

Отчет о проверке на заимствования №1

Автор: Тимофеева Анна Федоровна aftimofeeva@yandex.ru / ID: 37
Проверяющий: Тимофеева Анна Федоровна (aftimofeeva@yandex.ru / ID: 37)
Организация: Псковский государственный университет

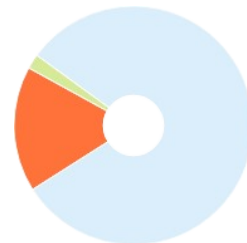
Отчет предоставлен сервисом «Антиплагиат» - <http://pskgu.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 161
 Начало загрузки: 18.06.2018 09:40:08
 Длительность загрузки: 00:01:06
 Имя исходного файла: VKR_Mavlyanov
 Размер текста: 9843 кБ
 Символов в тексте: 74702
 Слов в тексте: 9431
 Число предложений: 692

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
 Начало проверки: 18.06.2018 09:41:15
 Длительность проверки: 00:00:19
 Комментарии: не указано
 Модули поиска: Кольцо вузов, Модуль поиска общепотребительных выражений, Модуль поиска "ПсКГУ", Модуль поиска перефразирований Интернет, Модуль поиска Интернет, Модуль поиска переводных заимствований, Цитирование, Сводная коллекция ЭБС



ЗАИМСТВОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
16,87% ■	1,63% ■	81,5% ■

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
 Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общепотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
 Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
 Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
 Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
 Заимствования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
 Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Ссылка	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте
[01]	6,25%	7,42%	6 июня - День русского языка. Централ.	http://bibliopskov.ru	27 Окт 2017	Модуль поиска Интернет	17	27
[02]	0,79%	2,59%	Что такое SPA или одностраничный по..	http://codenet.ru	11 Апр 2016	Модуль поиска Интернет	4	5
[03]	1,83%	2,34%	Что такое SPA или одностраничный по..	http://codenet.ru	28 Янв 2017	Модуль поиска перефразирований Интернет	1	2
[04]	0,44%	1,7%	REST	http://ru.wikipedia.org	09 Сен 2017	Модуль поиска Интернет	2	10
[05]	0,53%	1,22%	Лучшие краеведческие издания 2013 г..	http://netess.ru	02 Дек 2016	Модуль поиска Интернет	7	11
[06]	1,04%	1,18%	Постановка целей и определение осно..	http://vikidalka.ru	19 Ноя 2016	Модуль поиска Интернет	12	15
[07]	1,01%	1,01%	Содрежание курса	http://mylektsii.ru	29 Янв 2017	Модуль поиска перефразирований Интернет	1	1
[08]	0,18%	0,93%	Сборник ПИС-2016.pdf	https://sibsutis.ru	01 Янв 2017	Модуль поиска перефразирований Интернет	1	3
[09]	0,71%	0,71%	Сборник материалов конференции за...	http://samsmu.ru	08 Янв 2017	Модуль поиска перефразирований Интернет	1	1
[10]	0,64%	0,64%	Принципы работы Hibernate — Merao...	http://megaobuchalka.ru	08 Янв 2017	Модуль поиска перефразирований Интернет	1	1
[11]	0%	0,61%	Содрежание курса	http://mylektsii.ru	16 Мая 2016	Модуль поиска Интернет	0	7
[12]	0,04%	0,58%	69993	http://e.lanbook.com	09 Мар 2016	Сводная коллекция ЭБС	1	8
[13]	0,56%	0,56%	Создание дизайна сайта: сколько нужн..	http://webstudio2u.net	13 Июл 2016	Модуль поиска Интернет	1	1
[14]	0,11%	0,56%	На областной конкурс «Псковская кни...	http://pln-pskov.ru	04 Мар 2015	Модуль поиска Интернет	2	6
[15]	0%	0,5%	На областной конкурс «Псковская кни...	http://pln-pskov.ru	05 Янв 2017	Модуль поиска перефразирований Интернет	0	2
[16]	0,13%	0,49%	АННОТАЦИЯ Рисунок 1 – Исходная ТРК.	http://tfolio.ru	18 Янв 2017	Модуль поиска Интернет	2	4
[17]	0,43%	0,43%	АННОТАЦИЯ Рисунок 1 – Исходная ТРК.	http://tfolio.ru	07 Янв 2017	Модуль поиска перефразирований Интернет	1	1
[18]	0,42%	0,42%	Развитие музыкально-творческих спо...	http://elibrary.ru	02 Янв 2018	Модуль поиска перефразирований	1	1

						Интернет		
[19]	0,1%	0,36%	Курсовая: "Основные возможности CS...	http://westud.ru	29 Мар 2016	Модуль поиска Интернет	2	2
[20]	0,35%	0,35%	Тонкости модульной системы ECMASc...	http://savepearlharbor.com	12 Апр 2016	Модуль поиска Интернет	1	1
[21]	0,05%	0,35%	Скачать - 0 байт	http://nauchkor.ru	12 Апр 2017	Модуль поиска Интернет	1	1
[22]	0%	0,31%	Современные веб-технологии	https://book.ru	03 Июл 2017	Сводная коллекция ЭБС	0	4
[23]	0%	0,31%	240404	http://biblioclub.ru	19 Апр 2016	Сводная коллекция ЭБС	0	4
[24]	0,29%	0,29%	СЛАВЯНСКАЯ МИФОЛОГИЯ И ЭТНОЛИ..	http://docs.gsu.by	29 Янв 2017	Модуль поиска перепаразирования Интернет	1	1
[25]	0,01%	0,27%	Домашнее задание к занятию 8 компь..	http://mir.zavantag.com	12 Мая 2016	Модуль поиска Интернет	1	1
[26]	0%	0,27%	Аннотированная программа конфере...	http://novsu.ru	26 Сен 2017	Модуль поиска Интернет	0	3
[27]	0,26%	0,26%	Курсовая: "Основные возможности CS...	http://westud.ru	05 Янв 2017	Модуль поиска перепаразирования Интернет	1	1
[28]	0,1%	0,2%	не указано	http://lib.druzya.org	16 Мая 2009	Модуль поиска Интернет	2	3
[29]	0,17%	0,17%	Разработка Web-сайта на основе HTML.	http://knowledge.allbest.ru	раньше 2011	Модуль поиска Интернет	5	5
[30]	0%	0,17%	Полная версия научной работы 2268 К..	http://scienceforum.ru	26 Сен 2017	Модуль поиска Интернет	0	1
[31]	0%	0,16%	Основы интернет-технологий. Учебно...	http://bibliorossica.com	26 Мая 2016	Сводная коллекция ЭБС	0	2
[32]	0,15%	0,15%	Большая книга CSS3	http://ibooks.ru	09 Дек 2016	Сводная коллекция ЭБС	1	1
[33]	0,09%	0,13%	Реферат - Учебник Macromedia Dreamw.	http://ronl.ru	раньше 2011	Модуль поиска Интернет	2	3
[34]	0,12%	0,12%	том 2	http://tusur.ru	14 Окт 2012	Модуль поиска Интернет	1	1
[35]	0,07%	0,07%	Разработка интернет-архива цветowych.	http://elib2.altstu.ru	26 Ноя 2016	Модуль поиска Интернет	1	1
[36]	0,7%	0%	не указано	не указано	раньше 2011	Модуль поиска общепотребительных выражений	14	50
[37]	0,93%	0%	не указано	не указано	раньше 2011	Цитирование	2	2

Текст документа

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования **36**

«Псковский государственный университет»

Физико-математический факультет

Кафедра прикладной информатики в образовании

Направление 09.03.03 «Прикладная информатика»

Профиль «Прикладная информатика в образовании»

«Допустить к защите»

Заведующий кафедрой

_____/В.Н. Мельник/

« ____ » _____ 20 ____ г.

Защищена с оценкой

« _____ »

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Мавлянова Азизбека Нажот угли

на тему

РАЗРАБОТКА ВЕБ-САЙТА “ТРАДИЦИОННЫЙ МИР ПСКОВСКИХ

КРЕСТЬЯН”

Руководитель

Кандидат технических наук,

доцент кафедры **36** прикладной информатики

в образовании _____/Кабаченко В.В./

Автор работы _____/Мавлянов А.Н./

Введение.....	4
Глава 1. Региональный этнолингвистический словарь «Традиционный быт псковских крестьян».....	8
1.1 Общие сведения о региональном этнолингвистическом словаре «Традиционный быт псковских крестьян».....	8
1.2 Состав этнолингвистического словаря «Традиционный быт псковских крестьян».....	9
1.3 Место этнолингвистического словаря «Традиционный быт псковских крестьян» в системе региональной лексикографии.....	11
1.4 Структура этнолингвистический словарь «Традиционный быт псковских крестьян».....	13
Вывод.....	17
Глава 2. Этапы разработки нового дизайна сайта.....	18
2.1 Анализ состояния старого дизайна сайта.....	18
2.2 Определения всех этапов разработки дизайна сайта.....	25
2.3 Создания прототипа дизайна сайта.....	31
2.4 Создания дизайна сайта.....	35
Вывод.....	42
Глава 3. Используемые программные обеспечения и языки программирования.....	43
3.1 Редактор кода Adobe Brackets.....	43
3.2 Язык разметки гипертекста “ 6 HTML”.....	45
3.3 Каскадные таблицы стилей “ 6 CSS”.....	51
3.4 Язык программирования “JavaScript”.....	57
3.5 Vue.js.....	61
3.6 Google ReCaptcha (invisible).....	68
3.7 Технология REST API.....	70
Вывод.....	72
Заключение	74
Список использованной литературы и источников.....	76
4	
Введение	
Объектом работы является сайт этнолингвистического словаря «Традиционный мир псковских крестьян». Первый опыт регионального этнолингвистического словаря отражает традиционный быт сельских жителей Псковщины. 1 Словарь создан в научно-образовательной лаборатории региональных филологических исследований Псковского государственного университета. В лаборатории филологического факультета сосредоточены ценнейшие материалы по диалектной лексике и фольклору, которые являются плодом многолетней экспедиционной работы студентов 18 и преподавателей кафедр русского языка и литературы. Словарь « 5 Традиционный быт псковских крестьян», 5 изданный в виде книги в 2012 г., представляет собой оригинальный лексикографический труд, отражающий через диалектное слово этнокультурные реалии Псковской земли. 5	
Словарь воплощает идею исторической и социокультурной значимости народной речи, в которой, как в зеркале, отражается вся полнота реального существования и мировосприятия человека. 1 Описывая этнографически	

значимые слова-реалии, словарь является средоточием разнообразной информации о традиционной народной культуре одного региона. Вместе с тем диалектный и фольклорный материал, зафиксированный на территории Псковщины, обладает многими общерусскими чертами, что расширяет территориальные границы этнолингвистического пространства словаря. Записи народной речи и фольклора относятся в основном к периоду второй половины XX – начала XXI вв. Однако, воспроизводя воспоминания диалектоносителей о недавнем и отдаленном прошлом, отражая многочисленные ссылки на то, как жили их предки, ¹ материалы словаря тем ¹

5
самым ¹ охватывают довоенный, а отчасти и дореволюционный период. Таким образом, значительно углубляется временной диапазон представленных материалов, что создает необходимую для этнолингвистического словаря степень историзма. ¹

В 2016 г. ³⁶ было принято решение о создании ³⁶ электронной версии данного этнолингвистического словаря. Была разработана программа, в которую была введена значительная часть вербальной информации. Кроме того, с целью содержательного расширения материала (выхода за пределы отражения только материальной культуры) в название электронного ресурса внесено изменение.

Целью настоящей работы является модернизация сайта «Традиционный мир псковских крестьян».

Данная цель обусловлена положением о том, что одним из важных преимуществ электронного ресурса является возможность его постоянного обновления и совершенствования.

Что составляет основу модернизации сайта и для чего нужна модернизация?

« Интернет-сайт, как и любой набор инструментов, имеет свойство устаревать. С течением времени требуются структурные изменения в любом механизме. Со временем также требуется определенная доработка сайта, которая реализуется в следующих аспектах:

Редизайн сайта

Изменения структуры сайта

Оптимизация навигации по сайту

Изменения набора функций сайта

Изменение структуры страниц

Добавление новых служб и сервисов.

Существует несколько основных причин, по которым следует прибегнуть к ³⁷

6
модернизации сайта». ³⁷

Так, существующее понятие интернет-«моды» диктует свои правила визуального представления, и устаревший дизайн не лучшим образом сказывается на привлечении посетителей, особенно молодого поколения.

Другая причина – техническая. В процессе эксплуатации сайта изменяются требования к функционалу, который был настроен на решение определенных задач и предоставление конкретных возможностей. Кроме того, появляются новые, более удобные системы управления сайтами. Расширяются технические возможности Интернета. К тому же, просуществовавший какое-то время сайт, зачастую превращается в хранилище самой разнообразной информации во многих случаях мешающей как посетителям, так и работе поисковых машин.

И, наконец, третья причина связана с технологическими возможностями сайта и его администрированием. С каждым днем растет количество владельцев сайтов, для которых имеют значение возможности управления сайтом.

Требования к квалификации редактора сайта, простота обслуживания системы, ограничения по хостинговым площадкам и многое другое складываются в

стоимость владения сайтом. Перевод сайта на высококачественную систему управления позволяет снизить постоянные расходы на его редактирование и последующие модернизации сайта.

Соответственно сформулированной цели в работе определены следующие задачи.

I. Комплексное описание этнолингвистического словаря «Традиционный мир псковских крестьян».

1. Дать общие сведения о словаре.
2. Рассмотреть состав словаря.
3. Показать возможности электронной формы по сравнению с традиционным книжным форматом.

7

II. Комплекс задач, связанных с обеспечением нового дизайна сайта, который дает возможность пользователю легко и быстро получить ту или иную информацию на сайте.

1. Обеспечить доступное восприятие информации для пользователя на сайте, используя подход минимализма в разработке нового дизайна сайта.
2. Обеспечить корректную цветовую палитру и декоративную графику, которая соответствует тематике сайта.
3. Обеспечить удобную компоновку блоков и элементов сайта для обеспечения легкой и удобной навигации по сайту.

III. Комплекс задач, направленных на изучение и применение современного программного обеспечения и информационных технологий.

1. Использование программного обеспечения Photoshop в разработке нового дизайна сайта.
2. Применения языков программирования, библиотек и фреймворков HTML5, CSS3, Javascript, Bootstrap, Vue.js в создании сайта.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка использованной литературы и источников. Результаты реализации практических задач отражены на сайте «Традиционный мир псковских крестьян» (www.xxx.ru).

8

Глава 1. Региональный этнолингвистический словарь «Традиционный быт псковских крестьян»

1.1 Общие сведения о региональном этнолингвистическом словаре

« Традиционный быт псковских крестьян»

Основными источниками словаря являются: картотека Псковского областного словаря (та ее часть, которая находится в ПсковГУ); псковский диалектный и фольклорно-этнографический архив (рукописный фонд и фонотека). К числу источников относятся также опубликованные выпуски Псковского областного словаря с историческими данными (Вып. 1–21. – Л. (СПб.), 1967–2009).

Словарь воплощает идею исторической и социокультурной значимости народной речи, в которой, как в зеркале, отражается вся полнота реального существования и мировосприятия человека. Описывая этнографически значимые слова-реалии, словарь является средоточием разнообразной информации о традиционной народной культуре одного региона. Вместе с тем диалектный и фольклорный материал, зафиксированный на территории Псковщины, обладает многими общерусскими чертами, что расширяет территориальные границы этнолингвистического пространства словаря. Записи народной речи и фольклора относятся в основном к периоду второй половины XX – начала XXI вв. Однако, воспроизводя воспоминания диалектоносителей о недавнем и отдаленном прошлом, отражая

многочисленные ссылки на то, как жили их предки, **1** материалы словаря тем самым **1** охватывают довоенный, а отчасти и дореволюционный период. Таким образом, значительно углубляется временной диапазон представленных материалов, что создает необходимую для этнолингвистического словаря степень историзма. **1**

9

1.2 Состав **1** этнолингвистического словаря « Традиционный быт псковских крестьян» **1**

Объектом словаря являются этнографизмы – комплексные единицы, отражающие понятия и реалии, характерные для традиционной русской, в том числе и псковской, деревни.

Если внимание этнографии сосредоточено на предметах мира человека, принадлежащего к определенному этносу или его части, то задачей лингвистики является описание наименований, отражающих мир реалий. Этнолингвистика как смежная и комплексная наука пытается синтезировать эти объекты. **1**

Определив этнографизм как категорию синкретичную, при работе над словарем необходимо было установить круг понятий, а следовательно и их наименований, которые могут соответствовать понятию этнографизма, т.к. содержание самого этого понятия не является однородным. Принято выделять две разновидности этнографизмов – общерусские и локальнорусские. Общенациональные этнографизмы обозначают предметы и явления русской жизни и русского быта, характерные для всего русского народа, воспринимаемые как типично русские явления, не свойственные другим народам. **1** При определении специфики этого вида этнографизмов русский язык противопоставляется другим национальным языкам. Примерами таких этнографических лексем являются изба, лапти, сарафан и под. Локальнорусские, или собственно этнографизмы, представлены диалектными наименованиями и принадлежат диалектной языковой системе. Таких единиц большинство как в языке в целом, так и в материалах настоящего словаря: **1** восьмикепина, ершевица, крутцовики, пуня и др.

В словаре получили описание около 650 единиц, среди них представлены как общерусские, так и локальнорусские этнографизмы, что позволяет раскрыть **24**

10

этнолингвистическую специфику Псковщины в общенациональном контексте **24** . **5**

Словник сформирован по тематическим группам с алфавитным принципом расположения заголовков словарных статей внутри каждой группы. **1**

Корпус первого опыта псковского этнолингвистического словаря составляют слова-объекты главным образом материальной культуры. Тематические группы, получившие описание в словаре, связаны с вещным миром псковского крестьянина: «Постройки. Традиционное жилище», «Прядение. Ткачество. Домотканое полотно», «Традиционная одежда, обувь», «Традиционная пища». **1**

Черты духовной культуры находят имплицитное отражение в привлечении данных местного фольклора.

При обращении к этнолингвистическому материалу авторы понимают, что привлекаемые наименования (слова или номинативные словосочетания)

обозначают понятия, явления и реалии, часто связанные со старым бытом, ушедшим традиционным укладом жизни. Этнографизмы, таким образом, могут носить архаический характер. Даже наблюдаемый в последние годы в деревне возврат к элементам натурального хозяйства не связан, как правило, с восстановлением традиционных форм хозяйствования, уклада жизни в целом.

Вместе с тем, следует подчеркнуть, что уход, утрата реалий традиционного быта не означает такого же стремительного исчезновения лексики, т.к. процесс устаревания слова более длительный. В связи с этим жизнь слова в памяти людей оказывается намного дольше бытования самой реалии. В то же время материалы словаря показывают и видоизменение смысла слова-этнографизма,

трансформацию области его применения.

Публикуя первый опыт регионального этнолингвистического словаря,

авторы продолжают работу над этой темой: ¹ уже выделены другие тематические группы, относящиеся к традиционным ремеслам (например, плетению из природных материалов, народной вышивке), и связанные с ними наименования хозяйственной утвари и корзин; выявлен состав традиционных групповых

11

микроэтнонимов, как хорошо известных на Псковщине, так и малоизвестных (например, ершееды, поляки, скобари и др.); предполагается расширить словарь уже выделенных тематических групп, а также усилить материалы, представляющие традиционную духовную культуру псковичей.

1.3 Место этнолингвистического словаря «Традиционный быт псковских крестьян» в системе региональной лексикографии

Особенности жанра этнолингвистического словаря опираются на само понимание этнолингвистики как междисциплинарной области. В целом можно сказать, что ³⁶ этнолингвистика как относительно молодая наука характеризуется нечеткостью границ, что отражается в неоднородности самого этнолингвистического направления [Герд 1995; Толстая, электронный ресурс; Березович 2007].

Теория отечественной этнолингвистики в новое время формировалась в тесной связи с решением научно-практических задач, а именно с созданием концепции лексикографического описания полевого экспедиционного материала при широком привлечении данных истории, этнографии, мифологии, фольклора и других смежных гуманитарных наук с целью реконструкции славянской духовной культуры. Так, в диссертации А.А. Плотниковой, выполненной под научным руководством Н.И. Толстого, этнолингвистический анализ оценивается с позиций взаимодействия трех аспектов: лингвистического, этнографического и фольклорного [Плотникова 1990], что указывает на расширенное понимание содержания этнолингвистики, выходящее за пределы только проявления признаков этноса через язык. Такой подход опирается на традицию, складывавшуюся в славистике на протяжении длительного времени [Бартминьский 2005; Плотникова 2000], что позволило в ходе многолетних полевых наблюдений, контактов и дискуссий создать свою школу

12

этнолингвистики: «Подлинным форумом польско-русского научного обмена стал люблинский ежегодный сборник "Этнолингвистика", основанный в 1988 году» [Бартминьский, электронный ресурс]. Основные положения московской школы этнолингвистики изложены в многочисленных публикациях ее основателей и сторонников [см.: Толстая, электронный ресурс]. Научно-практической реализацией идей основателя этой школы Н.И. Толстого явился фундаментальный этнолингвистический словарь «Славянские древности» (1995–2012 гг.).

Таким образом, мы наблюдаем классический пример единства научной теории и лексикографической практики, взаимно дополняющих друг друга. Причем в истории отечественной науки пример взаимосвязанного становления лингвистической теории и реализации лексикографических научно-практических проектов далеко не единственный, о чем свидетельствует, например, складывание научной школы Б.А. Ларина, в рамках одного из направлений которой сформировалась псковская диалектология, воплотившая идеи ее создателя в продолжающемся многотомном издании «Псковского областного словаря с историческими данными».

Предмет псковского этнолингвистического словаря находится в поле научно-практической интерпретации объектов региональной этнолингвистики средствами лексикографии. Примеров этнолингвистических словарей в

отечественной лексикографии не так много. Кроме уже выделенного словаря «Славянские древности», можно отметить издание, посвященное свадебному обряду, объединяющее в себе два корпуса: словарь свадебной терминологии и справочник свадебной символики [Подюков и др. 2004].

Недостаточная разработанность жанра этнолингвистического словаря отчасти объясняется тем, что лексикографирование культуры [Лиханова 2011] находит воплощение в лексико-фразеологическом фонде областных словарей. Ярким примером такого словаря является «Псковский областной словарь с

13 историческими данными». Будучи словарем полного типа, он содержит информацию о различных сферах материальной и духовной культуры псковских местных жителей в их прошлом и настоящем. Богатейшая картотека этого словаря, часть которой хранится в Псковском государственном университете, по объективным причинам не полностью находит отражение в самом словаре и оказывается доступной только специалистам. Кроме того, псковский архив содержит значительную по объему и представленности фольклорно-этнографическую часть, которая включает как рукописный фонд (экспедиционные тетради, третья часть которых – более 500 единиц – отсканирована), так и к настоящему времени оцифрованный аудиофонд (более 1055 час. звучания). Если материалы диалектной картотеки, хранящейся в Пскове, получают системное описание в «Псковском областном словаре», то фонды фольклорно-этнографического архива стали предметом активного исследования лишь в последние годы. Богатейшим источником этнолингвистической информации являются псковские сказки, изданные коллективом лаборатории, тексты которых содержат многочисленные указания и даже описания местных этнографических реалий [Народные сказки Псковского края].

Таким образом, к идее создания регионального этнолингвистического словаря привело наличие достаточной эмпирической базы, что позволило выявить и системно организовать материал, отражающий важные черты традиционной культуры, воплощенные в речи псковских крестьян.

1.4 Структура этнолингвистический словарь «Традиционный быт псковских крестьян»

Идеографический принцип организации словаря содержит в себе простор для расширения его корпуса без существенного изменения структуры. В словаре

14 разработаны фундаментальные с точки зрения этнографии тематические группы, однако это далеко не полный перечень 36 тем, лексика которых нуждается в этнолингвистическом описании. За пределами словаря остались темы, связанные с наименованиями хозяйственных предметов быта, домашней утвари, ремесел и изделий, в частности изготовленных плетением, орудий труда, обработки земли и мн. др.

Структура словарной статьи разработана авторским коллективом словаря. Реализованные принципы построения словарной статьи опираются на содержательный потенциал источников, которые включают не только объемный текстовый материал, но и данные о широкой вариативности диалектных наименований.

При широком понимании вариативности как репрезентативном множестве при наименовании одной и той же реалии неизбежно встает вопрос о наличии инварианта. Для практического решения вопроса заголовочного слова и статуса различных форм языковой репрезентации необходимо было разработать модель вариантно-инвариантных отношений внутри конкретных форм развертывания вариантного ряда. Основные формальные модификации наименования реализуются в следующих разновидностях: акцентологические варианты: бере'щеники, береще'ники,

берещеники'; бра'нина, брани'на; за'борка, забо'рка;

фонетические варианты: бульбени'ха, гульбени'ха (бульб.../

гульб...); ко'фта, ко'хта; ска'тереть, ска'терсть, ска'терть;

морфологические варианты (варианты грамматического рода,

вариантность форм единственного и множественного числа, вариантность основ

в пределах одного типа склонения): за'дво'рок – ед.ч., за'дворки – мн.ч.;

коню'шня, коню'шна; ста'тива – ж.р., ста'тиво – ср.р.;

словообразовательные параллели (однокоренные

разноаффиксные образования): блин, блине'ц, блино'к, блино'чек; за'уголок,

15

зауго'льник, зауго'льничек; кружа'стина, кружева'стина, кружева'тина;

номинативные параллели (близкие или тождественные по

смыслу и функции наименования в форме слова либо словосочетания):

барка'нник, морко'вник; е'вня / ё'вня, ови'н, рей, ри'га; малаха'й, треу'х,

уша'нка.

Представленные формы имеют разный языковой статус. Если первые три,

связанные с модификацией звучания и формально грамматическими

признаками, являются вариантами одного и того же слова **36**, то два последних

типа признаются авторами разными единицами. Причем словообразовательные

параллели, по общему мнению диалектологов, занимают промежуточное

положение между вариантами одного слова и разными словами-синонимами.

Таким образом, представленные в одном блоке в градационной

последовательности, разные формы наименования отражают свойственную

диалектной речи множественность номинации в пределах тождества и

отдельности слова.

Обратим внимание на то, что **36** указанные ряды включают также и

заголовочное слово, что должно продемонстрировать известную условность

вынесения в заголовок одного из вариантов (или отдельного наименования), т.к.

известно, что лексико-фонетическая, лексико-грамматическая и лексико-

словообразовательная репрезентация наименования этнографической реалии

может проявиться только в контексте – речевом, локальном (территориальном) и

социокультурном.

Приведем один из примеров словарной статьи, где этнографизм отражает

«сверхизбыточность» формы при отсутствии, к тому же, общерусского

наименования. Речь идет о названии особой веревочки, бечевки или специально

сплетенной из цветных нитей тесьмы, служившей для привязывания пучка

льняного волокна или шерсти к лопасти (личине) прялки. Изготовлением такой

тесьмы нередко занимались мужчины. Во время молодежных посиделок

16

(супрядок) парень мог сплести такой шнурок и подарить девушке в знак своего

расположения к ней, а также – перерезать его на прялке, что служило сигналом

окончания работы и начала молодежного гулянья.

Акцентологические варианты: мото'воз, мотово'з.

Фонетические варианты: мото'во'з, муту'воз, мутуо'с, муту'во'с,

мото'ус, мото'уз, мото'вуз.

Номинативные параллели: муту'вяз, мото'яз, муту'вис, мута'вис,

муту'вещ, мутузо'к, муту'вочка.

Далее следует зона иллюстративного материала, задача которого не просто

подтвердить использование слова в диалектной речи, но последовательно

отразить многообразие признаков описываемой реалии, а также языком самих

диалектоносителей дать функциональную, социальную, культурную оценку

этнографического и лингвистического факта.

Наименования этнографических реалий могут входить в состав

устойчивых выражений, что подчеркивает их функциональную широту в

диалектной речи. Отдельно приводятся примеры употребления слова в произведениях фольклорных жанров, собранных на территории современной Псковской области: сказках, легендах, преданиях, песнях, частушках, приметах, детском фольклоре. К сфере фольклора отнесены также пословицы и поговорки. Завершается словарная статья этимологической справкой, отражающей словообразовательные отношения производного слова внутри диалектной системы, а также раскрываются исторические связи слова или его корня со ссылкой на этимологические словари или специальные исследования, в том числе и авторские. Приведем пример такой справки для рассматриваемого слова.

Этимология. Название в различных вариантах известно всем русским говорам и многим славянским языкам: укр. мотоу́з, моту́з, блр. мо́туз, сербохорв. мату́зица, словен. motvòz, motòz и др. В русских памятниках письменности самые ранние фиксации относятся к XVI–XVII вв. Является

17
сложным словом, образованным от мотать < *motati и вязать, узел < *vǫzъ. Все многочисленные разновидности звучания (в русских говорах их число приближается к 50-ти единицам) являются результатом фонетических преобразований обоих корней. Древнее значение корня у глагола мотать связывают с понятием меры, измерения: общеславянское *motati, возможно, означало 'развивать и свивать с целью измерения, обмера чего-н.'. Ср. лит. matas 'мера', matuoti 'мерить, обмеривать' [Традиционный быт... 2012, с. 112–113].

18
Вывод
Краткий обзор материалов изданного словаря показывает, что при всей его информационной насыщенности рамки традиционного книжного формата издания не позволяют реализовать все этнолингвистические и культурологические возможности псковского диалектного и фольклорно-этнографического архива. Слабым местом книжного издания является отсутствие визуальных иллюстраций, имеющих особое значение для этнографизмов. Кроме того, в словаре остался недостаточно реализованным богатый потенциал псковского фольклорно-этнографического аудиоархива. Значительное число слов-реалий возможно проиллюстрировать функционированием в песенных и вербальных жанрах псковского фольклора. Все отмеченные «резервы» были очевидны для авторов в ходе подготовки издания, однако формат словаря не позволял полнее раскрыть культурную значимость языковых материалов и стоящих за ними этнографических реалий. С этой целью и осуществляется проект создания электронной версии псковского регионального этнолингвистического словаря.

Электронными словарями принято называть автоматизированные системы « 25 интеллектуального назначения с собранными и определенным образом упорядоченными наименованиями понятий в той отрасли, с которой работает или на которую опирается эта автоматизированная система» [37 Дубичинский 2008, с. 368].

Разработанная электронная база данных включает поля, соответствующие
19
структурным элементам словарной статьи в книжном издании. Помимо этого, имеется возможность прикрепления к уже сформированной, главным образом языковой, информации фотоматериалов, аудиофайлов.

Глава 2. Этапы разработки нового дизайна сайта
1.1 Анализ состояния старого дизайна сайта
Для того чтобы разработать новый дизайн сайта было проанализировано текущее на тот момент состояния регионального этнолингвистического словаря « 14 Традиционный быт псковских крестьян». 14

Рис.1 Старый дизайн сайта

И так, на рис.1. отображено существующий на тот момент старый дизайн сайта. Как мы можем увидеть, что это одностраничный веб сайт. Структура сайта состоит из следующих структурных блоков:

- 1) Header, то есть шапка сайта. (Рис.2.)
- 2) Блок для поле с поиском слов и фильтрации по категориям слов
- 3) Панель навигации по алфавиту. (Рис.4.)
- 4) Основной контент сайта. (Рис.5.)

21

- 5) Footer, то есть подвал сайта (Рис.6.)

Рис.2 Шапка старого дизайна сайта

22

Рис.3 Блок для поле с поиском слов и фильтрации по категориям слов

23

Рис.4 Панель навигации по алфавиту

24

Рис.5 Основной контент сайта

25

Рис.6 Подвал сайта

2.2 Определения всех этапов разработки дизайна сайта

26

Первым этапом определяем цель создания сайта и целевую аудиторию. Обсуждаем с заказчиком, кто и зачем должен посещать сайт? Какие действия должен совершить посетитель сайта, и что чем должно завершиться посещение сайта. Это помогает понять «механику» работы сайта и правильно сформировать структуру. Разрабатываем прототип сайта, где закрепляем и фиксируем принципы его работы и все взаимодействия внутри сайта. Совместно с заказчиком готовим контент. Ключевая идея - сайт это и есть контент, который оформляется определенным образом для удобства работы с ним посетителей сайта. Если у заказчика нет контента, нет понимая, что хочется донести до своих клиентов, то формировать структуру и дизайн сайта будет сложно или невозможно. Поэтому одной из ключевых задач на этапе подготовки к разработке дизайна – создание контента. Контент для сайта – это и текстовая информация и картинки (фотографии), видеоролики, схемы, таблицы, карточки товара, баннеры и т.д. Поэтому в соответствии с этим принципом 36 были предоставлены заказчиком следующие материалы:

- 1) Картинки по тематике сайта (Рис.7-14.)
- 2) Текстовая информация. Это книга “Традиционный быт псковских крестьян” (Рис.15.)
- 3) Контактная информация для обратной связи с администратором сайта. Это номер телефона, адрес электронной почты и сам юридический адрес.

27

Рис.7 Картинка заднего фона веб-сайта

28

Рис.8 Первая картинка слайдера на главной странице

Рис.9 Картинка заднего фона заголовка веб-сайта

29

Рис.10 Четвертая картинка слайдера на главной странице

Рис.11 Пятая картинка слайдера на главной странице

30

Рис.12 Шестая картинка слайдера на главной странице

Рис.13 Восьмая картинка слайдера на главной странице

31

Рис.14 Девятая картинка слайдера на главной странице

Рис.15 Словарь “Традиционный быт псковских крестьян”

32

Создание прототипов при веб-дизайне сайта — это основная графическая часть любого веб-дизайна. Их задачей является отображение структуры и схемы определенной страницы веб-проекта, которая не имеет цветов, шрифтов и каких-либо элементов брендинга.

Верстка в графическом редакторе.

Практически все используют Photoshop, как универсальное средство для работы с графикой. Обычно работа по созданию дизайна может занять от одного дня до нескольких недель. Это будет зависеть, сколько страниц стоит разрабатывать. Если нужна только главная и внутренняя – это несколько дней, если вы будете делать иные страницы, например, под какие-то сервисы, значит, стоит понимать, что это отдельный дизайн, который нужно делать заново, что займет больше времени.

2.3 Создания прототипа дизайна сайта

Создание прототипов является последним шагом перед переходом к созданию окончательного шаблона сайта. Эта одна из важных частей процесса, так как с ее помощью можно организовать все идеи, которые появились в момент создания набросков, и разработать окончательную разметку каждой страницы будущего сайта.

Большинство опытных дизайнеров советуют использовать Adobe Photoshop при создании прототипов. Занимаясь разработкой этого процесса, следует помнить, что прототипы не должны быть пиксель в пиксель и не стоит особо переживать по поводу цветов, типографии и других визуальных элементах. Необходимо сконцентрировать свое внимание на создании дизайна макета и на структуре страниц веб-проекта. Учитывая высшее упомянутые моменты при разработке прототипа сайта, я разработал достаточно понятный удобный прототип.

33

Наш прототип состоит из следующий стрех страниц:

- 1) Главная (Рис.16.)
- 2) Список слов (Рис.17.)
- 3) Контакты (Рис.18.)

Главная страницы прототипа подразделяется на следующие структурные блоки:

- 1) Шапка страницы
- 2) Hero section
- 3) Основной контент
- 4) Подвал страницы

Прототип страницы список слов подразделяется на следующие структурные блоки:

- 1) Шапка страницы
- 2) Секция с табами
- 3) Блок для на навигации по алфавиту
- 4) Подвал страницы

Прототип страницы контакты подразделяется на следующие структурные блоки:

- 1) Шапка страницы
- 2) Форма обратной связи
- 3) Подвал страницы

Рис 16. Прототип главной страницы веб-сайта

Рис 17. Прототип страницы “Список слов”

Рис.18 Прототип страницы “Контакты”

2.4 Создания дизайна сайта

Создание дизайна сайта в качестве одного из основных этапов включает обычно разработку дизайн-макета главной страницы, на основе которого затем уже выполняется разработка дизайна остальных страниц. При этом в практике современных веб-студий принято предлагать заказчикам дизайна сайтов сразу несколько вариантов дизайн-макета главной страницы, чтобы заказчик мог выбрать тот из них, который считает наиболее приемлемым. **13**

Фраза «красивый дизайн» может легко испугать опытного веб-мастера.

Самое главное — функциональность сайта. Уже затем быстрая загрузка и дружелюбный к пользователю интерфейс. А красота и великолепие — на заднем плане.

Поэтому, вряд ли кто сможет научить вас делать «красоту». Создание нового дизайна, прежде всего, заключается в создании функциональности. Задача в том, чтобы создать дизайн и расположить элементы сайта так, чтобы посетитель с легкостью нашел необходимую ему информацию. Вам нужно позаботиться о том, чтобы пользователь, зашедший на сайт, не сбежал оттуда немедленно, напугавшись или же сдерживая рвотные позывы. Чтобы так не случилось — используйте минимум анимации.

Вдобавок, анимированные объекты отвлекают внимание посетителя от главной части сайта — текста. Просто не используйте в оформлении различные «фишки» на подобию миленьких собачек или кошечек. В Сети и так полно мусора. Создавайте что-то действительно интересное.

В соответствии с прототипом, который был согласован с заказчиком, был разработан новый дизайн сайта. При разработке нового дизайна сайта были использована большая часть материалов, которых предоставил заказчик. На пример, для заднего фона всего сайта была использована картинка на рисунке выше. Для слайдера на главной страницы были использованы картинка с орнаментом и ряд других картинок, которые предоставил заказчик. Что касается палитры цветов, для нашего нового дизайна сайта был предложен голубой цвет как основной цвет сайта. Комбинирую все оттенки голубого, мне удалось нарисовать достаточно красивый дизайн сайта. Более того, я нарисовал совершенно новый логотип для сайта используя узоры, которые нарисованы на картинке с орнаментом.

В соответствии с прототипом был разработан дизайн трех следующих страниц:

- 1) Главная (Рис.19.)

- 2) Список слов (Рис.20.)
- 3) Контакты (Рис.21.)

Рис.19 Главная страница веб-сайта

Рис.20 Страница “Список слов” веб-сайта

Рис.21 Страница “Контакты” веб-сайта

На рис.19. отображен дизайн главной страницы, который был выбран и

одобрен заказчиком. На этой главной страницы мы можем увидеть следующие четыре основных секции страницы:

- 1) Шапка страницы
- 2) Секция с слайдером
- 3) Секция с основным контентом страницы
- 4) Подвал страницы

Как можно увидеть из рис.19. в секции “Шапка” расположены три блока.

Это блок с логотипом сайта, блок с навигации и блок с контактной информации сайта. С помощью навигационного блока пользователь легко и быстро может перейти на ту или иную страницы сайта без перезагрузки самого сайта, то есть, пользователю нет необходимости ждать пока страница перезагрузится и перенаправить его на нужную страницу. Данный функционал был реализован с помощью таких передовых фреймворком как Vue.js. Этот фреймворк будет рассмотрен более подробно на следующей главе данной работы.

Ниже секции “Шапка” расположена секция с слайдером в виде заставке для заднего фона данной секции. На переднем плане верхней части секции с слайдером находится название самого сайта. Ниже названии сайта расположен блок с полем для поиска слов. Это поле позволяет легко и быстро найти определения того слово которое пользователь будет искать. Плюс ко всему этому, реализована подсказка для поле с поиском в виде выпадающего списка слов. Этот функционал позволяет пользователя, вводя пару начальных букв того или иного слово в поле для поиска слов, увидеть подсказку в виде выпадающего списка. Из этого списка пользователь легко может выбрать то слово, которое он ищет и перейти на страницу, на которой подробно дается полная информация об этом слове. Что касается секции, с основным контентом, в этой секции есть обзорная информация в виде текста о самом словаре. Далее ниже секции с основным контентом расположена секция “Подвал”, в которой есть три блока. Это блок с логотипом, блок с гербом ПсковГУ и блок с контактной информации.

Что касается страницы “Список слов”, на этой страницы мы можем увидеть те самые нам знакомые секции как “Шапка” и “Подвал”, но с одним отличием. В секцию “Шапка” добавлен еще один блок поля для поиска слов. Кроме этого, на страницы “Список слов” есть еще два блока, таких как блок с определением слово и блок для навигации по алфавиту. В блоке с определением слово мы можем увидеть определения того или иного слово которые мы ищем. Что касается блока для навигации по алфавиту, мы можем легко и быстро найти то слово, используя данную навигация, нажимаю на ту или иную букву. При нажатии на любую букву, выпадает список слов начинающихся с этой буквой. Из списка может выбран одно слово. После выбора слово, выводится информация об этом слове в блоке с определением слово. И наконец-то, последняя страница сайта это страницы “Контакты”. На этой страницы расположены те же блоки, которые находятся на страницы “Список слов”, за исключением одного блока. Это блок с формой обратной связи. Эта форма позволяет пользователю обратиться к администратору сайта и задать свои вопросы, вводя свои данные на соответствующих полях формы. Ответ на запрос высылается на почту, указанную пользователем при заполнения полей формы, в виде электронного письма.

Вывод

В этой главе были рассмотрены следующие этапы разработки веб дизайна сайта:

Анализ текущего состояния веб сайта

Определения всех этапов разработки веб сайта

Разработка прототипа дизайна веб сайта

Создания нового дизайна веб сайта

Анализ текущего состояния веб сайта был проведен для того чтобы

выявить недостатки старого дизайна сайта и разработать новый дизайн сайта с

учетом современных и актуальных трендов разработки веб дизайна сайтов. В

ходе проведения анализа были найдены недостатки по структуре сайта, по

удобности использования интерфейса и по цветовой палитре сайта, которая не

соответствует тематике сайта.

В определении всех этапов разработки веб сайта были рассмотрены такие

важные моменты как определения цели создания сайта и целевую аудиторию.

Это помогло понять «механику» работы сайта и правильно сформировать

структуру.

Что касается разработки прототипа дизайна сайта, на этом этапе были

закреплены и зафиксированы принципы работы сайта и все взаимодействия

внутри сайта. Разработка прототипов веб проектов является последним шагом

перед переходом к созданию окончательного шаблона сайта.

На этапе создания нового дизайна веб сайта, сам дизайн сайта был

нарисован в соответствии с разработанным прототипом сайта. При разработки

нового дизайна сайта самым главным был функционал веб сайта. Уже затем

быстрая загрузка и дружелюбный к пользователю интерфейс. А красота и

великолепие — на заднем плане. И, между прочим, на вкус и цвет, как

говорится.

Глава 3. Использованные программные обеспечения и языки

44

программирования

3.1 Редактор кода Adobe Brackets

Конвертация любого веб-дизайна в реальный веб-сайт осуществляется с

помощью написания определенного кода, используя тот или иной язык

программирования. Для редактирования такого кода необходимо использовать

редактор кода. Редактор кода – это специальная программа, разработанная для

написания кода, в каком либо языке программирования. Суть такой программы,

заключается в максимальном облегчении жизни вебмастера. Такие редакторы

кода, как правило, сильно помогают в написании кода своим удобством. Я

использовал редактор кода Adobe Brackets для создания файловой структуры

проекта и для редактирования кода проекта. Я выбрал данный продукт выбрал

его по двум следующим причинам:

1) Бесплатность

2) Удобный интерфейс для написания и редактирования кода.

(Рис.23.)

Действительно, один из ключевых особенностей является его

бесплатность. Вы сможете быстро скачать этот продукт совершенно бесплатно с

официального сайта (www.brackets.io).

Что касается удобства использования, он обладает следующими удобными

свойствами:

1) Подсветка кода, что сильно облегчает навигацию в самом коде,

так как иногда код может быть очень большим. (Рис.24.)

2) Специальные кнопки, при нажатии на которые, автоматически

прописывается какая-нибудь функция, или, в случае с HTML – вставляется

тег.

Впрочем, если я начну перечислять, даже в общих чертах, все удобства,

получится книга не меньше, чем «Война и мир». Их может быть очень много.

45

Рис.23 Интерфейс редактора кода Adobe Brackets

Рис.24 Подсветка кода в редакторе кода Adobe Brackets.

3.2 Язык разметки гипертекста "HTML"

HTML (от англ. HyperText Markup Language — «язык разметки

гипертекста») — стандартный язык разметки документов в

Интернете. В основном Web-страницы создаются с помощью

языка HTML (или

XHTML). Язык HTML интерпретируется браузерами и отображается в виде

документа в удобной форме для человека. Гипертекст своему существованию

обязан интернету, хоть и создавался он совсем для другого. HTML позволяет

производить переход от одной части текста к другой, при этом эти части могут

храниться у разных пользователей на компьютерах. HTML создан специально

для разметки веб-страниц. Именно язык разметки дает браузеру необходимые

инструкции о том, как отображать тексты и другие элементы страницы на

мониторе. Не только различные браузеры, но и их версии могут по-разному

воспринимать и отображать на экране код. Следовательно, некоторые элементы

корректно выглядящие в браузере Mozilla Firefox могут выглядеть иначе в

Internet Explorer или Chrome. При верстке (создании страниц) HTML-страниц

необходимо просматривать их в самых популярных браузерах, например

Chrome, Mozilla Firefox, Opera, Internet Explorer, Safari. В дополнение к

официальной спецификации языка которая развивается, производители

браузеров от W3C (W3 Консорциум) их собственные элементы (расширения).

Затем некоторые из этих пунктов после всемирного признания в описании

следующих версиях языка получили участие. Инновационное решение

«фреймы» не были включены в спецификации. Но браузеры поддерживают

фреймы и многие книги по HTML, содержали описание без упоминания о том,

что это нестандартное изделие. Впоследствии, стандарт стал де-факто. В версии

4 они уже были включены в базу. HTML теги - это основа языка HTML. Каждый

тег несет в себе определенную информацию, он описывает документ или способ

47

форматирования текста. С помощью тегов задается все содержимое

вебстраницы. Тег помещается в угловые скобки <тег>. В большинстве случаев

для тега задается парный закрывающий тег, но в некоторых случаях он

отсутствует. Информация, заключенная между открывающим и закрывающим

тегом, называется его контейнером (содержимым тега). Общепринятым

современным стандартом считается версия HTML 5.0 Отличия XHTML от

HTML. XHTML (англ. Extensible HyperText Markup Language) - расширяемый

язык разметки гипертекста. Правильно не расширенный, а расширяемый. Это

означает, что XHTML дополняется (расширяется) до сих пор. Так что такое

XHTML? XHTML - это не новый язык, это набор дополнительных правил,

которые необходимо соблюдать при написании HTML кода. Главным различием

XHTML от HTML является способ обработки документа (интернет странички).

Необходимо дать еще одно определение. Парсер — программа или часть

программы, которая выполняет синтаксический анализ. Другое название -

синтаксический анализатор. Программа выполнения анализа всего строения

страницы т.е. кода страницы. В HTML (во время анализа) при нахождении

ошибки, она исправлялась, что затрачивало дополнительное время - браузеру

нужно было понять, что разработчик хотел написать. В XHTML не исправлялись

эти ошибки. Например, при ошибке в теге, он просто с остальным текстом

выводился. Еще одно отличие заключается в том, что все элементы должны быть

закреты, а одиночные теги должны после символов иметь знак /, например:
.

В моем проекте я придерживался метода блочной верстки в написании

HTML разметки. Для написания блочной разметки как правила используется

HTML тег "<div></div>". Так же рекомендуется использовать новые теги

HTML5 для повышения индексации сайта в поисковым рейтинге. В моем

проекте я использовал следующие HTML5 теги:

Header

48

Main

Section

Footer

Все эти теги кроме обладания всеми свойствами тега “<div></div>”, еще они обладают смысловым характером. Например тег “Header” предназначен для шапки веб сайта. Как правило, в секции “Шапка” находятся навигация сайта, логотип сайта и какие-нибудь контактные данные. В нашем случае в моем проекте в этой секции находятся логотип сайта, навигация и контактная информация. Что касается кода реализации секции “Шапка”, мы можем его увидеть в следующем куске кода:

```
<header class="header">

<div class="container">

<div class="logo">



</div>

<nav class="nav">

<ul class="nav__list">

<li class="nav__item">

<a href="index.html" class="nav__link">Главная</a>

</li>

<li class="nav__item">

<a href="listofwords.html" class="nav__link">Список слов</a>

</li>

<li class="nav__item">

<a href="contacts.html" class="nav__link">Контакты</a>

</li>

</ul>

</nav>
```

49

```
<div class="contacts">

<ul class="contacts__list">

<li class="contacts__item"><i class="fa fa-map-marker" aria-hidden="true"></i>

<span class="add">Адрес</span></li>

<li class="contacts__item"><i class="fa fa-phone" aria-hidden="true"></i>

<span>Телефон</span></li>

<li class="contacts__item"><i class="fa fa-envelope" aria-hidden="true"></i>

<span>Почта</span></li>

</ul>

</div>

</div>

</header>
```

Тег Main используется для описания основного контента страницы. Не рекомендуется использовать его больше одного раза. Реализацию кода с его участием можно посмотреть в следующем куске кода из нашего проекта:

```
<main>

<section class="hero">

<div class="hero__header">ТРАДИЦИОННЫЙ МИР ПСКОВСКИХ

КРЕСТЬЯН</div>

<div class="container">

<form action="" class="hero_search">

<input type="search" placeholder="НАЙТИ СЛОВО" class="search-input">

<button class="hero-submit">
```

```
<i class="fa fa-search" aria-hidden="true"></i>

</button>

</form>

</div>

</section>
```

50

```
<section class="about">

<div class="container">

<div class="acknowledgement">

<div class="acknowledgement__title">ПРЕДИСЛОВИЕ</div>

<div class="acknowledgement__text">

Здесь находится текст раздела предидловаия.

</div>

</div>

</div>

</div>

</section>

</main>
```

Обычно тер Section используется для разделения отдельных секции страницы которые не зависимы друг от друга. В отличие от тега Main, тега Section можно использовать по несколько раз. Реализацию кода с его участием можно посмотреть в следующем куске кода из нашего проекта:

```
<section class="hero">

<div class="hero__header">ТРАДИЦИОННЫЙ МИР ПСКОВСКИХ

КРЕСТЬЯН</div>

<div class="container">

<form action="" class="hero__search">

<input type="search" placeholder="НАЙТИ СЛОВО" class="search-input">

<button class="hero-submit">

<i class="fa fa-search" aria-hidden="true"></i>

</button>

</form>

</div>
```

51

```
</section>

<section class="about">

<div class="container">

<div class="acknowledgement">

<div class="acknowledgement__title">ПРЕДИСЛОВИЕ</div>

<div class="acknowledgement__text">

Здесь находится текст раздела предидловаия.

</div>

</div>

</div>

</div>

</section>
```

Тер Footer предназначен для описания так называемого подвала страницы. В секции подвал обычно находятся контактные данные сайта, навигация сайта, логотип самого сайта и логотипы партнеров и спонсоров проекта. В нашем случае на нашем сайте в секции Footer находятся логотип сайта, герб ПсковГу и контактная информация. Реализацию кода с его участием можно посмотреть в следующем куске кода из нашего проекта:

```
<footer class="footer">

<div class="container">

<div class="logo">



</div>
```

```

<div class="pskovgu-logo">



</div>

<div class="contacts footer__contacts">

<ul class="contacts__list">

52

<li class="contacts__item"><i class="fa fa-map-marker" aria-hidden="true"></i>

<span class="add">Адрес</span></li>

<li class="contacts__item"><i class="fa fa-phone" aria-hidden="true"></i>

<span>Телефон</span></li>

<li class="contacts__item"><i class="fa fa-envelope" aria-hidden="true"></i>

<span>Почта</span></li>

</ul>

</div>

</div>

</footer>

```

3.3 Каскадные таблицы стилей "CSS3" 33

CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки. В 6 основном используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к 6 разным XML-документам, например, к SVG или XUL. CSS 6 они служат для того чтобы отделить друг от друга 33 структуру страницы и ее содержание от ее внешнего вида. Если страница полностью написана на HTML, то каждый элемент кода определяет не только элемент содержимого страницы, но и его способ отображения. При использовании CSS кода все происходит по-другому. С помощью HTML описываются только очередность элементов содержания страницы и их классы. В CSS файле прописаны соответствующие классы. Каждому присвоен набор свойств. Когда в HTML элементу приписываем какой-то класс, то к нему применяются все свойства этого класса. Не надо писать все свойства каждый раз. Без CSS невозможно представить современную веб-разработку. Общепринятым современным стандартом считается версия CSS

53

3.0. Преимущества CSS3 - расширенной и усовершенствованной спецификации таблиц стилей. Оформление текстов и изображений на веб-страницах с помощью закругленных углов, теней, поворотов и трехмерных эффектов.

Особенности разработки меню, навигации и кнопок. В частности, теперь в стилях можно использовать анимационную трансформацию, переменные (возможно, но не точно), а также, что больше всего радует глаз - всякий 3D эффекты вроде теней над блоками 27 . 19

При разработке данного веб-сайта были использованы свойства CSS3 свойства. Например, для закругления углов кнопки "Приложения" на странице "Список слов" было использовано свойство "border-radius". Такие углы придают дизайну более красивый вид, нежели простые квадратики. 19 Код реализации закругления углов кнопки можно увидеть ниже.

```

.btn-success {
border-radius: 50px;
}

```

Данный код говорит о том, что для 36 элемента DOM дерева с классом "btn-success" применяется свойство CSS3 "border-radius". Это свойство закругляет углы кнопки "Приложения" на 50 пикселей.

В данной работе были использованы очень много полезных свойств CSS3, с помощью которых мне удалось красиво и удобно оформить разметку всего

сайта. Это можно увидеть на рис.23., рис.24., рис.25. Что касается реализации кода стилей разметки, мы можем увидеть кусок CSS3 кода для секции "Header"

ниже:

```
.header {
```

```
background-image: url("../images/header-bg.jpg");
```

```
background-position: center; 28
```

```
54
```

```
background-size: cover;
```

```
background-repeat: no-repeat; 28
```

```
padding: 10px 0; }
```

```
.logo {
```

```
width: 110px;
```

```
height: 78px;
```

```
display: inline-block; 35
```

```
vertical-align: middle; }
```

```
.logo__img {
```

```
display: block;
```

```
max-height: 100%;
```

```
max-width: 100%; }
```

```
.nav {
```

```
display: inline-block;
```

```
vertical-align: middle;
```

```
margin-left: 94px; }
```

```
55
```

```
.nav__list {
```

```
font-size: 0; }
```

```
.nav__list .nav__item {
```

```
font-size: 18px; }
```

```
.nav__item {
```

```
display: inline-block; }
```

```
.nav__item:first-child, .nav__item:nth-child(2) {
```

```
border-right: 2px solid #123762; }
```

```
.nav__link {
```

```
text-transform: uppercase;
```

```
text-decoration: none;
```

```
font-size: 18px;
```

```
font-weight: 600;
```

```
display: block;
```

```
color: #123762;
```

```
padding: 0 10px; }
```

```
.contacts {
```

```
56
```

```
display: inline-block;
```

```
vertical-align: middle;
```

```
margin-left: 480px; }
```

```
.contacts__item {
```

```
margin-top: 10px; }
```

```
.contacts__item:first-child {
```

```
margin-top: 0; }
```

```
.contacts__item .fa {
```

```
color: #123762;
```

```
font-size: 20px; }
```

```
.contacts__item .fa-envelope {
```

```
font-size: 18px; }

.contacts__item span, .contacts__item .fa {

color: #123762; }

.contacts__item span {

font-weight: 600;

margin-left: 5px; }

.contacts__item .add {

margin-left: 10px; }
```

57

```
.header-search {

display: inline-block;

margin-left: 220px;

position: relative; }

.header-search__input {

height: 40px;

width: 500px;

border: 2px solid #123762;

background-color: white;

padding: 10px; }

.header-submit {

position: absolute;

top: 0;

right: 0;

height: 40px;

width: 50px;

background: #123762;
```

58

```
border: none;

cursor: pointer; }

.header-submit .fa-search {

font-size: 25px;

color: #ffffff; }
```

3.4 Язык программирования “JavaScript” (ES2015)

JavaScript — это сценарный язык программирования, выполняемый непосредственно в браузере. Программы, написанные на нем, называют скриптами, и они интегрируются непосредственно в HTML страницу различными способами. Чаще всего код на языке JavaScript обрабатывают обычные браузеры. Практически все современные браузеры поддерживают JavaScript, например Internet Explorer, Opera, Google Chrome, FireFox и др. В браузерах есть специальный модуль, который может обрабатывать команды, написанные на этом языке и приводить их в понятный вид **17**. В **16** браузере пользователя исполняется код JavaScript, а браузеры еще называют «клиентами», то этот язык программирования еще называют клиентским. **16** JavaScript - одна из реализаций спецификации ECMAScript. Стандарт ECMAScript развивается и по мере этого переживает обновления. Одним из этих обновлений является ECMAScript 2015. ECMAScript 2015 - это новая версия стандарта ECMAScript. Она была утверждена в июне 2015 года. ES2015 включает в себя значительное обновление языка, которое сильно изменило то, к чему мы привыкли. После обновления JavaScript добавились следующие новые свойства:

Операторы let и const;

Стрелочные функции (arrow function);

Деструктуризация;

59

Классы;

Наследование;

(Default) export и import.

Переменная, объявленная с помощью оператора let будет видна в рамках блока, в котором объявлена. Таким образом, все, что нам необходимо для создания блочной области видимости (block scoping) - пара фигурных скобок { ... }. В частности это влияет на объявления внутри if, while или for. Рассмотрим следующий пример:

```
if (true) { let a = 15;
```

```
console.log(a); // 15
```

```
}
```

```
console.log(a); // ошибка, нет такой переменной
```

Это происходит потому, что переменная let видна непосредственно в том блоке, в котором объявлена.

Оператор const создает константу - переменную, которую нельзя изменить. Это можно увидеть в следующем коде:

```
const a = 10; a = 15 // ошибка
```

Что касается стрелочных функции (arrow funtion), в ES2015 появился новый синтаксис задания функций:

```
let sum = (a, b) => a + b; sum(5, 5); // 10
```

Эта запись примерно эквивалентна следующей записи:

```
let sum = function(a, b) { return a + b; } sum(5, 5); // 10
```

Следующее новое свойство JavaScript – деструктуризация.

Деструктуризация (destructuring assignment) - это особый синтаксис присваивания, при котором можно присвоить массив или объект сразу нескольким переменным, разбив его на части. Посмотрим на пример деструктуризации массива:

```
60
```

```
let [foo, bar] = [1, 2]; console.log(foo); // 1 console.log(bar); // 2
```

При таком присваивании первое значение массива пойдет в первую переменную foo, второе - в bar, а последующие, если есть, будут отброшены.

Классы, появившиеся в ECMAScript2015, представляют собой синтаксический сахар для существующего в языке прототипного наследования.

Таким образом, классы не вводят новую модель наследования, но они предоставляют более понятный и простой способ создания объектов. Для того чтобы создать класс необходимо воспользоваться ключевым словом class и указать имя класса.

```
class User() {
```

```
//Тело класса
```

```
}
```

Что касается свойства наследование, ключевое слово extends используется в объявлениях классов и выражениях классов для создания класса дочернего относительно другого класса. В примере ниже объявлено два класса: Foo и наследующего от него Bar:

```
class Foo {
```

```
constructor(name) { this.name = name; }
```

```
showName() { console.log(this.name); }
```

```
}
```

```
class Bar extends Foo {
```

```
showName() { console.log('\nI'm'); super.showName(); }
```

```
}
```

```
let bar = new Bar('bar'); bar.showName(); // I'm // bar;
```

Как мы видим, в классе Bar доступны как свои методы, так и методы родителя (через super). Т.е. методы родителя Foo можно переопределить в наследование. При этом для обращения к родительскому методу используется super.showName().

И последнее свойство, которое я рассмотрю – export (default) и import.

Оператор export (default) позволяет экспортировать сущности модуля, чтобы они были доступны из других модулей. У каждого модуля есть неявный объект `[[Exports]]`, в котором хранятся ссылки на все экспортируемые сущности, а ключом является идентификатор сущности (например, имя переменной). **20** Что касается свойства `import`, оно позволяет импортировать отдельные модули, которые экспортируются с помощью свойства `export`. Это можно увидеть в следующем примере кода:

```
Cube.js
export default function cube(x) {
  return x*x*x;
}

Use-cube.js
import cube from './Cube.js'
console.log(cube(3)); // 27
```

Вообще, JavaScript довольно мощный язык, на котором можно создавать целые веб-приложения.

В данном проекте язык программирования JavaScript был использован как основной язык всей динамики сайта. Данный язык был использован в паре с таким JavaScript фреймворком как Vue.js. Этот фреймворк будет более подробно рассмотрен далее в следующем разделе.

3.5 Vue.js

Vue.js – это тот прогрессивный фреймворк, который предназначен для создания пользовательских интерфейсов. В отличие от других фреймворков, Vue.js разработан пригодным для постепенного внедрения. Его ядро решает в первую очередь задачи уровня представления (view). Это упрощает интеграцию

с другими библиотеками и существующими проектами. С другой стороны, Vue.js полностью подходит и для создания сложных одностраничных приложений (SPA, Single-Page Applications), если использовать его совместно с современными инструментами разработки и дополнительными библиотеками.

Single Page Application – сокращенно SPA, в переводе на русский язык означает "Приложение одной страницы". Другими словами SPA – это web-приложение, размещенное на одной web-странице, которая для обеспечения работы загружает весь необходимый код вместе с загрузкой самой страницы. Приложение такого типа появились сравнительно недавно, с началом эры HTML5 и SPA является типичным представителем приложений на HTML5. **2** Давайте рассмотрим плюсы и **2** минусы данного подхода. Зачем всё это нужно и почему SPA так популярен? **2**

Первым плюсом стоит отметить тот факт, что приложения на SPA отлично работают на устройствах как стационарных, так и мобильных. "Большие" компьютеры, планшеты, смартфоны, и, в конце-концов, простые телефоны (некоторые) могут беспрепятственно работать с сайтами построенных по принципу SPA. Итак, первый "плюс" – работа на большом количестве устройств, а значит, создав одно приложение, вы получаете гораздо большую аудиторию пользователей нежели при использовании стандартного подхода. Далее второй "плюс" – богатый пользовательский интерфейс, так называемый User Experience. Так как web-страница одна, построить богатый, насыщенный пользовательский интерфейс гораздо проще. Проще хранить информацию о сеансе, управлять состояниями представлений (views) и управлять анимацией (в некоторых случаях).

Третий "плюс" – SPA существенно (в разы) сокращает так называемые "хождения по кругу", то есть загрузку одного и того же контента снова и снова.

Если ваш портал (сайт) использует шаблон, то вместе с основным содержанием какой-либо страницы посетитель сайта обязательно загружает разметку 3

63

шаблона. Да, кэширование данных на данном этапе развития WWW достигло высочайших результатов, но если нечего кэшировать, то и время, и ресурсы на это не тратятся.

Что касается минусов SPA, если вы программируете на C#, то единственным минусом SPA является необходимость изучения 3 JavaScript. Во всяком случае, других глобальных проблем мне выявить не удалось. 2

Данный веб-проект был разработан на основе принципов работы SPA, которые были рассмотрены выше. Для того чтобы разработать SPA веб-сайт был использован Vue.js фреймворк, который описывается в этом разделе выше.

Vue.js фреймворк позволил разделить весь веб-проект на так называемые компоненты приложения.

Компоненты — это переиспользуемые экземпляры Vue со своим именем. Мы можем использовать тот или иной компонент как пользовательский тег внутри корневого экземпляра Vue. Компоненты развивают идею плагинов. Каждый из них реализует какую-то свою возможность (а если нет существующих, то можно написать и свой). Если понадобится реализовать подобное в другом месте — легко переиспользовать плагин снова.

Взаимодействие можно описать простым интерфейсом: отправляем в плагин входные параметры, а для обратной связи можем отслеживать события.

Всё это справедливо и для компонентов. С тем лишь отличием, что компонент может представлять собой не только одну вещь (например, красивый и функциональный select), но и какую-то часть приложения, которая должна работать и выглядеть везде единообразно (например, форма комментирования, с аватаркой, редактором и красивым select'ом).

64

Рис.25 Иллюстрация разделения страницы на компоненты

Компонентный подход позволяет избежать мешанины кода и чётко выстраивать архитектуру приложения. Любую сложную страницу всегда можно разбить на меньшие составляющие. Каждую из таких частей при выделении в компонент проще поддерживать, а при необходимости повторять разбиение внутри компонента на ещё меньшие части. Это можно увидеть на рис.25.

Первым заметным плюсом подобного разбиения на компоненты будет удобство в поддержке — больше не нужно держать в голове логику всего приложения, можно сосредоточиться на конкретной его части. Вносить изменения или доработки нужно будет только в одном месте. Изолированность же компонентов избавит от появления конфликтов с другими частями приложения.

Поэтому применение компонентного подхода теперь широко используется во многих фреймворках. Vue не остался в стороне и предоставляет прекрасные возможности по работе с компонентами.

Что касается моего веб-проекта, он делится на следующие компоненты приложения:

Main.vue (компонент, который отвечает за отображения данных на главной странице)

AlphabetMenu.vue (компонент, который отвечает за отображения из

65

функционал навигации по алфавиту)

Contacts.vue (компонент, который отвечает за отображения формы обратной связи на странице “Контакты”)

Download.vue (компонент, который отвечает за скачивания материала, который прилагается для каждого слово)

Feedback.vue (компонент, который отвечает за отображения

сообщения говорящей о том, что валидация формы прошла успешна и сообщение было отправлено)

ItemCitations.vue (компонент, который отвечает за отображения информации о цитатах для каждого слово)

ItemDownload.vue (компонент, который отвечает за отображения списка скачиваемых материалов, которые прилагаются для каждого слово)

LetterDropdown.vue (компонент, который отвечает за отображения списка слов, который появляется после нажать на ту или иную букву в навигации по алфавиту)

Search.vue (компонент, который отвечает за поле для поиска слов)

SelectCategory.vue (компонент, который отвечает за список категорий)

Word.vue (компонент, который отвечает за отображения таб-виджетов на странице “Список слов”)

WordAccent.vue (компонент, который отвечает за отображения информации об ударениях для каждого слово)

WordCombinations.vue (компонент, который отвечает за отображения информации о разных комбинациях для каждого слово)

WordDescription.vue (компонент, который отвечает за отображения информации об определения слово)

66

WordEtymologies.vue (компонент, который отвечает за отображения информации об этимологии для каждого слово)

WordFolklor.vue (компонент, который отвечает за отображения информации о фольклорах для каждого слово)

WordVariants.vue (компонент, который отвечает за отображения информации о вариантах для каждого слово)

Все эти компоненты были собраны в один единый компонент, который называется App.vue. Более того, эти компоненты могут быть использованы по несколько раз в этом веб-проекты или в других проектах в целом.

При разработке данного проекта я использовал вторую версия данного фреймворка. Этот фреймворк мне позволил легко и быстро поделить весь проект на отдельные так называемые компоненты, которые значительно упростили разработку сайта. Кроме этого, данный фреймворк позволяет разработать такой функционал, который способен дать возможность пользователю перейти на другие страницы веб сайта без перезагрузки самого сайта. Ниже приведен основной код проекта с использованием фреймворка Vue.js.

```
App.vue :
<template>
<div id="app">
<header class="header clearfix">
<div class="container">
<div class="logo">

</div>
```

```
67
<nav class="nav">
<ul class="nav__list">
<li class="nav__item">
<a href="index.html" class="nav__link">Главная</a>
</li>
<li class="nav__item">
<a href="listofwords.html" class="nav__link">Список слов</a>
```

```
</li>

<li class="nav__item">

<a href="contacts.html" class="nav__link">Контакты</a>

</li>

</ul>

</nav>

<div class="contacts">

<ul class="contacts__list">

<li class="contacts__item"><i class="fa fa-map-marker" aria-hidden="true"></i>

<span class="add">Адрес</span></li>

<li class="contacts__item"><i class="fa fa-phone" aria-hidden="true"></i>

<span>Телефон</span></li>

<li class="contacts__item"><i class="fa fa-envelope" aria-hidden="true"></i>

<span>Почта</span></li>

</ul>

</div>
```

68

```
</div>

</header>

<div class="rout-views">

<router-view/>

</div>

<footer class="footer">

<div class="container">

<div class="logo">



</div>

<div class="pskovgu-logo">



</div>

<div class="contacts footer__contacts">

<ul class="contacts__list">

<li class="contacts__item"><i class="fa fa-map-marker" aria-hidden="true"></i>

<span class="add">Адрес</span></li>

<li class="contacts__item"><i class="fa fa-phone" aria-hidden="true"></i>

<span>Телефон</span></li>

<li class="contacts__item"><i class="fa fa-envelope" aria-hidden="true"></i>

<span>Почта</span></li>

</ul>

</div>
```

69

```
</div>

</footer>

</div>

</template>

<script>

export default {

name: 'App'

}

</script>
```

3.6 Google ReCaptcha (invisible)

Google ReCaptcha (invisible) - это капча, которая позволяет осуществлять фоновую валидацию действий пользователей на сайте. В отличие от ReCaptcha v2, она не требует от пользователей того, чтобы они нажимали на флажок (checkbox) "Я не робот". Она вызывается напрямую, т.е. тогда когда

пользователь нажимает на существующую кнопку на сайте или через вызов JavaScript API.

Суть её в том, что теперь не будет необходимости куда-либо нажимать, вводить или выбирать, чтобы подтвердить, что вы человек. Для пользователей всё будет прозрачно, если у «капчи» не будет сомнений в его «человечности». Google ReCaptcha (invisible) позволяет совершать пользователям на сайте какие-либо действия сразу, без ввода капчи. По умолчанию только самые подозрительные действия пользователей не позволят им пройти дальше. В этом случае Google ReCaptcha (invisible) предложит им решить капчу, то есть, определить действительно ли данную операцию совершает человек, а не робот. Чтобы начать использовать Google ReCaptcha (invisible), вам необходимо

70

зарегистрироваться для пары ключей API для вашего сайта. Пара ключей состоит из ключа сайта и секретного ключа. Ключ сайта используется для вызова службы ReCaptcha на вашем сайте или мобильном приложении. Секретный ключ разрешает связь между вашим сервером приложений и сервером ReCaptcha для проверки ответа пользователя. Для обеспечения безопасности секретный ключ должен быть безопасным. В нашем случае эта технология была использована для формы обратной связи, которая находится на странице “Контакты” (Рис.25). Google Recaptcha (invisible) проверяет самые подозрительные действия того или иного пользователя и не позволяет им пройти через валидацию формы. Рис.26 Технология Google ReCaptcha (invisible) на странице “Контакты”

3.8 Технология REST API

REST (сокр. от **4** англ. Representational State Transfer — « **21** передача состояния представления») — архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST представляет собой **9** **71** согласованный набор ограничений, учитываемых при проектировании распределённой гипермедиа системы. В определённых случаях (интернет магазины, поисковые системы, прочие системы, основанные на данных) это приводит к повышению производительности и упрощению архитектуры. В широком смысле компоненты в REST взаимодействуют наподобие взаимодействия клиентов и серверов во Всемирной паутине **9**.

Свойства архитектуры REST, которые зависят от ограничений, наложенных на REST-системы:

Производительность — взаимодействие компонентов системы может являться доминирующим фактором производительности и эффективности сети с точки зрения пользователя

Масштабируемость для обеспечения большого числа компонентов и взаимодействий компонентов. **4**

Преимущества архитектурного стиля REST:

Надёжность (за счёт отсутствия необходимости сохранять информацию о состоянии клиента, которая может быть утеряна);

Производительность (за счёт использования кэша);

Масштабируемость;

Прозрачность системы взаимодействия (особенно необходимая для приложений обслуживания сети);

Простота интерфейсов;

Портативность компонентов;

Лёгкость внесения изменений;

Способность эволюционировать, приспосабливаясь к новым требованиям (на примере Всемирной паутины). **10**

72

Вывод

В третьей главе 10 данной работы были рассмотрены все использованные программные обеспечения и языки программирования.

Один из таких программных обеспечений это - графический редактор Adobe Photoshop. С помощью данного графического редактора мне удалось очень легко и быстро разработать прототип веб сайта, а затем уже нарисовать новый дизайн сайта. Этому поспособствовали очень удобные инструменты интерфейса и разные полезные функции данного графического редактора, который был рассмотрен в данной главе.

Один из следующих программных обеспечений, который был использован это – Adobe Brackets. Adobe Brackets это редактор кода, который позволил мне создать структуру файлов проекта, а затем уже написать код проекта в соответствии с разработанным дизайном веб проекта. Все ключевые особенности, из-за которых я выбрал этот редактор кода были рассмотрены в этой главе.

Что касается языков программирования, я использовал для создания разметки HTML, который является языком разметки гипертекста. Именно этот язык разметки гипертекста позволил создать сам скелет кода разметки веб проекта без стилей.

CSS, который является каскадными стилями таблиц был использован как

73

формальный язык описания внешнего вида 6 веб 6 проекта. В основном

используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью 6 языка разметки гипертекста HTML .

Для создания динамики на сайте был использован такой язык программирования как JavaScript в паре с фреймворком “Vue.js”. JavaScript является интерпретируемым языком программирования. Веб-проект, который был создан мною был статичным, но идеи создания дизайна сайта предполагает создания динамики для некоторых элементов веб страницы. Одним из самых распространенных способов достижения данного результата является использование JavaScript на страницах сайта. Используя некоторые особенности этого языка программирования, я реализовал динамический функционал для веб проекта. Все ключевые особенности данного языка были рассмотрены в данной главе.

74

Заключение

Целью настоящей работы являлась модернизация сайта “Традиционный мир псковских крестьян”.

Цель была достигнута в ходе решения следующих задач. Для более четкой организации работы и структурирования текста задачи были объединены в три блока.

I. Комплекс задач, связанных с обеспечением нового дизайна сайта, который возможность пользователю легко и быстро получить информацию из словаря на сайте.

1. Обеспечить доступное восприятие информации для пользователя на сайте, используя подход минимализма в разработке нового дизайна сайта.

2. Обеспечить корректную цветовую палитру и декоративную графику, которая соответствует тематике сайта.

3. Обеспечить удобную компоновку блоков и элементов сайта для обеспечения легкой и удобной навигации по сайту.

II. Комплекс задач, связанных с обеспечением возможности загрузки файлов

75

различных форматов.

1. Обеспечить устойчивую возможность загрузки на сайт текстовых файлов в формате PDF.

2. Обеспечить загрузку аудиофайлов различного объема.

3. Обеспечить доступ к файлам в формате jpeg в базах данных.

III. Комплекс задач, направленных на изучение и применение современных информационных технологий.

1. В создании сайта использование языков программирования HTML, PHP, CSS, JavaScript, Vue.js.

2. Работа с базами данных, используемых СУБД MySQL.

В ходе выполнения данных задач я ознакомился на практике с библиотеками и языками программирования JavaScript, Vue.js, PHP и СУБД MySQL.

Выполненная работа принесла удовлетворение еще и потому, что была связана с реальными потребностями успешно работающего сайта, функциональные возможности которого значительно расширились.

76

Список использованной литературы и источников

1. W3Schools – Режим доступа: <https://www.w3schools.com> (дата обращения 15.02.2018).

2. htmlbook.ru – Режим доступа: <https://www.htmlbook.ru/> (дата обращения: 17.02.2018).

3. InfoDesigner – Режим доступа: <https://www.infodesigner.ru> (дата обращения: 24.02.2018).

4. Google ReCaptcha Режим доступа: <https://www.developers.google.com/recaptcha/intro/> (дата обращения: 7.06.2018).

5. Vue – Режим доступа: <https://www.vuejs.org> (дата обращения: 17.03.2018).

6. PHP – Режим доступа: <http://php.net/manual/ru/intro-what-is.php> (дата обращения: 11.04.2018).

7. PHP, MySQL и другие веб-технологии – Режим доступа: <http://www.php.su/> (дата обращения: 21.04.2018).

8. VikramVaswani XML and PHP; New Riders. – М., 2002. – 384 с.

9. Маркин А.В., Шкарин С.С. Основы Web-программирования на PHP. –М.: Диалог-МИФИ, 2012. – 256 с.

10.Большакова Н.В., Андреев В.К., Воробьева Л.Б., Лукьянова С.В., Моня Ю.Ю., 5

77

Площук Г.И., Побидько З.В., Смирнова А.Н. 5 Традиционный быт псковских крестьян - 5 Псков: ЛОГОС Плюс, 2012. – 284 с.

11. 5 Основы CSS и HTML: [Электронный ресурс]. – <http://www.web-lesson.ru> (дата обращения: 1.05.2018)

12.Печников В.Н. Создание Web-страниц и Web-сайтов. – М.: Триумф: 2010. – 370 с.

13.Томсон Л. Разработка Web-приложений на PHP и MySQL: Пер. с англ. /Лаура Томсон. – 2-е изд., испр. – СПб: ООО «ДиаСофтЮП», 2003. – 672 с.

14. Макфарланд Д. Большая книга CSS3. 3-е изд. — СПб.: Питер, 2014. — 608 с.: ил. — (Серия «Бестселлеры O'Reilly»).

15. 32 Роббинс Дж. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / Дженнифер Роббинс; [пер. с англ. М. А. Райтман]. — 4-е издание. — М. : Эксмо, 2014. — 528 с.

16.Айсманн, Кэтрин. Маски и композиция в Photoshop. : Пер. с англ. — М. : ООО "И. Д. Вильямс", 2007. — 560 с. : ил. — Парал. тит. англ.

17.Флэнаган Д. JavaScript. Подробное руководство. – Пер. с англ. – СПб: СимволПлюс, 2008. – 992 с., ил.

