**Documentation of Data masking**

Team number - 36

Team name - The Royal Flush

Respected Team of JA Assure,

This document explains on how to set up and run the application which has been developed by the team, The Royal Flush. We are extremely delighted to survive all the rounds and now we would love to present the application we developed to the team of JA Assure.

**Contents: -**

1. **Pip commands to set up the necessary environment and dependencies for the given Flask application.**
2. **Tesseract installation.**
3. **Sample file path for reference.**
4. **Acceptable File formats.**
5. **Detailed instructions**

**Documentation: -**

1. **Pip commands to set up the necessary environment and dependencies for the given Flask application.**
2. Flask and CORS:

pip install Flask Flask-Cors

1. PyMuPDF:

pip install pymupdf

1. Pandas:

pip install pandas

1. Pytesseract:

pip install pytesseract

1. Pillow:

pip install pillow

1. NumPy:

pip install numpy

1. OpenCV:

pip install opencv-python

1. SpaCy:

pip install spacy

1. Matplotlib:

pip install matplotlib

1. Download SpaCy model:

python -m spacy download en\_core\_web\_sm

1. **Tesseract installation.**

**Tesseract-OCR:**

* For Tesseract-OCR, you need to install it manually. Download and install it from [Tesseract at UB Mannheim](https://github.com/UB-Mannheim/tesseract/wiki) or use your system's package manager.

The below code when put in Git Bash,

Git clone <https://github.com/UB-Mannheim/tesseract.git>

This will download Tesseract. We have used **Tesseract 5.3.3**

1. **Sample file path for reference.**
2. File storage - C:\ADITYA\Python\OCR
   1. This will contain output, templates, upload, app.py too.
   2. Upload: Will contain all the files which have been uploaded from the Interactive Webpage
   3. Output: Will contain the masked output after we upload the file from the Webpage
   4. App.py contains the Python script developed by The Royal Flush
   5. Templates contain the Webpages code
3. Setting the path for tesseract- C:\Program Files\Tesseract-OCR\tesseract.exe
   1. This is a very crucial step, the location of the tesseract which has been downloaded has to be pasted in the file “app.py”, in the code, line 31
   2. The easiest way to not get any error is to follow the exact file path.
4. **Acceptable File formats.**

**Text-Based PDFs:**

**Searchable PDFs:** These PDFs are created from electronic sources and contain text that can be selected, copied, and searched.

**Tagged PDFs:** These PDFs include additional structural information to enhance accessibility and reflow for different devices.

**Image-Based PDFs:**

**Scanned PDFs:** Created by scanning physical documents, these PDFs are essentially images of the scanned pages. Text in these PDFs is not selectable or searchable without OCR (Optical Character Recognition).

**OCR PDFs:** These are scanned PDFs that have undergone OCR to convert the images of text into selectable and searchable text.

1. Detailed Instructions:

The above instructions when followed will allow us to run the application. Lets have a Check list,

1. Pip commands
2. Tesseract installation
3. Tesseract configurations with the app.py

After completion of these steps we can start with running the program. There are 2 ways to run the program,

* 1. Through CMD and then setup a local host
  2. Directly through Visual Studio code

1. **Through CMD and then setup a local host**

Using the cd command go to the directory where the application is stored. For example, cd C:\ADITYA\Python\OCR , this is the one where we have stored our application. After we are in the directory where the application is stored , we can use the command “python app.py” to launch a local instance. The local instance url should look like this : Running on <http://127.0.0.1:5000>. By following this link we can launch the application and upload a pdf file for data masking.

1. **Directly through Visual Studio code**

When it comes to Visual studio code we can click the run button which can be found on the top right corner, from which a local instance link will be generated in the CLI of VS Code.

After Uploading the PDF file for data masking we can the output will be displayed in the preview for easy access, to open the masked pdf we can check it in the output folder where the name of the file will be “masked\_file name”. There will also be another csv file, this file is present to view the extraction of the text from the uploaded PDF file.

The word [MASKED] is put on top of the sensitive content found by the regex patterns developed by The Royal Flush. The dynamic build of the application allows us to upload Any PDF file, if there are and sensitive images it will be completely blacked out or masked from the output pdf file.