

CS Career Roadmap for Beginners

Subtitle: A clear, step-by-step path to becoming a skilled developer and landing your first tech role.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>

Introduction

Starting a career in computer science can feel overwhelming with so many languages, tools, and paths. This roadmap simplifies the process, showing exactly what to learn, in what order, and how to apply it in real projects to get job-ready.

Step 1: Build a Strong Foundation

Every CS career starts with fundamentals:

1. Programming Basics

- Learn Python or JavaScript first.
- Practice variables, loops, conditionals, functions.
- Mini Exercise: Write a program that calculates the factorial of a number.

2. Data Structures & Algorithms (DSA)

- Focus on arrays, linked lists, stacks, queues, and basic sorting algorithms.
- Use platforms like LeetCode or HackerRank to practice.
- Mini Project: Build a simple text-based game using arrays and loops.

3. Computer Science Basics

- Learn about memory, CPU, OS basics, and networking fundamentals.
 - Helps in interviews and understanding real-world systems.
-

Step 2: Choose Your Path

Decide your main focus depending on career goals:

- **Frontend Development:** HTML, CSS, JavaScript, React.
- **Backend Development:** Node.js, Express, Python/Django, Databases.
- **Full-Stack:** Both frontend and backend skills.
- **Data/AI:** Python, Pandas, NumPy, Machine Learning Basics.
- **Mobile Apps:** React Native or Flutter.

Tip: Start with one path, then expand to complementary skills.

Step 3: Build Real Projects

Learning without doing won't get you hired:

- Start small: Portfolio website, To-Do app, Expense tracker.
- Intermediate: Blog platform, E-commerce site, Chat app.
- Advanced: Recommendation system, Task management SaaS, Mobile app with backend.

Mini Exercise: Plan one project for your chosen path, list features, tech stack, and deployment plan.

Step 4: Learn Tools & Best Practices

Employers value practical knowledge beyond coding:

- **Version Control:** Git & GitHub
- **Terminal & Command Line** basics
- **Debugging & Testing**
- **Clean Code Principles**
- **Basic Deployment:** Netlify, Vercel, Heroku, or AWS

Tip: Showcase these tools in your portfolio projects.

Step 5: Prepare for Interviews

Once you have skills and projects:

1. **Practice DSA regularly** – 1–2 hours daily.
 2. **Learn System Design Basics** – how apps scale, database design, APIs.
 3. **Mock Interviews** – Platforms like Pramp or interviewing.io.
 4. **Soft Skills** – Communication, problem explanation, teamwork.
-

Step 6: Build Your Portfolio & Network

- Create a portfolio website (link projects, GitHub, live demos).
 - Write short blogs explaining projects or concepts.
 - Join communities: Discord, LinkedIn, GitHub.
 - Apply for internships or freelance work to gain experience.
-

Key Takeaways

- Master **fundamentals** first, then specialize.
 - Apply knowledge in **real projects** to learn faster.
 - Learn **tools & best practices** to work like a professional.
 - Prepare for interviews **actively** with DSA, system design, and soft skills.
 - Build a portfolio that **proves your ability**, not just lists skills.
-

Visit [haas.dev https://dev-roast-app.vercel.app](https://dev-roast-app.vercel.app) for more step-by-step guides, resources, and career tips for CS students.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>