

Vue.js Full Roadmap (Basic to Advanced)

■ Basics (Start Here!)

- What is Vue.js and how it works
- Vue Instance & createApp()
- The Virtual DOM
- Vue Template Syntax
- Data Binding (v-bind, v-model)
- Event Handling (v-on, @click)
- Conditional Rendering (v-if, v-else, v-show)
- List Rendering (v-for)
- Computed Properties
- Methods vs Computed vs Watchers
- Class & Style Binding
- Basic Form Handling

■ Intermediate Topics (Getting Real)

- Creating Components (.vue files)
- Props (one-way binding)
- Emitting Events (\$emit)
- Slots (default, named, scoped)
- Dynamic Components & keep-alive
- Lifecycle Hooks (mounted, updated, etc.)
- Refs (ref, shallowRef)
- Provide/Inject (dependency injection)
- Built-in Directives (v-if, v-for, v-model, etc.)
- Custom Directives (like v-focus, v-click-outside)
- Vue CLI & Vite for project scaffolding
- Single File Components (SFC)
- Hot Module Replacement (HMR)

■ Advanced Core Concepts (Big Brain Time)

- Composition API vs Options API
- setup() function
- reactive, ref, computed, watch
- toRefs, toRaw, shallowReactive

- onMounted, onUnmounted (lifecycle hooks)
- defineExpose, defineEmits, defineProps
- Composable Functions (e.g., useUser, useFetch)
- Dynamic & Async Components
- Teleport, Suspense, Lazy loading components
- Writing & Using Vue Plugins
- Dependency Injection with Plugins

■ Vue Router (SPA Routing)

- Installing & configuring vue-router
- Route Parameters & Query Strings
- Nested Routes, Dynamic Routing
- Navigation Guards (beforeEnter, beforeEach)
- Lazy loading routes
- Programmatic navigation (router.push())

■ Vuex / Pinia (State Management)

- Vuex: State, Getters, Mutations, Actions
- Vuex Modules & Devtools
- Pinia: Setup & Installation
- Defining stores with defineStore
- Reactive stores, Actions, Getters
- Plugin support & Devtools integration

■ Testing Vue Apps

- Unit testing with Vue Test Utils + Jest/Vitest
- Snapshot Testing
- Mocking Props, Events, Stores
- End-to-End testing (e.g., Cypress or Playwright)

■ Vue + Backend / Fullstack

- Axios / Fetch API for backend communication
- Handling Authentication (JWT, OAuth)
- CRUD operations with API
- File upload & form data
- Vue + Firebase, Vue + Node.js (Express), Vue + Laravel

■ Performance, Optimization, Deployment

- Code Splitting & Lazy Load Routes/Components
- Tree-shaking unused code
- Dev vs Prod Builds
- SSR (Server-Side Rendering with Nuxt)
- Static Site Generation (SSG)
- Caching strategies
- Vue DevTools performance tab

■ Ecosystem Tools & Integrations

- Vue + TypeScript
- Vue + TailwindCSS / Bootstrap
- Vue + GraphQL
- Vue + Vitest / Cypress
- Vue + Capacitor / Cordova (Mobile apps)
- Vuetify, Element Plus, PrimeVue, Quasar (UI Frameworks)
- Vue + WebSockets

■ Meta Frameworks

- Nuxt.js: SSR and SSG out of the box
- Auto-imports, file-based routing
- API Routes, Nuxt Auth, Nuxt Content
- Nuxt 3 + Vue 3 integration

■ Best Practices

- Component naming conventions
- Atomic Design in Vue
- Managing large apps
- Reusability with Composables
- Accessibility in Vue (a11y)
- SEO with Vue/Nuxt