**Оглавление**

Введение стр. 2  
Тема 1 Изучение работы отделов предприятии стр.2  
1.1. Информация о предприятии стр.2  
1.2. Автоматизированные системы предприятия стр.4  
Тема 2 Выполнение обязанностей дублеров инженерно- технических работников стр.6  
2.1. Права и обязанности инженерно-технических работников. стр.6  
Тема 3 Выполнение работ и изучение материалов, предусмотренных программой междисциплинарного экзамена стр.8  
UML ДИАГРАММЫ стр.9  
Описание выбранных языков программирования стр.9  
ВЫБОР ТЕХНОЛОГИИ ДЛЯ КЛИЕНТСКОЙ ЧАСТИ стр.9  
Сравнение SVG и HTML 5 Canvas стр.10  
Преимущества стр.11  
Недостатки стр.11  
Язык программирования JavaScript стр.12  
Описание языка стр.12  
Структура языка стр.13  
ВЫБОР ТЕХНОЛОГИИ ДЛЯ СЕРВЕРНОЙ ЧАСТИ стр.14  
Ajax стр.14  
Comet стр.16  
SignalR стр.16  
ASP.NET стр.17  
IIS стр.18  
БАЗА ДАННЫХ стр.21  
MS SQL SERVER стр.21

Microsoft Visual code стр.22  
ReactJs стр.23

Redux стр.23

Npm стр.24

NodeJs стр.24

Express стр.25

MongoDB стр.25

Socket.Io стр.27

Заключение стр.27

**Введение**  
  
Большая часть работы программистов связана с написанием исходного кода, тестированием и отладкой программ на одном из языков программирования. Исходные тексты и исполняемые файлы программ являются объектами авторского права и являются интеллектуальной собственностью их авторов и правообладателей.  
Различные языки программирования поддерживают различные стили программирования (парадигмы программирования). Отчасти искусство программирования состоит в том, чтобы выбрать язык программирования, наиболее полно подходящий для решения поставленной задачи. Разные языки требуют от программиста различного уровня внимания к деталям при реализации алгоритма, результатом чего часто бывает компромисс между простотой и производительностью (или междувременем программистаивременем пользователя).  
Единственный язык, напрямую выполняемый ЭВМ— этомашинный язык(также называемыймашинным кодомиязыком машинных команд). Изначально все программы писались в машинном коде, но сейчас этого практически уже не делается. Вместо этого программисты пишутисходный кодна том или ином языке программирования, затем,используякомпилятор, транслируют его в один или несколько этапов в машинный код, готовый к исполнению на целевом процессоре, или в промежуточное представление, которое может быть исполнено специальныминтерпретатором— виртуальной машиной. Но это справедливо только дляязыков высокого уровня. Если требуется полный низкоуровневый контроль над системой на уровне машинных команд и отдельных ячеек памяти, программы пишут наязыке ассемблера, мнемонические инструкции которого преобразуются один к одному в соответствующие инструкции машинного языка целевого процессора ЭВМ. (По этой причине трансляторы с языков ассемблера— ассемблера— получаются алгоритмически простейшими трансляторами.)  
В некоторых языках вместо машинного кода генерируется интерпретируемый двоичный код «виртуальной машины», также называемый байт-кодом(byte-code). Такой подход применяется в Forth, некоторых реализациях Lisp,Java, Perl, Python, языках для.NET Framework.  
  
Тема 1 Изучение работы отделов предприятии  
1.1. Информация о предприятии

Проектная Академия «KAZGOR» - крупнейшая проектная организация в Казахстане, с даты основания в 1930 году,

является безусловным лидером в проектировании объектов жилищного и гражданского назначения.

В течение всего периода своей деятельности для нас главным было стремление создавать неповторимый облик

городов с комфортной и безопасной средой обитания, сохранение традиций и уникальных особенностей региона,

применение современных материалов и технологий, высокие конструктивные и эстетические требования

к возводимым зданиям и сооружениям независимо от их назначения и местонахождения.

Высокие стандарты, опирающиеся на национальные традиции и опыт лучших специалистов, применение

современных технологий и управления в соответствии с международными стандартами качества (ISO 9001:2008),

экологии (ISO 14001:2009), безопасности и охраны труда (BS OHSAS 18001:2007) – залог успеха в предоставлении

качественных услуг клиентам академии, осознании ответственности перед будущими поколениями за сохранение

окружающей среды, создании безопасных и здоровых условий труда для персонала.

Достижения академии получили признание и отмечены многочисленными наградами и званиями республиканских

и международных организаций. Самой высокой оценкой деятельности академии стало присуждение в ноябре

2007 г. звания лауреата премии Президента Республики Казахстан «За достижения в области качества»

Верный курс

В течение нескольких десятилетий работы у академии выработался свой стиль воплощения идей и замыслов,

отличительными чертами которого являются высокая степень ответственности, профессионализм, творческий

подход к процессу проектирования.

Зарубежные партнеры

Творческий потенциал и ресурсы академии позволили наладить тесное взаимодействие и партнерские связи с такими лидерами проектного бизнеса, как «Skidmore, Owings & Merrill LLP» («SOM»), «RAMSA», «Yost Grube Hall», «LERA», «ICC», «SNIP Register Inc.», «Foster & Partners», «Buro Happold», «Dizayn Grup Ltd», «Murat Sivka Group», «Boyut Architecture Co Ltd», «Space Group», «ЦНИИСК им. В.А.Кучеренко» и другие.

1.2. Автоматизированные системы предприятия  
В ТОО ПА “KAZGOR” пять компьютерных кабинетов, оснащенных компьютерным оборудованием.  
В здании 150 персональных компьютеров, 30 ноутбуков и 1 сервер.  
Средние технические характеристики персональных компьютеров:  
1. ОЗУ 4 Гб  
2. CPU 2.2 Ггц\*2  
3. Материнские платы фирмы Asus  
4. HDD 1000 Гб  
5. DVD дисководы  
6. USB 3.1\*4  
7. ВидеокартыPCI Express, Radeon HD 3830.  
Средние технические характеристики ноутбуков:  
1. ОЗУ 512 Мб  
2. CPU 2.2 Ггц  
3. материнские платы фирмы Acer  
4. HDD 150 Гб  
5. DVD дисководы  
6. USB 2.0\*2  
Технические характеристики сервера:  
1. ОЗУ 8 Гб  
2. CPU 3.0 Ггц\*4  
3. Материнская плата фирмы Asus  
4. HDD 2 Тб  
5. Дисковода нет в наличии  
6. USB 2.0\*4  
Помимо компьютеров, в колледже имеются 70 принтеров(20 струных и 50 лазерных), 30 сканеров, 25 проекторов, 10 факсов.  
Принтера  
Принтер | Картридж | Тонер |  
Samsung |  
ML-1210 | ML1210D3 | ML 1210 |  
Canon |  
iR 1018, 1020, 1022, 1024 | C-EXV18 | HP LJ 1200, Булат 1505 |  
FC 108/200/220/228/336PC 860/880/890 | E-16 | |  
LBP-800/810/1120 | EP-22 | HP LJ 1100, 1200, 1000w, 1300 |  
LBP-3250 | Canon 713 | Булат HP LJ 1505 Hp1005 |  
MF4430 | CE278A, Canon 728 | P1005, P1505, Булат 1505 |  
MF3228 | EP26/EP27 | HP LJ 1010, 1200,1505 |  
iR-1133 | C-EXV 40/ НР 505х/ CE505X | |  
HP |  
LJ P2015/M2727 | Q7553A, Q7553X | HP LJ 1010, 1200, 1300, 1320 |  
LJ 1010 | Q2612A | HP LJ 1010, 1015 |  
LJ 1100 | EP-22, C4092A | HP LJ 1100, 1200, 1000w, 1300 |  
LJ 1200 | C7115A | |  
LJ 1300 | Q2613X | |  
LJ 1005 | C7115A | |  
LJ 1320 |Q5949A | HP LJ 1320 |  
Сканеры  
Сканеры Hewlett Packard  
HP ScanJet 3400С  
К основным преимуществам модели относятся:  
- Подключение к компьютеру как через параллельный интерфейс, так и через шину USB;  
- Достаточно удобное программное обеспечение, позволяющее освоить работу со сканером даже новичку в самый короткий отрезок времени;  
- Три кнопки управления на передней панели сканера, которые во многом упрощают работу.  
Программа распознавания поддерживает следующие типы форматов файлов для сохранения результата сканирования:  
- Для фотографических изображений - BMP, TIFF, JPEG, PCX, FPX, GIF;  
- Для текста без графических изображений - TXT;  
- Для текста с графикой - PDF, RTF, HTML.  
Системные требования:  
- Pentium 90 MHz и выше;  
- 32 Мб оперативной памяти;  
- Примерно 150-200 Мб свободного пространства на жестком диске;  
- Windows7 и более новые.  
ScanJet 4200C  
Основными преимуществами данной модели являются простота в настройке и быстрая работа.  
Что касается минимальных системных требований, то для полноценной работы сканера необходим уже процессор не ниже intel core I5 6300 и операционная система Windows 7, а также обязательное наличие USB-шины  
HP ScanJet 4300  
Программа сканирования для данной модели используется точно такая же, как и для модели HP ScanJet 3400С. Но при этом она (и, конечно же, дополнительные драйвера) позволяет использовать сканер как копировальный аппарат. Пользователь имеет возможность выполнить любое количество копий, распечатав их при этом на принтере, а все управление производить при помощи панели управления сканера, которая кроме кнопок управления имеет небольшой жидкокристаллический дисплей. И для этого компьютер совершенно не нужен. С одного оригинала пользователь может сделать до 99 копий вне зависимости от того, черно-белые они или цветные. Так же при помощи панели можно настроить печать, то есть можно, например, с цветного изображения сделатьчерно-белые копии и т.д.  
Как можно видеть из технических характеристик сканеров HP ScanJet 3400С и HP ScanJet 4300С, если их сравнивать, модели практически не отличаются друг от друга. Даже программа сканирования используется одна и так же. Но все преимущества и новшества, отнесенные к сканеру HP ScanJet 4300С, обеспечиваются не чем иным, как программным обеспечением и более совершенными драйверами. Хотя многих функций не было бы без некоторых аппаратных новшеств сканера.  
HP ScanJet 5300  
Программа сканирования, использующаяся при работе с данным сканером, позволяет не только довольно быстро произвести сканирование или копирование документов, но и может автоматически распознать сканируемый текст и сохранить его в файл. При этом поддерживаются следующие типы форматов файлов:  
- Для графических изображений или фотографий - PDF, TIFF, BMP, PCX, JPEG, FPX, GIF, PNG, WMF;  
- Для текста - TXT, HTML, RTF;  
- Для текста с графическими изображениями или фотографиями - PDF, RTF, HTML.  
Также возможна работа со сканером через программу "AdobePhotoDeluxe", входящую в состав программного обеспечения на компакт- диске. Для этой программы поддерживаются следующие форматы файлов: PDD, PBD, PDF, PSD, BMP, FPX, JPG, JPE, PCT, PIC, PNG, TIF, EPS, GIF.  
Программное Обеспечение  
Все компьютеры в ТОО ПА“KAZGOR”работают на базе операционной системы семейства Windows, а именно Windows10 Windows 7, Windowsserver8.1.  
ДополнительноеПО  
MS Office 2007/2010, Nero 7.8, Foxit phantom, K-lite Codec pack, Win Rar, Win DJView, Boost Speed, Adobe Photoshop CS 5.  
Антивирусы  
Антивирус Касперского, NOD32 SmatSecurity.  
Тема 2 Выполнение обязанностей дублеров инженерно- технических работников  
2.1. Права и обязанности инженерно-технических работников.  
ОБЩИЕ ПОЛОЖЕНИЯ  
1.1. Настоящая должностная инструкция определяет функциональные обязанности, права и ответственность Программиста.  
1.2. Программистназначается на должность и освобождается от должности в установленном действующим трудовым законодательством порядке приказом Генерального директора по представлению руководителя отдела IT.  
1.3. Программист подчиняется непосредственно руководителю отдела IT.  
1.4. На должность Программиста назначается лицо, имеющее высшее техническое образование.  
1.5. Программист должен владеть компьютером на профессиональном уровне, в том числе уметь использовать и создавать специальные компьютерные программы.  
1.6. Программист должен знать:  
- законы, постановления, распоряжения, приказы, другие руководящие документы и нормативные акты, касающиеся работы предприятия;  
- законодательство о труде;  
- Правила внутреннего трудового распорядка;  
- правила и нормы охраны труда;  
- Правила техники безопасности, производственной санитарии и гигиены, противопожарной безопасности, гражданской обороны.  
1.7. Программист должен обладать коммуникабельностью, должен быть энергичен и позитивно настроен.  
ФУНКЦИОНАЛЬНЫЕ ОБЯЗАННОСТИ ПРОГРАММИСТА  
2.1. Программист:  
2.1.1. Разрабатывает на основе анализа математических моделей и алгоритмов решения экономических и других задач программы, обеспечивающие возможность выполнения алгоритма и соответственно поставленной задачи средствами вычислительной техники, проводит их тестирование и отладку.  
2.1.2. Разрабатывает технологию решения задачи по всем этапам обработки информации.  
2.1.3. Осуществляет выбор языка программирования для описания алгоритмов и структур данных.  
2.1.4. Определяет информацию, подлежащую обработке средствами вычислительной техники, ее объемы, структуру, макеты и схемы ввода, обработки, хранения и вывода, методы ее контроля.  
2.1.5. Выполняет работу по подготовке программ к отладке и проводит их отладку.  
2.1.6. Определяет объем и содержание данных контрольных примеров, обеспечивающих наиболее полную проверку соответствия программ их функциональному назначению.  
2.1.7.Осуществляет запуск отлаженных программ и ввод исходных данных, определяемых условиями поставленных задач.  
2.1.8. Проводит корректировку разработанной программы на основе анализа выходных данных.  
2.1.9. Определяет возможность использования готовых программных продуктов.  
2.1.10. Разрабатывает инструкции по работе с программами, оформляет необходимую техническую документацию.  
2.1.11. Осуществляет сопровождение внедренных программ и программных средств.  
2.1.12. Разрабатывает и внедряет системы автоматической проверки правильности программ.  
2.1.13. Выполняет работу по унификации и типизации вычислительных процессов.  
2.1.14. Принимает участие в создании каталогов и картотек стандартных программ, в разработке форм документов в электронном виде, подлежащих компьютерной обработке, в проектировании программ, позволяющих расширить область применения вычислительной техники.  
2.1.15. Обеспечивает правильную техническую эксплуатацию, бесперебойную работу компьютеров и отдельных устройств.  
2.1.16. Участвует в разработке перспективных и годовых планов и графиков работы, технического обслуживания и ремонта оборудования, мероприятий по улучшению его эксплуатации, предупреждению простоев в работе, повышению качества работы, эффективному использованию вычислительной техники.  
2.1.17. Осуществляет подготовку компьютеров и отдельных устройств к работе, их технический осмотр, проводит проверку наличия неисправностей, устраняет неисправности и предотвращает появление неисправностей в будущем.  
2.1.18. Принимает меры по своевременному и качественному выполнению ремонта компьютеров и отдельных устройств своими силами или силами третьих лиц.  
2.1.19. Принимает участие в проведении инвентаризаций.  
2.1.20. Должен беречь имущество предприятия, не разглашать информацию и сведения, являющиеся коммерческой тайной предприятия.  
2.1.21. Не дает интервью, не проводит встречи и переговоры, касающиеся деятельности предприятия, без разрешения руководства предприятия.  
2.1.22. Соблюдает трудовую и производственную дисциплину, правила и нормы охраны труда, требования производственной санитарии и гигиены, требования противопожарной безопасности, гражданской обороны.  
2.1.23. Исполняет распоряжения и приказы Генерального директора предприятия и руководителя отдела IT.  
2.1.24. Информирует руководство об имеющихся недостатках в работе предприятия, принимаемых мерах по их ликвидации.  
2.1.25. Способствует созданию благоприятного делового и морального климата на предприятии.  
ПРАВА ПРОГРАММИСТА  
3.1. Программист имеет право:  
3.1.1. Предпринимать соответствующие действия по устранению причин, создающих препятствия для осуществления программистом своих функциональных обязанностей.  
3.1.2. Вносить предложения администрации предприятия по улучшению работы, относящейся к функциональным обязанностям Программиста и всего предприятия в целом.  
УСЛОВИЯ РАБОТЫ ПРОГРАММИСТА  
5.1. Режим работы Программиста определяется в соответствии с Правилами внутреннего трудового распорядка, установленными на предприятии.  
  
Тема 3 Выполнение работ и изучение материалов, предусмотренных программой междисциплинарного экзамена  
В рамках индивидуального задания необходимо написать браузерную онлайн игру «Крестики-нолики пять в ряд» посредством HTML, JavaScript, CSS, C#.  
UML ДИАГРАММЫ  
  
UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это — открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.Диаграмма деятельности  
Диаграмма деятельности (Activity diagram) — диаграмма, на которой показано разложение некоторой деятельности на её составные части. Под деятельностью (англ. activity) понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий (англ. action), соединённых между собой потоками, которые идут от выходов одного узла к входам другого.  
Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.  
  
Диаграмма вариантов использования  
Диаграмма вариантов использования (Use case diagram, диаграмма прецедентов) — диаграмма, на которой отражены отношения, существующие между актёрами и вариантами использования.  
Основная задача — представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.  
  
Описание выбранных языков программирования  
  
HTML (от англ. HyperText Markup Language — «язык разметки гипертекста») — стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц создаются при помощи языка HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.  
HTML является приложением («частным случаем») SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO 8879. XHTML же является приложением XML.JavaScript — прототипно-ориентированный скриптовый язык программирования. Является диалектом языка ECMAScript.  
JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.  
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 не программистами. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.  
Название «JavaScript» является зарегистрированным товарным знаком компании Oracle Corporation.  
C# (произносится си шарп) — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework и впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.  
C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.  
Переняв многое от своих предшественников — языков C++, Java, Delphi, Модула и Smalltalk — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественное наследование интерфейсов).  
  
ВЫБОРТЕХНОЛОГИИ ДЛЯ КЛИЕНТСКОЙ ЧАСТИ  
Для написания клиентской части необходимо использовать SVG или Canvas и JavaScript.  
  
Сравнение SVG и HTML 5 Canvas  
HTML5 Canvas и SVG — веб-технологии, которые позволяют использовать высококачественную графику в браузерах, но фундаментально они в корне отличаются друг от друга.  
SVG — векторный графический формат, основанный на XML. SVG-контент может быть статическим, динамическим, интерактивным и анимированным — он очень гибок.  
С помощью SVG можно получить намного больше, чем просто векторную графику и анимацию. Можно разработать высокоинтерактивное веб-приложение с программированием, расширенными анимационными событиями, фильтрами и многое другое.  
Спецификация HTML5 Canvas определяет универсальный JavaScript API, позволяющий выполнять операции отрисовки объектов. Для рисования на канве можно использовать два разных подхода:  
\* 2D-подход,  
\* 3D-подход (WebGL).  
Первый лучше внедрён и доступен во всех современных веб-браузерах (за исключением IE), в то время как второй находится на ранней стадии определения, имея лишь несколько экспериментальных реализаций.  
  
Преимущества  
Таблица 1  
Canvas | SVG |  
\* Высокая производительность при отрисовке любых 2D объектов. \* Стабильная производительность — всё есть пиксель. Производительность падает только при увеличении разрешения изображения. \* Можно сохранить полученное изображение в PNG или JPG файл. \* Лучше всего подходит для создания растровой графики (например, в играх, фракталов и т.п.), редактирования изображений и операций, требующих манипулирования на уровне пикселей. | \* Нет зависимости от разрешения — SVG лучше подходит для кроссплатформенных пользовательских интерфейсов, так как позволяет масштабировать изображение при различных разрешениях экрана. \* SVG очень хорошо поддерживает анимацию. Элементы могут быть анимированы с использованием описательного синтаксиса или с помощьюJavaScript. \* Можно получить полный контроль над каждым элементом, используя SVG DOM API в JavaScript. \* SVG хранится в формате XML, что предоставляет больше возможностей браузерам по обеспечению доступности SVG документов по сравнению с элементом canvas. Таким образом, SVG выглядит лучшим решением для пользовательских интерфейсов веб-приложений. |  
  
Недостатки  
Таблица 2  
Canvas | SVG |  
\* Отрисовка основана на пикселях. \* Не существует API для анимации. Вам придется прибегать к использованию таймеров и других событий для обновления канвы. \* Слабые возможности по рендерингу текста. \* Возможно, не самый лучший выбор, когда доступность имеет решающее значение. Канва предоставляет вам поверхность для рисования в выбранном контексте (2D и 3D). Можно указать альтернативный контент внутри элемента canvas, который будет показан браузером при невозможности отображения графики. Кроме того, вы можете выполнить проверку доступности выбранного Canvas API с помощью JavaScript. На основе этого вы можете обеспечить различную функциональность для пользователей браузеров с разной поддержкой HTML 5 Canvas. \* HTML 5 Canvas не подходит для создания веб-сайтов или интерфейсов веб-приложений, так как пользовательские интерфейсы обычно должны быть динамическими и интерактивными, а Canvas требует от вас постоянной перерисовки каждого элемента в интерфейсе. | \* Низкая скорость рендеринга при увеличении сложности документа (рисунка), так как используется модель DOM \* Скорее всего, SVG не подходит для таких приложений как игры. Возможно лучшим выбором будет комбинация HTML Canvas + SVG. |  
Для курсовой работы была выбрана технология Canvas, так как его лучше всего использовать для создания растровой графики (например, в играх, фракталов и т.п.), редактирования изображений и операций, требующих манипулирования на уровне пикселей.  
  
Язык программирования **JavaScript**  
  
**JavaScript** (/ˈdʒɑːvɑːˌskrɪpt/; аббр. **JS** /ˈdʒeɪ.ɛs./) — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript (стандарт ECMA-262[7]).

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам[8].

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. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке[~ 1][9].

Название «JavaScript» является зарегистрированным товарным знаком корпорации Oracle в США[10].

В 1992 году компания *Nombas* (впоследствии приобретённая Openwave[en]) начала разработку встраиваемого скриптового языка *Cmm* (Си-минус-минус), который, по замыслу разработчиков, должен был стать достаточно мощным, чтобы заменить макросы, сохраняя при этом схожесть с Си, чтобы разработчикам не составляло труда изучить его[11]. Главным отличием от Си была работа с памятью. В новом языке всё управление памятью осуществлялось автоматически: не было необходимости создавать буферы, объявлять переменные, осуществлять преобразование типов. В остальном языки сильно походили друг на друга: в частности, *Cmm* поддерживал стандартные функции и операторы Си[12]. *Cmm* был переименован в *ScriptEase*, поскольку исходное название звучало слишком негативно, а упоминание в нём Си «отпугивало» людей[11][13]. На основе этого языка был создан проприетарный продукт *CEnvi*. В конце ноября 1995 года *Nombas* разработала версию *CEnvi*, внедряемую в веб-страницы. Страницы, которые можно было изменять с помощью скриптового языка, получили название *Espresso Pages* — они демонстрировали использование скриптового языка для создания игры, проверки пользовательского ввода в формы и создания анимации. *Espresso Pages* позиционировались как демоверсия, призванная помочь представить, что случится, если в браузер будет внедрён язык *Cmm*. Работали они только в 16-битовом Netscape Navigator под управлением Windows[14].

Самая первая реализация JavaScript была создана Бренданом Эйхом (Brendan Eich) в компании Netscape, и с тех пор обновляется, чтобы соответствовать ECMA-262 Edition 5 и более поздним версиям. Этот движок называется SpiderMonkey и реализован на языке C/C++. Движок Rhino создан Норрисом Бойдом (Norris Boyd) и реализован на языке Java. Как и SpiderMonkey, Rhino соответствует ECMA-262 Edition 5.

Структура языка  
  
Структурно JavaScript можно представить в виде объединения трёх чётко различимых друг от друга частей:  
\* ядро (ECMAScript),  
\* объектная модель браузера (Browser Object Model илиBOM (de)),  
\* объектная модель документа (Document Object Model или DOM).  
Если рассматривать JavaScript в отличных от браузера окружениях, то объектная модель браузера и объектная модель документа могут не поддерживаться.  
Объектную модель документа иногда рассматривают как отдельную от JavaScript сущность, что согласуется с определением DOM как независимого от языка интерфейса документа. В противоположность этому ряд авторов находят BOM и DOM тесно взаимосвязанными.

## Возможности языка

## JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование[35][36] 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, замыкания[37] — 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.

В языке отсутствуют такие полезные вещи[38], как:

* стандартная библиотека: в частности, отсутствует интерфейс программирования приложений по работе с файловой системой, управлению потоками ввода-вывода, базовых типов для бинарных данных;
* стандартные интерфейсы к веб-серверам и базам данных;
* система управления пакетами[~ 2], которая бы отслеживала зависимости и автоматически устанавливала их

ВЫБОР ТЕХНОЛОГИИ ДЛЯ СЕРВЕРНОЙ ЧАСТИ  
  
Ajax  
  
AJAX, Ajax (ˈeɪdʒæks, от англ. Asynchronous Javascript and XML — «асинхронный JavaScript и XML») — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.  
AJAX — не самостоятельная технология, а концепция использования нескольких смежных технологий. AJAX базируется на двух основных принципах:  
\* использование технологии динамического обращения к серверу «на лету», без перезагрузки всей страницы полностью, например:  
\* с использованием XMLHttpRequest (основной объект);  
\* через динамическое создание дочерних фреймов;  
\* через динамическое создание тега <script>.  
\* через динамическое создание тега <img>, как это реализовано в google analytics.  
\* Использование DHTML для динамического изменения содержания страницы;  
В качестве формата передачи данных могут использоваться фрагменты простого текста, HTML-кода, JSON или XML.  
Преимущества  
\* Экономия трафика  
Использование AJAX позволяет значительно сократить трафик при работе с веб-приложением благодаря тому, что часто вместо загрузки всей страницы достаточно загрузить только изменившуюся часть, как правило, довольно небольшую.  
\* Уменьшение нагрузки на сервер  
AJAX позволяет несколькоснизить нагрузку на сервер. К примеру, на странице работы с почтой, когда вы отмечаете прочитанные письма, серверу достаточно внести изменения в базу данных и отправить клиентскому скрипту сообщение об успешном выполнении операции без необходимости повторно создавать страницу и передавать её клиенту.  
\* Ускорение реакции интерфейса  
Поскольку нужно загрузить только изменившуюся часть, пользователь видит результат своих действий быстрее.  
\* Обращения сайта к серверу не ограничиваются щелчками по ссылкам и отправкой форм  
Например, при вводе запроса в Google выводится подсказка. На многих сайтах при регистрации пользователь вводит имя, и сразу же выводится, доступно это имя или нет; это тоже AJAX. AJAX удобен для программирования чатов, административных панелей и других инструментов, которые выводят меняющиеся со временем данные.  
Недостатки  
\* Отсутствие интеграции со стандартными инструментами браузера  
Динамически создаваемые страницы не регистрируются браузером в истории посещения страниц, поэтому не работает кнопка «Назад», предоставляющая пользователям возможность вернуться к просмотренным ранее страницам, но существуют скрипты, которые могут решить эту проблему.  
Другой недостаток изменения содержимого страницы при постоянном URL заключается в невозможности сохранения закладки на желаемый материал. Проблему можно успешно решить с помощью History.pushState.[4]  
\* Динамически загружаемое содержимое недоступно поисковикам (если не проверять запрос, обычный он или XMLHttpRequest)  
Поисковые машины не могут выполнять JavaScript, поэтому разработчики должны позаботиться об альтернативных способах доступа к содержимому сайта.  
\* Старые методы учёта статистики сайтов становятся неактуальными  
Многие сервисы статистики ведут учёт просмотров новых страниц сайта. Для сайтов, страницы которых широко используют AJAX, такая статистика теряет актуальность.  
\* Усложнениепроекта  
Перераспределяется логика обработки данных — происходит выделение и частичный перенос на сторону клиента процессов первичного форматирования данных. Это усложняет контроль целостности форматов и типов. Конечный эффект технологии может быть нивелирован необоснованным ростом затрат на кодирование и управление проектом, а также риском снижения доступности сервиса для конечных пользователей.  
\* Требуется включенный JavaScript в браузере  
JavaScript может быть выключен из соображений безопасности. И, конечно же, AJAX-страницы труднодоступны неполнофункциональным браузерам, роботам и веб-архивам.  
\* Низкая скорость при грубом программировании  
Казалось бы, AJAX предназначен именно для повышения скорости. Но, когда AJAX-запросов на одной странице много и, например, по каждому щелчку подгружается список, AJAX-страница становится даже медленнее традиционной.  
Comet  
Таблица 3  
Техника | Описание | Преимущества | Недостатки |  
Polling | Постоянный опрос сервера Ajax-запросами | + простота реализации+ поддержка во всех современных браузерах | - задержка в результатах- при уменьшении задержки существенно увеличивается нагрузка на сервер |  
Long Polling | Ajax-запросы, идущие один за другим, но каждый запрос держится открытым в течение нескольких минут | + сниженная нагрузка на сервер по сравнению с обычным Polling+ уменьшенный трафик+ поддержка во всех современных браузерах | - больше одновременно открытых соединений, т.к. каждый запрос живет дольше |  
Server-Sent Events | Новый стандарт HTML5, работающий поверх HTTP. Позволяет создавать долгоживущее соединение с сервером, чтобы сервер мог отправлять данные на клиент | + нет необходимости постоянно пересоединяться с сервером+ нет изменений на стороне сервера, поэтому работает на всех современных веб-серверах | - не поддерживается в IE (даже в IE10)  
- работает только в направлении сервер –> клиент (на сервер можно отправлять обычные Ajaxзапросы) |  
WebSockets | Новый протокол (ws:// и wss://), работающий поверх TCP на одном уровне с HTTP. Позволяет создавать двустороннее долгоживущее соединение с клиентом | + нет необходимости постоянно пересоединяться с сервером+ работает в двустороннем режиме | - поддерживается не во всех веб-серверах (IIS8)- поддерживается не во всех браузерах (в IE7-9, Android) |  
  
SignalR  
SignalR – это новая библиотека с открытым исходным кодом от Дэвида Фаулера и Дамьена Эдвардса из команды ASP.NET. Всё больше и больше приложений включают в себя компоненты реального времени, такие например, как чат или данные о биржевых котировках. В последнее время возникло множество новых фреймвоков, таких как socket.io и node.js для решения задач асинхронности и реального времени в веб. Вы уже слышали о таких технологиях, как веб-сокеты, вечные фреймы, длинный запросы и события, посылаемые сервером.  
SignalR – это библиотека для создания многопользовательских real-time ASP.NET (и не только) приложений. Она состоит из набора серверных и клиентских библиотек, и представляет собой абстракцию над целым набором транспортов. SignalR был придуман и реализован двумя разработчиками Microsoft: Damian Edwards и David Fowler.  
В целом, SignalR реализует следующие сценарии:  
1. клиент вызывает метод на сервере  
2. сервер вызывает метод на клиенте/клиентах  
3. передача состояния с клиента на сервер и обратно  
4. поддержка передачи сложных объектов (JSON сериализация)  
5. определение соединения, отсоединения и пересоединения клиентов  
6. обращение к клиентам извне хаба при помощи специального интерфейса (то есть любой код на сервере может оповестить клиенты о событии)  
7. асинхронные сценарии  
В официальной документации можно посмотреть более подробное описание всех API и их возможности, как на сервере, так и на клиентах:  
\* Hubs API  
\* PersistentConnection API  
\* JS API  
ASP.NET  
  
ASP.NET —технология создания веб-приложений и веб-сервисов от компании Майкрософт. Она является составной частью платформы Microsoft.NET и развитием более старой технологии Microsoft ASP. На данный момент последней версией этой технологии является ASP.NET 4.5.  
ASP.NET внешне во многом сохраняет схожесть с более старой технологией ASP, что позволяет разработчикам относительно легко перейти на ASP.NET. В то же время внутреннее устройство ASP.NET существенно отличается от ASP, поскольку она основана на платформе .NET и, следовательно, использует все новые возможности, предоставляемые этой платформой.  
ASP.NET не является платформой.  
Хотя ASP.NET берёт своё название от старой технологии Microsoft ASP, она значительно от неё отличается. Microsoft полностью перестроила ASP.NET, основываясь на Common Language Runtime (CLR), которая является основой всех приложений Microsoft .NET. Разработчики могут писать код для ASP.NET, используя практически любые языки программирования, входящие в комплект .NET Framework (C#, Visual Basic.NET и JScript .NET). ASP.NET имеет преимущество в скорости по сравнению со скриптовыми технологиями, так как при первом обращении код компилируется и помещается в специальный кэш, и впоследствии только исполняется, не требуя затрат времени на парсинг, оптимизацию, и т.д.  
  
Преимущества ASP.NET перед ASP  
  
Компилируемый код выполняется быстрее, большинство ошибок отлавливается ещё на стадии разработки  
Значительно улучшенная обработка ошибок времени выполнения, с использованием блоков try..catch  
Пользовательские элементы управления (controls) позволяют выделять часто используемые шаблоны, такие как меню сайта  
Использование метафор, уже применяющихся в Windows-приложениях, например, таких как элементы управления и события  
Расширяемый набор элементов управления и библиотек классов позволяет быстрее разрабатывать приложения  
ASP.NET опирается на многоязыковые возможности .NET, чтопозволяет писать код страниц на VB.NET, Delphi.NET, Visual C#, J# и т. д.  
Возможность кэширования всей страницы или её части для увеличения производительности  
Возможность кэширования данных, используемых на странице  
Возможность разделения визуальной части и бизнес-логики по разным файлам («code behind»)  
Расширяемая модель обработки запросов  
Расширенная событийная модель  
Расширяемая модель серверных элементов управления  
Наличие master-страниц для задания шаблонов оформления страниц  
Поддержка CRUD-операций при работе с таблицами через GridView  
Встроенная поддержка AJAX  
ASP.NET имеет преимущество в скорости по сравнению с другими технологиями, основанными на скриптах.  
Здесь можно привести определённые сравнения. Так, ASP — производная от Win32, XML и HTML; PHP — от XML, HTML, Java и CDI, тогда ASP.NET — от HTML и .NET(XML и XAML соответственно). При этом, если обычно Rich Media Application создают при помощи Flash, теперь это делается с помощью модуля Silverlight, так же через сам ASP.NET. ASP.NET — богатейшая среда для разработки и развёртывания веб-ресурсов. В ASP.NET можно работать с любым .NET языком, вплоть до Managed C++ и Visual Basic, что позволяет не задумываться о переходе на C#.  
IIS  
IIS (Internet Information Services, до версии 5.1 — Internet Information Server) — проприетарный набор серверов для нескольких служб Интернета от компании Майкрософт. IIS распространяется с операционными системами семейства Windows NT.  
Основным компонентом IIS является веб-сервер, который позволяет размещать в Интернете сайты. IIS поддерживает протоколы HTTP, HTTPS, FTP, POP3, SMTP, NNTP. По данным компании Netcraft на октябрь 2011 года, более 21 млн сайтов обслуживаются веб-сервером IIS, что составляет 12.46% от общего числа веб-сайтов.  
Основным компонентом IIS является веб-сервер — служба WWW (называемая также W3SVC), которая предоставляет клиентам доступ к сайтам по протоколам HTTP и, еслипроизведена настройка, HTTPS.  
Один сервер IIS может обслуживать несколько сайтов (IIS 6.0 и выше). Каждый сайт имеет следующие атрибуты:  
\* IP-адрес сайта;  
\* TCP-порт, на котором служба WWW ожидает подключений к данному сайту;  
\* Заголовок узла (Host header name) — значение заголовка Host запроса HTTP, указывающее обычно DNS-имя сайта.  
Таким образом, например, один сервер с одним IP-адресом может обслуживать на одном TCP-порту несколько сайтов. Для этого необходимо создать несколько DNS-записей, указывающих на IP-адрес сервера, и различать сайты по заголовкам узла.  
Для каждого сайта указывается домашний каталог — каталог в файловой системе сервера, соответствующий «корню» сайта.  
В IIS 6.0, доступном в составе систем Windows Server 2003, служба WWW претерпела серьёзные изменения. Был добавлен новый режим обработки запросов, называемый режимом изоляции рабочих процессов (англ. worker process isolation mode). В этом режиме все веб-приложения, обслуживаемые сервером, работают в разных процессах, что повышает стабильность и безопасность системы. Кроме того, для приёма запросов HTTP был создан новый драйвер http.sys, который работает в режиме ядра, что ускоряет обработку каждого запроса.  
Все запросы к статическому содержимому, не требующие исполнения скриптов, исполняются самим драйвером http.sys в ядре, что сближает веб-сервер IIS с серверами режима ядра.  
При этом запросы к динамическому содержимому исполняются рабочим процессом и загруженными в его адресное пространство модулями. С точки зрения пути исполнения запросов не существует центрального процесса, что повышает надежность в случае отказа, вызванного ошибкой в скрипте или ином модуле исполнения. Рабочие процессы автоматически перезапускаются при возникновении ошибок.  
Протокол SSL поддерживается отдельным процессом HTTP SSL, который служит мостом между протоколом TCP и драйвером http.sys.  
Веб-сервер IIS предоставляетнесколько способов разграничения доступа к сайтам и веб-приложениям. Служба WWW в составе IIS отличается от других веб-серверов тем, что функции обеспечения безопасности в ней тесно интегрированы с системой Windows NT, на основе которой она работает. В частности, чтобы получить доступ к защищённому ресурсу, посетитель должен ввести имя и пароль пользователя, существующего в системе Windows, на которой установлен IIS (или в домене Active Directory, если сервер принадлежит к домену). После этого пользователь работает с сайтом так же, как если бы он выполнил интерактивный вход в систему на сервере. К нему применяются установленные файловой системой NTFS разрешения на доступ к файлам и каталогам. Эта особенность IIS удобна для внутренних сайтов предприятий, однако практически неприменима для открытых сайтов Интернета, где невозможно создавать пользователя Windows для каждого зарегистрированного посетителя сайта. Поэтому в последнем случае разработчикам сайтов и веб-приложений обычно приходится использовать собственные механизмы ограничения доступа.  
Определённый пользователь Windows сопоставляется с каждым посетителем сайта даже в том случае, когда ограничение доступа не требуется. Этот режим называется режимом анонимного доступа. В этом случае посетитель представляется на сервере как специальный пользователь, имя которого обычно имеет формат IUSR\_xxxx (где xxxx — имя компьютера, на котором установлен IIS, в седьмой версии этот специальный пользователь не содержит имени компьютера, то есть просто IUSR). Этому пользователю должен быть разрешён доступ к ресурсам, которые открыты анонимным посетителям.  
Начиная с версии 6.0 служба WWW поддерживает следующие методы аутентификации, то есть определения личности пользователя по имени и паролю:  
\* Анонимная аутентификация (anonymous authentication) — определение личности пользователя не выполняется.  
\* Базовая аутентификация (basic authentication) — имя ипароль передаются по сети открытым текстом.  
\* Дайджест аутентификация (digest authentication) — пароль обрабатывается хеш-функцией перед отправкой по сети, что делает невозможным его прочтение в случае перехвата злоумышленником.  
\* Встроенная аутентификация Windows (integrated Windows authentication) — выполняется попытка входа на сервер с теми же учётными данными, под которыми работает браузер пользователя.  
\* Аутентификация для доступа к UNC-ресурсам (UNC authentication) — имя и пароль передаются удаленному серверу, на котором находится опубликованный в IIS UNC-ресурс, и удаленный сервер выполняет аутентификацию.  
\* Аутентификация с использованием .NET Passport (.NET Passport Authentication) (удалена в Windows Server 2008 и IIS 7.0) — для аутентификации используется служба .NET Passport.  
\* Аутентификация с использованием клиентского сертификата (certificate authentication) — для аутентификации пользователь должен предоставить SSL сертификат.  
Веб-сервер IIS поддерживает несколько различных технологий создания веб-приложений:  
\* ASP.NET — разработанная Microsoft технология; для IIS это — основное на сегодняшний день средство создания веб-приложений и веб-служб. IIS 6.0 поставляется вместе с операционными системами, в которые также изначально входит .NET Framework, так что поддержка ASP.NET как будто уже встроена в IIS 6.0; для более ранних версий необходимо отдельно загрузить и установить .NET Framework.  
\* ASP — предшествовавшая ASP.NET технология создания динамических веб-страниц на основе сценариев. Входит в поставку IIS начиная с версии 3.0.  
\* CGI — стандартная межплатформенная низкоуровневая технология создания динамических веб-страниц.  
\* FastCGI — клиент-серверный протокол взаимодействия веб-сервера и приложения.  
\* ISAPI — низкоуровневая технология, аналогичная интерфейсу модулей Apache, предоставляющая полный доступ ко всем возможностям IIS, возможностьразработки веб-приложений в машинном коде и возможность переопределения части функций IIS и добавления к нему функций, как связанных с генерацией контента, так и не связанных с этим. Подсистема исполнения скриптов ASP и подсистема ASP.NET выполнены как модули ISAPI.  
\* SSI — включение в одни страницы текста из других страниц. Строго говоря, веб-приложением не является, поскольку IIS поддерживает лишь ограниченный набор возможностей и без того малофункционального SSI. В частности, IIS5 поддерживает только статическое включение и игнорирует команды условного ветвления.  
Сам сервер поддерживает только CGI, FastCGI, ISAPI и SSI. Все остальные технологии являются надстройками, работающими через CGI, FastCGI или ISAPI.  
При помощи CGI приложения для IIS могут разрабатываться на основе практически любых, в том числе сторонних, инструментов, допускающих запись в стандартный поток вывода и чтение переменных среды — Perl, C/С++ и даже средствами интерпретатора командной строки Cmd.exe.  
Технология ISAPI позволяет, с одной стороны, создавать специальные приложения для IIS, требующие особенно тесного взаимодействия с механизмом сервера, а с другой стороны является удобной платформой для организации эффективного взаимодействия IIS с другими технологиями разработки веб-приложений — например, PHP и Perl.  
  
БАЗА ДАННЫХ  
  
MS SQL SERVER  
  
Microsoft SQL Server — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями.  
SQL Server включает несколько технологий управления данными и анализа данных.  
Database Engine  
Компонент Database Engine представляет собой основную службу для хранения, обработки и обеспечения безопасности данных. Этот компонент обеспечивает управляемый доступ кресурсам и быструю обработку транзакций, что позволяет использовать его даже в самых требовательных корпоративных приложениях обработки данных. Кроме того, компонент Database Engine предоставляет разносторонние средства поддержания высокого уровня доступности.  
Службы Data Quality Services  
Службы SQL Server Data Quality Services (DQS) являются решением для очистки данных на основе знаний. Службы DQS позволяют создать базу знаний, а затем выполнить в ней исправление данных и удаление дубликатов с помощью как автоматизированных, так и интерактивных средств. Можно использовать службы справочных данных на основе облачных вычислений, а также создавать решения по управлению данными, где службы DQS будут интегрированы со службами SQL Server Integration Services и Master Data Services.  
Службы Analysis Services  
Службы Службы Analysis Services — это платформа аналитических данных и набор средств для бизнес-аналитики на личном уровне, уровне рабочей группы и организации. Серверный и клиентский конструкторы поддерживают стандартные решения OLAP, новые решения для создания табличных моделей, а также самостоятельную аналитику и совместную работу с помощью PowerPivot, Excel и среды SharePoint Server. Службы Analysis Services также включают интеллектуальный анализ данных, который позволяет выявлять закономерности и связи на основе больших объемов данных.  
Integration Services  
Службы Integration Services представляют собой платформу для создания высокопроизводительных решений по интеграции данных, в том числе пакетов для хранения данных, обеспечивающих извлечение, преобразование и загрузку данных.  
Master Data Services  
Master Data Services — это решение SQL Server для управления основными данными. Решение, построенное на основе Master Data Services, позволяет обеспечить правильность информации, используемой для построения отчетов и выполнения анализа. С помощью Master Data Services можно создатьцентральный репозиторий основных данных и поддерживать запись этих данных по мере их изменения, защищенную и доступную для аудита.  
Репликация  
Репликация представляет собой набор технологий копирования и распространения данных и объектов баз данных между базами данных, а также синхронизации баз данных для поддержания согласованности. Благодаря репликации данные можно размещать в различных местах, обеспечивая возможность доступа к ним удаленных и мобильных пользователей по локальным или глобальным сетям, посредством коммутируемых и беспроводных соединений, а также через Интернет.  
Службы Reporting Services  
Службы Reporting Services предлагают средства создания корпоративных отчетов с поддержкой веб-интерфейса, которые позволяют включать в отчеты данные из различных источников, публиковать отчеты в разнообразных форматах, а также централизованно управлять безопасностью и подписками.

**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[8], инструменты для работы с Git[9], подсветку синтаксиса, IntelliSense[10] 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 кодом[11][12], но готовые сборки распространяются под проприетарной лицензией[13].

Visual Studio Code основан на Electron[14] и реализуется через веб-редактор Monaco[15], разработанный для Visual Studio Online.

Visual Studio Code был анонсирован 29 апреля 2015 года компанией Microsoft на конференции Build, и вскоре была выпущена бета-версия[16].

18 ноября 2015 года Visual Studio Code был выпущен под лицензией MIT, а исходный код был опубликован на GitHub. Анонсирована поддержка расширений[11].

14 апреля 2016 года Visual Studio Code вышел из стадии бета-тестирования[17].

**React** (иногда **React.js** или **ReactJS**) — JavaScript-библиотека[4] с открытым исходным кодом для разработки пользовательских интерфейсов.

React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций[5][6][7].

React может использоваться для разработки одностраничных и мобильных приложений. Его цель — предоставить высокую скорость, простоту и масштабируемость. В качестве библиотеки для разработки пользовательских интерфейсов React часто используется с другими библиотеками, такими как MobX, Redux и GraphQL[8].

React был создан Джорданом Валке, разработчиком программного обеспечения из Facebook. На него оказал влияние XHP — компонентный HTML-фреймворк для PHP[9]. Впервые React использовался в новостной ленте Facebook в 2011 году и позже в ленте Instagram в 2012 году[10]. Исходный код React открыт в мае 2013 года на конференции «JSConf US».

React Native анонсирован на конференции Facebook «React.js Conf» в феврале 2015 года, а исходный код открыт в марте 2015 года. Он позволяет разрабатывать нативные Android-, iOS- и UWP-приложения с использованием React.

18 апреля 2017 года Facebook анонсировал React Fiber, переписанную и оптимизированную версию React[11]. React Fiber станет основой разработки всех будущих функций и улучшений[12].

**Redux** — библиотека для JavaScript[2] с открытым исходным кодом, предназначенная для управления состоянием приложения.[3] Чаще всего используется в связке с React или Angular для разработки клиентской части.[4] Содержит ряд инструментов, позволяющих значительно упростить передачу данных хранилища через контекст. Создатели: Даниил Абрамов и Эндрю Кларк.

Библиотека была создана в 2015 году Даниилом Абрамовым и Эндрю Кларком.[6] Первую версию Абрамов создал во время подготовки к конференции React Europe. По словам Абрамова, он пытался создать реализацию идеи Flux с другой логикой. Абрамов был потрясён сходством между редьюсерами и Flux, так что он попытался совместить их.[7]

Абрамов обратился к Кларку (создателю Flummox, одной из реализаций Flux), и совместно они разработали Redux. Как утверждает Абрамов, именно благодаря Кларку был реализован комплекс экосистемных средств, логичный API, а также точки расширения, такие как связующее программное обеспечение и усилители хранилища.[8]

**npm** (англ. ***N****ode****P****ackage****M****anager*) — менеджер пакетов, входящий в состав Node.js.

Установка пакета производится при помощи команды:

npm install <packagename>

Все доступные для установки пакеты и их краткое описание:

npm search

Этой же командой можно производить выборочный поиск пакетов.

**Node** или **Node.js** — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, написанный на C++, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения (при помощи NW.js, AppJS или Electron для Linux, Windows и macOS) и даже программировать микроконтроллеры (например, tessel, low.js и espruino). В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом.

В 1996 году в компании Netscape были попытки создания серверного JavaScript (Server-side JavaScript — SSJS)[6][7], однако технология не получила распространения.

Райан Дал, создатель Node.js

Node.js разработал Райан Даль в 2009 году после двух лет экспериментирования над созданием серверных веб-компонентов. В ходе своих исследований он пришёл к выводу, что вместо традиционной модели параллелизма на основе потоков следует обратиться к событийно-ориентированным системам. Эта модель была выбрана из-за простоты, низких накладных расходов (по сравнению с идеологией «один поток на каждое соединение») и быстродействия. Целью Node является предложить «простой способ построения масштабируемых сетевых серверов».

Разработка Node.js спонсируется компанией Joyent.

В декабре 2014 года был создан форк io.js.

В мае 2015 года было принято решение о слиянии io.js и Node.js и дальнейшем развитии под эгидой Node.js Foundation[8].

8 сентября 2015 года вышел Node.js v4.0.0 как результат слияния Node.js v0.12.7 и io.js v3.3.0[9][10].

Важными событиями в развитии платформы стало появление Atomics и SharedArrayBuffer в Node.js 9, а также worker\_threads в Node.js 10.5 (и существенное развитие в Node.js 12)[11]. Это позволило создавать многопоточные параллельные приложения, реализовывать примитивы параллельного программирования и работать с разделяемой памятью[12].

**Express.js**, или просто Express, фреймворк web-приложений для Node.js, реализованный как свободное и открытое программное обеспечение под лицензией MIT. Он спроектирован для создания веб-приложений и API[2]. Де-факто является стандартным каркасом для Node.js. Автор фреймворка, TJ Holowaychuk, описывает его как созданный на основе написанного на языке Ruby каркаса Sinatra, подразумевая, что он минималистичен и включает большое число подключаемых плагинов. Express может являться backend'ом для программного стека MEAN, вместе с базой данных MongoDB и каркасом Vue.js, React или AngularJS для frontend'а.

**MongoDB** — документоориентированная система управления базами данных, не требующая описания схемы таблиц. Считается одним из классических примеров NoSQL-систем, использует JSON-подобные документы и схему базы данных. Написана на языке C++. Применяется в веб-разработке, в частности, в рамках JavaScript-ориентированного стека MEAN.

Система поддерживает ad-hoc-запросы: они могут возвращать конкретные поля документов и пользовательские JavaScript-функции. Поддерживается поиск по регулярным выражениям. Также можно настроить запрос на возвращение случайного набора результатов.[1].

Имеется поддержка индексов.

Система может работать с набором реплик[2], то есть содержать две или более копии данных на различных узлах. Каждый экземпляр набора реплик может в любой момент выступать в роли основной или вспомогательной реплики. Все операции записи и чтения по умолчанию осуществляются с основной репликой. Вспомогательные реплики поддерживают в актуальном состоянии копии данных. В случае, когда основная реплика дает сбой, набор реплик проводит выбор, которая из реплик должна стать основной. Второстепенные реплики могут дополнительно являться источником для операций чтения.

Система масштабируется горизонтально, используя технику сегментирования (англ. *sharding*) объектов баз данных — распределение их частей по различным узлам кластера. Администратор выбирает ключ сегментирования, который определяет, по какому критерию данные будут разнесены по узлам (в зависимости от значений хэша ключа сегментирования). Благодаря тому, что каждый узел кластера может принимать запросы, обеспечивается балансировка нагрузки.

Система может быть использована в качестве файлового хранилища с балансировкой нагрузки и репликацией данных (функция ***Grid File System***[3]; поставляется вместе с драйверами MongoDB). Предоставляются программные средства для работы с файлами и их содержимым. GridFS используется в плагинах для Nginx[4] и lighttpd.[5] GridFS разделяет файл на части и хранит каждую часть как отдельный документ.[6].

Может работать в соответствии с парадигмой MapReduce. Во фреймворке для агрегации есть аналог SQL-выражения GROUP BY. Операторы агрегации могут быть связаны в конвейер подобно UNIX-конвейрам. Фреймворк так же имеет оператор $lookup для связки документов при выгрузке и статистические операции такие как среднеквадратическое отклонение.

Поддерживается JavaScript в запросах, функциях агрегации (например, в MapReduce).

Поддерживает коллекции с фиксированным размером. Такие коллекции сохраняют порядок вставки и по достижении заданного размера ведут себя как кольцевой буфер.

В июне 2018 года (в версии 4.0) добавлена поддержка транзакций, удовлетворяющих требованиям ACID[7].

Запись записи в MongoDB с Robomongo 0.8.5.

Есть официальные драйверы для основных языков программирования (Си, C++, C#, Go, Java, Node.js, Perl, PHP, Python, Ruby, Rust, Scala, Swift). Существует также большое количество неофициальных или поддерживаемых сообществом драйверов для других языков программирования и фреймворков.

Основным интерфейсом к базе данных была командная оболочка «mongo». С версии MongoDB 3.2 в качестве графической оболочки поставляется «MongoDB Compass». Существуют продукты и сторонние проекты, которые предлагают инструменты с GUI для администрирования и просмотра данных.

**Socket.IO** — JavaScript-библиотека для веб-приложений и обмена данными в реальном времени. Состоит из двух частей: клиентской, которая запускается в браузере и серверной для node.js. Оба компонента имеют похожее API. Подобно node.js, Socket.IO событийно-ориентированная.

Socket.IO главным образом использует протокол WebSocket, но если нужно, использует другие технологии, например Flash Socket, AJAX Long Polling, AJAX Multipart Stream[2], предоставляя тот же самый интерфейс. Помимо того, что Socket.IO может быть использована как оболочка для WebSocket, она содержит много других функций, включая вещание на несколько сокетов, хранение данных, связанных с каждым клиентом, и асинхронный ввод/вывод.

Может быть установлена через npm (node package manager)[3].

С помощью Socket.IO можно реализовать аналитику в реальном времени, многопользовательские игры, обмен мгновенными сообщениями и совместную работу с документами в реальном времени.

Socket.IO довольно популярен, его используют Microsoft**,** Yammer**,** Zendesk**,** Trello и многие другие организации для создания систем реального времени.

Socket.IO работает на основе событий.

**Заключение**

Используя приобретенные навыки, сделал 2 To Do List’а на ReactJS и NodeJS, Quiz на ReactJS, интернет магазин Pizza на ReactJS, Сокращение ссылок на NodeJs, Прогноз погоды на ReactJS, онлайн-чат на ReactJS ,NodeJS, Socket.io.

Все эти работы можно посмотреть на моём GitHub’e: <https://github.com/Nurikon4/>