2-Year Weekly Plan: Mastering DSA and Development

This roadmap is tailored for a 5th-semester B.Tech CSE AI student aiming to excel in DSA, development, and internships while preparing for an impressive portfolio and competitive coding.

Year 1: DSA Foundation and Development Basics

Months 1-3: Core DSA Foundations (Arrays, Strings, Linked Lists, Stacks, Queues)

Week	Focus	Description	Resources
1-2	Arrays & Strings	Learn operations like sorting, searching, and manipulation.	GeeksforGeeks, LeetCode, Neetcode.io
3-4	Linked Lists	Single, doubly, and circular linked lists.	CodeChef, YouTube
5-6	Stacks & Queues	Implementation and problem-solving.	freeCodeCamp, HackerRank
7-8	Recap & Revision	Solve problems and revise key concepts.	Practice platforms: LeetCode, Codeforces
9-10	Mock Contests	Speed practice with mock coding challenges.	Codeforces, AtCoder

Months 4-5: Intermediate DSA (Recursion, Trees, Hashing)

Week	Focus	Description	Resources
11-12	Recursion	Basic to advanced recursion and backtracking problems.	LeetCode, GeeksforGeeks
13-14	Trees	Binary trees, BSTs, and tree traversals.	Abdul Bari YouTube, freeCodeCamp
15-16	Hashing	Learn hashing, hash maps, and problem applications.	HackerRank, CodeChef
17-18	Problem Solving	Combine recursion, trees, and hashing in problems.	LeetCode, AtCoder

Months 6-8: Advanced DSA (Heaps, Graphs, DP)

Week	Focus	Description	Resources
19-20	Heaps	Priority queues, heap operations, and applications.	Abdul Bari YouTube, HackerRank
21-24	Graphs	BFS, DFS, shortest paths, and topological sorting.	GeeksforGeeks, Codeforces

Week	Focus	Description	Resources
25-28	Dynamic Programming	Basic DP patterns like Knapsack, Fibonacci.	Striver's DP Series, LeetCode

Months 9-12: Introduction to Development

Week	Focus	Description	Resources
29-32	Frontend Basics	Learn HTML, CSS, and JavaScript fundamentals.	freeCodeCamp, Codecademy
33-34	Backend Basics	Introduction to Node.js or Flask for API development.	YouTube, freeCodeCamp
35-36	Mini Project	Build a simple web project (e.g., blog, to-do app).	GitHub, Netlify
37-38	Machine Learning	Learn Python basics for ML (NumPy, Pandas, Matplotlib).	Coursera, Kaggle
39-40	ML Mini Project	Build an ML project (e.g., spam classifier).	Kaggle, GitHub

---

Year 2: Specialization, Advanced DSA, and Career Prep

Months 1-3: Specialization (Pick One: AI/ML, Web, Mobile)

Week	Focus	Description	Resources
41-44	Deep Dive	Learn advanced concepts in your specialization area.	Udemy, Coursera, Documentation
45-48	Advanced Project	Build a significant project showcasing your skills.	GitHub

Months 4-5: Competitive Programming

Week	Focus	Description	Resources
49-52	Problem Solving	Practice advanced DSA problems.	Codeforces, AtCoder, HackerRank
53-54	Mock Contests	Participate in contests for optimization skills.	Codeforces, TopCoder

Months 6-9: Portfolio Building and Hackathon Preparation

Week	Focus	Description	Resources
55-56	Portfolio	Build a personal website showcasing projects.	GitHub Pages, Medium

Week	Focus	Description	Resources
57-60	Hackathons	Join hackathons, learn teamwork and rapid prototyping.	Devpost, MLH

Months 10-12: Internship Applications

Week	Focus	Description	Resources
61-64	Resume Building	Tailor your resume for internships and job applications.	LinkedIn, Glassdoor
65-68	Mock Interviews	Practice DSA + development interviews.	Pramp, InterviewBit

---

Key Milestones

- **6 Months: Strong foundation in DSA and basic projects.**
- **1 Year: Advanced DSA and entry-level development skills.**
- **1.5 Years: Specialization and a major project.**
- **2 Years: Internship-ready, with hackathon and competitive coding experience.**

Detailed Weekly Roadmap for Full-Stack Development (6 Months)

Week	Focus Area	Goals and Skills	Resources/Tasks/Projects
1	**HTML, CSS, and DOM**	- Understand HTML structure, forms, and semantics. - Learn CSS basics and responsive design. - Explore the DOM and JavaScript integration.	- Build a static webpage. - Add interactivity with DOM manipulation.
2	**JavaScript Basics**	- Master variables, loops, functions, and arrays. - Explore ES6 features like let, const, and arrow functions.	- Create a to-do list app. - MDN Web Docs
3	**Node.js Basics**	- Understand the JavaScript runtime environment. - Write HTTP servers. - Explore Node.js vs other runtimes (e.g., Bun, Cloudflare).	- Build a simple Node.js server. - Compare runtimes.
4	**HTTP Servers & Express**	- Learn HTTP basics, request-response cycle. - Create REST APIs using Node.js and Express.	- Build a REST API for a blog platform.
5	**NoSQL Databases (MongoDB)**	- Understand NoSQL concepts. - Perform CRUD operations with MongoDB. - Explore schema design and indexing.	- Build a database for a "Bookshelf App" using MongoDB.
6	**SQL Basics**	- Learn SQL syntax. - Perform CRUD operations with SQL databases (e.g., MySQL or PostgreSQL).	- Build a simple database for user data.

Week	Focus Area	Goals and Skills	Resources/Tasks/Projects
7	**ORMs (Sequelize/Mongoose)**	- Learn about Object-Relational Mapping. - Use Sequelize for SQL and Mongoose for MongoDB. - Understand model relationships.	- Build a project that integrates Sequelize or Mongoose.
8	**React Basics**	- Learn React fundamentals: JSX, components, props, and state. - Use React Developer Tools.	- Build a task tracker app.
9	**Styling React Apps**	- Style React apps using Tailwind CSS or styled-components. - Learn responsive design in React.	- Restyle your task tracker app. - Tailwind CSS Docs
10	**TypeScript Basics**	- Learn TypeScript syntax and static typing. - Integrate TypeScript with React. - Explore type declarations for APIs and models.	- Refactor the task tracker app to TypeScript.
11	**Next.js Basics**	- Learn server-side rendering (SSR) and static site generation (SSG). - Build SEO-friendly apps.	- Build a blog site using Next.js.
12	**Monorepos and Turborepo**	- Understand monorepos and how to organize multiple projects. - Use Turborepo for performance optimization.	- Create a monorepo for shared libraries in React and Node.js.
13	**Websockets & RTC Basics**	- Understand real-time communication with Websockets.	- Build a chat app with Websockets.

Week	Focus Area	Goals and Skills	Resources/Tasks/Projects
		- Learn RTC for video/audio communication.	- Explore RTC for a video conferencing feature.
14	**Testing Basics**	- Learn unit and integration testing. - Use Jest and React Testing Library. - Test APIs and frontend components.	- Write tests for your REST API and React components.
15	**Authentication & JWT**	- Understand secure user authentication. - Use JWT and cookies for session management.	- Add login/logout functionality to your blog platform.
16	**Database Optimization**	- Explore database optimization techniques (e.g., indexing, query optimization).	- Optimize queries for existing projects.
17	**Advanced Backend Concepts**	- Dive deeper into asynchronous programming and error handling. - Learn middleware chaining. - Implement complex backend logic.	- Refactor APIs with advanced features like rate limiting and complex queries.
18	**Ref, Populate, and API Design**	- Use ref and populate in Mongoose for relationships. - Learn API versioning and documentation (Swagger).	- Add relationships to your MongoDB models.
19	**Advanced Frontend Features**	- Learn dynamic form handling, animations, and accessibility features in React. - Explore Framer Motion for animations.	- Add form validation and animations to the task tracker.

Week	Focus Area	Goals and Skills	Resources/Tasks/Projects
20	**Project Planning**	- Plan a capstone project. - Define features, create high-level (HLD) and low-level designs (LLD).	- Create a project blueprint (e.g., e-commerce app or task tracker).
21	**Capstone Backend Development**	- Build the backend for your capstone project. - Implement CRUD APIs, authentication, and error handling.	- Set up Express.js, MongoDB, and authentication for the project.
22	**Capstone Frontend Development**	- Build the frontend for your capstone project. - Use React or Next.js for user interface. - Style the app with Tailwind CSS.	- Integrate the backend with the frontend.
23	**Real-Time Features in Capstone**	- Add real-time features using Websockets or RTC. - Implement live chat or real-time notifications.	- Enhance the capstone app with real-time functionality.
24	**Testing and Deployment**	- Test the app thoroughly. - Deploy the app to a cloud platform (e.g., Vercel, Heroku).	- Write unit and integration tests. - Deploy the capstone project.
25	**Linting and Performance Optimization**	- Set up linting tools (e.g., ESLint, Prettier). - Optimize frontend and backend performance. - Learn about caching strategies.	- Add ESLint and Prettier to your projects. - Optimize API response times and frontend load times.
26	**Final Enhancements & Portfolio**	- Polish the capstone project. - Add the project to a portfolio site. - Prepare for interviews with projects and a resume.	- Create a personal portfolio using Next.js. - Showcase all 10 projects in your portfolio.

Here's a list of **10 project ideas** aligned with the roadmap, along with a guide to building them in a tabular format:

Project Name	Description	Steps to Build	Tech Stack
Task Tracker App	A React app to track daily tasks with add, edit, and delete functionality.	1. Set up React. 2. Build a task list UI. 3. Add state management with React hooks. 4. Store tasks locally.	React, CSS
Bookshelf App	A MongoDB-powered app to manage a collection of books with search and CRUD operations.	1. Set up MongoDB. 2. Create Express APIs for CRUD operations. 3. Connect backend to a React frontend.	Node.js, Express, MongoDB, React
Blog Platform	A platform for users to post, edit, and delete blog entries. Includes JWT-based authentication.	1. Build REST APIs with Express. 2. Use MongoDB or PostgreSQL for storing blogs. 3. Add authentication.	Node.js, MongoDB/PostgreSQL, React
Chat App	Real-time chat app using Websockets with user authentication.	1. Set up Websockets on the backend. 2. Implement chat UI in React. 3. Add JWT authentication.	Node.js, Websockets, React
E-Commerce Store	A store with product listings, a cart system, and checkout functionality.	1. Design product APIs. 2. Add cart logic in React. 3. Integrate Razorpay/Stripe for payments.	React, Node.js, MongoDB, Payment APIs

Project Name	Description	Steps to Build	Tech Stack
Portfolio Website	A personal portfolio to showcase projects and skills.	1. Use Next.js for SSR. 2. Style with Tailwind CSS. 3. Add portfolio content dynamically using JSON.	Next.js, Tailwind CSS
Expense Tracker	A tool to track income and expenses with visualizations.	1. Create React components for transactions. 2. Store data in MongoDB. 3. Visualize data using Chart.js.	React, MongoDB, Chart.js
Online Classroom Platform	A platform for faculty to share documents, conduct live sessions, and create assignments.	1. Build backend with Express. 2. Implement user roles. 3. Add WebRTC for live classes.	Node.js, WebRTC, React
Video Conferencing App	A basic conferencing app using WebRTC for real-time communication.	1. Set up WebRTC APIs. 2. Build React frontend. 3. Manage connections using signaling servers.	WebRTC, React
Capstone E-Commerce App	A fully-featured app with real-time notifications, authentication, and performance optimization.	1. Plan HLD and LLD. 2. Develop backend APIs. 3. Add Websocket-based notifications. 4. Deploy on the cloud.	Node.js, React, MongoDB, Websockets

Details on Building the Projects:

1. **Progressive Complexity**: Projects grow in difficulty from simple React apps (e.g., Task Tracker) to advanced, multi-featured applications (e.g., Capstone E-Commerce App).
2. **Full Stack Integration**: Emphasizes both frontend and backend skills, including databases, APIs, and real-time communication.
3. **Deployment**: Includes a deployment step (e.g., using Vercel, Heroku, or AWS) to simulate production environments.

Let me know if you need detailed guides for specific projects!

JavaScript Learning Roadmap: Topics, Theory, and Projects

Step	Topic	Theory	Project Idea
1	Basics of JavaScript	Learn about variables, data types, operators, and expressions.	Build a **Simple Calculator** for basic arithmetic operations.
2	Control Structures	Study if, else, switch, and loops (for, while, do-while).	Create a **Guess the Number Game** where the user guesses a randomly generated number.
3	Functions	Understand function declaration, expressions, and arrow functions.	Develop a **Temperature Converter** using functions for conversions.
4	DOM Manipulation	Learn how to access and modify HTML elements with document.querySelector and document.getElementById.	Build a **Dynamic To-Do List** that lets users add, edit, and delete tasks.
5	Events and Event Listeners	Explore event types, event propagation, and how to handle events.	Create a **KeyPress Visualizer** that shows key codes when typing.

Step	Topic	Theory	Project Idea
6	Arrays	Study array methods like push, pop, filter, map, reduce, and forEach.	Create a **Grocery List Manager** with features like adding, editing, and sorting items.
7	Objects	Learn about objects, properties, methods, and how to loop through objects.	Build a **Student Grades Tracker** using objects for student data.
8	ES6 Features	Study modern features like let, const, template literals, destructuring, and the spread/rest operator.	Build a **Recipe Manager** app using ES6 features to store and edit recipes.
9	JSON and APIs	Understand JSON format and how to fetch data from APIs using fetch.	Fetch and display weather data using a **Weather App**.
10	Promises and Async/Await	Learn how to handle asynchronous operations with Promises and async/await.	Build a **GitHub User Finder** that fetches user data using the GitHub API.
11	Error Handling	Study try, catch, and finally for managing errors in code.	Create a **Form Validator** tha

Step	Topic	Theory	Project Idea
			t handles errors for invalid inputs.
12	Classes	Explore ES6 classes, constructors, inheritance, and methods.	Develop a **Library App** with book objects and class-based structure.
13	Modules	Learn how to use import and export to organize your code into modules.	Build a **Multi-Page Portfolio Website** using JavaScript modules for navigation.
14	Browser Storage	Study localStorage, sessionStorage, and cookies for storing data in the browser.	Create a **Notes App** that saves notes in localStorage.
15	Regular Expressions (Regex)	Learn basic Regex syntax for pattern matching and validation.	Build a **Password Strength Checker** using Regex to validate input.
16	Date and Time	Understand how to work with dates and times using the Date object.	Create a **Countdown Timer** for an event with real-time updates.

Step	Topic	Theory	Project Idea
17	Closures	Study closures and how they capture variables from their surrounding scope.	Develop a **Stopwatch App** that uses closures to manage time intervals.
18	Prototypes and Inheritance	Understand JavaScript's prototype chain and inheritance.	Build a **Shape Drawer** with different shapes inheriting from a common prototype.
19	Event Loop and Async Flow	Learn how JavaScript’s event loop and call stack work for asynchronous behavior.	Visualize the event loop with an **Interactive Event Loop Diagram** using animations.
20	Advanced Frameworks/Tools	Study frameworks like React or Node.js to expand your JavaScript knowledge into frontend or backend development.	Create a **Chat Application** with a real-time backend (e.g., Node.js and Socket.IO).

This roadmap covers foundational to advanced concepts, ensuring a balance between theoretical understanding and practical application. Each project complements its respective topic to strengthen hands-on skills.

Web Development Skills

Category	Skill
Front-End Development	HTML5
	CSS3
	JavaScript (ES6+)
	React.js
	Angular
	Vue.js
	Responsive Design
	Tailwind CSS
	Bootstrap
	SASS/SCSS
	TypeScript
	Web Accessibility (WCAG)
	Cross-Browser Compatibility
	Figma/XD Integration
	DOM Manipulation
	Web Animations (CSS, GSAP, Framer Motion)
	Styled Components
	Material-UI (MUI)
	Redux Toolkit/Zustand (State Management)
	Progressive Web Apps (PWAs)

Category	Skill
Back-End Development	Node.js
	Express.js
	Django
	Flask
	Ruby on Rails
	PHP
	RESTful API Development
	GraphQL
	Authentication & Authorization (OAuth, JWT)
	Server-Side Rendering (SSR)
Database Management	SQL (MySQL, PostgreSQL)
	NoSQL (MongoDB, Firebase)
	Database Optimization
	Cloud Databases (AWS RDS, Azure, Firestore)
DevOps & Deployment	Version Control (Git/GitHub/GitLab)
	CI/CD Pipelines (Jenkins, GitHub Actions)
	Docker & Containerization
	Cloud Platforms (AWS, Azure, GCP)
	Deployment (Netlify, Vercel, Heroku)
Web Performance & Security	Webpack/Parcel/Vite
	Image Optimization

Category	Skill
	Content Delivery Networks (CDNs)
	SSL/TLS Certificates
	Security Best Practices (XSS, CSRF, SQL Injection)
	Performance Monitoring Tools (Lighthouse, GTmetrix)
Additional Tools & Soft Skills	API Integration (REST, SOAP, GraphQL)
	Testing & Debugging (Jest, Cypress, Chrome DevTools)
	Agile Development Methodology
	Communication & Collaboration Tools (Slack, Trello)
	UX/UI Design Principles

Best YouTube Channels for Learning Web Development Skills

Skill	YouTube Resource
HTML, CSS, JS Basics	Web Development Full Course - Love Babbar
React	React Tutorial for Beginners - Chai aur code
JavaScript	JavaScript for Beginners - Namaste Javascript
MongoDB	MongoDB Crash Course - Thapa
Python	Python for Beginners - Code With Harry
TypeScript	TypeScript Crash Course - Hitesh Chaudhary
Flask	Flask for Beginners - Freecodecamp
SQL	SQL Tutorial for Beginners - Apna College
TypeScript	TypeScript Full Course - freeCodeCamp
Spring Boot	Spring Boot Tutorial - Amigoscode

Skill	YouTube Resource
Next.js	Next.js Crash Course - Traversy Media
Tailwind CSS	Tailwind CSS Tutorial - JS Mastery
Docker	Docker Tutorial - Harkirat Singh
Node.js	Node.js Crash Course - Traversy Media
GraphQL	GraphQL Tutorial - freeCodeCamp
Web Accessibility	Web Accessibility - freeCodeCamp
Agile Development	Agile Development Basics - Edureka

Project Playlist: **https://www.youtube.com/watch?v=RbxHZwFtRT4&list=PL6QREj8te1P6wX9m5KnicnDVEucbOPsqR**

50 tricks to identify DSA Patterns
Link: https://drive.google.com/drive/folders/1Da_v5uHIvBscWcRRgMsYGq-hJ00dQL9Y

Company Name	DSA Problem Link	Tricks for Solving Patterns
Google	Google DSA Problems	Focus on backtracking and dynamic programming.
Microsoft	Microsoft DSA Problems	Practice binary search, greedy algorithms.
Amazon	Amazon DSA Problems	Work on sliding window, heap, and DFS/BFS.
Facebook	Facebook DSA Problems	Master recursion, graphs, and bit manipulation.
Apple	Apple DSA Problems	Prioritize dynamic programming and divide and conquer.

Company Name	DSA Problem Link	Tricks for Solving Patterns
Adobe	Adobe DSA Problems	Practice sorting and searching techniques.
Goldman Sachs	Goldman Sachs DSA Problems	Focus on arrays, hashing, and string manipulation.
Uber	Uber DSA Problems	Work on graphs, BFS/DFS, and Dijkstra's algorithm.
LinkedIn	LinkedIn DSA Problems	Focus on two-pointer and sliding window techniques.
Netflix	Netflix DSA Problems	Prioritize dynamic programming and DP on trees.
Twitter	Twitter DSA Problems	Work on greedy algorithms and interval problems.
Dropbox	Dropbox DSA Problems	Master recursion and binary search on sorted arrays.
Airbnb	Airbnb DSA Problems	Practice graph traversal and backtracking.
Salesforce	Salesforce DSA Problems	Work on greedy algorithms and dynamic programming.
Oracle	Oracle DSA Problems	Focus on matrix traversal and dynamic programming.
PayPal	PayPal DSA Problems	Prioritize binary search and sorting problems.
Walmart	Walmart DSA Problems	Focus on hashing, prefix sums, and arrays.
Expedia	Expedia DSA Problems	Practice stack-based problems and recursion.
Snap	Snap DSA Problems	Focus on graphs and dynamic programming.
Yahoo	Yahoo DSA Problems	Work on linked lists and recursion.
DoorDash	DoorDash DSA Problems	Master binary trees and backtracking.

Company Name	DSA Problem Link	Tricks for Solving Patterns
Stripe	Stripe DSA Problems	Practice greedy algorithms and string manipulation.
Lyft	Lyft DSA Problems	Focus on two-pointer and sliding window techniques.
Intuit	Intuit DSA Problems	Work on backtracking and dynamic programming.
IBM	IBM DSA Problems	Master dynamic programming and recursion problems.
Atlassian	Atlassian DSA Problems	Focus on graph traversal and dynamic programming.
Reddit	Reddit DSA Problems	Work on hashing and bit manipulation.
Pinterest	Pinterest DSA Problems	Master recursion and divide and conquer techniques.
Spotify	Spotify DSA Problems	Focus on sorting, searching, and heaps.
Bloomberg	Bloomberg DSA Problems	Work on arrays, dynamic programming, and graphs.
Cisco	Cisco DSA Problems	Focus on linked lists and dynamic programming.
ByteDance	ByteDance DSA Problems	Master sorting algorithms and binary search.
Tesla	Tesla DSA Problems	Work on graph traversal and dynamic programming.
TikTok	TikTok DSA Problems	Prioritize dynamic programming and recursion.
Nvidia	Nvidia DSA Problems	Practice bit manipulation and backtracking.

Most Common Resume Projects and their Advanced Version

Sno	Project	Basic Version	Advanced Version	Resource Link
1	Tic Tac Toe Game	Basic Tic-Tac-Toe Game	🎖️ Multiplayer Tic-Tac-Toe using WebSockets	WebSockets Tutorial
2	Weather Application	Simple Weather App	🎖️ Weather Bot for Discord with Alerts	Discord.js Guide
3	Ecommerce Website	Basic Ecommerce Website	🎖️ AI-Driven Chrome Extension for Price Comparisons Using Web Scraping	Puppeteer Docs
4	Crypto Tracker	Static Crypto Tracker	🎖️ AI-Powered Crypto Trading Bot	Binance API
5	Portfolio Website	Static Personal Portfolio Website	🎖️ Interactive Portfolio with Real-Time Analytics & AI Bot	ReactJS
6	Calculator	Simple Calculator	🎖️ Advanced Financial Calculator with Tax, Loan, and Investment Features	MDN Calculator Guide
7	To-Do List	Basic To-Do App	🎖️ Collaborative To-Do App with Task Dependencies and Deadlines	Firebase Docs
8	Blog Platform	Simple Blog Platform	🎖️ Dynamic Blog Platform with NLP-based Content Recommendations and Summaries	TensorFlow.js
9	Social Media App	Basic Social Media Platform	🎖️ Decentralized Social Media App with User-Curated Content Streams	Web3.js
10	Language Translator	Basic Language Translator	🎖️ AI-Powered Real-Time Voice Translator with Sentiment Analysis	Google Cloud Translate

Sno	Project	Basic Version	Advanced Version	Resource Link
11	Task Management Tool	Basic Task Manager	🎖️ AI-Based Task Manager with Productivity Insights and Notifications	Microsoft AI API
12	Fitness Tracker	Simple Fitness Tracker	🎖️ AI-Powered Fitness Tracker with Workout Recommendations and Nutritional Analysis	Nutritionix API
13	Chatbot	Basic Chatbot	🎖️ AI Chatbot with Contextual Memory and Personalized Responses	Dialogflow
14	Music Streaming App	Simple Music Streaming App	🎖️ AI-Based Music Recommendation Platform with Mood Detection	Spotify API
15	Recipe Finder	Simple Recipe Search	🎖️ AI-Powered Recipe Finder with Dietary Preferences and Nutritional Analysis	Spoonacular API
16	Quiz App	Basic Quiz Application	🎖️ Gamified Quiz App with Leaderboards and Interactive Feedback	Open Trivia Database
17	Video Streaming App	Simple Video Streaming App	🎖️ AI-Based Video Summarization and Interactive Viewing Experience	Mux API
18	Stock Market Tracker	Basic Stock Market Tracking App	🎖️ Real-Time Stock Market Dashboard with Predictive Analytics	Alpha Vantage
19	Virtual Classroom	Basic Virtual Classroom App	🎖️ Interactive Classroom Platform with AI-Powered Participation Insights and Real-Time Feedback	BigBlueButton

Sno	Project	Basic Version	Advanced Version	Resource Link
20	Travel Planner	Simple Travel Itinerary App	🎖️ AI Travel Planner with Real-Time Location-Based Recommendations and Booking Integrations	Amadeus for Developers

3-Month Logic-Building & Problem-Solving Roadmap

Week	Focus Area	Daily Breakdown	Checkpoints
Week 1	**Foundation: Basics of Programming**	**Day 1-2:** Learn variables, data types, loops (for, while). **Day 3-4:** Conditionals, functions. **Day 5-6:** Arrays & Strings basics. **Day 7:** Solve 10 simple pattern-building questions (stars, triangles).	✅ Understand basic syntax and concepts. ✅ Solve **10 pattern-building problems**.
Week 2	**Introduction to Problem Solving**	**Day 1-2:** Learn dry-run and pseudocode. **Day 3-5:** Solve **10 simple DSA problems** (easy level, e.g., reverse array, Fibonacci). **Day 6-7:** Build **Mini Project 1: Calculator App**.	✅ Solve 10 DSA problems. ✅ Complete **Mini Project 1: Calculator App**.
Week 3	**Level Up: Arrays & Logic Thinking**	**Day 1-2:** Advanced array techniques (sliding window, prefix sum). **Day 3-6:** Solve **15 array-based problems** (medium level). **Day 7:** Build **Mini Project 2: Tic-Tac-Toe Game**.	✅ Solve 25 total DSA problems (15 new). ✅ Complete **Mini Project 2: Tic-Tac-Toe Game**.
Week 4	**Strings & Problem-Solving Practice**	**Day 1-2:** String manipulation (reversal, palindromes, substrings). **Day 3-5:** Solve **15 string-based problems** (easy-medium). **Day 6-7:** Build **Mini Project 3: Text Manipulation Tool (e.g., Uppercase, Reverse)**.	✅ Solve 40 total DSA problems (15 new). ✅ Complete **Mini Project 3: Text Tool**.
Week 5	**Recursion & Logic Expansion**	**Day 1-2:** Learn recursion basics (factorial, Fibonacci). **Day 3-5:** Solve **10 recursion-based problems** (medium level). **Day 6-7:** Build **Mini Project 4: Recursive Maze Solver**.	✅ Solve 50 total DSA problems (10 new). ✅ Complete **Mini Project 4: Recursive Maze Solver**.
Week 6	**Data Structures: Stacks & Queues**	**Day 1-2:** Learn stacks and queues basics.	✅ Solve 65 total DSA problems (15 new).

Week	Focus Area	Daily Breakdown	Checkpoints
		Day 3-6: Solve **15 problems** (balanced parentheses, queue reversal). **Day 7:** Work on **Mini Project 5: Browser History Tracker (using stack)**.	✅ Complete **Mini Project 5: Browser History Tracker**.
Week 7	**Dynamic Programming Introduction**	**Day 1-3:** Learn DP basics (knapsack, Fibonacci with memoization). **Day 4-6:** Solve **10 DP problems** (easy-medium). **Day 7:** Revise all past concepts/projects.	✅ Solve 75 total DSA problems (10 new). ✅ Master basic DP problems.
Week 8	**Advanced Problem Solving (Graph)**	**Day 1-2:** Learn graph representation (adjacency list/matrix). **Day 3-5:** Solve **10 graph-based problems** (BFS/DFS). **Day 6-7:** Build a **Mini Project: Path Finder Visualizer**.	✅ Solve 85 total DSA problems (10 new). ✅ Complete Graph-Based Mini Project.
Week 9	**Advanced DSA (Sorting/Greedy)**	**Day 1-2:** Learn sorting algorithms (merge sort, quicksort). **Day 3-5:** Solve **15 problems** on sorting/greedy (e.g., activity selection, job scheduling).	✅ Solve 100 DSA problems (15 new).
Week 10-11	**Integration: Complex Projects**	**Day 1-7:** Brainstorm and build **Project 1: Expense Tracker with Charts**. **Next 7 days:** Build **Project 2: Multiplayer Rock-Paper-Scissors Game** (WebSockets, if possible).	✅ Integrate past knowledge into real-world projects. ✅ Master real-world application-building.
Week 12	**Final Touches and Mock Practice**	**Day 1-3:** Revise all concepts learned. **Day 4-6:** Solve 10 problems from past mistakes or blindspots. **Day 7:** Build a final project of	✅ Complete final project. ✅ Feel confident in solving beginner-intermediate DSA problems.

Week	Focus Area	Daily Breakdown	Checkpoints
		your choice integrating at least one complex DSA concept.	

💡 **How to Think of Logic**

1. **Break Down the Problem**: Read the question twice. Identify inputs, outputs, and constraints.
2. **Start with Examples**: Create test cases manually and simulate the solution step-by-step.
3. **Write Pseudocode**: Draft a high-level plan before coding.
4. **Ask "Why?" at Every Step**: Understand each operation; don't memorize solutions.
5. **Visualize**: Use diagrams or dry-run tables to debug.
6. **Optimize Gradually**: Start with brute force; iterate to optimize for efficiency.

📚 **100 DSA Questions for Logic Building**

Category	Question Name	Platform	Link
Basics & Warm-Up	Print a pattern of stars (triangle, pyramid)	GeeksforGeeks	Link
	Reverse a number	GeeksforGeeks	Link
	Check if a number is palindrome	GeeksforGeeks	Link
	Count digits in a number	GeeksforGeeks	Link
	Find factorial of a number	HackerRank	Link
Arrays	Reverse an array	LeetCode	Link
	Find the maximum and minimum of an array	GeeksforGeeks	Link
	Rotate an array by K steps	LeetCode	Link
	Move all zeroes to the end	LeetCode	Link
	Kadane's Algorithm (Maximum Subarray Sum)	LeetCode	Link
Strings	Reverse a string	LeetCode	Link

Category	Question Name	Platform	Link
	Check if two strings are anagrams	LeetCode	Link
	Longest Common Prefix	LeetCode	Link
	Check if a string is a palindrome	LeetCode	Link
	Count and say	LeetCode	Link
Recursion	Fibonacci series using recursion	GeeksforGeeks	Link
	Tower of Hanoi	GeeksforGeeks	Link
	Factorial using recursion	LeetCode	Link
	Reverse a linked list using recursion	LeetCode	Link
	Permutations of a string	GeeksforGeeks	Link
Sorting	Bubble sort	GeeksforGeeks	Link
	Selection sort	GeeksforGeeks	Link
	Merge sort	GeeksforGeeks	Link
	Quick sort	GeeksforGeeks	Link
	Insertion sort	GeeksforGeeks	Link
Searching	Binary search	LeetCode	Link
	Linear search	GeeksforGeeks	Link
	Search in a rotated sorted array	LeetCode	Link
	First and last position in a sorted array	LeetCode	Link
	Square root of a number (using binary search)	LeetCode	Link
Linked List	Reverse a linked list	LeetCode	Link

Category	Question Name	Platform	Link
	Detect a cycle in a linked list	LeetCode	Link
	Merge two sorted linked lists	LeetCode	Link
	Remove Nth node from the end	LeetCode	Link
	Find the middle of a linked list	LeetCode	Link
Stacks & Queues	Implement a stack using arrays	LeetCode	Link
	Evaluate postfix expression	GeeksforGeeks	Link
	Balanced parentheses	LeetCode	Link
	Next greater element	LeetCode	Link
	Implement a queue using stacks	LeetCode	Link
Dynamic Programming	0/1 Knapsack problem	GeeksforGeeks	Link
	Fibonacci using dynamic programming	LeetCode	Link
	Longest common subsequence	LeetCode	Link
	Longest increasing subsequence	LeetCode	Link
	Minimum steps to reach the end	LeetCode	Link
Graphs	BFS traversal	GeeksforGeeks	Link
	DFS traversal	GeeksforGeeks	Link
	Detect cycle in an undirected graph	GeeksforGeeks	Link
	Shortest path in a graph (Dijkstra's)	GeeksforGeeks	Link
	Topological sort	GeeksforGeeks	Link

Project Overview	- Can you give an overview of your recent project?
	- What was the purpose of the project, and how did it solve the problem?
	- What technologies did you use, and why?
Challenges & Solutions	- What were the biggest challenges you faced during the project?
	- How did you handle performance bottlenecks or scaling issues?
	- Describe a time when something went wrong. How did you resolve it?
Technical Stack	- Why did you choose this particular stack for your project?
	- What design patterns or architectural decisions did you follow?
	- How did you ensure security in your application?
Team Collaboration	- Did you work in a team? If so, how did you manage code collaboration?
	- How did you handle merging conflicts and version control?
	- How did you communicate with team members about changes or blockers?
Scalability & Optimization	- How did you design your project to be scalable?
	- What kind of optimizations did you implement to improve performance?
	- How did you handle data-intensive operations or large datasets?
Testing & Quality	- How did you test your project?
	- What kind of testing framework or methodology did you use (unit, integration, e2e)?
	- How did you manage deployment and continuous integration?

Innovative Features	- Did you implement any innovative or unique features? - How did you incorporate AI/ML, blockchain, or other cutting-edge tech in your project?
Project Outcomes	- What was the final outcome or impact of your project? - How did users or stakeholders respond to your solution? - Can you provide any metrics that indicate the success of your project (e.g., performance gains, user adoption)?
Future Improvements	- If you could revisit this project, what improvements or changes would you make? - What additional features do you plan to add in the future?

Here are some resources and roadmaps to help you learn React, Python, Generative AI (Gen AI), Retrieval-Augmented Generation (RAG), and Augmented Reality (AR):

1. React

- **Roadmap**:
  - React Developer Roadmap
  - React Learning Path - freeCodeCamp
- **Courses & Tutorials**:
  - React – The Complete Guide (Udemy)
  - Frontend Masters React Learning Path

2. Python

- **Roadmap**:
  - Python Developer Roadmap
- **Courses & Tutorials**:
  - Python for Everybody (Coursera)

3. Generative AI (Gen AI)

- **Introduction Resources**:
  - Google's Generative AI Learning Path
  - OpenAI Resources
- **Roadmap**:
  - Generative AI Roadmap (GitHub)
- **Courses & Tutorials**:
  - Generative AI with Large Language Models

4. Retrieval-Augmented Generation (RAG)

- **Overview and Learning Resources**:
  - RAG Overview on GitHub
  - How RAG Works (Article)
- **Courses & Tutorials**:
  - Implementing RAG with OpenAI API (YouTube)

5. Augmented Reality (AR)

- **Official Documentation**:
  - ARCore Documentation (Google)

    - ARKit Documentation (Apple)
- **Roadmap**:
    - AR Developer Roadmap
- **Courses & Tutorials**:
    - Unity AR Tutorial
    - Coursera - AR Development with Unity

These resources provide comprehensive learning paths from beginner to advanced levels.

🦉 Night Owl Algorithm

Time	Activity
8:30 AM	Wake up and freshen up
9:00 AM - 1:00 PM	Internship work / Study
1:00 PM - 2:00 PM	Lunch break
2:00 PM - 5:00 PM	Project work / Study
5:30 PM - 8:00 PM	Physical Exercise
8:00 PM - 9:00 PM	Dinner
9:00 PM - 12:00 AM	Study / DSA Practice
12:30 AM - 1:30 AM	Self-learning
1:30 AM	Wind down and sleep

🤵 Weekends Algorithm

Time	Activity
6:30 AM	Wake up and freshen up
7:00 AM - 9:00 AM	Physical Exercise
9:00 AM - 12:00 PM	Self-learning
12:00 PM - 1:00 PM	Lunch break
1:00 PM - 4:00 PM	DSA Practice
4:00 PM - 6:00 PM	Rest / Relaxation
6:00 PM - 9:00 PM	Project work / Hackathon prep
9:00 PM - 11:00 PM	Internship work / Leisure time
11:00 PM	Wind down and relax

⏰ **Prioritizing Algorithm**

Time	Activity
6:00 AM	Start the day by prioritizing tasks
7:00 AM - 12:00 PM	Focus on highest priority task
12:00 PM - 1:00 PM	Lunch break
1:00 PM - 5:00 PM	Dedicated block for internships
5:00 PM - 7:00 PM	Physical exercise / Free time
7:00 PM - 9:00 PM	Study session / DSA Practice
9:00 PM - 11:00 PM	Project work / Hackathon prep

Company Name	DSA Problem Link	Tricks for Solving Patterns
Google	Google DSA Problems	Focus on backtracking and dynamic programming.
Microsoft	Microsoft DSA Problems	Practice binary search, greedy algorithms.
Amazon	Amazon DSA Problems	Work on sliding window, heap, and DFS/BFS.
Facebook	Facebook DSA Problems	Master recursion, graphs, and bit manipulation.
Apple	Apple DSA Problems	Prioritize dynamic programming and divide and conquer.
Adobe	Adobe DSA Problems	Practice sorting and searching techniques.
Goldman Sachs	Goldman Sachs DSA Problems	Work on dynamic programming and number theory.
Uber	Uber DSA Problems	Master graph traversal, greedy methods.
LinkedIn	LinkedIn DSA Problems	Work on graph algorithms and hashmaps.
Netflix	Netflix DSA Problems	Focus on greedy algorithms and dynamic programming.
Twitter	Twitter DSA Problems	Master sliding window and prefix sum techniques.
Dropbox	Dropbox DSA Problems	Work on recursion, memoization, and hashmaps.
Airbnb	Airbnb DSA Problems	Focus on backtracking and greedy algorithms.
Salesforce	Salesforce DSA Problems	Prioritize heap, greedy, and two-pointer techniques.
Oracle	Oracle DSA Problems	Master string manipulation and binary trees.
PayPal	PayPal DSA Problems	Focus on recursion and dynamic programming.

Company Name	DSA Problem Link	Tricks for Solving Patterns
Walmart	Walmart DSA Problems	Practice array manipulation and sorting algorithms.
Expedia	Expedia DSA Problems	Focus on DFS/BFS, recursion, and backtracking.
Snap	Snap DSA Problems	Practice two-pointer and graph traversal methods.
Yahoo	Yahoo DSA Problems	Master dynamic programming and recursion techniques.
DoorDash	DoorDash DSA Problems	Focus on sliding window and interval problems.
Stripe	Stripe DSA Problems	Practice dynamic programming and number theory.
Lyft	Lyft DSA Problems	Focus on recursion and hashmaps.
Intuit	Intuit DSA Problems	Master greedy algorithms and dynamic programming.
IBM	IBM DSA Problems	Focus on graph traversal and recursion techniques.
Atlassian	Atlassian DSA Problems	Practice dynamic programming and sliding window.
Reddit	Reddit DSA Problems	Focus on graph traversal and sorting algorithms.
Pinterest	Pinterest DSA Problems	Practice recursion and backtracking.
Spotify	Spotify DSA Problems	Master sliding window and string manipulation.
Bloomberg	Bloomberg DSA Problems	Work on greedy algorithms and dynamic programming.
Cisco	Cisco DSA Problems	Focus on recursion and divide and conquer.

Company Name	DSA Problem Link	Tricks for Solving Patterns
ByteDance	ByteDance DSA Problems	Master dynamic programming and graph traversal.
Tesla	Tesla DSA Problems	Practice heap, greedy, and backtracking algorithms.
TikTok	TikTok DSA Problems	Focus on recursion and sliding window techniques.
Nvidia	Nvidia DSA Problems	Work on dynamic programming and sorting algorithms.
Square	Square DSA Problems	Focus on graph traversal and string manipulation.
Accenture	Accenture DSA Problems	Master sorting, searching, and tree traversal.
Cognizant	Cognizant DSA Problems	Focus on recursion, hashmaps, and binary trees.
Infosys	Infosys DSA Problems	Practice dynamic programming and graph traversal.
TCS	TCS DSA Problems	Master greedy algorithms and two-pointer techniques.
Capgemini	Capgemini DSA Problems	Focus on divide and conquer and recursion.
HCL	HCL DSA Problems	Work on recursion, sorting, and searching.
Morgan Stanley	Morgan Stanley DSA Problems	Master dynamic programming and number theory.
JPMorgan Chase	JPMorgan Chase DSA Problems	Focus on greedy algorithms and backtracking.
Barclays	Barclays DSA Problems	Master dynamic programming and array manipulation.
American Express	American Express DSA Problems	Focus on heap and priority queue techniques.

Company Name	DSA Problem Link	Tricks for Solving Patterns
Dell	Dell DSA Problems	Work on recursion and graph traversal.
SAP	SAP DSA Problems	Master sliding window and hashmaps.
Qualcomm	Qualcomm DSA Problems	Focus on dynamic programming and backtracking.
Intel	Intel DSA Problems	Practice recursion and divide and conquer.
Samsung	Samsung DSA Problems	Work on sorting, searching, and dynamic programming.