

**Ministry of Higher Education and Scientific Research**  
**University of Abdelhafid Boussouf - Mila**  
**Institute of Mathematics and Computer Science**  
**Department of Computer Science**  
**Master 1 STIC - 2025/2026**

# **Advanced Web Development**

Chapter 1 : Review of Web Technologies

Presented by: **Dr. Brahim Benabderrahmane**

# Chapter 1 Overview

- 01 Introduction:** The Internet, the Web, and HTTP.
- 02 Client-Side Development:** HTML, CSS, JS, and Ajax.
- 03 Server-Side Development:** PHP, JSP/Servlets, and XML.

# The Global Infrastructure

**The Internet:** A "Network of Networks" using TCP/IP protocols to connect hardware globally.

**The World Wide Web (WWW):** An information space where documents and resources are identified by URLs and interlinked by hypertext.

## Historical Evolution:

- **Web 1.0:** Static pages, "Read-only".
- **Web 2.0:** Dynamic content, User-generated, "Read-Write".
- **Modern Web (2026):** Intelligent, decentralized, and real-time.

# The Communication Engine: HTTP Protocol

## Definition:

HyperText Transfer Protocol is the foundation of data exchange on the Web.

## Characteristics:

**Stateless:** The server does not retain user data between requests (handled by Sessions/Cookies).

**Client-Server Model:** The browser requests, the server responds.

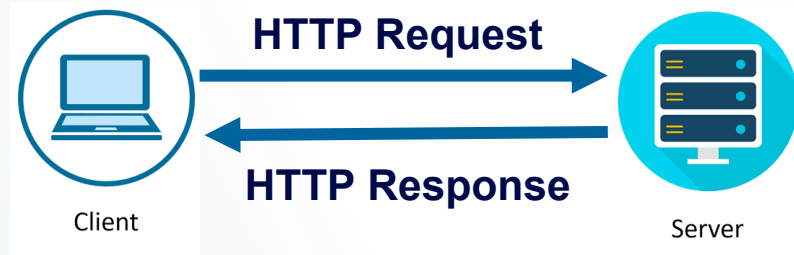
## Evolution:

Mentioning the shift from HTTP/1.1 (standard) to HTTP/2 and HTTP/3 (QUIC) for low-latency performance.

# Anatomy of an HTTP Request & Response

## Request Components:

- **Method/Verb:** GET (Read), POST (Create), PUT (Update), DELETE.
- **Headers:** Metadata (Content-Type, User-Agent, Authorization).



## Response Components:

- **Status Codes:** 200 (Success), 404 (Not Found), 500 (Server Error).
- **Payload:** The actual data (HTML, JSON, or XML).

# Client-Side Development (The Frontend)

## Role:

Designing the interface and interaction architecture.

## Core Technologies:



**HTML:** Structure and organization of pages.



**CSS:** Visual style and layouts (Syllabus reference: Style Sheets).



**JavaScript:** Interaction, navigation, and logic.



**Ajax:** Asynchronous updates without reloading the page.

# Server-Side Development (The Backend)

**Role:** Data processing, business logic, and security.

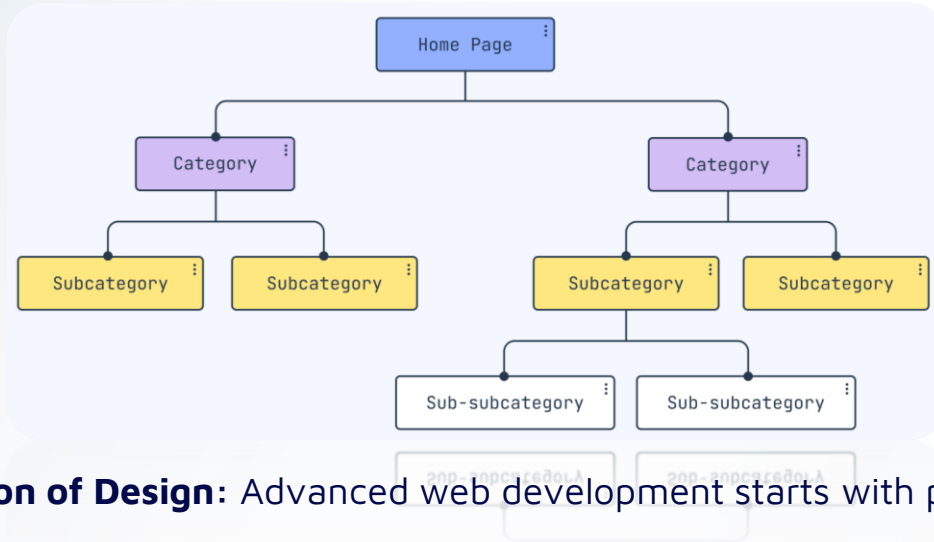
## Languages & Tools:

- **PHP:** Widely used scripting language for dynamic sites.
- **JSP & Servlets:** Java-based server-side technology for enterprise apps.

## Data Representation:

- **XML:** Extensible Markup Language for structured data.
- **Transition:** We will focus on JSON as the modern industry standard for APIs.

# Interaction Architecture & Arborescence



**The Foundation of Design:** Advanced web development starts with planning before coding.

## **Arborescence (Site Mapping):**

- The hierarchical "tree" structure of the application.
- Defining parent-child relationships between pages to ensure logical data flow.

## **Interaction Architecture:**

- The design of the "user journey", how a user moves from a request to a result.
- Organizing pages and navigation components to reduce cognitive load.

**Modern Context:** Moving from simple links to "**Single Page Application**" (SPA) routing where the architecture is **managed** by the **Frontend**.

# Professional Goals: Frameworks & CMS

## The Problem:

Manual "Vanilla" development is slow, error-prone, and difficult to scale.

## Content Management Systems (CMS):

- **Purpose:** Tools like Joomla or modern Headless CMS facilitate rapid deployment.
- **Core Benefit:** Decoupling content administration from the underlying code.

## Web Frameworks:

Pre-built structures that provide standardized solutions for common tasks (Routing, Auth, DB) e.g., WordPress, Drupal, or Joomla.

**Goal:** Drastically reduce development time while increasing security and performance

**Payload**

 **WORDPRESS**

  
**Drupal™**

 **strapi**

# Architecture Deep-Dive: The MVC Pattern

## **MVC (Model-View-Controller):**

The industry-standard architecture for separating concerns.

## **The Three Pillars:**

**Model:** Direct management of data, logic, and database interactions.

**View:** The presentation layer, everything the user sees (HTML/CSS).

**Controller:** The "Brain" that accepts input (forms/clicks) and converts it into commands for the Model or View.

**Implementation:** In the coming weeks, we will see how MVC handles complex Form Management and Database queries.

# Conclusion & Lab Kickoff

## Chapter 1 Summary:

- We have established the transition from basic HTTP protocols to complex application architectures.

## Looking Ahead:

- Next week begins our **Frontend Sprint (Chapter 2)**.

## Lab Requirements:

- Installation and configuration of the development environment (VS Code, Node.js, Git).
- Refresh your knowledge of basic HTML5 and CSS3.



**Good Luck**