CUDA SAXPY Benchmarking

What I learned?

- I learned how to setup a CUDA environment even without local GPU hardware.
- I used Google Colab with free access to NVIDIA T4 GPUs (as suggested).
- I modified CUDA code to benchmark the SAXPY algorithm (Single-precision A·X Plus Y).
- I used problem sizes ranging from 215215 to 225225.
- I profiled kernel execution time as well as total execution time.
- I learned how memory transfer between CPU and GPU affects overall performance.
- I gained experience with cudaEvent for accurate timing measurements.
- I visualized performance trends with Matplotlib and Jupyter Notebook.

My Insights:

- CUDA accelerates large data sizes very efficiently.
- Memory transfer (host ↔ device) overhead is substantial for small sizes.
- Profiling both total time and kernel time helps to separate compute vs transfer bottlenecks.
- This gives insight for deciding when hardware acceleration is worth it for a workload.

LLM prompts I used:

Prompt 1:
"Generate a CUDA SAXPY kernel where I can benchmark multiple N sizes and profile both kernel time and total time using cudaEvent."

Prompt 2:
"Give me Matplotlib Python code to read CSV results of SAXPY benchmarks and plot bar graphs for both kernel time and total time."

Prompt 3:
"Help me install CUDA kernel execution on Google Colab, starting from scratch."