Web Application

- A piece of software, which can be accessed from a browser.
- Multiple users, same version of an app.
- Doesn't need to be installed.
- Can be accessed through various platforms such as a desktop, laptop, or mobile.
- For a web app to operate, it needs-
    - Web Server
    - Database

Web Client

Something that can talk to a web server. Web client sends request to web servers
Example: Browsers → Safari, Chrome. Some apps → Daraz (jekhane kono product search diye Daraz er web server e request pathate partesi)

Web Server

- Serves web content. Listens on a port for a request via some Transport Protocol (HTTP, HTTPS)
- Response → Waits for a request and then sends a response along with some resources.
- Resource → html, json, pdf, image etc

** Web server wait korte thake kokhon ekta request ashbe, request ashle shei request e ja ja chawa hoise shegula tar storage theke pathay dey.

Protocol

Set of agreed upon rules between Client & Server.
Example: HTTP, HTTPS

Client-Server Communication

Basic concepts of web applications, how they work and the HTTP protocol

Static vs Dynamic Web Applications

https://www.linkedin.com/pulse/static-vs-dynamic-website-key-differences-which-use-w0qec/

URL

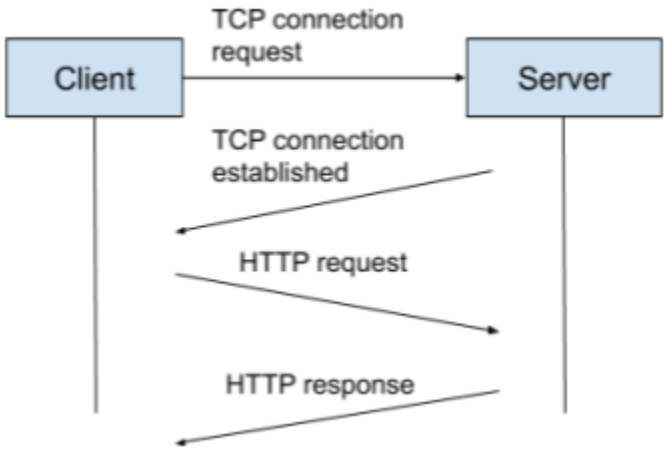
Know about HTTP URL

HTTP Protocol

The Hypertext Transfer Protocol is an application protocol for distributed, collaborative, hypermedia information systems that allows users to communicate data on the World Wide Web. HTTP clients generally use Transmission Control Protocol (TCP) connections to communicate with servers
https://www.tutorialspoint.com/http/index.htm
https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

- Connectionless Protocol - Client theke server e request pathacchi, server response back kortese. Client theke server e ei request ta jay ekta TCP/IP connection diye. Request jawar somoy connection ta established hoy, ar response chole ashle closed hoye jay. That means server ta client ke chintese shudhu request/response er time tuku. Er baire server client keu kauke chine na.
- Stateless Protocol - Meaning there is no link between two requests being successively carried out on the same connection (Ekta client er kach theke joto request ee pai, prottek ta request pawar por oi client ke ekta notun client hishebe handle korbe). This immediately has the prospect of being problematic for users attempting to interact with certain pages coherently, for example, using e-commerce shopping baskets. But while the core of HTTP itself is stateless, HTTP cookies allow the use of stateful sessions (will learn about it later).
- Application Layer Protocol - Jeno browser bujhte pare
- TCP/IP based
- Media independent - Response e jekono type media pathano jay. ‘Content type’ field e just media type ta boshay dite hobe.

How does connection establish between client & server?



1. **User enters the URL** of the website or file. The browser requests the DNS (Domain Name System) server for the corresponding IP address.
2. **DNS server provides the IP address** of the web server.

3. The **browser sends a TCP connection request** (also known as a **TCP handshake**) to the web server using the provided IP address. This establishes a reliable connection between the client (browser) and the server.
4. Once the TCP connection is established, the **browser sends an HTTP/HTTPS request** to the web server to request the necessary files.
5. The **server responds** with the website's files, and the browser renders the website.

From the connection request to the connection establishment is called one round trip time. Similarly, from the HTTP request to the HTTP response is one round trip.

HTTP Request, Response Message

https://mscancer22.tripod.com/applicationlayer/id19.html
https://medium.com/@adilrk/http-request-and-response-e7da8eb3a00c
https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages
https://developer.mozilla.org/en-US/docs/Web/HTTP/Session

HTTP Methods

https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods (Details in the corresponding links, except *Specifications*)
https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/HTTP-methods
https://sentry.io/answers/what-is-the-difference-between-post-and-put-in-http/
https://www.freecodecamp.org/news/http-request-methods-explained/ (HTTP Method FAQs)

HTTP Status Codes

https://developer.mozilla.org/en-US/docs/Web/HTTP/Status (You do not need to memorize all the codes, just know the basics)