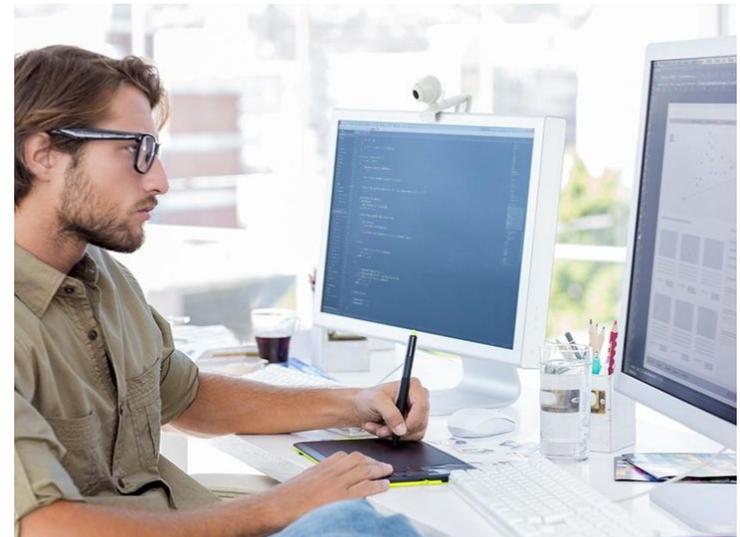


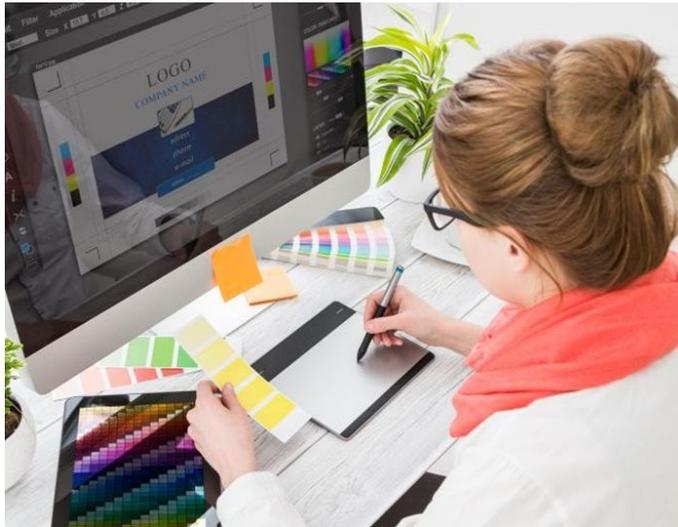
Актуальное техническое описание по компетенции «Веб-дизайн и разработка»

Описание профессиональной компетенции

- **Веб-разработчик является динамичной, постоянно меняющейся профессией, сферой деятельности которой является создание и функционирования веб-сайтов.**
- **Веб-разработчики используют для создания веб-сайтов специальные программы и языки программирования и разметки, которые связывают ссылки на различные веб-страницы, другие веб-сайты, графические элементы, текст и фото в единый функциональный и удобный информационный продукт.**



Описание профессиональной компетенции



- Компьютерные программы, заготовки и открытые электронные библиотеки используются в качестве технической базы.
- Чтобы пробудить интерес у посетителей сайта, последние обязаны изучать новые техники и технологии производства сайтов и использовать их при решении оригинальных задач.

Описание профессиональной компетенции

- Веб-разработчик осведомлен как в области технологий, так и в графическом дизайне.
- На сайтах веб-технология используется в том числе для автоматизации функций и помощи в управлении контентом.
- Творческие способности нужны веб-разработчикам при подборе цветов, шрифтов и графики, для поддержки эффективной рабочей коммуникации с профессиональными дизайнерами, а также при разработке структуры сайта.



Описание профессиональной компетенции

- Веб-разработчик так же обязан знать:
 - основы проектной работы;
 - продукцию, которой посвящен контент сайта;
 - технологии и методы программирования на стороне сервера и клиента;
 - основы программной архитектуры и базах данных для хранения информации и организации сложных веб-сервисов;
 - основы управления сайтом.
- Совместимость конечного продукта с современными версиями наиболее распространенных веб-браузеров, программам и устройств обязательна.

Типы веб-разработчик

- Front-end разработчик
- Back-end разработчик
- Full-stack разработчик

Front-end разработчик

- Эти программисты специализируются на интерфейсах. Все кнопки на экране, выпадающие меню, и весь внешний вид сайта или приложения – их сфера деятельности. Т.е. Front-end – это работа над динамикой интерактивных интерфейсов. Кроме того, при отсутствии верстальщика от Front-end разработчика могут потребовать работу с HTML-кодом (верстку шаблонов).
- Необходимо знать технологии:
 - HTML и CSS;
 - различные фреймворки;
 - препроцессоры (SASS/LESS);
 - JavaScript вместе с библиотеками;
 - системы отладки и сборки программ.

Back-end разработчик

- Программист Back-end занимается созданием программно-административной частью проекта – он создает архитектуру, продумывает логику, пишет исполняемый код.
- Такие специалисты работают в крупной или средней IT-компании. Вообще, разделение Front-end и Back-end – признак солидного проекта.
- Может пользоваться самыми разными языками программирования, фреймворками и библиотеками. Самые популярные:
 - PHP;
 - Python;
 - Symfony;
 - Yii;
 - Laravel.
- Также обязательно нужно изучить JavaScript, MySQL, PostgreSQL и NoSQL.

Full-stack разработчик

- Этот тип программистов объединяет в себе две описанные выше категории, обычно работает над WEB-проектами. Т.е. это тот самый программист, который сможет создать полноценный сайт «под ключ» самостоятельно.
- Эти специалисты очень нужны небольшим веб-студиям, а также на фрилансе. Объединяет в себе навыки и знания, описанные в предыдущих двух пунктах. Но оба направления обычно знает не столь глубоко, как в случае выбора узкой специализации. Впрочем, ему это и не нужно. Его ниша – несложные и не самые дорогие проекты.

Спецификация стандарта Ворлдскиллс по компетенции «Веб-дизайн и разработка»

Общие сведения о спецификации стандартов Worldskills

- WSSS определяет знание, понимание и конкретные компетенции, которые лежат в основе лучших международных практик технического и профессионального уровня выполнения работы.
- Она должна отражать коллективное общее понимание того, что соответствующая рабочая специальность или профессия представляет для промышленности и бизнеса.

Разделы Worldskills standard specification (wsss)

Организац
ия работы
и



Коммуникационн
ые и
м е



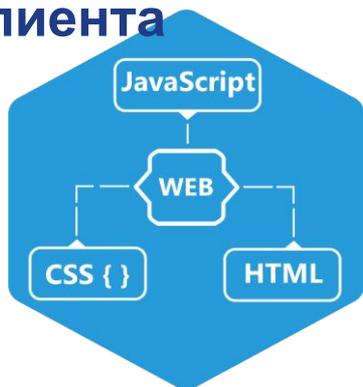
Графический
дизайн веб-
страниц



Верстк
а
страни



Программирован
ие на стороне
клиента



Программирован
ие на стороне
сервера



Системы
управления
контентом



Организация работы и управление

Специалист должен знать и понимать:

- принципы и практики, которые позволяют продуктивно работать в команде;**
- аспекты систем, которые позволяют повысить продуктивность и выработать оптимальную стратегию;**
- как проявить инициативу и предприимчивость в целях выявления, анализа и оценивания информации из различных источников.**

Организация работы и управление

Специалист должен уметь:

- решать распространенные задачи веб-дизайна и разработки кода;
- учитывать временные ограничения и сроки;
- производить отладку кода программ и находить ошибки;
- использовать компьютер или устройство и целый ряд программных пакетов;
- применять исследовательские приемы и навыки, чтобы быть в курсе последних отраслевых решений;
- планировать график рабочего дня с учетом требований;
- включать ссылки на изображения, шрифты и др. файлы при архивации данных;
- использовать систему контроля версий.

Коммуникационные и межличностные навыки

Специалист должен знать и понимать:

- как решить проблемы в общении, в том числе выявление проблемы, ее исследование, анализ, решение, макетирование, пользовательское тестирование и оценка результатов;
- принципы, лежащие в основе сбора и представления информации;
- дизайн-концепции и техники, в том числе черновое макетирование страниц (wireframing), объектно-событийное моделирование (storyboarding) и создание блок-схем;
- английский язык в рамках чтения и понимания официальной технической документации по используемым технологиями и языкам программирования.

Коммуникационные и межличностные навыки

Специалист должен уметь:

- представить продукт, который отвечает требованиям клиента и спецификации;
- собирать, анализировать и оценивать информацию;
- использовать навыки грамотности для толкования стандартов и требований;
- планировать и организовывать общение с клиентом;
- Критиковать свои проекты и идеи.

Графический дизайн веб-страниц

Специалист должен знать и понимать:

- структуру и общепринятые элементы веб-страниц различных видов и назначений;
- вопросы, связанные с когнитивными, социальными, культурными, технологическими и экономическими условиями при разработке дизайна;
- как создавать и оптимизировать графику для сети Интернет;
- как создавать дизайн по предоставляемым инструкциям и спецификациям;
- какие умения и навыки необходимы для выбора цвета, типографики и композиции;
- принципы и методы адаптации графики для использования ее на веб-сайтах;
- правила поддержания фирменного стиля, бренда и стилевых инструкций;
- ограничения, которые накладывают мобильные устройства и разрешения экранов при использовании их для просмотра веб-сайтов;
- принципы построения эстетичного и креативного дизайна;
- современные стили и тенденции дизайна.

Графический дизайн веб-страниц

Специалист должен уметь:

- создавать и анализировать разработанные визуальные ответы на поставленные вопросы, в том числе об иерархии, типографике, эстетике и композиции;
- создавать, использовать и оптимизировать изображения для веб-сайтов;
- анализировать целевой рынок и продукцию, которую продвигает, используя дизайн;
- выбирать дизайнерское решение, которое будет наиболее подходящим для целевого рынка;
- принимать во внимание влияние каждого элемента, который добавляется в проект во время разработки дизайна;
- использовать все требуемые элементы при разработке дизайна;
- учитывать существующие правила корпоративного стиля;
- создавать «отзывчивый» дизайн, который будет отображаться корректно на различных устройствах и при разных разрешениях;
- придерживаться оригинальной концепции дизайна проекта и улучшать его визуальную привлекательность;
- превращать идею в эстетичный и креативный дизайн.

Верстка страниц

Специалист должен знать и понимать:

- методы обеспечения доступа к страницам веб-сайтов аудитории с ограниченными возможностями;
- world Wide Web Consortium (W3C) стандарты HTML и CSS;
- методы верстки веб-сайтов и их стандартную структуру;
- web accessibility initiative (WAI);
- как применять соответствующие CSS правила и селекторы для получения ожидаемого результата;
- лучшие практики для Search Engine Optimization (SEO) и интернет-маркетинга;
- как встраивать и интегрировать анимацию, аудио, видео и другую мультимедийную информацию, управлять поведением остальных элементов на странице.

Верстка страниц

Специалист должен уметь:

- **создавать html-страницы сайта на основе предоставленных графических макетов их дизайна;**
- **корректно использовать CSS для обеспечения единого дизайна в разных браузерах;**
- **создавать адаптивные веб-страницы, которые способны оставаться функциональными на различных устройствах при разных разрешениях;**
- **создавать веб-сайты полностью соответствующие текущим стандартам W3C (<http://www.w3.org>);**
- **создавать и модифицировать сайты с учетом Search Engine Optimization.**

Программирование на стороне клиента

Специалист должен знать и понимать:

- возможности клиентского языка программирования JavaScript;
- как разрабатывать код с использованием открытых библиотек.

Специалист должен уметь:

- разрабатывать анимацию для веб-сайта для повышения его доступности и визуальной привлекательности;
- создавать и модифицировать JavaScript код для улучшения функциональности и интерактивности сайта;
- применять открытые библиотеки.

Программирование на стороне сервера

Специалист должен знать и понимать:

- как разрабатывать PHP, Python код на процедурном и объектно-ориентированном уровнях;
- как использовать открытые библиотеки и Фреймворки;
- распространенные модели организации и хранения данных и реализацию их с применением MySQL;
- FTP (File Transfer Protocol), особенности использования его на стороне сервера и клиента, а так же необходимое для этого программное обеспечение;
- как разрабатывать веб-сервисы с применением PHP, Python, XML (Extensible Markup Language) и JSON;
- различные методы программирования;
- как разрабатывать программный код в соответствии с паттернами (например, MVC (Model View Controller));
- Как разрабатывать безопасное веб-приложение.

Программирование на стороне сервера

Специалист должен уметь:

- создавать библиотеки и модули для выполнения повторяющихся задач;
- разрабатывать веб-приложения с доступом к базе данных MySQL и веб-сервисы по требованиям клиента;
- интерпретировать ER (Entity-Relationship) диаграммы в функционирующую базу данных;
- создавать SQL (Structured Query Language) запросы, используя корректный синтаксис (классический и PDO (PHP Data Object));
- обеспечивать безопасность (устойчивость веб-приложения к атакам и взлому);
- интегрировать существующий программный код с API (Application Programming Interfaces), библиотеками и фреймворками;
- разрабатывать объектно-ориентированный программный код.

Проектирование и архитектура

Специалист должен знать и понимать:

- **этапы жизненного цикла информационного продукта.**
- **принципы проектирования цифрового продукта и информационных систем на основе заданных целей и назначения;**
- **современные нотации и стандарты моделирования цифровых продуктов и информационных систем, в том числе UML-2;**
- **принципы проектирования информационных моделей баз данных и их оптимизации;**
- **этапы проектирования в различных методологиях управления созданием информационных продуктов.**
- **способы формального выбора альтернативных решений.**
- **Основы юзабилити, дизайна и восприятия графической информации.**

Проектирование и архитектура

Специалист должен уметь:

- планировать работу над проектом, визуально представлять план с помощью современных программных средств;
- формировать модели информационных систем на основе требований и функций и работать с ними;
- исследовать предметную область, анализировать аналоги и элементы информационной системы;
- выбирать технические средства и технологические платформы с учетом требований и ограничений;
- распределять функциональные характеристики по различным компонентам информационных систем;
- выбирать или разрабатывать протоколы и API для взаимодействия компонент системы;
- анализировать уровень защищенности цифровых решений.
- формировать набор функциональных, информационных и иных элементов на основе списка функций;
- располагать элементы на странице (форме) с учетом особенностей дальнейшего использования цифрового продукта (системы) или технологии и целевой аудитории;
- разрабатывать дизайн страницы или формы цифрового продукта.