

# Introduction to Web Development and E-commerce

Emmanuel BRUNO

Université de Toulon, LIS UMR CNRS 7020

emmanuel.bruno@univ-tln.fr

2026-10-05

## Table des matières

|     |                                                        |   |      |                                                              |   |
|-----|--------------------------------------------------------|---|------|--------------------------------------------------------------|---|
| 1   | Web Development for E-commerce                         | 2 | 6.9  | 2020s - Modern Web                                           | 4 |
| 1.1 | Web Development                                        | 2 | 7    | The Internet vs. the World Wide Web                          | 4 |
| 1.2 | E-commerce Web Development                             | 2 | 8    | The Role of Front-End Development in E-commerce              | 5 |
| 2   | E-commerce Models and Technical Examples               | 2 | 8.1  | Customer Experience and Engagement                           | 5 |
| 2.1 | B2B (Business-to-Business)                             | 2 | 8.2  | Online Transactions and Payments                             | 5 |
| 2.2 | B2C (Business-to-Consumer)                             | 2 | 8.3  | Marketing and Branding                                       | 5 |
| 2.3 | C2C (Consumer-to-Consumer)                             | 2 | 8.4  | Data Collection and Analysis                                 | 5 |
| 3   | Technical Focus for E-commerce                         | 3 | 9    | The Role of Back-End Development                             | 5 |
| 3.1 | User Experience                                        | 3 | 10   | Client / Server Architecture                                 | 5 |
| 3.2 | Performance and Scalability                            | 3 | 10.1 | Web Browser                                                  | 5 |
| 3.3 | Security                                               | 3 | 10.2 | Web Server                                                   | 6 |
| 4   | Key Components                                         | 3 | 11   | Introduction to HTTP Protocol                                | 6 |
| 4.1 | Front-End Development                                  | 3 | 11.1 | What is HTTP?                                                | 6 |
| 4.2 | Back-End Development                                   | 3 | 11.2 | How HTTP Works                                               | 6 |
| 5   | Why an E-commerce Generalist Should Know Web Languages | 3 | 11.3 | Common HTTP Methods                                          | 6 |
| 5.1 | Improved Communication                                 | 3 | 11.4 | HTTP Status Codes                                            | 6 |
| 5.2 | Enhanced Problem-Solving                               | 3 | 11.5 | Why HTTP Matters                                             | 6 |
| 5.3 | Competitive Advantage                                  | 4 | 12   | Sequence Diagram                                             | 6 |
| 5.4 | Better User Experience                                 | 4 | 13   | URL (Uniform Resource Locator)                               | 6 |
| 6   | History of the World Wide Web (WWW)                    | 4 | 14   | The Building Blocks of Websites                              | 7 |
| 6.1 | 1989 - Proposal                                        | 4 | 14.1 | HTML (HyperText Markup Language)                             | 7 |
| 6.2 | 1990 - First Website                                   | 4 | 14.2 | CSS (Cascading Style Sheets)                                 | 7 |
| 6.3 | 1991 - Public Release                                  | 4 | 14.3 | JavaScript                                                   | 7 |
| 6.4 | 1993 - Mosaic Browser                                  | 4 | 14.4 | PHP or JavaScript in Backend                                 | 7 |
| 6.5 | 1994 - Netscape Navigator                              | 4 | 15   | Key Trends                                                   | 7 |
| 6.6 | 1995 - Commercialization                               | 4 | 15.1 | Single Page Applications (SPAs)                              | 7 |
| 6.7 | 2000s - Web 2.0                                        | 4 | 15.2 | Responsive Web Design (RWD)                                  | 7 |
| 6.8 | 2010s - Mobile Web                                     | 4 | 15.3 | Server-Side Rendering (SSR) and Static Site Generation (SSG) | 7 |

|      |                                                     |   |
|------|-----------------------------------------------------|---|
| 15.4 | Headless CMS .....                                  | 7 |
| 15.5 | Progressive Web Apps (PWAs) .....                   | 7 |
| 15.6 | WebAssembly (Wasm) .....                            | 7 |
| 16   | Introduction to Web Development .....               | 7 |
| 16.1 | What is the Web? .....                              | 7 |
| 16.2 | The Role of a Web Developer .....                   | 8 |
| 16.3 | Basic Components of a Website ....                  | 8 |
| 17   | Web Development Stack .....                         | 8 |
| 17.1 | Front-End .....                                     | 8 |
| 17.2 | Back-End .....                                      | 8 |
| 18   | Development Tools .....                             | 8 |
| 18.1 | Integrated Development<br>Environments (IDEs) ..... | 8 |
| 18.2 | Version Control Systems .....                       | 8 |
| 19   | Questions? .....                                    | 8 |
| 20   | Key Takeaways 1 .....                               | 8 |
| 20.1 | E-commerce Models .....                             | 8 |
| 20.2 | Web Development for E-<br>commerce .....            | 8 |
| 20.3 | Technical Focus .....                               | 9 |
| 21   | Key Takeaways 2 .....                               | 9 |
| 21.1 | Importance of Web Languages .....                   | 9 |
| 21.2 | History of the WWW .....                            | 9 |
| 21.3 | HTTP Protocol .....                                 | 9 |
| 21.4 | Key Trends .....                                    | 9 |
| 21.5 | Development Tools .....                             | 9 |

# 1 Web Development for E-commerce

## 1.1 Web Development

Web development is the process of creating and maintaining websites and web applications. It involves a combination of front-end (client-side) and back-end (server-side) development. Front-end development focuses on the visual and interactive aspects of a website, while back-end development handles the server-side logic and database interactions.

## 1.2 E-commerce Web Development

E-commerce web development focuses specifically on building online stores and platforms for buying and selling products and services. It integrates various functionalities such as product catalogs, shopping carts, payment gateways, and user accounts. These features are essential for creating a seamless and secure shopping experience for customers.

## 2 E-commerce Models and Technical Examples

### 2.1 B2B (Business-to-Business)

B2B refers to businesses selling products or services to other businesses. In this model, companies often use APIs for seamless integration with client systems and ensure secure transactions and data handling. For example, Alibaba uses robust search algorithms and data analytics to facilitate B2B transactions, while Amazon Business employs scalable cloud infrastructure (AWS) to support its operations.

### 2.2 B2C (Business-to-Consumer)

B2C refers to businesses selling products or services directly to consumers. This model emphasizes the need for user-friendly interfaces and responsive design, as well as secure payment gateways and personalized recommendations. For instance, Amazon uses machine learning for personalized recommendations to enhance the shopping experience, and eBay implements real-time bidding systems to facilitate consumer-to-consumer transactions.

### 2.3 C2C (Consumer-to-Consumer)

C2C platforms enable consumers to sell to each other. These platforms require user authentication and secure payment processing, as well as rating and review systems to build trust among users. For example, eBay uses auction

algorithms and secure payment systems to facilitate C2C transactions.

## 3 Technical Focus for E-commerce

### 3.1 User Experience

User experience ensures that e-commerce sites are user-friendly, visually appealing, and functional across all devices. A good user experience leads to higher customer satisfaction and increased sales. It involves designing intuitive navigation, fast-loading pages, and responsive layouts that adapt to different screen sizes.

### 3.2 Performance and Scalability

Optimizing site performance is crucial for handling high traffic and large volumes of transactions. Scalability ensures that the platform can grow as the business expands. Techniques such as caching, load balancing, and database optimization are used to enhance performance and scalability.

### 3.3 Security

Implementing security measures is essential to protect sensitive customer data and prevent fraud. Security practices include using HTTPS, encrypting data, and regularly updating software to patch vulnerabilities. Ensuring compliance with industry standards and regulations is also important for maintaining customer trust.

## 4 Key Components

### 4.1 Front-End Development

Front-end development involves creating the visual and interactive aspects of a website. Key technologies include:

- **HTML (HyperText Markup Language):** Structures the content of e-commerce websites.
- **CSS (Cascading Style Sheets):** Styles the visual presentation of web pages.
- **JavaScript:** Adds interactivity and dynamic features to enhance user experience.

### 4.2 Back-End Development

Back-end development handles the server-side logic, database interactions, and business processes. Key technologies include:

- **Server-Side Languages (e.g., PHP, Node.js):** Handle server logic and database interactions.
- **Databases (e.g., MySQL, MongoDB):** Store and manage product information, user data, and transaction records.
- **APIs (Application Programming Interfaces):** Enable integration with third-party services like payment gateways and shipping providers.

## 5 Why an E-commerce Generalist Should Know Web Languages

### 5.1 Improved Communication

Understanding web languages helps in better communication with developers and technical teams. It facilitates clearer requirements and more effective collaboration on projects, ensuring that everyone is on the same page.

### 5.2 Enhanced Problem-Solving

Having knowledge of web languages enables e-commerce generalists to identify and troubleshoot basic issues on e-commerce sites. This helps in making informed decisions about website improvements and optimizations, leading to a better overall user experience.

### **5.3 Competitive Advantage**

Knowledge of web languages provides a competitive edge in the job market. It enables e-commerce generalists to contribute more effectively to the development and maintenance of online stores, making them valuable assets to their organizations.

### **5.4 Better User Experience**

Ensuring that e-commerce sites are user-friendly, visually appealing, and functional across all devices leads to higher customer satisfaction and increased sales. Understanding web languages helps in achieving these goals by enabling better design and development practices.

## **6 History of the World Wide Web (WWW)**

### **6.1 1989 - Proposal**

Tim Berners-Lee proposed the concept of the World Wide Web at CERN in 1989. This marked the beginning of a revolutionary way to share and access information globally.

### **6.2 1990 - First Website**

The first website (info.cern.ch) was created by Tim Berners-Lee in 1990. This site provided information about the World Wide Web project and how to create web pages.

### **6.3 1991 - Public Release**

In 1991, the World Wide Web was made available to the public, allowing people to access and share information over the internet.

### **6.4 1993 - Mosaic Browser**

The Mosaic browser was released in 1993, making the web more accessible to the general public. It was the first browser to display images inline with text, enhancing the user experience.

### **6.5 1994 - Netscape Navigator**

Netscape Navigator, launched in 1994, quickly became the most popular web browser. It played a significant role in popularizing the web and driving its rapid growth.

### **6.6 1995 - Commercialization**

The mid-1990s saw the rapid growth of commercial websites, with companies like Amazon and eBay launching their online platforms. This period marked the beginning of e-commerce as we know it today.

### **6.7 2000s - Web 2.0**

The 2000s brought the emergence of Web 2.0, characterized by user-generated content and social media. Platforms like Facebook, YouTube, and Wikipedia allowed users to create and share content, transforming the web into a more interactive and collaborative space.

### **6.8 2010s - Mobile Web**

The rise of smartphones in the 2010s led to the development of responsive web design, ensuring that websites function well on all devices. This period also saw the growth of mobile apps and the increasing importance of mobile-friendly websites.

### **6.9 2020s - Modern Web**

The 2020s have seen advancements in Progressive Web Apps (PWAs), Single Page Applications (SPAs), and WebAssembly. These technologies enhance the performance, functionality, and user experience of web applications, making them more powerful and versatile.

## **7 The Internet vs. the World Wide Web**

The World Wide Web (WWW) is a global system of interconnected documents and resources accessed through the internet. It spans the entire world, connecting users and re-

sources across geographical boundaries. The web is not just websites but also includes multimedia content, applications, and other digital assets. Web pages are linked together using hyperlinks, allowing users to navigate seamlessly between information. The web relies on the underlying infrastructure of the internet for communication and data transfer.

Essentially, the web is a layer built on top of the internet that provides a user-friendly way to access and interact with information. The internet is the foundation, providing the infrastructure, while the web is the application, providing the content and services.

## **8 The Role of Front-End Development in E-commerce**

### **8.1 Customer Experience and Engagement**

A well-designed website with intuitive navigation and engaging content can significantly improve the customer experience. Front-end development focuses on creating visually appealing and user-friendly interfaces that enhance customer engagement and satisfaction.

### **8.2 Online Transactions and Payments**

Web development is essential for creating secure and reliable online payment systems. Front-end developers ensure that payment forms are easy to use and visually consistent with the rest of the website, while back-end developers handle the secure processing of payment information.

### **8.3 Marketing and Branding**

Web development allows businesses to create a strong online presence, build brand awareness, and reach their target audience. A well-desi-

gned website reflects the brand's identity and values, helping to attract and retain customers.

## **8.4 Data Collection and Analysis**

Web development tools can be used to track user behavior, collect data, and analyze customer preferences. This information is valuable for making data-driven decisions and optimizing the website to better meet customer needs.

## **9 The Role of Back-End Development**

Back-end development handles the server-side logic of a web application. It involves data storage and management, interacting with databases to store and retrieve data. Back-end developers implement the business logic, which includes the rules and processes that govern the application's behavior. They also handle server configuration, setting up and managing web servers to host the application.

Back-end languages and frameworks include programming languages like Python, Ruby, Java, PHP, and Node.js, as well as frameworks like Django, Ruby on Rails, Spring Boot, and Express.js.

## **10 Client / Server Architecture**

### **10.1 Web Browser**

A web browser is a software application that interprets and displays web pages. Examples include Chrome, Firefox, Safari, and Edge. The browser fetches web pages from servers based on URLs, renders HTML, CSS, and JavaScript to create the visual interface, and handles user interactions and navigation.

## 10.2 Web Server

A web server is a computer system that stores and delivers web pages and other web content. Examples include Apache, Nginx, and IIS. The server receives HTTP requests from browsers, processes the requests, retrieves the requested content from storage, and sends the content back to the browser as an HTTP response.

## 11 Introduction to HTTP Protocol

### 11.1 What is HTTP?

HTTP (HyperText Transfer Protocol) is the foundation of data communication on the web. It is used for transmitting web pages and other resources between clients (browsers) and servers.

### 11.2 How HTTP Works

HTTP operates on a request-response model. The client (browser) sends an HTTP request to the server, which processes the request and sends back an HTTP response. This model enables the communication and data exchange that powers the web.

### 11.3 Common HTTP Methods

- **GET:** Requests data from a server, such as loading a web page.
- **POST:** Submits data to a server, such as submitting a form.

### 11.4 HTTP Status Codes

- **200 OK:** The request was successful.
- **404 Not Found:** The requested resource could not be found.
- **500 Internal Server Error:** The server encountered an error.

### 11.5 Why HTTP Matters

HTTP is the foundation of the web, enabling communication between browsers and servers.

Understanding HTTP is essential for grasping how the web works, making it a fundamental concept in web development and e-commerce.

## 12 Sequence Diagram

The sequence diagram illustrates the interaction between a browser and a web server. The browser sends an HTTP GET request for a URL, the server processes the request and retrieves the content, and then sends the HTTP response with the content back to the browser. Similarly, the browser can send an HTTP POST request with form data, the server processes the request and updates the data, and then sends an HTTP response with a success message. The browser can also send an HTTP request for an image, the server retrieves the image from storage, and sends the HTTP response with the image back to the browser.

## 13 URL (Uniform Resource Locator)

A URL is a unique address that identifies a specific resource on the internet. It consists of several components:

- **Protocol:** Specifies how the resource should be accessed (e.g., [http](#), [https](#), [ftp](#)).
- **Domain name:** The name of the website or server hosting the resource.
- **Path:** Indicates the specific location of the resource within the website's file structure.

Examples of URLs include <https://www.example.com> and <http://192.168.1.100/index.html>. In web development, URLs are used to link to other resources within a website, embed external content (e.g., images, videos, scripts), and create dynamic URLs based on user input or data.

## **14 The Building Blocks of Websites**

### **14.1 HTML (HyperText Markup Language)**

HTML defines the structure and content of a web page. It is the skeleton of the website, providing the basic framework for organizing text, images, links, and other elements.

### **14.2 CSS (Cascading Style Sheets)**

CSS styles the appearance of HTML elements, controlling their layout, colors, fonts, and more. It is like the clothing or design of the website, enhancing its visual appeal and user experience.

### **14.3 JavaScript**

JavaScript adds interactivity and dynamic behavior to web pages. It is the brain behind the website, allowing for things like user input, animations, and data updates.

### **14.4 PHP or JavaScript in Backend**

Server-side web page generation can be done using PHP or Node.js. The choice between PHP and Node.js often depends on factors like project requirements, team expertise, and performance needs.

## **15 Key Trends**

### **15.1 Single Page Applications (SPAs)**

SPAs enhance user experience by loading content dynamically without refreshing the page. They are commonly developed using frameworks like React, Angular, and Vue.js.

### **15.2 Responsive Web Design (RWD)**

RWD ensures that e-commerce sites look and function well on all devices, catering to

the growing number of mobile shoppers. It is achieved through CSS media queries and flexible grid layouts.

### **15.3 Server-Side Rendering (SSR) and Static Site Generation (SSG)**

SSR and SSG improve performance and SEO by pre-rendering pages on the server. Popular frameworks include Next.js (React) and Nuxt.js (Vue.js).

### **15.4 Headless CMS**

A headless CMS decouples the front-end and back-end, allowing for more flexibility and faster development. It uses APIs to deliver content, enabling seamless integration with various front-end technologies.

### **15.5 Progressive Web Apps (PWAs)**

PWAs combine the best of web and mobile apps, providing offline access, push notifications, and improved performance. They are built using standard web technologies like HTML, CSS, and JavaScript.

### **15.6 WebAssembly (Wasm)**

WebAssembly enables high-performance applications on the web, such as 3D product visualizations and augmented reality. It allows code written in languages like C, C++, and Rust to run in the browser.

## **16 Introduction to Web Development**

### **16.1 What is the Web?**

The web is a network of interconnected computers that allows websites to be accessed and displayed. It provides a platform for sharing information and interacting with content online.

## 16.2 The Role of a Web Developer

A web developer is responsible for creating, designing, and maintaining websites. They work on both the client-side (front-end) and server-side (back-end) components of a website.

## 16.3 Basic Components of a Website

- **Client-side:** HTML, CSS, JavaScript
- **Server-side:** PHP, Python, JavaScript
- **Databases:** MySQL, PostgreSQL, MongoDB

## 17 Web Development Stack

### 17.1 Front-End

The front-end stack includes:

- **HTML:** Defines the structure of web pages.
- **CSS:** Styles the appearance of web pages.
- **JavaScript:** Adds interactivity and dynamic behavior.
- **Frameworks:** React, Angular, Vue.js (simplify development and provide pre-built components).

### 17.2 Back-End

The back-end stack includes:

- **Programming languages:** Node.js, Python, PHP
- **Frameworks:** Express.js (Node.js), Django (Python), Laravel or Symphony (PHP)
- **Databases:** MySQL, PostgreSQL, MongoDB

## 18 Development Tools

### 18.1 Integrated Development Environments (IDEs)

- **Visual Studio Code:**
  - Features: Code editing, debugging, extensions, and version control integration.

- Benefits: Lightweight, highly customizable, and supports multiple programming languages.

- **WebStorm:**

- Features: Advanced code editing, debugging, and built-in tools for web development.
- Benefits: Specialized for JavaScript and web development, with strong support for frameworks like React, Angular, and Vue.js.

## 18.2 Version Control Systems

- **Git:**

- Features: Distributed version control, branching, and merging.
- Benefits: Facilitates collaboration, tracks changes, and allows for easy rollback.

- **GitHub:**

- Features: Code hosting, pull requests, and issue tracking.
- Benefits: Enhances collaboration, integrates with CI/CD pipelines, and provides project management tools.

## 19 Questions?

Feel free to ask any questions or share your thoughts.

## 20 Key Takeaways 1

### 20.1 E-commerce Models

- **B2B, B2C, C2C:**

- Examples: Alibaba, Amazon, eBay.

### 20.2 Web Development for E-commerce

- **Front-End:**

- HTML, CSS, JavaScript.

- **Back-End:**

- Server-side languages (PHP, JavaScript), databases, APIs.

## **20.3 Technical Focus**

- **User Experience, Performance, Security:**
  - User-friendly, optimized, and secure websites.

## **21 Key Takeaways 2**

### **21.1 Importance of Web Languages**

- **Communication, Problem-Solving, Competitive Advantage:**
  - Better collaboration, problem identification, effective contribution.

### **21.2 History of the WWW**

- **Evolution from 1989 to Present:**
  - From Tim Berners-Lee's proposal to modern advancements.

### **21.3 HTTP Protocol**

- **Request-Response Model:**
  - Methods: GET, POST.
  - Status Codes: 200, 404, 500.

### **21.4 Key Trends**

- **SPAs, RWD, SSR, SSG, Headless CMS, PWAs, WebAssembly:**
  - Enhancing user experience and performance.

### **21.5 Development Tools**

- **IDEs: Visual Studio Code, WebStorm.**
- **Version Control Systems: Git, GitHub.**