ELEVO LABS
LEARN. INNOVATE. ELEVATE.

Course Overview

Course Details

Course Name: Modern JavaScript Deep Dive: Advanced Concepts & Design Patterns

Instructor: Jane Smith

Created On: 16 June, 2025

Updated On: 16 June, 2025

Price: 4000 INR

Duration: 3 Weeks

Modules: 11

Language: English, Hindi

Level: Expert

Certifications: Yes

Course Description

Take your JavaScript skills from foundational to formidable with **Modern JavaScript Deep Dive: Advanced Concepts & Design Patterns**. This course is meticulously designed for developers who have a solid understanding of JavaScript fundamentals and are ready to tackle complex challenges, optimize performance, and write enterprise-grade code.

We begin by solidifying your grasp of advanced scope, closures, and the intricacies of the `this` keyword. The course then takes a deep dive into **asynchronous JavaScript**, mastering Promises, Async/Await, and understanding the Event Loop to handle complex operations gracefully. You'll explore advanced aspects of **Object-Oriented Programming (OOP) in**

JavaScript, breaking down each principle and applying them with prototypes and modern classes, along with **Functional Programming (FP) in JavaScript**, and common **Design Patterns** used in large-scale applications.

A significant focus is placed on the **modern JavaScript ecosystem**, covering essential tooling like NPM, Webpack/Vite, and Babel, crucial for efficient development workflows. You'll learn robust **error handling** and explore basic **testing strategies** to ensure code quality. Finally, we'll touch upon cutting-edge topics like TypeScript, preparing you for the demands of modern web development.

What You'll Learn:

- How JavaScript code is executed: Execution Context, Call Stack, Hoisting, Scope, Closures, and `this`.
- Modern ES6+ features: Arrow Functions, Destructuring, Spread/Rest Operators, Template Literals.
- Comprehensive Asynchronous JavaScript: Callbacks, Promises, Async/Await, Event Loop.
- **In-depth Object-Oriented Programming (OOP) principles and their application in JavaScript with prototypes and ES6 Classes.**
- Functional Programming paradigms and techniques.
- Implementing common JavaScript Design Patterns.
- Mastering the modern JavaScript tooling ecosystem: NPM, Module Bundlers (Webpack/Vite), Transpilers (Babel), Linters (ESLint).
- Advanced Error Handling and Custom Error Types.
- Introduction to JavaScript Testing with frameworks like Jest/Vitest.
- An introduction to TypeScript for type safety.
- Best practices for writing clean, maintainable, and scalable JavaScript code.

Who Is This Course For?

This course is for intermediate to advanced JavaScript developers who have a strong grasp of fundamental JavaScript concepts (variables, data types, basic functions, DOM manipulation, control flow). It's ideal for those looking to deepen their understanding, transition to building complex applications, optimize existing code, or prepare for senior front-end development roles.

Prerequisites:

- **Solid JavaScript Fundamentals:** A strong understanding of core JavaScript (covered in 'Vanilla JavaScript: The Absolute Beginner's Guide' or equivalent knowledge) is essential. This includes variables, data types, functions, arrays, objects, and basic DOM manipulation.
- **Basic HTML & CSS:** Familiarity with HTML structure and basic CSS styling.
- **A Computer/Laptop:** A working computer (Windows, macOS, or Linux) capable of running modern development tools.
- **Command Line Basics:** Comfort with basic command-line operations (e.g., navigating directories, running commands).
- **A Strong Desire to Master Advanced Concepts:** Be prepared for in-depth technical discussions and complex coding challenges.

Includes multiple challenging mini-projects, a comprehensive capstone project, and a certificate upon completion. Available in English and Hindi.

Course Curriculum Details

Module 1: Module 1: JavaScript Execution Model - How Code Runs

10 Lessons

1. 1. Introduction to JavaScript Execution: Interpreted vs. Compiled **20min**
2. 2. Understanding the JavaScript Engine (V8 Overview) **25min**
3. 3. The Global Execution Context and Its Components **30min**
4. 4. Function Execution Contexts and the Call Stack **35min**
5. 5. Hoisting: Variables and Functions in Detail **30min**
6. 6. Lexical Scoping and Scope Chain Deep Dive **35min**
7. 7. Closures: The Power of Persistent Lexical Scope **40min**

8. 8. The `this` Keyword: Understanding Context Binding **35min**

9. 9. `call()`, `apply()`, and `bind()` for Explicit `this` Control **30min**

10. 10. Recap & Practical Task: Trace Execution Flow and Implement Closures **1hr**

Module 2: Module 2: Modern JavaScript (ES6+) Features for Cleaner Code

8 Lessons

1. 11. Arrow Functions: Syntax, Benefits, and `this` Binding Differences **30min**

2. 12. Destructuring Assignment (Arrays and Objects) for Concise Code **35min**

3. 13. Spread Operator (`...`) for Array and Object Manipulation **30min**

4. 14. Rest Parameters (`...`) for Flexible Function Arguments **25min**

5. 15. Template Literals and Tagged Templates for Advanced String Handling **20min**

6. 16. Enhanced Object Literals (Shorthand Properties, Method Properties, Computed Property Names) **20min**

7. 17. Default Parameters and Named Parameters (via Destructuring) **20min**

8. 18. Recap & Practical Task: Refactor Existing Code Using Modern ES6+ Syntax **50min**

Module 3: Module 3: Mastering Asynchronous JavaScript

9 Lessons

1. 19. The Asynchronous Nature of JavaScript: Blocking vs. Non-Blocking **20min**

2. 20. The Event Loop, Callback Queue, and Microtask Queue in Depth **35min**

3. 21. Callbacks Revisited: Asynchronous Patterns and Callback Hell Solutions **30min**

4. 22. Promises: Introduction, States, and Chaining (`.then()`, `.catch()`, `.finally()`) **45min**

5. 23. Composing Promises: `Promise.all()`, `Promise.race()`, `Promise.allSettled()`, `Promise.any()` **40min**

6. 24. Async/Await: Writing Synchronous-Looking Asynchronous Code **50min**

7. 25. Error Handling in Asynchronous Code with Promises and Async/Await **30min**

8. 26. Fetch API: Advanced Usage, Request Options, and Interceptors (Conceptual) **40min**

9. 27. Recap & Practical Task: Build a Complex Data Dashboard with Asynchronous Operations **1hr 15min**

Module 4: Module 4: Deep Dive into Object-Oriented Programming (OOP) in JavaScript

14 Lessons

1. 28. Introduction to OOP: Fundamental Concepts and Paradigms **25min**

2. 29. OOP Principle: Encapsulation (Data Hiding and Bundling) **30min**

3. 30. OOP Principle: Inheritance (Code Reusability and Hierarchy) **30min**

4. 31. OOP Principle: Polymorphism (Many Forms, Single Interface) **25min**

5. 32. OOP Principle: Abstraction (Showing Essentials, Hiding Details) **25min**

6. 33. Prototypal Inheritance in JavaScript: The `__proto__` Chain **35min**

7. 34. Constructor Functions and the `new` Keyword **30min**

8. 35. ES6 Classes: Syntactic Sugar over Prototypes (Class Declaration, Expression) **40min**

9. 36. Class Methods: Instance Methods, Static Methods, and `this` in Classes **35min**

10. 37. Class Inheritance with `extends` and `super()` **35min**

11. 38. Getters and Setters for Controlled Property Access **25min**

12. 39. Private Class Fields (ECMAScript 2022) for True Encapsulation **20min**

13. 40. Mixins and Composition over Inheritance (Advanced OOP Patterns) **30min**

14. 41. Recap & Practical Task: Design and Implement an OOP-based Application (e.g., a Library Management System) **1hr 15min**

Module 5: Module 5: Functional Programming (FP) Concepts in JavaScript

7 Lessons

1. 42. Introduction to Functional Programming: Core Principles and Benefits **25min**

2. 43. Pure Functions, Side Effects, and Immutability **30min**

3. 44. Higher-Order Functions: Functions as First-Class Citizens **25min**

4. 45. Currying and Partial Application for Function Transformation **35min**

5. 46. Function Composition and Pipelining Data **30min**

6. 47. Recursion and Iteration in FP Context **25min**

7. 48. Recap & Practical Task: Refactor a Data Processing Pipeline Using FP Principles **50min**

Module 6: Module 6: JavaScript Design Patterns

8 Lessons

1. 49. Introduction to Design Patterns: Benefits and Categories (Creational, Structural, Behavioral) 20min
2. 50. Creational Patterns: Singleton Pattern 30min
3. 51. Creational Patterns: Factory Pattern 25min
4. 52. Structural Patterns: Module Pattern and Revealing Module Pattern 35min
5. 53. Structural Patterns: Facade Pattern 25min
6. 54. Behavioral Patterns: Observer Pattern (Pub/Sub) 30min
7. 55. Behavioral Patterns: Strategy Pattern 25min
8. 56. Recap & Practical Task: Apply a Design Pattern to Solve a Common Problem 50min

Module 7: Module 7: Modern JavaScript Modules and Tooling

7 Lessons

1. 57. ES Modules: Advanced Import/Export Syntax and Dynamic Imports 30min
2. 58. Node Package Manager (NPM) and Yarn: Advanced Usage 30min
3. 59. Module Bundlers: Why We Need Them (Webpack/Vite Concepts) 40min
4. 60. Transpilers: Babel Configuration and Presets 30min
5. 61. Code Linting with ESLint: Custom Rules and Integration 25min
6. 62. Code Formatting with Prettier 15min

7. 63. Recap & Practical Task: Set up a Modern JavaScript Project with Bundling and Linting **1hr**

Module 8: Module 8: Advanced Error Handling & Debugging

7 Lessons

1. 64. Custom Error Types and Extending `Error` **25min**
2. 65. Handling Asynchronous Errors (Promises, Async/Await Error Propagation) **30min**
3. 66. Global Error Handling (`window.onerror`, `unhandledrejection`) **25min**
4. 67. Debugging Strategies in Browser DevTools (Breakpoints, Call Stack, Scopes) **35min**
5. 68. Using Node.js Debugger **20min**
6. 69. Logging Best Practices **15min**
7. 70. Recap & Practical Task: Implement Robust Error Handling in a Web Application **45min**

Module 9: Module 9: Introduction to Testing JavaScript Applications

7 Lessons

1. 71. Importance of Testing: Unit, Integration, and End-to-End Testing Overview **20min**
2. 72. Test Driven Development (TDD) Workflow **25min**
3. 73. Setting up a Unit Testing Environment with Jest/Vitest **30min**
4. 74. Writing Effective Unit Tests: Assertions, Matchers, and Test Suites **40min**
5. 75. Mocking Dependencies and Spying on Functions **35min**

6. 76. Basic Integration Testing Concepts **20min**

7. 77. Recap & Practical Task: Write Unit Tests for a Complex JavaScript Module **1hr**

Module 10: Module 10: Introduction to TypeScript (Type-Safe JavaScript)

7 Lessons

1. 78. Why TypeScript? Benefits for Scalability and Maintainability **25min**

2. 79. Basic Types, Type Annotations, and Type Inference **30min**

3. 80. Interfaces and Type Aliases for Defining Shapes **25min**

4. 81. Classes and Enums in TypeScript **30min**

5. 82. Understanding the TypeScript Compiler (`tsc`) and `tsconfig.json` **25min**

6. 83. Integrating TypeScript into a JavaScript Project Workflow **20min**

7. 84. Recap & Practical Task: Convert a Small JavaScript Application to TypeScript **1hr**

Module 11: Module 11: Capstone Project: Building a Complex Single Page Application (SPA)

7 Lessons

1. 85. Project Planning and Architecture: Applying Advanced Concepts and Patterns **1hr**

2. 86. Setting Up the Project with a Modern Build Tool and Development Server **1hr**

3. 87. Developing Core Features with Advanced JS, Asynchronous Patterns, and OOP/FP **2hr**

4. 88. Implementing Robust Error Handling and Testing **1hr 30min**

5. 89. Deployment Strategies for Modern JavaScript Applications **45min**

6. 90. Final Project Presentation and Code Review **1hr**

7. 91. Recap & Course Completion **40min**

This curriculum is subject to minor adjustments to ensure the most up-to-date and effective learning experience.