YOUR LOGO HERE

[INSERT PROGRAMME NAME]

CANDIDATE INTERVIEW GUIDE – SOFTWARE TESTER

[INSERT DATE]

Purpose:

To assess a candidate's **foundational technical understanding**, **analytical thinking**, and **potential to learn software testing principles**.

Part 1: Technical Questions

(Evaluate understanding of software concepts and ability to apply logic to testing scenarios)

1. **Software Fundamentals**
    - What are the main stages in the **software development life cycle (SDLC)**?
    - What's the purpose of **testing** in software development?
    - How would you define **a bug** in your own words?
2. **Programming & Logic Awareness**
    - When writing code, how do you ensure it behaves as expected?
    - What is the difference between a **syntax error** and a **logical error**?
    - How would you test whether a function returns the correct output?
3. **Understanding User Behavior**
    - How would you test a simple **login form** (username and password)?
    - What kind of issues might users face that developers often overlook?
    - How would you handle testing if you don't have access to the full codebase?
4. **Basic Tool Familiarity & Curiosity**
    - Have you used tools like **Git**, **Postman**, or **Excel** for validation or checking results?
    - If you were asked to learn a testing tool (e.g., Selenium or Jira), how would you approach it?
    - What's the difference between **manual testing** and **automated testing**, as you understand it?

Part 2: Cognitive & Analytical Questions

(Assess reasoning, problem-solving, attention to detail, and communication skills)

1. **Analytical Thinking**
    - If an app crashes occasionally, how would you begin investigating the cause?
    - How would you explain to a developer that something isn't working as expected?
    - How do you decide when a program is "ready" to be released?
2. **Attention to Detail**
    - Describe a time you found a small mistake (in code, documentation, or data) that others missed.
    - How do you keep track of details when working on multiple small tasks?
    - Why do you think attention to detail is important in software testing?
3. **Problem-Solving & Learning Mindset**
    - How would you approach a task you've never done before?
    - When you encounter a bug or issue, what steps would you take to understand it?
    - Tell me about something new you taught yourself recently — how did you go about it?
4. **Teamwork & Communication**
    - How would you give feedback to a developer about an issue you discovered?
    - Describe a time when you worked in a team to solve a problem.
    - Why do you think good communication matters between developers and testers?