GENAI-BASED SOLUTION FOR AUDIT REQUESTS

A small sentence which explains all about this presentation

Presenter Name

OVERVIEW OF AUDIT REQUESTS AT BIC BANK

Frequent Audit Requests

BIC Bank receives very frequent audit requests from various parties.

Requesting Parties

The audit requests typically come from the Government, Regulatory bodies, and other relevant entities.

Reporting Requirements

The bank is required to report all data from Client's financial statements to the requesting parties.

OPERATIONAL CHALLENGES FACED

- Data Collection and Accuracy: Gathering accurate, complete financial data from various sources can be time-consuming and error-prone, especially with discrepancies in formats and reporting standards.
- Fraud Detection: Identifying anomalies or fraudulent activities within large volumes of transactions is challenging and requires sophisticated tools and expertise.
- Security and Confidentiality: Ensuring the security and confidentiality of sensitive financial information during the audit process is a critical operational challenge.

AUTOMATION GOALS FOR BIC BANK

Data Extraction and Processing: Automating the extraction of data from client financial statements, reducing manual effort, and speeding up the audit process.

Fraud Detection: Using AI to identify anomalies in transactions and suspicious patterns involving merchants or accounts , enabling detection of fraudulent activities by flagging transactions involving a big amount or frequent transactions with the same account onthe same day

```python
pytesseract.pytesseract.tesseract_cmd = r'/usr/bin/tesseract' # Update with the actual path if different

# Function to extract text from regular PDFs (non-scanned)
def extract_pdf_text(pdf_path):
    all_text = ""
    with pdfplumber.open(pdf_path) as pdf:
        for page_num, page in enumerate(pdf.pages, start=1):
            print(f"Extracting text from page {page_num}...")
            page_text = page.extract_text()
            if page_text:
                all_text += f"--- Page {page_num} ---\n{page_text}\n\n"
            else:
                print(f"Warning: No text extracted from page {page_num} using pdfplumber. Trying OCR...")
                all_text += ocr_extract_from_pdf_page(pdf_path, page_num)

    return all_text
```

DATA EXTRACTION USING MACHINE LEARNING

We automated the extraction of structured financial transactions from bank statements using a tech stack consisting of Streamlit for the frontend, SQLite for backend data storage, and Google's Generative AI (Gemini-Pro) for the core extraction. The AI model processes the input text, following strict formatting instructions to accurately extract and structure transactions, enabling streamlined auditing and review processes.

DATA LOADING ON UI SCREEN

Extracted Data

	client_name	bank_name	account_number	transaction_date	credit_debit	description
0	Mr. John Doe	B1 Bank	1261234567	08/19/2020	Credit	Correction: Cash Withdra
1	Mr. John Doe	B1 Bank	1261234567	08/19/2020	Credit	Correction: ATS Cash vat
2	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Credit	Banking App Payment Re
3	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Debit	Banking App Payment Lu
4	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Debit	Banking App Payment Fe
5	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Debit	ATM Balance Enquiry Fee
6	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Debit	ATM Cash Withdrawal Sp
7	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Debit	Cash Withdrawal Fee (AT
8	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Credit	Interest Received
9	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Debit	SMS Notification Fee

EXTRACTION ACCURACY SCORE

We interpreted accuracy by comparing entries in original table with the entries in the table obtained by extracting using AI

```
Accuracy for uploading the document B1 is: 83.47%
Accuracy for uploading the document B2 is: 59.23%
Accuracy for uploading the document B3 is: 72.35%
Accuracy for uploading the document B4 is: 78.59%
```

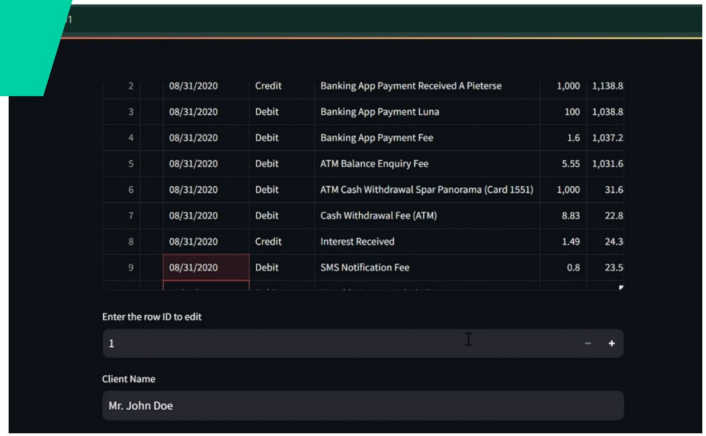
FRAUDULENT TRANSACTION CHECK

We evaluated transactions as being fraudulent on the basis of frequency of transactions made and/or the amount involved.

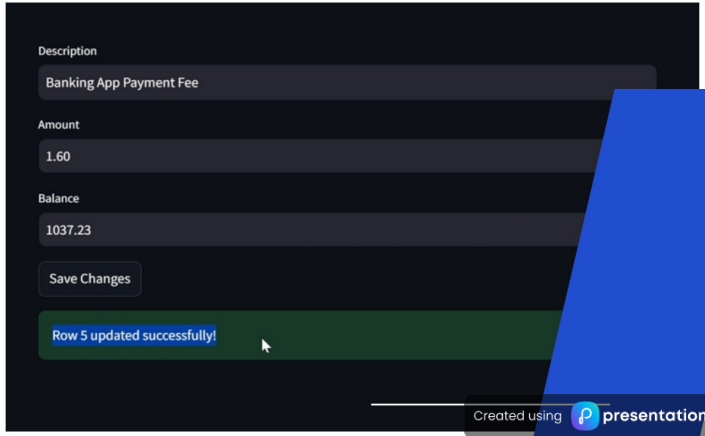
```python
def identify_fraudulent_transactions(bank_statements):
    # Defining threshold for high frequency and high amount
    high_frequency_threshold = 5
    high_amount_threshold = 10000

    # Dictionary to track transaction counts by date for frequency checking
    date_transactions = {}
    fraudulent_transactions = []

    for transaction in bank_statements:
        date = transaction[3]
        amount = transaction[6]

        # Track frequency of transactions for each date
        if date not in date_transactions:
            date_transactions[date] = 1
        else:
            date_transactions[date] += 1

        # Check if transaction qualifies as high frequency or h
        if date_transactions[date] > high_frequency_threshold:
            fraudulent_transactions.append(transaction)
        elif amount >= high_amount_threshold:
            fraudulent_transactions.append(transaction)

    return fraudulent_transactions

# Extract the fraudulent transactions based on the analysis
```

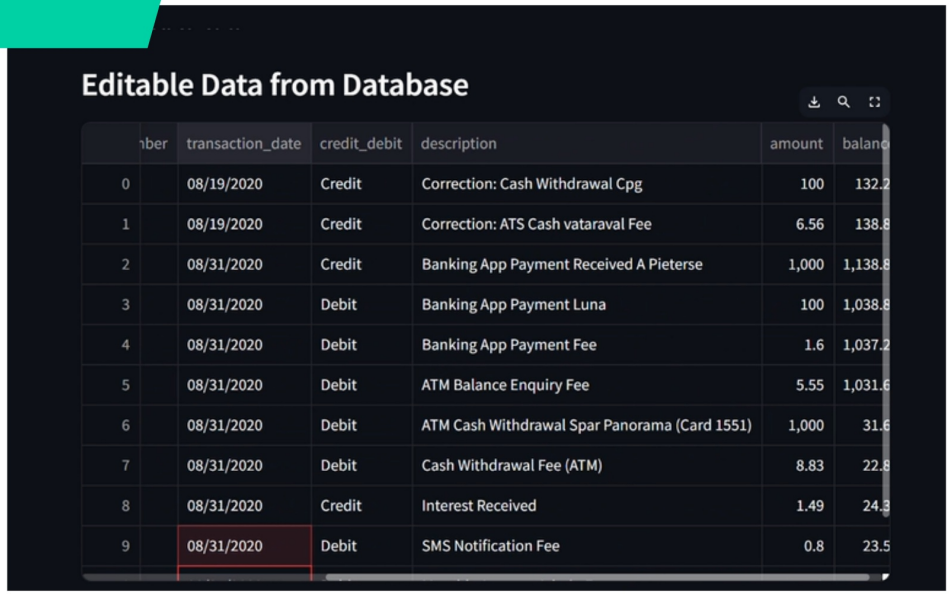
MANUAL REVIEW AND EDIT OPTIONS

Once the financial data is extracted from the PDF and displayed in the Streamlit interface, users can manually review each transaction in a table-like format. After editing, the data is immediately updated in the SQLite database using SQL UPDATE queries, allowing for quick and efficient corrections.



2	08/31/2020	Credit	Banking App Payment Received A Pieterse	1,000	1,138.8
3	08/31/2020	Debit	Banking App Payment Luna	100	1,038.8
4	08/31/2020	Debit	Banking App Payment Fee	1.6	1,037.2
5	08/31/2020	Debit	ATM Balance Enquiry Fee	5.55	1,031.6
6	08/31/2020	Debit	ATM Cash Withdrawal Spar Panorama (Card 1551)	1,000	31.6
7	08/31/2020	Debit	Cash Withdrawal Fee (ATM)	8.83	22.8
8	08/31/2020	Credit	Interest Received	1.49	24.3
9	08/31/2020	Debit	SMS Notification Fee	0.8	23.5





Editable Data from Database

	ıber	transaction_date	credit_debit	description	amount	balanc
0		08/19/2020	Credit	Correction: Cash Withdrawal Cpg	100	132.2
1		08/19/2020	Credit	Correction: ATS Cash vataraval Fee	6.56	138.8
2		08/31/2020	Credit	Banking App Payment Received A Pieterse	1,000	1,138.8
3		08/31/2020	Debit	Banking App Payment Luna	100	1,038.8
4		08/31/2020	Debit	Banking App Payment Fee	1.6	1,037.2
5		08/31/2020	Debit	ATM Balance Enquiry Fee	5.55	1,031.6
6		08/31/2020	Debit	ATM Cash Withdrawal Spar Panorama (Card 1551)	1,000	31.6
7		08/31/2020	Debit	Cash Withdrawal Fee (ATM)	8.83	22.8
8		08/31/2020	Credit	Interest Received	1.49	24.3
9		08/31/2020	Debit	SMS Notification Fee	0.8	23.5

DATA SAVING FUNCTIONALITY

After extracting data from the uploaded PDF and converting it into a DataFrame, it is saved to the SQLite database using SQL queries.

DATA EXPORT CAPABILITY

Data saved in SQLite can easily be converted into a CSV format. Once the data is fetched from the database into a pandas DataFrame, we exported it as a CSV file using the DataFrame.to_csv() method.

Extracted Data

	client_name	bank_name	account_number	transaction_date	credit_debit	description
0	Mr. John Doe	B1 Bank	1261234567	08/19/2020	Credit	Correction: Cash Withdra

	client_name	bank_name	account_number	transaction_date	credit_debit	c
0	Mr. John Doe	B1 Bank	1261234567	08/19/2020	Credit	(
1	Mr. John Doe	B1 Bank	1261234567	08/19/2020	Credit	(
2	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Credit	E
3	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Debit	E
4	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Debit	E
5	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Debit	A
6	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Debit	A
7	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Debit	(
8	Mr. John Doe	B1 Bank	1261234567	08/31/2020	Credit	I