

MERN = MongoDB + Express.js + React.js + Node.js



1 Prerequisites (Frontend + Core Web)

HTML5 (Day - 3)

- Structure [Link](#) & semantics [Link](#)
- Forms&input types [Like](#)
- Tables [Link](#)
- Audio/video [Like](#)
- SEO-friendly HTML [Like](#)
- Accessibility (ARIA roles, alt text) [Like](#)
- Html5form validation

```
<form action="" method="post">

    <fieldset>

        <legend>Booking Details</legend>

        <div>

            <label for="name">Name (required) :</label>

            <input type="text" id="name" name="name"
value="" aria-describedby="name-format">
```

```
        <span id="name-format" class="help">Format:  
firstname lastname</span>  
  
    </div>  
  
    <div>  
  
        <label for="email">Email (required) :</label>  
  
        <input type="text" id="email" name="email"  
value="">  
  
    </div>  
  
    <div>  
  
        <label>Website:</label>  
  
        <input type="text" id="website" name="website"  
value="">  
  
    </div>  
  
    <div>  
  
        <label for="numTickets"><abbr  
title="Number">No.</abbr> of Tickets (required) :</label>  
  
        <input type="text" id="numTickets"  
name="numTickets" value="">  
  
    </div>  
  
    <div class="submit">  
  
        <input type="submit" value="Submit">  
  
    </div>  
  
    </fieldset>  
  
/form>
```

 **CSS3**
(Day-9)

18-08-2025 to 26-08-2025

- Selectors, [Like](#) pseudo-classes [Like](#)
- Box model [Like](#)
- Positioning (absolute, relative, fixed) [Like](#)
- Display types: block, inline, inline-block [Like](#)
- Flexbox [Like](#)
- CSS Grid [Like](#)
- Responsive design (media queries, mobile-first) [Like](#)
- CSS variables [Like](#)
- Transitions & animations [Like](#)
- Preprocessors (SCSS/SASS - optional) [Like](#)
-

Project - (Day - 8,9,10) [Like](#)



JavaScript (ES6+) (Day - 30 Day) (Day - 40)

- 27-08-2025 to 3-10-2025**
- Data types, [Like](#) variables (var, let, const) [Like](#)
 - Operators [Like](#)
 - Conditionals (if, switch) [Like](#)

- Loops (for, while, for..in, for..of) [Like](#)
- Functions (declaration, expression, arrow) [Like](#)
- Objects [Like](#) & arrays [Like](#)
- Destructuring [Like](#)
- Spread & rest operators [Like](#)
- Template literals [Like](#)
- Classes [Like](#) & inheritance [Like](#)
- Promises [Like](#)
- Async/await [Like](#)
- Closures [Like](#)
- Hoisting [Like](#) & scope [Like](#)
- The `this` keyword [Like](#)
- ES modules (import/export) [Like](#)
- Error handling (try...catch) [Like](#)
- Event handling [Like](#)
- DOM manipulation [Like](#)
- Jquery [Like](#)

Project - (Day - 41-46)(Day - 6) [Like](#)

Version Control (Git & GitHub) [Like](#) (Day - 4)

- git init, clone, add, commit, push, pull
 - Branching & merging
 - Resolving merge conflicts
 - Pull requests & code reviews
 - Rebase
 - GitHub Actions (CI/CD)
-

□ 2 React.js (Frontend Library) (Day - 60)

4-10-2025 to 20-11-2025

□ React Basics [Like](#)

- JSX [Like](#)
- Components (functional + class) [Like](#)
- Props [Like](#)
- State [Like](#) [Like](#) [Like](#)
- Handling events [Like](#)
- Lists & keys [Like](#)
- Conditional rendering [Like](#)

□ React Advanced

- Hooks (useState [Like](#), useEffect [Like](#), useRef [Like](#), useContext [Like](#), useReducer [Like](#), useMemo [Like](#), useCallback [Like](#))
 - Context API [Like](#)
 - Error boundaries [Like](#)
 - Code splitting (React.lazy, Suspense) [Like](#)
 - Portals [Like](#)
 - Fragments [Like](#)
-

□ React Router [Like](#)

- Basic & nested routes
 - Dynamic routes (params)
 - Redirects
 - Route guards
-

⌚ State Management

- Redux (with Redux Toolkit) [Like](#)
- Thunks or Saga [Like](#)

- Zustand [Like](#)/ Jotai [Like](#)/ Recoil (lightweight alternatives)
-



Styling in React

- CSS Modules [Like](#)
- Styled Components [Like](#)
- Emotion
- Tailwind CSS [Like](#)
- Material-UI [Like](#) / Ant Design [Like](#)
-

Project -(Day - 10) 1. [Like](#) 2. [Like](#) 3.[Like](#) 4. [Like](#) 5. [Like](#)

□ React + TypeScript (2026+ standard) [Like](#)

- Typing props & state
 - Generics
 - Custom hooks with types
 - Event typing
-

□ 3 Node.js + Express.js (Backend) [Like](#)

21-12-2025 to 21-1-2025

Node.js

- Core modules (fs, path, http, os)
 - NPM modules
 - Event loop, callbacks
 - Streams, buffers
 - Process & env variables
 - File I/O
-

Express.js

- Routing
 - Middleware (custom + third-party)
 - Query params & URL params
 - Error handling
 - Serving static files
 - CORS
-

Authentication & Authorization

- JWT (JSON Web Token) [Like](#)

- OAuth (Google, GitHub) [Like](#)
 - bcrypt (password hashing) [Like](#)
 - Role-based access control [Like](#)
-

□ **Security**

- Helmet [Like](#)
 - Rate limiting [Like](#)
 - Input validation & sanitization [Like](#)
 - CSRF protection [Like](#)
-

□ **Testing** [Like](#)

- Unit testing (Mocha, Chai, Jest)
 - API testing (Supertest)
-

□ **4 MongoDB + Mongoose (Database)** [Like](#) (Day - 159 - 189)(Day - 20) **22-1-2025 to 22-2-2025**

□ **MongoDB**

- CRUD operations [Like](#)

- Data modeling [Like](#)
 - Indexes
 - Aggregation pipeline
 - Transactions (multi-document)
 - Embedding vs referencing
 - Relationships
-

□ **Mongoose**

- Schemas
 - Models
 - Virtuals
 - Middleware (pre/post hooks)
 - Population (populate)
 - Validation
-

❖ **MongoDB Atlas**

- Cluster creation
- Connection & security

- Backups
-

□ 5 Full-stack Integration (MERN together)

- Connecting React frontend with Express backend [Like](#)
- API consumption (fetch, axios) [Like](#)
- CORS handling [Like](#)
- Managing env variables (dotenv)
- Error handling across full stack [Like](#)
- File upload (Multer[Like](#) + Cloudinary [Like](#) / S3 [Like](#))
- Pagination [Like](#)
- Search & filters

Project 21-12-2025 to 20-1-2026

□ 6 DevOps & Deployment



Deployment **23-02-2026 to 23-03-2026**

- Build React apps (npm run build) [Like](#)
 - Serve React + Express [Like](#)
 - Vercel [Like](#), Netlify [Like](#), Render [Like](#), Heroku [Like](#)
 - Docker containerization [Like](#)
 - NGINX [Like](#) / Apache reverse proxy [Like](#)
 - GitHub Hosting [Like](#)
-

□ CI/CD

- GitHub Actions [Like](#)
 - Basic pipelines [Like](#)
-
- Linting& test automation
-



Monitoring

- Winston [Like](#)/ Morgan logging [Like](#)
 - Sentry [Like](#)
-

□ 7 Advanced Topics (2026+)

- **Next.js** (SSR [Like](#), static [Like](#) site generation) [Like](#)
 - **GraphQL + Apollo** [Like](#)
 - **WebSockets / Socket.io** (real-time apps) [Like](#)
 - **Microservices (Node.js + Docker)** [Like](#)
 - **Event-driven architecture (Kafka / RabbitMQ)** [Like](#)
 - **Serverless (AWS Lambda [Like](#), Vercel functions)**
 - **Progressive Web Apps (PWA)** [Like](#)
 - **AI/ML integration (OpenAI API, LangChain)** [Like](#)
 - **WebAssembly (Wasm basics)** [Like](#)
-

⚡ Bonus Skills

- Postman [Link](#) / Insomnia for API testing
 - Figma for UI prototyping [Figama](#) [Like](#)
 - SEO optimization for React [Like](#)
 - Webpack / Babel [Like](#) [Like](#)
 - Lighthouse performance checks [Like](#)
-

🛠 **MERN Project Ideas** **24-03-2026 to 24-04-2026**

- Blog CMS 1. [Like](#) 2. [Like](#) 3. [Like](#)
 - E-commerce store 1. [Like](#) 2. [Like](#) 3. [Like](#)
 - Social media clone 1. [Like](#) 2. [Like](#) 3. [Like](#) 4. [Like](#)
 - Real-time chat app 1. [Like](#) 2. [Like](#)
 - Project management dashboard 1. [Like](#) 2. [Like](#)
 - Video streaming app 1. [Like](#) 2.
-

□ Roadmap Summary Timeline

Month Focus

1-2 HTML, CSS, JS, Git

3-4 React (Core + Hooks + Router)

5-6 Node.js + Express

7 MongoDB + Mongoose

8 Full-stack integration

9 Build 1-2 full-stack projects

10-12 Advanced: TypeScript, Next.js, GraphQL, deployment

● Important topic

● OOPS [Like](#)(Day-15) **25-04-2026 to 6-05-2026**

● DSA [Like](#) **(2 Mathe)(9 - 10)**

□ Array [Like](#) [Like](#)

- Kadane's Algorithm (Maximum Subarray)
- Two Pointer Technique

- Sliding Window
- Merge Intervals
- Prefix Sum
- Rearrangement / Rotation
- Dutch National Flag Problem

String Like

-
- Palindrome check and longest palindrome
- Anagram, Substrings, Permutations
- KMP Algorithm (Pattern Matching)
- Z-Algorithm
- Rabin Karp Algorithm
- Trie (Prefix Tree)

Linked List Like

-
- Reverse a Linked List (Iterative and Recursive)
- Detect Loop (Floyd's Cycle Detection)
- Merge Two Sorted Linked Lists
- Intersection Point of Two Linked Lists
- LRU Cache (using DLL + HashMap)

◆ Stack Like

- Valid Parentheses
- Next Greater Element
- Stock Span Problem
- Min Stack
- Largest Rectangle in Histogram

◆ Queue Like / Deque Like

- Implement Stack/Queue using Queue/Stack
- Sliding Window Maximum (Monotonic Queue)
- Circular Queue
- LRU Cache

◆ Hashing Like

- Count frequency
- Longest Subarray with 0 sum
- Subarrays with given sum
- Detecting duplicates
- HashMap & HashSet based problems

◆ Recursion Like & Backtracking Like

- N-Queens Problem

- Rat in a Maze
- Sudoku Solver
- Subset/Combination Sum
- Permutations

• **Binary Tree Like / Binary Search Tree Like**

- Traversals (Inorder, Preorder, Postorder)
- Lowest Common Ancestor (LCA)
- Height/Depth of Tree
- Diameter of Binary Tree
- BST Operations (Insert/Delete/Search)
- Balanced BST (AVL/Red Black – theory)

• **Heap Like / Priority Queue Like**

- Kth Largest/Smallest Element
- Merge K Sorted Lists/Arrays
- Top K Frequent Elements
- Median in a Data Stream

• **Binary Search Like**

- Binary Search on Sorted Array
- Binary Search on Answer
- Search in Rotated Sorted Array

- Peak Element

◆ Greedy AlgorithmLike

- Activity Selection Problem
- Huffman Encoding
- Fractional Knapsack
- Job Sequencing
- Gas Station Problem

◆ Dynamic Programming (DP)Like

- 0/1 Knapsack
- Longest Common Subsequence (LCS)
- Longest Increasing Subsequence (LIS)
- Matrix Chain Multiplication
- DP on Trees / Grids
- DP with Bitmasking

◆ GraphLike

- BFS / DFS
- Detect Cycle (Undirected & Directed Graph)
- Topological Sort
- Dijkstra's Algorithm
- Bellman-Ford Algorithm

- Floyd-Warshall
- Union-Find (Disjoint Set Union - DSU)
- MST (Prim's & Kruskal's Algorithm)

• **Trie Like**

- Insert, Search, Delete
- Word Break Problem
- Autocomplete System
- Longest Prefix Matching

• **Bit Manipulation Like**

- Count Set Bits
- Power of 2
- XOR operations
- Single Number
- Subsets using Bits

• **Sliding Window Like**

- Maximum Sum Subarray of size k
- Minimum Window Substring
- Longest Substring Without Repeating Characters

 **Bonus Topics for Product-Based Companies**

- **Segment Tree / Fenwick Tree (Binary Indexed Tree)**
 - **Topological Sort**
 - **Heavy-Light Decomposition**
 - **Mo's Algorithm**
 - **Game Theory Basics**
-

If you'd like, I can give:

- Topic-wise problem list (Easy → Medium → Hard)
- Practice websites (LeetCode, GeeksforGeeks, etc.)
- Roadmap for completing all topics in 2–3 months

Would you like that?

- JavaScript Interview Questions
 1. [Like Like Like](#)
 2. [_____](#)
 3. [_____](#)

4. [Like](#)

● React js

1. [Like](#)

2.

● Node js

1. [Like](#)

● Mogodb

1. [Like](#)