

Who am I?

Humera Tariq

PhD, MS, MCS (Computer Science), B.E (Electrical)

Postdoc (Medical Image Processing, Deep Neural Networks)

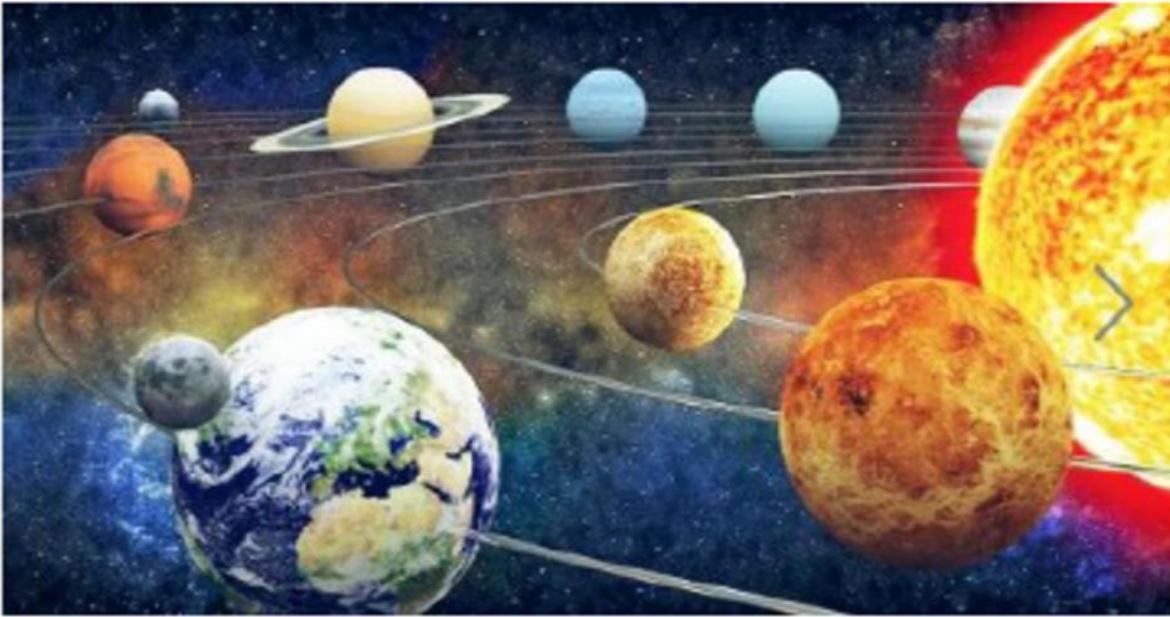
Email: humera@uok.edu.pk

Web: <https://humera.pk/>

Discord: <https://discord.gg/xeJ68vh9>

Starting in the name of Allah,

*the most beneficial,
the most merciful.*



۲۳
تہنیٰ

کیا انسان کو ہر وہ چیز حاصل ہے جس کی اس نے تمنا کی؟



UNIVERSITY OF
KARACHI

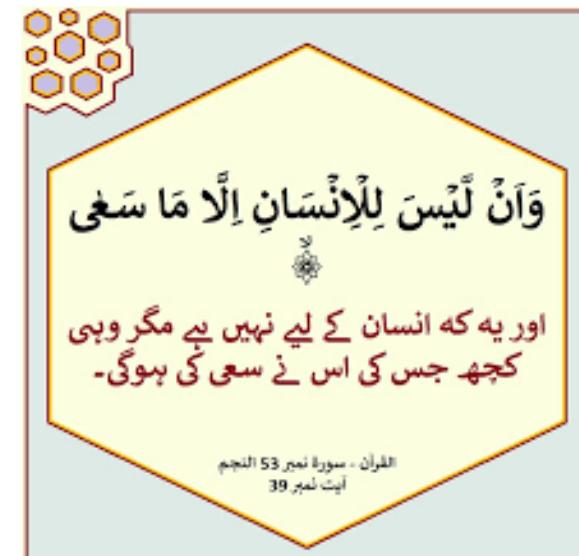
Surah An-Najm Chapter 53 Verse 39

اور یہ کہا سان گروہی ملنا ہے جس کی دُکوٹش کر رہے

(القرآن ۵۳:۳۹)



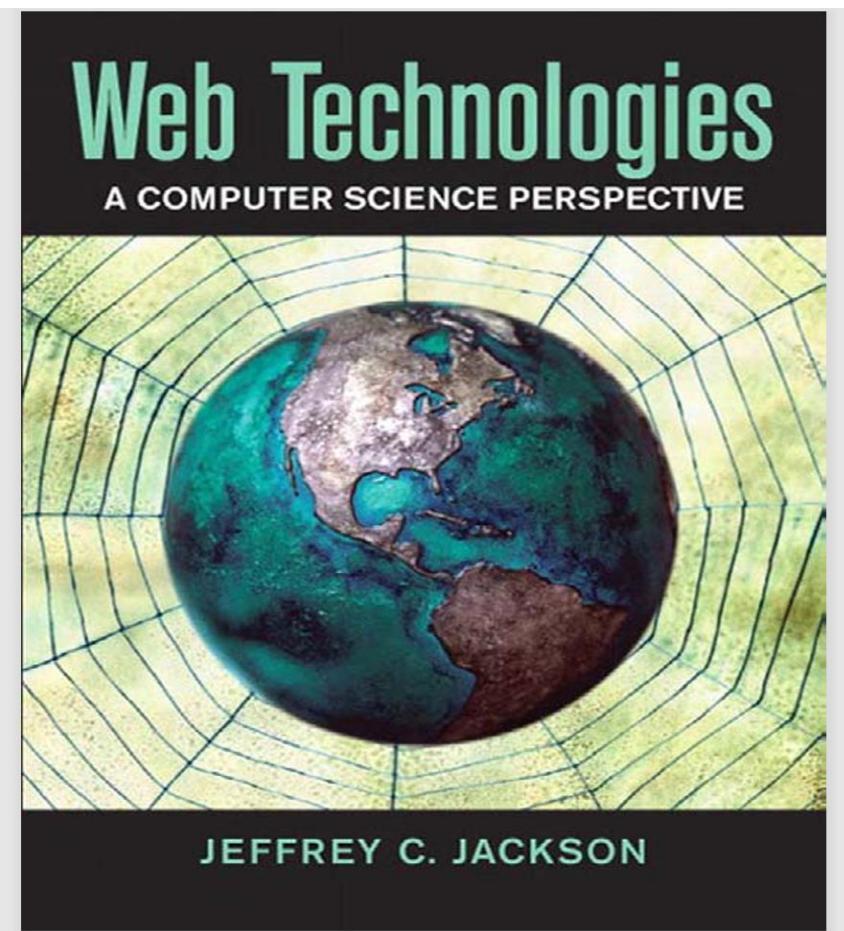
And there is not for man except that [good] for which he strives.



UNIVERSITY OF
KARACHI

Read, Read, Read..... Practice, Practice, Practice

[Vrije Universiteit Brussel | World University Rankings | THE](#)

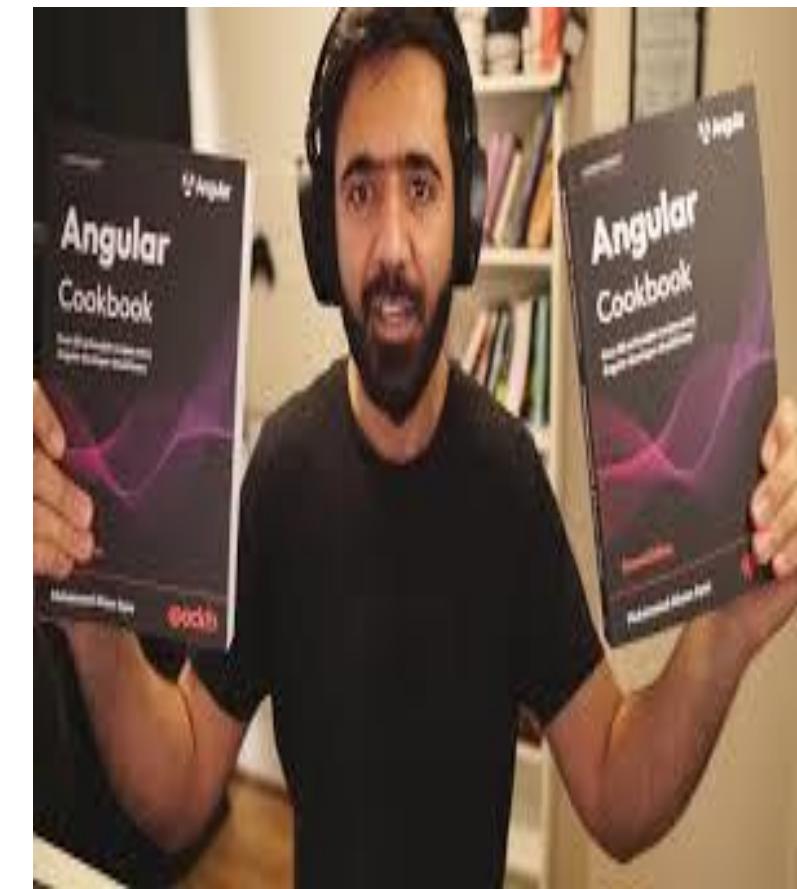
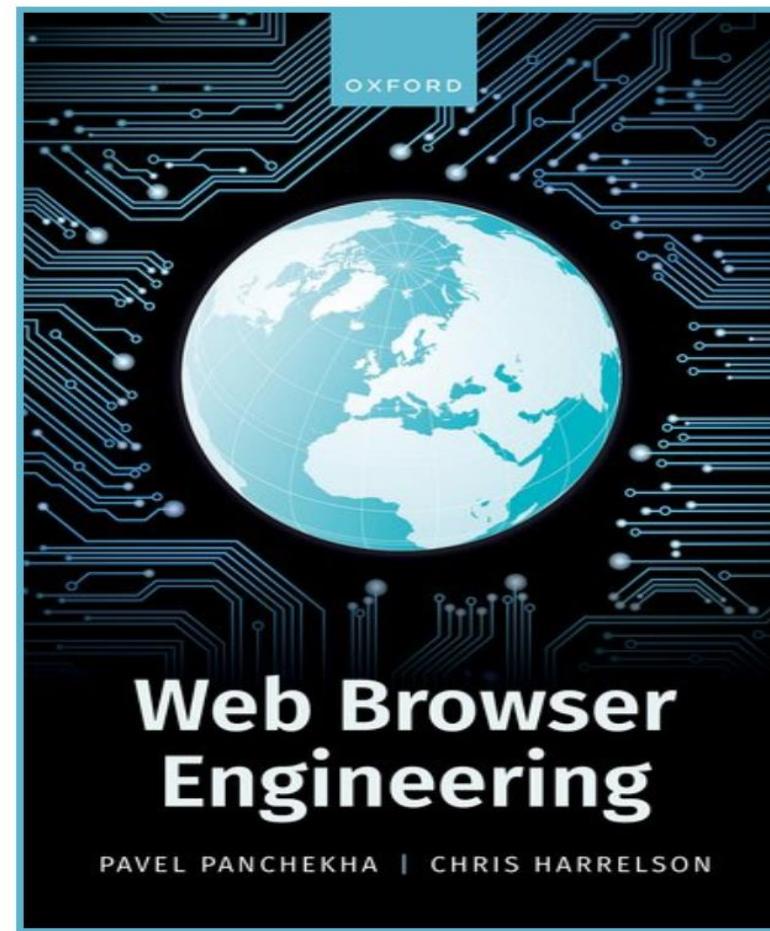


University of Utah

University of Washington

HAW Hamburg

The full source code is also available [on GitHub](#),
<https://browser.engineering/classes.html#university-of-utah>



UNIVERSITY OF
KARACHI



The Complete Reference

Covers
HTML5

HTML & CSS

Fifth Edition

- ▲ Design standards-based Web pages
- ▲ Learn markup and CSS best practices
- ▲ Deliver optimum client-side experiences

Thomas A.
Powell



UNIVERSITY OF
KARACHI

Week 02

Internet Application Development



Copyright © 2025, Humera Tariq

*Department of Computer Science (DCS/UBIT)
University of Karachi
January 2025*

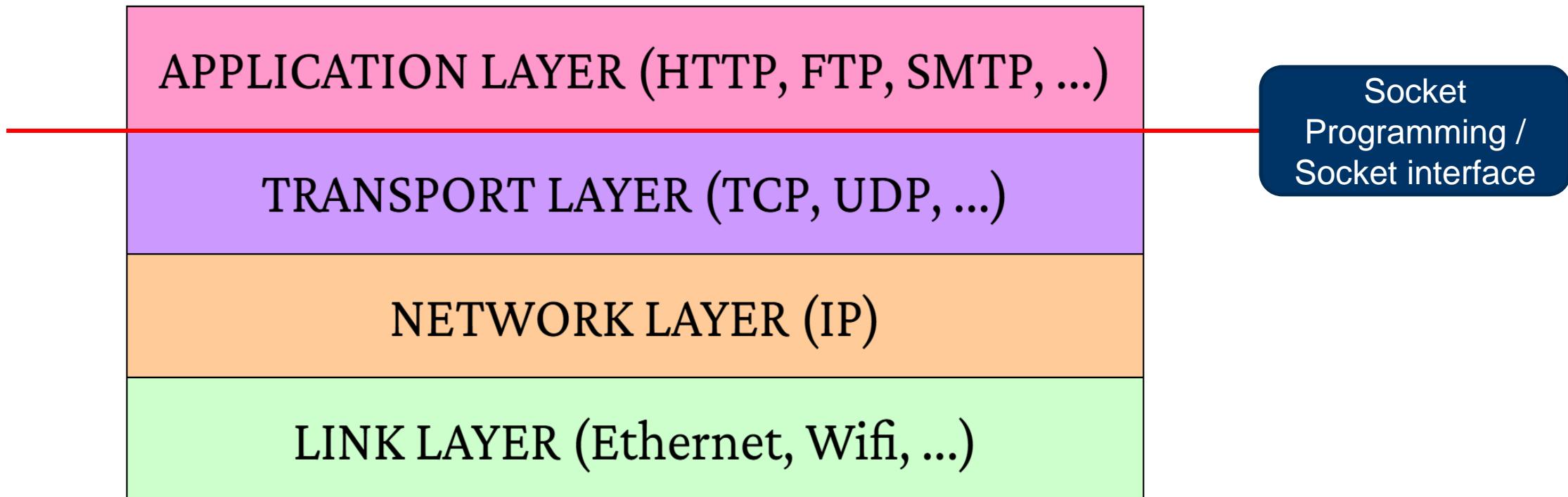
Background/Context

Internet Application Development

Review of Internet Layers

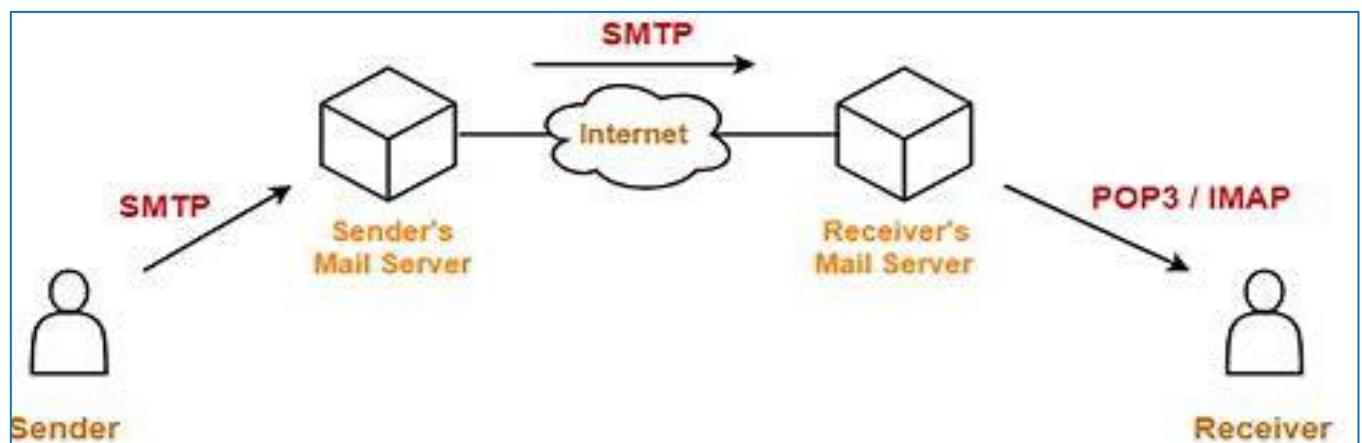
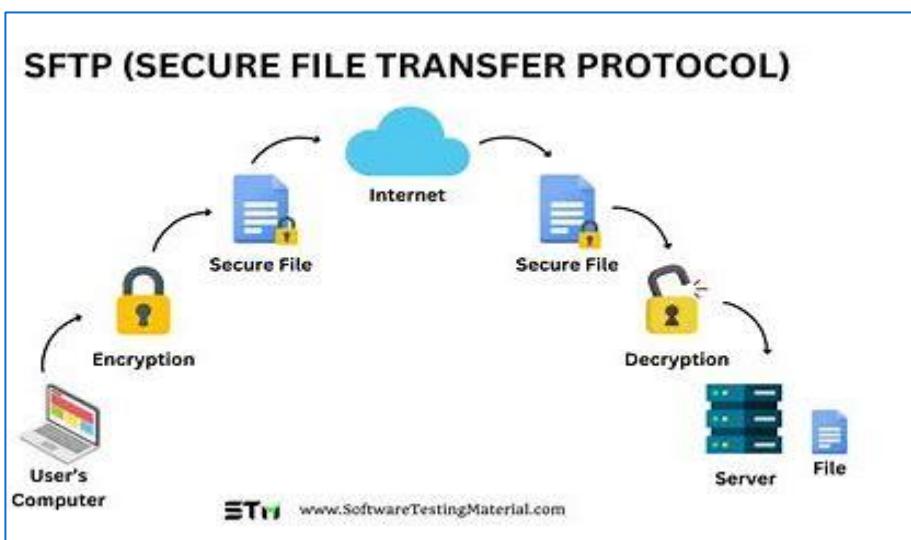
All Internet applications work over the exact same transport layers. The Internet says nothing about how these applications should work. It provides IP, TCP, and UDP, and that's it.

You can build anything on top of the Network & Transport Layer.

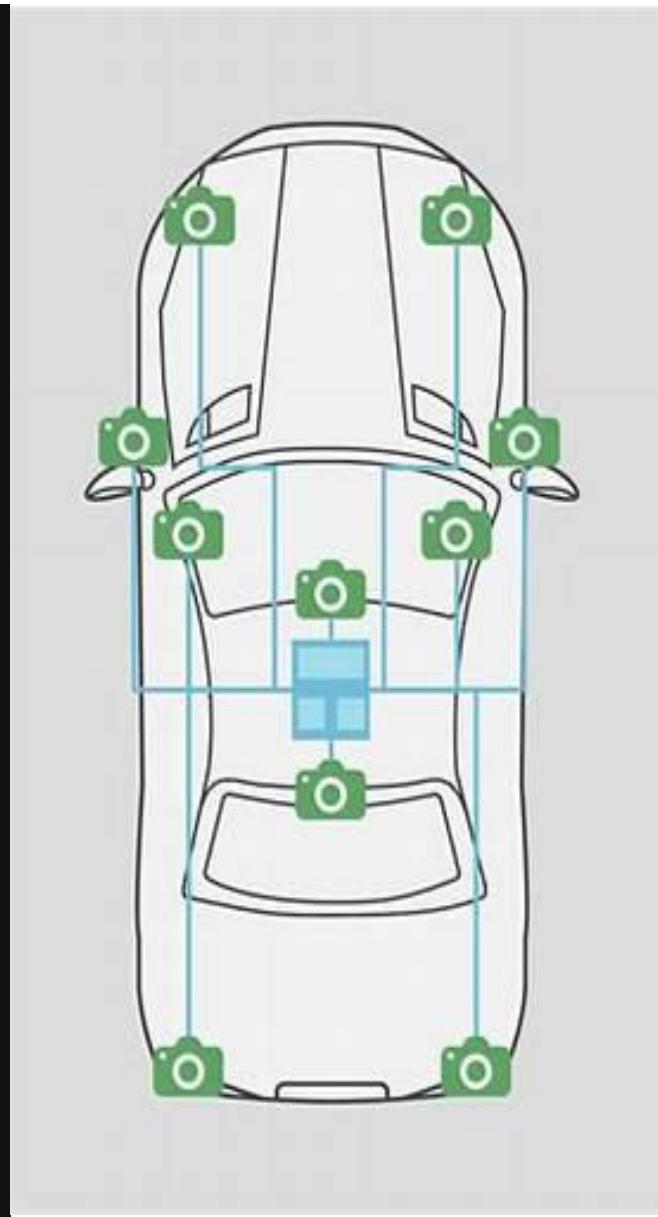


An Internet application does something for end users.

- ✓ <https://www.getmailbird.com/best-email-apps/>
- ✓ <https://www.anyviewer.com/how-to/network-file-transfer-software-2578.html>



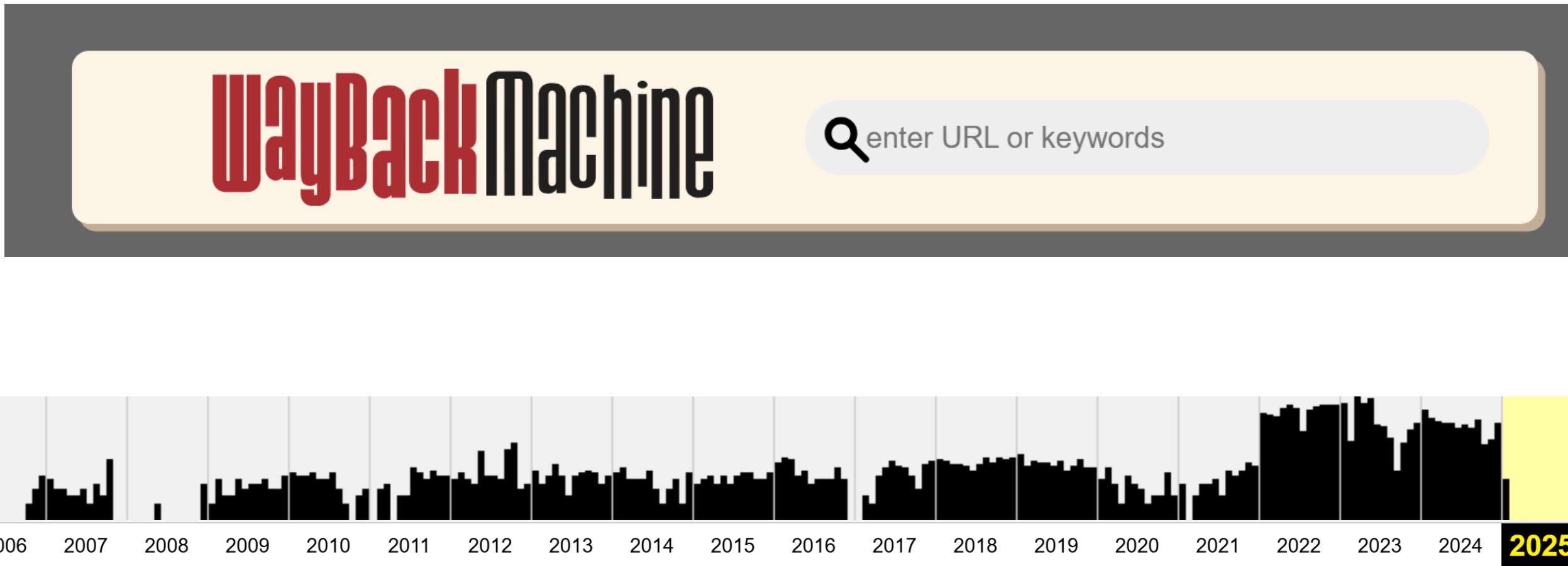
Which Internet Application I am thinking of ????



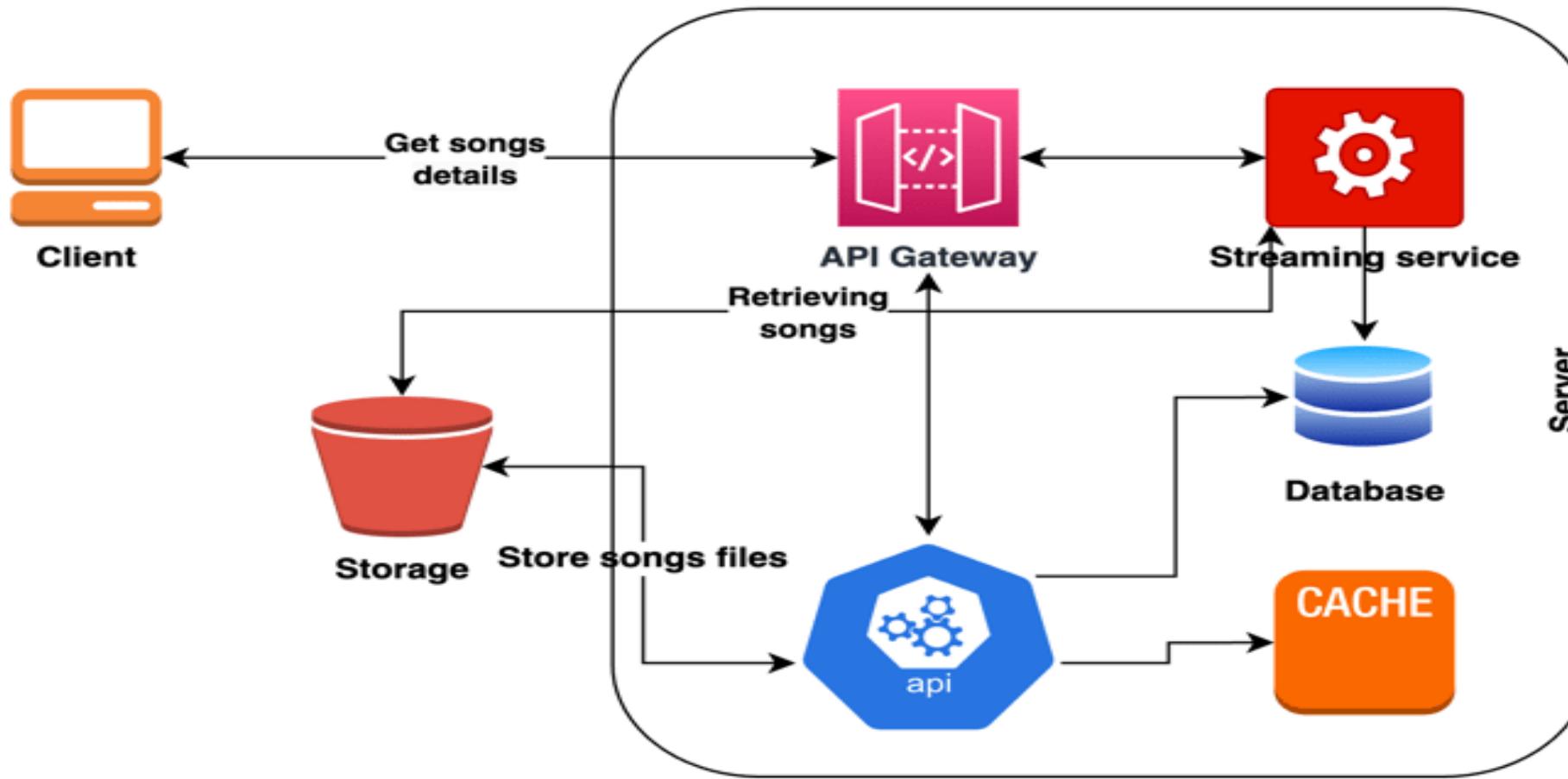
Navigation Server => Image/Video/Frame => Streaming



Which Internet Application I am thinking of ????



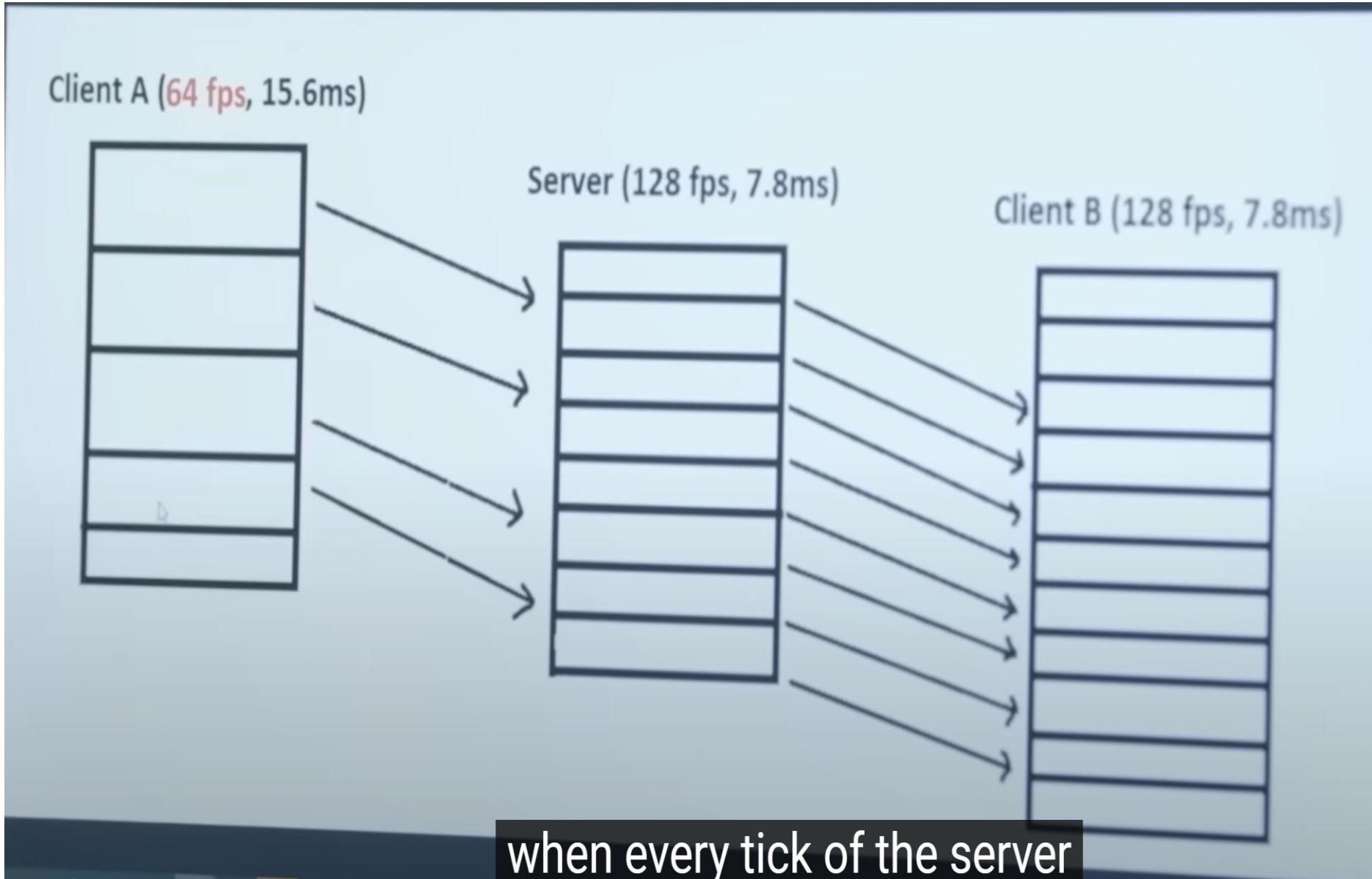
Which Internet Application I am thinking of ????



```
git clone -b v1.5 https://github.com/koladev32/golang-react-music-streaming.git  
cd golang-react-music-streaming  
make setup
```

Which Internet Application I am thinking of ????

<https://www.youtube.com/watch?v=YpNsGFd74fk>



Basics/Foundations

Internet Application Development

INTERNET

WWW

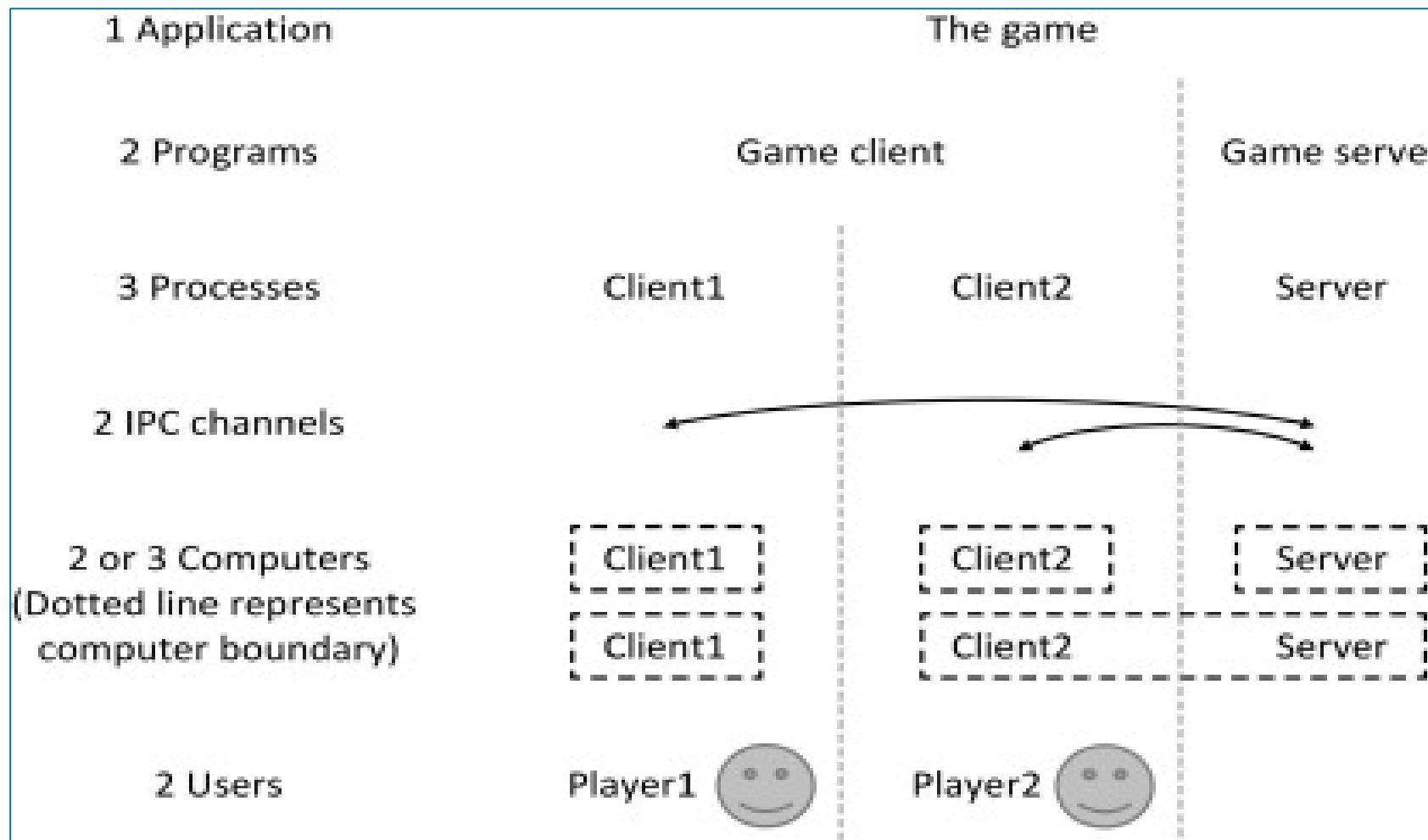
Internet is a global network of networks.	WWW stands for World wide Web.
Internet is a means of connecting a computer to any other computer anywhere in the world.	World Wide Web which is a collection of information which is accessed via the Internet.
Internet is infrastructure.	WWW is service on top of that infrastructure.
Internet can be viewed as a big book-store.	Web can be viewed as collection of books on that store.
At some advanced level, to understand we can think of the Internet as hardware.	At some advanced level, to understand we can think of the WWW as software.
Internet is primarily hardware-based.	WWW is more software-oriented as compared to the Internet.
It originated sometimes in late 1960s.	English scientist Tim Berners-Lee invented the World Wide Web in 1989.
Internet is superset of WWW.	WWW is a subset of the Internet.
The first version of the Internet was known as ARPANET.	In the beginning WWW was known as NSFNET.
Internet uses IP address.	WW uses HTTP.
It consists of interconnected computers, copper wires, fiber optics, and wireless system.	It comprises documents, files, and folders saved in various linked computers.

Writing Your Own Internet Applications

<https://www.sciencedirect.com/topics/computer-science/client-server-paradigm>

Current IoT solutions mostly rely on the centralized client-server paradigm where all the devices are identified, authenticated, and connected via cloud servers

Client Server paradigm



Internet Applications pretty much just need to know:

- 1) the **IP address** of the other party (what *host* the other party is running on—a network layer concept),
- (2) the **port number** of the application running at the other end (because the other machine might be running multiple services—a transport layer concept).

The application passes those two pieces of information to the transport layer to make the communication happen.

So, hosts have IP addresses and applications run on specific ports.



P2P paradigm

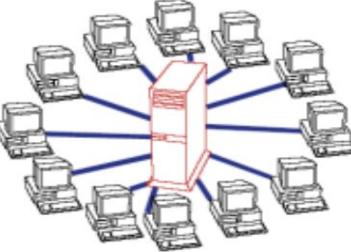
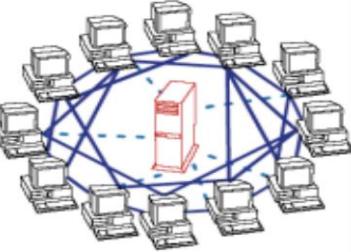
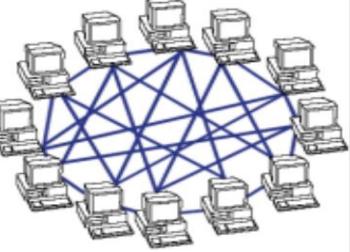
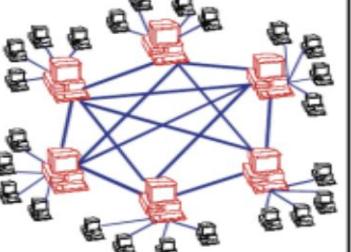
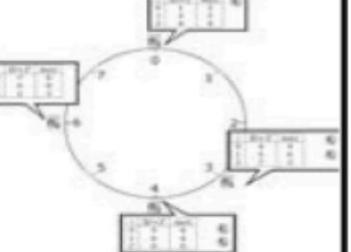
Peer-to-peer systems (P2P systems) represent a paradigm for the construction of distributed systems and applications in which data and computational resources are contributed by many hosts on the Internet.

P2P systems enable the sharing of data and resources on a very large scale by eliminating any requirement for separately managed servers and their associated infrastructure.



P2P systems have been used to provide **file sharing, web caching, information distribution and other services**, exploiting the resources of tens of thousands of machines across the Internet.

Summary of the characteristic features of Client-Server and P2P

Client-Server	Peer-to-Peer				
	Unstructured P2P			Structured P2P	
1st Generation		2nd Generation			
<p>1. Server is the central entity and only provider of service and content. → Network managed by the Server</p> <p>2. Server as the higher performance system.</p> <p>3. Clients as the lower performance system</p> <p>Example: WWW</p>	<p><i>Centralized P2P</i></p> <p>1. All features of Peer-to-Peer included</p> <p>2. Central entity is necessary to provide the service</p> <p>3. Central entity is some kind of index/group database</p> <p>Example: Napster</p>	<p><i>Pure P2P</i></p> <p>1. All features of Peer-to-Peer included</p> <p>2. Any terminal entity can be removed without loss of functionality</p> <p>3. → No central entities</p> <p>Examples: Gnutella 0.4, Freenet</p>	<p><i>Hybrid P2P</i></p> <p>1. All features of Peer-to-Peer included</p> <p>2. Any terminal entity can be removed without loss of functionality</p> <p>3. → dynamic central entities</p> <p>Example: Gnutella 0.6, JXTA</p>	<p><i>DHT-Based</i></p> <p>1. All features of Peer-to-Peer included</p> <p>2. Any terminal entity can be removed without loss of functionality</p> <p>3. → No central entities</p> <p>4. Connections in the overlay are "fixed"</p> <p>Examples: Chord, CAN</p>	
					

(Eberspächer, & Schollmeier 2005)



UNIVERSITY OF
KARACHI

Distributed Applications: browser execution + Server execution + DB execution

Descriptive models for distributed system design

Physical model

Architectural model

Architectural elements

Communicating entities

Processes

Objects

Components

Web Services

Communication paradigm

Inter-process communication

UDP sockets

TCP sockets

Multi-cast

Indirect communication

Remote invocation

Roles and responsibilities

Architectural styles

Client-server

Peer-to-peer

Placement

Multiple server

Proxy/Cache

Mobile code

Interaction model

Interaction model

Failure model

Security model

Architectural patterns

Vertical distribution

Multi-tier

Thin/Fat Client

Horizontal distribution

Web site – Web app – Web Service

What is the difference?

Check out the full Wikipedia articles on [web applications](#) , [web servers](#), and [web services](#).

Early Internet Applications!

A Primer on Internet TCP/IP Tools and Utilities 1997

Application	RFC	Port	Description
Daytime	867	13	When a client connects, the server sends back a string with the current date and time, then immediately closes the connection. (From the RFC: “There is no specific syntax for the daytime. It is recommended that it be limited to the ASCII printing characters, space, carriage return, and line feed. The daytime should be just one line.”)
Time	868	37	When a client connects, the server sends back a 32-bit time value, then immediately closes the connection. (From the RFC: “The time is the number of seconds since 00:00 (midnight) 1 January 1900 GMT, such that the time 1 is 12:00:01 am on 1 January 1900 GMT; this base will serve until the year 2036.”)
Quote of the Day	865	17	When a client connects, the server sends back a short message then immediately closes the connection. (From the RFC: “There is no specific syntax for the quote. It is recommended that it be limited to the ASCII printing characters, space, carriage return, and line feed. The quote may be just one or up to several lines, but it should be less than 512 characters.”)

DNS	1035	53	Clients ask the name service questions, usually of the type "what is the ip address for this domain name?" and the server responds with an answer of some sort.
Trivial File Transfer	1350	69 udp	TFTP. Much weaker than FTP, but is lightweight and simple. The sender and receiver exchange files packet by packet in lock step, with acknowledgements being part of the protocol itself.
Simple Mail Transfer	2821	25	SMTP. A very simple protocol for sending email.
Mailbox Access with POP3	1939	110	A simple protocol for managing email.
Mailbox Access with IMAP	3501	143	A modern alternative to POP3. IMAP clients can stay connected for longer times than POP3 clients, can have multiple clients attached to the same mailbox simultaneously, keep state on the server, fetch partial messages, and do other cool things.
File Transfer	959	21	A fully featured file transfer application using the File Transfer Protocol, FTP. Server runs on port 21 with data transferred through port 20.

World Wide Web	7230-7235	80	Servers usually run on port 80 (clear) or port 443 (secure) but really could run anywhere. Clients request resources via a uniform resource identifier (URI) and the server responds to the request. Request and response structures are quite detailed. The resources can be absolutely anything (so the responses usually contain a media type in their header).
News	3977	119	Network News Protocol, NNTP. Used for the reading and writing news articles structured into newsgroups.
Telnet	854	23	A very, very, generic communication protocol.
Secure Shell	4250-4256	22	A protocol for secure remote login, file transfer, and more.

Hands-on Exercise 1

Let's have a closer look at HTTP

Let's get our hands dirty

<https://blog.baens.net/posts/http-with-telnet/>

<https://blog.baens.net/posts/http-with-telnet/>

Most web pages are written using the Hypertext Markup Language, HTML, which along with HTTP is a fundamental web technology.

The Telnet communication protocol provides a way to establish a direct connection with a remote host. Although not a secure option for most tasks, there are use cases where Telnet is a viable option.

Talking to telnet

How to use telnet to connect to server?

How to create http request in telnet?

How to issue some basic HTTP commands to the web server?

Emulating an HTTP/1.1 Request using Telnet

Enable TELNET on windows

What is Telnet used for?

Terminal programs typically use TELNET to allow you to log into a remote host.

Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Windows\System32>dism /online /Enable-Feature /FeatureName:TelnetClient
```

```
Deployment Image Servicing and Management tool
Version: 10.0.22621.2792
```

```
Image Version: 10.0.22631.4602
```

```
Enabling feature(s)
[=====100.0%=====]
The operation completed successfully.
```

```
C:\Windows\System32>
```

Administrator: Command Prompt

Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>dism /online /En

Deployment Image Servicing and Management
Version: 10.0.22621.2792

Image Version: 10.0.22631.4602

Enabling feature(s)

[=====100.0%=====
The operation completed successfully

C:\Windows\System32>appwhiz.cpl
'appwhiz.cpl' is not recognized as a
operable program or batch file.

C:\Windows\System32>appwiz.cpl

C:\Windows\System32>

Programs and Features

← → ▲ ▼



> Control Panel > Programs > Programs and Features

Control Panel Home

View installed updates

Turn Windows features on or off

Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Organize ▾

Name

- Discord
- Easy Connection to Screen
- Git
- Google Chrome
- Grammarly for Windows
- HP LaserJet Pro M11-M13 S
- HPSSupply
- Intel(R) Chipset Device Software
- Intel(R) Wireless Bluetooth(F)
- Intel® Management Engine
- Intel® Serial IO
- Microsoft 365 Apps for enterprise
- Microsoft ASP.NET Core 7.0
- Microsoft Edge
- Microsoft OneDrive
- Microsoft OneNote - en-us
- Microsoft Teams Meeting Assistant
- Microsoft Update Health Tool

Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

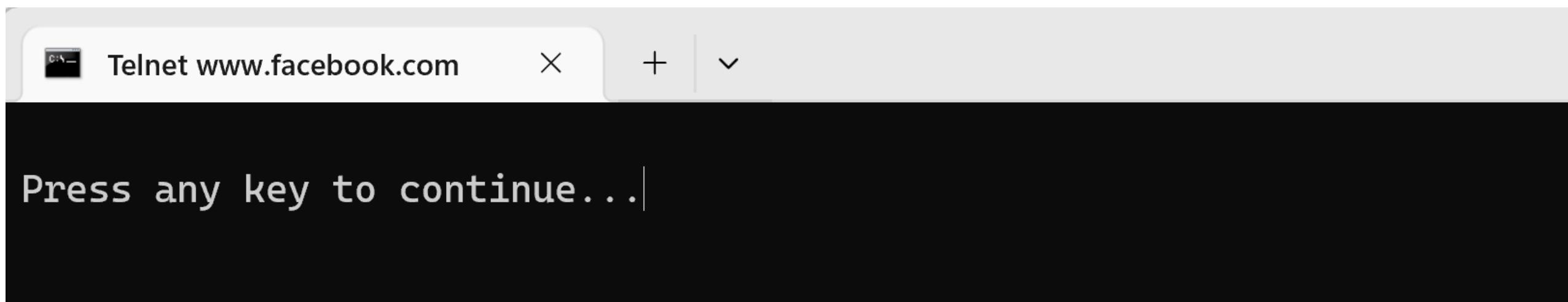
- + .NET Framework 3.5 (includes .NET 2.0 and 3.0)
- + .NET Framework 4.8 Advanced Services
- + Internet Information Services
 - Internet Information Services Hostable Web Core
- + Legacy Components
- + Media Features
- + Microsoft Message Queue (MSMQ) Server
 - Microsoft Print to PDF
 - Microsoft XPS Document Writer
- + Print and Document Services
 - Remote Differential Compression API Support
- + Simple TCPIP services (i.e. echo, daytime etc)
- + SMB 1.0/CIFS File Sharing Support
- + Telnet Client

OK

Cancel

Once you have telnet enabled, follow these steps:

1. Open a command prompt
2. Type in "telnet <IP ADDRESS OF SERVER PC> <PORT>" and press enter.
3. For example, you would type "telnet 123.45.67.89 1521"
4. If a *blank screen* appears then the port is open, and the test is successful.
5. If you receive a connecting... message or an error message then something is blocking that port. It could be the Windows firewall, a third party firewall like your anti-virus software, or an institutional hardware firewall between the workstation and the server.



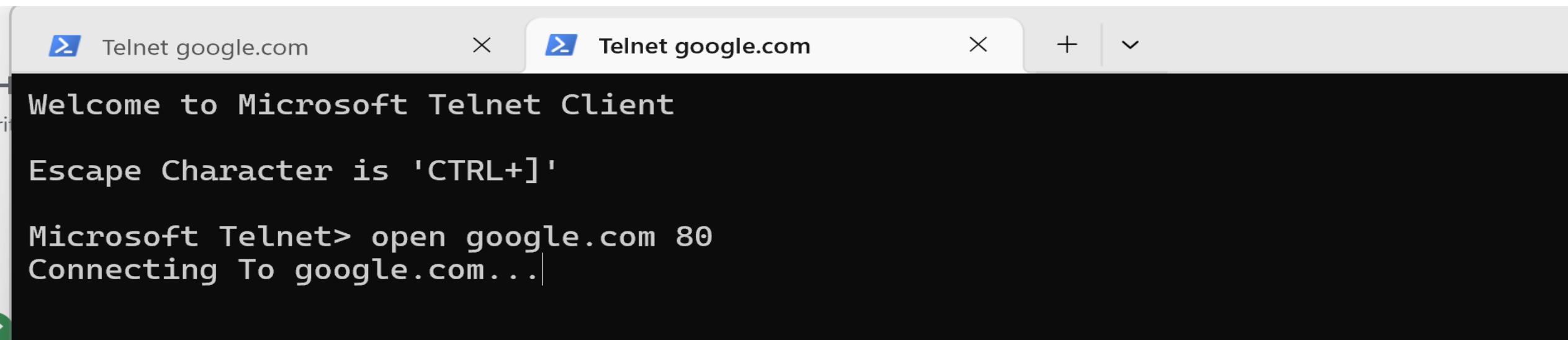
>> telnet

```
Telnet × + ▾  
elcome to Microsoft Telnet Client  
Microsoft Telnet> |  
Escape Character is 'CTRL+]'  
  
Microsoft Telnet> telnet www.facebook.com 80  
Invalid Command. type ?/help for help  
Microsoft Telnet> open www.facebook.com  
Connecting To www.facebook.com...Could not open connection to the host, on port 23: Connect failed  
Microsoft Telnet> open www.facebook.com 80  
Connecting To www.facebook.com...  
  
Connection to host lost.
```

```
Telnet ×  Telnet ×  
Welcome to Microsoft Telnet Client  
Escape Character is 'CTRL+]'  
  
Microsoft Telnet> open www.facebook.com 80  
Connecting To www.facebook.com...  
  
Connection to host lost.  
  
Microsoft Telnet> |
```

```
Microsoft Telnet> status  
Not Connected  
Microsoft Telnet> |
```

How to create a HTTP Request in Telnet



The screenshot shows the Microsoft Telnet Client interface with two windows. Both windows have the title 'Telnet google.com'. The left window displays the welcome message: 'Welcome to Microsoft Telnet Client' and 'Escape Character is 'CTRL+]''. The right window shows the command 'Microsoft Telnet> open google.com 80' followed by 'Connecting To google.com...'. The background of the client window is dark.

```
HTTP/1.0 400 Bad Request
Content-Length: 54
Content-Type: text/html; charset=UTF-8
Date: Thu, 09 Jan 2025 16:51:28 GMT

<html><title>Error 400 (Bad Request)!!1</title></html>

Connection to host lost.

Press any key to continue...|
```

Telnet google.com X + V - O X

HTTP/1.0 400 Bad Request
Content-Type: text/html; charset=UTF-8
Referrer-Policy: no-referrer
Content-Length: 1555
Date: Thu, 09 Jan 2025 16:39:24 GMT

```
<!DOCTYPE html>
  <html lang=en>
    <meta charset=utf-8>
      <meta name=viewport content="initial-scale=1, minimum-scale=1, width=device-width">
    <title>Error 400 (Bad Request)!!1</title>
    <style>
      *{margin:0;padding:0}html,code{font:15px/22px arial,sans-serif}html{background:#fff;color:#222;padding:15px}body{margin:7% auto 0;max-width:390px;min-height:180px;padding:30px 0 15px}* > body{background:url(/www.google.com/images/errors/robot.png) 100% 5px no-repeat;padding-right:205px}p{margin:11px 0 22px;overflow:hidden}ins{color:#777;text-decoration:none}a img{border:0}@media screen and (max-width:772px){body{background:none;margin-top:0;max-width:none;padding-right:0}}#logo{background:url(/www.google.com/images/branding/googlelogo/1x/googlelogo_color_150x54dp.png) no-repeat;margin-left:-5px}@media only screen and (min-resolution:192dpi){#logo{background:url(/www.google.com/images/branding/googlelogo/2x/googlelogo_color_150x54dp.png) no-repeat 0% 0%/100% 100%;-moz-border-image:url(/www.google.com/images/branding/googlelogo/2x/googlelogo_color_150x54dp.png) 0}}@media only screen and (-webkit-min-device-pixel-ratio:2){#logo{background:url(/www.google.com/images/branding/googlelogo/2x/googlelogo_color_150x54dp.png) no-repeat;-webkit-background-size:100% 100%}}#logo{display:inline-block,height:54px;width:150px}
    </style>
    <a href=/www.google.com/><span id=logo aria-label=Google></span></a>
      <p><b>400.</b> <ins>That's an error.</ins>
    <p>Your client has issued a malformed or illegal request. <ins>That's all we know.</ins>
```

Connection to host lost.

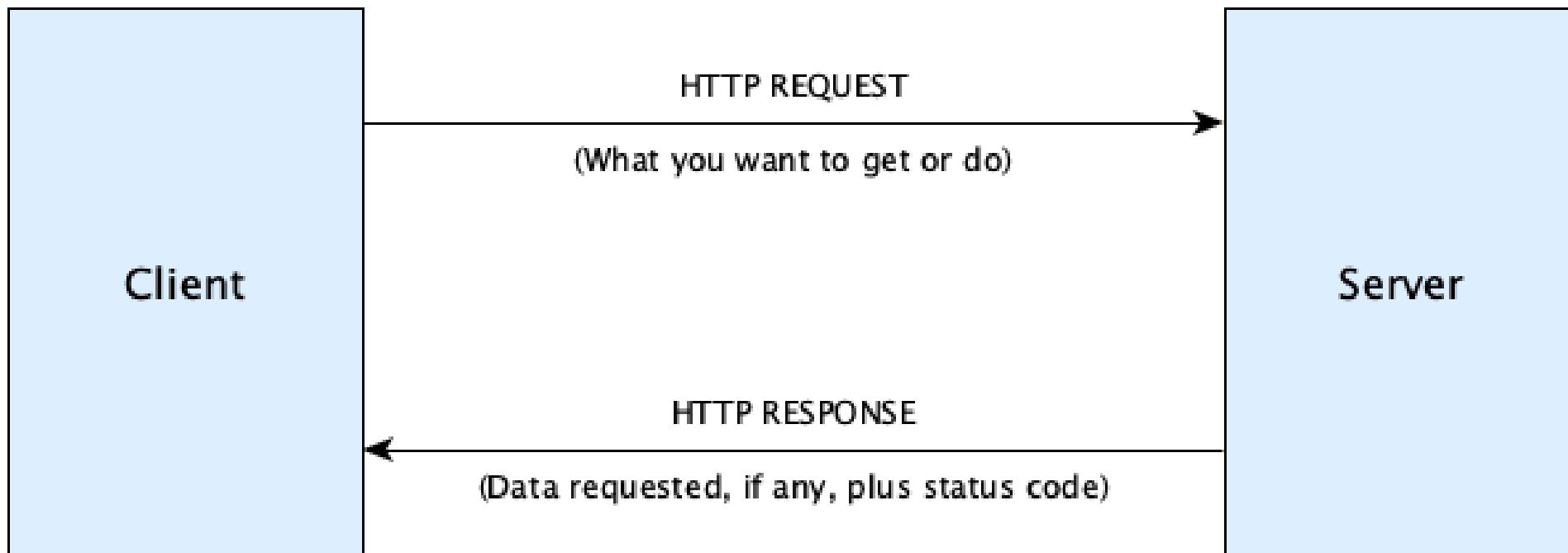
Press any key to continue...|

Website – Web app – web service

Web Application Program

A **web application** is a program that runs on a computer with a **web server** and is accessed via a **web browser** (the “client”) that communicates with the server via the HTTP or HTTPS protocol.

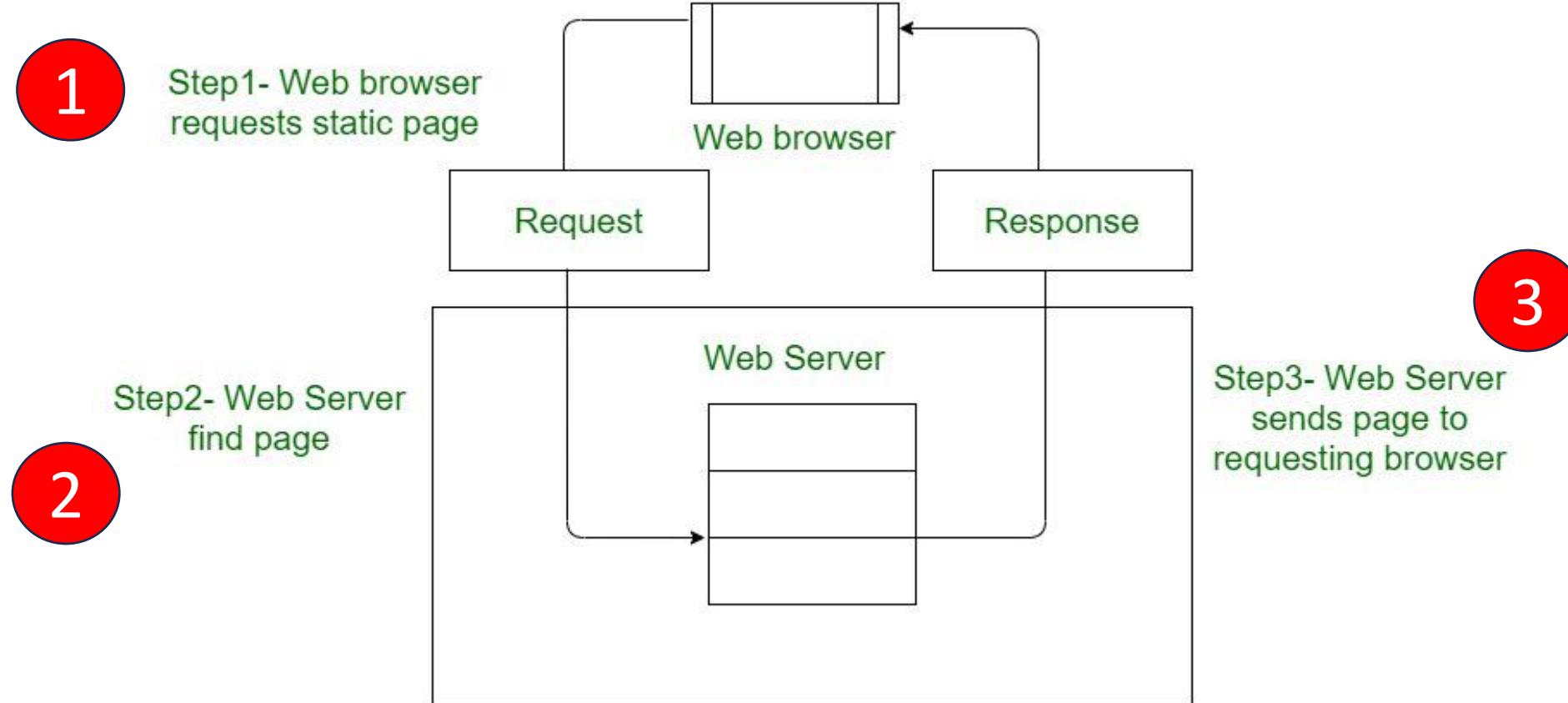
In HTTP(S), the client sends a **request** to the server and the server replies with a **response**:



Website - Web app - WebService

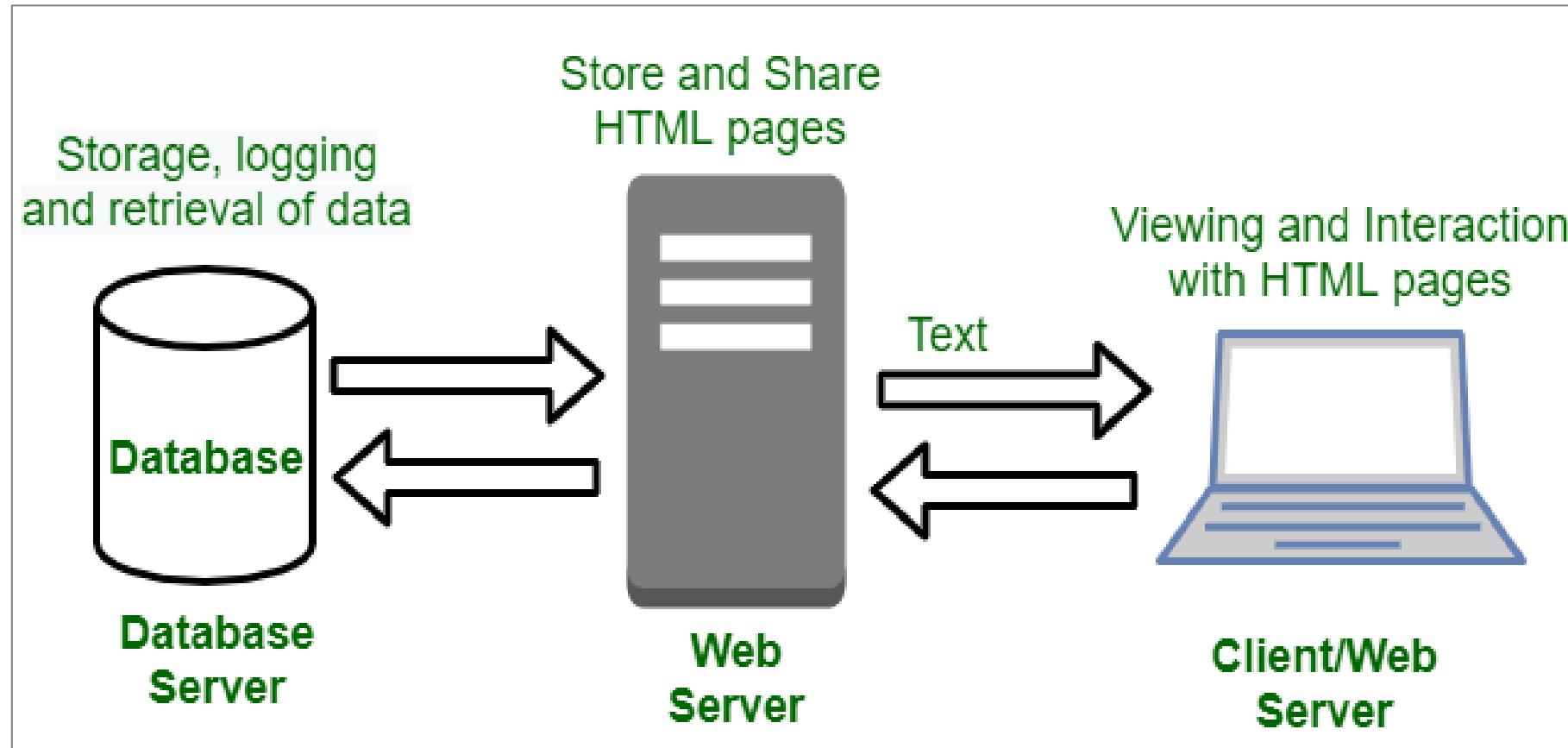
- ✓ If the server always sends back the same data in the form of **HTML**, **CSS**, **images**, **videos**, and similar resources *without any user interaction*, we call the app a **web site**.
- ✓ If there is user interaction (**mouse clicks**, **tapping and swiping**, **phone shaking**, **keypresses**, etc.) that affect the execution of the program, we have a **web app**.
- ✓ If the server only produces raw data (**generally in text or JSON**), then we speak of a **web service**.

Architecture of Static Website

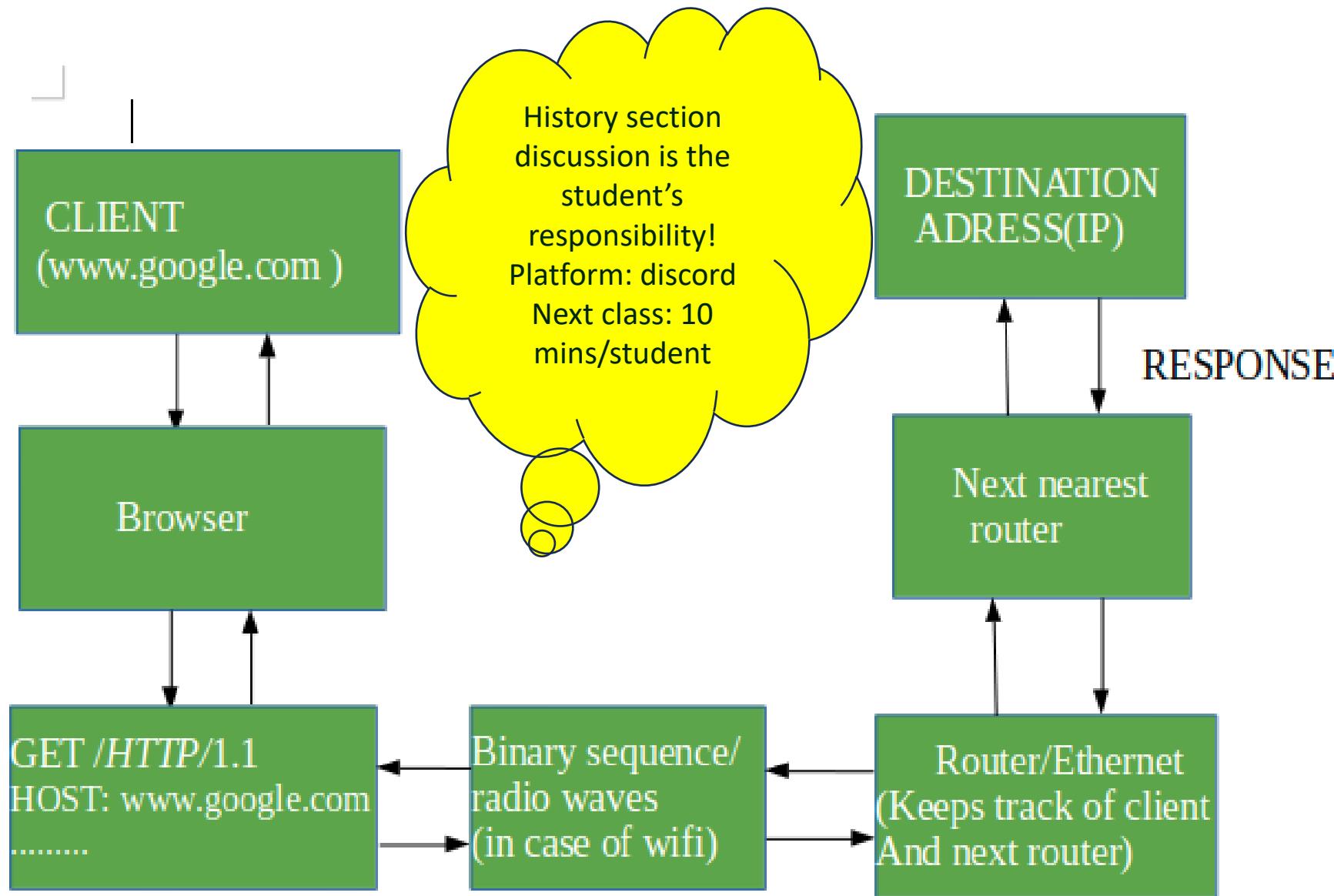


Note: Static does not mean that it will not respond to user actions,

Architecture of Dynamic Website



How Does Internet and Web Programming Work?



Client-side

- ✓ First, when we type a URL like **www.google.com**, the browser converts it into a file containing
 - GET /HTTP/1.1 (where GET means we are requesting some data from the server and HTTP refers to a protocol that we are using, 1.1 refers to the version of the HTTP request)
 - Host: www.google.com
 - And some other information
- ✓ Now this file is converted to binary code by the browser and it is sent down the wires if we are connected through Ethernet and if we are using WiFi, first it converts it to a radio signal which is decoded by a router in a very low level. It is converted to binary and then sent to the servers. This information or ‘binary codes’ go to the destination and responds if it is received by the sender only because of the IP address. One router will send the information to another and this keeps on going until the binary codes reach the destination.

Server-side

✓ Now the server receives the binary code and decodes it and sends the response in the following manner:

- HTTP/1.1 200 ok (where 200 ok is the status)
- Content-type : type/HTML
- Body of page

Now, this is converted back to binary by the server and sent to the [IP address](#) that is requesting it. Once the codes are received by the client, the browser again decodes the information in the following way

- First, it checks the status
- It starts reading the document from the HTML tag and constructs a Tree-like structure.
- The HTML tree is then converted to corresponding binary code and rendered on the screen.
- In the end, we see the website front-end.
- Soon we will understand the tree structure of the HTML document

The HTTP request/response transaction is summarized in Figure



Figure 4: An HTTP request and response pair are how a web browser gets web pages from a web server.

Now it is time for..... !

Writing and Deploying Simple Web Apps



Static Website	Dynamic Website
Content of Web pages can not be change at runtime.	Content of Web pages can be changed.
No interaction with <u>database</u> possible.	Interaction with <u>database</u> is possible
It is <u>faster</u> to load as compared to dynamic website.	It is <u>slower</u> than static website.
Cheaper Development costs.	More Development costs.
No feature of Content Management.	Feature of Content Management System (CMS).
HTML, CSS, Javascript is used for developing the website.	Server side languages such as PHP, Node.js are used.
Same content is delivered everytime the page is loaded.	Content may change everytime the page is loaded.

Angular or React?



INSTAGRAM



WHATSAPP



GENERAL MOTORS



UPWORK



FACEBOOK



AIRBNB



GOOGLE



HBO

Uber

UBER

NETFLIX

NETFLIX



Forbes

FORBES



DROPBOX

SONY

SONY

Flavors of HTML and XHTML

In the early days, the specification of HTML was somewhat fluid, and browser vendors of all sizes added their own elements. First the **Internet Engineering Task Force (IETF)** and later the **World Wide Web Consortium (W3C)** set standards for HTML and its cousin XHTML.

```
<p>This is a paragraph.  
<p>This is also a paragraph.
```

```
<p>This is a paragraph.</p>  
<p>This is also a paragraph.</p>
```

```
<br>
```

However, in XML markup variants, particularly XHTML, an unclosed tag is not allowed, so you need to close the tag

```
<br></br>
```

or, more commonly, use a self-identification of closure like so:

```
<br />
```

```

1  <!-- The original html recipe -->
2  <HTML>
3  <HEAD>
4  <TITLE>MY Donations to online thrift store</TITLE>
5  </HEAD>
6  <BODY>
7  <H3>My Donations to online thrift store</H3>
8  My favorite child hood books (urdu+english).
9  <H4>Description or Details</H4>
10 <TABLE BORDER="1">
11 <TR BGCOLOR="#308030"><TH>Qty</TH><TH>type</
12 TH><TH>Book</TH></TR>
13 <TR><TD>1</TD><TD>series</TD><TD>Dasstan-AmeerHamza</
14 TD></TR>
15 <TR><TD>3</TD><TD>series</TD><TD>Harry Potter</TD></
16 TR>
17 <TR><TD>1</TD><TD>seasons</TD><TD>Hatim Tai</TD></TR>
18 <TR><TD></TD><TD>suspense</TD><TD>Ishtique Ahmed</
19 TD></TR>
20 </TABLE>
<P>
<H4>Instructions</H4>
<OL>
<LI>Please read and share.....</LI>

```

Equivalent
XHTML ???

My Donations to online thrift store

My favorite child hood books (urdu+english).

Description or Details

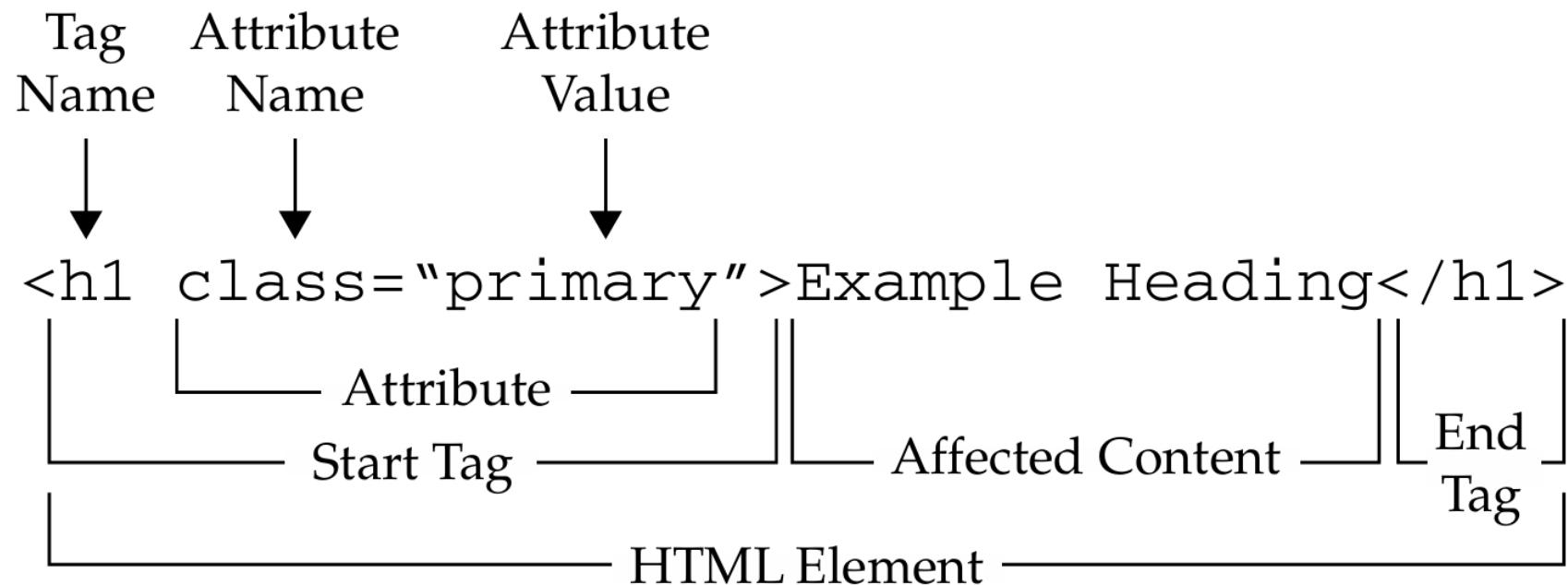
Qty	type	Book
1	series	Dasstan-AmeerHamza
3	series	Harry Potter
1	seasons	Hatim Tai
	suspense	Ishtique Ahmed

Instructions

1. Please read and share.....

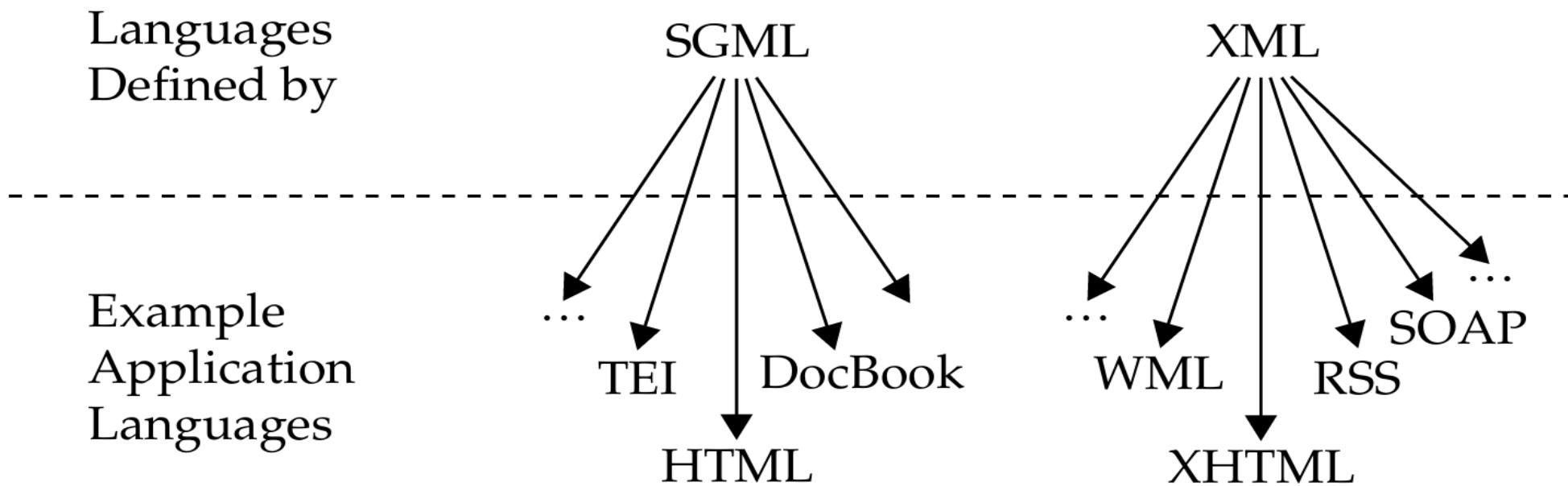
HTML markup syntax

A graphical overview of the HTML markup syntax shown so far is presented here:



The relationship between the various markup technologies

Traditionally, the W3C defined HTML as an application of the **Standard Generalized Markup Language** (SGML). SGML is a technology used to define markup languages by specifying the allowed document structure in the form of a **document type definition (DTD)**. A DTD indicates the syntax that can be used for the various elements of a language such as HTML.



Core HTML and XHTML Attributes Reference

The HTML and XHTML specifications provide four main attributes that are common to nearly all elements and have much the same meaning for all elements. These attributes are class, id, style, and title

class	id	style	title
-------	----	-------	-------

Attribute	Importance	Valid for
class	class for CSS selection	all elements
id	identifier for CSS and scripts	all elements
style	CSS style sheet	displayed elements
title	element title, additional info in the bubble	all elements
lang	element language	all elements
name	name for collaboration with other elements	different usability
access_key	function key	active elements
tab_index	Tab activation order	active elements
autofocus	the element automatically gets focus after the page loads	active elements
language	script language	all elements
director	text direction	especially <bdo>
contenteditable	user can edit content	about everything
enterkeying	defines the behavior of the Enter key	editable elements
translate	instructions for language translators	about everything
events	user events that trigger scripts	displayed elements

An element can also have **attributes**, which must appear in the start tag, and **content**, which appears between the tags. The HTML file does not define how the elements look nor how they behave.

COMMON PROGRAMMING ERROR

XHTML does not permit tags to overlap—a nested element's end tag must appear in the document before the enclosing element's end tag.

For example, the nested XHTML tags

```
<head><title>hello</head></title>
```

cause a syntax error, because the enclosing `head` element's ending

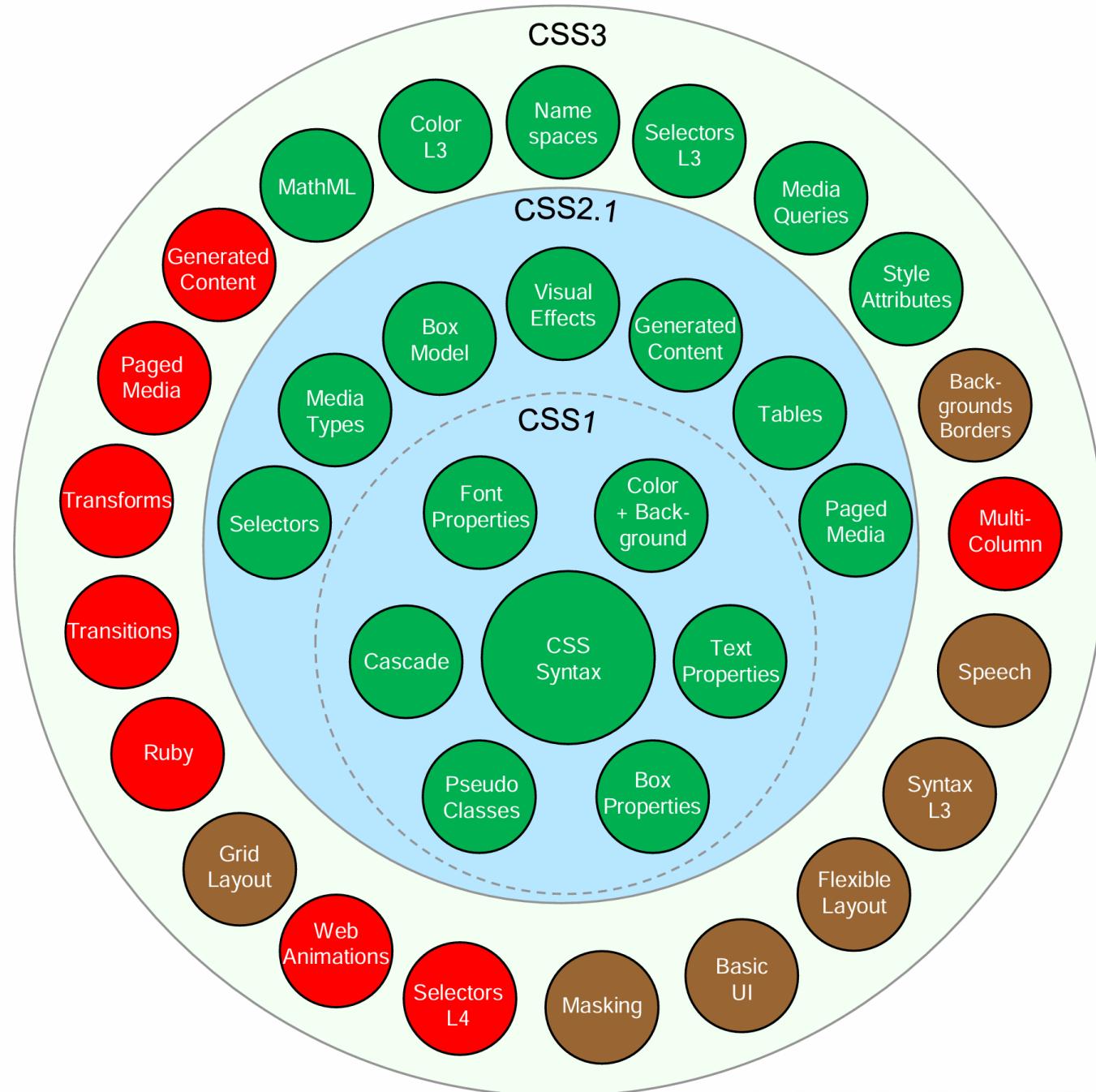
`</head>` tag appears before the nested `title` element's ending `</title>` tag.



● W3C Recommendation

● W3C Candidate Recommendation

● Working Draft



CSS1 specification published in 1996

- ✓ remember that HTML 3.2 introduced some elements and attributes (e.g. color) for the visual appearance in 1997

CSS2 specification published in 1998

- ✓ superset of CSS1
- ✓ functionality for relative, absolute and fixed positoning of elements
- ✓ media types

CSS3 divided into separate modules (documents)

- ✓ 2D & 3D transforms
- ✓ transitions
- ✓ Flexbox
- ✓ media queries
- ✓

What is CSS?

CSS is a language that applies styles to a HTML document and its elements to change the look and feel and is usually stored in separate .css style which can be re-used for all the web pages. A website is made up of HTML for content plus CSS for appearance.

HTML + CSS = WEB PAGE
(Content) (Presentation)

Disadvantage of CSS

Browser Compatibility: Some style sheet features are supported and some are not by the browsers.

CSS Advantages and Disadvantages

- ✓ **CSS saves time** - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- ✓ **Pages load faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply to all the occurrences of that tag. So less code means faster download times.
- ✓ **Easy maintenance** - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- ✓ **Superior styles to HTML** - CSS has a much wider array of attributes than HTML so you can give far better look to your HTML page in comparison of HTML attributes.
- ✓ **Multiple Device Compatibility** - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- ✓ **Global web standards** - Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

CSS Syntax

CSS rule consists of two parts

- a **selector**
 - HTML element
- a declaration block (at least one **property** and **value**)
 - surrounded by curly braces
 - multiple properties are separated with a semicolon

```
selector {  
    property1: value1;  
    property2: value2;  
}
```

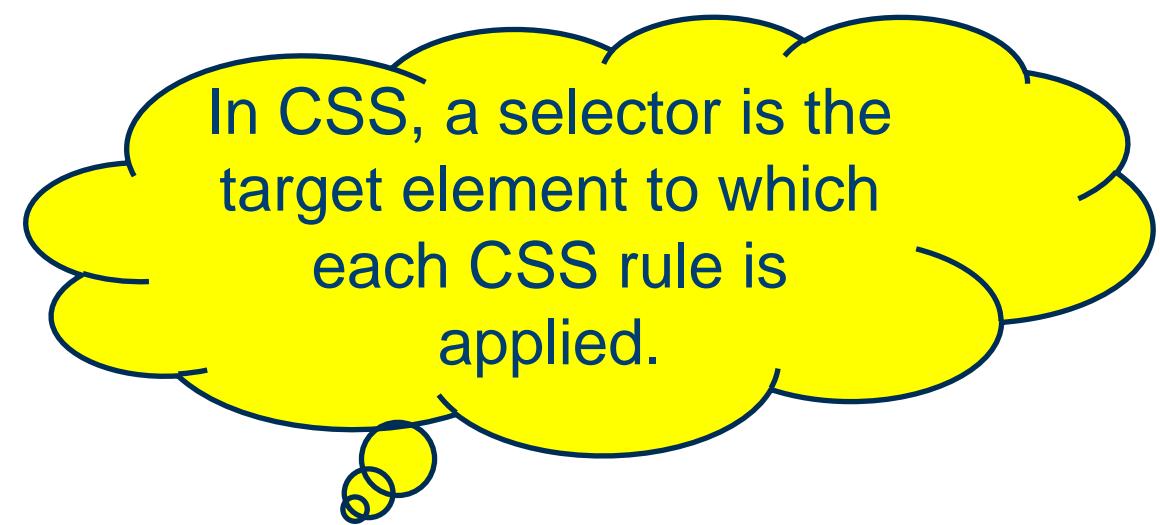
Example

```
h1 {  
    color: red;  
    font-size: 10px;  
}
```

CSS Selectors

Selectors are used to target rules to specific elements in an HTML document (case sensitive)

- ✓ Universal selector
- ✓ Element/Type selector
- ✓ ID selector
- ✓ Class selector
- ✓ Child selector
- ✓ Descendant selector
- ✓ Attribute selector
- ✓ Pseudo element selectors
- ✓ Pseudo class selector



UNIVERSITY OF
KARACHI

Let's create:

- (1) HTML to describe our app's **content**;
- (2) CSS to describe our app's **presentation**, and
- (3) JavaScript, to program the app's **interactivity**.

Note: We can include images, video, text, and other information as well.

All these resources are all placed on the server, and the clients will use a **web browser** to fetch and render them.

Now where can you **host** your web app? That is, where is your web server? You can get your own server of course, or use one provided by a service. Services for simple web apps include [The p5js Web Editor](#), [Replit](#), [Glitch](#), and [Code Sandbox](#).

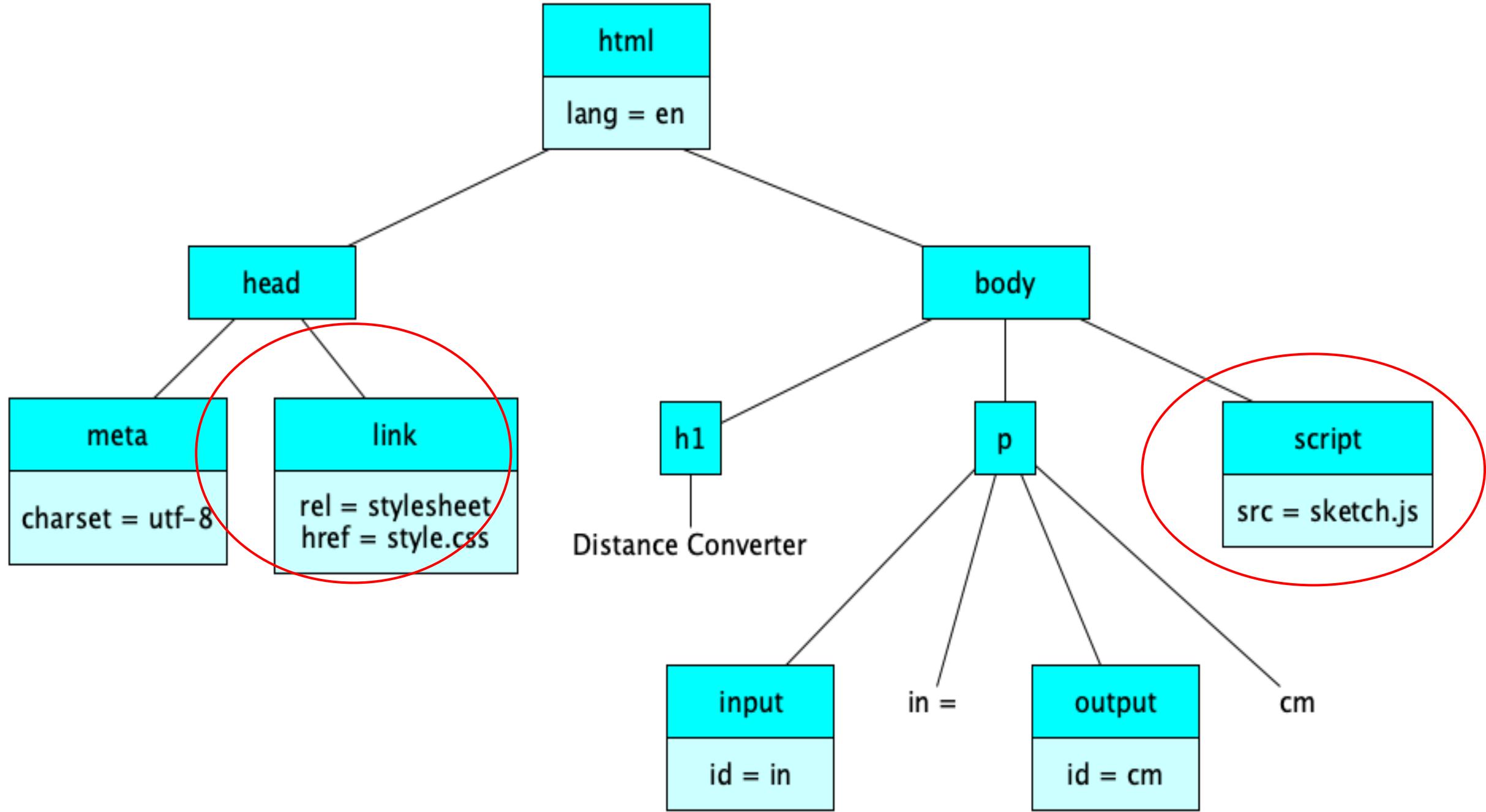
Example Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Distance Converter</h1>
    <p>
      <input id="in" type="text" value="in = ">
      <output id="cm" type="text" value="? cm">
    </p>
    <script src="script.js"></script>
  </body>
</html>
```

Distance Converter

 in = ? cm

An HTML file only states what **elements** should appear in your document. The head contains information about the document; the body contains what you see



What do we mean by presentation?

HTML defines only the structure of an app; CSS defines its **presentation**, sometimes called the **look-and-feel**. What do we mean by presentation? Lots of things:

- Text and Fonts
- Layout, Position, Alignment
- Border, Margins, Padding
- Colors
- Backgrounds
- Image Processing
- Scrolling and Overflow
- Paging
- Masking
- Transformations
- Transitions
- Audio and Speech

This stylesheet has **three style rules**. A style rule has a **selector** and a **property list**. The property list is a list of property-value pairs.

```
body {  
    margin: 0;  
    padding: 0;  
    background-color: skyblue;  
}  
  
h1 {  
    background-color: blue;  
    color: white;  
    margin-top: 0;  
    padding: 20px;  
    font-family: cursive;  
}  
  
p {  
    margin-left: 20px;  
}
```

Distance Converter

inches = ? cm

JavaScript is for **behavior**

Here's the JavaScript code: **script.js**

When event X occurs, perform action Y.

In our simple distance-converter app, we want to say:

When anything is input in our text box, update the output element with the new computed value.

```
const inchesInput = document.querySelector("#in")
const cmOutput = document.querySelector("#cm")

inchesInput.addEventListener("input", report)

function report() {
  const inches = inchesInput.value
  const centimeters = inches * 2.54
  if (isNaN(centimeters)) {
    cmOutput.textContent = "?"
  } else {
    cmOutput.textContent = centimeters
  }
}
```

Distance Converter

in = ? cm

JavaScript program has three main parts

1. Get HTML elements into JavaScript variables by calling `document.querySelector` with a **selector expression**.
2. Attach **event listeners** to certain elements, so that when an event happens on an element, a particular function is called.
3. Implement the functions to **handle the events**, generally by updating **properties on elements** (which will automatically make the browser update its display).

Two input boxes

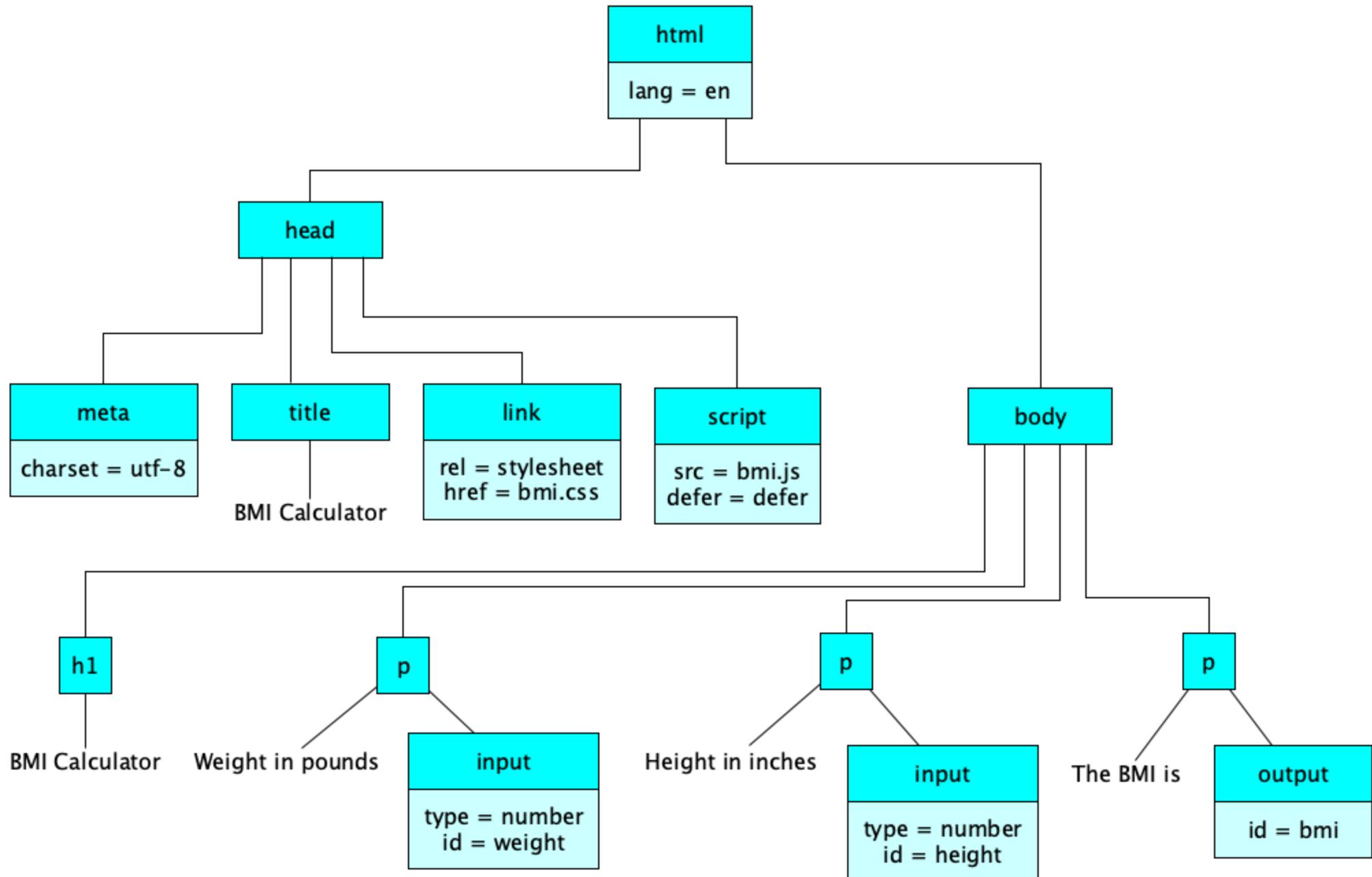
```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>BMI Calculator</title>
    <link rel="stylesheet" href="bmi.css">
    <script src="bmi.js" defer></script>
  </head>
  <body>
    <h1>BMI Calculator</h1>
    <p>Weight in pounds <input id="weight"></p>
    <p>Height in inches <input id="height"></p>
    <p>The BMI is <output id="bmi"></output></p>
  </body>
</html>
```

BMI Calculator

Weight in pounds

Height in inches

The BMI is





Save

Fork

Share



Static Starter



b Open in bolt.new | AI →



PROJECT



Create a repository



> INFO

FILES

bmi.css

index.html

package-lock.json

{...} package.json

page2.html

script.js

script.js

index.html X

bmi.css



...



stackblitzstartersvsvla...-40ls--8...



```
1  <!doctype html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>BMI Calculator</title>
6          <link rel="stylesheet" href="bmi.css">
7      </head>
8      <body>
9          <h1>BMI Calculator</h1>
10
11         <p>Weight in pounds <input id="weight"></p>
12         <p>Height in inches <input id="height"></p>
13         <p>The BMI is <output id="answer"></output></p>
14
15         <script src="bmi.js"></script>
16
17     </body>
18
19 </html>
```

JavaScript code can be placed in the `<head>`, in the `<body>` or in external files

BMI Calculator

Weight in pounds

Height in inches

The BMI is



UNIVERSITY OF
KARACHI



"Don't be satisfied with stories, how things have gone with others. Unfold your own myth." ~Rumi



UNIVERSITY OF
KARACHI



Department of Compute Science (UBIT Building), Karachi, Pakistan.

1200 Acres (5.2 Km sq.)

53 Departments

19 Institutes

25000 Students

My Homeland Pakistan

