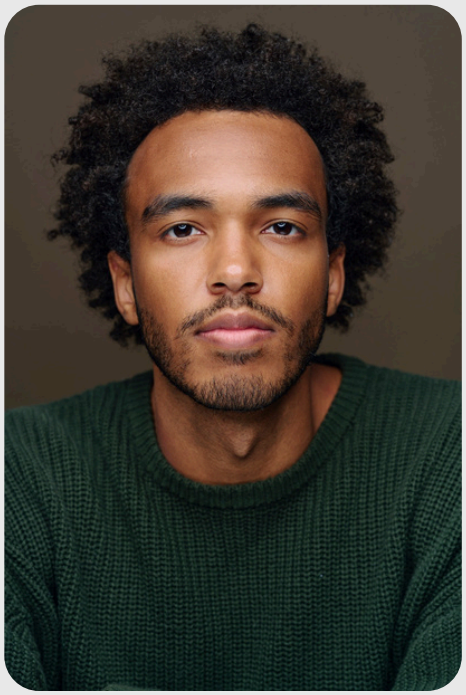
# *Resume Project Checklist to Land a $100,000+ Job in 2025*


BASHIRI SMITH

# Hey!



## A LITTLE ABOUT ME

My name is Bashiri Smith.

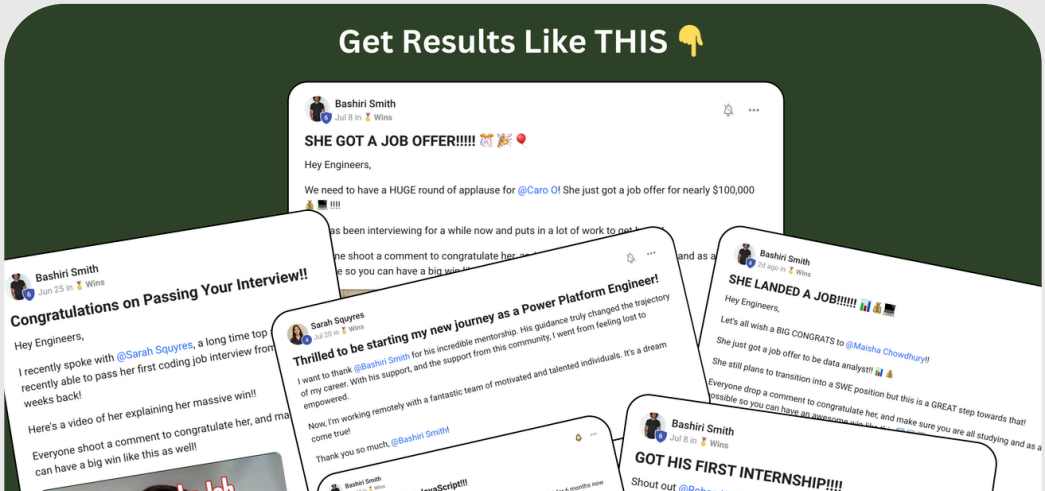
In just under 12 months & without a college degree or any work experience, I received 2 offers for more than $115,000 for my first job as a software engineer.

Since then, I have gone on to earn over $265,000p/y as a software engineer and I have also founded my own tech startup. (Interlade)

Throughout my journey to learn how to code, I've found that the unique information to ***ACTUALLY*** land a job is hard to come by regardless of if you go to college or a coding bootcamp.

My mission is to make it easy for anyone to change their life with coding!

**This guide is for aspiring coders who want to earn $100k+ as a software engineer ASAP.**



01

# Project Idea Prompt

Below is an example of a carefully crafted “prompt” you could give GPT-4, designed to elicit 10 unique, technically impressive, problem-solving coding projects that are robust enough to impress potential employers. Feel free to modify or refine this prompt to fit your specific needs or interests:

## Prompt:

Role: You are a highly experienced software engineer and tech career advisor with deep expertise in algorithm design, systems architecture, machine learning, distributed computing, and full-stack development.

Task: Propose 10 unique, technically impressive, problem-solving coding projects that a motivated software developer can build to significantly boost their portfolio and attract attention from top-tier tech companies. Each project should highlight diverse skill sets, showcase advanced problem-solving techniques, and have practical real-world value. For each of the 10 projects, include the following details:

1. Project Title
• Provide a concise, descriptive name.
2. Project Overview
• Give a high-level summary of what the application or system does and why it is impressive or valuable.
3. Key Features & Technical Complexity
• List the standout functionalities or unique selling points.
• Highlight the core problem-solving aspects, data structures, or algorithms involved.
• Emphasize the technical depth, such as concurrency, distributed systems, or advanced computations.
4. Technologies & Tools
• Specify relevant programming languages (e.g., Python, Go, TypeScript, Rust).
• Mention any major frameworks, libraries, or platforms (e.g., React, Flask, Node.js, Kubernetes, TensorFlow).
• Include any additional tools or cloud services (e.g., Docker, AWS, GCP).

02

# Project Idea Prompt

5. Challenges & Learning Opportunities
• Outline the main difficulties a developer would face and how they can tackle them.
• Note how building this project will enhance one's skill set and preparedness for technical interviews.
6. Potential Extensions
• Suggest ways to scale up, optimize further, or add more advanced features.
• Provide ideas for how this project could become even more impressive.
Additional Instructions:
• Ensure each project idea is unique and original. Avoid overlapping features or use-cases.
• Aim for a wide range of technologies, including web, mobile, data engineering, machine learning, systems design, and more.
• Assume the developer has intermediate coding skills but is looking to stretch their capabilities to a more advanced level.
• Where relevant, indicate how the project can be tested or deployed for maximum visibility to hiring managers.
• Make the projects sufficiently challenging to guarantee that a successful build will stand out during interviews and on GitHub/portfolio sites.
Format: Present your answer as a numbered list of 10 project ideas, with each idea structured using the headings outlined above (Project Title, Project Overview, Key Features & Technical Complexity, Technologies & Tools, Challenges & Learning Opportunities, and Potential Extensions).

Tip: When you submit this prompt to GPT-4, focus on the completion style of request (e.g., "Continue" or "Propose your list..."). This will help GPT-4 generate a comprehensive, clearly organized answer that meets all the specified requirements.

03

# What is a "Resume Quality" Project

**Qualities of a Resume-Level Project**

1. Complexity: Demonstrates a high level of technical skill and problem-solving ability.
2. Scalability: Shows consideration for scalability and performance.
3. Usability: Has a well-designed user interface and user experience.
4. Documentation: Includes comprehensive documentation for both users and developers.
5. Testing: Includes unit tests, integration tests, and possibly end-to-end tests.
6. Version Control: Uses version control with a well-documented commit history.
7. Deployment: Is deployed and accessible online, showcasing real-world usability.

**Checklist to Pick Your Own Resume-Level Project**

- Identify a Problem: Choose a real-world problem that interests you.
- Research Solutions: Investigate existing solutions and identify gaps or areas for improvement.
- Define Scope: Clearly define the scope of your project to ensure it is achievable within a reasonable timeframe.
- Plan Features: List the features your project will have and prioritize them.
- Design Architecture: Plan the architecture of your project, considering scalability and maintainability.
- Develop Incrementally: Build your project incrementally, testing each feature as you go.
- Document Thoroughly: Write comprehensive documentation for your project.
- Test Rigorously: Implement and run tests to ensure your project is robust and reliable.
- Deploy and Maintain: Deploy your project online and maintain it, fixing bugs and adding improvements as needed.

04

# Find & Work with Teammates

**The Importance of Teamwork**

Working on a team is crucial for creating a resume-worthy project. Teamwork not only simulates real-world software development environments but also offers numerous benefits:

- Effective Communication: Learn to articulate your ideas and understand others'.
- Diverse Perspectives: Gain insights and solutions from different viewpoints.
- Task Division: Leverage individual strengths to tackle various project components efficiently.

**Finding Teammates in a Community**

To help you find and team up with other members of the community, follow these steps:

1. Join this FREE software engineering community: https://www.skool.com/become-a-software-engineer/about
2. Introduce Yourself
    - Post a brief introduction in the community (under the TeamUp category), including your skills, interests, and the type of project you want to work on. This helps others with similar goals and complementary skills find you.
3. Create a Project Board
    - We have set up a dedicated space where you can pitch your project ideas and look for teammates. Each post should include:
        - Project Description: What the project is about.
        - Required Skills: What skills are needed to complete the project.
        - Commitment Level: The expected time and effort.
4. Participate in Networking Events
    - Join our regular virtual meetups and networking events. These sessions provide opportunities to discuss your projects and form teams. Keep an eye out for announcements about these events.
    - Meetups to be scheduled!
5. Utilize Collaboration Tools
    - We recommend using tools that facilitate teamwork, such as:
        - GitHub: For version control.
        - Trello: For project management.
        - Slack: For communication.

By following these steps and engaging with your peers, you'll be well on your way to creating a project that stands out on your resume. Teamwork makes the dream work!

05

# Get Your Project Idea Verified

Before you start working on your resume-level project, it's crucial to ensure that your idea meets the necessary criteria and will effectively showcase your skills to potential employers. To get your project idea verified, you can reach out to me directly.

## How to Verify Your Project Idea

1. Outline Your Idea: Write a brief outline of your project idea, including the problem it addresses, the features you plan to implement, and the technologies you intend to use.
2. Contact Me: You can DM me or use the link on the last page to book a call with me.
3. Receive Feedback: I will review your project idea and provide feedback, suggestions, and verification to ensure it aligns with the qualities of a resume-level project.

## Why Verification is Important

Verifying your project idea helps you:

- Ensure your project demonstrates a high level of technical skill.
- Align your project with industry standards and expectations.
- Receive guidance on best practices and potential improvements.
- Increase your chances of impressing potential employers with a well-crafted project.

*Don't hesitate to reach out for verification; it's a critical step in your journey to becoming a successful software engineer.*

06

# Still Need Help?

I know learning how to code on your own can be extremely daunting and just make you feel so lost. I felt the same way while I was on my journey. So now I help aspiring coders who feel lost land their first $100,000+ coding job!

This mentorship is ONLY for serious people who want to land a job ASAP.

If this sounds like you we can set up a call!

On the call, you and I will come up with a custom plan of action personalized for you so you can see exactly how to become a $100,000+ software engineer.

This call is completely FREE and is guaranteed to help you get crystal clear on your next steps.

Fill out the form below and I will reach out to you ASAP!

https://api.leadconnectorhq.com/widget/form/8VbHZnjfk0r1SxbmHQ5r