MERN+ADBMS Curriculum

Outcome

Students can build full-stack production-ready applications using MERN, apply advanced DBMS concepts practically with MongoDB & NoSQL systems, and design scalable real-world architectures.

Module 1: Web & JavaScript Foundations

⏱ 4 Hours

Topics

- Introduction to Web
- How the Browser Works
- JavaScript Basics (variables, data types, operators, control flow)

Students will be able to:

- Explain how the browser loads and executes JavaScript
- Write basic JavaScript logic without memorization
- Read and understand simple JS code written by others
- Build a correct mental model of how web pages actually work

LeetCode Problems Students Can Solve:

- **1342. Number of Steps to Reduce a Number to Zero**
- **1281. Subtract the Product and Sum of Digits of an Integer**

- **412. Fizz Buzz**
- **1528. Shuffle String**
- **2011. Final Value of Variable After Performing Operations**

Module 2: JavaScript Execution & Scope

⏱ 6 Hours

Topics

- Execution Context
- Hoisting
- Scope & Lexical Environment
- Closures

Students will be able to:

- Predict output of JavaScript code involving hoisting and scope
- Explain how variables are resolved during execution
- Use closures confidently in real applications
- Debug tricky output-based interview questions

LeetCode Problems Students Can Solve:

- **2620. Counter**
- **2665. Counter II**
- **2634. Filter Elements from Array**
- **2666. Allow One Function Call**

- **2694. Event Emitter (basic version)**

Module 3: Functions & Programming Paradigms

⏱ 3 Hours

Topics

- Function Declarations vs Expressions
- Arrow Functions
- Imperative vs Declarative Programming

Students will be able to:

- Choose the right function style for a given problem
- Write readable and maintainable JavaScript functions
- Refactor imperative code into cleaner declarative code
- Understand why JavaScript functions behave the way they do

LeetCode Problems Students Can Solve:

- **2626. Array Reduce Transformation**
- **2629. Function Composition**
- **2635. Apply Transform Over Each Element**
- **2631. Group By**
- **2667. Create Hello World Function**

Module 4: Asynchronous JavaScript

⏱ 7 Hours

Topics

- Callbacks
- Promises
- Async / Await
- Event Loop (Microtasks vs Macrotasks)
- Error Handling in Async Code

Students will be able to:

- Explain the JavaScript event loop clearly
- Write async code without callback hell
- Handle real-world async scenarios (API calls, retries, sequencing)
- Confidently answer async interview questions

LeetCode Problems Students Can Solve:

- **2621. Sleep**
- **2623. Memoize**
- **2627. Debounce**
- **2637. Promise Time Limit**
- **2648. Generate Fibonacci Sequence (Async)**

Module 5: DOM & Browser APIs

⏱ 7 Hours

Topics

- DOM Manipulation
- Events & Event Handling
- Event Propagation (Capturing & Bubbling)
- Event Delegation
- Debouncing & Throttling

Students will be able to:

- Build interactive UI features using pure JavaScript
- Handle complex event flows correctly
- Optimize UI performance using debouncing and throttling
- Solve machine-coding DOM questions in interviews

LeetCode Problems Students Can Solve:

- **2628. JSON Deep Equal**
- **2676. Throttle**
- **2721. Execute Asynchronous Functions in Parallel**
- **2722. Join Two Arrays by ID**
- **2675. Array of Objects to Matrix**

Module 6: Object-Oriented JavaScript

⏱ 5 Hours

Topics

- `this` Keyword
- Classes & Constructors
- Prototypes & Inheritance
- call, apply, bind

Students will be able to:

- Predict the value of `this` in any context
- Implement inheritance using prototypes and classes
- Write reusable and extensible object-oriented code
- Solve LLD-style JavaScript problems

LeetCode Problems Students Can Solve:

- **2622. Cache With Time Limit**
- **2693. Call Function with Custom Context**
- **2695. Array Wrapper**
- **2723. Add Two Promises**
- **2633. Convert Object to JSON String**

Module 7: Advanced JavaScript Internals

⏱ 4 Hours

Topics

- Higher Order Methods (map, filter, reduce)
- Polyfills

- Memory Management & Garbage Collection

Students will be able to:

- Write custom polyfills for core JavaScript methods
- Optimize code for performance and memory usage
- Debug memory leaks and unexpected behavior
- Think like an engineer instead of a syntax-writer

LeetCode Problems Students Can Solve:

- **2632. Curry**
- **2625. Flatten Deeply Nested Array**
- **2619. Array Prototype Last**
- **2692. Make Object Immutable**
- **2649. Nested Array Generator**

Module 8: Projects (Hands-on)

⏱ 6 Hours

Projects

- Weather App 🌦️
- Kanban Board 🗂️
- YouTube Clone 📺

Students will be able to:

- Design end-to-end frontend applications
- Structure code using LLD principles
- Handle real-world UI + state complexity
- Explain design decisions confidently in interviews

Module 9: Interview & Machine Coding Preparation

⏱ 3 Hours

Topics

- Machine Coding Patterns
- JavaScript Interview Questions
- Real Interview Simulations

Students will be able to:

- Solve machine-coding rounds within time limits
- Avoid common JavaScript interview traps
- Clearly explain thought process and trade-offs
- Perform confidently in frontend interviews

Module 10: ADBMS Fundamentals (Applied to MERN)

⏱ 6 Hours

Topics

- File System vs DBMS vs RDBMS
- Relational vs NoSQL
- CAP Theorem
- ACID vs BASE

- Normalization vs Denormalization
- Indexing Concepts
- Transactions
- Concurrency Control
- Sharding & Replication

Students will be able to:

- Understand when to use SQL vs NoSQL

- Design scalable databases

Module 11: MongoDB & NoSQL in Practice
⏱ 8 Hours

Topics

- MongoDB Architecture
- Documents & Collections
- CRUD Operations
- Schema Design Principles
- Embedding vs Referencing
- Indexes in MongoDB
- Aggregation Pipeline
- Transactions in MongoDB
- Mongoose ODM
- Pagination & Query Optimization

Students will be able to:

- Design optimized MongoDB schemas
- Integrate MongoDB with Express

Module 12: Object-Oriented & Advanced DB Concepts

⏱ 4 Hours

Topics

- Object-Oriented DBMS Concepts
- Object Identity
- Complex Data Types
- ORM vs ODM
- Mapping Objects to Documents
- Data Modeling for Full Stack Apps

Students will be able to:

- Bridge backend models with database design
- Think database-first in architecture

Total Duration

⏱ 42 – 44 Hours

Final Outcome 🎯

After completing this curriculum, students will be able to:

- Build production-ready frontend applications
- Understand JavaScript internals at a deep level
- Solve machine-coding and LLD interview problems
- Reason about performance, memory, and scalability