MERN ROADMAP

120 concepts to become a MERN stack developer

8 areas, 15 concepts each. Every concept explained in one line, the order to study each area, and two practice questions to test yourself.

How to use it: one area at a time, tick each concept only when you can explain it out loud in under a minute, then answer the two practice questions without notes.

@chalkesahil · chalkesahil.ai

AREA 01 / 08

HTML, CSS and JavaScript

Study order: Plain JavaScript first; React is just JavaScript underneath.

01	**Semantic HTML**	Meaningful tags.
02	**Flexbox and Grid**	Modern layouts.
03	**Responsive design**	Works on every phone.
04	**Tailwind CSS**	Utility classes for speed.
05	**let and const**	Modern variables.
06	**Arrow functions**	Short function syntax.
07	**Array methods**	map, filter, reduce.
08	**Destructuring**	Pull values out cleanly.
09	**Spread and rest**	Copy and merge.
10	**Promises**	Work that finishes later.
11	**async and await**	Readable async code.
12	**fetch**	Call APIs.
13	**Closures**	Functions remembering scope.
14	**Event loop**	Why async code runs when it does.
15	**ES modules**	import and export.

Practice

1. Build a responsive product card grid.
2. Fetch users from an API and show them in a list.

AREA 02 / 08

React

Study order: Build small apps until components feel natural.

01	**Components**	Reusable UI pieces.
02	**JSX**	HTML inside JavaScript.
03	**Props**	Pass data down.
04	**useState**	Data that changes.
05	**Events**	Clicks, inputs and forms.
06	**Lists and keys**	Render arrays correctly.
07	**Conditional rendering**	Show or hide UI.
08	**useEffect**	Side effects like fetching.
09	**Controlled forms**	Inputs tied to state.
10	**Lifting state**	Share state between siblings.
11	**React Router**	Pages and links.
12	**Fetching data**	Loading and error states.
13	**Component structure**	Folders that scale.
14	**Vite**	Fast React setup.
15	**DevTools**	Inspect components and state.

Practice

1. Build a todo app with filters.
2. Build a search page that calls an API as you type.

AREA 03 / 08

React advanced

Study order: These separate a beginner from someone who can join a team.

01	**Context API**	Avoid prop drilling.
02	**Custom hooks**	Reuse logic.
03	**useReducer**	Complex state transitions.
04	**Redux Toolkit**	Global state for big apps.
05	**React Query**	Server data with caching.
06	**Memoization**	useMemo and useCallback.
07	**Re-renders**	Why and when they happen.
08	**Code splitting**	Load pages on demand.
09	**Error boundaries**	Catch UI crashes.
10	**Forms libraries**	React Hook Form.
11	**Validation**	Zod or Yup schemas.
12	**Protected routes**	Pages only for logged-in users.
13	**Testing**	React Testing Library.
14	**Accessibility**	Keyboard and screen reader basics.
15	**Next.js basics**	Server rendering and routes.

Practice

1. Add a cart shared across pages without prop drilling.
2. Your list re-renders on every keystroke. Fix it.

AREA 04 / 08

Node.js

Study order: Understand Node itself before frameworks hide it.

01	**What Node is**	JavaScript on the server.
02	**npm**	Install and manage packages.
03	**package.json**	Scripts and dependencies.
04	**Modules**	require vs import.
05	**File system**	Read and write files.
06	**Event emitter**	Node's event pattern.
07	**Streams**	Handle big data in chunks.
08	**Async patterns**	Callbacks to async/await.
09	**Environment variables**	Config outside code.
10	**Error handling**	Uncaught errors and crashes.
11	**Path and OS modules**	Built-in helpers.
12	**Child processes**	Run other programs.
13	**Event loop phases**	How Node handles many requests.
14	**nodemon**	Restart on save.
15	**Debugging**	Breakpoints in VS Code.

Practice

1. Read a large CSV file without loading it all into memory.
2. Why can one Node server handle many requests?

AREA 05 / 08

Express

Study order: Build every route of a real API by hand.

01	**Express setup**	A server in minutes.
02	**Routing**	GET, POST, PUT, DELETE.
03	**Route params**	/users/:id.
04	**Query strings**	Filters and pagination.
05	**Request body**	JSON from the client.
06	**Middleware**	Code before each route.
07	**Error middleware**	One place for errors.
08	**Validation**	Reject bad input.
09	**Folder structure**	Routes, controllers, services.
10	**CORS**	Let React call your API.
11	**File uploads**	Multer for images.
12	**Rate limiting**	Stop abuse.
13	**Logging**	Morgan or pino.
14	**REST conventions**	Clean URLs and status codes.
15	**Postman**	Test every route.

Practice

1. Build a REST API for blog posts with comments.
2. Add pagination and search to a list endpoint.

AREA 06 / 08

MongoDB

Study order: Model data around how your app reads it.

01	**Documents and collections**	MongoDB's building blocks.
02	**CRUD operations**	Create, read, update, delete.
03	**Mongoose**	Schemas and models.
04	**Schema validation**	Required fields and types.
05	**Relationships**	Embed vs reference.
06	**populate**	Join referenced documents.
07	**Indexes**	Faster queries.
08	**Query operators**	$gt, $in, $regex.
09	**Aggregation pipeline**	Group and transform data.
10	**Pagination**	skip and limit, or cursors.
11	**Transactions**	Multi-document safety.
12	**MongoDB Atlas**	Free cloud database.
13	**Data modeling**	Design for your reads.
14	**Backups**	Never lose user data.
15	**SQL vs MongoDB**	When each fits.

Practice

1. Model users, posts and comments for a social app.
2. Find the top 5 authors by number of posts.

AREA 07 / 08

Auth and deployment

Study order: Add login to your app and ship it to a live URL.

01	**Password hashing**	bcrypt.
02	**JWT**	Signed tokens for login.
03	**Cookies**	httpOnly and secure.
04	**Refresh tokens**	Stay logged in safely.
05	**Google login**	OAuth.
06	**Roles**	Admin vs user.
07	**Protecting routes**	Backend and frontend.
08	**Environment config**	Secrets out of code.
09	**Git and GitHub**	Branches and pull requests.
10	**Deploy React**	Vercel or Netlify.
11	**Deploy Express**	Render or Railway.
12	**Connect frontend and backend**	URLs and CORS in production.
13	**Custom domain**	Point DNS to your app.
14	**Docker basics**	Same app everywhere.
15	**CI**	Tests on every push.

Practice

1. Walk through a JWT login from form to database.
2. Deploy a MERN app with a custom domain.

AREA 08 / 08

Projects and interviews

Study order: Three real MERN apps make a stronger resume than ten clones.

01	**Project 1: e-commerce**	Cart, payments test mode, orders.
02	**Project 2: real-time chat**	Socket.io.
03	**Project 3: dashboard**	Charts and filters.
04	**README**	Screenshots and live link.
05	**Portfolio site**	All projects in one place.
06	**JavaScript interview questions**	Closures, this, event loop.
07	**React interview questions**	Hooks and re-renders.
08	**Node interview questions**	Event loop and streams.
09	**Machine coding**	Build a UI feature in an hour.
10	**DSA basics**	Arrays, strings, hashing.
11	**System design basics**	Caching and load balancing.
12	**Resume**	Impact with numbers.
13	**LinkedIn**	Share what you build.
14	**Mock interviews**	Practise out loud.
15	**AI tools**	Speed up, still understand every line.

Practice

1. Build a star rating component in 30 minutes.
2. Explain how your chat app sends messages in real time.