Here's a **1-year comprehensive roadmap** that will guide you through mastering **Data Structures and Algorithms (DSA)** and becoming proficient in **Development**. This roadmap is designed to build deep understanding and expertise through a balanced mix of theory, practice, and project-building.

1-Year DSA and Development Roadmap

Phase 1: DSA Foundations & Basic Development (Months 1-3)

Month 1: Core DSA Basics

- **Objective:** Build a strong foundation in fundamental data structures and algorithms.

Week 1-2:

- **Concepts:**
  - Big O Notation (Time & Space Complexity)
  - Arrays & Strings
  - Linked Lists (Singly & Doubly)
- **Practice:** 10-15 basic problems per week on arrays, strings, and linked lists (LeetCode/Easy & Medium).

Week 3-4:

- **Concepts:**
  - Stacks & Queues
  - Recursion
  - Hashing (HashMaps/HashSets)
- **Practice:** 10-15 problems on recursion, stacks, and hash maps.

Month 2: Sorting, Searching & Development Introduction

- **Objective:** Learn intermediate DSA and start with basic web development.

Week 1-2:

- **Concepts:**
  - Sorting Algorithms (Merge Sort, Quick Sort)
  - Searching Algorithms (Binary Search)
- **Practice:** 10-15 problems on searching/sorting and interval problems.

Week 3-4:

- **Development Focus:**

    - HTML, CSS, JavaScript basics.
    - Build your first static web pages and personal portfolio website.

Month 3: Advanced DSA Structures & Basic Projects

- **Objective:** Dive into advanced data structures and build small development projects.

Week 1-2:

- **Concepts:**
    - Trees (Binary Trees, Binary Search Trees)
    - Heaps (Priority Queue)
- **Practice:** 10-15 problems involving trees and heaps.

Week 3-4:

- **Development Focus:**
    - JavaScript (DOM Manipulation, Events).
    - Build an interactive to-do list or simple web app.

Phase 2: Advanced DSA & Full-Stack Development (Months 4-6)

Month 4: Graphs, Greedy & Dynamic Programming Introduction

- **Objective:** Tackle more advanced DSA topics and deepen web development knowledge.

Week 1-2:

- **Concepts:**
    - Graphs (BFS, DFS)
    - Greedy Algorithms
- **Practice:** 10-15 problems on graph traversal, greedy algorithms, and optimization problems.

Week 3-4:

- **Concepts:**
    - Dynamic Programming (DP) Introduction
    - Basic DP Problems (Knapsack, Fibonacci, Coin Change)
- **Practice:** 10-15 DP problems (begin with easy, progress to medium).

Month 5: Full-Stack Development Foundations

- **Objective:** Start building full-stack projects, focusing on both frontend and backend.

Week 1-2:

- **Frontend:**
    - Learn React.js basics (components, state, props).
    - Build a basic React application (e.g., weather app or a movie search app).

Week 3-4:

- **Backend:**
    - Learn Node.js or Python with Flask/Django.
    - Build a simple API (e.g., a CRUD app for task management).

Month 6: Advanced Graphs, DP & Project Development

- **Objective:** Enhance problem-solving skills and start building larger full-stack applications.

Week 1-2:

- **Concepts:**
    - Advanced Graph Algorithms (Dijkstra's, Kruskal's, Prim's)
- **Practice:** 10-15 problems on shortest paths, minimum spanning trees, and graph problems.

Week 3-4:

- **Project:** Build a full-stack application that integrates React with a backend API (e.g., a blogging platform or a chat app).

Phase 3: Deep DSA & Larger Development Projects (Months 7-9)

Month 7: Advanced DSA Techniques

- **Objective:** Master complex DSA topics.

Week 1-2:

- **Concepts:**
    - Advanced Dynamic Programming (DP)
    - Advanced Trees (Segment Trees, Fenwick Trees)
- **Practice:** 10-15 problems on advanced DP and tree-based problems.

Week 3-4:

- **Concepts:**
    - Trie Data Structure
    - Bit Manipulation
- **Practice:** 10-15 problems on tries (e.g., autocomplete, prefix matching) and bit manipulation (e.g., XOR, power of 2).

Month 8: Building Scalable Applications

- **Objective:** Develop scalable and complex full-stack applications.

Week 1-2:

- **Frontend:** Master advanced React concepts (Context API, Hooks, Redux for state management).
- **Backend:** Dive deeper into database integration (SQL & NoSQL), and learn about data modeling and schema design.

Week 3-4:

- **Project:** Build a scalable full-stack app, such as a social media platform or an e-commerce store, with authentication, data persistence, and security features.

Month 9: DevOps and Advanced Backend

- **Objective:** Learn to deploy, scale, and secure applications.

Week 1-2:

- **Concepts:**
    - Introduction to Docker and Kubernetes for containerization and orchestration.
- **Practice:** Containerize your full-stack app using Docker.

Week 3-4:

- **Concepts:**
    - CI/CD pipelines (GitHub Actions, Jenkins).
    - Server and API security (JWT, OAuth).
- **Project:** Set up a CI/CD pipeline for automatic deployment and build a secure production-ready application.

Phase 4: Competitive Programming & Industry-Level Projects (Months 10-12)

Month 10: Competitive Programming Focus

- **Objective:** Sharpen DSA skills through competitive coding and optimization.

Weeks 1-4:

- **Practice:** Participate in coding competitions (Codeforces, CodeChef, HackerRank contests) and solve medium-to-hard problems on topics like DP, graphs, and segment trees.

Month 11: Open Source Contributions & Advanced Project

- **Objective:** Contribute to open-source projects and work on industry-level applications.

Weeks 1-2:

- **Contribute:** Find open-source projects on GitHub related to your stack and contribute to them (bug fixes, feature additions, documentation).

Weeks 3-4:

- **Project:** Start building a large-scale application that you can showcase in your portfolio. Focus on scalability, optimization, and user experience (e.g., a job portal, collaborative tool, or real-time application).

Month 12: Mastery & Portfolio Building

- **Objective:** Polish your portfolio, refine your skills, and prepare for job interviews.

Week 1-2:

- **Concepts:**
  - System Design Basics (CAP Theorem, Microservices, Event-Driven Architecture).
- **Project:** Design and implement a microservice or event-driven application.

Week 3-4:

- **Final Project & Portfolio:** Polish your projects, write clean documentation, and ensure your portfolio reflects your strongest work.

Job Preparation:

- **Interview Prep:** Start mock interviews and solve problems related to technical interviews (LeetCode Hard).
- **Resume:** Finalize your resume and portfolio.

General Guidelines:

1. **Daily DSA Practice:** Dedicate 1-2 hours daily to DSA.
2. **Weekly Project Development:** Spend weekends building projects.
3. **Coding Competitions:** Participate in contests every 2 weeks to gauge your progress.
4. **Reflect & Refine:** Revisit difficult DSA problems and optimize your code.
5. **Collaborate & Contribute:** Join developer communities and contribute to open-source.

End Goal:

By the end of this year, you should be highly proficient in DSA and have developed a strong portfolio of full-stack applications. You’ll be well-prepared for coding interviews and capable of building scalable industry-level applications.

Does this roadmap align with your goals? Let me know if you'd like any adjustments!