1. **Proposal title:** Tensor Execution engine on GPU

2. **Introduction:**
   The goal is to optimize the FasTensor tensor computing library to work efficiently on GPUs. Currently, it only works well on CPUs. This optimization would enable FasTensor to efficiently compute tensor contractions on GPUs while maintaining the structure-locality of tensor data. The project aims to create custom-defined computational operations (kernels) on GPUs to achieve this goal. This optimization is essential for many scientific applications, including advanced AI model training.

3. **Project goals:**
   a. The main objective of this project is to optimize FasTensor to work efficiently on GPUs, which will benefit a wide range of scientific applications that require tensor-based computing. The community will benefit from a more flexible and efficient tensor library that can run on different hardware platforms.
   b. The expected deliverables from this proposal include a working implementation of FasTensor on GPUs, a report on the performance of the execution engine, and documentation of the execution mechanism.
   c. Future work may involve further optimization of FasTensor for different types of hardware, as well as integration with other scientific computing libraries.

4. **Implementation plan:**
   a. The project will use a combination of programming languages, including C++ and CUDA, to implement the mechanism for moving user-defined computing kernels onto GPUs. My plan will be to interact with the mentor and other members of the open-source community through online discussions and code reviews.
   b. The technical elements of the project include implementing the necessary algorithms and data structures to maintain the structure-locality of tensor data on GPUs, as well as developing efficient memory access patterns and data movement strategies.
   c. The main challenges I anticipate include ensuring compatibility with different GPU architectures, managing memory and data movement efficiently, and ensuring that the execution engine can handle a wide range of user-defined computing kernels. Proposed solutions include using profiling and benchmarking to identify performance bottlenecks and optimizing the implementation accordingly.
   d. The project timeline includes an initial period of research and planning, followed by implementation and testing, and a final period of documentation and reporting. The planned deliverables include a working implementation of FasTensor on GPUs, performance benchmarks, and documentation of the execution mechanism.

5. **Project Timeline:**

a. **Week 1:** Research existing literature on tensor computing and GPU optimization techniques. Familiarize with Fastensor's codebase and identify areas that need to be modified for GPU optimization.
b. **Week 2-4:** Develop a plan for optimizing Fastensor on GPU and start implementing the changes.
   Deliverable: A detailed plan outlining the specific changes to be made to Fastensor to optimize it for GPU computing. Modified Fastensor codebase that runs efficiently on GPU
c. **Week 5-7**: Test and debug the modified code on a variety of GPU hardware and software configurations
   Deliverable: A report detailing the results of testing the modified Fastensor code on various GPU hardware and software configurations, including any bugs or issues encountered and how they were resolved.
d. **Week 8-10:** Evaluate the performance of the optimized Fastensor against existing tensor computing libraries on GPU.
   Deliverable: A report comparing the performance of the optimized Fastensor against existing tensor computing libraries on GPU, including any speedup achieved.
e. **Week 11-12:** Document the modifications made to Fastensor, including any performance improvements achieved and any challenges encountered.
   Deliverable: A comprehensive documentation of the modifications made to Fastensor for GPU optimization, including any challenges encountered and how they were addressed.

6. **Biographical information**:
   Name: Rishabh Ballabh Singh Email: rbs7261@nyu.edu
   Phone: 917-375-2150
   LinkedIn: https://www.linkedin.com/in/rishabh-singh-274120169/ Github: https://github.com/ris0801/
   Current Affiliation: New York University, Tandon School of Engineering
   Relevant experience and educational background:
   Experience:
   Senior Software Engineer at Arcesium(3 years):
   At Arcesium, I worked with the team on full-stack development, extensively on Java, Spring, MySQL for backend, ReactJS for frontend development, and AWS for building modern software solutions spanning the investment cycle. One of my most noteworthy projects was to increase the efficiency of reading data from upstream applications. Data was being read as quickly as within 5 mins compared to as worse as 6 hours before. It also reduced the number of data read issues by 99%. As a senior engineer, I supervised two other software engineers in developing a sustainable application as a team. We built a machine-learning suggestive application for the annual hackathon to help the Financial Operations team with highly positive reviews.

   Software Engineer Intern at Samsung(6 months):

At Samsung, I worked in the Voice Intelligence team, where we succeeded in deploying General Adversarial Networks as part of the Natural Language Generation Team using Python, Keras, TensorFlow, and scikit-learn after reading numerous research papers. We constructed a primary pipeline with 88% accuracy from datasets of queries and expected responses requested from Bixby over the years. Bixby was generating answers for the first time instead of replying from a fixed set of responses.

Education:
Masters in Computer Engineering (New York University, Tandon School of Engineering, Sep 2022 - Present)
Relevant coursework:

- Machine Learning, Deep Learning, High Performance Machine Learning (HPML), Artificial Intelligence, Data Structure and Algorithm

Technical interests and strengths:

- **Programming Languages:** Python, Java, C++, MATLAB
- **Technical**: Deep Learning, Tensorflow, Pytorch, Keras, Scikit-learn, GPU architecture, CUDA programming, GPU memory management, Git, SQL, Pandas, AWS, Linux