

FORM MITHRA

Presented By:

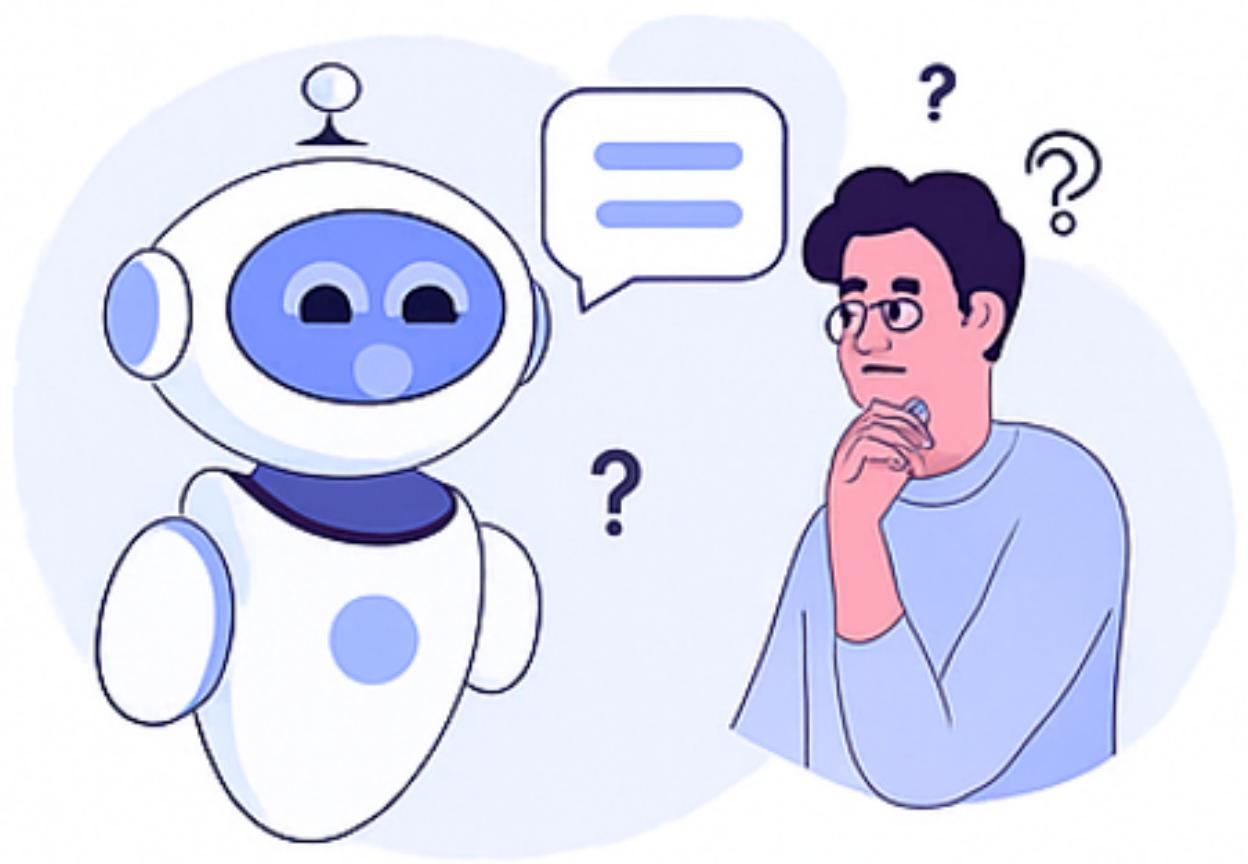
- 1.M.Saichandra (22WU0104075)
- 2.K.Sai Meghana (22WU0104110)
- 3.G.Shravinya (22WU0104131)

Supervised by:-

Prof. Meher Gayatri Devi Tiwari,
Assistant Professor.

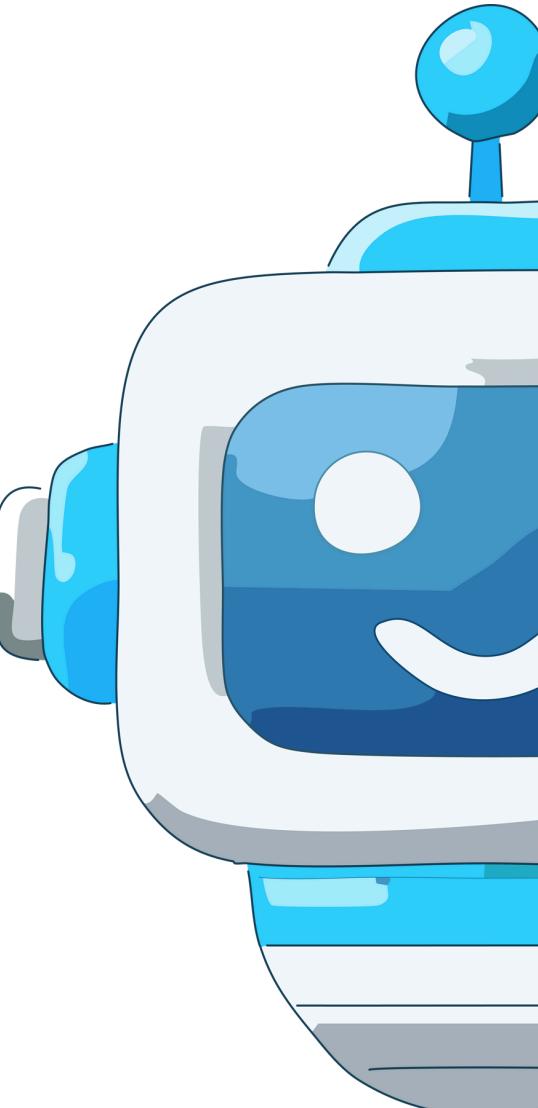
Introduction – Background & Significance

- Banking processes still depend heavily on manual form validation.
- Manual verification → time-consuming, error-prone, inconsistent.
- Rise of AI and NLP enables automation of document understanding.
- Goal: Build an AI-driven system for accurate, automated form validation.



Problem Statement

- Human validation leads to delays and inconsistent interpretations.
- Lack of unified policy reference → frequent policy violations.
- Need for an automated system to extract, understand, and validate form data against policy documents.



Literature / Existing Work

Paper	Solution	Limitation
Automating Banking Forms with Intelligent Document Processing [1].	Extracts and processes data from semi/unstructured bank forms using AI-driven IDP.	Needs well-scanned documents; may struggle with handwritten fields.
How RPA and OCR Are Revolutionizing Process Automation [2].	Automates document field extraction using OCR and robotic process automation.	Template-dependent; weak for unusual layouts or poor scan quality.
Bank Form Automation[3].	Rule-based extraction of fields from account and loan forms using deep learning.	Effective mainly for standard templates; limited adaptability.

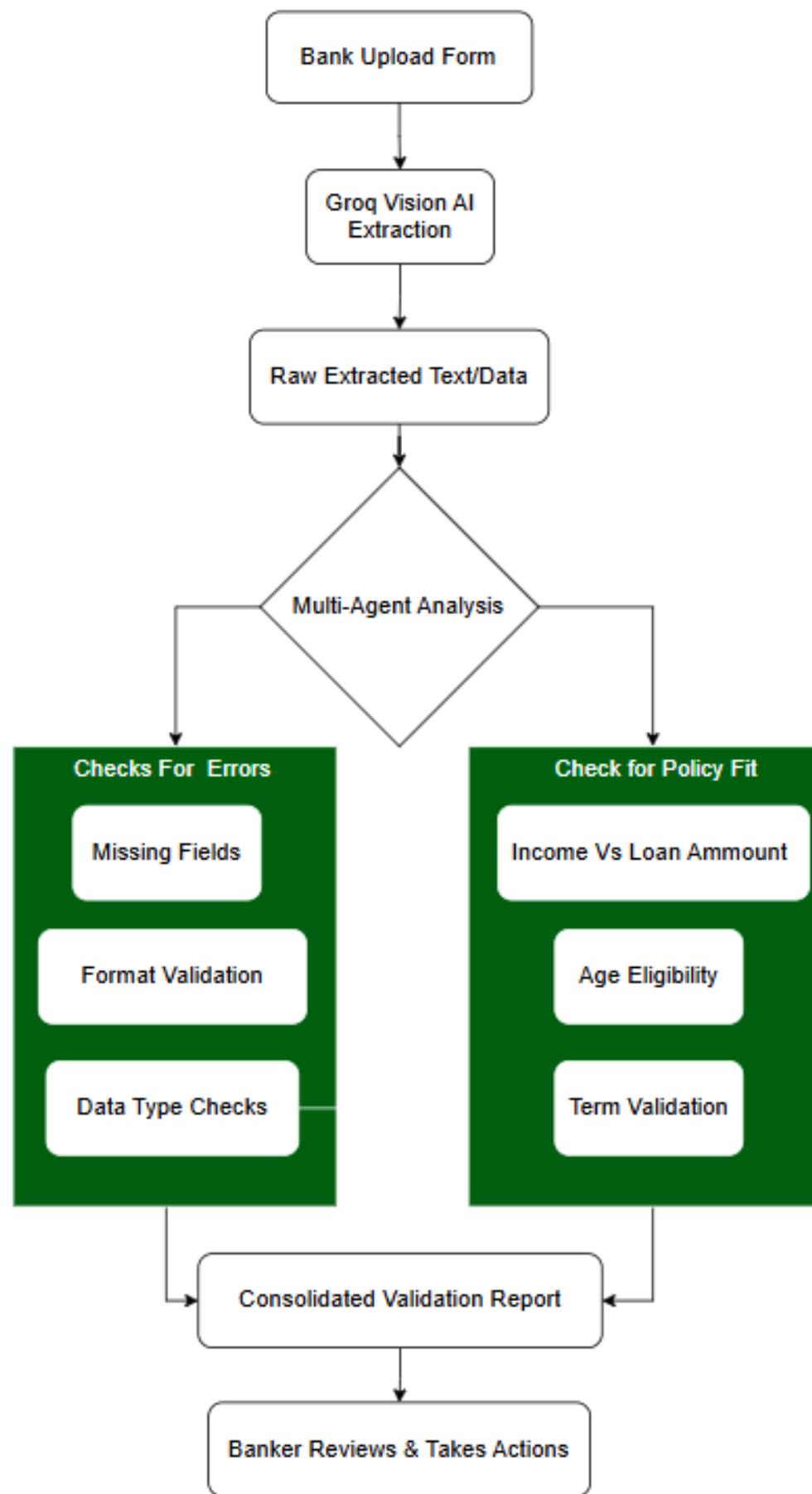
Objectives

- To automate the process of extracting data from scanned banking forms using Vision AI.
- To validate the extracted data against the official policy document using RAG.
- To containerize and automate deployment for scalability and CI/CD integration.

Novelty

- 1. Agentic RAG-Based Validation:** Integration of Agentic AI and RAG enables autonomous, policy-aware validation of bank forms.
- 2. Shift from Rule-Based to Intelligent Agents:** The system replaces static rule-based checks with intelligent multi-agent reasoning, improving accuracy, adaptability, and compliance automation.

Workflow



End-to-End Workflow

- User uploads form via Streamlit UI.
- Extraction Agent (Groq Vision AI) → structured JSON data.
- RAG system (FAISS + Sentence Transformers) retrieves relevant policy chunks.
- Validation Agent (Groq LLM) checks compliance and generates results.
- Streamlit displays final decision - Approved/Rejected/Needs Review.

Methodology

Frontend (User Interaction Layer)

Tool: Streamlit

Provides an intuitive web interface for uploading bank forms (PDF/Image).

Displays extracted fields, validation results, and policy-based recommendations.

Backend (AI & Processing Layer)

a. Extraction Phase

Tool: Groq Vision (LLaMA-3.2-90B-Vision)

Extracts structured data (e.g., name, account number, amount) from uploaded forms.

Converts document data into JSON format for downstream validation.

b. Retrieval Phase

Tools: FAISS + Sentence Transformers

Retrieves top-N relevant policy chunks from a vector store containing 67 policy documents.

Ensures semantic, context-aware retrieval beyond keyword matching.

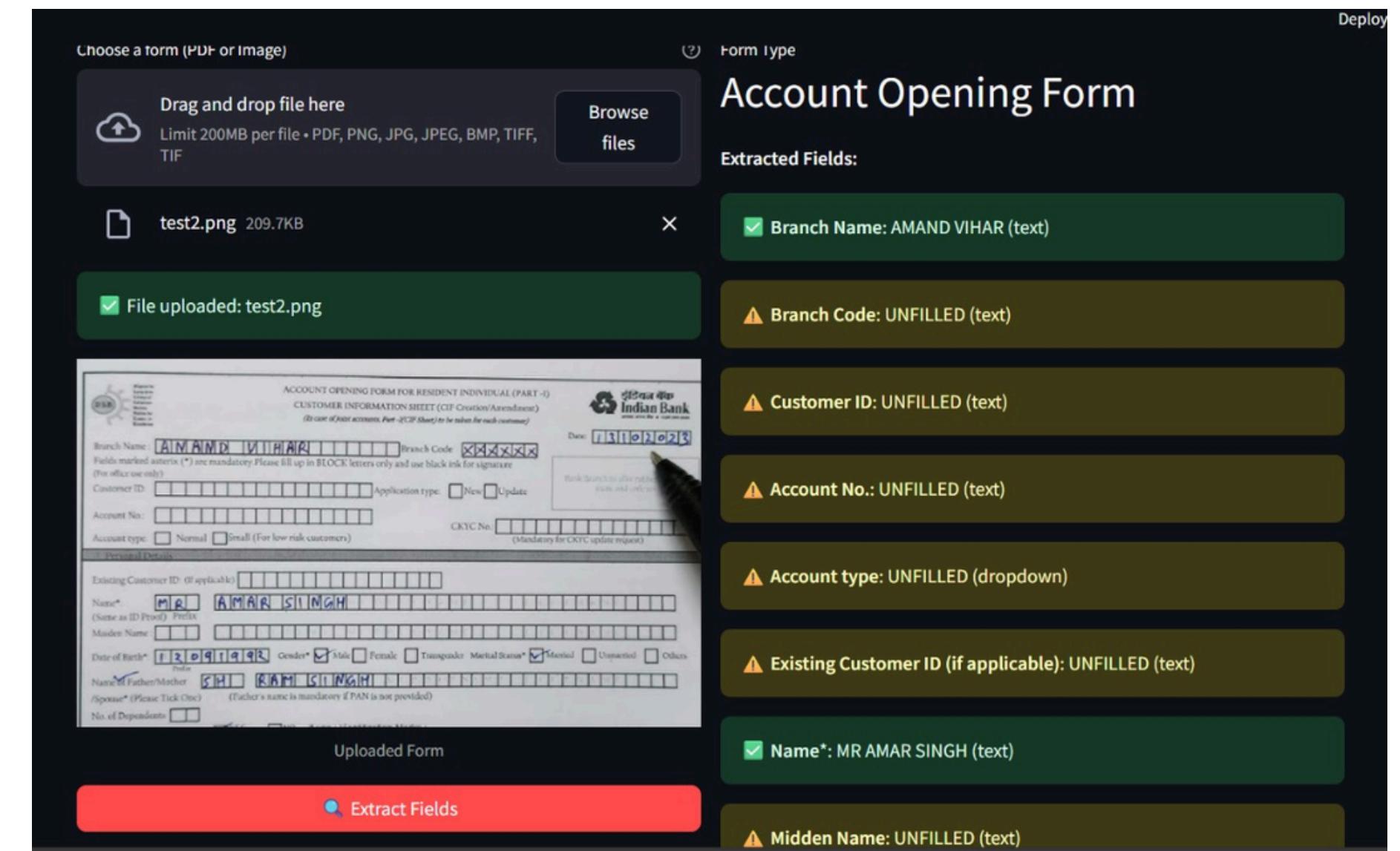
Methodology

c. Validation Phase

- Tool: Groq LLaMA-3.1-70B-Versatile
- The Validation Agent reasons over extracted data and retrieved policies.

Infrastructure & Deployment

- Containerization: Docker ensures consistent and portable deployment.
- Automation & CI/CD: GitHub Actions automates testing and builds.
- Supporting Tools:
 - Tesseract OCR - fallback extraction for scanned/low-quality images.
 - PyMuPDF, OpenCV for preprocessing and text extraction.



The screenshot displays a user interface for validating an account opening form. At the top, there's a file upload area where 'test2.png' has been uploaded. Below the file list is a success message: 'File uploaded: test2.png'. To the right, the 'Account Opening Form' PDF is shown with various fields. A sidebar on the right lists 'Extracted Fields' with corresponding validation status:

- Branch Name: AMAND VIHAR (text) - Valid (green checkmark)
- Branch Code: UNFILLED (text) - Invalid (orange warning triangle)
- Customer ID: UNFILLED (text) - Invalid (orange warning triangle)
- Account No.: UNFILLED (text) - Invalid (orange warning triangle)
- Account type: UNFILLED (dropdown) - Invalid (orange warning triangle)
- Existing Customer ID (if applicable): UNFILLED (text) - Invalid (orange warning triangle)
- Name*: MR AMAR SINGH (text) - Valid (green checkmark)
- Middle Name: UNFILLED (text) - Invalid (orange warning triangle)

Tools & Technologies

Programming & Frameworks:

- Python – Core development language
- Streamlit – Web UI for user interaction
- Groq AI (Vision + LLM) – Form extraction & validation

AI & NLP Components:

- FAISS (Facebook AI Similarity Search) – Vector storage & semantic retrieval
- Sentence Transformers (all-MiniLM-L6-v2) – Text embeddings
- Tesseract OCR – Backup text extraction

Infrastructure & Deployment:

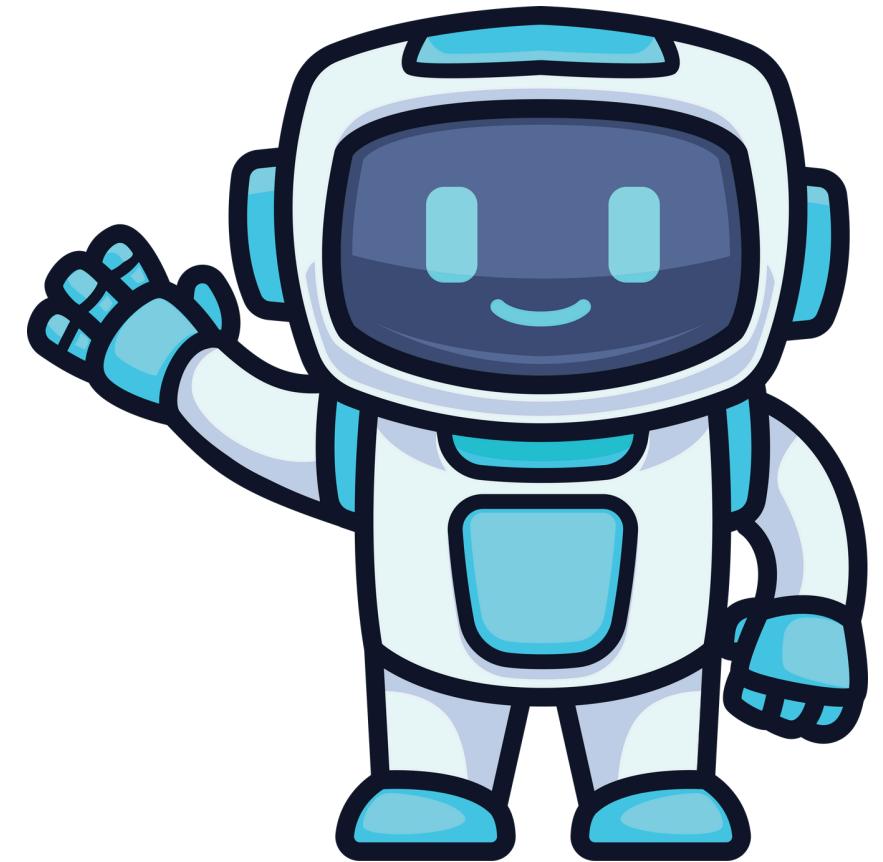
- Docker & Docker Compose – Containerization and reproducible environments
- GitHub Actions (CI/CD) – Automated testing and deployment

Supporting Libraries:

- OpenCV, PyMuPDF, PDFplumber – Image & PDF preprocessing

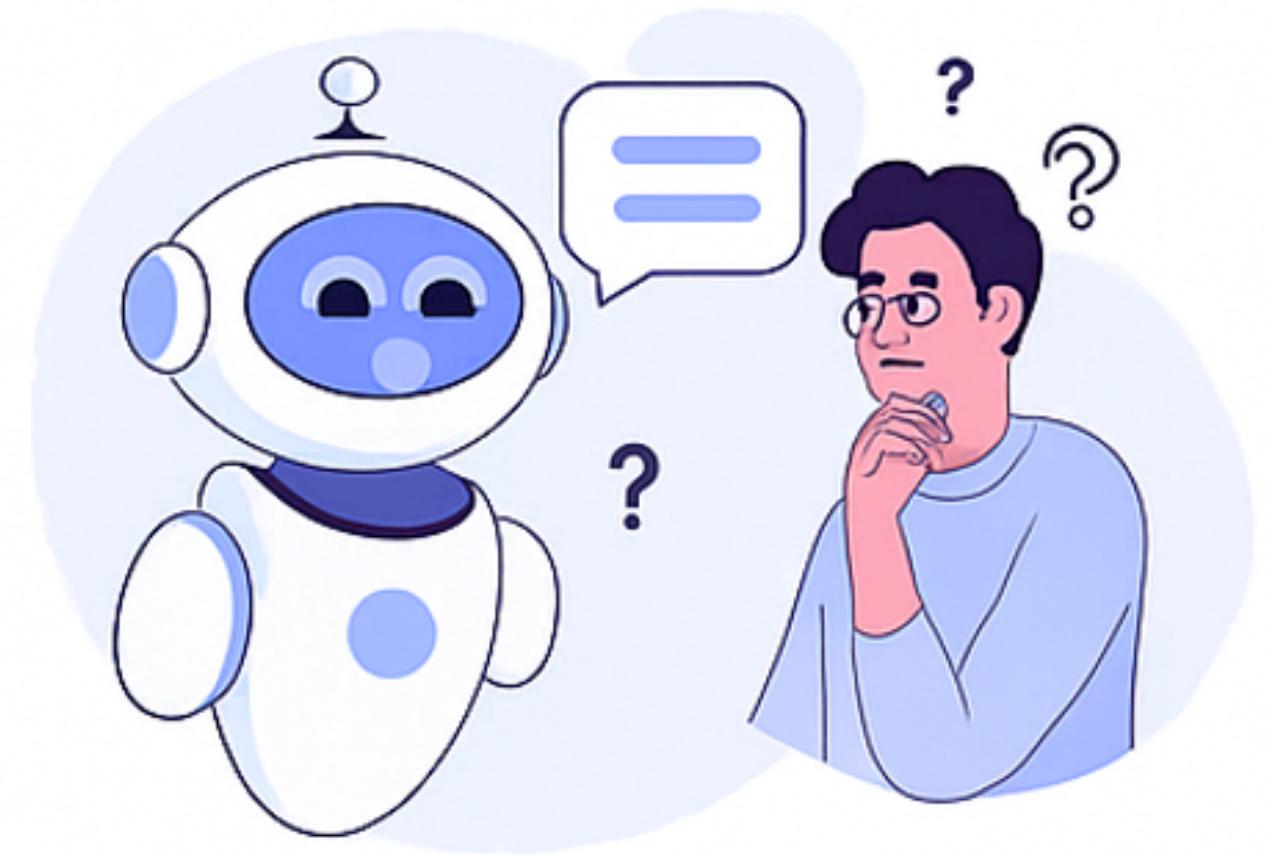
Purpose Summary:

→ Extraction + Validation + Deployment in a unified, automated pipeline



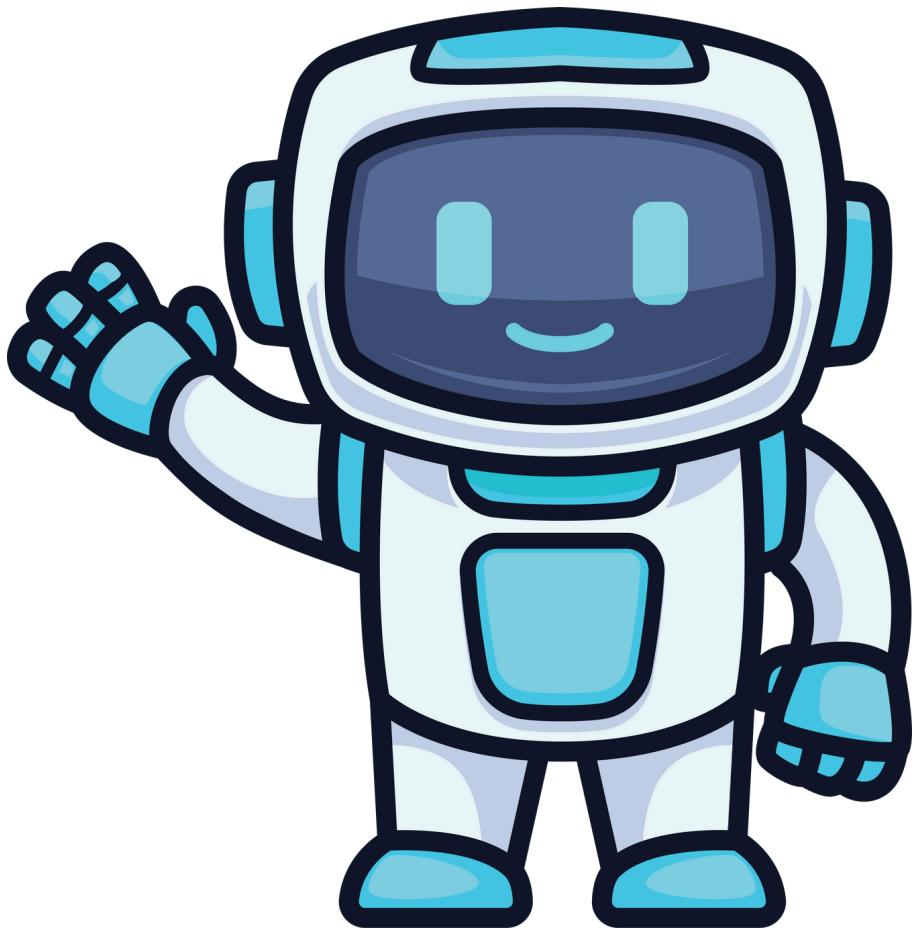
Challenges Faced

- Handling varied form layouts and poor scan quality.
- Extracting accurate fields from handwritten or mixed data.
- Large policy PDFs → long embedding & retrieval time.
- Maintaining consistency of vector embeddings after updates.



Conclusion

- Automated data extraction from scanned/digital bank forms using Groq Vision AI, achieving high accuracy and reduced manual effort.
- Implemented RAG-based validation to verify extracted data against official policy documents for policy-aware compliance.
- Ensured scalable and reliable deployment through Docker containerization and CI/CD integration using GitHub Actions.
- Delivered a fast, accurate, and AI-driven validation system that enhances operational efficiency and compliance transparency in banking.



References

1. Gupta, P. (2023). Automating Banking Forms with Intelligent Document Processing. ProcessMaker Blog.
2. Arya AI. (2024). How Intelligent Document Processing is Transforming Banking.
3. HyperVerge. (2025). How RPA and OCR Are Revolutionizing Process Automation.

THANK YOU