Once known as the “Information Superhighway” the internet is not a system of pneumatic tubes, nor was it invented by Al Gore - the Internet is a giant network of computers which can actually be dated back to the 1960’s.

When visiting a site like https://www.techtonicgroup.com/ several (usually 5) steps happen before the webpage is displayed on your computer. First your computer (the client) sends a request for the webpage, which is actually a document (usually an HTML file). Next, the computer listening to the request on the other side (the server) responds by informing the client that it is checking if the document is available. The client computer then tells the server it will wait. The server then responds by sending the requested document to the client, if it’s available. Finally the client computer informs the server it has received the document.

For data to reach the client and for the server to be accessed there is a specific procedure for the network to function. When you enter https://google.com (which is actually known as a domain name) into a browser, a DNS (dynamic name server) is queried for it’s actual address - known as an ip address. Ipv4 is the current standard, which limites ip addresses between 0.0.0.0 and 255.255.255.255. Once your computer has an IP address, it attempts to access the website with the protocol you requested, in this case the protocol is known as https://, which is specifically designed to fetch HTML documents. Other protocols exist, such as FTP (file transfer protocol), which is designed to access a file system on a remote computer.

For a website to actually be displayed in a browser, it has to receive the HTML, CSS and Javascript files from a host server, in addition to any images or videos that need to be displayed. Many websites have extensive databases that are not sent to the client, only the bare minimum is sent - for security and performance reasons. For example if you log into amazon.com you can only view information that is connected to your amazon login. This info is inserted into the HTML/CSS/JS files you receive, and all other info remains on the server (in this case their database), unavailable from the client.

The software written for a server is different than that of a client, since it performs a vastly different function. This is known as server-side code. It is used to fetch, modify or create permanent data, usually stored on a database, that is meant to be secure from some or all of the public. This data is then inserted into a HTML/CSS/JS file for the clients browser to render. Examples of these functions include: user validation - like logging into a bank; retrieving data - displaying your bank account balance; or changing data - withdrawing or adding money to your bank account.

Conversely, software written for a client is intended to be fast and modifiable without the need to contact a database stored on a server remotely. This allows webpages to change quickly for the client with little to no interaction from the host server. The end result is a more interactive webpage, with fast response times and less overhead for the network.

Every time a web page is opened your web browser requests a new set of HTML/CSS/JS documents to be rendered in your browser. Opening a new tab and going to the website again will result in a completely separate copy of these HTML/CSS/JS files being downloaded. This is intended behavior, since it is always preferred to download the newest version of the website available. For example: if you leave a browser tab open on news.google.com for a day, and open up a new tab you will receive a different version of the document residing at news.google.com.

Alternatively, server side code only has one version of HTML/CSS/JS available at one time - the most current version uploaded to the web server. This is intended since we want websites to be consistent when viewed across multiple devices or computers. When a website is updated (manually or automatically) these HTML/CSS/JS documents are replaced completely with a new set. The old documents are usually archived if needed on the web server's storage.

Runtime is software designed to perform the execution of computer code written in some specific language or platform. For example: if you download a program written in C++, not all of the required data may be included in the executable - you may need to download addition runtime libraries so the program will function. The addition of these runtime libraries is sometimes automated by the install program, but not always. In this case the runtime library must be manually installed in order for the program to function.

At least one instance of a database connected to a server application are available at one time. Sometimes more instances of these databases are created, in order to relieve load on the server if the architecture of the application requires it.