YOUR LOGO HERE

[INSERT PROGRAMME NAME]

CANDIDATE INTERVIEW GUIDE – SOFTWARE DEVELOPER

[INSERT DATE]

Entry-Level Software Developer Interview Guide

Purpose:
To assess both **technical capability** and **cognitive thinking skills** of candidates applying for an entry-level software developer role.

Part 1: Technical Questions

(Assess understanding of programming fundamentals, problem-solving, and applied knowledge)

1. **Programming Basics**
    - Explain the difference between a **compiled** and an **interpreted** language.
    - What are **variables**, and how are they used in a program?
    - What is the difference between **arrays** and **lists**?
2. **Object-Oriented Programming (OOP)**
    - Describe the four main principles of OOP.
    - Give a real-world example of **inheritance** or **polymorphism**.
    - How would you decide when to use a **class** vs. a **function**?
3. **Data Structures & Algorithms**
    - What is the time complexity of a **binary search**?
    - When would you use a **hash map** vs. a **linked list**?
    - Write or explain logic to reverse a string or check if a word is a palindrome.
4. **Databases**
    - What's the difference between **SQL** and **NoSQL** databases?
    - How would you retrieve all users from a database table called Users?
    - Explain what **normalization** is and why it's important.
5. **Web & Version Control**
    - What happens when you type a URL in a browser and press Enter?
    - Why is **Git** used, and what does the command git commit do?
    - Describe the difference between **frontend** and **backend** development.

Part 2: Cognitive & Problem-Solving Questions

(Assess reasoning ability, adaptability, and creative thinking)

1. **Analytical Thinking**
    - You're given a bug you can't immediately fix — how would you approach solving it?
    - How would you optimise slow code without knowing where the problem is?
    - Explain how you'd break down a large project into smaller tasks.
2. **Learning Agility**
    - Tell me about a time you had to learn a new language or framework quickly.
    - How do you stay up to date with new technologies?
    - If you were assigned a task you've never done before, what steps would you take?
3. **Communication & Teamwork**
    - How would you explain a technical concept to a non-technical person?
    - Describe a time you disagreed with a teammate — how did you resolve it?
    - What does good collaboration look like to you in a development team?
4. **Creativity & Adaptability**
    - If you could improve one aspect of a product you've used recently, what would it be and why?
    - Describe a time when you had to change your approach midway through a project.
    - How do you handle feedback or code reviews?