1 Team Plan and Work Distribution

This project was a collaborative effort between two members, focusing on a high-performance CUDA AutoEncoder. The workload was balanced between **System Architecture/Optimized Kernels** and the **Computational Pipeline/Naive Implementations**.

1.1 Work Distribution

Member	**Primary Roles**	**Key Contributions**
Nguyen Long	System Architecture & GPU Optimization	Designed the core framework and smart memory management. Implemented Phase 2.3 (Shared Memory Conv2D) and Phase 3 optimization strategies. Developed core layer logic, bias addition, and performance profiling.
Nguyen Minh Nhat	Computational Engine & GPU Implementation	Implemented Phase 2.2 (Naive GPU kernels) and Phase 2.4 (Implicit im2col and GEMM-based convolution). Developed ReLU, Upsampling, and MaxPool kernels. Integrated the SVM training and evaluation pipeline.

Table 1: Summary of Team Responsibilities

1.2 Task Breakdown and Timeline

The development followed a four-phase approach, concluding with the final presentation on December 21, 2025.

- **Phase 1: Foundation & Research (Oct 31 – Nov 28)**
  - Initial project setup, CMake configuration, and image loading.
  - Implementation of baseline CPU convolution.
- **Phase 2: Core GPU Engine (Dec 3 – Dec 10)**
  - **Minh Nhat:** Developed Phase 2.2 Naive GPU implementation (1:1 thread mapping).
  - **Long:** Established Tensor classes, layer protocols, and memory reuse logic.
  - Implementation of ReLU and MaxPool CUDA kernels.
- **Phase 3: Advanced Optimization (Dec 11 – Dec 18)**
  - **Long:** Implemented Phase 2.3 (Shared memory tiling and kernel-level batching).
  - **Long:** Integrated warp-level shuffle reductions for gradient computation.

    - **Minh Nhat:** Developed implicit im2col mapping and GEMM-style kernels.
- **Phase 4: Finalization & Evaluation (Dec 19 – Dec 21)**
    - Completion of Phase 2.4 GEMM-based convolution for high-performance training.
    - **Minh Nhat:** Finalized SVM feature evaluation and classification logic.
    - **Team:** Documentation, README completion, and Video Presentation.

1.3 Contribution Percentage

Student Name	**Responsibility**	**Contribution %**
Nguyen Long	Architecture / Shared Memory / Warp Reductions	50%
Nguyen Minh Nhat	Naive GPU / GEMM / im2col / SVM	50%

Technical Note: *The architectural design utilizes shared memory to allow layers to exchange data efficiently, while the implicit im2col implementation optimizes memory bandwidth, resulting in an end-to-end speedup of* $\approx 25x$ *over the CPU baseline.*