Comprehensive Frontend Developer Roadmap: Basic to Advanced

1. HTML Fundamentals

- **Basic HTML Structure**
  - Document structure (`<!DOCTYPE>`, `<html>`, `<head>`, `<body>`)
  - Tags, attributes, and elements
  - Semantic HTML (`<header>`, `<nav>`, `<main>`, `<section>`, `<article>`, `<footer>`)
  - Block vs. inline elements
- **HTML Forms**
  - Form elements and attributes
  - Input types (text, email, number, checkbox, radio, etc.)
  - Form validation (required, pattern, min/max)
  - Fieldset and legend
- **Media Elements**
  - Images (`<img>` and attributes)
  - Audio and video elements
  - Responsive media
  - Figure and figcaption
- **Tables**
  - Basic table structure
  - Table headers, rows, and cells
  - Spanning rows and columns
  - Table accessibility
- **HTML5 APIs**
  - Local Storage and Session Storage
  - Geolocation API
  - Canvas basics
  - Drag and Drop API

2. CSS Essentials

- **CSS Fundamentals**
    - Selectors (element, class, ID, attribute)
    - Cascading and specificity
    - Box model (content, padding, border, margin)
    - Units (px, em, rem, %, vh/vw)
- **Typography**
    - Web fonts and font-face
    - Text properties (size, weight, style, alignment)
    - Line height and letter spacing
    - Web-safe fonts vs. custom fonts
- **Colors and Backgrounds**
    - Color formats (hex, RGB, HSL)
    - Background properties
    - Gradients
    - Opacity and transparency
- **Layout Basics**
    - Display property (block, inline, inline-block)
    - Positioning (static, relative, absolute, fixed, sticky)
    - Floats and clears
    - z-index and stacking context
- **Responsive Design Basics**
    - Media queries
    - Viewport meta tag
    - Mobile-first approach
    - Breakpoints

3. CSS Advanced

- **Flexbox**
    - Flex container properties
    - Flex item properties
    - Alignment and justification
    - Flex direction and wrapping
- **CSS Grid**

    - Grid container properties
    - Grid item placement
    - Grid template areas
    - Implicit vs. explicit grids

- **Animations and Transitions**

    - Transition properties
    - Keyframe animations
    - Animation timing functions
    - Performance considerations

- **CSS Preprocessors**

    - SASS/SCSS basics
    - Variables and mixins
    - Nesting and partials
    - Functions and operators

- **CSS Architecture**

    - BEM methodology
    - SMACSS
    - OOCSS
    - CSS Modules

- **Advanced Selectors**

    - Pseudo-classes (`:hover`, `:nth-child`, etc.)
    - Pseudo-elements (`::before`, `::after`)
    - Attribute selectors
    - Combinators (descendant, child, adjacent, general sibling)

4. JavaScript Fundamentals

- **JavaScript Syntax**

    - Variables and data types
    - Operators and expressions
    - Control flow (if/else, switch, loops)
    - Functions (declarations, expressions, arrow functions)

- **Working with Data**

    - Objects and properties
    - Arrays and array methods
    - String manipulation

    - JSON parsing and stringifying

- **DOM Manipulation**

    - Selecting DOM elements
    - Modifying element content
    - Creating and removing elements
    - Traversing the DOM

- **Events**

    - Event listeners
    - Event propagation (bubbling and capturing)
    - Event delegation
    - Common events (click, submit, load, etc.)

- **Scope and Closures**

    - Global vs. local scope
    - Lexical scope
    - Closures and practical uses
    - Execution context

5. JavaScript Advanced

- **Asynchronous JavaScript**

    - Callbacks
    - Promises
    - Async/await
    - Error handling with try/catch

- **ES6+ Features**

    - Template literals
    - Destructuring
    - Spread/rest operators
    - Default parameters
    - Optional chaining and nullish coalescing

- **Functional Programming Concepts**

    - Pure functions
    - Higher-order functions
    - Map, filter, reduce
    - Immutability principles

- **Object-Oriented Programming**

    - Prototypal inheritance
    - Constructor functions
    - ES6 Classes
    - Encapsulation, inheritance, polymorphism
- **JavaScript Modules**
    - CommonJS vs. ES Modules
    - Import and export syntax
    - Dynamic imports
    - Module bundling concepts
- **Web APIs**
    - Fetch API
    - History API
    - Intersection Observer
    - Web Workers

6. TypeScript Fundamentals

- **TypeScript Basics**
    - Static typing
    - Type annotations and interfaces
    - Union and intersection types
    - Type inference
- **Advanced Types**
    - Generics
    - Utility types
    - Type guards
    - Enums and literal types
- **TypeScript with React**
    - Typing props and state
    - Event handling with TypeScript
    - Generic components
    - Using .tsx files

7. React Fundamentals

- **React Core Concepts**

    - JSX syntax
    - Components (functional and class-based)
    - Props and prop types
    - State and lifecycle

- **Component Patterns**

    - Controlled vs. uncontrolled components
    - Lifting state up
    - Composition vs. inheritance
    - Children props

- **React Hooks Basics**

    - useState
    - useEffect
    - useContext
    - useRef

- **Conditional Rendering**

    - If/else in JSX
    - Ternary operators
    - Short-circuit evaluation
    - Switch statements

- **Lists and Keys**

    - Rendering arrays
    - Key prop importance
    - Array manipulation techniques
    - List performance

8. React Advanced

- **Advanced React Hooks**

    - useReducer
    - useCallback
    - useMemo
    - Custom hooks

- **Context API**

    - Creating context
    - Provider and Consumer
    - Context with hooks

    - Multiple contexts
- **React Router**
    - Route configuration
    - Dynamic routes
    - Nested routes
    - Protected routes
- **Advanced Component Patterns**
    - Higher-Order Components (HOCs)
    - Render props
    - Compound components
    - Custom hooks as patterns
- **Optimization Techniques**
    - Memoization (React.memo)
    - Virtual DOM understanding
    - Profiling with React DevTools
    - React.lazy and Suspense

9. State Management

- **Local State Management**
    - useState best practices
    - useReducer for complex state
    - Prop drilling solutions
    - State design patterns
- **Redux Fundamentals**
    - Actions and action creators
    - Reducers
    - Store configuration
    - Connect API vs. hooks API
- **Redux Middleware**
    - Redux Thunk
    - Redux Saga basics
    - Async actions
    - Middleware patterns
- **Redux Toolkit**

    - createSlice
    - configureStore
    - createAsyncThunk
    - RTK Query
- **Alternative State Management**
    - Context + useReducer
    - Zustand
    - Jotai/Recoil
    - MobX basics
- **Server State Management**
    - React Query fundamentals
    - SWR
    - Caching strategies
    - Optimistic updates

10. Forms in React

- **Form Basics**
    - Controlled forms
    - Form submission
    - Input validation
    - Multi-step forms
- **Form Libraries**
    - React Hook Form
    - Formik
    - Yup validation
    - Zod schema validation
- **Advanced Form Techniques**
    - Dynamic form fields
    - Form arrays
    - Field-level validation
    - Form persistence

11. Styling in React

- **CSS-in-JS**

    - Styled-components
    - Emotion
    - CSS Modules with React
    - Style props
- **CSS Frameworks with React**
    - Tailwind CSS integration
    - Bootstrap with React
    - Material UI
    - Chakra UI
- **Theming**
    - ThemeProvider pattern
    - Dark/light mode toggle
    - Dynamic theming
    - Design system implementation

12. Testing

- **Testing Fundamentals**
    - Unit vs. integration vs. E2E testing
    - Test-driven development
    - Testing pyramids
    - Writing testable code
- **React Testing Tools**
    - Jest basics
    - React Testing Library
    - Component testing patterns
    - Mocking in tests
- **Advanced Testing**
    - Testing hooks
    - Testing Redux stores
    - Testing asynchronous code
    - Snapshot testing
- **End-to-End Testing**
    - Cypress basics
    - Playwright
    - Testing user flows

- Visual regression testing

13. Performance Optimization

- **React Performance**
  - Profiling React applications
  - Identifying performance bottlenecks
  - Code splitting strategies
  - Bundle size optimization
- **Runtime Performance**
  - Avoiding unnecessary renders
  - useCallback and useMemo usage
  - Virtual list techniques
  - React.lazy and Suspense
- **Network Performance**
  - HTTP/2 and HTTP/3 considerations
  - Code splitting
  - Resource prioritization
  - Caching strategies
- **Web Vitals**
  - Core Web Vitals metrics
  - Measuring performance (Lighthouse, WebPageTest)
  - First Contentful Paint optimization
  - Cumulative Layout Shift reduction

14. Advanced React Patterns

- **Data Fetching Patterns**
  - Data fetching with hooks
  - SWR/React Query patterns
  - Loading and error states
  - Pagination and infinite scroll
- **Architectural Patterns**
  - Container vs. presentation components
  - Feature-based organization
  - Atomic design in React

    - Micro-frontends

- **Authentication Patterns**

    - JWT auth flow
    - OAuth integration
    - Protected routes
    - Role-based access control

- **Internationalization (i18n)**

    - React Intl
    - i18next
    - Dynamic language switching
    - Right-to-left support

15. Next.js Fundamentals

- **Next.js Basics**

    - Pages and routing
    - Static site generation (SSG)
    - Server-side rendering (SSR)
    - API routes

- **Data Fetching**

    - getStaticProps
    - getServerSideProps
    - Incremental Static Regeneration
    - SWR with Next.js

- **Next.js Features**

    - Image optimization
    - Font optimization
    - Layouts and nested layouts
    - Middleware

- **Next.js Deployment**

    - Vercel deployment
    - Static exports
    - Environment variables
    - Serverless functions

16. Build Tools and Workflows

- **Package Managers**
    - npm
    - yarn
    - pnpm
    - Package.json configuration
- **Bundlers**
    - Webpack basics
    - Vite
    - Rollup
    - esbuild
- **Task Runners**
    - npm scripts
    - Build pipelines
    - Watch mode
    - Dev server configuration
- **Transpilers and Polyfills**
    - Babel configuration
    - TypeScript compilation
    - Browserslist
    - Core-js polyfills

17. Version Control and Collaboration

- **Git Fundamentals**
    - Basic commands
    - Branching strategies
    - Merge vs. rebase
    - Resolving conflicts
- **GitHub/GitLab Workflows**
    - Pull/merge requests
    - Code reviews
    - GitHub Actions basics
    - Issue management
- **Code Quality Tools**

    - ESLint
    - Prettier
    - Husky pre-commit hooks
    - TypeScript strict mode

18. Accessibility (a11y)

- **Accessibility Fundamentals**
    - WCAG guidelines
    - Semantic HTML for accessibility
    - Keyboard navigation
    - Screen reader testing
- **React Accessibility**
    - Accessible forms
    - Focus management
    - ARIA attributes in React
    - Accessible routing
- **Testing for Accessibility**
    - Automated a11y testing
    - Manual testing procedures
    - axe-core integration
    - Accessibility audits

19. Security

- **Frontend Security Basics**
    - XSS prevention
    - CSRF protection
    - Content Security Policy
    - Secure authentication practices
- **API Security**
    - Authorization headers
    - CORS understanding
    - JWT handling
    - Secure storage (cookies vs. localStorage)
- **Security Testing**

    - Security audits
    - Dependency scanning
    - OWASP Top 10 for frontend
    - Security headers

20. Advanced Concepts

- **Server Components**
    - React Server Components
    - Streaming SSR
    - Hydration strategies
    - Islands architecture
- **Web Assembly**
    - WASM basics
    - Integrating WASM with React
    - Performance considerations
    - Use cases for frontend
- **Progressive Web Apps**
    - Service workers
    - Web App Manifest
    - Offline capabilities
    - Push notifications
- **GraphQL with React**
    - Apollo Client
    - Query and mutation hooks
    - Cache management
    - State management with Apollo
- **Micro-Frontends**
    - Module Federation
    - Single-spa
    - Architecture considerations
    - Communication between micro-frontends

21. Deployment and DevOps

- **CI/CD for Frontend**

    - Continuous integration setup
    - Deployment strategies
    - Environment management
    - Automated testing in CI

- **Containerization**

    - Docker basics for frontend
    - Multi-stage builds
    - Docker Compose for local development
    - Container orchestration concepts

- **Monitoring and Logging**

    - Error tracking (Sentry)
    - Performance monitoring
    - Analytics integration
    - Logging best practices

- **Infrastructure as Code**

    - Netlify/Vercel configuration
    - AWS Amplify
    - Terraform basics
    - GitHub Actions workflows

22. Professional Skills

- **Technical Documentation**

    - README documentation
    - JSDoc comments
    - Storybook for component documentation
    - Architecture decision records

- **Code Review**

    - Code review best practices
    - Giving constructive feedback
    - Addressing technical debt
    - Review automation

- **Technical Decision Making**

    - Evaluating libraries and frameworks
    - Technical spike methodology
    - ROI analysis for technical decisions

    - Tech radar concept

- **Team Collaboration**
    - Working with designers (design systems)
    - Backend API collaboration
    - Cross-functional communication
    - Mentoring junior developers