REFLECTION PROCESS FOR SELF-AWARENESS

Step	Action	Example
Identify specific episodes	• Take memorable projects, tasks, or interactions you had. • Choose examples from different spheres: work, school, sports...	• Group Project at university developing a small web application with a team. • Open-Source Contribution where you fixed bugs or added new features to a repository. • Hackathon Participation creating a prototype under time pressure. • Side Project building a personal website or mobile app to learn a new technology.
Ask targeted questions	• "Which tasks did I enjoyed the most?" • "Where did I encounter the most challenges?" • "How did I respond setbacks or obstacles?" • "Did I enjoy working alone or in a group?"	• You might realize you felt very motivated when working on front-end development (designing user interfaces). • Perhaps debugging complex back-end issues was draining or exhilarating. • If there was a technical roadblock, did you enjoy researching solutions or prefer seeking mentorship from a senior engineer?

REFLECTION PROCESS FOR SELF-AWARENESS

Step	Action	Example
Look for patterns	• Observe common themes from your reflections. • Identify whether you enjoy solving complex algorithmic challenges, designing user interfaces, or collaborating in a team.	• Pattern 1: You frequently found yourself taking the lead in code reviews or mentoring peers, indicating a natural inclination for collaboration and teaching. • Pattern 2: You enjoyed exploring new frameworks or libraries, suggesting you thrive in innovative or exploratory environments.
Document your insights	• Keep a journal or spreadsheet to record observations. • Note each significant task, what you liked/disliked, and the key strengths you demonstrated.	• Create columns labeled: "Project Name," "Energizing or Draining," "Technologies Used," "Skills Demonstrated," and "Key Takeaways." • Example entry: Hackathon Project; "Energizing: quick prototyping and teamwork. Draining: lack of clear structure. Skills Used: React, team coordination. Takeaways: need clear task division."

SEEKING EXTERNAL FEEDBACK

Step	Action	Example
Identify Trustworthy People	• Choose people who have observed you in coding, debugging, teamwork, or project leadership. • Prioritize those who can provide constructive, honest insights.	• Teammates: people who collaborated on a recent sprint. • Former Manager: Someone who supervised your performance • Mentor: A senior engineer with whom you've regularly discussed career progression.
Prepare Specific Questions	• Right down open-ended questions that address your technical abilities, communication style, and teamwork. • Avoid vague requests like "What do you think of me?"	• "How would you rate my approach to debugging complex issues?" • "Which areas of my coding style need the most improvement?" • "In team discussions, do I communicate my ideas clearly?" • "How can I add more value during code reviews?"

SEEKING EXTERNAL FEEDBACK

Step	Action	Example
Choose the Right Format	• Decide if you want to get feedback via a 1:1 conversation, email, or an other way (phone, survey, text). • Pick the one that is the best for you and the person you are considering.	• One-on-one Chat: Schedule a 15-minute Zoom call • Email • Whats'App text
Listen Actively & Ask for Clarification	• When receiving feedback, stay open-minded, avoid being defensiveness. • Ask follow-up questions if the feedback is unclear or too general.	• Someone says, "Your code is sometimes hard to read." You could answer: "Could you show me an example or explain which aspects make it unclear? Is it variable naming, lack of comments, or something else?"

SEEKING EXTERNAL FEEDBACK

Step	Action	Example
Identify Relevant and Constructive Feedback	• Focus on feedback that is concrete and actionable. • Be cautious about overly vague or unconstructive criticism.	• – If a colleague says, "Your Git commits need more detail," that's concrete (you can fix it by adding descriptive commit messages). • – "You're just not a team player" is vague—probe further: "Could you explain what actions led you to feel that way?"
Take Action on Feedback	• Identify 2–3 improvement areas and create a plan with specific goals or tasks. • Incorporate changes into your daily workflow or coding practices.	• Feedback: "Your pull requests are too large and hard to review." • Action Plan: Create smaller, more frequent commits, and request reviews more regularly.
Follow Up	• After applying changes for a few weeks, reconnect with the same people for updated feedback.	• Example: "Last month, you mentioned my SQL queries could be optimized. I've done some reading on indexing and adjusted my approach. Thank you so much for your insights and feeback