Credit Card Statement Parser - Project Description

Introduction

This project is a comprehensive solution for automatically extracting and structuring information from credit card statements. It addresses the common challenge of manually reviewing and organizing data from monthly credit card statements by leveraging artificial intelligence to parse PDF documents and extract relevant financial information.

Problem Statement

Credit card statements contain crucial financial information, but extracting this data manually is time-consuming and error-prone. Users often need to:

- Track spending across multiple billing cycles
- Compare transactions between different months
- Export data for budgeting or accounting purposes
- Quickly access key information like due dates and balances

Traditional solutions require either manual data entry or expensive enterprise software. This project provides a free, accessible alternative that works with statements from any major credit card issuer.

Solution Overview

The Credit Card Statement Parser is a web-based application that automates the extraction of financial data from credit card statements. Users simply upload their PDF statement, and the system automatically identifies and extracts key information using AI-powered natural language processing.

Core Functionality

PDF Processing

The application uses pdfplumber, a robust Python library, to extract text content from PDF files. This approach handles:

- Multi-page statements
- Various PDF formats and encodings
- Both text-based and slightly formatted content
- Table structures commonly found in transaction listings

The extraction process preserves the document structure while converting it into machine-readable text that can be analyzed by the AI model.

AI-Powered Data Extraction

The heart of the system is Google's Gemini AI model, specifically the Gemini 2.5 Flash variant. This large language model has been optimized for:

- Understanding financial document structures
- Identifying key data points across different statement formats
- Handling variations in terminology and layout
- Extracting structured data from unstructured text

The AI receives the extracted text along with a carefully crafted prompt that specifies exactly what information to extract and in what format. This prompt engineering ensures consistent, structured output regardless of the input statement's format.

Data Validation and Cleaning

After extraction, the system validates and cleans the data through several processes:

Date Normalization: Dates are converted to a standard YYYY-MM-DD format, handling various input formats like MM/DD/YYYY, DD/MM/YYYY, or written formats like "January 15, 2024".

Amount Processing: Financial amounts are cleaned by removing currency symbols, commas, and handling different representations of negative numbers (parentheses, minus signs, etc.).

Card Number Validation: The system verifies that card numbers contain exactly four digits and are formatted consistently.

Transaction Validation: Each transaction is checked for completeness and validity, with malformed entries either corrected or flagged for user review.

User Interface

The application features a Streamlit-based web interface that provides:

Simple Upload Mechanism: Users can drag and drop PDF files or browse their file system to select a statement.

Progress Feedback: Visual indicators show the processing stages: extraction, AI analysis, and validation.

Organized Data Display: Extracted information is presented in logical sections with clear labels and formatting.

Export Capabilities: Users can download the extracted data in JSON format for programmatic use or CSV format for spreadsheet applications.

The interface is designed to be intuitive, requiring no technical knowledge from the user while providing detailed information for those who want it.

Technical Architecture

Components

PDFExtractor Class: Handles all PDF reading operations. It opens PDF files, iterates through pages, extracts text content, and identifies any table structures present in the document.

GeminiParser Class: Manages interaction with the Gemini AI API. It constructs prompts, sends requests, handles responses, and parses the returned JSON data.

DataValidator Class: Implements validation logic for all extracted fields. It checks data types, formats dates consistently, cleans numeric values, and flags any anomalies.

Streamlit Interface: Provides the web-based user experience, handling file uploads, displaying results, and managing user interactions.

Data Flow

The processing pipeline follows a clear sequence:

1. User uploads a PDF file through the web interface
2. PDFExtractor reads the file and extracts all text content
3. Extracted text is sent to the Gemini AI model with structured instructions
4. AI returns a JSON object containing identified data points
5. DataValidator processes the JSON, cleaning and validating each field
6. Results are displayed in the web interface with organized sections
7. User can review data and export in their preferred format

Error Handling

The system includes comprehensive error handling at each stage:

- PDF reading errors are caught and reported with specific messages
- API failures trigger appropriate user notifications
- Invalid data is flagged but doesn't prevent processing of valid fields
- Warnings inform users of potential issues without blocking functionality

Data Extraction Details

Card Information

The system identifies the credit card issuer by recognizing common bank names and card program identifiers. It extracts the card variant (Platinum, Gold, Rewards, etc.) and locates the last four digits of the card number, which are typically prominently displayed.

Billing Information

Billing cycle dates are extracted by identifying date ranges that span approximately one month. The payment due date is found by looking for phrases like "Payment Due", "Due Date", or similar terminology, then extracting the associated date.

Financial amounts are identified by their context and position within the statement. The system distinguishes between total balance, minimum payment, previous balance, and new charges by analyzing the surrounding text and common statement layouts.

Transaction History

Transactions are extracted by identifying table-like structures or repeated patterns that include dates, descriptions, and amounts. The system handles:

- Standard purchases and charges
- Credits and refunds (represented as negative amounts)
- Fees and interest charges
- Various description formats and lengths

Each transaction is parsed into a structured format with date, description, and amount fields, making it easy to analyze or export the data.

Additional Metrics

When available, the system also extracts credit limit, available credit, and calculates credit utilization. These metrics provide useful insights into credit health and spending patterns.

Use Cases

Personal Finance Management

Individuals can use the parser to quickly extract transaction data for budgeting purposes. Rather than manually entering each transaction into budgeting software, users can export the CSV and import it directly.

Business Expense Tracking

Business owners and freelancers can streamline expense reporting by automatically extracting business expenses from company credit cards. The structured data can be easily integrated with accounting software.

Financial Analysis

Users interested in analyzing spending patterns can accumulate data from multiple billing cycles and perform trend analysis, category comparison, or spending forecasting.

Record Keeping

The tool provides an easy way to digitize and organize credit card statements for long-term record keeping, making it simple to search and retrieve specific transactions or billing periods.

Technology Stack

Python: The core programming language, chosen for its excellent libraries for PDF processing, data manipulation, and web development.

Streamlit: A Python framework for building data applications with minimal code. It provides the web interface, handles file uploads, and manages the user experience.

pdfplumber: A pure Python library for extracting text and tables from PDFs. It's reliable, actively maintained, and handles various PDF formats well.

Google Generative AI (Gemini): Google's advanced language model API provides the intelligence for understanding and extracting data from unstructured text.

Pandas: Used for data manipulation, particularly when creating CSV exports and analyzing transactions.

Design Decisions

Why AI Instead of Rules-Based Parsing

Traditional PDF parsers rely on rigid rules and patterns specific to each document format. This approach requires extensive customization for each credit card issuer and breaks when statement formats change.

Using an AI model provides flexibility and adaptability. The model understands context and can extract information even when layouts vary. This significantly reduces maintenance requirements and improves compatibility across different issuers.

Local PDF Processing

Rather than sending PDFs directly to external services, the application processes PDFs locally and only sends extracted text to the AI API. This approach:

- Protects user privacy by avoiding full document uploads
- Reduces API costs since text requires fewer tokens than images
- Improves reliability by handling PDF complexity locally
- Provides better error messages for PDF-related issues

Security and Privacy Considerations

Data Handling

The application processes statements locally on the user's machine (when run locally) or on the server (when deployed). PDFs are not permanently stored; they exist only in memory during processing.

API Communication

Only extracted text content is sent to the Gemini API for analysis. This means:

- No complete PDFs leave the system
- Sensitive information like full card numbers are not transmitted (only last 4 digits are extracted)
- Users maintain control over their financial documents

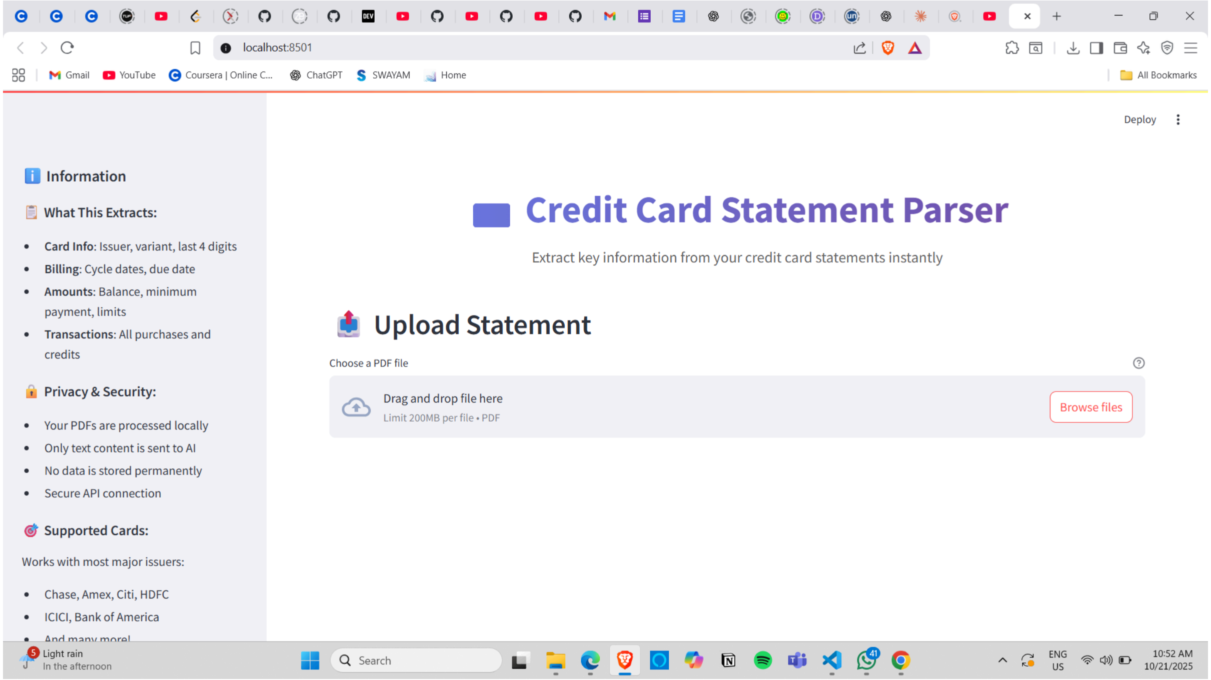
API Key Management

The application is designed to use a single API key managed by the application administrator. This approach:

- Simplifies user experience (no need for users to obtain API keys)
- Allows for usage monitoring and cost control
- Centralizes security management

For production deployments, API keys should be stored as environment variables rather than hardcoded, following security best practices.

Output



Information

What This Extracts:

- **Card Info**: Issuer, variant, last 4 digits
- **Billing**: Cycle dates, due date
- **Amounts**: Balance, minimum payment, limits
- **Transactions**: All purchases and credits

Privacy & Security:

- Your PDFs are processed locally
- Only text content is sent to AI
- No data is stored permanently
- Secure API connection

Supported Cards:

Works with most major issuers:

- Chase, Amex, Citi, HDFC
- ICICI, Bank of America
- And many more!

Credit Card Statement Parser

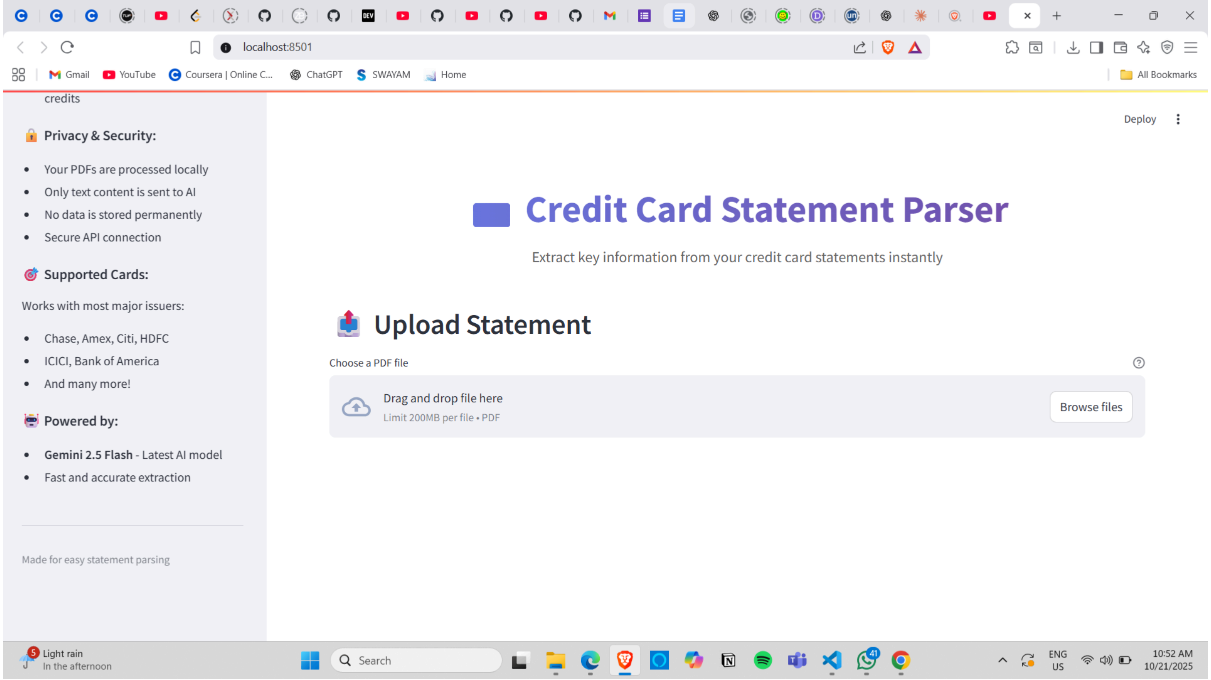
Extract key information from your credit card statements instantly

Upload Statement

Choose a PDF file

Drag and drop file here
Limit 200MB per file • PDF

Browse files



credits

Privacy & Security:

- Your PDFs are processed locally
- Only text content is sent to AI
- No data is stored permanently
- Secure API connection

Supported Cards:

Works with most major issuers:

- Chase, Amex, Citi, HDFC
- ICICI, Bank of America
- And many more!

Powered by:

- **Gemini 2.5 Flash** - Latest AI model
- Fast and accurate extraction

Made for easy statement parsing

Credit Card Statement Parser

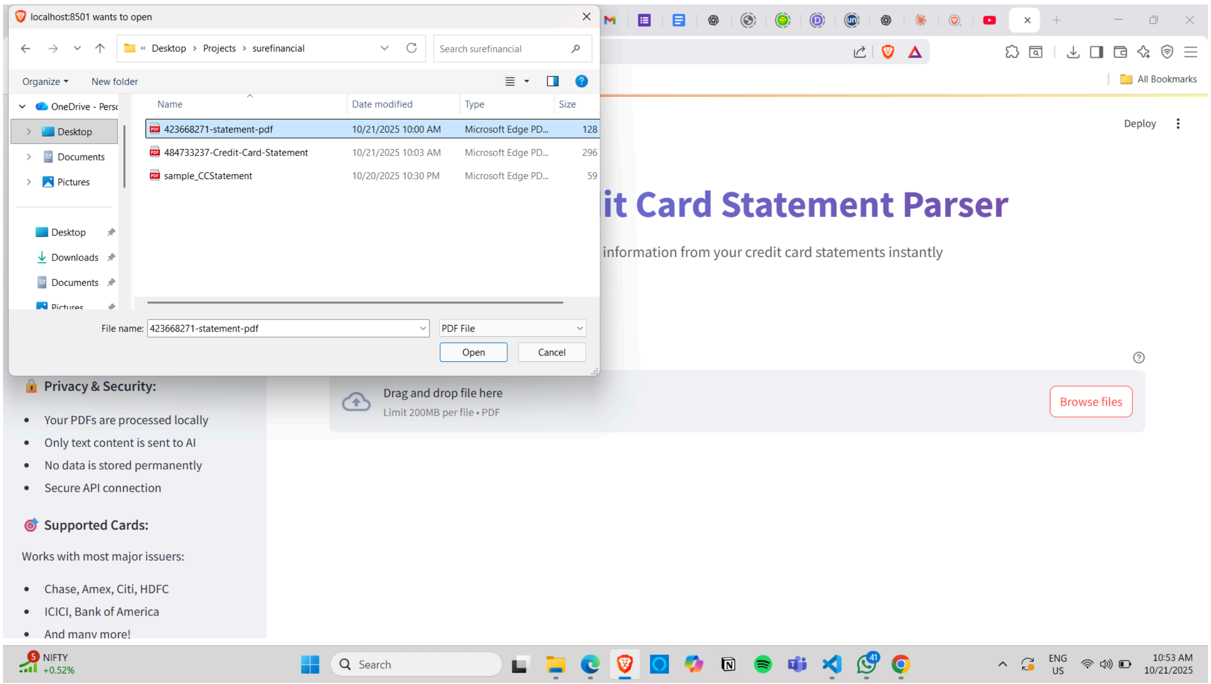
Extract key information from your credit card statements instantly

Upload Statement

Choose a PDF file

Drag and drop file here
Limit 200MB per file • PDF

Browse files



localhost:8501 wants to open

Desktop › Projects › surefinancial

Name	Date modified	Type	Size
423668271-statement-pdf	10/21/2025 10:00 AM	Microsoft Edge PD...	128
484733237-Credit-Card-Statement	10/21/2025 10:03 AM	Microsoft Edge PD...	296
sample_CCStatement	10/20/2025 10:30 PM	Microsoft Edge PD...	59

File name: 423668271-statement-pdf

PDF File

it Card Statement Parser

information from your credit card statements instantly

Drag and drop file here

Limit 200MB per file • PDF

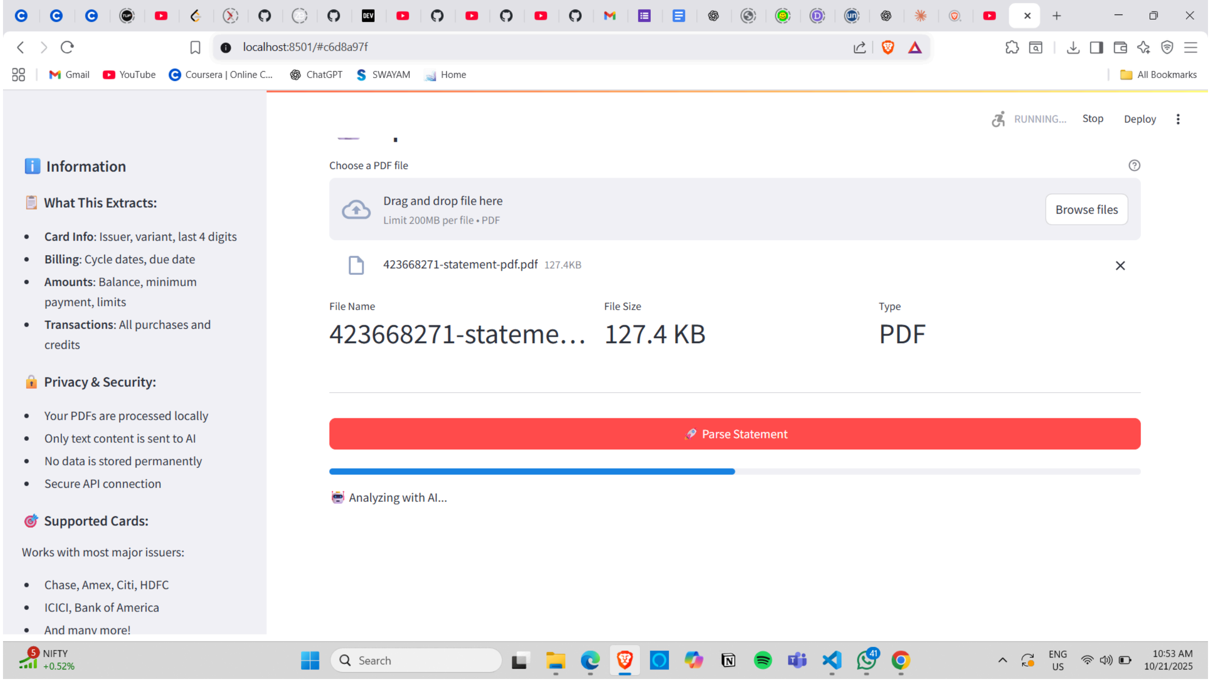
🔒 **Privacy & Security:**

- Your PDFs are processed locally
- Only text content is sent to AI
- No data is stored permanently
- Secure API connection

🎯 **Supported Cards:**

Works with most major issuers:

- Chase, Amex, Citi, HDFC
- ICICI, Bank of America
- And many more!



ℹ️ **Information**

📋 **What This Extracts:**

- **Card Info**: Issuer, variant, last 4 digits
- **Billing**: Cycle dates, due date
- **Amounts**: Balance, minimum payment, limits
- **Transactions**: All purchases and credits

🔒 **Privacy & Security:**

- Your PDFs are processed locally
- Only text content is sent to AI
- No data is stored permanently
- Secure API connection

🎯 **Supported Cards:**

Works with most major issuers:

- Chase, Amex, Citi, HDFC
- ICICI, Bank of America
- And many more!

Choose a PDF file

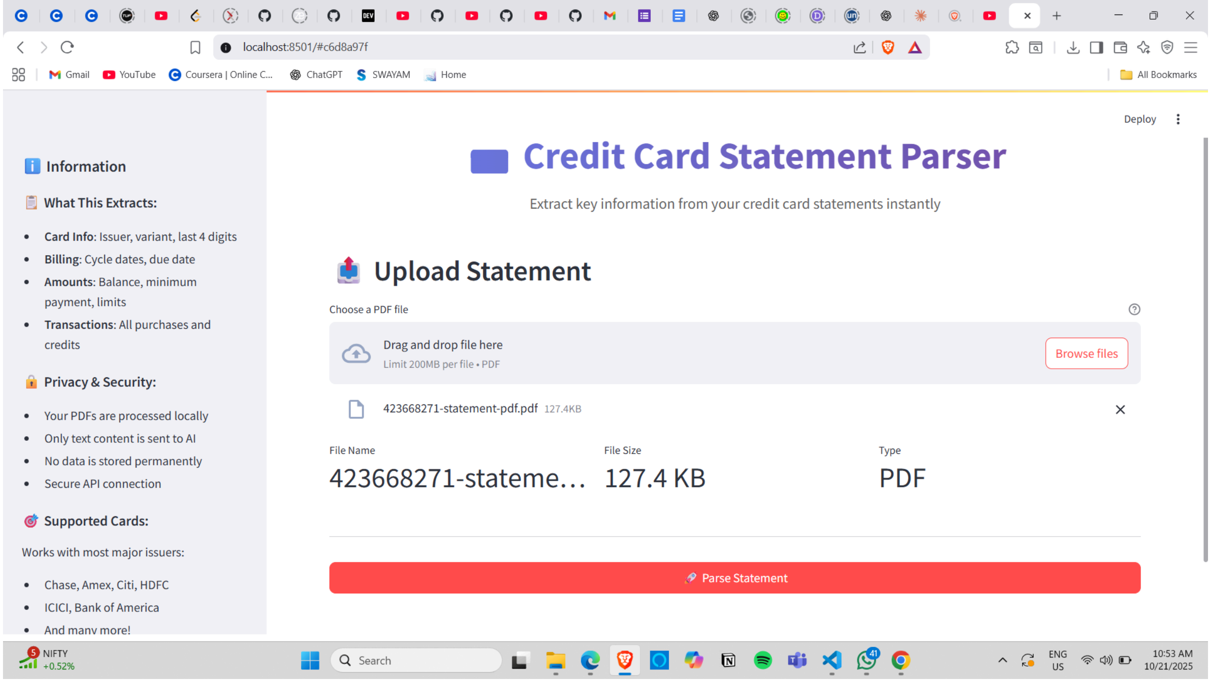
Drag and drop file here

Limit 200MB per file • PDF

423668271-statement-pdf.pdf 127.4KB

File Name	File Size	Type
423668271-stateme...	127.4 KB	PDF

Analyzing with AI...



Information

What This Extracts:

- **Card Info**: Issuer, variant, last 4 digits
- **Billing**: Cycle dates, due date
- **Amounts**: Balance, minimum payment, limits
- **Transactions**: All purchases and credits

Privacy & Security:

- Your PDFs are processed locally
- Only text content is sent to AI
- No data is stored permanently
- Secure API connection

Supported Cards:

Works with most major issuers:

- Chase, Amex, Citi, HDFC
- ICICI, Bank of America
- And many more!

Credit Card Statement Parser

Extract key information from your credit card statements instantly

Upload Statement

Choose a PDF file

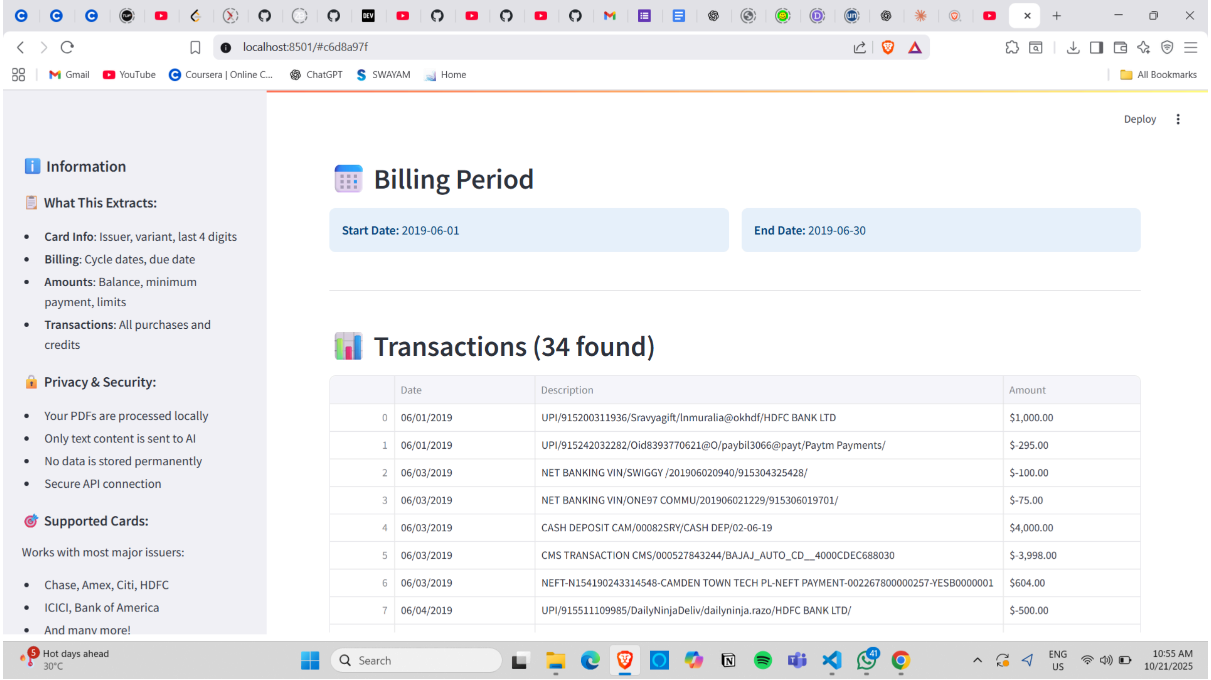
Drag and drop file here
Limit 200MB per file • PDF

Browse files

423668271-statement-pdf.pdf 127.4KB

File Name	File Size	Type
423668271-stateme…	127.4 KB	PDF

Parse Statement



Information

What This Extracts:

- **Card Info**: Issuer, variant, last 4 digits
- **Billing**: Cycle dates, due date
- **Amounts**: Balance, minimum payment, limits
- **Transactions**: All purchases and credits

Privacy & Security:

- Your PDFs are processed locally
- Only text content is sent to AI
- No data is stored permanently
- Secure API connection

Supported Cards:

Works with most major issuers:

- Chase, Amex, Citi, HDFC
- ICICI, Bank of America
- And many more!

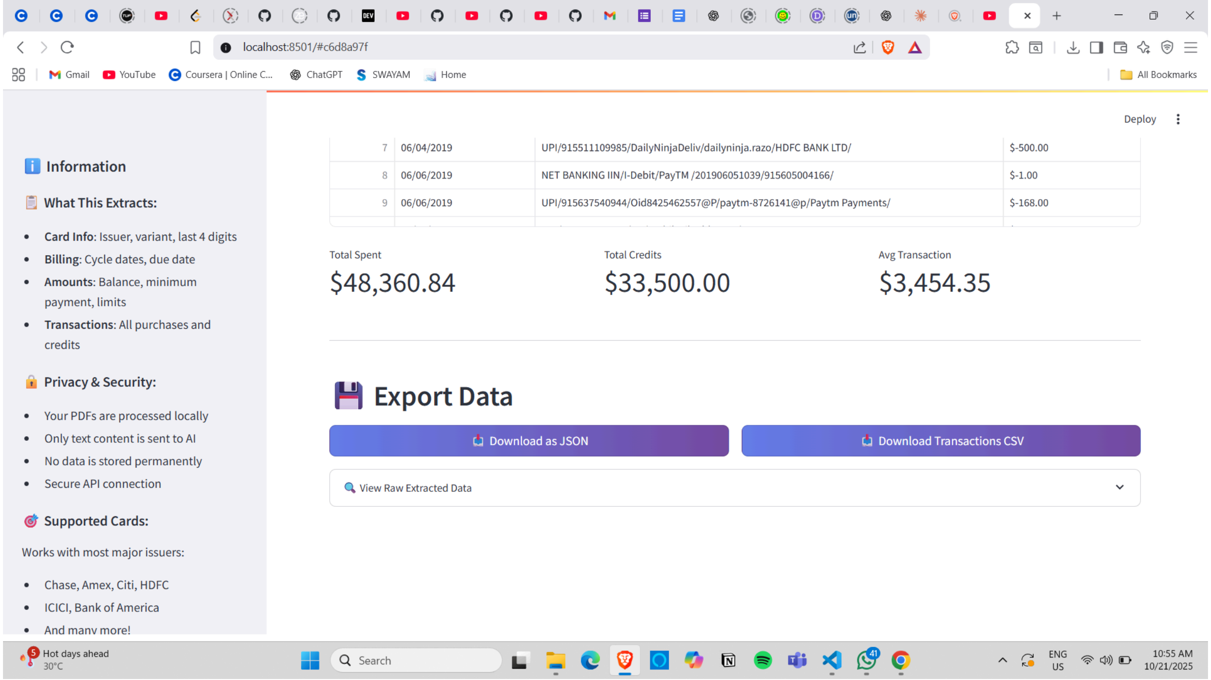
Billing Period

Start Date: 2019-06-01

End Date: 2019-06-30

Transactions (34 found)

	Date	Description	Amount
0	06/01/2019	UPI/915200311936/Sravyagift/lnmuralia@okhdf/HDFC BANK LTD	$1,000.00
1	06/01/2019	UPI/915242032282/Oid8393770621@O/paybil3066@payt/Paytm Payments/	$-295.00
2	06/03/2019	NET BANKING VIN/SWIGGY /201906020940/915304325428/	$-100.00
3	06/03/2019	NET BANKING VIN/ONE97 COMMU/201906021229/915306019701/	$-75.00
4	06/03/2019	CASH DEPOSIT CAM/00082SRY/CASH DEP/02-06-19	$4,000.00
5	06/03/2019	CMS TRANSACTION CMS/000527843244/BAJAJ_AUTO_CD__4000CDEC688030	$-3,998.00
6	06/03/2019	NEFT-N154190243314548-CAMDEN TOWN TECH PL-NEFT PAYMENT-002267800000257-YESB0000001	$604.00
7	06/04/2019	UPI/915511109985/DailyNinjaDeliv/dailyninja.razo/HDFC BANK LTD/	$-500.00



Information

What This Extracts:

- **Card Info**: Issuer, variant, last 4 digits
- **Billing**: Cycle dates, due date
- **Amounts**: Balance, minimum payment, limits
- **Transactions**: All purchases and credits

Privacy & Security:

- Your PDFs are processed locally
- Only text content is sent to AI
- No data is stored permanently
- Secure API connection

Supported Cards:

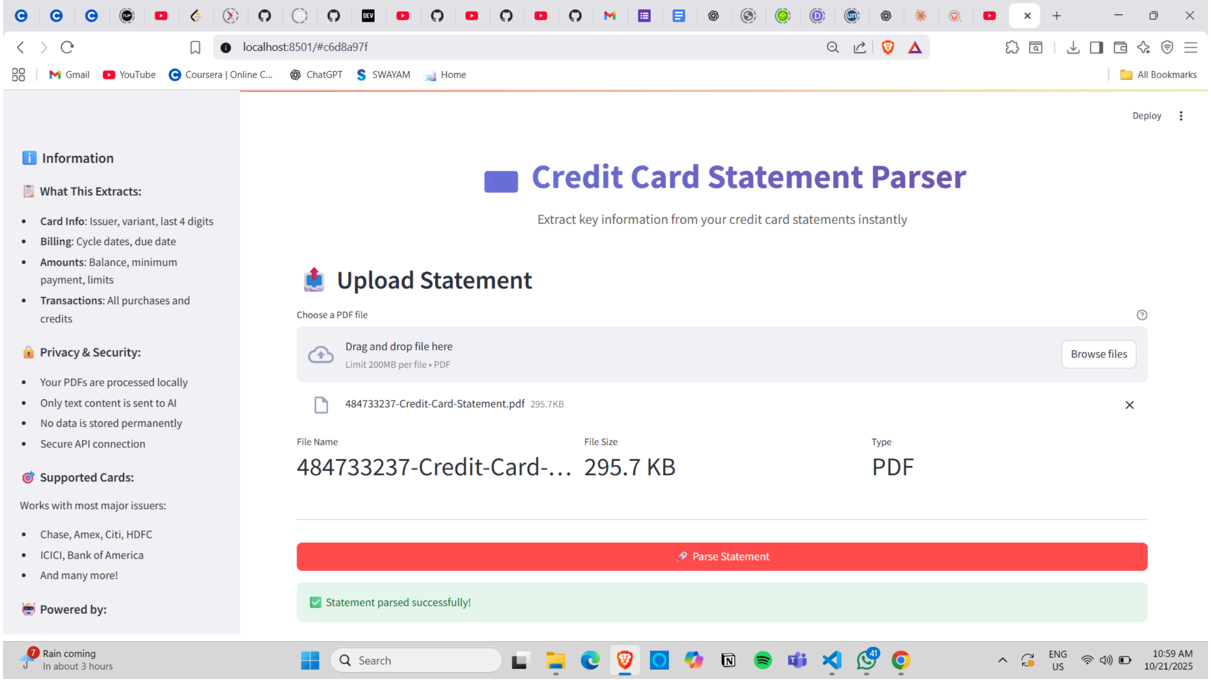
Works with most major issuers:

- Chase, Amex, Citi, HDFC
- ICICI, Bank of America
- And many more!

7	06/04/2019	UPI/915511109985/DailyNinjaDeliv/dailyninja.razo/HDFC BANK LTD/	$-500.00
8	06/06/2019	NET BANKING IIN/I-Debit/PayTM /201906051039/915605004166/	$-1.00
9	06/06/2019	UPI/915637540944/Oid8425462557@P/paytm-8726141@p/Paytm Payments/	$-168.00

Total Spent	Total Credits	Avg Transaction
$48,360.84	$33,500.00	$3,454.35

Export Data



Information

What This Extracts:

- **Card Info**: Issuer, variant, last 4 digits
- **Billing**: Cycle dates, due date
- **Amounts**: Balance, minimum payment, limits
- **Transactions**: All purchases and credits

Privacy & Security:

- Your PDFs are processed locally
- Only text content is sent to AI
- No data is stored permanently
- Secure API connection

Supported Cards:

Works with most major issuers:

- Chase, Amex, Citi, HDFC
- ICICI, Bank of America
- And many more!

Powered by:

Credit Card Statement Parser

Extract key information from your credit card statements instantly

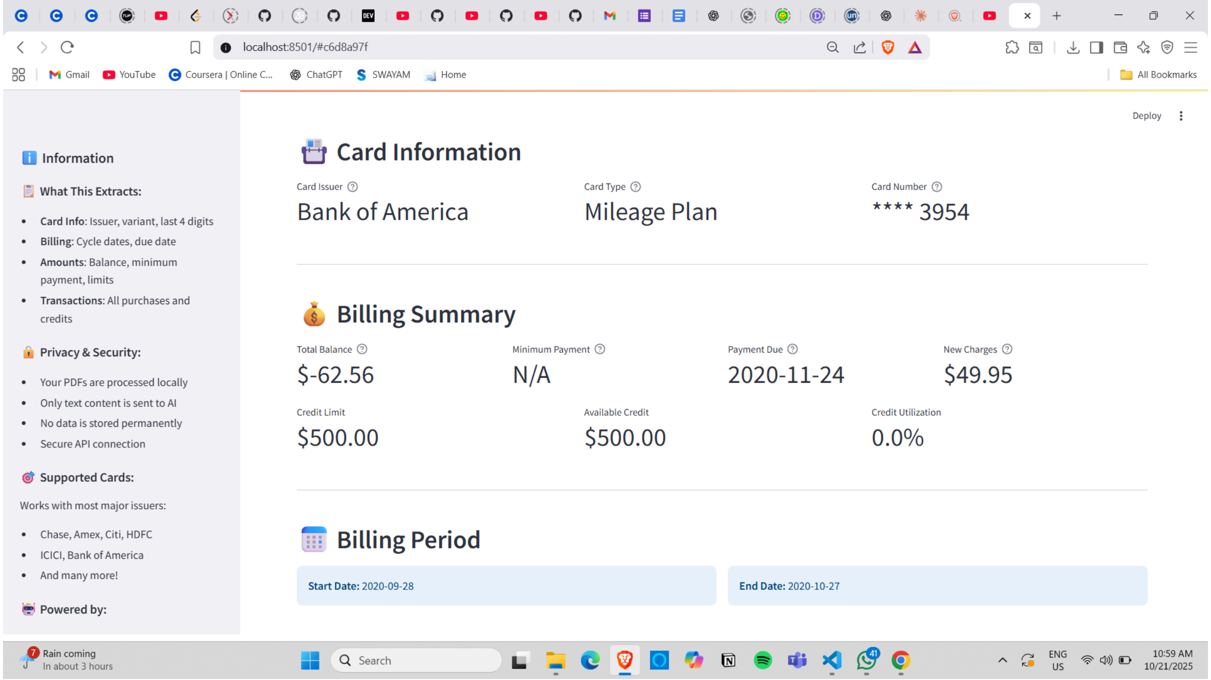
Upload Statement

Choose a PDF file

484733237-Credit-Card-Statement.pdf 295.7KB

File Name	File Size	Type
484733237-Credit-Card-...	295.7 KB	PDF

Statement parsed successfully!



Information

What This Extracts:

- **Card Info**: Issuer, variant, last 4 digits
- **Billing**: Cycle dates, due date
- **Amounts**: Balance, minimum payment, limits
- **Transactions**: All purchases and credits

Privacy & Security:

- Your PDFs are processed locally
- Only text content is sent to AI
- No data is stored permanently
- Secure API connection

Supported Cards:

Works with most major issuers:

- Chase, Amex, Citi, HDFC
- ICICI, Bank of America
- And many more!

Powered by:

Card Information

Card Issuer: Bank of America

Card Type: Mileage Plan

Card Number: **** 3954

Billing Summary

Total Balance: $-62.56

Minimum Payment: N/A

Payment Due: 2020-11-24

New Charges: $49.95

Credit Limit: $500.00

Available Credit: $500.00

Credit Utilization: 0.0%

Billing Period

Start Date: 2020-09-28

End Date: 2020-10-27

Conclusion

The Credit Card Statement Parser demonstrates how modern AI capabilities can solve practical everyday problems. By combining robust PDF processing, intelligent data extraction, and a user-friendly interface, it transforms the tedious task of manual data entry into a simple, automated process.