

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Механико-математический факультет  
Кафедра веб-технологий и компьютерного моделирования

## **Курсовая работа**

**«ASP.NET MVC. Веб-проект. Интернет-магазин»**

Выполнил студент:

2 курса 2 группы

Специальность:

математика (информационные технологии)

Гончаров Александр Витальевич

Научный руководитель:

кандидат физико-математических наук, доцент

Кремень Елена Васильевна

Минск, 2019

## СОДЕРЖАНИЕ

Введение .....	3
1. Постановка задачи.....	4
1.1 Описание предметной области .....	4
1.2 Модель предметной области .....	<b>Ошибка! Закладка не определена.</b>
1.3 Модель сайта .....	4
2. Обзор инструментальных средств, используемых при создании веб-сайта .....	<b>Ошибка! Закладка не определена.</b>
2.1 Задачи, решаемые при разработке веб-сайта.....	<b>Ошибка! Закладка не определена.</b>
2.2 Методы создания сайта.....	<b>Ошибка! Закладка не определена.</b>
2.3 Выбор и обоснование CMS для создания сайта ....	<b>Ошибка! Закладка не определена.</b>
3. Разработка сайта.....	6
3.1 Подготовка к установке Joomla .....	6
3.2 Установка дистрибутива Denwer.....	<b>Ошибка! Закладка не определена.</b>
3.3 Установка и настройка Joomla.....	<b>Ошибка! Закладка не определена.</b>
3.4 Разработка дизайна.....	13
3.5 Разработка веб-сайта .....	<b>Ошибка! Закладка не определена.</b>
3.6 Создание шаблона сайта .....	<b>Ошибка! Закладка не определена.</b>
4. Описание сайта.....	16
4.1 Программно-аппаратные ресурсы ПК .....	16
4.2 Тестирование .....	16
4.3 Демонстративный пример работы программы .....	17
Заключение .....	23
Список использованных источников .....	24
Приложение А .....	25

## ВВЕДЕНИЕ

Тема данной курсовой работы – «ASP.NET MVC. Веб-проект. Интернет-магазин».

Данный курсовой проект был выполнен с целью практического освоения основных приемов и правил создания веб-сайта.

В настоящее время Интернет представляет собой один из самых активно развивающихся средств информации. По статистике, значительное число населения имеют дома компьютер и имеют доступ к сети Интернет.

В современном обществе больше всего цениться время, люди очень много тратят времени на покупки. Для того что бы экономить время на покупки начали создаваться интернет - магазины. Интернет - магазин - это сайт, на котором при помощи браузера можно заказать необходимые товары, которые курьер доставит домой или в офис. Преимущества такого вида покупки очевидны:

- не надо никуда идти и стоять в очереди;
- не надо думать о доставке товара - это сделают сотрудники интернет-магазина;
- можно сэкономить время;
- просто и удобно оформить заказ.

Настоящая работа посвящена разработке интернет-магазина на технологии ASP.NET MVC 5 с использованием движка визуализации Razor, Entity Framework, Ninject, Bootstrap.

## **1. ПОСТАНОВКА ЗАДАЧИ**

### **1.1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

Предметной областью называется часть реального мира, представляющая интерес для данного исследования.

Для проектируемого сайта предметной областью является информация о фильмах, имеющихся в продаже на сайте.

В разделе «Магазин» посетитель может узнать информацию о фильмах, имеющихся в продаже (жанр, стоимость, название, описание) и при желании добавить фильм в свою корзину покупок.

Раздел «Корзина» предоставляет информацию о товарах, находящихся в корзине, а также позволит посетителю удалить из своей корзины нежелательный фильм, вернуться к списку фильмов и продолжить добавление их в корзину, либо оформить заказ с существующим списком покупаемых товаров.

Раздел «Оформление заказа» предоставляет пользователю возможность ввести для последующей доставки товара.

Раздел «Страница администрации» является недоступным для обычного пользователя, для доступа к нему необходима авторизация с использованием данных администратора.

Раздел «Вакансии» и «Контакты» расскажет об актуальных вакансиях предприятия и соответственно всех актуальных контактах: телефоны, электронный адрес, физический адрес, реквизиты.

Целью разработки интернет-магазина является последующая продажа своих товаров посетителям.

### **1.2 МОДЕЛЬ САЙТА**

Одним из важных моментов является разграничение прав доступа у различных категорий пользователей.

1. Посетитель магазина.
2. Администратор.

Посетитель магазина при входе на сайт обладает базовыми правами. Он может просматривать всю информацию о фильме: название, описание, жанр, стоимость. При желании он может добавить товар в корзину и позже оформить заказ. Также, при необходимости, посетитель может удалить товар из своей корзины.

Администратор может авторизоваться на сайте для того использовать CRUD (create, read, update, delete)-средства для управления каталогом товаров.

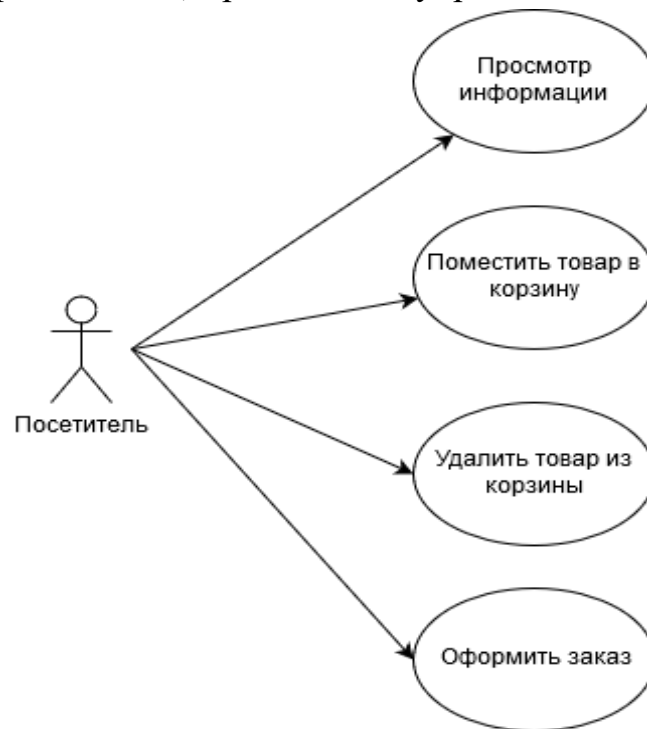


Рисунок 1.2.1 Диаграмма прецедентов категории "Посетитель"

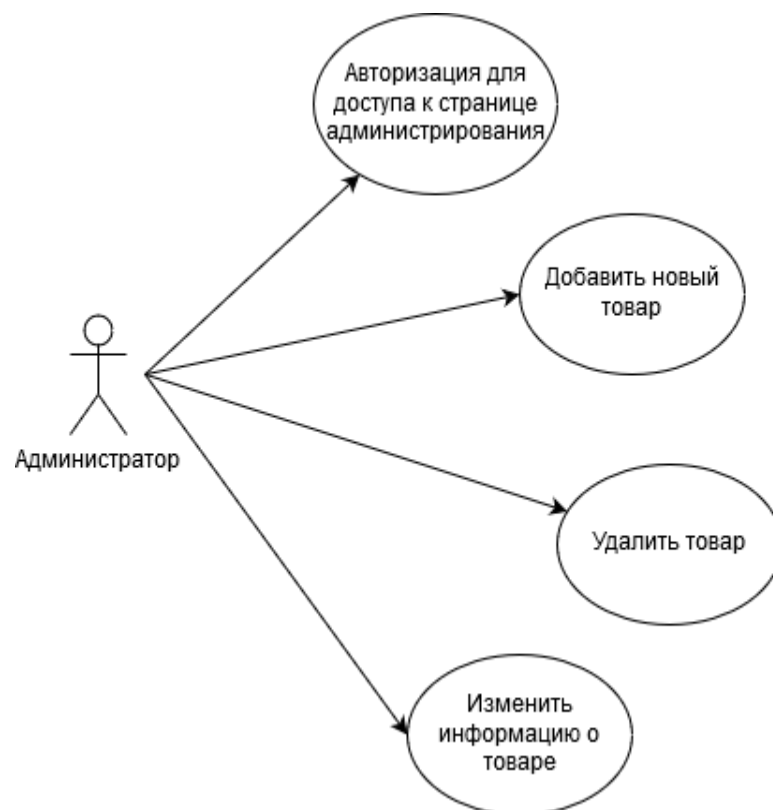


Рисунок 1.2.2 Диаграмма прецедентов категории "Администратор"

## 2. РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА

### 2.1 ВВЕДЕНИЕ В РАЗРАБОТКУ

- Выбранный язык программирования: C#;
- Используемые библиотеки: ASP.NET MVC, Ninject, Entity Framework, Bootstrap, jQuery.
- Среда разработки: Visual Studio 2019\2017.
- SQL сервер: Microsoft SQL Server 2012 Express

### 2.2 СОЗДАНИЕ СТРУКТУРЫ ИНТЕРНЕТ-МАГАЗИНА

Интернет-магазин будет следовать классическому подходу, который повсеместно используется в интернет-магазинах. Мы создадим онлайн-каталог товаров, который пользователи могут просматривать по категориям и страницам, корзину для покупок, куда пользователи могут добавлять и удалять товары, и форму оплаты, где пользователи могут вводить сведения, связанные с доставкой. Кроме того, будет создана административная область, которая включает в себя для управления каталогом товаров. Она будет защищена для предотвращения нежелательных манипуляций со списком товаров.

Имя проекта	Шаблон проекта Visual Studio	Назначение
Domain	Class Library (Библиотека классов)	Содержит сущности и логику предметной области; настраивается на обеспечение постоянства посредством хранилища, которое создано с помощью инфраструктуры Entity Framework.
WebUI	ASP.NET MVC Web Application. (Веб-приложение ASP.NET MVC 5)	Содержит контроллеры и представления; выступает в качестве пользовательского интерфейса для приложения .

UnitTests	Unit Test Project (Проект модульного тестирования)	Содержит модульные тесты для других двух проектов
-----------	--	---

Таблица 2.2.1 Три проекта для приложения интернет-магазин

Во избежание проблем на этапе компиляции необходимо настроить зависимости между проектами, а также от определенных сборок Microsoft.

Имя проекта	Зависимости внутри решения	Зависимости от сборок
Domain	-	System.ComponentModel.DataAnnotations.
WebUI	Domain	-
UnitTests	Domain, WebUI	System.Web, Microsoft.CSharp

Таблица 2.2.2 Обязательные зависимости между проектами

Следующий шаг – создание первой сущности предметной области, товара (фильма). Структура сущности:

- FilmId (int)
- Name (string)
- Description (string)
- Genre (string)
- Price (decimal)
- ImageData (byte[])
- ImageMimeType (string)

Далее необходимо создать абстрактное хранилище. Поля и методы интерфейса IFilmRepository:

- IEnumerable<Film> Films {get;}
- Void SaveFilm (Film film)
- Film DeleteFilm(int filmId)

Интерфейс использует другой интерфейс IEnumerable<T>, чтобы позволить вызывающему коду получать последовательность объектов Film, ничего не сообщая о том, как или где хранятся или извлекаются данные. Класс, зависящий от интерфейса IFilmRepository, может получить объекты Film, ничего не зная о том, откуда они поступают или каким образом класс реализации будет их доставлять. В этом и состоит суть шаблона хранилище (repository).

Также необходимо создать контроллер FilmController, в конструкторе которого нужно объявить зависимость от интерфейса IFilmRepository. Это приведет к внедрению библиотекой Ninject данной зависимости для хранилища фильмов при создании экземпляра класса контроллера. Вдобавок нужно

добавить метод действия List(), который будет визуализировать представление, отображающее полный список товаров.

Прежде чем можно будет реализовать настоящее хранилище, необходимо настроить базу данных и заполнить ее данными. Для базы данных будет применяться SQL Server 2012, а доступ к ней будет осуществляться с помощью Entity Framework (EF) – инфраструктуры ORM для платформы .NET. Инфраструктура ORM представляет таблицы, столбцы и строки реляционной базы данных с помощью обычных объектов C#. Следует упомянуть, что LINQ может работать с различными источниками данных, и один из них - Entity Framework.

Это область, где доступен широкий диапазон инструментальных средств и технологий. Можно взаимодействовать не только с разными реляционными базами данных, но также с хранилищами объектов, хранилищами документов и некоторыми вообще экзотическими альтернативами. Кроме того, существуют другие инфраструктуры ORM для .NET, каждая из которых использует слегка отличающийся подход - подобное разнообразие позволяет выбрать то, что наиболее подходит для конкретных проектов.

В проекте использована инфраструктура Entity Framework по нескольким причинам: она проста и ее легко запустить в работу; она прекрасно интегрирована с LIN. В ранних версиях этой библиотеки имелись серьезные недостатки, но в текущих версиях эти недостатки были исправлены, и на данный момент библиотека эффективна и обладает большими функциональными возможностями.

Для создания базы данных используем инструменты Visual Studio и заполняем созданную базу данных, например, таким образом:

Name	Description	Genre	Price
Начало	Кобб — талантливый вор, лучший из лучших в опасном искусстве извлечения: он крадет ценные секреты из глубин подсознания во время сна, когда человеческий разум наиболее уязвим. Редкие способности Кобба сделали его ценным игроком в привычном к	Драма	24,91



	предательству мире промышленного шпионажа, но они же превратили его в извечного беглеца и лишили всего, что он когда-либо любил.		
--	--	--	--

Таблица 2.2.3 Пример заполнения базы данных

FilmId-столбец идентичности, для которого SQL Server будет генерировать уникальное значение при переходе на следующую строку. Изображение корректно добавлять вручную на сайте.

Далее нужно использовать библиотеку Ninject для создания специального распознавателя зависимостей, который MVC Framework будет применять при создании экземпляров объектов внутри приложения. В проекте WebUI создается соответствующий файл класса с именем NinjectDependencyResolver.cs, в который записывается нужная реализация.

Следующий шаг – создание контекста Entity Framework. Последние версии Entity Framework включают замечательное средство, которое называется *сначала код (code-first)*. Идея состоит в том, что в модели можно определить классы, а затем на их основе сгенерировать базу данных. Это очень удобно для проектов, создаваемых с нуля.

Создается класс контекста, который будет ассоциировать модель с базой данных. Чтобы воспользоваться преимуществом подхода "сначала код", потребуется создать класс, производный от System.Data.Entity.DbContext. Этот класс затем автоматически определяет свойство для каждой таблицы базы данных, с которой необходимо работать.

Имя свойства указывает таблицу, а параметр типа результата DbSet - тип модели, который инфраструктура Entity Framework должна использовать для представления строк в этой таблице. В рассматриваемом случае именем свойства является Films, а параметром типа - Film, т.е. для представления строк в таблице Films инфраструктура Entity Framework будет применять тип модели Film.

Далее Entity Framework нужно указать, как подключаться к базе данных, и это делается путем добавления в файл Web.config проекта WebUI строки подключения к базе данных с таким же именем, как у класса контекста:

```
<configuration>
  <connectionStrings>
    <add name="EFDbContext" connectionString="Data Source=(localdb)\v11.0;Initial Cata-
log=FilmStore;Integrated Security=True" providerName="System.Data.SqlClient" />
  </connectionStrings>
```

Логика модели и хранилища определяются в проекте Domain, но информация подключения к базе данных помещается в файл Web.config проекта WebUI.

Все, что осталось сделать – создать хранилище EFFilmRepository, реализующий интерфейс IFilmRepository и использует экземпляр EFDbContext для извлечения данных из базы данных посредством Entity Framework. Также в NinjectDependencyResolver добавляется привязка к хранилищу.

Для создания представления интернет-магазина необходимо создать стандартное представление метода List() FilmController, а позже редактировать его. Средствами Visual Studio будет создан не только файл List.cshtml, но также и файлы \_ViewStart.cshtml и Shared/\_Layout.cshtml. В файл \_Layout.cshtml заносится необходимая компоновка, а в файл List.cshtml код, состоящий из HTML-разметки и несколько выражений Razor для отображения полной информации о товарах.

На этом этапе создана структура интернет-магазина и интернет-магазин извлекает сведения о товарах из базы данных. На сайте интернет-магазина добавлено представление товаров, имеющихся в базе данных.

## 2.3 РАЗБИЕНИЕ НА СТРАНИЦЫ, СТИЛИЗАЦИЯ КОНТЕНТА И НАВИГАЦИЯ

Следующий этап разработки – создание разбиения контента на стнициы, стилизация контента, и реализация навигации для упрощенного доступа пользователей к интересующим их товарам.

На этом этапе были достигнуты следующие результаты:

- Добавлены различные URL для доступа к различными страницам и жанрам фильмов.

URL	Что выводит
/	Выводит первую страницу списка товаров всех категорий
/Page2	Выводит указанную страницу (в этом случае страницу 2), отображая товары всех категорий

/Драма	Отображает первую страницу элементов указанного жанра(в этом случае фильма жанра "Драма")
/Симулятор/Page2	Отображает заданную страницу (в этом случае страницу 2) элементов указанного жанра (Драма)

Таблица 2.3.1 Сводка маршрутизации

- Создано и стилизовано меню поиска фильмов по жанрам

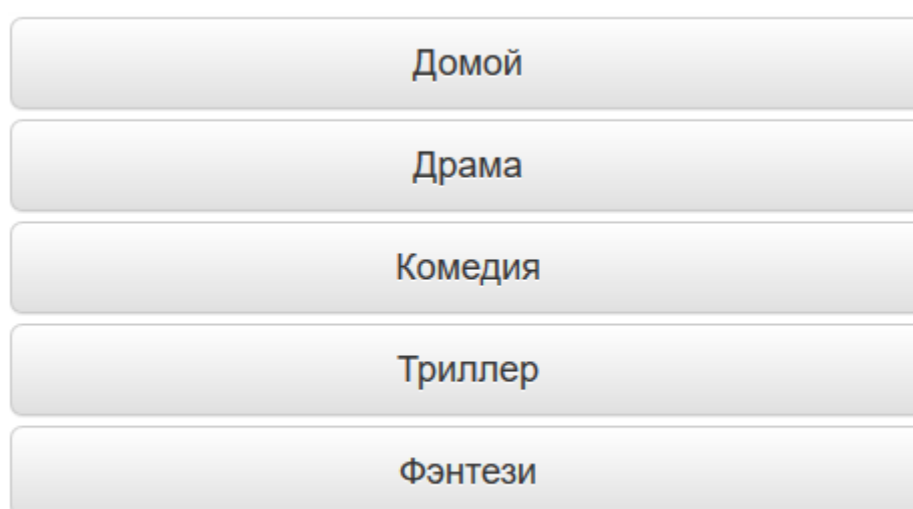


Рисунок 2.3.1 Отображение меню на сайте

## 2.4 СОЗДАНИЕ КОРЗИНЫ ПОКУПОК И ОТПРАВКИ ЗАКАЗОВ

Для реализации функционала сайта необходимо создать корзину покупок, в которую пользователь может добавлять понравившиеся ему фильмы, а также дать пользователю возможность оформить заказ.

На данном этапе разработки были достигнуты следующие результаты:

- Стилизовано и оформлено верхнее меню, содержащее название сайта и переход к корзине покупок



Рисунок 2.4.1 Отображение верхнего меню на сайте

– Создана корзина покупок

Ваша корзина

Кол-во	Фильм	Цена	Общая цена	
1	Начало	24,91 руб	24,91 руб	<a href="#">Удалить</a>
Итого:			24,91 руб	
<div><a href="#">Продолжить покупки</a><a href="#">Оформить заказ</a></div>				

### Рисунок 2.4.2 Корзина покупок

- Создана страница оформления заказа

Оформить заказ сейчас

Пожалуйста введи ваши контактные данные, и мы сразу отправим товар!

## Данные

Ваше имя:

Адрес доставки

Адрес

Город

Страна

## Опции

☐ Завернуть диск в фирменную подарочную упаковку?

Обработать заказ

### Рисунок 2.4.3 Оформление заказа

- Доставка сообщения о заказе на почту администратору (в данном случае в файл на компьютер).

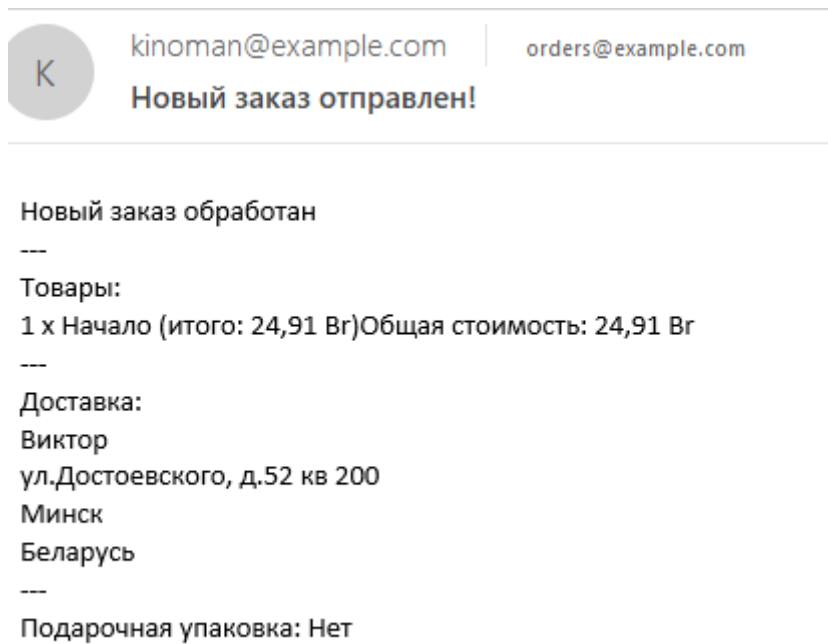


Рисунок 2.4.4 Сообщение о заказе

## 2.5 РАЗРАБОТКА СТРАНИЦ АДМИНИСТРИРОВАНИЯ

Необходимо предоставить возможность администрации манипулировать списком товаров, а также предотвратить неавторизованный доступ к страницам администрирования.

На данном этапе разработки были достигнуты следующие результаты:

- Создана страница авторизации для доступа к ресурсам администрации

### Вход в систему

Пожалуйста, войдите в систему, чтобы получить доступ к админ. панели:

Имя пользователя:

Пароль:

Войти

Рисунок 2.5.1 Страница авторизации администратора

- Разработана страница администрации, на которых администратор может использовать CRUD-средства.

Список фильмов			
ID	Название	Цена	Действия
2	<a href="#">Начало</a>	24,91 руб	<a href="#">Удалить</a>
3	<a href="#">Остров проклятых</a>	19,99 руб	<a href="#">Удалить</a>
4	<a href="#">Форрест Гамп</a>	29,89 руб	<a href="#">Удалить</a>
5	<a href="#">Криминальное чтиво</a>	10,99 руб	<a href="#">Удалить</a>
6	<a href="#">Зеленая миля</a>	9,99 руб	<a href="#">Удалить</a>
8	<a href="#">Операция «Ы» и другие приключения Шурика</a>	14,99 руб	<a href="#">Удалить</a>
9	<a href="#">Властелин колец: Две крепости</a>	20,99 руб	<a href="#">Удалить</a>
<a href="#">Добавить фильм</a>			

Рисунок 2.5.2 Страница администрирования списка фильмов

## Редактирование фильма «Начало»

### Название

Начало

### Описание

Кобб — талантливый вор, лучший из лучших в опасном искусстве извлечения: он крадет ценные секреты из глубин подсознания во время сна, когда человеческий разум наиболее уязвим. Редкие способности Кобба сделали его ценным игроком в привычном к предательству мире промышленного шпионажа, но они же превратили его в извечного беглеца и лишили всего, что он когда-либо любил.

### Жанр

Драма

### Цена (руб)

24,91

### Картинка [Выберите файл...](#)



Сохранить

Отменить изменения и вернуться к списку

Рисунок 2.5.3 Страница редактирования информации о фильме

По завершению этого этапа разработки сайт готов к развертке: В корзину безошибочно добавляются товары, заказа оформляется с верными данными, также администратор может легко добавлять, изменять и удалять фильмы. Для доступа к администраторской панели необходима авторизация, благодаря чему неавторизованный доступ к данным о товарах невозможен.

## **4. ОПИСАНИЕ САЙТА**

### **4.1 ПРОГРАММНО-АППАРАТНЫЕ РЕСУРСЫ ПК**

В процессе работы использовались следующие технические и программные средства:

Технические средства:

- ноутбук ASUS K53SV-SX732;
- процессор Intel® Core™ i5-2410M (2,30 ГГц, 3 МБ L3 Cache);
- оперативная память 4Gb;
- графический адаптер NVIDIA GeForce GT 540M;
- локальная видеопамять 1Gb;
- жесткий диск Samsung на 500Gb.

Программные средства:

- Windows10 Pro (64 Бит);
- Word 2016;
- конструктор сайта Joomla 3.5;
- Джентльменский Набор Web-Разработчика (PHP 5.3.3, MySQL 5.1, PostgreSQL 8.4 etc.);
- Dreamweaver;
- Adobe photoshop;
- Яндекс.Браузер.

### **4.2 ТЕСТИРОВАНИЕ**

Задача тестирования — не доказать правильность работы сайта, а выявить неизвестные ошибки. Требуется обнаружить условия, при которых сайт



работает неверно. Принципы и методы тестирования определяются назначением и структурой программы.

На этапе тестирования на работоспособность проверяют, как функционирует веб-сайт, используя те же условия, при которых с ним будет работать пользователь. Поработайте с веб-сайтом в различных браузерах и посмотрите, как выглядит Ваш сайт в каждом из них. Постарайтесь оценить время загрузки страниц, что очень важно.

Для тестирования на удобство пользования интерфейсом крупные компании приглашают специальные группы людей. Можно пригласить своих друзей и, не давая им никаких инструкций, посмотреть, как они будут пользоваться вашим веб-сайтом. Нужно обратить внимание на то, как они перемещаются по веб-сайту. Где возникают паузы? Когда пользователи испытывают трудности? И при этом не подсказывая им! Такие наблюдения дадут вам много ценной информации. А если пользователи будут выполнять неправильные действия, то это уже недостаток вашей разработки и, значит, над веб-сайтом следует еще поработать.

Виды тестирования программ. Тестирование программы обычно происходит в два этапа, о которых говорят, как об альфа-тестировании и бета-тестировании. Альфа-тестирование проводит сам разработчик, бета-тестирование — заказчик или потенциальный. Бета-тестирование может быть открытым для всех желающих или закрытым. При закрытом бета-тестировании программа предоставляется ограниченному кругу лиц, отобранных разработчиком.

Если в ходе тестирования были обнаружены ошибки, то после их исправления все тесты необходимо провести заново. К сожалению, исправление выявленных ошибок в программе нередко сопровождается появлением новых ошибок.

### 4.3 ДЕМОНСТРАТИВНЫЙ ПРИМЕР РАБОТЫ ПРОГРАММЫ

Для работы с сайтом, входим в браузер, например, Яндекс.Интернет и в строку для ввода веб-адреса вводим rgns.by (рисунок 4.3.1)



Рисунок 4.3.1 – Вход на сайт

При загрузке сайта появляется главная страница сайта. На главной странице отображается информация о предприятии, предоставляемые услуги, контакты и переход к карте сайта. С этой страницы можно переходить по всем ссылкам сайта (Рис.4.3.2).



Рисунок 4.3.2 – Главная страница сайта

На рисунке 4.3.3 отображена страница где рассказывается подробно о самом предприятии и истории Руденской ГНС. Появляется по переходу пункта меню О предприятии → О предприятии.



Рисунок 4.3.3 – О предприятии

На рисунке 4.3.4 показана страница О предприятии → Фотогалерея. Тут можно увидеть несколько фотографий Газонаполнительной станции.

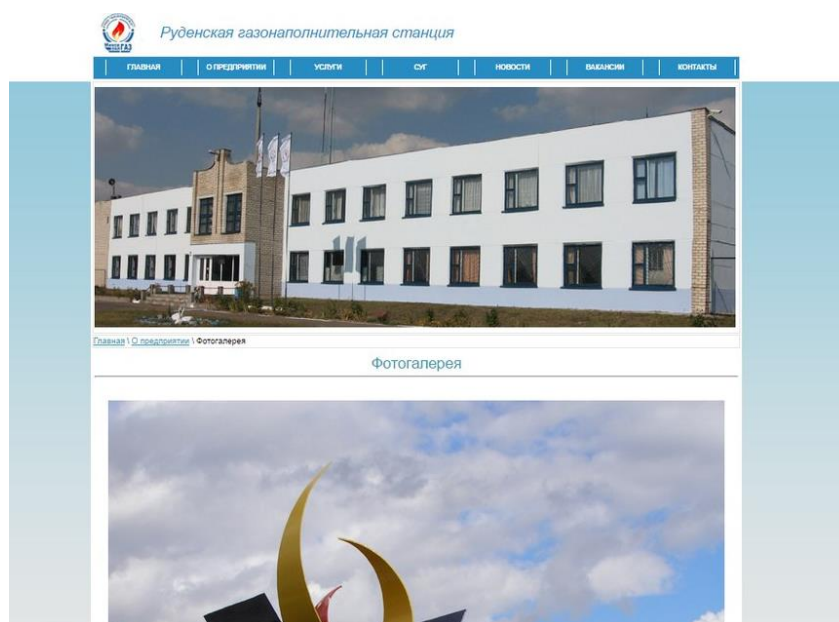


Рисунок 4.3.4 – Фотогалерея

На рисунке 4.3.5 показана страница О предприятии → Руководители. Отображается информация о руководстве предприятия: должность, имя, контактные телефоны.

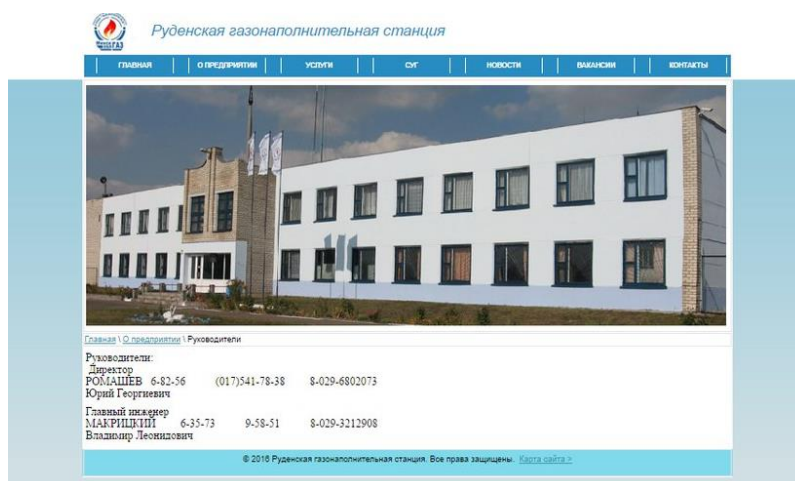


Рисунок 4.3.5 – Руководители

На рисунке 4.3.6 показана страница Услуги. Перечислены все предоставляемые услуги, получаемые виды документов, контакты, режим работы.

Информация предоставлена в виде слайдера, то есть изменяющийся текст определенной услуги при нажатии по соответствующей ссылке.



Рисунок 4.3.6 – Услуги, предоставляемые Руденской ГНС

На рисунке 4.3.7 показана страница СУГ → Что такое СУГ. Предоставлена информация для ознакомления о сжиженном углеводородном газе.



Рисунок 4.3.7 – Что такое СУГ

На рисунке 4.3.8 показана страница СУГ → Техника безопасности. Тут рассказывается о правилах использования газа в быту.





Рисунок 4.3.8 – Техника безопасности

На рисунке 4.3.9 показана страница Новости. В этом разделе размещены полезные статьи и новости станции.

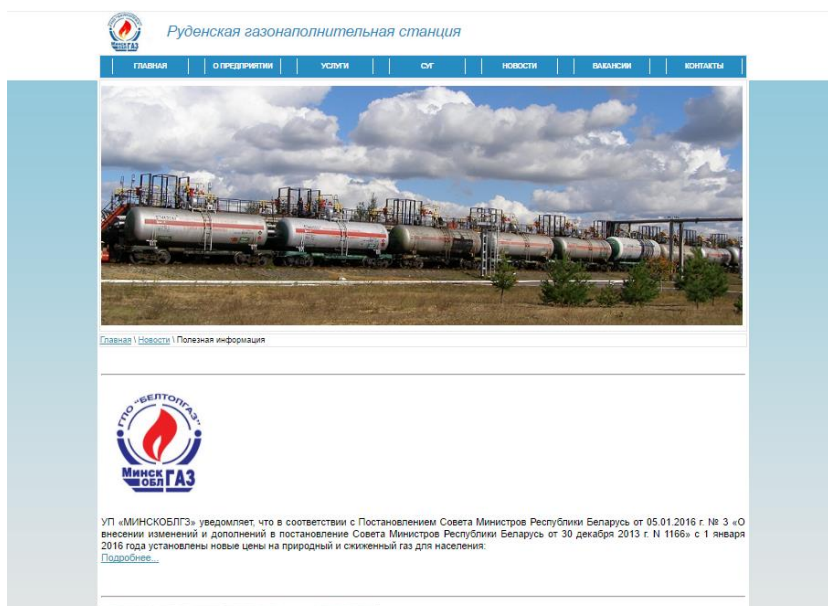


Рисунок 4.3.9 – Новости

На рисунке 4.3.10 – показана страница Вакансии. Указываются все актуальные вакансии и контактные телефоны.



Рисунок 4.3.10 – Актуальные вакансии

На рисунке 4.3.11 – показана страница Контакты. На этой странице указан адрес, контактные телефоны и реквизиты предприятия.

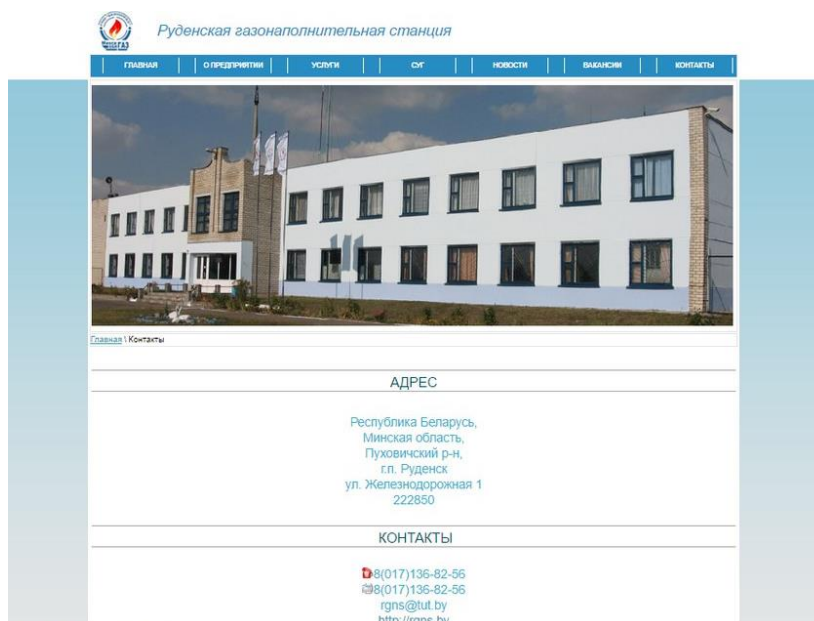


Рисунок 4.3.11 – Контакты

Так же имеется карта сайта, представленная на рисунке 4.3.12. На нее можно попасть с главной страницы сайта, либо внизу любой странице где указана защита прав сайта – «Подвал сайта».



Рисунок 4.3.12 – Карта сайта

## ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы был получен полнофункциональный web-сайт, полностью готовый к применению. Данный сайт создан для ознакомления с предоставляемыми услугами и информации об организации.

При разработке web-сайта были проанализированы современные web-технологии, позволяющие создавать интерактивные web-страницы. Наиболее подходящими для выполнения поставленной задачи оказались Denwer, Dreamweaver, Adobe Photoshop и Joomla.

В качестве дальнейшего совершенствования web-сайта представляется возможным доработка интерфейса сайта с целью дальнейшего повышения его информативности, привлекательности и удобства.

Тестирование сайта производилось в браузерах: Google Chrome, Яндекс.Интернет.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ:

- 1 «Денвер – локальный сервер» [Электронный ресурс] - режим доступа: <http://www.denwer.ru>;
- 2 «Учебник по интернет-программированию» [Электронный ресурс] – режим доступа: <http://www.des4web.ru/InternetDevelopment/soder.html>;
- 3 Internet-энциклопедия «Википедия» [Электронный ресурс] - режим доступа: <http://ru.wikipedia.org/CMS>;
- 4 Joomla! CMS по-русски. [Электронный ресурс] – режим доступа: <http://joomlaportal.ru/content/view/32/48/>;
- 5 «10 легких шагов к освоению Joomla! 3.0» автор – Хаген Граф, перевод – Алексей Баскинов;
- 6 Видеоуроки от Виталия Куликова – «Joomla 3» [Электронный ресурс] - <https://vitaliykulikov.ru>;
- 7 Видеоуроки от Сергея Патины - Joomla! Ваш первый сайт [2014].



## ПРИЛОЖЕНИЕ А

(обязательное)

Листинг программы

### index.php

```
<?php
    defined('_JEXEC') or die;
    JHtml::_('behavior.framework', true);
    $app = JFactory::getApplication();
?>
<?php echo '<?'; ?>xml version="1.0" encoding="<?php echo $this->_char-
set ?>"?>
    <!DOCTYPE html>
    <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo
$this->language; ?>" lang="<?php echo $this->language; ?>" dir="<?php echo
$this->direction; ?>" >
        <head>
            <jdoc:include type="head" />
            /*Ssylki na stili CSS*/
            <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/sys-
tem/css/system.css" type="text/css" />
            <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/sys-
tem/css/general.css" type="text/css" />
            <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
echo $this->template ?>/css/template.css" type="text/css" />
            <link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php
echo $this->template ?>/css/menu.css" type="text/css" />
        <?php
        if($this->countModules('left and right') == 0) $contentwidth = "100";
        ?>
        <?php $this->setGenerator('mysite'); ?>
        <link rel="icon" href="/templates/trip/favicon.ico" type="image/x-icon">
        <link rel="shortcut icon" href="/template/trip/favicon.ico" type="image/x-
icon">
    </head>
    <body>
        <div id="page">
            <div id="top">
                <div id="logo" style="cursor: pointer;" onclick="window.locat-
tion.href='http://localhost/trip/'">
                    <h1><?php echo $app->getCfg('sitename'); ?></h1>
                </div>
```

```

<div id="user1">
<jdoc:include type="modules" name="user1" style="xhtml" />
</div>
</div>
<?php if($this->countModules('user2')) : ?>
<div id="user2 ">
<jdoc:include type="modules" name="user2" style="xhtml" />
</div>
<?php endif; ?>
<?php if($this->countModules('header')) : ?>
<div id="header ">
<jdoc:include type="modules" name="header" style="xhtml" />
</div>
<?php endif; ?>
<?php if($this->countModules('user3')) : ?>
<div id="user3">
<jdoc:include type="modules" name="user3" style="xhtml" />
</div>
<?php endif; ?>
<div id="content"<?php echo $contentwidth; ?>">
<jdoc:include type="message" />
<jdoc:include type="component" style="xhtml" />
</div>
<?php if($this->countModules('footer')) : ?>
<div id="footer">
<jdoc:include type="modules" name="footer" style="xhtml" />
</div>
<?php endif; ?>
</div>
</body>
</html>

```

### **templateDetails.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<extension
    version="3.5"
    type="template"
    client="site">
    <name>rgns</name>
    <version>1.0</version>
    <creationDate>01/02/16</creationDate>
    <author>Prischep Anastasia</author>

```

```

<authorEmail></authorEmail>
<authorUrl>http://vk.com/id80275126</authorUrl>
<copyright>GNU/GPL</copyright>
<description>simple_template_for_studing</description>
<files>
    <!--Perechen failov i papok shablona-->
    <folder>css</folder>
    <folder>images</folder>
    <folder>html</folder>
    <filename>index.php</filename>
    <filename>favicon.ico</filename>
</files>
<positions>
    <!--Perechen vseh posicij dla modyla-->
    <position>user1</position>
    <position>user2</position>
    <position>user3</position>
    <position>header</position>
    <position>footer</position>
</positions>
</extension>

```

### **template.css**

```

@charset "utf-8";
* {
margin:0;
padding:0;
}

/* nachalnie nastroiiki zagolovkov vseh urovnei */
h1, h2, h3, h4, h5, h6,
h1 a, h2 a, h3 a, h4 a, h5 a, h6 a
h1 a:hover, h2 a:hover, h3 a:hover, h4 a:hover, h5 a:hover, h6 a:hover
h1 a:visited, h2 a:visited, h3 a:visited, h4 a:visited, h5 a:visited, h6 a:visited
{
font-weight: normal;
font-style: normal;
text-decoration: none;
}
/* oformlenie ssylok na saite */
a{
text-decoration: underline;
color:#004f64;
}

```

```

}
/*ssylka neposeshenay */
a:link
{
    color: #004f64;
}
/*ssylka poseshenay */
a:visited{
    color:#3b98b0;
}
/*ssylka pri navedenii*/
a:hover{
    text-decoration: none;
    color: #fe1919;
}

/*zagolovok uroven 1 */
h1, h1 a, h1 a:link, h1 a:visited, h1 a:hover{
font-size:22px;
color: #3b98b0;
text-align:left;
margin-bottom:5px;
}
/*zagolovok uroven 2 */
h2, h2 a, h2 a:link, h2 a:visited, h2 a:hover{
font-size:20px;
color: #004f64;
text-align:left;
border-bottom:1px solid #a7a6aa;
margin-bottom:5px;
}
/*zagolovok uroven 3 */
h3, h3 a, h3 a:link, h3 a:visited, h3 a:hover{
font-size: 18px;
color: #30b2cf;
text-align:left;
}
/* zagolovoki uroveni 4-6 */
h4, h4 a, h4 a:link, h4 a:visited, h4 a:hover,h5, h5 a, h5 a:link, h5 a:visited,
h5 a:hover,h6, h6 a, h6 a:link, h6 a:visited, h6 a:hover{
font-size:16px;
color: #3b98b0;
text-align:left;
}

```

```

/*oformlenie kartinok na saite */
/*kartinka kak ssylka */
a img {
    border: none;
}
/*kartinka v tekste */
p img {
    margin:0;
}

/*oformlenie abzacev na saite */
p {
    margin: 0 0 10px 0;
}

/* blok body */
body {
    margin:0 auto;
    padding:0;
    font:14px Arial, Helvetica, sans-serif ;
    line-height:1.3;
    color:#000;
    text-align:left;
    background-color:#e7e5e5;
    background-image: url('../images/body_bg.png');
    background-repeat:repeat-x;
}

/*blok page */
#page {
    width:950px;
    margin:auto;
    background-color:#fff;
    overflow:hidden;
}

/*blok top dlya razmeshenia div id="logo" div id="user1" */
#top{
    height:60px;
    width:100%;
    margin:0;
}
/* div id="logo"*/

```

```

#logo {
height:60px;
width: 600px;
float: left;
background-image: url('../images/logo.jpg');
}
/*oformlenie nazvania saita */
#logo h1 {
font:bold 24px Arial, Helvetica, sans-serif small-caps;
font-style:italic;
text-align:left;
color:#278cc2;
padding:15px 0 0 100px;
border:none;
}
/*div id="user1" */
#user1 {
height:60px;
width: 350px;
float: left;
background-color:#FFFFFF
}

/*oformlenie modulya poiska po сайту syffiks _poisk */
.moduletable_poisk {
background-color:#fff;
}

/*oformlenie bloka poiska po сайту */
#user1 input {
background:url('../images/search.png') no-repeat;
height:20px;
width:245px;
font:12px Arial, Helvetica, sans-serif;
margin:15px 0 0 80px;
font-style:italic;
border:3px double #3b98b0;
}

/*div id="user2"*/
#user2 {
width: 100%;
clear:both;
}

```

```

/*div id=header */
#header {
width:100%;
}

.moduletable_header{
margin-bottom:0px;
}

/*oformlenie kartinok v module s suffiksom _header */
.moduletable_header img{
margin:0;
}
.moduletable_header p{
margin:0;
}

/*div id="user3" */
#user3 {
width: 100%;
}
/*module News Show Pro GK4 */
.moduletable_gk4{
font:12px Arial, Helvetica, sans-serif ;
color:#000;
font-style:italic;
background:url('../images/user3_bg.png') repeat-y;}

.moduletable_gk4 h4{
padding: 2px 0px;
font:bold 14px Arial, Helvetica, sans-serif;
color:#000;
font-style:italic;}

.moduletable_gk4 h4 a{
padding: 2px 0px;
font:bold 14px Arial, Helvetica, sans-serif;
color:#000;
font-style:italic;}

.moduletable_gk4 a{
font-size:12px;
color:#000;

```

```
padding:5px;
text-decoration:underline;}
```

```
.moduletable_gk4 a:hover {
color:#fe1919;
text-decoration:none;}
```

```
/*vivod contenta razmerom 60% */
#content60 {
float:left;
width:540px;
padding: 0 5px 0 5px;
```

```
}
/*vivod contenta razmerom 80% */
#content80 {
float:left;
width:740px;
padding: 0 5px 0 5px;
}
```

```
/*vivod contenta razmerom 100% */
#content100 {
float:left;
width:940px;
padding: 0 5px 0 5px;
}
```

```
/*oformlenie teksta v statyah */
#content60,#content80,#content100 {
text-align:justify;
margin-top:5px;
}
```

```
#content60 p img,#content80 p img ,#content100 p img {
padding:5px;
}
```

```
/*oformlenie knopok na saite */
.button,input.button,button.validate {
background:#fe1919;
color:#fff;
margin:2px;
padding:2px 10px 2px 10px;
cursor:pointer;
```



```

}

/*oformlenie moduley na saite */
.moduletable {
font:12px Arial, Helvetica, sans-serif;
color:#000;
line-height:1.5;
text-align:left;
padding:0px;
margin:2px;
border:2px solid #e7e5e5;
}

/*oformlenie zagolovka modulia */
.moduletable h3{
font: bold 12px Arial, Helvetica, sans-serif;
text-transform: uppercase;
text-align:left;
padding:5px 0 5px 5px;
color:#fe1919;
background-color:#e7e5e5;
}

/*oformlenie teksta v modulyah */
.moduletable p{
padding:5px;
}

/*oformlenie kartinok v modulyah */
.moduletable img{
margin:2px;
}

/*oformlenie spiskov v modulyah */
.moduletable ul li{
background: url('../images/blok.gif') no-repeat;
padding: 0 0 0 12px;
margin:5px;
list-style:none;
}

/*oformlenie modulya navigacii s suffiksom _nav */
.moduletable _nav {
font-style:italic;

```

```
padding:5px;
margin:0;
background-color:#e7e5e5;
}
```

```
/*div id="footer" */
#footer {
width:100%;
clear:both;
background-color:#80daeb;
}
```

```
/*oformlenie modulya s suffiksom _footer */
.moduletable_foot {
text-align:center;
padding:10px;
}
```

### **menu.css**

```
@charset "utf-8";
```

```
/*gorizontalnoe menu c sufiksom _topmenu */
```

```
.moduletable_topmenu ul {
    display: none;
    background-color: #278cc2;
    position: absolute;
    color: #278cc2;
    float:left;
    left:-4%;}
.moduletable_topmenu li:hover ul { display: block; }
.moduletable_topmenu, .moduletable_topmenu ul {
    margin: 0;
    padding: 0;
    list-style-type: none;}
.moduletable_topmenu {
    font: bold 10px Arial, Helvetica, sans-serif;
    text-align: left;
    text-transform: uppercase;
    height: 33px;
    background-color: #278cc2;}
.moduletable_topmenu li {
    float: left;
```

```

    position: relative;
    height: 100%;}
.moduletable_topmenu li a {
    display: block;
    padding: 5px;
    width: 100px;
    color: #ffffff;
    text-decoration: none;
    text-align: center;
    border-left:2px ridge #fff;
    border-right:2px ridge #fff;}
.moduletable_topmenu ul li {
    list-style:none;
    position:relative;
    float: none;
}
.moduletable_topmenu li:hover { background-color: #278cc2;}
.moduletable_topmenu ul li:hover { background-color: #80daeb;}

```

### default.php

```

<?php
/**
 * @package    Joomla.Site
 * @subpackage Templates.beez3
 *
 * @copyright   Copyright (C) 2005 - 2012 Open Source Matters, Inc. All
rights reserved.
 * @license    GNU General Public License version 2 or later; see LI-
CENSE.txt
 */
defined('_JEXEC') or die;
?>
<div class="breadcrumbs"<?php echo $moduleclass_sfx; ?>">
<?php if ($params->get('showHere', 1))
{
    echo '<span class="showHere">' .JText::_('MOD_BREAD-
CRUMBS_HERE').</span>';
}
?>
<?php for ($i = 0; $i < $count; $i ++):
    // Workaround for duplicate Home when using multilanguage
    if ($i == 1 && !empty($list[$i]->link) && !empty($list[$i - 1]->link)
&& $list[$i]->link == $list[$i - 1]->link) {

```

```

        continue;
    }
    // If not the last item in the breadcrumbs add the separator
    if ($i < $count - 1)
    {
        if (!empty($list[$i]->link)) {
            echo ' <a href="'. $list[$i]->link. '" class="path-
way">'. $list[$i]->name. '</a>';
        } else {
            echo '<span>';
            echo $list[$i]->name;
            echo '</span>';
        }
        if ($i < $count - 2)
        {
            echo ' '.$separator.' ';
        }
    } elseif ($params->get('showLast', 1)) { // when $i == $count -1 and
'showLast' is true
        if ($i > 0)
        {
            echo ' '.$separator.' ';
        }
        echo '<span>';
        echo $list[$i]->name;
        echo '</span>';
    }
endfor; ?>
</div>

```