What is the Internet?

The Internet is a vast **interconnection of networks** that links computer systems across the globe. Every device connected to this network is assigned a unique **IP address**, which acts like a digital fingerprint—ensuring each computer can be identified and located within the network.

Network Connection

Multiple systems interconnected through protocols and infrastructure

IP Addresses

Unique identifiers that distinguish each computer on the network

Data Exchange

Information flows between systems seamlessly across the globe

Through this infrastructure, data exchange happens continuously between systems—powering everything from simple emails to complex web applications. The Internet forms the backbone of modern digital communication.

WEB DEVELOPMENT

Building the Interactive Web

Businesses recognized the Internet's potential and began creating **interactive webpages** to showcase their work, products, and services. This gave birth to a new development discipline: **web development**—where developers build pages that clients can interact with through their browsers.

How It Works

1. Client opens a browser and requests a website
2. Client connects to ISP (Internet Service Provider)
3. ISP queries DNS to translate the website name into an IP address
4. Client uses the IP address to connect directly to the server
5. Server responds with HTML, CSS, and JavaScript files
6. Browser renders the interactive webpage

DNS Server
ISP
ISP
Web Server
Web Server
192.168.x.x
1168.x.x
User Computer

Full Stack Development

Full stack development encompasses all layers of web application development—from what users see and interact with, to the server logic that powers it, to the data storage that makes it all persistent.

01

Frontend (FE)

HTML, CSS, and JavaScript create the user interface and handle client-side interactions that users see and interact with directly.

02

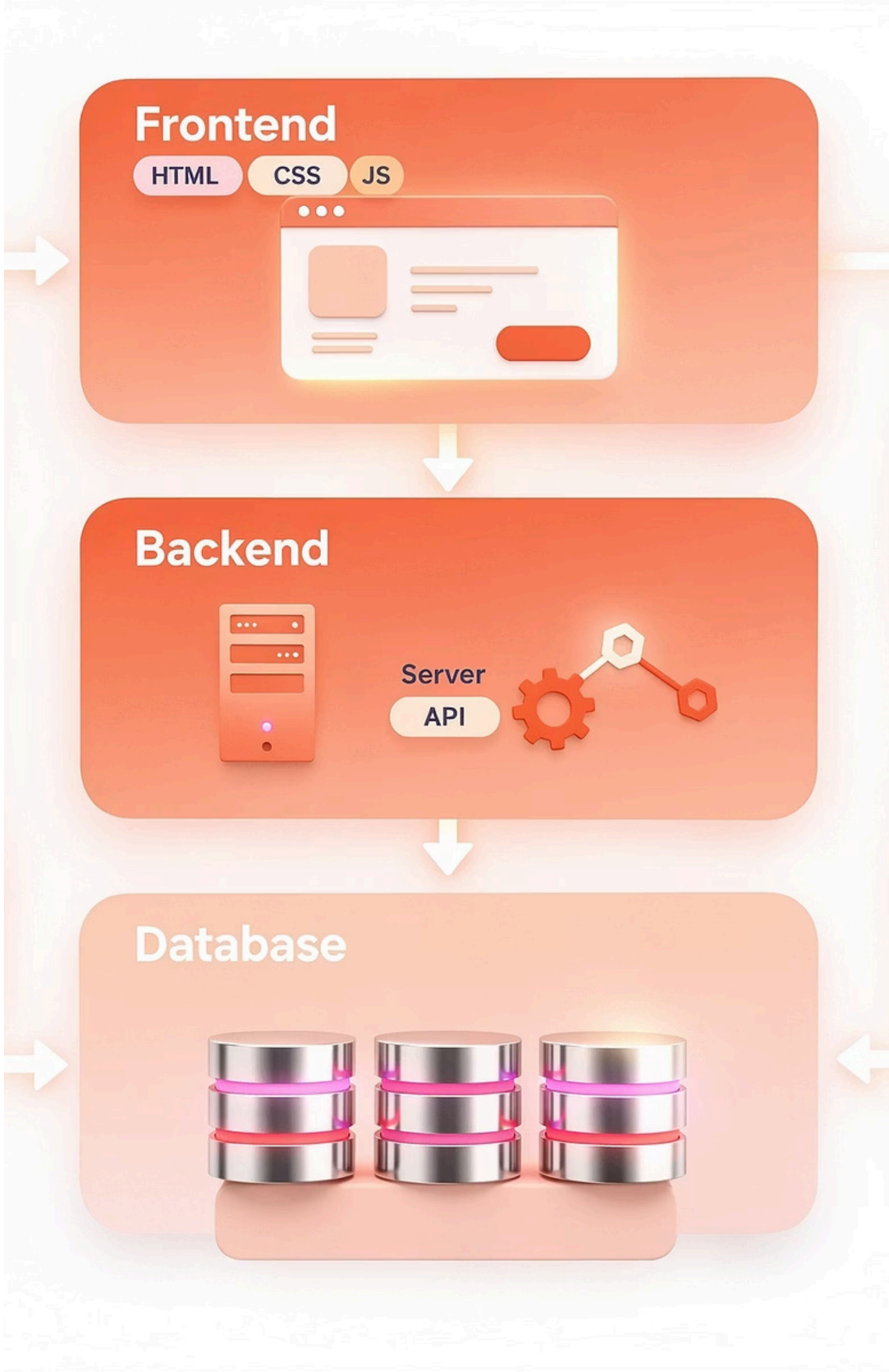
Backend (BE)

Server-side response management processes requests, implements business logic, and manages data flow between the frontend and database.

03

Database

Bulk data storage and management systems that efficiently store, retrieve, and organize large amounts of application data.



HTML

HyperText Markup Language

HTML helps us present plain text in a structured and formatted way. It provides the building blocks for creating webpages by defining elements like paragraphs, buttons, headings, and titles.

Plain Text vs. HTML Structure

See how simple plain text transforms into structured web content with HTML:

Plain Text

```
Welcome to My Website
This is a paragraph of text.
Click here
```

With HTML

```
<h1>Welcome to My
Website</h1>
<p>This is a paragraph of text.
</p>
<button>Click here</button>
```

<h1"Welcome to Your Creative Space!"
<p Design. Code. Inspire.p
<button. Geet Started.ne>
<form....formse>

HTML Element

An element is a complete structure made of tags and content. HTML provides standard elements that browsers recognize and render. To create different HTML elements, we use HTML tags.

1

Opening Tag

Marks the beginning of the element, for example, `<p>` or `<h1>`.

2

Content

The actual text or data inside the element, e.g., "Hello World!".

3

Closing Tag

Marks the end of the element, for example, `</p>` or `</h1>`.

Here are some examples of complete HTML elements:

```
<p>This is a paragraph.</p>
```

```
<h1>This is a heading.</h1>
```

```
<button>Click Me</button>
```

HTML Tags

A tag is a keyword written inside angle brackets ‹ ›. HTML tags typically come in pairs—an opening tag and a closing tag. The content goes between these tags.

<p> Paragraph Tag

Used to define a block of text content. Browser renders it as a text block with spacing above and below.

```
<p>This is a paragraph.</p>
```

<img> Image Tag

Used to embed images in a webpage. This is a self-closing tag (no closing tag needed). The `src` attribute specifies the image source, and `alt` provides alternative text.

```
<img src="photo.jpg" alt="description">
```