This is a simple test to see whether or not our implementation works. Firstly we test if text works, if form characters that get introduced by converting to pdf get eliminated, special characters like ,, ., -, “”, $, &, etc. get eliminated.
Then if this works, the next step would be to get rid of word that are not of interest in a phrase net. Words like the, a, an, by, etc. however this step need to keep the word to be searched for in mind. It would not be smart to eliminate the word the user wants to look for.
Speaking of, the next step is to look for the word the user has specified and then return frequency of word occurrence for now.
After this a benchmark with a larger file should give us an idea, if what is implemented so far has the potential to work fast for large documents. If not …. to be discussed.
As of writing this, the base for the http server in python exists and works with GET and POST requests :) .

Lastly, we should also look into what happens if the user selects a pdf-document that is not exclusivity text. Things like pictures, tables, or simply a power point converted to pdf could throw our project into problems. But that is for the future….