МИНИСТЕРСТВО НАУКИ и высшего образования РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ»

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

филиал «РКТ» МАИ в г. Химки Московской области

**Специальность 09.02.03 «Программирование в компьютерных системах»**

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

**ПМ.02«Разработка и администрирование баз данных**»

**Студент**

**Группы МП-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )**

**Руководитель**

**практики от организации \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)**

**Руководитель**

**практики от филиала Шумаев А.Ю. / ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )**

**2020г.**

**ПРОГРАММА ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

По специальности 09.02.03 «Программирование в компьютерных системах»

|  |  |  |
| --- | --- | --- |
| Дата | Наименование выполняемых работ | Количество дней практики |
| 11.06.20 | Ознакомление с программой практики. Прохождение инструктажа по технике безопасности и охране труда, изучение внутреннего распорядка организации и правил работы. Разработка задания на производственную практику. | 4 |
| 12.06.20 | Описание структуры сети, в которой может функционировать разработанная база данных. | 4 |
| 13.06.20 | Описание сетевого оборудования необходимого для работы с базой данных. | 4 |
| 15.06.20 | Описание программного обеспечения необходимого для работы с базой данных по сети. | 4 |
| 16.06.20-18.06.20 | Разработка клиентских интерфейсов для клиент-серверных приложений. | 16 |
| 20.06.20 | Создание инфокоммуникационной системы.  Выполнение сетевых настроек для взаимодействия с СУБД. | 4 |
| 22.06.20-24.06.20 | Создание концептуальной модели данных.  Создание логической модели данных.  Создание физической модели данных.  Описание механизмов обеспечения целостности базы данных. | 16 |
| 25.06.20-01.07.20 | Описание СУБД представленной базы данных и ее возможностей. Внесение различных данных в базу данных.  Структурирование запросов базы данных.  Определение методов создания хранимых процедур и триггеров. Создание хранимых процедур и триггеров базы данных. | 50 |
| 02.07.20-03.07.20 | Определение способов управления правами пользователей.  Описание распределения прав пользователей и управления ими в базе данных. | 12 |
| 03.07.20-04.07.20 | Определение методов создания и синхронизации реплик базы данных. Описание существующих механизмов репликации в базе данных. Определение методов создания резервных копий базы данных. Описание существующих механизмов резервного копирования в базе данных. | 12 |
| 06.07.20 | Составление отчёта по практике | 6 |
| 07.07.20 | Подготовка к защите отчёта по практике. | 6 |
| 08.07.20 | Итоговая аттестация по производственной практике - зачёт | 6 |

Руководитель практики от филиала «РКТ» МАИ преподаватель

Шумаев А.Ю.

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

**АТТЕСТАЦИОННЫЙ ЛИСТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*ФИО*

обучающийся (аяся) на 3-ем курсе по специальности СПО 09.02.03 «Программирование в компьютерных системах»

успешно прошел(ла) производственную практику по профессиональному модулю   
**ПМ.02 «Разработка и администрирование баз данных»**

в объеме 144 часа с 11.06.2020г. по 08.07.2020г.

в организации филиала «РКТ» МАИ

**Виды и качество выполнения работ**

|  |  |  |
| --- | --- | --- |
| Виды работ, выполненных обучающимся(ейся) во время практики | Объем работ | Качество выполнения работ в соответствии с технологией и (или) требованиями организации, в которой проходила практика |
| Ознакомление с программой практики. Прохождение инструктажа по технике безопасности и охране труда, изучение внутреннего распорядка организации и правил работы. Разработка задания на производственную практику. | 4 |  |
| Описание структуры сети, в которой может функционировать разработанная база данных. | 4 |  |
| Описание сетевого оборудования необходимого для работы с базой данных. | 4 |  |
| Описание программного обеспечения необходимого для работы с базой данных по сети. | 4 |  |
| Разработка клиентских интерфейсов для клиент-серверных приложений. | 16 |  |
| Создание инфокоммуникационной системы.  Выполнение сетевых настроек для взаимодействия с СУБД. | 4 |  |
| Создание концептуальной модели данных.  Создание логической модели данных.  Создание физической модели данных.  Описание механизмов обеспечения целостности базы данных. | 16 |  |
| Описание СУБД представленной базы данных и ее возможностей. Внесение различных данных в базу данных.  Структурирование запросов базы данных.  Определение методов создания хранимых процедур и триггеров. Создание хранимых процедур и триггеров базы данных. | 50 |  |
| Определение способов управления правами пользователей.  Описание распределения прав пользователей и управления ими в базе данных. | 12 |  |
| Определение методов создания и синхронизации реплик базы данных. Описание существующих механизмов репликации в базе данных. Определение методов создания резервных копий базы данных. Описание существующих механизмов резервного копирования в базе данных. | 12 |  |
| Составление отчёта по практике | 6 |  |
| Подготовка к защите отчёта по практике. | 6 |  |
| Итоговая аттестация по производственной практике - зачёт | 6 |  |

Руководитель практики от филиала «РКТ» МАИ Шумаев А.Ю.

|  |  |  |
| --- | --- | --- |
| Дата\_\_\_\_\_\_\_\_\_\_\_\_\_ | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *подпись* | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *Расшифровка подписи* |

**ДНЕВНИК** **ПРОХОЖДЕНИЯ ПРАКТИКИ**

|  |  |  |
| --- | --- | --- |
| Дата | Наименование выполняемых работ | Подпись  руководителя |
| 11.06.20 | Ознакомление с программой практики. Прохождение инструктажа по технике безопасности и охране труда, изучение внутреннего распорядка организации и правил работы. Разработка задания на производственную практику. |  |
| 12.06.20 | Описание структуры сети, в которой может функционировать разработанная база данных. |  |
| 13.06.20 | Описание сетевого оборудования необходимого для работы с базой данных. |  |
| 15.06.20 | Описание программного обеспечения необходимого для работы с базой данных по сети. |  |
| 16.06.20-18.06.20 | Разработка клиентских интерфейсов для клиент-серверных приложений. |  |
| 20.06.20 | Создание инфокоммуникационной системы.  Выполнение сетевых настроек для взаимодействия с СУБД. |  |
| 22.06.20-24.06.20 | Создание концептуальной модели данных.  Создание логической модели данных.  Создание физической модели данных.  Описание механизмов обеспечения целостности базы данных. |  |
| 25.06.20-01.07.20 | Описание СУБД представленной базы данных и ее возможностей. Внесение различных данных в базу данных.  Структурирование запросов базы данных.  Определение методов создания хранимых процедур и триггеров. Создание хранимых процедур и триггеров базы данных. |  |
| 02.07.20-03.07.20 | Определение способов управления правами пользователей.  Описание распределения прав пользователей и управления ими в базе данных. |  |
| 03.07.20-04.07.20 | Определение методов создания и синхронизации реплик базы данных. Описание существующих механизмов репликации в базе данных. Определение методов создания резервных копий базы данных. Описание существующих механизмов резервного копирования в базе данных. |  |
| 06.07.20 | Составление отчёта по практике |  |
| 07.07.20 | Подготовка к защите отчёта по практике. |  |
| 08.07.20 | Итоговая аттестация по производственной практике - зачёт |  |

СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

# ВВЕДЕНИЕ

Современный период развития цивилизованного общества по праву называют этапом информатизации. Информатизация общества предполагает всестороннее и массовое внедрение методов и средств сбора, обработки, передачи и хранения. Оператор электронно-вычислительных и вычислительных машин – профессия настоящего и будущего, так как в современном обществе является одной из самых востребованных профессий.

Практика – одна из важнейших составляющих профессиональной подготовки студента. Она является составной частью основной образовательной программы среднего профессионального образования.

Цель учебной практики — углубить и закрепить научно-теоретические знания по выбранной специальности, научиться применять их на практике, а также выработать навыки практической работы.

Объект – профессиональная деятельность оператора электронно-вычислительных и вычислительных машин.

Цель – применение на практике знаний, полученных в ходе изучения общепрофессиональных и специальных дисциплин, отработка практических профессиональных умений и навыков по профессии.

В процессе прохождения практики были освоены и использованы:

* языки программирования: HTML, CSS, JavaScript;
* метаязыки: SASS;
* препроцессоры: LESS;
* программные продукты: Adobe Dreamweaver, Visual Studio Code;
* дополнительное ПО:Denwer;
* утилиты: PHPMyAdmin;
* фреймворки: Node.js, Express, Handlerbars, Bootstap 4;
* веб-плагины: Colorbox, Font Awesome, Greensock, Isotope, Processbar, Scroll Magic, EasIng, Circle Progress, Owl-Carousel;
* библиотеки скриптов: JQuery, MySQL 2.0.

СОДЕРЖАНИЕ

1. Теоретические основы проекта
   1. Git

Git — распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года.

Система спроектирована как набор программ, специально разработанных с учётом их использования в сценариях. Это позволяет удобно создавать специализированные системы контроля версий на базе Git или пользовательские интерфейсы. Например, Cogito является именно таким примером оболочки к репозиториям Git, а StGit использует Git для управления коллекцией исправлений (патчей).

Репозиторий Git представляет собой каталог файловой системы, в котором находятся файлы конфигурации репозитория, файлы журналов, хранящие операции, выполняемые над репозиторием, индекс, описывающий расположение файлов, и хранилище, содержащее собственно файлы. Структура хранилища файлов не отражает реальную структуру хранящегося в репозитории файлового дерева, она ориентирована на повышение скорости выполнения операций с репозиторием. Когда ядро обрабатывает команду изменения (неважно, при локальных изменениях или при получении патча от другого узла), оно создаёт в хранилище новые файлы, соответствующие новым состояниям изменённых файлов. Существенно, что никакие операции не изменяют содержимого уже существующих в хранилище файлов. Любое файловое дерево в системе можно превратить в репозиторий git, отдав команду создания репозитория из корневого каталога этого дерева (или указав корневой каталог в параметрах программы). Репозиторий может быть импортирован с другого узла, доступного по сети. При импорте нового репозитория автоматически создаётся рабочая копия, соответствующая последнему зафиксированному состоянию импортируемого репозитория (то есть не копируются изменения в рабочей копии исходного узла, для которых на том узле не была выполнена команда commit).

Репозиторий Git бывает локальный и удалённый. Локальный репозиторий — это подкаталог git, создаётся (в пустом виде) командой git init и (в непустом виде с немедленным копированием содержимого родительского удалённого репозитория и простановкой ссылки на родителя) командой git clone.

Практически все обычные операции с системой контроля версий, такие, как коммит и слияние, производятся только с локальным репозиторием. Удалённый репозиторий можно только синхронизировать с локальным как «вверх» (push), так и «вниз» (pull).

Наличие полностью всего репозитория проекта локально у каждого разработчика даёт Git ряд преимуществ перед SVN. Так, например, все операции, кроме push и pull, можно осуществлять без наличия интернет-соединения.

Команда push передаёт все новые данные (те, которых ещё нет в удалённом репозитории) из локального репозитория в репозиторий удалённый. Для исполнения этой команды необходимо, чтобы удалённый репозиторий не имел новых коммитов в себя от других клиентов, иначе push завершается ошибкой, и придётся делать pull и слияние.

Команда pull — обратна команде push. В случае, если одна и та же ветвь имеет независимую историю в локальной и в удалённой копии, pull немедленно переходит к слиянию.

Имя ветви по умолчанию: master. Имя удалённого репозитория по умолчанию, создаваемое git clone во время типичной операции «взять имеющийся проект с сервера себе на машину»: origin.

Таким образом, в локальном репозитории всегда есть ветвь master, которая есть последний локальный коммит, и ветвь origin/master, которая есть последнее состояние удалённого репозитория на момент завершения исполнения последней команды pull или push.

Команда fetch (частичный pull) — берёт с удалённого сервера все изменения в origin/master, и переписывает их в локальный репозиторий, продвигая метку origin/master.

* 1. Cit Hup

[GitHub](https://github.com/) — сервис онлайн-хостинга репозиториев, обладающий всеми функциями распределённого контроля версий и функциональностью управления исходным кодом — всё, что поддерживает Git и даже больше. Обычно он используется вместе с Git и даёт разработчикам возможность сохранять их код онлайн, а затем взаимодействовать с другими разработчиками в разных проектах.

Также GitHub может похвастаться контролем доступа, багтрекингом, управлением задачами и вики для каждого проекта. Цель GitHub — содействовать взаимодействию разработчиков.

К проекту, загруженному на GitHub, можно получить доступ с помощью интерфейса командной строки Git и Git-команд. Также есть и другие функции, такие как документация, запросы на принятие изменений (pull requests), история коммитов, интеграция со множеством популярных сервисов, email-уведомления, эмодзи, графики, вложенные списки задач, система @упоминаний, похожая на ту, что в Twitter, и т.д.

Git — это инструмент, позволяющий реализовать распределённую систему контроля версий, а GitHub — это сервис для проектов, использующих Git.

[GitHub](https://github.com/) также является одним из [крупнейших онлайн-хранилищ](https://octoverse.github.com/) (англоязычной) совместной работы по всему миру.

Репозиторий

Репозиторий или хранилище — это каталог, в котором хранятся файлы проекта. Он может быть расположен в хранилище GitHub или в локальном хранилище на компьютера. Можно хранить файлы кодов, изображения, аудио или всё, что связано с проектом, в хранилище.

Система “Звезд”

Одной из отличительных особенностей GitHub является система звёзд. Чтобы выразить интерес к репозиторию, его нужно отметить звездой. Это можно сделать с помощью кнопки «Star». Что позволяет отслеживать интересные проекты и находить похожие. Это также один из самых важных рейтинговых механизмов, поскольку чем больше звезд имеет репозиторий, тем он популярнее. Поэтому в результатах поиска он будет в топовых местах. Крупные проекты могут иметь десятки тысяч звезд.

Также в GitHub есть трендовая страница, на которой представлены репозитории, которые получают наибольшее количество звезд за определенный период времени (за день, неделю, месяц).

Fork - это копия репозитория. Это то же самое, что branch в Git.

Если есть возможность внести вклад в уже существующие проекты, в которые нет прав на внесения изменений путем отправки (push) изменений, можно создать собственное ответвление (“fork”) проекта. Это означает, что GitHub создаст собственную копию проекта, данная копия будет находиться в репозитории и можно легко делать изменения путем отправки (push) изменений. Также другой человек может разветвить репозиторий, внести некоторые изменения, а затем создать запрос на внесение этих изменений.

Pull Request

Человек может создать свое собственное ответвление (“fork”) проекта, внести некоторые изменения и затем сделать Pull Request., чтобы замерджили эти изменения.

Как только сделано Pull Request, рассматривается основными разработчиками проекта. В зависимости от количества и сложности изменений, которое внесено в код, разработчику может потребоваться разное количество времени, чтобы убедиться, что изменения совместимы с проектом.

У проекта может быть четкий график изменений, которые разработчики хотят внедрить. Тогда запросы на внесение изменений в код будут рассмотрены быстро. Но запрос может и не быть рассмотрен вовсе.

Управление проектами

Кроме обратной связи и новых знакомств, GitHub также предоставляет некоторые функции по управлению проектами.

Одна из таких функций - Projects. Нововведение в GitHub. Она основывается на методологии Канбан, которая помогает эффективно организовать работу.

Wiki предназначен для использования в качестве документации для пользователей. Одним из самых впечатляющих видов использования Wiki, является язык программирования Go GitHub Wiki.

Еще один популярный инструмент для управления проектами -вехи. Это часть страницы "issues ". Используя эту функцию можно назначать контрольные точки проекта.

Тэги в Git - это указатель на конкретный коммит, и если он выполняется последовательно, он помогает вернуться к предыдущей версии вашего кода, не ссылаясь на конкретные коммиты.

Сравнение коммитов

GitHub предлагает множество инструментов для работы с вашим кодом.

Одна из самых важных вещей, которые нужно сделать - сравнить одну ветку с другой. Или сравнить последний коммит с используемой версией, чтобы увидеть, какие изменения были внесены.

Ветка

Ветка это копия репозитория. Можно использовать ветку, когда есть возможность сделать разработку изолированно.

Работа с веткой не повлияет на центральное хранилище или другие ветки. Если сделана эта работа, то есть возможность объединить собственную ветку с другими ветками и центральным репозиторием, используя запрос на извлечение.

Запрос на извлечение

Запрос на извлечение означает, что сообщаете другим, что вы передали изменения, внесённые вами в ветке, в главный репозиторий. Соавторы хранилища могут принять или отклонить запрос на извлечение. После его открытия обсуждается и анализируется работа с соавторами.

Инструкции по созданию запросов на извлечение в GitHub:

* Перейти в хранилище и найти ветку меню
* В меню выбрать ветку, которая содержит нужный коммит;
* Нажать кнопку «Новый запрос» на извлечение рядом с меню ветки;
* Вставить заголовок и описать запрос;
* Нажать кнопку «Создать запрос» на извлечение

Форкинг репозитория

Форкинг репозитория означает, что создаётся новый проект на основе существующего репозитория, разветвление репозитория означает, что копируется существующий репозиторий, нужно внести некоторые необходимые изменения, сохранить новую версию в качестве нового репозитория и называть собственным проектом.

Это отличная функция, которая ускоряет разработку проекта. Поскольку это совершенно новый проект, центральное хранилище не будет затронуто. Если «главный» репозиторий обновлён, то также можно применить это обновление к текущему форку.

Инструкции по созданию репозиторий в GitHub:

* Найти репозиторий, который необходимо разветвлять
* Найти кнопку Форк

GitHub не ограничен только для разработчиков. GitHub это отличная платформа, которая меняет работу разработчиков. Тем не менее, каждый, может эффективно управлять своим проектом и работать совместно.

* 1. Языки программирования

Язык программирования — формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель под её управлением.

### Язык гипертекстовой разметки HTML

HTML -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.

Язык HTML до 5-й версии определялся как приложение SGML (стандартного обобщённого языка разметки по стандарту ISO 8879). Спецификации HTML5 формулируются в терминах DOM (объектной модели документа).

Язык XHTML является более строгим вариантом HTML, он следует синтаксису XML и является приложением языка XML в области разметки гипертекста.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием шифрования.

Текстовые документы, содержащие разметку на языке HTML (такие документы традиционно имеют расширение .html или .htm), обрабатываются специальными приложениями, которые отображают документ в его форматированном виде. Такие приложения, называемые «браузерами» или «интернет-обозревателями», обычно предоставляют пользователю удобный интерфейс для запроса веб-страниц, их просмотра (и вывода на иные внешние устройства) и, при необходимости, отправки введённых пользователем данных на сервер.

### Язык разметки CSS

**CSS (Cascading Style Sheets)** — язык таблиц стилей, который позволяет прикреплять стиль (например, шрифты и цвет) к структурированным документам (например, документам HTML и приложениям XML). Обычно CSS-стили используются для создания и изменения стиля элементов веб-страниц и пользовательских интерфейсов, написанных на языках HTML и XHTML, но также могут быть применены к любому виду XML-документа, в том числе XML, SVG и XUL. Отделяя стиль представления документов от содержимого документов, CSS упрощает создание веб-страниц и обслуживание сайтов.

CSS поддерживает таблицы стилей для конкретных носителей, поэтому авторы могут адаптировать представление своих документов к визуальным браузерам, слуховым устройствам, принтерам, брайлевским устройствам, карманным устройствам и т.д.

Применение CSS к документам HTML основано на принципах наследования и каскадирования. Принцип наследования заключается в том, что свойства CSS объявленные для элементов-предков, почти всегда, наследуются элементами-потомками.

Принцип каскадирования применяется в случае, когда какому-то элементу HTML одновременно поставлено в соответствие более одного правила CSS, то есть, когда происходит конфликт значений этих правил. Чтобы разрешить такие конфликты, вводятся правила приоритета.

* Наиболее низким приоритетом обладает стиль браузера;
* Следующим по значимости является стиль, заданный пользователем браузера в его настройках;
* Наиболее высоким приоритетом обладает стиль, заданный непосредственно автором страницы.

### Язык программирования JavaScript

JavaScript – это интерпретируемый язык программирования, разработанный для взаимодействия с веб-страницами. JavaScript представляет собой реализацию ECMAScript. ECMAScript – это просто описание языка, который определён в стандарте ECMA-262.

В браузерах по умолчанию встроено специальное программное обеспечение, называемое интерпретатором JavaScript, это сделано для того, чтобы браузер мог выполнять написанный на языке JavaScript код. Как правило, JavaScript называют клиентским языком, подчеркивая тем самым, что сценарий исполняется на клиентском компьютере в браузере, а не на веб-сервере.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса .

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке [4].

JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам — функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкани — что придаёт языку дополнительную гибкость

Несмотря на схожий с Си синтаксис, JavaScript по сравнению с языком Си имеет коренные отличия:

* объекты с возможностью интроспекции;
* функции как объекты первого класса;
* автоматическое приведение типов;
* автоматическая сборка мусора;
* анонимные функции [4].

В JavaScript доступ к отладчикам становится особенно полезным при разработке крупных нетривиальных программ из-за различий в реализациях разных браузеров (в частности, в отношении объектной модели документа). Полезно иметь доступ к отладчику для каждого из браузеров, в которых будет работать веб-приложение [4].

По состоянию на ноябрь 2009 года, Internet Explorer, Opera, Firefox, Safari, и Google Chrome имеют отладчики сценариев.

Internet Explorer имеет три отладчика: Microsoft Visual Studio — самый полный из них, за ним следует Microsoft Script Editor и наконец, свободный Microsoft Script Debugger, гораздо более простой, чем два других. Бесплатный Microsoft Visual Web Developer Express предоставляет ограниченную версию с отладочной функцией JavaScript в Microsoft Visual Studio.

JavaScript может быть подключен или внедрен в веб-документ тремя способами:

* Расположение внутри страницы. Можно использовать теги <script></script>которые рекомендуется, но не обязательно, помещать внутри контейнера <head>. Контейнеров <script> в одном документе может быть сколько угодно;
* Расположение внутри тега. Спецификация HTML описывает набор атрибутов, используемых для задания обработчиков событий;
* Вынесение в отдельный файл. Подключение JavaScript — написать скрипт в отдельном файле, а потом подключить его с помощью конструкции <script type="application/javascript" src="http://Путь\_к\_файлу\_со\_скриптом"></script> [4].

### Язык программирования PHP

PHP - скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

В области веб-программирования, в частности серверной части, PHP — один из популярных сценарных языков.

Перечень преимуществ, которые делают широко применимым его в веб-разработке:

* разработка с помощью PHP дает много возможностей. При должном уровне владения, с помощью шаблонизатора можно создавать не только сценарии для веб-приложений, но и полноценные программы. Существуют решения, позволяющие создавать мобильные приложения на PHP;
* изучение PHP не требует много времени. Это одновременно и плюс, и минус. Ведь основательное знание требует практики, но об этом позже;
* кроссплатформенность. PHP может быть запущен в любой операционной системе, включая юниксоиды;
* поддержка веб-серверов. Сложно найти тот, который бы не работал с PHP;
* Присоединить его можно четырьмя способами:
* include;
* require;
* include\_once;
* require\_once.

Подключаем его в файле index.php:

<?php

for($i=0;$i<2;$i++){

include ("Variable.php");

echo 'Значение переменной - '.$count.' ';

$count++;

}

Последующие способы идентичны синтаксису, необходимо только менять «include» на последующие варианты.

* 1. Вспомогательные технологии разработки
     1. Библиотеки скриптов

jQuery – это кросс-платформенная библиотека функций синтаксиса JavaScript взаимодействующая с HTML для упрощения процесса создания клиентских решений. Была разработана Джоном Резигом в 2006 году.

В способности реализацией библиотекой входят такие возможности как:

* Обход документа HTML – отображение браузером веб-страницы известно, как объектная модель документа (Document Object Model). Эту модель можно концептуально отобразить в виде древа с элементами. Для поиска и внесения необходимых корректировок элементов, библиотека позволяет запрашивает эту модель, и редактировать ее с помощью документированного API;
* Манипуляция документами HTML – функции, отвечающие за видоизменение документа, определить высоту или ширину документа, окна; получить значения из элементов; переключать имена классов и так далее;
* Обработка событий – реагирование веб-страницы на происходящие события со стороны пользователя (например, при нажатии клавишей на клавиатуре или мышью по элементу), что позволяет использовать преимущества взаимодействия пользователя с браузером;
* Анимация – библиотека jQuery содержит эффекты для создания анимации элементам, имеющие обратную связь с пользователями;
* Ajax – функция jQuery.ajax() осуществляет вызов на сервер, обрабатывает ответ с обновлением части страницы. Технология обеспечивает аккуратные функциональные возможности в контексте страницы или веб-приложения при правильном использовании.

Помимо этого, jQuery способна выполнять другие мощные операции в кроссплатформенном режиме. Например, jQuery UI – является дополнением для основной библиотеки, позволяющая создавать виджеты на веб-сайтах, или jQuery Mobile – библиотека для создания веб-сайтов для смартфонов, планшетов и других настольных устройств.

MySQL – это система управления реляционными базами данных, с открытым исходным кодом, созданная компанией MySQL AB в 1995 году.

База данных представляет собой структурированную совокупность данных, которые могут представлять из себя списки, перечни документов и другого большого количества информации. В реляционных базах данных содержимое хранится не в одном скопе, а в отдельных таблицах. Таблицы связываются между собой с помощью отношений, что позволяет создавать запросы, состоящие из нескольких таблиц. Программное обеспечение MySQL является системой клиент-сервер, содержащая в себе многопоточный SQL-сервер, обеспечивающий поддержку вычислительных машин баз данных, клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов.

Основные характеристики MySQL:

* Разработан на языке программирования С++;
* Кроссплатформенный;
* Многопоточный за счет использования потоков ядра процессора;
* Быстрая работа с таблицами;
* Быстрое распределение памяти;
* Быстрые соединения между таблицами (с использованием метода one-sweep multi-join);
* Быстрые SQL-функции, реализованные с помощью оптимизированных библиотек классов;
* Доступность использования хеш-таблиц в памяти компьютер, в качестве запасных таблиц.
  + 1. Препроцессоры

Less — это препроцессор CSS, позволяющий использовать переменные, функции, циклы и другие технологии для упрощения работы со стилями. Препроцессор в данном случае означает, что мы имеем дело с динамическим языком стилей, который преобразуется в CSS. Таким образом, на выходе получаем стандартный стилевой файл.

Less преследует благородные цели — облегчить работу разработчикам сайтов дав им подходы и приёмы, которые в CSS на данный момент отсутствуют, но характерны для языков программирования. Например, те же переменные позволяют задать какое-либо значение, а затем использовать его многократно, подставляя лишь имя переменной.

Разработал Less Алексис Селье на языке Ruby, а затем под влиянием нарастающей популярности Node.js переписал код на JavaScript. Характерной особенностью Less стал синтаксис основанный на CSS. Это сразу же снизило порог вхождения и изучения новой технологии, к тому же не возникает никаких проблем с редакторами кода, они прекрасно поддерживают подсветку синтаксиса. В дальнейшем Less оказал своё влияние на препроцессор Sass, в котором появился новый похожий синтаксис. Но в целом, вопрос, кто на кого оказал влияние открыт, потому что многие полезные вещи были заимствованы друг у друга. Так что в настоящий момент и Less и Sass похожи как братья, различаясь лишь в деталях.

* + 1. Метаязыки: SASS

SASS расшифровывается как Syntactically Awesome Style Sheets – если переводить дословно, то это звучит как: «Синтаксически потрясающие таблицы стилей».

Эта технология была придумана и воплощена Хэмптоном Катлином (Hampton Catlin). SASS манипулирует CSS-правилами, используя переменные, так называемые миксины (mixins), наследование и вложенность.

Исходные и скомпилированные файлы имеют расширения .sass и .scss, соответственно. Исходники переводятся в хорошо отформатированный CSS-код с помощью командной строки или веб-плагина.

SASS упрощает написание CSS-кода и позволяет динамически им манипулировать. Это отличный способ создания более функциональных CSS-кодов, который позволяет ускорить выполнение ежедневной работы веб-разработчиков и дизайнеров.

Особенности SASS:

1 Вложенность

Одна из главных особенностей SASS – вложенность. Она облегчает написание написание стилей. К примеру данный код:

#container p {

font-family: Arial;

font-size: 13px;

}

#container h1 {

font-family: Tahoma;

font-size: 15px;

}

#container h2 {

font-family: Helvetica;

font-size: 14px;

}

Для SASS будет выглядеть иначе:

$myFontsize1: 13px;

$myFontsize2: 18px;

$myFontsize3: 25px;

$myWidth: 500px;

$myMargin: 0px auto;

#container {

width: $myWidth;

margin: $myMargin;

p {

font-family: Arial;

font-size: $myFontsize1;

}

h1 {

font-family: Tahoma;

font-size: $myFontsize3;

}

h2 {

font-family: Helvetica;

font-size: $myFontsize2;

}

}

В данном коде все стили элементов расположены под идентификатором container, вместо того, чтобы предварять этим id каждый из них.

2.1.2 Переменные

Переменные SASS объявляются с предваряющим их название символом $ и записываются аналогично CSS-свойствам. С помощью SASS, вы можете определять переменные для таких стилей, как font size, margin, padding и так далее.

Использование переменных дает вам возможность повторного использования заданных ранее значений.

В SASS существует шесть разных типов переменных: Строковые, Цветовые, Логические, Списковые, Тип null.

2. Миксины.

Миксины позволяют определять общие свойства, а затем использовать их повторно, то есть это реализация наследования. Миксины определяются с помощью директивы @mixin и включают в себя блок кода, который затем можно использовать с помощью директивы @include. Внизу приведен пример использования миксина:

@mixin border {

border: 1px solid red;

}

#container {

width: 960px;

margin: 0 auto;

@include border;

}

Директива @mixin была использована для создания стиля свойства border, затем эта же директива была включена в стиль идентификатора container с помощью директивы @include.

* + 1. Фреймворки

Node.js — это кроссплатформенная среда с открытым исходным кодом для разработки серверных и сетевых приложений. Приложения написаны на JavaScript и могут выполняться в среде исполнения Node.js на ОС X, Microsoft Windows и Linux. Node.js использует управляемую событиями, неблокирующую модель ввода-вывода, которая делает его простым и эффективным, идеальным для приложений с интенсивным использованием данных в реальном времени, работающих через распределенные устройства.

Самое главное, что нужно понимать, это то, что Node.js — не сервер. Нет в нем конфигурационных файлов, как, например, в Apach, и сам по себе Node ничего не делает. Это среда выполнения JavaScript кода — еще один способ выполнить его на компьютере. И если нам нужно, чтобы эта платформа была HTTP-сервером, то мы должны написать его сами.

Приложения Node.js написаны на JavaScript и могут выполняться в среде исполнения на ОС X, Windows и Linux. Node.js отлично подходит для масштабируемых высоконагруженных проектов. С его помощью также можно создавать синхронные кроссплатформенные решения — например, когда отправляем сообщение с телефона, оно отображается и в веб-версии приложения, и в десктопной. И чаще всего Node.js используется для создания веб-серверов.

Использование Node.js в Full-stack разработке (разработке полного цикла) позволяет писать и front-end — пользовательскую часть приложения, и back-end — серверную часть приложения на одном и том же языке программирования JavaScript. Это довольно удобно для связывания частей приложения. Также это позволяет front-end разработчикам достаточно просто освоить и back-end разработку.

Преимущества NodeJS

Node.js — это среда выполнения JavaScript на стороне сервера, которая используется для построения быстрых, масштабируемых сетевых приложений. Основные преимущества использования:

* + - * удобство для построения быстрых приложений, поскольку Node способен обрабатывать огромное количество одновременных соединений с высокой пропускной способностью. Это привлекает множество программистов, которые хотят создавать сайты с поддержкой миллионов посетителей;
      * эффективность. В веб-приложении больше всего времени требуется, чтобы выполнить все ваши запросы к базе данных. С помощью Node.js вы можете выполнить все их сразу, уменьшая время отклика для медленных запросов;
      * язык разработки JavaScript. Вы можете разделить код между браузером и вашим back-end. JavaScript на пути к тому, чтобы стать действительно универсальным языком;
      * скорость выполнения. Движок V8 постоянно расширяет границы и является одним из самых быстрых интерпретаторов динамических языков на планете. Кроме того, средства ввода/вывода Node имеют действительно легкий вес, в результате чего вы становитесь настолько близки к полному использованию потенциала вашей системы ввода/вывода, насколько это возможно.

Express.js

Express.js — это самый простой и быстрый фреймворк Node.js, который используется в качестве промежуточного обработчика для управления серверами и маршрутами.

Express.js подходит для разработки простых приложений, которые могут обрабатывать несколько запросов одновременно и опираются на возможности технологии Express.

Особенности Express.js:

* + - * Полностью настраиваемый;
      * Низкая кривая обучаемости;
      * Ориентирован на браузер.

Handlebars

Handlebars - это шаблонный процессор, который динамически генерирует HTML-страницу, что экономит время на ручном обновлении.

Handlebars генерирует HTML, используя структуру JSON и запуская ее через шаблон. Эти шаблоны написаны в основном в обычном HTML и набиты заполнителями, которые позволяют вам при необходимости вводить данные.

Handlebars дает возможность создать собственный пользовательский помощник. Просто зарегистрируем свою функцию в Handlebars, и любой шаблон, который затем компилируем, может получить доступ к помощнику. Есть два типа помощников, которые можно сделать:

* + - * Функциональные помощники - это в основном обычные функции, которые после регистрации могут быть вызваны в любом месте шаблона. Handlebars записывает возвращаемое значение функции в шаблон.
      * Блок-помощники похожи по своему характеру на помощников if, each и т.д. Они позволяют изменить контекст того, что внутри.

Bootstrap 4

Bootstrap — это открытый и бесплатный HTML, CSS и JS фреймворк, который используется веб-разработчиками для быстрой вёрстки адаптивных дизайнов сайтов и веб-приложений.

Фреймворк Bootstrap используется по всему миру не только независимыми разработчиками, но иногда и целыми компаниями. На Bootstrap создано очень много различных сайтов.

Основная область его применения – это разработка фронтенда сайтов и интерфейсов админок. Среди аналогичных систем (Foundation, UIkit, Semantic UI, InK и др.) фреймворк Bootstrap является самым популярным.

Фреймворк Bootstrap представляет собой набор CSS и JavaScript файлов. Чтобы его использовать эти файлы необходимо просто подключить к странице. После подключения станут доступны инструменты данного фреймворка: колоночная система (сетка Bootstrap), классы и компоненты.

Bootstrap состоит из:

* + - * сетки;
      * классов для стилизации текста, изображений, таблиц и другого контента;
      * компонентов предназначенных для создания кнопок, различных форм на странице, горизонтальных и вертикальных навигационных меню, слайдеров, выпадающих списков, аккордеонов, модальных окон, всплывающих подсказок и других элементов интерфейса;
      * классов для решения вспомогательных задач наиболее часто возникающими перед веб-разработчиками (выравнивание текста, скрытие или отображение элемента, задания цвета и фона элементу, задание margin и padding отступов, и др.).
    1. Веб-плагины

ColorBox

ColorBox — новый лайтбокс плагин для jQuery

**Итак, что же позволяет ColorBox:**

* Поддерживает показ фото, группы фото, слайд-шоу, AJAX, инлайн и iframe содержание
* Вес: только 9KB  JavaScript
* Внешний вид полностью контролируется с помощью CSS так что пользователи могут изменять его на свой вкус
* Настройки отображения можно можно изменять без изменения ColorBox JavaScript файла
* Не требует никаких изменений в существующим HTML
* Предзагрузка фоновых изображений, чтобы избежать «бликов» при первом использовании
* Позволяет разделить фото на группы
* Написанн как плагин для jQuery и может быть связан с другими jQuery командами
* Генерирует W3C валидный XHTML и CSS, JS не добавляет глобальные переменные

# Font Awesome

Font Awesome – это шрифт, созданный с помощью значков (иконок), но это ни в коем случае не набор картинок. Почему это различие так важно? Поскольку шрифт Font Awesome:

Векторный – это означает, что он отлично смотрится независимо от того, в каком размере он отображается.

Настраиваемый – вы можете манипулировать им, как и любым другим шрифтом. Изменяйте цвет, добавляйте анимацию и многое другое!

Кросс-браузерная совместимость – иконочный шрифт должен работать практически в любом браузере.

GreenSock

GreenSock-позволяет вам анимировать все, что вы можете получить с помощью JavaScript, включая DOM, Canvas и CSS, а также свои собственные пользовательские объекты. GSAP также помогает разрешить несогласованности браузера для более легкого тестирования пользователей, позволяет использовать объекты для управления сложной анимацией и работает до 20 раз быстрее, чем jQuery. Он стал стандартом в отрасли и использовался на бесчисленных удостоенных наград веб-сайтах.

Isotope

Isotope— это прекрасное дополнение для адаптивной верстки. С помощью этого бесплатного jQuery плагина, можно при заставить блоки динамически перемещаться по странице, при изменении разрешения экрана (или размера страницы). Также можно добавить плавную CSS3 анимацию.

Progress bar

Progress bar– это элемент графического интерфейса пользователя, представляющий собой прямоугольную (в большинстве случаев) панель для отображения индексации ходи или выполнения какой либо задачи, например загрузки файла по сети. Обычно бегунок индикатора процесса заполняется слева направо. Часто встречается текстовое сопровождение выполненных процентов. Такие элементы используются в том случае, если масштаб задачи известен, либо ход выполнения задачи требует отображения информации.

ScrollMagic

ScrollMagic-Создание анимации при прокрутке страницы

ScrollMagic поможет вам легко реагировать на текущую позицию прокрутки (скролла). Это идеальная библиотека для вас, если вы хотите, чтобы …

* анимацию на основе положения прокрутки — либо вызвать анимацию или синхронизировать ее с движением скроллбара.
* прикрепить (pin) элемент, начиная с определенной позиции скролла — на неопределенный или лимитированный промежуток скролла (липкие элементы)
* переключать CSS классы элементов на основе позиции скролла.
* легко добавлять параллакс эффекты на ваш сайт.
* создать бесконечно прокручиваемую страницу (ajax загрузка дополнительного контента).
* добавить callbacks на специфичную скролл позицию или при прокрутки через специфичную секцию, передавая параметр прогресса.

EASING

EASING-один из основных инструментов анимации WEB-страниц. С его помощью можно анимировать все: текст, блоки и другие элементы.

Для всего этого есть куча разнообразных эффектов и их параметров, одним из которых есть "easing" - скорость анимации. Про него и поговорим.

Параметр easing определяет динамику выполнения анимации.

От этого параметра зависит то, как будет происходить анимация. Например ускорятся к завершению, или создать эффект "скачка"

Circle progress

Circle progress -Универсальный плагин jQuery для создания красивых прогресс-баров. При загрузке странице инструмент запускает плавную анимацию для круговых прогресс-баров, "заполняя" их соответственно с параметрами в настройках плагина.

Owl Carousel

Owl Carousel— это слайдер-карусель написанный на jQuery. Данный плагин предлагает нам очень крутые фишки:

* очень гибкая кастомизация ( настройка ) слайдера ( более 60 опций)
* полностью адаптивен
* поддерживает
* совместимость со всеми браузерами
* поддерживает прикосновений и перетаскиваний на мобильных устройствах
  1. Среды разработки
     1. Программные продукты

Программа Adobe Dreamweaver, по праву, занимает лидирующее положение в широкой линейке программ для разработки и создания веб-сайтов. Ядро программы Dreamweaver состоит из мощного HTML-редактора, но разработку веб-документов возможно осуществлять не только в режиме ручного кодирования.

Dreamweaver является и полноценным визуальным редактором, что позволяет работать в режиме визуального представления документа. На рисунке 1 представлен вид окна в режиме “Разделение”.

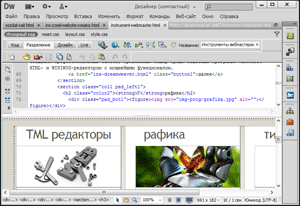


Рисунок 1 – окно “Разделение”.

В программе Dreamweaver заложены средства автоматизации разработки и управления веб-сайтом. В составе визуального редактора имеются множество полезных инструментов, упрощающих и ускоряющих разработку: загрузка сайта на сервер, готовые макеты сайтов, всевозможных вариантов меню и других объектов веб-страниц.

Dreamweaver гибок в настройке, позволяет создать индивидуальную рабочую среду, удобную для конкретного вебмастера. Кроме того, Dreamweaver позволяет создавать шаблоны с редактируемыми и неизменяемыми областями, одинаковыми на всем сайте или в одном из разделов сайта. Этот механизм существенно облегчает процесс обновления и модернизации всего сайта целиком.

Функционал Adobe Dreamweaver: резиновые макеты - инструмент, позволяющий создавать сложные веб-сайты, меняющие отображение в зависимости от размера экранов компьютеров и мобильных устройств. Переходы CSS3 - позволяют создавать динамические веб-страницы с анимацией и привлекательными эффектами.

Многоэкранный просмотр - инструмент, с помощью которого можно проверить отображения для смартфонов, планшетников и компьютеров с различным разрешением экранов. Создание приложений для мобильных устройств. Ускоренная многопоточная передача данных по FTP - актуально для приличных по размеру сайтов. Режим интерактивного просмотра Live - отображение веб-страницы в самой программе, как в браузере. Окно программы в режиме Live представлен на рисунке 2.

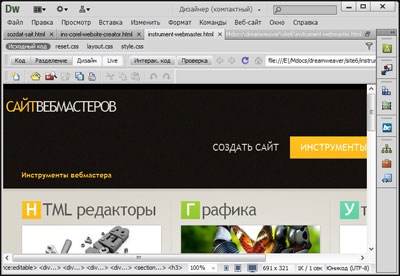


Рисунок 2 - Окно программы в режиме “Live”.

Особенности Dreamweaver - полная поддержка HTML5 и CSS3 с возможностью проверки кода веб-документа на ошибки. Режим работы сервиса BrowserLab - позволяет проверить динамические веб-страницы, находящиеся на локальном компьютере, на возможную разницу отображения в разных браузерах.

Поддержка интеграции с CMS WordPress, Joomla и Drupal и возможность проверки результата в режиме просмотра Live.

Visual Studio Code — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодомно готовые сборки распространяются под проприетарной лицензией.

Visual Studio Code основан на Electron и реализуется через веб-редактор Monaco, разработанный для Visual Studio Online.

Так же поддерживает ряд языков программирования, подсветку синтаксиса, IntelliSense, рефакторинг, отладку, навигацию по коду, поддержку Git и другие возможности. Многие возможности Visual Studio Code недоступны через графический интерфейс, зачастую они используются через палитру команд или JSON-файлы (например, пользовательские настройки). Палитра команд представляет собой подобие командной строки, которая вызывается сочетанием клавиш, позволяет заменять кодовую страницу при сохранении документа, символы перевода строки и язык программирования текущего документа, имеет поддержку плагинов, доступных через Visual Studio Marketplace. Они могут включать в себя дополнения к редактору, поддержку дополнительных языков программирования, статические анализаторы кода. На рисунке 3 представлен вид интерфейса Visual Studio Code.

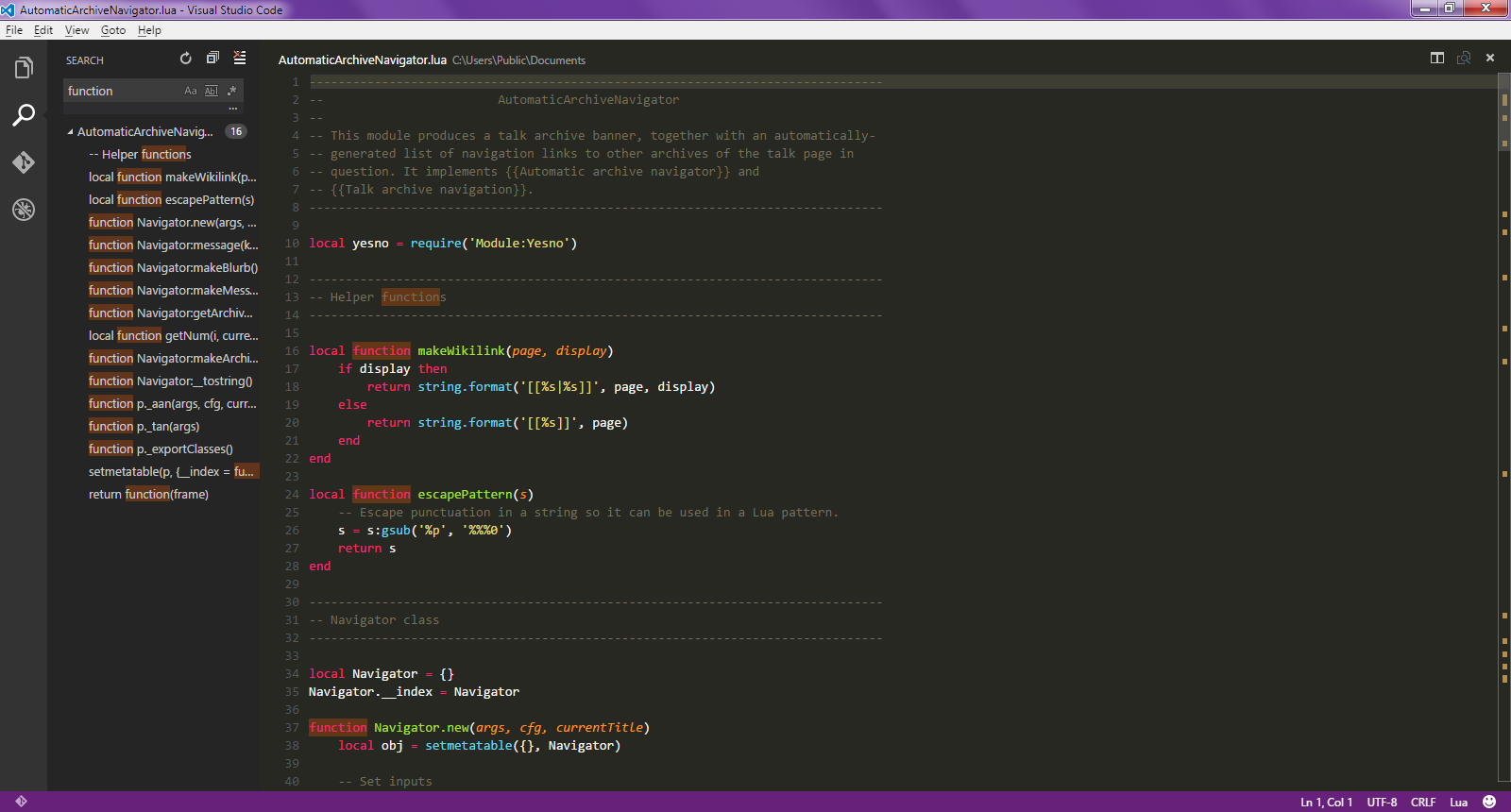


Рисунок 3 - Вид интерфейса Visual Studio Code.

* + 1. Дополнительно программное обеспечение

Денвер– специальное программное обеспечение, которое предназначено для создания и корректировки сайтов. Устанавливается на ПК на базе ОС Windows, при этом нет необходимости в подключении к Интернету. Разработчиками данного ПО являются Котеров Дмитрий (автор), Сущев Антон и Ливач Михаил.

Работа над созданием программы началась в 1999 году после написания Дмитрием Котеловым, одним из разработчиков, пошаговой инструкции с Windows 95/98. В марте 2002 года официально выпустили стартовую версию Denwer. Это был простейший установщик, копирующий документы и файлы в указываемую папку и добавляющий соответствующий ярлык в Windows-автозагрузка.

Долгое время людям, далеким от программирования, было непонятно, для чего нужен Денвер, однако сегодня, когда каждому хочется собственный сайт, правила работы с Денвером «для чайников» пользуются популярностью. Современный Localhost Denwer — пакет бесплатных специализированных программ, который просто устанавливать. Он без проблем копируется на жесткие диски и Flash-карты. Перенести сайт на Денвер способен любой человек, умеющий устанавливать программы самостоятельно.

Базовая комплектация программы включает web-сервер по типу Apache, установщик программной оболочки PH5, реляционную систему MySQL5, web-приложение phpMyAdmin, способное управлять базами данных сайта, имитатор отправки электронных сообщений и инструмент для SMTP-сервера.

С помощью этих компонентов можно запустить простейшие скрипты и директивы SSI без подключения к интернету.

Основные преимущества программы

* Готовая подборка программ, необходимых для отладки.
* Автономность. Ее легко установить и удалить.
* Возможность создать и наладить работу интернет-сайта без покупки хостинга.
* Автоматическое исправление системных файлов.
* Бесплатный доступ.

Ключевая особенность Денвера — поддержка работы сразу с несколькими проектами, каждый из которых располагается на отдельном виртуальном хосте. Виртуальные хосты для проектов создаются автоматически: например, вам достаточно скопировать файлы проекта в /home/ИмяПроекта/www, и он тут же станет доступен по адресу http://ИмяПроекта (DOCUMENT\_ROOT также будет корректным). Это особенно удобно в работе веб-студий, разрабатывающих параллельно несколько сайтов, а также «в связке» с системами контроля версий CVS или Subversion. Схема именования директорий может быть легко настроена персонально на ваш хостинг в шаблоне виртуальных хостов (см. комментарии в httpd.conf).

Все компоненты Денвера уже настроены и готовы для работы (в частности, корректно настроена русскоязычная кодировка MySQL, SSL и т. д.). Кроме того, вы можете обновлять любой из сервисов Денвера ([Apache](http://wikireality.ru/wiki/Apache" \o "Apache), [PHP](http://wikireality.ru/wiki/PHP), [MySQL](http://wikireality.ru/wiki/MySQL" \o "MySQL) и т. д.) вручную, просто копируя новые версии дистрибутивов поверх старых.

Денвер автономен: он может располагаться в любой директории на диске (или даже на флэш-накопителе). Он также не изменяет системных файлов Windows, так что может быть деинсталлирован путем простого удаления своей папки.

Состав базового пакета Денвера:

* Apache 2 с поддержкой SSL и mod\_rewrite.
* PHP5: выполняемые файлы, модуль для веб-сервера Apache, дистрибутивный и адаптированный конфигурационный файл, библиотека GD, mодули поддержки MySQL и sqLite.
* MySQL5 с поддержкой InnoDB, транзакций и русских кодировок (windows-1251).
* phpMyAdmin — панель управления базой данных MySQL, а также скрипт, упрощающий добавление нового пользователя MySQL.
* Отладочный эмулятор sendmail (/usr/sbin/sendmail), не отправляющий письма, а записывающий их в директорию /tmp/!sendmail.
* Система автоматического поиска виртуальных хостов и обновления системного файла hosts, а также конфигурации Apache. Благодаря ей добавление нового виртуального хоста (или домена третьего уровня) заключается в простом создании каталога в /home (см. по аналогии с уже существующими хостами) и перезапуске комплекса. Все изменения вносятся в конфигурационные и системные файлы автоматически, но вы можете управлять этим процессом при помощи механизма шаблонов хостов (см. /usr/local/apache/conf/httpd.conf за детальными разъяснениями).

На официальном сайте Денвера доступны дополнения («пакеты расширения»), расширяющие возможности базового комплекта:

* PHP версии 3 в виде CGI-программы;
* PHP версии 4 в виде CGI-программы;
* дополнитльные модули для Apache;
* дополнительные модули для PHP;
* полная версия ActivePerl;
* интерпретатор ActivePython.
* сервер MySQL версии 4;
* модули поддержки технологии Parser;
* СУБД PostgreSQL;
* СУБД FireBird версий 2 и 1.3

<https://semantica.in/blog/denver-dlya-razrabotchikov-i-ne-tolko.html> -

**[Денвер](https://semantica.in/blog/denver-dlya-razrabotchikov-i-ne-tolko.html" \t "https://yandex.ru/search/_blank)**[для разработчиков и не только](https://semantica.in/blog/denver-dlya-razrabotchikov-i-ne-tolko.html" \t "https://yandex.ru/search/_blank)

<https://webstudio2u.net/ru/programming/183-what-is-denwer.html> -

[Веб-программирование](https://webstudio2u.net/ru/programming/183-what-is-denwer.html" \t "https://yandex.ru/search/_blank)

<http://wikireality.ru/wiki/Denwer> -

**[Denwer](http://wikireality.ru/wiki/Denwer" \t "https://yandex.ru/search/_blank)**

phpMyAdmin-это бесплатный программный инструмент, написанный на [РНР](https://php.net/), предназначен для решения вопросов администрирования: [MySQL](https://www.mysql.com/) через web. phpMyAdmin поддерживает широкий спектр операций на MySQL и MariaDB. Часто используемые операции (управление базами данных, таблицами, столбцы, отношения, индексы, пользователи, разрешения и т. д.) могут быть выполнены с помощью пользовательский интерфейс, в то время как у все еще есть возможность непосредственно выполнить любую инструкцию SQL.

phpmyadmin – это простое, удобное и хорошо документированное решение, которое переведено на множество языков, в том числе и русский. Даже если у пользователя возникают какие-то вопросы по работе с этим приложением, в Интернете можно без труда найти ответы на большинство стандартных вопросов. Phpmyadmin написан на языке PHP.

Что можно делать с помощью phpmyadmin:

* Создавать и корректировать базы данных, таблицы, записи;
* Создавать пользователей;
* Возможность исполнять SQL-команды;
* Система поиска по базе данных;

В общем, есть все необходимые средства, которые необходимы для работы с базой данных.

Нужно отметить, что практически все хостеры устанавливают именно phpmyadmin  в свои панели управления для работы с базой данных. Это приложение является одним из самых популярных среди других приложений в этом классе.

Все операции с ячейками и их содержимым – просмотр, копирование, удаление, вставка производятся в один клик. Наиболее часто веб-разработчиками и администраторами сайта используется возможность моментального создания так называемого дампа базы (копии) для переноса сайта с одного хоста на другой. При желании его можно получить сразу в заархивированном виде.

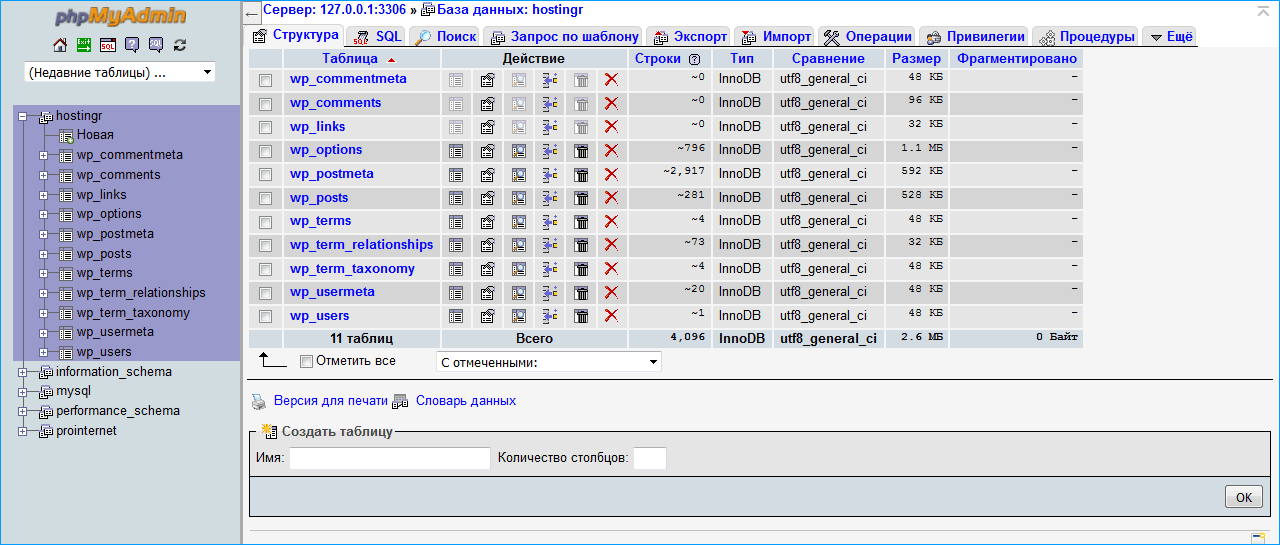
Еще одной интересной возможностью PHPMyAdmin является восстановление пароля для входа в административную часть сайта. Достаточно зайти в ячейку Users или аналогичную ей и скопировать в вашей учетной записи искомые логин и пароль. Эта программа может с успехом заменить административную часть сайта в случае ее неисправности.

Единственным, да и то весьма спорным, ее недостатком является представление содержимого сайта в виде кода HTML.

Интерфейс PhpMyAdmin

Программа управления базами данных открывается в окне браузера. Ее интерфейс лаконичен и понятен без дополнительного инструктажа. Существует и русифицированная версия. Интерфейс PHPMyAdmin состоит из окна, поделенного на две части.

Слева находится узкий столбец, в котором представлены список баз данных и древовидная структура каждой из них. Справа – рабочее окно, в котором отражаются содержимое ячеек базы, меню команд и служебные кнопки.



Причины популярности PhpMyAdmin

Простота пользования и удобство интерфейса PHPMyAdmin предопределили его популярность и широту распространения во Всемирной сети. Эту программу включают в пакет серверного софта большинство хостеров, предоставляющих виртуальные площадки для размещения сайтов. Кроме того, она является частью пакета программ локальных серверов, используемых веб-разработчиками для отладки сайтов на домашних компьютерах. Например, она входит в состав популярных пакетов серверных программ Denwer и Apache.

<https://www.phpmyadmin.net/> - Приведение MySQL в сеть

<https://ipipe.ru/info/phpmyadmin> - PhpMyAdmin.

1. Разработка программного продукта

   2. Разработка клиентской части
   3. Разработка серверной части

Для разработки серверной части проекта была выбрана среда Node.js c фреймворком Express.js. Nodee.js — это среда выполнения JavaScript, которая в основном используется для создания веб-приложений. Другими словами, это реализация JavaScript на стороне сервера, используемая для написания серверной части приложения. (Хотя многие платформы Node.js также могут работать с внешним интерфейсом.) Ниже приведено несколько примеров того, что можно создать с помощью Node.js.

* Одностраничные приложения. Это веб-приложения, которые работают в браузере и не требуют повторной загрузки страницы при каждом ее использовании для получения новых данных. Некоторые примеры одностраничных приложений включают в себя приложения социальных сетей, электронную почту, приложения карт, текст в Интернете, инструменты для рисования и т. д.
* Приложения в режиме реального времени. Это веб-приложения, которые способствуют пользователям получать информацию сразу после публикации автором, а не требовать периодических проверок источника на наличие обновлений.
* Приложения потоковой передачи данных. Это приложения (или службы), которые отправляют данные или содержимое по мере их поступления (или создания), сохраняя подключение для продолжения загрузки дополнительных данных, содержимого или компонентов по мере необходимости. К некоторым примерам относятся приложения потоковой передачи видео и аудио.
* API REST. Это интерфейсы, которые предоставляют данные для взаимодействия с веб-приложением другого пользователя. Например, служба API календаря может предоставлять даты и время для концертного помещения, которое может использоваться веб-сайтом локальных событий другого пользователя.
* Готовые к просмотру приложения на стороне сервера (SSR). Эти веб-приложения могут работать как на клиенте, так и на сервере, позволяя динамически отображать любое известное содержимое и быстро перехватывать неизвестное по мере его доступности.
* Средства командной строки. Они позволяют автоматизировать повторяющиеся задачи, а затем распространить ваше средство по обширной экосистеме Node.js.
* Программирование оборудования. Хотя Node.js не так популярен, как веб-приложения, в настоящее время Node.js пользуется все большей популярностью для сбора данных с датчиков, маяков, передатчиков, двигателей или всего, что генерирует большие объемы данных. Node.js может обеспечить сбор данных, анализ этих данных, обмен данными между устройством и сервером и принятие мер на основе анализа.

Express предоставляет ряд готовых абстракций, которые упрощают создание сервера и серверной логики, в частности, обработка отправленных форм, работа с куками, CORS и так далее.

Код созданного сервера изображен на рисунке 1.



Рисунок 1 – Созданный сервер приложения

Все модули подключения с помощью метода require(). Для использования Express в начале надо создать объект, который будет представлять приложение. Для приложения устанавливается шаблонный процессор handlebars, указывается папка, содержащая статичные файлы, и парсер данных. Далее создается объект шаблонного процессора и указывается папка с частичными представлениями (блоки кода, которые можно использовать много раз в разных частях приложения). Затем создается объект подключения к базе данных MySQL и объект системы маршрутизации, которой передается объекты приложения и подключения к базе данных. Затем сервер начинается прослушивание по указанному порту и выводит в консоли сообщение о том, что сервер запущен.

На рисунке 2 изображен код файла подключения к базе данных MySQL.

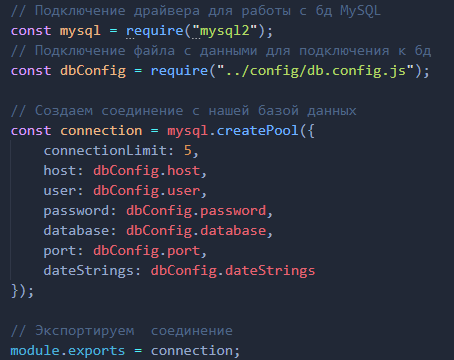


Рисунок 2 – Подключение к базе данных MySQL

Для подключения к базе данных MySQL используется драйвер mysql2. Для создания подключения используется метод createConnection() или createPool(), которые в качестве параметра принимают объект с свойствами, содержащими необходимые данные для успешного соединения. Запросы к базе данных осуществляются с помощью метода query(sqlString, callback) или execute(sqlString, callback), где sqlString - выполняемая SQL-команда, а callback - функция обратного вызова, через параметры которой мы можем получить результаты выполнения SQL-команды или возникшую ошибку.

На рисунке 3 изображен код системы маршрутизации.

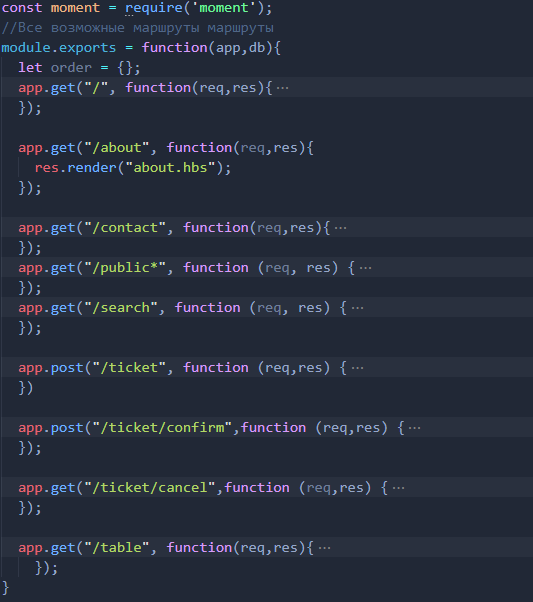


Рисунок 3 – Система маршрутизации

При переходе по одному из таких маршрутов происходит выполнение каких-либо действий или запросов и у клиента отображается соответствующая страница с помощью метода render(), в параметры которого передается строковый аргумент с названием файла страницы, а во второй параметр могут передаваться объекты, которые необходимо использовать на страниц, или происходит переадресация на другой маршрут. Для обработки данных по определенному маршруту можно использовать ряд функций, в частности:

* use
* get
* post
* put
* delete

В качестве первого параметра эти функции могут принимать шаблон адреса, запрос по которому будет обрабатываться. Второй параметр функций представляет функцию, которая будет обрабатывать запрос по совпавшему с шаблоном адресу. Когда приходит запрос Express сопоставляет запрошенный адрес с каждым из маршрутов. Затем выбирается первый совпавший маршрут. При совпадении маршрута вызывается его функция обработчика.

f

* + 1. Разработка базы данных

1. Отладка и тестирование продукта