МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ФГБОУВО «ВГТУ»)

Факультет информационных технологий и компьютерной безопасности

(факультет)

Кафедра систем автоматизированного проектирования информационных систем

КУРСОВАЯ РАБОТА

по дисциплине Методы разметки Web-документов на основе HTML5

Тема Интернет-магазин

Расчетно-пояснительная записка

Разработал студент мРИС-201 А.Р. Марченко

Подпись, дата Инициалы, фамилия

Руководитель Н.А. Рындин

Подпись, дата Инициалы, фамилия

Члены комиссии

Подпись, дата Инициалы, фамилия

Подпись, дата Инициалы, фамилия

Нормоконтролер

Подпись, дата Инициалы, фамилия

Защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата

2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ФГБОУВО «ВГТУ»)

Кафедра систем автоматизированного проектирования и информационных систем

ЗАДАНИЕ

на курсовую работу

по дисциплине Методы разметки Web-документов на основе HTML5

Тема проекта Интернет-магазин

Студент группы мРИС-201 Марченко Алина Романовна

Фамилия, имя, отчество

Технические условия PC с ОС Windows 10, 8 Гб ОЗУ, процессор с тактовой частотой 2,5 ГГц,

Содержание и объем проекта (графические работы, расчеты и прочее)

21 страница, 4 рисунка

Срок защиты курсового проекта 30.06.2021

Руководитель Н.А. Рындин

Подпись, дата Инициалы, фамилия

Задание принял студент А.Р. Марченко

Подпись, дата Инициалы, фамилия

Содержание

[Введение 4](#_Toc75206373)

[1. Теоретическая часть 5](#_Toc75206374)

[1.1 Электронная коммерция 5](#_Toc75206375)

[1.2 Понятие интернет магазина и его преимущества 6](#_Toc75206376)

[1.3 Язык разметки HTML5 9](#_Toc75206377)

[1.4 Каскадные таблицы стилей 11](#_Toc75206378)

[1.5 JavaScript 11](#_Toc75206379)

[2. Практическая часть 13](#_Toc75206381)

[2.1 Создание страниц сайта 13](#_Toc75206382)

[2.2 HTML 14](#_Toc75206383)

[2.3 CSS 16](#_Toc75206380)

[2.4 JavaScript 18](#_Toc75206380)

[Заключение 20](#_Toc75206384)

[Список литературы 21](#_Toc75206385)

[Приложение А 22](#_Toc75206386)

Введение

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

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

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

Целью данной курсовой работы является рассмотрение web-ориентированных языков программирования, языка html разметки, а также разработка интернет-магазина.

1. Теоретическая часть
   1. Электронная коммерция

Интернет-торговля — это динамично развивающаяся отрасль во всем мире. Благодаря активному распространению широкополосного доступа к сети Интернет на быстроразвивающихся рынках, перед компаниями сектора розничной торговли открываются совершенно новые рынки. Интернет также предоставляет все больше возможностей потребителю для изучения продукции, так как потребитель может быстро ознакомиться с продукцией и определить, является ли товар, предлагаемый местными торговыми компаниями, продукцией высокого или низкого качества. Таким образом, продвижение бизнеса в Интернете и продажи через Интернет могут стать «и благом, и проклятьем».

Рынок интернет-торговли в России не так развит, как в странах Западной Европы или Северной Америки. Чтобы получить более четкое представление о российском рынке, объем которого оценивается более чем в 100 млрд. рублей, компания Pricewaterhouse Coopers провела исследование, в ходе которого был получен ряд интересных результатов, в частности:

- 80% респондентов когда-либо совершали покупку в интернет-магазине;

- Самой популярной формой оплаты до сих пор остается оплата наличными при доставке товара, хотя все шире применяется и такая форма оплаты, как электронные платежи;

- Неуверенность в надежности продавца и качестве услуг по доставке товара — основные факторы, удерживающие потенциальных клиентов интернет-магазинов от покупок в Сети;

- Социальные медиа оказывают значительное влияние на решение интернет-пользователей о совершении той или иной покупки. Члены социальной сети гораздо больше доверяют мнению других ее участников о продуктах и производителях, чем рекламе.

Электронная коммерция — это сфера экономики, которая включает в себя все финансовые и торговые транзакции, осуществляемые при помощи компьютерных сетей, и бизнес-процессы, связанные с проведением таких транзакций. Если упростить, то любые действия с коммерческим уклоном в глобальной сети подпадают под определение онлайн-коммерция. Эта сфера зародилась в США, потом получила развитие в Европе и в конце 90-х годов прошлого века стала активно развиваться в Китае и России.

Электронная коммерция предоставляет огромные возможности для развития своего дела. Однако для того чтобы использовать ее действительно эффективно мало быть талантливым предпринимателем, точно так же, как и мало уметь ориентироваться в безграничном виртуальном пространстве. Необходимо еще обладать рядом высокотехнологичных инструментов для ведения электронного бизнеса. Одним из них и является интернет-магазин.

* 1. Понятие интернет магазина и его преимущества

Интернет-магазин является одной из разновидностей электронной коммерции. Продажа собственных товаров и услуг через интернет – основной способ коммерческого использования сайтов для большинства производственных и торговых предприятий. Компания создает сайт и размещает на нем информацию о своих товарах и услугах, ценах и гарантиях для покупателей. Интернет-магазины становятся одним из необходимых и эффективных инструментов увеличения числа продаж предприятия, увеличения оборота и повышения имиджа, успешного развития. Существует несколько определений «что такое интернет-магазин».

Интернет-магазин (англ. online shop или e-shop) - специализированный вид бизнеса, основанный на дистанционном обслуживании и дающий возможность продавцу и покупателю совершать сделки по покупке-продаже того или иного вида товаров посредством сети Интернет.

Интернет-магазин - это интерактивный веб-сайт, рекламирующий товар или услугу, принимающий заказы на покупку, предлагающий пользователю выбор варианта расчета, способа получения заказа и выписывающий счет на оплату.

Интернет-магазин - это совокупность программ, работающих на Web-сайте, которые позволяют покупателю дистанционно выбрать товар из каталога и оформить его заказ.

Интернет-магазин является посредником между покупателем и складом производителя, которому достаточно иметь ресурс в Интернете и телефон. Магазин виртуального типа не нуждается в представительном офисе, складских помещениях, не несет расходы на содержание помещений, владелец магазина может управлять продажами даже из своей квартиры. Именно потому цены в таких магазинах минимальны. Купленные вещи подвозят потребителю непосредственно со склада производителя.

Интернет-магазин обладает рядом преимуществ по сравнению с обычными магазинами:

1. Интернет-магазин работает 24 часа в сутки, 365 дней в году, без перерыва на обед, без выходных и праздничных дней.

2. Необязательно иметь в наличии товар. Но в этом случае обязательно иметь сеть поставщиков, работающих по принципу «точно вовремя». Это гарантирует доставку товара в заранее оговоренные минимальные сроки.

3. Не нужно помещение для продаж. Витрины, стеллажи, прилавки, квадратные метры и арендная плата – на всем этом можно существенно сэкономить. На чем сэкономить нельзя – так это на качественно выполненном дизайне сайта для интернет-магазина, поскольку именно внешний вид веб-страниц будет заменять покупателям и стеллажи, и витрины.

4. Срок и стоимость создания интернет-магазина несоизмеримо ниже, чем обычного магазина.

5. Свобода передвижений продавца. Поскольку интернет-магазин – это точка продаж в виртуальном пространстве, вы можете управлять им из любого уголка земного шара, где есть доступ в Интернет.

6. С интернет-магазином вы получаете возможность расширить географию своего бизнеса вплоть до мировых рынков. С оговоркой, что содержание сайта будет понятно даже иностранцам.

7. Можно не нанимать продавцов, администраторов, менеджеров, товароведов, кассиров, охранников и еще кого бы то ни было. Профессионально грамотно созданный интернет-магазин может работать полностью автономно.

8. Интернет-магазин дает вам возможность самим распределять свое рабочее время. Вы можете тратить на его обслуживание столько часов в сутки, сколько захотите – один, два, а то и все двадцать четыре. И когда хотите – ночью, днем, утром, вечером.

Однако помимо преимуществ ведения торговли через интернет-магазин есть и существенные риски:

1. Атака хакеров. В «лучшем» – внесут путаницу в данные о заказах, отправках, расчетах, и Вам будет необходимо объясняться со всеми своими клиентами на предмет «Извините, это наша ошибка». В худшем случае у Вас могут украсть базу данных с номерами кредитных карт клиентов, а Вы погрязнете в судебных разбирательствах. Однако в Интернете есть средства обеспечивающие безопасность электронной торговли, поэтому имеется возможность интегрировать в интернет-магазин новейшие защитные разработки.

2. «Баги». Это ошибки в программном обеспечении, сделанные самими разработчиками. Ошибки есть всегда. В любом программном обеспечении для интернет-торговли. Это может привести к тому, что в определенный момент, при определенных условиях Ваш магазин начинает «глючить» или вообще «откажется» работать. Проблему с багами помогает решить четко функционирующая служба технической поддержки при компании-разработчике.

3. Клиенты легко приходят и уходят. К счастью, в интернет-магазин не надо «идти» – важно знать адрес. Но, к сожалению, чтобы его покинуть, также достаточно одного щелчка «мышью». Все, чем Вы можете привлечь клиентов на свой сайт – это оригинальный дизайн, выгодные предложения, хорошие тексты-описания товаров. Никаких эмоций, никакого общения, никаких старых знакомых клиентов. И никакой быстрой реакции на ситуацию. Однако и эта проблема в последнее время становится разрешимой. Появляется такое программное обеспечение, которое позволяет буквально отслеживать посетителей вашего сайта – по полу, возрасту, профессии, месту жительства и т. п., а также добавляется интерактивность.

Тем не менее, интернет-магазины завоевывают всё больше доверия и популярность, и, несомненно, их ожидает большое будущее.

* 1. Язык разметки HTML5

Выбор технологии программирования и разработки – стартовая точка в реализации любого веб-проекта. От этого зависит функциональность и удобство будущего ресурса.

Каждый веб-ресурс разрабатывается для упрощения какой-либо работы и удобства пользованием, поэтому первым мы рассмотрим клиентские технологии.

Основополагающим является HTML – это язык гипертекстовой разметки, стандартизированный язык разметки в сети интернет. Большинство веб-ресурсов содержат описание разметки страниц на HTML или XHTML.

Полученная в результате интерпретации браузерами языка HTML страница отображается на экране монитора компьютера или мобильного устройства.

HTML5 — это не продолжатель языка разметки гипертекста, а новая открытая платформа, предназначенная для создания веб-приложений использующих аудио, видео, графику, анимацию и многое другое.

HTML давно перестал быть просто языком программирования. Человек, изучавший этот язык, обретает возможность делать сложные вещи простыми способами и, главное, быстро, что в компьютерном мире не так уж и мало. Гипертекст подходит для включения элементов мультимедиа в традиционные документы. Практически именно благодаря развитию гипертекста, большинство пользователей получило возможность создавать собственные мультимедийные продукты. Такие информационные системы, выполненные в виде набора HTML-страниц, не требует разработки специальных программных средств, так как все необходимые инструменты для работы с данными (WEB-браузеры) стали частью стандартного программного обеспечения большинства персональных компьютеров. От пользователя требуется выполнить только ту работу, которая относится к тематике разрабатываемого продукта: подготовить тексты, нарисовать рисунки, создать HTML-страницы и продумать связь между ними.

Особенностями языка разметки являются:

* Простота в освоении и использовании: HTML имеет некоторые теги, которые служат определенной цели. Один может легко понять код другого и внести в него изменения, если это необходимо, так как в нем несложно разобраться.
* Доступность: одним из самых больших преимуществ HTML является то, что он бесплатен и нет необходимости приобретать конкретное программное обеспечение. Не следует иметь дело с различными плагинами, необходимыми для работы на любом программном обеспечении, поскольку HTML не требует никаких плагинов. Таким образом, это очень экономично в соответствии с бизнес-перспективой, поскольку нет никаких затрат на покупку лицензии, если весь сайт разрабатывается на языке HTML;
* Кросс-браузерность: почти все браузеры по всему миру поддерживаются HTML. Таким образом, нет необходимости беспокоиться о веб-сайте, написанном в HTML для поддержки браузера, поскольку веб-сайт будет легко отображаться во всех браузерах.
* Интеграция с другими языками: HTML может быть легко интегрирован с несколькими языками. Например, JavaScript, PHP, Node.js, CSS и многое другое, мы пишем код этих языков между HTML, и он очень легко смешивается с ними.
  1. Каскадные таблицы стилей

Аббревиатура CSS расшифровывается как Cascading Style Sheets, что в переводе означает «каскадные таблицы стилей». Это язык разметки, используемый для визуального оформления веб-сайтов.

Объекты, расположенные на странице, размещаются с помощью HTML. А вот CSS отвечает за то, как эти объекты выглядят. Их размер, цвет, фоновое изображение, степень прозрачности, расположение относительно других элементов, поведение при наведении курсора, визуальное изменение кнопок при нажатии и т.п.

Язык CSS быстро стал стандартом в веб-разработке, потому что он позволяет быстро изменить визуальное оформление сайта, не прибегая к использованию более сложных языков программирования.

Применение HTML в совокупности с CSS куда лучше, чем просто HTML, так как вы получите ряд преимуществ:

* Детально разработанный дизайн.
* Возможность управления любыми документами при помощи единой таблицы.
* Доступность функции настройки отображения веб-страницы для всех современных устройств, начиная от ПК, и заканчивая различными гаджетами.
  1. JavaScript

Интерактивные элементы сайтов и мобильных приложений часто выполняются на языке JavaScript. Он хорошо интегрируется с кодом HTML/CSS, поддерживается основными браузерами и включен в них по умолчанию. Поэтому никаких вопросов с запуском веб-ресурсов не возникает, они работают без участия пользователя.

Инструмент JavaScript (сокращенно JS) относится к языкам программирования высокого уровня с возможностью встраивания в другие приложения. Все типы функциональных модулей создаются в виде сценариев. По синтаксису он схож с языком Java, но общая у них только часть наименования. Платформа регулярно обновляется, код становится все более рациональным.

Пользователь любым своим действием на сайте вызывает событие, будь это прокрутка страницы, клик по пункту меню или перемещение курсора «внутри экрана». На этом и базируется применение скриптов, которые запросто «вкладываются» друг в друга, если некоторые события предполагают разную реакцию браузера.

Реагирование на событие становится возможным, если программист внес в шаблон сайта (страницы) код JS-скрипта и поставил ссылку на него в нужном месте (команда onclick). Если этого не сделать, ничего не произойдет. Те же формы станут «обычными картинками».

Возможностями языка JS являются:

* Динамическое изменение страниц, добавление текста или тегов, изменение стилей элементов;
* Реакция на определенные события, например, скрипт может ожидать определенного действия (клик мыши, загрузка страницы) и реагировать на него выполнением нужной функции;
* Выполнение запросов и загрузка данных без перезагрузки всей страницы целиком;
* Устанавливать и считывать небольшой фрагмент данных (cookie), отправленный веб-сервером и хранимый на компьютере пользователя, вывод сообщений и множество других возможностей.

JavaScript полностью интегрирован с браузером и поддерживается почти везде. Такие технологии как Flash, Silverlight, Java, не полностью интегрированы с браузером, но и работают в своем окружении.

1. Практическая часть
   1. Создание страниц сайта

Разрабатываемый магазин компьютерных игр имеет страницу с контактными данными и каталог с играми, а также корзину.

Блок «О нас» содержит текстовую информацию.

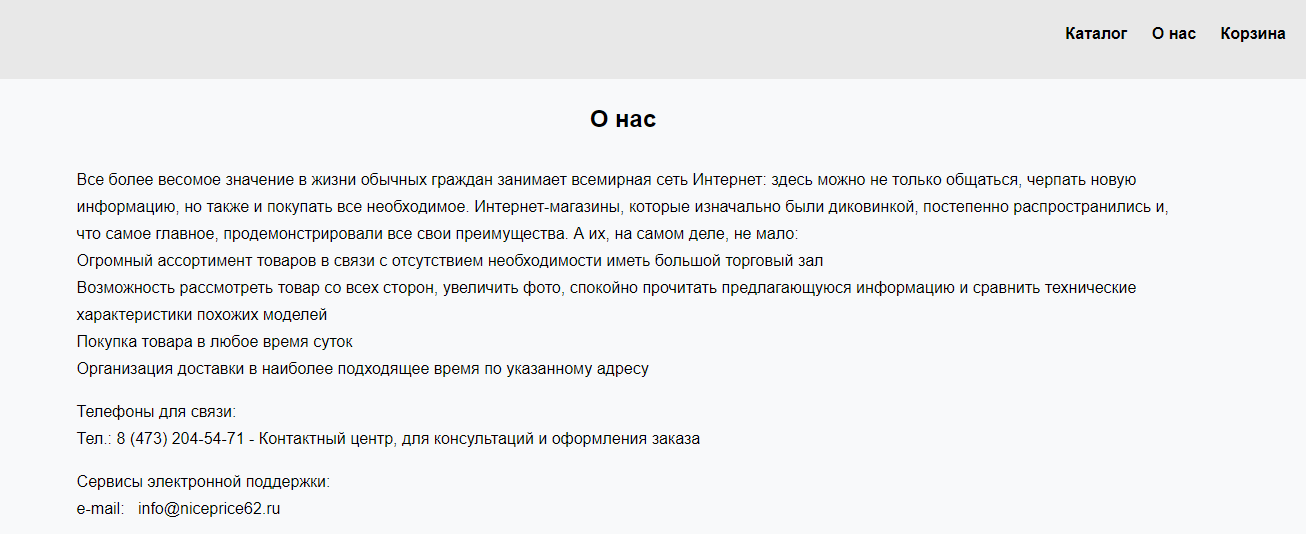


Рисунок 1 – Блок О Нас

Теперь посмотрим на каталог игр.

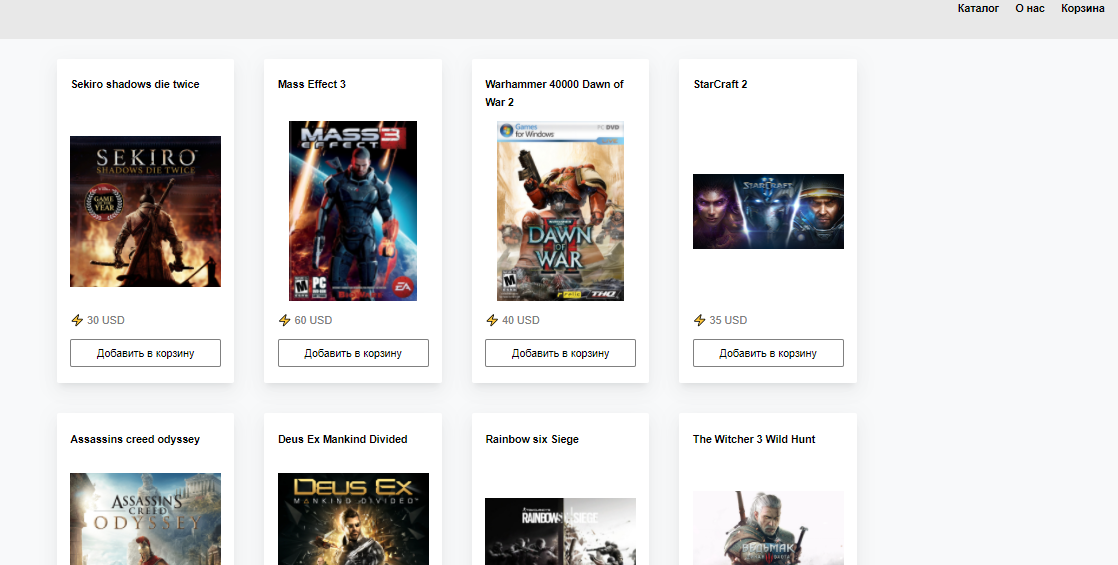


Рисунок 2 – Каталог игр

Здесь видим товары, по 3 в ряд, есть названия игр и их цена. У каждого товара имеется кнопка «Добавить в корзину», при нажатии на которую данная кнопка изменяется на «Удалить из корзины», а количество товаров в корзине пополняется.

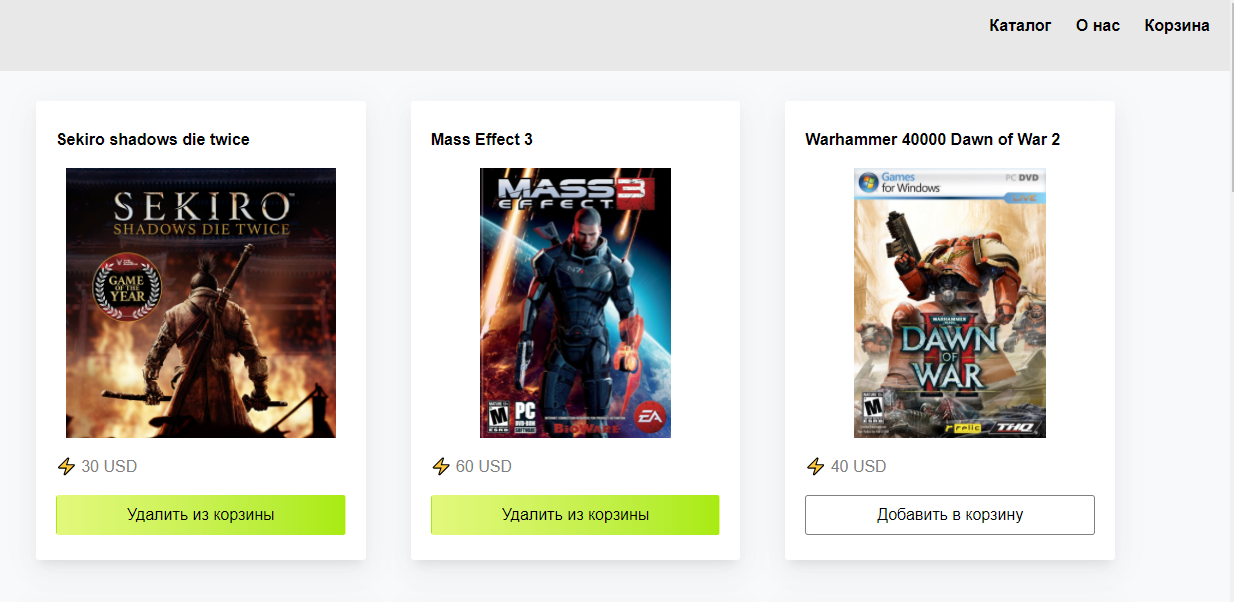


Рисунок 3 – Товар в корзине

И товар появился в корзине.

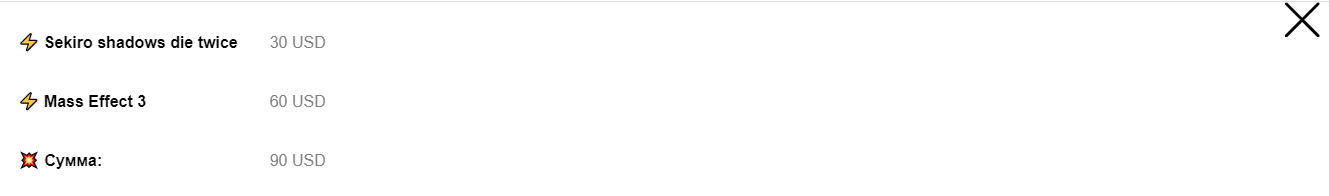


Рисунок 4 – Корзина

* 1. HTML

При написании HTML файлов курсовой работы использовались такие основные теги как:

<!--...--> — тег для добавления комментариев в документ. Помещённые внутри него теги не интерпретируются браузером.

<!DOCTYPE> — показывает браузеру тип документа, сообщает его версию и язык.

<html></html> — корневой тег, который сообщает браузеру, что это HTML-документ. Все остальные элементы помещаются внутри него.

<head></head>— контейнер, в который помещаются метаданные документа, не видимые пользователям, но считываемые поисковыми роботами: например, <title> или <style>.

<meta> — тег для оформления метаданных документа, используемых браузером для обработки страницы, а поисковиками — для индексации.

<body></body> — тег, обрамляющий видимую пользователям часть документа. Всё, что указано внутри этого контейнера, отобразится на странице. Тег <body> имеет несколько атрибутов, позволяющих управлять цветами.

<title></title>— метатег, который задаёт название страницы, отображаемое на вкладке браузера.

<header></header> — определяет содержимое блока с вводной информацией сайта или группой ссылок.

<h1></h1>......<h6></h6>. — теги заголовков, от самого большого к самому маленькому.

<b></b> — жирный текст для придания важности выделенному фрагменту.

<p></p> — контейнер для абзаца.

<br> — переносит текст на другую строку без создания абзаца.

<ul> </ul> - используется для создания маркированных списков.

<li> </li> — формирование элемента списка (цифра или значок в зависимости от типа списка).

<a></a> — добавление гиперссылки в текст. Имеет обязательный атрибут href, в котором указывается ссылка или якорь. Внутри контейнера помещается текст, при нажатии на который происходит переход на другую страницу или другое место на этой же странице.

<img></img> — вставка изображения. Имеет атрибут src, который указывает на адрес нужного файла. Есть и другие атрибуты:

<script></script> — определяет выполнение сценария на стороне посетителя сайта. Может содержать сам скрипт или иметь ссылку на  внешний источник.

<table></table> — размещает таблицу. Все остальные теги для форматирования таблиц должны находиться внутри это контейнера.

<thead></thead> — определяет заголовок.

<td></td> — создаёт одну ячейку.  
 <th></th> — указывает на заголовок ячейки.

<tr> — создание одной строки.

<form></form> — создание форм на странице. С помощью HTML описывается только внешний вид, для выполнения функций требуется запуск соответствующих скриптов на сервере.

<button></button> — создаёт интерактивную кнопку. Внутри контейнера нужно поместить текст или изображение.

<div> — является блочным элементом и предназначен для выделения фрагмента документа с целью изменения вида содержимого. Как правило, вид блока управляется с помощью стилей. Чтобы не описывать каждый раз стиль внутри тега, можно выделить стиль во внешнюю таблицу стилей, а для тега добавить атрибут class или id с именем селектора.

<link> — Устанавливает связь с внешним документом вроде файла со стилями или со шрифтами. В отличие от тега <a>, тег <link> размещается всегда внутри контейнера <head> и не создает ссылку.

# CSS

Все стили были подключены с помощью <link> и каждому HTML файлу соответствует свой собственный CSS файл.

<link rel="stylesheet" href="Главная.css" media="screen">

# Основные стили CSS используемые в данной курсовой:

font - универсальное свойство, которое позволяет одновременно задать несколько характеристик шрифта и текста.В качестве обязательных значений свойства font указывается размер шрифта и его семейство.

**Color** — Данный стиль задает цвет текста. Для задания цвета можно использовать как хекс-значение цвета (color:#FFF), так и ряд ключевых слов (color:black, color:red ...)

**text-align** — Данный стиль задает выравнивание текста внутри родительского блока. Может иметь значения left, right, center.

**line-height** — Данный стиль задает расстояние между строк в текстовом блоке.

**letter-spacing** — межсимвольное расстояние. Значение данного свойства указывает в единицах длины (пиксели, дюймы, pt), либо относительные единицы - em.

**font-weight** — Чтобы сделать текст блока жирным - используется стиль font-weight:bold. Чтобы убрать жирное выделение - font-weight:normal.

position:absolute — Данное свойство вырывает блок (применять это свой свойство нужно к блоку div) из нормального потока формирования страницы. И далее блок позиционируется относительно верхнего угла браузера путем указания свойств left и top в процентах или пикселях.

Используя абсолютное позиционирование, чтобы избежать непредвиденных обстоятельств, необходимо задать значения ширины и высоты блока в пикселях (свойства width и height).

position:relative — Разновидность абсолютного позиционирования. В данном случае блок смещается заданием значений left и top относительно места своего нормального положения. То есть блок выводится там, где он и должен быть в нормальном потоке и сдвигается на заданные значения.

Если у родительского блока указан стиль position:relative, то вложенный блок с указанным стилем position:absolute будет смещаться относительно левого верхнего угла родительского блока.

Padding — Данное свойство формирует отступ внутри блока (значение в пикселах).

Margin — задает отступы от границ блока.

Cursor — устанавливает форму курсора, когда он находится в пределах элемента. Вид курсора зависит от операционной системы и установленных параметров.

box-shadow — добавляет тень к элементу. Допускается использовать несколько теней, указывая их параметры через запятую, при наложении теней первая тень в списке будет выше, последняя ниже.

background — Универсальное свойство  позволяет установить одновременно до пяти характеристик фона. Значения могут идти в любом порядке, браузер сам определит, какое из них соответствует нужному свойству. В CSS3 допустимо указывать параметры сразу нескольких фонов, перечисляя их через запятую.

# JavaScript

С помощью Javascript была реализована корзина покупок. Скрипт содержит 3 основных раздела:

Header – отвечает за шапку, где содержится ссылка на корзину а также счётчик товаров в корзине.

Products – служит для отображения каталога товаров на странице, а так же со

Shopping – раздел корзины, отображает каталог товаров, которые были занесены в корзину.

Массив с товарами находится в файле catalog.js и содержит:

id – уникальный идентификатор.

name – наименование товара.

img – ссылка на картинку для товара.

price – цена товара.

Поскольку данная курсовая работа не предусматривает использование серверной части, то для хранения информации о выбранных товарах будет использовано локальное хранилище localStorage.

Общая логика работы:

Когда пользователь нажимает на кнопку «Добавить в корзину», то создаётся локальное хранилище, в которое будет помещено название товара и его цена. После этого кнопка у товара изменит свой вид.

Если локальное хранилище не является пустым, то при загрузке страницы происходит обработка массива добавленных товаров с помощью функции forEach, важно заметить, что перед работой с хранилищем необходимо его преобразовать с помощью функции JSON.parse, поскольку оно хранится в текстовом формате. Если локальное хранилище пусто, то в корзину не будут выведены товары.

На каждой итерации цикла прохода по массиву будет создана строка таблицы с названием товаров, ценой. А к итоговой сумме цен товаров в корзине приплюсуется цена товара.

Кнопка «Удалить из корзины» осуществляет удаление товара из корзины и локального хранилища.

Заключение

В ходе выполнения курсовой работы были проанализированы и изучены средства разработки web-сайтов.

- Язык разметки HTML версии 5, с помощью которого были разработаны страницы нашего интернет-магазина;

- Каскадные стили таблиц, благодаря которым был реализован внешний вид;

- Язык сценариев JavaScript был использован для придания динамичности и реализации работы корзины магазина с использованием локального хранилища пользователя.

В результате был разработан интернет-магазин компьютерных игр, который имеет возможность взаимодействия с товарами, добавление их в корзину.

Список литературы

1. Брезгунова И. В. Интернет–технологии для пользователя.— Мн.: 2001,

205 с.

2. Климченя Л.С. Электронная коммерция.— Мн.: 2004- 412 с.

3. Коляда А. Маркетинг, который работает. Создание эффективной маркетинговой стратегии в отечественных условиях, Мн.: Изд. Гревцова, 2006 – 240с

4. Гаевский, А.Ю., Романовский, В.А. Самоучитель по созданию Web-страниц: HTML, JavaScript и Dynamic HTML. – К. : А.С.К., 2015г.

5. Панфилов К. По ту сторону веб-страницы. – СПб.: ДМК Пресс, 2011г.

6. Пауэл, Т. HTML: справочник программиста / Т.А. Пауэл, Д. Уитворт. – М.: АСТ, 2015.

7. Печников В.Н. Самоучитель Web-страниц и Wеb-сайтов. – М.: Триумф, 2014г.

Приложение А

Index.html

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

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

<title>Game Shop</title>

<link href="img/favicon.png" rel="icon" type="image/png">

<link href="css/index.css" rel="stylesheet">

</head>

<body>

<div id="header"></div>

<div id="products"></div>

<div id="shopping"></div><!-- Constants and Utils -->

<script src="constants/catalog.js">

</script>

<script src="constants/root.js">

</script>

<script src="utils/localStorageUtil.js">

</script> <!-- Header -->

<script src="components/Header/Header.js">

</script>

<link href="components/Header/Header.css" rel="stylesheet"><!-- Products -->

<script src="components/Products/Products.js">

</script>

<link href="components/Products/Products.css" rel="stylesheet"><!-- Shopping -->

<script src="components/Shopping/Shopping.js">

</script>

<link href="components/Shopping/Shopping.css" rel="stylesheet">

<footer class="header-container1">

<p align="center">Администрация не несет ответственности за содержание. Просьба не ругаться</p>

</footer>

</body>

</html>

Products.js

class Products {

constructor() {

this.classNameActive = 'products-element\_\_btn\_active';

this.labelAdd = 'Добавить в корзину';

this.labelRemove = 'Удалить из корзины';

}

handleSetLocationStorage(element, id) {

const {

pushProduct,

products

} = localStorageUtil.putProducts(id);

if (pushProduct) {

element.classList.add(this.classNameActive);

element.innerHTML = this.labelRemove;

} else {

element.classList.remove(this.classNameActive);

element.innerHTML = this.labelAdd;

}

headerPage.render(products.length);

}

render() {

const productsStore = localStorageUtil.getProducts();

let htmlCatalog = '';

CATALOG.forEach(({

id,

name,

price,

img

}) => {

let activeClass = '';

let activeText = '';

if (productsStore.indexOf(id) === -1) {

activeText = this.labelAdd;

} else {

activeClass = ' ' + this.classNameActive;

activeText = this.labelRemove;

}

htmlCatalog += `

<li class="products-element">

<span class="products-element\_\_name">${name}</span>

<img class="products-element\_\_img" src="${img}" />

<span class="products-element\_\_price">

⚡️ ${price.toLocaleString()} USD

</span>

<button class="products-element\_\_btn${activeClass}" onclick="productsPage.handleSetLocationStorage(this, '${id}');">

${activeText}

</button>

</li>

`;

});

const html = `

<ul class="products-container">

${htmlCatalog}

</ul>

`;

ROOT\_PRODUCTS.innerHTML = html;

}

}

const productsPage = new Products();

productsPage.render();

Shopping.js

class Shopping {

handleClear() {

ROOT\_SHOPPING.innerHTML = '';

}

render() {

const productsStore = localStorageUtil.getProducts();

let htmlCatalog = '';

let sumCatalog = 0;

CATALOG.forEach(({ id, name, price }) => {

if (productsStore.indexOf(id) !== -1) {

htmlCatalog += `

<tr>

<td class="shopping-element\_\_name"> ${name}</td>

<td class="shopping-element\_\_price">${price.toLocaleString()} USD</td>

</tr>

`;

sumCatalog += price;

}

});

const html = `

<div class="shopping-container">

<div class="shopping\_\_close" onclick="shoppingPage.handleClear();"></div>

<table>

${htmlCatalog}

<tr>

<td class="shopping-element\_\_name">Сумма:</td>

<td class="shopping-element\_\_price">${sumCatalog.toLocaleString()} USD</td>

</tr>

</table>

</div>

`;

ROOT\_SHOPPING.innerHTML = html;

}

}

const shoppingPage = new Shopping();

Catalog.js

const CATALOG = [

{

id: 'el1',

name: 'Sekiro shadows die twice',

img: 'https://image.api.playstation.com/vulcan/img/rnd/202010/2723/knxU5uU5aKvQChKX5OvWtSGC.png',

price: 30,

},

{

id: 'el2',

name: 'Mass Effect 3',

img: 'https://static.wikia.nocookie.net/masseffect/images/b/be/ME3\_Cover\_Art.png/revision/latest?cb=20120711194022',

price: 60,

},

{

id: 'el3',

name: 'Warhammer 40000 Dawn of War 2',

img: 'https://upload.wikimedia.org/wikipedia/ru/4/4d/Warhammer40kdow2\_boxart.jpg',

price: 40,

},

{

id: 'el4',

name: 'StarCraft 2',

img: 'https://bnetcmsus-a.akamaihd.net/cms/blog\_header/2g/2G4VZH5TIWJF1602720144046.jpg',

price: 35,

},

{

id: 'el5',

name: 'Assassins creed odyssey',

img: 'https://image.api.playstation.com/cdn/EP0001/CUSA12042\_00/mvdOphV0uyg1PcARccoFZnrWWZG5EGeU.png',

price: 60,

},

{

id: 'el6',

name: 'Deus Ex Mankind Divided',

img: 'https://image.api.playstation.com/cdn/EP0082/CUSA01836\_00/6JDF9vIBtjnoSwVbjcXaBmj1GrNRdqGl.png',

price: 5,

},

{

id: 'el7',

name: 'Rainbow six Siege',

img: 'https://cdnimg.rg.ru/img/content/190/02/13/Rainbow\_Six\_Siege\_t\_650x433.jpg',

price: 10,

},

{

id: 'el8',

name: 'The Witcher 3 Wild Hunt ',

img: 'https://image.api.playstation.com/vulcan/img/rnd/202009/2913/09xDToHiETDsJxINSzIVOQ4t.jpg',

price: 20,

},

{

id: 'el9',

name: 'Battlefield 2042',

img: 'https://image.api.playstation.com/vulcan/ap/rnd/202106/0123/jLyvJuE1N7lDHpWqG0kHhaWd.png',

price: 60,

},

{

id: 'el10',

name: 'Cyberpunk 2077',

img: 'https://newxboxone.ru/wp-content/uploads/2020/12/ecd9803e87d6025fd76b.jpg',

price: 50,

}

];

LocalStorageUtil.js

class LocalStorageUtil {

constructor() {

this.keyName = 'products';

}

getProducts() {

const productsLocalStorage = localStorage.getItem(this.keyName);

if (productsLocalStorage !== null) {

return JSON.parse(productsLocalStorage);

}

return [];

}

putProducts(id) {

let products = this.getProducts();

let pushProduct = false;

const index = products.indexOf(id);

if (index === -1) {

products.push(id);

pushProduct = true;

} else {

products.splice(index, 1);

}

localStorage.setItem(this.keyName, JSON.stringify(products));

return {

pushProduct,

products

}

}

}

const localStorageUtil = new LocalStorageUtil();