- 📝 Interview Preparation Instructions
  - 🔷 Objective
    - Understand the code you have written (use the self-evaluation checklist)
    - Documentaion requirements
    - Logging Requirements
- Sample Project
  - Face Recognition Project
  - Gesture Recognition

📝 Interview Preparation Instructions

🔷 Objective

This document provides **step-by-step instructions** for candidates to prepare for the interview.

You will be evaluated on:

1. Code analysis and understanding of real-world repositories.
2. Documentation of architectures and component flows.
    1. Quality and clarity of diagrams and Markdown documentation.
3. Functional Implementation and meaningful extensions.
4. Logging, performance tracking, and code readability.

Understand the code you have written (use the self-evaluation checklist)

- **Understand the code you have written** — including all packages, libraries, classes, and functions used.
- Before the interview, walk through the provided **self-evaluation checklist document** and mark off each item to confirm coverage.
- Be able to explain the **overall objective** of your solution.
- For each function and class, identify and note:

    - **Inputs** and **outputs**
    - Purpose and logic implemented

- You are ***not* expected to memorize everything**. You may refer to your notes during the interview, but ensure you can walk us through the structure and logic clearly.

- Questions will focus on the **fundamental concepts** implied by the features implemented in your code.

Documentaion requirements

1. A **high-level architecture diagram** of your codebase.
2. **Component-level details** describing:
    - Inputs received by each component
    - Processing performed within each component
    - Outputs generated by each component

Logging Requirements

- Please enhance your codebase with a dedicated logger (e.g., using Python's logging module).
- Your logs should include:Metadata about inputs and outputs of key functions.
- Object data length and any significant object changes.
- Execution time and space/memory usage for major processes and functions.
- Time and space complexity observations wherever feasible.
- Exceptions, errors, and any unusual behavior during code execution.
- Prepare your log files and notes for discussion; we will cover how logging improves code reliability and tracking.

This preparation will help us have a productive and in-depth discussion during the interview, ***please feel free to reschedule the interview if the preparation is not sufficient*.**

Sample Project

Refer to the sample project's documentation provided below, and follow a similar structure and organization for your codebase.

Face Recognition Project

Github Repo Link	**Link**
Code	Link
Documentation - Architecture diagram	Link
Documentation - Component Diagram	Link

Gesture Recognition

Github Repo Link	**Link**
Code	
Documentation - Project overview	Link
Documentation - Component Diagram	Link