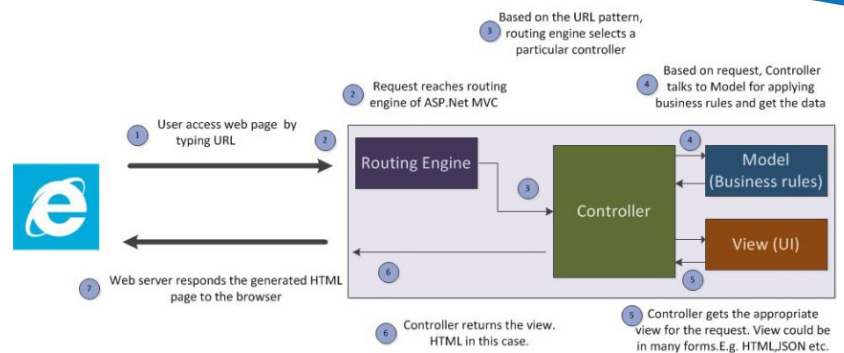
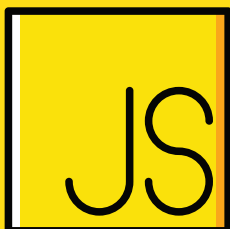
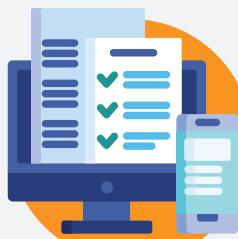


# LEARN TO DEVELOP A SIMPLE WEB APPLICATION



## EPISODE 1: The “HOW”

A Series By  
Ahmed Dawuda



## INTRODUCTION

Do you know the most popular and oldest platform is the web? I am pretty sure you've heard of the phrase “**Web application**”. If in one way or the other, you've never heard of this phrase, you're lucky and this book is rightfully yours. A web application is a software application that runs on a **remote server**. You've asked for help from other people, haven't you? When you ask for something from people, you are the client asking for that service from the person. The person granting you the service is the **server** because he/she is serving you. This should give you an idea of what a server is. Basically, a **server** is a computer just like your laptop, which runs one of the several operating systems we have, most of which is Linux operating system. **Operating systems** themselves are software applications that runs on the bare computer. They are important because they provide a conducive environment for all other software to run in. Usually these servers are cloud based. Web applications reside and run on servers. **Web browsers** are used to access the web applications, over a network, such as the internet. Some web applications are used in **intranets**. An intranet is a private network that only few selected computers can have access to., usually owned by organizations.

## ABOUT THIS BOOK

In this book, I'll be teaching you ways to create basic and fully functional web applications with modern frameworks. This tutorial will be delivered to you in a series of episodes which will help you understand the basic concepts required to develop web applications to solve your daily problems. This book will introduce you to HTML, CSS, JavaScript and several JavaScript frameworks including Vue.js, Angular, React.js etc. You will also learn backend technologies such as node.js, php, Laravel, databases etc.

## ASSUMPTIONS

- ▶ You're interested in learning how to develop your own web application.
- ▶ You have a basic understanding of programming and you can write some code, this could be the traditional "Hello World" program.
- ▶ You will study hard and follow this series, read every episode all the time
- ▶ You will practice the questions and exercises in all the episodes of this series.
- ▶ You have a computer.
- ▶ You have questions to ask.
- ▶ You do not expect this book to transform you from novice to pro or a ninja.
- ▶ You can install a program on your computer.
- ▶ You can type as fast as you can.
- ▶ You can use the terminal or command prompt.
- ▶ You don't expect to find this book boring or offending.
- ▶ You know how to copy and past code from external source.

If at least one of these assumptions about you is mistaken, then sorry this book is not for you. There are several books written by good writers to help you. Bye. Otherwise seat, relax, grab a cup of Coffee (wait, I prefer to chew than to sip, anyways) and enjoy this episode.

## TOOLS AND TECHNOLOGY CHOICES

In this book, I decided to use VSCode, VSCode is a text editor which we will use for writing our codes. Download and install VSCode here <https://code.visualstudio.com/download> . I will be using chrome browser.

## IS THIS BOOK FOR YOU?

This book is not designed to make you a “ninja” or a “rock star” or even a particularly good computer programmer. It won’t prepare you for immediate employment, nor can I promise that it will show you “the right way” to do things. On the other hand, it will give you a solid foundation in the essential topics that you’ll need in order to understand how the pieces of a modern web app fit together, and it will provide a launching point to further study on the topic. If you work your way through this book, you’ll know everything that I wish I had known when I was first starting web development.

## WHEN YOU NEED HELP



<https://github.com/kofi-dalvik/rapid-web-dev-for-the-weak>



@awesome\_kofi



+233547406827



kofidahmed@gmail.com

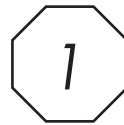
## ERRATA

This book may contain unrecognized errors or grammatical errors as it has not been thoroughly reviewed. Please report any error in this book to the contact details above.

## ACKNOWLEDGEMENT

Thanks to the leads of Google Developer Students Club for giving me the opportunity to lead the web development for the society. Also, thanks to anyone reading this book. A big thank you to my family and all those who helped my journey of attaining knowledge.

# *EPISODE*



*The “HOW”*

Have you ever wondered how the web and its applications work? I assume you've used a browser before. Get your laptop, open chrome browser, type in your favorite site mine is youtube.com. Notice how long it took to open. Has it opened yet? Guess what happened behind the scene.

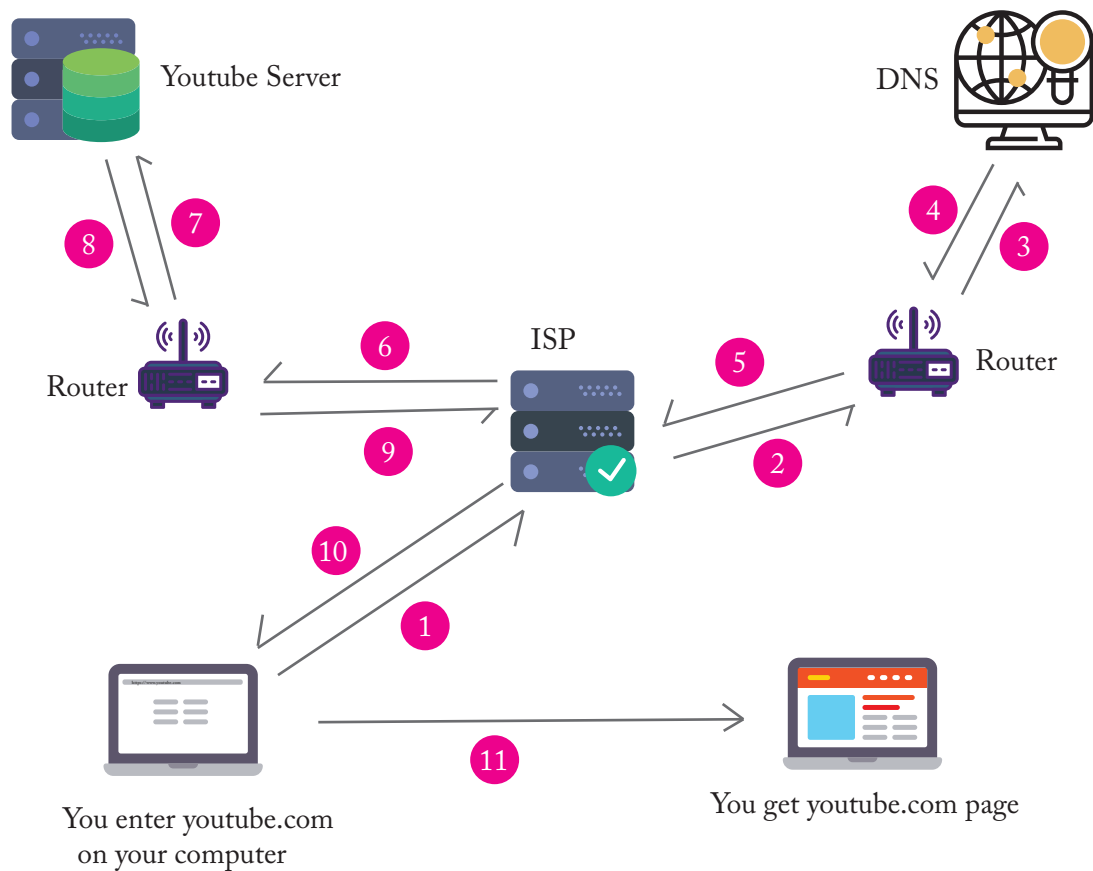
Basically, what happened is pretty simple. Someone has developed the application and its running on a **remote server**. Remote servers are usually servers connected to the **internet**. The **internet** is just a large network of computers of which your mobile device and any other devices connected to the internet form part of. Devices in the internet must be uniquely identified so each device is assigned an **ID**. This ID is the **Internet Protocol (IP)**. **Protocol** in Computer Science simply refers to a set of principles or rules governing how computers communicate with one another within the internet. Examples of protocols are **STMP, HTTP, HTTPS, TLS** etc. (Please, read more about the protocols above)

The IP consist of a set of numbers separated by dots like this **12.34.56.78, 198.34.2.65, 127.0.0.1** etc. In other to identify a web server, we need to remember the respective IP address of the server that our app is located on. But because keeping numbers is more difficult than words, IP addresses can be represented by words. Browsers need to contact our web servers using the server's IP address. The browser makes a search to other servers called **domain name servers (DNS)**.

Basically domain name servers are specialized servers whose work is to keep mapping of domain names (website names) to their respective IP addresses. Say you entered a link xyz.com, the browsers looks up the name xyz.com to get its ip address from a series of name servers. Details of the look up is outside the scope of this book.

The name server then returns to the browser with the IP address of the remote server the browser wants to access. The browser then makes a request to the remote server throug **ISPs**. The server responds to the browser by giving the name of the file the browser is requesting for.

## A DIAGRAM OF HOW THE WEB WORKS



The steps are bellow:

1

You type in <https://youtube.com> and press enter in your browser. The browser sends your link to an **ISP**. **ISP** (Internet Service Provider) is an organization that provides services for accessing, using, or participating in the Internet. Internet service providers may be organized in various forms, such as commercial, community-owned, non-profit, or otherwise privately owned. The browser then waits for a response from the ISP.

2

3

The ISP makes a request to a DNS to resolve the domain name for youtube.com through a router. The **router** is a device that looks for

the less expensive path to send a request to a destination. The router tries to find the quickest way to the DNS server through step 3.

4

5

The DNS Server looks up its folders and files to find the IP address that corresponds to the given domain name. i.e youtube.com. Once it finds the IP address, it returns the address to the ISP through the router.

6

7

8

9

Once the ISP has the IP address corresponding to youtube.com, it makes a request to the youtube.com server once again through the router. Based on the pattern of the **url** entered, the web server sends back a response to the ISP through the router. The response can be of various forms, this could be error, success etc. The ISP then sends the response to the browser and you tube page is shown.

## BASIC WEB TECHNOLOGIES

**Web technologies** is a general term referring to the many languages and multimedia packages that are used in conjunction with one another, to produce dynamic web sites such as <http://www.alphadevx.com>.

Each separate technology is fairly limited on it's own, and tends to require the dual use of at least one other such technology. Therefore we can conclude that all of the components that make up a site are interdependent on one another.

This guide is intended to briefly introduce the you to each of these separate technologies, without attempting to teach you how to program. If you intend to study web development further, make sure you follow this series. Read every episode.



## **HTML: The foundations of every web site.**

HTML (Hyper Text Mark-up Language) is the glue that holds together every web site. Like building a house, you always build a strong foundation first.

For any site, HTML is that foundation. HTML is an open source language (i.e. not owned by anyone), which is easy to learn, and requires no fancy (or expensive!) packages to start using it. All you need is something to type with, such as Windows Notepad, VSCode, and a lot of time, patience and something to chew.

## **CSS: The Scintillator.**

CSS (Cascading Style Sheets) is a relatively new language, designed to expand upon the limited style properties of HTML. Easy to learn and implement, CSS is an excellent way to control the style of your site, such as text styles like size, colour and font.

CSS may also be placed inside the HTML page or in separate files. The real advantage of having all of the style properties for your entire site in one single CSS file is that you may edit that single file to effect changes on the whole site, rather than having to go through each HTML file one at a time. For this reason, it is perhaps the most useful web technology and certainly one of my favourites.

**CSS-P (CSS-Positioning)** is a sub-set of CSS, and is concerned mainly with the layout of your HTML pages. It allows the web designer to place any element (text, graphic etc.) exactly on the screen where they want it, to the pixel.

## **JavaScript: The Event Organizer.**

JavaScript is a 'scripting' language. A bit like a script in a feature film, it is used to decide 'what happens next'. This may be a sequence of screen events, where one event is initiated by the end of another, or it could be a programmed response to a user interacting with the page in some way, e.g. moving their mouse over a link. JavaScript is a complex and powerful language, and may be placed directly inside a HTML page, or in a separate JavaScript file.

## The Backend: The logic.

If you will forgive me for continuing with the house analogy; if HTML forms the foundation and structure of the house, (the bits you see), then the backend would form the functional workings parts of the house that you can not see such as the plumbing, electrical wiring and heating.

The backend has certain technologies beign used. These include PHP, Perl, Node.js, MySQL etc

We can differentiate between the presentation (the bit the visitor sees) and the functional (the bit they don't see) parts of a site using the terms “**front-end**” and “**back-end**” respectively. Therefore, PHP is often referred to as a back-end technology. It operates on the server that hosts your site, rather the within the browser window of a visitor to your site. On the other hand, HTML, CSS and Javascript are front-end technologies because they run on the browser.

## IN A NUTSHELL

We define the structure of our web site in a file using HTML, thus the HTML tells the browser which build-in components we intend to use. These components can be a list with bullets, a paragraph, a link, image, videos etc.

Once we define the structure, we use CSS to define the appearance of the built-in components we added by HTML. With CSS, we apply colors, font styles and sizes and most importantly position the elements on the browser window.

We now have a nice looking website on the browser window. You know computers listens to user events such as clicking, dragging, hovering etc. These are events that are native to the browser.

You may want to do something when a user clicks on a particular item on the browser window. This is where javascript comes in. We use javascript to add interactivity to the website.

## EXERCISE

Read the history of the web, how it works, the various web technologies available.

