To ensure my code is functional and secure, I prioritize testing and follow secure coding practices from the start. I use unit and integration tests to check that each part works as expected and meets specific requirements. Additionally, I follow best practices for security, like input validation and secure data handling, and make sure to review any potential vulnerabilities regularly.

When it comes to interpreting user needs, I find it helps to spend time understanding the target audience and their goals. I ask questions, gather feedback, and create a list of user stories or requirements to guide development. Regular check-ins with users (or stakeholders) are essential for making sure the program stays aligned with their expectations, and I’m always open to refining features based on this feedback.

For designing software, I start by mapping out a clear structure, often sketching an outline of the system’s core components and interactions. I also think through how each part will work together, considering scalability and maintainability, and choosing tools or frameworks that best fit the project’s goals. Planning before diving into code helps create a more organized, adaptable design, making future updates or feature additions much smoother.