

Credit Card Statement Parser

Project Overview

This project reads credit card statement PDFs and extracts important information automatically.

It helps in converting unstructured text from the PDF into a structured format (JSON).

Features

- Reads PDF credit card statements
- Extracts key details like:
 - Card provider (HDFC, SBI, etc.)
 - Last 4 digits of the card
 - Payment due date
 - Total amount due
- Returns clean and structured data in JSON format
- Simple backend built using Flask

How It Works

1. The user uploads a credit card statement (PDF).
2. The program extracts text from the PDF using PyMuPDF.
3. Regular Expressions (Regex) are used to find important details like due date, total, and card number.
4. The extracted data is organized and sent back as a JSON response.

Technologies Used

Python

The main programming language used for the project.

It handles text, files, and automation easily.

Flask

A lightweight Python framework used to create the backend server.
It allows users to upload PDF files and get the extracted data.

PyMuPDF (fitz)

Used to read and extract text from PDF credit card statements.

Regular Expressions (Regex)

Used to search and extract patterns like dates, numbers, and card digits from the PDF text.

Postman / cURL

Used to test the API by sending a PDF file and checking the extracted results.

Installation and Setup

1. Install all required libraries

```
pip install -r requirements.txt
```

2. Run the Flask application

```
python app.py
```

3. Upload a PDF file using Postman or cURL

```
curl -X POST http://127.0.0.1:5000/upload -F "file=@sample_credit_card_statement.pdf"
```

Future Improvements

- Add support for more banks
- Add error messages for invalid PDFs
- Create a dashboard to analyze spending
- Create more interactive and attractive dashboard