Comparable app: Camscanner.com

What is needed: A python script that takes parameters via command line and returns 2 different possible results. **We only want the python script,** we will be communicating with the python script from a php script.

Result 1) returns a preview image and the coordinance of edges (the edges of a 4 sided document). This is taken from the python output and sent to a user

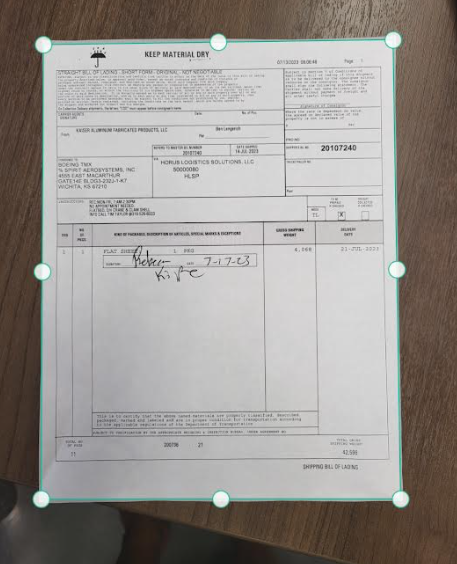
Result 2) after confirmation of the preview + modified edges are returned to the python script, its to produce a camscanner-like pdf file.

Attributes of the expected result:

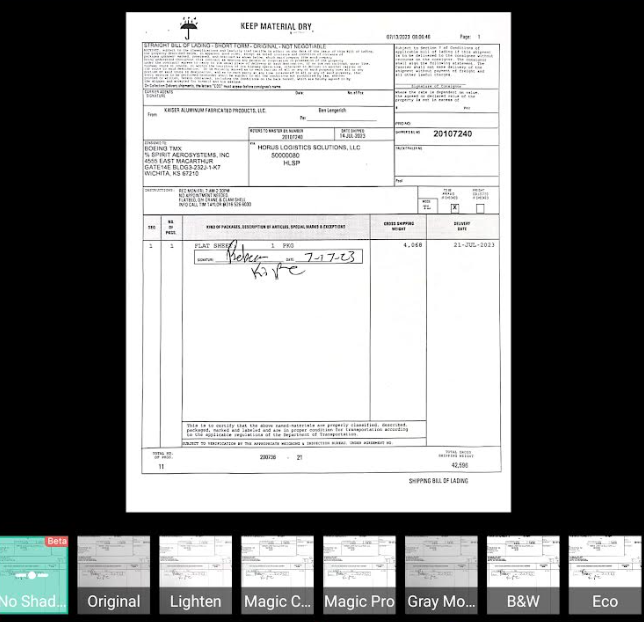
1. White background. One primary reason to use camscanner is so that the department that prints this document is not wasting a ton of black ink
2. Perspective flattened and cropped to the edges of the document
3. Black and white, “scanned-in” look.
4. Clear, legible document with minimal noise
5. Some effort to optimize PDF file size. Some effort to minimize the size of the preview while maintaining quality.

**Step-by-step what the user will do**

1. We take a photo from via mobile on our website, the photo is sent via params to the python script, the python script returns the **Detected Edges**



1. The edges are returned to the php script running the python, where user will maybe modify the edges to their liking (this entire interface will be made by us, not you)
2. These edges are sent back to python script so the image may be **cropped & perspective-flattened**
3. A preview is sent back to the user with it cropped + flattened + default filter (Black & White scanner look). It may be necessary to provide additional filters to clean/darken/lighten the filter. Please allow these settings to be passed in via parameters.



1. Upon confirmation the final product is then converted to a PDF

**A “Document session”**

We will need the concept of a document session which we can use a hash for (perhaps the hash would also be the file names used) so that we can scan multiple pages through this process and the end result, they will end up in the same PDF file. It will be the webapps responsibility to provide this hash to the python script via parameters. And if this hash matches the files already produced by the python script, then the final PDF will include all of those pages.

I am thinking we would end up with files like

E36325cf86fce71d878824fa76ba96a1da2158d9\_1\_preview.jpg

E36325cf86fce71d878824fa76ba96a1da2158d9\_2\_preview.jpg

E36325cf86fce71d878824fa76ba96a1da2158d9\_1\_cropped.jpg

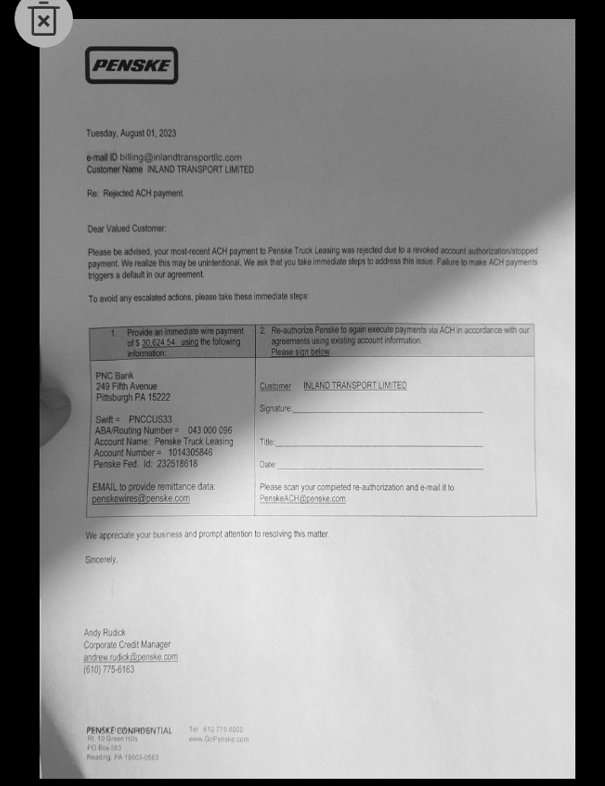
E36325cf86fce71d878824fa76ba96a1da2158d9\_2\_cropped.jpg

E36325cf86fce71d878824fa76ba96a1da2158d9.pdf

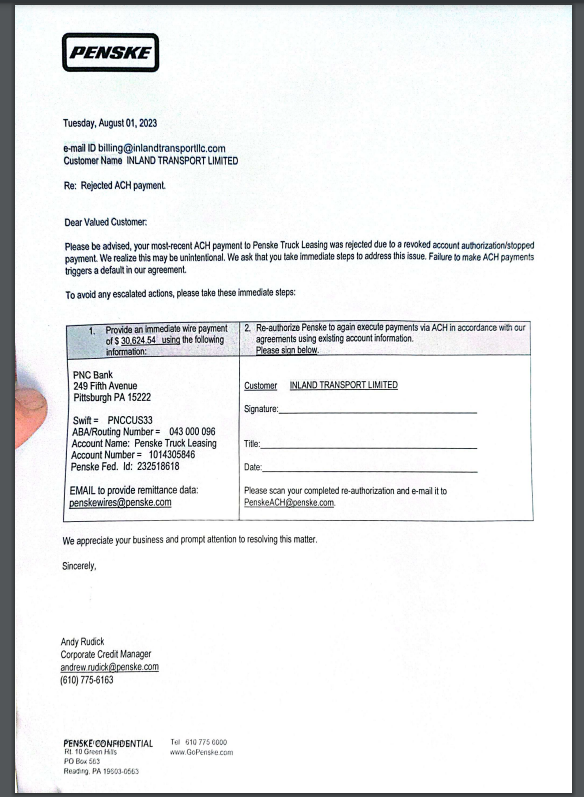
(containing both pages 1 and 2, of the cropped+flattened+filtered result)

**Real world example + Shadow removal:**

These photos are being taken on the road and often contain shadows that need to be removed.



**Expected Result:**



Please feel free to ask [loadcompanydev@gmail.com](mailto:loadcompanydev@gmail.com) for further clarification