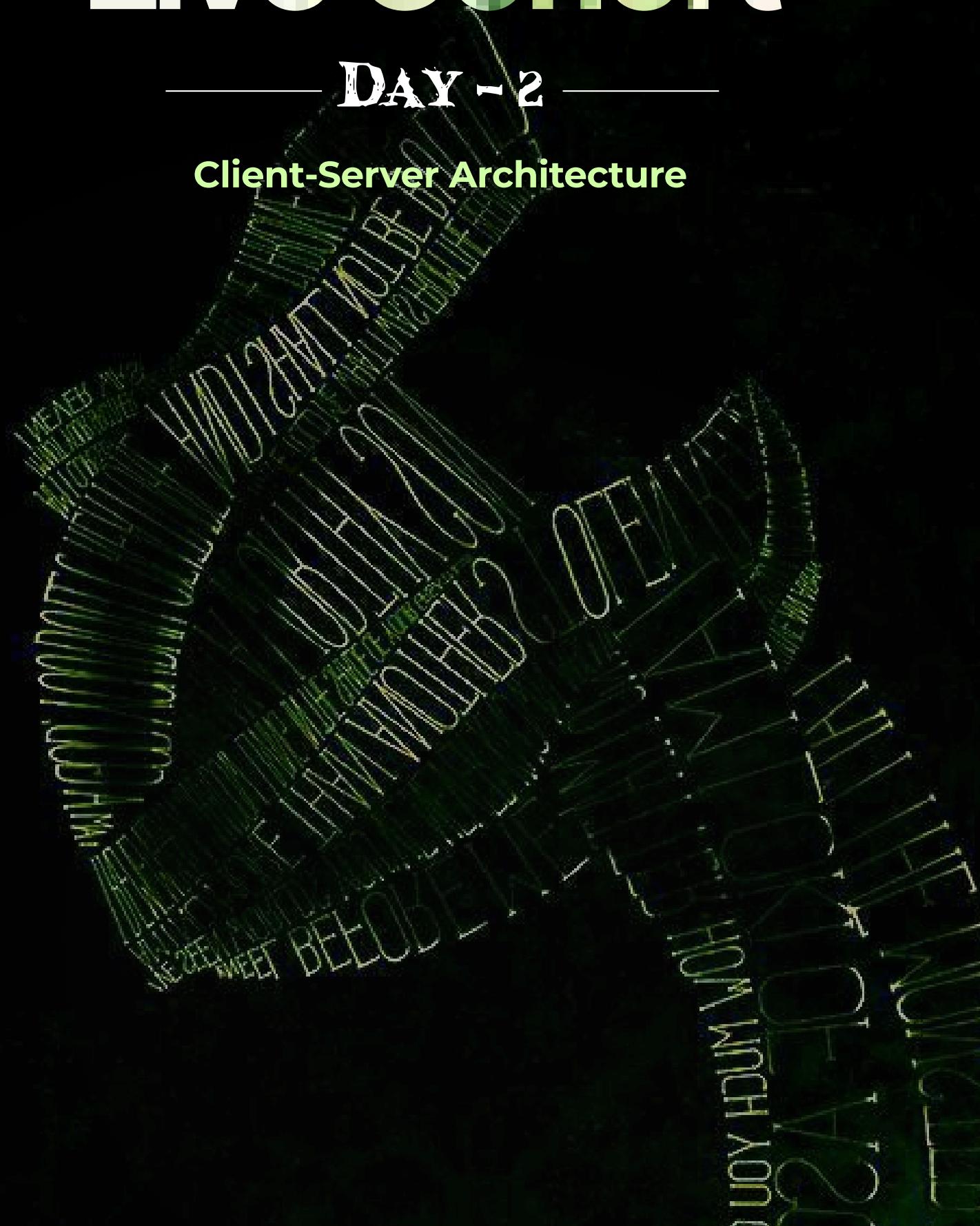


Live Cohort

DAY - 2

Client-Server Architecture



Client-Server Architecture

1. What is Client-Server Model

The client-server model is the foundation of how the internet works.

- A **Client** is a device or program (like a browser or mobile app) that requests data or services.
- A **Server** is a computer system that stores, processes, and delivers the requested data to clients.
- Relationship: **Client asks → Server responds.**

◆ Example

You (client) request google.com in your browser → Google's server responds with the webpage.

2. Difference Between Client and Server

◆ Client (Browser)

- Runs on the user's device (laptop, phone).
- Sends requests for resources like HTML, CSS, JavaScript, or images.
- Displays the fetched content to the user.

◆ Server

- A more powerful computer that stores websites, applications, or databases.
- Listens for incoming requests and responds accordingly.
- Stays **always online** and **connected** to ensure availability.

Client-Server Architecture

3. HTTP Request-Response Cycle

The communication between client and server happens in a cycle known as the **HTTP request-response cycle**.

1. **User Action** → You type a URL or click a link.
2. **Browser sends an HTTP request to the server.**
3. **Server processes the request** (fetch data, execute logic, access database).
4. **Server sends an HTTP response back** (HTML, JSON, images, or files).
5. **Browser renders the page** and shows it to the user.

👉 This entire cycle usually takes place within **milliseconds**.

4. What Happens When You Visit a Website

1. You type www.example.com in your browser.
2. Browser contacts the **DNS** to get the IP address of the website.
3. Browser connects to the correct server via your **ISP (Internet Service Provider)**.
4. Browser sends an HTTP/HTTPS request.
5. Server locates the requested page or data and sends it back.
6. Browser displays the website to you.

5. Frontend vs Backend

◆ Frontend (Client-Side)

- The **visible part** of a website or application.
- Built using **HTML, CSS, JavaScript**.
- Focuses on **UI/UX** (buttons, forms, visuals, design).
- ↗ Example: The login form you see on your screen.

Client-Server Architecture

◆ Backend (Server-Side)

- The hidden part of the system that runs on the server.
- Handles **logic, authentication, database operations, APIs**.
- ✅ Example: Checking whether your login credentials are correct.

6. Static vs Dynamic Websites

◆ Static Website

- Shows the **same content to everyone**.
- Built with **pre-written HTML files**.
- Very fast but limited in interactivity.
- ✅ Example: Portfolio sites, simple company info pages.

◆ Dynamic Website

- Content **changes based on user input or data**.
- Built using **backend logic + database integration**.
- ✅ Example: Facebook's news feed or Amazon's product pages.

7. What is Web Hosting

Web hosting means renting space on a server to store your website files so that they can be accessed online.

- Hosting providers like **Hostinger, AWS, GoDaddy** keep your site running and accessible 24/7.
- Without hosting, your site cannot be seen on the internet.

Client-Server Architecture

◆ Types of Web Hosting

- **Shared Hosting:** Multiple websites share one server (cheapest, but slower).
- **VPS / Dedicated Hosting:** A server is divided or dedicated for one website (more power, more cost).
- **Cloud Hosting:** Scalable and flexible hosting (e.g., AWS, Google Cloud, Microsoft Azure).

