



TEAM 2

WEBSITE DESIGN & DEVELOPMENT

CLASS: SE06303

MEMBER: DIEP VAN HUY

NGUYEN TIEN DUNG

NGUYEN CHI KIEN

NGUYEN HUU HIEU



I. IDENTIFY THE PURPOSE AND TYPES OF DNS, INCLUDING EXPLANATIONS ON HOW domain names are organised and managed.

01

What is a website? Basic components of a website

A website is a collection of web pages linked together, stored on a server and accessible via the internet. A website provides information, services or applications to users via a web browser.



Basic components of a website include:

- Domain name: This is the website address that users enter into the browser to access. For example: www.example.com.
- Hosting: Is a service that stores data and source code of a website on a server, helping the website to operate 24/7 and be accessed from anywhere.
- Source code: Are files containing the content, design and functions of the website, written in programming languages such as HTML, CSS, JavaScript, PHP, ...





Static and dynamic web:



Static web: Is a type of website with content that does not change or changes little. When a page is downloaded, the content is always the same and there is no dynamic interaction from the user. Usually only HTML, CSS is used for design.

Dynamic web: Is a website with content that changes based on user interaction or changes over time. Dynamic websites use databases and programming languages such as PHP, ASP.NET, ... to generate content automatically.



What is a URL? Distinguishing URL and domain name:



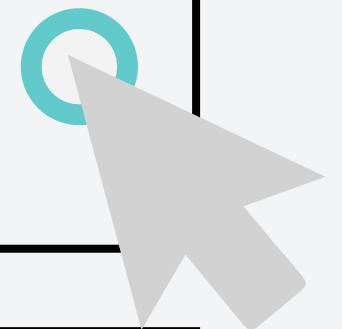
URL (Uniform Resource Locator): Is the complete address of a resource on the web, including the protocol (http/https), domain name and specific path to the resource. For example: <https://www.example.com/page1.html>.

Domain name: Just the domain name part of the URL, used to identify the website. For example: www.example.com.

Difference: The domain name is part of the URL. The URL points to a specific resource on the website, while the domain name identifies the entire website.

The user enters the URL into the browser.

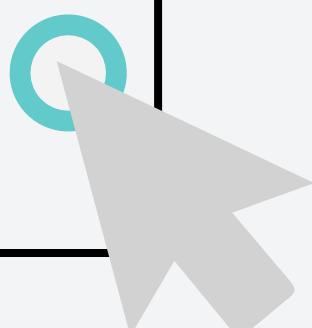
The browser sends a request to the website server via the HTTP or HTTPS protocol.



The server processes the request and sends back a response including HTML source code, CSS, JavaScript, etc. to the browser.



The browser receives the data and displays the website content on the user's screen.



What is a domain name? Domain name structure? How are domain names organized and managed

Domain name: An easy-to-remember address of a website instead of an IP address (a series of numbers), helping users easily access the website without having to remember the IP address of the server. For example: google.com, facebook.com.

Structure of a domain name:

Top-Level Domain (TLD): Is the extension of the domain name, located after the dot. For example: .com, .net, .org.

Second-Level Domain (SLD): Is the main domain name you choose. For example: in google.com, "google" is the SLD.

Subdomain: Is the prefix added before the SLD, separated by dots. For example: in "mail.google.com", "mail" is the subdomain.

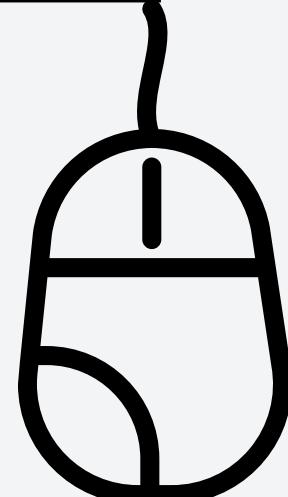




How domain names are organized and managed:

Domain names are managed by organizations such as ICANN (Internet Corporation for Assigned Names and Numbers) through domain name registrars. TLD domain names such as .com, .net, ... are assigned and managed by international organizations.

Users register domain names through registrars to own a domain name for their website for a period of time (usually from 1 to 10 years).



Domain Name Resolution System (DNS): Is a system that converts domain names into IP addresses, helping browsers and servers understand and connect with each other.

When users enter a domain name into the browser, DNS will perform the process of searching for the IP address corresponding to that domain name.

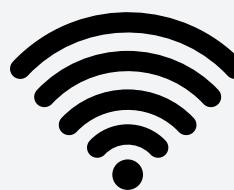
Structure of a domain name resolution system:

DNS Resolver: Is an intermediate server that receives requests from the browser and queries other DNS servers to find the IP address of the domain name.

Root DNS Server: Is the root server in the DNS system, holding information about TLDs (.com, .org, ...).

TLD DNS Server: Stores information about domain names belonging to each specific TLD.

Authoritative DNS Server: Is a server that holds accurate information about the IP address of a specific domain name.



How does the Domain Name System work?



The browser sends a request to the DNS Resolver to find the IP address corresponding to the domain name.

The DNS Resolver checks its cache. If it does not find one, it queries the Root DNS Server.

The Root DNS Server returns the address of the corresponding TLD DNS Server (for example, .com).

The DNS Resolver then queries the TLD DNS Server to find the address of the Authoritative DNS Server.

Finally, the Authoritative DNS Server provides the IP address of the domain name to the DNS Resolver, and the browser then uses the IP address to connect to the server and load the web page content.



II. EXPLAIN THE PURPOSE AND RELATIONSHIPS BETWEEN COMMUNICATION PROTOCOLS, SERVER HARDWARE, OPERATING SYSTEMS AND WEB SERVER SOFTWARE WITH REGARDS TO DESIGNING, PUBLISHING AND ACCESSING A WEBSITE.

01

What is a communication protocol?

- A communication protocol is a set of standard rules used to represent data, signal, authenticate, and detect data errors. These protocols allow computers and devices to connect and exchange information with each other efficiently and accurately.
- Communication protocols define how data is packaged, sent, received, and decoded to ensure that information is transmitted correctly.

Communication protocols used in the web

- HTTP (Hypertext Transfer Protocol): Hypertext Transfer Protocol, not secure.
- HTTPS (HTTP Secure): Secure version of HTTP, encrypts data transmitted.
- FTP (File Transfer Protocol): File transfer protocol, often used to transfer files between servers and clients.

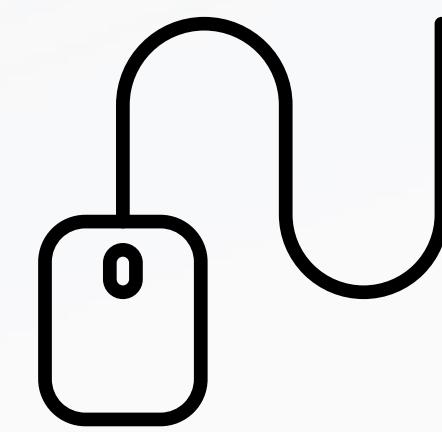




Comparison table of communication protocols:



Criteria	HTTP	HTTPS	FTP
Protocol type	Hypertext transfer protocol	Secure hypertext transfer protocol	File transfer protocol
Default port	80	443	20 (Data), 21 (Control)
Security	No data encryption	Data encryption using SSL/TLS	No encryption
Application	Web information transmission	Secure web information transmission	File transfer between server and client
Performance	High, no encryption required	Lower than HTTP due to encryption	High performance for large file transfers
Common use	Websites that do not require security	Secure websites (transactions, personal information)	Managing and transferring files between systems





Definition:

Server hardware is a set of physical devices that make up a server, allowing data processing and storage, providing services to clients in the network. The basic components include:

- CPU (Central Processing Unit): Processes commands and calculations.
- RAM (Random Access Memory): Temporary memory helps process data quickly.
- Hard Drive (Hard Drive/SSD): Stores data and applications.
- Network Interface Card: Allows connection and communication with other devices in the network.
- Power Supply: Provides power to all hardware components.

Role:

The server acts as a coordination center, processing requests from clients and providing data or services such as web, email, database.

Ensures continuity, stability, and security for large network systems, serving many users at the same time.



Server Operating System



An Operating System (OS) is a software that manages and coordinates computer hardware, providing a platform for applications and services to operate. It controls the execution of programs and handles resources such as CPU, RAM, hard drive.

Server operating systems (Server OS) are specifically designed to manage and support server services, ensuring stability, security and performance for multi-user applications.

Popular operating systems for web servers:

- Linux (Ubuntu Server, CentOS, Red Hat Enterprise Linux): Open source, good security, and often used for Apache, Nginx web servers.
- Windows Server: Microsoft's commercial operating system, good support for applications running on the .NET platform and IIS (Internet Information Services) server.
- macOS Server: A few web servers use macOS Server, mainly for Apple applications. These operating systems provide a stable platform for deploying web services, databases, and many other applications



Server software is installed and configured according to the function of each type of server to serve specific services such as web, email, database, or file transfer. Below are the common types of servers and their corresponding software:

1. Web server:

Web server is installed with specialized software to manage and operate websites.

Popular software:

XAMPP: Software package including Apache (web server), MySQL (database), PHP, and Perl, which helps deploy web applications.

IIS (Internet Information Services): Microsoft's web server, used for web applications on the Windows platform.

Visual Studio: Integrated development environment (IDE) that supports the development and running of web applications.

2. Email Server:

This server supports sending, receiving, and storing emails for users.

Popular software:

Microsoft Exchange Server: Microsoft's dedicated email management software.

Postfix: Open source email sending server for Linux systems.





3. Database Server:



The database server stores, manages, and processes application data.

Popular software:

MySQL: Open source database management system, often used in web applications.

Microsoft SQL Server: Microsoft database management software, widely used in enterprise applications.

4. FTP Server:



Supports file transfer via FTP protocol between the server and client devices.

Popular software:

FileZilla: Open source FTP software used to transfer files via FTP.

CuteFTP: A simple FTP application that allows managing and transferring files over the network.

The Role of Server Software



Ensures Service Uptime: These software help servers perform specific tasks, such as managing web, email, data, or files.

System Resource Management: Server software coordinates and manages the use of CPU, RAM, and network bandwidth to optimize system performance and security.



1. Hardware:

Hardware (server) provides the environment for installing the operating system and software. From there, it supports the process of designing, operating and publishing the website. For example, the server stores web data, processes requests from users and ensures the website operates stably.

2. Operating system:

The operating system acts as an intermediary between the hardware and the user, providing a platform for developers to write, run web applications, and control access to the hardware. The operating system helps manage resources and supports the services needed for the website to operate.

3. Communication protocol:

Communication protocols such as HTTP, HTTPS are responsible for transmitting data from the web server (web server) to the user's browser and vice versa. It allows users to access and interact with the website. This protocol can be implemented by hardware, software or a combination of both.

4. Web Software:

Web software is installed on a web server to support the design, publishing, and management of websites. Tools such as XAMPP, IIS, and FTP help develop websites, publish content, and provide services necessary for users to access and use the website.



III: DISCUSS THE CAPABILITIES AND RELATIONSHIPS BETWEEN FRONT-END AND BACK-END WEBSITE TECHNOLOGIES AND EXPLAIN HOW THESE RELATE TO PRESENTATION AND APPLICATION LAYERS

Front-end and Back-end Technologies

HTML (Front-End Technology)

Capabilities:

- HTML (HyperText Markup Language) is the core language for structuring content on the web. It defines elements like headings, paragraphs, lists, images, and links using specific tags (e.g., `<h1>`, `<p>`, ``). HTML serves as the essential building block that browsers interpret to render web pages.

Key Features:

- Semantic Structure: HTML tags convey specific meanings (e.g., `<header>`, `<footer>`, `<nav>`), aiding browsers and search engines in understanding content.
- Multimedia Support: Allows embedding of images, audio, video, and interactive elements like forms.
- Extensibility with Attributes: Elements can have attributes (e.g., `class`, `id`, `href`) that provide additional information or link to styling (CSS) and behavior (JavaScript).

Example:

```
<html>
  <head>
    <title>My Website</title>
  </head>
  <body>
    <h1>Welcome to My Website</h1>
    <p>This is a simple HTML page with some text and an image.</p>
    
  </body>
</html>
```

CSS (Front-End Technology)

Capabilities:

- CSS (Cascading Style Sheets) is used to control the visual appearance and layout of HTML elements on a web page. While HTML structures content, CSS handles presentation aspects like colors, fonts, layout, and responsive design. It allows for visually appealing websites that adapt to various screen sizes, separating content from design.

Key Features:

- Styling: Controls text colors, background images, and layouts using grid and flexbox.
- Responsive Design: Media queries enable websites to adapt to different screen sizes for optimal viewing on mobile and desktop.
- Animation: Adds transitions and animations to elements, enhancing user experience with visual feedback and interactivity.

Example:

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f0f0f0;  
    color: #333;  
}  
  
h1 {  
    color: #4CAF50;  
}  
  
p {  
    font-size: 16px;  
    line-height: 1.5;  
}  
  
img {  
    max-width: 100%;  
    height: auto;  
}
```

JavaScript (Front-End Technology)

Capabilities:

- JavaScript is a programming language that adds interactivity and dynamic behavior to web pages, enabling client-side scripts to run in the browser. It enhances user experience through real-time interactions, such as form validation and animations.
- Essential for modern web development, it transforms static HTML and CSS into interactive interfaces, powering complex single-page applications (SPAs) with frameworks like React, Angular, and Vue.js.

Key Features:

- DOM Manipulation: Dynamically modifies the Document Object Model (DOM) for real-time content changes without page reloads.
- Asynchronous Processing: Uses technologies like AJAX to retrieve data from servers in the background, allowing updates without refreshing.
- Event Handling: Listens to user actions (e.g., clicks, key presses) and enables interactive features like dropdown menus and form validation.

Example:

```
document.querySelector('button').addEventListener('click', function() {
  alert('Button clicked!');
});

// Dynamic content update without reloading the page
fetch('https://api.example.com/data')
  .then(response => response.json())
  .then(data => {
    document.querySelector('#content').innerHTML = data.message;
});
```

PHP (Back-End Technology)

Capabilities:

- PHP (Hypertext Preprocessor) is an open-source server-side scripting language widely used for dynamic content generation, form processing, session management, and database interactions. It's the backbone of many web applications, including platforms like WordPress.
- As a back-end technology, PHP processes user requests and generates HTML content for browsers.

Key Features:

- Easy Integration with HTML: PHP can be directly embedded in HTML, simplifying dynamic page creation.
- Database Interaction: Works well with various databases, particularly MySQL, making it suitable for data-driven applications.
- Session Management: Effectively manages user sessions, maintaining state between page loads.

Example:

```
<?php  
    // Connect to a database  
    $conn = new mysqli("localhost", "username", "password", "database");  
  
    // Fetch user data from the database  
    $result = $conn->query("SELECT * FROM users WHERE id = 1");  
  
    // Display the user's name  
    $user = $result->fetch_assoc();  
    echo "Welcome, " . $user['name'];  
?>
```

ASP.NET (Back-End Technology)

Capabilities:

- ASP.NET is a server-side framework by Microsoft for building dynamic websites and web applications. It's robust, scalable, and integrates well with the .NET ecosystem, making it suitable for enterprise-level solutions.
- Supports multiple programming languages, including C#, and offers high performance and security, favored by large organizations.

Key Features:

- Integrated Framework: Works seamlessly with Microsoft technologies like Visual Studio, Azure, and SQL Server.
- High Performance: Compiled code leads to faster page loads and improved server-side performance.
- Scalability: Easily scales to accommodate increased user demand, ideal for large-scale applications.

Example:

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        ViewBag.Message = "Welcome to ASP.NET!";
        return View();
    }
}
```

Java (Back-End Technology)

Capabilities:

- Java is a versatile, object-oriented language ideal for back-end development, especially for large-scale, enterprise applications. Its cross-platform compatibility, security, and scalability make it suitable for robust systems.
- Commonly used with frameworks like Spring, Java enables the creation of secure and scalable web applications, supported by a strong ecosystem of libraries.

Key Features:

- Platform Independence: Java runs on any platform with the Java Virtual Machine (JVM), ensuring portability.
- Scalability: Designed to handle significant traffic and data, making it ideal for large applications.
- Strong Ecosystem: Rich in libraries and frameworks (e.g., Spring, Hibernate) that facilitate efficient web development.

Example:

```
@RestController
public class UserController {

    @GetMapping("/user/{id}")
    public User getUser(@PathVariable String id) {
        return userService.getUserById(id);
    }
}
```

Ruby (Back-End Technology)

Capabilities:

- Ruby is a dynamic, object-oriented programming language, especially powerful with the Ruby on Rails framework for building web applications quickly and simply.
- Its flexibility makes it ideal for startups and small teams, emphasizing convention-over-configuration to minimize repetitive coding.

Key Features:

- Developer-Friendly: Focuses on simplicity and productivity, enabling less code for more functionality.
- Rails Framework: Facilitates rapid development with built-in solutions for routing, databases, and templating.
- Community Support: Strong community with numerous pre-built libraries (gems) that enhance development speed.

Example:

```
class UsersController < ApplicationController
  def show
    @user = User.find(params[:id])
  end
end
```

Relationship Between Front-End and Back-End Technologies with Application Layer and Presentation Layer

Front-end and back-end technologies in web development can be compared to the presentation layer and application layer in the OSI (Open Systems Interconnection) model as follows:

Front-End and Presentation Layer:

- The front-end includes HTML, CSS, and JavaScript, responsible for user interface and experience. It manages user interactions, such as form submissions and link clicks.
- The presentation layer displays data received from the back-end in a visually appealing format. For example, when a user submits a form, the front-end sends the data to the back-end for processing.

Back-End and Application Layer:

- The back-end includes technologies like PHP, ASP.NET, Java, and Ruby, handling business logic, data storage, and user requests. It processes data and returns information to the front-end.
- The application layer encompasses tasks performed by the back-end to ensure accurate data processing and delivery to the front-end.

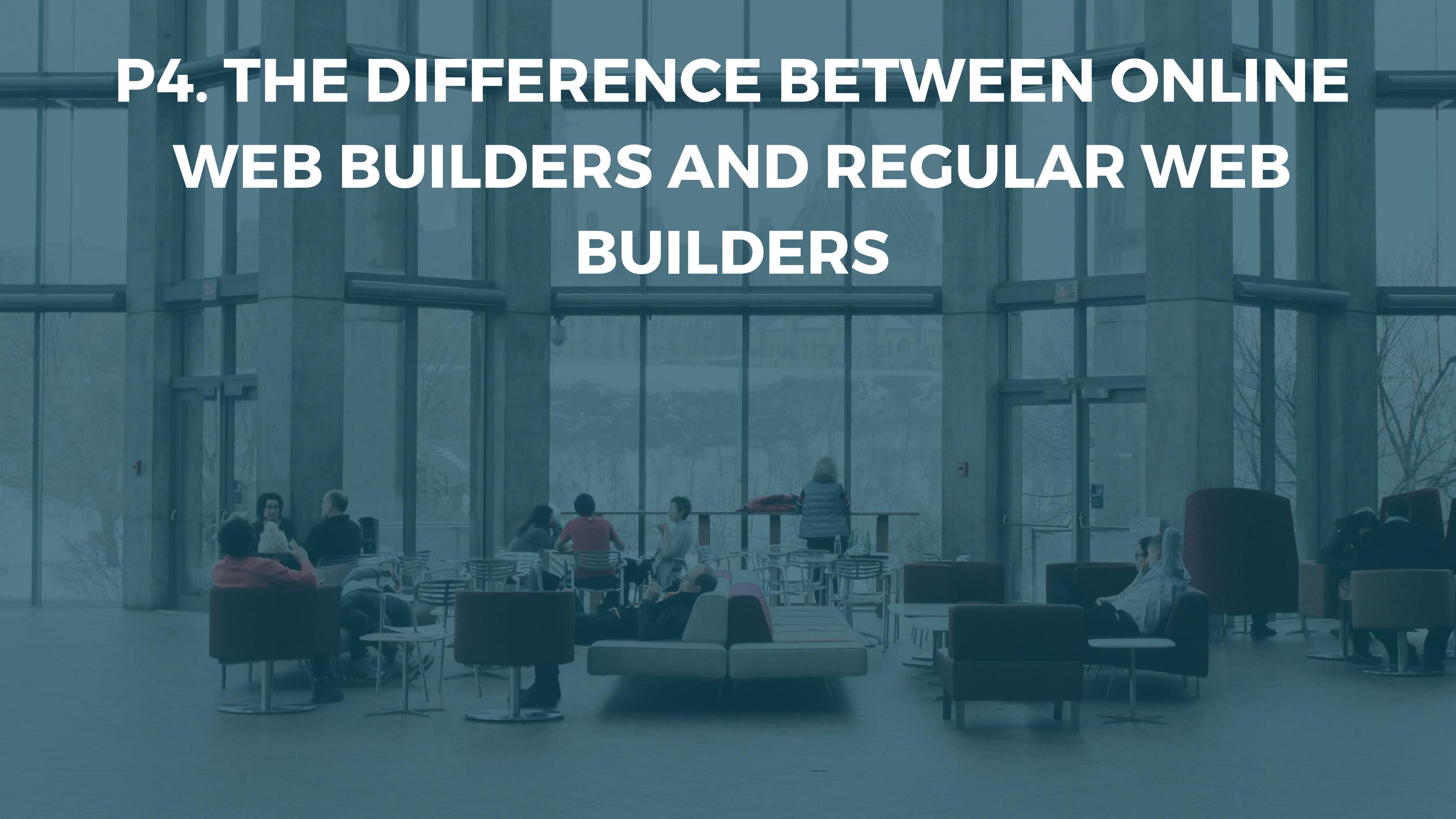
Flow of Data:

- When a user submits a form, the front-end sends the data to the back-end, which processes it and returns results for display.
- In the OSI model, data flows from the user through the application layer to the presentation layer, where it is formatted for viewing, and vice versa.

Summary:

- The front-end and back-end are closely linked, much like the presentation and application layers in the OSI model. The front-end interacts with users, while the back-end manages data processing and business logic, ensuring effective communication within the web application.

P4. THE DIFFERENCE BETWEEN ONLINE WEB BUILDERS AND REGULAR WEB BUILDERS

A large, modern building with a glass facade and many windows. In the foreground, there is a lounge area with several people sitting on couches and chairs, looking out at the view. The building has a minimalist design with a grid of windows and a dark, textured facade.

THE DIFFERENCE

Criteria	Online web creation tool	General web creation tools
Flexibility in design	Limited customization, depends on available templates	Customize freely with HTML/CSS source code
Efficiency	Automatic optimization, but sometimes slow due to platform server dependency	High performance if well optimized
Function	Supports various plugins and simple drag and drop features	Add functions yourself, need programming skills
User Experience (UX)	Simple interface, easy to use for non-professionals	Depends on the UX design skills of the programmer
User Interface (UI)	Pre-designed interface, easy to edit via intuitive tools	Freelance design, need UI-savvy programmer

ADVANTAGES AND DISADVANTAGES OF SOME ONLINE WEB CREATION TOOLS

Tools	Advantage	Disadvantages
WordPress	<ul style="list-style-type: none">- Easy to use, popular- Supports many powerful plugins	<ul style="list-style-type: none">- Limited deep customization- Sometimes requires payment
Wix.com	<ul style="list-style-type: none">- Easy drag and drop- Integrated with many beautiful templates	<ul style="list-style-type: none">- Poor performance on complex pages
Joomla	<ul style="list-style-type: none">- Flexible with many different types of websites	<ul style="list-style-type: none">- Programming learning curve is a bit high for beginners
SiteSpinner	<ul style="list-style-type: none">- Create website quickly- Newbie friendly	<ul style="list-style-type: none">- Limited customization options
CoffeeCup Editor	<ul style="list-style-type: none">- Free- Simple and fast HTML support	<ul style="list-style-type: none">- Only suitable for people with knowledge of HTML



Wix is a popular online website building platform that allows users to create professional websites without any programming skills. Wix offers easy-to-use drag-and-drop templates, with built-in features such as blogging, e-commerce, and SEO. Users can customize the interface, add multimedia content, and connect with third-party apps to extend functionality. Wix has free and paid plans, suitable for individuals and businesses who want to build a website quickly and efficiently.

COMPARE ONLINE WEB BUILDERS AND REGULAR WEB BUILDERS

Comparison factor	Online web creation tool	General web creation tools
Easy to use	Very easy, no programming required	Requires programming skills and technical knowledge
Flexibility	Limited customization to individual needs	Complete freedom to customize every detail
SEO Optimization	Provide basic SEO tools	Optimize as desired by tweaking the code
Scalability	Depends on available plugins	Any feature can be added programmatically
Expense	Free with limitations, paid for advanced features	Free if self-deployed, hosting fee applies

**THANK'S FOR
WATCHING**

