**Федеральное агентство связи**

**Федеральное государственное образовательное бюджетное учреждение высшего**

**профессионального образования**

**«Сибирский государственный университет телекоммуникаций и информатики»**

**(ФГОБУ ВПО «СибГУТИ»)**

Новосибирск - 2014

Форма утверждена

научно-методическим

советом ФГОБУ ВПО «СибГУТИ»

Протокол №3 от 15.04.2014 г.

Кафедра

Допустить к защите

зав. кафедрой

Разработка Интернет-магазина для ремонтно-отделочной организации

**ВЫПУСКНАЯ РАБОТА БАКАЛАВРА**

Пояснительная записка

ФИВТ.10126-и ПЗ

Студент:

Факультет Группа

Руководитель

Федеральное агентство связи

Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования

«Сибирский государственный университет телекоммуникаций и информатики»

(ФГОБУ ВПО «СибГУТИ»)

Форма утверждена

научно-методическим

советом ФГОБУ ВПО «СибГУТИ»

Протокол №2 от 04.03.2014 г.

**КАФЕДРА**

**ЗАДАНИЕ**

|  |  |
| --- | --- |
| СТУДЕНТУ | ГРУППЫ |

«УТВЕРЖДАЮ»

зав. Кафедрой

Новосибирск, 2014 г.

1. Тема проекта: «Разработка Интернет-магазина для ремонтно-отделочной организации» утверждена указом по университету от  № 4
2. Срок сдачи студентом законченной работы:
3. Исходные данные по проекту (эксплуатационно-технические данные):

* *Учебные онлайн-курсы по CMS 1С-БИТРИКС: Управление сайтом* [*http://dev.1c-bitrix.ru/learning/index.php*](http://dev.1c-bitrix.ru/learning/index.php)
* *Документация для разработчиков 1C-Битрикс: Управление сайтом* [*http://dev.1c-bitrix.ru/api\_help/*](http://dev.1c-bitrix.ru/api_help/)
* *Интернет-ресурс по верстке* [*http://htmlbook.ru/*](http://htmlbook.ru/)
* *Руководство по PHP* [*http://www.php.net/manual/ru/*](http://www.php.net/manual/ru/)

1. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов) и сроки выполнения по разделам:

|  |  |
| --- | --- |
| Задача | Время выполнения, ч |
| Верстка сайта, натяжка шаблона на Битрикс | 30 |
| Выполнение текстовых страниц | 0.5 |
| Выполнение страниц «Новости», «Статьи», «Отзывы», «Фотогалерея» | 4 |
| Написание компонента разделов каталога | 3 |
| Написание компонента элементов раздела | 2 |
| Написание компонента детального просмотра товара | 3 |
| Интеграция хлебных крошек в компоненты для навигации по сайту | 0.5 |
| Написание компонента просмотренных товаров | 2 |
| Написание компонента маленькой корзины | 1 |
| Написание компонента полной корзины | 4 |
| Написание компонента добавления элемента в инфоблок | 2 |
| Написание компонента оформления заказа | 3 |

Дата выдачи задания:

Задание принял к исполнению

Федеральное агентство связи

Федеральное государственное образовательное бюджетное учреждение

высшего профессионального образования

«Сибирский государственный университет телекоммуникаций и информатики»

(ФГОБУ ВПО «СибГУТИ»)

Форма утверждена

научно-методическим

советом ФГОБУ ВПО «СибГУТИ»

Протокол №2 от 04.03.2014 г.

**ОТЗЫВ**

на бакалаврскую работу студент

группы

|  |  |  |  |
| --- | --- | --- | --- |
| Работа имеет практическую ценность |  | Тема предложена студентом |  |
| Рекомендация к внедрению |  | Тема является фундаментальной |  |
| Рекомендация к опубликованию |  | Рекомендую студента в магистратуру |  |
| Тема предложена предприятием |  | Рекомендую студента в аспирантуру |  |

|  |  |
| --- | --- |
|  |  |
| (Перцев Игорь Владимирович) | |
|  | |

**АННОТАЦИЯ**

бакалаврской работы студента

по теме «Разработка интернет-магазина для ремонтно-отделочной организации»

Объём бакалаврской работы 62 страницы, на которых размещены 23 рисунка и 1 таблица. При написании работы использовалось 9 источников.

Ключевые слова: интернет-магазин, сайт, 1С-Битрикс: Управление сайтом редакция Старт.

Работа выполнена на Кафедре ПМиК СибГУТИ.

Руководитель – ,

Целью бакалаврской работы было написание интернет-магазина на системе управления сайтом компании 1С-Битрикс редакции Старт, в которой заведомо нет возможности разработать интернет-магазин.

В рамках дипломного проекта были разработаны компоненты, взаимодействующие между собой, которые позволяют реализовать оформление и продажу товара с сайта. Проведено экспериментальное исследование алгоритмов, составляющих компоненты.

**Graduation thesis abstract**

of

on the theme « Development of Internet-shop for repair-finishing company »

The paper consists of 62 pages, with 23 figures and 1 tables. While writing the thesis 9 reference sources were used.  
  
Keywords: online store site, 1C-Bitrix Site Manager Start edition.  
  
The thesis was written at department of Applied Mathematics and Cybernetics at SIBSUTIS.   
Scientific supervisor – Ph. D Associate Professor Pertsev I.V., 

The subject of the paper is writing an online store on the content management system of 1C-Bitrix company.

The aim of this thesis is to create an online store on the content management system of 1C-Bitrix company.

The result of the project is own module, which includes components. Components that interact with each other were developed, they allow to implement the design and sale of goods from the site. An experimental investigation of algorithms which compose the components was held.

Содержание

[1 Введение 8](#_Toc390166435)

[1.1 Преимущества и недостатки интернет-магазинов 8](#_Toc390166436)

[1.2 Создание интернет-магазина 10](#_Toc390166437)

[1.2.1 Рассмотрение функционала CMS 1С-Битрикс: Управление Сайтом 11](#_Toc390166438)

[2 Теоретическая подготовка 13](#_Toc390166439)

[2.1 PHP................ 13](#_Toc390166440)

[2.2 HTML+CSS 14](#_Toc390166441)

[2.3 jQuery........... 15](#_Toc390166442)

[3 Программная часть 17](#_Toc390166443)

[3.1 Создание простого компонента для Битрикса 17](#_Toc390166444)

[3.2 Написание логики компонентов 20](#_Toc390166445)

[3.2.1 Выборка разделов из инфоблока CIBlockSection::GetList 20](#_Toc390166446)

[3.2.2 Добавление товаров в корзину и в категорию просмотренных.... 21](#_Toc390166447)

[3.2.3 Выборка элементов инфоблока CIblockElement::GetList 22](#_Toc390166448)

[3.2.4 Добавление элемента с определенными свойствами в инфоблок...... 22](#_Toc390166449)

[3.2.5 Функция валидации почты 23](#_Toc390166450)

[4 АДМИнистративная часть 25](#_Toc390166451)

[4.1 Установка 1С-Битрикс: Управление сайтом. 25](#_Toc390166452)

[4.2 Включение шаблона 26](#_Toc390166453)

[4.3 Настройка компонентов 27](#_Toc390166454)

[5 Пользовательская часть 29](#_Toc390166455)

[6 Заключение 36](#_Toc390166456)

[Приложение А 37](#_Toc390166457)

[Приложение б 38](#_Toc390166458)

[ПРИЛОЖЕНИЕ В 39](#_Toc390166459)

# Введение

Интернет прочно вошел в нашу жизнь. Он объединил огромное число компьютеров и других устройств, расположенных по всему Земному шару, в единое целое. С его помощью люди могут общаться друг с другом, используя электронную почту, системы электронных досок объявлений и даже видеоконференции. Ученые получили возможность решать задачи, требующие огромных вычислительных ресурсов, объединив множество компьютеров энтузиастов в один гигантский суперкомпьютер с помощью grid-технологий. Примерами таких задач являются программы поиска внеземного разума и создание глобальных климатических моделей.

Интернет уже превратился в полноценный источник разнообразной полезной информации для любой категории пользователей. В это время любой человек, владеющий компьютером, сможет "скачать" с его помощью статьи, иллюстрации, видео- или аудиоинформацию по любой интересующей его теме. Он сможет получить эту информацию тогда, когда он этого пожелает.

Также интернет открыл дорогу электронной коммерции: всевозможным интернет-магазинам, посадочным страницам и т.д. Сейчас не нужно выходить из дома, чтобы купить пищу, электронику, сувениры и множество других товаров.

Преимущества виртуального магазина перед реальным очевидны. Уменьшается численность персонала за счет сокращения объема взаимодействия с клиентами, аренда дискового пространства и размещение «электронной витрины» дешевле и проще аренды торговых помещений и размещения товаров на полках, нет нужды в кассовом обслуживании и т.д. Решая вопрос о подключении Интернет-магазина к той или иной платежной системе, его руководителям необходимо заранее оценить количество пользователей системы, которым будет интересен ассортимент Интернет-магазина.

Интернет-коммерция зарождалась как рынок розничной торговли. Но

постепенно на рынке стали появляется решения, ориентированные не на конечных потребителей, а на организации, так называемый рынок B2B, или business-to-business (в противоположность B2C, business-to-customer). Первоначально, на рынке B2B предлагались продукты, связанные с собственно организацией розничной торговли (готовые Интернет — магазины, услуги по рекламе, внедрение Интернет — торговли в традиционные бизнес — процессы и т.п.). Но постепенно через Интернет стали продаваться решения, непосредственно с Интернет не связанные (коммерческое программное обеспечение, услуги по автоматизации, оптовая торговля, брокерские услуги, консалтинг и т.п.).

## Преимущества и недостатки интернет-магазинов

Интернет-магазин во многом выигрывает перед реальным магазином. Уменьшается численность персонала за счет сокращения объема взаимодействия с клиентами, аренда дискового пространства и размещение «электронной витрины» дешевле и проще аренды торговых помещений и размещения товаров на полках, нет нужды в кассовом обслуживании и т.д. Так же виртуальный магазин можно использовать как эффективный способ маркетингового исследования, тем более, что сегодня эта услуга довольно дорога в маркетинговых агентствах. Любой пользователь сети Интернет может быстро заполнить анкету, предлагаемую ему магазином через компьютер. Это позволяет без особых затрат изучить потребности и вкусы потенциальных покупателей и учесть результаты маркетингового обследования в своей работе.

В чем недостаток Интернет-магазина? На необходимости иметь хорошие каналы связи и аппаратно-программное обеспечение, да и доля доставки в себестоимости существенно возрастает. Проигрывает и на "синдроме недоверия", поскольку в Интернет-торговле покупатель менее защищен от недобросовестного продавца, да и постоянно присутствующий в Интернете хакерский фактор существенно повышает риск сделки.  
 Основные преимущества Интернет-магазинов для потребителей - экономия времени.

Человеку, работающему 6 дней в неделю с 10 до 19, ему бывает некогда сходить в магазин. Интернет-магазин позволяет сделать покупку не выходя из офиса в любое время, а выбор и заказ товара займет у него несколько минут, если он точно знает, что хочет купить. Служба доставки интернет-магазина доставит выбранный товар в удобно время и место. Кроме этого выбор и оценка свойств товара происходит в интернет-магазине намного быстрее и удобнее чем в обычном магазине. Представьте, что вы пришли в обычный магазин бытовой техники и хотите сравнить параметры 20—30 радиотелефонов. Для этого вы должны изучить все ценники: запомнить характеристики, цены и названия моделей. Можно обратиться к продавцу-консультанту, если он не занят, с просьбой рассказать о всех этих товарах. Но обычно ни один продавец не имеет столько свободного времени для работы с одним клиентом, и нет гарантии, что он владеет всей информацией.  
 В Интернет-магазине можно  неограниченно увеличивать ассортимент и информативность.

Кроме того ассортимент интернет-магазина ничем не ограничен (как, например, ассортимент обычного магазина ограничен площадью торгового павильона). А если предусмотрен поиск по параметрам, то Вы просто можете указать характеристики, которым должен соответствовать товар, и затем уже выбирать из списка моделей, удовлетворяющих вашему запросу. И еще одно очень важное замечание — ни один менеджер по продажам не в состоянии помнить столько информации о товаре, сколько может предоставить Вам интернет-магазин. Кроме того, интернет-магазин в состоянии выдать Вам такую информацию, как покупательский рейтинг, советы и отзывы о товаре, статьи о товаре, которые могут предопределить Ваш выбор.   
 Следующий плюс - экономия денег. Затраты на работу интернет-магазина, включая доставку, существенно ниже чем у обычного. В отличие от обычного магазина интернет-магазин способен обслужить несколько сотен клиентов одновременно. Хотя на практике такого не встретишь. Кроме того, если покупатель проживает в другом городе, он получает возможность не тратиться на междугородние телефонные консультации. Всю информацию он может найти на страницах интернет-магазина.

К недостаткам интернет-магазинов для покупателя относится:

* нельзя "пощупать", нельзя узнать больше, чем написано (пример: мебель, одежда);
* проблемы гарантии, сопровождения;
* зачастую долгая доставка (до 6 недель), особенно если мы говорим о международности.

Электронная торговля товарами имеет и недостатки. Так, исследования, проведенные в начале 2000 г. английской фирмой Armor Group, показали, что 35 % дорогих товаров, продающихся через магазины, работающие в Интернет, являются подделками.

Другими словами, например, предлагаемый покупателю «Panasonic» может оказаться не самим «Panasonic» этой известной фирмы, а его азиатской подделкой, произведенной где-нибудь в Юго-Восточной Азии, но украшенной громким товарным знаком. Исследования также показали большую доверчивость покупателей. Например, покупатели легко готовы поверить во всемирную известность несуществующей торговой марки. Так, о знакомстве с никогда не существовавшей торговой маркой «Royal Alliance Insuarance» заявило 80 % опрошенных респондентов. Простота доступа в Интернет, массовость аудитории и потенциальная анонимность рекламодателей делает интернет идеальной платформой для недобросовестных предпринимателей.

## Создание интернет-магазина

В наше время достаточно много веб-студий, занимающихся разработкой сайтов, а так же людей (фрилансеров), которые за определенное вознаграждение напишут интернет-магазин. Также можно воспользоваться Интернет-ресурсами, которые позволяют сконструировать интернет-магазин. И последний способ, все сделать самому.

Рассмотрим достоинства и недостатки данных методов.

Метод первый, заказ у студии.

К достоинствам студии можно отнести:

* Возможность выполнения сайта с уникальным дизайном
* Возможность выполнения сайта с определенным функционалом, ориентированным на нужды клиента
* Более стабильные сроки выполнения заказа
* Техподдержка, по различным вопросам за n-сумму вам ответят на любые вопросы

К недостаткам можно отнести стоимость выполнения заказа, но разобравшись в процессе написания сайта – среднерыночная стоимость достаточно адекватная. Так как студии нужно платить аренду, зарплату специалистам, за электроэнергию, налоги и много чего другого.

Метод второй, заказ у фрилансера.

К достоинствам относится только низкая стоимость, но нужно быть готовым на неопределенные сроки, никакой поддержки, никакой гарантии на уникальность дизайна.

Метод третий, воспользоваться шаблонизатором.

К достоинствам относится бесплатность ресурса. А к недостаткам также отсутствие поддержки, шаблонный дизайн.

Наконец, метод четвертый, сделать интернет-магазин самому.

Недостатком данного способа является то, что нужно знать множество технологий, такие как PHP, AJAX, JavaScript, HTML, CSS, SQL. Достоинства: бесплатность создания магазина, бесплатность поддержки, независимость от других, собственный дизайн и функционал.

Также можно использовать какую-нибудь Систему Управления Сайтом (CMS). Этот подход позволяет сократить количество потраченного времени, написанного кода, потрепанных нервов. Многие CMS предлагают даже избыточный функционал для Вашего сайта. В моей работе используется CMS 1С-Битрикс: Управление Сайтом, редакция старт.

### Рассмотрение функционала CMS 1С-Битрикс: Управление Сайтом

Данная CMS предлагает огромный функционал, от вывода новостей до интеграции с «1С: Предприятие 8.2». Она предоставляет простое, интуитивно понятное управление пользователю, обеспечивает высокую степень защиты веб-безопасности, и сотрудники техподдержки с радостью ответят на Ваши вопросы.

Так как в редакцию старт данной CMS не входит модуль «Интернет-магазин», то мы не можем реализовать магазин.

Таблица 1.1 – Сравнение редакций 1С-Битрикс: Управление сайтом

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | [**Первый сайт**](https://www.1c-bitrix.ru/products/cms/editions/first.php) | [**Старт**](https://www.1c-bitrix.ru/products/cms/editions/start.php) | [**Стандарт**](https://www.1c-bitrix.ru/products/cms/editions/standard.php) | [**Эксперт**](https://www.1c-bitrix.ru/products/cms/editions/expert.php) | [**Малый бизнес**](https://www.1c-bitrix.ru/products/cms/editions/smallbusiness.php) | [**Бизнес**](https://www.1c-bitrix.ru/products/cms/editions/business.php) | [**Веб-кластер**](https://www.1c-bitrix.ru/products/cms/editions/portal.php) |
| [Главный модуль](https://www.1c-bitrix.ru/products/cms/features/main.php) | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | http://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |
| [Управление структурой](https://www.1c-bitrix.ru/products/cms/features/filemanager.php) | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |
| [Медиабиблиотека](https://www.1c-bitrix.ru/products/cms/features/filemanager.php#media) | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |

Продолжение таблицы 1.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Информационные блоки](https://www.1c-bitrix.ru/products/cms/features/infoblocks.php) | 2/2\* | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |
| [Highload Блоки](https://www.1c-bitrix.ru/products/cms/new/new12.php#tab-orm-link!hl) |  | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |
| [SEO-модуль](https://www.1c-bitrix.ru/products/cms/features/seo.php) |  | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |
| [Поиск](https://www.1c-bitrix.ru/products/cms/features/search.php) |  | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |
| [Фотогалерея 2.0](https://www.1c-bitrix.ru/products/cms/features/photo2.php) |  |  | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |
| [База знаний (Wiki)](https://www.1c-bitrix.ru/products/cms/features/wiki.php) |  |  |  | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |  | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |
| [Реклама](https://www.1c-bitrix.ru/products/cms/features/advert.php) |  |  |  | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |  | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |
| [Универсальные списки](https://www.1c-bitrix.ru/products/cms/features/lists.php) |  |  |  |  |  | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |
| [Документооборот](https://www.1c-bitrix.ru/products/cms/features/workflow.php) |  |  |  |  |  | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |
| [**Интернет-магазин**](http://www.1c-bitrix.ru/products/cms/features/eshop.php) |  | | | | **Базовый**[**\*\*\***](https://www.1c-bitrix.ru/products/cms/editions/#restr) | **Расширенный** |  |
| [Конструктор отчетов](https://www.1c-bitrix.ru/products/cms/features/reports.php) |  |  |  |  |  | https://www.1c-bitrix.ru/images/icons/bullet-n.gif | https://www.1c-bitrix.ru/images/icons/bullet-n.gif |

Данная таблица взята с официального сайта компании 1С-Битрикс, из нее видно, что в редакции «Старт» нет модуля «Интернет-магазин». Но, понимая структуру интернет-магазина как набор таблиц где хранятся товары, где хранятся данные о заказах и остальное, и зная, что инфоблоки в 1С-Битрикс – тоже таблицы, можно организовать структуру интернет-магазина при данном функционале этой CMS.

# Теоретическая подготовка

## PHP

PHP (Hypertext Preprocessor) – наиболее простой скриптовый язык программирования, широко применяющийся при создании динамически генерируемых веб-страниц. Основная масса Интернет ресурсов, на данный момент, написана с использованием именно этого языка программирования. При всей своей простоте, PHP позволяет разрабатывать профессиональные веб-проекты любой сложности, от небольших сайтов до крупных порталов.

PHP-код программы выполняется на стороне сервера. После того, как пользователь совершил на сайте некое действие, например клик по ссылке в меню, с целью перейти на другую страницу сайта, браузер посылает запрос серверу на соответствующую страницу с PHP-кодом. Далее, PHP-код обрабатывается интерпретатором PHP и генерируется HTML-код, который возвращается серверу. Сервер в свою очередь, передаёт этот HTML-код обратно браузеру. В результате пользователь видит отображение в браузере новой страницы, имеющей свой HTML-код. При просмотре же исходного кода этой страницы виден будет только HTML-код, а PHP-код остается недоступен для просмотра.

Большой плюс языка PHP состоит в том, что PHP-код можно внедрять непосредственно в HTML-файлы. PHP-код встраивается в HTML-страницы при помощи угловых скобок и знака вопроса: Сами же файлы, в которых присутствует PHP-код, имеют расширение \*\*\*.php.

Для создания веб-проектов на языке php, необходимо программировать, используя либо установленный локальный сервер на своём компьютере, либо работая с помощью удалённого сервера. Удалённый сервер не всегда удобен, да и, как правило, за это надо платить. Для создания сервера на своём компьютере понадобятся следующие программы: Apache или Nginx (сервер), MySQL (базы данных), PHP. Все программы актуальных версий можно найти на официальных сайтах разработчиков. Создание PHP-файлов, написание кода и работа с ним ничем не отличается от того же процесса, что и при работе с HTML.

Работать с php-кодом можно также в обычном текстовом редакторе, но делать это с помощью php-редактора намного удобнее. Можно в принципе обойти процесс установки серверных программ на компьютер и долгой их настройки. Это возможно при использовании уже готовых сборок в виде одной целой программы. Например, Top Server, и подобные программы. Установив такую программу на компьютер, вы получаете уже готовый набор всех серверных программ, с отдельной панелью управления ими. Такой вариант идеален для создания и разработки сайтов на компьютере и проверки их в «живую». Для использования компьютера в качестве сервера такой вариант, конечно, использовать нельзя. Для тех, кто только начинает знакомиться с языком программирования PHP и не имеет особого навыка в программировании, кому, нужно лишь только написав некоторый код проверить его в действии, такой вариант будет самым удобным.

## HTML+CSS

HTML (Hyper Text Markup Language) означает язык разметки гипертекста. Этот язык был разработан Тимом Бернерсом-Ли в рамках создания проекта распределенной гипертекстовой системы, которую он назвал World Wide Web (WWW) или Всемирная паутина. HTML предназначен для написания гипертекстовых документов, публикуемых в World Wide Web. Документ на языке HTML может включать следующие компоненты:

* стилизованный и форматированный текст;
* команды включения графических и звуковых файлов;
* гиперсвязи с различными ресурсами Internet;
* скрипты на языке JavaScript и VBScript;
* различные объекты, например Flash-анимацию.

Документы HTML являются обычными текстовыми файлами, содержащими специальные теги (или управляющие элементы) разметки. Теги разметки указывают браузеру Web, как надо вывести страницу.  
В HTML-документах то, что мы обычно видим на страничках, перемежается невидимым для просмотра из браузера кодом. Вот этот-то невидимый код и есть язык разметки HTML. 

HTML — это не язык программирования, — он служит лишь для разметки странички, придания определенного вида тому или иному элементу, будь то таблица, текст или картинки. 

Осуществляется это путем присвоения каждому элементу своих параметров, которые распознает браузер. Параметры эти могут быть заданы как для одного, так и для группы или типа элементов. Тип элементов может быть таким: таблицы, ячейки, ссылки, текст и т.п. То есть что-то, что можно назвать одним термином. Отдельные свойства можно присваивать и выбранным элементам персонально. 

Параметры отображения элементов задаются при помощи тегов, в которых и задается желаемый вид того или иного элемента нашей странички.

CSS (Cascading Style Sheets- каскадные таблицы стилей) - технология, использующаяся для оформления веб-страниц с помощью стилей.

Основная идея CSS состоит в том, чтобы разделить структуру и содержание веб страниц от их оформления:

* c помощью HTML создается страница - ее каркас, а также вносится ее содержимое.
* с помощью CSS создается дизайн - задается оформление для элементов страницы.

Стили - это набор  правил форматирования, применяемых к элементам страницы для изменения их свойств и параметров представления на странице: в основном это касается их внешнего вида и занимаемого положения.

Стили представлены свойствами CSS. Каждое свойство по своему влияет на оформление того или иного элемента страницы. Некоторые свойства не применимы к некоторым элементам.

Элементы страницы - это теги HTML или их содержимое.

## jQuery

jQuery— это JavaScript-библиотека, обеспечивающая кроссбраузерную поддержку приложений (работает в Internet Explorer 6.0+, Mozilla Firefox 2+, Safari 3.0+, Opera 9.0+ и Chrome).

Автор библиотеки Джон Резиг (John Resig) впервые представил свое творение в январе 2006 года на компьютерной конференции в Нью-Йорке, а в августе того же года была выпущена первая стабильная версия библиотеки.

За прошедшие годы библиотека претерпела множество изменений и на текущий день содержит функционал, полезный для максимально широкого круга задач. Она имеет небольшой размер (например, размер минимизированной версии— 55,9 Кбайт, а сжатой версии— 19 Кбайт) и не засоряет глобальное пространство имен тривиальными идентификаторами.

Потрясающие возможности механизма селекторов, позволяющие легко получить доступ к любому элементу объектной модели документа, сделали библиотеку jQuery очень популярной.

Судите сами. Чтобы получить ссылку на DOM-элемент с помощью Javascript, обычно используется метод getElementByld (). Например, изменим HTML-код элемента с идентификатором divl.

Листинг 2.1 – Доступ к DOM-элементу средствами JavaScript

|  |
| --- |
| Document.getElementByld (“divl”).innerHTML = "Новый текст"; |

Код на jQuery, выполняющий то же самое действие, будет в два раза короче.

Листинг 2.2 – Доступ к DOM-элементу средствами jQuery

|  |
| --- |
| $("#divl").html("Новый текст"); |

Конечно, ради одной этой строки не имеет смысла подключать целую библиотеку. Но все дело в том, что функционал селекторов далеко не ограничивается одним идентификатором. Возможности селекторов можно сравнить разве что с регулярными выражениями языка Perl. В качестве примера изменим цвет текста во всех элементах А, в параметре href которых содержится ссылка на HTML-документ, причем элемент А должен быть расположен внутри элемента DIV, имеющего стилевой класс clsl.

Листинг 2.3 – Простое использование селекторов в jQuery

|  |
| --- |
| $("div.clsl a[href$=`.html`]").css("color", "red"); |

Попробуйте выполнить аналогичную операцию с помощью JavaScript. Для этого понадобится далеко не одна строка кода.

Еще одной отличительной особенностью библиотеки jQuery является возможность составлять цепочки из вызовов методов, так как большинство методов jQuery возвращает объект, с которым можно производить дальнейшие манипуляции.

Листинг 2.4 – Составление цепочки из выводов методов

|  |
| --- |
| $("#message") // Получили ссылку на элемент с id=message.html("Сообщение") // Изменили текст внутри элемента.parent() // Получили ссылку на родительский элемент.ess("background-color", "#fff4dd") // Задалицветфона.width(300) // Ширина.height(200) // Высота// Плавно отобразили элемент за счет изменения прозрачности.fadeln(3000); |

Библиотека jQuery не оставила без внимания и технологию AJAX, позволяющую обмениваться данными с сервером без перезагрузки веб-страницы.

Благодаря своей универсальности библиотека jQuery будет полезна практически любому разработчику. Она подходит новичку, так как позволяет забыть о проблеме с кроссбраузерностью приложения (вышел новый веб-браузер— сменил версию библиотеки и все опять работает). Библиотека идеальна для профессионалов, так как позволяет сократить код минимум в три раза. А это в свою очередь позволит написать очень сложный код с минимальными усилиями и потерей времени.

# Программная часть

Для написания Интернет-магазина на 1С-Битрикс: Управление сайтом редакции «Старт» нужно было разработать несколько своих компонентов, которые будут выводить товар, класть в корзину и оформлять заказы. Каждый элемент можно было разбить на несколько компонентов, отвечающих за свой функционал: вывод всех разделов каталога, вывод одной секции детально, вывод товаров секции или каталога, вывод детально товара, вывод общей суммы заказов, вывод полной корзины, оформление заказа.

Так как хотелось написать качественный сайт, было принято решение добавить на сайт разделы «Статьи», «Новости», выполненные стандартными компонентами Битрикса, раздел «Отзывы», где сами отзывы тоже есть стандартный функционал Битрикса, но форма добавления – авторский компонент. Также организовал вывод просмотренных ранее товаров.

## Создание простого компонента для Битрикса

Для создания компонента лучше сразу определить свою область видимости. Например, в папке /bitrix/components/ была создана папка DmitryTelepnev, чтобы хранить там все свои компоненты.

Создаем папку с именем компонента. Обычно структура папок и файлов компонента выглядит так:

* /help
* /images
* /lang
* /templates
* .description.php
* .parameters.php
* component.php

Обязательными из них являются /templates, .description.php, .parameters.php, component.php.

 В папке templates лежат все шаблоны компонента. Можно сразу сделать дефолтных шаблон для нашего компонета. Полный путь до него будет выглядеть так:

bitrix/components/DmitryTelepnev/sections.list/templates/.default/template.php.

Файл .description.php отвечает за описание компонента и отображение компонента в списке компонентов в визуальном редакторе.

Файл .parameters.php отвечает за набор параметров, с которыми будет вызван наш компонент.

Файл component.php является головным файлом, он отвечает за всю логику.

Листинг 3.1 – Пример файла .description.php

|  |
| --- |
| <?  if (!defined("B\_PROLOG\_INCLUDED") || B\_PROLOG\_INCLUDED!==true) die();  $arComponentDescription = array(  "NAME" => 'Prmedia: список разделов инфоблока',  "DESCRIPTION" => 'Список разделов инфоблока',  "ICON" => "/images/sections\_list.gif",  "CACHE\_PATH" => "Y",  "PATH" => array(  "ID" => "utility",  ),  );  ?> |

В этом файле просто содержится массив.

Листинг 3.2 – Пример файла .parameters.php

|  |
| --- |
| <? if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  $arComponentParameters = array(  'PARAMETERS' => array(  'IBLOCK\_ID' => array(  'NAME' => 'Id инфоблока',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'CACHE\_TIME' => array('DEFAULT'=>3600),  ),  );  ?> |

В этом файле также массив, ключи которого являются параметрами, а значения в свою очередь являются массивами со свойствами данных параметров.

Ключ PARAMETERS массива $arComponentParameters представляет собой массив с описанием параметров. Пусть у компонента будет только один параметр - IBLOCK\_ID (ID инфоблока, из которого мы будем выводить разделы). Ключи NAME - название параметра, TYPE - тип, MULTIPLE - множественность (если 'Y', то наш параметр сможет принимать массив значений), PARENT - родитель (параметры можно делить на группы. В этой статье не рассматривается). Параметр CACHE\_TIME - параметр по умолчанию для компонентов Битрикса и устанавливает время кеширования.

Из массива $arComponentParameters будет сгенерирован массив $arParams, который будет использоваться в основном файле компонента - component.php

Листинг 3.3 – Содержание файла component.php

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  // echo '<pre>'; print\_r($arParams); echo '</pre>';  CModule::IncludeModule('iblock');  if ($this->StartResultCache(3600))  {  $iblock\_id = $arParams['IBLOCK\_ID'];  $arFilter = array('IBLOCK\_ID'=>$iblock\_id);  $db\_list = CIBlockSection::GetList(array('NAME'=>'ASC'), $arFilter, true, array("ID", "NAME", "CODE"));  while($ar\_result = $db\_list->GetNext())  {  $arResult[] = array(  "ID" => $ar\_result['ID'],  "CODE" => $ar\_result['CODE'],  "NAME" => $ar\_result['NAME'],  "ELEMENT\_CNT" => $ar\_result['ELEMENT\_CNT']  );  }  // echo '<pre>'; print\_r($arResult); echo '</pre>';  $this->IncludeComponentTemplate();  }  ?> |

В файле component.php содержится вся логика компонента. Задача этого файла - сформировать из полученных параметров ($arParams) массив $arResult, который впоследствии попадет в шаблон компонента.

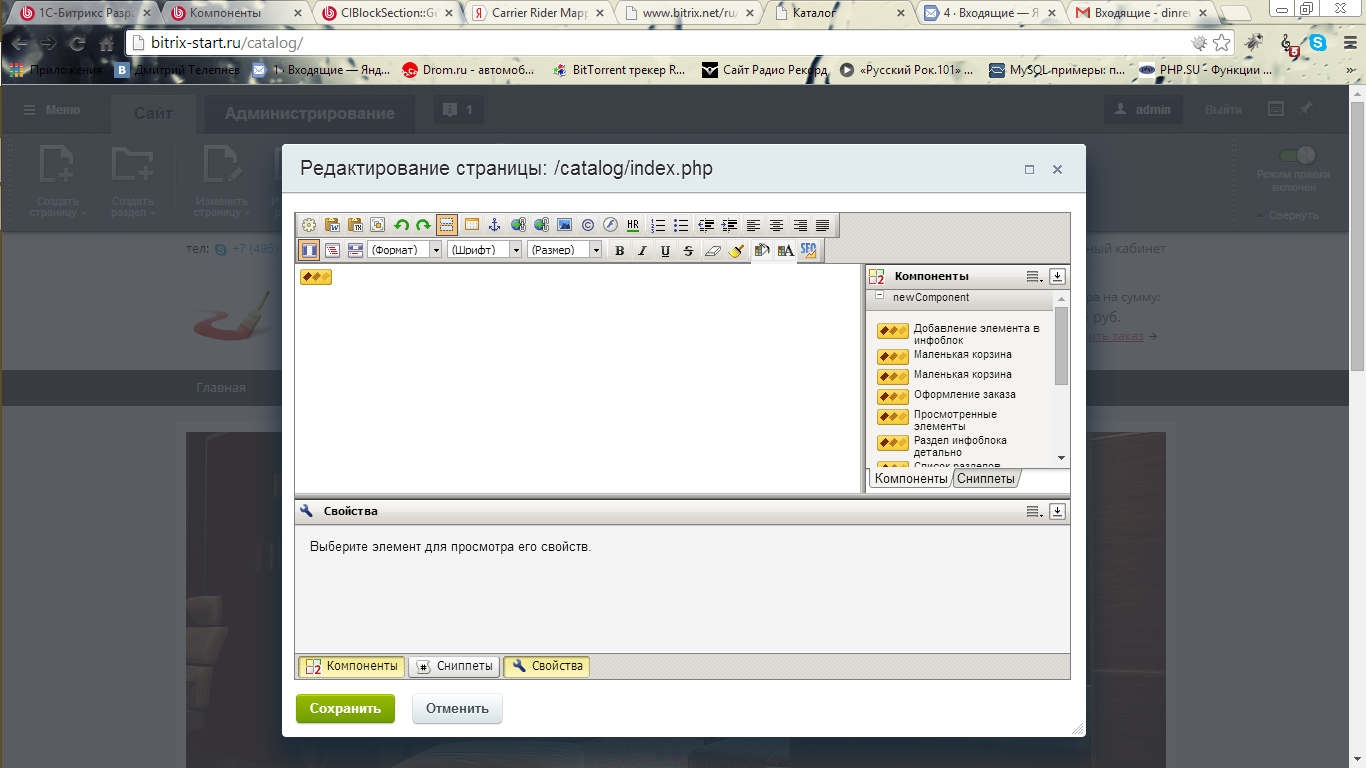
Сам код компонента не сложен - в зависимости от полученого параметра (ID инфоблока) делается выборка разделов и сохраняется в массив $arResult. Следует обратить внимание на вызов метода $this->StartResultCache(). Он проверяет, есть ли у компонента актуальный кеш. Если он есть, то выводится информация из кеша. Следовательно, запроса к базе нету, массив $arResult не генерируется, и даже шаблон не подлючается (метод $this->IncludeComponentTemplate()).

Листинг 3.4 – Содержимое шаблона template.php

|  |
| --- |
| <h2>Рубрики блога</h2>  <ul>  <? foreach($arResult as $category): ?>  <li><a href="/posts/<?=$category['CODE']?>/" title="<?=$category['NAME']?>"><?=$category['NAME']?> (<?=  $category['ELEMENT\_CNT']?>)</a></li>  <? endforeach; ?>  </ul> |

Шаблон принимает массив $arResult, сгенерированный в файле component.php и выводит его содержимое в браузер.

Все, компонент готов, осталось вызвать его на нужной странице, как показано на рисунке 3.1.



3

22

1

Рисунок 3.1 – Порядок вызова и настройки компонента

В визуальном редакторе открываем страницу и помещаем на нее компонент (действия 1-2), производим настройку параметров в графическом интерфейсе (действие 3), и все.

## Написание логики компонентов

### Выборка разделов из инфоблока CIBlockSection::GetList

Практически все описанные компоненты строятся на выборке значений из базы данных.

Листинг 3.5 – Фрагмент кода из компонента section.list

|  |
| --- |
| $arResult = array();  $arRes = array();  $db\_list = CIBlockSection::GetList($arOrder, $arFilter, true);  while($ar\_result = $db\_list->GetNext())  {  $arRes[] = $ar\_result;  } |

Здесь представлена стандартная выборка разделов из инфоблока средствами Битрикса. Рассмотрим подробнее. В документации сказано: «Возвращает список разделов, отсортированных в порядке*arOrder* по фильтру *arFilter*.». Параметр arOrder отвечает за сортировку, arFilter за выборку нужных нам параметров (Фильтрацию). Также у метода есть еще параметр bIncCnt, который указывает методу, возвращать ли нам количество элементов в разделе, и параметр arSelect, который отвечает за то, что нам нужно выбрать.

Каждый из параметров, кроме bIncCnt – это массив вида “КЛЮЧ” => “ЗНАЧЕНИЕ”. Для массива arOrder ключ представляет поле, по которому нужно сортировать, а значение поля – сортировка по убыванию или по возрастанию (ASC or DESC). Все поля можно найти в документации Битрикса.

Для массива arFilter ключ представляет поле, которое нужно выбрать, а значение – условие, которому поле должно удовлетворять. Этот массив также поддерживает написание сложных логических фильтров.

Массивы arSelect и arNavStartParams отсутствуют в запросе, так как нам нужно выбрать все поля из таблицы базы данных. Вкратце, в массиве arSelect ключом является поле, которое нам нужно выбрать, если массива с полями нет – то по умолчанию выбираются все поля.

### Добавление товаров в корзину и в категорию просмотренных через технологию сессий

Добавление товаров в корзину и в категорию просмотренных было решино сделать через технологию сессий в php. Сессии предназначены для хранения сведений о пользователях при переходах между несколькими страницами. При использовании сессий данные сохраняются во временных файлах на сервере.

При работе с сессиями различают следующие этапы:

* открытие сессии
* регистрация переменных сессии и их использование
* закрытие сессии

Так как в Битриксе используются сессии в самом ядре, то нам

понадобится только этап 2: регистрация переменных сессии и их использование.

Листинг 3.6 – Фрагмент кода добавления ID товара в корзину и в категорию просмотренных

|  |
| --- |
| if($arParams["ADD\_TO\_BASKET"]=="Y") {  $arResult["FORM\_DATA"] = array();  if($\_SERVER["REQUEST\_METHOD"]=="POST" && $\_POST["guard"] == "yes") {  //debug($\_POST);  $\_SESSION["ADDED\_ITEMS"][] = array("ID"=>$\_POST["ITEM"],"IBLOCK\_ID"=>$\_POST["IBLOCK\_ID"]);  }  }  if($arParams["ADD\_TO\_VISITED"]=="Y") {  $\_SESSION["VISITED"][] = array("ID"=>$arResult["ID"], "IBLOCK\_ID"=>$arResult["IBLOCK\_ID"]);  //debug($\_SESSION["VISITED"]);  } |

В данном примере если в настройке компонента разрешено добавление товара в корзину и в просмотренные товары (соответственно, ($arParams["ADD\_TO\_BASKET"] == "Y" и $arParams["ADD\_TO\_VISITED"] =="Y"), то мы заполняем свой определенный ключ массивом из ID товаров. А далее в процессе мы их просто выбираем из базы данных выборкой CIblockElement::GetList.

### Выборка элементов инфоблока CIblockElement::GetList

Выборка элементов инфоблока выполняется аналогично выборке разделов инфоблока, описанная в главе 2.2.1.

Листинг 3.7 – Пример выборки элементов инфоблока

|  |
| --- |
| $res = CIBlockElement::GetList(Array(), $arFilter, false, Array("nPageSize"=>$arParams["COUNT\_SECTION\_ELEMENT"]));  while($ob = $res->GetNext())  {  $arItems[] = $ob;  } |

Параметры, принимаемые данным методом, не сильно отличаются от CIblockSection::GetList. Помимо тех, что присутствуют в методе выбора разделов, есть еще массив для группировки элементов arGroupBy.

Для реализации определенного массива arResult иногда приходилось пользоваться условиями, в котором находились определенные фильтры для методов. Иногда приходилось из результата выборки приходилось формировать массив другого вида, для удобства применения его в шаблоне template.php.

### Добавление элемента с определенными свойствами в инфоблок

Добавление элемента используется в компоненте добавления отзывы и в компоненте оформления заказа. Заказы у нас хранятся в отдельном инфоблоке, у элемента которого название «Заказ покупателя … от #ДАТА#», в дополнительных полях хранится телефон, e-mail, адрес доставки, и купленные товары. Купленные товары реализованы привязкой по ID товара инфоблока «Каталог».

Листинг 3.8 – Процесс добавления элемента в инфоблок на примере оформления заказа

|  |
| --- |
| $el = new CIBlockElement;  $PROP = array();  $arItemsId = array();  foreach($\_SESSION["BUY\_ITEMS"] as $tmp) {  $arItemsId[] = $tmp["ID"];  }  // debug($arItemsId);  $PROP = array(  $arParams["USER\_NAME"]=>$\_POST["name"],  $arParams["USER\_PHONE"]=>$\_POST["phone"],  $arParams["USER\_EMAIL"]=>$\_POST["email"],  $arParams["USER\_ADDRESS"]=>$\_POST["address"],  $arParams["TOTAL\_PRICE"]=>"",  $arParams["PURCHASED\_GOODS"]=>$arItemsId,);  $arProps = Array(  "MODIFIED\_BY" => $USER->GetID(), // элемент изменен текущим пользователем  "IBLOCK\_SECTION\_ID" => false, // элемент лежит в корне раздела  "IBLOCK\_ID" => $arParams["IBLOCK\_ID"],  "PROPERTY\_VALUES"=> $PROP,  "NAME" => "Заказ от ".date("d.m.Y H:i")." покупателя ".$\_POST["name"],  );  // debug($arProps);  if($PRODUCT\_ID = $el->Add($arProps)) {  unset($\_SESSION["BUY\_ITEMS"]);  unset($\_SESSION["ADDED\_ITEMS"]);  $arEventFields = array(  "USER\_NAME"=>$\_POST["name"],  "USER\_PHONE"=>$\_POST["phone"],  "USER\_EMAIL"=>$\_POST["email"],  "USER\_ADDRESS"=>$\_POST["address"],  "TOTAL\_PRICE"=>"",  "PURCHASED\_GOODS"=>$arItemsId,  );  CEvent::Send("PURCHASE", SITE\_ID, $arEventFields);  LocalRedirect($arParams["LINK\_SUCCESS"]); |
|  |

Здесь создали объект класса CIBlockElement, сформировали массив PROP, который представляет собой набор значений полей элемента инфоблока, определили в какой инфоблок будем добавлять элемент и вызвали метод CIblockElement::Add, передав ему полученный массив. Метод CEvent::Send позволил нам отослать письмо на почту покупателя с уведомлением о покупке. Передаваемые параметры были название шаблона письма, ID сайта и массив данных, которые будут вставлены в письмо.

Функция LocalRedirect позволила перенаправить пользователя на страницу об успешном оформлении заказа.

### Функция валидации почты

Функция валидации почты – это функция, написанная мной с целью проверить поступающую мне строку на то, похож ли синтаксис ее на синтаксис электронной почты. К примеру, строка «asdwweracx» не может быть почтой, а строка «asda@xzvc.ru» - может.

Листинг 3.9 – Функция валидации e-mail

|  |
| --- |
| function \_\_checkEmail($email)  {  if(!preg\_match('|([a-z0-9\_\.\-]{1,20})@([a-z0-9\.\-]{1,20})\.([a-z]{2,4})|is', $email)) {  //echo "false";  return FALSE;  } else {  return true;  }  } |

Основным элементом, отвечающим за «шаблон» строки является регулярное выражение '|([a-z0-9\_\.\-]{1,20})@([a-z0-9\.\-]{1,20})\.([a-z]{2,4})|is'. Используя функцию PHP preq\_match() проверяем строку на соответствие шаблону и возвращаем результат true или false.

# АДМИнистративная часть

## Установка 1С-Битрикс: Управление сайтом.

Для установки Битрикса на локальный компьютер требуется локальный сервер – ПО, которое может эмулировать сервер на Вашем компьютере сервера, наличие сервера БД, наличие PHP и его следующих расширений:

* GD – библиотека для работы с изображениями, требуется для построения графиков и диаграмм для модулей статистики, рекламы, техподдержки. Используется для работы механизма CAPTCHA.
* PHP XML – используется для работы системы обновлений. Библиотека по умолчанию включена в стандартной установке PHP. Для версии под Windows - поддержка встроенная.
* FreeType – библиотека необходима для корректной работы механизма CAPTCHA.
* Поддержка регулярных выражений (POSIX и Perl-compatible) – необходима для корректной работы внутренних механизмов продукта.
* Zlib compression – библиотека компрессии используется для работы модуля Компрессии и при работе системы обновлений для уменьшения количества передаваемых данных от сервера к клиенту.

Далее есть несколько вариантов установки. Можно скачать с официального сайта ПО, запустив которое на своем компьютере, вы установите Битрикс, или скачать скрипт на языке PHP, положив который в корень сайта и запустив в браузере, вы запустите установку CMS.

Откройте страницу http://<ваш сайт>/bitrixsetup.php в браузере, заменив строку <ваш сайт> на реальный адрес вашего сайта. В браузере отобразится страница загрузки дистрибутива, изображенная на рисунке 4.1.

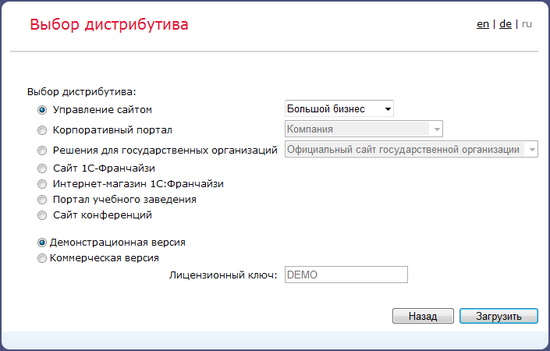


Рисунок 4.1 – Окно выбора дистрибутива

В поле Выбор дистрибутива выберите Управление сайтом и с помощью выпадающего списка укажите редакцию продукта, которую вы хотите установить.

Укажите версию продукта, которая вам необходима: демонстрационная или коммерческая.

Если вы выбрали коммерческую версию, то введите лицензионный ключ в поле Лицензионный ключ.

Примечание: Установка коммерческой версии без лицензионного ключа невозможна. Установка демоверсии возможна как с демонстрационным ключом, полученным на сайте компании «1С-Битрикс», так и без ключа вообще.

Нажмите кнопку Загрузить. Начнется процесс загрузки дистрибутива на сайт, рисунок 4.2.

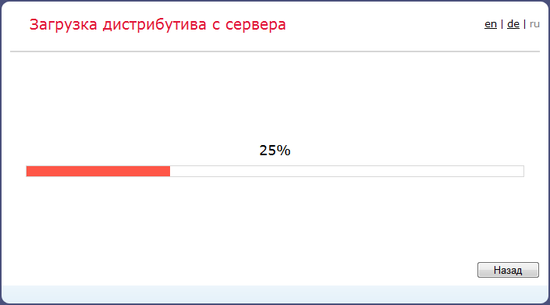


Рисунок 4.2 – Окно процесса загрузки дистрибутива

С помощью скрипта **BitrixSetup** будет установлено соединение вашего сервера непосредственно с сервером компании «1C-Битрикс». Дистрибутив выбранной редакции продукта будет скопирован в корневую директорию сайта на сервере, автоматически распакован, а затем в браузере откроется окно Мастера установки продукта (например «1C-Битрикс: Управление сайтом» или «Битрикс24 в коробке»).

С помощью кнопки Назад можно вернуться в раздел Выбор дистрибутива, чтобы изменить значения параметров загрузки (например, редакцию дистрибутива).

## Включение шаблона

После успешной установки вы увидите свой сайт. Если Вам требуется поменять шаблон, сделать это можно в Административном разделе сайта. Перейдя по ссылке «Администрирование» в верхней части сайта, нужно зайти в Настройки\Настройки продукта\Сайты\Список сайтов. Появится список сайтов. Здесь нужно выбрать свой сайт, выбирая по активности и названию. Зайти в этот сайт и произвести настройку, изображенную на рисунке 4.3.

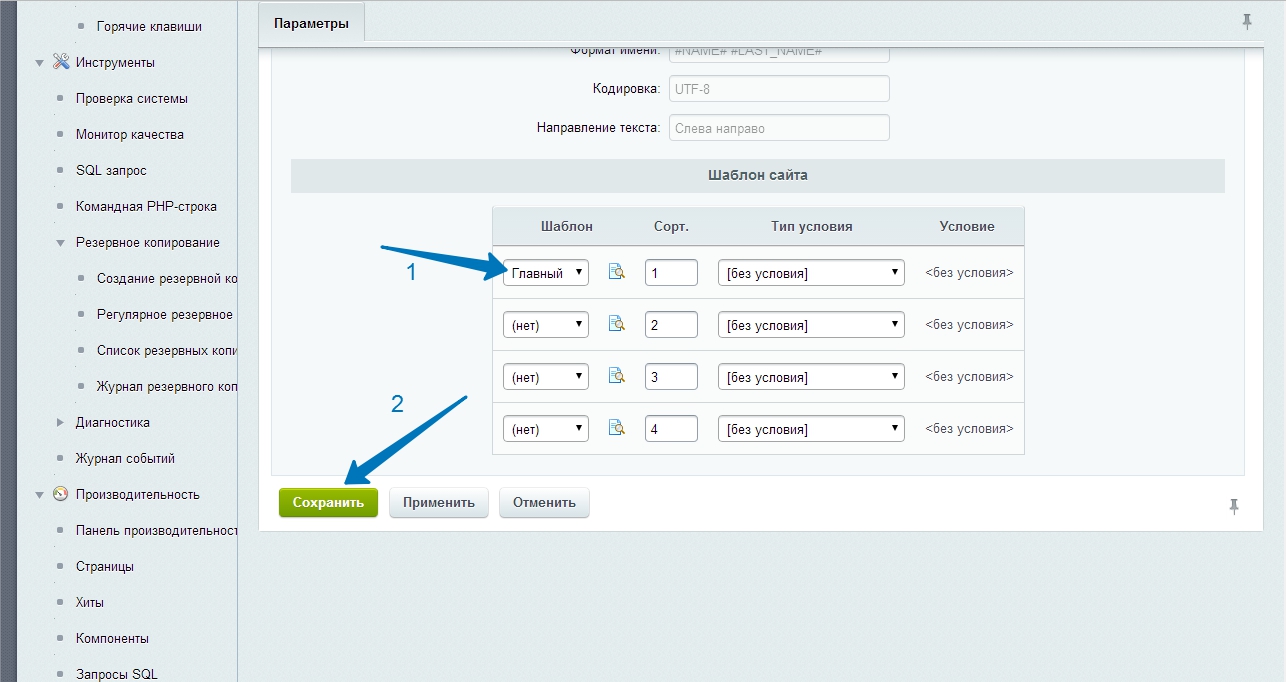


Рисунок 4.3 – Окно настройки сайта

Нажав на поле шаблон будет выдан список имеющихся на сайте шаблонов, выбрав один из которых вы сможете его применить, нажав кнопку или «Сохранить», или «Применить».

## Настройка компонентов

Для быстрого размещения компонентов на странице разработан интерфейс на базе визуального HTML-редактора, рисунок 4.4.

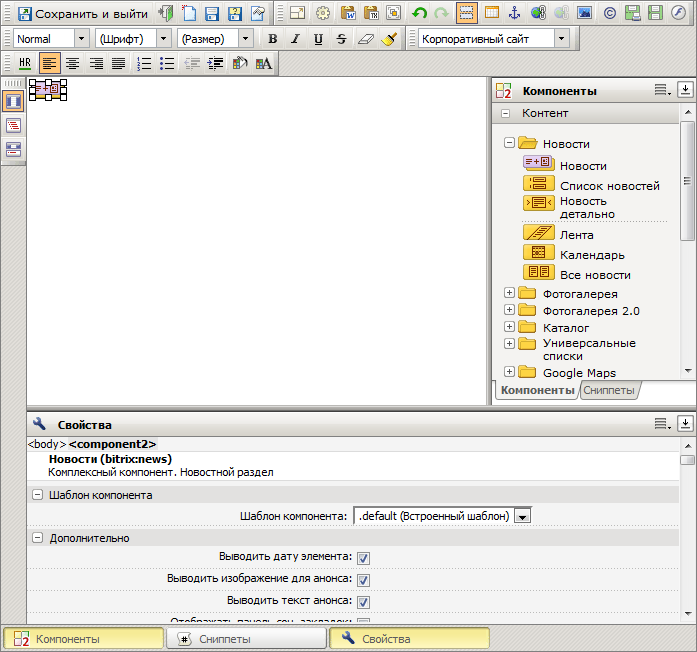


Рисунок 4.4 – Окно быстрого размещения компонента

Все визуальные компоненты расположены на специальной панели компонентов. Отображение этой панели зависит от настроек визуального редактора (переход к настройкам осуществляется при помощи кнопки http://dev.1c-bitrix.ru/images/content_manager/components_2/but_vis_editor_set.png).

Нажав кнопку «Сохранить и выйти», в следующий раз можно обратиться к настройке компонента через Панель управления компонентом, выделенной красным на рисунке 4.5.

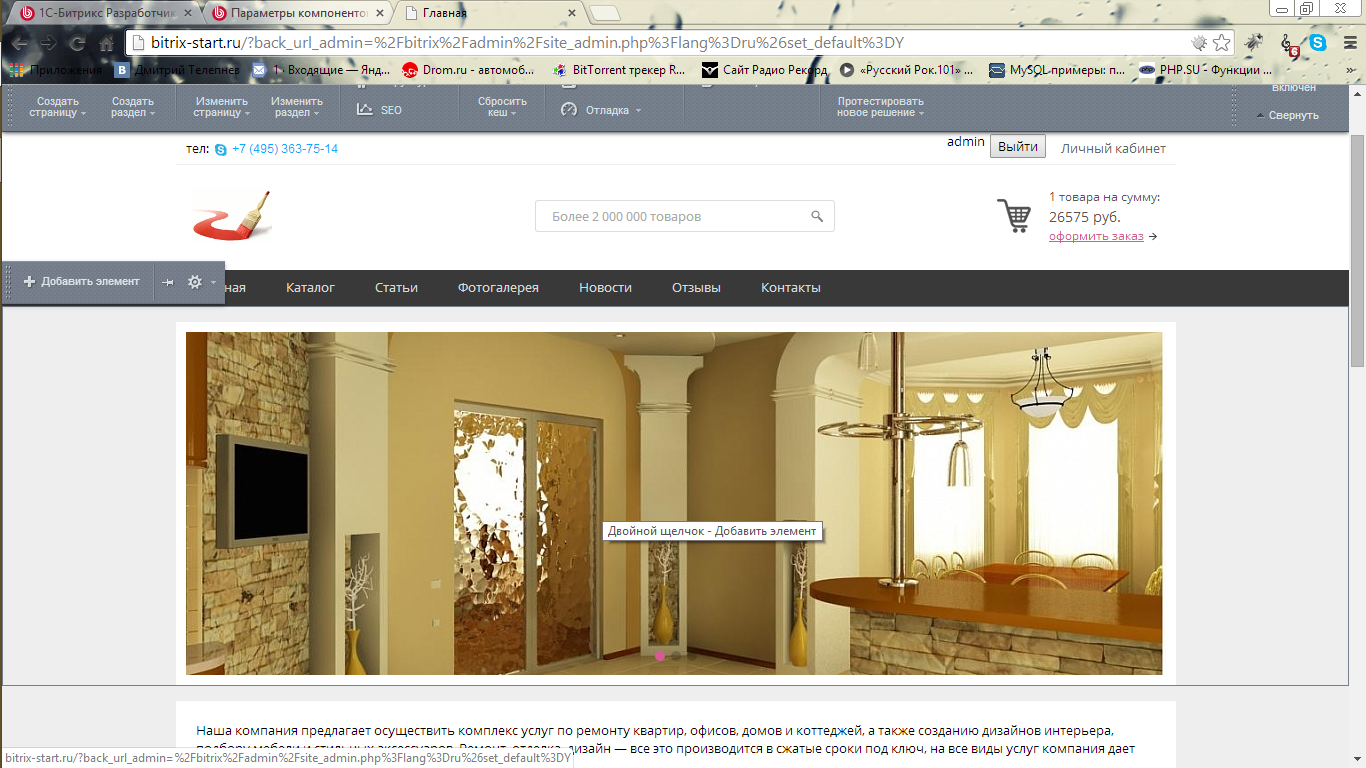


Рисунок 4.5 – Изображение панели управления компонентом

Кликом на шестеренку вызывается окно настройки параметров компонента, рисунок 4.6.

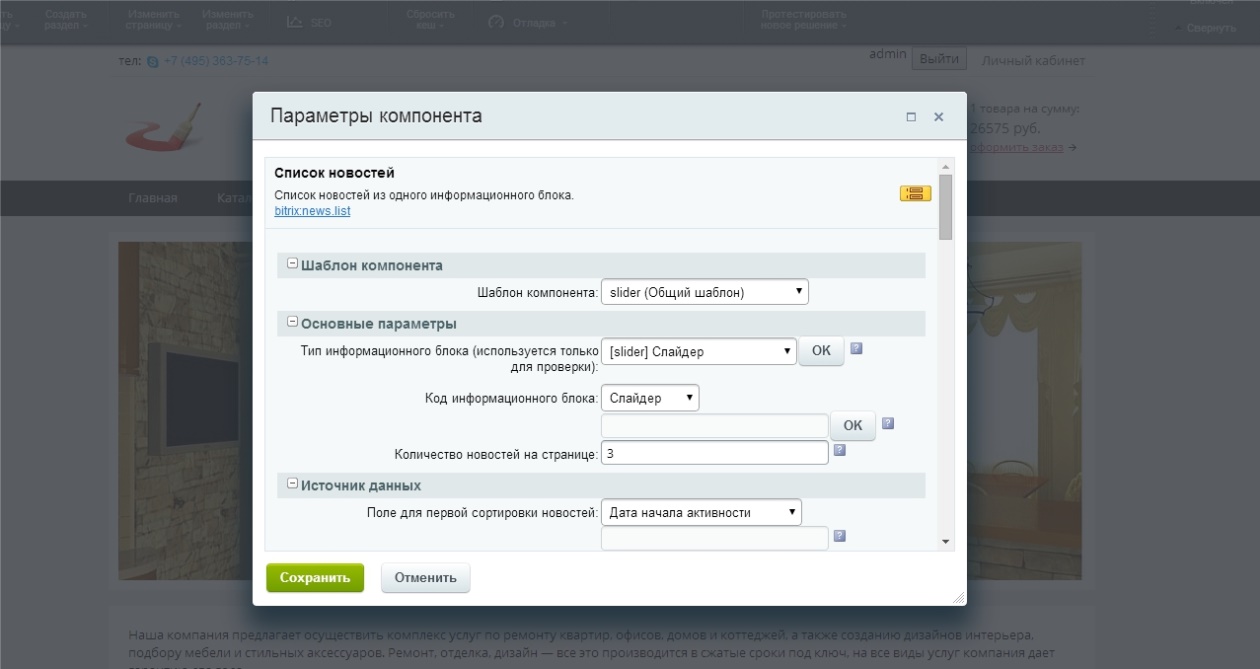


Рисунок 4.6 – Окно настройки параметров компонента

Здесь можно выбрать вид отображения информации – шаблон компонента, в основных параметрах выбирается источник данных (откуда будут приходить данные, инфоблок и тип инфоблока). Также можно настроить сортировку, количество выведенных элементов и т.д.

Соответственно, все настройки сайта и его компонентов доступны только администратору сайта и кругу лиц, имеющих право редактировать сайт.

# Пользовательская часть

Так как работа больше ориентирована на клиента, то есть на пользователя, то рассмотрим интерфейс получившегося сайта, его достоинства.

Пользовательский интерфейс получился довольно простым и интуитивно понятным. Это хорошо для владельца сайта, так как клиент придет и купит товар.

Верхняя часть главной страницы изображена на рисунке 5.1.

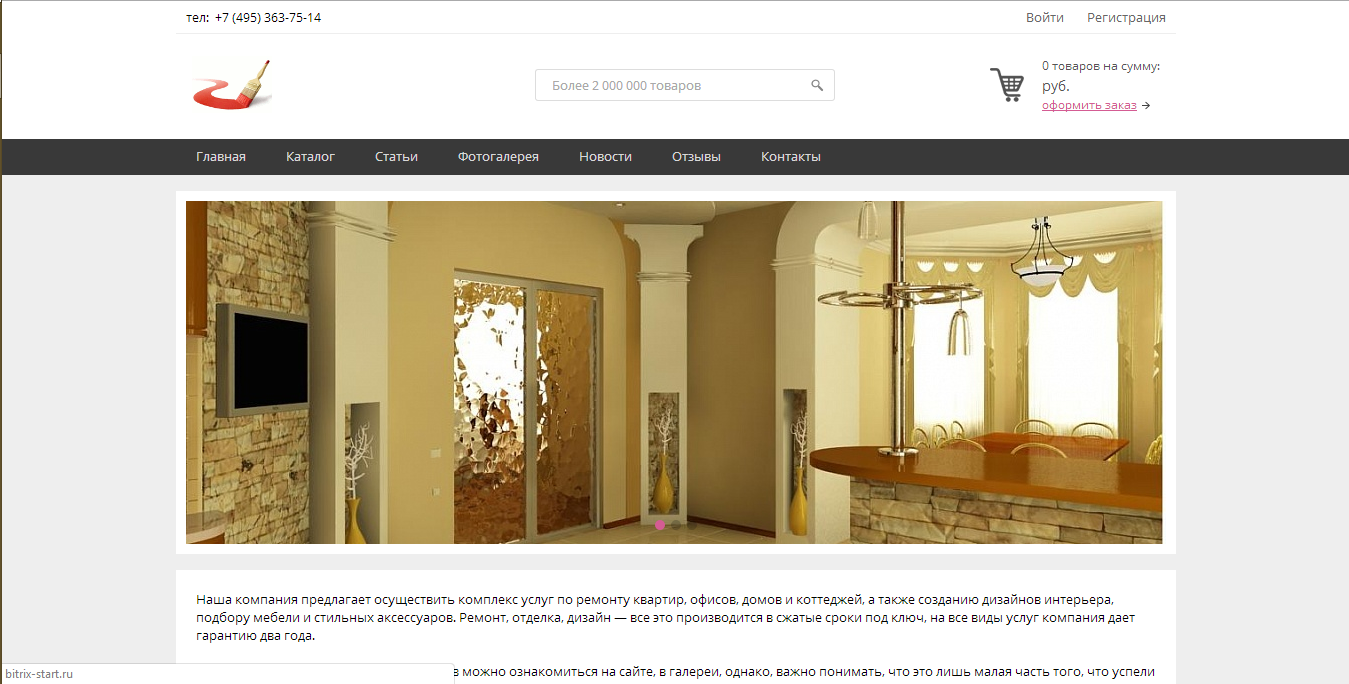


Рисунок 5.1 – Верх главной страницы

На главной странице из функционала расположились поиск по сайту, ссылки на вход и регистрацию, маленькая корзина, оповещающая о количестве товара в корзине и общей сумме. Под ними расположилось главное меню и ниже слайдер.

Нижняя часть главной страницы информативная. Она показана на рисунке 5.2.

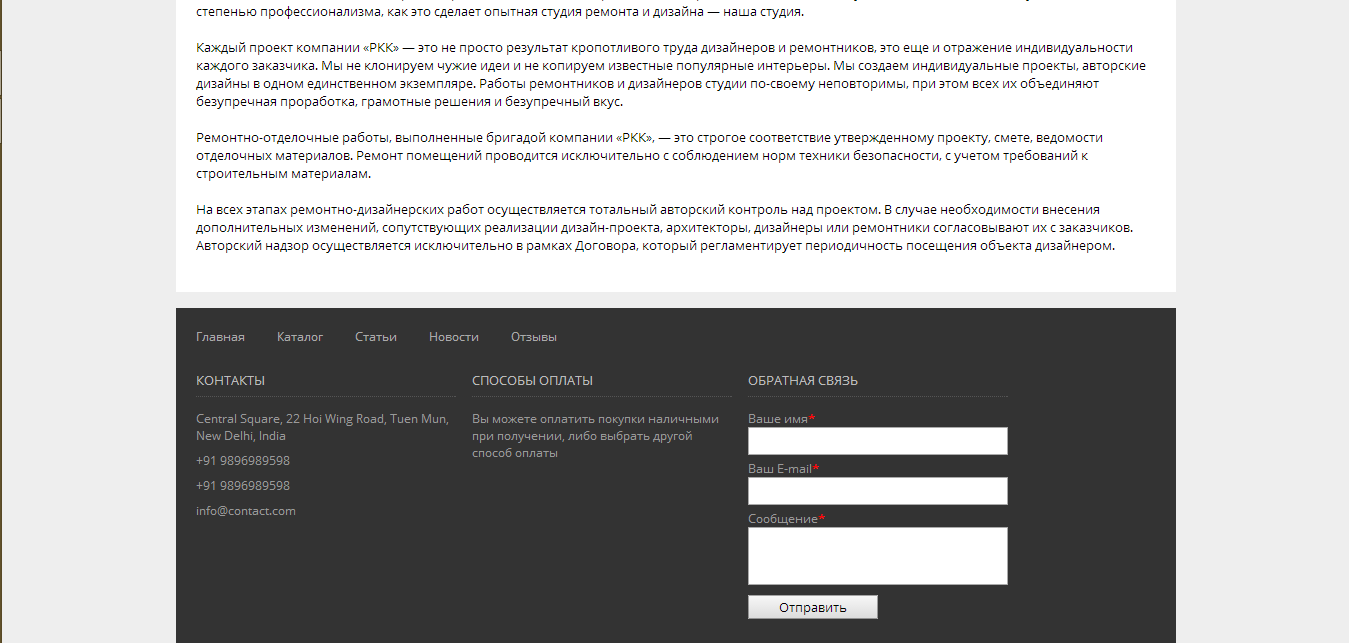


Рисунок 5.2 – Нижняя часть главной страницы

Здесь находится также дублер главного меню, краткая информация о контактах, способах оплаты и форма обратной связи. Последняя нужна для связи клиента и администратора, чтобы можно было оперативно получить интересующую клиента информацию.

Страница контакты выполнена с помощью карты сайта Google, так как она более удобна в использовании. Карта представлена на рисунке 5.3.

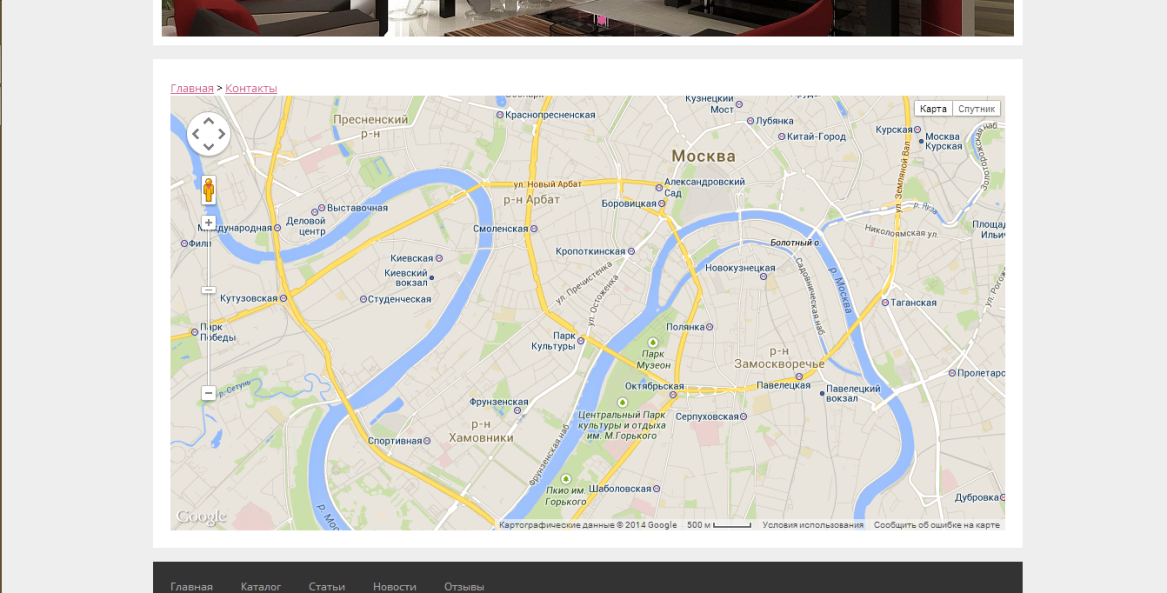


Рисунок 5.3 – Страница контактов

На сайте выполнены разделы новости и статьи. Это хорошо тем, что новости оповещают клиента о событиях компании, и, тем самым сближают клиента и продавца, позволяя клиенту легче расставаться с деньгами.

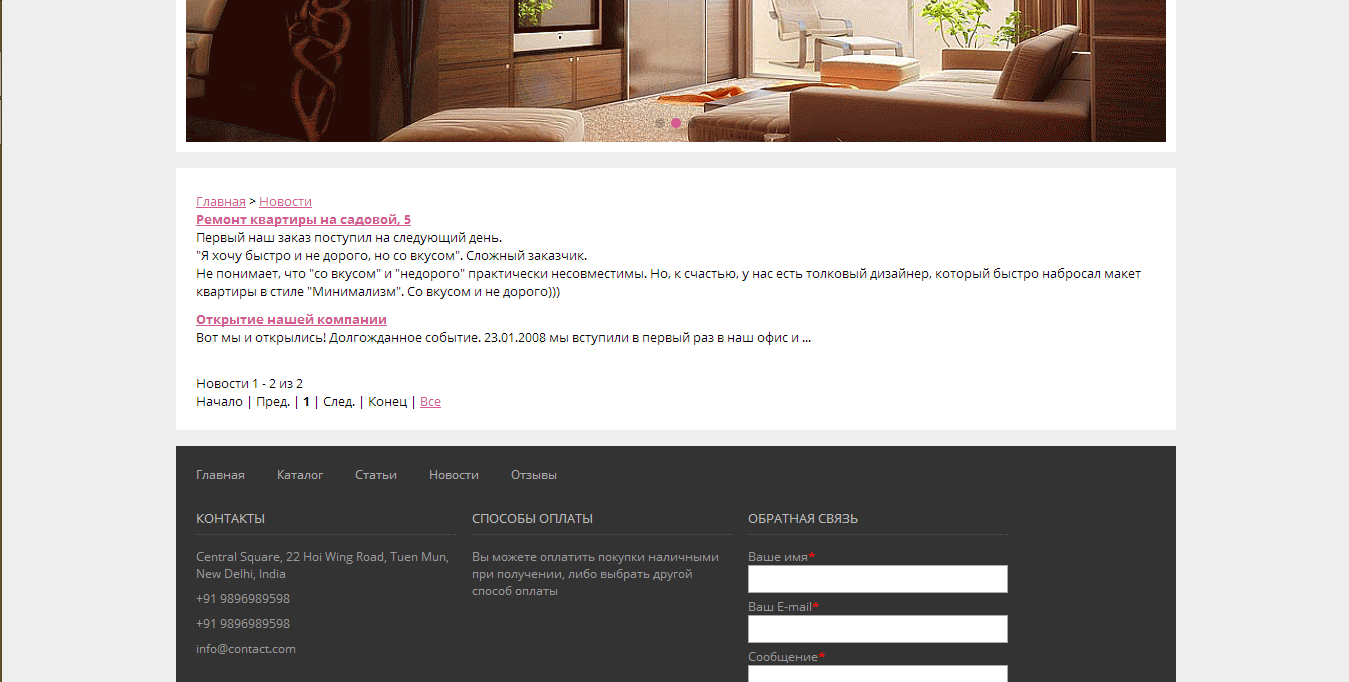


Рисунок 5.4 – Список новостей

Так же, по желанию пользователя можно зайти в определенную новость и прочитать подробнее, показанную на рисунке 5.4.

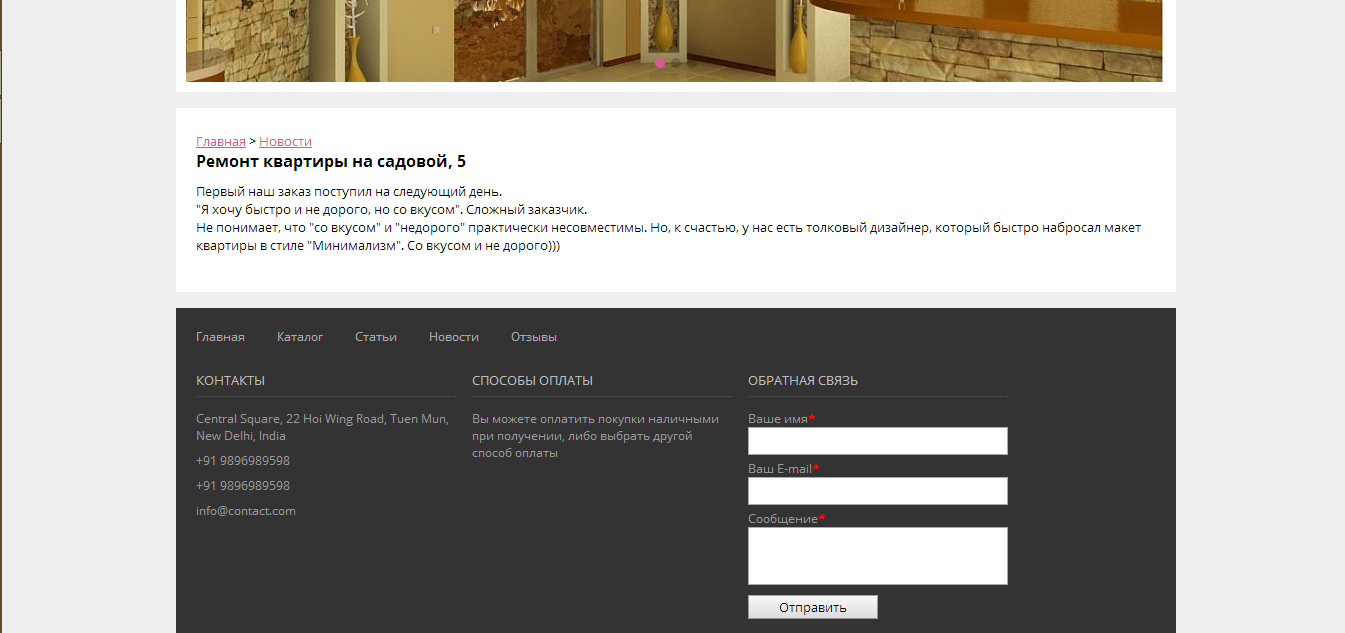


Рисунок 5.5 – Новость детально

Раздел статьи похож по функционалу на раздел новостей, также есть список, также можно подробно прочесть статью. Раздел статьи нужен для получения клиентом более полной информации, нежели от консультации с администратором. Это создает доверительное отношение к компании, что так же позволяет пользователю оставлять на сайте за товар свои кровные денежки.

На сайте присутствует фотогалерея, где расположены фотографии ремонта, товаров и другого. Фотогалерея изображена на рисунке 5.6.

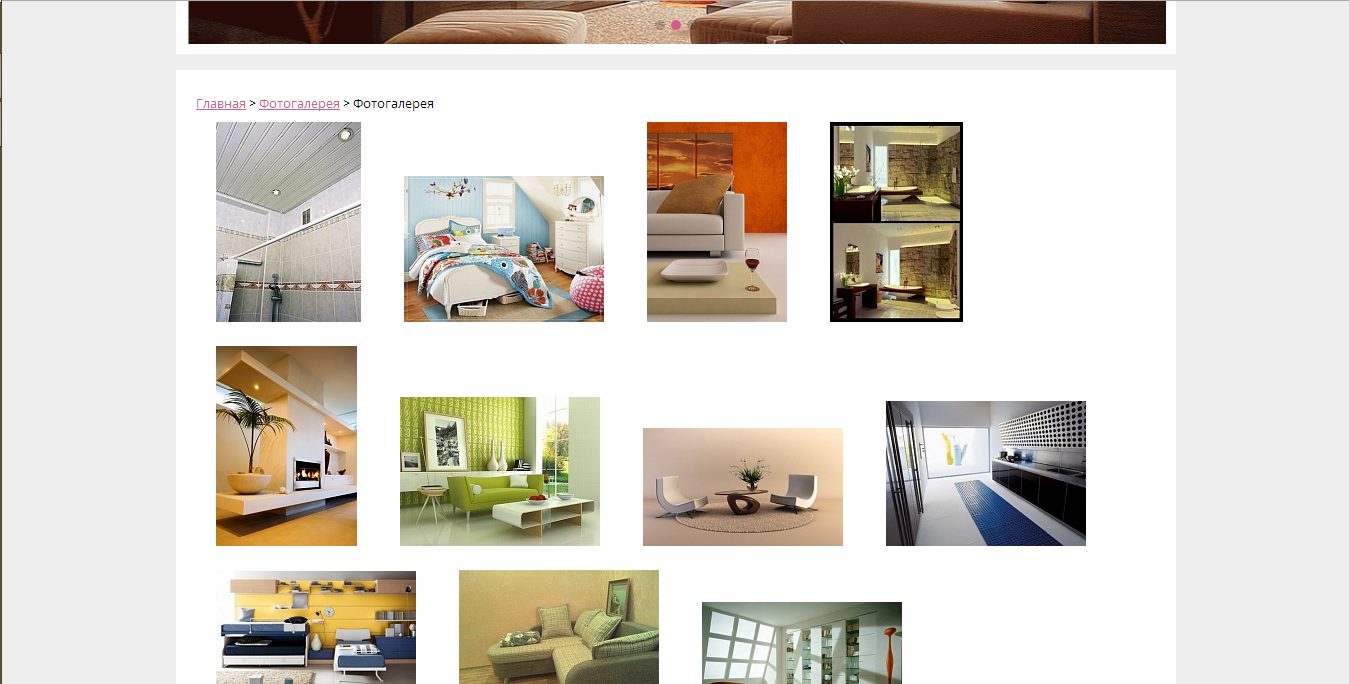


Рисунок 5.6 – Фотогалерея

Пользователь при определенном желании может зарегистрироваться на сайте, рисунок 5.7, 5.8. Но в то же время пользователь так же и в «анонимном режиме», не регистрируясь, может совершать покупки. Выполнено это в таком стиле для того, чтобы не отбить желание клиента потратить деньги своей формой регистрации, так как большинство пользователей не любят регистрироваться, да и пришли они не регистрироваться, а покупать.

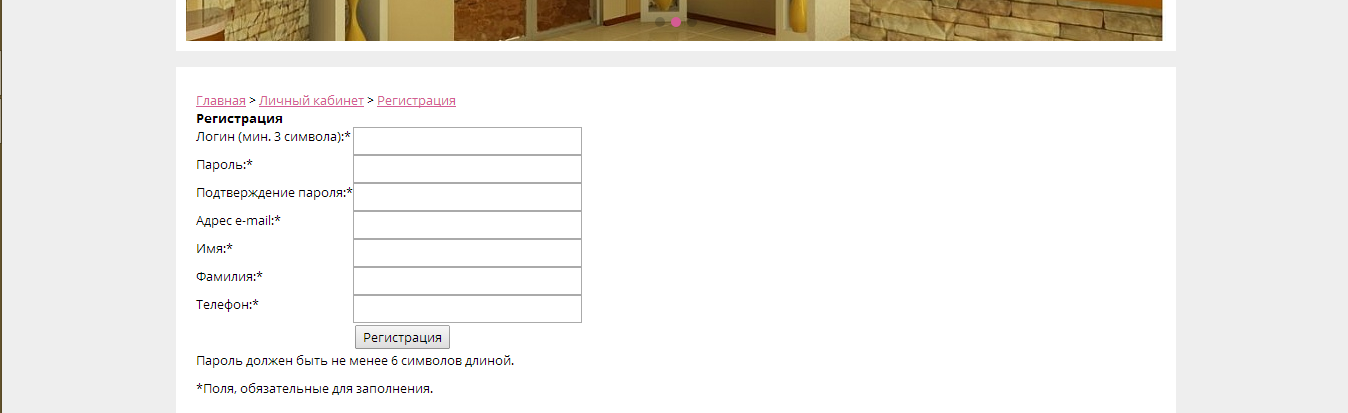


Рисунок 5.7 - Форма регистрации

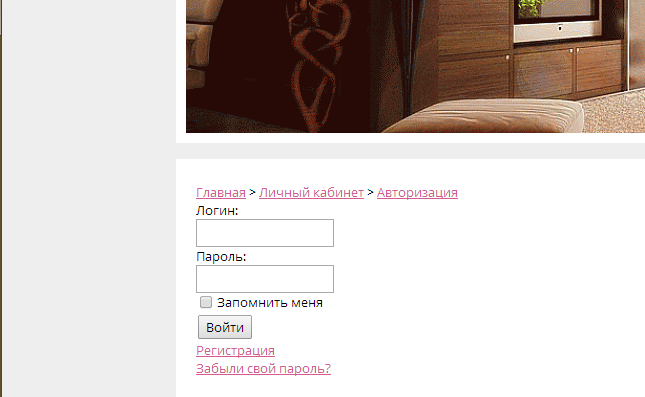


Рисунок 5.8 – Форма входа

На сайте есть раздел, где выводятся просмотренные товары, показанной на рисунке 5.9. Это очень удобно для навигации пользователя: вдруг закрыл понравившийся товар, а искать не хочется. Все просмотренные пользователем товары добавляются в этот раздел, для каждого пользователя там находятся свои товары, которые просмотрел именно он.

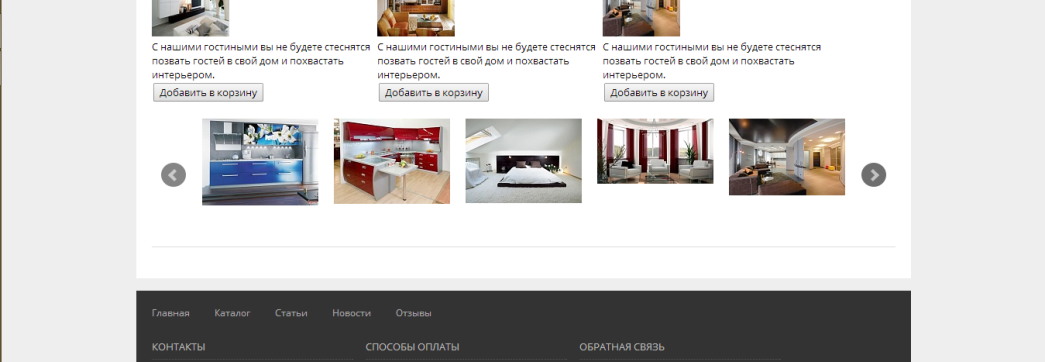


Рисунок 5.9 – Просмотренные товары

Страница отзывов, изображенная на рисунке 5.10, также располагает клиента к компании, потому что все мы, перед тем, как делать где-нибудь заказ, узнаем о компании отзывы, репутацию.

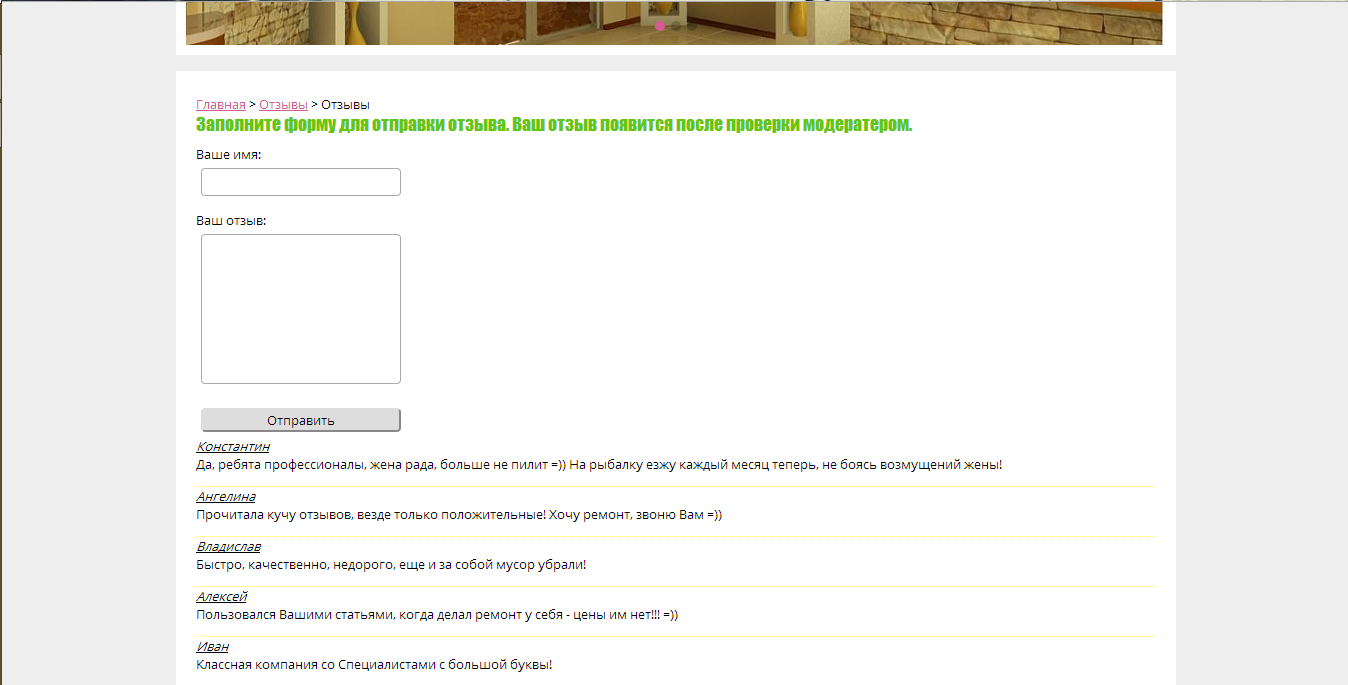


Рисунок 5.10 – Страница отзывов

Здесь же клиент может оставить свой отзыв, будем надеяться, что только положительный.

Теперь перейдем в каталог, корзину и прилегающие части.

Условно каталог разбит на секции(рисунок 5.11), в которых хранятся товары(рисунок 5.12). Это удобно для поиска нужного товара, так как не все в одной куче.

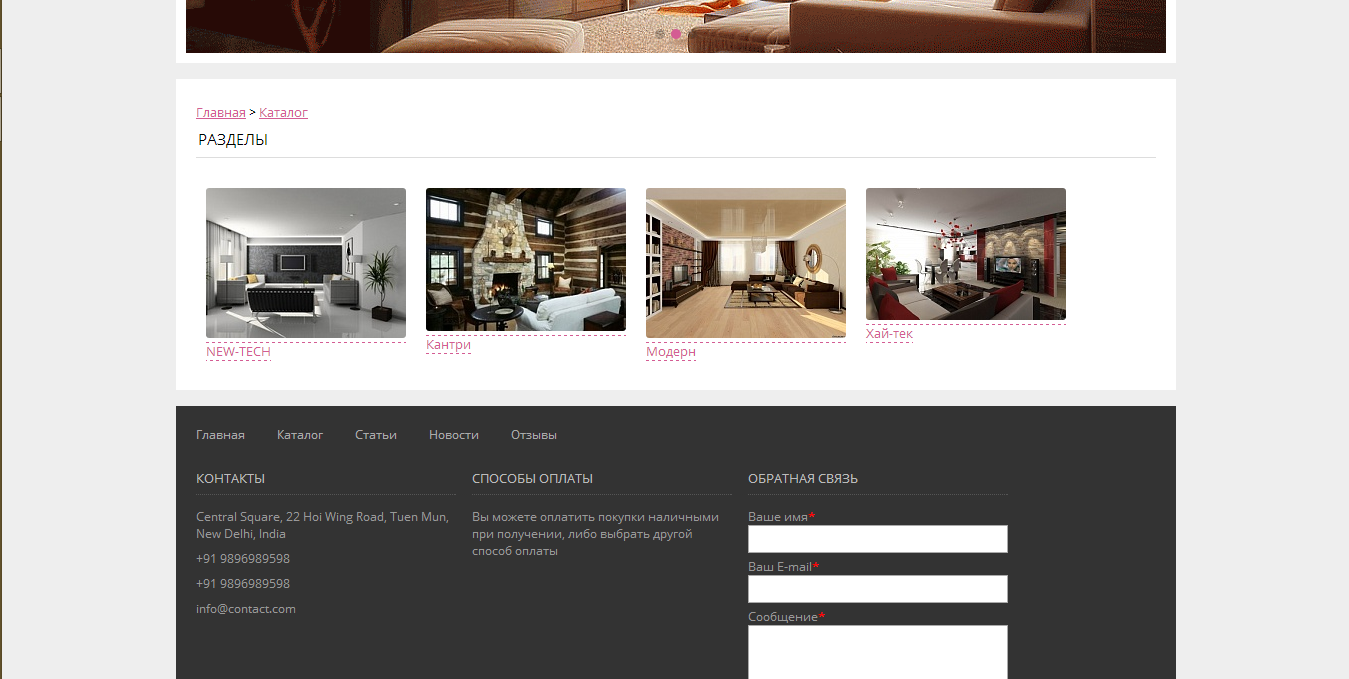


Рисунок 5.11 – Разделы каталога



Рисунок 5.12 – Список товаров

Отсюда можно добавить товар в корзину, если знаем, что он нам точно нужен, или же перейти в детальный просмотр товара, кликнув по названию-ссылке(рисунок 5.13).

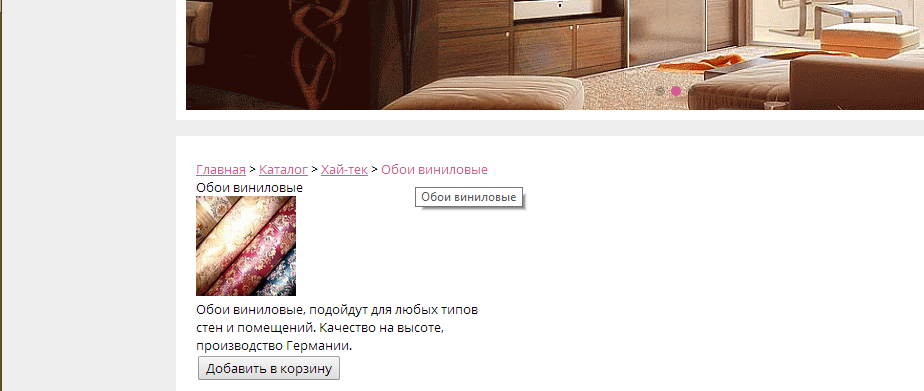


Рисунок 5.13 – Детальный просмотр товара

Обратите внимание, на всех страницах у есть хлебные крошки – список ссылок, по которым нетрудно определить, в каком разделе сайта и каталога находится пользователь.

Набрав товаров, клиент переходит в раздел оплаты и доставки. Сделать это можно кликнув по ссылке в малой корзине, изображенной на рисунке 5.14.

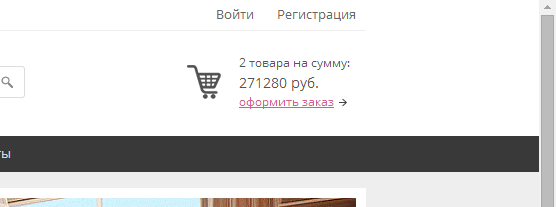


Рисунок 5.14 – Малая корзина

Клиент попадет в раздел с полной корзиной, рисунок 5.15. Здесь можно удалить ненужный товар, в случае если он нечаянно его добавили к себе в корзину.

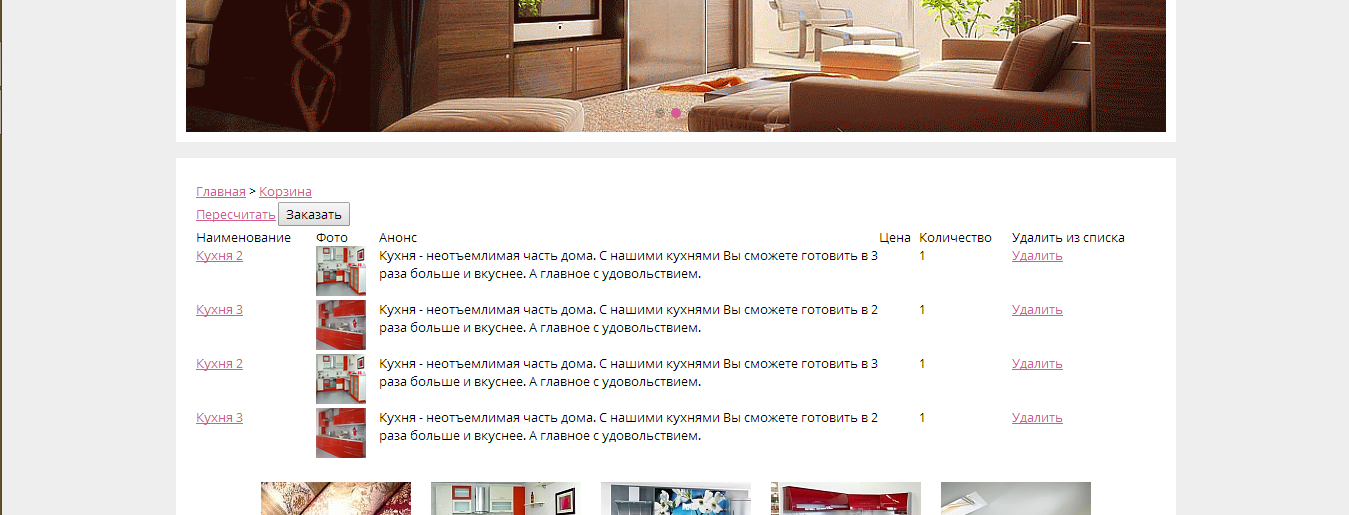


Рисунок 5.15 – Полная корзина

Также здесь выводится информация о товарах, его количество. Можем снова перейти в этот товар, кликнув по названию товара. Если устраивает свой список товаров, клиент переходит по ссылке «Заказать» и попадает на форму оформления заказа, изображенную на рисунке 5.16.

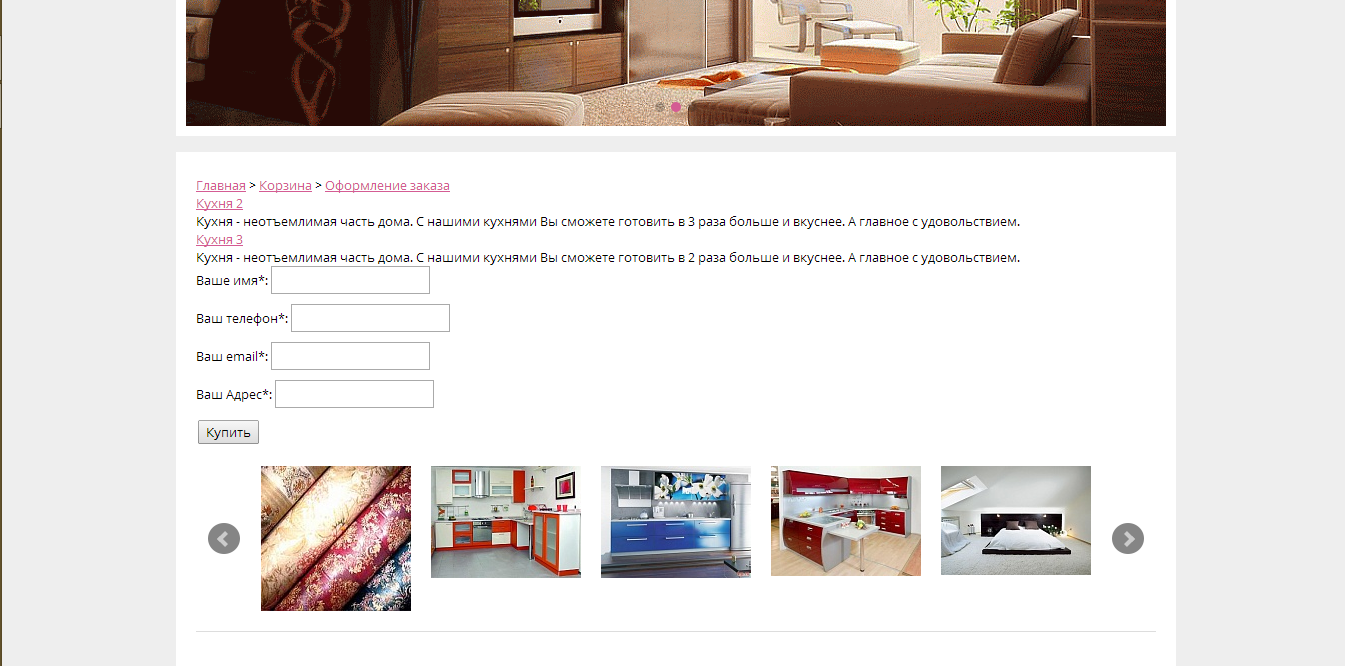


Рисунок 5.16 – Форма оформления заказа

Здесь он заполняет поля «Имя», «Телефон», «E-mail», «Ваш адрес». Эта информация понадобится продавцу для доставки и уточнения доставки. И помимо этого, продавцу понадобится e-mail для его клиентской базы. Все, товар куплен и ждет доставки.

# Заключение

В данной работе было разработано решение для CMS 1С-Битрикс: Управление сайтом редакции «Старт», чтобы сделать интернет-магазин. Оно было практически испытано и готово выйти на рынок в качестве авторского модуля или авторского готового решения для продажи в Магазине 1С-Битрикс. В работе не приведена верстка сайта, так как целью было написание программной части интернет-магазина, используя возможности и функционал Битрикса.

Приложение А

(справочное)

Библиография

1. Джек Д. Харрингтон PHP. Трюки / Джек Д. Харрингтон – Питер, 2008. – 445 с.
2. Николай Прохоренок HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Николай Прохоренок – БХВ-Петербург ,3-е изд., 2010. – 900 с.
3. Кузнецов М.В, Симдянов И.В. PHP. Практика создания Web-сайтов / Кузнецов М.В, Симдянов И.В.– БХВ-Петербург, 2009. – 1244 с.
4. Роберт Басыров 1С-Битрикс. Постройте профессиональный сайт сами! / Роберт Басыров, - Питер, 2009. – 304с.
5. Создание компонента Битрикс, URL: http://www.bxdev.ru/posts/components/bitrix\_components\_creating\_simple\_component/ (Дата последнего обращения: 23.04.2014)
6. Учебные онлайн-курсы по CMS 1С-БИТРИКС: Управление сайтом, URL: <http://dev.1c-bitrix.ru/learning/index.php>(Дата последнего обращения: 23.02.2014)
7. Документация для разработчиков 1C-Битрикс: Управление сайтом, URL: <http://dev.1c-bitrix.ru/api_help/>(Дата последнего обращения: 15.05.2014)
8. Интернет-ресурс по верстке, URL: <http://htmlbook.ru/>(Дата последнего обращения: 13.03.2014)
9. Руководство по PHP, URL: <http://www.php.net/manual/ru/>(Дата последнего обращения: 23.04.2014)

Приложение б

Список основных сокращений

B2B (англ. Business to Business, буквально бизнес для бизнеса) — термин, определяющий вид информационного и экономического взаимодействия, классифицированного по типу взаимодействующих субъектов, в данном случае — это юридические лица, которые работают не на конечного рядового потребителя, а на такие же компании, то есть на другой бизнес.

B2C  (Business-to-consumer, рус. Бизнес для Потребителя) — термин, обозначающий коммерческие взаимоотношения между организацией (Business) и частным, так называемым «конечным» потребителем (consumer)

CMS (Content Management System - система управления содержимым) — информационная система или компьютерная программа для обеспечения и организации совместного процесса создания, редактирования и управления контентом.

PHP (англ. PHP: Hypertext Preprocessor — «PHP: препроцессор гипертекста»; первоначально Personal Home Page Tools[[4]](http://ru.wikipedia.org/wiki/PHP#cite_note-off_history-4) — «Инструменты для создания персональных веб-страниц») — [скриптовый язык](http://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)программирования общего назначения, интенсивно применяемый для разработки веб-приложений.

AJAX(англ. Asynchronous Javascript and XML — «асинхронный [JavaScript](http://ru.wikipedia.org/wiki/JavaScript) и XML») — подход к построению интерактивных пользовательских интерфейсов [веб-приложений](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), заключающийся в «фоновом» обмене данными браузера с [веб-сервером](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80).

SQL(англ. structured query language — «структурированный язык запросов») — формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД).

HTML (от англ. HyperText Markup Language — «язык гипертекстовой разметки») — стандартный язык разметки документов во Всемирной паутине. Большинство [веб-страниц](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

СУБД (Система управления базами данных) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

ПРИЛОЖЕНИЕ В

Листинг B.1.1 – Файл component.php компонента section

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  // echo '<pre>'; print\_r($arParams); echo '</pre>';  CModule::IncludeModule('iblock');  if ($this->StartResultCache(3600) && $arParams["IBLOCK\_ID"]!="")  {  // debug($\_REQUEST["ELEMENT\_CODE"]);  $arResult = array();  $arResult["ITEMS"] = array();  $section\_id = "";  $iblock\_id = $arParams['IBLOCK\_ID'];  $arFilter = array('IBLOCK\_ID'=>$iblock\_id);  if($arParams["IBLOCK\_SECTION\_ID"]) {  $arFilter = array('IBLOCK\_ID'=>$iblock\_id, "SECTION\_ID" => $arParams["IBLOCK\_SECTION\_ID"]);  $section\_id = $arParams["IBLOCK\_SECTION\_ID"];  }  if($arParams["IBLOCK\_SECTION\_CODE"]) {  $res = CIBlockSection::GetList(array(), array('IBLOCK\_ID' => $iblock\_id, 'CODE' => $arParams["IBLOCK\_SECTION\_CODE"]));  $section = $res->Fetch();  if($res->SelectedRowsCount() == 1) {  $arFilter = array\_merge(array('SECTION\_ID'=>$section["ID"]),$arFilter);  $section\_id = $section["ID"];  }  }  $tmpItems = array();  $db\_list1 = CIBlockElement::GetList(array(), $arFilter);  while($ar\_result1 = $db\_list1->GetNext())  {  $arResult["ITEMS"][] = $ar\_result1;  }  if($arParams["SHOW\_SECTIONS"]=="Y") {  $db\_list = CIBlockSection::GetList(array(), $arFilter);  while($ar\_result = $db\_list->GetNext())  {  $arResult["SECTIONS"][] = $ar\_result;  }  }  if($arParams["ADD\_TO\_BASKET"]=="Y") {  $arResult["FORM\_DATA"] = array();  if($\_SERVER["REQUEST\_METHOD"]=="POST" && $\_POST["guard"] == "yes") {  $\_SESSION["ADDED\_ITEMS"][] = array("ID"=>$\_POST["ITEM"],"IBLOCK\_ID"=>$\_POST["IBLOCK\_ID"]);  }  }  if(isset($arParams["PRICE\_ELEM"]) && !empty($arParams["PRICE\_ELEM"])) {  foreach($arResult["ITEMS"] as $key=>$arItem) {  $res = CIBlockElement::GetProperty($iblock\_id, $arItem["ID"], "sort", "asc", array("ID" => $arParams["PRICE\_ELEM"]));  if ($ob = $res->GetNext())  {  $arResult["ITEMS"][$key]["PRICE"] = $ob['VALUE'];  }  }  }  if($arParams["ADD\_CHAIN\_ITEM"]=="Y") {  $arResult["PATH"] = array();  $nav = CIBlockSection::GetNavChain($iblock\_id, $section\_id);  while($res = $nav->GetNext()) {  $arResult["PATH"][] = $res;  }  foreach($arResult["PATH"] as $arItem) {  $APPLICATION->AddChainItem($arItem["NAME"],$arItem["SECTION\_PAGE\_URL"]);  }  }  $this->IncludeComponentTemplate();  } else {  if($arParams['SET\_404']=="Y") {  CHTTP::SetStatus("404 Not Found");  }  echo "Неправильная настройка компонента!";  }  ?> |

Листинг В.1.2 – Файл .parametres.php компонента section

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  CModule::IncludeModule("iblock");  $dbIBlockType = CIBlock::GetList(  array("sort" => "asc"),  array("ACTIVE" => "Y")  );  while ($arIBlockType = $dbIBlockType->Fetch())  {  $arIblockType[$arIBlockType["ID"]] = "[".$arIBlockType["ID"]."] ".$arIBlockType["NAME"];  }  $arComponentParameters = array(  'PARAMETERS' => array(  'IBLOCK\_ID' => array(  'NAME' => 'Id инфоблока',  'TYPE' => 'LIST',  'VALUES' => $arIblockType,  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  'REFRESH' => 'Y',  ),  'IBLOCK\_SECTION\_ID' => array(  'NAME' => 'Id раздела',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'IBLOCK\_SECTION\_CODE' => array(  'NAME' => 'Код раздела',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'SHOW\_SECTIONS' => array(  'NAME' => 'Показывать разделы',  'TYPE' => 'CHECKBOX',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'ADD\_TO\_BASKET' => array(  'NAME' => 'Использовать добавление в корзину',  'TYPE' => 'CHECKBOX',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'ADD\_CHAIN\_ITEM' => array(  'NAME' => 'Включать раздел в цепочку навигации',  'TYPE' => 'CHECKBOX',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'SET\_TITLE' => array(  'NAME' => 'Устанавливать заголовок страницы',  'TYPE' => 'CHECKBOX',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'SET\_404' => array(  'NAME' => 'Устанавливать 404 ошибку,если раздел или элемент не найдены',  'TYPE' => 'CHECKBOX',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'CACHE\_TIME' => array('DEFAULT'=>3600),  ),  );  //debug($arCurrentValues);  if (0 < intval($arCurrentValues['IBLOCK\_ID']))  {  $arPropList = array();  $rsProps = CIBlockProperty::GetList(array(),array('IBLOCK\_ID' => $arCurrentValues['IBLOCK\_ID']));  while ($arProp = $rsProps->Fetch())  {  $arPropList[$arProp['ID']] = $arProp['NAME'];  }  $arComponentParameters['PARAMETERS']['PRICE\_ELEM'] = array(  'NAME' => 'Свойство для вывода цены',  'TYPE' => 'LIST',  'VALUES' => $arPropList,  'PARENT' => 'BASE',  );  }  ?> |

Листинг В.1.3 – файл template.php компонента section

|  |
| --- |
| <?  //debug($arResult);  ?>  <?if(!empty($arResult["SECTIONS"])) {?>  <div class="sections">  <?foreach($arResult["SECTIONS"] as $arSections) {?>  <div class="elem-item">  <span class="name"><a href="<?=$arSections["SECTION\_PAGE\_URL"]?>"><?=$arSections["NAME"]?></a></span>  <?if(!empty($arSections["PREVIEW\_PICTURE"])) {?><div><img src=""></div><?}?>  <?if(!empty($arSections["PREVIEW\_TEXT"])) {?><div><?=$arSections["PREVIEW\_TEXT"]?></div><?}?>  </div>  <?}?>  </div>  <?}?>  <?if(!empty($arResult["ITEMS"])) {?><div class="section-elements">  <?//debug($\_SESSION["ADDED\_ITEMS"]);?>  <?foreach($arResult["ITEMS"] as $arItem) {?>  <div class="elem-item">  <?$file = CFile::ResizeImageGet($arItem['PREVIEW\_PICTURE'], array('width'=>150, 'height'=>150), BX\_RESIZE\_IMAGE\_PROPORTIONAL, true);?>  <span class="name"><a href="<?=$arItem["DETAIL\_PAGE\_URL"]?>"><?=$arItem["NAME"]?></a></span>  <?if(!empty($arItem["PREVIEW\_PICTURE"])) {?><div><img src="<?=$file["src"]?>"/></div><?}?>  <?if(!empty($arItem["PREVIEW\_TEXT"])) {?><div class="prev-text">  <?=$arItem["PREVIEW\_TEXT"]?>  </div><?}?>  <?if($arParams["ADD\_TO\_BASKET"]=="Y") {  ?>  <form method="post">  <input type="hidden" name="guard" value="yes"/>  <input type="hidden" name="ITEM" value="<?=$arItem["ID"]?>"/>  <input type="hidden" name="IBLOCK\_ID" value="<?=$arItem["IBLOCK\_ID"]?>"/>  <input type="submit" value="Добавить в корзину"/>  </form>  <?  }?>  </div>  <?}?>  </div>  <?}?> |

Листинг В.2.1 – Файл component.php компонента section.list

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  CModule::IncludeModule('iblock');  if ($this->StartResultCache(3600) && $arParams['IBLOCK\_ID']!="")  {  $iblock\_id = $arParams['IBLOCK\_ID'];  $section\_id = "";  $arFilter = array('IBLOCK\_ID'=>$iblock\_id);  $arOrder = array();  if($arParams["SUBSECTIONS"]=="Y") {  $arOrder = array("left\_margin"=>"asc");  } else {  $arFilter = array\_merge(array('DEPTH\_LEVEL'=>1),$arFilter);  }  if($arParams["SECTION\_ID"]!="") {  $arFilter = array\_merge(array('SECTION\_ID'=>$arParams["SECTION\_ID"]),$arFilter);  $section\_id = $arParams["SECTION\_ID"];  }  if($arParams["SECTION\_CODE"]!="") {  $res = CIBlockSection::GetList(array(), array('IBLOCK\_ID' => $iblock\_id, 'CODE' => $arParams["SECTION\_CODE"]));  $section = $res->Fetch();  if($res->SelectedRowsCount() == 1) {  $arFilter = array\_merge(array('SECTION\_ID'=>$section["ID"]),$arFilter);  $section\_id = $section["ID"];  }  }  $arResult = array();  $arRes = array();  $db\_list = CIBlockSection::GetList($arOrder, $arFilter, true);  while($ar\_result = $db\_list->GetNext())  {  $arRes[] = $ar\_result;  }  if($arParams["SHOW\_SECTION\_ELEMENT"]=="Y" && $arParams["COUNT\_SECTION\_ELEMENT"]>0) {  $arSectId = array();  $arItems = array();  foreach($arRes as $tmp) {  $arSectId[] = $tmp["ID"];  }  $arFilter = Array("IBLOCK\_ID"=>$iblock\_id, "ACTIVE\_DATE"=>"Y", "ACTIVE"=>"Y");  $res = CIBlockElement::GetList(Array(), $arFilter, false, Array("nPageSize"=>$arParams["COUNT\_SECTION\_ELEMENT"]));  while($ob = $res->GetNext())  {  $arItems[] = $ob;  }  foreach($arRes as $k=>$tmp) {  foreach($arItems as $item) {  if($item["IBLOCK\_SECTION\_ID"]==$tmp["ID"]) {  $arRes[$k]["ITEMS"][] = $item;  }  }  }  }  if($arParams["SUBSECTIONS"]=="Y") {  for($i=0;$i<count($arRes);$i++) {  if($arRes[$i]["DEPTH\_LEVEL"]==1)  {  $arResult[$i] = $arRes[$i];  } elseif($arRes[$i]["DEPTH\_LEVEL"]>1) {  $arResult[$i-1]["CHILD\_LIST"][] = $arRes[$i];  }  }  } else {  $arResult = $arRes;  }  $this->IncludeComponentTemplate();  } else {  echo "Неправильная настройка компонента!";  }  ?> |

Листинг В.2.2 – файл .parametres.php компонента section.list

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  CModule::IncludeModule("iblock");  $dbIBlockType = CIBlock::GetList(  array("sort" => "asc"),  array("ACTIVE" => "Y")  );  while ($arIBlockType = $dbIBlockType->Fetch())  {  $arIblockType[$arIBlockType["ID"]] = "[".$arIBlockType["ID"]."] ".$arIBlockType["NAME"];  }  $arComponentParameters = array(  'PARAMETERS' => array(  'IBLOCK\_ID' => array(  'NAME' => 'Id инфоблока',  'TYPE' => 'LIST',  'VALUES' => $arIblockType,  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'SECTION\_ID' => array(  'NAME' => 'Id раздела',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'SECTION\_CODE' => array(  'NAME' => 'Код раздела',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'SUBSECTIONS' => array(  'NAME' => 'Показывать подразделы(подразделы третьей и более вложенности будут находится на одном уровне с разделами второй вложенности)',  'TYPE' => 'CHECKBOX',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'SHOW\_SECTION\_ELEMENT' => array(  'NAME' => 'Показывать элементы раздела',  'TYPE' => 'CHECKBOX',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'COUNT\_SECTION\_ELEMENT' => array(  'NAME' => 'Количество показываемых элементов раздела',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'CACHE\_TIME' => array('DEFAULT'=>3600),  ),  );  ?> |

Листинг В.2.3 – Файл template.php компонента section.list

|  |
| --- |
| <h2>Разделы</h2>  <ul>  <? foreach($arResult as $arSections): ?>  <li><a href="<?=$arSections["SECTION\_PAGE\_URL"]?>" title="<?=$arSections["NAME"]?>"><?=$arSections["NAME"]?> (<?=$arSections['ELEMENT\_CNT']?>)</a></li>  <? endforeach; ?>  </ul> |

Листинг В.3.1 – Файл component.php компонента section.element.detail

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  // echo '<pre>'; print\_r($arParams); echo '</pre>';  CModule::IncludeModule('iblock');  if ($this->StartResultCache(3600) && $arParams["IBLOCK\_ID"]!="" && ($arParams["IBLOCK\_SECTION\_CODE"]!="" || $arParams["IBLOCK\_SECTION\_ID"]!=""))  {  $arResult = array();  $section\_id = "";  $iblock\_id = $arParams['IBLOCK\_ID'];  $arFilter = array('IBLOCK\_ID'=>$iblock\_id);  if($arParams["IBLOCK\_SECTION\_ID"]!="") {  $arFilter = array('IBLOCK\_ID'=>$iblock\_id, "SECTION\_ID" => $arParams["IBLOCK\_SECTION\_ID"]);  $section\_id = $arParams["IBLOCK\_SECTION\_ID"];  }  if($arParams["IBLOCK\_SECTION\_CODE"]!="") {  $res = CIBlockSection::GetList(array(), array('IBLOCK\_ID' => $iblock\_id, 'CODE' => $arParams["IBLOCK\_SECTION\_CODE"]));  $section = $res->Fetch();  if($res->SelectedRowsCount() == 1) {  $arFilter = array\_merge(array('SECTION\_ID'=>$section["ID"]),$arFilter);  $section\_id = $section["ID"];  // debug($section\_id);  }  }  //debug($section\_id);  if($arParams["IBLOCK\_ELEMENT\_ID"]) {  $arFilter = array\_merge(array('ID'=>$arParams["IBLOCK\_ELEMENT\_ID"]),$arFilter);  }  if($arParams["IBLOCK\_ELEMENT\_CODE"]) {  $arFilter = array\_merge(array('CODE'=>$arParams["IBLOCK\_ELEMENT\_CODE"]),$arFilter);  }  $db\_list = CIBlockElement::GetList(array(), $arFilter);  if($ar\_result = $db\_list->GetNext())  {  $arResult = $ar\_result;  }  if(empty($arResult) && $arParams['SET\_404']=="Y") {  CHTTP::SetStatus("404 Not Found");  }  if($arParams["ADD\_TO\_BASKET"]=="Y") {  $arResult["FORM\_DATA"] = array();  if($\_SERVER["REQUEST\_METHOD"]=="POST" && $\_POST["guard"] == "yes") {  //debug($\_POST);  $\_SESSION["ADDED\_ITEMS"][] = array("ID"=>$\_POST["ITEM"],"IBLOCK\_ID"=>$\_POST["IBLOCK\_ID"]);  }  }  if($arParams["ADD\_TO\_VISITED"]=="Y") {  $\_SESSION["VISITED"][] = array("ID"=>$arResult["ID"], "IBLOCK\_ID"=>$arResult["IBLOCK\_ID"]);  //debug($\_SESSION["VISITED"]);  }  if(isset($arParams["PRICE\_ELEM"]) && !empty($arParams["PRICE\_ELEM"])) {  // debug($arParams["PRICE\_ELEM"]);  $res = CIBlockElement::GetProperty($iblock\_id, $arResult["ID"], "sort", "asc", array("ID" => $arParams["PRICE\_ELEM"]));  if ($ob = $res->GetNext())  {  $arResult["PRICE"] = $ob['VALUE'];  }  }  // debug($\_SESSION["ADDED\_ITEMS"]);  if($arParams["ADD\_CHAIN\_ITEM"]=="Y") {  $arResult["PATH"] = array();  // debug($iblock\_id);  // debug($section\_id);  $nav = CIBlockSection::GetNavChain($iblock\_id, $section\_id);  while($res = $nav->GetNext()) {  $arResult["PATH"][] = $res;  }  // debug($arResult["PATH"]);  foreach($arResult["PATH"] as $arItem) {  $APPLICATION->AddChainItem($arItem["NAME"],$arItem["SECTION\_PAGE\_URL"]);  }  $APPLICATION->AddChainItem($arResult["NAME"],$arResult["DETAIL\_PAGE\_URL"]);  }  $this->IncludeComponentTemplate();  } else {  if($arParams['SET\_404']=="Y") {  CHTTP::SetStatus("404 Not Found");  }  echo "Неправильная настройка компонента!";  }  ?> |

Листинг В.3.2 – Файл .parametres.php компонента section.element.detail

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  CModule::IncludeModule("iblock");  $dbIBlockType = CIBlock::GetList(  array("sort" => "asc"),  array("ACTIVE" => "Y")  );  while ($arIBlockType = $dbIBlockType->Fetch())  {  $arIblockType[$arIBlockType["ID"]] = "[".$arIBlockType["ID"]."] ".$arIBlockType["NAME"];  }  $arComponentParameters = array(  'PARAMETERS' => array(  'IBLOCK\_ID' => array(  'NAME' => 'Id инфоблока',  'TYPE' => 'LIST',  'VALUES' => $arIblockType,  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  'REFRESH' => 'Y'  ),  'IBLOCK\_SECTION\_ID' => array(  'NAME' => 'Id раздела',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'IBLOCK\_SECTION\_CODE' => array(  'NAME' => 'Код раздела',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'IBLOCK\_ELEMENT\_ID' => array(  'NAME' => 'Id элемента',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'IBLOCK\_ELEMENT\_CODE' => array(  'NAME' => 'Код элемента',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'ADD\_TO\_BASKET' => array(  'NAME' => 'Использовать добавление в корзину',  'TYPE' => 'CHECKBOX',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'ADD\_TO\_VISITED' => array(  'NAME' => 'Запоминать просмотренные элементы',  'TYPE' => 'CHECKBOX',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'ADD\_CHAIN\_ITEM' => array(  'NAME' => 'Включать раздел в цепочку навигации',  'TYPE' => 'CHECKBOX',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'SET\_TITLE' => array(  'NAME' => 'Устанавливать заголовок страницы',  'TYPE' => 'CHECKBOX',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'SET\_404' => array(  'NAME' => 'Устанавливать 404 ошибку,если раздел или элемент не найдены',  'TYPE' => 'CHECKBOX',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'CACHE\_TIME' => array('DEFAULT'=>3600),  ),  );  if (0 < intval($arCurrentValues['IBLOCK\_ID']))  {  $arPropList = array();  $rsProps = CIBlockProperty::GetList(array(),array('IBLOCK\_ID' => $arCurrentValues['IBLOCK\_ID']));  while ($arProp = $rsProps->Fetch())  {  $arPropList[$arProp['ID']] = $arProp['NAME'];  }  $arComponentParameters['PARAMETERS']['PRICE\_ELEM'] = array(  'NAME' => 'Свойство для вывода цены',  'TYPE' => 'LIST',  'VALUES' => $arPropList,  'PARENT' => 'BASE',  );  }  ?> |

Листинг В.3.3 – Файл templete.php компонента section.element.detail

|  |
| --- |
| <?  //debug($arResult);  ?>  <div class="section-elements">  <div class="elem-item">  <span class="name"><?=$arResult["NAME"]?></span>  <?$file = CFile::ResizeImageGet($arResult['PREVIEW\_PICTURE'], array('width'=>150, 'height'=>150), BX\_RESIZE\_IMAGE\_PROPORTIONAL, true);?>  <?if(!empty($arResult["PREVIEW\_PICTURE"])) {?><div><img src="<?=$file["src"]?>"/></div><?}?>  <?if(!empty($arResult["PREVIEW\_PICTURE"])) {?><div class="prev-text">  <?=$arResult["PREVIEW\_TEXT"]?>  </div><?}?>  <?if($arParams["ADD\_TO\_BASKET"]=="Y") {  ?>  <form method="post">  <input type="hidden" name="guard" value="yes"/>  <input type="hidden" name="ITEM" value="<?=$arResult["ID"]?>"/>  <input type="hidden" name="IBLOCK\_ID" value="<?=$arResult["IBLOCK\_ID"]?>"/>  <input type="submit" value="Добавить в корзину"/>  </form>  <?  }?>  </div>  </div> |

Листинг В.4.1 – Файл component.php компонента sale.order

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  // echo '<pre>'; print\_r($arParams); echo '</pre>';  CModule::IncludeModule('iblock');  function \_\_checkEmail($email)  {  if(!preg\_match('|([a-z0-9\_\.\-]{1,20})@([a-z0-9\.\-]{1,20})\.([a-z]{2,4})|is', $email)) {  //echo "false";  return FALSE;  } else {  return true;  }  }  if ($this->StartResultCache(3600))  {  $arResult = array();  if(!empty($\_SESSION["BUY\_ITEMS"])) {  $arItemsId = array();  foreach($\_SESSION["BUY\_ITEMS"] as $tmp) {  $arItemsId[] = $tmp["ID"];  }  $arFilter = Array("ID"=>$arItemsId, "ACTIVE\_DATE"=>"Y", "ACTIVE"=>"Y");  $res = CIBlockElement::GetList(Array(), $arFilter, false);  $arPict = array();  while($ob = $res->GetNext())  {  $arResult[] = $ob;  }  if($\_SERVER["REQUEST\_METHOD"]=="POST" &&  $\_POST["action"]=='buy' &&  !empty($\_POST["email"]) &&  !empty($\_POST["phone"]) &&  !empty($\_POST["name"]) &&  !empty($\_POST["address"])  )  {  if(\_\_checkEmail($\_POST["email"])) {  //echo "12";  $el = new CIBlockElement;  $PROP = array();  $arItemsId = array();  foreach($\_SESSION["BUY\_ITEMS"] as $tmp) {  $arItemsId[] = $tmp["ID"];  }  // debug($arItemsId);  $PROP = array(  $arParams["USER\_NAME"]=>$\_POST["name"],  $arParams["USER\_PHONE"]=>$\_POST["phone"],  $arParams["USER\_EMAIL"]=>$\_POST["email"],  $arParams["USER\_ADDRESS"]=>$\_POST["address"],  $arParams["TOTAL\_PRICE"]=>"",  $arParams["PURCHASED\_GOODS"]=>$arItemsId,);  $arProps = Array(  "MODIFIED\_BY" => $USER->GetID(), // элемент изменен текущим пользователем  "IBLOCK\_SECTION\_ID" => false, // элемент лежит в корне раздела  "IBLOCK\_ID" => $arParams["IBLOCK\_ID"],  "PROPERTY\_VALUES"=> $PROP,  "NAME" => "Заказ от ".date("d.m.Y H:i")." покупателя ".$\_POST["name"],  );  // debug($arProps);  if($PRODUCT\_ID = $el->Add($arProps)) {  unset($\_SESSION["BUY\_ITEMS"]);  unset($\_SESSION["ADDED\_ITEMS"]);  $arEventFields = array(  "USER\_NAME"=>$\_POST["name"],  "USER\_PHONE"=>$\_POST["phone"],  "USER\_EMAIL"=>$\_POST["email"],  "USER\_ADDRESS"=>$\_POST["address"],  "TOTAL\_PRICE"=>"",  "PURCHASED\_GOODS"=>$arItemsId,  );  CEvent::Send("PURCHASE", SITE\_ID, $arEventFields);  LocalRedirect($arParams["LINK\_SUCCESS"]);  }  }  }  }  $this->IncludeComponentTemplate();  }  ?> |

Листинг В.4.2 – Файл .parametres.php компонента sale.order

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  CModule::IncludeModule("iblock");  $dbIBlockType = CIBlock::GetList(  array("sort" => "asc"),  array("ACTIVE" => "Y")  );  while ($arIBlockType = $dbIBlockType->Fetch())  {  $arIblockType[$arIBlockType["ID"]] = "[".$arIBlockType["ID"]."] ".$arIBlockType["NAME"];  }  $arFilter = array(  "LID" => "ru"  );  $rsET = CEventType::GetList($arFilter);  while ($arET = $rsET->Fetch())  {  $arEventType[$arET["ID"]] = "[".$arET["ID"]."] ".$arET["EVENT\_NAME"];  }  $arComponentParameters = array(  'PARAMETERS' => array(  'IBLOCK\_ID' => array(  'NAME' => 'Id инфоблока',  'TYPE' => 'LIST',  'VALUES' => $arIblockType,  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  'REFRESH' => 'Y',  ),  'EVENT\_TYPE' => array(  'NAME' => 'Тип почтового события',  'TYPE' => 'LIST',  'VALUES' => $arEventType,  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'LINK\_SUCCESS' => array(  'NAME' => 'Путь в случае успешного оформления заказа',  'TYPE' => 'STRING',  ),  'CACHE\_TIME' => array('DEFAULT'=>3600),  ),  );  if (0 < intval($arCurrentValues['IBLOCK\_ID']))  {  $arPropList = array();  $rsProps = CIBlockProperty::GetList(array(),array('IBLOCK\_ID' => $arCurrentValues['IBLOCK\_ID']));  while ($arProp = $rsProps->Fetch())  {  $arPropList[$arProp['ID']] = $arProp['NAME'];  }  $arComponentParameters['PARAMETERS']['USER\_PHONE'] = array(  'NAME' => 'Свойство для хранения телефона пользователя',  'TYPE' => 'LIST',  'VALUES' => $arPropList,  'PARENT' => 'BASE',  );  $arComponentParameters['PARAMETERS']['USER\_NAME'] = array(  'NAME' => 'Свойство для хранения имени пользователя',  'TYPE' => 'LIST',  'VALUES' => $arPropList,  'PARENT' => 'BASE',  );  $arComponentParameters['PARAMETERS']['USER\_EMAIL'] = array(  'NAME' => 'Свойство для хранения пользовательского email',  'TYPE' => 'LIST',  'VALUES' => $arPropList,  'PARENT' => 'BASE',  );  $arComponentParameters['PARAMETERS']['USER\_ADDRESS'] = array(  'NAME' => 'Свойство для хранения адреса доставки',  'TYPE' => 'LIST',  'VALUES' => $arPropList,  'PARENT' => 'BASE',  );  $arComponentParameters['PARAMETERS']['TOTAL\_PRICE'] = array(  'NAME' => 'Свойство для хранения общей стоимости заказа',  'TYPE' => 'LIST',  'VALUES' => $arPropList,  'PARENT' => 'BASE',  );  $arComponentParameters['PARAMETERS']['PURCHASED\_GOODS'] = array(  'NAME' => 'Свойство для хранения приобретенных товаров',  'TYPE' => 'LIST',  'VALUES' => $arPropList,  'PARENT' => 'BASE',  );  }  ?> |

Листинг В.4.3 – Файл template.php компонента sale.order

|  |
| --- |
| <?if(!empty($arResult))  {?>  <div class="basket-element">  <form method="post">  <input type="hidden" name="action" value="buy"/>  <?foreach($arResult as $arItem) {?>  <div class="basket-item">  <div class="name-element"><a href="<??>"><?=$arItem["NAME"]?></a></div>  <?if(!empty($arItem["PREVIEW\_PICTURE"])) {?><div class="img"></div><?}?>  <?if(!empty($arItem["PREVIEW\_TEXT"])) {?><div class="text"><?=$arItem["PREVIEW\_TEXT"]?></div><?}?>  <span class="cost-element"></span>  </div>  <?}?>  <p>Ваше имя\*: <input type="text" name="name"/></p>  <p>Ваш телефон\*: <input type="text" name="phone" /></p>  <p>Ваш email\*: <input type="text" name="email"/></p>  <p>Ваш Адрес\*: <input type="text" name="address"/></p>  <input type="submit" value="Купить"/>  </form>  </div>  <?  }  ?> |

Листинг В.5.1 – файл component.php компонента form.element.add

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  // echo '<pre>'; print\_r($arParams); echo '</pre>';  CModule::IncludeModule("iblock");  if ($this->StartResultCache(3600))  {  //echo $\_POST["guard"];  if($\_SERVER["REQUEST\_METHOD"]=="POST"  && $\_POST["guard"]=="yes"  && !empty($\_POST["name"])  && !empty($\_POST["message"])  ) {  //debug($\_POST);  $el = new CIBlockElement;  $arProps = Array(  "MODIFIED\_BY" => $USER->GetID(), // элемент изменен текущим пользователем  "IBLOCK\_SECTION\_ID" => false, // элемент лежит в корне раздела  "IBLOCK\_ID" => $arParams["IBLOCK\_ID"],  "NAME" => $\_POST["name"],  "PREVIEW\_TEXT" => $\_POST["message"],  );  // debug($arProps);  if($PRODUCT\_ID = $el->Add($arProps)) {  echo "Отзыв отправлен";  }  }  $this->IncludeComponentTemplate();  }  ?> |

Листинг В.5.2 – Файл .parametres.php компонента form.element.add

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  CModule::IncludeModule("iblock");  $dbIBlockType = CIBlock::GetList(  array("sort" => "asc"),  array("ACTIVE" => "Y")  );  while ($arIBlockType = $dbIBlockType->Fetch())  {  $arIblockType[$arIBlockType["ID"]] = "[".$arIBlockType["ID"]."] ".$arIBlockType["NAME"];  }  $arFilter = array(  "LID" => "ru"  );  $rsET = CEventType::GetList($arFilter);  while ($arET = $rsET->Fetch())  {  $arEventType[$arET["ID"]] = "[".$arET["ID"]."] ".$arET["EVENT\_NAME"];  }  $arComponentParameters = array(  'PARAMETERS' => array(  'IBLOCK\_ID' => array(  'NAME' => 'Id инфоблока для сохранения',  'TYPE' => 'LIST',  'VALUES' => $arIblockType,  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  'REFRESH' => 'Y',  ),  'EVENT\_TYPE' => array(  'NAME' => 'Тип почтового события',  'TYPE' => 'LIST',  'VALUES' => $arEventType,  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'LINK\_SUCCESS' => array(  'NAME' => 'Путь в случае успешного добавления элемента',  'TYPE' => 'STRING',  ),  'CACHE\_TIME' => array('DEFAULT'=>3600),  ),  );  ?> |

Листинг В.5.3 – Файл template.php компонента form.element.add

|  |
| --- |
| <div class="review-add">  <form method="post">  <input type="hidden" name="guard" value="yes" />  <p>Ваше имя: <input type="text" name="name"/></p>  <p>Ваш отзыв:<textarea name="message"></textarea></p>  <input type="submit" value="Отправить"/>  </form>  </div> |

Листинг В.6.1 – Файл component.php компонента element.visited

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  // echo '<pre>'; print\_r($arParams); echo '</pre>';  CModule::IncludeModule('iblock');  if ($this->StartResultCache(3600))  {  $arResult = array();  $tmp = array();  if(!empty($\_SESSION["VISITED"])) {  for($i=0;$i<count($\_SESSION["VISITED"]);$i++) {  if(!in\_array($\_SESSION["VISITED"][$i]["ID"],$tmp)) {  $tmp[] = $\_SESSION["VISITED"][$i]["ID"];  $newResult[] = $\_SESSION["VISITED"][$i];  }  }  $arItemsId = array();  foreach($newResult as $arItems) {  $arItemsId[] = $arItems["ID"];  }  $db\_list = CIBlockElement::GetList(array(), array("ID"=>$arItemsId,"IBLOCK\_ID"=>$arItems["IBLOCK\_ID"]));  while($ar\_result = $db\_list->GetNext())  {  $arResult[] = $ar\_result;  }  }  $this->IncludeComponentTemplate();  }  ?> |

Листинг В.6.2 – файл template.php компонента element.visited

|  |
| --- |
| <?if(!empty($arResult))  {?>  <div class="basket-element">  <?foreach($arResult as $arItem) {?>  <div class="basket-item">  <?$file = CFile::ResizeImageGet($arItem['PREVIEW\_PICTURE'], array('width'=>150, 'height'=>150), BX\_RESIZE\_IMAGE\_PROPORTIONAL, true);?>  <div class="name-element"><a href="<?=$arItem["DETAIL\_PAGE\_URL"]?>"><?=$arItem["NAME"]?></a></div>  <?if(!empty($arItem["PREVIEW\_PICTURE"])) {?><div class="img"><img src="<?=$file["src"]?>"/></div><?}?>  <?if(!empty($arItem["PREVIEW\_TEXT"])) {?><div class="text"><?=$arItem["PREVIEW\_TEXT"]?></div><?}?>  <span class="cost-element"></span>  </div>  <?}?>  </div>  <?  }  ?> |

Листинг В.7.1 – Файл component.php компонента basket.link

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  // echo '<pre>'; print\_r($arParams); echo '</pre>';  CModule::IncludeModule('iblock');  if ($this->StartResultCache(3600))  {  $arResult = array();  $arQuery = array();  if(!empty($\_SESSION["ADDED\_ITEMS"])) {  $arItemsId = array();  foreach($\_SESSION["ADDED\_ITEMS"] as $tmp) {  $arItemsId[] = $tmp["ID"];  }  $db\_list = CIBlockElement::GetList(array(), array("ID"=>$$arItemsId),false,false,array("PROPERTY\_PRICE"));  while($ar\_result = $db\_list->GetNext())  {  $arResult["PRICE"] += $ar\_result["PROPERTY\_PRICE\_VALUE"];  }  }  if($arParams["BASKET\_PATH"]!="")  $arResult["PATH\_TO\_BASKET\_PAGE"] = $arParams["BASKET\_PATH"];  if($arParams["ORDER\_PATH"]!="")  $arResult["PATH\_TO\_ORDER\_PAGE"] = $arParams["ORDER\_PATH"];  $arResult["COUNT\_ELEM"] = count($\_SESSION["ADDED\_ITEMS"]);  $this->IncludeComponentTemplate();  }  ?> |

Листинг В.7.2 – Файл .parameters.php компонента basket.link

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  $arComponentParameters = array(  'PARAMETERS' => array(  'BASKET\_PATH' => array(  'NAME' => 'Путь к странице корзины',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'ORDER\_PATH' => array(  'NAME' => 'Путь к странице оформления заказа',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'CACHE\_TIME' => array('DEFAULT'=>3600),  ),  );  ?> |

Листинг В.7.3 – Файл template.php компонента basket.link

|  |
| --- |
| <?if(!empty($arResult))  {?>  <div class="basket-link">  <div class="cnt"><?=$arResult["COUNT\_ELEM"]?></div>  <div class="price"><?=$arResult["PRICE"]?></div>  <?  // debug($arResult);  if(isset($arResult["PATH\_TO\_BASKET\_PAGE"]) && $arResult["PATH\_TO\_BASKET\_PAGE"]!="") {?>  <a href="<?=$arResult["PATH\_TO\_BASKET\_PAGE"]?>">Перейти в корзину</a>  <?}?>  <?//debug(isset($arResult["PATH\_TO\_ORDER\_PAGE"]));?>  <?if(isset($arResult["PATH\_TO\_ORDER\_PAGE"]) && $arResult["PATH\_TO\_ORDER\_PAGE"]!="") {?>  <a href="<?=$arResult["PATH\_TO\_ORDER\_PAGE"]?>">Перейти к оформлению</a>  <?}?>  </div>  <?  }  ?> |

Листинг В.8.1 – Файл component.php компонента basket

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  // echo '<pre>'; print\_r($arParams); echo '</pre>';  CModule::IncludeModule('iblock');  if ($this->StartResultCache(3600))  {  $arResult = array();  $arQuery = array();  $tmp = array();  if(!empty($\_SESSION["ADDED\_ITEMS"])) {  for($i=0;$i<count($\_SESSION["ADDED\_ITEMS"]);$i++) {  if(!in\_array($\_SESSION["ADDED\_ITEMS"][$i]["ID"],$tmp)) {  $tmp[] = $\_SESSION["ADDED\_ITEMS"][$i]["ID"];  $newResult[] = $\_SESSION["ADDED\_ITEMS"][$i];  } else {  $index = array\_search($\_SESSION["ADDED\_ITEMS"][$i]["ID"],$tmp);  $newResult[$index]["COUNT"] ++ ;  }  }  if($\_SERVER["REQUEST\_METHOD"]=="GET" && $\_GET["action"]=='delete' && is\_numeric($\_GET["id"])) {  foreach($newResult as $key=>$arItem) {  if($arItem["ID"]==$\_GET["id"]) {  unset($newResult[$key]);  }  }  }  if($\_SERVER["REQUEST\_METHOD"]=="POST" && !empty($newResult)) {  $\_SESSION["BUY\_ITEMS"] = $newResult;  LocalRedirect($arParams["ORDER\_PATH"]);  }  if($\_SERVER["REQUEST\_METHOD"]=="POST" && $\_POST["action"]=='refresh' && !empty($newResult)) {  $\_SESSION["BUY\_ITEMS"] = $newResult;  }  $\_SESSION["ADDED\_ITEMS"] = $newResult;  $arItemsId = array();  foreach($newResult as $tmp) {  $arItemsId[] = $tmp["ID"];  }  $db\_list = CIBlockElement::GetList(array(), array("ID"=>$arItemsId,"IBLOCK\_ID"=>$arItems["IBLOCK\_ID"]));  while($ar\_result = $db\_list->GetNext())  {  $arResult[] = array\_merge($ar\_result, array("COUNT"=>(empty($arItems["COUNT"])?"1":$arItems["COUNT"])));  }  }  $this->IncludeComponentTemplate();  }  ?> |

Листинг В.8.2 – Файл .parametres.php компонента basket

|  |
| --- |
| <?  if(!defined("B\_PROLOG\_INCLUDED")||B\_PROLOG\_INCLUDED!==true)die();  $arComponentParameters = array(  'PARAMETERS' => array(  'BASKET\_PATH' => array(  'NAME' => 'Путь к странице корзины',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'ORDER\_PATH' => array(  'NAME' => 'Путь к странице оформления заказа',  'TYPE' => 'STRING',  'MULTIPLE' => 'N',  'PARENT' => 'BASE',  ),  'CACHE\_TIME' => array('DEFAULT'=>3600),  ),  );  ?> |

Листинг В.8.3 – Файл template.php компонента basket

|  |
| --- |
| <?if(!empty($arResult))  {?>  <div class="basket-element">  <form method="post">  <?foreach($arResult as $arItem) {?>  <div class="basket-item">  <div class="name-element"><a href="<?=$arItem["DETAIL\_PAGE\_URL"]?>"><?=$arItem["NAME"]?></a></div>  <?$file = CFile::ResizeImageGet($arItem['PREVIEW\_PICTURE'], array('width'=>150, 'height'=>150), BX\_RESIZE\_IMAGE\_PROPORTIONAL, true);?>  <?if(!empty($arItem["PREVIEW\_PICTURE"])) {?><div class="img"><img src="<?=$file["src"]?>"/></div><?}?>  <?if(!empty($arItem["PREVIEW\_TEXT"])) {?><div class="text"><?=$arItem["PREVIEW\_TEXT"]?></div><?}?>  <span class="cost-element"></span>  <span class="delete-element">  <a href="?id=<?=$arItem["ID"]?>&action=delete">Удалить</a>  </span>  </div>  <?}?>  <a href="?action=refresh">Пересчитать</a>  <input type="submit" value="Заказать"/>  </form>  </div>  <?  }  ?> |

Листинг В.9 – Содержание файла init.php

|  |
| --- |
| <?  function debug($var) {  echo "<pre>"; print\_r($var); echo "</pre>";  }  ?> |