HOW INTERNET WORKS?

- Website need to be accessible to anyone from anywhere at anytime. A powerful external computer connected to the Internet, called a server, stores these Website's file(HTML,CSS,JS).
- When you enter a URL like https://sadiq.cs/hello-world, your browser has to figure out which server on the Internet is hosting the site. It does this by looking up the domain, sadiq.cs, to find the address.
- Each device on the Internet have a unique address called an IP address. Eg: 194.185.221.7
- Now domain names come in. sadiq.cs is much easier to remember than 194.185.221.7.
- DNS helps our browser (and us) find servers on the Internet.
- We enter https://sadiq.cs/hello-world, DNS will search the corresponds IP and will go to that IP.

THE PROCESS:

1. **Let's break down the parts of this URL you typed in to get here:**
   - https:// is the scheme. HTTPS stands for Hypertext Transfer Protocol Secure. This scheme tells the browser to make a connection to the server using Transport Layer Security. TLS is an encryption protocol to secure communications over the Internet
   - sadiq.cs is the domain name of the site. It is the memorable address and points to a specific server's IP address.
   - Sometimes there is an additional path to the resource in the URL. For example, for this URL,https://sadiq.cs/the-path-to/hello-world, the-path-to is the path on the server to the requested resource, hello-world.
   - So instead of taking u to homepage of sadiq.cs it will take you to hello-world which belong to sadiq.cs.

2. **Browser looks up IP address for the domain:**
   - After you've typed the URL into your browser and pressed enter, the browser needs to figure out which server on the Internet to connect to.
   - To do that, it needs to look up the IP address of the server hosting the website using the domain you typed in. It does this using a DNS lookup.

- DNS data is cached at different layers between your browser and at various places across the Internet.
- Your browser checks its own cache, the operating system cache, a local network cache at your router, and a DNS server cache your internet service provider (ISP).
- If the browser cannot find the IP address at any of those cache layers, the DNS server at your ISP does a recursive DNS lookup.
- A recursive DNS lookup asks multiple DNS servers around the Internet, which in turn ask more DNS servers for the DNS record until it is found.

3. **Browser initiates TCP connection with the server:**

   - Using the public Internet routing infrastructure, packets from a client browser request get routed through the router, the ISP, through an internet exchange to switch networks, all using transmission control protocol, more commonly known as TCP, to find the server with the IP address to connect to.
   - Many sites use a content delivery network, or CDN, to cache static and dynamic content closer to the browser.
   - A CDN is a globally distributed network of caching servers that improve the performance of your site or app (the origin) by bringing the content closer to your users.(It cache on a server nearby us)
   - Requests from the client browser get to take advantage of this private network that has ultra-low latency and high availability.
   - Instead of relying on the public internet routing infrastructure and being subject to extra hops, redeliveries, and packet loss, the client browser request use CDN. The request is intelligently routed through the most performant location to deliver content to your browser.
   - Once the browser finds the server on the Internet, it establishes a TCP connection with the server and if HTTPS is being used, a TLS handshake takes place to secure the communication.
   - Once the browser has established a connection with the server, the next step is to send the HTTP request to get the resource, or the page.

4. **Browser sends the HTTP request to the server:**

   - Now that the browser has a connection to the server, it follows the rules of communication for the HTTP(s) protocol.
   - It starts with the browser sending an HTTP request to the server to request the contents of the page.

- The HTTP request contains a request line, headers and a body.
- The request line contains information that the server can use to determine what the browser wants to do.
- The request line contains the following:
  - a request method, which is one of GET, POST, PUT, PATCH, DELETE, or a handful of other HTTP verbs.
  - the path, pointing to the requested resources.
  - the HTTP version to communicate with.
- The request line for the URL request looks like this: GET /hello-world HTTP/1.1
- The request line tells the server that you want to GET resource at /hello-world and to communicate with HTTP/1.1.
- Once the server has received the request from the client, the server processes it and sends back a response.

5. **Server processes request and sends back a response:**

- The server takes the request and based on the info in the request line, headers, and body, decides how to process the request.
- For the request, GET /hello-world/ HTTP/1.1, the server gets the content at this path, constructs the response and sends it back to the client.
- The response contains the following:
  - a status line, telling the client the status of the request.
  - response headers, telling the browser how to handle the response.
  - the requested resource at that path, content like HTML, CSS, Javascript, or image files, or data.
- Now that you know how the server builds the response to send back to the browser, let's take a look at how the browser handles the response.

6. **Browser renders the content:**

- Once the browser has received the response from the server, it inspects the response headers for information on how to render the resource. The Content-Type header above tells the browser it received an HTML resource in the response body.
- As the browser is parsing and rendering the HTML, it is making additional requests to get Javascript, CSS, images, and data. It can do much of this in parallel, but not always.

7. **Wrapping Up:**
    - You traced a URL request from the browser all the way to the server hosting it and it's response back to the browser to be rendered.
    - For review, here are those six steps:
        - You type a URL in your browser and press Enter.
        - Browser looks up IP address for the domain.
        - Browser initiates TCP connection with the server.
        - Browser sends the HTTP request to the server.
        - Server processes request and sends back a response.
        - Browser renders the content.