

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 About the Lecturer	1
2 Please Answer the Following Questions: .	1
3 Purpose of This Lecture	2
4 Evaluation	2
5 A Word About Large Language Models (AI - LLM)	2
6 Web Development for E-commerce	2
7 Overview of the Full Lecture	2
8 E-commerce Models and Technical Examples	3
8.1 B2B (Business-to-Business)	3
8.2 B2C (Business-to-Consumer)	3
8.3 C2C (Consumer-to-Consumer)	3
9 Technical Focus for E-commerce	3
10 Why an E-commerce Generalist should know web languages ?	4
11 History of the World Wide Web (WWW)	4
12 The Internet vs. the World Wide Web . . .	4
13 Client / Server Architecture	5
14 Introduction to HTTP Protocol	5
14.1 Sequence	6
15 URL (Uniform Resource Locator)	6
16 Key Trends	6
17 Usefull Development Tools	7
18 Questions?	7
19 Key Takeaways	7

1 About the Lecturer

Emmanuel BRUNO

- **Position:** Associate Professor at the University of Toulon
- **Department:** Information System Laboratory LIS - (UMR CNRS 7020)
- **Teaching Areas:**
 - Advanced Coding (mainly Java and Android)
 - Collaborative Tools (Continuous Integration, etc.)
 - Master Degree of Science and School of Engineering
- **Research Interests:**
 - Information Retrieval
 - Recommendation Systems
 - Data Querying and Analysis

2 Please Answer the Following Questions:

Raise your hand if you have experience with any of the above technologies.

1. **HTML (HyperText Markup Language)**
 - Have you ever created a web page using HTML?
 - Are you familiar with HTML tags and elements?
2. **CSS (Cascading Style Sheets)**
 - Have you used CSS to style a web page?
3. **JavaScript**
 - Have you written any JavaScript code?
 - In any other language ?
4. **Relational Databases**

- Have you worked with any relational databases (e.g., MySQL, PostgreSQL)?
- Do you understand basic SQL queries (SELECT, INSERT, UPDATE, DELETE)?

3 Purpose of This Lecture

- For All Students
 - **Mandatory General Culture:**
 - Understand the basics of web development and e-commerce.
 - Gain insights into how websites and online stores are built and maintained.
 - Learn about the key technologies and concepts that drive the web.
- For Technically Interested Students
 - **Technical Bootstrap:**
 - Dive deeper into HTML, CSS, JavaScript, and relational databases.
 - Get hands-on experience with building and styling web pages.
 - Explore advanced topics and tools used in modern web development.

4 Evaluation

- **Final Exam:**
 - Individual quiz on key concepts covered in the lecture.
- **Group Project:**
 - Collaborate in groups (up to five students).
 - Apply knowledge and skills in web development and e-commerce.
 - Final project deadline will be after the end of the lecture (details to come).

5 A Word About Large Language Models (AI - LLM)

- **Usage Inquiry:**
 - Are you using any AI tools? (Raise your hand)

- Which ones? (ChatGPT, Copilot, Gemini, etc.)
- What do you use them for?

- **Guidelines:**

- Avoid using AI tools directly to complete exercises; it may hinder your learning process.
- Use AI tools to:
 - Explain errors
 - Generate sample code
 - Provide insights and suggestions

- **Productivity Tips:**

- I will demonstrate how to use AI tools effectively to enhance your productivity and learning experience.

6 Web Development for E-commerce

- **Web Development:**
 - The process of creating and maintaining websites and web applications.
 - Involves a combination of front-end (client-side) and back-end (server-side) development.
- **E-commerce Web Development:**
 - Focuses specifically on building online stores and platforms for buying and selling products and services.
 - Integrates various functionalities such as product catalogs, shopping carts, payment gateways, and user accounts.

7 Overview of the Full Lecture

- *Objective 1:* Understand web front-end and create a simple web page
 - **Session 1: Introduction to Web Development and E-commerce**
 - **Session 2: HTML Fundamentals**
 - **Session 3: CSS Fundamentals**
 - **Session 4: Introduction to HTML Forms**

- *Objective 2:* Understand data storage and management
 - **Session 5: Introduction to Relational Databases and SQL**
- *Objective 3:* Gain insight into creating a real fullstack website
 - **Session 6: Introduction to Back-End Development**
 - **Session 7: FullStack Web Development with Remix**
 - **Session 8: Deployment and Performance Optimization**

- **Technical Aspects:**
 - User-friendly interfaces and responsive design.
 - Secure payment gateways and personalized recommendations.
- **Examples:**
 - **Lacoste:** Offers an online store with a user-friendly interface and secure payment options.
 - **Amazon:** Uses machine learning for personalized recommendations.
 - **eBay:** Implements real-time bidding systems.

8 E-commerce Models and Technical Examples

8.1 B2B (Business-to-Business)

- **Definition:**
 - Businesses sell products or services to other businesses.
- **Technical Aspects:**
 - **APIs (Application Programming Interfaces):**
 - Enable seamless integration with client systems.
 - **Secure Transactions:**
 - Ensure secure handling of transactions and data.
- **Examples:**
 - **Salesforce:**
 - Provides CRM (Customer Relationship Management) software to businesses, enabling them to manage customer relationships and data effectively.
 - **SAP:**
 - Offers enterprise resource planning (ERP) software that integrates various business processes, including finance, HR, and supply chain management.

8.2 B2C (Business-to-Consumer)

- Businesses sell products or services directly to consumers.

8.3 C2C (Consumer-to-Consumer)

- Platforms enable consumers to sell to each other.
- **Technical Aspects:**
 - User authentication and secure payment processing.
 - Rating and review systems to build trust.
- **Examples:**
 - **eBay:** Uses auction algorithms and secure payment systems.
 - **Airbnb:** Facilitates peer-to-peer trust user reviews and ratings.

9 Technical Focus for E-commerce

- **User Experience:**
 - Ensures that e-commerce sites are user-friendly, visually appealing, and functional across all devices.
 - Leads to higher customer satisfaction and increased sales.
- **Performance and Scalability:**
 - Optimizes site performance to handle high traffic and large volumes of transactions.
 - Ensures that the platform can scale as the business grows.
- **Security:**

- Implements security measures to protect sensitive customer data and prevent fraud.
- Ensures compliance with industry standards and regulations.

10 Why an E-commerce Generalist should know web languages ?

- **Improved Communication:**
 - Understanding web languages helps in better communication with developers and technical teams.
 - Facilitates clearer requirements and more effective collaboration on projects.
- **Enhanced Problem-Solving:**
 - Ability to identify and troubleshoot basic issues on e-commerce sites.
 - Helps in making informed decisions about website improvements and optimizations.
- **Competitive Advantage:**
 - Knowledge of web languages provides a competitive edge in the job market.
 - Enables e-commerce generalists to contribute more effectively to the development and maintenance of online stores.
- **Better User Experience:**
 - Ensures that e-commerce sites are user-friendly, visually appealing, and functional across all devices.
 - Leads to higher customer satisfaction and increased sales.

11 History of the World Wide Web (WWW)

- **1989 - Proposal:**
 - Tim Berners-Lee proposes the concept of the WWW at CERN.
- **1990 - First Website:**
 - First website (info.cern.ch) created by Tim Berners-Lee.
- **1991 - Public Release:**

- WWW made available to the public.

- **1993 - Mosaic Browser:**
 - Mosaic browser released, making the web more accessible.
- **1994 - Netscape Navigator:**
 - Netscape Navigator browser launched, popularizing the web.
- **1995 - Commercialization:**
 - Rapid growth of commercial websites (e.g., Amazon, eBay).
- **2000s - Web 2.0:**
 - Emergence of user-generated content and social media.
- **2010s - Mobile Web:**
 - Rise of smartphones and responsive web design.
- **2020s - Modern Web:**
 - Advancements in PWAs, SPAs, and WebAssembly.

12 The Internet vs. the World Wide Web

World Wide Web (WWW):

- A global system of interconnected documents and resources.
- Includes websites, multimedia content, applications, and digital assets.
- Uses hyperlinks to connect web pages, enabling seamless navigation.
- Relies on the internet for communication and data transfer.
- Facilitates machine-to-machine communication for automated data exchange.

Key Differences:

- **Internet:** The networking foundation providing the infrastructure.
- **Web:** The application layer providing content and services.

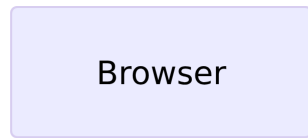
13 Client / Server Architecture

- **Web Browser**
 - A software application that interprets and displays web pages.
 - Examples: Chrome, Firefox, Safari, Edge
 - Functionality:
 - Fetches web pages from servers based on URLs.
 - Renders HTML, CSS, and JavaScript to create the visual interface.
 - Handles user interactions and navigation.
- **Web Server**
 - A computer system that stores and delivers web pages and other web content.
 - Examples: Apache, Nginx, IIS
 - Functionality:
 - Receives HTTP requests from browsers.
 - Processes requests and retrieves the requested content from storage.
 - Sends the content back to the browser as an HTTP response.
- **GET:** Requests data from a server (e.g., loading a web page).
- **POST:** Submits data to a server (e.g., submitting a form).
- **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.
- **Why HTTP Matters:**
 - **Foundation of the Web:** Enables communication between browsers and servers.
 - **User Experience:** Ensures web pages load and function correctly.
- **Conclusion:**
 - Understanding HTTP is essential for grasping how the web works, making it a fundamental concept in web development and e-commerce.

14 Introduction to HTTP Protocol

- **What is HTTP?**
 - HTTP (HyperText Transfer Protocol) is the foundation of data communication on the web.
 - Used for transmitting web pages and other resources.
- **How HTTP Works:**
 - **Request-Response Model:**
 - Client (Browser) sends an HTTP request to the server.
 - Server processes the request and sends back an HTTP response.
- **Common HTTP Methods:**

14.1 Sequence



Browser



Browser

15 URL (Uniform Resource Locator)

- A unique address that identifies a specific resource on the internet.
- 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:
 - <https://www.example.com>
 - <http://192.168.1.100/index.html>

- Role in web development:
 - Used to link to other resources within a website.
 - Used to embed external content (e.g., images, videos, scripts).
 - Used to create dynamic URLs based on user input or data.

- Processes request, retrieves content

16 Key Trends

Single Page Applications (SPAs)

- Enhance user experience by loading content dynamically without refreshing the page.
- Commonly developed using frameworks like React, Angular, and Vue.js.

Responsive Web Design

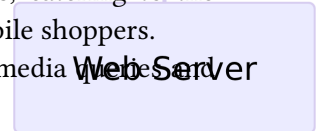
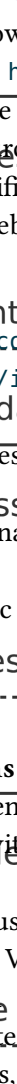
- Ensures e-commerce sites look and function well on all devices, catering to the growing number of mobile shoppers.
- Achieved through CSS media queries and flexible grid layouts.

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

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



Web Server



Web Server

Sends HTTP GET request for URL

Sends HTTP response with content

Sends HTTP POST request with form data

Sends HTTP response with success message

Sends HTTP request for image

Sends HTTP response with image

Processes request, retrieves content

Processes request, updates data

Retrieves image from storage

- **Headless Content Management Systems (CMS)**
 - Decouples the front-end and back-end, allowing for more flexibility and faster development.
 - Uses APIs to deliver content, enabling seamless integration with various front-end technologies.
- **Progressive Web Apps (PWAs)**
 - Combine the best of web and mobile apps, providing offline access, push notifications, and improved performance.
 - Built using standard web technologies like HTML, CSS, and JavaScript.
- **WebAssembly (Wasm)**
 - Enables high-performance applications on the web, such as 3D product visualizations and augmented reality.
 - Allows code written in languages like C, C++, and Rust to run in the browser.

17 Useful Development Tools

- **Integrated Development Environments (IDEs):**
 - **Visual Studio Code (with extension):**
 - 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.
- **Version Control Systems:**
 - **Git:**

- Features: Distributed version control, branching, and merging.
- Benefits: Facilitates collaboration, tracks changes, and allows for easy roll-back.
- **GitHub:**
 - Features: Code hosting, pull requests, and issue tracking.
 - Benefits: Enhances collaboration, integrates with CI/CD pipelines, and provides project management tools.

18 Questions?

- Feel free to ask any questions or share your thoughts.

19 Key Takeaways

- **Web and Its Architecture:**
 - Client-Server Model
 - HTTP/HTTPS Protocols
 - RESTful APIs
- **Web Development for E-commerce:**
 - **Front-End:**
 - Technologies: HTML, CSS, JavaScript
 - Focus: Creating user-friendly and visually appealing interfaces.
 - **Back-End:**
 - Technologies: Server-side languages, databases, APIs
 - Focus: Handling server logic, database interactions, and business processes.
- **Technical Focus:**
 - User Experience
 - Performance Optimization
 - Security Measures
- **E-commerce Models:**
 - B2B (Business-to-Business)
 - B2C (Business-to-Consumer)
 - C2C (Consumer-to-Consumer)
- **Key Trends:**
 - Single Page Applications (SPAs)
 - Responsive Web Design

- Server-Side Rendering (SSR) and Static Site Generation (SSG)
- Headless CMS
- Progressive Web Apps (PWAs)
- WebAssembly (Wasm)
- **Why an E-commerce Generalist Should Know Web Languages:**
 - Better Collaboration with Technical Teams
 - Informed Decision-Making
 - Enhanced Problem-Solving Skills