Module 2 Assignment:

1. Take note of at least one technology, tool, or term, with which you are unfamiliar, from each listing (10).
2. Search online for a definition or description of that technology or term. Try to understand what role it plays in the context of how web applications were described in that chapter.
3. Open a Word or Text file and paste in your description for each of the technologies you researched.

Website functionalities

Website Designer

Front-end Developer – client-side

Back-end Developer – DBs or server side

Full Stack Developer – works on both Client and Server side and sometimes even a design. Mastery over different layers is in Hight demand.

Web Developer – development and analysis.

**Apache Server -** is free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

It is built for Unix-like systems as well as Windows. It is the most widely used web server software stimated to serve 46% of all active websites and 43% of the top million websites. Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support Perl, Python, Tcl and PHP

**PHP -** (recursive acronym for **PHP**: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. SERVER-SIDE Scripting

**Ajax -** is a client-side script that communicates to and from a server/database without the need for a postback or a complete page refresh. The best definition I've read for **Ajax** is “the method of exchanging data with a server, and updating parts of a web page – without reloading the entire page. **JAX** typically involves sending HTTP requests from client to server and processing the server's response, without reloading the entire page. (Asynchronously). ... **Ajax** stands for **Asynchronous Javascript And Xml**. The only part of **Ajax** that you need is the XMLHttpRequest object from javascript.

**React.js / -** is a declarative, efficient, and flexible **JavaScript** library for building user interfaces. ... Your components tell **React** what you want to render – then **React** will efficiently update and render just the right components when your data changes. Here, Shopping List is a **React** component class, or **React** component type.

**Redux -** Redux is one particular implementation of a **subset** of flux concepts. Redux is astonishingly simple. It uses a function called a reducer (a name derived from the JavaScript reduce method) that takes two parameters: An action, and a next state. The reducer has access to the current (soon to be previous) state, applies the given action to that state, and returns the desired next state.

**JQuery -** is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. The purpose of jQuery is to make it much easier to use JavaScript on your website. jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

**Restful API -** REpresentational State Transfer is an application program interface ([API](http://searchexchange.techtarget.com/definition/application-program-interface)) that uses [HTTP](http://searchwindevelopment.techtarget.com/definition/HTTP) requests to GET, PUT, POST and DELETE data. This Technology is generally preferred to the more robust Simple Object Access Protocol (SOAP) technology because REST leverages less bandwidth, making it more suitable for internet usage. A RESTful API breaks down a [transaction](http://searchcio.techtarget.com/definition/transaction) to create a series of small modules. Each [module](http://searchenterpriselinux.techtarget.com/definition/module) addresses a particular underlying part of the transaction. This modularity provides developers with a lot of flexibility, but it can be challenging for developers to design from scratch.

**JS Frameworks -** A JavaScript framework is an application framework written in JavaScript. It differs from a JavaScript library in its control flow: a library offers functions to be called by its parent code, whereas a framework defines the entire application design. JS framework is particularly helpful for applications where much of the **business logic** is going to take place on the **client-side** – routing, templating, first-pass model validation, table building and pagination – pretty much whatever you might have used the server for in the past, but now without the latency and overhead that additional HTTP calls would have incurred.

**Node.js -** is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. **Node**.**js** uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. Here is how Node.js handles a file request:

1. Sends the task to the computer's file system.
2. Ready to handle the next request.
3. When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the waiting, and simply continues with the next request.

Node.js runs single-threaded, non-blocking, asynchronously programming, which is very memory efficient.

* Node.js can generate dynamic page content
* Node.js can create, open, read, write, delete, and close files on the server
* Node.js can collect form data
* Node.js can add, delete, modify data in your database

What is a Node.js File?

* Node.js files contain tasks that will be executed on certain events
* A typical event is someone trying to access a port on the server
* Node.js files must be initiated on the server before having any effect
* Node.js files have extension ".js"
* **Ruby on Rails -** Node.js can generate dynamic page content
* Node.js can create, open, read, write, delete, and close files on the server
* Node.js can collect form data
* Node.js can add, delete, modify data in your database

## What is a Node.js File?

* Node.js files contain tasks that will be executed on certain events
* A typical event is someone trying to access a port on the server
* Node.js files must be initiated on the server before having any effect
* Node.js files have extension ".js"

**AngularJS -**

**SASS/LESS -**

**Jade -**

**EJS -**

**MongoDB -**

**Postgres -**