MOUNT ZION COLLEGE OF ENGINEERING AND TECHNOLOGY

SECURE PII FILE REDACTION SYSTEM

REVIEW - 0

TEAM MEMBERS

NOOR MOHAMED K (911722104083)

DHAYALAN M (911722104027)

PAULSON J (911722104084)

EDWIN RICHARD A (911722104030)

TEAM GUIDE

ELAVARASI D M.E., PH.D.,*
ASSISTANT PROFESSOR
DEPARTMENT OF CSE

PROBLEM STATEMENT

- Sensitive personal information like Aadhaar, PAN, phone number, email, and driving licence number is often present inside digital documents.
- When these documents are shared without redaction, it leads to privacy risks, identity theft, and data misuse.
- Many organizations share documents in large volumes, making manual redaction slow, tedious, and error-prone.
- There is no simple automated system that can detect PII accurately across multiple file types like PDF, Word, images, and scanned documents.
- Existing tools lack AI-based detection, policy-aware redaction, and support for Indian government ID formats.
- Users and organizations need a fast, reliable, and automatic solution that can identify and hide sensitive information before sharing documents.

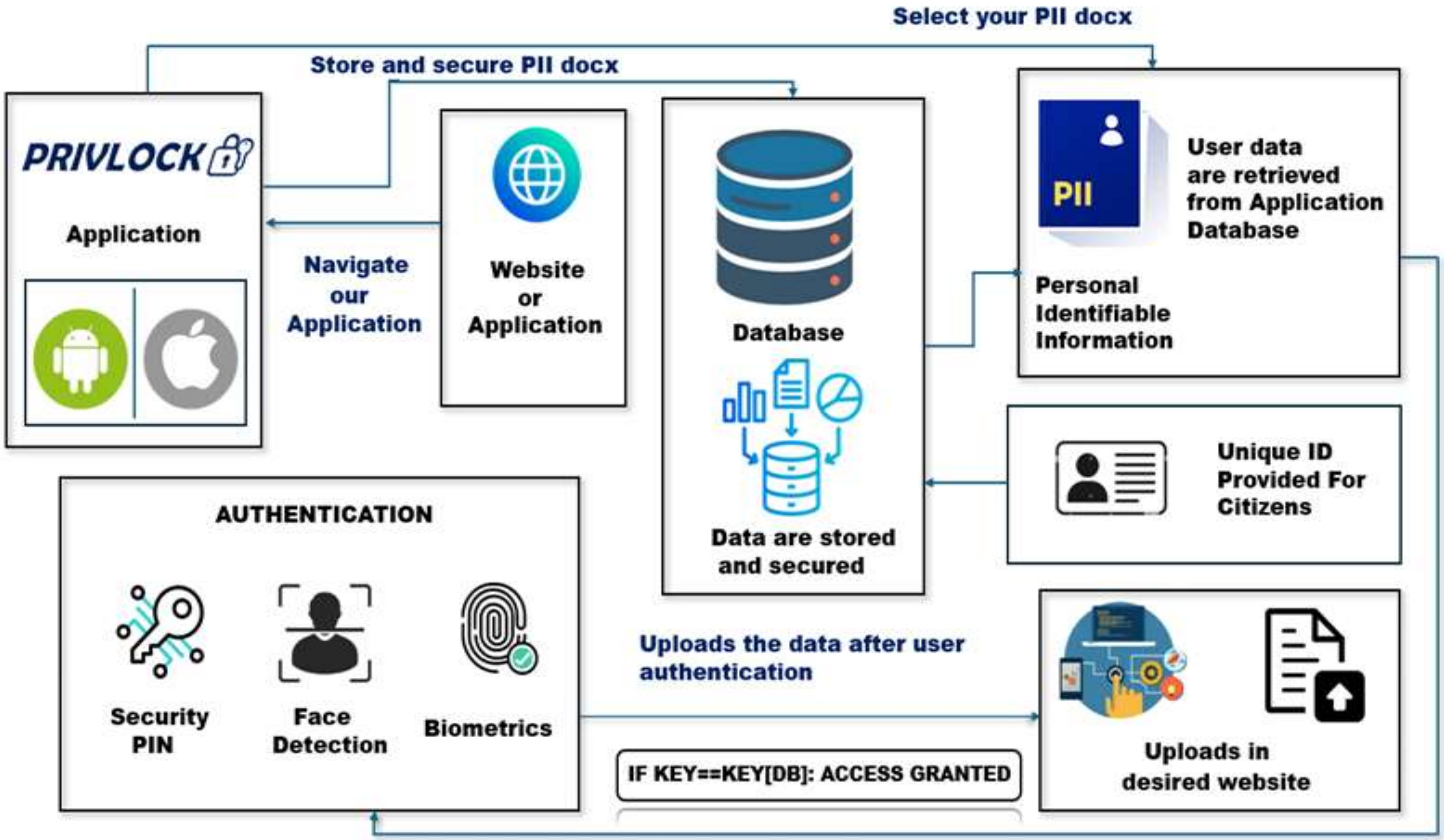
ABSTARCT

- This project provides a secure mobile application that helps users protect sensitive personal information in documents before sharing them.
- The user first unlocks the app using PIN or biometric authentication to ensure only authorised access.
- The user selects a document (PDF, Word, or image), and the system extracts text using OCR Tesseract or normal text reading.
- The application automatically detects personal information like Aadhaar, PAN, phone numbers, emails, and driving licence numbers using regex and AI/ML based NER models.
- A RAG based AI module analyzes privacy rules and decides which information should be masked or fully redacted.
- The system then creates a clean, redacted version of the document that removes all sensitive data.
- MySQL stores the redaction history and document activity securely.
- The final output allows users to safely share documents without exposing personal information.

TECHNOLOGY STACK

- **Frontend:** Flutter(Dart) – cross-platform mobile app for document upload & preview
- **Backend:** FastAPI (Python) – fast and lightweight API for AI processing
- **OCR & Processing:** Tesseract OCR – extract text from PDFs, images, and Word files
- **AI / ML:** Regex, NER, RAG – detect and decide sensitive information to redact
- **Database:** MySQL – store user logs, document metadata, and redaction history securely
- **Security:** AES-256 encryption, PIN/Biometric login – protect documents and user access

SAMPLE FLOW DIAGRAM



Select your PII docx
Store and secure PII docx
PRIVLOCK
Application
Navigate our Application
Website or Application
Database
Data are stored and secured
PII
User data are retrieved from Application Database
Personal Identifiable Information
Unique ID Provided For Citizens
AUTHENTICATION
Security PIN
Face Detection
Biometrics
Uploads the data after user authentication
IF KEY==KEY[DB]: ACCESS GRANTED
Uploads in desired website

METHODOLOGY

- **User Authentication**

  User unlocks the app using PIN or biometric verification.

- **Document Selection**

  User selects a PDF, Word file, or image to process.

- **Text Extraction**

  OCR (Tesseract) and parsers extract text from the document.

- **PII Detection**

  Regex and NER models identify sensitive data (Aadhaar, PAN, phone, email, etc.).

- **RAG Decision**

  RAG model decides which information should be redacted or masked.

- **Redaction**

  System automatically hides or masks the detected PII.

- **Output**

  A secure, redacted document is returned to the user for preview or sharing.

ALGORITHM

OCR Algorithm

- Input: Image or scanned PDF
- Preprocess (grayscale, denoise, threshold)
- Extract text using **Tesseract OCR**
- Send extracted text to PII detection

PII Detection Algorithm

Regex:

- Detect Aadhaar (4-4-4), PAN (ABCDE1234F), phone, email
- Store matched values + positions

NER (AI):

- Identify PERSON, ID_NUMBER, EMAIL, ADDRESS, etc.
- Combine Regex + NER results

RAG-Based Redaction Decision

- Input: PII list + document text
- Retrieve policy rules from vector DB
- LLM decides: **REDACT / MASK / KEEP**
- Output: redaction plan for each PII

Redaction / Masking Algorithm

- Locate each PII in the document
- REDACT → replace with █████
- MASK → partially hide (e.g., ******3210)
- Generate final redacted document and return to user

REFERNECE RESEARCH PAPER

- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10916629

- https://link.springer.com/chapter/10.1007/978-981-19-5689-8_8

- https://ieeexplore.ieee.org/abstract/document/11140717

- https://ieeexplore.ieee.org/abstract/document/10945230

- https://ieeexplore.ieee.org/abstract/document/11076720

THANK YOU
Supermeme.ai