МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

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

Факультет математики и информационных технологий

Кафедра информатики и информационных технологий

КУРСОВАЯ РАБОТА  
 по дисциплине «Дополнительные главы информатики»

СОВРЕМЕННЫЕ WEB-ТЕХНОЛОГИИ ДЛЯ СОЗДАНИЯ РАСПРЕДЕЛЁННЫХ ПРИЛОЖЕНИЙ

Моисейченков Дмитрий Юрьевич

4 курс, 44 группа

Руководитель:

Шедько Василий Викторович

старший преподаватель

Витебск, 2019

**Реферат**

Курсовая работа 30 с., 6 рис., 9 источников.

WEB-ТЕХНОЛОГИИ ПРИЛОЖЕНИЕ ИНТЕРНЕТ-МАГАЗИН

Объект исследования – методы и технологии создания распределённых приложений.

Предмет исследования − реализация в программировании методов и технологий создания распределённых приложений.

Цель курсовой работы – ознакомиться с различными методами, технологиями по созданию распределённых приложений, сделать сравнительную характеристику данных методов и технологий.

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

СОДЕРЖАНИЕ

[Введение 4](#_Toc27636711)

[1.Программное обеспечение по технологии RAD 7](#_Toc27636712)

[2. Интерфейс интернет-магазина 9](#_Toc27636713)

[2.1 Основные инструменты для создания вёрстки сайта 10](#_Toc27636714)

[3.Основные особенности интерфейса интернет-магазина 18](#_Toc27636715)

[3.1 Шапка сайта 18](#_Toc27636716)

[3.3 Корзина интернет-магазина 19](#_Toc27636717)

[3.4 Каталог товаров 19](#_Toc27636718)

[3.5 Товарные страницы 20](#_Toc27636719)

[3.6 Форма обратной связи 20](#_Toc27636720)

[4. Практическая реализация вёрстки страниц интернет магазина. Адаптивность страниц. 21](#_Toc27636721)

[5. Тестирование страничной вёрстки интернет-магазина 23](#_Toc27636722)

[5.1 Результаты тестирования 26](#_Toc27636723)

[Заключение 27](#_Toc27636724)

[Список использованных источников 29](#_Toc27636725)

[ПРИЛОЖЕНИЕ А 30](#_Toc27636726)

# Введение

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. Данная курсовая работа 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.

В данной курсовой работе 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 [1]:

* 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.

Цель данной работы – ознакомиться с различными методами, технологиями по созданию распределённых приложений, сделать сравнительную характеристику данных методов и технологий создания пользовательского интерфейса.

Можно также выделить основные задачи, которые следует решить в ходе курсовой работы:

1. Определиться с основным инструментарием для создание пользовательского интерфейса
2. Определить структуру приложения
3. Использовать современные веб-технологии для разработки чего-либо
4. Сделать качественную вёрстку сайта с помощью специального сборщика проектов Gulp
5. Провести нужные тесты на адаптивность страниц

# 1.Программное обеспечение по технологии RAD

Жизненный цикл ПО по методологии RAD состоит из четырех фаз [2]:

* фаза анализа и планирования требований;
* фаза проектирования;
* фаза построения;
* фаза внедрения.

На фазе анализа и планирования требований пользователи системы определяют основные функции, которые она должна будет выполнять, выделяют наиболее приоритетные из них, описывают информационные потребности. Ограничивается масштаб проекта, определяются временные рамки для каждой из последующих фаз. Кроме того, определяется сама возможность реализации данного проекта в установленных рамках финансирования, на данных аппаратных средствах и т.п. Результатом данной фазы должны быть список и приоритетность функций будущей информационной системы, предварительные функциональные и информационные модели информационных систем.

На фазе проектирования часть пользователей принимает участие в техническом проектировании системы под руководством специалистов-разработчиков. CASE-средства используются для быстрого получения работающих прототипов приложений. Пользователи, непосредственно взаимодействуя с ними, уточняют и дополняют требования к системе, которые не были выявлены на предыдущей фазе. Более подробно рассматриваются процессы системы. Анализируется и, при необходимости, корректируется функциональная модель. Каждый процесс рассматривается детально. При необходимости для каждого элементарного процесса создается частичный прототип: экран, диалог, отчет, устраняющий неясности или неоднозначности. Определяются требования разграничения доступа к данным. На этой же фазе происходит определение набора необходимой документации.

После детального определения состава процессов оценивается количество функциональных элементов разрабатываемой системы и принимается решение о разделении информационных систем на подсистемы, поддающиеся реализации одной командой разработчиков за приемлемое для RAD-проектов время - порядка 60 - 90 дней. С использованием CASE-средств проект распределяется между различными командами (делится функциональная модель). Результатом данной фазы должны быть:

* общая информационная модель системы;
* функциональные модели системы в целом и подсистем, реализуемых отдельными командами разработчиков;
* точно определенные с помощью CASE-средства интерфейсы между автономно разрабатываемыми подсистемами;
* построенные прототипы экранов, отчетов, диалогов.

Все модели и прототипы должны быть получены с применением тех CASE-средств, которые будут использоваться в дальнейшем при построении системы.

Данное требование вызвано тем, что в традиционном подходе при передаче информации о проекте с этапа на этап может произойти фактически неконтролируемое искажение данных. Применение единой среды хранения информации о проекте позволяет избежать этой опасности.

# 2. Интерфейс интернет-магазина

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].

**2.1 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.

# 3.Основные особенности интерфейса интернет-магазина

Сегодня важно помнить, что изо дня в день на потенциальных клиентов обрушивается поток информации и различных рекламных сообщений, и что в плане завоевания их внимания существует предельно жесткая конкуренция. Веб-сайт, способный привлечь внимание и вызвать любопытство, побудит клиентов не только просмотреть оставшиеся страницы и совершить покупки, но и снова посетить его через некоторое время, а также рекомендовать своим друзьям и знакомым.

Поэтому следует внимательно подходить к выбору цвета, разметки, функциональности интернет магазина.

Главные элементы структуры

В идеале интерфейс магазина включает следующие основные элементы структуры:

* шапка;
* поиск;
* корзина;
* каталог товаров;
* товарные страницы;
* форма обратной связи.

Следует остановиться на каждом из них подробнее.

**3.1 Шапка сайта**

Шапка онлайн-магазина – информативный верхний блок, который бросается первым в глаза на этапе беглого ознакомления с сайтом. Он не должен быть нагромождён сложными композициями из огромного количества рисунков.

В самом верху слева нужно обязательно разместить логотип магазина, кратко описывающий вид деятельности компании. На видном месте в правом верхнем углу должна присутствовать корзина. По центру – краткая контактная информация (номер телефона/электронная почта). Сразу под шапкой следует расположить меню.

**3.2 Поиск**

Функция поиска облегчает и ускоряет процедуру выбора нужного товара. Поисковую строку реализуют во всех популярных магазинах. Особенно важна эта опция для компаний с обширным товарным ассортиментом.

Поиск должен быть быстрым, интерактивным, мгновенно подстраивать результаты, подходящие под введенные значения. Желательно, чтобы выдача могла выполняться под разные параметры: артикул, название продукта, бренд.

**3.3 Корзина интернет-магазина**

Важно обеспечить клиенту возможность добавления товарных позиций в корзину без принудительной регистрации и других навязчивых процедур. Хорошо, когда сам процесс сопровождается откликом интерфейса: в верхнем правом углу страницы отображается отдельная и суммарная стоимость отложенных продуктов, изменяясь по мере их добавления.

Содержимое корзины нужно сделать наглядным, предельно понятным: картинка-превью, наименование, краткое описание, стоимость каждой позиции, общая сумма заказа.

**3.4 Каталог товаров**

Для понятной и быстрой навигации по продуктам создается специальный каталог, в котором имеющийся ассортимент распределяется по разделам и категориям. Это может быть обычное меню с названиями разделов, динамическая рамка с кнопками-разделами, отдельные страницы с миниатюрами фотографий.

Ключевое предназначение рубрикатора – помочь пользователю отыскать нужную страницу за минимальное количество кликов. Причем у него должна всегда оставаться возможность вернуться к предыдущему действию с помощью простой навигационной цепочки. Удобным дополнительным инструментом для структурирования товаров служат фильтры. Они выстраивают позиции в соответствии с заданными критериями: цена, производитель, материал изготовления, алфавит.

**3.5 Товарные страницы**

Идеальная [карточка товара](https://www.insales.com.ua/blogs/blog/kartochki-tovarov-internet-magazina) является лаконичной и простой. Ее оформление сосредоточено непосредственно на продукте и его характеристиках. Товарная страница плавно подводит потенциального покупателя к целевому действию, всячески помогает совершить покупку. Нужно постараться объединить в ней все, что хочет знать пользователь о товаре/услугах – подробные параметры, [качественные фото](https://www.insales.com.ua/blogs/blog/foto-dlya-internet-magazina), наличие на складе, реальная стоимость, отзывы.

Информацию о доставке и способах оплаты можно расположить в виде коротких блоков, разворачивающихся при нажатии. Обязательный элемент карточки – конверсионная кнопка «Купить» или «Добавить в корзину». Лучше выделить ее, используя цветовые акценты.

**3.6 Форма обратной связи**

Принципиально важный момент – общение с клиентом. Поскольку покупатели часто не уверены в своем выборе, они ищут на сайте помощи знающих профессионалов. Чтобы иметь возможность оперативно сориентировать пользователя и довести его до финальной ступеньки [воронки продаж](https://www.insales.com.ua/blogs/blog/voronka-prodazh-dlya-internet-magazina), необходимо предусмотреть на сайте формы обратной связи.

Также огромным преимуществом считается наличие рекламы самых популярных или новинок на главной странице сайта. Также неотъемлемой частью интернет-магазина должна служить правильная авторизация и регистрация. Хорошо, когда и незарегистрированный пользователь может купить товар, однако он должен заполнить специальную форму-анкету перед покупкой. Зарегистрированный же и авторизованный пользователь может не переживать по этому поводу, так как все его данные уже есть в базе данных и всё , что ему остаётся-это просто заполнить корзину и оплатить товар без всяких других анкет и форм.

Все выше указанные элементы и их преимущества были учтены и реализованы непосредственно в момент создания интерфейса интернет-магазина. Вся необходимая информация по практической части находится в 4 разделе курсовой работы.

# 4. Практическая реализация вёрстки страниц интернет магазина. Адаптивность страниц.

Все перечисленные выше элементы сайта были реализованы с помощью языков разметки HTML,CSS. Все ошибки были исправлены посредством взаимодействия с веб-сборщиком Gulp. Также вместо обычного языка CSS был задействован препроцессор SCSS. С помощью специального gulp-файла была реализована строгая структура сайта.

Используя спиралевидную технологию проектирования, следует предусмотреть адаптивность под различные устройства для всех страниц интернет-магазина. Адаптивная верстка меняет дизайн страницы в зависимости от поведения пользователя, платформы, размера экрана и ориентации девайса и является неотъемлемой частью современной веб-разработки. Она позволяет существенно экономить и не отрисовывать новый дизайн для каждого разрешения, а менять размеры и расположение отдельных элементов. Данную возможность помогли реализовать специальные медиа-запросы, которые в зависимости от ширины экрана меняют некоторые свойства определённых элементов.

Медиа запросы появились в спецификации CSS3 и на сегодняшний день поддерживаются во всех современных браузерах (Chrome 4+, Firefox 3.5+, IE 9+, Opera 9+, Safari 4+).

Для создания медиа запросов используется синтаксис как в листинге 4.1:

Листинг 4.1

@media условие {

/\* стили (они будут выполняться, если устройство соответствует указанному условию)

}

Для создания дизайна, позволяющего лучшим образом отображать сайт на различных устройствах, используют общие стратегии медиа-запросов:

* Уменьшение количества колонок (столбцов) и постепенная отмена обтекания для мобильных устройств.
* Использование свойства max-width вместо width при задании ширины блока-контейнера.
* Уменьшение полей и отступов на мобильных устройствах (например, нижних отступов между заголовком и текстом, левого отступа для списков и т.п.).
* Уменьшение размеров шрифтов для мобильных устройств.
* Превращение линейных навигационных меню в раскрывающиеся.
* Скрытие второстепенного содержимого на мобильных устройствах с помощью display: none.
* Подключение фоновых изображений уменьшенных размеров.

# 5. Тестирование страничной вёрстки интернет-магазина

После успешной реализации всех запланированных этапов и после того, как все условия курсовой работы были соблюдены. Можно сделать вывод, что цель, поставленная в начале курсовой работы, была достигнута. Однако теперь следует протестировать интерфейс сайта. Оценить всю визуальную составляющую.

Результаты тестирования представлены на рисунках 5.1- 5.6

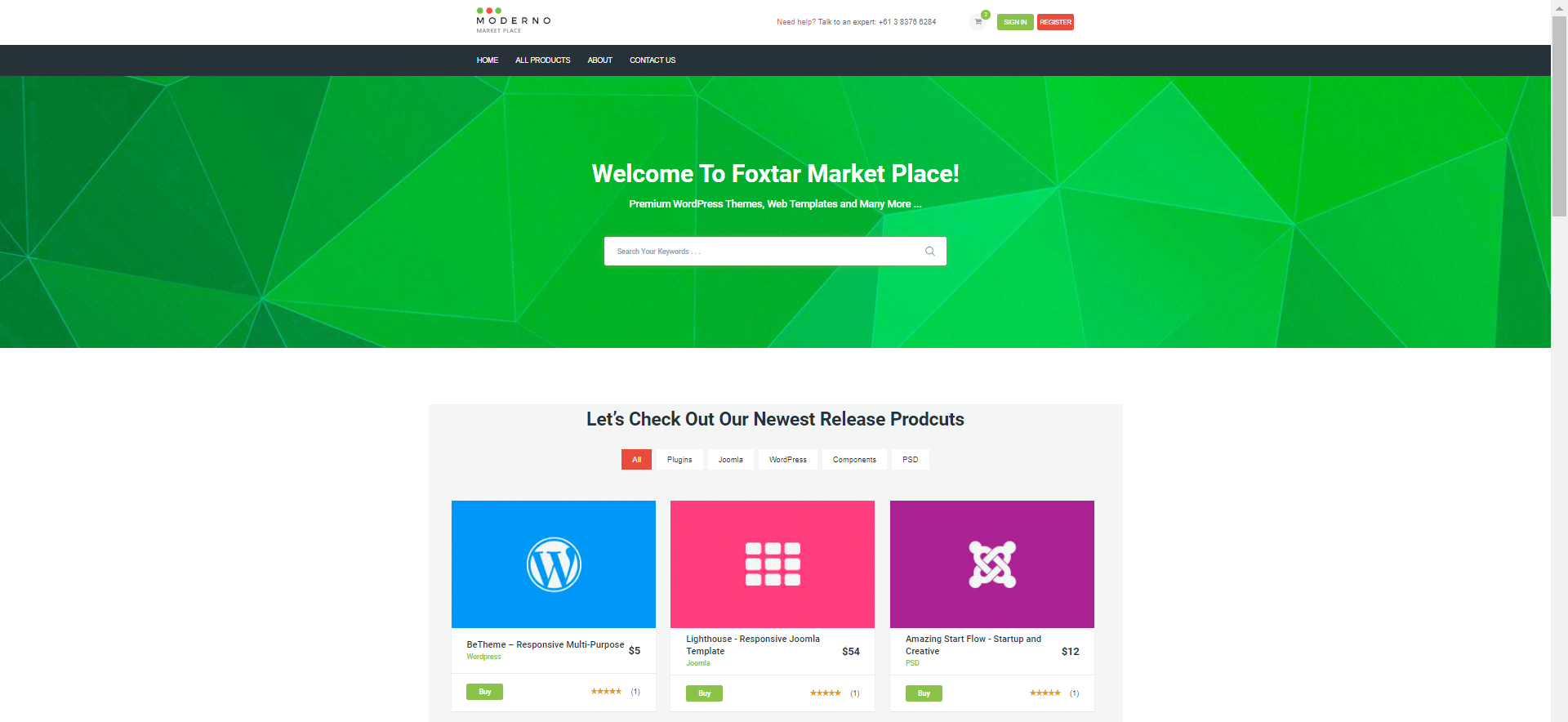


Рисунок 5.1 – Главная страница интернет-магазина (1 часть)

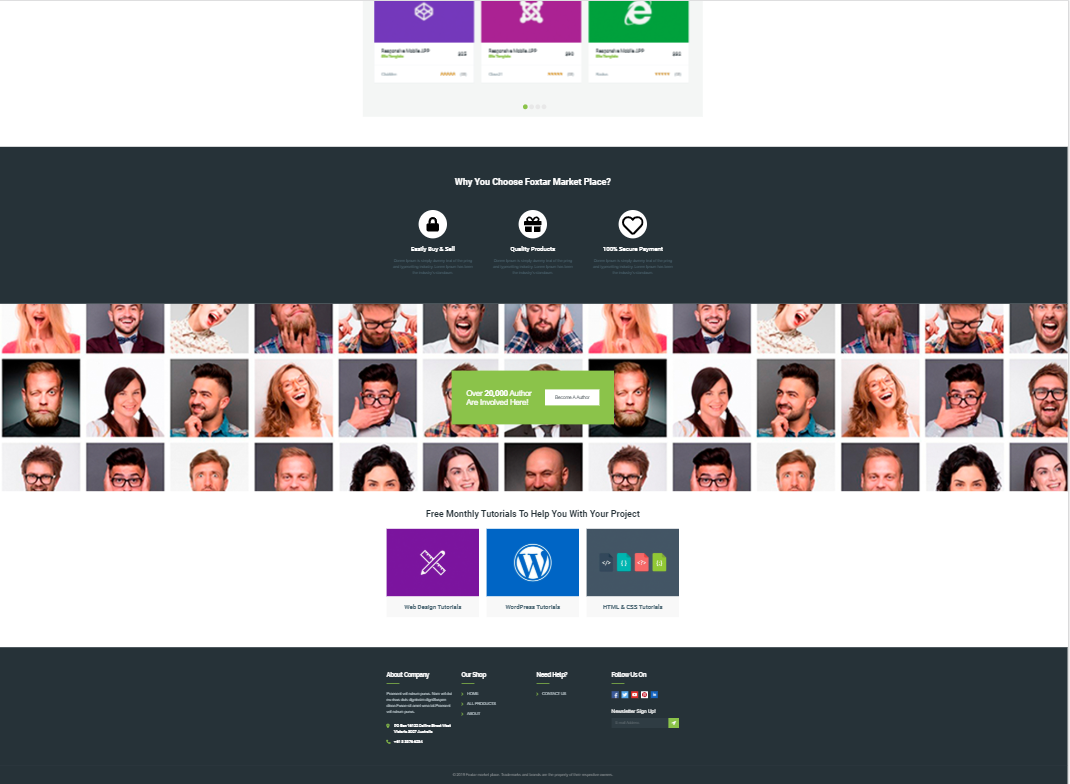


Рисунок 5.2 – Главная страница интернет-магазина (2 часть)

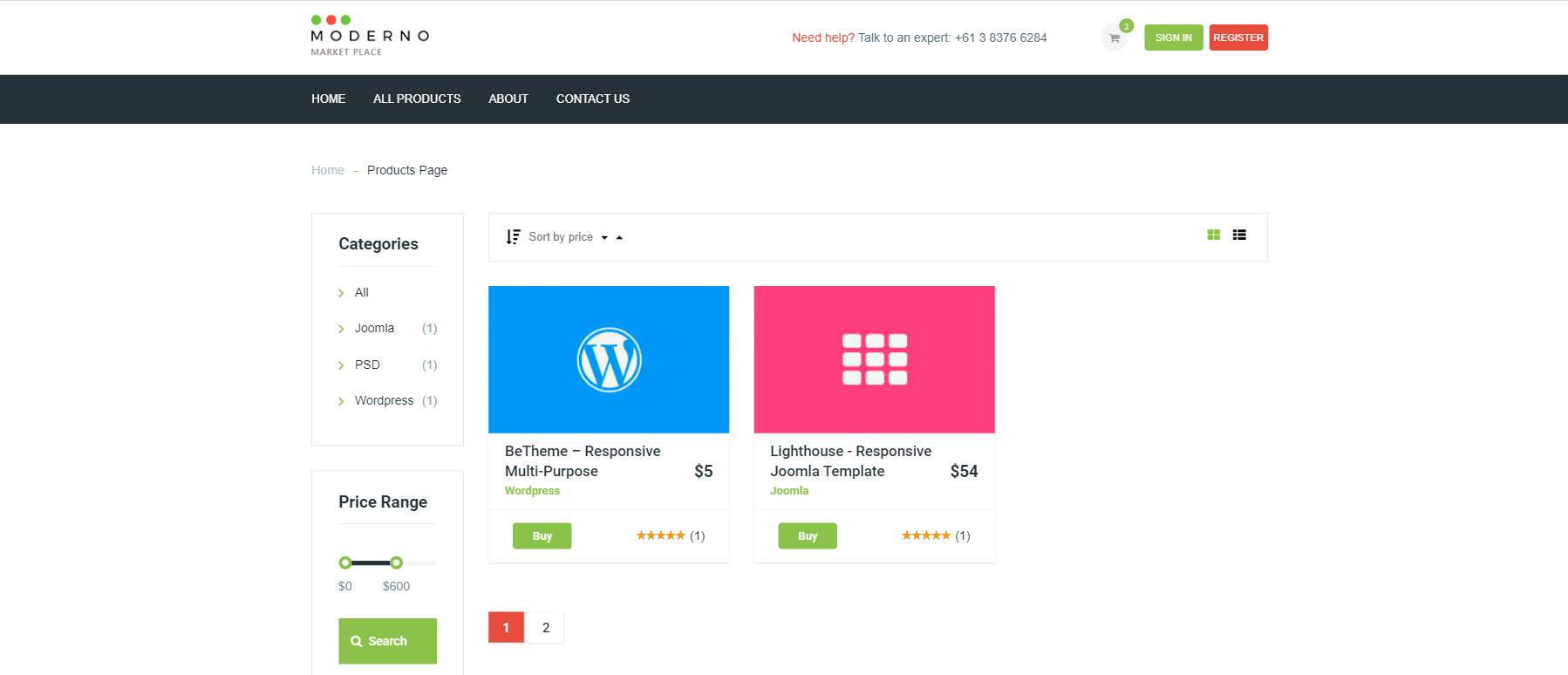


Рисунок 5.3 – Каталог товаров

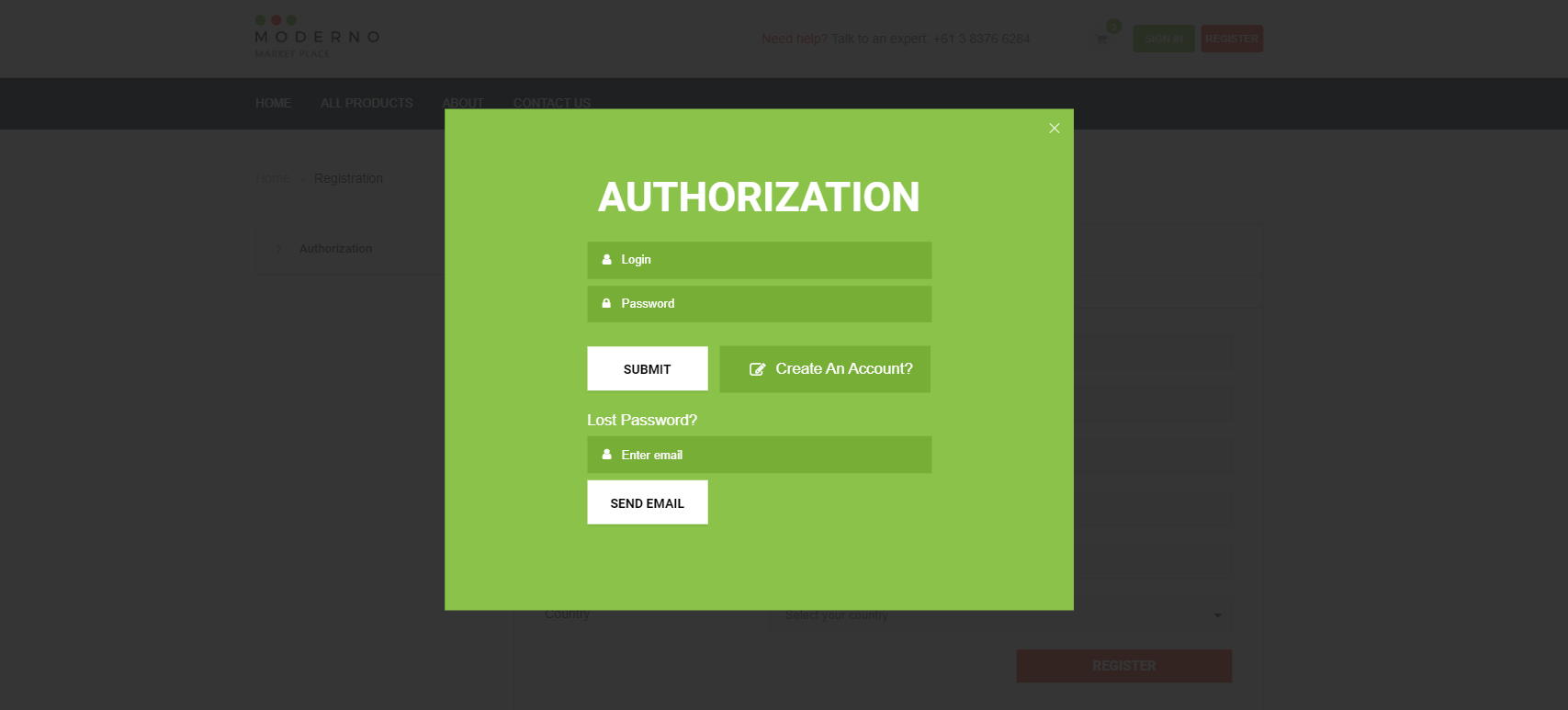


Рисунок 5.4 – Окно авторизации

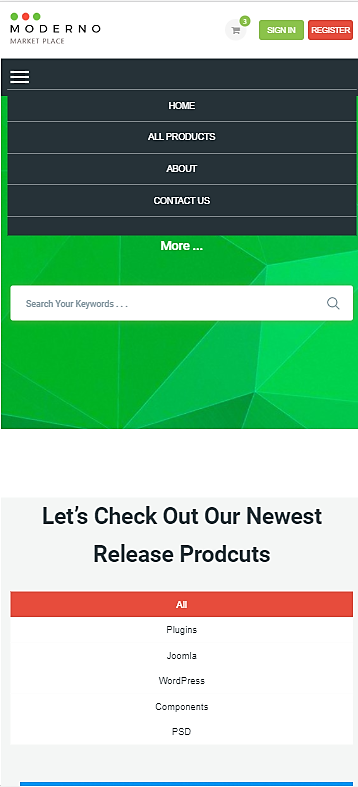


Рисунок 5.5 – Адаптивность главной страницы

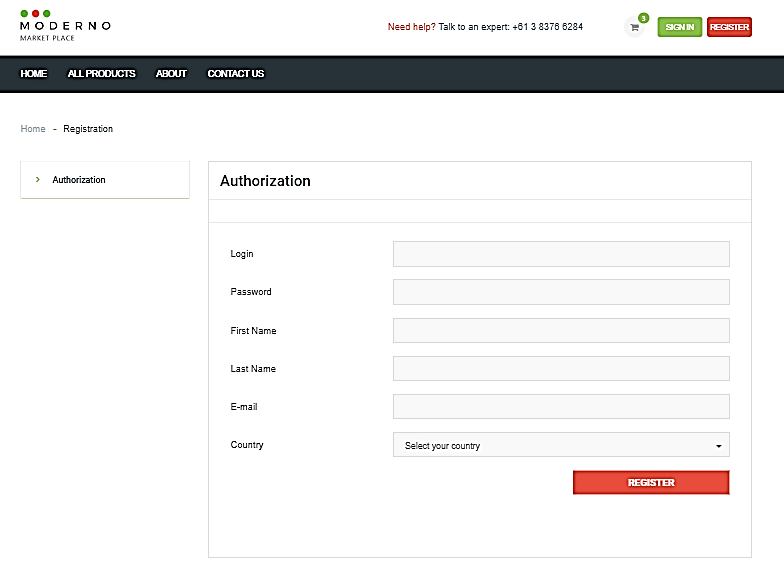


Рисунок 5.6 – Страница регистрации

## 5.1 Результаты тестирования

Как видно из тестов, вёрстка работает без ошибок. При переходе на другие страницы сайта ошибок обнаружено не было. Адаптивность страниц сайта работает при любом расширении экрана. Значит, на телефонах и планшетах любой пользователь сможет без проблем пользоваться данным сайтом. Интерфейс сайта прост и удобен. Все главные элементы сайта выполняют свои функции правильно. Следовательно, можно сделать вывод, что интерфейс интернет-магазина был создан успешно. Значит цель, которая была поставлена в начале курсовой работы, была достигнута.

# Заключение

В результате выполнения курсовой работы 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.

Кроме того, были углублены и закреплены практические навыки работы с HTML, CSS, Gulp, SCSS. При тестировании вёрстки сайта каких-либо сбоев в работе новостного сайта не произошло. Также были расширены знания и практические навыки применения в области технологии проектирования и тестирования программного обеспечения

Интерфейс интернет-магазина − главная цель работы − является удобным и простым в понимании интерфейсом. Сама вёрстка работает правильно, без ненужных прерываний и задержек при переходе на страницы. Адаптивность всех страниц присутствует.

Можно сделать вывод, что вся работа была построена на использовании нисходящей технологии проектирования, заключающейся в том, что данная разработка приложения начинается с создания его компоновки и определения структуры, на основе которых затем моделируются входящие в изделие другие детали и узлы [9].

В процессе проектирования были задействованы следующие фазы:

1) дивергенция — расширение границ проектной ситуации с целью обеспечения более обширного пространства поиска решения;

2) трансформация — стадия создания принципов и концепций (исследование структуры проблемы);

3) конвергенция — охватывает традиционное проектирование (кодирование, отладка, проработка деталей).

В ходе разработки вёрстки была присуща спиралевидная технология проектирования, так как при адаптивности страниц приходилось возвращаться назад, можно сказать по спирали, к уже готовым элементам на предыдущем этапе с целью дальнейшего их улучшения и доработки.

По степени использования типовых проектных решений было задействовано

оригинальное (индивидуальное), когда проектные решения разрабатываются «с нуля» в соответствии с требованиями заказчика. Характеризуется тем, что все виды проектных работ ориентированы на создание индивидуальных для каждого объекта проектов, которые в максимальной степени отражают все его особенности.

Следует отметить, что в ходе создания интерфейса для интернет-магазина была задействована визуальная технология программирования. Так как визуальное программирование является в настоящее время одной из наиболее популярных парадигм программирования. Оно состоит в автоматизированной разработке программ с использованием особой диалоговой оболочки. В последнее время визуальному программированию стали уделять намного больше внимания — в связи с развитием мобильных сенсорных устройств (КПК, планшеты).

# Список использованных источников

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

2. RAD\_(программирование) − Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/RAD\_(программирование)− Дата доступа: 17.12.2019

3. Интерфейс\_пользователя − Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Интерфейс\_пользователя −Дата доступа: 17.12.2019

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

5. Даккет, Д. HTML и CSS. Разработка и дизайн веб-сайтов/ Д.Даккет. −М.:Эксмо Т.А., 2014.− 480 с.

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

7. Sass − sass-scss.ru [Электронный ресурс]. – Режим доступа: https://sass-scss.ru/guide −Дата доступа: 17.12.2019

8. Gulp − gulpjs.com [Электронный ресурс]. – Режим доступа: https://gulpjs.com пользователя −Дата доступа: 17.12.2019

9. Макаровских, Т. Проектирование программных средств/ Т.Макаровских. −М.:Либроком Т.А., 2019.− 362 с.

# ПРИЛОЖЕНИЕ А

**Содержание электронного носителя**

На электронном носителе расположены следующие директории и файлы:

− Исходный код страниц сайта

− Файл «Курсовая работа «СОВРЕМЕННЫЕ WEB-ТЕХНОЛОГИИ ДЛЯ СОЗДАНИЯ РАСПРЕДЕЛЁННЫХ ПРИЛОЖЕНИЙ» с документацией к курсовой работе

− Презентация « СОВРЕМЕННЫЕ WEB-ТЕХНОЛОГИИ ДЛЯ СОЗДАНИЯ РАСПРЕДЕЛЁННЫХ ПРИЛОЖЕНИЙ Моисейченков Д.Ю»