🎯 Complete Intermediate JavaScript Roadmap with Projects

PHASE 1: ES6+ Modern JavaScript (2-3 weeks)

📚 Topics to Learn:

Week 1: Core ES6 Syntax

- `let` and `const` (block scoping)
- Arrow functions (`=>`)
- Template literals (backticks)
- Destructuring (arrays & objects)
- Default parameters
- Spread operator (`...`)
- Rest parameters

Week 2: Advanced ES6+

- Classes and inheritance
- Modules (import/export)
- Optional chaining (`?.`)
- Nullish coalescing (`??`)
- Maps and Sets
- WeakMap and WeakSet
- Symbols

Week 3: Iterators & Advanced

- for...of loops
- Iterators and Generators
- Array methods (map, filter, reduce, find, some, every)
- String methods (includes, startsWith, endsWith)
- Object methods (Object.keys, values, entries, assign, freeze)

🛠️ Projects:

Project 1: Task Manager with ES6 Classes

- Use classes to create Task objects
- Implement modules (separate files for Task, TaskList, UI)
- Use Maps to store tasks with unique IDs
- Practice destructuring and spread operators
- localStorage for persistence

Project 2: Movie Search App

- Fetch data from OMDB API
- Use template literals for dynamic HTML
- Destructure API responses
- Use array methods to filter/sort results
- Optional chaining for nested data

Project 3: Shopping Cart

- ES6 classes for Product and Cart
- Use reduce() to calculate totals
- Spread operator for state updates
- Modules for cart logic, UI, and storage
- Sets to track unique items

PHASE 2: Asynchronous JavaScript (3-4 weeks)

📚 Topics to Learn:

Week 4: Callbacks & Promises

- Synchronous vs Asynchronous code
- Callbacks and callback hell
- Promises (then, catch, finally)
- Creating promises
- Promise chaining
- Error handling in promises

Week 5: Async/Await & Event Loop

- async/await syntax
- Error handling with try/catch
- Event loop explained
- Call stack, callback queue, microtasks
- setTimeout and setInterval
- Understanding execution order

Week 6: Advanced Async Patterns

- Promise.all() - parallel execution
- Promise.race() - first to resolve
- Promise.allSettled() - wait for all
- Promise.any() - first successful
- Sequential vs parallel async operations
- Async iterators

Week 7: Fetch API & Real APIs

- Fetch API basics
- GET, POST, PUT, DELETE requests
- Headers and request options
- Handling JSON
- CORS understanding
- HTTP status codes
- Error handling in fetch

🛠️ Projects:

Project 4: Weather Dashboard

- Fetch weather data from OpenWeatherMap API
- Use async/await throughout
- Handle loading states
- Error handling for failed requests
- Show 5-day forecast
- Geolocation API integration

Project 5: GitHub User Finder

- Search GitHub users via API
- Display repositories
- Use Promise.all() to fetch multiple endpoints
- Loading spinners during fetch
- Error handling for user not found
- Pagination for repositories

Project 6: Async Image Gallery

- Load images from API (Unsplash or Pixabay)
- Implement infinite scroll

- Lazy loading with Intersection Observer
- Download images functionality
- Search with debouncing
- Handle rate limiting

Project 7: Currency Converter

- Real-time exchange rates API
- Convert between multiple currencies
- Use async/await for API calls
- Cache results for 1 hour
- Handle offline scenarios
- Historical rate charts (bonus)

PHASE 3: Advanced Functions & Closures (2 weeks)

📚 Topics to Learn:

Week 8: Closures & Scope

- Lexical scope
- Closures explained
- Practical use cases
- Module pattern
- Private variables
- IIFE (Immediately Invoked Function Expressions)

Week 9: Function Techniques

- Higher-order functions
- Function currying
- Function composition

- Recursion (base case, recursive case)
- Memoization
- Debounce and throttle
- `this` keyword (context)
- call, apply, bind methods

🛠️ Projects:

Project 8: Autocomplete Search with Debounce

- Implement debounce function from scratch
- Search suggestions from API
- Highlight matching text
- Keyboard navigation (up/down arrows)
- Close suggestions on outside click

Project 9: Calculator with Advanced Operations

- Use closures for calculator state
- Chain operations
- Memory functions (M+, M-, MR, MC)
- History of calculations
- Keyboard support
- Scientific calculator (bonus)

Project 10: Infinite Scroll Blog

- Throttle scroll events
- Fetch posts from API
- Intersection Observer for loading
- Memoize expensive calculations
- Recursive comments rendering

PHASE 4: Object-Oriented JavaScript (2 weeks)

📚 Topics to Learn:

Week 10: Prototypes

- Prototype chain
- `__proto__` vs `prototype`
- Constructor functions
- Object.create()
- instanceof operator
- Prototypal inheritance

Week 11: Modern OOP

- ES6 Classes deep dive
- Inheritance with extends
- super keyword
- Static methods
- Getters and setters
- Private fields (#)
- Composition vs inheritance

🛠️ Projects:

Project 11: Library Management System

- Book class with inheritance (EBook, AudioBook)
- Member class with borrowing limits
- Library class managing collection
- Use Maps for fast lookups
- Implement search, sort, filter

- Late fee calculation with getters

Project 12: Game: RPG Character System

- Character base class
- Warrior, Mage, Archer subclasses
- Inventory system
- Battle mechanics with polymorphism
- Level up system
- Save/load with localStorage

PHASE 5: Browser APIs & DOM Mastery (2-3 weeks)

📚 Topics to Learn:

Week 12: DOM & Events

- DOM manipulation (createElement, appendChild, etc.)
- Event listeners
- Event bubbling and capturing
- Event delegation
- preventDefault and stopPropagation
- Custom events

Week 13: Storage & Modern APIs

- localStorage and sessionStorage
- IndexedDB basics
- Cookies
- Service Workers (introduction)
- Web Workers
- Geolocation API

- Notification API

Week 14: Advanced Browser APIs

- Canvas API basics
- Drag and Drop API
- History API (pushState, replaceState)
- Intersection Observer
- MutationObserver
- Clipboard API
- File API

🛠️ Projects:

Project 13: Note Taking App (Advanced)

- Rich text editing
- Categories with colors
- IndexedDB for storage
- Export notes to JSON
- Search and filter
- Drag and drop to reorder
- Dark mode with localStorage

Project 14: Drawing App

- Canvas API for drawing
- Different tools (pen, eraser, shapes)
- Color picker
- Undo/redo functionality
- Save drawing as image
- Load images onto canvas

Project 15: Real-time Location Tracker

- Geolocation API
- Map integration (Leaflet.js)
- Track movement path
- Calculate distance traveled
- Speed calculation
- Save locations to IndexedDB

Project 16: Drag & Drop Kanban Board

- Drag and Drop API
- Multiple columns (Todo, In Progress, Done)
- Add/edit/delete tasks
- localStorage persistence
- Search and filter tasks
- Due dates with notifications

PHASE 6: Error Handling & Debugging (1 week)

📚 Topics to Learn:

Week 15: Error Management

- try/catch/finally blocks
- throw keyword
- Error types (SyntaxError, TypeError, etc.)
- Custom error classes
- Error handling in async code
- Debugging techniques
- Chrome DevTools mastery

- Source maps

🛠️ Projects:

Project 17: Form Validator with Custom Errors

- Email, phone, password validation
- Custom error messages
- Real-time validation
- RegEx patterns
- Try/catch for async validation
- API validation (check if email exists)

PHASE 7: JavaScript Engine & Performance (1-2 weeks)

📚 Topics to Learn:

Week 16: Under the Hood

- Execution context
- Call stack
- Memory heap
- Hoisting explained
- Scope chain
- Event loop deep dive
- Microtasks vs macrotasks

Week 17: Performance

- Time complexity basics
- Memory leaks
- Garbage collection

- Performance.now()
- RequestAnimationFrame
- Code optimization techniques
- Lazy loading
- Tree shaking concepts

🛠️ Projects:

Project 18: Performance Dashboard

- Measure page load time
- Track API response times
- Memory usage monitoring
- Visualize performance metrics
- Lazy load images
- Optimize heavy calculations with Web Workers

PHASE 8: Data Structures in JavaScript (2 weeks)

📚 Topics to Learn:

Week 18: Built-in Structures

- Arrays (advanced methods)
- Objects (deep dive)
- Maps vs Objects
- Sets vs Arrays
- WeakMap and WeakSet
- Typed Arrays

Week 19: Custom Structures

- Stack implementation
- Queue implementation
- Linked List basics
- Hash table concepts
- Tree basics (binary tree)
- Graph basics

🛠️ Projects:

Project 19: Data Structure Visualizer

- Visualize stack operations (push, pop)
- Queue operations (enqueue, dequeue)
- Linked list traversal
- Tree traversal animations
- Interactive controls

Project 20: LRU Cache Implementation

- Implement Least Recently Used cache
- Use Map for O(1) operations
- Set size limit
- Remove least used items
- Test with real API caching

PHASE 9: Functional Programming (1-2 weeks)

📚 Topics to Learn:

Week 20: FP Concepts

- Pure functions
- Immutability

- Side effects
- First-class functions
- Function composition
- Pipe and compose
- Partial application
- Point-free style

Week 21: FP in Practice

- Array methods (map, filter, reduce deep dive)
- Avoiding mutations
- Immutable data updates
- Ramda or Lodash/fp (optional)

🛠️ Projects:

Project 21: Data Transformation Pipeline

- Process large datasets
- Chain transformations (map, filter, reduce)
- No mutations allowed
- Compose reusable functions
- Performance comparison with imperative style

Project 22: Shopping Cart (FP Style)

- Immutable state management
- Pure functions for all operations
- Compose discount calculators
- No side effects
- State history with undo/redo

PHASE 10: APIs & HTTP Mastery (1-2 weeks)

📚 Topics to Learn:

Week 22: REST APIs

- RESTful principles
- HTTP methods (GET, POST, PUT, PATCH, DELETE)
- Status codes (200, 201, 400, 401, 404, 500)
- Headers (Content-Type, Authorization)
- Query parameters
- Request body formats
- CORS in detail
- Authentication (JWT, OAuth basics)

Week 23: Advanced API Handling

- Rate limiting
- Retry logic
- Request cancellation (AbortController)
- Axios library
- API error handling patterns
- Pagination strategies
- Caching strategies

🛠️ Projects:

Project 23: REST API Client Library

- Create reusable API client
- Handle authentication
- Automatic retry on failure

- Request/response interceptors
- Cancel pending requests
- Error handling wrapper

Project 24: Social Media Dashboard

- Integrate multiple APIs (Twitter, GitHub, etc.)
- Display combined feed
- Handle rate limits
- Implement infinite scroll
- Real-time updates (polling)
- Error recovery

PHASE 11: Regular Expressions (1 week)

📚 Topics to Learn:

Week 24: RegEx

- Pattern syntax
- Character classes
- Quantifiers
- Anchors (^, $)
- Groups and capturing
- Lookahead and lookbehind
- Flags (g, i, m, s)
- Common patterns (email, phone, URL)
- test() and match() methods
- replace() with patterns

🛠️ Projects:

Project 25: Advanced Form Validator

- Email validation
- Phone number (multiple formats)
- Password strength checker
- Credit card validation
- URL validator
- Date format validator
- Highlight invalid fields

Project 26: Text Analyzer Tool

- Word count
- Find and highlight patterns
- Email extractor
- URL extractor
- Phone number finder
- Replace patterns
- Regex tester/debugger

PHASE 12: Testing & Quality (1-2 weeks)

📚 Topics to Learn:

Week 25: Testing Basics

- Why testing matters
- Unit testing concepts
- Jest basics
- Writing test cases
- Assertions

- Mocking
- Code coverage

Week 26: Best Practices

- Clean code principles
- DRY, KISS, YAGNI
- Code comments
- Naming conventions
- Error handling best practices
- Security basics (XSS, CSRF)

🛠️ Projects:

Project 27: Tested Utility Library

- Create utility functions
- Write comprehensive tests
- Achieve 100% coverage
- Document with JSDoc
- Publish to npm (optional)

CAPSTONE PROJECTS (Final 2-4 weeks)

🏆 Build One Large Project Combining Everything:

Option 1: Full-Featured Todo App

- User authentication (localStorage simulation)
- Projects and sub-tasks
- Tags and priorities
- Due dates with notifications
- Drag and drop

- Search and filters
- Dark mode
- Export/import data
- Offline support
- Statistics dashboard

Option 2: E-commerce Store

- Product catalog
- Shopping cart
- Checkout flow
- Order history
- Product search and filters
- User reviews
- Wishlist
- LocalStorage + IndexedDB
- Payment simulation
- Admin panel

Option 3: Social Media Clone

- User profiles
- Posts (create, edit, delete)
- Comments and likes
- Follow system
- Real-time feed (simulated)
- Image uploads
- Search users and posts
- Notifications
- Dark/light theme

- Responsive design

Option 4: Project Management Tool

- Multiple projects
- Tasks with subtasks
- Team members
- Kanban board
- Gantt chart
- Time tracking
- Reports and analytics
- File attachments (simulated)
- Activity log
- Export to PDF

📊 Complete Timeline Summary

Phase	Duration	Topics	Projects
1	2-3 weeks	ES6+ Features	3 projects
2	3-4 weeks	Async JavaScript	4 projects
3	2 weeks	Functions & Closures	3 projects
4	2 weeks	OOP JavaScript	2 projects
5	2-3 weeks	Browser APIs & DOM	4 projects
6	1 week	Error Handling	1 project
7	1-2 weeks	Engine & Performance	1 project
8	2 weeks	Data Structures	2 projects
9	1-2 weeks	Functional Programming	2 projects
10	1-2 weeks	APIs & HTTP	2 projects
11	1 week	Regular Expressions	2 projects
12	1-2 weeks	Testing & Quality	1 project

Phase	Duration	Topics	Projects
Final	2-4 weeks	Capstone	1 large project

Total: 20-30 weeks (5-7 months) of focused learning