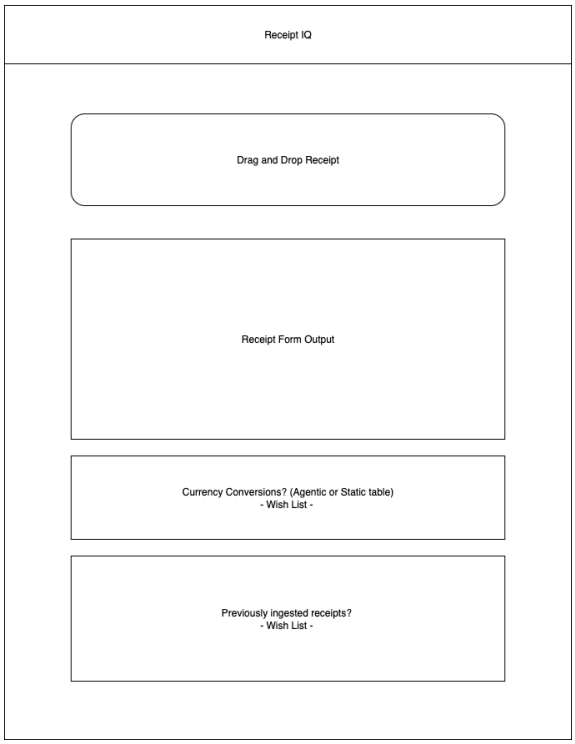
Receipt IQ

UI:

Next JS
TypeScript
Tailwind

Wishlist:
Prisma
AG grid Table

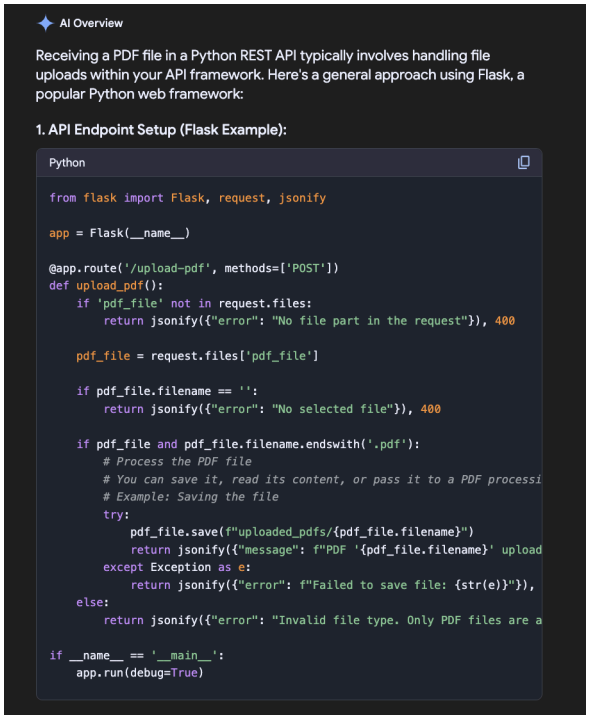


Server:

Python - Flask API

- Install Larry's OCR dependencies into flask API project

- Route should accept PDF file (POST)



AI Overview

Receiving a PDF file in a Python REST API typically involves handling file uploads within your API framework. Here's a general approach using Flask, a popular Python web framework:

1. API Endpoint Setup (Flask Example):

Python

```python
from flask import Flask, request, jsonify

app = Flask(__name__)

@app.route('/upload-pdf', methods=['POST'])
def upload_pdf():
    if 'pdf_file' not in request.files:
        return jsonify({"error": "No file part in the request"}), 400

    pdf_file = request.files['pdf_file']

    if pdf_file.filename == '':
        return jsonify({"error": "No selected file"}), 400

    if pdf_file and pdf_file.filename.endswith('.pdf'):
        # Process the PDF file
        # You can save it, read its content, or pass it to a PDF processi
        # Example: Saving the file
        try:
            pdf_file.save(f"uploaded_pdfs/{pdf_file.filename}")
            return jsonify({"message": f"PDF '{pdf_file.filename}' upload
        except Exception as e:
            return jsonify({"error": f"Failed to save file: {str(e)}"}),
    else:
        return jsonify({"error": "Invalid file type. Only PDF files are a

if __name__ == '__main__':
    app.run(debug=True)
```

- Upon receiving the file, use Larry's OCR functions to parse the receipt and return json to the response of the post request.

Agentic (potential):

- Live currency conversion (Use a consistent source)
    - https://www.oanda.com/currency-converter/live-exchange-rates/
    - After parsing the receipt into json, use agentic AI to reference the URL above and via an optional parameter, parse the currency values into the correct currency with a type return specifying what currency is represented in the response.
- Foreign Language translation
    - After parsing the receipt into json, use agentic AI to translate the json values into the proper language via an optional parameter that is english by default.

Project Structure:

- Use Docker Network
- Is Larry's project part of flask api project? Probably?
    - https://github.com/goagiq/VisionOCR/tree/main
- Initial UI<->Server communication
- README for monorepo should be able to have someone clone the repository and run the project with a few simple commands.
- All dependencies for OCR should be installed in the python service.
- Should be able to blow project away and re-clone and only use the README multiple times.