Web Development Basics

- Presenting by Nishitha S

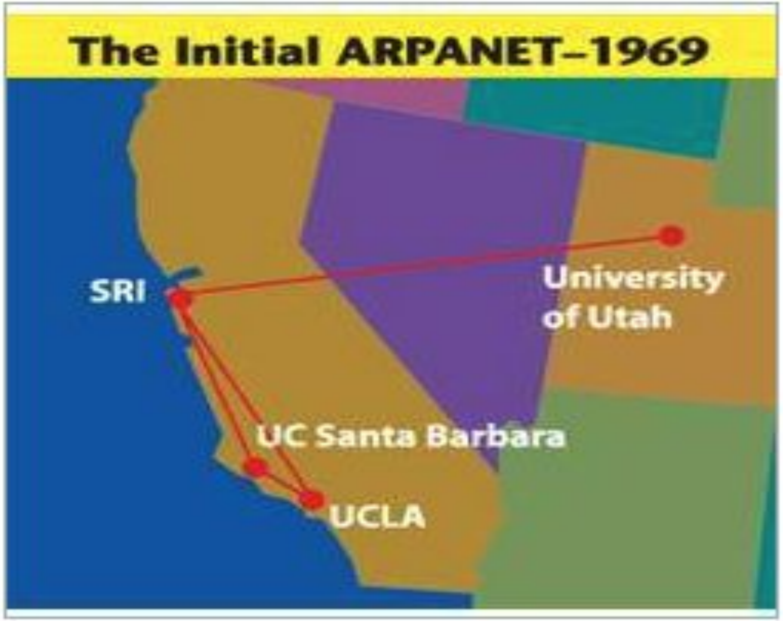
</> HTML,CSS,JAVASCRIPT
& JQUERY </>

Architecture of the Web

From Internet Origins to Modern Web Technologies

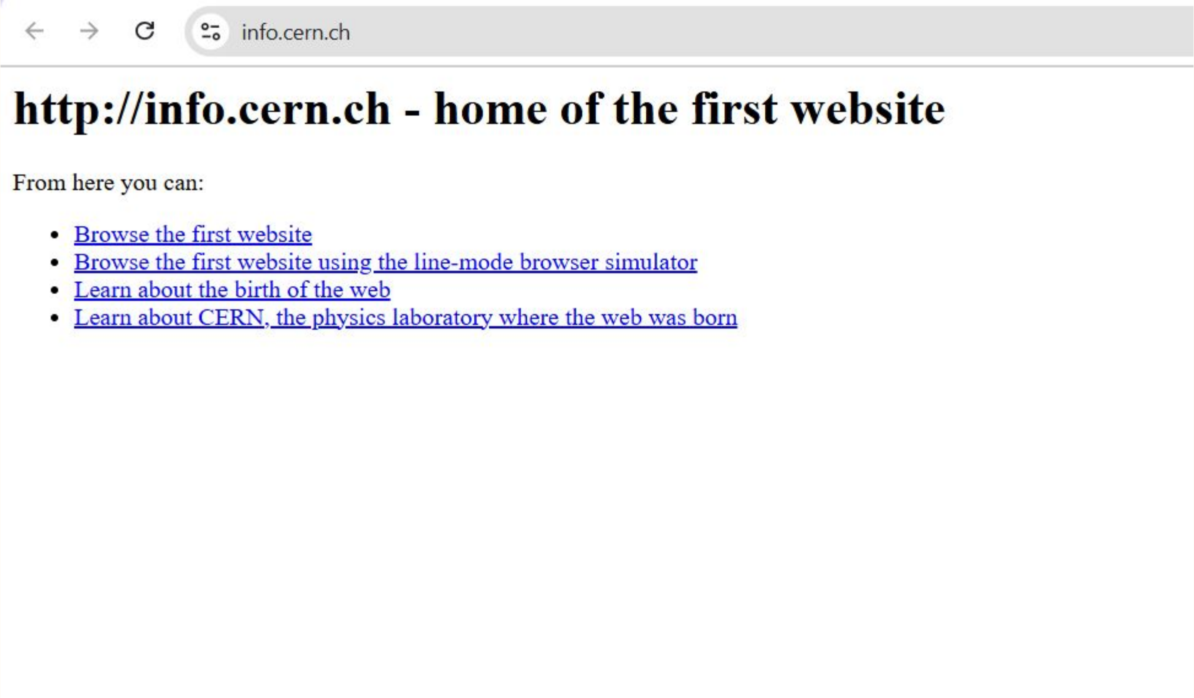
The Internet Timeline

- **1957**: ARPA established
- **1969**: ARPANET connects first 4 nodes
- **1983**: TCP/IP protocols adopted
- **1989**: World Wide Web invented by Tim Berners-Lee
- **1993**: Mosaic browser released
- **2000s**: Web 2.0 emerges
- **2010s**: Mobile dominance
- **2020s**: Web 3.0 concepts



The humble beginnings of today's global network

First Web Page Ever



http://info.cern.ch - home of the first website

From here you can:

- Browse the first website
- Browse the first website using the line-mode browser simulator
- Learn about the birth of the web
- Learn about CERN, the physics laboratory where the web was born

Created by Tim Berners-Lee at CERN

How Does the Internet Actually Work?

- Internet backbone
- ISPs (Internet Service Providers)
- Data centers

Data Transmission

- Data as packets
- Packets travel independently through the network
- Packets are reassembled at the destination

Internet Protocol: The Addressing System

IPv4

Format:

- IPv4 addresses are **32-bit** numeric addresses.
- Written in **dotted decimal format**, divided into **four octets** (each 8 bits).
- Example: `192.168.1.1`

Range: Total possible addresses: **2^{32} = ~4.3 billion**

IPv6

Format:

- Pv6 addresses are **128-bit** hexadecimal addresses.
- Written in **colon-separated** format.
- Example: `2001:0db8:85a3:0000:0000:8a2e:0370:7334`

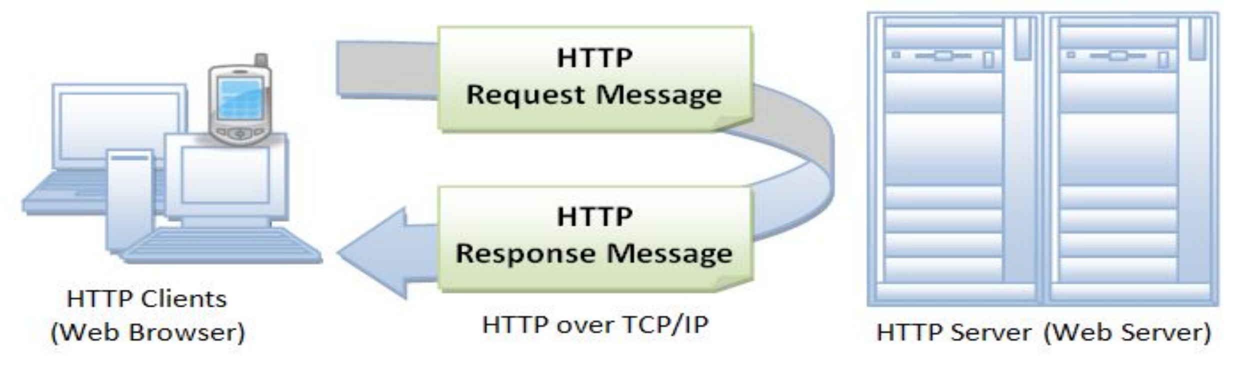
Range: Total possible addresses: $\mathbf{2^{128} = 3.4\times10^{38}}$

What's Your IP Address?

HTTP: The Language of the Web

Application protocol for transmitting hypermedia documents

Client-server model: browser requests, server responds



HTTP Evolution

- **HTTP 0.9** (1991): Simplest version, only GET method
- **HTTP 1.0** (1996): Added headers, status codes, multiple content types
- **HTTP 1.1** (1997): Added persistent connections, pipelining, caching controls
- **HTTP/2** (2015): Binary instead of textual, multiplexed connections, server push
- **HTTP/3** (2022): faster connection establishment

HTTP Methods

- **GET**: Request data from a resource
- **POST**: Submit data to be processed
- **HEAD**: Same as GET but returns only headers
- **PUT**: Update a resource
- **DELETE**: Remove a resource

HTTP Status Codes: The Web's Response System

- 1xx: Informational
- 2xx: Success (200 OK)
- 3xx: Redirection (301, 302)
- 4xx: Client Error (404, 403,401)
- 5xx: Server Error (500)

Domain Names: The Web's Address Book

User → Browser → OS Cache → DNS Resolver → Root Server

↓

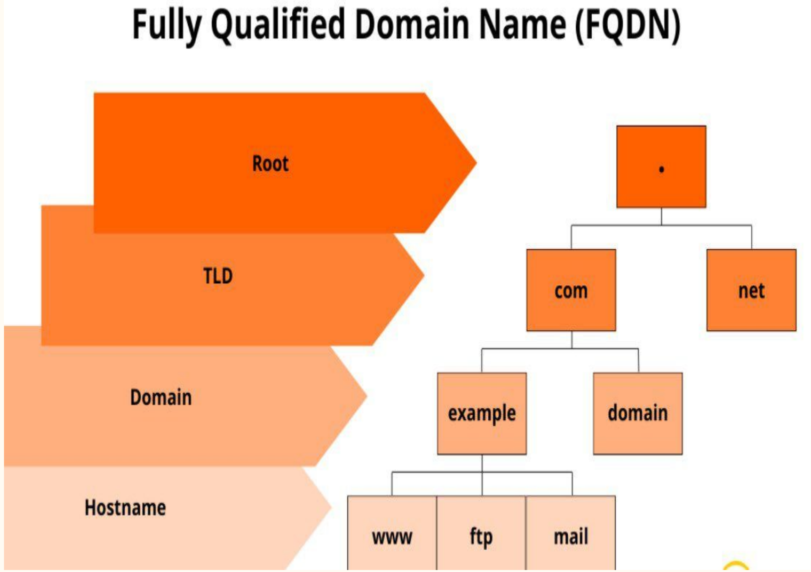
TLD Server (.com)

↓

Authoritative Server (example.com)

↓

IP Returned to User



How DNS Works: Step by Step

Step 1: Check browser cache

Step 2: Check OS cache

Step 3: Query recursive DNS server

Step 4: Query authoritative DNS server

HTTP's Memory Problem

Stateless Protocol

- Each request/response cycle is independent
- Server doesn't maintain information about past requests

How do websites remember you between visits?

Solutions to Statelessness

- **Cookies**: Small data files stored in client browser
- **Sessions**: Server-side storage with client identifier
- **LocalStorage/SessionStorage**: Client-side browser storage

HTTPS (HTTP Secure)

- Encrypted version of HTTP using SSL/TLS
- Provides:
  - Authentication (website is who it claims to be)
  - Data integrity (content hasn't been modified)
  - Confidentiality (communications are encrypted)

Architecture of the Web

- Client-server model
- Three-tier architecture
- Modern components

Client-server model

Client: User's device running a web browser

Server: Computer that hosts websites and services

Request-response cycle

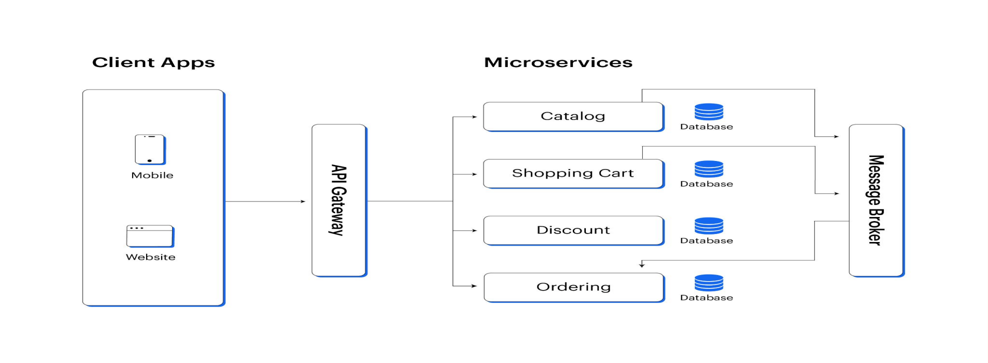
Three-Tier Architecture

- **Presentation tier**: User interface (browser)
- **Application tier**: Business logic processing
- **Data tier**: Database and data storage

Modern Web Architecture Components

- **Load Balancers**: Distribute traffic across multiple servers
- **CDNs (Content Delivery Networks)**: Distribute content geographically closer to users
- **Caching Systems**: Store copies of data for faster retrieval
- **Microservices**: Breaking applications into smaller, specialized services

MicroService Architecture



Web Servers: Powering the Internet

A **web server** is a combination of **hardware and software** that stores, processes, and delivers web pages to users over the **Internet** using the **HTTP or HTTPS protocol**.

Popular Web servers

Apache HTTP Server : Open-source, cross-platform

NGINX : Known for high performance and low resource usage

Microsoft IIS (Internet Information Services) : Windows-based web server