

# JavaScript Full-Stack Web Development — Course Guide

L3 LPECMN — Fall 2025

Emmanuel BRUNO  
Université de Toulon, LIS UMR CNRS 7020  
emmanuel.bruno@univ-tln.fr

2025-10-12

**Résumé** Course lectures and practices for JavaScript full-stack web development with AI-assisted workflows.

## Table des matières

1 What You'll Learn .....	2
2 Capstone mini-project (throughout the course) .....	2
3 Session Format .....	2
4 Schedule — 6 sessions × 3 h .....	2
4.1 Session 1 — Introduction & Foundations: HTML/CSS & LLMs (3 h) .....	3
4.2 Session 2 — Full-Stack Architecture & JavaScript Fundamentals (3 h) .....	3
4.3 Session 3 — React Frontend Development (3 h) .....	3
4.4 Session 4 — Backend Fundamentals: Node.js & Express APIs (3 h) .....	3
4.5 Session 5 — Database Integration & Advanced Patterns (3 h) .....	3
4.6 Session 6 — Full-Stack Integration & Assessment (3 h) .....	4
5 Assessment & grading .....	4
5.1 Attendance & Participation (10%) .....	4
5.2 Final Quiz (60%) .....	4
5.3 Group Project: Multi-Source Product Review Aggregator (30%) .....	4
5.3.a Optional Group Extension Project .....	5
6 Submission Deadlines .....	5
7 Course Policies .....	5
7.1 Late Submission Policy .....	5
7.2 Academic Integrity & LLM Usage .....	6
7.3 Support & Communication .....	6
7.4 Team Formation & Collaboration .....	6

7.5 Attendance Requirements .....	6
8 FAQ .....	6

## 1 What You'll Learn

In this intensive 18-hour course (6 sessions × 3h), you'll discover modern full-stack JavaScript development by building a complete Multi-Source Product Review Aggregator. You'll learn React for interactive frontends, Node.js and Express for REST APIs, MySQL for data persistence, and how to integrate all three layers into a working web application. Along the way, you'll learn to leverage LLMs (ChatGPT, Copilot) effectively while maintaining code quality. you'll be proposed to use professional tools (Docker, Git, VS Code)

### Prerequisites:

- **Essential:** Basic HTML and CSS (semantic elements, selectors, responsive concepts)
- **Helpful:** Some programming experience (PHP, Python, Java, or similar) for logical thinking
- **Not required:** Prior JavaScript experience (we start from fundamentals)
- **Mindset:** Willingness to learn AI-assisted development workflows

## 2 Capstone mini-project (throughout the course)

- **Project:** Multi-Source Product Review Aggregator
- **Description:** Build a full-stack web application that aggregates product reviews from multiple e-commerce platforms (Amazon, BestBuy, etc.) into a unified interface. Features include:
  - Product catalog with search and filtering
  - Fetch reviews on-demand from external sources (simulated scraper)
  - Display aggregated reviews with statistics (average rating, count by source)

- Persistent storage of reviews in MySQL database
- React frontend with interactive components
- REST API backend with Express
- **Tech stack:** React (Vite) + Node.js + Express + MySQL + Docker Compose
- **Milestones:**
  - End of Session 1: Static HTML/CSS prototypes with Tailwind
  - End of Session 2: Interactive vanilla JS catalog with fetch basics
  - End of Session 3: React components with mock API integration
  - End of Session 4: Express backend with REST endpoints
  - End of Session 5: MySQL persistence with migrations
  - End of Session 6: Full-stack integration + demo
- **See** <https://github.com/ebpro/fullstack-minimal-app/> for the project source code.
- **See** [https://github.com/ebpro/fullstack-minimal-app/blob/develop/LEARNING\\_NOTES.md](https://github.com/ebpro/fullstack-minimal-app/blob/develop/LEARNING_NOTES.md) for learning notes about the project.

## 3 Session Format

Each 3-hour session typically follows this structure: two hours of lecture followed by one hour of hands-on practice. Lectures introduce key concepts and patterns, while practices provide guided exercises to apply what you've learned. You will work individually on progressive milestones during Sessions 1-5, then form teams for final integration and demo in Session 6.

## 4 Schedule — 6 sessions × 3 h

#### 4.1 Session 1 — Introduction & Foundations: HTML/CSS & LLMs (3 h)

- Lecture:  
webdev\_00\_L\_00\_Introduction.qmd — « Introduction — Web Development & E-commerce » (course overview, full-stack landscape, development environment setup).
- Lecture:  
webdev\_01\_L\_02A\_LLMQuickStart.qmd — « LLM Quick Start for Developers » (AI-assisted development workflow, GitHub Copilot setup, CRISP framework).
- Lecture:  
webdev\_01\_L\_03\_HTMLCSSEssentials.qmd — « HTML & CSS Essentials for AI-Assisted Development » (semantic HTML, Tailwind CSS, accessibility).
- Practice: webdev\_01\_P\_HTMLCSS-LLMs.qmd — « Practice 1 — HTML/CSS with LLM Assistance » (create static prototypes, use Tailwind via CDN, verify LLM outputs).
- Session goals & deliverables: repo & dev-server setup, semantic product-card prototype.

#### 4.2 Session 2 — Full-Stack Architecture & JavaScript Fundamentals (3 h)

- Lecture:  
webdev\_02\_L\_01\_BackAndFront.qmd — « Back vs Front: Full-Stack Overview » (full-stack architecture, client-server model, solution patterns).
- Lecture:  
webdev\_02\_L\_02\_JavaScriptForReact.qmd — « JavaScript for React » (essential foundations, async programming, DOM basics, React motivation).
- Practice:  
webdev\_02\_P\_JavaScriptForReact.qmd — « Practice: JavaScript for React — Product

Review Interface » (building frontend foundation for review aggregator) (90 min).

- Session goals & deliverables: interactive product interface with vanilla JavaScript, fetch API integration, understanding of React benefits.

#### 4.3 Session 3 — React Frontend Development (3 h)

- Lecture: webdev\_03\_L\_ReactFrontendDevelopment.qmd — « React Frontend Development » (components, JSX, hooks, state management, interactive features).
- Practice:  
webdev\_03\_P\_ReactReviewAggregator.qmd — « Practice 3 — React Review Aggregator Frontend » (build interactive product review interface with React).
- Session goals & deliverables: Complete React frontend with components (ProductHeader, ReviewFetcher, ReviewStats, ReviewList), mock API integration.

#### 4.4 Session 4 — Backend Fundamentals: Node.js & Express APIs (3 h)

- Lecture:  
webdev\_04\_L\_NodeExpressBackend.qmd — « Node.js & Express Backend Development » (Express fundamentals, REST APIs, middleware, database integration basics).
- Practice:  
webdev\_04\_P\_NodeExpressBackend.qmd — « Practice 4 — Node.js Express Backend » (- build REST API for review aggregator with MySQL integration).
- Session goals & deliverables: Express server with REST endpoints for review aggregation, MySQL database integration, external scraper simulation.

#### 4.5 Session 5 — Database Integration & Advanced Patterns (3 h)

- **Lecture:**  
webdev\_05\_L\_DatabaseIntegration.qmd — « Database Integration for Full-Stack Apps » (Express + MySQL integration patterns, connection pools, transactions, optimization).
- **Practice:**  
webdev\_05\_P\_DatabaseIntegration.qmd — « Practice 5 — Database Integration » (replace in-memory arrays with MySQL persistence, optimize queries).
- **Team Formation:** Form teams.
- **Session goals & deliverables:** MySQL connection pool integration, optimized database queries, transaction handling for data consistency.

## 4.6 Session 6 — Full-Stack Integration & Assessment (3 h)

- **Final Quiz:** Individual written quiz (30 min) covering full-stack concepts with emphasis on integration (React ↔ Express ↔ MySQL, REST APIs, data flow).
- **Practice:**  
webdev\_06\_P\_FullStackIntegration.qmd — « Practice 6 — Full-Stack Integration » (connect frontend to backend, complete end-to-end integration).
- **Session goals & deliverables:** Complete full-stack application, team formation completed, integration demo ready.

## 5 Assessment & grading

The final grade is based on three components: Attendance & Participation, a Final Quiz, and a Group Project (completed after the course).

### 5.1 Attendance & Participation (10%)

- **Attendance tracking:** 6 sessions × 3h = 18h total
- **Grading:**
  - Full attendance (6/6 sessions): 10%

- 5 sessions: 8%
- 4 sessions: 6%
- 3 sessions or fewer: 0%

- **Participation bonus:** Active engagement during lectures and practices (questions, helping peers, demonstrating understanding)

### 5.2 Final Quiz (60%)

- **Format:** Individual written quiz (30 minutes)
- **Timing:** Beginning of Session 6, before starting full-stack integration practice in groups (24th October 2025)
- **Coverage:** Fundamental concepts with **emphasis on full-stack integration**
- **Topics: (can be adjusted depending on class progress)**
  - **Full-stack integration** (React ↔ Express ↔ MySQL, REST APIs, data flow) (20%)
  - **Backend & databases** (Express routing, middleware, SQL queries) (15%)
  - **Frontend frameworks** (React components, hooks, state management, JSX) (10%)
  - **JavaScript fundamentals** (async/await, promises, DOM manipulation, ES6+) (8%)
  - **HTTP & REST concepts** (methods, status codes, request/response cycle) (5%)
  - **HTML/CSS & accessibility** (semantic HTML, responsive design) (2%)
- **Question types:** Multiple choice, code snippet analysis.
- **Focus:** Understanding how different layers communicate and integrate

### 5.3 Group Project: Multi-Source Product Review Aggregator (30%)

- **Format:** Group project (5 students per team), merges and extends the work completed throughout the course by the students, polish and add optional features.
- **Timeline:**

- **Sessions 1-6:** Students work individually on progressive milestones
- **End of Session 5:** Teams form (instructor assigns team numbers)
- **Session 6:** Team consolidation work, merging individual codebases, planning integration
- **Submission deadline:** 1 month after Session 6 (1st December 2025)
- **Submission requirements:**
  - Email and optional Source code repository (GitHub) including:
    - a clear README.md in markdown describing the functionalities
    - a One-page document LLM\_Usage.md explaining LLM usage (prompts, verification, challenges)
    - 3-minute maximum demo video via a private Youtube link (screencast)

#### Grading breakdown:

- **Functionality (15%):** Product catalog, review fetching, aggregation/statistics, MySQL persistence, full-stack integration
- **Code Quality & Documentation (8%):** Project structure, README, code comments/naming, error handling, Git history
- **LLM Usage Documentation (4%):** Prompts with context, verification/corrections, critical reflection
- **Demo Video (3%):** Features demo, full-stack integration, technical challenges

#### 5.3.a Optional Group Extension Project

If you are motivated, you can implement additional features beyond the core requirements to demonstrate deeper understanding and ambition. This will be rewarded with bonus points. Possible features include:

- Those presented in the lectures and practical work

- Some ambitious teams may choose to implement more feature for bonus points (some ideas):
  - **Real scraper integration** with actual e-commerce APIs
  - **Advanced analytics dashboard** with charts (Chart.js/Recharts)
  - **Admin panel** for review moderation and flagging

## 6 Submission Deadlines

- **Quiz date:** at the beginning of Session 6: **24th October 2025**
- **Group project due:** 1 month after Session 6: **1st December 2025** (with 3-min video, README, LLM\_Usage.md)
- **All submissions via:**
  - **Email:** emmanuel.bruno@univ-tln.fr with subject « WebDev L3 LPECMN - Team[XX] » (where XX is your assigned team number). The email must include:
    - Source code repository link (GitHub) OR .zip archive if no repository
    - README.md (included in repository/archive)
    - LLM\_Usage.md (included in repository/archive)
    - Demo video link (YouTube private/unlisted link)
  - **Optional:** Submit via GitHub Classroom link (if provided)

## 7 Course Policies

### 7.1 Late Submission Policy

- **Group Project:**
  - Up to 24 hours late: –10% penalty
  - 24-48 hours late: –20% penalty
  - More than 48 hours late: –50% penalty
  - More than 1 week late: Not accepted (0%)

## 7.2 Academic Integrity & LLM Usage

**Permitted:** - Using LLMs (ChatGPT, Claude, GitHub Copilot, etc.) to generate code snippets, explanations, or debugging help - Consulting documentation, tutorials, Stack Overflow, and other learning resources - Collaborating with team members on group project - Discussing concepts with classmates (but not sharing code for individual work during Sessions 1-6)

**Required:** - Document your LLM usage in LLM\_Usage.md (prompts, verification steps, corrections made) - Understand and be able to explain all code you submit (instructor may ask questions) - **Verify and test all LLM-generated code (do not blindly accept outputs)**

**Prohibited:** - Copying entire solutions from other students or online sources without attribution - Submitting LLM-generated code without understanding or verification - Having someone else complete your work (during individual phases or quiz) - Sharing your individual work (Sessions 1-6) with other students before team formation

**Violations:** Academic integrity violations will result in a grade of 0 for the affected component and may be reported to university administration.

## 7.3 Support & Communication

- **Instructor:** Emmanuel Bruno (emmanuel.bruno@univ-tln.fr)
- **Questions:**
  - During sessions: Ask during Q&A time or practice periods
  - Outside sessions: Use the discord channel of your group or private message the instructor on Discord
- **Technical Issues:**
  - ask during sessions or use Discord

## 7.4 Team Formation & Collaboration

- **Team Assignment:** Teams are built at end of Session 5
- **Team Numbers:** Teams will be numbered G\_T for example G01\_T01 for the Team 1 in Groupe 1 (use this number in email subject lines or on Discord).
- **Team Dynamics:** All team members expected to contribute. **If there are team conflicts, contact instructor immediately.**
- **Individual Accountability:** During Sessions 1-5, each student must complete their own milestones. After team formation, collaboration is expected.

## 7.5 Attendance Requirements

- **Mandatory Attendance:** Regular attendance is required to keep up with course material and participate in practices.
- **Absences:** Notify instructor in advance when possible. Excused absences (medical, university-approved) may be accommodated but you are responsible for catching up on content.

## 8 FAQ

**Q: Can I use a different tech stack (Vue instead of React, PostgreSQL instead of MySQL)?**

A: No. The course is designed around React + Express + MySQL. Assessment and support are based on this stack.

**Q: What if my team member isn't contributing?**

A: Document the issue and contact the instructor immediately. In severe cases, individual grades may be adjusted.

**Q: Can I submit the project individually instead of with my team?**

A: No. The project is team-based only. If you

want individual bonus work, focus on exceeding requirements in the main group project and document it in the README.

**Q: How much should I rely on LLMs?**

A: LLMs are tools, not replacements for learning. Use them to accelerate coding and explain concepts, but you must understand and verify everything. **You'll be tested individually on the quiz.**

**Q: What if Docker doesn't work on my computer?**

A: Use Option 2 (local installation with Node.js + MySQL). Bring your laptop if you need setup help or ask on Discord.

**Q: Will solutions be provided?** A: The sample project is available on GitHub. However, you should not copy it directly. The goal is to learn by doing. You will extend it for your project.