Frontend разработка включает в себя работу с такими языками программирования как html, css, javascript, Backend разработка – python, php, ruby, Java, C#.

Html – язык гипертекстовой разметки, это основа любого сайта. Он обеспечивает базовую структуру страницы и включает такие важные элементы как заголовки, абзацы, ссылки и т.д.

Css – каскадные таблицы стилей, определяет цвет, вид, стиль элементов. Применение стилей помогает выделить и разнообразить страницу делая ее более привлекательной и красивее.

Javascript – этот язык помогает добавить странице динамичность и интерактивность.

Ruby –  это объектно-ориентированный язык программирования, в котором особое внимание уделяется простоте и производительности.

Php – популярный язык сценариев на стороне сервера, известный своей простотой и удобством использования. На нем написана значительная часть веб-страниц, включая крупные платформы.

Python – это универсальный язык программирования высокого уровня.

Java – платформонезависимый объектно-ориентированный язык программирования, известный своей производительностью, совместимостью и универсальностью. Он широко используется предприятиями для разработки крупномасштабных веб-приложений

C# -  мощный современный язык программирования, разработанный компанией Microsoft. Это универсальный язык, позволяющий создавать сложные веб-приложения, игры и многое другое.

Веб-приложения - это программное обеспечение, которое работает на сервере и обслуживает запросы от клиентских браузеров. Архитектура веб-приложений - это структура и организация приложения, которая определяет, как компоненты приложения связаны между собой и какие принципы необходимо соблюдать при проектировании.

**Компоненты веб-приложений. Веб-приложения состоят из трех главных компонентов:**

**1. Клиентский интерфейс**

Клиентский интерфейс - это пользовательский интерфейс, который расположен на стороне клиента. Это может быть HTML, CSS, JavaScript или другие технологии, которые использовались для создания интерфейса.

**2. Серверный интерфейс**

Серверный интерфейс - это часть приложения, которая расположена на сервере. Он обрабатывает запросы от клиентского интерфейса и взаимодействует с базой данных.

**3. База данных**

База данных хранит данные приложения. База данных может быть локальной или удаленной.

**Архитектура веб-приложений**

Существует несколько подходов к архитектуре веб-приложений, но одним из самых распространенных является Model-View-Controller (MVC).

**1. Model-View-Controller (MVC)**

MVC - это паттерн проектирования, который разделяет приложение на три основных компонента: модель (Model), представление (View) и контроллер (Controller). Модель отвечает за управление данными, представление - за отображение этих данных, а контроллер - за управление потоком данных между моделью и представлением.

**2. Другие подходы**

Некоторые другие подходы к архитектуре веб-приложений включают Component-Based Architecture (CBA), Service-Oriented Architecture (SOA) и Domain-Driven Design (DDD).

**Принципы проектирования**

Проектирование веб-приложений включает несколько принципов, которые должны быть соблюдены, чтобы создать эффективное и стабильное приложение. Ниже перечислены наиболее важные принципы.

**1. Принцип единственной ответственности**

Каждый компонент должен иметь только одну ответственность.

**2. Принцип открытости / закрытости**

Код должен быть открыт для расширения, но закрыт для изменения. Это означает, что если необходимо внести изменения, они должны быть выполнены без изменения уже существующего кода.

**3. Принцип разделения интерфейсов**

Приложение должно быть разделено на логические части, которые не зависят друг от друга.

**4. Принцип инверсии зависимостей**

Зависимости между компонентами должны быть определены на уровне абстракции, а не на уровне реализации.

**5. Принцип композиции**

Компоненты должны быть построены из более мелких компонентов, что позволяет легко добавлять, изменять или удалять функциональность.