МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ВИТЕБСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ П.М. МАШЕРОВА»

Факультет математики и информационных технологий

Кафедра информатики и информационных технологий

Допущена к защите

«\_\_» \_\_\_\_\_\_\_\_ 20\_\_ г.

Заведующий кафедрой

\_\_\_\_\_\_\_ Е.А.Витько

**ДИПЛОМНЫЙ ПРОЕКТ**

Проектирование и практическая реализация интернет-магазина

Специальность 1-40 01 01 Прогрммное обеспечение информационных технологий

Моисейченков Дмитрий Юрьевич

4 курс, 44 группа

Руководитель:

Шедько Василий Викторович

старший преподаватель

Витебск, 2020

# Реферат

Курсовой проект 0 с., 0 рис., 0 источников, 0 прил.

ИНТЕРНЕТ-МАГАЗИН, ПРИЛОЖЕНИЕ, ИНТЕРФЕЙС, БАЗА ДАННЫХ, MYSQL, PHP

Объект исследования – технологии создания современных распределённых приложений.

Предмет исследования − технологии разработки интерактивного интернет-магазина.

Цель работы – изучить различные методы, технологии по созданию распределённых приложений, создать и реализовать клиентскую и серверную часть интернет-магазина, продемонстрировать их работоспособность посредством тестирования.

Методы исследования: анализ, синтез, изучение литературы, практическая реализация.

Результатом является web-приложение, позволяющее пользователям совершать выбор товара и его покупку в интернет-магазине.

Теоретическая и практическая значимость. Теоретическая значимость заключается в получении навыков работы с современными технологиями распределённых приложений, обобщении и систематизации знаний, полученных в рамках выполнения проекта. Практическая значимость заключается в возможности применения разрабатываемого интернет-магазина на реальных платформах.

# Обозначения и сокращения

БД – база данных.

СУБД – система управления базой данных.

URL – Uniform Resource Locator (единообразный локатор ресурса).

UI – User Interface (пользовательский интерфейс).

CSS – Cascading Style Sheets (каскадные таблицы стилей).

Scss – Sassy CSS  ([метаязык](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%B0%D1%8F%D0%B7%D1%8B%D0%BA) на основе [CSS](https://ru.wikipedia.org/wiki/CSS)).

HTML – HyperText Markup Language (язык гипертекстовой разметки).

JavaScript – [мультипарадигменный](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BF%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F).

PHP – Hypertext Preprocessor ([препроцессор](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80) [гипертекста](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82)).

Gulp – task-менеджер для автоматического выполнения часто используемых задач.

SQL – Structured Query Language (язык структурированных запросов).

MySQL –  [свободная](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%9F%D0%9E) [реляционная система управления базами данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94).

phpMyAdmin – [веб-приложение](https://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) с открытым кодом, реализованное на языке [PHP](https://ru.wikipedia.org/wiki/PHP) и представляющее собой веб-интерфейс для администрирования СУБД MySQL.

Содержание

[Реферат 2](#_Toc39505499)

[Обозначения и сокращения 3](#_Toc39505500)

[Введение 5](#_Toc39505501)

[1.Предметная область 7](#_Toc39505502)

[1.1 Постановка задачи 7](#_Toc39505503)

[1.2 Выбор технологий 7](#_Toc39505504)

[2. Программное проектирование серверной части интернет-магазина 10](#_Toc39505505)

[3. Практическая реализация серверной части интернет-магазина 11](#_Toc39505506)

[3.1 Создание базы данных 12](#_Toc39505507)

[3.2 Создание сессии 13](#_Toc39505508)

[3.3 Подключение к базе данных 14](#_Toc39505509)

[3.4 Регистрация пользователя 14](#_Toc39505510)

[3.5 Создание фильтра товаров на главной странице магазина 16](#_Toc39505511)

[3.6 Авторизация пользователя 17](#_Toc39505512)

[4. Тестирование работы серверной части интернет-магазина 19](#_Toc39505513)

[4.1 Результаты тестирования 21](#_Toc39505514)

[Заключение 22](#_Toc39505515)

[Список использованных источников 24](#_Toc39505516)

[ПРИЛОЖЕНИЕ А 25](#_Toc39505517)

# Введение

Электронный магазин — это прикладная система, построенная с использованием технологии системы электронной торговли. Подобно обычному магазину, электронный магазин реализует следующие основные функции: представление товаров (услуг) покупателю, обработку заказов, продажу и доставку товаров. Однако электронный магазин требует от программистов-разработчиков определённых навыков работы с такими системами и способностей ими управлять. Здесь, самая главная проблема-учесть все необходимые нюансы на стадии разработки программного обеспечения и, если вдруг понадобится что-то новое, например задача от заказчика, – суметь сделать этот функционал уже на стадии работы с интернет -магазином в интернете. В данном дипломном проекте будут приложены все усилия, чтобы проблема с разработкой многофункционального интернет-магазина была решена современными методами разработки ПО.

Современное информационное поле состоит из массы событий, объектов и явлений. Открытое программное обеспечение стало основным структурным элементом при создании некоторых крупнейших веб-сайтов. С началом стремительного роста этих веб-сайтов возникли практические передовые методы и основные руководящие принципы их архитектуры. Данный дипломный проект стремится охватить некоторые ключевые вопросы, которые следует учитывать при проектировании больших веб-сайтов, интернет-магазинов, а также некоторые базовые компоненты, используемые для достижения этих же целей.

Что именно означает создание и управление масштабируемым веб-сайтом или приложением? На примитивном уровне это просто соединение пользователей с удаленными ресурсами через Интернет. А ресурсы или доступ к этим ресурсам, которые рассредоточены на множестве серверов и являются звеном, обеспечивающим масштабируемость веб-сайта.

Выбранная тема считается актуальной на сегодняшний день, так как сегодня миллионы людей ежедневно, не выходя из дому, покупают различные товары в электронных магазинах. Поэтому, с точки зрения разработки программного обеспечения, интернет-магазин сегодня является действительно удобным интерактивным приложением с современным интерфейсом и методами разработки. Сегодня существует огромное количество языков программирования, которые с лёгкостью справляются с такой большой задачей, как поддержка в постоянной работе многопользовательского приложения. Заказы на разработку интернет-магазинов сегодня всё больше и больше востребованы. А, следовательно, можно сделать вывод, что спрос на многофункциональные веб-приложения постоянно растёт. Также в мире, а в частности в Беларуси, огромными темпами растет количество пользователей Internet и, как следствие, количество «электронных» покупателей и потенциальных «электронных» покупателей.

Цель данного проекта – изучить различные методы, технологии по созданию распределённых приложений, создать и реализовать клиентскую и серверную часть интернет-магазина, продемонстрировать их работоспособность посредством тестирования.

Можно также выделить основные задачи, которые следует решить в ходе дипломного проекта:

1. Определиться с основным инструментарием для создания пользовательского интерфейса
2. Определить структуру приложения
3. Сделать качественную вёрстку сайта с помощью специального сборщика проектов Gulp. Адаптивность страниц.
4. Создать базу данных интернет-магазина
5. Реализовать серверную часть приложения
6. Реализовать в полной мере регистрацию и авторизацию пользователя, фильтр товаров на главной странице и в каталоге, возможность оплачивать товар.

# 1.Предметная область

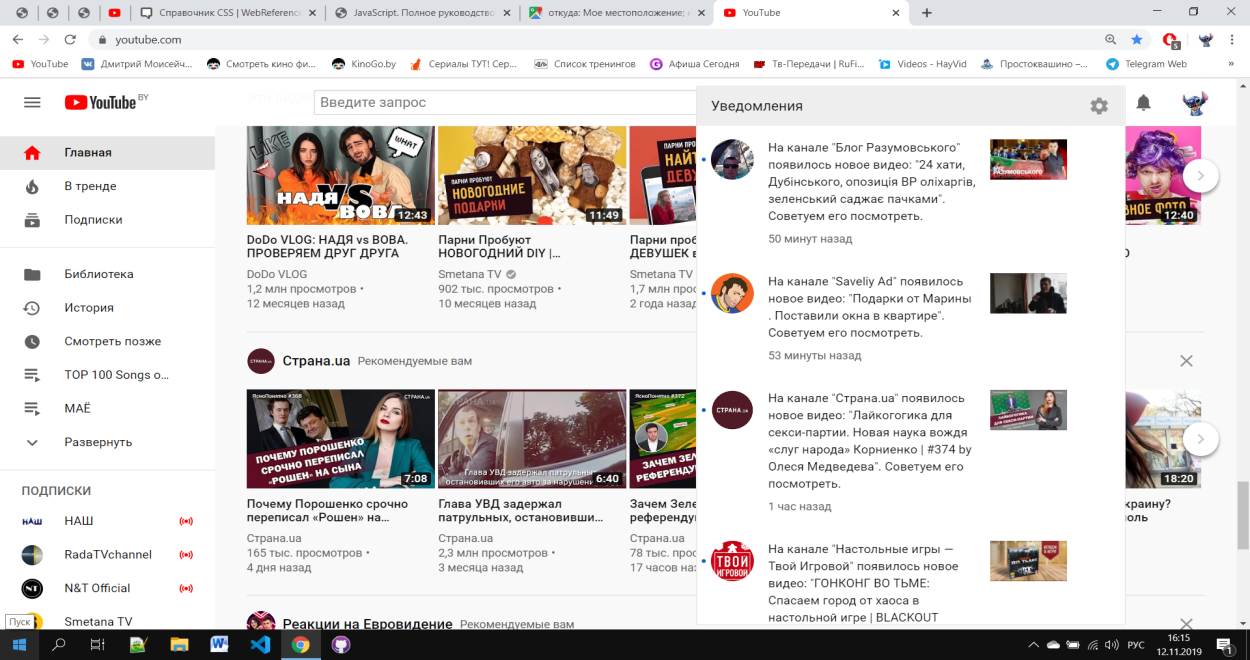
## 1.1 Постановка задачи

Сегодня электронный магазин требует от программистов-разработчиков определённых навыков работы с такими сложными системами и способностей ими правильно управлять. На данный момент, самая главная проблема в разработке интернет – магазина учесть все необходимые нюансы на стадии создания программного обеспечения и, если вдруг понадобится какой-либо новый функционал – суметь быстро реализовать его уже на стадии работы с интернет-магазином в интернете. В данном дипломном проекте будут приложены все усилия, чтобы проблема с разработкой многофункционального интернет-магазина была решена современными методами разработки ПО, так как именно современный инструментарий для разработки такого сложного приложения и делают данную задачу намного легче в реализации.

## 1.2 Выбор технологий

Система построена в виде клиент-серверного web-приложения. Для разработки выбран язык программирования PHP, т.к. он является одним из наиболее распространенных языков для серверных приложений.

Все данные интернет-магазина хранятся в СУБД MySQL. При помощи приложения MySQL Workbench была создана база данных, схема которой представлена на рисунке 1.1.



**Рисунок 1.1 – Схема БД приложения.**

Для взаимодействия с базой данных и PHP был выбран phpMyAdmin, который сам является удобным [веб-приложение](https://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)м с открытым кодом, написанным на языке [PHP](https://ru.wikipedia.org/wiki/PHP) и представляет собой современный веб-интерфейс для администрирования СУБД MySQL.

Для создания пользовательского интерфейса использовалась технология php-страниц с HTML-тегами, а также каскадные таблицы стилей CSS и специальный сборщик frontend-части приложения Gulp, который использовал SCSS стили для приложения и увеличивал скорость разработки интерфейса интернет-магазина. Вся интерактивность была реализована с помощью языка JavaScript.

В данном дипломном проекте упор будет поставлен на использование нисходящей технологии проектирования, так как она наиболее лучше может отразить все этапы создания пользовательского интерфейса. Как большинство вещей в нашей жизни, время, которое потрачено заранее на планирование и построение веб-службы, может помочь в дальнейшем; понимание некоторых компромиссов, стоящих позади больших веб-сайтов, может принести плоды в виде более умных решений при создании меньших веб-сайтов. Ниже рассмотрены самые ключевые принципы, влияющие на проектирование крупных веб-систем:

* Доступность: длительность работоспособного состояния веб-сайта критически важна по отношению к функциональности и репутации многих компаний. Для некоторых наиболее крупных розничных онлайн-магазинов, недоступность, даже в течение нескольких минут, может привести к тысячам или миллионам долларов потерянного дохода. Таким образом, разработка постоянно доступных и эластичных к отказу систем, в свою очередь, является и фундаментальным, и главным технологическим требованием. Высокая доступность в распределенных системах требует внимательного рассмотрения избыточности для ключевых компонентов, быстрого восстановления после частичных системных отказов и сглаженного сокращения возможностей при возникновении небольших проблем в работе.
* Производительность: Производительность любого веб-сайта стала очень важным показателем для большинства современных сайтов. Скорость веб-сайта влияет на работу и удовлетворенность пользователей, а также ранжирование поисковыми системами (Google, Яндекс) — фактор, который непосредственно влияет на удержание аудитории и доход сайта. В результате, ключом является создание системы, которая оптимизирована для быстрых ответов и низких задержек.
* Надежность: система должна быть действительно надежной, так, чтобы определенный запрос на получение данных единообразно возвращал определенные, нужные в данную секунду, данные. В случае изменения данных или обновления, то тот же запрос должен возвращать уже совершенно новые данные. Пользователи должны знать, если что-то записано в систему или храниться в ней, то можно быть уверенным, что оно и будет оставаться на своем месте для возможности извлечения данных впоследствии следующего использования сайта в своих целях.
* Масштабируемость: Когда дело доходит до любой крупной распределенной системы, размер оказывается всего лишь одним пунктом из целого списка, который необходимо учитывать. Не менее важным являются усилия, направленные на увеличение пропускной способности для обработки больших объемов нагрузки, которая обычно и называется «масштабируемостью системы». Масштабируемость может относиться к различным параметрам системы: количество выделяемого дополнительного трафика, с которым она без проблем сможет справиться, насколько легко можно нарастить ёмкость запоминающего устройства, или насколько много транзакций может быть обработано в определённый момент времени.
* Управляемость: проектирование системы, которая проста в эксплуатации также еще один важный фактор. Управляемость системы можно приравнять к масштабируемости операций «обслуживание» и «обновления». Для обеспечения управляемости необходимо рассмотреть вопросы простоты диагностики и понимания возникающих проблем, легкости проведения обновлений или модификаций, прихотливости системы в эксплуатации. (То есть, работает ли она как положено без отказов или исключений?)
* Стоимость: Стоимость является важным фактором. Она, очевидно, может включать в себя расходы на аппаратное и программное обеспечение, однако важно также рассмотреть и другие аспекты, необходимые для развертывания и поддержания системы в работе. Можно выделить основные: количество времени разработчиков, требуемое для построения системы, объем оперативных усилий, необходимые для запуска системы и даже достаточный уровень обучения — все должно быть предусмотрено на этапе запуска. Стоимость же будет представлять собой общую стоимость владения.

# 1.Создание интерфейса интернет-магазина

## 1.1 Характеристика современного пользовательского интерфейса

User Interface или пользовательский интерфейс (UI) – все, что видит посетитель, заходя на веб-сайт. Кнопки, картинки, текст, меню, разделы, блоки отражают ассортимент компании, указывают на возможности для покупателя и предоставляют способы их реализации [3].

Это наиболее важное средство взаимодействия с потенциальным клиентом, основной рекламный инструмент, который способен, как привлекать, так и отталкивать посетителей. В большинстве случаев именно на стадии знакомства с интерфейсом принимается решение о покупке товара или закрытии вкладки с ресурсом.

Работая над созданием или модернизацией функционирующего проекта в e-commerce, необходимо понимать, что каждый элемент структуры служит органичной частичкой крупного паззла. Несущественных деталей на сайте нет. Складываясь в единую общую картинку, они формируют положительное или отрицательное мнение о бренде.

UI онлайн-магазина выполняет три основных задачи:

Ознакомительная. Знакомит [целевую аудиторию](https://www.insales.com.ua/blogs/blog/tselevaya-auditoriya-internet-magazina) с компанией и конкретным товаром.

Маркетинговая. При условии соответствия требованиям клиента дает компании эффективную бесплатную рекламу.

Мотивирующая. Располагает пользователя к покупкам, ускоряет процесс конвертации случайного посетителя в постоянного покупателя.

При проектировании интерфейса должны быть учтены вероятные ожидания аудитории. Открывая страничку любого интернет-магазина, пользователь, в первую очередь, рассчитывает на:

* доступность, простоту подачи, быструю загрузку информации;
* логичное расположение элементов;
* эффективное решение актуальных проблем;
* отсутствие сложностей в процессе совершения каких-либо действий.

Правильный UI, удовлетворяющий потребности покупателя, составляет оптимальный баланс между навигацией и информационной архитектурой сайта. Благодаря этому клиент:

* легко может взаимодействовать с ресурсом;
* понимает процесс выполнения нужных операций на своём интуитивном уровне без дополнительных усилий и подсказок;
* оперативно получает исчерпывающую информацию о товаре;
* совершает целевое действие, проходя минимальный путь.

С помощью только одного грамотно разработанного интерфейса интернет-магазин завоевывает доверие аудитории, получает [прирост трафика](https://www.insales.com.ua/blogs/blog/chto-takoe-trafik) и постоянных клиентов, приумножает объемы продаж и наращивает прибыль [4].

**1.2 Основные инструменты для создания вёрстки сайта**

В своей курсовой работе был задействован особенный набор инструментов для создания нужного интерфейса интернет-магазина, который бы с лёгкостью удовлетворял всем потребностям, описанным в пункте 2. Выбор каждого инструмента был особенным, так как в своей курсовой работе использовались только самые удобные и оптимизированные инструменты для разработки внешнего вида интернет-магазина, его вёрстки.

В самом начале работы над интерфейсом был создан макет интернет-магазина в редакторе Adobe Photoshop. Были нарисованы сразу все возможные разновидности страниц сайта: главная, страница товаров, страница информации, страница контактов, страница авторизации и регистрации, страница корзины. Основной упор ставился на то, чтобы внешний вид сайта был привлекателен для будущих покупателей.

Все страницы имеют относительно одинаковый дизайн, что является наиболее разумным в создании интерфейса.

Для вёрстки было принято решение использовать такие языки разметки, как HTML5, и формальный язык описания внешнего вида документа, написанный с использованием языка разметки CSS.

Следует отметить, что на сегодняшний момент, HTML5 – современная версия языка гипертекстовой разметки, которая получила ряд дополнений и новых возможностей. В состав рабочей группы ее разработчиков входят представители таких гигантов IT-сферы, как Google, Apple, IBM, Microsoft и другие.

Многие эксперты считают HTML5 не продолжением развития языка HTML, а новой открытой программной платформой, которая применяется для создания веб-интерфейсов, использующих текстовые и мультимедийные инструменты (видеоролики, графика, аудиозаписи и другие элементы). При этом в пятой версии сохраняется обратная совместимость и удобочитаемость кода для пользователей.

Также хорошим дополнением для HTML стали CSS (Cascading Style Sheets) – каскадные таблицы стилей, которые содержат описание особенностей внешнего вида веб-документа, созданного на основе инструментов гипертекстовой разметки. В основном CSS используется для визуального оформления веб-страниц. То есть, HTML задает структуру ресурса, а таблицы стилей определяют то, как он будет выглядеть [5].

Именно CSS отвечают за привлекательность и дизайн страниц. Эта технология позволяет сделать сайт привлекательным, за счёт прописи основным элементам стилей.

CSS может применяться не только в языках разметки HTML и XHTML, но и работать в XML-формате (включая документы SVG и HUL).

CSS3 – третья версия таблицы каскадных стилей, которая значительно расширила возможности предыдущих поколений. Особенность CSS3 состоит в том, что с ее помощью можно создавать анимированные элементы без помощи JS. Она включает поддержку различных градиентов и теней, использует новые формы сглаживания и т. д. Использование набора технологий HTML5 + CSS3 открывает разработчикам огромный арсенал программных средств, с которыми значительно легче стало создавать удобные, стильные, современные и функциональные веб-ресурсы [6].

Если для разработки использовались эти средства, то они ускорят загрузку страниц и привнесут ряд привлекательных новаций. CSS3 позволит создать уникальный дизайн сайта, благодаря использованию большого количества визуальных эффектов. А главное вы сможете обойтись без использования языка [Javascript](https://astwellsoft.com/ru/blog/tehnology/javascript.html).

Поэтому выбор данных языков разметки можно считать действительно приемлемым и оптимальным, а главное, определённо самым подходящим для реализации вёрстки страниц интернет-магазина и выполнения поставленных мною задач.

Однако процесс написания HTML и CSS может оказаться несколько изнурительным и требовать множества одних и тех же задач снова и снова. Таких задач, как закрытие тегов в HTML или монотонный просмотр шестнадцатеричных значений цвета в CSS.

Такие разные задачи, как правило, небольшие, чуть-чуть снижают эффективность. К счастью, эти и несколько других неэффективных задач были признаны и вызов им бросили препроцессоры.

Препроцессор — это программа, которая берёт один тип данных и преобразует его в другой тип данных. В случае HTML и CSS, одни из наиболее популярных языков препроцессора — это Sass, SCCS.

Для своей работы был выбран препроцессор SCSS, так как он является одним из наиболее широко используемых CSS-препроцессоров сегодня,  он полностью совместим с CSS. Кроме того, SCSS является хорошо читаемым, так как конкретные конструкции уже имеют смысл. Если в коде встречается @mixin, то можно сразу установить, что это объявление примеси; если в коде встречается  @include - это вызов примеси. У него нет никаких привязок, и всё имеет смысл без интерпретацииp [7].

Пример синтаксиса препроцессора SCSS представлен в Листинге 2.1.1

Листинг 2.1.1

@mixin button-base() {

@include typography(button);

@include ripple-surface;

@include ripple-radius-bounded;

display: inline-flex;

position: relative;

height: $button-height;

&:hover { cursor: pointer; }

&:disabled {

color: $mdc-button-disabled-ink-color;

cursor: default;

pointer-events: none;

}}

Также для наиболее оптимальной и продуктивной работы над вёрсткой сайта было необходимо прибегнуть к помощи сборщика веб-проектов Gulp.

Gulp – это инструмент автоматизации front-end разработки. Он помогает автоматизировать рутинные задачи программиста и как следствие ускоряет процесс разработки. Gulp построен на Node.js, а файл сборки пишется на JavaScript.

Основные задачи, которые вам поможет решить Gulp:

* Использование CSS препроцессоров Sass, Less, Stylus и других;
* Минимизация и конкатенация CSS/JS файлов;
* Минимизация HTML;
* Оптимизация изображений;
* Автоматическая простановка вендорных префиксов;
* Перезагрузка страницы при изменении файлов;
* Удаление неиспользуемого CSS.

Если разработчик сегодня действительно настроен на то, чтобы ценить своё время и время клиентов, следовательно, он вынужден использовать все методы для повышения продуктивности и скорости его работы. Особенно это касается верстки сайтов, так как этот процесс занимает немало времени. По многочисленным наблюдениям специалистов было доказано, что Gulp ускоряет процесс верстки на 20-30%, в зависимости от опыта программиста, его имений использования препроцессоров и скорости печати. С этим инструментом разработка сайта может стать очень быстрым, простым и интересным занятием. Очень важно, что Gulp еще и следит за ошибками в коде, в случае сбоя он выводит сообщение с описанием проблемы и показывает название файла и номер строки, где собственно и произошла ошибка. Потому теперь не нужно тратить очень много времени на поиски очередной ошибки или опечатки в коде, так как c этой задачей легко справляется Gulp, показывая, что нужно исправить и в каком месте.

Поэтому в современной разработке просто глупо не использовать такой сборщик веб-проектов, как Gulp. Листинг 2.1.2 содержит код gulp-файла, который был задействован для того, чтобы ускорить работу над вёрсткой сайта, автоматически перезагружать страницы при изменении сайта [8].

Листинг 2.1.2

let gulp = require('gulp'),

sass = require('gulp-sass'),

rename = require('gulp-rename'),

browserSync = require('browser-sync'),

autoprefixer = require('gulp-autoprefixer'),

concat = require('gulp-concat'),

uglify = require('gulp-uglify'),

cssmin = require('gulp-cssmin');

gulp.task(

'sass',

function() {

return gulp.src('app/scss/\*\*/\*.scss')

.pipe(sass({ outputStyle: 'expanded' }))

.pipe(rename({ suffix: '.min' }))

.pipe(autoprefixer({

overrideBrowserslist: ['last 8 versions']}))

.pipe(gulp.dest('app/css'))

.pipe(browserSync.reload({ stream: true }))

}

);

gulp.task(

'script',

function() {

return gulp.src(

[

'node\_modules/slick-carousel/slick/slick.js',

'node\_modules/magnific-popup/dist/jquery.magnific-popup.js',

'node\_modules/mixitup/dist/mixitup.js',

'node\_modules/rateyo/src/jquery.rateyo.js',

'node\_modules/ion-rangeslider/js/ion.rangeSlider.js', 'node\_modules/@fancyapps/fancybox/dist/jquery.fancybox.min.js',

'node\_modules/jquery-form-styler/dist/jquery.formstyler.js'

]

)

.pipe(concat('libs.min.js'))

.pipe(uglify())

.pipe(gulp.dest('app/js'))

}

);

gulp.task(

'style',

function() {

return gulp.src([

'node\_modules/normalize.css/normalize.css',

'node\_modules/slick-carousel/slick/slick.css',

'node\_modules/magnific-popup/dist/magnific-popup.css',

'node\_modules/rateyo/src/jquery.rateyo.css',

'node\_modules/ion-rangeslider/css/ion.rangeSlider.css',

'node\_modules/@fancyapps/fancybox/dist/jquery.fancybox.css',

'node\_modules/jquery-form-styler/dist/jquery.formstyler.css',

'node\_modules/jquery-form-styler/dist/jquery.formstyler.theme.css'

])

.pipe(concat('libs.min.css'))

.pipe(cssmin())

.pipe(gulp.dest('app/css'))

}

);

gulp.task(

'html',

function() {

return gulp.src('app/\*.html')

.pipe(browserSync.reload({ stream: true }))

});

gulp.task(

'php',

function() {

return gulp.src('app/\*.php')

.pipe(browserSync.reload({ stream: true }))

});

gulp.task(

'js',

function() {

return gulp.src('app/js/\*.js')

.pipe(browserSync.reload({ stream: true }))

});

gulp.task('browser-sync', function() {

browserSync.init({

server: {

baseDir: "app/"

},

host: ' 192.168.100.2'

});

});

gulp.task('watch', function() {

gulp.watch('app/scss/\*\*/\*.scss', gulp.parallel('sass'));

gulp.watch('app/\*.html', gulp.parallel('html'));

gulp.watch('app/\*.php', gulp.parallel('php'));

gulp.watch('app/js/\*.js', gulp.parallel('js'));

});

gulp.task('default', gulp.parallel('style', 'script', 'sass', 'watch', 'browser-sync'))

Однако самым главным преимуществом именно Gulp-сборщика является организация файловой структуры проекта. Так как все документы с расширениями .html, .js, .css строго расформированы уже в самом gulp-файле.

# 2. Программное проектирование серверной части интернет-магазина

Вначале перед созданием самого интерактивного сайта составим список требований-пунктов, который будет в дальнейшем являться главным ориентиром в процессе написания самого приложения.

Этапы реализации серверной части интернет-магазина [4]:

1. Создать таблицы базы данных для хранения информации о продуктах, пользователях и содержании корзины.

2. С помощью языка программирования создать структура сайта, организовать его работоспособность, основываюсь на уже готовом и реализованном интерфейсе.

Требования к интернет-магазину:

1. Сайт должен иметь шапку с разделами по тематике.

2. На главной странице сайта должны публиковаться все посты, хранящиеся в базе данных и сортироваться по тематикам с помощью горячих кнопок.

3. На странице каталога товаров должен быть предусмотрен быстрый отбор, навигация, сортировка всех товаров в интернет-магазине.

4. Если пользователь не зарегистрирован он также должен иметь право добавлять товары в корзину, но после этого отправив специальную форму со своими данными.

5. Если пользователь уже зарегистрирован (находится в базе данных), то специальная форма заказа заполняется автоматически. Пользователю нужно только добавить понравившиеся ему товары в свою корзину.

6. Также следует предусмотреть безопасность входа в магазин (кодирование паролей). Все пароли должны храниться в базе данных только в закодированном виде.

7. Все вносимые пользователем изменения должны синхронизироваться с базой данных и наоборот.

Все вышеперечисленные требования были учтены и реализованы на практике.

# 3. Практическая реализация серверной части интернет-магазина

Для создания серверной части интерактивного интернет-магазина, постоянно ссылающего на базу данных, было принято решение использовать язык программирования PHP. Так как он имеет открытый исходный код и распространяется по GPL лицензии. Он позволяет довольно быстро создавать прототипы и готовые сайты. PHP совместим с большинством операционных систем, таких как Windows, Linux, Solaris, Mac OSX, etc. Он также поддерживает все серверы, лидирующие по количеству инсталляций (Apache, IIS, iPlanet, etc.). Бизнес-приложение или сайт на базе PHP, большой или маленький, может быть с легкостью разработан в условиях ограниченного времени и бюджета. Более продвинутое PHP программирование подразумевает создание CRM решений, интернет-магазинов, веб-сайтов с базами данных, и др.

PHP широко используется для разработки удобных веб-сайтов и привлечения качественной аудитории на ваш сайт, что в итоге ведет к повышению рентабельности инвестиций (ROI). Таким образом данный язык полностью удовлетворяет всем требованиям разработки интернет-магазина. Также продуманная структура кода HTML, PHP, JavaScript и CSS даст нам возможность создавать множество самых разнообразных элементов интерфейса и сетку сайта [5]-[6].

Для создания базы данных, а также для дальнейшей работы с ней, будет использоваться MySQL — свободная реляционная система управления базами данных и PHPMyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL.

Было принято решение выбрать именно систему MySQL, так как одними из главных плюсов MySQL, благодаря которым программа и была выбрана, является высокая скорость обработки информации и функциональность СУБД. Как было уже упомянуто выше, данная система представляет собой программное обеспечение с открытым кодом, и она доступна любому пользователю абсолютно бесплатно, что тоже склоняет чашу весов в сторону MySQL. Большинство современных хостингов используют именно MySQL.

Веб-мастер получает возможность совершенствовать код и вносить в него собственные изменения, что делает программу гибкой. Она многофункциональна и с её помощью можно корректировать, создавать и убирать данные из таблиц, запрашивать из них информацию, создавать сами таблицы и т.д. Кроме MySQL, сегодня на просторах Интернета вы можете найти огромное количество других систем управления БД, таких как PostgreSQL, mSQL и т.д. Все они обладают рядом преимуществ и недостатков, однако именно программа MySQL является самой универсальной, удобной и гибкой среди всех остальных. Поэтому для данного курсового проекта MySQL подходит как нельзя хорошо. И со всеми поставленными задачами по созданию базы данных для интернет-магазина она справится без проблем.

3.1 Создание базы данных

База данных db\_shop – главное хранилище данных работы, в соответствии с планом, должна содержать информацию обо всех товарах сайта, его пользователях и информацию о товарах в корзине определённого пользователя. Следовательно, для запланированной правильной работы сайта потребуется 3 таблицы: table\_products (таблица для хранения товаров магазина), reg\_user (таблица для хранения логинов, паролей и другой личной информации) и cart (таблица для хранения текущего состояния в корзине определённого пользователя). С полями данных таблиц можно подробнее ознакомиться на рисунке 3.1.1.

Следующим действием заполним созданные таблицы данными. Таким образом, в таблице table\_products имеются такие поля как: название, цена, тип, описание, картинка, дата, индикатор принадлежности к какому-либо разделу. В таблице корзины пользователя содержится информация о ip пользователя и id товара, который находится в корзине на данный момент. В таблице пользователей помимо логина и пароля хранится имя, фамилия, email, страна пользователя.

На этом этапе можно сделать вывод, что база данных – каркас сайта − была создана успешно. Поэтому можно переходить уже непосредственно к разработке серверной части сайта, постоянно взаимодействующей с базой db\_shop.

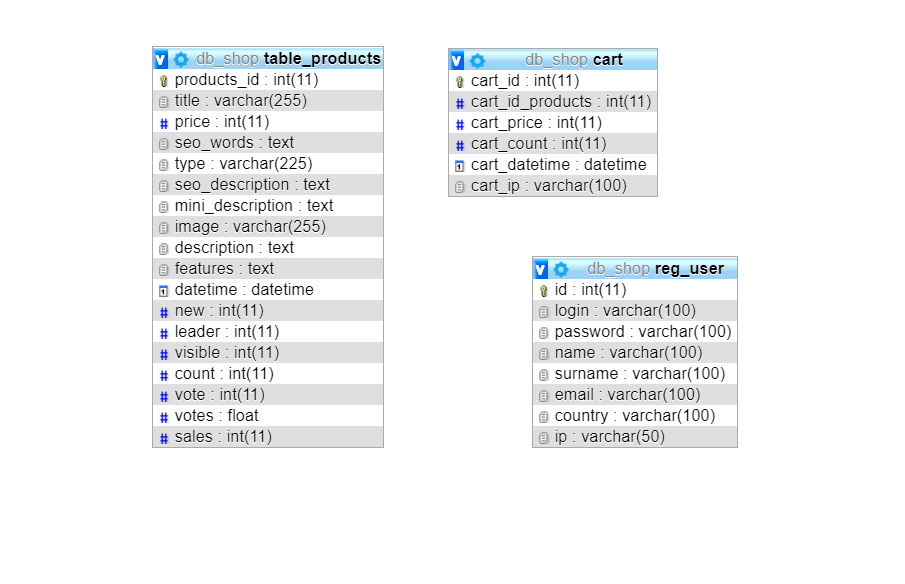


Рисунок 3.1.1 – Структура полей таблиц базы данных

3.2 Создание сессии

Использование сессий становится незаменимым при работе с такими сайтами, как интернет – магазины, ведь в этом случае нужно сохранять данные о пользователе на протяжении нескольких страниц.

Всю сессию можно разделить на три этапа:

* открытие сессии (когда пользователь начинает работу с определенным сайтом),
* учет переменных сессии (при переходе на различные страницы),
* завершение сессии.

Пример создания сессии внутри интернет-магазина представлен в листинге 3.2.1

Листинг 3.2.1

$result = mysqli\_query($link, "SELECT \* FROM reg\_user

WHERE login='$login' AND password='$pass'");

*if* (mysqli\_num\_rows($result) > 0) {

$row = mysqli\_fetch\_array($result);

$\_SESSION['auth\_login'] = $row['login'];

$\_SESSION['auth\_pass'] = $row['password'];

$\_SESSION['auth\_name'] = $row['name'];

$\_SESSION['auth\_surname'] = $row['surname'];

$\_SESSION['auth\_email'] = $row['email'];

echo 'yes\_auth';

$\_SESSION['auth'] = 'yes\_auth';

}

Сессионные переменные хранятся в глобальном массиве $\_SESSION. Для начала работы необходим вызов PHP-функции session\_start(). При этом создается временный файл, где хранятся сессионные переменные. При начале новой сессии создается новый файл, что позволяет отслеживать действия каждого посетителя. Сессия по умолчанию сессия через определенное время (обычно 24 минуты) после закрытия браузера. Однако может быть задано другое время (в секундах) жизни сессии, которое указывается в файле php.ini [8].

Для того чтобы создать шапку интернет-магазина создадим php – файл header.php, в котором определим сетку сайта, разметку полей, колонок, определим наличие изображения на главной полосе сайта.

3.3 Подключение к базе данных

Для того чтобы подключится к базе данных был создан специальный php-файл database.php. В данном файле происходит подключение к базе данных с помощью функции mysqli(), которая принимает 4 аргумента: имя сервера базы данных, имя пользователя, пароль и название базы данных [7]. Также следует установить нужную кодировку во избежание ошибок в воспроизведении текста на страницах сайта. Данное подключение представлено в листинге 3.3.1.

Листинг 3.3.1

<?php  
$link = new mysqli('localhost','dima','dima','news');  
$link->set\_charset("utf8");

## 3.4 Регистрация пользователя

Для любого интернет-магазина важно, чтобы пользователи регистрировались на сайте и оставляли о себе много информации. Это позволяет увеличить средний чек и сильно повышает шансы интернет-магазина превратить разового покупателя в постоянного.

Сама регистрация должна быть очень простой, чтобы не стать барьером для пользователя, а интернет-магазин должен стимулировать пользователей зарегистрироваться. Некоторые крупные интернет-магазины запрещают совершать покупки незарегистрированным пользователям – это довольно спорный способ заставить зарегистрироваться, но он показывает важность регистрации на сайте. Мотивация зарегистрироваться может быть самой разной: специальная скидка для зарегистрированных пользователей, доступ к любым акциям только для зарегистрированных, особые условия зарегистрированным и т.д.

Для того чтобы зарегистрироваться пользователь должен заполнить небольшую форму на вкладке с регистрацией. Если при валидации формы не будет выявлено никаких ошибок, то можно заносить данные сведения о пользователе в саму базу данных (таблица reg\_user). Стоит обратить внимание на то, что все пароли должны храниться в закодированном виде внутри базы данных. Также стоит запросить email пользователя, так как если придётся восстанавливать пароль, то он сможет с лёгкостью сделать данную процедуру, указав свой email в специальном поле-новый пароль придёт на указанный адрес. Исходный код регистрации пользователя представлен в листинге 3.4.1.

Листинг 3.4.1

<?php

session\_start();

include('../include/dbconnect.php');

include('../functions/functions.php');

$error = array();

$login = $\_POST["reg\_login"];

$pass = $\_POST['reg\_password'];

$fname = $\_POST['reg\_fname'];

$lname = $\_POST['reg\_lname'];

$email = $\_POST['reg\_email'];

$country = $\_POST['reg\_country'];

*if* (strlen($login) < 5 or strlen($login) > 15) {

    $error[] = 'LOGIN: 5-15 symbols!'.$login.'';

}

*else* {$result = mysqli\_query($link, "SELECT login FROM reg\_user WHERE login='$login'");

*if* (mysqli\_num\_rows($result) > 0) {

        $error[] = 'Login already exists';

    }

}

*if*(count($error)){

    echo implode('<br/>',$error);

}*else*{

    $pass=md5($pass);

    $pass=strrev($pass);

    $ip=$\_SERVER['SERVER\_ADDR'];

    mysqli\_query($link,"INSERT INTO reg\_user(login,password,name,surname,email,country,ip) VALUES('".$login."','".$pass."','".$fname."','".$lname."','".$email."','".$country."','".$ip."')");echo 'true';

}

session\_reset();?>

## 3.5 Создание фильтра товаров на главной странице магазина

Фильтры — элементы управления интерфейсом сайта интернет-магазина, позволяющие посетителям выбирать товары.

Фильтры в интернет-магазинах помогают посетителям выбирать товары, соответствующие их потребностям, и, соответственно, повышают удобство работы с сайтом.

Главная страница сайта (файл index.php) будет представлена таким образом: вверху размещается шапка сайта с логотипом, меню, горячими кнопками регистрации и авторизации, а также иконка корзины. Ниже будет размещать большой блок, созданный для фильтрации товаров магазина по типу, популярности, новизне. Реализацию блока с фильтрацией товаров можно увидеть в листинге 3.5.1.

Листинг 3.5.1

 <div *class*="products\_\_inner-box">

<?php

$result = mysqli\_query($link, "SELECT \* FROM table\_products");

*if* (mysqli\_num\_rows($result) > 0) {

$row = mysqli\_fetch\_array($result);

*do* {

$img\_path = '';

*if* ($row["image"] != "" && file\_exists("./upload\_images/" . $row["image"])) {

          $img\_path = $row["image"];

          } *else* {

          $img\_path = 'no-img.jpg';

                                        }

*if* ($row["new"] == 1) {

         $new = 'new';

         } *else* {

         $new = '';

                                        }

*if* ($row["leader"] == 1) {

         $leader = 'popular';

          } *else* {

         $leader = '';

                                        }

        echo '

<div id=' .$row["products\_id"]. ' class="product\_\_item mix category-' . $new . ' category-' . $leader . ' category-' . strtolower($row["type"]) . '">

<a href="#" class="product\_\_item-img" style="background-image: url(upload\_images/' . $img\_path . ')"></a>

<div class="product\_\_item-content">

<div class="product\_\_item-name">

<a href="#" class="product\_\_item-title">

  ' . $row["title"] . '

</a>

<a href="#" class="product\_\_item-category">

' . $row["type"] . '</a>

</div><div class="product\_\_item-price">

$' . $row["price"] . '

                     </div>

                     </div>

<div class="product\_\_item-info">

<a tid=' .$row["products\_id"]. ' href="#" class="product\_\_item-btn">

<span>Buy</span>

</a>

<div class="product\_\_item-star">

<div class="ret-star"></div>(<span>' . $row["votes"] . '</span>)

</div>

</div>

</div>

                                        ';

} *while* ($row = mysqli\_fetch\_array($result));

                                }

                                ?></div></div>

## 3.6 Авторизация пользователя

Авторизация является функцией определения прав доступа к ресурсам и управления этим доступом. Авторизация — это не то же самое что идентификация и аутентификация: идентификация — это называние лицом себя системе; аутентификация — это установление соответствия лица названному им идентификатору; а авторизация — предоставление этому лицу возможностей в соответствие с положенными ему правами или проверка наличия прав при попытке выполнить какое-либо действие. Например, авторизацией являются лицензии на осуществление определённой деятельности. Авторизация в интернет-магазине реализуется таким образом: пользователь вводит логин и пароль, эти данные сравниваются с паролями и логинами в базе данных, причем пароли сравниваются только в зашифрованном виде. Если данные совпадают с теми, что есть в базе, то создаётся специальная сессия именно для обладателя данного логина и данного пароля. Реализацию авторизации можно увидеть в листинге 3.6.1

Листинг 3.6.1

<?php

*if* ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    include('../include/dbconnect.php');

    session\_start();

    $login = $\_POST['login'];

    $pass = md5($\_POST['password']);

    $pass = strrev($pass);

$result = mysqli\_query($link, "SELECT \* FROM reg\_user

WHERE login='$login' AND password='$pass'");

*if* (mysqli\_num\_rows($result) > 0) {

        $row = mysqli\_fetch\_array($result);

        $\_SESSION['auth\_login'] = $row['login'];

        $\_SESSION['auth\_pass'] = $row['password'];

        $\_SESSION['auth\_name'] = $row['name'];

        $\_SESSION['auth\_surname'] = $row['surname'];

        $\_SESSION['auth\_email'] = $row['email'];

        echo 'yes\_auth';

        $\_SESSION['auth'] = 'yes\_auth';

    }

}

# 4. Тестирование работы серверной части интернет-магазина

После успешной реализации всех запланированных этапов и после того, как все условия курсового проекта были соблюдены. Можно сделать вывод, что цель, поставленная в начале курсового проекта, была достигнута. Однако теперь следует протестировать работоспособность интернет-магазина [9]-[10].

Результаты тестирования представлены на рисунках 4.1- 4.5

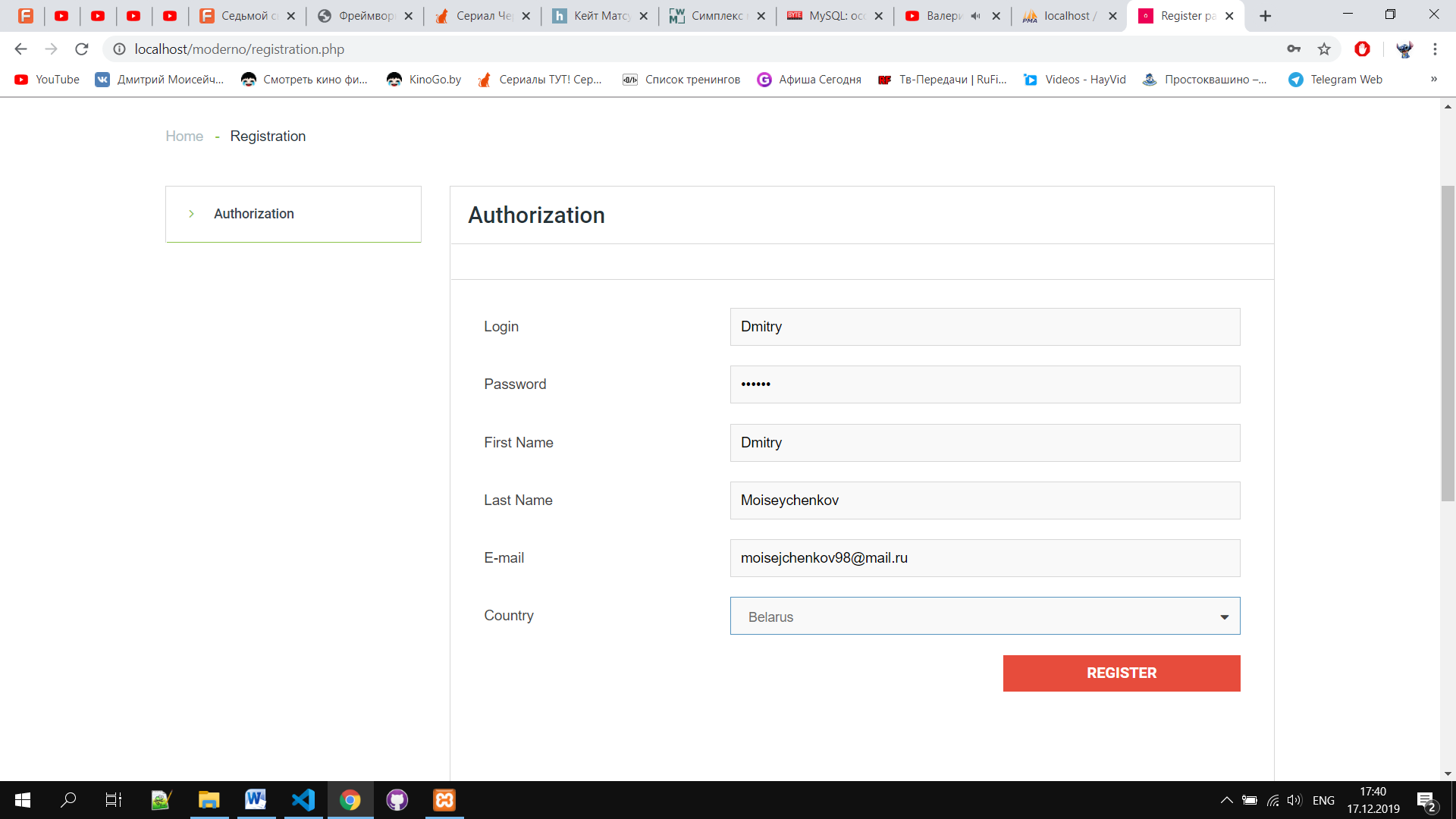


Рисунок 4.1 – Регистрация нового пользователя

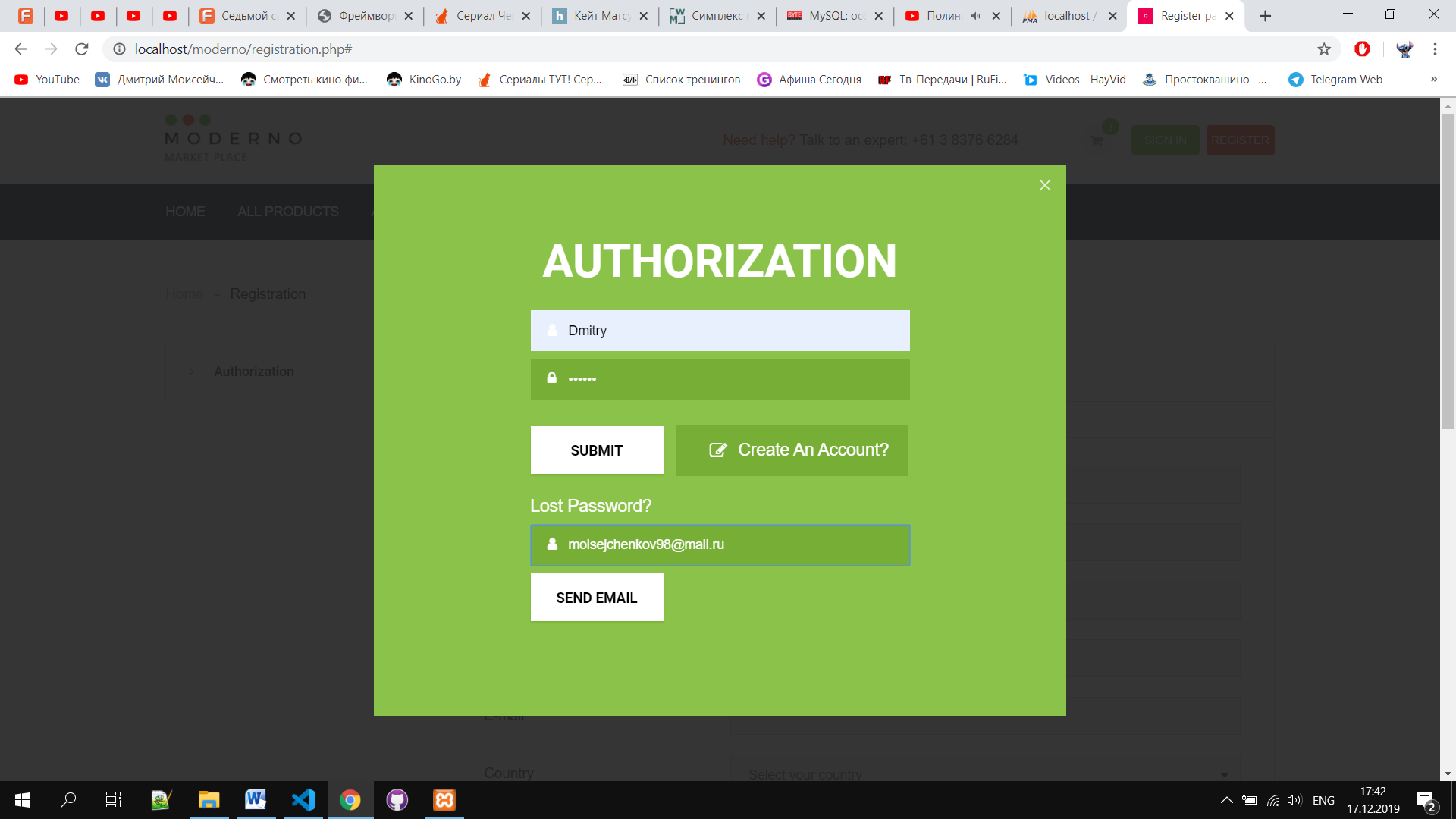
****

Рисунок 4.2 – Авторизация пользователя

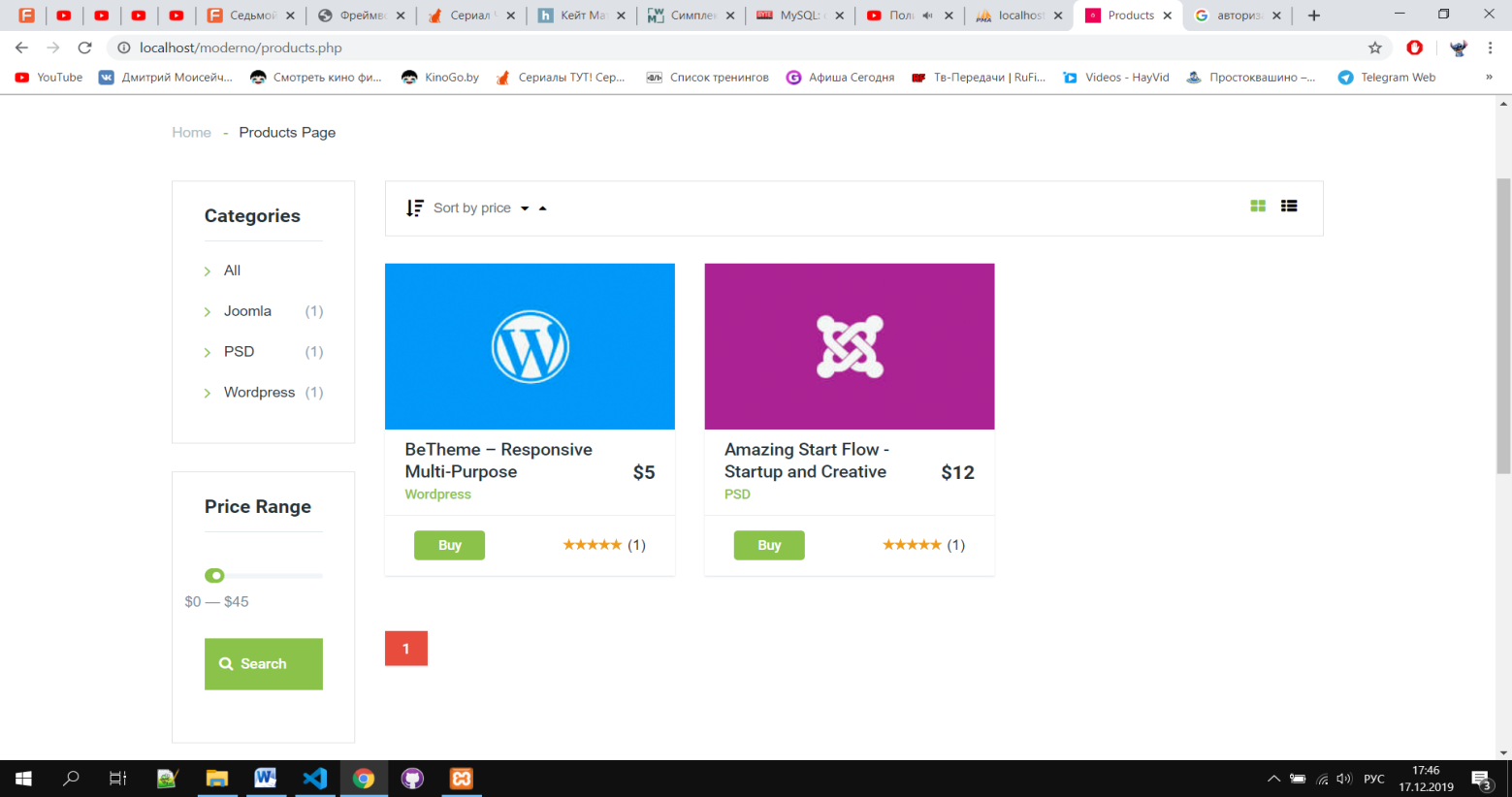
****

Рисунок 4.3 – Каталог товаров, сортировка по цене

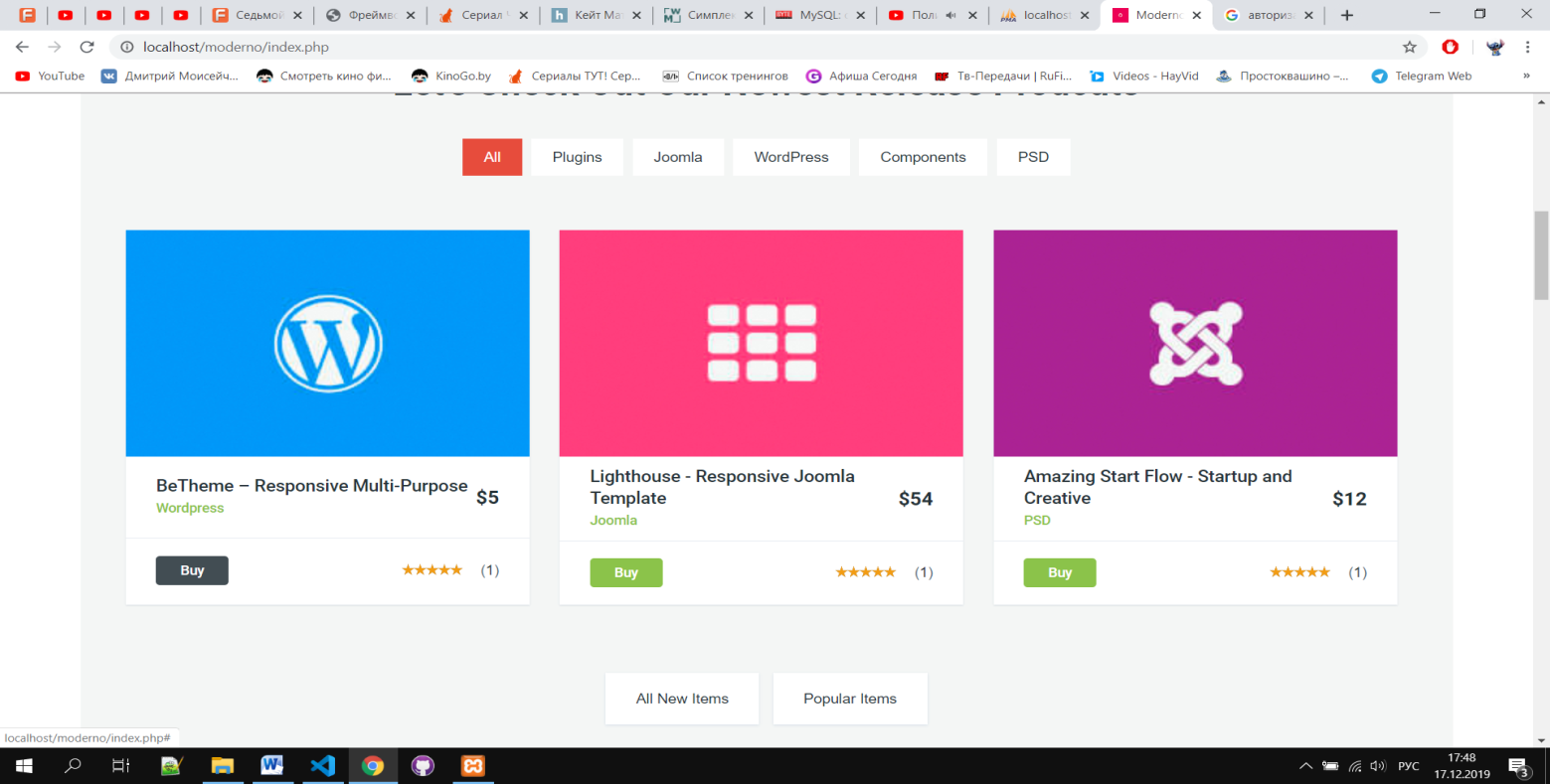
****

Рисунок 4.4 – Добавление товара в корзину

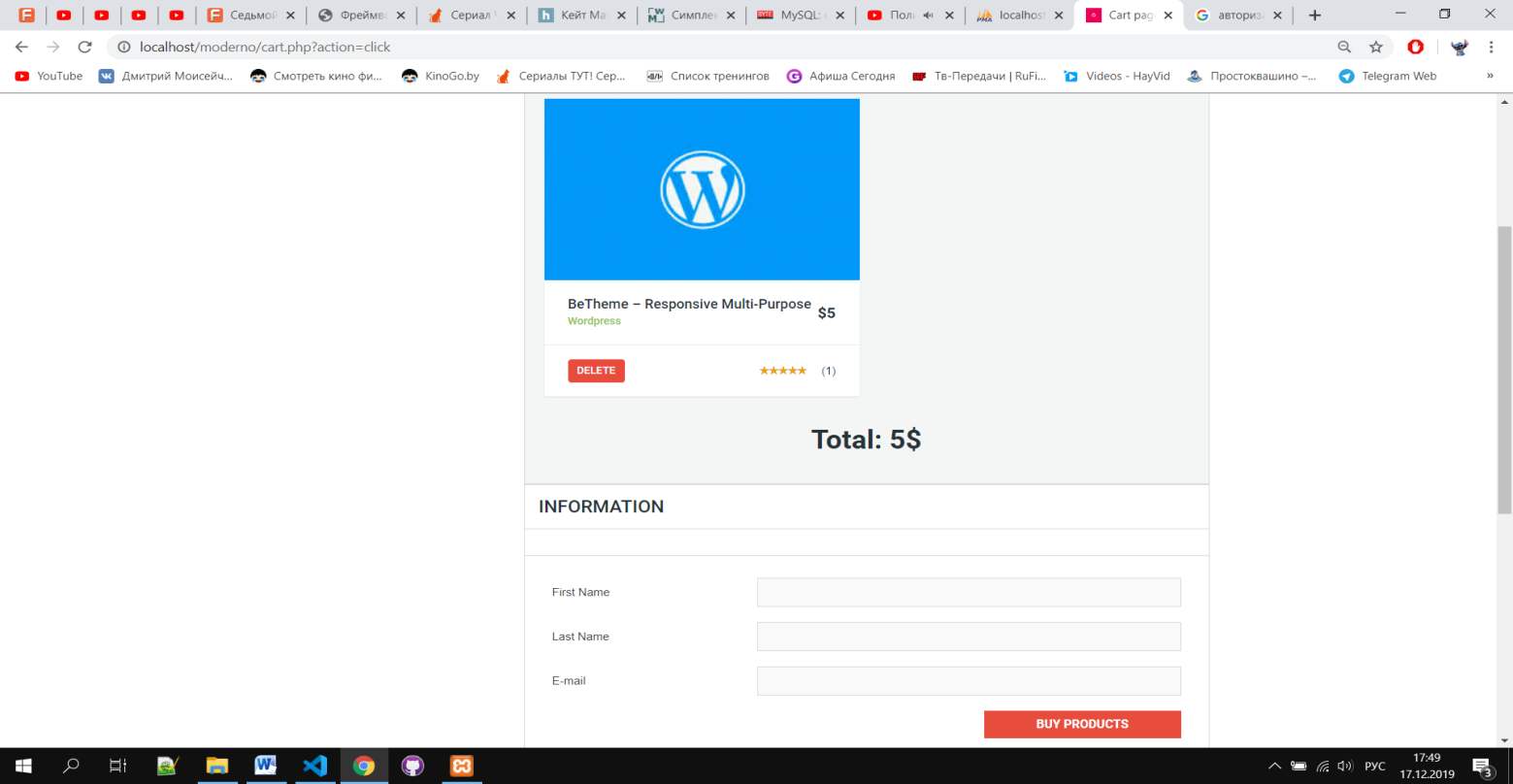
****

Рисунок 4.5 – Товар в корзине

## 4.1 Результаты тестирования

Как видно из тестов, интернет-магазин работает без ошибок. При переходе на другие страницы сайта ошибок обнаружено также не было. Регистрация и авторизация пользователя работает правильно и без ошибок. Все товары фильтруются по цене, алфавиту и по типу без ошибок. Интерфейс сайта прост и удобен.

Все главные элементы сайта выполняют свои функции правильно. На главной странице с помощью горячих кнопок можно без труда найти новинки, тематические товары, самые популярные товары в магазине. Форма регистрации проста и удобна. При добавлении товара в корзину пользователь видит уменьшенные копии данных товаров, а также итоговую сумму, которую следует оплатить. Он может, как удалить старый товар, так и добавить новый. Все действия пользователя в интернет-магазине привязаны к ip, поэтому корзина пользователя будет принадлежать лишь ему одному.

Выход из интернет-магазина также работает без нареканий. Благодаря созданию сессии все действия зарегистрированного пользователя принадлежат только ему и никому другому. Следовательно, можно сделать вывод, что интерактивный интернет-магазин был создан успешно. Значит цель, которая была поставлена в начале курсового проекта, была достигнута.

# Заключение

Несмотря на широкий спектр доступных инструментов для реализации серверной части интернет-магазина, в курсовом проекте были рассмотрены наиболее распространенные подходы и технологии, знание которых существенно облегчает работу программиста.

В результате выполнения курсового проекта были получены навыки работы и поиска учебной литературы, улучшены навыки изучения и анализа различных источников, были подробно описаны все этапы работы и использовании представленных алгоритмов.

Следует отметить, что в разработанном интернет-магазине удачно сочетается интерфейс магазина с его функциональностью и простотой использования.

Интерактивный интернет-магазин − главная цель проекта − является удобным и простым в понимании сайтом. Сам магазин работает правильно, без ненужных прерываний и задержек при переходе со страницы на страницу. В ходе выполнения курсового проекта была разработана база данных для интернет-магазина, посредством использования софта для администрирования MySQl, была протестирована работоспособность базы данных, безошибочность подключения к базе данных. Был создан интерактивный интернет-магазин, который постоянно взаимодействует с базой данных, путём отправки данных о товарах, пользователях и состоянии корзины определённого пользователя интернет-магазина. Была продемонстрирована его безошибочная работа. Посредством языка программирования PHP была реализована серверная часть магазина, которая основывается на уже готовом пользовательском интерфейсе.

Интернет-магазин имеет удобный и понятный интерфейс, благодаря которому не требуется излишних усилий в освоении работы с сайтом. Вся информация не содержит ничего лишнего.

Результаты проверки показали, что система работает без ошибок, имеется возможность работы с веб-сайтом во всех популярных браузерах.

В результате разработки курсового проекта были приобретены практические умения PHP, а так же использования системы управления содержимым сайта.

Модульная система php-скрипта открывает широкие возможности и перспективы электронному магазину, позволяя ему изменяться в ногу со временем, совершенствуя и оттачивая искусство взаимодействия с клиентом.

В ходе работы над интернет-магазином была задействована сессионная методика и были решены все задачи, поставленные в начале курсового проекта. Регистрация и авторизация пользователей происходит без заминок, фильтр товаров работает правильно, сессии вовремя создаются и сбрасываются.

# Список использованных источников

1. MySQL − Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/MySQL> − Дата доступа: 17.12.2019

2. MySQL − mysql.com [Электронный ресурс]. – Режим доступа: https://www.mysql.com/ − Дата доступа: 17.12.2019

3. Дронов, В.  PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов/ В.Дронов, −М.: BHV 2016.− 688 с.

4. Дари К., Баланеску Э. PHP и MySQL: создание интернет-магазина/ К.Дари,Э.Баланеску. −М.: Диалектика-Вильямс., 2011.− 640 с.

5. PHP − php.net [Электронный ресурс]. – Режим доступа: https://www.php.net/ − Дата доступа: 17.12.2019

6. PhpMyAdmin − Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/PhpMyAdmin − Дата доступа: 17.12.2019

7. Подключение в PHP к MySQL и выполнение запросов – metanit.com [Электронный ресурс]. – Режим доступа: https://metanit.com/web/php/7.2.php − Дата доступа: 17.12.2019

8. Сессии в PHP − php.su [Электронный ресурс]. – Режим доступа: http://www.php.su/articles/?cat=examples&page=070− Дата доступа: 17.12.2019

9. Янк, К. PHP и MySQL. От новичка к профессионалу/ К.Янк. − М.: Эксмо, 2017.− 384 с.

10. Шиколенков, Т. Ваш интернет-магазин от А до Я/ Т.Шиколенков. −М.:Шиколенко Т.А., 2018.− 978 с.

# ПРИЛОЖЕНИЕ А

Содержание электронного носителя

На электронном носителе расположены следующие директории и файлы:

− Исходный код программы

Файл «Курсовой проект «РАЗРАБОТКА И РЕАЛИЗАЦИЯ СЕРВЕРНОЙ ЧАСТИ САЙТА-МАГАЗИНА Моисейченков Д.Ю.» с документацией к курсовому проекту

− Презентация

«РАЗРАБОТКА И РЕАЛИЗАЦИЯ СЕРВЕРНОЙ ЧАСТИ САЙТА-МАГАЗИНА Моисейченков Д.Ю»