

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

**УЧЕТ ЗАКАЗОВ ДЕРЕВООБРОБАТЫВАЮЩЕЙ
КОМПАНИИ**

Выпускная квалификационная работа
обучающегося по направлению подготовки 02.03.02 Фундаментальная
информатика и информационные технологии
заочной формы обучения, группы 07001250
Стародубцева Александра Семёновича

Научный руководитель
доцент
Бурданова Е.В.

БЕЛГОРОД 2017

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. ЗАДАЧА АВТОМАТИЗАЦИИ ОБРАБОТКИ ИНФОРМАЦИИ ПО УЧЕТУ ТОВАРОВ	
1.1 Основные характеристики производства деревянных лестниц....	6
1.2 Классификация лестниц.....	7
1.3 Определение требований к информационной системе обработки информации по учету заказов деревообрабатывающего предприятия	11
1.4 Выбор СУБД.....	13
2. РАЗРАБОТКА СИСТЕМЫ	
2.1 Обзор моделей представления данных.....	19
2.2 Разработка модели данных.....	21
2.3 Описание проектируемой БД.....	25
3. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ЗАДАЧИ АВТОМАТИЗАЦИИ ОБРАБОТКИ ИНФОРМАЦИИ ПО УЧЕТУ ЗАКАЗОВ ДЛЯ ПРЕДПРИЯТИЯ ДЕРЕВООБРАБОТКИ	
3.1 Построение инфологической модели.....	27
3.2 Построение даталогической модели.....	27
3.3 Физическая модель Д.....	33
4. РАЗРАБОТКА И ОПИСАНИЕ САЙТА, РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.	
4.1 Руководство пользователя.....	37
4.2 Руководство администратора.....	41
4.3 Руководство разработчика.....	46
ЗАКЛЮЧЕНИЕ.....	48

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	49
ПРИЛОЖЕНИЕ 1.....	51
ПРИЛОЖЕНИЕ 2.....	52
ПРИЛОЖЕНИЕ 3.....	53
ПРИЛОЖЕНИЕ 4.....	56
ПРИЛОЖЕНИЕ 5.....	60
ПРИЛОЖЕНИЕ 6.....	63

ВВЕДЕНИЕ

Прогресс, достигнутый за последние несколько лет во всех аспектах вычислительной техники, включая теорию, технологию и приложения, привели к значительному расширению области применения компьютеров и росту числа их пользователей. Существенной частью современного общества являются разнообразные системы доступа и хранения информации, которые являются неотъемлемой составляющей современного научно-технического прогресса.

В настоящее время, когда интернет распространился очень широко и стоит на одной ступени с другими средствами массовой информации, очень важно для предпринимателей иметь в интернете свое представительство.

Также во многих организациях сегодня возникает проблема такой обработки и управления данными, которые обеспечили бы наиболее эффективную работу. Большинство организаций используют для этого компьютеризированные СУБД, позволяющие эффективно хранить, извлекать информацию и управлять большими объемами данных.

Поэтому было принято решение о разработке сайта включающего в себя базу данных товаров для предприятия деревообработки.

Сайт должен иметь удобную навигацию по разделам и возможность его администрирования.

Цель дипломного проекта – разработка веб-сайта, включающего в себя автоматизированную систему обработки заказов для предприятия деревообработки имеющего:

- хорошую информационную базу о товарах, а также их характеристиках;
- возможность быстрого обновления информации доступной на сайте;
- возможность ведения опроса клиентов сайта для ведения

статистической базы.

Для достижения поставленной цели необходимо решить следующие задачи:

—ознакомиться и изучить программный инструментарий, применяемый для разработки и создания Web-сайтов такие как PHP, MySQL, Apache;

—выявить и учесть методы и способы представления на Web-страницах различных видов информации, не препятствующие их доступности;

—ознакомиться с основными правилами и рекомендациями по разработке и созданию Web-сайтов;

—определиться со структурой Web-страниц.

Задачи, решаемые информационной системой:

- ввод и хранение информации о товарах предприятия;
- хранение данных;
- исключение избыточности данных;
- обеспечение целостности данных;
- обновление информации в базе данных;
- резервное архивирование данных;
- формирование и печать отчетных документов.

Выпускная квалификационная работа содержит 47 страниц, 29 рисунков, 1 таблицу и 6 приложения на 71 странице.

5. ЗАДАЧА АВТОМАТИЗАЦИИ ОБРАБОТКИ ИНФОРМАЦИИ ПО УЧЕТУ ТОВАРОВ

5.1 Основные характеристики производства деревянных лестниц

Без лестницы можно обойтись разве что в одноэтажном частном доме, да и то если в нем нет чердака. Этот крайне важный элемент сооружения гарантирует безопасность, удобство и уют для жильцов, и потому его изготовление является весьма ответственной задачей.

Динамика российского бизнеса и постоянные изменения в законодательстве создают все больше сложностей в управлении производственным предприятием. Помимо этого, специфика работы компании в деревообрабатывающей отрасли выдвигает дополнительные требования для обеспечения успешной работы на рынке.

Сегодня производители предлагают большое разнообразие лестниц для дома, выполненных из различных материалов – дерева, металла, комбинированные и т.д. Однако вот уже не одно столетие лидирующие позиции занимают деревянные лестницы, которые отличаются своим особенным внешним видом и энергетикой.

Любая конструкция состоит из ряда элементов, некоторые из которых обязательные (постоянные), а другие могут добавляться в зависимости от дизайнерской задумки.

Любая лестница состоит из следующих элементов:

- опорные балки (тетива или косоур);
- опорный столб (для винтовой лестницы);
- ступени (проступи и подступенки);
- перила (балясины и поручни) (см Рисунок 1.1)



Рисунок 1.1 – Элементы лестницы

5.2 Классификация лестниц

Выбор лестницы зависит от предпочтений жильцов, размеров помещения, стиля интерьера и назначения.

Лестницы можно классифицировать (см. Приложение 1):

1. По конструкции: с подступенками, без подступенков, с тетивами, на косоурах, винтовые (с центральной стойкой и без нее), со ступенями в виде консолей, подвесные, выдвижные и т. д.

2. По материалу: деревянные, стальные, каменные или бетонные.

3. По форме:

- прямые лестницы. Они бывают одномаршевыми или многомаршевыми, в зависимости от наличия площадок;

- лестницы с поворотом маршей (ломаные). У таких лестниц направление маршей изменяется. В месте поворота делают лестничную площадку. В зависимости от количества маршей, различают двух-, трех- и многомаршевые лестницы. Распашными называют такие лестницы, у

которых первый марш ведет к площадке, а над площадкой сооружены два марша в противоположном направлении;

- лестницы с забежными ступенями. К ним относятся одномаршевые лестницы с поворотом на 180° и с одним или двумя поворотами на 90° . Забежные ступени могут находиться внизу, наверху или посередине. Лестницы такой формы часто устанавливают в небольших индивидуальных домах;

- криволинейные лестницы. Такие лестницы (например, полукруглые) состоят из одних только забежных ступеней;

- винтовые лестницы. Это лестницы со средней линией в виде окружности и круглым пролетом. Их чаще можно встретить в просторных жилых домах;

- винтовые лестницы с центральной стойкой. Их стойка несет нагрузку;

- двухмаршевые криволинейные лестницы с промежуточной площадкой.

Винтовые лестницы вообще используются достаточно редко. Они могут разрешить вопрос с поднятием на второй этаж в условиях ограниченного пространства. Подниматься по ним быстро не получится, зато они отлично экономят место и украшают собой интерьер (см. Рисунок 1.2).



Рисунок 1.2 – Винтовая лестница

Маршевыми лестницами (см. Рисунок 1.3) называются потому, что состоят из маршей или лестничных пролетов. Это ступени, разделенные площадками. На лестнице может быть от одной до 3-х и более площадок, однако для подъема на второй этаж обычно достаточно 2-х. Маршевые лестницы гораздо проще в строительстве и расчетах, но при этом могут иметь самые разнообразные конструкции.



Рисунок 1.3 – Маршевая лестница

В отличие от винтовых, маршевые конструкции занимают довольно много места, поэтому некоторые хозяева для экономии площади строят три марша, делая расчет лестницы с поворотом 90° или два марша с поворотом 180° .

Марш, в свою очередь, может быть криволинейным с угловым заходом на лестницу и поворотными ступенями. В двух- и трехмаршевых конструкциях вместо пролетов могут быть установлены поворотные ступени.

Существует три основных группы лестниц по используемому материалу:

- деревянные
- металл с деревом
- бетонные с деревом (см. Рисунки 1.4 – 1.5_



Рисунок 1.4 – Деревянная лестница



Рисунок 1.5 – Комбинированная лестница

Для изготовления лестничных конструкций используется различная древесина (см. Таблицу 1.1). Твердость древесины делится на четыре основных типа: мягкая, средняя твердость, твердая, очень твердая.

-мягкая древесина-ель, сосна.

-средняя твердость-орех, лиственница, груша, черная сосна.

-твердая древесина-ясень, дуб, бук.

Рекомендуется использовать древесины средней твердости, твердых и очень твердых пород.

Таблица 1.1

Характеристики древесины

Древесина	Цвет	Твердость	Подверженность гниению	Износостойкость	Стоимость
Сосна	Желтовато-белый	Мягкая	Низкая	Низкая	Низкая
Лиственница	Красно-бурый	Средне твердая	Низкая	Высокая	Средняя
Дуб	Светло-желтый, коричневый	Твердая	Низкая	Высокая	Высокая
Ясень	Желтовато-серый	Твердая	Средне стойкая	Высокая	Высокая
Орех	От темно серого до коричневого	Твердая	Низкая	Высокая	Средняя
Бук	Бледно желтый - розоватый	Очень твердая	Средне стойкая	Высокая	Средняя

5.3 Определение требований к информационной системе обработки информации по учету заказов деревообрабатывающего предприятия

На торгово-производственных предприятиях любого масштаба и формы работы должна быть внедрена система автоматизации торговли и учета товаров. Компании-интеграторы, производящие свое собственное программное обеспечение, максимально заинтересованы в его продвижении и зачастую предлагают варианты неадекватные тому, что требуется достичь. В этом и заключается значительная сложность, так как при существующем изобилии подобных систем, нелегко запутаться.

Деревообрабатывающая организация занимается изготовлением и продажей деревянных лестниц. Деятельность компании организована следующим образом: компания производит и продает деревянные лестницы

из определенных материалов. Каждый товар (лестница) характеризуется ценой, параметрами изготовления и признаком наличия или отсутствия доставки.

В компанию обращаются заказчики. Для каждого из них необходимо запомнить в проектируемой базе данных стандартные данные (ФИО, адрес, телефон) и составить по каждому заказу документ, запоминая наряду с заказчиком количество купленного им товара и дату покупки. Доставка разных товаров может производиться разными способами, различными по цене и скорости. Нужно хранить информацию по тому, какой вид доставки (а, соответственно, и какую стоимость доставки) выбрал клиент при заключении сделки.

Для правильного выбора системы автоматизации торговли и учета товаров должен быть обязательно проведен тщательный анализ всех бизнес-процессов, и детально согласовано с клиентом техническое задание.

Таким образом, основными требованиями к разработке системы автоматизации торговли являются:

- 1) Быстрое внедрение и четкая бесперебойная работа системы
- 2) Возможность сравнения товаров, а также различных групп товаров по разным показателям

- 3) Простота в использовании, быстрое создание нужной отчетности

- 4) Четкая защита от несанкционированного доступа

- 5) Возможность модернизации программы

Возможность удобного архивирования данных.

Необходимость разработки собственного web-сайта организации обусловлена следующими преимуществами:

- круглосуточный представитель организации
- реклама для широчайшей аудитории
- расширение географии бизнеса
- вся нужная информация в одном месте
- мобильность общения

- экономия времени
- увеличение дохода
- поддержание имиджа

Доступ для пользователей информации к официальному сайту организации обеспечивается хостинговой компанией, с которой администрация заключает ежегодный договор. Контроль за выполнением договора осуществляется главой компании.

Поддержка работы сайта, его продвижение в поисковых системах сети Интернет, внедрение новых программных разработок и проектов осуществляется на основании ежегодно заключаемого договора о технической поддержке. Контроль над выполнением договора осуществляется главой компании.

Администратор сайта организует его информационное наполнение и размещает информацию, предоставленную главой компании и ее сотрудниками.

5.4 Выбор СУБД

Традиционных возможностей файловых систем недостаточно для построения даже простых информационных систем из-за возникающих потребностей, которые не покрываются возможностями систем управления файлами:

- а) поддержание логически согласованного набора файлов;
- б) обеспечение языка манипулирования данными;
- в) восстановление информации после разного рода сбоев;
- г) реально параллельная работа нескольких пользователей.

Можно считать, что если прикладная информационная система опирается на некоторую систему управления данными, обладающую этими свойствами, то эта система управления данными является системой управления базами данных (СУБД).

Произведем анализ наиболее популярных СУБД.

СУБД Microsoft Access

СУБД Access входит в состав широко распространенного семейства офисных приложений Microsoft Office. Microsoft Access на сегодняшний день является одним из самых распространенных настольных приложений для работы с базами данных. Изначально Access имела ряд уникальных возможностей, таких как умение сводить воедино информацию из самых разных источников (электронных таблиц, текстовых файлов, других баз данных), представление данных в удобном для пользователя виде с помощью таблиц, диаграмм, отчетов, интеграция с другими компонентами Microsoft Office.

Access представляет собой локальную СУБД, ориентированую на разработку баз данных малой и средней сложности, с которыми одновременно могут работать один или несколько пользователей. Для защиты целостности данных при обращении к ним производится блокировка (два пользователя не могут одновременно изменить содержимое одной и той же ячейки).

Microsoft Access является настольной СУБД реляционного типа, которая имеет все необходимые средства для выполнения необходимых функций. Достоинством Access является то, что она имеет очень простой графический интерфейс, который позволяет не только создавать собственную базу данных, но и разрабатывать простые и сложные приложения.

Access позволяет использовать электронные таблицы и таблицы из других настольных и серверных баз данных для хранения информации, необходимой приложению.

В отличие от других настольных СУБД, Access хранит все данные в одном файле, хотя и распределяет их по разным таблицам, поэтому можно создать сколько угодно таблиц.

Access предоставляет простые и удобные возможности для разработки структуры БД и пользовательского интерфейса. Одно и то же приложение

служит как инструментом создания баз данных, так и средством работы с готовыми БД.

База данных Access может содержать объекты следующих видов:

- Таблицы
- Запросы
- Формы
- Отчеты
- Страницы
- Макросы
- Модули

Все эти объекты являются инструментами, позволяющими выполнять различные действия над данными.

СУБД MySQL

Программное обеспечение MySQL представляет собой очень быстрый многопоточный, многопользовательский надежный SQL-сервер баз данных (SQL - язык структурированных запросов). Сервер MySQL предназначен как для критических по задачам производственных систем с большой нагрузкой, так и для встраивания в программное обеспечение массового распространения.

MySQL написан на C и C++. Протестирован на множестве различных компиляторов.

Основные характеристики MySQL:

- а) работает на различных платформах;
- б) API для C, C++, Eiffel, Java, Perl, PHP, Python, Ruby и Tcl;
- в) полностью многопоточный с использованием потоков ядра. Это означает, что, если такая возможность обеспечивается, можно легко организовать работу с несколькими процессорами;
- г) очень быстрые дисковые таблицы на основе B-деревьев со сжатием индексов;

д) очень быстрая базирующаяся на потоках система распределения памяти;

е) очень быстрые соединения, использующие оптимизированный метод однопроходного мультисоединения;

ж) хеш-таблицы в памяти, используемые как временные таблицы;

SQL-функции реализованы при помощи хорошо оптимизированной библиотеки классов, поэтому они выполняются настолько быстро, насколько это возможно

СУБД MySQL поддерживает большое количество типов данных: целочисленные со знаком и беззнаковые, длиной в 1, 2, 3, 4 и 8 байтов, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET и ENUM.

MySQL обеспечивает поддержку SQL92. Имена функций не конфликтуют с именами таблиц и столбцов. Например, ABS является корректным именем столбца. В одном и том же запросе могут указываться таблицы из различных баз данных (с версии 3.22).

MySQL поддерживает такие расширения, как процедуры и функции.

Можно добавлять свои функции двумя методами:

а) добавить функцию посредством интерфейса определяемых пользователем функций (user-definablefunction - UDF);

б) добавить функцию, как встроенную функцию MySQL. Эти функции компилируются вместе с остальным кодом сервера mysqld и становятся постоянно доступными.

Система безопасности основана на привилегиях и паролях, за счет чего обеспечивается гибкость и безопасность, и с возможностью верификации с удаленного компьютера. Пароли защищены, т.к. они при передаче по сети при соединении с сервером шифруются.

Масштабируемость и ограничения

Управляет очень большими базами данных. Компания MySQLAB. использует MySQL для работы с несколькими базами данных, которые

содержат 50 миллионов записей, кроме того, нам известны пользователи, использующие MySQL для работы с 60000 таблицами, включающими около 5000000000 строк.

Для каждой таблицы разрешается иметь до 32 индексов. Каждый индекс может содержать от 1 до 16 столбцов или частей столбцов. Максимальная ширина индекса 500 бит (это значение может быть изменено при компиляции MySQL). Для индекса может использоваться префикс поля CHAR или VARCHAR.

СУБД InterBase

InterBase SQL Server разрабатывается и продается фирмой Interbase Software Corporation (ISC).

СУБД InterBase очень проста в установке. На Win32-платформах инсталлятор IB имеет хороший графический интерфейс (InstallShield), а на UNIX-платформах - традиционный (для этих платформ) интерфейс командной строки.

Одним из достоинств СУБД InterBase является простота администрирования. Главными предпосылками при этом являются маленькое количество опций настройки, отсутствие привязки к аппаратным особенностям (например – использования низкоуровневого обращения к аппаратуре (Informix, Sybase, Oracle)) и особенностям операционной системы. Все это, конечно, приводит к упрощению администрирования, но в ущерб производительности и гибкости.

InterBase поддерживает синтаксис языка SQL, совместимый с начальным уровнем стандарта SQL-92 и некоторые возможности SQL 3. Так, InterBase поддерживает механизм ролей, хранимые процедуры, функции, транзакции.

Используя InterBase API можно создавать как внешние процедуры, так и внешние функции. В IB создано процедурное расширение языка SQL. С его помощью можно писать триггеры и хранимые процедуры. Элементы этого языка нельзя использовать при интерактивной работе.

IBM поддерживает механизм доменов, позволяющий создать тип с ограничениями (например - >0 и <100), но не позволяет создать совершенно новый тип (с методами, атрибутами, в общем, с нестандартной структурой и с возможностью расширенной обработки этого типа), как, например, это можно сделать в Oracle и Informix.

В IBM также невозможно расширение SQL. Хранимые процедуры и триггеры в IBM можно писать только на встроенном языке.

IBM не поддерживает никаких объектных расширений.

Обоснование выбора СУБД

Проведенный анализ показал, что наиболее оптимальным решением будет использование СУБД MySQL, т.к. MySQL обладает очень широким диапазоном средств для ввода, анализа и представления данных. Эти средства являются не только простыми и удобными, но и высокопродуктивными, что обеспечивает высокую скорость разработки приложений и позволяет эффективно строить системы любой сложности, а также реализуется одновременная работа нескольких пользователей, что необходимо для решения поставленной задачи.

6. РАЗРАБОТКА СИСТЕМЫ

6.1 Обзор моделей представления данных

Модель представления данных - есть множество элементов (объектов, типов данных) и связей (отношений) между ними, ограничений операций над типами данных и отношениями.

Классификация моделей данных (см. Рисунок 2.1)

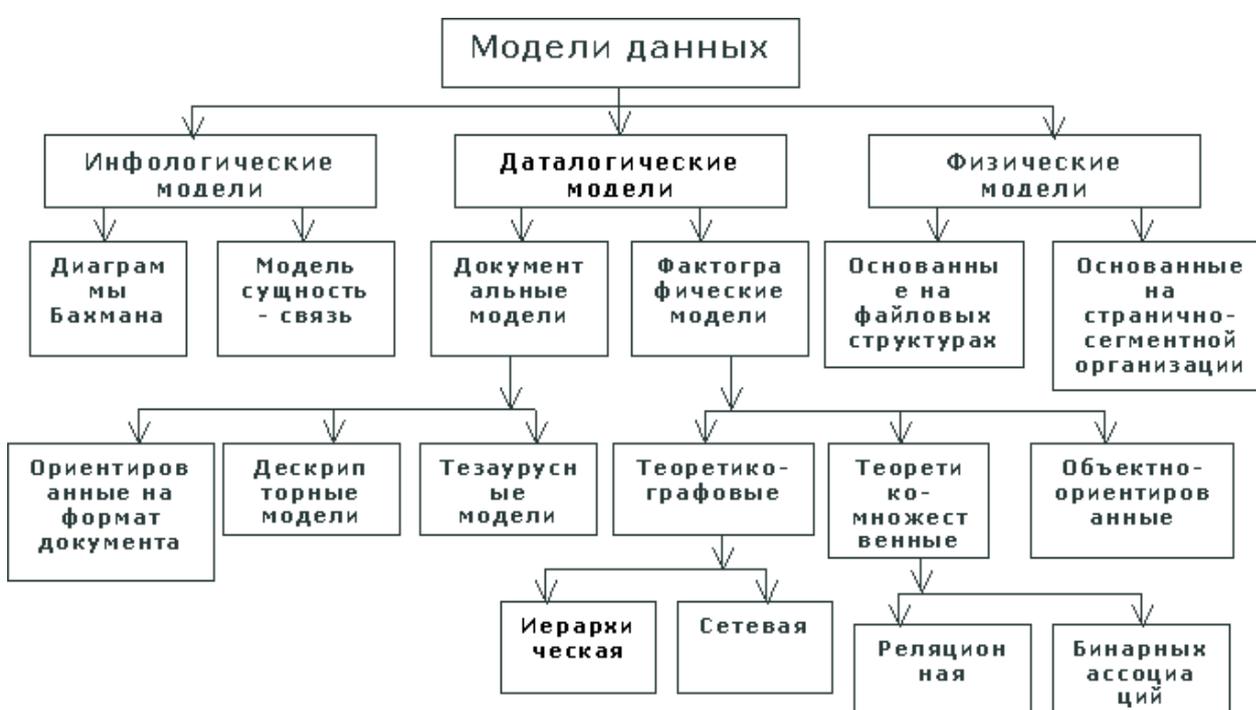


Рисунок 2.1 – Классификация МД

Множество допустимых типов данных и их отношений образует структуру данных. В модели данных, следовательно, выделяется три компонента: структура данных; ограничения, определяющие допустимое состояние БД; множество операций, применяемых для поиска и обновления данных. Эти компоненты отображаются языковыми и программными средствами описания и манипулирования данными.

На практике для реализации БД используются фактографические модели представления данных. Рассмотрим основные разновидности фактографических моделей представления данных.

Иерархическая модель. Модель описывает объекты, находящиеся между собой в отношении подчинения.

Сетевая модель. Сетевая модель позволяет описывать более сложные виды взаимоотношений между данными. Однако расширение возможностей достигается за счет большей сложности реализации самой модели и трудности манипулирования данными.

Реляционная модель. В реляционной модели данные представляются в виде таблиц, состоящих из строк и столбцов. Каждая строка таблицы - информация об одном конкретном объекте, столбцы содержат свойства этого объекта. Взаимоотношения между объектами задаются с помощью связей между столбцами таблиц. Реляционная модель на сегодняшний день наиболее распространена. Она достаточно универсальна и проста в проектировании.

Объектная модель. В этой модели данные представляются в форме объектов. Объект имеет набор свойств, называемых атрибутами, и может включать в себя также процедуры для обработки данных, которые называют методами. В настоящее время не существует единого подхода к реализации объектных баз данных. Объектный подход - набор общих принципов, которые могут применяться при проектировании различных приложений.

Гибридные модели. Пример гибридной модели — объектно-реляционная модель. В ней использовано некоторое сходство между реляционной и объектной идеологией. Строки таблиц реляционной модели соответствуют объектам объектной модели, столбцы таблиц - атрибутам объектов. Таблицы в целом являются аналогом классов.

Начальной стадией проектирования системы баз данных является построение семантической модели предметной области, которая базируется на анализе свойств и природы объектов предметной области и

информационных потребностей будущих пользователей разрабатываемой системы. Эту стадию принято называть концептуальным проектированием системы, а ее результат – концептуальной моделью.

Назначение концептуальных моделей определяет и некоторые специфические требования к средствам их представления.

6.2 Разработка модели данных

Задача автоматизации учета заказов товаров в деревообрабатывающей организации является частью общей задачи бухгалтерского учета. Она не является отдельной областью, а тесно связана с другими областями. Поэтому необходимо четко определить границы поставленной задачи, чтобы впоследствии избежать необходимости изменения структуры программы в связи с появившимися новыми подзадачами с одной стороны, и не повторять обработку данных из других областей, для работы с которыми уже разработаны другие программные средства.

Ассортимент организации достаточно велик, поэтому существует необходимость инвентаризации. Имея полные и конкретные данные легче вести продажу и контроль над доставкой и продажей.

Одним из самых распространенных инструментальных средств является ER-модель (от английского Entity-Relationship). Моделирование предметной области базируется на использовании графических диаграмм, включающих небольшое число разнородных компонентов.

Основными понятиями ER-модели являются сущность, связь и атрибут.

Сущность (или объект) - это совокупность объектов предметной области, обладающих одинаковым набором свойств. Это может быть личность, вещь, событие и т.п.

В проектируемой предметной области можно выделить такие сущности: ЛЕСТНИЦА, ФОРМА, МАТЕРИАЛ, ЦЕНА, ЦВЕТ, КЛИЕНТ, ДОСТАВКА, ЗАКАЗ.

В диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности.

Под атрибутом будем понимать логически неделимый элемент структуры информации, характеризуемый множеством атомарных значений. Каждый атрибут идентифицируется именем.

Каждой сущности соответствует набор атрибутов, характеризующих свойства данного объекта. Например, для сущности КЛИЕНТ можно выделить следующие атрибуты: фамилия, имя, отчество, адрес, телефон.

Под экземпляром сущности (или объекта) понимается реализация сущности в конкретных значениях атрибутов. Например, экземпляру сущности ДОСТАВКА может соответствовать такая информация: "ПЭК, 7000".

При отражении в информационной системе каждый объект (сущность) представляется своим идентификатором, который должен быть уникальным для каждого экземпляра объекта.

Связь - это графически изображаемая ассоциация, устанавливаемая между двумя объектами. В любой связи выделяются два конца, на каждом из которых указывается имя конца связи, степень конца связи (сколько экземпляров данной сущности связывается), обязательность связи (т.е. любой ли экземпляр данной сущности должен участвовать в данной связи).

Различают связи типа "один ко одному" (1 : 1), "один ко многим" (1 : M), "многие к одному" (M : 1) и "многие ко многим" (M : M).

Первоначально перечень атрибутов формируется по одному сообщению или группе сообщений. После обработки сообщений формируется общий перечень атрибутов, а затем атрибуты агрегируются в объекты.

Агрегация атрибутов

Когда перечень атрибутов составлен, очередная задача - агрегация атрибутов в объекты, или другими словами, разбиение перечня атрибутов на некоторые исходные сущности.

После того, как объекты определены строится ER-диаграмма и выясняются типы связей между объектами: 1 : 1, 1: M, M : M (см.Рисунок 2.2).

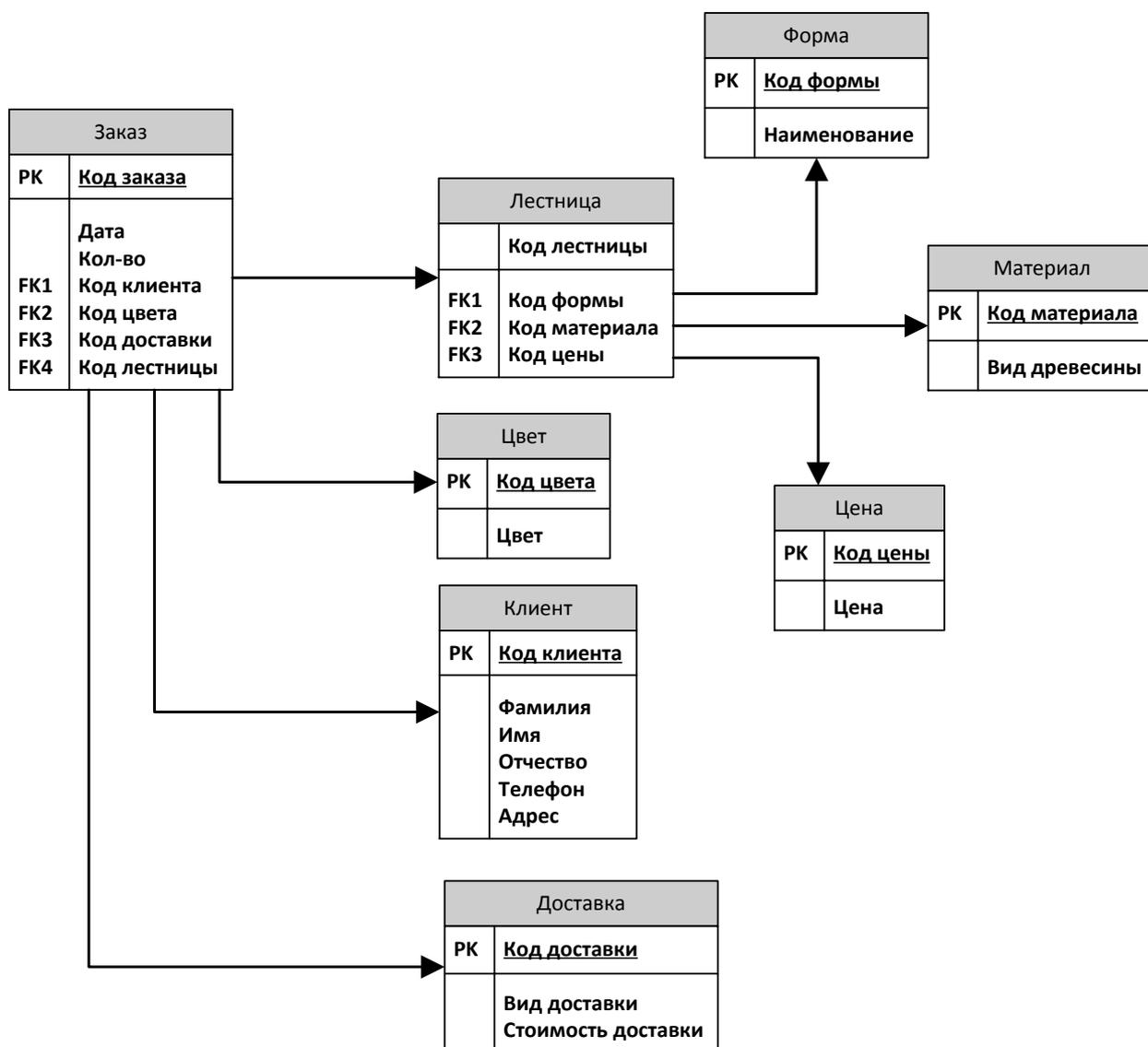


Рисунок 2.2 – Концептуальная модель

Лестница

В этой таблице будут использоваться следующие поля: Код лестницы, Код формы, Код материала, Код цены. В качестве ключевого поля будет использоваться Код лестницы. Данная таблица связана с таблицами Заказ, Форма, Материал и Цена.

Форма

В этой таблице приводятся варианты форм лестниц. Она содержит столбцы: Код формы, Наименование. В качестве ключевого поля будет использоваться Код формы. Данная таблица связана только с таблицей Лестница.

Материал

В этой таблице поля содержат виды древесины для производства лестниц. Она содержит столбцы: Код материала, Вид древесины. В качестве ключевого поля будет использоваться Код материала. Данная таблица связана только с таблицей Лестница.

Цена

В этой таблице используются следующие поля: Код цены, Цена. В качестве ключевого поля будет использоваться Код цены. Данная таблица связана только с таблицей Лестница.

Цвет

В этой таблице приводятся варианты цветового решения для лестниц. Она содержит столбцы: Код цвета, Цвет. В качестве ключевого поля будет использоваться Код цвета. Данная таблица связана только с таблицей Лестница.

Клиент

В этой таблице будут использоваться следующие поля: Код клиента, Фамилия, Имя, Отчество, Телефон, Адрес. В качестве ключевого поля будет использоваться Код клиента. Данная таблица связана с таблицей Заказ.

Доставка

В этой таблице необходимы следующие характеристики: Код доставки, Вид доставки, Стоимость доставки. В качестве ключевого поля будет использоваться Код доставки. Данная таблица связана с таблицей Заказ.

Заказ

В этой таблице описываются заказы: Код заказа, Дата, Кол-во, Код лестницы, Код клиента, Код цвета, Код доставки. В качестве ключевого поля

будет использоваться Код заказа. Данная таблица связана с таблицами Лестница, Цвет, Клиент, Доставка.

6.3 Описание проектируемой БД

Проектирование реализации – это последний этап проектирования, он состоит из двух шагов: конструирования схемы и реализации БД, а затем разработки программного обеспечения.

Разработка технологии ведения ИС. Программное обеспечение технологии ведения ИС составляют сервисные средства, необходимые для выполнения большинства процессов, включенных в технологию:

- ввод информации в систему;
- защита данных;
- управление использованием данных;
- управление эффективностью системы.

Это могут быть стандартные программные продукты (из состава СУБД или независимо предоставляемые), либо оригинальные программные разработки. Определяя состав программного обеспечения, проектировщик ИС оговаривает его состав, а для оригинальных программ разрабатываются их алгоритмы.

В структуру БД входят различные компоненты – агрегаты, массивы, файлы и др.

Режимы работы с базами данных

Обычно с базами данных работают две категории пользователей. Первая категория – проектировщики. Их задача состоит в разработке структуры таблиц базы данных и согласование ее с заказчиком.

Вторая категория исполнителей, работающих с базами данных, - пользователи. Они получают исходную базу данных от проектировщиков и занимаются ее наполнением и обслуживанием. В общем случае пользователи не имеют средств доступа к управлению структурой базы – только к данным,

да и то не ко всем, а к тем, работа с которыми предусмотрена на конкретном рабочем месте.

Соответственно СУБД имеет два режима работы: проектировочный и пользовательский. Первый режим предназначен для создания или изменения структуры базы и создания ее объектов. Во втором режиме происходит использование ранее подготовленных объектов для наполнения базы или получения данных из нее.

Запросы используются для просмотра, изменения и анализа данных различными способами. Запросы также можно использовать в качестве источников записей для форм, отчетов и страниц доступа к данным. В MySQL есть несколько типов запросов:

- Запросы на выборку
- Запросы с параметрами
- Перекрестные запросы
- Запросы на изменение
- Запросы SQL

В данной дипломной работе необходимы следующие запросы:

- 1) Сортировка списка заказов по определенному параметру
- 2) Поиск по определенным характеристикам
- 3) Вычисление стоимости лестниц в зависимости от характеристик
- 4) Удаление данных о товарах, или перенос в архив и др.

Для быстрого поиска по таблицам создаются запросы. При нажатии на запрос, требуется ввести данные, например, характеристики товара или цену товара.

Для ввода исходных данных используются формы. Для просмотра имеющихся данных так же используются формы.

7. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ЗАДАЧИ АВТОМАТИЗАЦИИ ОБРАБОТКИ ИНФОРМАЦИИ ПО УЧЕТУ ЗАКАЗОВ ДЛЯ ПРЕДПРИЯТИЯ ДЕРЕВООБРАБОТКИ

7.1 Построение инфологической модели

На данном этапе проектирования БД проводится анализ предметной области, определяются семантические (смысловые) утверждения, которые ограничивают предметную область, информационные потоки, существующие в ней (например, предприятия), входные и выходные документы для решаемой задачи.

Нельзя сказать, что в настоящее время существует какой-либо стандарт или хотя бы общепринятый способ построения инфологической модели. Для описания ИЛМ используются как языки аналитического (описательного) типа, так и графические средства. Графическое представление является наиболее наглядным и простым для восприятия и анализа.

Данные заносятся в таблицы 3.1 – 3.9 (см. Приложение 4)

7.2 Построение даталогической модели

Любая СУБД оперирует с допустимыми для нее логическими единицами данных. Кроме того, многие СУБД накладывают количественные и иные ограничения на структуру базы данных.

Поэтому, прежде чем приступить к построению даталогической модели, необходимо детально изучить особенности различных СУБД, определить факторы, влияющие на выбор проектного решения, провести анализ известных средств автоматизации, сделать свой выбор, учитывая специфику конкретной ситуации.

Даталогическое проектирование является проектированием логической структуры базы данных с учетом влияния возможности физической организации данных, предоставляемых конкретной СУБД.

Поэтому знание особенностей физической организации данных является полезным при проектировании логической структуры.

Конечным результатом даталогического проектирования является описание логической структуры базы данных на языке описания данных. Однако если проектирование выполняется «вручную», то для большей наглядности сначала строится схематическое графическое изображение структуры базы данных. При этом должно быть обеспечено однозначное соответствие между конструкциями языка описания данных и графическими обозначениями информационных единиц и связей между ними. Графическое представление используется и при автоматизированном проектировании структуры базы данных как интерфейсное средство общения с проектировщиком, и при документировании проекта.

Спроектировать логическую структуру базы данных означает определить все информационные единицы и связи между ними, задать их имена; если для информационных единиц возможно использование разных типов, то необходимо определить их тип. Следует также задать некоторые количественные характеристики, например длину поля.

При проектировании логической структуры БД осуществляются преобразование исходной инфологической модели в модель данных, поддерживаемую конкретной СУБД, и проверка адекватности полученной даталогической модели отображаемой предметной области.

Каждому типу модели данных и каждой разновидности модели, поддерживаемой конкретной СУБД, присущи свои специфические особенности. Вместе с тем имеется много общего во всех структурированных моделях данных и принципах проектирования БД в их среде. Все это дает возможность использовать единый методологический подход к проектированию структуры базы данных.

В БД отражается определенная предметная область. Поэтому процесс проектирования БД предусматривает предварительную классификацию объектов предметной области, систематизированное представление информации об объектах и связях между ними.

На проектные решения оказывают влияние особенности требуемой обработки данных. Поэтому соответствующая информация должна быть определенным образом представлена и проанализирована на начальных этапах проектирования БД.

Данные о предметной области и особенностях обработки информации в ней фиксируются в инфологической модели. В связи с этим одним из первых шагов проектирования является определение состава БД, т.е. перечня тех показателей, которые целесообразно хранить в БД.

При проектировании логической структуры БД осуществляются преобразование исходной инфологической модели в модель данных, поддерживаемую конкретной СУБД, и проверка адекватности полученной даталогической модели отображаемой предметной области.

Для любой предметной области существует множество вариантов проектных решений ее отображения в даталогической модели. Методика проектирования должна обеспечивать выбор наиболее подходящего проектного решения.

Переход от ER-модели к схеме реляционной базы данных

1. Каждая простая сущность превращается в таблицу (отношение). Имя сущности становится именем таблицы. Каждый простой атрибут становится столбцом таблицы с тем же именем.

2. Компоненты уникального идентификатора сущности превращаются в первичный ключ. Если имеется несколько возможных уникальных идентификаторов, выбирается наиболее используемый. Учитываются также следующие факторы:

- длина ключа – в качестве первичного ключа выбирается, как правило, самый короткий из вероятных ключей;

- стабильность – желательно выбирать в качестве первичного ключа атрибуты, которые не изменяются;
- мнемоничность – при прочих равных условиях следует отдавать предпочтение тем из вероятных ключей, которые легче запомнить.

Некоторые СУБД позволяют автоматически генерировать в качестве ключа таблицы поле типа «счетчик». Этот искусственный код можно использовать для простых объектов, если в предметной области не предполагается применение другой системы кодирования.

Если в состав уникального идентификатора входят связи, то к числу столбцов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на дальнем конце связи (процесс может продолжаться рекурсивно).

3. Каждому из многозначных атрибутов ставится в соответствие отношение, полями которого будут идентификатор, выбранный в качестве первичного ключа, и многозначный атрибут. Ключ этого отношения будет составным, включающим оба эти атрибута.

4. Если сущность имеет необязательный атрибут, возможны два варианта:

- если таким свойством обладают многие экземпляры объекта, его можно хранить как обычный атрибут в той же таблице (столбец может содержать неопределенные значения);
- если свойством обладает малое число экземпляров, то можно выделить отношение, включающее идентификатор и соответствующий атрибут. Отношение будет содержать столько строк, сколько объектов имеет свойство.

5. Если сущность имеет составной атрибут, то возможны два варианта:

- составному свойству ставится в соответствие отдельное поле;
- каждому из составляющих элементов составного свойства ставится в соответствие отдельное поле.

Выбор варианта зависит от характера обработки данных. При реализации запросов проще объединить поля, чем выделить часть поля. Если предполагается использование компонентов атрибута, лучше вариант 2, иначе – вариант 1.

6. Бинарные связи один-к-одному и один-ко-многим становятся внешними ключами. Создается копия уникального идентификатора с конца связи «один», и соответствующие столбцы составляют внешний ключ.

Связь один-к-одному между сущностями встречается редко. Если класс принадлежности обеих сущностей является обязательным, то для отображения обеих связанных сущностей можно использовать одну таблицу:

Однако таким решением злоупотреблять не следует. Если для каждого объекта потребуются свои связи или в запросах потребуется информация по каждой сущности, то выбранное решение усложнит или замедлит работу с БД.

Если для каждой сущности создаются отдельные отношения, то информацию о связях можно отразить, включив в одно из отношений идентификатор из другого отношения. Причем это можно сделать в любом из отношений.

Если класс принадлежности одной из сущностей является необязательным, то идентификатор сущности с необязательным классом добавляется в отношение, соответствующее сущности с обязательным классом принадлежности.

Если класс принадлежности обеих сущностей является необязательным, то, чтобы избежать наличия пустых полей, следует использовать три отношения: по одному для каждой сущности и одно – для отображения связи между ними.

7. Преобразование бинарной связи один-ко-многим (1:N) зависит только от класса принадлежности N-связной сущности. Если он является обязательным, то можно использовать два отношения (по одному для каждой

сущности). В отношении для N-связной сущности добавляется идентификатор 1-связной сущности.

Если класс принадлежности N-связной сущности является необязательным, то для отображения связи создается третье отношение, которое будет содержать ключи каждой из связанных сущностей.

8. Для бинарной связи многие-ко-многим (M:N) потребуются три отношения: по одному для каждой сущности и одно дополнительное – для отображения связи между ними. Последнее отношение будет содержать идентификаторы связанных объектов. Ключ этого отношения будет составным.

9. В случае N-арной связи необходимо использовать (n+1) отношение – по одному для каждой сущности, и одно для связи. Идентификатор каждой сущности станет первичным ключом соответствующего отношения. Отношение, порожаемое связью, будет иметь среди своих атрибутов ключи каждой сущности. Если связь имеет атрибуты, то они становятся атрибутами отношения связи.

10. Обобщающей сущности соответствует одно отношение, причем ключ сущности становится ключом отношения. Этому отношению приписываются общие для всех ролевых сущностей атрибуты. Ролевые элементы и связи, их соединяющие, порождают такое число отношений, которое определяется ранее описанными правилами, причем каждая роль трактуется как обычная сущность. Связываются отношения с помощью ключевого атрибута. Каждому значению ключевого атрибута ролевой сущности соответствует одна запись в обобщающем отношении с таким же значением ключа.

11. Индексы создаются для первичного ключа (уникальный индекс), а также внешних ключей и тех атрибутов, которые будут часто использоваться в запросах.

Итак, выполним правила перехода от инфологической модели данных к даталогической модели данных.

Мы имеем следующие объекты: «Лестница», «Форма», «Материал», «Цена», «Цвет», «Клиент», «Доставка», «Заказ». Объекты Лестница – Форма; Лестница – Материал, Лестница – Цена, Заказ – Клиент, Заказ – Цвет, Заказ – Доставка имеют связь 1:М, поэтому вводим у каждого объекта уникальный ключ. А объекты Заказ – Лестница имеют связь М:М, поэтому вводим дополнительную таблицу в которой уникальные ключи таблиц «Заказ» и «Лестница» станут внешними. Итак, после второго этапа проектирования базы данных мы получили даталогическую модель данных, представленную в таблицах 3.10-3.18 (см. Приложение 5).

7.3 Физическая модель БД

При создании базы данных были созданы и заполнены таблицы: «Ladders» – Лестница, «Forms» – Форма, «Prices» – Цена, «Materials» – Материал, «Delivery» – Доставка, «Colors» – Цвет, «Clients» – Клиент, «Orders» – Заказ (см. Рисунки 3.19 – 3.27).

Table Name	Columns
u144736188_lad_forms	id : int(10) unsigned type : varchar(255)
u144736188_lad_prices	id : int(10) unsigned price : int(10) unsigned
u144736188_lad_materials	id : int(10) unsigned type : varchar(255)
u144736188_lad_ladders	id : int(10) unsigned idForm : int(10) unsigned idMaterial : int(10) unsigned idPrice : int(10) unsigned
u144736188_lad_delivery	id : int(10) unsigned type : varchar(255) price : int(10)
u144736188_lad_colors	id : int(10) unsigned color : varchar(255)
u144736188_lad_clients	id : int(10) unsigned surname : varchar(255) name : varchar(255) secondName : varchar(255) phone : varchar(255) address : text
u144736188_lad_orders	id : int(10) unsigned date : int(10) amount : int(10) idClient : int(10) idLadder : int(10) idColor : int(10) idDelivery : int(10) isArchive : int(10) unsigned

Рисунок 3.19 – Таблицы базы данных учета заказов

В главной таблице «Ladders» храниться информация о лестницах. Поля, находящиеся в этой таблице: id, idForm, idMaterial, idPrice. Эта таблица объединяет в себе информацию характеризующую лестницы, такую как форма, вид древесины и цена. В свою очередь эти характеристики отражены в таблицах «Forms», «Materials», «Prices».

				id	idForm	idMaterial	idPrice
<input type="checkbox"/>		Изменить		Копировать		Удалить	1
<input type="checkbox"/>		Изменить		Копировать		Удалить	2
<input type="checkbox"/>		Изменить		Копировать		Удалить	3
<input type="checkbox"/>		Изменить		Копировать		Удалить	4
<input type="checkbox"/>		Изменить		Копировать		Удалить	5
<input type="checkbox"/>		Изменить		Копировать		Удалить	6
<input type="checkbox"/>		Изменить		Копировать		Удалить	7
<input type="checkbox"/>		Изменить		Копировать		Удалить	8
<input type="checkbox"/>		Изменить		Копировать		Удалить	9
<input type="checkbox"/>		Изменить		Копировать		Удалить	10
<input type="checkbox"/>		Изменить		Копировать		Удалить	11
<input type="checkbox"/>		Изменить		Копировать		Удалить	12
<input type="checkbox"/>		Изменить		Копировать		Удалить	13
<input type="checkbox"/>		Изменить		Копировать		Удалить	14
<input type="checkbox"/>		Изменить		Копировать		Удалить	15

Рисунок 3.20 – Таблица «Ladders»

				id	type			
<input type="checkbox"/>		Изменить		Копировать		Удалить	1	Прямая
<input type="checkbox"/>		Изменить		Копировать		Удалить	2	Г-образная
<input type="checkbox"/>		Изменить		Копировать		Удалить	3	П-образная

Рисунок 3.21 – Таблица «Forms»

				id	type			
<input type="checkbox"/>		Изменить		Копировать		Удалить	1	Сосна
<input type="checkbox"/>		Изменить		Копировать		Удалить	2	Лиственница
<input type="checkbox"/>		Изменить		Копировать		Удалить	3	Дуб
<input type="checkbox"/>		Изменить		Копировать		Удалить	4	Бук
<input type="checkbox"/>		Изменить		Копировать		Удалить	5	Ясень

Рисунок 3.22 – Таблица «Materials»

					id	price		
<input type="checkbox"/>		Изменить		Копировать		Удалить	1	50000
<input type="checkbox"/>		Изменить		Копировать		Удалить	2	60000
<input type="checkbox"/>		Изменить		Копировать		Удалить	3	70000
<input type="checkbox"/>		Изменить		Копировать		Удалить	4	80000
<input type="checkbox"/>		Изменить		Копировать		Удалить	5	90000
<input type="checkbox"/>		Изменить		Копировать		Удалить	6	110000
<input type="checkbox"/>		Изменить		Копировать		Удалить	7	115000
<input type="checkbox"/>		Изменить		Копировать		Удалить	8	120000
<input type="checkbox"/>		Изменить		Копировать		Удалить	9	115000
<input type="checkbox"/>		Изменить		Копировать		Удалить	10	120000

Рисунок 3.23 – Таблица «Prices»

Таблица «Clients» – Клиент отражает поля, связанные с данными о клиентах организации. Поля, находящиеся в этой таблице: id, surname, name, secondName, phone, address.

id	surname	name	secondName	phone	address
9	Иванов	Антон	Петрович	89161248520	г.Москва, пр.Кирова, 56, кв.255
10	Яковлева	Елена	Сергеевна	89161111111	г.Москва, ул.Победы, 15, кв.158
11	Ткачев	Петр	Владимирович	89161248522	г.Санкт-Петербург, ул.Первой Конной Армии, 12, кв....
12	Петров	Семен	Иванович	+380631234568	г.Москва, пр.Кирова, 56, кв.255
13	Крсанова	Алена	Николаевна	8-920-209-888	г.Белгород, ул. Ленина, 26
14	Тихненко	Наталья	Олеговна	310450	г. Курск, ул. Менделеева, 49
15	Зуева	Ирина	Олеговна	9152413364	г. Курск, пр-т Хрущева, 13

Рисунок 3.24 – Таблица «Clients»

В таблице «Delivery» хранятся данные о типах и стоимости доставки. Поля, имеющиеся в таблице: id, type, price.

					id	type	price		
<input type="checkbox"/>		Изменить		Копировать		Удалить	1	ПЭК	7000
<input type="checkbox"/>		Изменить		Копировать		Удалить	2	Деловые линии	5000
<input type="checkbox"/>		Изменить		Копировать		Удалить	3	Самовывоз	0

Рисунок 3.25 – Таблица «Delivery»

Создана таблица «Orders» в которой объединяются все таблицы и соответственно все данные заказа. Поля, имеющиеся в таблице: id, date, amount, idClient, idLadder, idColor, idDelivery.

id	date	amount	idClient	idLadder	idColor	idDelivery
9	02062017	1	9	9	4	3
10	02062017	2	10	11	1	2
11	03062017	1	11	13	2	1
13	03062017	1	13	8	2	3
14	03062017	1	14	1	1	1

Рисунок 3.26 – Таблица «Orders»

Для нормализации связи М – М между объектами «Лестница» и «Заказ» была создана таблица «Заказано», в которой производится соответствие между полями код товара и код продажи.

В ходе проектирования были созданы следующие запросы:

- 1) Сортировка списка заказов по клиентам
- 2) Поиск заказов по параметрам
- 3) Вычисление количества заказов сделанных определенным клиентом за определенный день
- 4) Сумма заказов за день
- 5) Количество заказов в архиве.

Программный код и результат созданных запросов приведены в приложении В.

Также были созданы отчеты, показанные на разработанном в рамках диплома сайте.

Созданные отчеты подробно рассматриваются в руководстве для пользователя.

8. РАЗРАБОТКА И ОПИСАНИЕ САЙТА, РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ.

8.1 Руководство пользователя

Сайт организации «Дерево» будет находится поадресу: <http://ladder-online.esy.es/>. При открытии в браузере взору посетителя предстаеглавная страница (см. Рисунок 4.1).

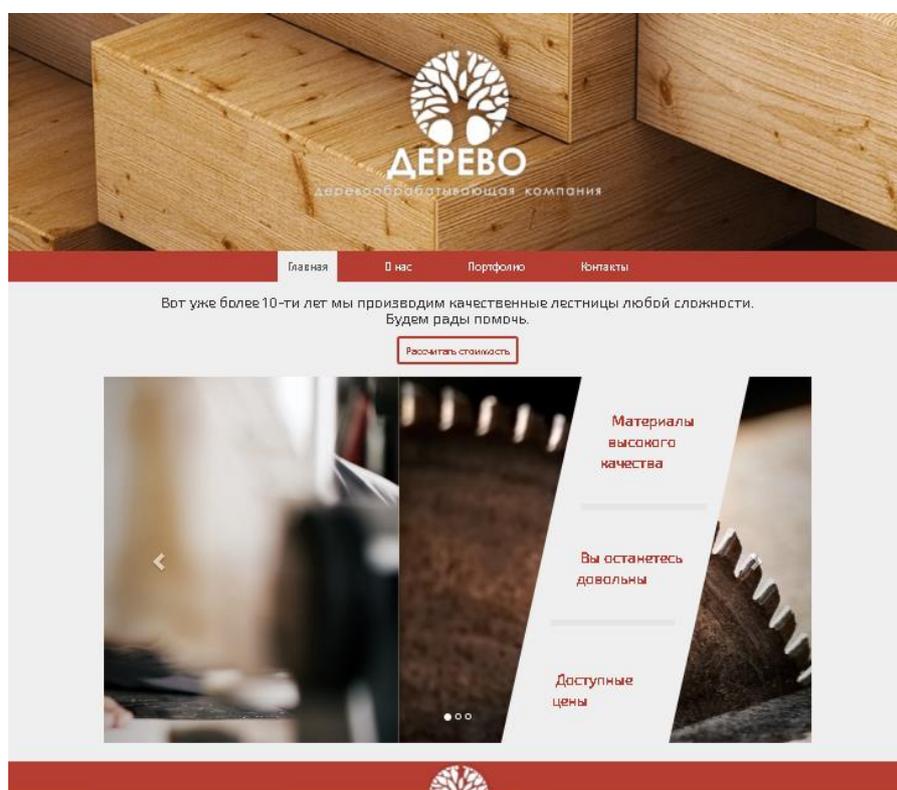


Рисунок 4.1 – Главная страница сайта организации «Дерево»

На этой странице посетителю предлагается выбрать и посетить один из представленных разделов сайта – «О нас», «Портфолио», «Контакты» (см. Рисунки 4.2, 4.3, 4.4), а также возможность рассчитать стоимость лестницы по параметрам, удовлетворяющим посетителя.

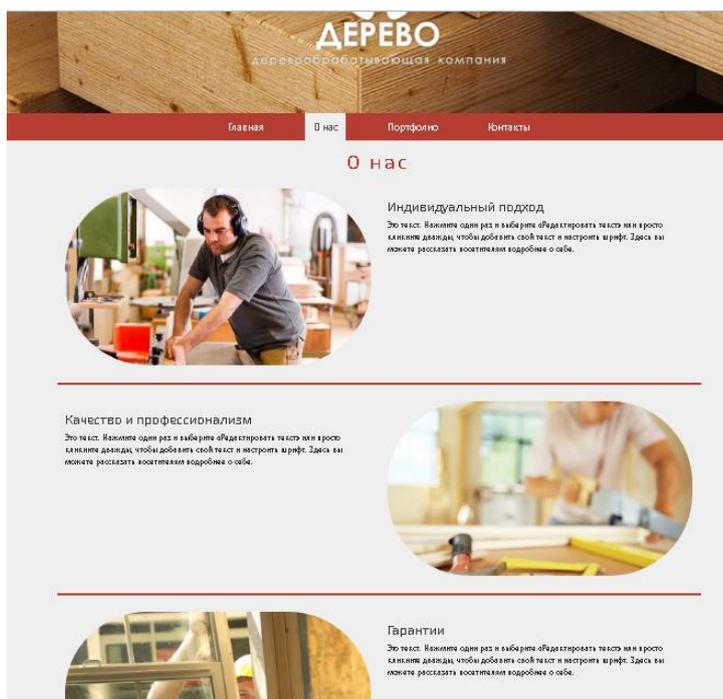


Рисунок 4.2 – Раздел сайта «О нас»

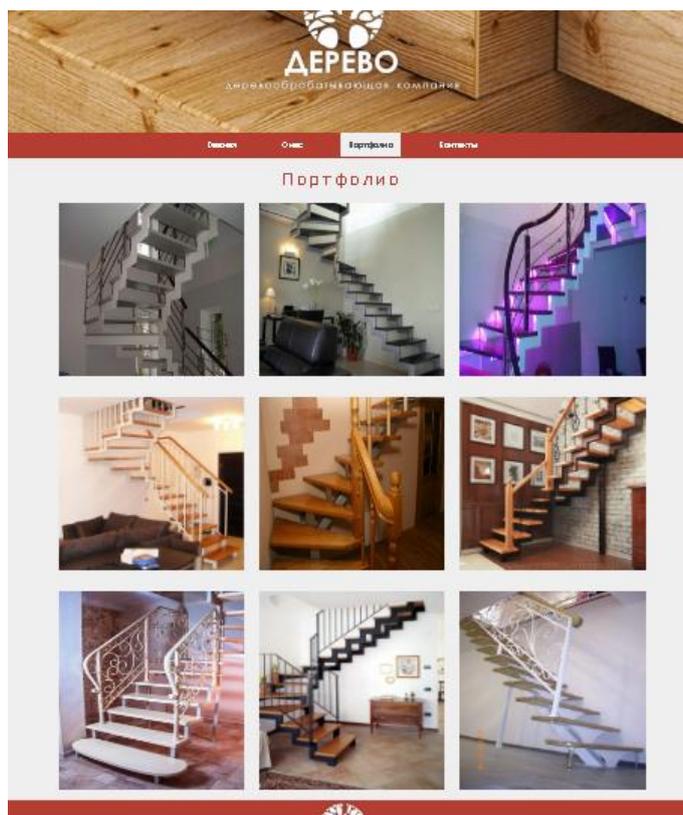


Рисунок 4.3 – Раздел сайта «Портфолио»

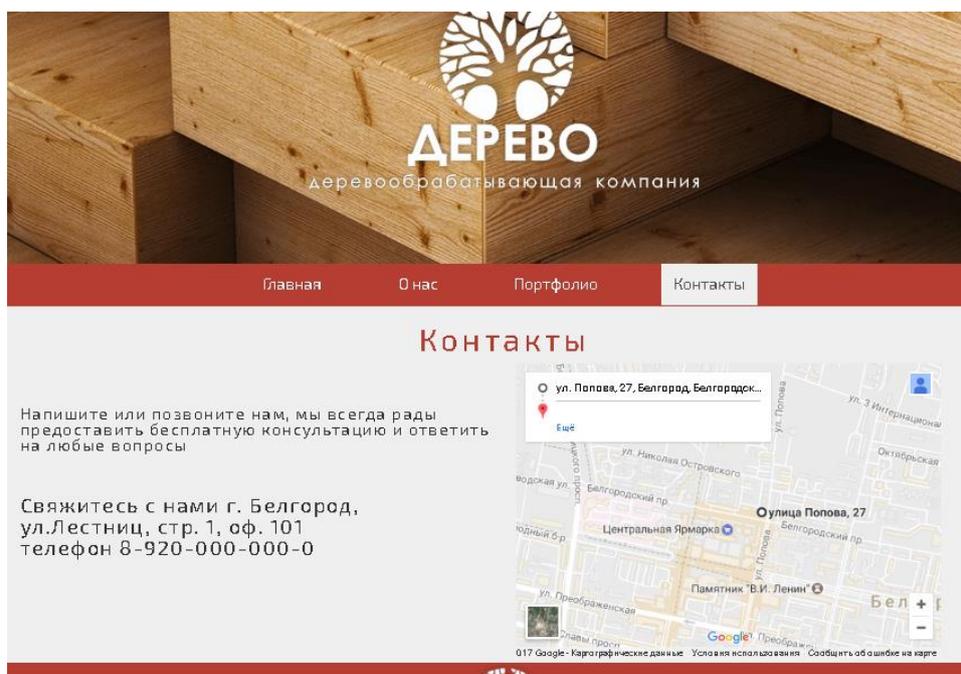


Рисунок 4.4 – Раздел сайта «Контакты»

При нажатии на главной странице сайта на кнопку «Рассчитать стоимость» посетителю откроется страница на которой он может выбрать необходимые для расчета характеристики лестницы (см. Рисунок 4.5).

Рисунок 4.5 – Страница расчета стоимости лестницы

Для расчета стоимости посетителю необходимо выбрать форму лестницы, вид древесины из которой она будет изготовлена, а также цвет и нажать на кнопку «Расчитать цену». После расчета стоимости производства лестницы посетитель может сделать заказ на ее изготовление, нажав кнопку «Заказать». При этом он будет перенаправлен на страницу заказа (см. Рисунок 4.6).

Главная О нас Портфолио Контакты

Оформление заказа

Для окончания оформления заказа заполните, пожалуйста, данную форму
(все поля обязательные для заполнения)

Параметры заказа
Форма лестницы - Г-образная
Материал - Ясень
Цвет - Орех

Количество
1

Имя
Стржев

Отчество
Александр

Фамилия
Викторович

Телефон
89153247631

Адрес
г. Белгород, ул. Советская, 47

Вид доставки
Деловые линии (5000 рублей)

Заказать

Рисунок 4.6 – Страница заказа

Для оформления заказа на изготовление лестницы посетителю необходимо заполнить все поля выводимой на странице сайта формы. После заполнения необходимо нажать кнопку «Заказать».

Если все поля корректно заполнены, посетитель увидит следующее окно (см Рисунок 4.7)

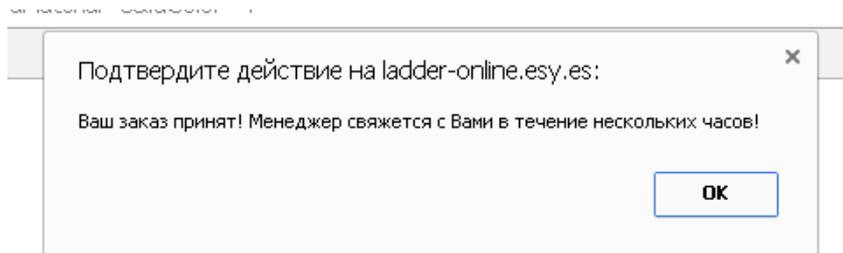


Рисунок 4.7 – Подтверждение приема заказа

Это окно означает, что заказ принят в обработку и в ближайшее время менеджер перезвонит посетителю для подтверждения заказа, а также уточнения необходимых деталей.

Для наиболее удобной и эффективной работы посетителю должно быть все понятно по навигации и по содержанию страниц сайта.

Сайт организации «Дерево» создавался специально в простой форме без различных премудростей, чтобы любой посетитель, будь то программист или человек близко не знакомый с программированием, мог разобраться в навигации и найти необходимую для него информацию на сайте.

8.2 Руководство администратора

Для того чтобы разграничить доступ для сотрудников организации «Дерево» и посетителей сайта была введена авторизация пользователей с доступом к корректировке заказов.

Для того чтобы иметь доступ к просмотру и изменению заказов необходимо в строке браузера ввести адрес <http://ladder-online.esy.es/admin/>.

Перейдя по данному адресу пользователь попадает на страницу авторизации, представленную на Рисунке 4.8.



Рисунок 4.8 – Страница авторизации

Введя в представленных полях имя пользователя – «admin» и пароль – «admin», пользователь попадает на страницу со списком активных заказов (см. Рисунок 4.9)

Личный кабинет: Администратор [Выйти](#)

Список заказов
[Архив заявок](#)

[В архив](#) [Удалить](#)

Поиск по дате (формат ДД.ММ.ГГГГ) [Искать](#)

[Сортировка по возрастанию даты](#)

[Сортировка по убыванию количества](#)

	Дата	Форма	Материал	Цвет	Сумма заказа (с учетом доставки)	Количество	Вид доставки	ФИО	Телефон	Адрес
<input type="checkbox"/>	06.06.2017	Г-образная	Ясень	Орех	от 115000	1	Деловые линии	Викторович Строев Александр	89153247631	г. Белгород, ул. Советская, 47
<input type="checkbox"/>	06.06.2017	П-образная	Лиственница	Орех	от 165000	2	Деловые линии	Тест Тест Тест	89161248520	г. Москва, ул. Победы, 15, кв. 158
<input type="checkbox"/>	05.06.2017	Прямая	Сосна	Венге	от 50000	1	Самовывоз	Стародубцев Александр Семенович	920000000	г. Белгород, ул. Ленина, 26
<input type="checkbox"/>	05.06.2017	Г-образная	Бук	Серый	от 335000	3	Деловые линии	сакратов степан степанович	778899	г. Белгород, ул. Ленина, 26

Рисунок 4.9 – Страница «Список заказов»

На данной странице пользователь может просматривать активные заказы и всю характеризующую их информацию, сведенную в одну таблицу.

Также при нажатии на кнопку «В архив» пользователь может переместить активный заказ в архив, тем самым изменив его статус. При нажатии на ссылку «Архив заявок» пользователь попадает на соответствующую страницу (см Рисунок 4.10).



Рисунок 4.10 – Архив заявок

Пользователь имеет возможность просматривать заказы за определенный день. Для этого необходимо ввести дату в формате дд.мм.гггг в соответствующее поле и нажать на кнопку «Искать» (см Рисунок 4.11). Также предусмотрена сортировка заказов по дате и по убыванию количества (см. Рисунки 4.11, 4.12, 4.13).

Архив заявок

	Дата	Форма	Материал	Цвет	Сумма заказа (с учетом доставки)	Количество	Вид доставки	ФИО	Телефон	Адрес
<input type="checkbox"/>	06.06.2017	Г-образная	Ясень	Орех	от 115000	1	Деловые линии	Викторович Строев Александр	89153247631	г. Белгород, ул. Советская, 47
<input type="checkbox"/>	06.06.2017	П-образная	Лиственница	Орех	от 165000	2	Деловые линии	Тест Тест Тест	89161248520	г. Москва, ул. Победы, 15, кв. 158
<input type="checkbox"/>	05.06.2017	Прямая	Сосна	Венге	от 50000	1	Самовывоз	Стародубцев Александр Семенович	920000000	г. Белгород, ул. Ленина, 26
<input type="checkbox"/>	05.06.2017	Г-образная	Бук	Серый	от 335000	3	Деловые линии	сакратов степан степанович	778899	г. Белгород, ул. Ленина, 26

Рисунок 4.11 – Поиск заказов по дате

Архив заявок

	Дата	Форма	Материал	Цвет	Сумма заказа (с учетом доставки)	Количество	Вид доставки	ФИО	Телефон	Адрес
<input type="checkbox"/>	05.06.2017	Г-образная	Бук	Серый	от 335000	3	Деловые линии	сакратов степан степанович	778899	г. Белгород, ул. Ленина, 26
<input type="checkbox"/>	05.06.2017	Прямая	Сосна	Венге	от 50000	1	Самовывоз	Стародубцев Александр Семенович	920000000	г. Белгород, ул. Ленина, 26
<input type="checkbox"/>	06.06.2017	П-образная	Лиственница	Орех	от 165000	2	Деловые линии	Тест Тест Тест	89161248520	г. Москва, ул. Победы, 15, кв. 158
<input type="checkbox"/>	06.06.2017	Г-образная	Ясень	Орех	от 115000	1	Деловые линии	Викторович Строев Александр	89153247631	г. Белгород, ул. Советская, 47

Рисунок 4.12 – Результат сортировки по возрастанию даты заказа

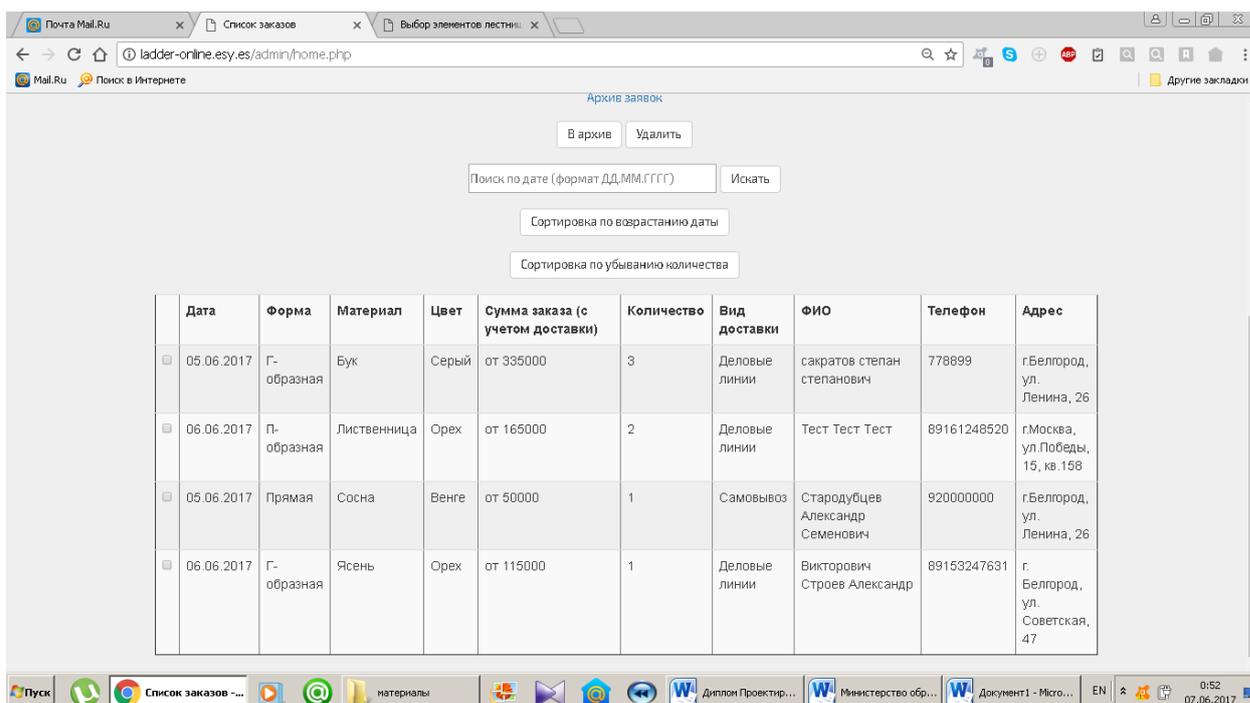


Рисунок 4.13 – Результат сортировки по убыванию количества

Пользователь имеет возможность не только сортировать, искать и изменять статус заказа, но еще и удалять заказы из базы данных. Для этого необходимо нажать кнопку «Удалить». После чего пользователю будет выведено окно для подтверждения удаления заказа (см. Рисунок 4.14).

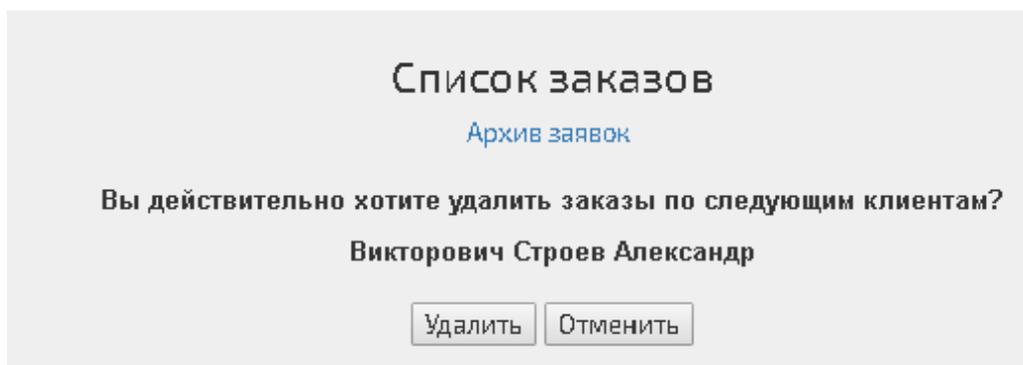


Рисунок 4.14 – Удаление заказа из базы данных

8.3 Руководство разработчика

К настоящему моменту PHP стал одним из самых популярных скриптовых языков, используемых для разработки простых и средних по уровню сложности сайтов. Поддержка этого языка входит в состав услуг большинства виртуальных хостингов в России и других странах. А установка PHP под Windows является несложной задачей. Подобная доступность делает этот язык еще более привлекательным для разработки сайтов и вынуждает хостинговые компании устанавливать и поддерживать PHP.

Разрабатываемый в данном дипломе сайт будет целиком написан на языке программирования PHP с использованием базы данных MySQL.

PHP лучше всего охарактеризовать как работающий на стороне сервера встроенный язык web-сценариев, позволяющий разработчикам быстро и эффективно создавать динамические web-приложения.

За последние годы PHP прошел долгий путь от набора скриптов, позволяющих следить за посетителями страницы, до одного из самых известных и широко распространенных скриптовых языков для создания сложных, динамических, интерактивных web-сайтов.

MySQL – это одна из самых популярных и самых распространенных СУБД (система управления базами данных) в интернете. Она не предназначена для работы с большими объемами информации, но ее применение идеально для интернет-сайтов, как небольших, так и достаточно крупных.

MySQL отличается хорошей скоростью работы, надежностью, гибкостью. Работа с ней, как правило, не вызывает больших трудностей. Поддержка сервера MySQL автоматически включается в поставку PHP.

Взаимодействие с базой данных происходит при помощи Системы Управления Базой Данных (СУБД), которая расшифровывает запросы и производит операции с информацией в базе данных.

Для администрирования и редактирования баз данных MySQL нами

используется программа phpMyAdmin. Она вызывается по команде в адресной строке <https://cpanel.hostinger.com.ua/hosting/index/aid/28627643>

В программе есть класс с именем “DB” для работы с базой данных. В данный момент в программе есть следующие функции:

- добавление клиента
- добавление заказа
- перемещение в архив
- перемещение из архива обратно
- запрос на получение всех данных из любой таблицы
- запрос на получение всех данных из таблицы по определенному условию
- запрос на получении цены лестницы по заданным форме и материалу
- запрос на удаление
- сортировка
- поиск.

Программные коды всех перечисленных выше запросов к СУБД приведены в приложении 3.

ЗАКЛЮЧЕНИЕ

Результатом выполнения дипломного проекта является разработка системы обработки информации по учету заказов для деревообрабатывающего предприятия.

Система позволяет решать следующие задачи:

1. Ввод и хранение информации о товарах деревообрабатывающего предприятия
2. Ввод и хранение информации о клиентах организации
3. Формирование и печать отчетных документов:
 - сортирование списка заказов по определенным характеристикам;
 - поиск по параметрам;
 - список активных заказов;
 - список заказов в архиве
 - расчет цены по характеристикам.

Система разработана в СУБД MySQL и включает в себя БД для учета заказов и пользовательское приложение для работы с ней, реализованное на разработанном в рамках дипломного проекта сайте.

В настоящее время система работает с реальными данными в режиме тестовой эксплуатации.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дейт К. Введение в системы баз данных, 6-е изд. М.: Диалектика, 1998. В. Benz, R. Oliver, «Lotus Notes and Domino 6 Programming Bible», Indianapolis, 2003
2. Кузнецов С.Д. СУБД (системы управления базами данных) и файловые системы. М.: Майор, 2001. – 176 с. – (Мой компьютер).
3. Григорьев Ю.А., Ревунков Г.И. Банки данных: Учеб. Для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 320 с. (Сер. Информатика в техническом университете).
4. Архангельский А.Я. Программирование Access – Москва, 2005
5. Гагарина Л.Г. Разработка и эксплуатация автоматизированных информационных систем: // Учебное пособие. – М.: ФОРУМ; ИНФРА-М, 2007. – 384 с.
6. Гвоздева В.А. Основы построения автоматизированных информационных систем: // Учебник. – М.: ИНФРА-М; ФОРУМ, 2007. – 320 с.
7. Дейт К. Введение в системы баз данных. – К: "Диалектика", 1998.
8. Емельянова Н.З. Основы построения автоматизированных информационных систем: // Учебное пособие. – М.: ФОРУМ; ИНФРА.
9. Кузин А.В. Демин В.М. Разработка баз данных в системе MICROSOFT ACCESS: // Учебник. – М.: ФОРУМ; ИНФРА-М, 2005. – 224 с.
10. Коннолли Т.А., Бег К.В. Базы данных. Проектирование, реализация и сопровождение - Москва, 2001
11. Фаронов В.В. Программирование баз данных в Access – Санкт-Петербург, 2003.
12. Информатика. Базовый курс / Симонович С.В. И другие – Санкт-Петербург: Издательство “Питер”, 2000 г. – 640 с.
13. Web сайт www.azsoft.ru

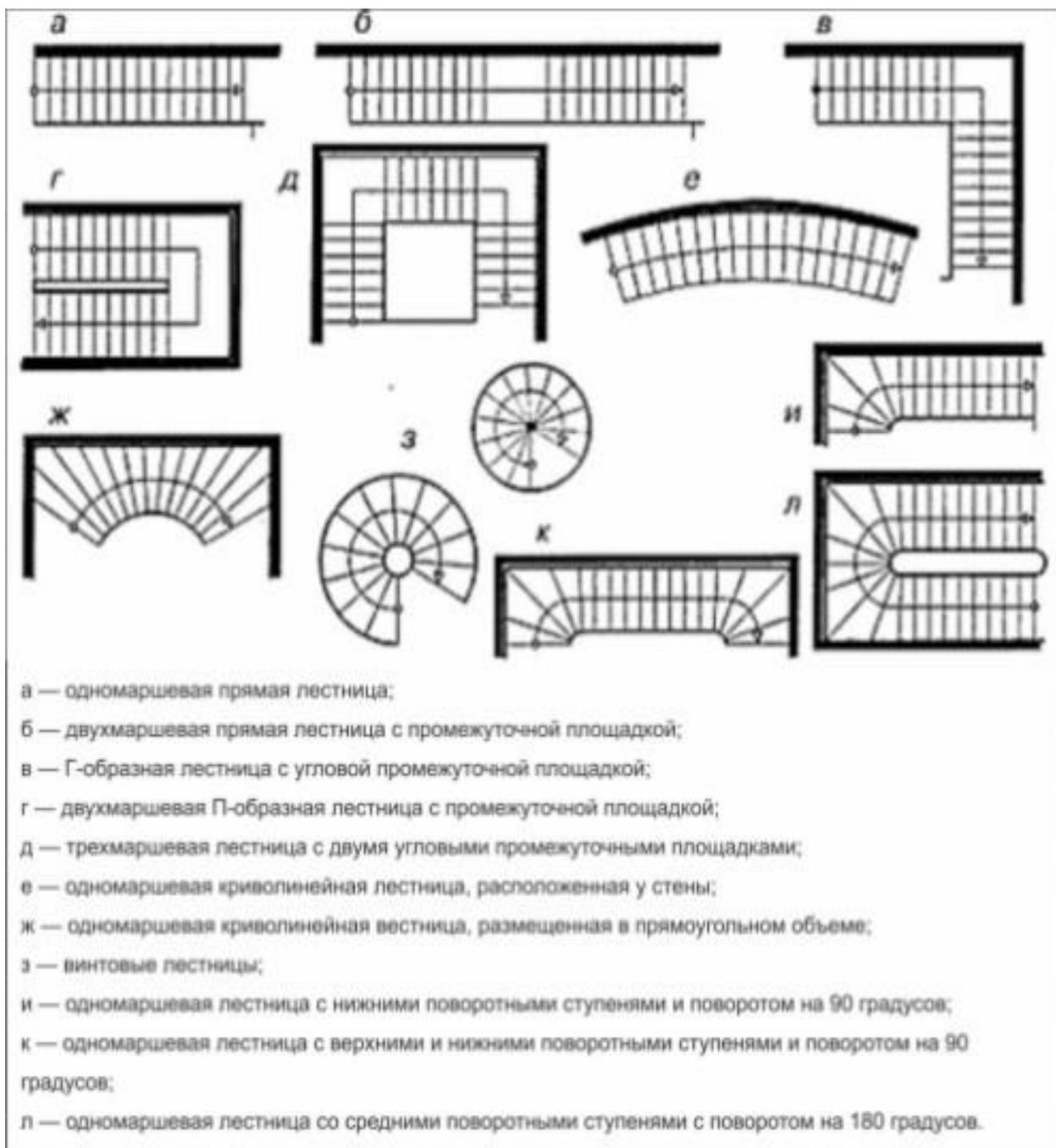
14. Web сайт ru.wikipedia.org
15. Web сайт www.press.unipack.ru

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1



ПРИЛОЖЕНИЕ 2



ПРИЛОЖЕНИЕ 3

Программные коды запросов

1. *Добавление клиента*

```
public function addClient($name, $secondName, $surname, $phone,
$address) {
    $this->db->query("INSERT INTO `clients` (`name`,
`secondName`, `surname`, `phone`, `address`)
    VALUES ('$name', '$secondName', '$surname', '$phone',
'$address')");
}
```

2. *Добавление заказа*

```
public function addOrder($amount, $idClient, $idLadder, $idColor,
$idDelivery) {
    $this->db->query("INSERT INTO `orders` (`amount`, `date`,
`idClient`, `idLadder`, `idColor`, `idDelivery`)
    VALUES ('$amount', '".time()."', '$idClient', '$idLadder',
'$idColor', '$idDelivery')");
}
```

3. *Перемещение в архив, перемещение из архива обратно*

```
public function moveToArchive($id, $isArchive) {
    $this->db->query("UPDATE `orders` SET
`isArchive`='$isArchive' WHERE `id` = '$id'");
}
```

4. Запрос на получение всех данных из любой таблицы

```
public function getAll($table) {
    $result_set = $this->db->query("SELECT * FROM `{$table}`");
    return $result_set;
}
```

5. Запрос на получение всех данных из таблицы по определенному условию

```
public function getRowWhere($table, $columnName, $where) {
    $result_set = $this->db->query("SELECT * FROM `{$table}`
WHERE `{$columnName}`='{$where}'");
    return $result_set;
}
```

6. Запрос на получении цены лестницы по заданным форме и материалу

```
public function getPrice($idForm, $idMaterial) {
    $result_set = $this->db->query("SELECT * FROM `ladders`
WHERE (`idForm`={$idForm} AND `idMaterial`={$idMaterial})");
    return $result_set;
}
```

7. Запрос на удаление

```
public function delete($table, $id) {
    $this->db->query("DELETE FROM `{$table}` WHERE `id` =
'{$id}'");
}
```

8. *Сортировка*

```
public function sortAllOrders($sort, $isArchive) {  
    $result_set = $this->db->query("SELECT * FROM `orders` WHERE  
isArchive = $isArchive ORDER by $sort");  
    return $result_set;  
}
```

9. *Поиск*

```
public function search($time, $isArchive){  
    $result_set = $this->db->query("SELECT * FROM `orders` WHERE  
`date` >= '$time' AND `date` < ('$time' + 86400) AND isArchive = 0");  
    return $result_set;  
}
```

ПРИЛОЖЕНИЕ 4
Инфологическая модель

Таблица 3.1

Объект Лестница

Название	Тип данных	Описание
Код лестницы	INTEGER	Первичный ключ
Код формы	INTEGER	Внешний ключ
Код материала	INTEGER	Внешний ключ
Код цены	INTEGER	Внешний ключ

Таблица 3.2

Объект Форма

Название	Тип данных	Описание
Код формы	INTEGER	Первичный ключ
Наименование	VARCHAR(255)	

Таблица 3.3

Объект Материал

Название	Тип данных	Описание
Код материала	INTEGER	Первичный ключ
Вид древесины	VARCHAR(255)	

Таблица 3.4

Объект Цена

Название	Тип данных	Описание
Код цены	INTEGER	Первичный ключ
Цена	INTEGER	

Таблица 3.5

Объект Цвет

Название	Тип данных	Описание
Код цвета	INTEGER	Первичный ключ
Цвет	VARCHAR(255)	

Таблица 3.6

Объект Клиент

Название	Тип данных	Описание
Код клиента	INTEGER	Первичный ключ
Фамилия	VARCHAR(255)	
Имя	VARCHAR(255)	
Отчество	VARCHAR(255)	
Телефон	VARCHAR(255)	
Адрес	TEXT	

Таблица 3.7

Объект Доставка

Название	Тип данных	Описание
Код доставки	INTEGER	Первичный ключ
Вид доставки	VARCHAR(255)	
Стоимость доставки	INTEGER	

Таблица 3.8

Заказ

Название	Тип данных	Описание
Код заказа	INTEGER	Первичный ключ
Дата	DATE	
Количество	INTEGER	
Код клиента	INTEGER	Внешний ключ
Код лестницы	INTEGER	Внешний ключ
Код цвета	INTEGER	Внешний ключ
Код доставки	INTEGER	Внешний ключ

Таблица 3.9

Связи

Название связи	Объекты связи	Тип связи
Имеет	Лестница – Форма	1 – М
Состоит	Лестница – Материал	1 – М
Стоит	Лестница – Цена	1 – М
Продается	Заказ – Лестница	М – М
Имеет	Заказ – Цвет	1 – М
Покупает	Заказ – Клиент	М – 1
Имеет	Заказ – Доставка	1 – М

ПРИЛОЖЕНИЕ 5

Даталогическая модель данных

Таблица 3.10

Таблица «Лестница»

Отношение	Ключ	Физическая характеристика	Логические ограничения
Лестница			
Код лестницы	PK	Числовой, основной	Notnull;>0
Код формы	FK	Числовой, основной	Not null;>0
Код материала	FK	Числовой, основной	Not null;>0
Код цены	FK	Числовой, основной	Not null;>0

Таблица 3.11

Таблица «Форма»

Отношение	Ключ	Физическая характеристика	Логические ограничения
Форма			
Код формы	PK	Числовой, основной	Notnull;>0
Наименование		Текстовый	Not null

Таблица 3.12

Таблица «Материал»

Отношение	Ключ	Физическая характеристика	Логические ограничения
Материал			
Код материала	PK	Числовой, основной	Notnull;>0
Вид древесины		Текстовый	Not null

Таблица 3.13

Таблица «Цена»

Отношение	Ключ	Физическая характеристика	Логические ограничения
Цена			
Код цены	PK	Числовой, основной	Notnull;>0
Цена		Текстовый	Not null

Таблица 3.14

Таблица «Цвет»

Отношение	Ключ	Физическая характеристика	Логические ограничения
Цвет			
Код цвета	РК	Числовой, основной	Notnull;>0
Цвет		Текстовый	Not null

Таблица 3.15

Таблица «Клиент»

Отношение	Ключ	Физическая характеристика	Логические ограничения
Клиент			
Код клиента	РК	Числовой, основной	Notnull;>0
Фамилия		Текстовый	Not null
Имя		Текстовый	Not null
Отчество		Текстовый	Not null
Телефон		Числовой	Notnull
Адрес		Текстовый	Not null

Таблица 3.16

Таблица «Доставка»

Отношение	Ключ	Физическая характеристика	Логические ограничения
Доставка			
Код доставки	РК	Числовой, основной	Notnull;>0
Вид доставки		Текстовый	Not null
Стоимость доставки		Числовой	Not null

Таблица 3.17

Таблица «Заказ»

Отношение	Ключ	Физическая характеристика	Логические ограничения
Заказ			
Код заказа	PK	Числовой, основной	Not null;>0
Дата		Дата/Время	Not null
Количество		Числовой	Not null
Код клиента	FK	Числовой, основной	Not null;>0
Код цвета	FK	Числовой, основной	Not null;>0
Код доставки	FK	Числовой, основной	Not null;>0

Таблица 3.18

Таблица «Заказано»

Отношение	Ключ	Физическая характеристика	Логические ограничения
Заказано			
Код заказанного	PK	Числовой, основной	Notnull;>0
Код лестницы	FK	Числовой, основной	Not null;>0
Код заказа	FK	Числовой, основной	Not null;>0

ПРИЛОЖЕНИЕ 6

Программный код

archive

?php

```
session_start();

require_once "../lib/checkWasUserLoggedInAndIsAdmin.php";
require_once "../lib/db.php";

$success = 0;

if(!isset($_SESSION["success"])) {
    $success = 0;
}

else $success = $_SESSION["success"];

$db = DB::getObject();
$result_set = $db->getAllOrders(1);

if(isset($_POST["deleteMarkedItems"])) {
    $idItem = $_POST["idItem"];
    if(!isset($idItem))
        $error_delete = "Не было выбран ни один заказ";
    else
        $addConfirmButton = true;
}

else if(isset($_POST["archive"])) {
```

```
$idItem = $_POST["idItem"];
if(!isset($idItem))
    $error_delete = "Не было выбран ни один заказ";
else {
    foreach($idItem as $value) {
        $db->removeFromArchive($value);
    }

    $_SESSION["success"] = 1;

    header("Location: archive.php");
    exit;
}

if(isset($_POST["confirmDelete"])) {
    $idItem = $_POST["idItem"];
    foreach($idItem as $value) {
        $db->delete("orders", $value);
    }

    $_SESSION["success"] = 1;

    header("Location: archive.php");
    exit;
```

```

    }

    if(isset($_POST["cancelDelete"])) {
        $addConfirmButton = false;
    }
?>

<html>
<head>
    <title>Списокзаказов</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="../css/libs.min.css">
    <link rel="stylesheet" type="text/css"
href="https://getbootstrap.com/dist/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="../css/styles.css">
</head>

<body id="content">
    <header>
        <div class="row">
            <div class="admin-header-banner text-center">
                <a href="http://ladder-online.esy.es">
                    
                </a>
            </div>
        </div>
    </header>

```

```

        </div>

</header>

<?php require_once "../partsOfPages/welcome.php"; ?>

<main class="main">

    <div class="container">

        <h3>Списокзаказов</h3>

        <a href="home.php">Назад</a><br /><br />

        <? if($success == 1) {?>
            <p class="success">Данныеуспешносохранены!</p>
            <br />
        <?}?>

        <? if($addConfirmButton) {?>
            <form name="comfirmDelete" action="" method="post">
                <p><b>Вы действительно хотите удалить
заказы по следующим клиентам?</b></p>
                <?
                    $i = 0;
                    foreach($idItem as $value) {
                        $result_set8 = $db-
>getRowWhere("orders", "id", $value);
                        $row8 = $result_set8->fetch_assoc();

```

```

                                $result_set9 = $db-
>getRowWhere("clients", "id", $row8[idClient]);

                                $row9 = $result_set9->fetch_assoc();

                                ?>

                                <input type="hidden"
name="idItem[]" value="<?=$idItem[$i]?>">

                                <? echo "<b>".$row9[surname].
".$row9[name]." ".$row9[secondName]."</b>."<br />";

                                $i++;

                                }

                                ?><br />

                                <input class="standartButton" type="submit"
name="confirmDelete" value="УДАЛИТЬ">

                                <input class="standartButton" type="submit"
name="cancelDelete" value="ОТМЕНИТЬ">

                                </form>

                                <hr><br />

                                <?}?>

                                <form name="" action="" method="post">

                                <span style="color: red";><?=$error_delete ?></span>

                                <input class="longButton lock btn btn-default"
type="submit" name="archive" value="Вернуть из архива">

                                <input class="standartButton de btn btn-default"
type="submit" name="deleteMarkedItems" value="УДАЛИТЬ"><br /><br />

                                <table class="table table-striped" border='1'>

                                <tr>

```

```

        <th></th>
        <th>Дата</th>
        <th>Форма</th>
        <th>Материал</th>
        <th>Цвет</th>
        <th>Цена</th>
        <th>Количество</th>
        <th>Вид доставки</th>
        <th>ФИО</th>
        <th>Телефон</th>
        <th>Адрес</th>
    </tr>
    <? while (($row = $result_set->fetch_assoc()) !=
false) { ?>
        <?
            $result_set1 = $db-
>getRowWhere("ladders", "id", $row[idLadder]);
            $row1 = $result_set1->fetch_assoc();

            $result_set2 = $db-
>getRowWhere("forms", "id", $row1[idForm]);
            $row2 = $result_set2->fetch_assoc();

            $result_set3 = $db-
>getRowWhere("materials", "id", $row1[idMaterial]);
            $row3 = $result_set3->fetch_assoc();

```

```

                                $result_set4 = $db-
>getRowWhere("prices", "id", $row1[idPrice]);

                                $row4 = $result_set4->fetch_assoc();

                                $result_set5 = $db-
>getRowWhere("colors", "id", $row[idColor]);

                                $row5 = $result_set5->fetch_assoc();

                                $result_set6 = $db-
>getRowWhere("delivery", "id", $row[idDelivery]);

                                $row6 = $result_set6->fetch_assoc();

                                $result_set7 = $db-
>getRowWhere("clients", "id", $row[idClient]);

                                $row7 = $result_set7->fetch_assoc();

                                ?>

                                <tr>

                                <td><input type="checkbox"
name="idItem[]" value="<?=$row[id] ?>"></td>

                                <td><?=date('d.m.Y', $row["date"])
?></td>

                                <td><?=$row2[type] ?></td>
                                <td><?=$row3[type] ?></td>
                                <td><?=$row5[color] ?></td>
                                <td><?=$row4[price] ?></td>
                                <td><?=$row[amount] ?></td>

```

```

        <td><?=$row6[type] ?></td>
        <td><?=$row7[surname]
?><?=$row7[name] ?><?=$row7[secondName] ?></td>
        <td><?=$row7[phone] ?></td>
        <td><?=$row7[address] ?></td>
    </tr>
    <?}?>
</table>
</form>
</div>
</main>
</body>
</html>
<?  $_SESSION["success"] = 0; ?>
Home
<?php
    session_start();
    require_once "../lib/checkWasUserLoggedInAndIsAdmin.php";
    require_once "../lib/db.php";

    $success = 0;
    if(!isset($_SESSION["success"])) {
        $success = 0;
    }
    else $success = $_SESSION["success"];

```

```
$db = DB::getObject();

$result_set = $db->getAllOrders(0);

if(isset($_POST["deleteMarkedItems"])) {

    $idItem = $_POST["idItem"];

    if(!isset($idItem))

        $error_delete = "Не был выбран ни один заказ";

    else

        $addConfirmButton = true;

}

else if(isset($_POST["archive"])) {

    $idItem = $_POST["idItem"];

    if(!isset($idItem))

        $error_delete = "Не был выбран ни один заказ";

    else {

        foreach($idItem as $value) {

            $db->moveToArchive($value);

        }

        $_SESSION["success"] = 1;

        header("Location: home.php");

        exit;

    }

}
```

```
if(isset($_POST["confirmDelete"])) {  
    $idItem = $_POST["idItem"];  
    foreach($idItem as $value) {  
        $db->delete("orders", $value);  
    }  
  
    $_SESSION["success"] = 1;  
  
    header("Location: home.php");  
    exit;  
}  
  
if(isset($_POST["cancelDelete"])) {  
    $addConfirmButton = false;  
}  
  
if(isset($_POST["sortByDate"])) {  
    $result_set = $db->sortAllOrders("date", 0);  
}  
  
if(isset($_POST["sortByAmount"])) {  
    $result_set = $db->sortAllOrders("amount DESC", 0);  
}
```

```
if(isset($_POST["searchButton"])) {
    $search = $_POST["search"];

    $error_search = "";
    $error = false;

    if(strlen($search) == 0) {
        $error_search = "Незаполнено поле";
        $error = true;
    }

    if(!$error) {
        header("Location: search.php?search=$search");
        exit;
    }
}

?>

<html>
<head>
    <title>Список заказов</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="../css/libs.min.css">
    <link rel="stylesheet" type="text/css"
href="https://getbootstrap.com/dist/css/bootstrap.min.css">
```

```
<link rel="stylesheet" type="text/css" href="../css/styles.css">
</head>

<body id="content">
  <header>
    <div class="row">
      <div class="admin-header-banner text-center">
        <a href="http://ladder-online.esy.es">
          
        </a>
      </div>
    </div>
  </header>
  <?php require_once "../partsOfPages/welcome.php"; ?>
  <main class="main">
    <div class="container">
      <h3>Списокзаказов</h3>

      <a href="archive.php">Архивзаявок</a><br /><br />

      <? if($success == 1) {?>
        <p class="success">Данныеуспешносохранены!</p>
        <br />
      <?}?>
    </div>
  </main>
</body>
</html>
```

```

<? if($addConfirmButton) {?>
    <form name="comfirmDelete" action="" method="post">
        <p><b>Вы действительно хотите удалить
заказы по следующим клиентам?</b></p>
        <?
            $i = 0;
            foreach($idItem as $value) {
                $result_set8 = $db-
>getRowWhere("orders", "id", $value);
                $row8 = $result_set8->fetch_assoc();

                $result_set9 = $db-
>getRowWhere("clients", "id", $row8[idClient]);
                $row9 = $result_set9->fetch_assoc();
                ?>
                <input type="hidden"
name="idItem[]" value="<?=$idItem[$i]?>">
                <? echo "<b>".$row9[surname].
".$row9[name]." ".$row9[secondName]."</b>".<br />";
                $i++;
            }
        ?><br />
        <input class="standartButton" type="submit"
name="comfirmDelete" value="Удалить">
        <input class="standartButton" type="submit"
name="cancelDelete" value="Отменить">
    </form>
    <hr><br />

```

```
<?}?>
```

```
<form name="" action="" method="post">
    <span style="color: red";><?=$error_delete ?></span>
    <input class="standartButton lock btn btn-default"
type="submit" name="archive" value="Архив">
    <input class="standartButton de btn btn-default"
type="submit" name="deleteMarkedItems" value="Удалить"><br /><br />
```

```
<input class="standartInput searchInput" type="text"
name="search" placeholder="Поиск по дате (формат ДД.ММ.ГГГГ)"
value="<?=$_POST["search"]?>">
```

```
<input class="standartButton btn btn-default"
type="submit" name="searchButton" value="Искать"><br />
```

```
<span style="color: red";><?=$error_search
?></span><br />
```

```
<input class="standartButton btn btn-default"
type="submit" name="sortByDate"
value="Сортировка по возрасту даты"><br /><br />
```

```
<input class="standartButton btn btn-default"
type="submit" name="sortByAmount"
value="Сортировка по убыванию количества"><br /><br />
```

```
<table class="table table-striped" border="1">
```

```
<tr>
```

```
<th></th>
```

```
<th>Дата</th>
```

```
<th>Форма</th>
```

```

        <th>Материал</th>
        <th>Цвет</th>
        <th>Сумма заказа (с учетом
доставки)</th>
        <th>Количество</th>
        <th>Вид доставки</th>
        <th>ФИО</th>
        <th>Телефон</th>
        <th>Адрес</th>
    </tr>
    <? while (($row = $result_set->fetch_assoc()) !=
false) { ?>
        <?
            $result_set1 = $db-
>getRowWhere("ladders", "id", $row[idLadder]);
            $row1 = $result_set1->fetch_assoc();

            $result_set2 = $db-
>getRowWhere("forms", "id", $row1[idForm]);
            $row2 = $result_set2->fetch_assoc();

            $result_set3 = $db-
>getRowWhere("materials", "id", $row1[idMaterial]);
            $row3 = $result_set3->fetch_assoc();

            $result_set4 = $db-
>getRowWhere("prices", "id", $row1[idPrice]);

```

```

$row4 = $result_set4->fetch_assoc();

$result_set5 = $db-
>getRowWhere("colors", "id", $row[idColor]);

$row5 = $result_set5->fetch_assoc();

$result_set6 = $db-
>getRowWhere("delivery", "id", $row[idDelivery]);

$row6 = $result_set6->fetch_assoc();

$result_set7 = $db-
>getRowWhere("clients", "id", $row[idClient]);

$row7 = $result_set7->fetch_assoc();

?>

<tr>

<td><input type="checkbox"
name="idItem[]" value="<?=$row[id] ?>"></td>

<td><?=date('d.m.Y', $row["date"])
?></td>

<td><?=$row2[type] ?></td>

<td><?=$row3[type] ?></td>

<td><?=$row5[color] ?></td>

<td>or<?=( $row4[price]*$row[amount]+$row6[price]) ?></td>

<td><?=$row[amount] ?></td>

<td><?=$row6[type] ?></td>

```

```

                <td><?=$row7[surname]
?><?=$row7[name] ?><?=$row7[secondName] ?></td>
                <td><?=$row7[phone] ?></td>
                <td><?=$row7[address] ?></td>
            </tr>
        <?}?>
    </table>
</form>
</div>
</main>
</body>
</html>
<?  $_SESSION["success"] = 0; ?>
Index
<?php
    session_start();
    require_once "../lib/db.php";

    $user = DB::getObject();
    $auth = $user->isAuth();
    if(isset($_POST["auth"])) {
        $login = $_POST["login"];
        $password = $_POST["password"];
        $auth_success = $user->login($login, $password);
        if($auth_success){
            $_SESSION["isLogin"] = "true";

```

```
$role = $user->getRole($login);
$_SESSION["role"] = $role;

$result_set = $user->getUserByLogin($login);
$row = $result_set->fetch_assoc();
$_SESSION["id"] = $row["id"];

switch($role) {
    case "1": {header("Location: home.php"); break;}
}
exit;
}
}
?>

<html>
<head>
    <title>Авторизация</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="../css/libs.min.css">
    <link rel="stylesheet" type="text/css"
href="https://getbootstrap.com/dist/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="../css/styles.css">
</head>
<body>
```

```

<header>
  <div class="row">
    <div class="admin-header-banner text-center">
      <a href="http://ladder-online.esy.es">
        
      </a>
    </div>
  </div>
</header>
<body id="content" class="fonForBody">
  <div class="authPage">
    <form name="auth" action="" method="post">
      <h3>Вход для зарегистрированных
пользователей</h3>
      <div class="row">
        <div class="form-group col-md-2 col-md-offset-
5">
          <input class="standartInput form-control"
type="text" name="login" placeholder="Имяпользователя">
        </div>
      </div>
      <div class="row">
        <div class="form-group col-md-2 col-md-offset-
5">
          <input class="standartInput form-control"
type="password" name="password" placeholder="Пароль">
        </div>
      </div>
    </form>
  </div>

```

```

        </div>

        <input class="standartButton btn btn-default"
type="submit" name="auth" value="ВОЙТИ">

    </form>

</div>

<div>

    <?php

        if($_POST["auth"]) {

            if(!$auth_success) echo "<h3 id=\"wrongAuth\"
style='color: #8f1f1d; font-weight: bold; text-align: center; margin: 15px 10px 5px
10px'>Неверные имя пользователя и/или пароль</h3>";

            }

        ?>

    </div>

</div>

<script type="text/javascript" src="../js/libs.min.js"></script>

</body>

</html>

ПОИСК

<?php

    session_start();

    require_once "../lib/db.php";

    $user = DB::getObject();

```

```
$auth = $user->isAuth();  
if(isset($_POST["auth"])) {  
    $login = $_POST["login"];  
    $password = $_POST["password"];  
    $auth_success = $user->login($login, $password);  
    if($auth_success){  
        $_SESSION["isLogin"] = "true";  
        $role = $user->getRole($login);  
        $_SESSION["role"] = $role;  
  
        $result_set = $user->getUserByLogin($login);  
        $row = $result_set->fetch_assoc();  
        $_SESSION["id"] = $row["id"];  
  
        switch($role) {  
            case "1": {header("Location: home.php"); break;}  
        }  
        exit;  
    }  
}  
?>  
  
<html>  
<head>  
    <title>Авторизация</title>
```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="../css/libs.min.css">
<link rel="stylesheet" type="text/css"
href="https://getbootstrap.com/dist/css/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="../css/styles.css">
</head>
<body>
<header>
  <div class="row">
    <div class="admin-header-banner text-center">
      <a href="http://ladder-online.esy.es">
        
      </a>
    </div>
  </div>
</header>
<body id="content" class="fonForBody">
  <div class="authPage">
    <form name="auth" action="" method="post">
      <h3>Вход для зарегистрированных
пользователей</h3>
      <div class="row">
        <div class="form-group col-md-2 col-md-offset-
5">
          <input class="standartInput form-control"
type="text" name="login" placeholder="Имяпользователя">

```

```

        </div>
    </div>
    <div class="row">
        <div class="form-group col-md-2 col-md-offset-
5">
            <input class="standartInput form-control"
type="password" name="password" placeholder="Пароль">
        </div>

        </div>

        <input class="standartButton btn btn-default"
type="submit" name="auth" value="ВОЙТИ">
    </form>
</div>

<div>
    <?php
        if($_POST["auth"]) {
            if(!$auth_success) echo "<h3 id=\"wrongAuth\"
style='color: #8f1f1d; font-weight: bold; text-align: center; margin: 15px 10px 5px
10px'>Неверные имя пользователя и/или пароль</h3>";
        }
    ?>
</div>

</div>

<script type="text/javascript" src=" ../js/libs.min.js"></script>
</body>

```

```
</html>
```

```
about_us
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Дерево</title>
```

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

```
<link rel="stylesheet" type="text/css" href="css/libs.min.css">
```

```
<link rel="stylesheet" type="text/css"  
href="https://getbootstrap.com/dist/css/bootstrap.min.css">
```

```
<link rel="stylesheet" type="text/css" href="css/styles.css">
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<div class="row">
```

```
<div class="header-banner text-center">
```

```
<a href="http://ladder-online.esy.es">
```

```

```

```
</a>
```

```
</div>
```

```
</div>
```

```
<nav class="navbar navbar-default text-center">
```

```
<ul class="nav navbar-nav">
```

```
<li><a href="http://ladder-online.esy.es">Главная</a></li>
```

```
<li class="active"><a href="/about_us.php">О нас</a></li>
```

```
<li><a href="/portfolio.php">Портфолио</a></li>
```

```
<li><a href="/contacts.php">Контакты</a></li>
```

```
</ul>
```

```
</nav>
```

```
</header>
```

```
<section id="about">
```

```
<div class="container">
```

```
<div class="row">
```

```
<h1>О нас</h1>
```

```
<div class="row quality">
```

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

```

```

```
</div>
```

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

```
<h3>Индивидуальный подход</h3>
```

```
<span>
```

Это текст. Нажмите один раз и выберите «Редактировать текст» или просто кликните дважды, чтобы добавить свой текст и настроить шрифт. Здесь вы можете рассказать посетителям подробнее о себе.

```
</span>
```

```
</div>
```

```
</div>
```

```
<hr>
```

```
<div class="row quality grey">
```

```
<div class="col-md-6" style="margin-bottom: 20px;">
```

```
<h3>Качество и профессионализм</h3>
```

```
<span>
```

Это текст. Нажмите один раз и выберите «Редактировать текст» или просто кликните дважды, чтобы добавить свой текст и настроить шрифт. Здесь вы можете рассказать посетителям подробнее о себе.

```
</span>
```

```
</div>
```

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

```

```

```
</div>
```

```
</div>
```

```
<hr>
```

```
<div class="row quality">
```

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

```

```

```
</div>
```

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

```
<h3>Гарантии</h3>
```

```
<span>
```

Это текст. Нажмите один раз и выберите «Редактировать текст» или просто кликните дважды, чтобы добавить свой текст и настроить шрифт. Здесь вы можете рассказать посетителям подробнее о себе.

```
</span>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<footer class="text-center">
```

```
<div class="container">
```

```

```

```
<h3>Бесплатная консультация</h3>
```

```
<h3>Звоните: 8-920-000-000-0</h3>
```

```
</div>
```

```
</footer>
```

```
<script type="text/javascript" src="js/libs.min.js"></script>
```

```
</body>
```

```
</html>
```

```
calcAndOrder
```

```
<?php
```

```
    session_start();
```

```
require_once "lib/db.php";

$success = 0;

if(!isset($_SESSION["success"])) {
    $success = 0;
}

else $success = $_SESSION["success"];

$db = DB::getObject();

$result_set = $db->getAll("forms");
$result_set1 = $db->getAll("materials");
$result_set2 = $db->getAll("colors");

if(isset($_POST["calc"])) {
    $idForm = $_POST["idForm"];
    $idMaterial = $_POST["idMaterial"];
    $idColor = $_POST["idColor"];
    // echo "idForm: ".$idForm."<br />";

    $error_idForm = "";
    $error_status = "";
    $error_status = "";
    $error = false;

    if($idForm == "Форма лестницы") {
```

```
        $error_idForm = "Не сделан выбор";
        $error = true;
    }
    if($idMaterial == "Материал") {
        $error_idMaterial = "Несделанвыбор";
        $error = true;
    }
    if($idColor == "Цвет") {
        $error_idColor = "Несделанвыбор";
        $error = true;
    }

    if(!$error) {
        $calculate = true;

        $result_set6 = $db->getPrice($idForm, $idMaterial);
        $row6 = $result_set6->fetch_assoc();

        $result_set6 = $db->getRowWhere("prices", "id",
$row6[idPrice]);
        $row6 = $result_set6->fetch_assoc();
    }
}

if(isset($_POST["makeOrder"])) {
    $idForm = $_POST["idForm"];
```

```
$idMaterial = $_POST["idMaterial"];
$idColor = $_POST["idColor"];

$error_idForm = "";
$error_status = "";
$error_status = "";
$error = false;

if($idForm == "Формалестницы") {
    $error_idForm = "Несделанвыбор";
    $error = true;
}

if($idMaterial == "Материал") {
    $error_idMaterial = "Несделанвыбор";
    $error = true;
}

if($idColor == "Цвет") {
    $error_idColor = "Несделанвыбор";
    $error = true;
}

if(!$error) {
    header("Location:
orderDetail.php?idForm=$idForm&idMaterial=$idMaterial&idColor=$idColor");
    exit;
}
```

```

    }
?>

<html>
<head>
    <title>Выборэлементовлестницы</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <link rel="stylesheet" type="text/css" href="css/libs.min.css">
    <link rel="stylesheet" type="text/css"
href="https://getbootstrap.com/dist/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="css/styles.css">
</head>
<header>
    <div class="row">
        <div class="header-banner text-center">
            <a href="http://ladder-online.esy.es">
                
            </a>
        </div>
    </div>
    <div>
        <nav class="navbar navbar-default text-center">
            <ul class="nav navbar-nav">
                <li><a href="http://ladder-online.esy.es">Главная</a></li>
                <li><a href="/about_us.php">Онас</a></li>
                <li><a href="/portfolio.php">Портфолио</a></li>

```

```

        <li><a href="/contacts.php">Контакты</a></li>
    </ul>
</nav>
</header>
<body>
    <main id="content" class="main">
        <div class="container text-center">
            <h3>Выбор элементов лестницы</h3>
            <p>Для расчета цены и осуществления заказа лестницы
выберите, пожалуйста, форму, материал и цвет лестницы.</p>

            <? if($success == 1) {?>
                <p class="success">Ваш заказ принят! Менеджер
свяжется с Вами в течение нескольких часов!</p>
                <br />
            <?}?>

            <form name="" action="" method="post">
                <div class="row">
                    <div class="form-group col-md-4 col-md-offset-
4">
                        <label
for="idForm">Формалестницы</label>
                        <select class="form-control standartSelect"
name="idForm">
                            <? if($idForm == "" | $idForm ==
"Формалестницы") { ?>

```

```

selected="selected"></option>-->
                                <!--<option
selected="selected"></option>-->
                                <?}
                                else {
                                    $result_set3 = $db-
>getRowWhere("forms", "id", $_POST["idForm"]);
                                    $row3 = $result_set3-
>fetch_assoc(); ?>
                                <option selected="selected"
value="<?=$_POST["idForm"] ?>"><?=$row3["type"] ?></option>
                                <?}??>
                                <? while (($row = $result_set-
>fetch_assoc()) != false) { ?>
                                    <? if($row3["type"] !=
$row[type]) { ?>
                                        <option
value="<?=$row[id] ?>"><?=$row[type] ?></option>
                                        <?}??>
                                        <?}??>
                                    </select><br />
                                    <span style="color: red";><?=$error_idForm
?></span>
                                </div>
                            </div>
                        <div class="row">
                            <div class="form-group col-md-4 col-md-offset-
4">
                                <label for="idMaterial">Материал</label>

```



```

4">
    <div class="form-group col-md-4 col-md-offset-
    <label for="idColor">Цвет</label>
    <select class="form-control standartSelect"
name="idColor">
    <? if($idColor == "" | $idColor ==
"Цвет") { ?>
        <!--<option
selected="selected">Цвет</option>-->
        <?>
        else {
            $result_set5 = $db-
>getRowWhere("colors", "id", $_POST["idColor"]);
            $row5 = $result_set5-
>fetch_assoc(); ?>
            <option selected="selected"
value="<?=$_POST["idColor"] ?>"><?=$row5["color"] ?></option>
            <?>?>
            <? while (($row2 = $result_set2-
>fetch_assoc()) != false) { ?>
                <? if($row5["color"] !=
$row2[color]) {?>
                    <option
value="<?=$row2[id] ?>"><?=$row2[color] ?></option>
                    <?>?>
                <?>?>
            </select><br />
            <span style="color:
red";><?=$error_idColor?></span>

```

```

        </div><br />
    </div>

    <? if($calculate) {?>
        <span
class="success">Цена данной лестницы составляет <?=$row6[price]?> рублей.
</span><br /><br />
        <?}?>

        <input class="standartButton btn btn-default"
type="submit" name="calc" value="Рассчитать цену">

        <input class="standartButton btn btn-default"
type="submit" name="makeOrder" value="Заказать">

    </form>

</div>

</main>

<footer class="text-center">
    <div class="container">
        
        <h3>Бесплатная консультация</h3>
        <h3>Звоните: 8-920-000-000-0</h3>
    </div>

</footer>

<script type="text/javascript" src="js/libs.min.js"></script>

</body>

</html>

```

```
<? $_SESSION["success"] = 0; ?>
```

```
Contacts
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Дерево</title>
```

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

```
<link rel="stylesheet" type="text/css" href="css/libs.min.css">
```

```
<link rel="stylesheet" type="text/css"  
href="https://getbootstrap.com/dist/css/bootstrap.min.css">
```

```
<link rel="stylesheet" type="text/css" href="css/styles.css">
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<div class="row">
```

```
<div class="header-banner text-center">
```

```
<a href="http://ladder-online.esy.es">
```

```

```

```
</a>
```

```
</div>
```

```
</div>
```

```
<nav class="navbar navbar-default text-center">
```

```
<ul class="nav navbar-nav">
```

```
<li><a href="http://ladder-online.esy.es">Главная</a></li>
```

```
<li><a href="/about_us.php">Онас</a></li>
```

```
<li><a href="/portfolio.php">Портфолио</a></li>
```

```
<li class="active"><a href="/contacts.php">Контакты</a></li>
```

```
</ul>
```

```
</nav>
```

```
</header>
```

```
<section id="contacts">
```

```
<div class="container">
```

```
<div class="row">
```

```
<h1>Контакты</h1>
```

```
</div>
```

```
<div class="row">
```

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

```
<h4 id="contact-text">
```

Напишите или позвоните нам, мы

всегда рады предоставить бесплатную консультацию и ответить
на

любые вопросы

```
</h4>
```

```
<h3 id="contact-info">
```

Свяжитесь с нами

г. Белгород, ул.Лестниц,

стр. 1, оф. 101

телефон 8-920-000-000-0

</h3>

</div>

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

<iframe

src="https://www.google.com/maps/embed?pb=!1m23!1m12!1m3!1d5064.824852504646!2d36.58644363492807!3d50.60087351427694!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!4m8!3e6!4m5!1s0x41266af83533c7c5%3A0x9f89df9d5678530a!2z0YYPQuy4g0J_QvtC_0L7QstCwLCAyNywg0JHQtdC70LPQvtGA0L7QtCwg0JHQtdC70LPQvtGA0L7QtNGB0LrQsNGPINC-0LHQuy4sINCg0L7RgdGB0LjRjywgMzA4MDA5!3m2!1d50.60261!2d36.590188!4m0!5e0!3m2!1sru!2sua!4v1496556982052"

width="500" height="350" frameborder="0" style="border:0" allowfullscreen></iframe>

</div>

</div>

</div>

```
</section>
```

```
<footer class="text-center">
```

```
<div class="container">
```

```

```

```
<h3>Бесплатная консультация</h3>
```

```
<h3>Звоните: 8-920-000-000-0</h3>
```

```
</div>
```

```
</footer>
```

```
<script type="text/javascript" src="js/libs.min.js"></script>
```

```
</body>
```

```
</html>
```

```
notRightsEnough
```

```
<?php
```

```

if(isset($_GET["goToLoginPage"])) {
    session_start();
    session_destroy();
    header("Location: admin/index.php");
    exit;
}
?>

<html>
<head>
    <title>Недостаточно прав!</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <link rel="stylesheet" type="text/css" href="css/libs.min.css">
    <link rel="stylesheet" type="text/css"
href="https://getbootstrap.com/dist/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="css/styles.css">
</head>
<header>
    <div class="row">
        <div class="header-banner text-center">
            <a href="http://ladder-online.esy.es">
                
            </a>
        </div>
    </div>
</div>

```

```

<nav class="navbar navbar-default text-center">
    <ul class="nav navbar-nav">
        <li><a href="http://ladder-online.esy.es">Главная</a></li>
        <li><a href="/about_us.php">О нас</a></li>
        <li><a href="/portfolio.php">Портфолио</a></li>
        <li><a href="/contacts.php">Контакты</a></li>
    </ul>
</nav>
</header>
<body>
    <main id="content" class="main mainStartPage" style="min-height:
280px;">
        <div class="container text-center">
            <div class="notRightsEnough" style="margin-top: 80px">
                <p><b>У вас недостаточно прав для просмотра этой
страницы. </b></p>
                <p>Вы можете вернуться на страницу входа и войти
под другими учетными данными.</p>
                <div class="">
                    <a class="download"
href="\notRightsEnough.php?goToLoginPage=1">Страница входа</a>
                </div>
            </div>
        </div>
    </main>
<footer class="text-center">
    <div class="container">

```

```

<h3>Бесплатная консультация</h3>
<h3>Звоните: 8-920-000-000-0</h3>
</div>
</footer>
<script type="text/javascript" src="js/libs.min.js"></script>
</body>
</html>
```

orderDetail

```
<?php
    session_start();
    require_once "lib/db.php";

    $db = DB::getObject();

    if(isset($_GET["idForm"])) {
        $idForm = $_GET["idForm"];
        $idMaterial = $_GET["idMaterial"];
        $idColor = $_GET["idColor"];
    }

    if(isset($_POST["makeOrder"])) {
        $idForm = $_POST["idForm"];
```

```
$idMaterial = $_POST["idMaterial"];
$idColor = $_POST["idColor"];

$amount = $_POST["amount"];
$name = $_POST["name"];
$secondName = $_POST["secondName"];
$surname = $_POST["surname"];
$phone = $_POST["phone"];
$address = $_POST["address"];
$idDelivery = $_POST["idDelivery"];

$error_amount = "";
$error_name = "";
$error_secondName = "";
$error_surname = "";
$error_phone = "";
$error_address = "";
$error_idDelivery = "";
$error = false;

if(strlen($amount) == 0) {
    $error_amount = "Незаполненополе";
    $error = true;
}

if(strlen($name) == 0) {
```

```
        $error_name = "Незаполненополе";
        $error = true;
    }
    if(strlen($secondName) == 0) {
        $error_secondName = "Незаполненополе";
        $error = true;
    }
    if(strlen($surname) == 0) {
        $error_surname = "Незаполненополе";
        $error = true;
    }
    if(strlen($phone) == 0) {
        $error_phone = "Незаполненополе";
        $error = true;
    }
    if(strlen($address) == 0) {
        $error_address = "Незаполненополе";
        $error = true;
    }
    if($idDelivery == "Виддоставки") {
        $error_idDelivery = "Несделанвыбор";
        $error = true;
    }

    if(!$error) {
```

```

$db->addClient($name, $secondName, $surname, $phone,
$address);

$clientId = $db->getLastID("clients");

$result_set = $db->getPrice($idForm, $idMaterial);

$row = $result_set->fetch_assoc();

$db->addOrder($amount, $clientId, $row[id], $idColor,
$idDelivery);

$_SESSION["success"] = 1; ?>

<script language = 'javascript'>
    alert("Ваш заказ принят! Менеджер свяжется с Вами в
течение нескольких часов!");
    document.location.href="calcAndOrder.php";
</script>

<?
}
}

$result_set = $db->getRowWhere("forms", "id", $idForm);
$row = $result_set->fetch_assoc();

$result_set1 = $db->getRowWhere("materials", "id", $idMaterial);
$row1 = $result_set1->fetch_assoc();

```

```
$result_set2 = $db->getRowWhere("colors", "id", $idColor);
```

```
$row2 = $result_set2->fetch_assoc();
```

```
$result_set3 = $db->getAll("delivery");
```

```
?>
```

```
<html>
```

```
<head>
```

```
<title>Оформлениезаказа</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

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

```
<link rel="stylesheet" type="text/css" href="css/libs.min.css">
```

```
<link rel="stylesheet" type="text/css"
href="https://getbootstrap.com/dist/css/bootstrap.min.css">
```

```
<link rel="stylesheet" type="text/css" href="css/styles.css">
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<div class="row">
```

```
<div class="header-banner text-center">
```

```
<a href="http://ladder-online.esy.es">
```

```

```

```
</a>
```

```
</div>
```

```
</div>
```

```
<nav class="navbar navbar-default text-center">
```

```

<ul class="nav navbar-nav">
    <li><a href="http://ladder-
online.esy.es">Главная</a></li>
    <li><a href="/about_us.php">О нас</a></li>
    <li><a href="/portfolio.php">Портфолио</a></li>
    <li><a href="/contacts.php">Контакты</a></li>
</ul>
</nav>
</header>
<main id="content" class="main">
    <div class="container text-center">
        <h3>Оформление заказа</h3>
        <p>Для окончания оформления заказа заполните,
пожалуйста, данную форму <br /> (все поля обязательные для
заполнения)</p>
        <h4>Параметры заказа</h4>
        <b>Форма лестницы -</b><?=$row[type]?<br />
        <b>Материал -</b><?=$row1[type]?<br />
        <b>Цвет -</b><?=$row2[color]?<br /><br />
        <form name="" action="" method="post">
            <div class="row">
                <div class="form-group col-md-4 col-md-offset-
4">
                    <label for="amount">Количество</label>

```

```

        <input id="amount" class="standartInput
form-control" type="number" placeholder="Количество" required
name="amount" min="1" step="1" value="<?=$_POST["amount"]?>">
        <span style="color: red";><?=$error_amount
?></span>
    </div>
</div>
<div class="row">
    <div class="form-group col-md-4 col-md-offset-
4">
        <label for="name">Имя</label>
        <input id="name" class="standartInput form-
control" name="name" type="text" placeholder="Имя"
value="<?=$_POST["name"]?>">
        <span style="color: red";><?=$error_name
?></span>
    </div>
</div>
<div class="row">
    <div class="form-group col-md-4 col-md-offset-
4">
        <label
for="secondName">Отчество</label>
        <input id="secondName"
class="standartInput form-control" name="secondName" type="text"
placeholder="Отчество" value="<?=$_POST["secondName"]?>">
        <span style="color:
red";><?=$error_secondName ?></span>
    </div>

```

```

</div>

<div class="row">
    <div class="form-group col-md-4 col-md-offset-
4">
        <label for="surname">Фамилия</label>
        <input class="standartInput form-control"
name="surname" type="text" placeholder="Фамилия"
value="<?=$_POST["surname"]?>">
        <span style="color:
red";><?=$error_surname ?></span>
    </div>
</div>
<div class="row">
    <div class="form-group col-md-4 col-md-offset-
4">
        <label for="phone">Телефон</label>
        <input id="phone" class="standartInput
form-control" name="phone" type="text" placeholder="Телефон"
value="<?=$_POST["phone"]?>">
        <span style="color: red";><?=$error_phone
?></span>
    </div>
</div>
<div class="row">
    <div class="form-group col-md-4 col-md-offset-
4">
        <label for="address">Адрес</label>

```

```

        <input id="address" class="standartInput
form-control" name="address" type="text" placeholder="Адрес"
value="<?=$_POST["address"]?>">

        <span style="color: red";><?=$error_address
?></span>

        </div>

    </div>

    <div class="row">

        <div class="form-group col-md-4 col-md-offset-
4">

            <label
for="idDelivery">Виддоставки</label>

            <select class="standartSelect form-control"
name="idDelivery">

                <? if($idDelivery == "" | $idDelivery
== "Виддоставки") { ?>

                    <!--<option
selected="selected">Виддоставки</option>-->

                    <?>

                    else {

                        $result_set4 = $db-
>getRowWhere("delivery", "id", $_POST["idDelivery"]);

                        $row4 = $result_set4-
>fetch_assoc(); ?>

                        <option selected="selected"
value="<?=$_POST["idDelivery"] ?>"><?=$row4["type"]?>
(<?=$row4["price"]?>рублей)</option>

                    <?>?>

                    <? while (($row3 = $result_set3-
>fetch_assoc()) != false) { ?>

```

```

$row3[type]) {?>
<? if($row4["type"] !=
value="<?=$row3[id] ?>"><?=$row3[type]?>
(<?=$row3["price"]?>рублей)</option>
<?}?>
<?}?>
</select><br />
<span style="color:
red";><?=$error_idDelivery ?></span>
</div>
</div>
<input type="hidden" name="idForm"
value="<?=$idForm ?>">
<input type="hidden" name="idMaterial"
value="<?=$idMaterial ?>">
<input type="hidden" name="idColor"
value="<?=$idColor ?>">
<input class="standartButton btn btn-default"
type="submit" name="makeOrder" value="Заказать">
</form>
</div>
</main>
<footer class="text-center">
<div class="container">

<h3>Бесплатная консультация</h3>

```

```
<h3>Звоните: 8-920-000-000-0</h3>
```

```
</div>
```

```
</footer>
```

```
<script type="text/javascript" src="js/libs.min.js"></script>
```

```
</body>
```

```
</html>
```

Portfolio

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Дерево</title>
```

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

```
<link rel="stylesheet" type="text/css" href="css/libs.min.css">
```

```
<link rel="stylesheet" type="text/css"
href="https://getbootstrap.com/dist/css/bootstrap.min.css">
```

```
<link rel="stylesheet" type="text/css" href="css/styles.css">
```

```
<link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/ekko-lightbox/5.1.1/ekko-
```

```
lightbox.min.css">
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<div class="row">
```

```
<div class="header-banner text-center">
```

```
<a href="http://ladder-online.esy.es">
```

```

```

```
</a>
```

```
</div>
```

```
</div>
```

```
<nav class="navbar navbar-default text-center">
```

```
<ul class="nav navbar-nav">
```

```
<li><a href="http://ladder-online.esy.es">Главная</a></li>
```

```
<li><a href="/about_us.php">Онас</a></li>
```

```
<li class="active"><a href="/portfolio.php">Портфолио</a></li>
```

```
<li><a href="/contacts.php">Контакты</a></li>
```

```
</ul>
```

```
</nav>
```

```
</header>
```

```
<section id="portfolio">
```

```
<div class="container">
```

```
<div class="row">
```

```
<h1>Портфолио</h1>
```

```
<div class="row">
```

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

```

```

```
</div>
```

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

```

```

```
</div>
```

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

```

```

```
</div>
```

```
</div>
```

```
<div class="row">
```

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

```

```

```
</div>
```

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

```

```

```
</div>
```

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

```

```

```
</div>
```

```
</div>
```

```
<div class="row">
```

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

```

```

```
</div>
```

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

```

```

```
</div>
```

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

```

```

```
</div>
```

```
</section>
```

```
<footer class="text-center">
```

```
<div class="container">
```

```

```

```
<h3>Бесплатная консультация</h3>
```

```
<h3>Звоните: 8-920-000-000-0</h3>
```

```
</div>
```

```
</footer>
```

```
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/ekko-lightbox/5.1.1/ekko-lightbox.min.js"></script>
```

```
</body>
```

```
</html>
```

