

# Chapter 0: Introduction

## 1 Objectives of This Class

### Objectives

- Understanding static and dynamic web development.
- Implementing interactive web features.
- Exploring the client-server architecture.
- Learning key web technologies: HTML, CSS, JavaScript, PHP, and MySQL.

## 2 Clearing Misconception: Internet $\neq$ Web

### Internet $\neq$ Web

Many people mistakenly believe that the Internet and the Web are the same, but they are **two distinct technologies**. The Internet, developed in the **1960s**, is a global network that connects computers, while the Web, introduced in **1989**, is a service that operates on top of the Internet. We will explore this distinction in more detail in the upcoming section.

### 2.1 The Internet

#### Internet

As previously discussed, the Internet is older than the Web. It was developed in the 1960s by 3 researchers : Leonard Kleinrock, Vinton Cerf, and Robert Kahn, with the goal of **making communication between distant computers possible**.

Their vision was to create a global network that would hold an online library for sharing and accessing and even updating vast amounts of data, a concept that later evolved into what we see today with platforms like Wikipedia.

But to reach the Internet as we know it today, many technologies had to be developed to solve various challenges. We will list them below.

### 2.1.1 Modem

## Modem

A modem is a hardware device that uses telephone lines to enable communication between distant machines. It operates using two key functions:

- **Modulator:** Converts an analog signal into a binary signal.
- **Demodulator:** Converts a binary signal back into an analog signal.

The name "modem" comes from the combination of these two functions: **MO**dulator and **DE**Modulator.

## Note

A legitimate question arises: why did modems rely on telephone lines? Why not create a technology to transmit binary signals directly without conversion?

The answer is that cities were already connected by telephone cables. Building an entirely new infrastructure for digital communication would have been extremely costly and time-consuming. Instead, leveraging the existing telephone network was the most practical and efficient solution.

### 2.1.2 Automatic Switching (Switches)

## Switches

Linking each machine or location directly to all others is impractical. We need a technology that can automatically determine the most efficient route (shortest path) for communication. This is why switches were invented.

Additionally, switches provide **resilience** (Fault-tolerant), ensuring that if one path fails, an alternative route can be used to maintain communication.

### 2.1.3 IP Addresses

## IP

As the number of machines grew exponentially, using simple labels to identify them became impractical. A new system was needed to uniquely identify each machine while remaining scalable for the future. This led to the creation of **IP addresses**.

The format:

`XXX.XXX.XXX.XXX : Port` where each  $XXX \in [0, 255]$

## Note

- We will discuss **ports** further in the course.
- Some countries assign **dynamic IP addresses** to routers, but internet providers can still track history and identify the original source.
- Over the years, the internet has become a global network, with undersea cables linking countries.

## 2.2 Web

### Web

The web emerged much later than the internet. It is a **service** that leverages the internet, similar to how software (the web) operates on hardware (the internet).

It was created in **1989** by **Tim Berners-Lee** to help physicists at **CERN** organize and access their research easily, regardless of their machine's operating system. This led to the birth of the first web page, built using **HTML** and hyperlinks, giving rise to the **World Wide Web (WWW)**.

## Note

- Tim Berners-Lee created the foundations of the web, including the first web server, **HTML**, and **HTTP**.
- He has also expressed disappointment that the web did not evolve as he originally envisioned, as many people now associate it solely with social media, moving away from its original purpose of sharing and accessing information.

### 2.2.1 DNS

### DNS

To host a website, we use servers that are accessible on the internet through a public IP address. The issue is that while IP addresses work well for machines, most humans cannot easily remember them.

This is why we needed a way to label websites with **domain names**, managed by the **DNS (Domain Name System)**. DNS stores a table that maps domain names to their corresponding IP addresses. This allows users to browse the web using domain names, and then the browser requests the IP address from the **DNS**, which redirects the request to the correct server.

## Note

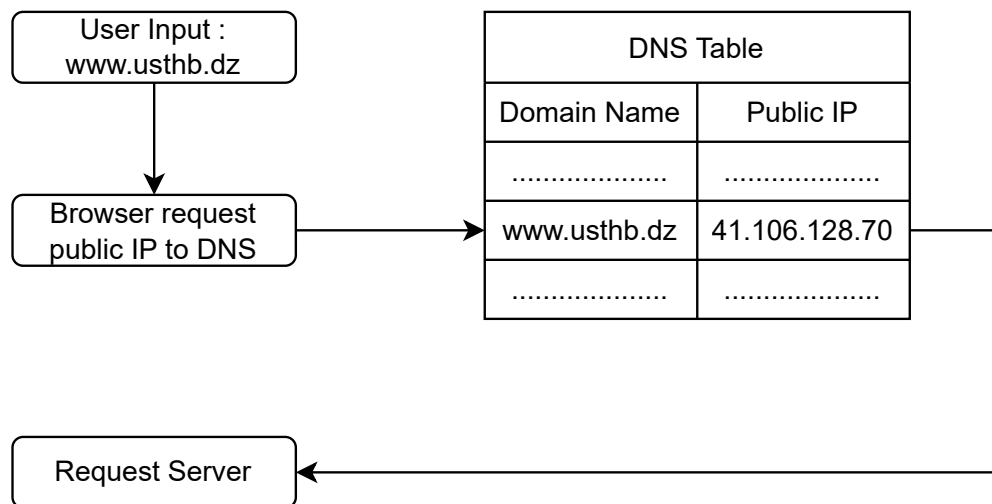
- Both the domain name and its corresponding IP address work when accessing a website in the browser.
- Some people try to exploit domain names by creating fake websites with similar names to steal user data. However, DNS security has improved, and modern browsers now flag uncertified and unsecure websites.
- It is possible to get the public ip of a domain using the **nslookup** command

## Nslookup Example

```
C:\Users\Administrator>nslookup www.usthb.dz
Server:   dev.opt
Address:  192.168.100.1

Non-authoritative answer:
Name:     portail2023.usthb.dz
Address:  41.106.128.70
Aliases:  www.usthb.dz
```

## DNS Diagram



### 2.2.2 Website & Web Pages

## Website & Web Pages

A website consists of multiple web pages that users can browse through hyperlinks, whether they are internal (linking within the same website) or external (linking to a different website). Each web page has its own **URL (Uniform Resource Locator)**, and each website has a domain name.

### 2.2.3 Technologies Involved in Creating a Website

## Technologies

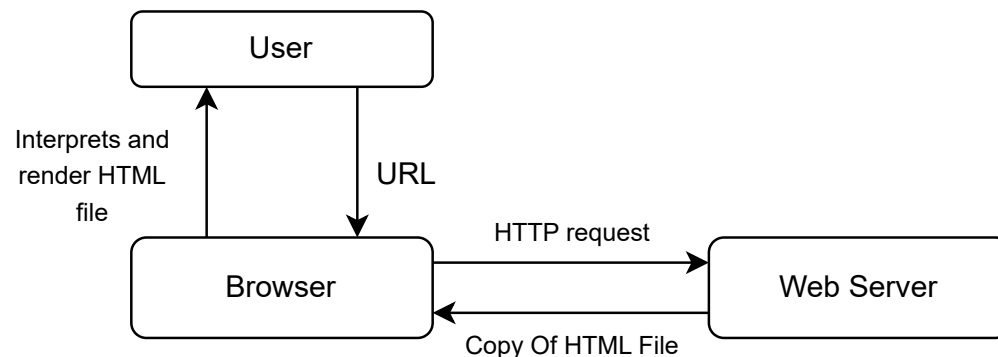
- **HTML (HyperText Markup Language):** The backbone and structure of a website.
- **CSS (Cascading Style Sheets):** Used to style fonts, colors, and layouts.
- **JS (JavaScript):** Makes the website interactive.
- **Server-Side Languages:** Various languages can be used, such as PHP, Java, Python, etc.
- **DBMS (Database Management System):** Ensures data persistence, e.g., MySQL, Oracle, etc.

### 2.2.4 Abstraction to the User

## Abstraction

There is an abstraction for the user. One might wonder what really happens after entering a **URL**.

When a URL is entered, an **HTTP (HyperText Transfer Protocol)** request is sent between the web browser and the web server. The server responds by sending a copy of the HTML file, which the browser then interprets and renders. Users can inspect and even modify the HTML in their browser. However, since it is just a local copy, any changes made will not affect the server. If the page is refreshed, it will revert to its original state.



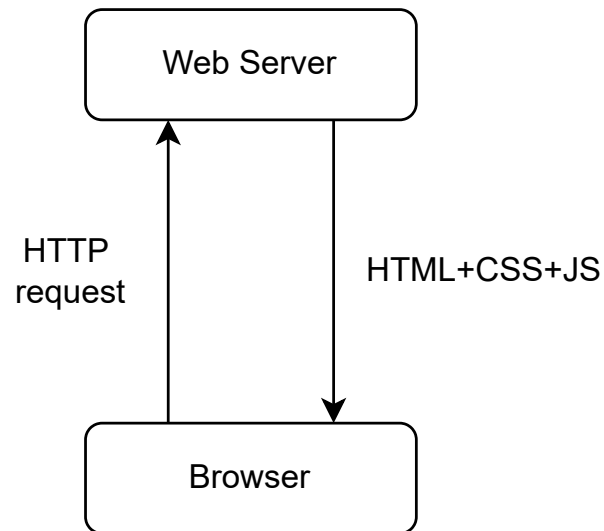
### 2.2.5 Static & Dynamic Websites

## Static & Dynamic

The terms **static** and **dynamic** refer to how a website's content is updated and displayed:

- **Static Website:**
  - The coder must manually update the content in the code.
  - No database is used.
  - The website's content remains the same unless manually updated.
- **Dynamic Website:**
  - Content changes dynamically based on user requests.
  - Uses a database to store and retrieve data.
  - Requires a server-side language (e.g., PHP, Java, Python).

### Static Website



## Dynamic Website

