**Протоколы**

**HTTP** (HyperText Transfer Protocol)— широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов (то есть документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам). Обеспечивает связь между сервером и компьютером (устройством) пользователя.

**HTTPS** (HyperText Transfer Protocol Secure) – Оно реализует упаковку передаваемых данных в криптографический протокол. HTTPS широко используется для защиты информации от перехвата, а также, как правило, обеспечивает ОТНОСИТЕЛЬНУЮ защиту от атак — в том случае.

**URI, URL, URN**

URL - Uniform Resource Locator (унифицированный определитель местонахождения ресурса)

URN - Unifrorm Resource Name (унифицированное имя ресурса)

URI - Uniform Resource Identifier (унифицированный идентификатор ресурса)

**URL**: синоним термина веб-адрес. URL определяет местонахождение ресурса в сети и способ его (ресурса) извлечения. Это позволяет нам полностью узнать: как, кому и где можно достать требуемый ресурс, вводя понятия схемы, данных авторизации и местонахождения.

**HTML** (HyperText Markup Language) – это язык гипертекстовой разметки

**Языки WEB программирования**

**CSS -**

**JavaScript –**

**PHP** (Hypertext Preprocessor ) **-** этот язык был создан специально для интернета и позволял органически соединить HTML и синтаксически близкий к C язык программирования.

**Веб-приложения**

Веб-приложения можно разделить на несколько типов, в зависимости от разных сочетаний его основных составляющих:

1. **Backend** (бэкенд или серверная часть приложения) работает на удаленном компьютере, который может находиться где угодно. Она может быть написана на разных языках программирования. Если создавать приложение используя только серверную часть, то в результате любых переходов между разделами, отправок форм, обновления данных, сервером будет генерироваться новый HTML-файл и страница в браузере будет перезагружаться.

2. **Frontend** (фронтенд или клиентская часть приложения) выполняется в браузере пользователя. Эта часть написана на языке программирования Javascript. Приложение может состоять только из клиентской части, если не требуется хранить данные пользователя дольше одной сессии. Это могут быть, например, фоторедакторы или простые игрушки.

3. **Single page application** (SPA или одностраничное приложение). Более интересный вариант, когда используются и бэкенд и фронтенд. С помощью их взаимодействия можно создать приложение, которое будет работать совсем без перезагрузок страницы в браузере. Или в упрощенном варианте, когда переходы между разделами вызывают перезагрузки, но любые действия в разделе обходятся без них.

**Фронтенд** — все, что браузер может читать, выводить на экран и / или запускать. То есть это HTML, CSS и JavaScript.

**Бэкенд** — все, что работает на сервере, то есть «не в браузере» или «на компьютере, подсоединенном к сети (обычно к Интернету), который отвечает на сообщения от других компьютеров».

**Фреймворки**

**Фреймворки -** По сути это такие «заготовочки» - шаблоны, чтобы ускорить процесс разработки. Они позволяют вместо написания функционала заново сделать настройки (те самый минимальные изменения) и получить готовый продукт. Также фреймворки дают возможность подключаться к различным типам СУБД (Систе́ма управле́ния ба́зами да́нных) без погружения в специфику организации инфраструктуры, создавать элементы интерфейса и ускорять процесс верстки