**Summary of Web Development All in One For Dummies Book**

After reading the Web Development book, I need to try to put in my own words the essence of Web Development. What is it? Essentially Web Development is using high level and abstract computer tools to create digital products that let people interact with the Internet in order to gain some sort of value.

What is the internet? The internet in essence is a way to remotely connect and collectivize massive groups of people towards some action or purpose. With the internet, we can educate people, entertain and narcotize people, undertake commerce over vast distances, perform surveillance and organize massive amounts of data and intelligence into usable models that can help somebody or some group, distribute useful information and knowledge, perform smoother and less frictional financial transactions, and form daily comforting relationships with others. The ability of the internet to reach across the planet and provide intelligence and coordination among billions at the speed of light is a very valuable tool.

The hardware of the internet are millions and millions of fiber optic cables, and satellites emitting electromagnetic waves, controlled by vast server farms as well as millions of individual personal computing devices have almost infinite uses limited only by the imagination. It is the radio towers networked across countries. It’s the hundreds of thousands of Wifi routers linked up to Television coaxial cables which ultimately end back up at the server farms that manage all the googles of signals.

On top of this hardware layer, there is the kernel or hardware drivers that convert all this analog light, electrical and radio signal into 1s and 0s or essentially digital signal that computers can interpret. It’s a level of abstraction that converts analog into digital. It’s still based on the universal concept of duality. In computer science, this would be called the Virtual Machine… Which is essentially structuring all these oscillating electric and photonic and radio waves into something more easily visualized. It’s a pretend and more structured world created from a less structured electromagnetic reality.

Finally, we get into the level where I’m at. This is the software world where we create usable applications and codes that allow all these hard lines and radio towers and computer hardware to have some use to people. That’s essentially what web development is. Part of creating usable applications is making them lively, fun, and attractive. Web development also deals with debugging errors, security, and combining with other sub disciplines such as machine learning, mobile app development, hardware, , gaming, video communications, banking, etc…

So as far as I can see based on the book, Web development can be divided into three abstractions. First there is the front-end part, then there is the pipeline and then there is the back-end part of web development. The front end part of web development is essentially dealing with the stuff that happens in the users’ face. This will be the web page on a personal computer, the app that is installed on the cell phone, the desktop application that the user immediately accesses on his or her device. It’s in front of your face. The back-end part of web development is the programming and coding part that the user does not directly experience. For example, it’s were all the user data for a website is stored such as bank account info, orders, characters in a web game, videos and blog posts, etc… Back-End Web development is the storing of this data, it’s organization, securing this data, and accessing it in an efficient manner so that a user can gain value from it. Finally, there is the pathway that connects Front-End and Back-End. How do you create a pathway that allows the web page or user interface to communicate with the stored data in the servers… And how do you build a pathway that allows updated back-end data to be displayed upfront to the users?

There are certain coding tools that are used for the front-end, pipeline, and back-end of web-development.

**Front-End Web Development**

So what are the tools of front end web development? First, there are **HTML (Hypertext Markup Language)** and **CSS(Cascading Style Sheets)**. HTML creates the basic outline of a webpage using things called tags. It adds text, spacing, page breaks, allows one to add images, headers and other things. What is a markup language? It is a system for annotating a document so that when the document is displayed, the annotations are not shown. It essentially just formats the document. What are Cascading Style Sheets? It is a tool that allows one to standardize and create the look of a webpage. It allows one to separate the presentation from the content(HTML). CSS controls fonts, colors, page layout and can allow you to apply the same styles to multiple web pages without having to repeat code over and over again.

Another tool for coding the front end is **JavaScript**. HTML and CSS is based off the JavaScript language. They are essentially more abstracted versions of JavaScript. The reason we move down a level of abstraction to use JavaScript directly. JavaScript is what programs the internet. The reason we would want to go down to the more fundamental level of JavaScript is that it allows greater control, power and customizability of the Web Application… At the expense of having to know more stuff. JavaScript is an actual programming language. So what are some concrete examples of what JavaScript can do? It will allow you to write scripts that display pop up boxes, add content to a webpage on the fly, make something happen when a user clicks a button, make an image change when the mouse hovers over it, calculate the total for an order form, calculate astrological birth chart, ask a web server for data and then display that data on a webpage, modify or remove page text/HTML tags/CSS properties, display messages to the user and ask for user info, collect and retrieve data for users, etc..

JavaScript can’t do these things: write data permanently to an existing file, access files on the server, glean information about the user including email or IP addresses, submit credit-card based purchases for authorization and payment, create multiplayer games, get data directly from server database, or handle file uploads.

The next tool for front end development is named **jQuery**. jQuery is basically a library based off JavaScript, that has many premade functions and tools for web design so that you don’t have to waste time recreating these tools yourself from raw JavaScript. Some things that jQuery allows you to do are: adding an element to a page using a consistent format, replacing an element’s HTML, replacing an element’s text, modifying CSS, manipulating classes, tweaking HTML attributes, adding animations and events, etc… There is also a jQuery UI or user interface that lets you use a graphical user interface to perform jQuery functions instead of just using the command line.

**Back-End**

Now I’ll discuss the tools for Pipeline development. The first tool is **PHP**. PHP is a server-side programming language. It is the pipeline connecting the front-end or client side to the back-end or server side of the web application. PHP generates HTML on the server-side that is output to the client-side and displayed on the browser or whatever output method. Part of PHP is learning to debug PHP. PHP is the language that you use to communicate with the MySQL database that stores data server-side.

The next tool is called **MySQL**. This is a database system. It stores and organizes different types of data, that is fetched by the PHP structures and displayed on the Front-End. With MySQL, you can store multiple related databases that are linked together. These databases can hold various things: single webpages, banking data, user posting data, product data, image data, personal information, videos. PHP code is the thing that communicates with the MySQL database.

**The Pipeline between Front and Back-End**

Now we discuss the pipeline part of the Web Stack. The pipeline allows the web page of user side to interact with the server to ask for and receive server data and you need to give the back-end server side a way to return data in a format the webpage can understand and manipulate. We have two tools for this: **Ajax** is for sending data back and forth between the web page and server. **JSON** is a tool for putting that data into a format that’s easily read by your web page code.

So, what specifically is **Ajax (Asynchronous JavaScript and XML)**? Well, in the past, the only way to see new data in a web page was to reload the entire webpage from the server. This was an extremely inefficient method, especially back in the days of dial-up internet. Ajax solves this problem and although it is a very complicated technology, it essentially inserts a layer called an Ajax engine between the web page and the web server. Asynchronous means that the web page doesn’t have to wait for the server to resend the entire page when the page changes. Instead the Ajax engine asks the server for data while also keeping the page displayed so the user can interact with it. JavaScript is the language used by Ajax to send requests to the server and to handle the response. To make this process easier, we use jQuery. Finally XML or eXtensible Markup Language is the data format that Ajax uses to send data to the server and to receive data from the server. JavaScript is like the messenger. XML is the format of the actual content of data that is exchanged back and forth.

So now, what specifically is **JSON (Javascript Object Notation)**? You can’t just send pure MySQL data back to the web browser because there are no front-end tools that can work with that format of data. So you need to convert that data into a special format called JSON. JSON is especially useful when you want to return a complex set of data from the server and display it on the web page. This usually means an array of database records. JSON allows you to process the data and make it more presentable for the user.

**Other things to Consider:**

So the above basically explained the Full-Stack pipeline of developing a Web Application so that users can take full advantage of the internet. One thing of massive importance is to design a **Secure** web application. I don’t know the specifics, but security against attacks and exploitation of data is a very important aspect of web development. Another thing to consider is designing web applications for multiple kinds of output. As of this time, the main outputs are through Desktop PCs, laptops, and mobile devices. The difference in screen sizes entails different design considerations and this has to be considered as a web developer. Everything in this essay is general and high level, but when I get down to the development, I’ll also have to consider things like managing user login information, web forms, forgotten passwords, the planning process for building a web application and so on.

But in summary, this is the basic outline for what a web developer has to deal with. The technologies will evolve and change, but for the most part, it seems that the basic abstraction and virtual machine software paradigm will stay the same. What I now have to learn are the syntax and commands and language…. The actual coding part of Web Development.