Detailed 45-Day Plan

Phase 1: JavaScript Core Fundamentals (Days 1-8)

Goal: Master JS building blocks with problem-solving.

Tool: Use LeetCode, HackerRank, or Codewars for problems; Node.js for execution.

- **Day 1: Variables, Scope, Hoisting**
  - JS: Solve "Reverse a String" (var/let/const differences); "Max Variable Scope" (hoisting/closure).
  - TS: N/A (focus on JS first).
- **Day 2: Functions & Closures**
  - JS: "Sum All Arguments" (args handling); "Counter Closure" (closure practice).
  - TS: N/A.
- **Day 3: Arrays & Iteration**
  - JS: "Filter Duplicates" (array methods); "Sum Even Numbers" (forEach/map/reduce).
  - TS: N/A.
- **Day 4: Objects & Prototypes**
  - JS: "Deep Clone Object" (recursion); "Inheritance Chain" (prototype basics).
  - TS: N/A.
- **Day 5: Async Basics (Promises)**
  - JS: "Fetch Mock Data" (Promise chaining); "Delay Sequence" (setTimeout + Promises).
  - TS: N/A.
- **Day 6: Async/Await**
  - JS: "Retry Failed Request" (async/await); "Parallel Promises" (Promise.all).
  - TS: N/A.
- **Day 7: Error Handling**
  - JS: "Try-Catch Validator" (input validation); "Custom Error Class" (throw errors).
  - TS: N/A.
- **Day 8: ES6+ Features**
  - JS: "Destructure Nested Object" (destructuring); "Spread Merge Arrays" (spread/rest).
  - TS: N/A.

Rest Day: Day 8 or 9.

Phase 2: TypeScript Core Fundamentals (Days 9-16)

Goal: Add TS rigor to JS knowledge.

Tool: TypeScript Playground or Node.js with TS setup.

- **Day 9: Types & Interfaces**
    - JS: "Array Sum" (plain JS revisit).
    - TS: Type "Array Sum" (basic types); "User Profile" (interface definition).
- **Day 10: Generics**
    - JS: "Merge Arrays" (plain JS).
    - TS: "Generic Merge" (generic function); "Typed Queue" (generic class).
- **Day 11: Type Narrowing**
    - JS: "Check Type" (typeof/instanceof).
    - TS: "Narrow Input" (union types); "Guard Function" (type guards).
- **Day 12: Enums & Tuples**
    - JS: "Status Switch" (plain JS).
    - TS: "Status Enum" (enum usage); "Coordinate Tuple" (tuple problem).
- **Day 13: Utility Types**
    - JS: "Pick Properties" (plain JS).
    - TS: "Pick<User>" (Pick utility); "Partial Update" (Partial utility).
- **Day 14: Async TS**
    - JS: "Fetch Data" (plain JS revisit).
    - TS: "Typed Fetch" (async function typing); "Promise<Error>" (error typing).
- **Day 15: Modules & Namespaces**
    - JS: "Export Helpers" (plain JS modules).
    - TS: "Typed Module" (export/import types); "Namespace Group" (namespace usage).
- **Day 16: TS with Classes**
    - JS: "Bank Account" (plain JS class).
    - TS: "Typed Bank" (class with access modifiers); "Interface Extend" (class + interface).

Rest Day: Day 16 or 17.

Phase 3: Intermediate JS/TS Problem Solving (Days 17-24)

Goal: Backend-relevant algorithms and data handling.

Tool: LeetCode, custom problems with Node.js/TS.

- **Day 17: Strings & Regex**
    - JS: "Parse URL Params" (string manipulation).
    - TS: "Typed Regex Validator" (email check).
- **Day 18: Trees & Recursion**
    - JS: "Binary Tree Depth" (recursion).
    - TS: "Typed Tree Sum" (tree traversal).
- **Day 19: Hash Maps**
    - JS: "Two Sum" (hash map approach).
    - TS: "Typed Frequency Counter" (map usage).

- **Day 20: Sorting & Searching**
  - JS: "Quick Sort" (implementation).
  - TS: "Typed Binary Search" (typed array).
- **Day 21: CRUD Operations**
  - JS: "In-Memory DB" (array CRUD).
  - TS: "Typed CRUD" (interface-based DB).
- **Day 22: Rate Limiting Logic**
  - JS: "Throttle Function" (debounce/throttle).
  - TS: "Typed Rate Limiter" (endpoint simulation).
- **Day 23: Pagination**
  - JS: "Paginate Array" (slice logic).
  - TS: "Typed Paginator" (generic pagination).
- **Day 24: Middleware Pattern**
  - JS: "Chain Functions" (middleware sim).
  - TS: "Typed Middleware" (type-safe chain).

Rest Day: Day 24 or 25.

Phase 4: Backend Endpoint Development (Days 25-32)

Goal: Build standalone endpoint functions with Node.js/Express/TS.

Tool: Node.js, Express, TS, Postman for testing.

- **Day 25: GET Endpoint**
  - JS: "Get Users" (array return).
  - TS: "Typed GET" (response typing).
- **Day 26: POST Endpoint**
  - JS: "Create User" (body parsing).
  - TS: "Typed POST" (input validation).
- **Day 27: PUT Endpoint**
  - JS: "Update User" (find + update).
  - TS: "Typed PUT" (partial updates).
- **Day 28: DELETE Endpoint**
  - JS: "Delete User" (filter logic).
  - TS: "Typed DELETE" (ID typing).
- **Day 29: Error Handling**
  - JS: "404 Handler" (custom errors).
  - TS: "Typed Errors" (union error types).
- **Day 30: Authentication**
  - JS: "Simple Auth" (token check).
  - TS: "Typed Auth" (middleware typing).
- **Day 31: Data Validation**
  - JS: "Validate Input" (manual checks).
  - TS: "Zod Validation" (typed schema).

- **Day 32: Database Mock**
  - JS: "Mock DB Query" (array sim).
  - TS: "Typed DB" (interface for rows).

Rest Day: Day 32 or 33.

Phase 5: Full-Stack Integration & Advanced Topics (Days 33-40)

Goal: Connect backend to React; explore advanced JS/TS.
Tool: Node.js/Express/TS + React.

- **Day 33: API + React**
  - JS: "Fetch API" (plain JS fetch).
  - TS: "Typed API Call" (React + TS).
- **Day 34: State Management**
  - JS: "Redux Clone" (plain JS).
  - TS: "Typed Reducer" (React + TS).
- **Day 35: File Handling**
  - JS: "Read File" (fs module).
  - TS: "Typed File Parser" (endpoint).
- **Day 36: Streams**
  - JS: "Stream Data" (basic stream).
  - TS: "Typed Stream" (pipe logic).
- **Day 37: WebSockets**
  - JS: "Chat Message" (ws module).
  - TS: "Typed WS" (event typing).
- **Day 38: Performance**
  - JS: "Memoize Function" (optimization).
  - TS: "Typed Memo" (generic memo).
- **Day 39: Testing**
  - JS: "Unit Test" (Jest basics).
  - TS: "Typed Test" (typed assertions).
- **Day 40: CI/CD Mock**
  - JS: "Deploy Script" (simulated).
  - TS: "Typed Config" (env typing).

Rest Day: Day 40 or 41.

Phase 6: Interview Prep & Polish (Days 41-45)

Goal: Ace mid-level interviews.
Tool: LeetCode, system design resources, mock interviews.

- **Day 41: Algorithm Review**
  - JS: "Top K Frequent" (heap sim).

- ○ TS: "Typed Solution" (generic).
- **Day 42: System Design**
  - ○ Sketch: "URL Shortener" (API + DB).
  - ○ Code: TS endpoint for it.
- **Day 43: Behavioral Prep**
  - ○ Practice: STAR method stories.
  - ○ Code: "Rate Limiter" revisit (TS).
- **Day 44: Mock Interview**
  - ○ Simulate: 1 JS + 1 TS problem.
  - ○ Review: Weak spots.
- **Day 45: Final Polish**
  - ○ Portfolio: Add 1-2 projects (e.g., API).
  - ○ Review: Key concepts/notes.