и дополнительные возможности разным сегментам потребителей, возможность персонализации.

В качестве ключевых компонентов корпоративного сайта выделяют:

- наличие контактных данных (адрес, телефон и пр.);
- интуитивно понятный интерфейс (логичная структура, читабельность текстов);
- внешняя привлекательность;
- наличие корпоративной информации (отчеты, посредники и точки продаж, возможности трудоустройства);
- возможности персонализации и сохранения персональных настроек,
- отслеживание нахождения товара по трек-номеру;
- возможности совершения онлайн-покупки (возможность оплаты, особенности ценообразования).

Сайт должен включать такую характеристику, как «демонстрация надежности», и быть основанным на корпоративных имидже и репутации, которые существуют за пределами сети Интернет [22, с. 107]. Помимо этой характеристики еще к главным составляющим сайта относят:

- тип доменного имени;
- информацию о продукте;

- языковые возможности (для экспортера важно использование тех языков, носителями которых являются ЦА);
- возможность персонализации;
- контактную информацию.

Таким образом, существует внушительный перечень характеристик web-сайта коммерческой компании.

Основные характеристики сайта коммерческой компании представлены в таблице 1.1.

Таблица 1.1 Основные выделенные характеристики сайта

№ п/п	Характеристика
<i>Информация</i>	
1	Наличие информации о функциях, свойствах, ценах товара
2	Наличие корпоративной информации (финансовые отчеты, вакансии, миссии, цели)
3	Наличие контактной информации (телефон, адрес)
4	Наличие информации о тенденциях отрасли, отраслевые обзоры
5	Наличие раздела «часто задаваемые вопросы»
6	Наличие отзывов о компании и товарах
7	История компании
<i>Привлекательность</i>	
8	Наличие фото товара и видео о товаре
9	Оформление и привлекательность сайта
10	Фотографии клиентов
11	Фотографии персонала компании (в том числе руководителей компании)
<i>Простота</i>	
12	Простота навигации
13	Ясность и понятность текста
<i>Дополнительные возможности</i>	
14	Наличие инструкций по использованию, советов по подбору моделей, цветов, размеров
15	Возможность сохранения персональных настроек
16	Наличие рекламы других компаний
17	Наличие онлайн-игр и развлечений
18	Возможность бесплатного скачивания дополнительных программ
<i>Обратная связь</i>	
18	Наличие форм обратной онлайн-связи
19	Возможность осуществления онлайн-покупки
20	Онлайн-диагностика возникающих проблем при использовании товара
21	Возможность отслеживания статуса заказа и его местоположения

Компания, где целевая аудитория - потребители, которые массово приобретают фирменный товар, обычно не нуждается в дополнительных вложениях денежных средств в развитие. Основной задачей такой компании будет использование своего web-сайта как дополнительного инструмента формирования имиджа. Главными требованиями к подобным сайтам обычно являются: внешняя привлекательность и дизайн, информация о товаре и специальных предложениях.

Для всех информационных проектов контент – важнейшая составляющая, которая должна привлечь новых посетителей, сделать их лояльными, а также должна гарантировать эффективную монетизацию.

Сейчас медиа-контент – это уже не только информация, представленная в виде текстов (и сопутствующая ей HTML-вёрстка с форматированием и гиперссылками), но и комплекс других форм передачи информации: изображения, видео, музыка, а также множество иных современных решений, включая карты и структурированные материалы.

Классификация информации по форме подачи:

1. Текст. Сегодня это понятие включает в себя целый ряд решений: встроенные изображения, видеоролики, музыка и прочее. Любой текст в интернете представлен в виде HTML, следовательно, элементом контента будет и встроенное форматирование, которое включает в себя выделение заголовков (жирный и курсив), а также встроенные гиперссылки.

2. Фотографии. Наиболее популярный вид контента. Большинство интернет-сайтов использует такой контент не как отдельную составляющую, а вместе с текстом, которая отлично дополняет его. После того, как цифровые фотоаппараты и мобильные телефоны со встроенными камерами заняли свое место в жизни современного человека, стремительно выросли и фотохостинги, которые представляют собой крупнейшие хранилища фотоконтента.

3. Видео. Такой вид контента стал развиваться позже фотографий, так как требует более широких каналов доступа к интернету для просмотра и

загрузки, а также цифровых видеокамер или мобильных телефонов с возможностью записи видео, которые появились несколько позже, чем первые цифровые фотоаппараты. Первым видеохостингом стал YouTube, который произвел революцию в жанре любительского видео, после чего стали появляться и другие видеосервисы.

4. Аудио. Целый ряд хостингов дают возможность загрузить свои собственные подкасты и прослушать их. Довольно часто аудиофайлы становятся частью тестового контента электронных СМИ.

5. Другие виды контента. Нынешние технологии предлагают достаточно много интереснейших решений для создания и доставки контента, включая видеотрансляции в реальном времени (прямые эфиры), интерактивные карты и прочее. Перечень видов контента будет расти с появлением новых технологий и потребностей, и этот список можно будет все время дополнять.

1.3 Актуальность использования информационных технологий в ресторанах

Современный рынок услуг и продукции предприятий общественного питания активно развивается, происходит усовершенствование применяемых технологий. Все это сказывается на том, что изменяются масштабы деятельности предприятий и потенциальные возможности.

Непременным условием роста экономической эффективности для любой фирмы является расширение применяемых информационных ресурсов.

В ресторанном бизнесе применяется, в основном, сетевой формат организации производства, что приводит к усложнению организационной структуры и некоторых функций управления. Поэтому появляется необходимость использовать сложные информационные системы, иногда

индивидуальной разработки, с учетом специфики сферы общественного питания.

Для примера, если автоматизировать бизнес-процесс производства, то у шеф-повара появится возможность мгновенно составлять меню, редактировать рецептуру и прочее; технологу представится возможность высчитывать себестоимости всех блюд и их экономические показатели. Если автоматизировать склад и систему логистики, то станет намного легче выполнять функции менеджеров по закупке, приемке и хранению продуктов.

Кроме того, back-офис позволяет менеджерам контролировать и управлять всеми подразделениями предприятия, а управляющему становится доступной вся необходимая информация, на основе которой он может грамотно оценить эффективность и результативность своего заведения.

Для сотрудников всех уровней, а также владельца бизнеса подобный автоматизированный комплекс предоставляет большое количество возможностей, которые должны оказать положительное влияние на организацию работы предприятия и уровень сервиса. Основа всех систем автоматизации ресторана - механизм отчетов, настройка которого определяет результативность всей системы. Можно уточнить, что сами отчеты – это не только контролирующие средства, но еще и эффективный инструмент, который служит для сбора необходимой маркетинговой информации.

Автоматизация ресторана является задачей прикладной. Поэтому успех разработанного приложения, в основном, зависит не столько от умения программировать, сколько от знаний и понимания специфики ресторанной сферы. Отсюда следует, что успешными становятся те предприятия, которые начинали свои разработки больше 10 лет назад. Они смогли накопить достаточное количество знаний и опыта в работе с предприятиями общественного питания, отладили и протестировали свои программные продукты.

Как правило, владельцы предприятий общественного питания выбирают те ИС, которые прошли испытание временем и считаются

надежными, но при этом большое значение имеет стоимость системы. Например, для маленького предприятия нет смысла покупать дорогую ИС, с расширенным набором возможностей, так как большинство функций использоваться не будет. В этом случае ИС подбирается с учетом минимальной функциональности, что позволяет экономить денежные средства. Тогда владелец ресторана, в первую очередь, обращает внимание на надежность и стабильность предлагаемого ПО, его популярность (нет смысла учить персонал на малоизвестной системе) и функциональность. Но также немаловажную роль играет постгарантийное и гарантийное обслуживание.

Система автоматизации ресторанный бизнеса должна решать целый ряд управленческих задач: помогать активно привлекать новых посетителей, обеспечивать актуальной информацией о финансах, создавать эффективные алгоритмы мотивации персонала, снижать уровень злоупотреблений, оптимизировать процесс закупок и процесс хранения товарных резервов, обладать удобным аналитическим аппаратом.

Основные отличия всех систем автоматизации у разных компаний – разработчиков заключаются в таких понятиях, как надежность, удобство работы и стоимость предлагаемых решений. Делая выбор в пользу какой-либо системы для ресторана, стоит обратить внимание на:

- список ресторанов, которые уже установили ИС: фирмы, которые серьезно относятся к своей работе, сами предлагают список крупных ресторанов, уже работающих на их ИС;
- системы, установленные в других городах (только достаточно надежное ПО сможет выполнять работу на существенном расстоянии от разработчиков);
- статистику выездов на предприятие (рестораны, которые работают давно, вызывают специалистов не чаще одного раза в месяц);
- комфорт и понятность ИС для разного уровня подготовки пользователей. Чтобы оценить этот показатель, лучше всего прислушаться к

мнению управляющего и технолога – их мнение будет экспертным, в то время, как мнения обслуживающего персонала (официанты, бармены и прочие) учитывать не всегда целесообразно, так как они видят деятельность ресторана только с точки зрения своего ограниченного участка работы;

- стоимость обслуживания системы в случае внештатных ситуаций (выход техники из строя, недопустимые перепады напряжения в сети, обрыв кабеля);
- возможность предоставить замену оборудования в случае необходимости проведения ремонтных работ;
- возможность предоставить штатного сотрудника, при условии наличия компьютеров в сети от 5 штук.

Автоматизация бизнес-процессов позволяет значительно сократить объемы ручного труда, так как позволяет учитывать перемещения продукции, рассчитывать себестоимость готовой продукции и значительно упрощать анализ эффективности работы всех подразделений предприятия общепита.

Внедрение автоматизированной ИС в деятельность ресторана позволяет решить вопрос хищений, а также предотвратить другие возможные нарушения со стороны сотрудников. Система обеспечивает контроль выполнения технологических процессов к приготовлению блюд, ограничение права доступа в систему для персонала, контроль за удалениями и ручными корректировками в системе, контроль и мониторинг закупочных цен.

Информационная система позволяет сократить время обслуживания посетителей ресторана, делает возможным повышение точности в подсчетах расхода продуктов, дает возможность четко отслеживать работу всего предприятия и производить анализ данных, полученных из системы.

В настоящее время ресторан, не имея личного сайта, рекламирует свои услуги и проводимые акции при помощи тематических ежемесячных журнальных изданий и в сети Интернет на сторонних сайтах со схожей ЦА.

Проектирование личного web-сайта ресторана начинается с исследования ключевых, вспомогательных и обслуживающих процессов при помощи организационной модели.

На рисунке 1.3 приведена укрупненная схема процессов и событий в ресторане.



Рисунок 1.3 Модель организации процессов в ресторане

Базовыми видами деятельности ресторана являются:

- обслуживание посетителей заведения обеспечивают официанты, соблюдающие дресс-код, которые обязаны знать и соблюдать стандарты сервиса при общении с посетителями, при подаче блюд и напитков, приготовленных командой поваров или барменом по заказу;
- контроль деятельности ресторана и всего персонала осуществляют директор и управляющий;

- обеспечение всеми видами ресурсов для всех сфер деятельности ресторана (имеются в виду пищевые, человеческие, развлекательные, материальные, производственные).

Внедрение ИС также способно повысить лояльность клиентов.

Лояльность — это отношения, построенные на положительных эмоциях, которые играют важнейшую роль при принятии любых решений, связанных с объектом лояльности.

Потребительская лояльность — особое расположение покупателей к определенной торговой марке, которая мотивируется основательной эмоциональной привычкой к определенному товару, изделию или услуге.

В мегаполисах число предприятий общепита постоянно растет. И чем больше вариантов есть у потребителя, тем острее для рестораторов стоит вопрос об удержании клиентов и о привлечении новых. Клиент осознает свою значимость, отчего предъявляет повышенные требования к сервису. И уже недостаточно предложить качественное меню, вежливого и квалифицированного официанта, приемлемого уровня цен.

Поэтому были разработаны программы лояльности клиентов, которые помогают сохранить старых клиентов и привлечь новых. Сохранение старых клиентов – это не просто привлечение для повторного посещения, но и формирование особых, преданных отношений. Программы лояльности проектируются и разрабатываются с учетом специфик бизнеса, его целей, сферы деятельности и прочих факторов.

К программам лояльности относятся:

- дисконтные программы (разовые скидки на первое или последующие посещения);
- подарочные сертификаты;
- бонусные программы (бонусами можно оплатить сами услуги);
- организация безналичного расчета за обеды для корпоративных клиентов;
- прогрессивная шкала скидок;

- призы, награды, лотереи, подарки и т.п.;
- особые условия обслуживания.

В настоящее время существует множество инструментов для привлечения потребителя, которые доказали свою эффективность. Главное, чтобы программа лояльности была запущена вовремя, реализована грамотно и закончена в срок.

Этапы развития, выполнения и завершения программ лояльности должны рассматриваться отдельно, так как у каждого этапа есть свои характерные черты и условия, которые обязательны для выполнения.

Итак, в первой главе дипломной работы рассмотрена история формирования и развития ресторанного бизнеса, нормативная база, которая регулирует отношения предприятий общепита, а также тенденции развития данной отрасли. Успешность предприятия общественного питания тесно связана с вопросом использования информационных технологий, которые оптимизируют работу ресторана с целью повышения качества услуг. Использование web-сайта для ресторана позволяет автоматизировать некоторые процессы и дает возможность дополнительного распространения информации об услугах, которые оказывает предприятие и, соответственно, помогает наладить связь между менеджментом и клиентами.

2 РЕАЛИЗАЦИЯ ПРОЕКТА САЙТА РЕСТОРАНА

2.1 Выбор информационной системы для организации процесса создания, редактирования и управления содержимым сайта

Сайт - это связанные между собой различные интернет-страницы, которые можно найти по определенным адресам (URL). Можно, конечно, сверстать эти страницы по-отдельности и дать им необходимые адреса, но внести изменения или отредактировать такие страницы будет нелегко и потребует временных затрат.

Для решения этой проблемы были придуманы системы управления контентом (CMS).

CMS — информационная система, которая нужна для организации процесса создания, редактирования и управления содержимым любого сайта.

Общие принципы систем управления контентом (CMS):

1. **Доступность.** Система устанавливается на web-сервер и все операции с контентом осуществляются через любой интернет-браузер (Internet Explorer, Mozilla, Opera, Safari, GoogleChrome и т.п.), следовательно, нет необходимости в установке специализированного ПО.

2. **Функциональность.** Большая часть всех CMS разработана по принципу модульности: в основе лежит определенная база, к которой можно подключить дополнительные модули в случае необходимости. Модули могут быть уже включены в состав комплекса, могут быть подключены отдельно, кроме того, есть системы, которые разрешают самостоятельную разработку и включение дополнительных модулей.

3. **Надежность.** Большая часть нынешних систем предоставляют ряд средств для безопасной работы с сайтом, защиты информации и обеспечения целостности структуры.

4. **Общая структура.** В конечном итоге, все системы включают в себя оболочку (программа управления) и базу данных. Оболочка — это набор программ, которые после запроса пользователей берут информацию из базы

данных и передают ее в виде отдельной страницы браузеру. Таким образом страницы генерируются по заданному шаблону.

Важнейшие функции CMS:

- наличие инструментов, необходимых для создания содержимого и для организации коллективной работы над содержимым;
- управление содержимым: хранение, контроль версий, выполнение контроля доступа, управление потоком документов и т. п.;
- публикация содержимого;
- удобное представление информации для поиска и навигации.

В CMS системах в качестве содержимого могут храниться самые разные данные: документы, видео, изображения, контактные данные и прочее. Такие системы часто используются с целью сохранения, управления, модификации и публикации документов. Главным преимуществом можно считать контроль, то есть изменять содержимое может группа лиц пользователей.

На данный момент уже есть достаточное количество готовых систем управления содержимым сайтов, и такие системы по способу работы делятся на три типа:

1. Генерация страниц по запросу. В основе систем данного типа связка: «Модуль редактирования → База данных → Модуль представления». Модуль представления генерирует страницу с содержанием при запросе на него, на основе информации из базы данных. Информация в базе данных изменяется с помощью модуля редактирования. Страницы создаются сервером заново при каждом запросе, что создаёт дополнительную нагрузку на системные ресурсы. Нагрузку можно снизить, если использовать средства кэширования, которые предусмотрены в современных web-серверах.

2. Генерация страниц при редактировании. Системы этого типа создают набор статических страниц при редактировании содержимого сайта. Недостатком такой системы будет недостаточная степень взаимодействия между посетителями сайта и его содержимым.

3. Смешанный тип. Сочетает в себе преимущества первых двух. Такой тип реализуется путём кэширования. В этом случае модуль представления один раз генерирует страницу, тогда из кэша она загружается намного быстрее. Кэш будет обновлен автоматически, спустя некоторое время или после внесения изменений в определённые разделы сайта, а также кэш может быть обновлен вручную после команды администратора. Другой подход — сохранение определённых информационных блоков на этапе редактирования сайта и проектирование страницы из этих блоков при запросе соответствующей страницы пользователем.

Рынок CMS четко разделяется на коробочные системы управления сайтом и внутренние системы. Основная особенность коробочных систем состоит в том, что такие системы могут быть установлены и введены в эксплуатацию без участия разработчика. Внутренние системы, как правило, не являются отдельно продаваемыми продуктами и обычно поставляются в стандартной комплектации вместе с разрабатываемым сайтом.

Прежде чем сделать выбор в пользу какого-либо вида CMS, стоит рассмотреть и проанализировать плюсы и минусы каждого.

Таблица 2.1 Плюсы и минусы коробочных продуктов: «1С-Битрикс», UMI.CMS, ABO.CMS, Twilight и др.

Плюсы	Минусы
<ul style="list-style-type: none"> • отчуждаемость, возможность установки собственными силами; • наличие документации по системе, как для пользователей, так и для разработчиков; • развитая служба технической поддержки; • развитые партнерские сети: при создании сайта, к примеру, на Урале, пользователь без проблем найдет техподдержку в своем регионе, не обращаясь в центральный офис разработчика. 	<p>Коробки имеют типовой набор функционала и менее пригодны для нетипичных ситуаций, в отличие от индивидуальной разработки.</p>

Таблица 2.2 - Сравнение преимуществ и недостатков внутренних систем: RBC Contents, CMS ContentRight, DynaSite, Art-Publishing и др.

Плюсы	Минусы
<ul style="list-style-type: none"> • возможность оперативного изменения и доработки под нужды каждого конкретного проекта: т.н. «кастомизация» функционала системы и ее интерфейсов; • возможность интеграции с другими комплексными системами (например, информационными или бухгалтерскими). 	<ul style="list-style-type: none"> • отсутствие документации; • невозможность самостоятельного внедрения продукта (это прерогатива исключительно разработчика ПО); • при смене разработчика клиент несет определенные риски: внутренняя CMS способна развиваться только со своим производителем; • Сложности развития системы самим разработчиком. Внутренние системы постепенно отстают от «коробочных» по темпам своего развития, поскольку это развитие требует значительных инвестиций. Кроме того, уход ключевых разработчиков из компании часто ставит крест на дальнейшем развитии созданной ими внутренней CMS; • Внутренние системы, как правило, не лицензируются, поэтому нередко клиент попадает в зависимость от разработчика. Заключая договор с интернет-компанией, предлагающей внутренний продукт, необходимо настаивать на том, чтобы к договору прилагалось подробное техническое задание, где в деталях бы описывался функционал и особенности работы системы.

Очевидно, что в дальнейшем изучении и в выборе информационной системы для сайта «Бирхоф» внутренние CMS участвовать не будут. Минусов у таких систем гораздо больше, чем плюсов.

Критерии выбора коробочных CMS:

1. Назначение сайта. Наиболее значимым критерием выбора системы управления контентом (CMS) является назначение сайта. Ведь очень многое будет зависеть от того, для чего нужен сайт и какие задачи должны быть решены. Если сайт ориентирован, в первую очередь, на продажу товаров и услуг, то нужен будет один вид CMS, которая будет специализироваться на торговле. Если сайт новостной или информационный портал — это другой вид системы. Если планируется сайт для большой компании, то нужна будет корпоративная система. Все будет зависеть от направления деятельности и назначения web - сайта.

2. Функционал. После определения назначения сайта, возникает список функций, которые должна обеспечивать система: размещение разных форм информационного контента (блог новостей, статьи, термины и понятия, вопросы и ответы и т.п.), размещение предложений о товарах, перечень предлагаемых услуг, специальные формы для оформления заказов и заявок, функция обратной связи, поддержки, общения и т.п. Все эти задачи должны быть обеспечены функционалом сайта. Поэтому стоит особое внимание уделить функциональным возможностям сайта при выборе CMS.

3. Удобство. Следующим шагом стоит определить: кто, где и как часто станет работать с информацией на сайте, оценить его навыки работы с компьютером. Если подразумевается, что над сайтом будут работать несколько авторов, то CMS должна предоставлять такую возможность.

4. Технические требования к хостингу. Так как система должна быть установлена на сервер (хостинг), то у всех CMS есть несколько обязательных требований к серверу, на котором она будет работать. Должны быть различные БД, должны поддерживаться языки программирования, доступное пространство на диске и прочие особенности подобного рода. Если хостинг – площадка у предприятия уже есть, то выбирая CMS следует обратить внимание на техническую возможность использовать систему на

существующей площадке. Достаточно сложные, многофункциональные CMS способны создавать большую нагрузку на сервер, что обязательно стоит учитывать на начальном этапе.

5. Безопасность. Существует два типа программных разработок: системы с открытым кодом (исходный код CMS открыт для модифицирования) и коммерческие системы с закрытым кодом (исходный код CMS является собственностью компании - разработчика и не может быть модифицирован). Для первого типа характерно наличие большого количества разных модулей и достаточно простое изменение функционала сайта. Такие системы поддерживаются на добровольной основе сообществом разработчиков. Для коммерческих (закрытых) систем важны безопасность и поддержка. И естественно, что для любых CMS важны обновления и исправления в системах безопасности.

6. Сколько стоит. Существуют закрытые и свободно распространяемые системы управления контентом. Причем среди закрытых CMS есть бесплатные или условно бесплатные редакции. И чаще всего такие бесплатные варианты содержат весь необходимый функционал для небольших интернет-проектов (блоги, интернет-магазины, сайты-каталоги).

7. Поддержка. Необходимо решить, кто и каким образом будет заниматься установкой и поддержкой CMS для проекта. Это может быть специалист на предприятии или же необходимо будет заключить договор с компанией-разработчиком. При этом важную роль будут играть качество поддержки и своевременность. Для систем с открытым кодом важно присутствие активных участников в сообществе пользователей, надежные каналы поддержки (официальные форумы, вики, базы знаний и т.п.).

8. Простота развертывания. Важным условием является простота установки и обновления системы. Установка и настройка системы, требующая длительного вмешательства эксперта, будет свидетельствовать о невнимательности разработчиков к администраторам и пользователям CMS.

Для принятия решения о том, какая CMS будет использована для сайта «Бирхоф», был проведен сравнительный анализ шести наиболее востребованных и популярных систем на российском рынке, а именно ABO.CMS, NetCat, WordPress, Twilight, UMI.CMS, 1С-Битрикс.

Таблица 2.3 - Сравнение российского рынка коробочных CMS и их ключевые особенности

	<u>ABO.CMS</u>	<u>NetCat</u>	<u>Word Press</u>	<u>Twilight</u>	<u>UMI.CMS</u>	<u>1С-Битрикс</u>
Встроенный редактор изображений	+	-	+	+	-	-
Отмена внесенных изменений	-	-	+	+	+	-
Корзина (восстановление удаленных объектов)	+	-	+	-	+	-
Копирование объектов системы	+	-	+	+	+	+
Высокий уровень кастомизации (гибкость пользовательских настроек)	-	+	+	-	+	+
«Дружественные» URL (понятные адреса страниц)	+	+	+	+	+	+
Автокорректор URL	-	-	+	-	+	-
Мультисайтовость	+	+	+	+	+	+
Многоязычность	+	+	+	+	+	+
Программная платформа	PHP5	PHP4	PHP4	Perl	PHP5	PHP4
Цена минимальной редакции, руб.	8700	5880	0	3750	2990	4900
Цена максимальной редакции, руб.	29700	32200	0	27000	29990	48900

Выбор системы управления контентом – сложный процесс, требующий учета многих факторов, которые были приведены выше в таблице, где видно, что система WordPress – единственная система, которая отвечает всем требованиям и является бесплатной.

Поэтому для разработки сайта ресторана «Бирхоф» принято решение использовать CMS WordPress. И тогда необходимо добавить, что выбранная система обладает еще рядом некоторых существенных преимуществ:

- ориентированность на разработчиков и дизайнеров. Благодаря весьма простой модификации внешнего вида сайта можно в несколько кликов внести изменения или переделать сайт полностью. Это позволит максимально персонализировать сайт, подстраивая его внешний вид под необходимые нужды;

- наличие большого количества дополнительных компонентов, расширений, плагинов, инструментов. Для любого популярного продукта с открытым исходным кодом характерна подобная ситуация. Это дает возможность встраивать в сайт почти все, что угодно. В этом случае, важным преимуществом WordPress будет то, что этой системе необходимо намного меньше дополнительных компонентов, чем многим другим системам. К тому же встраиваемые компоненты почти не будут влиять на производительность системы WordPress в целом;

- внешне привлекательный готовый продукт. Даже если нет опыта работы с сайтом, то все равно можно получить качественный продукт. Это весомый аргумент для тех, кто хочет создать коммерческий сайт с целью качественно презентовать свой продукт для пользователей;

- своевременный выход обновлений, а также постоянное повышение качества и безопасности системы. Сообщество WordPress считается одним из самых крупных. Над обновлениями этой CMS работает огромное количество разработчиков, которые все время выпускают новые расширения, а также выявляют недоработки и способы улучшения работы

системы. Как следствие, с течением времени, пользователи получают наиболее качественную систему;

- **многофункциональность:** потенциал WordPress почти неограничен. Система позволяет проектировать сайты практически любой сложности и содержания, используя минимум дополнительных инструментов и расширений;

- **скорость работы сайта.** WordPress предоставляет вместе с установкой небольшой объем занимаемой памяти, что позволяет осуществлять меньшую нагрузку на сервер, и, следовательно, сайт будет работать быстрее.

- **сайт на WordPress** достаточно просто продвигать и оптимизировать. Поддержка сайтов на WordPress считается одной из самых простых, потому что легко подыскать грамотного оптимизатора, знакомого с данной CMS, или контент-менеджера.

2.2 Выбор средств разработки интернет-сайта ресторана

2.2.1 Выбор хостинга

Для организации системы и ее работы во всемирной сети необходимо прежде всего определиться с выбором хостинга [7].

С технической точки зрения систему предполагается разместить на платном хостинге одного из хостинг-провайдеров, по крайней мере, на начальном этапе его развития.

Пользователю необходимо рассмотреть примерную структуру и требования к хостингу.

При выборе хостинга принимаются во внимание следующие параметры:

1. Местоположение дата-центра.

2. Технические характеристики:

- место на диске;
- ограничения по трафику;
- число почтовых ящиков;
- число FTP-аккаунтов;
- количество баз данных и их типы;
- наличие, версии и режимы работы серверных языков программирования.

3. Наличие более высоких тарифных планов (актуально для перспективы развития проекта).

4. Ограничения на хостинге[47]:

- лимиты на максимальное число отправляемых писем в час/день;
- максимальное время выполнения PHP-скрипта;
- лимит по расходуемой памяти на скрипт;
- количество одновременных соединений с базой данных MySQL;
- максимальная загрузка процессора сервера;
- ограничения по содержанию материалов сайтов;
- другие ограничения.

5. Поддержка:

- методы связи;
- наличие круглосуточной online-поддержки;
- качество работы поддержки;
- статус хостера;
- наличие официальной регистрации;
- юридическое лицо, физическое лицо;
- наличие офиса.

6. Отзывы пользователей на качество работы хостинга.

Условия эксплуатации требуют: размещения в специальном помещении, где есть кондиционирование, резервное питание, стойки, и квалифицированный обслуживающий персонал.

Такие характеристики зависят прежде всего от планируемого количества посетителей в сутки, к примеру:

- Рекомендуемая конфигурация для проектов с посещаемостью менее 50000 в день: Processor Intel Pentium III 800 MHz, 512Mb RAM, 100Mb HDD свободного места на диске[45].
- Рекомендуемая конфигурация для проектов с посещаемостью более 50000 в день: Processor Intel Pentium IV 1.7 GHz, 1Gb RAM, 100Mb HDD свободного места на диске.

Хостинг для размещения проектируемой системы должен обладать следующими характеристиками:

- поддержка PHP;
- поддержка MySQL;
- количество баз данных – одна;
- дисковое пространство – до 10 Гб;
- количество доменов на аккаунте – 1;
- наличие статистики по сайту.

Данными характеристиками в настоящее время обладает большинство хостинг-провайдеров, поэтому имеет смысл сравнение по стоимости их услуг. Сравнение тарифных планов различных хостинговых провайдеров приведено в таблице 2.4.

Таблица 2.4 Сравнение тарифных планов различных хостинговых провайдеров

№ пп	Наименование хостинг-провайдера	Стоимость пакета услуг, рублей в год.
1.	Джино.ру	2573
2.	Мастерхост	2980
3.	Мажордомо	2760
4.	Макхост	3689
5.	Agava	4200
6.	Best-hoster	3300
7.	AlfaHosting	3900
8.	Hosting RuNet	4000

Таким образом, наиболее дешевым является хостинг от компании Джино. Важным преимуществом данной компании также является возможность самостоятельного комплекта покупаемых услуг.

2.2.2 Обоснование проектных решений программному обеспечению

Программное обеспечение системы должно быть достаточным для выполнения реализуемых функций системы, а также иметь средства организации требуемых процессов обработки данных, позволяющих своевременно выполнять необходимые функции во всех режимах функционирования системы[18].

Для того, чтобы обосновать проектные решения, которые были приняты для реализации проекта, необходимо изучить следующий ряд программных продуктов и технологий:

- операционная система;
- языки программирования для сети интернет;
- технологии создания веб-проектов;
- базы данных;
- web-серверы.

Для нормального функционирования уже выбранной системы необходимо наличие операционной системы Microsoft Windows Vista/7/8.

Для того, чтобы выбрать язык программирования, следует обозначить требования к среде программирования.

Требования к среде проектирования:

- кроссплатформенность, которая необходима для интеграции с иными подсистемами;
- гибкость, которая позволяет изменять и дорабатывать систему без дополнительных расходов;

- безопасность;
- язык программирования должен предоставлять широкие возможности по работе с современными СУБД[48];
- открытость исходного кода, которая позволит динамическое совершенствование ИС и исправление ошибок[26].

В настоящее время наиболее распространены следующие скриптовые языки веб-программирования[27].

- JSP;
- ASP.NET;
- Ruby;
- PHP;
- Perl.

Сравнительный анализ перечисленных языков программирования по шести критериям представлен в таблице 2.5.

Таблица 2.5 Сравнительный анализ языков программирования

Критерий	JSP	ASP.NET	Ruby	PHP	Perl
Кроссплатформенность	+	+	+	+	+
Гибкость	+	+	-	+	+
Безопасность	+	+	+	+	-
Поддержка современных СУБД	+	+	+	+	+
Универсальность	-	-	-	+	-
Открытость исходного кода	-	-	-	+	+

Далее рассмотрены имеющиеся на данный момент языки программирования и технологии создания веб-проектов. Их немного, но у каждого языка программирования и технологии есть свои преимущества и особенности.

1. PHP – это язык программирования, который применяется с целью генерации страниц сайта, которые ориентированы на гипертекстовую разметку. На сегодняшний день PHP является самым распространенным

направлением программирования для сети Интернет. На основе этого языка разработано уже больше десяти миллионов русскоязычных сайтов. PHP – довольно простой язык, который обладает высокой скоростью обработки и позволяет подключать другие модули. При работе с PHP в связке с другими модулями появляется возможность работать с криптографической защитой, графикой, базами данных, документацией и т.д. Интерпретаторы – аппаратное либо программное средство, выполняющее анализ, обработку и выполнение исходного кода или программы. Для языка PHP сегодня существуют интерпретаторы для всех типов ОС.

2. ASP – технология, представляющая собой некоторый набор страниц на сервере, которые при вызове выполняют определенный программный код. Такая технология позволяет создавать страницы сайтов в виде небольших приложений, которые похожи на обычные программы. Чаще всего весь код в ASP делится на клиентскую и серверную части. Клиентская предназначена для того, чтобы сформировать визуальное представление web-сайта, а вот серверная часть отвечает за выполнение таких действий, как аутентификация пользователя на сервере, работа с БД, передача и обмен информацией, сохранение данных. Чисто технически - это текстовые файлы с частями HTML-кода, скриптами на Java и Basic.

3. ASP.NET является продолжением ASP-технологии с расширенным набором функций. Для наиболее качественного процесса программирования создана новая среда разработки и отладчик, которые смогли без проблем выявлять многие скрытые ошибки еще на этапе написания кода. Также изменилась компиляция и реализация кода между try и catch операторами. Добавились новые элементы управления, что в значительной мере упростило разработку самого интерфейса для пользователя.

4. Perl использовался с целью формирования отчетов, в основе которых лежат собранные данные из хранилища, поэтому применялся он для работы с большими базами данных, из которых было необходимо выделить

определенные материалы и представить их в виде наиболее удобном для восприятия. Кроме того, Perl активно использовался при создании сайтов, так как может качественно работать с базами данных и взаимодействовать с большим количеством инструментов для работы с текстом. На данный момент Perl применяется уже не так часто, однако некоторые серверные утилиты для обработки текста и энциклопедические сайты строятся непосредственно на нем.

5. Ruby – молодой, но уже достаточно популярный объектно-ориентированный язык программирования. Ruby имеет простой, понятный и четкий синтаксис, дает возможность легко переопределять операторы. Этот язык характеризуется быстрым циклом разработки (редактирование - запуск - редактирование), использованием интерпретатора и нетипизированных переменных, которые не требуют объявления. Модель прямого выполнения и динамического контроля типов дает возможность разрабатывать код поэтапно, причем удобно, что можно добавить функцию и сразу же ее протестировать. Однако углубленное обучение языку выше начального уровня может быть непростым. Информационных ресурсов, позволяющие изучить Ruby, в настоящее время не достаточно. Ruby медленно разрабатывается и развивается.

6. Python/Zope считаются языками высокого уровня. Создатели постарались сделать так, чтобы каждому программисту было комфортно работать с готовым кодом, а ПО под управлением Python обладало значительной быстротой исполнения. Непосредственно сам язык создан таким образом, что в нем встречаются элементы объектно-ориентированного и императивного программирования, а ядро содержит минимальный набор команд. Реализуя классы и функции, в конечном результате можно получить несколько модулей. И, следовательно, Python лучше всего использовать для разработки приложения для математических расчётов или работы с текстами.

7. JSP – технология, представляющая собой страницы, написанные на Java и выполняемые на сервере, что больше всего зависит от компилятора.

При использовании Jasper первоначально создается высоко оптимизированный байтовый код, который затем передается Java-машине для выполнения. JSP является достаточно частым решением для создания интернет-магазинов или аукционов. Секрет языка в том, что отлаженный код может выполняться как на сервере, так и в пользовательском браузере, также имеется возможность полноценного программирования с классами, разделением прав доступа и прочими удобствами, предоставляемыми полноценной Java.

8. HTML можно смело назвать основой всего интернета. Разработчику можно использовать различные фреймы и писать на любом языке программирования, но только при помощи Hyper Text Markup Language возможно графическое представление данных в окне браузера. Готовая страница, как правило, состоит из набора тегов, среди которых находится сам текст и выполняется подключение других сторонних компонентов. В целом, теги используются парные – один открывает и содержит в себе все настройки текста, альтернативные записи, а второй закрывает команду. Бывают и одинарные теги. Благодаря HTML можно работать с графикой, дополнениями браузера, таблицами, документами, шрифтами, фреймами, другими технологиями программирования и т.д.

9. Dynamic HTML использует статический язык разметки HTML, каскадные таблицы стилей CSS для своевременного изменения внешнего вида компонента при программировании, а также используются возможности JavaScript, которые позволяют внести изменения без обновления страницы. Страницы, которые написаны с применением такой технологии, грузятся дольше по времени, но зато потом сам пользователь может уверенно переключать вкладки, загружать картинки, менять положение объектов, обновлять содержимое и тому подобное без загрузки лишнего байта. А самое важное преимущество состоит в том, что все скрипты выполняются непосредственно на машине пользователя. По этой причине динамические страницы могут быть использованы на любом сервере.

10. XHTML является представителем гипертекстовой разметки, но основан не на тегах, а на иерархиях XML. С технической точки зрения такая разметка почти не отличается от оригинала, однако по визуальному представлению информации значительно выигрывает, поскольку в XML все должно располагаться согласно системе и по порядку. Такое новое решение оказалось удобным и быстрым. Структурированная иерархия избавляет ядро браузера от бесконечного хранения шаблонов, а также уменьшает время на обработку и отображение страницы, что дает возможность использовать это техническое решение на мобильных устройствах.

11. CSS используется программистами для разработки наборов стилей на основе объектно-ориентированного программирования. Суть языка заключается в возможности описания классов на странице и всех входящих объектов, где можно указать цвет, шрифт, способ отображения, размер и другие свойства для каждого объекта и использовать готовый стиль нужное количество раз без дополнительного описания. Таким образом, использование каскадных таблиц стилей значительно сокращает размер HTML-кода, а также позволяет настраивать отображение сайта для разных браузеров и устройств. При этом можно подключить другой набор стилей, и документ примет новый вид без изменения кода. Технология каскадных стилей позволяет объединять объекты разных категорий в один контейнер и задавать один стиль для большого количества объектов одновременно.

12. JavaScript характеризуется как прототипно-ориентированный язык программирования. Чаще всего используется в качестве встраиваемого языка для доступа к объектам приложений. Обширно используется в браузерах, как язык сценариев для реализации интерактивных web-страниц. Главной задачей при создании этого языка ставилась доступность языка для понимания программистов-непрофессионалов. Модель работы: текстовый файл с кодом подключается к документу и выполняется средствами клиентской машины. Достаточно часто эта технология применяется для корректировки работы органов управления сайта и контроля за действиями

браузера. На практике же JavaScript используют в сфере рекламы для разработки баннеров и контекстной рекламы, которая не доступна для обнаружения поисковой машиной.

13. SSI переводится как включение от сервера. Это значит, что разработчики часто повторяющиеся фрагменты в функциях или коде выносят для того, чтобы он подгружался в документ только, если возникает такая потребность. Эта технология чаще всего встречается в рекламной индустрии. Повторяющиеся баннеры при загрузке страницы выбираются при помощи сgi-скриптов, которые подключаются посредством SSI. Но такая функция предоставляется только на платных хостингах вычислительной мощности для физического расположения информации на сервере, которая всегда находится в сети интернет, так как существует большая вероятность вирусной атаки.

14. AJAX – технология, которая появилась не так давно, однако она уже стала достаточно популярной. С технической точки зрения AJAX - асинхронный JavaScript и XML, которые во время работы пользователя на странице осуществляют обмен данными без лишнего обновления этой страницы. Это значит, что пользователь совершает определенные действия - переходит по вкладкам, просматривает фотографии, статьи, читает сообщения – на одной странице. Это очень удобно с точки зрения скорости работы и простоты навигации. Однако у такой технологии существует несколько весомых минусов. Во-первых, при низкой скорости канала работа AJAX становится видимой. Во-вторых, переходы по странице не фиксируются в истории браузера, следовательно, вернуться к уже просмотренным материалам невозможно.

15. XML является иерархической технологией для хранения данных в текстовом виде. В некоторой степени написание скриптов на XML относится к объектно-ориентированному программированию. Возможно отметить большое число объектов, которые обладают собственным набором свойств и обычно являются либо элементом управления, либо

подключаемым модулем. С целью настройки каждого объекта используется описание его параметров с учетом вложенности свойств, которые имеют каждые свои свойства. XML является строгим языком и он не используется сам по себе, а применяется как составная часть динамической технологии визуализации страницы или для разработки собственной БД.

Таким образом, определив плюсы и минусы рассмотренных выше технологий, выбор для сайта «Бирхоф» был такой: HTML – для разметки страниц сайта, CSS – для оформления стиля страницы, JavaScript и базирующийся на нем AJAX – для быстрой работы сайта и проверки переданных данных на корректность, PHP – как основной язык проекта, поскольку он как раз предназначен для создания подобных разработок, а также по работе данным языком всегда можно найти большое количество информационных источников в интернете, а результат работы можно видеть в режиме реального времени на своем домашнем/рабочем ПК без обращений к удаленному серверу.

Подбор наиболее подходящей под задачи СУБД всегда представляет собой тяжелую задачу с различными требованиями и ключевыми элементами, но является чуть ли не самым важным этапом всей разработки приложений баз данных.

Чтобы выбрать СУБД, необходимо провести анализ свободно распространяемых баз данных. В качестве альтернатив рассматриваются преимущества и недостатки следующих СУБД:

1. SQLite;
2. MySQL;
3. PostgreSQL.

Преимущества SQLite:

- вся база данных состоит из одного файла, поэтому её очень легко перемещать на разные машины;

- в процессе разработки приложений возникает необходимость масштабирования и SQLite предлагает всё, что нужно для таких целей, так как состоит всего из одного файла и библиотеки, написанной на языке C.

Недостатки SQLite:

- обычно используется для небольших приложений, но для более крупных она не подойдет, так как отсутствует система управления правами доступа пользователей;

- нет возможность увеличения производительности.

Преимущества MySQL[9]:

- просто устанавливается;
- поддерживает большую часть возможностей SQL;
- имеет больше количество функций, установленных по умолчанию;

- легко справляется с большими объемами данных и хорошо масштабируется;

- позволяет существенно повысить производительность, за счет упрощения некоторых стандартов.

Недостатки MySQL:

- согласно задумке в MySQL заложены определенные ограничения функционала, которые просто необходимы в некоторых приложениях;

- из-за некоторых способов обработки данных (связи, транзакции, аудиты) данная СУБД иногда уступает другим СУБД по надежности;

- существуют жалобы на медленный процесс разработки, хотя MySQL технически открытое ПО.

Достоинства PostgreSQL:

- Бесплатное ПО с открытым исходным кодом. Эта СУБД является очень мощной системой.

- Существует достаточно крупное сообщество пользователей, в котором можно легко найти ответы на вопросы, возникающие при работе.

- Имеется немало дополнений, которые позволяют разрабатывать данные для этой СУБД и управлять ими, даже несмотря на огромное количество встроенных функций.

- Это не только реляционная СУБД, но в тоже время и объектно-ориентированная с поддержкой наследования и прочего.

Недостатки PostgreSQL:

- При простых операциях чтения PostgreSQL могут существенно замедлить сервер, что делает данную СУБД медленнее своих конкурентов, например, MySQL.

- Данная СУБД не может похвастаться своей популярностью, хотя и существует довольно большое сообщество пользователей.

- В силу выше перечисленных факторов иногда достаточно сложно найти хостинг, который бы поддерживал эту СУБД.

На основании представленного сравнения для дальнейшего использования была выбрана база MySQL.

Для нормальной работы реализуемой информационной системы необходим web-сервер – специальное программное обеспечение (сервер), которое принимает HTTP-запросы от пользовательских браузеров, и выдает им HTTP-ответ, чаще всего совместно с HTML-страницей, файлом, изображением, медиа-контентом или другими данными.

Web-сервером может быть назван как программное обеспечение, которое выполняет функции такого сервера, так и конкретно сам компьютер, на котором это ПО действует. Каждый пользователь получает доступ к серверу по URL-адресу через выделенный web-интерфейс. Существуют также дополнительные функции web-серверов, как например:

- ведение журнала с перечислением абсолютно всех обращений пользователей к web-ресурсам;
- аутентификация и авторизация пользователей;
- поддержка защищенных каналов HTTPS для передачи данных.

На сегодняшний день существует пара наиболее распространенных web-серверов:

1. Apache — web-сервер с открытым исходным кодом, часто применяемый в Unix-подобных системах;
2. IIS — web-сервер компании Microsoft, который в основном применяется на ОС семейства Microsoft Windows.

К основным плюсам Apache относятся стабильность работы и гибкость конфигурации. Сервер дает возможность подключать внешние модули для обмена данными, применять различные СУБД с целью авторизации пользователей, редактировать сообщения об ошибках. Основной недостаток: недружелюбный интерфейс для администрирования.

Преимуществом IIS будет считаться возможность интеграции с другими продуктами семейства Microsoft. А недостатком данного сервера будет сложная интеграция самого сервера с интерпретатором PHP, что делает затруднительной последующую настройку и использование этого сервера.

Поэтому, решение использовать сервер Apache будет наиболее рациональным.

Таким образом, для нормальной работы web-ориентированной системы нужно дополнительно установить следующие компоненты программного обеспечения:

- PHP 5.0;
- MySQL;
- Веб-сервер Apache.

2.3 Разработка карты сайта и шаблона дизайна

Разработка шаблона начинается с подготовки дизайн - макета главной и второстепенных страниц сайта.

Макет главной страницы (рисунок 2.1) включает несколько основных позиций:

- Логотип ресторана.
- Главное меню: переход к меню компании и оформлению заявки, просмотр оказываемых услуг, просмотр контактной информации.
- Вспомогательное меню: переход к разделам «О ресторане» и «Меню».
- Модуль слайд-шоу из трех фотографий и статей.
- Модуль быстрого поиска информации по сайту.
- Модуль вывода информации о сайте (подвал) и контактного номера телефона.

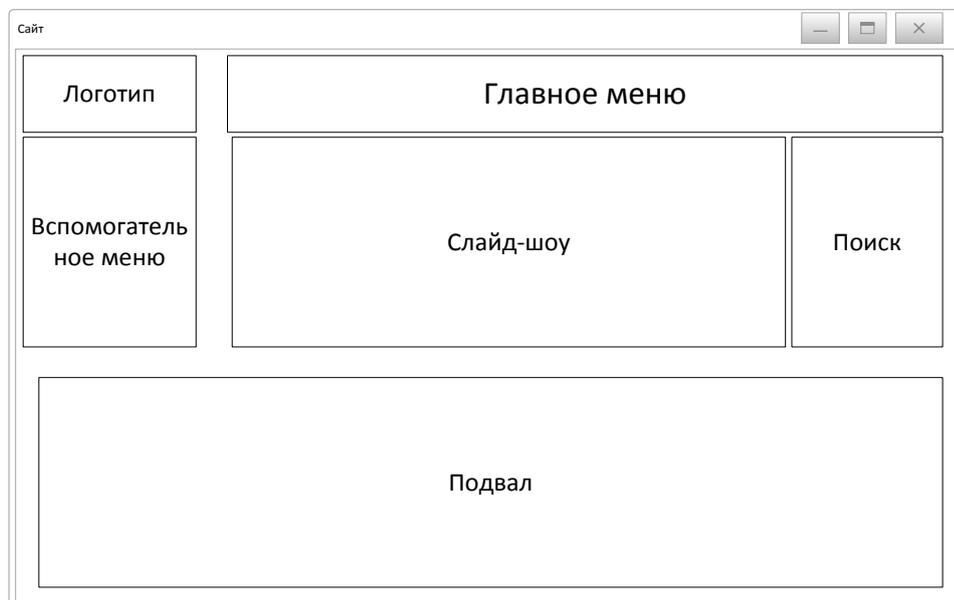


Рисунок 2.1 Макет главной страницы

После разработки дизайн - макета производится непосредственно разработка шаблона для выбранной системы управления содержимым CMS WordPress.

Шаблон для системы управления содержимым WordPress – это набор файлов, которые управляют представлением контента. Шаблон — это еще не сайт, но его и нельзя назвать полным дизайном сайта. Шаблон — это только основа, на которой строится дизайн сайта под управлением WordPress. Чтобы

добиться необходимого внешнего вида, шаблон должен работать непосредственно с содержанием, хранящимся в базе данных WordPress.

Общая структура для шаблонов WordPress представлена в таблице 2.6.

Таблица 2.6 - Общая структура для шаблонов WordPress

Наименование	Назначение
{root}\templates\{шаблон}\	основная директория шаблона
component.php	отдельный шаблон вывода компонента
index.php	непосредственно шаблон
params.ini	файл для хранения параметров шаблона
templateDetails.xml	файл описания шаблона
template_thumbnail.png	превью шаблона в административной панели
offline.php	шаблон, который загружается при выключенном сайте
error.php	шаблон для вывода ошибок
favicon.ico	иконка шаблона
index.html	файл для запрета листинга
{root}\templates\{шаблон}\html\	директория для шаблонизации расширений
modules.php	файл описания вывода стилей модулей
pagination.php	шаблон пагинации

Если некоторые из файлов отсутствуют в шаблоне, то их можно скопировать из директории {root}\templates\system\ в собственный шаблон и непосредственно изменять их в шаблоне.

Практика использования шаблонов в CMS WordPress имеет преимущества и недостатки:

- Контент и оформление полностью отделены друг от друга, особенно это заметно при использовании для верстки CSS (в отличие от использования таблиц в файле index.php). На данный момент это один из основных критериев соответствия современным стандартам сети.

- Применить новый шаблон можно практически мгновенно. При этом измениться могут не только графика и цветовая палитра, но и расположение и порядок отображения контента на странице.
- Если в разных разделах сайта потребуется разная компоновка страниц, то будет трудно достигнуть желаемого эффекта.
- Конечно, к разным страницам могут быть применены разные шаблоны, но подобное использование встроенной функциональности не вполне корректно. Значительно лучше будет применить выражения PHP и создать разметку, которая будет регулировать динамически число колонок в зависимости от опубликованного контента.

Способы разработки шаблона:

1. Создать собственный шаблон самостоятельно.
2. Создать шаблон, используя специальные программы (например, Artisteer).

Рассмотрим способы разработки шаблона более детально.

Разработка шаблона может выполняться вручную с использованием программ Adobe (Photoshop и Dreamweaver), но в этом случае процесс разработки шаблона будет достаточно трудоемким и потребует владения хорошими навыками работы в сфере веб-дизайна, понимание структуры шаблона для системы управления содержимым, а также глубокого владения языками HTML, CSS, PHP, AJAX и т.п.

Автоматизировать разработку шаблона можно с помощью программы типа Artisteer или CorelDraw Web-Design, при этом трудоемкость процесса разработки снижается, но все же требует высокой квалификации от разработчика.

Рассмотрим одну из самых популярных программ создания шаблонов для систем управления содержимым - Artisteer. Программа работает в системах Windows и Mac. Программа платная и триальная версия не позволяет произвести экспорт файлов в шаблон. Назначение программы Artisteer: создание шаблонов web-страниц и сайтов. С помощью Artisteer

создаются шаблоны и для систем управления содержимым, в частности, для WordPress. Artisteer – классический визуальный редактор. Единственное отличие от других подобных редакторов - это наличие режима предложений, который использует большую библиотеку. Редактор поддерживает HTML5 и CSS3. При открытии программы появится окно «Новый проект», в котором необходимо сделать выбор шаблона из предложенных категорий: бизнес, искусство, мода, медицина, магазины и прочее. После чего выбранный шаблон откроется в окне программы. Редактирование шаблона происходит в визуальном режиме (есть возможность добавить собственный код при создании страницы).

При разработке шаблона страница будет выглядеть следующим образом: область заголовка «header» (может содержать баннер, анимацию, слайды, заголовок и подзаголовок сайта, горизонтальное меню и пр.), область вертикального меню (слева или справа), область «block», область нижнего колонтитула «footer» и, конечно, область основного содержания страницы (content).

Изначально шаблон сайта состоит из двух страниц: главной страницы и страницы контактов. При дальнейшей работе есть возможность добавить сколько угодно дополнительных страниц. Все страницы сайта будут показаны на левой панели редактора. Можно редактировать предложенный шаблон, пользуясь панелью инструментов редактора. Панель инструментов состоит из нескольких вкладок. Вкладка «Начало» необходима для того, чтобы начать работу над шаблоном. Есть возможность выбрать дизайн веб-сайта, изменить дизайн, создать содержание и экспортировать результат. Вкладка «Правка» служит для редактирования шаблона в визуальном режиме. Можно добавить объекты, создать ссылки, вставить изображения, видео, кнопки и таблицы, есть возможность редактировать исходный HTML-код для каждой выделенной области.

Прочие вкладки предназначены с целью выбора цветовой палитры и шрифтов для макета: выбор и настройки фонового цвета или изображения,

создание верхнего и нижнего колонтитулов, создание горизонтального или вертикального меню и прочее. В свойствах страницы можно определить параметры оптимизации поисковой системы (SEO). Разрабатываемый шаблон возможно просматривать предварительно. Область такого просмотра можно использовать также с целью отредактировать текст и стили заголовков сайта, меню, содержание и прочее. Шаблоны, которые были созданы в Artisteer, можно экспортировать.

При экспорте шаблон будет сохранен в выбранной папке на ПК. Если шаблон предназначен для CMS (Artisteer или WordPress), то формат, созданный Artisteer будет темой WordPress или шаблоном WordPress. Также есть возможность использовать встроенную программу передачи файлов (FTP). Также есть некоторые бесплатные дополнения к программе. Но к сожалению, нет возможности импортировать шаблоны в Artisteer (даже созданные в этой программе). Чтобы сделать изменения в шаблоне, необходимо наличие файла проекта Artisteer.

3. Создать шаблон, изменив внешний вид уже созданного шаблона сторонними разработчиками.

Такой подход выбран в данной работе для сайта «Бирхоф», так как является менее затратным в финансовом отношении (программа Artisteer платная) и менее трудоемким в плане реализации.

После нахождения шаблона для сайта компании «Бирхоф» и его скачивания производится его установка на сайт. Порядок установки шаблона на сайт следующий:

1. Запустить web-сервер и сайт (ввести в браузере адрес в данном случае localhost/site).
2. В отдельной вкладке войти в административный раздел сайта (ввести в браузере адрес - в данном случае localhost/site/administrator, после чего ввести логин admin и пароль admin).
3. Перейти в раздел «Расширения – Установить / Удалить».

4. В разделе «Загрузить файл пакета» через кнопку «Обзор» выбрать скаченный архив для установки шаблона «Site.zip».

5. Нажать кнопку «Загрузить файл» и установить.

Таким образом, шаблон будет установлен. Отметим, что устанавливать шаблон можно и из папки, прописав путь к разархивированным файлам шаблона.

В целях повышения информационной безопасности сайта неиспользуемые шаблоны рекомендуется удалить через раздел «Расширения – Установить/Удалить – Шаблоны – выбрать неиспользуемый шаблон – Удалить – ОК».

После установки шаблона его необходимо сделать используемым по умолчанию, т.е. основным. Для этого необходимо перейти в раздел «Расширения – Менеджер шаблонов». С помощью переключателя отметить шаблон «Site» и нажать кнопку «По умолчанию». Шаблон будет применен к сайту, останется обновить содержимое вкладки и посмотреть на новый внешний вид.

Поскольку шаблон разработан сторонней организацией, то необходимо его видоизменить в соответствии с требованиями заказчика.

В процессе изменения шаблона существенную помощь оказывает плагин FireBug для браузера FireFox. Данный плагин не требует оплаты и дает возможность посмотреть внешний вид страницы, не меняя при этом код в шаблоне. Видоизмененный дизайн будет виден только пользователям FireBug в режиме реального времени. Это удобно, например, если есть необходимость посмотреть, как будет выглядеть страница после редактирования размера шрифта, изменения фона и т.д. Кроме того, плагин FireBug помогает отслеживать возникающие ошибки на сайте и взглянуть скорость загрузки элементов страницы сайта и многое другое.

Для редактирования отобранного готового шаблона сайта компании «Бирхоф» предлагает следующий алгоритм действий.

После установки плагина FireBug (через пункт меню браузера «Инструменты – Дополнения») работа осуществляется следующим образом: в правом нижнем углу браузера нажать на иконку расширения, в появившемся окне выбрать инструмент «Инспектировать» и на странице загруженного в браузере сайта выбрать любой объект дизайна, который требуется видоизменить. После этого в правой панели расширения появятся названия файлов и код, отвечающие за отображение объекта в окне браузера. Можно в интерактивном режиме изменять код и наблюдать последствия данного изменения.

На конечном этапе, если результат изменения кода устраивает, остается открыть файл шаблона и записать в него изменения в требуемых участках.

Итак, при помощи плагина FireBug были внесены все необходимые изменения в дизайн-шаблоне сайта. Непосредственное сохранение файлов производилось в редакторе кода NotePad++. Стоит обратить внимание на то, что для корректного отображения русскоязычных элементов интерфейса файлы необходимо сохранять в кодировке UTF-8.

3 ОПИСАНИЕ СОЗДАННОГО САЙТА РЕСТОРАНА

3.1 Программная реализация основных разделов сайта

После установки веб-сервера, системы управления содержимым, шаблона сайта производится общая настройка сайта.

Общие настройки содержат три блока: «Сайт», «Система», «Сервер».

Первый блок настроек сайта – «Сайт». Здесь представлены следующие настройки:

- «Название сайта» – данное название указывается при установке. Введенное значение будет использоваться в заголовках страниц.

- «Офлайн сообщение» отображается, когда сайт выключен, что может использоваться, например, при смене дизайна сайта или при тестировании расширений.

- «Визуальный редактор по умолчанию» – необходимо выбрать тот, который будет использовать администратор при подготовке материалов.

- «Настройки метаданных» необходимы для заполнения тегов META поисковой оптимизации.

- «Установки SEO» позволяют также оптимизировать сайт для более успешной индексации его поисковыми системами.

Второй блок – «Система» включает настройки пользователя. Поскольку в данном случае сайт реализует функцию оформления заказов через скрипт Интернет - магазина VirtueMart, то необходимо в обязательном порядке в данном блоке активировать функцию «Разрешить регистрацию пользователей». Другие настройки в блоке «Система» остаются по умолчанию («Системные установки», «Режим отладки», «Установки кэша и сессий», «Установки медиаменеджера»).

Третий блок – «Сервер» включает группы настроек сервера, FTP, базы данных и почты.

После выполнения общей настройки сайта можно приступать к созданию разделов и категорий, а затем можно будет наполнять сайт

нужными материалами. После выполнения этих действий можно приступать к разработке навигации по сайту.

Материал для сайта - это некоторая информация, которая была специально подготовлена для публикации. Как правило, для сайтов используют тексты и графику (материалы часто называют статьями), но вполне возможно использовать материалы в формате аудио или видео. Именно материалы формируют основную часть информационного контента.

Материал следует рассматривать как чистую информацию, независимую от оформления, разработанного дизайнерами для сайта. Ведь одна и та же статья будет по-разному выглядеть на нескольких страницах сайта с разным оформлением. Она может отличаться, например, по цвету и размеру шрифта, оформлением заголовков и фона, но содержание статьи при этом не изменяется. Создавая материалы (статьи), стоит уделить внимание содержанию и при этом можно совсем не думать о том, как эти материалы будут представлены на сайте (это зависит от встроеного шаблона).

Разделы и категории в иерархии сайта служат для группировки материалов по смыслу. Иерархия выглядит следующим образом: раздел включает в себя одну или несколько категорий. Категории содержат в себе материалы сайта, при этом каждый материал имеет право находиться только в одной определенной категории любого раздела.

Материал, который содержится в определенной категории, называется категоризованным. Помимо категоризованных материалов могут быть некатегоризованные (статические) материалы, которые не принадлежат ни к одной из категорий. Некатегоризованные материалы нужны для создания на сайте общей информации или для незавершенных статей, которым категория еще не присвоена.

Иерархическая структура сайта «Бирхоф» приведена на рисунке 3.1.

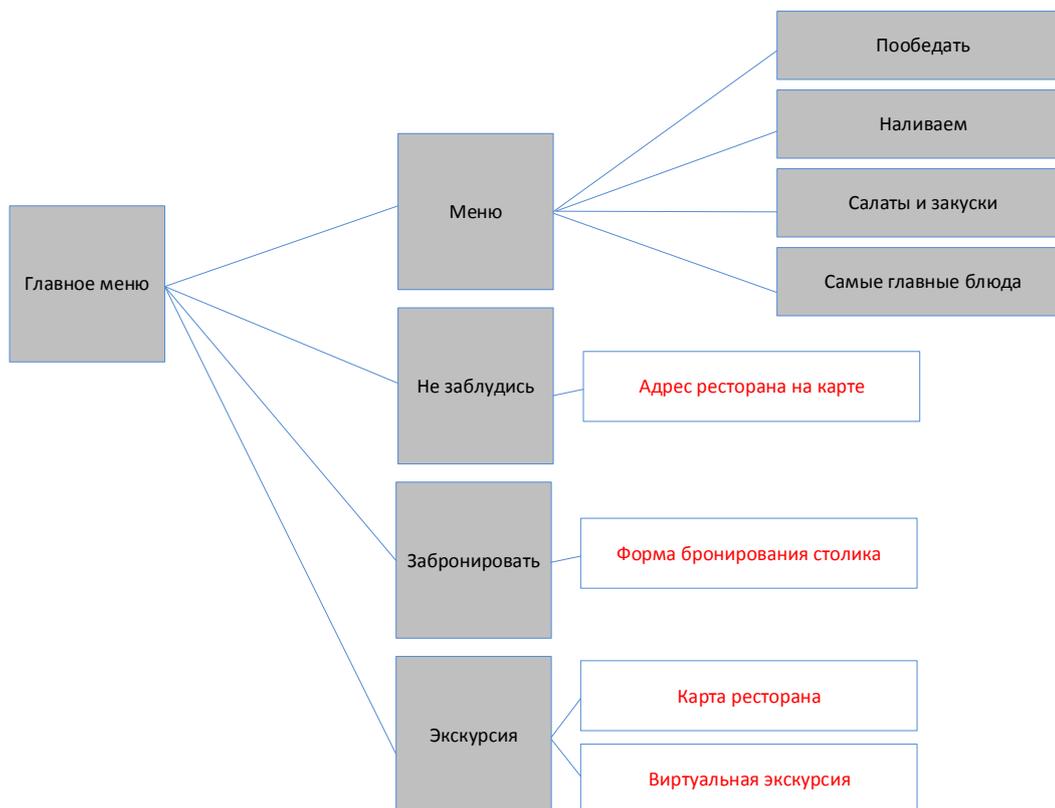


Рисунок 3.1 - Иерархическая структура сайта с разделами, категориями, материалами

Для отображения иерархической структуры сайта существуют несколько типов навигационного меню: «Шаблон блога раздела» и «Стандартный шаблон раздела», «Шаблон блога категории» и «Шаблон списка категории». Эти типы меню удобно использовать, чтобы показать на сайте материалы, которые содержатся в категориях и разделах. После того, как будет создан новый категорированный материал, он будет автоматически опубликован на странице, которая отображает данную категорию.

Также статьи для определенных категорий можно располагать в нужном порядке: по времени создания и редактирования статьи, по названию статьи, по количеству показов на сайте, либо в произвольном порядке.

Важно различать понятия: иерархическая структура материалов сайта и иерархия уровней навигационного меню. Разница в том, что структура материалов на сайте имеет жесткую трехуровневую структуру «Раздел –

Категория – Материал», а навигационное меню таких жестких ограничений не имеет, и число уровней может быть произвольным.

Сайты чаще всего содержат большое количество материалов (более 100 статей), что создает определенные сложности для администратора сайта. Но при использовании разделов и категорий можно отфильтровать материалы, оставляя для показа только те, что входят в нужные на данный момент раздел и категорию. Это позволит заметно сократить время на поиск нужных материалов.

Следует обратить внимание, что термин «Категория», в отличие от термина «Раздел», используется не только при группировке материалов, но и при использовании некоторых встроенных компонентов, таких, как «Баннер», «Контакты», «Ленты новостей», «Каталог ссылок». Категории в этих компонентах никак не связаны с категориями материалов, и для управления ими используются разные панели администратора.

В данной работе созданы три раздела: «Меню», «Подробнее», «Описание» и одноименные категории, которые принадлежат соответствующим разделам. Создание выполняется через пункт меню «Материалы – Менеджер разделов – Менеджер категорий».

После создания разделов и категорий были созданы семь материалов через пункт меню «Материалы – Менеджер материалов». Три материала включены в раздел и категорию «Меню» (содержат тексты статей для модуля слайд-шоу).

Посетителю сайта доступны следующие возможности:

- Ознакомление с главной страницей (рис. 3.2 и рис. 3.3).

Здесь посетитель сайта может ознакомиться с последними новостями или интересными статьями о ресторане «Бирхоф» и его жизни. Также на этой странице доступна галерея фотографий интерьера ресторана и фотографии, сделанные вовремя мероприятий, которые проводились на территории «Бирхофа». Находясь на главной странице, посетитель сайта имеет

возможность ознакомиться с контактными данными ресторана «Бирхоф» и режимом его работы (рис. 3.4).

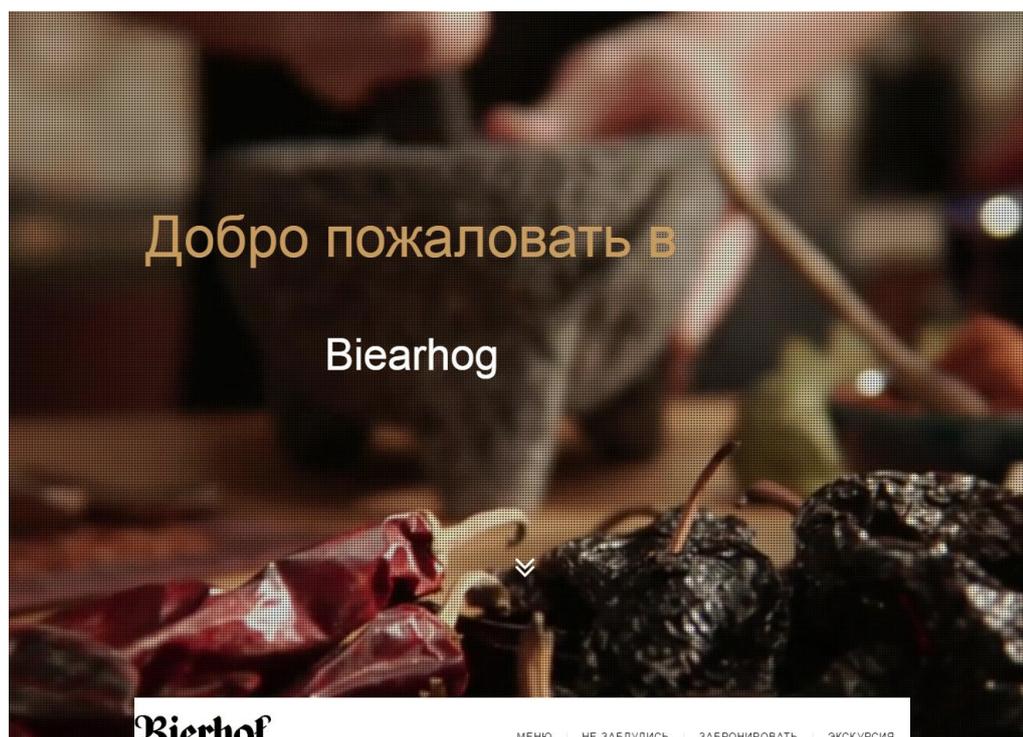


Рисунок 3.2 - Главная страница сайта

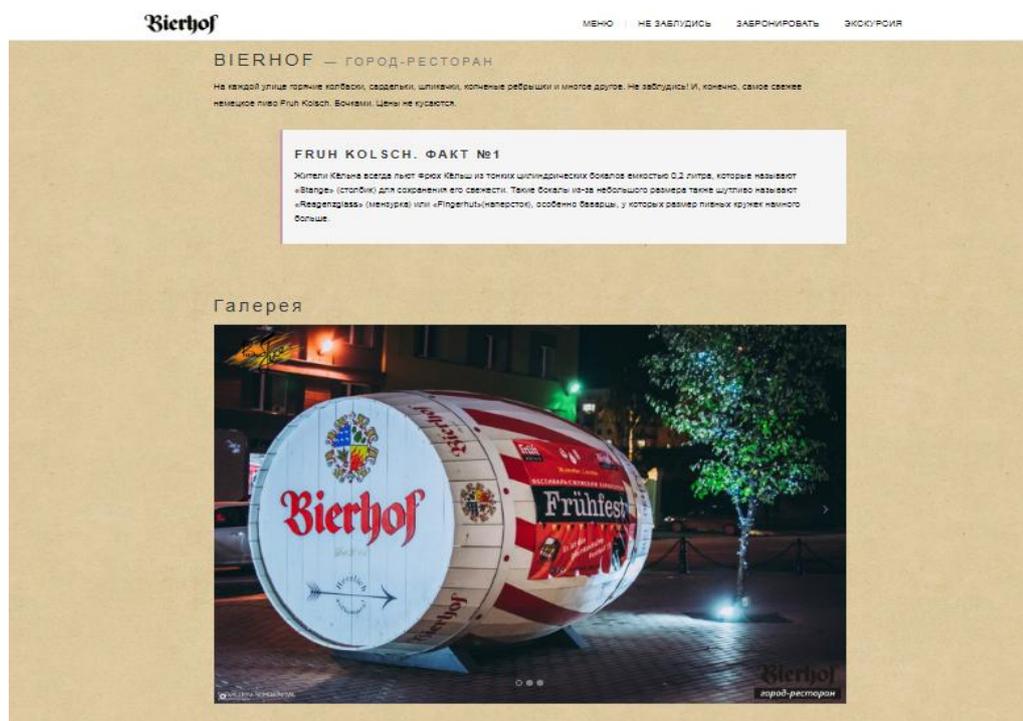


Рисунок 3.3 - Главная страница сайта



Рисунок 3.4 - Главная страница сайта (контактная информация и режим работы)

- Ознакомление с меню (рис. 3.5).

Во вкладке «Меню» на сайте «Бирхоф» можно заранее ознакомиться с меню бизнес-ланча, основным меню и с картой бара.

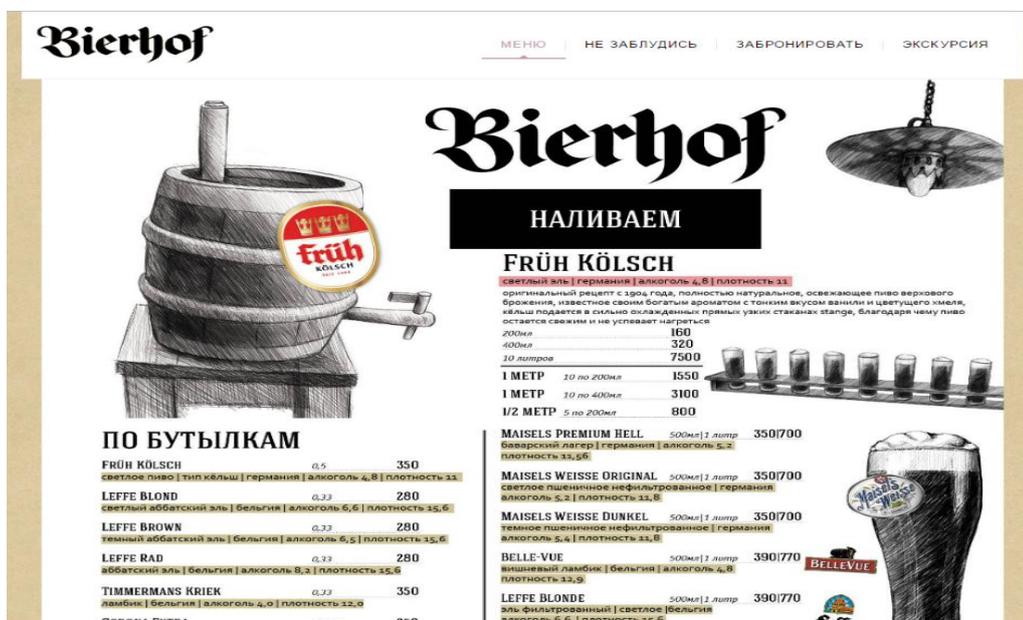


Рисунок 3.5 - Меню

- Заказ столика (рис. 3.6).

Посетителю сайта ресторана предоставляется возможность забронировать столик на определенное количество гостей, на нужную дату и время, без звонка администратору ресторана.

Рисунок 3.6 - Форма заказа столика

- Ознакомление с картой проезда к ресторану «Бирхоф» (рис. 3.7).

Вкладка «Не заблудись» содержит в себе информацию о том, как добраться до ресторана.

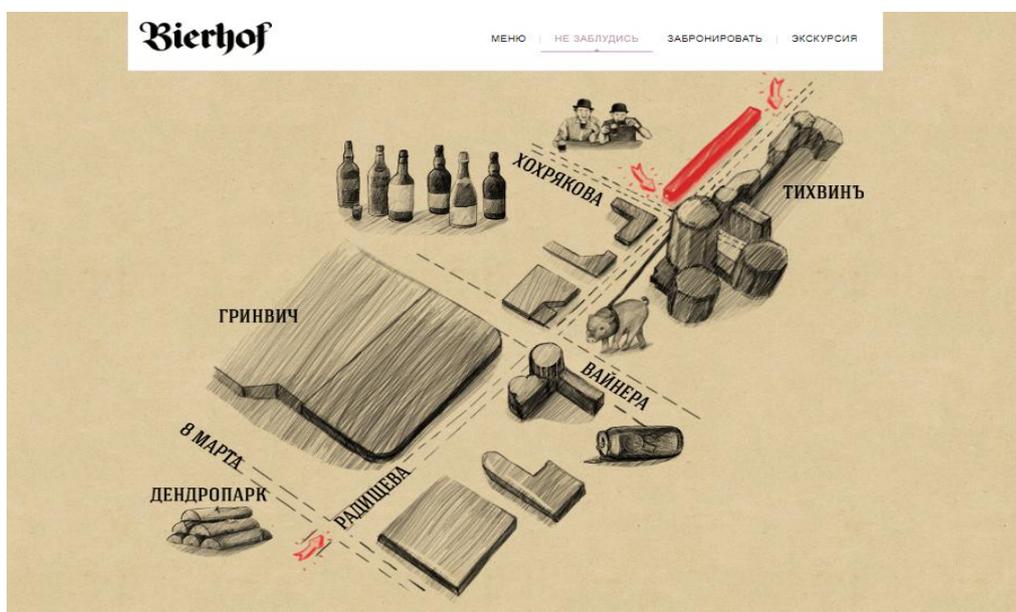


Рисунок 3.7 – Карта проезда к ресторану

- Ознакомление с картой ресторана (рис. 3.8).

Во вкладке «Экскурсия» у посетителя сайта есть возможность ознакомиться заранее с картой ресторана, понять расположение залов, расстановку столиков в этих залах, определить количеством мест и т.д.

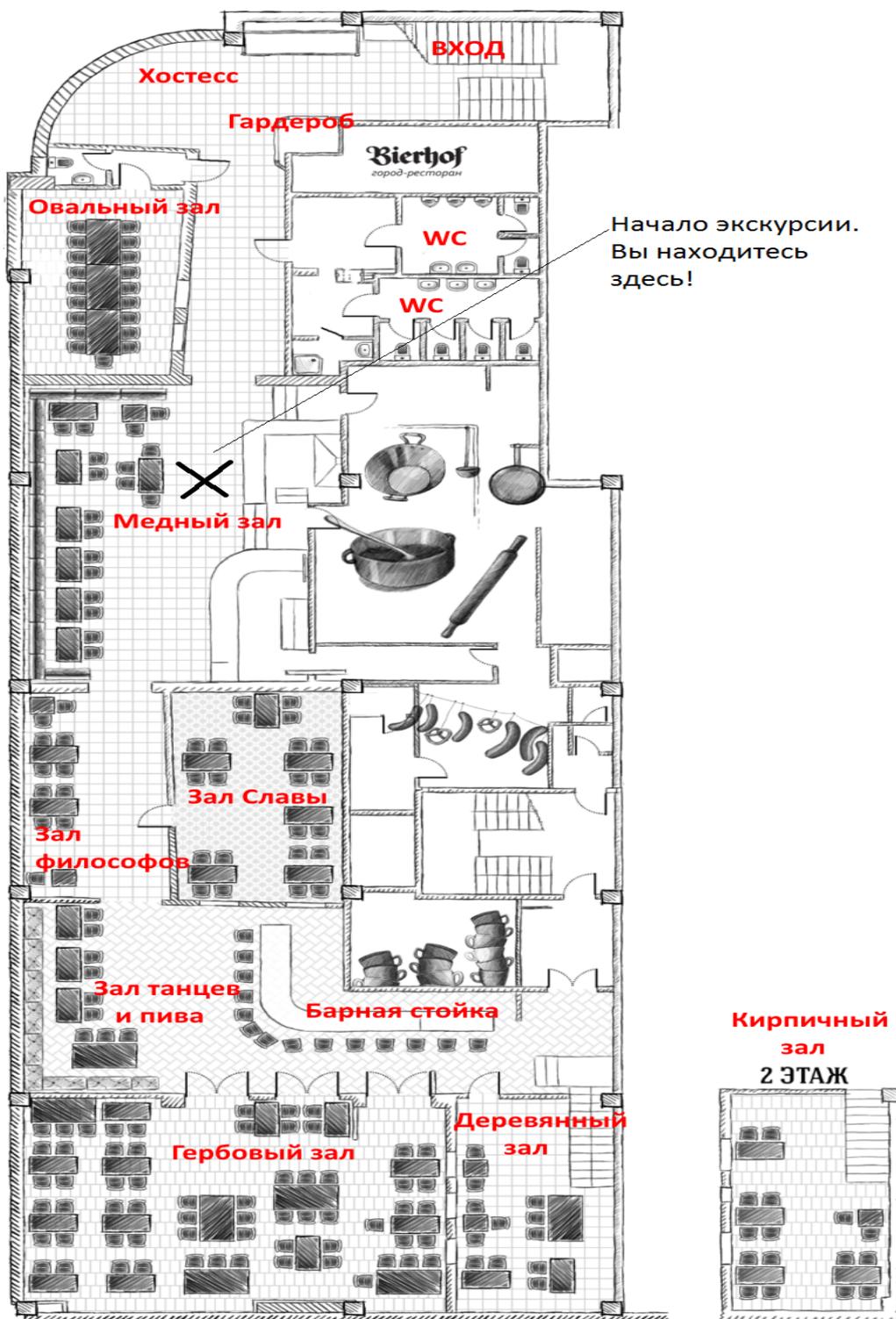


Рисунок 3.8 – Карта ресторана

- Ознакомление с панорамной экскурсией по залам ресторана.

У посетителя сайта «Бирхоф» есть уникальная возможность прогуляться по залам ресторана, не выходя из дома. Встроенную панорамную экскурсию можно найти во вкладке «Экскурсия» (рис.3.9, рис.3.10, рис.3.11, рис.3.12).

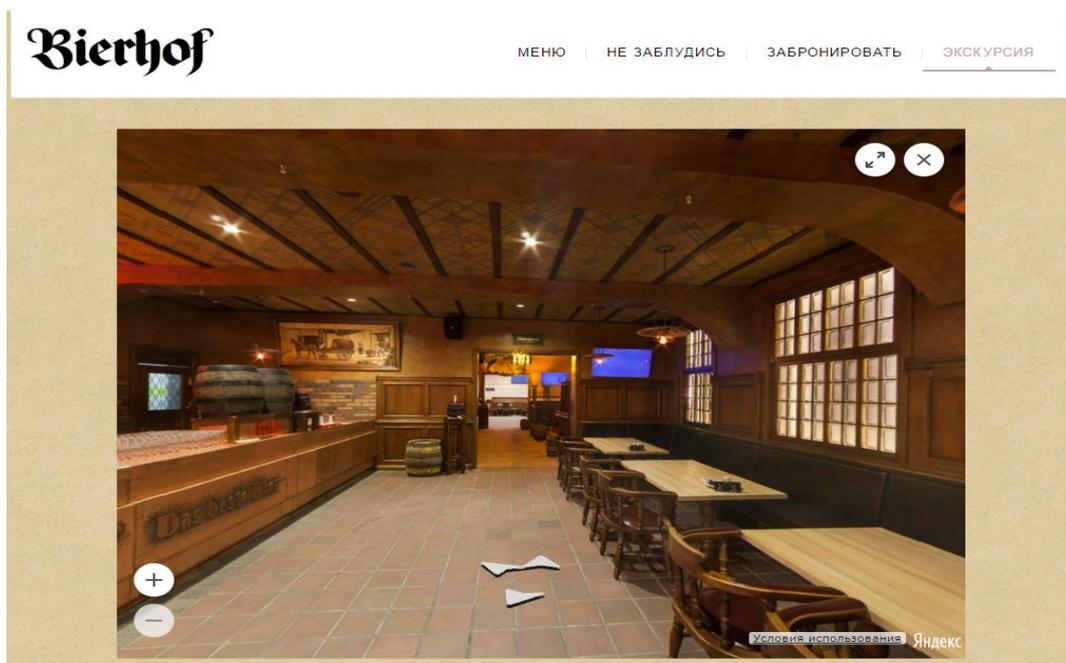


Рисунок 3.9 - Панорамы помещений ресторана («Медный» зал)

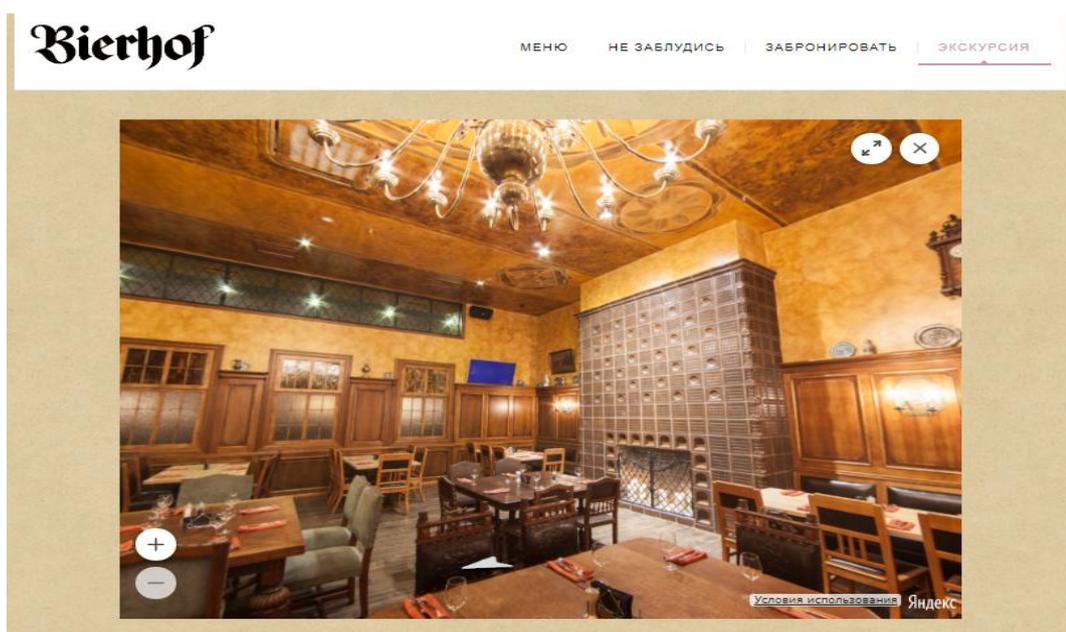


Рисунок 3.10 – Панорамы помещений ресторана («Гербовый» зал)



Рисунок 3.11 – Панорамы помещений ресторана («Деревянный» зал)



Рисунок 3.12 – Панорамы помещений ресторана («Овальный» зал)

Технические возможности панорамной экскурсии:

1. Передвижение по залам происходит при помощи специальных стрелок (рис.3.13).
2. Возможность приближения картинки встроенными кнопками «+» и «-» или используя «зум», управляя колесом компьютерной «мышки».
3. Полноэкранный презентация



Рисунок 3.13 – Стрелки, указывающие направление передвижения по залам во время панорамной экскурсии

Таким образом, разработанный сайт выполняет следующие функции:

- служит web-представительством ресторана в сети Интернет;
- знакомит посетителя с особенностями ресторана;
- сообщает контактные данные и меню ресторана.

После размещения сайта на удаленном сервере необходимо осуществлять его информационную поддержку.

Информационная поддержка сайта включает в себя следующие мероприятия:

- наполнение сайта тематическими статьями для лучшего ранжирования поисковыми машинами;
- размещение информации о работе сайта в виде новостей в блоге;
- обновление и поддержание в актуальном состоянии каталога сайта (состояние цен, проведение акций и других мероприятий по привлечению клиентов);
- работа по продвижению сайта в поисковой выдаче.

Кроме того, непосредственно управлять работой сайта должен специально назначенный сотрудник. В рассматриваемом случае

предполагается назначить одного из менеджеров ответственным за продажи на сайте. Вначале данный сотрудник должен будет совмещать свои основные обязанности с работой по сопровождению сайта.

При работе с сайтом в обязанности данного сотрудника будет входить: выполнение мероприятий по наполнению каталога сайта и сопровождение заявок, поступивших через сайт. В дальнейшем, при увеличении количества посетителей сайта, предполагается освободить администратора от выполнения других обязанностей.

3.2 Описание порядка реализации виртуальной экскурсии

Для получения панорамы необходимо выполнить все фотографии с одной освещённостью и соблюдением горизонтальной ориентации.

Далее необходимо загрузить снимки в приложение «Image Composite Editor 2.0.3.0» и на выходе получить готовые сферические панорамы. Данное приложение предназначено для автоматической склейки фотографий в различные варианты панорам (цилиндрические, сферические и т.д.).



Рисунок 3.14 Пример панорамы



Рисунок 3.15 Пример панорамы

При этом необходимо сохранять панорамы в двух разрешениях (высоком и низком), так как этого требует сервис «Панорамы» от Яндекс. После того, как панорамы будут созданы, их необходимо нарезать на сегменты размерами 256x256 пикселей, поскольку панорамы накладываются на сферу только сегментами. Нарезка панорам выполняется пакетом «ImageMagick-7.0.5» через утилиту командной строки (пути задаем свои):

```
magick.exe convert C:\WEB\Sites\htdocs\tiles\hq1.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\1\hq\[filename:tile].jpg"
magick.exe convert C:\WEB\Sites\htdocs\tiles\lq1.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\1\lq\[filename:tile].jpg"

magick.exe convert C:\WEB\Sites\htdocs\tiles\hq2.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\2\hq\[filename:tile].jpg"
magick.exe convert C:\WEB\Sites\htdocs\tiles\lq2.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\2\lq\[filename:tile].jpg"

magick.exe convert C:\WEB\Sites\htdocs\tiles\hq3.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\3\hq\[filename:tile].jpg"
magick.exe convert C:\WEB\Sites\htdocs\tiles\lq3.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\3\lq\[filename:tile].jpg"

magick.exe convert C:\WEB\Sites\htdocs\tiles\hq4.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\4\hq\[filename:tile].jpg"
magick.exe convert C:\WEB\Sites\htdocs\tiles\lq4.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\4\lq\[filename:tile].jpg"

magick.exe convert C:\WEB\Sites\htdocs\tiles\hq5.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\5\hq\[filename:tile].jpg"
magick.exe convert C:\WEB\Sites\htdocs\tiles\lq5.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\5\lq\[filename:tile].jpg"

magick.exe convert C:\WEB\Sites\htdocs\tiles\hq6.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\6\hq\[filename:tile].jpg"
magick.exe convert C:\WEB\Sites\htdocs\tiles\lq6.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\6\lq\[filename:tile].jpg"

magick.exe convert C:\WEB\Sites\htdocs\tiles\hq7.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\7\hq\[filename:tile].jpg"
magick.exe convert C:\WEB\Sites\htdocs\tiles\lq7.jpg -crop 256x256 -set filename:tile
"%[fx:page.y/256]-%[fx:page.x/256]" "C:\WEB\Sites\htdocs\tiles\7\lq\[filename:tile].jpg"
```

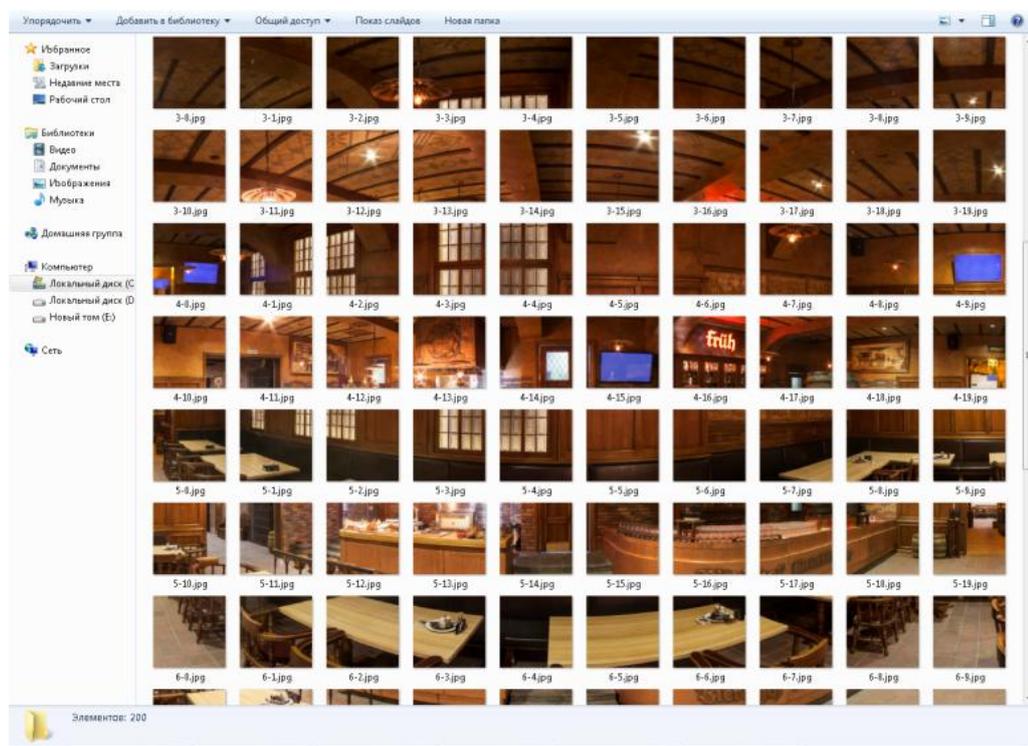


Рисунок 3.16 Фотографии

После того, как нарезка готова, создается JavaScript файл, который описывает панорамы и переходы между ними.

Файлы проекта находятся в папке «panorama». Нарезка панорам находится в папке «tiles». Файл JavaScript, описывающий панорамы и переходы между ними называется, «panorama_custom.js» (содержит комментарии с описанием кода). Исходные кода файлов приведены в Приложениях 1 и 2.

Таким образом, в третьей главе ВКР определены основные составляющие компоненты, необходимые для разработки содержания сайта, на основании чего сформулирована и построена структура сайта.

Для разработки интерфейса проанализированы и сформулированы основные требования к элементам интерфейса сайта, в том числе требования к навигации и структуре сайта.

Далее описаны основные информационные страницы сайта и их особенности. В завершении было дано описание технологии создания панорамных изображений и процесс их внедрения на web-сайт.

ЗАКЛЮЧЕНИЕ

Сфера ресторанного бизнеса активно развивается. Количество предприятий общественного питания увеличивается как в мегаполисах, так и в малонаселенных городах. В следствии этого владельцам ресторанов необходимо работать над лояльностью посетителей и над привлечением новых, путем использования различных способов информационной поддержки деятельности ресторана. И в качестве эффективной информационной поддержки предлагается использовать web-сайт.

В первой главе дипломной работы рассмотрена история формирования и развития ресторанного бизнеса, нормативная база, которая регулирует отношения предприятий общепита, а также тенденции развития данной отрасли. Успешность предприятия общественного питания тесно связана с вопросом использования информационных технологий, которые оптимизируют работу ресторана с целью повышения качества услуг. Использование web-сайта для ресторана позволяет автоматизировать некоторые процессы и дает возможность дополнительного распространения информации об услугах, которые оказывает предприятие и, соответственно, помогает наладить связь между менеджментом и клиентами.

Во второй главе проведен анализ существующих средств и технологий для проектирования web-сайтов и дано обоснование проектных решений, которые был приняты в процессе создания сайта ресторана «Бирхоф». В этой же главе описан порядок разработки карты сайта и его дизайна.

В третьей главе определены основные составляющие компоненты, необходимые для разработки содержания сайта, на основании чего сформулирована и построена структура сайта. Также в конце третьей главы была описана, разработанная технология создания панорамных изображений и их процесс внедрения на web-сайт ресторана «Бирхоф».

Таким образом, задачи, которые были поставлены в начале ВКР выполнены, а цель достигнута. Предполагается, что сайт позволит ресторану «Бирхоф» увеличить количество посетителей и расширит их географию.

СПИСОК ЛИТЕРАТУРЫ

1. Автоматизированные информационные системы, базы и банки данных. Вводный курс: Учебное пособие, М.: Гелиос АРВ, 2011. – 368 с.
2. Астелс Дэвид Практическое руководство по экстремальному программированию, Пер. с англ. / Астелс Дэвид, Миллер Гранвилл, Новак Мирослав - М.: Вильямс, 2012. - 320 с.
3. Ашманов, И. П. Оптимизация и продвижение сайтов в поисковых системах / Ашманов И.П., Иванов А. С., Питер, 2011 год, 400 с.
4. Баженова, И. Ю. Основы проектирования приложений баз данных / Баженова И. Ю. – Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2012 г., 328 с.
5. Бодров О.А., Предметно-ориентированные экономические информационные системы / Бодров О.А., Медведев Р.Е. - М, Финансы и статистика, 2012 г., 224 с.
6. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / Вендров, А.М. - М.: Финансы и статистика, 2011 г, 456 с.
7. Вигерс Карл, Разработка требований к программному обеспечению, пер, с англ. / Вигерс Карл - М.: Русская Редакция, 2012, 576 с.
8. Гвоздева, Т. В. Проектирование информационных систем / Гвоздева, Т. В., Баллод Б. А. – М: Феникс, 2012 г., 512 с.
9. Голицына О. Л., Информационные технологии / Голицына О.Л., Попов И. И., Максимов Н. В., Партыка Т. Л. - М, Издательство Инфра-М, 2012 г., 608 с.
10. ГОСТ Р 50762-95 «Классификация предприятия общественного питания».
11. ГОСТ Р 50-763-95 «Общественное питание. Кулинарная продукция, реализация населению. Общие технические условия».
12. ГОСТ Р 50-764-95 «Услуги предприятий общественного питания».

13. ГОСТ Р ИСО/МЭК 12207/99. Государственный стандарт РФ. «Информационная технология. Процессы жизненного цикла информационных систем».
14. Дейв Крейн, AJAX в действии: Учебник – М.: Вильямс, 2011. 450с. – 490с.
15. Дэвид Флэнаган, JavaScript. Подробное руководство: Учебник – М.: Символ Плюс, 2011. 243 – 249 с.
16. Емельянова, Н. З. Проектирование информационных систем / Емельянова Н. З., Партыка Т. Л., Попов И. И. - М, Издательство Форум, 2011 г., 432 с.
17. Емельянова, Н. З. Основы построения автоматизированных информационных систем / Емельянова Н. З., Партыка Т. Л., Попов И. И. - М, Издательство Форум, 2012 г., 416 с.
18. Илюшечкин, В. М. Основы использования и проектирования баз данных / Илюшечкин В. М. - М, Издательство Юрайт, 2012 г., 224 с.
19. Исследования рынков. Готовые маркетинговые исследования [Электронный ресурс] / URL: <http://marketing.rbc.ru> (дата обращения: 19.04.2017).
20. Кодекс WordPress [Электронный ресурс] / URL: <https://codex.wordpress.org> (дата обращения: 19.04.2017).
21. Котляров, В. П. Основы тестирования программного обеспечения/ Котляров В.П., Коликова Т.В. - Бином. Лаборатория знаний, 2011 г., 288 с.
22. Кристиан Дари, AJAX и PHP. Разработка динамических веб-приложений: Учебник / Кристиан Дари, Богдан Бринзаре, Филип Черчез-Тоза, Михай Бусика – М.: Символ Плюс, 2012, 289 с.
23. Критерии выбора CMS [Электронный ресурс] / URL: http://cmslist.ru/articles/kriterii_vybora_cms (дата обращения: 19.04.2017).
24. Кузин А. В., Базы данных / Кузин А.С., Левонисова С.В. - М, Академия, 2011 г., 320 с.

25. Кузнецов С. Д., Основы баз данных / Кузнецов С.Д. - М, Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2012 г., 488 с.
26. Любимова Л. А., Теоретические аспекты обеспечения конкурентоспособности предприятий непроизводственной сферы / Любимова Л. А., Морева С. Н. - Непроизводственная сфера в социально ориентированной рыночной экономике: коллективная монография. Тамбов, 2012.
27. Незнанов, А.А. Программирование и алгоритмизация / Незнанов А.А. - М, Академия, 2012 г., 304 с.
28. Пирогов В. Ю., Информационные системы и базы данных. Организация и проектирование / Пирогов В.Ю. - БХВ-Петербург, 2011 г., 528с.
29. Потеряхин, А. Лояльность персонала как условие профессиональной мотивации [Электронный ресурс] / URL: <http://www.hr-portal.ru> (дата обращения: 05.05.2017).
30. Российский рынок общественного питания [Электронный ресурс] / URL: <http://carbofood.ru/obzory-rynkov/rossiiskii-rynok-obshestvennogo-pitaniia-2013> (дата обращения: 05.05.2017).
31. Свой ресторан: подводные камни ресторанного бизнеса [Электронный ресурс] / портал бизнес-планов и руководств для малого и среднего бизнеса / URL: <http://www.openbusiness.ru/html/dop5/piter-rest1.html> (дата обращения 05.05.2017).
32. Средства создания сайтов [Электронный ресурс] / URL: <http://www.in-internet.narod.ru/teor/sreda.html> (дата обращения: 19.04.2017).
33. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. ГОСТ 19.701-90 (ИСО 5807-85) / Государственный комитет СССР по управлению качеством продукции и стандартам, 01.01.1992.

34. Экономическая эффективность внедрения ИТ-системы [Электронный ресурс] / URL: <https://studfiles.net> (дата обращения: 05.05.2017).

35. Энди Орам, Идеальная разработка ПО. Рецепты лучших программистов / Энди Орам, Грегори Уилсон - Издательство: Питер, Спб, 2013 год, 592 с.

36. Энтони Молинаро, SQL. Сборник рецептов / Энтоним Молинаро - Издательство: Символ-Плюс, 2011 г., 672 с.

37. Эрих Гамма, Приемы объектно-ориентированного проектирования. Паттерны проектирования / Эрих Гамма, Ричард Хелм, Ральф Джонсон, Джон Влссидес - Издательство: Питер, М., 2013 год, 368 с.

Приложение 1. Исходный код файла «panorama_custom.js»

```
ymaps.ready(function () {
    // Для начала проверим, поддерживает ли плеер браузер пользователя.
    if (!ymaps.panorama.isSupported()) {
        // Если нет, то ничего не будем делать.
        return;
    }
    // Создаем объект, содержащий данные связанных панорам.
    var panoData = {
        // Данные первой панорамы.
        Pano1: {
            type: 'custom',
            angularBBox: [0.5 * Math.PI, 2 * Math.PI, -0.5 * Math.PI, 0],
            position: [0, 0, 0],
            tileSize: [256, 256],
            tileLevels: [
                {
                    // URL тайлов для высокого уровня детализации.
                    getTileUrl: function (x, y) {
                        return '/panorama/tiles/1/lq/' + y + '-' + x + '.jpg';
                    },
                    // Размер изображения для высокого уровня детализации.
                    getImageSize: function () {
                        return [1024, 512];
                    }
                },
                {
                    // URL тайлов для высокого уровня детализации.
                    getTileUrl: function (x, y) {
                        return '/panorama/tiles/1/hq/' + y + '-' + x + '.jpg';
                    },
                    // Размер изображения для высокого уровня детализации.
                    getImageSize: function () {
                        return [5120, 2560];
                    }
                }
            ],
            // Переходы на панораме по стандартной стрелке.
            thoroughfares: [
                {
                    panoID: 'Pano2',
                    direction: [(0/180)*Math.PI, 0]
                },
                {
                    panoID: 'Pano7',
                    direction: [-(45/180)*Math.PI, 0]
                },
                {
                    panoID: 'Pano5',
                    direction: [(200/180)*Math.PI, 0]
                }
            ],
            // Переходы на панораме через маркеры.
            markerConnections: [/*{
                panoID: 'Pano2',
                iconSrc: {
                    'default': '/panorama/images/marker-info-default.png',
                    'hovered': '/panorama/images/marker-info-hovered.png'
                },
                iconPosition: [-0.36, -0.8, 0]
            }*/]
        },
        // Данные второй панорамы.
        Pano2: {
            type: 'custom',
            angularBBox: [0.5 * Math.PI, 2 * Math.PI, -0.5 * Math.PI, 0],
            position: [0, 0, 0],
            tileSize: [256, 256],
            tileLevels: [{
                // URL тайлов для высокого уровня детализации.
                getTileUrl: function (x, y) {
                    return '/panorama/tiles/2/lq/' + y + '-' + x + '.jpg';
                },
                // Размер изображения для высокого уровня детализации.
                getImageSize: function () {
                    return [1024, 512];
                }
            }
        ]
    }
}
```

```

    },
    {
        // URL тайлов для высокого уровня детализации.
        getTileUrl: function (x, y) {
            return '/panorama/tiles/2/hq/' + y + '-' + x + '.jpg';
        },
        // Размер изображения для высокого уровня детализации.
        getImageSize: function () {
            return [5120, 2560];
        }
    }
],
// Добавим обратный переход на первую панораму.
thoroughfares: [
    {
        panoID: 'Pano1',
        direction: [(45/180)*Math.PI, 0]
    },
    {
        panoID: 'Pano3',
        direction: [(230/180)*Math.PI, 0]
    },
    {
        panoID: 'Pano4',
        direction: [(180/180)*Math.PI, 0]
    },
    {
        panoID: 'Pano6',
        direction: [(130/180)*Math.PI, 0]
    },
    {
        panoID: 'Pano7',
        direction: [(90/180)*Math.PI, 0]
    }
],
markerConnections: []
},
Pano3: {
    type: 'custom',
    angularBBox: [0.5 * Math.PI, 2 * Math.PI, -0.5 * Math.PI, 0],
    position: [0, 0, 0],
    tileSize: [256, 256],
    tileLevels: [{
        // URL тайлов для высокого уровня детализации.
        getTileUrl: function (x, y) {
            return '/panorama/tiles/3/lq/' + y + '-' + x + '.jpg';
        },
        // Размер изображения для высокого уровня детализации.
        getImageSize: function () {
            return [1024, 512];
        }
    }
],
    {
        // URL тайлов для высокого уровня детализации.
        getTileUrl: function (x, y) {
            return '/panorama/tiles/3/hq/' + y + '-' + x + '.jpg';
        },
        // Размер изображения для высокого уровня детализации.
        getImageSize: function () {
            return [5120, 2560];
        }
    }
],
// Переходы на панораме по стандартной стрелке.
thoroughfares: [
    {
        panoID: 'Pano2',
        direction: [(35/180)*Math.PI, 0]
    }
],
// Переходы на панораме через маркеры.
markerConnections: []
},
Pano4: {
    type: 'custom',
    angularBBox: [0.5 * Math.PI, 2 * Math.PI, -0.5 * Math.PI, 0],
    position: [0, 0, 0],
    tileSize: [256, 256],

```

```

tileLevels: [{
  // URL тайлов для высокого уровня детализации.
  getTileUrl: function (x, y) {
    return '/panorama/tiles/4/lq/' + y + '-' + x + '.jpg';
  },
  // Размер изображения для высокого уровня детализации.
  getImageSize: function () {
    return [1024, 512];
  }
}],
  {
  // URL тайлов для высокого уровня детализации.
  getTileUrl: function (x, y) {
    return '/panorama/tiles/4/hq/' + y + '-' + x + '.jpg';
  },
  // Размер изображения для высокого уровня детализации.
  getImageSize: function () {
    return [5120, 2560];
  }
}
],
// Переходы на панораме по стандартной стрелке.
thoroughfares: [
  {
    panoID: 'Pano2',
    direction: [-(110/180)*Math.PI, 0]
  },
],
// Переходы на панораме через маркеры.
markerConnections: [],
  getDefaultDirection: function () {
    return [180, 180];
  },
},

Pano5: {
type: 'custom',
angularBBox: [0.5 * Math.PI, 2 * Math.PI, -0.5 * Math.PI, 0],
position: [0, 0, 0],
tileSize: [256, 256],
tileLevels: [{
  // URL тайлов для высокого уровня детализации.
  getTileUrl: function (x, y) {
    return '/panorama/tiles/5/lq/' + y + '-' + x + '.jpg';
  },
  // Размер изображения для высокого уровня детализации.
  getImageSize: function () {
    return [1024, 512];
  }
}],
  {
  // URL тайлов для высокого уровня детализации.
  getTileUrl: function (x, y) {
    return '/panorama/tiles/5/hq/' + y + '-' + x + '.jpg';
  },
  // Размер изображения для высокого уровня детализации.
  getImageSize: function () {
    return [5120, 2560];
  }
}
],
// Переходы на панораме по стандартной стрелке.
thoroughfares: [
  {
    panoID: 'Pano1',
    direction: [(10/180)*Math.PI, 0]
  },
],
// Переходы на панораме через маркеры.
markerConnections: [],
  getDefaultDirection: function () {
    return [180, 180];
  },
},

Pano6: {
type: 'custom',
angularBBox: [0.5 * Math.PI, 2 * Math.PI, -0.5 * Math.PI, 0],
position: [0, 0, 0],
tileSize: [256, 256],

```

```

tileLevels: [{
  // URL тайлов для высокого уровня детализации.
  getTileUrl: function (x, y) {
    return '/panorama/tiles/6/lq/' + y + '-' + x + '.jpg';
  },
  // Размер изображения для высокого уровня детализации.
  getImageSize: function () {
    return [1024, 512];
  }
},
  {
  // URL тайлов для высокого уровня детализации.
  getTileUrl: function (x, y) {
    return '/panorama/tiles/6/hq/' + y + '-' + x + '.jpg';
  },
  // Размер изображения для высокого уровня детализации.
  getImageSize: function () {
    return [5120, 2560];
  }
}
],
// Переходы на панораме по стандартной стрелке.
thoroughfares: [
  {
    panoID: 'Pano2',
    direction: [-(140/180)*Math.PI, 0]
  },
],
// Переходы на панораме через маркеры.
markerConnections: [],
  getDefaultDirection: function () {
    return [180, 180];
  },
},

Pano7: {
type: 'custom',
angularBBox: [0.5 * Math.PI, 2 * Math.PI, -0.5 * Math.PI, 0],
position: [0, 0, 0],
tileSize: [256, 256],
tileLevels: [{
  // URL тайлов для высокого уровня детализации.
  getTileUrl: function (x, y) {
    return '/panorama/tiles/7/lq/' + y + '-' + x + '.jpg';
  },
  // Размер изображения для высокого уровня детализации.
  getImageSize: function () {
    return [1024, 512];
  }
},
  {
  // URL тайлов для высокого уровня детализации.
  getTileUrl: function (x, y) {
    return '/panorama/tiles/7/hq/' + y + '-' + x + '.jpg';
  },
  // Размер изображения для высокого уровня детализации.
  getImageSize: function () {
    return [5120, 2560];
  }
}
],
// Переходы на панораме по стандартной стрелке.
thoroughfares: [
  {
    panoID: 'Pano2',
    direction: [-(110/180)*Math.PI, 0]
  },
  {
    panoID: 'Pano1',
    direction: [-(40/180)*Math.PI, 0]
  },
],
// Переходы на панораме через маркеры.
markerConnections: [],
  getDefaultDirection: function () {
    return [180, 180];
  },
},
},
};

```

```

// Функция для извлечения данных нужной панорамы из объекта panoData.
function getConnectedPanoramaData(panoID) {
    return panoData[panoID];
}
// Функция, загружающая изображение маркера с сервера.
function loadImage(src) {
    return new ymaps.vow.Promise(function (resolve) {
        var image = new Image();
        image.onload = function () {
            resolve(image);
        };
        image.crossOrigin = 'anonymous';
        image.src = src;
    });
}

// Создаем класс, описывающий переход между панорамами по стандартной стрелке.
function Thoroughfare(currentPanorama, direction, nextPanorama) {
    this.properties = new ymaps.data.Manager();
    this._currentPanorama = currentPanorama;
    this._direction = direction;
    this._connectedPanorama = nextPanorama;
}

ymaps.util.defineClass(Thoroughfare, {
    getConnectedPanorama: function () {
        // Если переход будет осуществляться на пользовательскую панораму,
        // то создаем объект панорамы MyPanorama.
        // Если нужно перейти на Яндекс.Панораму, то для получения объекта
        // панорамы воспользуемся функцией ymaps.panorama.locate.
        if (this._connectedPanorama.type == 'custom') {
            return ymaps.vow.resolve(new MyPanorama(this._connectedPanorama));
        } else if (this._connectedPanorama.type == 'yandex') {
            return ymaps.panorama.locate(this._connectedPanorama.coords).then(
                function(panoramas) {
                    if (panoramas.length) {
                        return panoramas[0];
                    } else {
                        return ymaps.vow.reject(new Error('Панорама не нашлась.'));
                    }
                }
            );
        }
    },
    // Направление взгляда на панораму, на которую будет осуществляться переход.
    getDirection: function () {
        return this._direction;
    },
    // Ссылка на текущую панораму, из которой осуществляется переход.
    getPanorama: function () {
        return this._currentPanorama;
    }
});

// Создаем класс, описывающий маркер-переход.
function MarkerConnection(currentPanorama, imgSrc, position, nextPanorama) {
    // В классе должно быть определено поле properties.
    this.properties = new ymaps.data.Manager();
    this._panorama = currentPanorama;
    this._position = position;
    this._imgSrc = imgSrc;
    this._connectedPanorama = nextPanorama;
}

ymaps.util.defineClass(MarkerConnection, {
    getIconSet: function () {
        return ymaps.vow.Promise.all([
            loadImage(this._imgSrc.default),
            loadImage(this._imgSrc.hovered)
        ]).spread(function (defaultImage, hoveredImage) {
            return {
                'default': {
                    image: defaultImage,
                    offset: [0, 0]
                },
                hovered: {
                    image: hoveredImage,
                    offset: [0, 0]
                }
            };
        });
    }
});

```

```

    });
  },
  // Текущая панорама, из которой осуществляется переход.
  getPanorama: function () {
    return this._panorama;
  },
  // Позиция маркера на текущей панораме.
  getPosition: function () {
    return this._position;
  },
  // Чтобы по клику на маркер осуществлялся переход на другую панораму,
  // реализуем метод getConnectedPanorama.
  getConnectedPanorama: function () {
    if (this._connectedPanorama.type == 'custom') {
      return ymaps.vow.resolve(new MyPanorama(this._connectedPanorama));
    } else if (this._connectedPanorama.type == 'yandex') {
      return ymaps.panorama.locate(this._connectedPanorama.coords).then(
        function(panoramas) {
          if (panoramas.length) {
            return panoramas[0];
          } else {
            return ymaps.vow.reject(new Error('Панорама не нашлась.'));
          }
        }
      );
    }
  }
});

// Класс панорамы.
function MyPanorama(obj) {
  ymaps.panorama.Base.call(this);
  this._angularBBox = obj.angularBBox;
  this._position = obj.position;
  this._tileSize = obj.tileSize;
  this._tileLevels = obj.tileLevels;
  // Получаем массив экземпляров класса, описывающего переход по стрелке из
  // одной панорамы на другую.
  this._thoroughfares = obj.thoroughfares.map(function (thoroughfare) {
    return new Thoroughfare(
      this, // Текущая панорама.
      thoroughfare.direction, // Направление взгляда на панораму, на которую делаем
переход.
      getConnectedPanoramaData(thoroughfare.panoID) // Данные панорамы, на которую
делаем переход.
    );
  }, this);
  // Получаем массив маркеров-переходов.
  this._connections = obj.markerConnections.map(function (marker) {
    return new MarkerConnection(
      this, // Текущая панорама.
      marker.iconSrc, // Изображение маркера.
      marker.iconPosition, // Позиция маркера.
      getConnectedPanoramaData(marker.panoID) // Данные панорамы, на которую делаем
переход.
    );
  }, this);
}

ymaps.util.defineClass(MyPanorama, ymaps.panorama.Base, {
  // Чтобы добавить на панораму стандартные стрелки переходов,
  // реализуем метод getThoroughfares.
  getThoroughfares: function () {
    return this._thoroughfares;
  },
  // Чтобы добавить на панораму маркеры-переходы,
  // нужно реализовать метод getConnections.
  getConnections: function () {
    return this._connections;
  },
  getAngularBBox: function () {
    return this._angularBBox;
  },
  getPosition: function () {
    return this._position;
  },
  getTileSize: function () {
    return this._tileSize;
  },
});

```

```
    getTileLevels: function () {
        return this._tileLevels;
    },
    getCoordSystem: function () {
        return ymaps.coordSystem.cartesian;
    }
});

var panorama = new MyPanorama(panoData.Pano1);

// Отображаем панораму на странице.
var player = new ymaps.panorama.Player('panoramaplayer', panorama, {
    direction: [0, 0],
    span: [100,100],

});
});
```

Приложение 2. Исходный код файла «index.html» (подключение панорамы в HTML-коде страницы)

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <title>Yandex Panorama</title>
  <meta charset="utf-8">

  <style>
    html, body, .panoramaplayer {
      width: 100%;
      height: 100%;
      margin: 0;
      padding: 0;
    }
  </style>

  <script src="https://api-maps.yandex.ru/2.1/?lang=ru_RU&load=package.full"
type="text/javascript"></script>
  <script src="panorama_custom.js"></script>
</head>
<body>
  <div id="panoramaplayer" class="panoramaplayer"></div>
</body>
</html>
```

Приложение 3. Список условных обозначений

БД – база данных

ИС – информационная система

ОС – операционная система

ПК – персональный компьютер

ПО – программное обеспечение

СМИ – средства массовой информации

СУБД – система управления базами данных

ЦА – целевая аудитория

AJAX – Asynchronous Javascript

CMS – Content management system

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

IIS – Internet Information Services

SEO – Search Engine Optimization

PHP – Personal Home Page

URL – Uniform Resource Locator

XML – eXtensible Markup Language