

Task 1: Comparison of Job Adverts

For this comparison, I chose two recent job adverts for Software Developer positions one posted by an NHS Trust in 2024 and the other drawn from the UK National Careers Service's software developer profile. Both offer detailed expectations for what it takes to succeed in this field, but the way they prioritize skills and describe the work environment tells us a lot about current industry standards. The NHS advert doesn't just look for technical know-how. It stresses the need for sharp problem-solving, keen analytical thinking, collaboration, and clear communication. There's a real emphasis on personal attributes, too adaptability, strong organisation, self-drive, and comfort working both solo and in a team really stand out. On the technical side, the role requires familiarity with multiple programming languages, understanding the basics of software development, and the ability to test systems reliably. There's also notable attention paid to working with stakeholders; developers need to explain their ideas, understand user needs, and help guide projects to completion. Turning to the National Careers Service profile, the core skill set overlaps quite a bit with the NHS advert. Analytical thinking, creativity, and problem-solving are foundational, alongside technical strengths such as coding and software design. This profile also points to the importance of understanding user requirements, creating robust software solutions, and maintaining those systems over time. What stands out in this description is the expectation that developers blend technical mastery with creative vision after all, building effective, user-friendly software demands both. Put side by side, these two adverts reinforce a clear picture of what employers actually want. Technical skills are non-negotiable, but they're matched by the need for good communication, attention to detail, an ability to work well with others, and a strong sense of accountability. Essentially, successful developers bring a balanced mix of hard and soft skills to their teams. Yes, you need to write clean code, but you also need to work through project challenges, explain your thought process, and juggle responsibilities across different departments or with non-technical colleagues. If you had to single out the most critical competency, though, it's problem-solving. No matter where you work, software development is all about finding practical solutions to complex challenges. Whether it's tracking down bugs, refining performance, or designing new features that users actually want.

Effective problem-solving powers innovation. It's not just about fixing what's broken but about imagining what could work better, and then bringing that vision to life. This is why employers put so much weight on this skill it's at the heart of the job.

Task 2: Self-Assessment and Evidence of Readiness

To get a realistic sense of where I stand as an aspiring software developer, I turned to a SWOT analysis, mapping out where my strengths lie, what holds me back, and where the road could lead. One major strength: my analytical mindset. Breaking apart a big task into manageable chunks isn't just something I can do, it's the way I naturally think. This helps enormously when the pressure's on to untangle a tricky bit of code or plan the build of a new feature from scratch. My academic grounding also supports me here; the fundamentals of programming, testing, and development tools come naturally thanks to coursework and lots of practice. Another thing in my favour is my motivation to keep learning. Technology's always on the move, and I actively search out tutorials, online courses, and real-world exercises to stay sharp. That hunger for improvement helps me adapt quickly and keep up with rapid industry changes. Plus, I'm comfortable in group projects. I make it a point to listen, explain my reasoning, and help move the team forward. There are, of course, some gaps to close. Most noticeably, I haven't spent much time in real-world development environments. Academic projects are a good start, but I'm aware that working with live users, real deadlines, and changing project requirements is a different game. Deep knowledge in areas like algorithms or data structures is another weak spot for now. Mastering these topics is essential for writing fast, efficient code, and I know I need to reinforce my understanding here. Time management is another area I can't ignore; sometimes, I overestimate how much I can get done at once, and that can hurt my ability to meet deadlines. On the opportunity side, the path forward is open if I'm proactive. Internships and work placements could give me the hands-on experience I'm missing. By building a portfolio, small apps, websites, or even contributions to open-source projects. I can showcase what I'm capable of when it comes time to apply for jobs. Online learning, from crash courses to coding challenges, makes it possible to fill in technical gaps and sharpen my skills on my

own schedule. But there are threats to consider. The job market is intense, packed with candidates who have years of experience or advanced credentials. Technology evolves quickly, too, and falling behind on new languages or frameworks could make it tough to stay relevant. Keeping up will demand constant learning and adaptability. Despite those challenges, I can point to real progress. My academic record shows I've built up essential programming, problem-solving, and critical thinking skills through both structured coursework and practical assignments. Personal projects like creating apps and websites on my own and back up those classroom lessons with hands-on experience. In group settings, I've developed communication and teamwork abilities, learning how to share ideas, respond to feedback, and make projects stronger by working together. Tutor and peer feedback has been invaluable, guiding me to sharpen strengths and tackle weaknesses head-on. Above all, my drive to keep learning sets me apart. Coding regularly and making use of free and paid resources, I know I'm always moving forward. There's still plenty of ground to cover, but my foundation is strong, and I'm committed to doing what it takes to make software development my career.

Task 3: Employment Goals and Personal Development Plan

Looking at my current position and aspirations, I've set both immediate and long-term goals to guide my journey in software development. In the short term, I want to strengthen my grasp on advanced technical topics, especially algorithms and data structures, since these underpin efficiency in every major project. I'll also work on growing my portfolio, building at least three substantial projects to showcase my skills, and seek out entry-level work or internships to get that vital hands-on exposure. Long term, my sights are set on securing a professional role in a respected organisation. I'm especially interested in carving out a niche, perhaps in web development, with the eventual aim of moving into senior-level responsibilities where I can take on larger projects and mentor newer developers. The appeal here is about more than job security, it's about finding a role where I can grow, make an impact, and keep learning as the field evolves. To make this happen, I've mapped out a personal development plan. First up, online courses deep-dives into

programming topics and algorithm design are already on my schedule. Each new course finishes with a project, and by the end of six months, I'll have at least three completed pieces for my portfolio. Regular practice, daily coding, timed challenges, and problem sets will reinforce that learning. I'll also apply for at least two internships or entry-level positions each month, refining my CV and interview skills as I go. And I'm actively seeking a mentor, someone who can give real-world feedback and guidance as I move forward. I'm tracking everything: which courses I complete, portfolio projects finished, interview feedback, and feedback from mentors or peers. The first six months focus on technical improvement. The next period, up to a year, is all about experience, whether through paid roles or substantial personal work. From there, it's about locking in that first full-time job and pushing toward my longer-term aspirations. By structuring my development and holding myself accountable to tangible milestones, I'm setting myself up for a successful transition into the software development field. It's a challenging path, but I'm determined to keep moving forward and build the career I want.