PROJECT ABSTRACT

E-Signature Document Portal with Audit Trail

An e-signature portal where PDFs get signature fields, signers sign in sequence, and every event is sealed into a SHA-256 hash-chained audit trail. Delivered built-to-order with the working application, report, PPT and viva kit.

01. Title

E-Signature Document Portal with Audit Trail

02. Introduction / Background

Agreements still travel as PDFs over email: slow, untraceable, and trivially forgeable by pasting a signature image. Real signing platforms solve this with two mechanisms — an ordered signing ceremony with identity checks, and a tamper-evident audit trail where each event is hash-chained to the last so any later edit breaks the seal. This project implements both from scratch: field placement, sequential signing, drawn/typed signatures, and a verifiable audit certificate, with an honest treatment of the legal context (India's IT Act 2000).

03. Problem Statement

Students hear 'digital signature' and think of pasting a PNG. The real engineering is the integrity model: how do you prove, months later, who signed what, in which order, and that nothing changed since? The project addresses this by building a hash-chained event log — every view, signature, reminder and download hashed with SHA-256 and linked to the previous hash — plus a verification view that recomputes the chain.

04. Objectives

- Upload PDFs and place signature, initial, date and text fields by drag-and-drop with pixel positioning.
- Run sequential signing ceremonies: signers notified in order, with reminders and decline/void flows.
- Capture drawn (pointer) or typed signatures composited tamper-evidently onto the PDF.
- Record every event in an immutable SHA-256 hash-chained audit log (IP, timestamp, user agent).
- Generate sealed PDFs with a verification summary page plus a downloadable audit certificate.
- Deliver the complete student kit: application source, deployment setup, report, PPT and viva Q&A.;

05. Proposed Methodology

The system is developed in three stages. (1) Document model: the uploaded PDF's SHA-256 becomes the envelope's genesis hash; fields are stored with page coordinates

and assignee. (2) Ceremony: single-use JWT signer links drive the signing pages; each signature rasterizes onto the page, the new document version is hashed and chained, and the next signer is notified. (3) Verification: a verification view recomputes the hash chain and signature list, proving integrity to any third party holding the sealed PDF.

06. System Architecture / Working

Express serves the portal and signing API; pdf-lib composites signatures onto pages; PostgreSQL stores envelopes, fields and the hash-chained event log. The sender uploads, places fields, sets the order and sends; signer 1 gets a single-use link, signs (draw/type), and the server chains the new version hash; reminders fire on schedule; on completion the sealed PDF and audit certificate are emailed. Nodemailer handles invites, reminders and completion mails.

07. Hardware / Software Requirements

- Node.js 18+ with Express
- pdf-lib (PDF field compositing)
- PostgreSQL (envelopes, fields, hash-chained events)
- Nodemailer/SMTP for invites and certificates
- Modern browser with pointer events (signature pad)
- Docker + docker-compose

08. Expected Outcomes

Complete signing ceremonies for envelopes with up to ~20 signers (design target); an audit trail whose hash chain verifies end-to-end; sealed PDFs that fail verification if any byte is altered afterwards; and a report explaining the threat model and the honest boundary between hash-chained integrity and licensed digital-signature certificates.

09. Applications

- Internship offer letters and freelance agreements
- Vendor NDAs and onboarding documents
- College society consent forms and event waivers
- Internal HR policy acknowledgements

10. Future Scope

- Integration with licensed Certifying Authority certificates for legal-grade signatures
- Government-ID / video-KYC identity verification step
- Bulk-send: one template to many independent signers
- Long-term validation (LTV) timestamps on sealed documents