### Task: 1.58-bit BitNet Model fine-tuning on Mobile GPUs (Adreno & Mali) Using llama.cpp

**Problem Statement**

The team has pre-trained a Small Language Model (SLM) with 2 billion parameters using a **1.58-bit BitNet architecture** for efficient mobile deployment. The goal is to **enable fine-tuning** using the llama.cpp framework on smartphones equipped with Adreno and Mali GPUs.

## Objective

Develop a comprehensive write-up plan to **enable 1.58-bit BitNet model architecture fine-tuning** using the llama.cpp framework on smartphones equipped with Adreno and Mali GPUs.

Furthermore, you must also build a proof of concept (PoC) implementation to develop a **1-bit matrix multiplication kernel** using **Vulkan for a single-layer inference pass** in a **standard GPT transformer-based** model. The implementation must demonstrate successful operation on **at least one GPU platform (NVIDIA, AMD, Intel, Adreno, Mali or other GPU)**. You must include a test harness with sample input/output validation.

Your proposal must, at a minimum, address the following aspects:

1. How to design BitNet fine-tuning kernels for 1.58-bit weights optimized for Adreno or Mali GPUs using Vulkan as the backend.
2. Describe how you would extend and modify the llama.cpp framework to support fine-tuning on Adreno and Mali GPUs across cross-platforms (i.e., Android, Windows, and Linux). Clearly identify the files and modules in llama.cpp that would require modification or extension.
3. Detail the profiling methodology for Adreno and Mali GPUs, including tools and techniques for collecting fine-tuning performance metrics and defining concrete optimization targets.
4. List all frameworks, libraries, and SDKs required for development, deployment, and performance tuning.
5. Explain the kernel- and shader- level implementation optimization strategies that will be employed.
6. Define success criteria and evaluation benchmarks for fine-tuning performance on Adreno and Mali GPUs.

7. Identify key smartphone hardware constraints and propose strategies for achieving low-power fine-tuning, minimizing memory bandwidth usage, and reducing compute overhead.
8. Present a phased schedule that includes milestones and deliverables for each stage of the project, as well as progress tracking checkpoints.
9. Highlight potential risks and outline contingency plans or mitigation strategies.
10. State all underlying assumptions and critical dependencies that may influence project execution.
11. Include any other information or constraints relevant to the proposal.

## Proof of Concept (PoC) Development

1. Develop a **1-bit matrix multiplication kernel using Vulkan for a single-layer inference pass** in a **standard GPT transformer-based model**. The implementation must demonstrate successful operation on **at least one GPU platform** (NVIDIA, AMD, Intel, Adreno, Mali or other GPU). You must include a test harness with sample input/output validation.

   **Bonus:** if you can show your integration with llama.cpp

## Deliverables

Submit both:

- **Scripts containing your source code**
- **A technical plan report (PDF)**

**Due Date:** Submission must be completed within **10 days**.

Be prepared to **present and justify your approach** in an upcoming interview.