ГБОУ ВПО МО «Финансово-технологическая академия»

«Колледж космического машиностроения и технологий»

Курсовой проект

На тему: «Разработка интернет магазина «Электроники»»

Выполнил: Проверил:

Абдуллаев М.С. Гусятинер Л.Б.

Группа: К46АСУ

Подпись: Подпись:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Королёв. 2014.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc386064133)

[1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ 6](#_Toc386064134)

[2. АНАЛИТИЧЕСКИЕ СВЕДЕНИЯ 11](#_Toc386064135)

[2.1. Бизнес-процесс AS-IS 12](#_Toc386064137)

[2.3. Анализ имеющихся программ 12](#_Toc386064138)

[3. СВЕДЕНИЯ О ПРОЕКТЕ 13](#_Toc386064139)

[3.1. Выбор системы 13](#_Toc386064140)

[3.2. Установка системы в помещениях 13](#_Toc386064141)

[3.3. Бизнес-процесс TO-BE 13](#_Toc386064142)

[3.4. Программное обеспечение 14](#_Toc386064143)

[*3.4.1.* *Сценарий работы* 14](#_Toc386064144)

[*3.4.2.* *Структура БД* 15](#_Toc386064145)

[*3.4.3.* *Структура клиентской части* 16](#_Toc386064146)

[*3.4.4.* *Руководство пользователя* 16](#_Toc386064147)

[3.5. Техническое обеспечение 32](#_Toc386064148)

[3.6. Обязанности организации для защиты информации 33](#_Toc386064149)

[4. ЭКОНОМИЧЕСКИЕ СВЕДЕНИЯ 34](#_Toc386064150)

[4.1. Расчёт стоимости одного часа работы ПЭВМ 34](#_Toc386064151)

[*4.1.1.* *Расчет заработной платы обслуживающего персонала* 34](#_Toc386064152)

[*4.1.2.* *Расчет взносов социальное страхование и обеспечение* 35](#_Toc386064153)

[*4.1.3.* *Расчет накладных расходов* 36](#_Toc386064154)

[*4.1.4.* *Расчет амортизационных отчислений* 36](#_Toc386064155)

[*4.1.5.* *Расчет затрат на ремонт оборудования* 37](#_Toc386064156)

[*4.1.6.* *Расчет затрат на электроэнергию* 37](#_Toc386064157)

[*4.1.7.* *Расчет затрат на материалы* 38](#_Toc386064158)

[*4.1.8.* *Расчет стоимости одного часа работы ПЭВМ* 39](#_Toc386064159)

[4.2. Расчет затрат на составление и отладку программы 40](#_Toc386064160)

[*4.2.1.* *Расчет стоимости решения задачи на ЭВМ* 40](#_Toc386064161)

[*4.2.2.* *Расчет заработной платы руководителя* 40](#_Toc386064162)

[*4.2.3.* *Расчет заработной платы студента* 41](#_Toc386064163)

[*4.2.4.* *Расчет взносов на социальное страхование и обеспечение* 41](#_Toc386064164)

[*4.2.5.* *Расчет накладных расходов* 41](#_Toc386064165)

[*4.2.6.* *Расчет затрат на материалы, используемые при составлении и отладке программы* 42](#_Toc386064166)

[*4.2.7.* *Расчет стоимости решения задачи вручную* 42](#_Toc386064167)

[4.3. Расчет экономической эффективности 43](#_Toc386064168)

[*4.3.1.* *Расчет экономии труда за счет применения ПЭВМ* 43](#_Toc386064169)

[*4.3.2.* *Расчет затрат времени на составление и отладку программы* 43](#_Toc386064170)

[*4.3.3.* *Расчет времени решения задачи на ПЭВМ* 43](#_Toc386064171)

[*4.3.4.* *Расчет предполагаемого числа обращений к программе* 44](#_Toc386064172)

[*4.3.5.* *Расчет денежной экономической эффективности применения ПЭВМ* 44](#_Toc386064173)

[*4.3.6.* *Расчет затрат на работу программы* 44](#_Toc386064174)

[*4.3.7.* *Расчет роста производительности труда от применения ПЭВМ* 44](#_Toc386064175)

[5. ЗАКЛЮЧЕНИЕ 45](#_Toc386064176)

[6. СПИСОК ЛИТЕРАТУРЫ 46](#_Toc386064177)

[ПРИЛОЖЕНИЯ 47](#_Toc386064178)

[Приложения А. Скрипт на создание БД 47](#_Toc386064179)

[Приложения Б. Модуль клиентской части 59](#_Toc386064180)

# ВВЕДЕНИЕ

В настоящее время Интернет становиться все более развитой средой для осуществления коммуникаций с потребителями. В тоже время, существенным является и тот факт, что Интернет становиться удобной и достаточно дешевой «торговой площадкой». Все большее количество фирм старается представить свою продукцию в on-line среде. При этом такое представление не ограничивается только лишь созданием промо-сайтов и размещением рекламных баннеров и статей в электронных журналах и на информационных порталах. С развитием Интернет-среды развивается и само предложение. Теперь люди могут не только получать интересующую их информацию, но и совершать покупки. При этом с помощью Интернет-магазинов можно приобретать товары совершенно разных категорий, как элементарные потребительские, так и высокотехнологичные.

Такое положение вещей обусловлено, во-первых, постоянным, стабильным ростом аудитории пользователей глобальной сети.

Основную часть аудитории сети Интернет составляют люди, работающие в офисах и проводящие большую часть времени за компьютером. Как правило, их образ жизни не позволяет им тратить большое количество времени на походы по off-line магазинам в поисках именно того, что им нужно. Возможность заказать интересующий продукт в Интернете является для них действительно выходом из ситуации.

Выбранная мною тема считается актуальной на сегодняшний день, так как сегодня миллионы людей ежедневно, не выходя из дому, покупают различные товары в электронных магазинах. В мире, а в частности России огромными темпами растет количество пользователей internet и как следствие количество «электронных» покупателей, потенциальных «электронных» покупателей.

Электронные магазины существенно уменьшают издержки производителя, сэкономив на содержании обычного магазина, расширяют рынки сбыта, так же как и расширяет возможность покупателя - покупать любой товар в любое время в любой стране. Это дает электронным магазинам неоспариваемое преимущество перед обычными магазинами. Этот момент является существенным при переходе производителей с «обычной» торговли на «электронную».

# ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

**Web-сайт** – это информация, представленная в определенном виде, которая располагается на Web-сервере и имеет свое имя (адрес). Для просмотра Web-сайтов на компьютере пользователя используются специальные программы, которые называются браузерами. В зависимости от того, какое имя (адрес) сайта мы зададим в строке "Адрес", браузер будет загружать в свое окно соответствующую информацию.

Web-сайт состоит из связанных между собой Web-страниц. Web-страница представляет собой текстовый файл с расширением \*.htm, который содержит текстовую информацию и специальные команды – HTML-коды, определяющие в каком виде эта информация будет отображаться в окне браузера. Вся графическая, аудио- и видео-информация непосредственно в Web-страницу не входит и представляет собой отдельные файлы с расширениями \*.gif, \*.jpg (графика), \*.mid, \*.mp3 (звук), \*.avi (видео). В HTML-коде страницы содержатся только указания на такие файлы.

Каждая страница Web-сайта также имеет свой Internet адрес, который состоит из адреса сайта и имени файла, соответствующего данной странице. Таким образом, Web-сайт – это информационный ресурс, состоящий из связанных между собой гипертекстовых документов (Web-страниц), размещенный на Web-сервере и имеющий индивидуальный адрес. Посмотреть Web-сайт может любой человек, имеющий компьютер, подключенный к Internet.

**Web-сервер** - программа, запущенная на узле сети Интернет и выдающая посетителям этого узла web-страницы по запросам. Также web-сервером часто называется узел, на котором эта программа запущена, или даже компьютер, являющийся таким узлом.

**Сервер** *-* любой отдельно взятый компьютер в Интернете, который позволяет другим машинам, грубо говоря, использовать себя в качестве «посредника» при передаче данных.

Сервер - это именно машина («железо»), а не логическая часть Сети, он может иметь несколько различных IP-адресов (не говоря уже о доменных именах), так что вполне может выглядеть из Интернета как несколько независимых систем. Отличительной чертой сервера является то, что он использует один единственный стек TCP/IP, т.е. на нем запущено только по одному «экземпляру» драйверов протоколов.

У термина «сервер» есть и еще другое, определение - это программа (в терминологии, TCP - процесс), обрабатывающая запросы клиентов. Например, приложение, обслуживающее пользователей WWW, называется Web-сервером.

# Классификация Web-сайтов

По используемым технологиям:

* Статические сайты. Изначально все сайты были статическими. Они созданы по классической технологии HTML. Статические сайты представляют собой набор HTML-страниц, связанных друг с другом гиперссылками и загруженных на web-сервер. После набора адреса, соответствующего статическому сайту, пользователь видит в браузере одно и то же содержание, так как страницы этих сайтов располагаются на web-сервере в неизменном виде. Страницы таких сайтов пользователь может скачать себе на компьютер и просматривать локально. Когда необходимо изменить содержимое страниц статических сайтов, автор при помощи специальной программы (HTML-редактор) или обычного текстового редактора модифицирует станицы на своем рабочем компьютере и заново выкладывает на web-сервер.
* Динамические сайты. Состоят из скриптов, шаблонов, информации и т.п. объектов в виде отдельных файлов. Содержимое такого сайта изменяется средствами самого сайта без применения пользователем дополнительного программирования. Страница динамического сайта с одним и тем же адресом может выглядеть по-разному и в зависимости от

введенного запроса, прав доступа пользователя и других условий. Это обусловлено тем, что в динамических сайтах при запросе информация извлекается из базы данных, вставляется в шаблон, формируя новую web-страницу, которая отображается в браузере пользователя.

* Флеш-сайты. Технология флэш позволяет создавать эффектные интерактивные сайты со звуком и анимацией. Флэш сайты в общем понимании представляют собой совокупность сменяющих друг друга кадров в определенный промежуток времени.

Основной задачей флэш-сайта является произвести сильное впечатление на пользователя, поразить яркой «живой» анимацией, привлечь максимальной интерактивностью, возможностью менять элементы оформления. Один из серьезных недостатков флэш-сайта – это долгая загрузка из-за значительного «веса».

По видам доступа:

* Открытые. Все сервисы сайта полностью доступны для всех посетителей. Чтобы воспользоваться всеми возможностями сайта достаточно просто на него зайти.
* Полуоткрытые. Для доступа к сервисам необходимо пройти процедуру регистрации на сайте. Процесс регистрации на разных сайтах может отличаться. Как правило, он заключается в заполнении анкеты с информационными полями: обязательными, такими как «имя пользователя», «пароль», «электронный адрес», и необязательными, такими как «пол», «возраст», «город» и т.п. Далее предлагается ознакомиться с пользовательским соглашением. Последним шагом регистрации обычно является ее подтверждение по средствам кода или через указанный электронный адрес. Код подтверждения служит защитным механизмом с целью определения, робот регистрируется или человек. Теперь при входе на сайт необходимо ввести логин и пароль, чтобы получить доступ к дополнительным возможностям (скачать файлы, разместить комментарии, прочесть скрытый текст и т.д.). На некоторых сайтах регистрация платная.
* Закрытые. Сайты с закрытым доступом. Это могут быть служебные корпоративные сайты, личные сайты частных лиц или группы лиц. Такие сайты доступны для узкого круга пользователей. Доступ новым пользователям обычно предоставляется через специальные приглашения – инвайты. Инвайт – это код доступа (набор символов) или ссылка, содержащая в себе код, для регистрации в закрытом сервисе, форуме или партнерской программе. Приглашение обычно можно получить у уже зарегистрированного участника, либо у службы поддержки, либо выполнив определенные условия, установленные администрацией закрытого интернет-проекта. Инвайт используют для ограничения доступа к сервису сайта от посторонних пользователей. Закрытыми также могут быть проекты на стадии начального тестирования их сервисов.

По типам представляемых сервисов:

* Сайт-визитка. Небольшого объема сайт, представляющий собой визитную карточку фирмы либо частного лица в сети интернет. Такие сайты кратко рассказывают о товарах или услугах, предоставляемых организацией или индивидуальным предпринимателем. Как правило, на сайте-визитке содержится информация о виде деятельности, контактные данные, прайс-лист, схема проезда, форма обратной связи. Чаще всего сайт-визитка делается на основе шаблонного дизайна.
* Интернет магазин. Интерактивный сайт с каталогом продукции, рекламирующий товары или услуги. С его помощью можно оформить заказ на покупку, получить счет на оплату, служащий подтверждением заказа, выбрать способ оплаты и доставки товара. В каталоге имеется подробное описание товаров или услуг, представленных в интернет-магазине, сертификаты качества, лицензии, отзывы экспертов и потребителей. Создаются такие сайты с использованием систем управления контентом.
* Поисковая система. Программно-аппаратный комплекс, оказывающий помощь пользователю в поиске информации в сети интернет. Дает возможность искать по изображениям, видео, картам, новостям, блогам, словарям и т.д. Поисковая система базируется на поисковом движке – комплексе программ, реализующих функциональности поисковой системы и являющихся коммерческой тайной фирмы-разработчика. Поисковые системы состоят из агента, который индексирует информацию, перемещаясь по глобальной сети, базы данных и поискового механизма.
* Почтовый сервис. Предоставление услуг по отправке и получению электронных сообщений между пользователями сети интернет. Пересылка сообщений организуется специальными почтовыми серверами. При регистрации в почтовом сервисе пользователь получает электронный адрес и ему выделяется место на сервере для хранения писем. К электронным сообщениям, кроме информации, пользователь может добавлять документы, фотографии, видео, музыку и другие файлы. Существуют как платные, так и бесплатные почтовые сервисы. Отличаются они скорее не качеством и удобством предоставляемых услуг, а уникальностью и престижем почтового адреса.
* Блог. Также называемый интернет-журнал, онлайн-дневник. Представляет собой регулярно добавляемые не слишком большие статьи, которые могут, кроме текста, содержать изображения, аудио и видео файлы. Обычно блог ведется одним пользователем (личный блог), но встречаются также групповые блоги (социальные, корпоративные, клубные). Для блогов свойственна возможность добавления другими пользователями комментариев к уже опубликованным статьям, создавая, таким образом, среду сетевого общения. Блоги могут быть тематическими либо общего содержания. Блоги располагаются на блог-платформе – специальном сервисе, позволяющем пользователю вести свой интернет-дневник, не углубляясь при его создании в программирование.
* Социальная сеть. Многопользовательский интерактивный сайт, содержимое которого в основном формируется и наполняется самими участниками. Социальные сети предназначены главным образом для общения, поиска людей. Участники социальной сети объединяются в сообщества по интересам. Общение между пользователями сети осуществляется посредством внутренней почты, обмена мгновенных сообщений. Для регистрации в социальной сети пользователь создает свой аккаунт (учетную запись, необходимую для идентификации участника) и заполняет анкету, в которой указывает сведения о себе. Регистрация может быть платной и бесплатной. Сети бывают открытые и закрытые.
* Интернет-портал. Крупный информационный ресурс, объединяющий различные сервисы. Портал предоставляет широкий спектр информации по определенной тематике. В рамках интернет-портала пользователи могут общаться друг с другом посредством форума, чата, мгновенных сообщений и других средств взаимодействия. Также портал может содержать развлекательный контент.

Данный курсовой проект, является полуоткрытым, динамическим, интернет магазином.

# АНАЛИТИЧЕСКИЕ СВЕДЕНИЯ

## Бизнес-процесс AS-IS

**Бизнес процесс покупки товара**

Рисунок 2 - Схема бизнес-процесса AS-IS часть 1

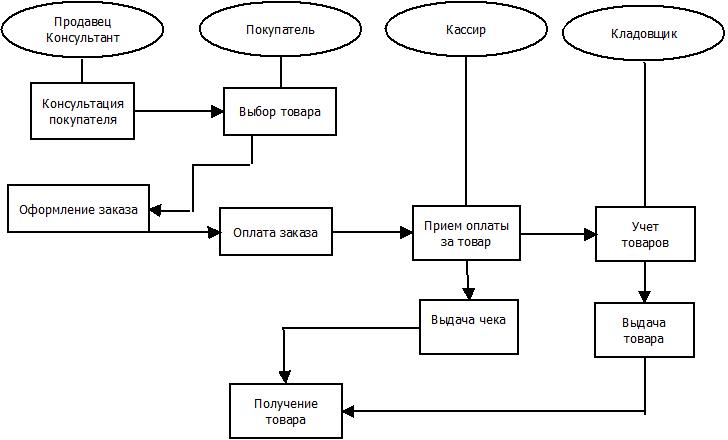


Рисунок 3 - Схема бизнес-процесса AS-IS часть 2

## Рисунок 1 – Бизнес-процесс AS-IS покупки товара

## Анализ имеющихся программ

Просмотрев большинство проектов, были выявлены некоторые недочёты, а именно:

* Отсутствие мобильной версии;
* Отсутствие модуля «Так же покупают»;
* Отсутствие списка популярных товаров;
* Отсутствие возможности редактирования меню;
* Отсутствие исходного кода;

Проанализировав все эти недочёты, было принято решения устранить их в данном курсовом проекте.

# СВЕДЕНИЯ О ПРОЕКТЕ

## Выбор системы

Было принято решения создать новую систему, так как функциональные возможности готовых систем не соответствует поставленным задачам.

В готовых системах не были реализованы следующие функции:

* Отсутствие мобильной версии;
* Отсутствие возможности редактирования меню;
* Отсутствие исходного кода;

## Установка системы в помещениях

Для установки системы нужен один компьютер, выступающий в виде сервера, который может располагаться в любом кабинете или в серверной.

## Бизнес-процесс TO-BE

**Бизнес процесс покупки товара**

Рисунок 5 - Схема бизнес-процесса TO-BE часть 1

Рисунок 6 - Схема бизнес-процесса TO-BE часть 2

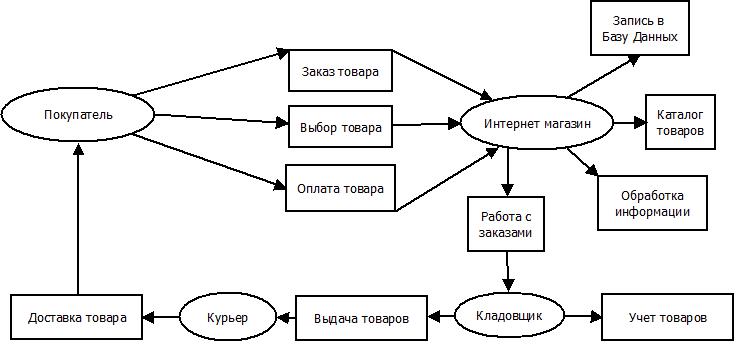


Рисунок 7 - Схема бизнес-процесса TO-BE часть 3

## Рисунок 2 – Бизнес-процесс TO-BE покупки товара

## Программное обеспечение

### *Сценарий работы*

Сценарий работы программы при входе от имени, анонимного пользователя выглядит следующим образом:

* Пользователь может просматривать товары
* Оставлять отзывы к товарам
* Поиск по товарам
* Зарегистрироваться
* Обратная связь с администрацией

Сценарий работы программы при входе от имени, зарегистрированного пользователя выглядит следующим образом:

* Пользователь может просматривать товары
* Оставлять отзывы к товарам
* Поиск по товарам
* Заказывать товары
* Обратная связь с администрацией
* Просматривать список своих заказов
* Сменить пароль

Сценарий работы программы при входе от имени администратора выглядит следующим образом:

* Пользователь может просматривать товары
* Оставлять отзывы к товарам
* Поиск по товарам
* Заказывать товары
* Сменить пароль
* Добавлять категории
* Редактировать категории
* Добавлять подкатегории
* Редактировать подкатегории
* Добавлять товары
* Редактировать товары
* Просматривать список всех заказов
* Просматривать и редактировать отзывы ко всем товарам
* Выводить список всех заказов в Word, Excel

### *Структура БД*

### *C:\Users\Мурад\Desktop\схема БД.png*

Рисунок 3 - Схема базы данных

### *Структура клиентской части*

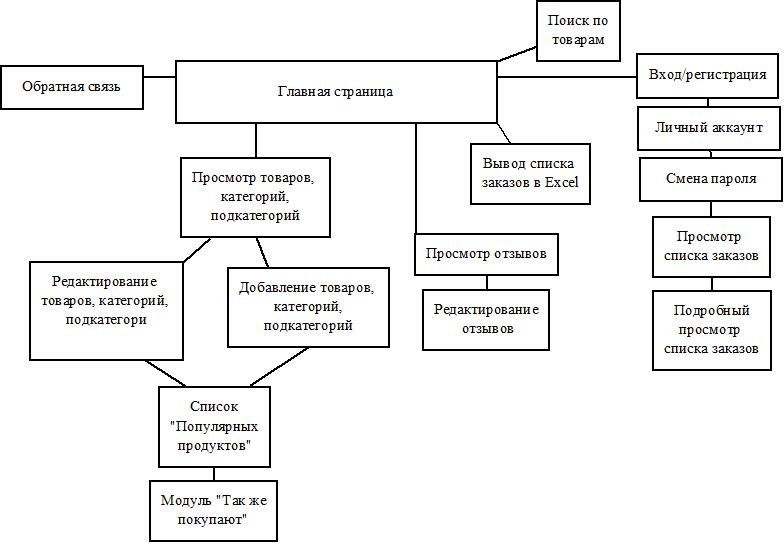


Рисунок 9 - Структурная схема работы клиентской части программы

### *Руководство пользователя*

*Руководство пользователя клиентской части программы:*

1. **Первый запуск**
2. Вход от имени зарегистрированного пользователя

У пользователя есть свой личный кабинет, попасть в него, он может, кликнув ЛКМ на кнопку аккаунт (см. рис. 4).

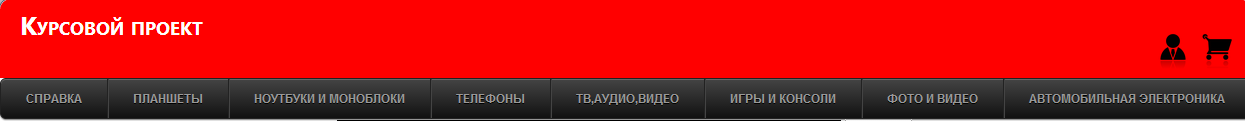


Рисунок 4 – Личный кабинет

В личном кабинете, у пользователя есть возможность просмотреть список заказов, сменить пароль, а так же выйти из системы (см. рис. 5).

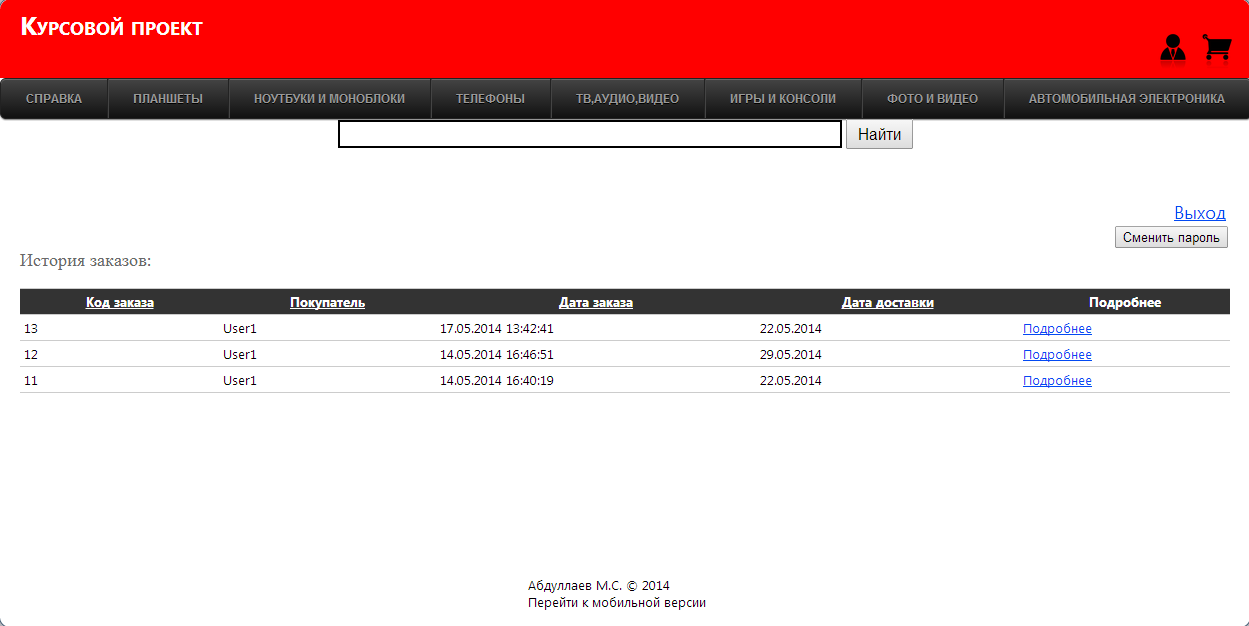


Рисунок 5 – Личный кабинет

Он может просматривать и заказывать товары, а так же оставлять отзывы к ним (см. рис. 6, 7, 8 ,9).

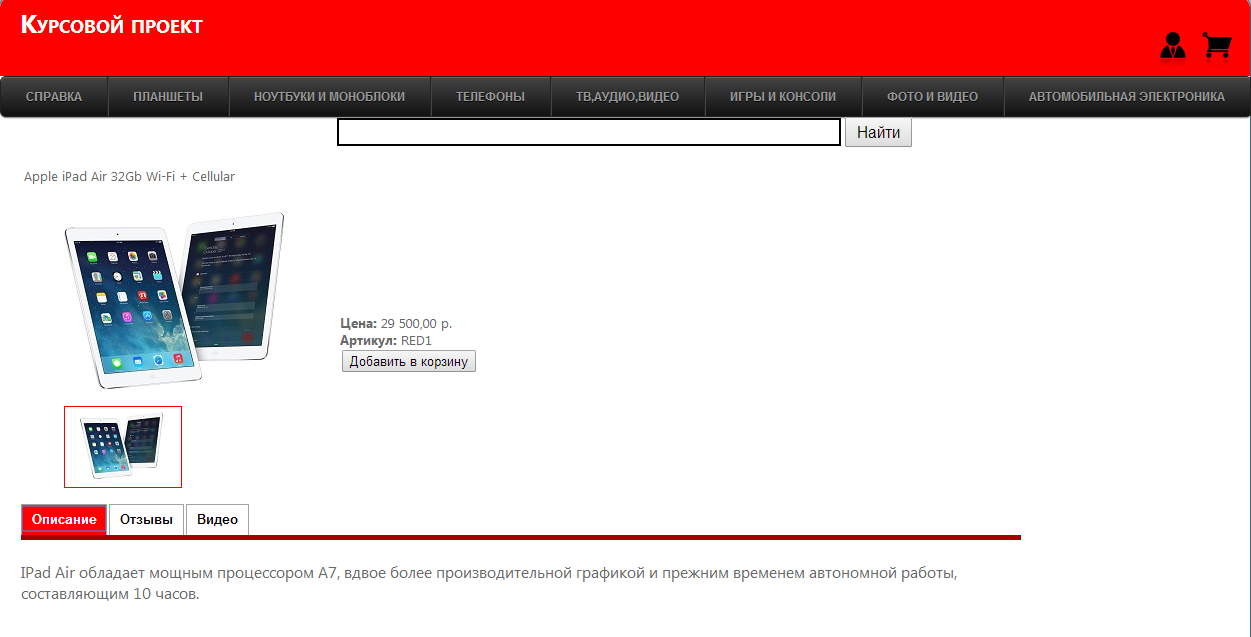
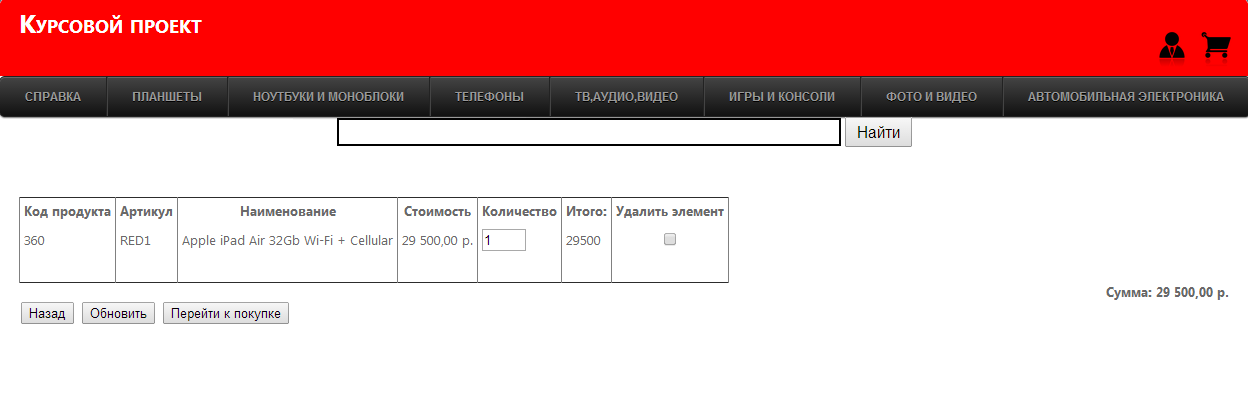
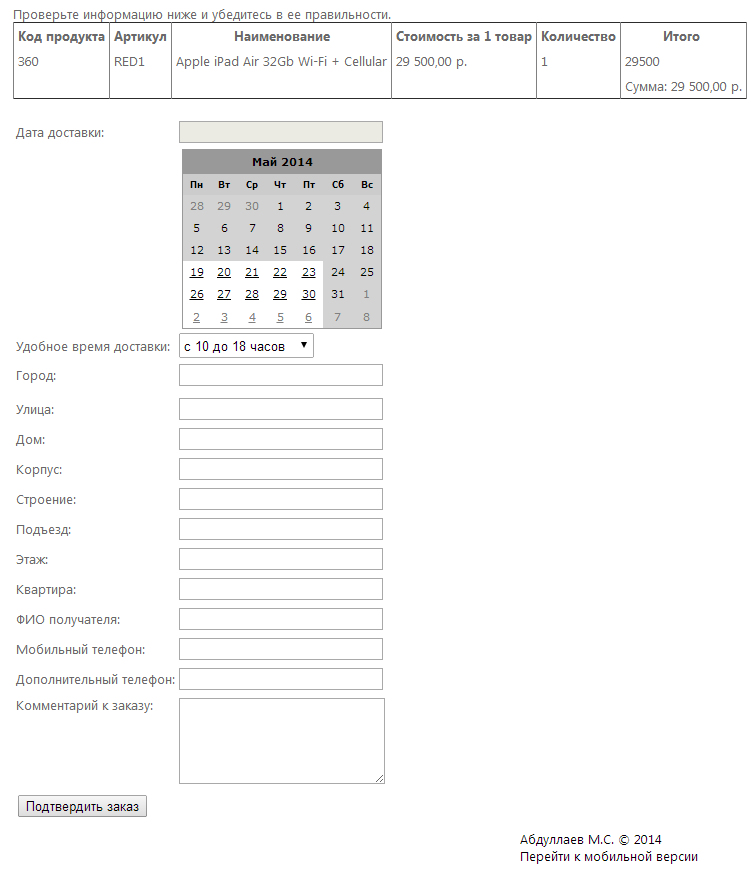


Рисунок 6 – Просмотр товаров

 Рисунок 7 – Корзина

 Рисунок 8 – Оформление заказа

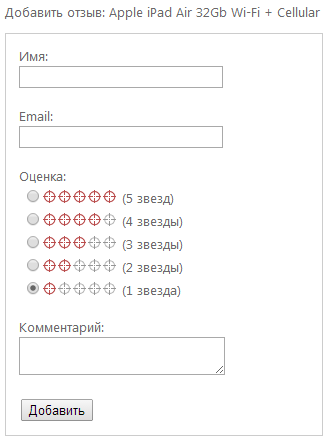


Рисунок 9 – Добавление отзыва

1. Вход от имени администратора

У администратора так же есть такой же личный кабинет, как и у простого пользователя + панель администратора. (см. рис.10).



Рисунок 10 – Кнопка панели администратора

Нажав на эту кнопку мы попадаем в панель администратора, где можем добавлять/редактировать категории, добавлять/редактировать подкатегории, добавлять/редактировать товары, просматривать все заказы, выводить их в Word, Excel, редактировать/просматривать все отзывы.

Для того, чтобы отредактировать нужную категорию, переходим к панели “Изменить”, там выбираем “Категории”. Слева появится список, категорий. Кликнув ЛКМ на нужную категорию, мы сможем ee отредактировать, а так же удалить, или же добавить новую. Кликаем “правка” и изменяем нашу категорию (см. рис.11).

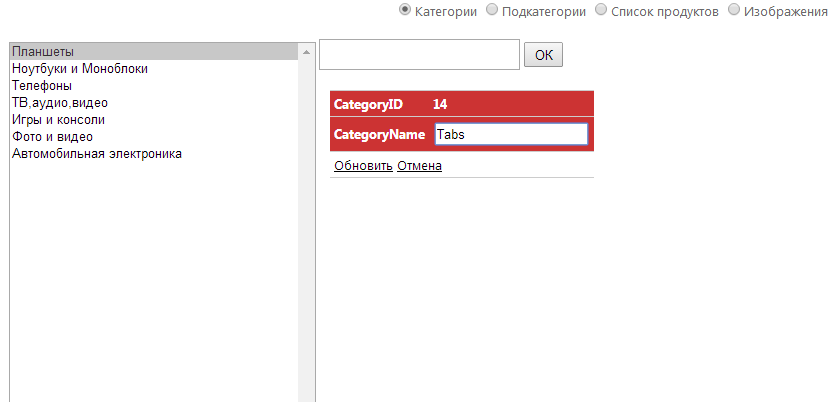


Рисунок 11 – Редактирование категории

Нажимаем кнопку “Обновить” и эта категория обновится. Чтобы проверить, кликните ЛКМ по кнопке “Назад” в главном меню. (см. рис.12).

C:\Users\Мурад\Desktop\Пояснительные записки ТРПП и РиЭАИС\Клиентская часть\9.png

Рисунок 12 – Главное меню панели администратора

Получим:

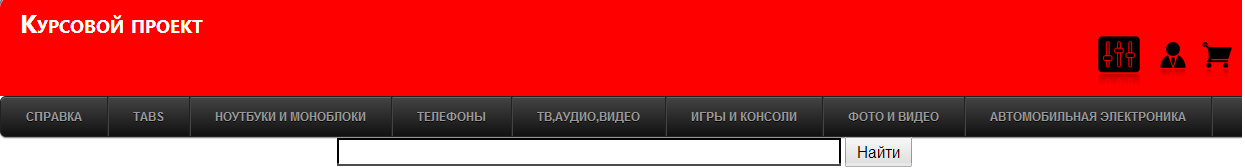


Рисунок 13 – Главное меню домашней страницы

Аналогичным образом можно создавать, редактировать и удалять, подкатегории, товары.

*Руководство пользователя в серверной части программы:*

1. **Установка Microsoft SQL Server 2008 R2**

Запускаем установщик Microsoft SQL Server 2008 R2 и нажимаем "Новая установка или добавление компонентов к существующей установке" (см. рис.14).

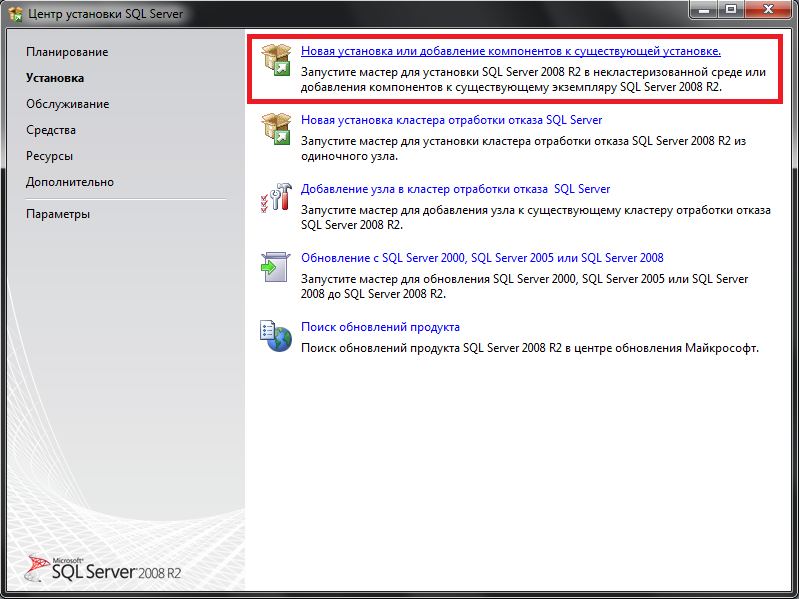


Рисунок 14 - Окно установки Microsoft SQL Server 2008 R2

В отчете все должно быть "Выполнено". Нажимаем кнопку "ОК" (см. рис. 15).

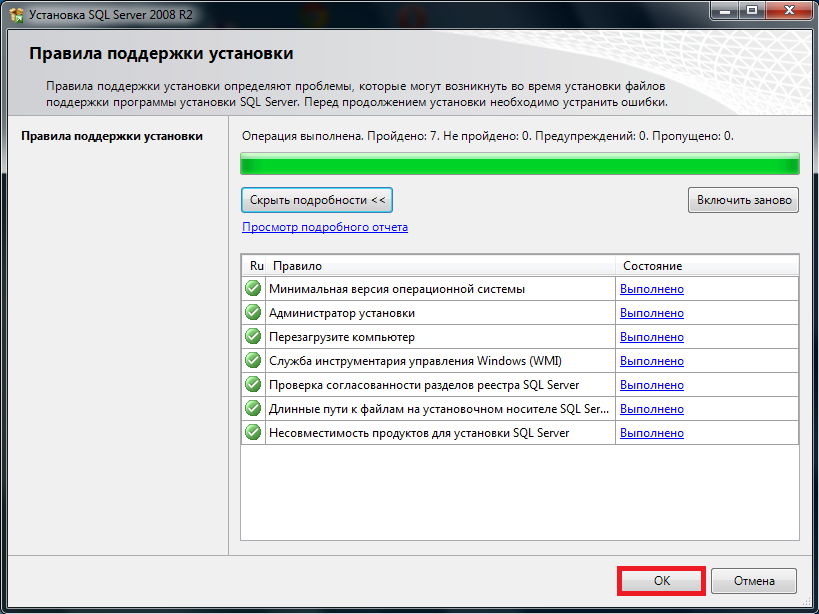


Рисунок 15 – Правила поддержки установки

Отмечаем галочкой "Я принимаю условия лицензионного соглашения". Нажимаем кнопку "Далее" (см. рис.16).

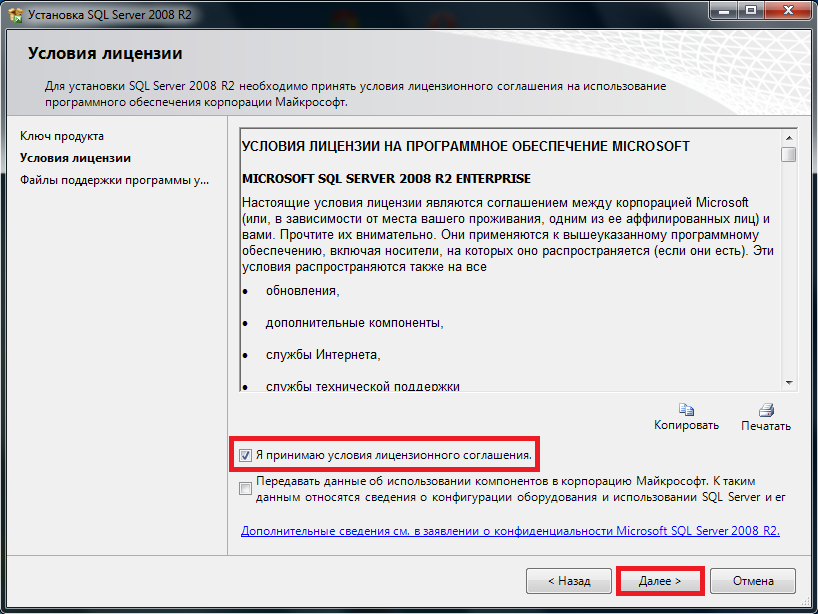


Рисунок 16 – Условия лицензии

Нажимаем кнопку "Установить". В отчете все должно быть "Выполнено" (Если Брандмауэр показывает "Внимание!", то не обращаем на него и продолжаем установку). Нажимаем кнопку "Далее" (см. рис.17).

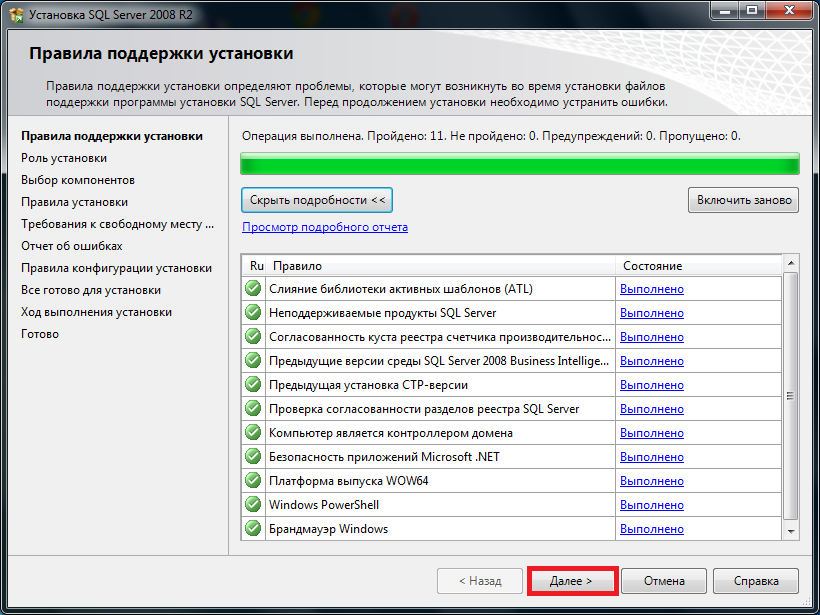


Рисунок 17 – Правила поддержки установки

Нажимаем кнопку "Выделить все" и нажимаем кнопку "Далее" (см. рис.18).

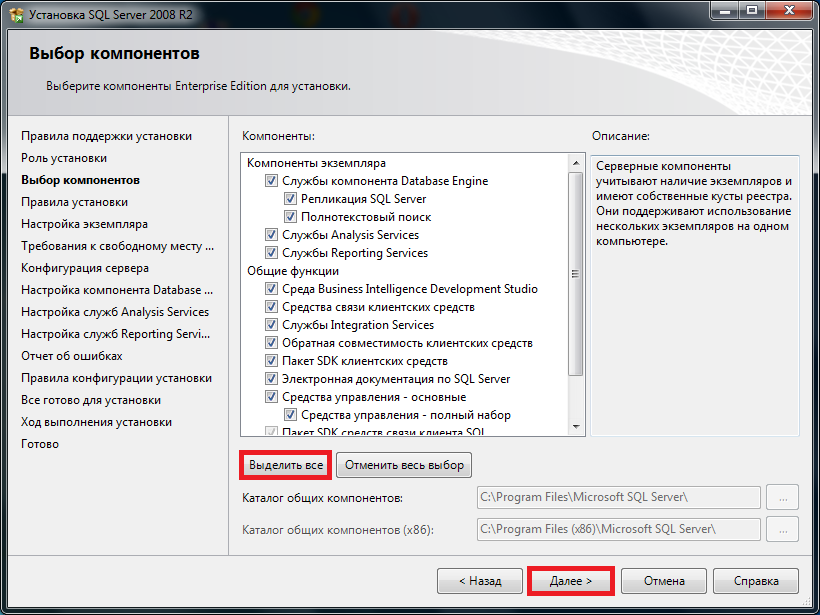


Рисунок 18 – Выбор компонентов

В отчете все должно быть "Выполнено" и "Неприменимо". Нажимаем кнопку "Далее" (см. рис. 19).

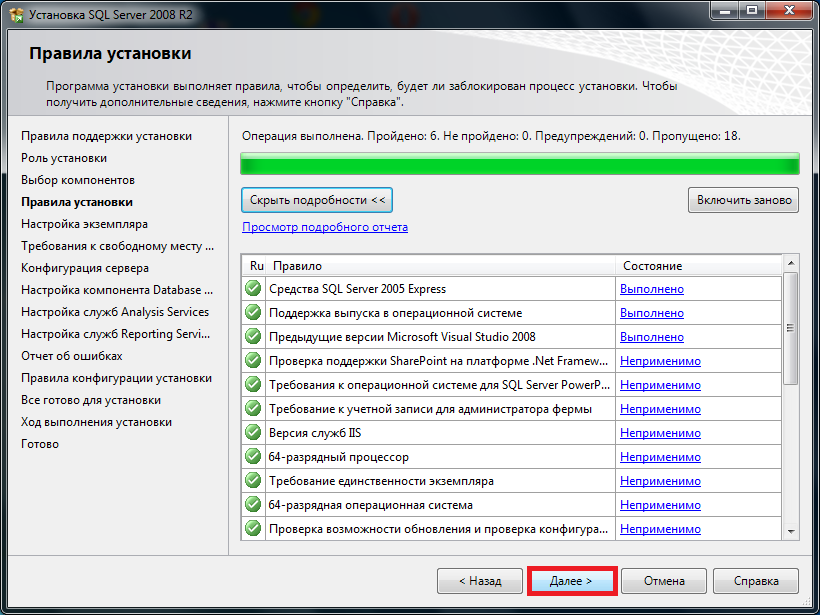


Рисунок 19 – Правила установки

В "Имя учетной записи" и "Тип запуска" выставляем все как на скрине. Нажимаем кнопку "Далее" (см. рис. 20).

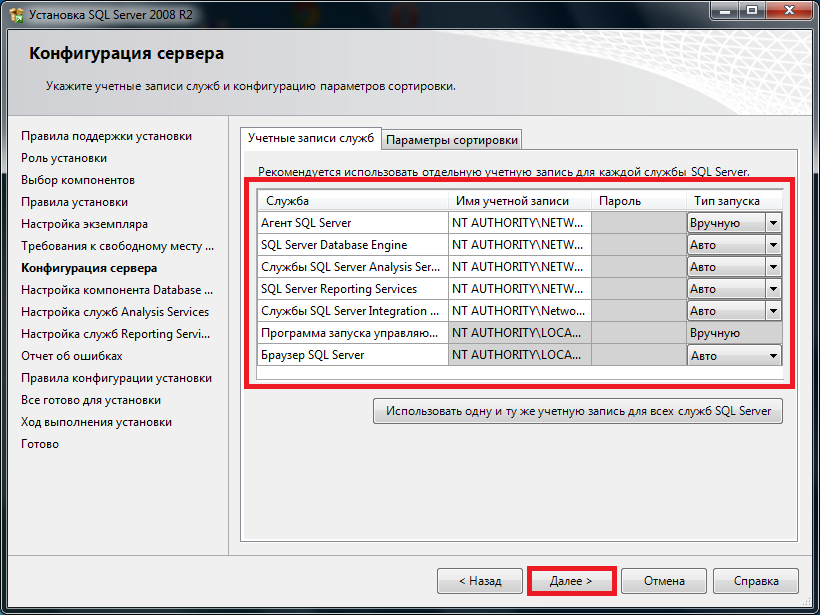


Рисунок 20 – Конфигурация сервера

Выбираем "Смешанный режим". Вводим пароль (его надо запомнить). Повторяем пароль. Нажимаем кнопку "Добавить текущего пользователя". Нажимаем кнопку "Далее" (см. рис. 21).

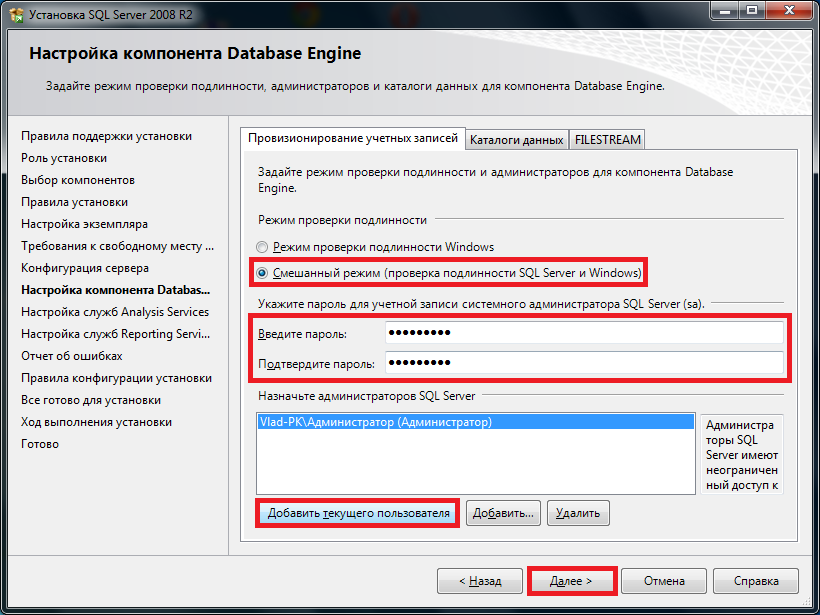


Рисунок 21 – Настройка компонента Database Engine

Нажимаем кнопку "Далее". Далее, нажимаем кнопку "Установить" (см. рис. 22).

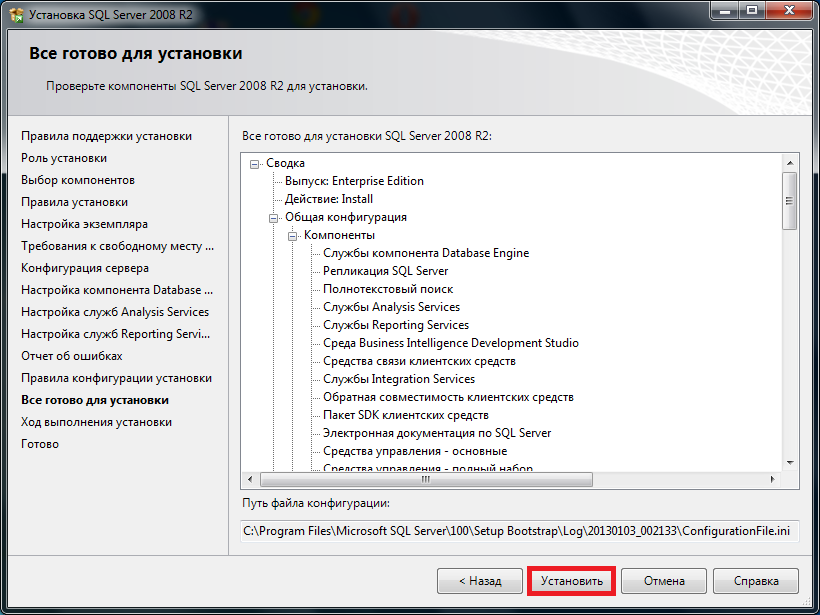


Рисунок 22 – Все готово для установки

Ждем конца установки и нажимаем кнопку "Закрыть". Поздравляем, установка программы завершена.

1. Техническое обеспечение

На серверном компьютере должны быть примерно следующие технические характеристики:

* Двух ядерный процессор с тактовой частотой 2,26 гигагерц (ГГц) или выше;
* 4 гигабайт (ГБ) оперативной памяти (ОЗУ);
* не менее 2 гигабайт (ГБ) свободного дискового пространства;
* 512 мегабайт (МБ) видео памяти;
* операционную систему: Windows Vista или новее, Windows Server 2008 или новее.

На серверном компьютере должны быть установлены следующие программы:

* Microsoft SQL Server 2008 R2 или выше
* .Net Framework 4.0 или выше
* Веб-браузер последней версии, с поддержкой HTML5
* IIS 6.5 или выше
* Microsoft Redistributable 2005-2013

Клиентские компьютеры должны быть примерно со следующими техническими характеристиками:

* Процессор с тактовой частотой 2,26 гигагерц (ГГц) или выше;
* 512 мегабайт (МБ) оперативной памяти (ОЗУ) или выше;
* не менее 512 мегабайт (ГБ) свободного дискового пространства;
* 512 мегабайт (МБ) видео памяти;
* операционную систему: Windows XP (SP3) или новее.

На клиентских компьютерах должны быть установлены следующие программы:

* Net Framework 4.0 или выше
* Веб-браузер последней версии, с поддержкой HTML5
* Microsoft Redistributable 2005-2013

Компьютер, выступающий в виде сервера должен иметь внешний статический IP адрес.

# ЭКОНОМИЧЕСКИЕ СВЕДЕНИЯ

## Расчёт стоимости одного часа работы ПЭВМ

**Сэвм = С / (Т×q)**

где **СЭВМ** - стоимость 1 часа работы ПЭВМ;

**С**  - затраты на содержание и обслуживание ПЭВМ;

**Т** - полезный фонд времени работы ПЭВМ;

**q** - количество машин, на которых установлена программа.

**С = Сзп+Сссо+Сн+Сам+Срем+Сэл+См,**

где **СЗзп** - заработная плата обслуживающего персонала;

**Сссо** – взносы на социальное страхование и обеспечение;

**Сн** – накладные расходы;

**Сам** – амортизационные отчисления;

**Срем** – затраты на ремонт оборудования;

**Сэл** - затраты на электроэнергию;

**См** - затраты на материалы.

### *Расчет заработной платы обслуживающего персонала*

Таблица 1. Обслуживающий персонал

|  |  |  |  |
| --- | --- | --- | --- |
| Обслуживающий персонал | Заработная плата за месяц  (руб) | Количество работников  (чел) | Фонд  заработной платы  (руб) |
| Начальник ВЦ | 20000 | 1 | 20000 |
| Инженер-программист | 20000 | 2 | 40000 |
| Оператор | 10000 | 1 | 10000 |
| ИТОГО: | | | 70000 |

Заработная плата сотрудника определяется умножением тарифного коэффициента на ставку 1 разряда.

Если количество машин, обслуживаемых работником, больше количества машин, на которых будет использована Ваша программа, определяем заработную плату работника, приходящуюся на необходимое количество машин:

**ЗПi = (ЗПj / Q) × q,**

ЗП1=(20000/15)×15=20000,00 руб.

ЗП2=(40000/15)×15=40000,00 руб.

ЗП1=(10000/15)×15=10000,00 руб.

**Сзп = Σ ЗПi,**

Сзп=20000,00+40000,00+10000,00=70000,00 руб.

где ЗПj – заработная плата работника в месяц (данные приведены в таблице;

Q - количество машин, обслуживаемых работником;

q - количество машин, на которых установлена программа.

### *Расчет взносов социальное страхование и обеспечение*

**Cссо = (Сзп × % ССО) / 100%,**

Cссо=(70000×30,2)/100%=21140,00 руб.

где %ССО – ставка страховых взносов во внебюджетные фонды

Таблица 2. Ставка страховых взносов во внебюджетные фонды

|  |  |
| --- | --- |
| Государственные внебюджетные фонды | Ставка взносов (%) |
| 1.Пенсионный фонд РФ | 22 |
| 2.Фонд социального страхования РФ | 2,9 |
| 3.Фонд обязательного медицинского страхования РФ | 5,1 |
| 4.Страховые взносы на социальное страхование от несчастных случаев на производстве и профзаболеваний | 0,2 |
| ИТОГО: | 30,2 |

Ставка по взносам на социальное страхование от несчастных случаев на производстве и профессиональных заболеваний зависит от того, какой вид деятельности у организации. Для расчета курсовой работы возьмем 0,2 %.

### *Расчет накладных расходов*

**Сн = (Сзп × %Н) / 100,**

Сн=(70000×84)/100=58800,00 руб.

где %Н – ставка накладных расходов.

### *Расчет амортизационных отчислений*

**Сам = (Сп × Nа) / (100 × 12),**

Сам=(350000×12)/(100×12)=3500,00 руб.

где Nа – ставка годовой нормы амортизации (%).

Таблица 3. Состав оборудования

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование  оборудования | Цена за единицу  (руб) | Количество машин, на которых установлена программа | Стоимость оборудования  (руб) |
| Компьютер серверный…….. | 60000 | 1 | 60000 |
| Компьютер ……. | 20000 | 14 | 280000 |
| Принтер………… | 10000 | 1 | 10000 |
| ИТОГО: | | | 350000 |

### *Расчет затрат на ремонт оборудования*

**Срем = (Сп × %Р) / (100 × 12),**

Срем=(350000×12)/(100×12)=3500,00 руб.

где % Р – ставка годовых плановых затрат на ремонт оборудования(%).

**Расчеты в пунктах 4 и 5 ведутся для одного месяца года, поэтому в знаменателе ставится число 12.**

### *Расчет затрат на электроэнергию*

**Сэл =P × T ×Z,**

Сэл=11×122,55×4,01=5405,68 руб.

где P – общая мощность оборудования;

T – полезный фонд времени работы ПЭВМ;

Z – цена 1 Квт/ч.

Т = Треж - Трем

Т=129-6,45=122,55 ч.

Трем = (Треж × %Рем) / 100,

Трем=(129×5)/100=6,45 ч.

где Треж  – режимный фонд работы оборудования (час);

%Рем – ставка плановых потерь на профилактику и ремонт оборудования (%).

Треж = (Д - Дв – Дпр) × А × S – Д 'пр × t,

Треж= (30–4–0) ×5×1–1×1=129 ч.

где Д – количество календарных дней за отчетный период;

Дв – количество выходных дней за отчетный период;

Дпр – количество праздничных дней за отчетный период;

А - длительность смены (час);

S - количество смен;

t - количество часов, на которые сокращается рабочий день в предпраздничные дни;

Д 'пр – количество предпраздничных дней за отчетный период.

Таблица 4. Общая мощность оборудования

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование  оборудования | Мощность единицы оборудования (Квт/час) | Количество оборудования | Суммарная мощность (Квт/час) |
| Компьютер серверный…….. | 1 | 1 | 1 |
| Компьютер ……. | 0,7 | 14 | 9,8 |
| Принтер………… | 0,2 | 1 | 0,2 |
| ИТОГО: | | | 11 |

### *Расчет затрат на материалы*

Величина затрат на материалы См= 600,00 руб.

### *Расчет стоимости одного часа работы ПЭВМ*

**Сэвм = С / (Т × q),**

С=70000+21140+58800+3500+3500+5405,68+600=162945,68 руб.

Сэвм=163945,68/(122,55×15)=88,64 руб./час

**Структура стоимости 1 часа работы ПЭВМ**

Таблица 5. Структура стоимости 1 часа работы ПЭВМ

|  |  |  |
| --- | --- | --- |
| Элементы затрат | Сумма (руб) | Удельный вес (%) |
| 1.Заработная плата обслуживающего персонала | 70000,00 | 42,959 |
| 2.Взносы на социальное страхование и обеспечение | 21140,00 | 12,974 |
| 3.Накладные расходы | 58800,00 | 36,086 |
| 4.Амортизационные отчисления | 3500,00 | 2,148 |
| 5.Затраты на ремонт оборудования | 3500,00 | 2,148 |
| 6.Затраты на электроэнергию | 5405,68 | 3,317 |
| 7.Затраты на материалы | 600 | 0,368 |
| ИТОГО: | 162945,68 | 100 |

## Расчет затрат на составление и отладку программы

### *Расчет стоимости решения задачи на ЭВМ*

Стоимость решения задачи на ЭВМ рассчитывается по формуле:

**Sо = ЗПрук + ЗПстуд + С 'ссо + С 'н + С 'м + Тотл × Сэвм,**

Sо=450,00+69,23+135,90+436,15+215,00+36×88,64=2770,24 руб.

где ЗПрук – заработная плата руководителя;

ЗПстуд – заработная плата студента;

С 'ссо - взносы на социальное страхование и обеспечение;

С 'н - накладные расходы;

С 'м - затраты на материалы, используемые при составлении и

отладке программы;

Тотл - затраты времени на отладку программы и решение её на

ПЭВМ;

Сэвм - стоимость одного часа работы ПЭВМ.

### *Расчет заработной платы руководителя*

**ЗПрук = (Nч ×Тч) / Nст,**

ЗПрук=(36×300)/24=450,00 руб.

где ЗПрук – заработная плата руководителя;

Nч – время, отводимое для консультации курсовой работы;

Nст - количество студентов в группе.

Тч – часовая ставка руководителя.

### *Расчет заработной платы студента*

**ЗПстуд = (Сст × Тотл) / Ту,**

ЗПстуд=(300×36)/156=69,23 руб.

где ЗПстуд – заработная плата студента;

Сст - стипендия студента;

Тотл - затраты времени на отладку программы и решение задачи

на ПЭВМ;

Ту - количество учебных часов в месяце (36 учебных часов в

неделю).

**Тотл = S1 × d,**

Тотл= 6×6=36 ч.

где S – средняя занятость в течение дня (час);

d – количество дней для отладки программы.

Ту=(30-4-9) ×6=156 ч.

### *Расчет взносов на социальное страхование и обеспечение*

**С 'ссо = (ЗПрук × %ССО) / 100,**

С'ссо=(450,00×30,2)/100=135,90 руб.

где %ССО – ставка страховых взносов во внебюджетные фонды

### *Расчет накладных расходов*

**С 'н = ((ЗПрук + ЗПст) × %Н) / 100,**

С'н=((450,00+69,23)×84)/100=436,15 руб.

где %Н – ставка накладных расходов (%).

### *Расчет затрат на материалы, используемые при составлении и отладке программы*

Таблица 6. Материалы, используемые при составлении и отладке программы

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование  материалов | Цена за единицу  (руб) | Количество | Стоимость (руб) |
| Бумага | 1 | 50 | 50 |
| Ручка | 5 | 1 | 5 |
| Диск | 150 | 1 | 150 |
| Картридж | 0,1 | 100 | 10 |

Sо=50+5+150+10=215 руб.

### *Расчет стоимости решения задачи вручную*

**Sp = Тр × Ч × ( 1 + (%ССО / 100) + (%Н / 100)),**

Sp=2×1,92×(1+(30,2/100)+(84/100))=8,23 руб.

где Тр – затраты времени на решение задачи вручную (час);

Ч – часовая ставка студента, решающего задачу (руб).

**Ч = Сст / Ту**

Ч=300/156=1,92 руб./час

Так как решение задачи ведет студент, то на сумму стипендии взносы на социальное страхование и обеспечение не начисляются.

## Расчет экономической эффективности

### *Расчет экономии труда за счет применения ПЭВМ*

**Эт = Тр × N − ( То + N × Тэвм),**

Эт=2×1200−(90+1200×0,03543)=2267,48 руб.

где Тр – затраты времени на решение задачи вручную (час);

N - предполагаемое число обращений к программе;

То – затраты времени на составление и отладку программы на

ПЭВМ;

Тэвм- время решения задачи на ПЭВМ.

### *Расчет затрат времени на составление и отладку программы*

**То = S1 × К,**

То=6×15=90 ч.

где S – средняя занятость в течение дня (час);

К – количество дней на составление и отладку программы на

ПЭВМ.

### *Расчет времени решения задачи на ПЭВМ*

**Тэвм = Тв + Тм + Тп,**

Тэвм=0,025+0,00043+0,01=0,03543

где Тв – время ввода исходной информации;

Тм – машинное время;

Тп – время вывода информации на печать.

### *Расчет предполагаемого числа обращений к программе*

Для проведения лабораторных работ данной программой пользуется несколько групп. Каждый студент из группы обращается к программе по одному разу. Из-за ряда причин половине групп пришлось обратиться к программе повторно.

**N = (Nст × 1) + (Nст / 2) ×1**

N=(800×1)+(800/2)×1=1200 об.

### *Расчет денежной экономической эффективности применения ПЭВМ*

**Эд = Sр × N − (Sо + N × Sэвм),**

Эд=8,23×1200−(4497,39+1200×3,14)=1604,26 руб.

где Sр – стоимость решения задачи вручную;

N – предполагаемое число обращений к программе;

Sо – стоимость решения задачи на ПЭВМ;

Sэвм – затраты на работу программы.

### *Расчет затрат на работу программы*

**Sэвм = Тэвм × Сэвм,**

Sэвм=0,03543×88,64=3,14 руб.

где Тэвм – время решения задачи на ПЭВМ;

Сэвм – стоимость часа работы ПЭВМ.

### *Расчет роста производительности труда от применения ПЭВМ*

**∆П = (Эт / (Тр × N)) × 100%**

∆П =(2267,48/(2×1200))×100%=94,4785 %

# ЗАКЛЮЧЕНИЕ

Результатом моей курсовой работы стал готовый к работе сайт Интернет-магазин по продаже электроники. В ходе проведения работы я выделил основные задачи сайта, проанализировал различные пути решения и выбрал наиболее подходящие в данной ситуации. Оценивая проделанную работу можно сделать выводы:

* Сайт предоставляет пользователям из целевой группы наиболее нужные им сервисы, а именно - возможность ознакомиться со списком продуктов.
* Дизайн сайта соответствует предполагаемым предпочтениям целевой группы, времени и целям нахождения потенциальных клиентов на сайте.
* Реализован сервис поиска по сайту.
* Реализована навигация с помощью меню по страницам сайта. Ссылки легко отличимы от основного текста, находятся в одном месте.
* Реализована сортировка продукта, что помогает различным категориям целевой аудитории легко ориентироваться в товарах.
* Уделено особое внимание безопасности пользователей, контролю над корректностью входных данных, защита от переполнения базы данных.
* Реализован интерфейс администратора, что позволяет работать с сайтом малоквалифицированным работникам и совершать меньше ошибок при работе с базой данных.

Экономия труда за счёт применения ПЭВМ составил 2267,48 руб., а рост производительности труда составил 94,4785%.

К данной системе были написаны руководство пользователя для сервера и клиента системы.

Данный курсовой проект был выполнен согласно поставленной задачи, и проверен на работоспособность.

# СПИСОК ЛИТЕРАТУРЫ

* + - Миронов М.Г., Загородников С.В. «Экономика отрасли (машиностроение)»: Учебник.- М.: ФОРУМ: ИНФРА-М, 2005. – 320 с.
    - Мартин Грабер. «Введение в SQL.2010»
    - Никита Культин «Основы программирования в Microsoft Visual C# 2010»
    - [Мэтью Мак-Дональд, Адам Фримен, Марио Шпушта Microsoft ASP.NET 4.0 с примерами на C# 2010 для профессионалов, 4-е издание, 2011](http://www.williamspublishing.com/Books/978-5-8459-1702-7.html)

# ПРИЛОЖЕНИЯ

# Приложения А. Скрипт на создание БД

-- Creating table 'Brands'

CREATE TABLE [dbo].[Brands] (

[ID] int NOT NULL,

[BrandName] nvarchar(max) NULL,

[BrandImage] nvarchar(max) NULL,

[ProductID] int NULL

);

GO

-- Creating table 'Categories'

CREATE TABLE [dbo].[Categories] (

[CategoryID] int IDENTITY(1,1) NOT NULL,

[CategoryName] nvarchar(50) NULL

);

GO

-- Creating table 'Images'

CREATE TABLE [dbo].[Images] (

[ImagesID] int IDENTITY(1,1) NOT NULL,

[ImageIcons1] nvarchar(max) NULL,

[ImageIcons2] nvarchar(max) NULL,

[ImageSmall1] nvarchar(max) NULL,

[ImageSmall2] nvarchar(max) NULL,

[ImageBig1] nvarchar(max) NULL,

[ImageBig2] nvarchar(max) NULL,

[ProductID] int NULL

);

GO

-- Creating table 'OrderDetails'

CREATE TABLE [dbo].[OrderDetails] (

[Id] int IDENTITY(1,1) NOT NULL,

[OrderID] int NULL,

[ProductID] int NULL,

[Quantity] int NULL,

[UnitCost] decimal(19,4) NULL,

[TimeDelivery] nvarchar(max) NULL,

[City] nvarchar(max) NULL,

[Street] nvarchar(max) NULL,

[House] nvarchar(50) NULL,

[Housing] nvarchar(150) NULL,

[Structure] nvarchar(100) NULL,

[Porch] nvarchar(50) NULL,

[Floor] nvarchar(50) NULL,

[Apartment] nvarchar(50) NULL,

[FIO] nvarchar(max) NULL,

[Mobile] nvarchar(100) NULL,

[DopMobile] nvarchar(100) NULL,

[MessageText] nvarchar(max) NULL

);

GO

-- Creating table 'Orders'

CREATE TABLE [dbo].[Orders] (

[OrderID] int IDENTITY(1,1) NOT NULL,

[CustomerName] nvarchar(256) NOT NULL,

[OrderDate] datetime NOT NULL,

[ShipDate] nvarchar(max) NOT NULL

);

GO

-- Creating table 'Products'

CREATE TABLE [dbo].[Products] (

[ProductID] int IDENTITY(1,1) NOT NULL,

[CategoryID] int NOT NULL,

[ModelNumber] nvarchar(50) NULL,

[ModelName] nvarchar(50) NULL,

[UnitCost] decimal(19,4) NOT NULL,

[Description] nvarchar(3800) NULL,

[SubcategoriesID] int NULL,

[Video] nvarchar(max) NULL

);

GO

-- Creating table 'Reviews'

CREATE TABLE [dbo].[Reviews] (

[ReviewID] int IDENTITY(1,1) NOT NULL,

[ProductID] int NOT NULL,

[CustomerName] nvarchar(50) NULL,

[CustomerEmail] nvarchar(50) NULL,

[Rating] int NOT NULL,

[Comments] nvarchar(3850) NULL

);

GO

-- Creating table 'ShoppingCarts'

CREATE TABLE [dbo].[ShoppingCarts] (

[RecordID] int IDENTITY(1,1) NOT NULL,

[CartID] nvarchar(50) NULL,

[Quantity] int NOT NULL,

[ProductID] int NOT NULL,

[DateCreated] datetime NOT NULL

);

GO

-- Creating table 'Subcategories'

CREATE TABLE [dbo].[Subcategories] (

[SubcategoriesID] int IDENTITY(1,1) NOT NULL,

[SubcategoriesName] nvarchar(100) NULL,

[CategoryID] int NOT NULL,

[SImage] nvarchar(max) NULL

);

GO

-- Creating table 'sysdiagrams'

CREATE TABLE [dbo].[sysdiagrams] (

[name] nvarchar(128) NOT NULL,

[principal\_id] int NOT NULL,

[diagram\_id] int IDENTITY(1,1) NOT NULL,

[version] int NULL,

[definition] varbinary(max) NULL

);

GO

-- Creating table 'VewOrderDetails'

CREATE TABLE [dbo].[VewOrderDetails] (

[ProductID] int NOT NULL,

[ModelNumber] nvarchar(50) NULL,

[ModelName] nvarchar(50) NULL,

[Quantity] int NULL,

[UnitCost] decimal(19,4) NULL,

[OrderID] int NULL

);

GO

-- Creating table 'ViewAlsoPurchaseds'

CREATE TABLE [dbo].[ViewAlsoPurchaseds] (

[ProductID] int NULL,

[ModelName] nvarchar(50) NULL,

[TotalNumPurchased] int NULL,

[OrderID] int NULL,

[Products\_ProductID] int NOT NULL

);

GO

-- Creating table 'ViewCarts'

CREATE TABLE [dbo].[ViewCarts] (

[ProductID] int NOT NULL,

[ModelNumber] nvarchar(50) NULL,

[ModelName] nvarchar(50) NULL,

[UnitCost] decimal(19,4) NOT NULL,

[Quantity] int NOT NULL,

[CartID] nvarchar(50) NULL

);

GO

-- Creating table 'ViewCategs'

CREATE TABLE [dbo].[ViewCategs] (

[CategoryName] nvarchar(50) NULL,

[SubcategoriesName] nvarchar(100) NULL,

[CategoryID] int NOT NULL

);

GO

-- Creating table 'ViewImages'

CREATE TABLE [dbo].[ViewImages] (

[ImageIcons1] nvarchar(max) NULL,

[ImageIcons2] nvarchar(max) NULL,

[ImageSmall1] nvarchar(max) NULL,

[ImageSmall2] nvarchar(max) NULL,

[ImageBig1] nvarchar(max) NULL,

[ImageBig2] nvarchar(max) NULL,

[ModelNumber] nvarchar(50) NULL,

[ModelName] nvarchar(50) NULL,

[UnitCost] decimal(19,4) NOT NULL,

[Description] nvarchar(3800) NULL,

[ProductID] int NOT NULL,

[SubcategoriesID] int NULL,

[Video] nvarchar(max) NULL

);

GO

-- Creating table 'ViewPopularItems'

CREATE TABLE [dbo].[ViewPopularItems] (

[ProductID] int NOT NULL,

[ModelName] nvarchar(50) NULL,

[UnitCost] decimal(19,4) NOT NULL,

[ImageIcons1] nvarchar(max) NULL

);

GO

-- --------------------------------------------------

-- Creating all PRIMARY KEY constraints

-- --------------------------------------------------

-- Creating primary key on [ID] in table 'Brands'

ALTER TABLE [dbo].[Brands]

ADD CONSTRAINT [PK\_Brands]

PRIMARY KEY CLUSTERED ([ID] ASC);

GO

-- Creating primary key on [CategoryID] in table 'Categories'

ALTER TABLE [dbo].[Categories]

ADD CONSTRAINT [PK\_Categories]

PRIMARY KEY CLUSTERED ([CategoryID] ASC);

GO

-- Creating primary key on [ImagesID] in table 'Images'

ALTER TABLE [dbo].[Images]

ADD CONSTRAINT [PK\_Images]

PRIMARY KEY CLUSTERED ([ImagesID] ASC);

GO

-- Creating primary key on [Id] in table 'OrderDetails'

ALTER TABLE [dbo].[OrderDetails]

ADD CONSTRAINT [PK\_OrderDetails]

PRIMARY KEY CLUSTERED ([Id] ASC);

GO

-- Creating primary key on [OrderID] in table 'Orders'

ALTER TABLE [dbo].[Orders]

ADD CONSTRAINT [PK\_Orders]

PRIMARY KEY CLUSTERED ([OrderID] ASC);

GO

-- Creating primary key on [ProductID] in table 'Products'

ALTER TABLE [dbo].[Products]

ADD CONSTRAINT [PK\_Products]

PRIMARY KEY CLUSTERED ([ProductID] ASC);

GO

-- Creating primary key on [ReviewID] in table 'Reviews'

ALTER TABLE [dbo].[Reviews]

ADD CONSTRAINT [PK\_Reviews]

PRIMARY KEY CLUSTERED ([ReviewID] ASC);

GO

-- Creating primary key on [RecordID] in table 'ShoppingCarts'

ALTER TABLE [dbo].[ShoppingCarts]

ADD CONSTRAINT [PK\_ShoppingCarts]

PRIMARY KEY CLUSTERED ([RecordID] ASC);

GO

-- Creating primary key on [SubcategoriesID] in table 'Subcategories'

ALTER TABLE [dbo].[Subcategories]

ADD CONSTRAINT [PK\_Subcategories]

PRIMARY KEY CLUSTERED ([SubcategoriesID] ASC);

GO

-- Creating primary key on [diagram\_id] in table 'sysdiagrams'

ALTER TABLE [dbo].[sysdiagrams]

ADD CONSTRAINT [PK\_sysdiagrams]

PRIMARY KEY CLUSTERED ([diagram\_id] ASC);

GO

-- Creating primary key on [ProductID] in table 'VewOrderDetails'

ALTER TABLE [dbo].[VewOrderDetails]

ADD CONSTRAINT [PK\_VewOrderDetails]

PRIMARY KEY CLUSTERED ([ProductID] ASC);

GO

-- Creating primary key on [Products\_ProductID] in table 'ViewAlsoPurchaseds'

ALTER TABLE [dbo].[ViewAlsoPurchaseds]

ADD CONSTRAINT [PK\_ViewAlsoPurchaseds]

PRIMARY KEY CLUSTERED ([Products\_ProductID] ASC);

GO

-- Creating primary key on [ProductID], [UnitCost], [Quantity] in table 'ViewCarts'

ALTER TABLE [dbo].[ViewCarts]

ADD CONSTRAINT [PK\_ViewCarts]

PRIMARY KEY CLUSTERED ([ProductID], [UnitCost], [Quantity] ASC);

GO

-- Creating primary key on [CategoryID] in table 'ViewCategs'

ALTER TABLE [dbo].[ViewCategs]

ADD CONSTRAINT [PK\_ViewCategs]

PRIMARY KEY CLUSTERED ([CategoryID] ASC);

GO

-- Creating primary key on [UnitCost], [ProductID] in table 'ViewImages'

ALTER TABLE [dbo].[ViewImages]

ADD CONSTRAINT [PK\_ViewImages]

PRIMARY KEY CLUSTERED ([UnitCost], [ProductID] ASC);

GO

-- Creating primary key on [ProductID], [UnitCost] in table 'ViewPopularItems'

ALTER TABLE [dbo].[ViewPopularItems]

ADD CONSTRAINT [PK\_ViewPopularItems]

PRIMARY KEY CLUSTERED ([ProductID], [UnitCost] ASC);

GO

-- --------------------------------------------------

-- Creating all FOREIGN KEY constraints

-- --------------------------------------------------

-- Creating foreign key on [ProductID] in table 'Brands'

ALTER TABLE [dbo].[Brands]

ADD CONSTRAINT [FK\_Brands\_Products]

FOREIGN KEY ([ProductID])

REFERENCES [dbo].[Products]

([ProductID])

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- Creating non-clustered index for FOREIGN KEY 'FK\_Brands\_Products'

CREATE INDEX [IX\_FK\_Brands\_Products]

ON [dbo].[Brands]

([ProductID]);

GO

-- Creating foreign key on [CategoryID] in table 'Products'

ALTER TABLE [dbo].[Products]

ADD CONSTRAINT [FK\_Products\_Categories]

FOREIGN KEY ([CategoryID])

REFERENCES [dbo].[Categories]

([CategoryID])

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- Creating non-clustered index for FOREIGN KEY 'FK\_Products\_Categories'

CREATE INDEX [IX\_FK\_Products\_Categories]

ON [dbo].[Products]

([CategoryID]);

GO

-- Creating foreign key on [CategoryID] in table 'Subcategories'

ALTER TABLE [dbo].[Subcategories]

ADD CONSTRAINT [FK\_Subcategories\_Categories]

FOREIGN KEY ([CategoryID])

REFERENCES [dbo].[Categories]

([CategoryID])

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- Creating non-clustered index for FOREIGN KEY 'FK\_Subcategories\_Categories'

CREATE INDEX [IX\_FK\_Subcategories\_Categories]

ON [dbo].[Subcategories]

([CategoryID]);

GO

-- Creating foreign key on [ProductID] in table 'Images'

ALTER TABLE [dbo].[Images]

ADD CONSTRAINT [FK\_Images\_Products]

FOREIGN KEY ([ProductID])

REFERENCES [dbo].[Products]

([ProductID])

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- Creating non-clustered index for FOREIGN KEY 'FK\_Images\_Products'

CREATE INDEX [IX\_FK\_Images\_Products]

ON [dbo].[Images]

([ProductID]);

GO

-- Creating foreign key on [OrderID] in table 'OrderDetails'

ALTER TABLE [dbo].[OrderDetails]

ADD CONSTRAINT [FK\_Order\_OrderDetails]

FOREIGN KEY ([OrderID])

REFERENCES [dbo].[Orders]

([OrderID])

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- Creating non-clustered index for FOREIGN KEY 'FK\_Order\_OrderDetails'

CREATE INDEX [IX\_FK\_Order\_OrderDetails]

ON [dbo].[OrderDetails]

([OrderID]);

GO

-- Creating foreign key on [SubcategoriesID] in table 'Products'

ALTER TABLE [dbo].[Products]

ADD CONSTRAINT [FK\_Products\_Subcategories]

FOREIGN KEY ([SubcategoriesID])

REFERENCES [dbo].[Subcategories]

([SubcategoriesID])

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- Creating non-clustered index for FOREIGN KEY 'FK\_Products\_Subcategories'

CREATE INDEX [IX\_FK\_Products\_Subcategories]

ON [dbo].[Products]

([SubcategoriesID]);

GO

-- Creating foreign key on [ProductID] in table 'Reviews'

ALTER TABLE [dbo].[Reviews]

ADD CONSTRAINT [FK\_Reviews\_Products]

FOREIGN KEY ([ProductID])

REFERENCES [dbo].[Products]

([ProductID])

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- Creating non-clustered index for FOREIGN KEY 'FK\_Reviews\_Products'

CREATE INDEX [IX\_FK\_Reviews\_Products]

ON [dbo].[Reviews]

([ProductID]);

GO

-- Creating foreign key on [ProductID] in table 'ShoppingCarts'

ALTER TABLE [dbo].[ShoppingCarts]

ADD CONSTRAINT [FK\_ShoppingCart\_Products]

FOREIGN KEY ([ProductID])

REFERENCES [dbo].[Products]

([ProductID])

ON DELETE NO ACTION ON UPDATE NO ACTION;

-- Creating non-clustered index for FOREIGN KEY 'FK\_ShoppingCart\_Products'

CREATE INDEX [IX\_FK\_ShoppingCart\_Products]

ON [dbo].[ShoppingCarts]

([ProductID]);

GO

# 

# Приложения Б. Модуль клиентской части

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using Shop.Data\_Access;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.Security;

namespace Shop

{

public struct ShoppingCartUpdates

{

public int ProductId;

public int PurchaseQuantity;

public bool RemoveItem;

}

public partial class MyShoppingCart

{

public const string CartId = "Shop\_CartID";

public String GetShoppingCartId()

{

if (Session[CartId] == null)

{

Session[CartId] = System.Web.HttpContext.Current.Request.IsAuthenticated ? User.Identity.Name : Guid.NewGuid().ToString();

}

return Session[CartId].ToString();

}

public void AddItem(string cartID, int productID, int quantity)

{

using (ShopDBEntities db = new ShopDBEntities())

{

try

{

var myItem = (from c in db.ShoppingCarts where c.CartID == cartID && c.ProductID == productID select c).FirstOrDefault();

if (myItem == null)

{

ShoppingCart cartadd = new ShoppingCart();

cartadd.CartID = cartID;

cartadd.Quantity = quantity;

cartadd.ProductID = productID;

cartadd.DateCreated = DateTime.Now;

db.ShoppingCarts.AddObject(cartadd);

}

else

{

myItem.Quantity += quantity;

}

db.SaveChanges();

}

catch (Exception exp)

{

throw new Exception("ОШИБКА! Не удается добавить элемент в корзину - " + exp.Message.ToString(), exp);

}

}

}

public decimal GetTotal(string cartID)

{

using (ShopDBEntities db = new ShopDBEntities())

{

decimal cartTotal = 0;

try

{

var myCart = (from c in db.ViewCarts where c.CartID == cartID select c);

if (myCart.Count() > 0)

{

cartTotal = myCart.Sum(od => (decimal)od.Quantity \* (decimal)od.UnitCost);

}

}

catch (Exception exp)

{

throw new Exception("ОШИБКА! Не удалось вычислить сумму заказа - " + exp.Message.ToString(), exp);

}

return (cartTotal);

}

}

public void RemoveItem(string cartID, int productID)

{

using (ShopDBEntities db = new ShopDBEntities())

{

try

{

var myItem = (from c in db.ShoppingCarts where c.CartID == cartID && c.ProductID == productID select c).FirstOrDefault();

if (myItem != null)

{

db.DeleteObject(myItem);

db.SaveChanges();

}

}

catch (Exception exp)

{

throw new Exception("ОШИБКА! Не удалось удалить элемент корзины - " + exp.Message.ToString(), exp);

}

}

}

public void UpdateItem(string cartID, int productID, int quantity)

{

using (ShopDBEntities db = new ShopDBEntities())

{

try

{

var myItem = (from c in db.ShoppingCarts where c.CartID == cartID && c.ProductID == productID select c).FirstOrDefault();

if (myItem != null)

{

myItem.Quantity = quantity;

db.SaveChanges();

}

}

catch (Exception exp)

{

throw new Exception("ОШИБКА! Не удалось обновить элемент корзины - " + exp.Message.ToString(), exp);

}

}

}

public void UpdateShoppingCartDatabase(String cartId, ShoppingCartUpdates[] CartItemUpdates)

{

using (ShopDBEntities db = new ShopDBEntities())

{

try

{

int CartItemCOunt = CartItemUpdates.Count();

var myCart = (from c in db.ViewCarts where c.CartID == cartId select c);

foreach (var cartItem in myCart)

{

// Итерация по всем строкам в списке корзины

for (int i = 0; i < CartItemCOunt; i++)

{

if (cartItem.ProductID == CartItemUpdates[i].ProductId)

{

if (CartItemUpdates[i].PurchaseQuantity < 1 || CartItemUpdates[i].RemoveItem == true)

{

RemoveItem(cartId, cartItem.ProductID);

}

else

{

UpdateItem(cartId, cartItem.ProductID, CartItemUpdates[i].PurchaseQuantity);

}

}

}

}

}

catch (Exception exp)

{

throw new Exception("ОШИБКА! Не удалось обновить базу данных корзины - " + exp.Message.ToString(), exp);

}

}

}

public void MigrateCart(String oldCartId, String UserName)

{

using (ShopDBEntities db = new ShopDBEntities())

{

try

{

var myShoppingCart= from cart in db.ShoppingCarts

where cart.CartID == oldCartId

select cart;

foreach (ShoppingCart item in myShoppingCart)

{

item.CartID = UserName;

}

db.SaveChanges();

Session[CartId] = UserName;

}

catch (Exception exp)

{

throw new Exception("ОШИБКА! Не удалось переместить корзину - " + exp.Message.ToString(), exp);

}

}

}

public bool SubmitOrder(string UserName)

{

using (ShopDBEntities db = new ShopDBEntities())

{

try

{

//------------------------------------------------------------------------+

// Добавление новой записи заказа |

//------------------------------------------------------------------------+

Order newOrder = new Order();

newOrder.CustomerName = UserName;

newOrder.OrderDate = DateTime.Now;

newOrder.ShipDate = Session["Date"].ToString();

//newOrder.ShipDate = CalculateShipDate();

db.Orders.AddObject(newOrder);

db.SaveChanges();

String cartId = GetShoppingCartId();

var myCart = (from c in db.ViewCarts where c.CartID == cartId select c);

foreach (ViewCart item in myCart)

{

int i = 0;

if (i < 1)

{

OrderDetail od = new OrderDetail();

od.OrderID = newOrder.OrderID;

od.ProductID = item.ProductID;

od.Quantity = item.Quantity;

od.UnitCost = item.UnitCost;

od.TimeDelivery = Session["Time"].ToString();

od.Apartment = Session["Apartment"].ToString();

od.City = Session["City"].ToString();

od.MessageText = Session["Comments"].ToString();

od.FIO = Session["FIO"].ToString();

od.House = Session["House"].ToString();

od.Housing = Session["Housing"].ToString();

od.Floor = Session["Floor"].ToString();

od.DopMobile = Session["DopMobile"].ToString();

od.Mobile = Session["Mobile"].ToString();

od.Street = Session["Street"].ToString();

od.Porch = Session["Porch"].ToString();

od.Structure = Session["Structure"].ToString();

db.OrderDetails.AddObject(od);

i++;

}

var myItem = (from c in db.ShoppingCarts where c.CartID == item.CartID && c.ProductID == item.ProductID select c).FirstOrDefault();

if (myItem != null)

{

db.DeleteObject(myItem);

}

}

db.SaveChanges();

}

catch (Exception exp)

{

throw new Exception("ОШИБКА! Не удалось отправить заказ - " + exp.Message.ToString(), exp);

}

}

return (true);

} } }