Изм.

Лист

№ докум.

Подпись

Дата

Лист

4

кр.ПМ.03.МДК.03.01.09.02.03.19.17.00.00.пз

Разраб.

Шабанов И.Ф.

Провер.

Шепилова А.А.

Н. Контр.

Утверд.

Разработка Web–сайта туристического агентства

Лит.

Листов

38

4–П–2

Содержание

[Введение 5](#_Toc3805864)

[1 Постановка задачи 6](#_Toc3805865)

[Жизненный цикл программного обеспечения 6](#_Toc3805866)

[Постановка задач и целей при создании сайтов 7](#_Toc3805867)

[2 Описание предметной области 9](#_Toc3805868)

[3 Структура веб–сайтов 9](#_Toc3805869)

[4 Принцип коммутации с веб–сервером, на котором хранится веб–сайт 12](#_Toc3805870)

[Веб–приложения 12](#_Toc3805871)

[Три технологии, которыми должен владеть каждый веб–разработчик 14](#_Toc3805872)

[HTML 14](#_Toc3805873)

[CSS 14](#_Toc3805874)

[JavaScript 15](#_Toc3805875)

[phpMyAdmin 16](#_Toc3805876)

[4 Описание алгоритмов обработки данных 19](#_Toc3805877)

[5 Обоснование выбора программных и аппаратных средств для реализации поставленной задачи 20](#_Toc3805878)

[6 Описательная часть разработанного приложения (базы данных) 21](#_Toc3805879)

[7 Программный код с пояснениями 22](#_Toc3805880)

[8 Методы тестирование и отладки 22](#_Toc3805881)

[Заключение 23](#_Toc3805882)

[Список использованных источников 24](#_Toc3805883)

[Список стандартов по курсовому проектированию 25](#_Toc3805884)

[Приложение 26](#_Toc3805885)

# Введение

Курсовая работа – мероприятие, направленное на восполнение отсутствующих, систематизацию имеющихся и накопление новых, необходимых для выполнения (самостоятельно выбранного) задания знаний, а также, это замечательная возможность закрепить и отработать полученные в ходе обучения умения на практике. К сожалению, неиспользуемый материал очень быстро «стирается из памяти» (забывается), курсовая работа – возможность получить необходимые, комплексные знания, необходимые для её выполнения. IT–сфера оказала огромное влияние на современный бизнес, невооружённым глазом заметна повсеместная тенденция к автоматизации. Существует множество направлений, в которых задействованы IT–специалисты: проектирование и разработка веб–сайтов (вёрстка удобных интерфейсов и разработка функционала сайта), создание мобильных и настольных приложений, а также множество смежных специальностей, которые сотрудничают бок о бок на благо одной цели – проектировщики программного обеспечения, администраторы баз данных, системные аналитики (DevOps'ы – специалисты, стремящиеся сократить переход с одной стадии жизненного цикла программного обеспечения к другой, не теряя, а даже улучшая качество разрабатываемого продукта), разработчики интерфейсов (UX&UI–дизайнеры), тестировщики и другие. Хочется отметить и то, что студент имел возможность самостоятельно выбрать тему курсовой работы, по моему мнению, это является исключительно хорошей практикой, так как каждый студент может начать (продолжить) развиваться в интересном ему направлении. На данный момент, каждому верстальщику необходимо иметь в своём арсенале опыт работы со следующими технологиями: Photoshop, стандартом HTML5, CSS3 и jQuery (JavaScript (ES6)), либо TypeScript, при работе с фреймворком Angular. Фреймворки используются для создания веб–приложений. Верстальщику же, для трудоустройства (успешного фриланса), сейчас, достаточно знания библиотеки jQuery. Курсовая работа является важным этапом в формировании конкурентоспособного специалиста.

# 1 Постановка задачи

Планирование – важный процесс, позволяющий смоделировать многие детали будущей информационной системы или веб–сайта, не приступая к этапу программирования: интерфейс, функциональные возможности, используемые технологии и многое другое.

При отсутствии хорошего планирования (используемых технологий, интерфейса, функциональных возможностей проекта, требований к скорости выполнения), проект часто получается совсем не таким, каким изначально задумывался. Повышается сложность его разработки и сопровождения, так как программисту неизвестно в каком направлении двигаться. Хорошее, детальное планирование – возможность максимально близко приблизиться к желаемому результату.

Несколько цитат: «Кто не знает, куда направляется, очень удивится, попав не туда» – Марк Твен.

«Очень сложно прийти туда, куда нужно, если вы не знаете, чего конкретно хотите» – zirina479.livejournal.com/519609

«Кто не знает, куда идет, вероятно, придет не туда» – Питер Лоуренс.

Задача технического задания – это свести к минимуму разницу между представлениями о проекте обеих сторон: заказчика и исполнителя. Один раз хорошо обсудив все детали проекта, обе стороны останутся в плюсе: у программиста сократится количество лишней работу, а заказчик, в свою очередь быстрее получит желаемый результат, ведь останется больше времени на правку.

## Жизненный цикл программного обеспечения

ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes – (международный) стандарт ISO, описывающий процессы жизненного цикла программного обеспечения. Данный стандарт, используя устоявшуюся терминологию, устанавливает общую структуру процессов жизненного цикла программных средств, на которую можно ориентироваться в программной индустрии. Стандарт определяет процессы, виды деятельности и задачи, которые используются при приобретении программного продукта или услуги, а также при поставке, разработке, применении по назначению, сопровождении и прекращении применения программных продуктов.

До начала разработки веб – сайта, следует разбить жизненный цикл веб –сайта на несколько этапов, а также постараться представить, что и как будет реализовано на каждом временном промежутке. Также, следует оценить какие результаты должны быть достигнуты при помощи созданного функционала (какие должны быть использованы технологии, желаемое время выполнения имеющихся функций и так далее).

## Постановка задач и целей при создании сайтов

Во всех случаях процессу непосредственной разработки любого сайта предшествует выработка четкой мотивации, осознание необходимости существования интернет – ресурса. В определенное время человек начинает понимать, что наличие сайта становится необходимостью то ли для реализации какой–нибудь бизнес идеи, то ли для поиска и сплочения вокруг себя единомышленников в сфере определенного интереса, увлечения. Осознав проблему, человек начинает заниматься поиском путей ее реализации. Очень важно, чтоб на этом этапе уже были сформированы конкретные задачи и цели намеченного проекта. В контексте планируемого бизнес проекта к основным целям относятся:

1. Возрастание рейтинга бренда или компании, рост лояльности клиентов, узнаваемость среди аналогичных предложений
2. Рост клиентурной сети и компаньонов фирмы;
3. Наращивание выпуска продукции и возрастание продаж;
4. Непосредственный контакт с потребителями как существующими, так и потенциальными;
5. Информирование по актуальным вопросам, оперативное реагирование на предложения, жалобы;
6. Развитие партнерских связей;
7. Реализация через сеть Интернет услуг и товаров;
8. Исследования в сфере маркетинга.

Когда уже намечены конкретные цели, следует приступать к формированию задач, создать план конкретных действий, направленных на достижение поставленных целей. Главное внимание следует обратить на:

1. Разработку уникального, соответствующего целям проекта дизайна;
2. Удобную навигацию, доступный поиск информации на сайте;
3. Разработку дополнительных привлекательных и полезных пользователю сервисов;
4. Создание условий для общения и обсуждения (наличие форумов, блогов, досок объявлений)
5. Реализацию маркетингового инструментария (наличие функций обратной связи, организация опросов, анкетирования, тестирования).

В большинстве случаев эти задачи ставятся перед профессионалами в этой области. Существует большое количество специализированных компаний, веб–дизайн студий, а также частных лиц. И следующим шагом будет поиск оптимального варианта исполнителя заказа, постановка конкретных требований к веб–мастерам, бюджетные вопросы, временные условия реализации проекта.

Полный цикл создания сайтов

1. Поиск заказчика (бриф/разговор о создаваемом сайта): опрос, позволяющий получить от заказчика информацию об организации/лице, для которого создаётся сайт. В первую очередь пожелания о внешнем виде, содержимом сайта (сайт–визитка, интернет–магазин или одностраничный сайт (landing page), выстроенный по всем маркетинговым принципам). Данный опрос, можно провести используя Google Forms.
2. Прототипирование и копирайтинг (составление «скелета сайта», макета). Прототипирование сайта можно произвести в программе Adobe InDesign и на веб–сайте: ninjamock.com. На данном этапе важно понять следующее: кто является целевой аудиторией проектируемого (будущего) сайта, а также понять как правильно составлять «продающие тексты», которые будут мотивировать пользователя остаться на сайте и совершить покупку.
3. Дизайн–концепция. На данном этапе моделируется внешний вид компьютерной и мобильной версии будущего сайта, но функционал, например, переход с одной страницы на другую, возможность оплаты, будет реализована на следующем этапе.

4.1. Вёрстка и программирование. На этом этапе, «сайт оживает», то есть работает (сайт воплощён в коде в виде блоков (при помощи HTML и CSS), а также, реализован весь необходимый функционал, при помощи языка программирования).

4.2. Посадка на систему управления (CMS: MODX/WordPress/Joomla) –обеспечение возможности добавления, редактирования или удаления контента (материалов) с сайта, не требующее знаний вёрстки/программирования. Это очень востребовано заказчиками.

5. Наполнение и администрирование. Получение от заказчика реальной информации (об оказываемых частным лицом/компанией услугах), фотографий и их дальнейшая публикация на сайте. Администрирование подразумевает возможность передачи управления доступом к сайту заказчику (прав администратора).

6. Публикация и тесты. На данном этапе происходит размещение сайта на хостинге (сайт привязывается к приобретённому домену, а также, оплачивается хранение содержимого сайта на веб–сервере) и он становится доступным для общего доступа. Помимо этого, на этом этапе происходит тестирование скорости загрузки сайта и её оптимизация за счёт уменьшения размера картинок (путём сохранения .png картинок в .jpg формате), кроме этого, обязательно тестируется работа сайта на различных устройствах и разрешениях экрана.

7. Продвижение и развитие сайта. Продвижение сайта его разработчиком или другими людьми, занимающимся контекстной рекламой (в поисковых системах Google и Yandex) и SEO–продвижением – действиями, направленными на создание таких условий, когда сайт сам начнёт «органически» поднимать себя вверх в выдаче поисковой сети. Это и использование ключевых слов и описания в мета–тэгах каждой страницы, заголовков страниц, тексты, в которых будут использованы упоминания/ссылки на другие страницы сайта, которые точно так же будут содержать мета–тэги.

8. Масштабирование сайта – этап, до которого развиваются немногие сайты. Масштабирование сайта – возможность простого, беспроблемного внедрения новых разделов, модулей или скриптов, это и возможность простого редизайна сайта.

От разработчиков, обладающих всеми описанными выше навыками, при предложении собственных услуг, зачастую можно услышать фразу: «создание сайта под ключ».

# 2 Описание предметной области

Веб–сайт–прекрасная возможность донести до потенциального клиента сведения об оказываемых компанией услугах (в данном случае, о предлагаемых вариантах экскурсий), организовать и упростить обратную связь. Таким образом компания увеличивает шанс на получение дополнительной прибыли и снимает с себя часть работы, отвечая на часто задаваемые вопросы.

Веб–страница – простой документ, отображаемый на экране компьютера посредством браузера. Такой документ написан языком HTML (который мы рассмотрим более детально в других статьях). Веб–страница может содержать множество различных материалов, таких как:

1. Cтилевая информация – контролирование страницы по восприятию и ощущению
2. Cкрипты – которые делают страницу более динамичной и удобной в использовании для пользователей
3. Медиа – изображения, музыка и видео.

Примечание: браузеры зачастую могут отображать некоторые документы в формате PDF файла или изображения, но термин веб–страница больше относится непосредственно к HTML–документам.

# 3 Структура веб–сайтов

Веб–сайт – это совокупность веб–страниц, объединённых, общим дизайном (стилем). Обычно, файлы сайта хранятся либо в одной директории (папке), либо в подкаталогах главной папки проекта, но, всегда соответственно их типу: например, хорошим тоном является наличие в папке с проектом папок dist и src. В папке src хранятся исходные файлы, над которыми ещё ведётся работа. В этой папке, может быть создана ещё одна папка с именем img, хранящая изображения, так как название папки должно передавать содержимое. Стили хранятся в папке с именем css, скрипты в папке с именем js, также, может быть создана папка less, libs и другие, css, img, js – минимальный набор.

Каждая страница сайта содержит прямые ссылки (практически всегда выделенные части текста, по которым можно кликнуть мышью), что позволяет пользователю быстро переходить от одной страницы веб–сайта к другой.

Чтобы получить доступ к веб–сайту, следует набрать его доменное имя в адресной строке браузера, и браузер отобразит главную страницу сайта или, по–другому, домашнюю страницу. Веб–страницу и веб–сайт особенно легко спутать между собой, когда сайт содержит всего одну страницу. Такой сайт иногда называют одностраничным веб–сайтом.

Веб–сервер – компьютер, предоставляющий в сеть один или множество веб–сайтов (хостинг). Понятие "хостинг" – означает, что все страницы и прикрепленные к ним файлы содержатся на данном компьютере. То есть, веб–сервер будет отправлять любую страницу с сайта по запросу любого пользователя, что и будет хостингом для браузера пользователя.

Не следует путать понятия веб–сайта и веб–сервера. Например, когда кто–либо говорит: "Мой веб–сайт не отвечает", на самом деле это означает, что это веб–сервер не отвечает на запрос, и поэтому недоступен и сам сайт. Более того, так как веб–сервер может разместить несколько сайтов, термин веб–сервер никогда не используется для обозначения веб–сайта, так как это могло бы привести к большой путанице. Вернемся к предыдущему примеру: если бы мы сказали: "Мой веб–сервер не отвечает", это значило бы, что на этом сервере нет доступных сайтов в данный момент.

Поисковые системы являются распространенной причиной путаницы в сети. Поисковая система – это специальный вид веб–сайта, который помогает пользователям найти нужные страницы других сайтов.

Наиболее популярные поисковые системы: Google, Bing, Yandex, DuckDuckGo, и многие другие. Некоторые из них универсальны, а какие–то ориентированы на определенную область.

Раньше интернет–сайты создавали на чистом HTML, и это было неудобно, так как все данные были представлены как отдельные HTML–файлы. Нельзя было осуществлять поиск, группировку, сортировку информации. К тому же, информация могла часто дублироваться. При появлении PHP (JavaScript и вариантов его исполнения в виде различных фреймворков) у веб–мастеров появилась возможность разделения сайта на его каркас и данные в базе. Теперь структуру сайта можно хранить отдельно от контента, что позволяет быстрее и удобнее администрировать веб–ресурс, легко дорабатывать его дизайн и функционал.

Структура веб–ресурса хранится в коде или в отдельных шаблонах (специальных файлах). Контент размещается в базе данных – определенном наборе таблиц с однотипными данными.

Допустим, необходимо создать обычный сайт–визитку. У разрабатываемого ресурса будет своя, отдельная структура и база данных. В базе будут представлены несколько таблиц: первая – с содержимым страниц, вторая – с новостной лентой, третья – с фотогалереей.

# 4 Принцип коммутации с веб–сервером, на котором размещён веб–сайт

Как уже говорилось ранее, выполнение многих процессов становится всё более автоматизированным и менее времязатратным, так, например, пользователи при открытии браузера, вводят в адресной строке не ip–адрес, состоящий из цифр, а название нужного сайта. Данную возможность обеспечивает система доменных имён (DNS), которая хранит соответствие имени сайта его ip–адресу в виде таблицы. Возможно, каждый пользователь сталкивался с тем, что при первом обращении к тому или иному сайту, процесс полной загрузки сайта длится дольше, чем последующие. Это объясняется тем, что веб–браузер кэширует (сохраняет) на компьютер пользователя необходимые для загрузки страницы данные, в том числе информацию об ip–адресе сайта, что, также, обеспечивает более быстрый доступ к нужному сайту. В случае, если на компьютере нет информации об ip–адресе запрашиваемого сайта, происходит обращение к resolver’у (ISP–провайдеру), каждый интернет–провайдер знает адрес корневого сервера, именно resolver (в порядке очереди) обращается к корневому серверу, который хранит информацию о доменах верхнего уровня (.com, .org, .ru и так далее). Всего, на данный момент существует 13 корневых серверов.

Веб–приложения

Веб–приложение–клиент–серверное приложение, в котором клиент взаимодействует с сервером при помощи браузера, а за сервер отвечает веб–сервер. Логика веб–приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб–приложения являются межплатформенными службами.

Любое веб приложение состоит из двух частей: Back–End и Front–End. Соответственно и программистов делят также, по сути это две несвязанные между собой касты разработчиков.

Бекэнд–серверная часть приложения, которая не видна пользователям. Сюда относится: авторизация, хранение и обработка данных, email рассылки и тому подобное. Скажем так бекэнд это то, что под капотом. Языков программирования тут множество, вот самые распространенные из них:

1. PHP
2. ASP.NET
3. Java
4. Python
5. Ruby
6. Node.js

С текущими требованиями к программистам, для достижения уровня middle пройдет не менее нескольких лет, поэтому, следует сосредоточиться на одном.

Сервер во многом используется для работы с данными. Поэтому нужно будет выучить как минимум бы одну из реляционных баз данных:

1. Oracle
2. Microsoft SQL Server
3. MySQL
4. PostgreSQL

По сложности это примерно тоже, самое, что и освоение языка программирования. Во всех перечисленных БД есть часть, которая одинакова везде–так называемый SQL (язык запросов), а также, процедурный язык который в каждой базе свой. Так как SQL все–таки един, а он основа всему, что связано с обработкой данных, то переход на другую базу данных, не является такой уж и проблемой. Изучив, необходимый набор технологий единожды, на них можно проработать длительное время.

Язык программирования тут один – JavaScript, но есть свои нюансы. JavaScript на текущий момент времени имеет два стандарта, которые широко используются: ECMAScript 5 и ECMAScript 6. Разница между ними состоит в том, что в ECMAScript 5 отсутствует полноценное ООП. В ECMAScript 6 оно есть, но поскольку этот стандарт появился относительно недавно, уже большая часть приложений написана на ECMAScript 5, в результате чего нужно будет освоить оба стандарта и как следствие два разных подхода к разработке.

Еще один момент, который нужно учитывать это то что один и тот же код, написанный на JavaScript может по–разному отрабатывать в разных браузерах и на разных ОС. С текущими требованиями к приложениям как правильно нужно будет писать так называемый кроссбраузерный код, то есть код, результат выполнения которого будет давать везде один и тот же результат.

Описанные проблемные моменты решены в современных JavaScript фреймворках (например, в таких как Angular и React). Но, в проектах, которые написаны без использования фреймворков, эти моменты остаются.

Для Front–End характерно очень быстрое развитие трендов. Постоянно обновляются версии фреймворков, появляются новые библиотеки.

Три технологии, которыми должен владеть каждый веб–разработчик

HTML

HTML (от английского HyperText Markup Language – «язык гипертекстовой разметки») – стандартизированный язык разметки документов во Всемирной паутине. Большинство веб–страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства. Не является языком программирования. Технология CSS позволяет придать нужный вид (стиль) созданной с помощью HTML разметки.

CSS

Cascading Style Sheets (CSS) – это язык иерархических правил (таблиц стилей), используемый для представления внешнего вида документа, написанного на HTML или XML (включая различные языки XML, такие как SVG и XHTML). CSS описывает, каким образом элемент должен отображаться на экране, на бумаге, голосом или с использованием других медиа средств.

CSS является одним из основных языков свободной веб–разработки, который стандартизован спецификацией W3C. Стандарт CSS делится на уровни: CSS1 в настоящее время устарел, CSS2.1 – рекомендован для применения, а CSS3, разбитый на более мелкие модули, развивается на пути стандартизации.

JavaScript

JavaScript – это кросс–платформенный, объектно–ориентированный скриптовый язык, являющийся небольшим и легковесным. Внутри среды исполнения JavaScript может быть связан с объектами данной среды и предоставлять программный контроль над ними.

JavaScript включает стандартную библиотеку объектов, например, Array, Date и Math, а также базовый набор языковых элементов, например, операторы и управляющие конструкции. Ядро JavaScript может быть расширено для различных целей путем добавления в него новых объектов, например:

JavaScript на стороне клиента расширяет ядро языка, предоставляя объекты для контроля браузера и его Document Object Model (DOM). Например, клиентские расширения позволяют приложению размещать элементы в форме HTML и обрабатывать пользовательские события, такие как щелчок мыши, ввод данных в форму и навигация по страницам.

JavaScript на стороне сервера расширяет ядро языка, предоставляя объекты для запуска JavaScript на сервере. Например, расширение на стороне сервера позволяет приложению соединяться с базой данных, обеспечивать непрерывность информации между вызовами приложения или выполнять манипуляции над файлами на сервере.

JavaScript и Java похожи в одних отношениях, но фундаментально отличны в других. Javascript напоминает Java, но не имеет статической типизации и строгой проверки типов, которыми обладает Java. JavaScript следует большей части синтаксиса Java в выражениях, соглашениях об именовании и основного потока управления конструкциями, поэтому он был переименован из LiveScript в JavaScript.

По сравнению с Java, JavaScript является языком с очень свободной формой. Вам не надо объявлять переменные, классы и методы. Вам не надо беспокоиться о том, являются ли методы публичными (public), приватными (private) или защищенными (protected), а также вам не надо реализовывать интерфейсы. Переменные, параметры и возвращаемые функциями типы не являются явно типизированными.

Java – это язык программирования, основанный на классах и предназначенный для быстрого выполнения и безопасности типов. Безопасность типов означает, например, что вы не можете привести тип integer к типу object reference или получить доступ к приватной памяти, изменяя байт–код Java. Ориентированная на классы модель Java означает, что программы состоят исключительно из классов и их методов. Наследование классов и строгая типизация в Java обычно требуют тесно связанные иерархии объектов. Эти требования делают программирование на Java более комплексным, чем программирование на JavaScript.

По духу JavaScript происходит от небольших, динамически типизированных языков, таких как HyperTalk и dBASE. Эти скриптовые языки предлагают инструменты программирования для гораздо более широкой аудитории благодаря более простому синтаксису, специализированной встроенной функциональности и минимальным требованиям для создания объектов.

phpMyAdmin

phpMyAdmin – веб – приложение с открытым кодом, написанное на языке PHP и представляющее собой веб–интерфейс для администрирования СУБД MySQL. PHPMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб–разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд, предоставляя дружественный интерфейс.

На сегодняшний день PHPMyAdmin широко применяется на практике. Последнее связано с тем, что разработчики интенсивно развивают свой продукт, учитывая все нововведения СУБД MySQL. Подавляющее большинство российских провайдеров используют это приложение в качестве панели управления для того, чтобы предоставить своим клиентам возможность администрирования выделенных им баз данных.

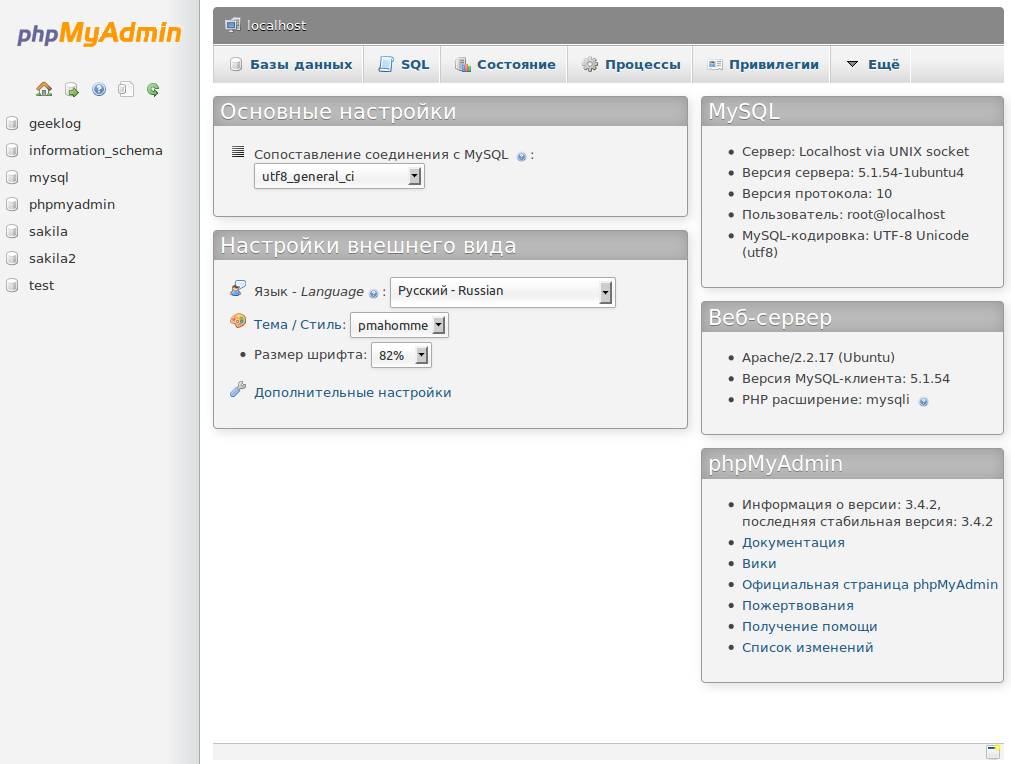


Рисунок 1 – Главная страница пвеб–приложения phpMyAdmin

На рисунке показан интерфейс веб–приложения phpMyAdmin. В левой части расположены загруженные на локальный (localhost/127.0.0.1) или веб–сервер базы данных. Там же, в виде иконок в левой и в верхней части окна расположена панель навигации.

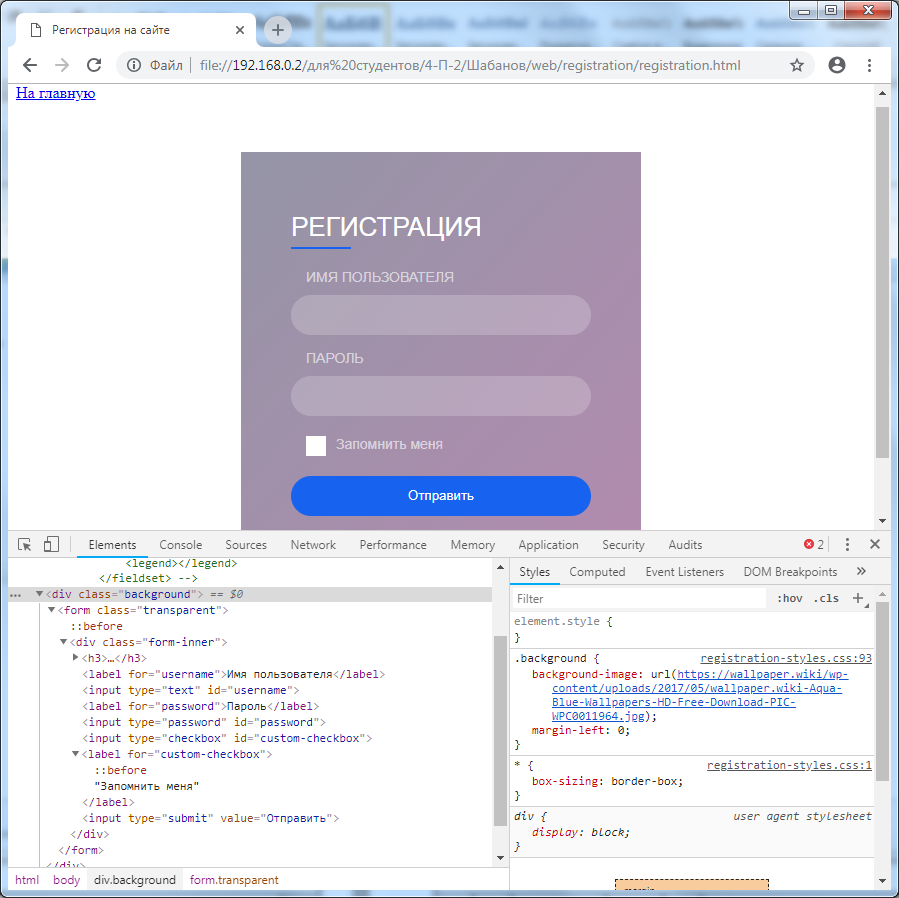


Рисунок 2 – инструменты разработчика в Google Chrome

Компания Google в своём браузере Google Chrome активно развивает встроенные инструменты разработчика: DevTools, доступ к которым можно получить используя горячую клавишу: F12 (сочетание клавиш Ctrl+Shift+I), либо, нажав правой кнопкой мыши в окне браузера и выбрав в выпадающем контекстном меню пункт: просмотреть код. В левой части инструментов разработчика расположен HTML код, а в правой, CSS код.

# 5 Описание алгоритмов обработки данных

Так как на одной из страниц: about.html была подключена карта от компании Yandex, которой требуется загрузка API Яндекс.Карт, встраивание было осуществлено через тэг–контейнер: <iframe>…</iframe>, также, это можно было осуществить вторым способом: осуществив подключение внешних скриптов, а именно, используя элемент script в заголовке HTML–документа. Например:

Способ 1 (используя тэг iframe):

<iframe src="https://yandex.ru/map–widget/v1/?um=constructor%3A48d30462b18dfa9d34b9136975ff13e69145320b392c26e0a42cd5e4ea004bcd&amp;source=constructor" width="1102" height="551" class="bordernone"></iframe>

Способ 2 (подключив внешний скрипт):

<head>

<script src="https://apimaps.yandex.ru/2.1/?apikey=<ваш API–ключ>&lang=ru\_RU" type="text/javascript">

</script>

</head>

В стандартном браузере мобильной операционной системы Android и Apple iOS версии ниже 3.2 жест масштабирования над картой приводит к увеличению масштаба всей страницы средствами браузера. Для того чтобы отключить обработку жеста масштабирования, необходимо добавить в тег head страницы следующий код:

<meta name="viewport" content="initial–scale=1.0, user–scalable=no, maximum–scale=1" />

Готовность API

Компоненты API Яндекс.Карт всегда загружаются асинхронно. Это происходит даже в том случае, если для подключения API используется тег <script> и никаких специальных действий для асинхронной загрузки не производилось.

Чтобы быть уверенным, что компоненты загружены и готовы к использованию, необходимо использовать функцию ready или параметр загрузки onload.

Использование функции ready()

<script src="https://api–maps.yandex.ru/2.1?apikey=<ваш API–ключ>&load=package.full&lang=ru\_RU">

</script>

<script type="text/javascript">

ymaps.ready(function () {

var myMap = new ymaps.Map("YMapsID", {

center: [55.76, 37.64],

zoom: 10

});

});

</script>

// Формирование div–контейнер карты.

<div id="YMapsID" style="width: 450px; height: 350px;"></div>

Использование параметра загрузки onload

<script src="https://https://api–maps.yandex.ru?apikey=<ваш API–ключ>&load=package.full&lang=ru\_RU&onload=init">

</script>

<script type="text/javascript">

// Сразу после загрузки API будет

// вызвана функция init.

// На момент ее исполнения div–контейнер

// карты уже будет готов.

function init (ymaps) {

var myMap = new ymaps.Map("YMapsID", {

center: [55.87, 37.66],

zoom: 10

});

}

</script>

// Формирование div–контейнер карты.

<div id="YMapsID" style="width: 450px; height: 350px;"></div>

Возникновение событий загрузки DOM–дерева или документа не сигнализирует об окончании загрузки API. То есть использование обработчиков событий типа document.ready, window.onload, jQuery.ready и пр. не позволяет определить, готовы ли компоненты для использования.

Для инициализации карты необходимо, чтобы в DOM–дереве находился элемент, в котором она размещается.

Функция ready исполняет включенный в нее код после того, как будет загружены компоненты API и DOM–дерево документа.

Функция, переданная в параметр onload вызывается после загрузки API, но не отслеживает готовность DOM–дерева. В этом случае отслеживать доступность HTML–элемента, в который помещается карта, необходимо самостоятельно. Например, при помощи обработчиков событий, перечисленных выше.

Использование параметра onload дает возможность инициализировать карту, не дожидаясь, пока DOM будет сформирован полностью. Поэтому, данный способ является самым быстрым способом загрузки API.

Карта может быть размещена в любом блочном HTML–элементе и полностью заполняет занимаемую им прямоугольную область. Вычисление размеров области производится в момент инициализации карты. Если в момент вызова конструктора карты HTML–контейнер не сформирован в DOM–дереве или его размеры не определены, будет создана карта нулевых размеров (то есть, не будет отображена).

# 6 Обоснование выбора программных и аппаратных средств для реализации поставленной задачи

Код любого, даже самого сложного сайта, веб–разработчики пишут в текстовых редакторах, используя технологии HTML, CSS и JavaScript. При разработке данного сайта использовался популярный текстовый редактор «Visual Studio Code», в котором веб–разработчику, «out of the box – прямо из коробки» доступна подсветка синтаксиса написанного кода для упомянутых выше технологий, данный текстовый редактор отличается хорошей репутацией и очень быстрой поддержкой новейших технологий, является кросплатформенным, то есть поддерживается многими операционными системами. Помимо этого, в данном текстовом редакторе, присутствует цветовая палитра, подобной той, что есть в Photoshop. Благодаря этому, веб–разработчику не нужно тратить лишнее время на поиск нужного ему цвета на веб–ресурсах или в Photoshop. Эта и многие другие встроенные функции, призваны облегчить и ускорить скорость разработки. Другим, используемым при создания проекта инструментом является browser–sync, устанавливаемый через пакетный менеджер для технологии Node.js, идущий с ним в одной поставке, который носит название npm. Node.js – среда выполнения для языка JavaScript, умеющая работать вне браузера. Node.js создан на движке V8 (разработчики – сотрудники компании Google), который был написан используя JavaScript и движок WebAssembly, написанные на C++. Browser–sync – программное обеспечение, которое позволяет отслеживать любую модификацию написанного кода, автоматически обновлять открытую в браузере страницу (запускается из терминала, запущенного в папке с проектом, в ту же директорию можно перейти, используя команду cd в терминале и введя: browser–sync start – –server – –files ”\*.html”). Это позволяет не тратить деньги на приобретение доменного имени и хоста – места на веб–сервере. Browser–sync и инструменты разработчика в браузерах позволяют моделировать подключение к сайту с использованием медленного интернет–соединения. Полезно при тестировании скорости загрузки.

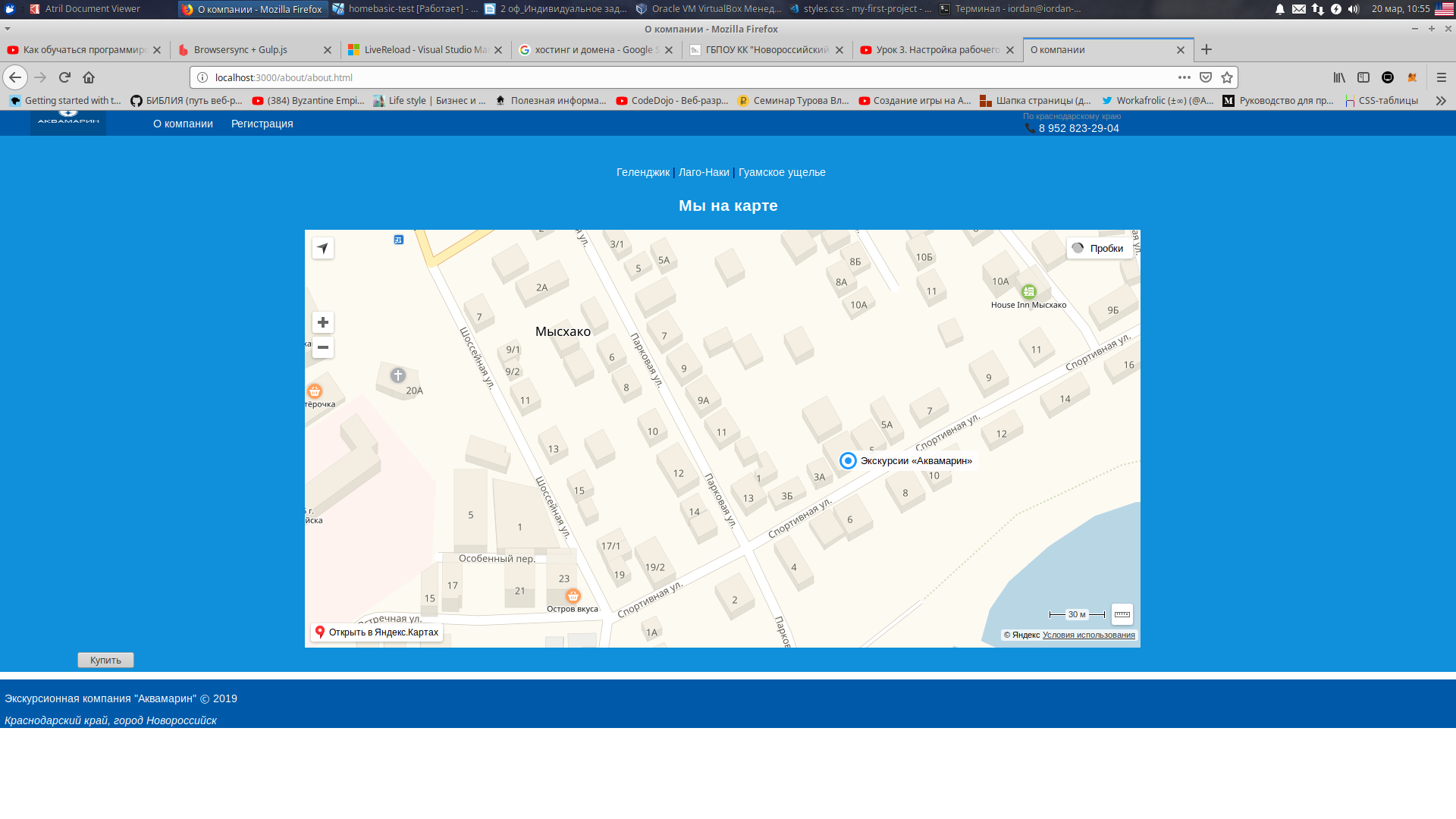
7 Описательная часть разработанного сайта

Рисунок 3 – вкладка «О компании»

Пользователь сайта, может найти веб–страницу «О компании», кликнув по одноимённой ссылке, расположенной в шапке (навигации) главной страницы. При нажатии на ссылку–номер телефона, указанный в верхней правой части страницы, данный номер автоматически подставится для набора, а пользователю останётся только нажать кнопку вызова. По центру страницы размещена Яндекс.Карта, встроенная при помощи API Яндекс.Карт.

# 8 Программный код с пояснениями

<!DOCTYPE html>

<html lang="ru–RU">

<head> <!– открытие контейнера со служебной информацией (метаданными) ––>

<meta http–equiv="X–UA–Compatible" content="IE=edge"/>

<meta charset="UTF–8"/>

<title>Аквамарин</title>

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

<meta name="Keywords" content="экскурсии,экскурсии новороссийск,экскурсии Крым,экскурсии по новороссийску,экскурсии геленджик,экскурсии Анапа,Горгиппия,

Горгиппия анапа, гора Митридат, Аквамарин"/>

<meta name="description" content="Страница экскурсионной компании Аквамарин."/>

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

</head>

HTML – не чувствительный к регистру язык разметки. Ниже, будет дано описание используемых HTML–тэгов:

<!DOCTYPE html> – задать тип документа как HTML версии 5.

<html lang=”ru\_RU”> – язык страницы (документа).

<head></head> – тэг контейнер, в котором размещается служебная информация: название документа, используемая кодировка (в данный момент рекомендуется использовать кодировку UTF–8, это может помочь устранить проблему некорректного отображения символов).

Тэг–контейнер <title></title> задаёт заголовок документа, который отображается в названии вкладки с открытым документом.

В течение некоторого времени после загрузки документа (набора страниц сайта) на веб–сервер, специальные роботы поисковых систем добавят загруженные веб–разработчиком страницы и станут видны всем пользователям в выдаче поисковой системе по соответствующему запросу. Тэг meta со значением атрибута name=”keywords” позволяет сообщить поисковому роботу ключевые слова (то, какую информацию содержит веб–страница). Тэг meta со значением атрибута name=”description” задаёт описание страницы. Использование тэга meta со значением атрибута name=”keywords” и name”description” может помочь в продвижении (поднять сайт в поисковой выдаче) разрабатываемого сайта.

Также, если необходимо предотвратить индексироавние некоторых разделов сайта поисковыми роботами, необходимо использовать текстовый файл robots.txt. Используемый синтаксис:

User–agent: \* – применить к поисковым роботам всех поисковых систем.

Disallow: / – запретить индексирования всех страниц сайта.

# 9 Методы тестирования и отладки

TDD, test–driven development или разработка через тестирование – это методология разработки ПО, которая основывается на повторении коротких циклов разработки: изначально пишется тест, покрывающий желаемое изменение, затем пишется программный код, который реализует желаемое поведение системы и позволит пройти написанный тест, а затем проводится рефакторинг написанного кода с постоянной проверкой прохождения тестов.

Тестирование ПО – это процедура, которая позволяет подтвердить или опровергнуть работоспособность кода и корректность его работы. При тестировании приложению передаются входные данные и запрашивается выполнение некой команды, после чего производится проверка полученных результатов на соответствие эталону, если результат соответствует ожидаемому – тест считается пройденным. Эта процедура может быть автоматизирована, в этом случае проверка работоспособности и правильности работы приложения осуществляется гораздо быстрее, полноценнее и чаще (в сравнении с ручным тестированием). Методика разработки через тестирование заключается в организации автоматического тестирования разрабатываемых приложений путем написания модульных, интеграционных и функциональных тестов, определяющих требования к коду непосредственно перед написанием этого самого кода. Сначала пишется тест, который проверяет корректность работы еще ненаписанного программного кода. Этот тест, разумеется, не проходит. После этого разработчик пишет код, который выполняет действия, требуемые для прохождения теста. После того, как тест успешно пройден, по необходимости осуществляется рефакторинг (доработка и переработка) написанного кода, причём рефакторинг осуществляется под контролем прохождения тестов.

Эта методология позволяет добиться создания пригодного для автоматического тестирования приложения и очень хорошего покрытия кода тестами, так как ТЗ переводится на язык автоматических тестов, то есть всё, что программа должна делать, проверяется. Также TDD часто упрощает программную реализацию: исключается избыточность реализации – если компонент проходит тест, то он считается готовым.

Архитектура программ­ных продуктов, разрабатываемых таким образом, обычно лучше (в приложениях, которые пригодны для автоматического тестирования, обычно очень хорошо распределяется ответственность между компонентами, а выполняемые сложные процедуры декомпозированы на множество простых). Стабильность работы приложения, разработанного через тестирование, выше за счёт того, что все основные функциональные возможности программы покрыты тестами и их работоспособность постоянно проверяется. Сопровождаемость проектов, где тестируется всё или практически всё, очень высока – разработчики могут не бояться вносить изменения в код, если что–то пойдёт не так, то об этом сообщат результаты автоматического тестирования.

# Заключение

Курсовая работа является важным этапом учебного процесса, способствующим началу самостоятельной, исследовательской работы каждого студента в выбранном им направлении, формировании необходимых каждому разработчику навыков: умение работать с техническим заданием, проектирование внешнего вида и логики будущей информационной системы, кодирование, тестирование, работа с документацией и её последующая разработка, анализ возможных способов решения поставленной задачи (на основании эффективности существующих алгоритмов и используемых технологий, их отладка и модификация).

Иначе говоря, во время написания курсовой работы, студент погружается в рабочую атмосферу: стадии, включающие в себя последовательные этапы разработки программного обеспечения, с которыми сталкивается каждый разработчик (это особенно актуально для разработчиков, работающих в команде). Этапы разработки программного обеспечения на профессиональном языке называют циклом разработки программного обеспечения.

# Список использованных источников

1. Макфарланд Д. «Новая большая книга CSS» – СПб.: Питер, 2016;
2. Гоше Х.Д. «HTML5. Для профессионалов» – СПб.: Питер, 2014;

Материал из сети Internet

1. Tproger – создано программистами для программистов [Электронный ресурс]. – URL: https://tproger.ru/
2. Википедия – свободная энциклопедия [Электронный ресурс]. – URL: https://ru.wikipedia.org/
3. Справочники веб–дизайнера – ценные русскоязычные ресурсы для тех, кто изучает веб–технологии и создаёт сайты: htmlbook.ru, WebReference, HTML5BOOK.RU – [Электронные ресурсы]. – URL: htmlbook.ru, webref.ru, html5book.ru
4. Бесплатный образовательный веб–сайт для изучения веб–технологий (и не только) в Интернете [Электронный ресурс]. – URL: https://www.w3schools.com/
5. Coursera – образовательный онлайн–проект, позволяющий получить доступ к урокам от самых лучших университетов мира [Электронный ресурс]. – URL: https://www.coursera.org/
6. Новейшие рекомендации и техническая спецификация по различным веб–технологиям [Электронный ресурс]. – URL: https://www.w3.org/TR/
7. Разработка адаптивных сайтов, используя технологию Flexbox (обучение в игровой форме) [Электронный ресурс]. – URL: http://flexboxfroggy.com/#ru
8. Разработка адаптивных сайтов, используя (набирающую популярность) технологию CSS Grid (обучение в игровой форме) [Электронный ресурс]. – URL: https://cssgridgarden.com/#ru
9. Хабрахабр – интересные публикации и новости из мира IT [Электронный ресурс]. – URL: https://habr.com/ru/

# Список стандартов по курсовому проектированию

В настоящем положении использованы ссылки на следующие стандарты:

ГОСТ 2.004–88 ЕСКД. Общие требования к выполнению конструкторских и технологических документов на печатающих и графических устройствах вывода ЭВМ

ГОСТ 28388–89 Системы обработки информации. Документы на магнитных носителях данных. Порядок выполнения и обращения

ГОСТ 2.104–2006 ЕСКД. Основные надписи

ГОСТ 2.105–95 ЕСКД. Общие требования к текстовым документам

ГОСТ 2.106–96 ЕСКД. Текстовые документы

ГОСТ 3.1102–81 Стадии разработки и виды документов.

ГОСТ 3.1103–82 Основные надписи

ГОСТ 3.1109–82 ЕСТД. Термины и определения основных понятий

ГОСТ 7.1–2003 Библиографическая запись. Библиографическое описание. Общие требования и правила.

ГОСТ 7.12–93 Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила.

ГОСТ 19.101–77 ЕСПД. Виды программ и программных документов

ГОСТ 19.201–78 ЕСПД. Техническое задание. Требования к содержанию и оформлению

ГОСТ 19.401–78 Текст программы. Требования к содержанию и оформлению

ГОСТ 34.601–90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания

ISO/IEC 12207:2008 «System and software engineering – Software life cycle processes» (российский аналог – ГОСТ Р ИСО/МЭК 12207–2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств).

# Приложение

Приложение 1.Техническое задание

МИНИСТЕРСТВО ОБРАЗОВАНИЯ НАУКИ И МОЛОДЕЖНОЙ ПОЛИТИКИ КРАСНОДАРСКОГО КРАЯ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ КРАСНОДАРСКОГО КРАЯ

«НОВОРОССИЙСКИЙ КОЛЛЕДЖ РАДИОЭЛЕКТРОННОГО

ПРИБОРОСТРОЕНИЯ»

(ГБПОУ КК «НКРП»)

УТВЕРЖДАЮ

Председатель ЦК

О.А. Афиногенова

\_\_.\_\_.2019

Разработка Web–сайта туристического агентства

Техническое задание на курсовой проект

Листов 3

Руководитель, преподаватель \_\_\_\_\_\_\_\_\_\_ А.А. Шепилова

Исполнитель, студент 4–П–2\_\_\_\_\_\_\_\_\_\_\_ И.Ф. Шабанов

Новороссийск, 2019

1. Введение

Наличие веб – сайта – возможность продвижения собственного бизнеса в сети Интернет (получение новых клиентов за счёт (SEO–продвижения) рекламы), обеспечение обратной связи, ответы на часто задаваемые вопросы и многое другое. Правильное использование веб–технологий разработчиком (верстальщиком) позволяет одинаково хорошо отображать сайт на устройствах, независимо от установленной операционной системы (кроссплатформенность), поддерживающих самое различное разрешение экрана: начиная от экранов мобильных телефонов, планшетных компьютеров и заканчивая самыми современными панелями сверхвысокой чёткости (UHD), поддерживающими разрешение экрана: 7680x4320 пикселей.

Современные системы управления сайтом (CMS: WordPress, MODX), позволят предпринимателю (владельцу сайта) донести до своих посетителей всю необходимую информацию. С другой стороны (со стороны разработчика), вёрстка (дизайн) веб–сайтов – востребованная, творческая IT–профессия. Изучение веб–технологий (и, в первую очередь практика) – обязательный шаг, который при должном упорстве и частоте занятий приведёт человека к успеху: возможности получения прибыли. Возможно, это мнение ошибочно, но, кажется, что порог вхождения в вёрстку ниже, чем, например, в те IT–специальности, которые требует знания математики на высоком уровне, такие как Back–end (Full Stack: компетентность в Front и Back – end), разработка всевозможных приложений (для настольных и мобильных платформ). Всё это вызывает искренний интерес к данной специальности.

1. Основание для разработки
   1. Основанием для данной работы служит задание на курсовое проектирование.
   2. Наименование работы: веб–сайт экскурсионной компании «Аквамарин».
   3. Исполнители: студент группы 4–П–2 Шабанов И.Ф., руководитель Шепилова А.А.
2. Назначение работы
3. Оказание помощи в масштабировании бизнеса.
4. Освоение необходимых компетенций для старта в профессии верстальщика.

Наличие веб–сайта должно помочь организации автоматизировать некоторые бизнес–процессы, по ведению учета, ответу на часто задаваемые вопросы, о вариантах экскурсий и, (возможно, в будущем будет реализована выдача чека), что позволит частично снизить нагрузку на сотрудников организации и при этом, что очень важно привлечь новых клиентов.

1. Технические требования
   1. Требования к функциональным характеристикам.
      1. Состав выполняемых функций:

Разрабатываемый веб–сайт должен обеспечивать:

* Хранение информации о вариантах экскурсий;
* Хранение информации о выборе заказа;
* Управление данными (добавление, удаление, редактирование);
* Быть адаптивным (отзывчивым к используемому разрешению экрана).
  + 1. Организация входных и выходных данных:

Исходная информация вносится в систему через поля ввода, затем, сохраненная в базе данных информация, отображается пользователям.

* 1. Условия эксплуатации и требования к составу и параметрам технических средств:

Информационная система имеет минимальные системные требования, равные системным требованиям используемого браузера.

* 1. Требования к транспортировке и хранению:

Программа и программная документация поставляется в электронном виде.

* 1. Специальные требования:

1. Веб–сайт должен иметь дружественный и понятный интерфейс, рассчитанный на неопытного пользователя;
2. Пользователь системы должен обладать общими навыками работы на персональном компьютере;
3. Требования к программной документации

Состав программной документации, предъявляемой на испытании:

* описание программы (ГОСТ 19.402–78);
* программа и методика испытаний (ГОСТ 19.301–79);
* текст программы (ГОСТ 19.401–78).

Основными документами, регламентирующими разработку будущих программ, должны быть документы Единой Системы Программной документации (ЕСПД): руководство пользователя, руководство администратора, описание применения.

1. Технико–экономические показатели

Эффективность системы определяется удобством использования веб-сайта для ознакомления и заказа предлагаемых компанией экскурсий.