

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

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК

КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ

Разработка информационной системы для ООО «ЭНКОМ» по учёту заказов
потребителей

Выпускная квалификационная работа
обучающегося по направлению подготовки 02.03.02 Фундаментальная
информатика и информационные технологии
очной формы обучения, группы 07001401
Горбачева Дмитрия Павловича

Научный руководитель
старший преподаватель кафедры
МиПОИС,
Ерошенко Яна Борисовна.

БЕЛГОРОД 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. Программные средства разработки и предметная область.....	5
1.1 Понятие разработки клиент-серверного приложения.....	5
1.2 СУБД MySQL и утилита phpMyAdmin	12
1.3 HTML.....	13
1.4 JavaScript	14
1.5 PHP.....	15
1.6 AJAX.....	16
1.7 Бизнес-логика.....	18
1.8 ООО "ЭНКОН"	21
2. Проектирование приложения.....	22
2.1 Создание базы данных	22
2.2 Модульная схема приложения	24
3. Реализация клиент-серверного приложения	26
ЗАКЛЮЧЕНИЕ	32
СПИСОК ЛИТЕРАТУРЫ.....	33
ПРИЛОЖЕНИЕ	37

ВВЕДЕНИЕ

В настоящее время во многих сферах жизни внедряются современные технологии, что подразумевает реализацию множества задач на ЭВМ. В силу быстрого научного прогресса и темпов развития общества, количество информации возрастает. В результате этого возникает потребность в организации информации. Один из способов организации – создание клиент-серверных приложений. В частности, с помощью таких приложений решаются задачи, возникающие в сфере бизнеса. К таким задачам относятся расчет стоимости товара, обработка торговых транзакций, учет товара, клиентов и средств. В данной работе реализована клиент-серверная информационная система для учета заказов потребителей. Такая система актуальна в современных реалиях и может быть плодотворно внедрена в работу компании.

Целью выпускной квалификационной работы является разработка информационной системы по учету заказов потребителей.

В область задач входят следующие направления деятельности:

- Выполнить формальную постановку задачи;
- Провести анализ предметной области;
- Построить логическую и физическую модели БД;
- Создать БД;
- Разработать web-приложение.

В работе приведена информационная система по учёту заказов потребителей для ООО «ЭНКОМ». При разработке были использованы Java Script, PHP и СУБД MySQL, а также реализованы понятный, удобный интерфейс, богатый функционал и поддержка взаимодействия с базой данных.

Данная работа содержит 3 основных главы:

1. Первая глава – проектирование Web-приложения. В

данной главе описаны программные средства и непосредственно само проектирование приложения.

2. Вторая глава – разработка Web-приложения. В данной главе было рассмотрено создание базы данных и структуры приложения.

3. Третья глава – реализация Web-приложения. Здесь приведены твердые копии экрана, отражающие работу приложения.

Данная выпускная квалификационная работа содержит 52 страницы, 12 изображений, 3 листинга кода, 1 таблицу и 1 приложение.

1. Программные средства разработки и предметная область

1.1 Понятие разработки клиент-серверного приложения

Информация, доступная пользователям Internet, располагается на компьютерах (Web-серверах), на которых установлено специальное программное обеспечение. Значительная часть этой информации организована в виде Web-сайтов. Web-сайт – это информация, представленная в определенном виде, которая располагается на Web-сервере и имеет свое имя (адрес)[1].

Для просмотра Web-сайтов на компьютере пользователя используются специальные программы, которые называются браузерами. Наиболее распространенные браузеры: Opera, Mozilla Firefox, Flock, Internet Explorer, Maxthon.

В зависимости от того, какое имя (адрес) сайта мы зададим в строке "Адрес", браузер будет загружать в свое окно соответствующую информацию. Web-сайт состоит из связанных между собой Web-страниц. Web-страница представляет собой текстовый файл с расширением *.htm, который содержит текстовую информацию и специальные команды – HTML-коды, определяющие в каком виде эта информация будет отображаться в окне браузера.

Вся графическая, аудио - и видео-информация непосредственно в Web-страницу не входит и представляет собой отдельные файлы с расширениями *.gif, *.jpg (графика), *.mid, *.mp3 (звук), *.avi (видео).

В HTML-коде страницы содержатся только указания на такие файлы. Каждая страница Web-сайта также имеет свой Internet адрес, который состоит из адреса сайта и имени файла, соответствующего данной странице.

Таким образом, Web-сайт – это информационный ресурс, состоящий из связанных между собой гипертекстовых документов (Web-страниц), размещенный на Web-сервере и имеющий индивидуальный адрес. Вопрос:

Что нужно для того, чтобы посмотреть какой-либо Web-сайт? Ответ: Посмотреть Web-сайт может любой человек, имеющий компьютер, подключенный к Internet, Web-браузер и адрес сайта.

В настоящее время во всемирной паутине размещено несколько миллионов Web-сайтов и их число постоянно растет.

Кроме Web-браузеров, необходимо иметь программы для редактирования HTML-документов. Сейчас на рынке программного обеспечения представлено большое количество продуктов для создания Web-страниц или HTML-документов: от мощнейших профессиональных пакетов типа FrontPage и Dreamweaver до таких популярных программ семейства текстовых процессоров, как Microsoft Word Pad, Microsoft Word и Corel WordPerfect.

Достаточно востребованным для создания Web-сайтов является ряд визуальных приложений, называемых WYSIWYG-редакторами (What You See Is What You Get — что видишь, то и получишь). Эти программы ускоряют создание Web-документов и избавляют от необходимости вручную набирать основной программный HTML-код. Все необходимые операции выполняются автоматически.

Так же Web-страницы можно создавать вручную с помощью языка HTML (HyperText Mark-up Language – язык разметки гипертекста), при этом ввод HTML-кода выполняется в любом текстовом редакторе или с помощью HTML-редакторов. Для создания элементов страниц используются следующие инструментальные средства: графические редакторы (Adobe Photoshop, Fireworks, Paint Shop Pro, Painter и др.) для создания графических файлов, текстовые редакторы для создания текстов, звуковые редакторы для создания звуковых файлов. Обычно новичок думает следующим образом: «Все свои привычные документы я создаю посредством программ WYSIWYG (<что вижу, то и получаю>) - тексты я создаю в MS Word, презентации - в MS PowerPoint, так возьму-ка я и для создания web-

странички подобную программу - MS FrontPage». Приняв такое решение, вы дважды обкрадываете себя[3].

Первый раз - в смысле рационального использования web-пространства. Дело в том, что все визуальные редакторы web-страниц, к которым относится и упомянутый MS FrontPage, вставляют в создаваемые страницы отсебятину - множество лишних ненужных тегов. Исключением, пожалуй, является Macromedia Dreamweaver (за что он снискал себе заслуженную популярность, как среди новичков, так и среди профессионалов). Но даже он в этом плане не идеален - любит засорять исходный текст кавычками (в большинстве случаев совершенно ненужными), а также вставлять символы неразрывного пробела в самых неподходящих для этого местах. Справедливости ради стоит отметить, что все визуальные редакторы предоставляют пользователю возможность работать с исходным кодом создаваемой страницы, но столь любимый многими FrontPage вновь переделает все по-своему, стоит вам только переключиться снова в визуальный режим.

Выделяют следующие этапы разработки Web-сайта: - планирование, реализация, публикация, продвижение, поддержка.

На стадии планирования определяется следующее:

- цели создания Web-сайта (Зачем? Какие задачи он должен выполнять и на какую аудиторию он рассчитан?);
- характер содержимого;
- структура (Юзабилити – удобство пользования);
- особенности оформления (определяется структура каждой страницы и разрабатывается графика).

Любую страницу можно оценить по трем параметрам: контенту, внешнему виду и навигации. Если сайт рассчитан на долгое вдумчивое чтение, то он должен иметь хорошую читабельность, меньше отвлекающих динамических эффектов, не утомляющее цветовое сочетание фона и текста. И наоборот: сократите текстовые блоки до минимума, если Вы создаете сайт,

дающий посетителю, прежде всего визуальную и другую мультимедийную информацию. При создании достаточно больших документов надо помнить о том, что не все из посетителей имеют высокоскоростной доступ в Internet.

Реализация. Это и есть работа по созданию сайта. На этом этапе проводится подготовка текстового и графического материала (печать, сканирование). Материал разбивается по файлам в соответствии со структурой. Организуются ссылки между файлами сайта. Рекомендуется создать шаблон-заготовку страницы с основными структурными областями и стилевым оформлением и использовать ее для создания всех страниц узла. Меняйте в каждой новой странице только содержимое и адресацию ссылок, такая организация работы сократит время, потраченное на каждую из них.

Тестирование. Завершив работу по размещению страниц на Web-сайте, необходимо выполнить тестирование. Оно состоит из двух этапов: тестирование на работоспособность и тестирование на удобство пользования интерфейсом. На этапе тестирования на работоспособность проверяют, как функционирует Web-сайт, используя те же условия, при которых с ним будет работать пользователь. Поработайте с Web-сайтом в различных браузерах и посмотрите, как выглядит Ваш сайт в каждом из них. Постарайтесь оценить время загрузки страниц, что очень важно[7].

Публикация. Готовый Web-сайт необходимо опубликовать на Web-сервере, чтобы он был доступен через Internet. Если ваш сайт создан посредством редактора FrontPage, то на сервере должны быть установлены серверные расширения FrontPage, что обеспечит полную поддержку доступных в FrontPage компонентов, которые были помещены на странице в процессе создания сайта. Если у вас нет собственного сервера, то в Сети можно найти огромное количество ссылок на free web pages, где некоторые провайдеры предоставляют своим клиентам бесплатное место под страницу.

Однако для того чтобы окончательно выбрать сервер для размещения своего сайта необходимо обратить внимание на следующее:

- отсутствие дискриминации по контенту. Некоторые сервера ставят условия, касающиеся содержимого Вашего сайта;
- надежность Web-сервера. Посетите их Web-узел в пиковые часы и убедитесь, что проблем с доступом не возникает;
- отсутствие скрытых завуалированных видов оплаты места в виде всевозможных доплат за различные услуги и рекламы;
- техническая поддержка. Предоставят ли Вам возможность общения с администратором сервера, есть ли возможность доступа по протоколу FTP (File Transfer Protocol) и т.д.;
- объем выделяемого Вам пространства. Обычно это 5–10 мегабайт;
- дополнительные возможности. Не будет лишним, если сервер предложит Вам: бесплатную почту, счетчик посещений, гостевую книгу, чат, различные CGI-скрипты и т.д.;
- какое программное обеспечение установлено на сервере. Например, UNIX-сервера чувствительны к регистру букв в названиях файлов, т.е. если у Вас в исходном коде будет стоять ссылка на файл background.gif, а файл будет назван background.Gif или Background.gif, то он не будет визуализирован. Однако у таких серверов более высокая степень защищенности от взломов;
- есть ли на сервере автоматическая перекодировка содержимого страниц, в соответствии с пользовательскими настройками браузера. Если таковой нет, то у Вас могут возникнуть проблемы с отображением русских кодировок.

Рекламирование сайта. Существует множество приемов рекламирования сайта: размещение информации о нем на поисковом Web-сайте, организация взаимных ссылок с другими сайтами и т.д. Как привлечь пользователя? Красиво оформленные страницы Web-сайта – это только половина дела. В первую очередь страницы должны быть содержательными. Основное требование к содержимому Web-сайта – полнота и достоверность.

Информация должна быть представлена таким образом, чтобы пользователь, однажды посетивший Web-сайт, еще не раз обратился к нему.

Сопровождение сайта. Содержимое Web-сайта может подвергаться неоднократным изменениям. Важно, чтобы предоставляемая на Web-сайте информация всегда была актуальной, поэтому как можно чаще обновляйте информацию на своем Web-сайте, по возможности расширяйте материал, улучшайте дизайн. Обязательное правило. Web-сайт должен обновляться не реже одного раза в месяц. В противном случае вы потеряете не только потенциальных, но и уже состоявшихся посетителей. Рекомендуется создать на своем компьютере копию Web-сайта, вносить в нее изменения и новую версию передавать для размещения на сервере в завершенном виде.

Страница сайта является документом, созданным на основе языка HTML. Это, по сути, и есть Internet-страничка при условии, что ее файл будет размещен в Internet. Значки “<” и “>”. Эти элементы в угловых скобках называются HTML-дескрипторами или HTML-тегами. Именно дескрипторы сообщают Web-браузеру информацию о структуре и особенностях форматирования Web-страницы. Большинство дескрипторов являются парными, т.е. состоят из открывающей и закрывающей частей и воздействуют на текст, помещенный между ними. Обратите внимание, что признак закрывающего дескриптора — обратная косая черта ” / “. Следующий важный термин, относящийся к основным понятиям языка HTML,— элемент. Элемент — конструкция языка HTML. Он состоит из открывающего дескриптора, содержания и закрывающего дескриптора. Содержанием называется все, что находится между открывающим и закрывающим дескрипторами. Атрибуты — дополнительные к дескрипторам параметры или свойства элементов (например, цвет или размер шрифта для одного абзаца, где абзац является элементом).

Структура документа HTML. Согласно спецификациям HTML 4.01, разработанным W3C, для всех документов HTML обязательны следующие условия.

В документе должна присутствовать директива (или пролог, как его иногда называют) DOCTYPE (Document Type Declaration). Пролог содержит объявление, информирующее о том, какая именно версия HTML использовалась при составлении Web-страницы, и указывает Web-браузеру, каким образом обрабатывать данный документ. Пролог представляет собой одиночный дескриптор специального вида. Данный дескриптор всегда вставляется в самом начале HTML-документа перед открывающим дескриптором `<html>`.

Запомните, что `<!doctype` — единственный дескриптор, который пишется прописными буквами. Все остальные дескрипторы нужно набирать только строчными буквами!

HTML-документ всегда должен начинаться открывающим дескриптором `<html>` и заканчиваться закрывающим дескриптором `</html>`. Далее следуют два дескриптора заголовка `<head>` и `</head>`. Между этими дескрипторами обычно находится пара дескрипторов `<title>` и `</title>`, определяющих название страницы. Обратите внимание: именно это название отображается в заголовке окна Web-браузера. Далее следует пара дескрипторов содержимого документа `<body>` и `</body>`. Между этими дескрипторами и размещают всю информацию, которая планируется быть представленной на Web-странице.

Добавление комментариев. Комментарии — это определенное текстовое сообщение, которое автор вставляет в Web-страницу, чтобы отметить для себя нужные детали. Например, как частный случай — сделать для себя пометку о необходимости добавления рисунков в данный раздел либо глобально — добавить информацию, помогающую впоследствии разобраться в отдельных вехах создания сайта или управлять им. Комментарии не отображаются Web-браузерами на экране, но нужно

помнить, что любой пользователь может прочесть данные комментарии. Для добавления комментариев используют специальные парные дескрипторы — открывающий `<!--` и закрывающий `-->`. Чтобы внести в HTML-код Web-страницы текстовую конструкцию комментариев, выполните такие действия.

1. Введите дескриптор `<!--` вместе начала текста комментария. Наберите собственно требуемый текст комментария. Завершите блок комментария закрывающим дескриптором `-->`, например `<!-- А в это место Web-страницы нужно вставить красивые картинки или фотографии. Не забыть!-->`.

2. Как обычно, сохраните текст, вернитесь в окно Web-браузера и обновите изображение открытой Web-страницы. Обратите внимание, что на Web-странице текст комментариев не отображается.

Текст комментариев можно отделять дополнительными пробелами от разделительных дескрипторов, чтобы HTML-код было проще читать.

1.2 СУБД My SQL и утилита phpMyAdmin

Для работы с базой данных необходима СУБД (система управления базами данных), т.е. программа, которая берет на себя все заботы, связанные с доступом к данным. Она содержит команды, позволяющие производить различные действия с базой данных.

MySQL – это реляционная система управления базами данных. То есть данные в ее базах хранятся в виде логически связанных между собой таблиц, доступ к которым осуществляется с помощью языка запросов SQL. MySQL – свободно распространяемая система, т.е. платить за ее применение не нужно. Кроме того, это достаточно быстрая, надежная и, главное, простая в использовании СУБД, вполне подходящая для не слишком глобальных проектов.

Строки таблиц могут быть связаны друг с другом одним из трех способов. Простейшее отношение — "один к одному". В этом случае

строка первой таблицы соответствует одной единственной строке второй таблицы.

Отношение "один ко многим" означает ситуацию, когда строка одной таблицы соответствует нескольким строкам другой таблицы. Это наиболее распространенный тип отношений.

Наконец, при отношении "многие ко многим" строки первой таблицы могут быть связаны с произвольным числом строк во второй таблице.

В текстовом режиме работа с базой данных выглядит просто как ввод команд в командную строку. Но работать с MySQL можно не только в текстовом режиме, но и в графическом. Существует очень популярный визуальный интерфейс для работы с этой СУБД - PhpMyAdmin. Этот интерфейс позволяет значительно упростить работу с базами данных в MySQL. Утилита phpMyAdmin реализована на языке PHP и доступна для свободной загрузки с сайта www.phpmyadmin.net. Этот инструмент обладает дружественным интерфейсом и позволяет пользоваться всеми достоинствами браузера. Многие из базовых SQL-функций работы с данными в PhpMyAdmin сведены к интуитивно понятным интерфейсам и действиям, напоминающим переход по ссылкам в Internet.

1.3 HTML

HTML (HyperText Markup Language, язык гипертекстовой разметки) - специальные инструкции браузеру, с помощью которых создаются web-страницы.

Т.е. web-страницы - это документы в формате HTML, содержащие текст и специальные тэги (дескрипторы) HTML. По большому счету тэги HTML необходимы для форматирования текста (т.е. придания ему

нужного вида), который "понимает" браузер. Документы HTML хранятся в виде файлов с расширением .htm или .html.

Тэги HTML сообщают браузеру информацию о структуре и особенностях форматирования web-страницы. Каждый тэг содержит определенную инструкцию и заключается в угловые скобки <>. Большинство тэгов состоят из открывающей и закрывающей частей и воздействуют на текст, заключенный внутри.

1.4 JavaScript

JavaScript - это объектно-ориентированный язык программирования, предназначенный для написания сценариев, работающих как на стороне клиента, так и на стороне сервера. Поэтому он не является "полноценным" языком программирования, а ориентирован на использование возможностей той среды, в которой сценарии исполняются.

Web-обозреватель, работающий на компьютере-клиенте, обеспечивает среду, в которой JavaScript имеет доступ к объектам, которые представляют собой окна, меню, диалоги, текстовые области, фреймы, куки и ввод-вывод в web-страницу. Кроме того, обозреватель позволяет присоединить сценарии на языке JavaScript к таким событиям, как загрузка и выгрузка страниц и графических образов, нажатие клавиш и движение мыши, выбор текста и пересылка форм. При этом программный код сценариев только реагирует на события и поэтому не нуждается в главной программе. Набор объектов, предоставляемых обозревателем, известен под названием Document Object Model (DOM).

JavaScript — это язык программирования, основанный на объектах: и языковые средства, и возможности среды представляются объектами, а сценарий (программа) на JavaScript — это набор взаимодействующих объектов. JavaScript содержит несколько встроенных объектов, таких,

как Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. Кроме того, JavaScript содержит набор встроенных операций, которые, строго говоря, не обязательно являются функциями или методами, а также набор встроенных операторов (операторы ветвления, операторы цикла и т. д.), управляющих логикой выполнения программ.

JavaScript-код включается в HTML-код страницы и исполняется интерпретатором, встроенным в браузер.

1.5 PHP

PHP— язык программирования, созданный для генерирования HTML-страниц на веб-сервере и работы с базами данных.

Синтаксис PHP подобен синтаксису языка Си. Некоторые элементы, такие как ассоциативные массивы и цикл `foreach`, заимствованы из Perl.

Для работы программы не требуется описывать какие-либо переменные, используемые модули, и т.п. Любая программа может начинаться непосредственно с оператора PHP.

«PHP может все», – заявляют его создатели. В первую очередь PHP используется для создания скриптов, работающих на стороне сервера, для этого его, собственно, и придумали.

Всего выделяют три основные области применения PHP.

- Первая область, как уже говорилось, – это создание приложений (скриптов), которые исполняются на стороне сервера. PHP наиболее широко используется именно для создания такого рода скриптов. Для того чтобы работать таким образом, понадобится PHP-парсер (т.е. обработчик php-скриптов) и web-сервер для обработки скрипта, браузер для просмотра результатов работы скрипта, ну, и, конечно, какой-либо текстовый редактор для написания самого php-кода.

- Вторая область – это создание скриптов, выполняющихся в командной строке. То есть с помощью PHP можно создавать такие скрипты, которые будут исполняться, вне зависимости от web-сервера и браузера, на конкретной машине. Для такой работы потребуется лишь парсер PHP (в этом случае его называют интерпретатором командной строки). Этот способ работы подходит, например, для скриптов, которые должны выполняться регулярно с помощью различных планировщиков задач или для решения задач простой обработки текста.

- И последняя область – это создание GUI-приложений (графических интерфейсов), выполняющихся на стороне клиента. В принципе это не самый лучший способ использовать PHP, особенно для начинающих, но если вы уже досконально изучили PHP, то такие возможности языка могут оказаться полезны. Для применения PHP в этой области потребуется специальный инструмент – PHP-GTK, который является расширением PHP.

Если говорить о возможностях сегодняшнего PHP, то они выходят далеко за рамки тех, что были реализованы в его первых версиях. С помощью PHP можно создавать изображения, PDF-файлы, флэш - ролики, в него включена поддержка большого числа современных баз данных, встроены функции для работы с текстовыми данными любых форматов, включая XML, и функции для работы с файловой системой. PHP поддерживает взаимодействие с различными сервисами посредством соответствующих протоколов.

1.6 AJAX

AJAX - подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных,

веб-страница не перезагружается полностью и веб-приложения становятся более быстрыми и удобными.

AJAX — не самостоятельная технология, а концепция использования нескольких смежных технологий. AJAX базируется на двух основных принципах:

- использование технологии динамического обращения к серверу «на лету», без перезагрузки всей страницы полностью, например:
 - с использованием XMLHttpRequest (основной объект);
 - через динамическое создание дочерних фреймов;
 - через динамическое создание тега <script>.
- использование DHTML для динамического изменения содержания страницы/

Преимущества

- *Экономия трафика*

Использование AJAX позволяет значительно сократить трафик при работе с веб-приложением благодаря тому, что часто вместо загрузки всей страницы достаточно загрузить только изменившуюся часть, как правило довольно небольшую.

- *Уменьшение нагрузки на сервер*

AJAX позволяет несколько снизить нагрузку на сервер. К примеру, на странице работы с почтой, когда вы отмечаете прочитанные письма, серверу достаточно внести изменения в базу данных и отправить клиентскому скрипту сообщение об успешном выполнении операции без необходимости повторно создавать страницу и передавать её клиенту.

- *Ускорение реакции интерфейса*

Поскольку нужно загрузить только изменившуюся часть, пользователь видит результат своих действий быстрее.

1.7 Бизнес-логика клиент-серверного приложения

Бизнес-логика — в разработке информационных систем — совокупность правил, принципов, зависимостей поведения объектов предметной области (области человеческой деятельности, которую система поддерживает). Иначе можно сказать, что бизнес-логика — это реализация правил и ограничений автоматизируемых операций. Является синонимом термина «логика предметной области» (англ. domain logic).

Проще говоря, бизнес-логика — это реализация предметной области в информационной системе. К ней относятся, например, формулы расчёта ежемесячных выплат по ссудам (в финансовой индустрии), автоматизированная отправка сообщений электронной почты руководителю проекта по окончании выполнения частей задания всеми подчиненными (в системах управления проектами), отказ от отеля при отмене рейса авиакомпанией (в туристическом бизнесе) и т. д.

В фазе бизнес-моделирования и разработки требований бизнес-логика может описываться в виде:

текста;

концептуальных аналитических моделей предметной области (онтологии);

бизнес-правил;

разнообразных алгоритмов;

диаграмм деятельности;

графов и диаграмм перехода состояний;

моделей бизнес-процессов.

В фазе анализа и проектирования системы бизнес-логика воплощается в различных диаграммах языка UML или ему подобных. В фазе программирования бизнес-логика воплощается в коде классов и их методов, в случае использования объектно-ориентированных языков программирования, или процедур и функций, в случае применения процедурных языков.

На жаргоне разработчиков программного обеспечения «бизнес-логикой» также называются программные модули, её реализующие, и уровень системы, на котором эти модули находятся (англ. business logic layer, domain logic layer).

В многоуровневых (многослойных) информационных системах этот уровень взаимодействует с нижележащим уровнем инфраструктурных сервисов (англ. infrastructure layer), например, интерфейсом доступа к базе данных или файловой системе (англ. data-access layer, DAL) и вышележащим уровнем сервисов приложения (англ. application services layer), который уже, в свою очередь, взаимодействует с уровнем пользовательского интерфейса (англ. user interface layer) или внешними системами.

В клиент-серверном приложении имеются два звена, что приводит к созданию как минимум двух слоев. На начальном этапе сервер рассматривался только как удаленная база данных, и деление было как на рисунке – приложение на клиенте и данные на сервере. Обычно вся бизнес логика находилась на клиенте, перемешанная с остальными слоями, такими как пользовательский интерфейс.

Достаточно быстро стало понятно, что можно сократить нагрузку на сеть и централизовать логику для уменьшения постоянных затрат на развертывание, перенеся большую часть бизнес логики на сервер. Архитектурно сервер был хорошо подготовленным местом в клиент-

серверной системе, но база данных как платформа давала мало возможностей. Базы данных были спроектированы для хранения и выдачи и в их архитектуру не были заложены возможности расширения в направлении бизнес логики. Языки хранимых процедур в базах данных были разработаны для базовых преобразований данных, чтобы поддержать то, на что не хватало SQL. Языки хранимых процедур разработали для быстрого исполнения, а не для обслуживания сложных задач бизнес логики.

Это решение выигрывало, и часть бизнес логики переехала в хранимые процедуры. На самом деле, я готов поспорить, что бизнес логика была ужата и вбита в рамки хранимых процедур, исключительно с прагматической точки зрения. В двухзвенной архитектуре это было не идеальным, но все-таки гораздо лучшим.

Когда проблема клиент-серверной архитектуры стала явной, возросла популярность трехзвенного подхода. Наибольшей и самой тяжелой проблемой того времени было количество подключений. Сейчас многие базы данных могут обрабатывать тысячи единовременных подключений, в девяностых большинство баз данных падали где-то на 500 подключений. Сервера зачастую лицензировались по количеству клиентских подключений. Это все и привело к тому, что потребовалось сократить количество подключений к базе данных.

Стало популярным объединение подключений в пул, однако для реализации пула подключений в системе с множеством отдельных клиентов, необходимо внедрить третье звено между клиентом и сервером. Среднее звено так и стало называться «среднее звено». В большинстве случаев среднее звено существовало только для управления пулом соединений, но в некоторых случаях бизнес логика начала перемещаться в среднее звено потому, что языки разработки (C++, VB, Delphi, Java) гораздо лучше подходили для реализации бизнес логики, чем языки хранимых процедур. Вскоре стало очевидно, что среднее

звено –это наилучшее место для бизнес-логики.

1.8 ООО «ЭНКОМ»

ООО "ЭНКОМ" – студия веб-дизайна, которая осуществляет свою деятельность с 2006 года и специализируется в области веб-дизайна, программирования, хостинга и раскрутки сайтов в Белгороде и других городах России. Основное направление деятельности компании - создание сайтов "под ключ", от проектирования структуры и дизайна сайта до их размещения и продвижения в сети Интернет.

Услуги ООО «ЭНКОМ»:

- Создание сайтов и web-дизайн
- Регистрация доменов в зонах RU и РФ
- Хостинг
- Раскрутка сайтов (продвижение)
- Реклама в Интернет
- Редизайн существующих сайтов
- Разработка логотипа и фирменного стиля
- Визитки, фирменные бланки, папки

2. Проектирование приложения

Проектирование web-приложения подразумевает решение следующих задач:

- Создание базы данных;
- Создание приложения для управления БД;
- Начало реализации спроектированного web-приложения.

2.1 Создание базы данных

В начале работы была разработана логическая модель базы данных, представляющая собой данные в контексте предметной области и приведенная на рисунке 2.1.

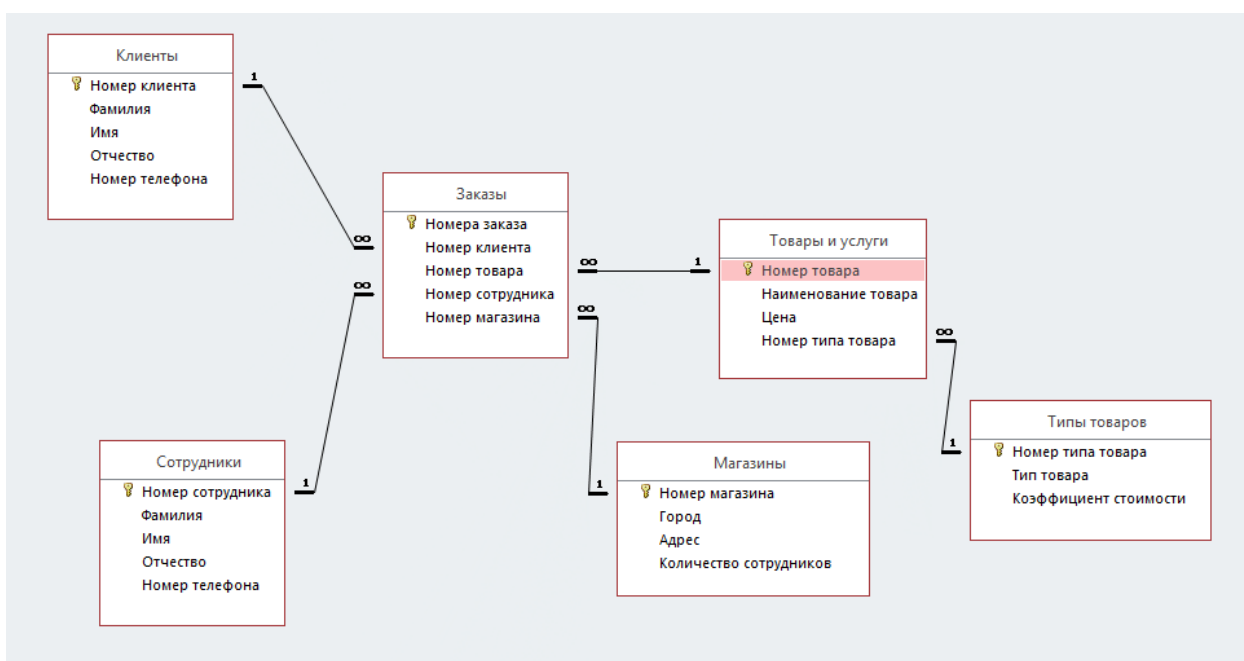


Рис. 2.1. Логическая модель базы данных

Далее на основе логической модели была разработана физическая модель, содержащая конкретные названия полей базы данных. Физическая модель базы данных представлена на рисунке 2.2.

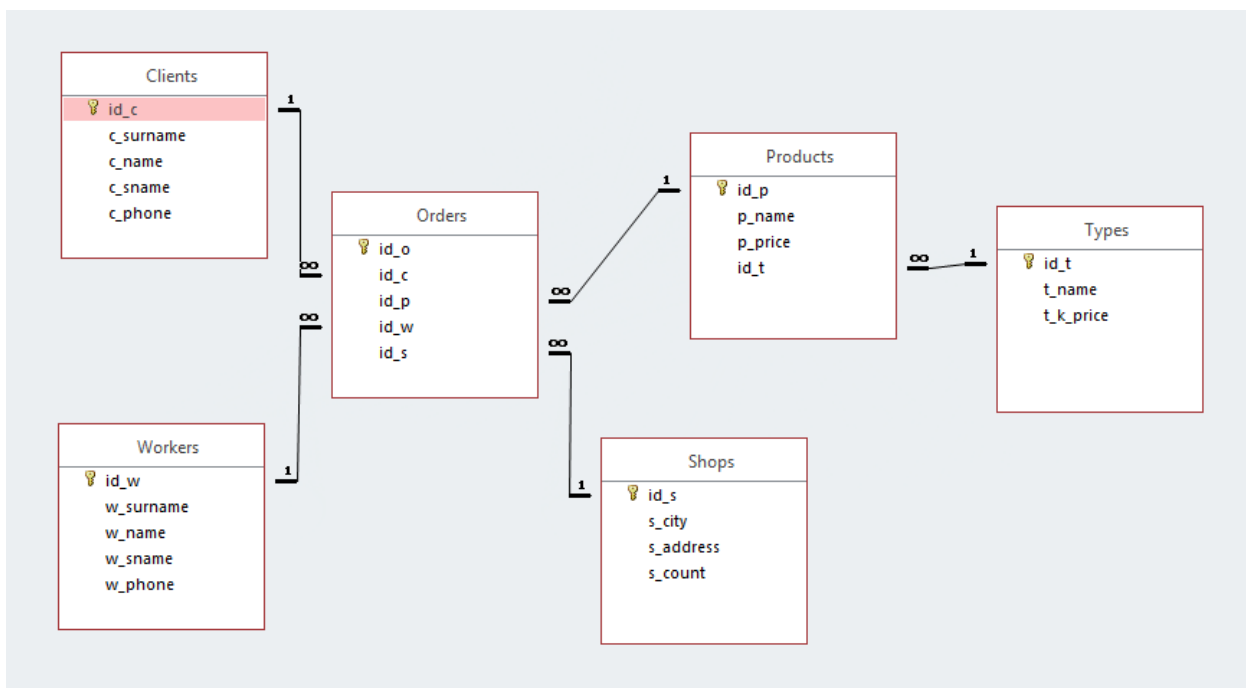


Рис.2.2. Физическая модель базы данных

В таблице ниже приведена подробная структура таблиц с указанием типов данных и атрибутов полей. Очень важно проработать структуру базы данных заранее, так как верная модель и правильно подобранные типы данных позволяют оптимизировать работу базы. Стоит отметить, что поле с телефонным номером имеет тип char, а не int. Это позволяет вводить в это поле коды стран, скобки, а также работает быстрее в целом, так как текстовое поле не является вычисляемым.

Имя таблицы	Поле	Тип	Not Null	Primary key	Extra
Orders	id_o	int(10)	Да	Да	auto_increment
	id_c	int(10)	Да	Нет	
	id_p	int(10)	Да	Нет	
	id_w	int(10)	Да	Нет	
	id_s	int(10)	Да	Нет	
Clients	id_c	int(10)	Да	Да	auto_increment
	c_surname	char(30)	Да	Нет	
	c_name	char(25)	Да	Нет	
	c_sname	char(30)	Нет	Нет	
	c_phone	char(15)	Да	Нет	
Types	id_t	int(10)	Да	Да	auto_increment
	t_name	char(20)	Да	Нет	
	t_k_price	int	Да	Нет	

Products	id_p	int(10)	Да	Да	auto_increment
	p_name	char(60)	Да	Нет	
	p_price	int	Да	Нет	
	id_t	int	Да	Нет	
Shops	id_s	int(10)	Да	Да	auto_increment
	s_city	char(20)	Да	Нет	
	s_address	char(50)	Да	Нет	
	s_count	int	Да	Нет	
Workers	id_w	int(10)	Да	Да	auto_increment
	w_surname	char(30)	Да	Нет	
	w_name	char(25)	Да	Нет	
	w_sname	char(30)	Нет	Нет	
	w_phone	char(15)	Да	Нет	

Таблица 2.1. Типы данных полей БД

2.2 Модульная схема приложения

Далее была разработана модульная схема приложения. Она представляет из себя схему взаимодействия файлов клиент-серверного приложения, что позволяет в дальнейшем облегчить процесс написания кода приложения.

В модульной схеме отображены связи между отдельными файлами приложения. Так, index.html является главной страницей сайта, переходы на остальные страницы осуществляются уже непосредственно оттуда. Модульная схема приложения приведена ниже.

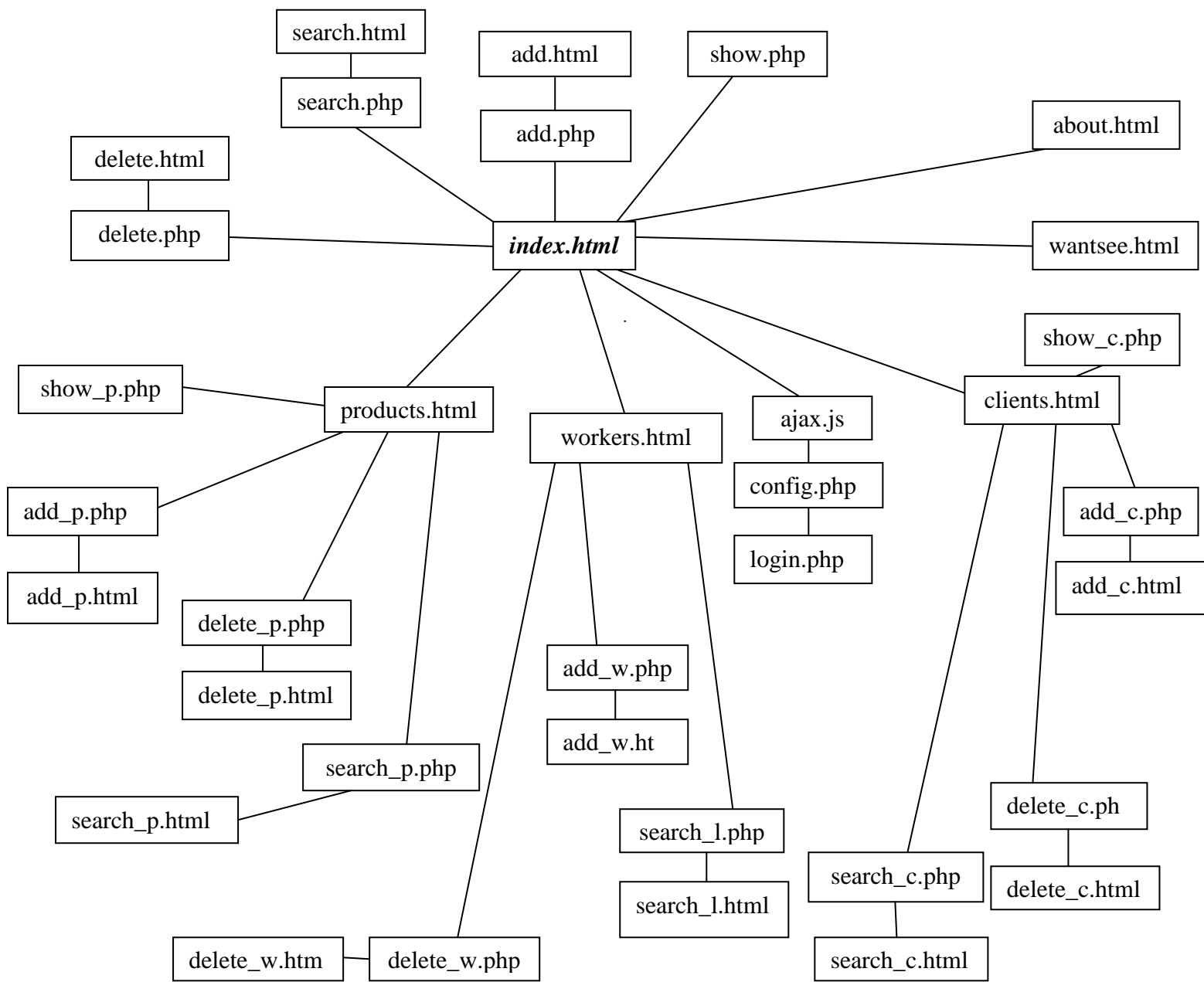


Рис. 2.3. Модульная схема приложения

3. Реализация клиент-серверного приложения

На рисунке 3.1 приведена главная страница приложения.

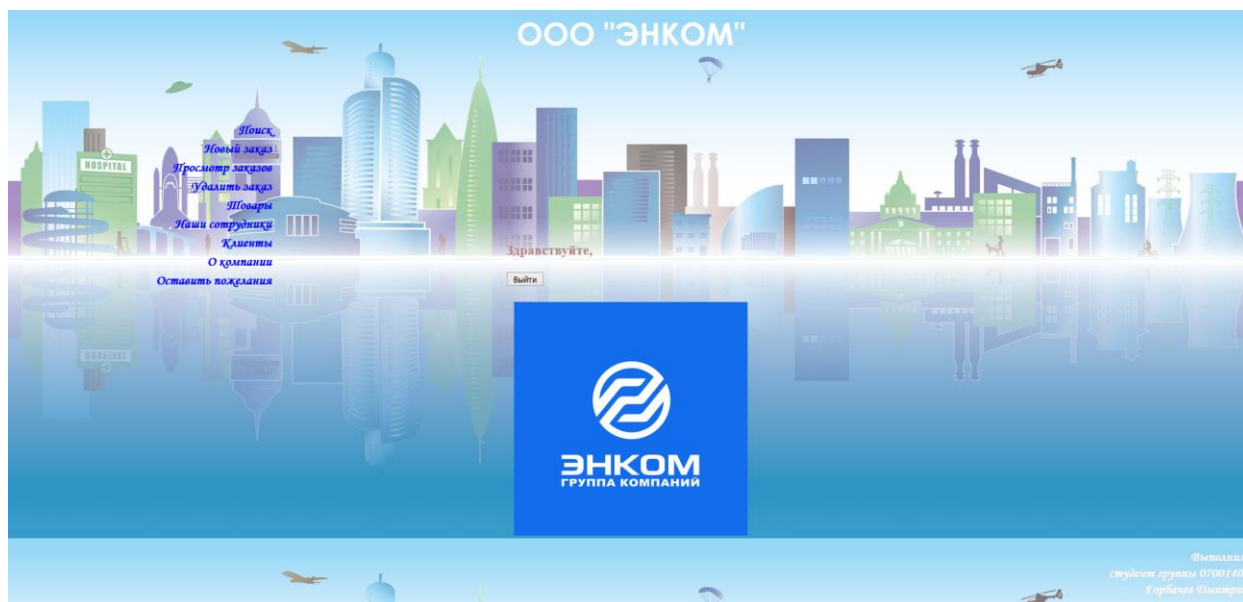


Рис 3.1. Главная страница web-приложения.

Как видно из рисунка, слева находится боковое меню сайта. Фоновое изображение подобрано под цвет логотипа. Ссылки в боковом меню меняют цвет при нажатии и наведении

На рисунке 3.2 приведена форма добавления сотрудника.

Рис 3.2. Страница добавления сотрудника.

Далее в листинге 3.1 приведен программный код файла add_w.html

Листинг 3.1. Программный код add_w.html

```
<html>
<head>
<title>000 "ЭНКОМ"</title>
  <link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
<meta http-equiv='Content-Type' content='text/html; charset='utf-8'>
</head>
<body>
<p>Введите информацию о сотруднике:</p><br>
<form action="add_l.php" method="post">
<p>Фамилия:</p> <input type="text" name="l_surname" size="30"> <br>
<p>Имя:</p> <input type="text" name="l_fname" size="30"> <br>
<p>Отчество:</p> <input type="text" name="l_sname" size="30"> <br>
<input type="submit" value="Отправить данные"><br>
</form>
<p class="go-home"><a onmouseup="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'" id="m1" style="visibility: visible"
onmouseover="this.style.color='#fff5ee'" href="index.html">На главную</a></p>
</body>
</html>
```

Конец листинга 3.1.

Как видно из листинга, данные из формы ввода обрабатываются в файле add_w.php. Ниже приведен листинг данного файла.

Листинг 3.2. Программный код add_w.php

```
<html><head>
<title>Запись:</title> <link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
</head>
<body>
<h1>Добавление информации в БД:</h1>
<?php
require "add_l.html";
$_w_surname=trim($_POST['w_surname']);
$_w_fname=trim($_POST['w_name']);
$_w_sname=trim($_POST['w_sname']);
if (!$w_surname || !$w_name || !$w_sname)
{
echo "<script type=\"text/JavaScript\">alert(\"Заполнены не все поля!\")</script>";
exit;
}
@ $db = mysql_pconnect('localhost', '0700', '1488');
if (!$db)
{
echo "<script type=\"text/JavaScript\">alert(\"Ошибка соединения с БД!\")</script>";
exit;
}
mysql_select_db('db_1409131');
$query="INSERT INTO Workers VALUES (NULL, '$_w_surname.', '$_w_name.',
'$_w_sname.')";
$result = mysql_query($query);
if ($result) echo "<script type=\"text/JavaScript\">alert(\"Ваши данные добавлены в
список сотрудников\")</script>";
?></body></html>
```

Конец листинга 3.2.

В листинге создаются переменные `w_surname`, `w_fname` и `w_name`, куда попадают данные из формы, которые передаются POST-методом. Далее формируется запрос к базе данных на добавление указанных значений.

На рисунке 3.3 приведен результат успешного добавления сотрудника в таблицу.

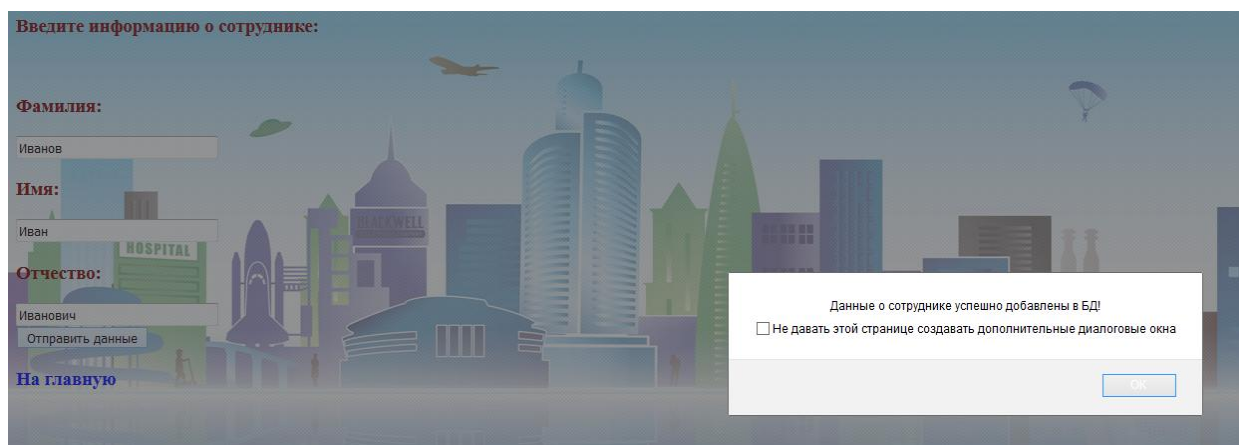


Рис 3.3. Успешное добавление сотрудника.

Ниже приведен снимок экрана с поисковым запросом.



Рис 3.4. Поиск товара.

Рассмотрим программные коды формы поиска и ее обработчика на PHP, приведенные в листинге 3.3.

Листинг 3.3. Программный код search_p.html

```
<html>
<head>
  <link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
<meta http-equiv='Content-Type' content=text/html; charset=utf-8>
<title>ООО "ЭНКОМ"</title>
</head>
<body>
<form action="search_g.php" method="post">
<p>Выберите тип поиска:</p><br><br>
<select name="search_type">
<option value="id_p">По номеру</option>
<option value="p_name">По названию</option>
</select>
<br><br>
<p>Введите данные:</p><br><br>
<input name="to_search" type="text">
<br /><br>
<input type="submit" value="Поиск">
</form>
<p class="go-home"><a
onmouseup="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseover="this.style.color='blue'" id="m1" style="visibility: visible"
onmouseout="this.style.color='#fff5ee'" href="index.html">На главную</a></p>
</body>
</html>
```

Конец листинга 3.3.

Общий принцип работы поиска аналогичен принципу работы добавления данных, однако у поиска есть одна особенность. В листинге 3.3 опциям выбора присвоены значения, названия которых совпадают с названиями полей базы данных.

В результате это используется при формировании запроса, когда нужные поля подставляются в текст запроса.

На рисунке 3.5 приведен результат работы поиска.



Рис 3.5. Результат работы поиска товара.

На рисунке 3.6 приведена страница about.html, содержащая краткие сведения об ООО «ЭНКОМ».

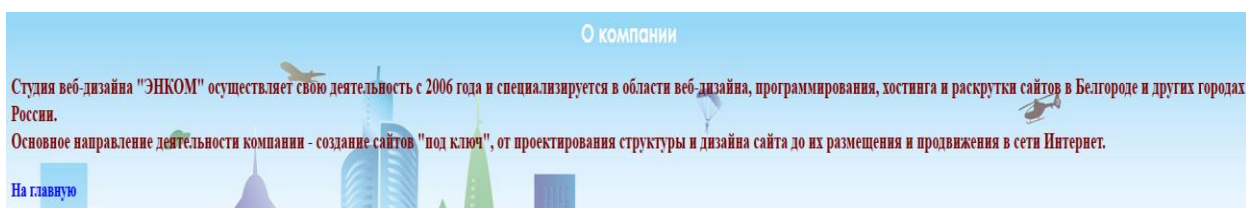


Рис 3.6. Страница «О компании».

Далее, на рисунке 3.7 приведен результат отображения данных в таблице Workers, где хранятся данные о работниках предприятия.



Рис 3.7. Список сотрудников в базе данных.

На рисунке 3.8 приведен аналогичный результат аналогичного запроса для таблицы Shops, в которой хранятся сведения о филиалах предприятия



Рис 3.8. Список магазинов в базе данных.

И наконец, на рисунке 3.9 приведена таблица со списком товаров и услуг.

Список товаров и услуг:

№	Наименование товара	Стоимость	Тип товара
1	Интернет-магазин	15000	Разработка
2	SEO-оптимизация сайта	7000	Продвижение
3	Сопровождение сайта	71	Разработка
4	Консультация по администрированию	150	Техническая поддержка

[На главную](#)

Рис 3.9. Список товаров и услуг в базе данных.

Примечательно, что для этого запроса происходит выборка сразу по двумя таблицам (также из таблицы типов), а в результате на экран выводится выборка сразу из двух таблиц. Это обусловлено тем, что база данных находится в третьей нормальной форме, поэтому в таблице товаров в поле «Тип товара» указан порядковый номер соответствующей записи в таблице типов.

Работоспособность приложения проверена на тестовых примерах, а само приложение представлено в ООО «ЭНКОМ» и получило положительные отзывы от компании.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были исследованы различные программные средства и СУБД. Выявлены достоинства и недостатки программных средств, определены возможности реализации тех или иных функций приложения. На основании проведенного исследования разработаны алгоритмы, предусматривающие программную реализацию данного приложения.

В соответствии с постановкой задачи проекта на основании разработанных алгоритмов определена модульная структура программы. Исходя из специфики решаемых задач, были использованы такие программные средства, как HTML, PHP, JavaScript и AJAX. В конечном итоге было реализовано приложение, взаимодействующее с разработанной базой данных. Реализованное приложение имеет удобный интерфейс и относительно богатый функционал. Кроме того, имеется возможность регистрации. Работоспособность разработанного приложения проверена на тестовых примерах, приведены сведения об интерфейсе, показаны варианты обработки данных и взаимодействие с БД.

Представленное программное обеспечение может быть использовано в качестве web-приложения для ООО «ЭНКОМ».

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Java Script. 3 – е изд. – Дунаев В.В. – СПб.: Питер, 2008. –400 с.: ил.
2. «Основы веб-мастеринга» К.Жаринов - СПб.: БХВ-Петербург, - 2003г.К. Дж. Дейт. Введение в системы баз данных / Пер. с англ. — 8-е изд. — М.: Вильямс, 2005, 1328 стр.
3. Лаура Томсон и Люк Веллинг. Разработка Web-приложений на PHP и MySQL: Пер. с англ./Лаура Томсон, Люк Веллинг. — 2-е изд., испр. — СПб: ООО «ДиаСофтЮП», 2003. — 672 с..
4. Кристиан Уэнц JavaScript. Карманный справочник = JavaScript Phrasebook. — М.: «Вильямс», 2007, 272 стр.
5. Дмитрий Котеров, Алексей Костарев. PHP. — СПб.: «БХВ-Петербург», 2005. — С. 1120. — (В подлиннике). — ISBN 5-94157-245-X.
6. Костарев А. Ф. PHP 5. — СПб.: «БХВ-Петербург», 2008. — С. 1104. — ISBN 978-5-9775-0315-0.
7. Мэтт Зандстра. PHP: объекты, шаблоны и методики программирования = PHP Objects, Patterns and Practice, Third Edition. — 3-е издание. — М.: «Вильямс», 2010. — С. 560. — ISBN 978-5-8459-1689-1.
8. Кристиан Дари, Эмилиан Баланеску. PHP и MySQL: создание интернет-магазина = Beginning PHP and MySQL E-Commerce: From Novice to Professional. — М.: «Вильямс», 2010. — ISBN 978-5-8459-1602-0.
9. Джейсон Ленгсторф. PHP и jQuery для профессионалов = Pro PHP and jQuery. — М.: «Вильямс», 2010. — С. 352. — ISBN 978-5-8459-1693-8.
10. Стив Суэринг, Тим Конверс, Джойс Парк. PHP и MySQL = PHP 6 and MySQL 6 Bible. — 2-е издание. — М.: «Диалектика», 2010. — С. 912. — (Библия программиста). — ISBN 978-5-8459-1640-2.
11. Квентин Зервас. Web 2.0: создание приложений на PHP = Practical Web 2.0 Applications with PHP. — М.: «Вильямс», 2009. — С. 544. — ISBN 978-5-8459-1590-0.

12. Кузнецов Максим, Симдянов Игорь. PHP 5/6. — СПб.: «БХВ-Петербург», 2009. — С. 1024. — ISBN 978-5-9775-0304-4.
13. Кузнецов Максим, Симдянов Игорь. Объектно-ориентированное программирование на PHP. — СПб.: «БХВ-Петербург», 2007. — С. 608. — ISBN 978-5-9775-0142-2.
14. Эд Леки-Томпсон, Алек Коув, Стивен Новицки, Хьяо Айде-Гудман. PHP 5 для профессионалов = Professional PHP 5. — М.: «Диалектика», 2006. — С. 608. — ISBN 0-7645-7282-2.
15. Кузнецов Максим, Симдянов Игорь. Самоучитель PHP 5/6. — 3-е изд., перераб. и доп. — СПб.: «БХВ-Петербург», 2009. — С. 672. — ISBN 978-5-9775-0409-6.
16. Флэнаган Д. JavaScript. Карманный справочник. Сделайте веб-страницы интерактивными! / Перевод А.Г. Сысолюк. — Москва.: Издательский дом "Вильямс", 2015. — С. 320. — 1000 экз. — ISBN 978-5-8459-1948-9 (рус.).
17. Д. Рейсиг. 4. Инструменты отладки и тестирования // JavaScript. Профессиональные приёмы программирования = Pro JavaScript™ Techniques / Перевод Н. Вильчинский. — СПб.: Питер, 2008. — С. 76. — (Библиотека программиста). — 2500 экз. — ISBN 978-5-91180-904-1.
18. Christian Johansen. Chapter 1. Automated Testing // Test Driven JavaScript Development. — Addison-Wesley Professional, 2010. — P. 3. — (Rough Cuts). — ISBN 0-321-68391-9, 978-0-321-68391-5.
19. Christian Johansen. 3.1.4. Features of xUnit Test Frameworks. The Test Runner // Test Driven JavaScript Development. — Addison-Wesley Professional, 2010. — P. 24, 25. — (Rough Cuts). — ISBN 0-321-68391-9, 978-0-321-68391-5.
20. Christian Johansen. 3.3.1. Crosscheck // Test Driven JavaScript Development. — Addison-Wesley Professional, 2010. — P. 28. — (Rough Cuts). — ISBN 0-321-68391-9, 978-0-321-68391-5.

21. Christian Johansen. 3.3.2. env.js // Test Driven JavaScript Development. — Addison-Wesley Professional, 2010. — P. 28, 29. — (Rough Cuts). — ISBN 0-321-68391-9, 978-0-321-68391-5.
22. Christian Johansen. 3.3.3. Blue Ridge // Test Driven JavaScript Development. — Addison-Wesley Professional, 2010. — P. 29. — (Rough Cuts). — ISBN 0-321-68391-9, 978-0-321-68391-5.
23. Christian Johansen. 3.3.4. The Issue With Headless Test Runners // Test Driven JavaScript Development. — Addison-Wesley Professional, 2010. — P. 29. — (Rough Cuts). — ISBN 0-321-68391-9, 978-0-321-68391-5.
24. C. Titus Brown, Gheorghe Gheorghiu, Jason Huggins. 3.1. A Brief History of Selenium // An Introduction to Testing Web Applications with twill and Selenium. — O'Reilly Media, Inc, 2007. — 60 p.
25. Christian Johansen. 3.4. One Test Runner To Rule Them All // Test Driven JavaScript Development. — Addison-Wesley Professional, 2010. — P. 29. — (Rough Cuts). — ISBN 0-321-68391-9, 978-0-321-68391-5.
26. Christian Johansen. 3.4.1. How JsTestDriver Works // Test Driven JavaScript Development. — Addison-Wesley Professional, 2010. — P. 29. — (Rough Cuts). — ISBN 0-321-68391-9, 978-0-321-68391-5.
27. Резиг, Джон TestSwarm (англ.) (недоступная ссылка — история) (25 August 2009).
28. Christian Johansen. 3.1.6. Dependencies // Test Driven JavaScript Development. — Addison-Wesley Professional, 2010. — P. 25. — (Rough Cuts). — ISBN 0-321-68391-9, 978-0-321-68391-5.
29. Christian Johansen. 3.2.1. YUI Test // Test Driven JavaScript Development. — Addison-Wesley Professional, 2010. — P. 26. — (Rough Cuts). — ISBN 0-321-68391-9, 978-0-321-68391-5.
30. Christian Johansen. 3.2.2. Other In-Browser Testing Frameworks // Test Driven JavaScript Development. — Addison-Wesley Professional, 2010. — P. 28. — (Rough Cuts). — ISBN 0-321-68391-9, 978-0-321-68391-5.

31. Крокфорд, Дуглас Малыш-программист на JavaScript (англ.). — Статья, раскрывающая связь между JavaScript и Scheme.
32. Толксдорф Р. Языки программирования для JavaScript (англ.). is-research. — Подборка реализаций языков программирования на JavaScript.. Проверено 12 октября 2009.
33. hotruby. Ruby on JavaScript and Flash (англ.).
34. Cooper P. HotRuby: A JavaScript & Flash Virtual Machine That Runs Ruby (англ.). Rubi Inside. Проверено 12 октября 2009.
35. Б. Маклафлин. Изучаем Ajax = Head Rush Ajax. — СПб.: Питер, 2007. — ISBN 978-5-91180-322-3.
36. Стивен Хольцнер. Ajax Библия программиста = Ajax Bible. — М.: Диалектика, 2009. — С. 553. — ISBN 978-5-8459-1502-3.
37. Дейв Крейн, Бер Бибо, Джордон Сонневельд. Ajax на практике = Ajax in Practice. — М.: Вильямс, 2007. — ISBN 978-5-8459-1327-2.

ПРИЛОЖЕНИЕ

index.html

```
<html>
<head>
  <link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
  <script src="ajax.js" language="javascript"></script>
  <title>
    Сайт учета заказа потребителей ООО "ЭНКОМ"
  </title>
</head>
<body onLoad="javascript:Check()">
  <p class="main"> ООО "ЭНКОМ"</p><br><br>
<table width="100%"><tr><td>
  <table class="sm_table" id="table">
    <tr>
      <td>
        <p class="sometext">
          <a onMouseup="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'" id="m1" style="visibility: visible"
onmouseover="this.style.color='white'" href="search.php">Поиск</a></td>
      </tr>
      <tr>
        <td>
          <p class="sometext">
            <a onMouseup="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'" id="m1" style="visibility: visible"
onmouseover="this.style.color='white'" href="add.php">Новый заказ</a></td>
          </tr>
          <tr>
            <td>
              <p class="sometext">
                <a onMouseup="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'" onmouseover="this.style.color='white'"
id="m3" style="visibility: visible" href="show.php">Просмотр заказов</a></td>
            </tr>
          <tr>
            <td>
```

```

        <p class="sometext">
        <a onclick="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'"          onmouseover="this.style.color='white'"
id="m4" style="visibility: visible" href="delete.php">Удалить заказ</a></td>
    </tr>
<tr>
    <td>
        <p class="sometext">
        <a onclick="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'"          onmouseover="this.style.color='white'"
id="m4" style="visibility: visible" href="genres.html">Товары</a></td>
    </tr>
<tr>
    <td>
        <p class="sometext">
        <a onclick="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'"          onmouseover="this.style.color='white'"
id="m4" style="visibility: visible" href="librarians.html">Наши
сотрудники</a></td>
    </tr>
<tr>
    <td>
        <p class="sometext">
        <a onclick="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'"          onmouseover="this.style.color='white'"
id="m4" style="visibility: visible" href="readers.html">Клиенты</a></td>
    </tr>
<tr><td>
    <p class="sometext">
    <a onclick="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'"          onmouseover="this.style.color='white'"
id="m4" style="visibility: visible" href="about.html">О компании</a></td>
    </tr>
<tr>
    <td>
        <p class="sometext">
        <a onclick="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'"          onmouseover="this.style.color='white'"

```

```

id="m4" style="visibility: visible" href="wantsee.html">Оставить
пожелания</a></td>
</tr>
</table></td>
<td>
<form action="javascript:login()" method="post">
<p class="login" style="margin-left:285px;" name="forlog">Введите логин:
<input name="loginLogin" type="text" id="login" name="forlog" value=""/></p>
<p class="login" style="margin-left:285px;" name="forlog">Введите
пароль:<input name="pswLogin" type="password" id="pswLogin" name="forlog"
value=""/></p>
<p class="login" style="margin-left:285px;" name="forlog"><input type="submit"
name="forlog" value="Войти"/>
</form>
<input type="button" value="Регистрация" name="forlog"
onClick="document.location.href='registration.php';"></p>
<p class="loginresponse" style="margin-left:285px;" id="login_response"></p>
<form action="javascript:quit()" id="qqq" method="post" style="visibility:
hidden"><p class="login" style="margin-left:285px;"><input type="submit"
value="Выйти"></p></form>
</tr></table>
<div class="imgCenter" id="image">

<p class="sometext" align="right" id="done">
Выполнил:<br>студент группы 07001401 <br>Горбачев Дмитрий</p>
</div>
</body>
</html>

```

add.html

```

<html>
<head>
<title> ООО «ЭНКОМ»</title>
<link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
<meta http-equiv='Content-Type' content='text/html; charset='utf-8'>
</head>
<body>
<p>Введите информацию о товаре:</p><br>
<form action="add.php" method="post">
<p>Название товара:</p><input type="text" name="id_b" size="60"><br><br>
<p>Ваше Ф.И.О.:</p> <input type="text" name="author" size="60"> <br>
<p>Продавец:</p> <input type="text" name="id_g" size="2"> <br>
<input type="submit" value="Отправить данные"><br>
</form>

```

```

class="go-home"><a
onmouseup="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'" id="m1" style="visibility: visible"
onmouseover="this.style.color='#fff5ee'" href="index.html">На главную</a></p>
</body>
</html>

```

add.php

```

<html>
<head>
<title>Запись:</title>
<link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
</head>
<body>
<h1>Добавление информации в БД:</h1>
<?php
require "add.html";
$хid_b=trim($_POST['id_b']);
$хauthor=trim($_POST['author']);
$хid_g=trim($_POST['id_g']);
$хyear=trim($_POST['year']);
if (!$хid_b || !$хauthor || !$хid_g || !$хyear)
{
echo "<script type=\"text/JavaScript\">alert(\"Заполнены не все
поля!\")</script>";
exit;
}
@ $хdb = mysql_pconnect('localhost', '1409131', 'm2mnearl');
if (!$хdb)
{
echo "<script type=\"text/JavaScript\">alert(\"Ошибка соединения с
БД\")</script>";
exit;
}
mysql_select_db('db_0700');
$хquery="INSERT INTO books VALUES (NULL, '$хid_b.', '$хauthor.',
'$хid_g.', '$хyear.')";
$хresult = mysql_query($хquery);
if ($хresult) echo "<script type=\"text/JavaScript\">alert(\"Товар добавлен в
БД\")</script>";
?>
</body>
</html>

```


delete.html

```
<html>
<head>
<title> ООО «ЭНКОМ»</title>
  <link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
<meta http-equiv='Content-Type' content=text/html; charset=utf-8>
</head>
<body>
<form action="delete.php" method="post">
<p>Введите номер товара:</p><br>
<input name="id_b" type="text"><br>
<input type="submit" value="Удалить">
</form>
<p
                                                                 class="go-home"><a
onmouseup="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'"   id="m1"   style="visibility:   visible"
onmouseover="this.style.color='#fff5ee'" href="index.html">На главную</a></p>
</body>
</html>
```

delete.php

```
<html>
<head>
<title>Удаление:</title>
  <link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
</head>
<body>
<h1>Удаление информации из БД</h1>
<?php
require "delete.html";
$del_id=trim($_POST['id_b']);
if (!$del_id || $del_id=="*")
{
  echo  "<script   type=\"text/JavaScript\">alert(\"Вы   не   ввели   номер
товара!\")</script>";
  exit;
}
@ $db = mysql_pconnect('localhost', '1409131', 'm2mnear1');
if (!$db) {
  echo  "<script   type=\"text/JavaScript\">alert(\"Ошибка   соединения   с
БД\")</script>";
```

```

    exit;
}
mysql_select_db('db_0700');
$result = mysql_query("delete from books where id_b=$del_id");
echo "<script type=\"text/JavaScript\">alert(\"Удаление      произведено
успешно\")</script>";
?>
</body>
</html>

```

search.html

```

<html>
<head>
  <link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
<meta http-equiv='Content-Type' content=text/html; charset=utf-8>
<title> ООО «ЭНКОМ»</title>
</head>
<body>
<form action="search.php" method="post">
<p>Выберите тип поиска:</p><br><br>
<select name="search_type">
<option value="id_b">По номеру</option>
<option value="b_name">По наименованию</option>
<option value="author">По покупателю</option>
<option value="genre">По цене</option>
</select>
<br><br>
<p>Введите данные:</p><br><br>
<input name="to_search" type="text">
<br /><br>
<input type="submit" value="Поиск">
</form>
<p
                                                                 class="go-home"><a
onmouseup="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'"   id="m1"   style="visibility:   visible"
onmouseover="this.style.color='#fff5ee'" href="index.html">На главную</a></p>
</body>
</html>

```

search.php

```

<html>

```

```

<head>
  <link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
</title>Поиск:</title>
</head>
<body>
<h1>Поиск товара</font></h1>
<?php
require "search.html";
$type=trim($_POST['search_type']);
$par=trim($_POST['to_search']);
if (!$type || !$par)
{
echo "<script type=\"text/JavaScript\">alert(\"Вы не ввели параметр
поиска!\")</script>";
exit;
}
$par = addslashes($par);
@ $db = mysql_pconnect('localhost', '1409131', 'm2mnearl');
if (!$db)
{
echo "<script type=\"text/JavaScript\">alert(\"Ошибка соединения с
БД\")</script>";
exit;
}

mysql_select_db('db_0700');
//mysql_query("SET CHARACTER SET utf8.general_ci");
//mysql_query("SET NAMES utf8.general_ci");
$result = mysql_query("select * from books left join genres on
books.id_g=genres.id_g where ".$type." like '%".$par.%'");
if (!mysql_num_rows($result)) echo "<script
type=\"text/JavaScript\">alert(\"Поиск не дал результатов\")</script>";
else {
echo "<p>Найденные записи (" .mysql_num_rows($result)." шт.):</p>";
for ($i=0; $i<mysql_num_rows($result); $i++)
{
$rows = mysql_fetch_array($result);
echo "<br><p>Номер товара: ";
echo stripslashes($rows['id_b']);
echo '<br><b>Наименование: ';
echo stripslashes($rows['b_name']);
echo '<br><b>Цена: ';
echo stripslashes($rows['author']);
echo '<br><b>Тип: ';
echo stripslashes($rows['genre']);
}
}
}

```

```

echo '<br><b>Год: ';
echo stripslashes($rows['year']);
echo '</p><hr>';
}}
?>
</body>
</html>

```

show.php

```

<html>
<head>
<title> ООО «ЭНКОМ»</title>
  <link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
</head>
<body>
<h1>Просмотр данных</h1>
<?php
@$db = mysql_pconnect('localhost', '1409131', 'm2mnear1');
if (!$db)
{
echo    "<script    type=\"text/JavaScript\">alert(\"Ошибка    соединения    с
БД\")</script>";
exit;
}
mysql_select_db('db_0700');
$result = mysql_query("select * from products left join genres on
books.id_g=genres.id_g");
echo "<p>Найденные записи (".mysql_num_rows($result)." шт.):</p>";
for ($i=0; $i<mysql_num_rows($result); $i++)
{
$rows = mysql_fetch_array($result);
echo "<br><p>Номер товара: ";
echo stripslashes($rows['id_b']);
echo '<br><b>Наименование: ';
echo stripslashes($rows['b_name']);
echo '<br><b>Тип: ';
echo stripslashes($rows['author']);
echo '<br><b>Цена: ';
echo stripslashes($rows['genre']);
echo '</p><hr>';
}
?>

```

```

class="go-home"><a
onmouseup="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'" id="m1" style="visibility: visible"
onmouseover="this.style.color='#fff5ee'" href="index.html">На главную</a></p>
</body>
</html>

```

wantsee.html

```

<html>
<head>
<title>О компании</title>
<link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
</head>
<body>
<h1>Ваши пожелания</h1>
<table width="70%" align="center" border="2"><form name="wantsee"
action="mailto:lib_fourier@mail.ru">
<tr><td><p>Фамилия : </td><td><input type="text" size="25"></td></tr>
<tr><td><p>Имя : </td><td><input type="text" size="25"></td></tr>
<tr><td><p>Отчество : </td><td><input type="text" size="25"></td></tr>
<tr><td><p>Пожелания: </td><td><input type="text" size="45"></td></tr>
<tr><td><input type="submit" value="Отправить"></td><td><input
type="reset"> <input type="button" value="На главную"
onClick="document.location.href='index.html';"></td></tr></p>
</form></table>
</body>
</html>

```

registration.html

```

<html>
<head>
<link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
<meta http-equiv='Content-Type' content=text/html; charset=utf-8>
<title> ООО «ЭНКОМ»
</title>
</head>
<body>
<form action="registration.php" method="post">
<p>Введите ваши данные:</p><br><br>
<p>Логин:</p><input type="text" name="log" size="60"><br><br>

```

```

<p>Пароль:</p> <input type="password" name="psw" size="60"> <br>
<p>Повторите пароль:</p> <input type="password" name="rpsw" size="60">
<br>
<br><br>
<input type="submit" value="Регистрация">
</form>
<p
                                class="go-home"><a
onmouseup="this.style.fontStyle='italic';this.style.color='brown'"
onmousedown="this.style.fontStyle='italic';this.style.color='red'"
onmouseout="this.style.color='blue'" id="m1" style="visibility: visible"
onmouseover="this.style.color='#fff5ee'" href="index.html">На главную</a></p>
</body>
</html>

```

registration.php

```

<html>
<head>
<title>Регистрация:</title>
  <link REL="STYLESHEET" TYPE="text/css" HREF="css/style.css">
</head>
<body>
<h1>Регистрация на сайте:</h1>
<?php
require "registration.html";
$login=trim($_POST['log']);
$psw=$_POST['psw'];
$rpsw=$_POST['rpsw'];
if (!$login || !$psw || !$rpsw)
{
echo    "<script    type=\"text/JavaScript\">alert(\"Заполнены    не    все
поля!\")</script>";
exit;
}
if ($psw!=$rpsw)
{
echo    "<script    type=\"text/JavaScript\">alert(\"Пароли    не
совпадают!\")</script>";
exit;
}
@ $db = mysql_pconnect('localhost', '0700', '1488');
if (!$db)
{

```

```

echo "<script type=\"text/JavaScript\">alert(\"Ошибка соединения с
БД\")</script>";
exit;
}
mysql_select_db('db_0700');
@ $check=mysql_query("SELECT * FROM user WHERE login=\".$login.\"");
if(mysql_num_rows($check)<1)
{
$query="INSERT INTO user VALUES (\".$login.\", \".$psw.\",)";
$result = mysql_query($query);
if ($result) echo "<script type=\"text/JavaScript\">alert(\"Регистрация прошла
успешно\")</script>";
}
else echo "<script type=\"text/JavaScript\">alert(\"Пользователь с таким именем
уже существует\")</script>";
?>
</body>
</html>

```

config.php

```

<?php
$db_host="localhost";
$db_name="db_0700";
$username="0700";
$password="1488";
$db_con=mysql_connect($db_host,$username,$password);
$connection_string=mysql_select_db($db_0700);
mysql_connect($db_host,$username,$password) or die('Соединение не
установлено');
mysql_select_db($db_name) or die('База не подключена');
?>

```

login.php

```

<?php include('config.php');
if($_GET['quit']==0){
if(isset($_GET['login']) && isset($_GET['psw'])){

$login = $_GET['login'];
$psw = $_GET['psw'];

$getuser_sql = 'SELECT * FROM user WHERE login="'. $login . "' AND psw="
. $psw . "'";

```

```

$getUser = mysql_query($getUser_sql);
$getUser_result = mysql_fetch_row($getUser);
$getUser_RecordCount = mysql_num_rows($getUser);
if(!$getUser_RecordCount < 1)
{
$sid=rand(0, getrandmax());
setcookie("sid", $sid);
mysql_query("update user set sid=".$sid." where login=".$login."");
echo $getUser_result[0];
}
}}
else
{
$query = "SELECT * FROM user WHERE sid=".$_COOKIE['sid'];
$result = mysql_query($query);
$get_result=mysql_fetch_row($result);
mysql_query("update user set sid=0 where login=".$get_result[0]."");
setcookie("sid", -1);
}
?>

```

ajax.js

```

function createObject() {
var request_type;
var browser = navigator.appName;
if(browser == "Microsoft Internet Explorer"){
request_type = new ActiveXObject("Microsoft.XMLHTTP");
}else{
request_type = new XMLHttpRequest();
}
return request_type;
}

var http = createObject();

var nocache = 0;
function Check()
{
http.open('get', 'check.php?nocache='+nocache);
http.onreadystatechange = loginCheck;
http.send(null);
}
function login() {

```



```

document.getElementById('login_response').innerHTML = "Загрузка..."
// Проверяем, что все поля не пустые. Используем encodeURIComponent() для
кодирования недопустимых символов в запросе.
var login = encodeURIComponent(document.getElementById('login').value);
var psw = encodeURIComponent(document.getElementById('pswLogin').value);
nocache = Math.random();
http.open('get',
'login.php?quit=0&login='+login+'&psw='+psw+'&nocache='+nocache);
http.onreadystatechange = loginReply;
http.send(null);
}
function loginReply() {
if(http.readyState == 4){
var response = http.responseText;
if(response == "")
{
document.getElementById('login_response').innerHTML = 'Неверный логин
или пароль';
}
else
{
document.getElementById('login_response').innerHTML = 'Здравствуйте,
'+response;
var log = document.getElementsByName('forlog');
for (var i=0; i<log.length; i++)
{
log[i].style.visibility='hidden';
}
document.getElementById('qqq').style.visibility='visible';
}
}
}
function loginCheck() {
if(http.readyState == 4){
var response = http.responseText;
if(response != "")
{
document.getElementById('login_response').innerHTML = 'Здравствуйте,
'+response;
var log = document.getElementsByName('forlog');
for (var i=0; i<log.length; i++)
{
log[i].style.visibility='hidden';
}
document.getElementById('qqq').style.visibility='visible';
}
}
}

```

```

    }
  }
}
function quit()
{
http.open('get', 'login.php?quit=1&nocache='+nocache);
document.getElementById('login_response').innerHTML = "";
document.getElementById('qqq').style.visibility='hidden';
var log = document.getElementsByName('forlog');
for (var i=0; i<log.length; i++)
{
log[i].style.visibility='visible';
}
http.onreadystatechange = null;
http.send(null);
}

```

style.css

```

body {
  background : url(..img/pic.gif);
}
h1 {
  font-family: Century Gothic;
  font-size: 18pt;
  color: #FFFFFF;
  font-weight: bold;
  text-align:center; }
p{
  font-family: Times New Roman;
  font-size: 14pt;
  color: #fff5ee;
  font-weight: bold;
  text-align: left;
}
.sometext {
font-family: Monotype Corsiva;
font-size: 21px;
color: "#FFFFFF";
text-align:right;
}
.login {
font-family: Monotype Corsiva;
font-size: 21px;
color: "#FFFFFF";

```

```

text-align:left;
}
.loginresponse
{
font-family: Times New Roman;
font-size: 21px;
color: #c7fcec;
text-align:left;
}

.s_table {
border: 0;
width: 20%;
background : url(../img/table.png);}
.sm_table {
border: 0;
width: 85%;
background : url(../img/table.png);}

.main {
font-family: Century Gothic;
font-size: 36pt;
color: #FFFFFF;
font-weight: bold;
text-align:center; }

.go-home{
font-family: Times New Roman;
font-size: 14pt;
color: #800000;
font-weight: bold;
}

a:link, a:visited {
color: blue;
text-decoration: none;
}

a:hover {
color: brown;
text-decoration: underline;
}

a.active, a:link.active, a:visited.active, a:hover.active {
color: blue;
}

```

```
}
```

```
.imgCenter {text-align:center;}
```