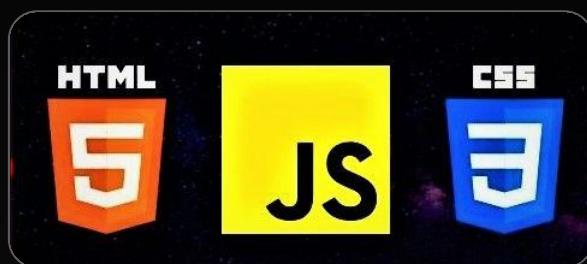




Sheryians Coding  
School

# Live Cohort

## Notes Day 1



# Topic 1: How the Internet Works

## 1 . Brief History: Web 1.0 → Web 3.0

### Web 1.0 (1990s – early 2000s) – The "Read-Only" Web

- Websites were static with no interactivity.
- Built using HTML and CSS (no JavaScript).
- Users could only read content; there was no user engagement.
- No social media or real-time interactions.
- Example: Early Yahoo! pages, simple blogs.

### 🌐 Web 2.0 (2004 – Present) – The "Read-Write" Web

- Enabled dynamic and interactive experiences.
- Technologies like AJAX and JavaScript allowed live updates without page reloads.
- Social media, e-commerce, and cloud applications emerged.
- Users could generate content (comments, reviews, blog posts, real-time chats).
- Example: Facebook, YouTube, Twitter, Amazon.

### 🚀 Web 3.0 (Emerging) – The "Decentralized" Web

- Uses blockchain technology to ensure decentralization.
- AI-driven and privacy-focused to give users more control over data.
- Example: Cryptocurrencies, smart contracts, decentralized apps (Ethereum, Filecoin).

# Topic 1: How the Internet Works

## 2 . Basics of Computer Communication & Data Transmission

### 📡 Data Transmission & Packets

- The internet sends data in packets, which are small chunks of information.
- Each packet has:
  - Header (destination and sender information).
  - Payload (actual data being transferred).
- Uses TCP/IP protocols to ensure data reaches the right destination.

### 🔗 Two Types of Communication

- Wired (Ethernet, Fiber Optics) – More stable and faster; used in offices.
- Wireless (Wi-Fi, 5G) – Provides flexibility but can be slower.

# Topic 1: How the Internet Works

## 3 . Domain Names, IP Addresses, MAC Addresses, and Routing

### IP Address (Internet Protocol Address)

- A unique number assigned to every internet-connected device.
- IPv4 (xxx.xxx.xxx.xxx) has 4.3 billion addresses.
- IPv6 provides a larger address space for the future.

### MAC Address (Media Access Control)

- A unique identifier assigned to a device's network adapter.
- Example: 00:1A:2B:3C:4D:5E.
- Used within local networks but not on the broader internet.

### Domain Names (Example: www.google.com)

- Human-friendly names mapped to IP addresses via DNS (Domain Name System).
- DNS servers resolve domain names into IP addresses.

### Routing

- Routers and gateways direct traffic efficiently.
- Packets take different routes to reach the destination efficiently.

# Topic 1: How the Internet Works

## 4 . Overview of ISP & DNS Functionality

### Internet Service Providers (ISP)

- Companies like Jio, Airtel, BSNL provide internet access.
- They act as intermediaries between users and the global internet.

### How DNS Works

- A user types www.google.com in a browser.
- The browser contacts a DNS server to get the IP address.
- The DNS translates the domain into an IP (e.g., 142.250.183.206).
- The browser connects to the website's server using the IP.

# Topic 2: Client-Server Architecture

## 1 . Introduction to the Client-Server Model

### 📌 What is a Client?

- A user's device (computer, phone, tablet) that requests data from a server.
- Examples: Web browsers, mobile apps.

### 📌 What is a Server?

- A powerful computer that stores and delivers content or services.
- Examples: Web servers, API servers, database servers.

### 📌 Example:

A client is like a customer at a restaurant, and the server is like a waiter bringing food from the kitchen.

# Topic 2: Client-Server Architecture

## 2 . Differences Between Client and Server

Feature	Client	Server
Who uses it?	End-user	Handles requests
Example	Web browser	Website backend
Processes	Displays UI	Manages logic, databases

Example:

- You visit YouTube → The client sends a request.
- The YouTube server processes it and responds with videos.

# Topic 2: Client-Server Architecture

## 3 . HTTP Request/Response Cycle

### How It Works

- Client sends a request (e.g., GET /index.html).
- Server processes the request.
- Server responds with data.
- Client displays the content.

### HTTP Methods

- GET → Retrieve data.
- POST → Send data (forms, logins).
- PUT → Update data.
- DELETE → Remove data.

### Example:

A restaurant order – You (client) order pizza (GET), the kitchen (server) prepares it, and you receive it.

# Topic 2: Client-Server Architecture

## 4 . Frontend vs Backend, Static vs Dynamic Websites

### ★ Frontend (Client-Side)

- Code that runs in the browser.
- Technologies: HTML, CSS, JavaScript.

### Backend (Server-Side)

- Code that runs on the server.
- Technologies: Node.js, Python, PHP.

### Static vs Dynamic Websites

Type	Static Websites	Dynamic Websites
Example	Simple blogs, portfolios	Facebook, Amazon
Data Changes	Fixed content	Updates dynamically
Backend Needed?	✗ No	✓ Yes

### 📌 Example:

A static website is like a printed book, while a dynamic website is like a news website that updates regularly.

# Topic 2: Client-Server Architecture

## 5 . Basics of Web Hosting

- Web hosting stores websites for online access.
- Examples: Shared Hosting, VPS, Cloud Hosting.
- Popular Providers: GoDaddy, Hostinger, AWS, Netlify.

### 📌 Analogy:

Hosting is like renting an apartment for your website