**Problem:** [import torch: How to fix OSError WinError 126, error loading fbgemm.dll or dependencies](https://stackoverflow.com/questions/78114412/import-torch-how-to-fix-oserror-winerror-126-error-loading-fbgemm-dll-or-depen)

Solution: Pasted **libomp140.x86\_64.dll in Windows/System32**

<https://www.dllme.com/dll/files/libomp140_x86_64/00637fe34a6043031c9ae4c6cf0a891d/download>

and torch reinstallation with NumPy downgrade

**Problem**: Unable to link React app with Flask Backend

Access to XMLHttpRequest at 'http://127.0.0.1:5000/upload' from origin 'http://localhost:3000' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.

Solution: Solved it by keeping the CORS(Cross Origin Resource Sharing) configuration along with the route.py file containing the API Functions

Adding the required headers manually using after\_request Also worked but the inbuilt approach is more efficient.

**Problem:** Handling large files for summarization and stopping the process when needed.

Solution: Implement text chunking with threading and use a stop button to halt processing. The button's visibility and functionality are managed based on file size and process status.

**Problem:** Finding it hard to add stop processing button separately on large files for a splitting and processing and acceptable files which are shorter

Solution: Implement conditional rendering for the "Stop Processing" button based on file size and processing status, and manage button text and state based on the process status and file size.