PRD —

NYAYA SETU (MVP)

Hackathon Build | Citizen Legal Document Decoder (India-first)

---

1) Project Overview

Problem

Common citizens in India often receive **legal notices** and other legal documents that are:

- hard to understand (legal jargon, formal writing)
- stressful / panic-inducing
- not accessible in the user's **mother tongue**
- difficult to interpret without professional help

MVP Goal

Build **NYAYA SETU**, a mobile-responsive web app where a citizen can:

- upload a legal document (PDF/image)
- get a **simplified, vernacular explanation**
- understand key facts (sender, deadline, demands, etc.)
- receive a clear action outcome: **"Consult a lawyer."**

Success Metric

Primary KPI:

- **User trust & satisfaction** measured via **1–5 rating + optional feedback**

---

2) Core Requirements

Platform

- **Web app (mobile responsive)**

Target User

- **Common citizen in India** (no legal background)

Authentication

- **Mandatory login**
- **Phone OTP** authentication
- Session stays logged in for **30 days**

Supported Inputs

- **PDF**
- **JPG/PNG images**
- Multi-image upload supported (treated as pages)
- Max upload size: **10MB**
- **Printed documents only** (no handwritten)

Document Support Scope

- MVP focuses on **Legal Notices**
- If user uploads other legal docs, app must:
    - still decode them
    - auto-detect and label document type

Language Support (MVP)

Total supported languages = **8**

- English
- Hindi
- Marathi
- Tamil
- Telugu
- Bengali
- Gujarati
- Punjabi

Language behavior:

- Ask the user to select preferred output language **every time** during decoding
- Show **one language at a time** (via dropdown)

Storage & Retention

- Store data **locally on browser/device** (no cloud sync)
- Keep only **latest 3 documents**
    - uploading a 4th auto-deletes the oldest
- User can delete a document (with confirmation)
- After delete: show **Undo toast (5 seconds)**

Performance

- Accuracy is priority
- Decoding can take **up to 60 seconds**

Safety / Trust

Top risk to avoid:

- **Wrong meaning / mistranslation causing panic**

Mitigation requirements:

- Show a results warning banner: **“This may be inaccurate.”**
- Show **Original Extracted Text** (collapsible)
- Show **document preview** (first page only)

---

3) Core Features

A) Upload + Decode

- Upload PDF or images (multiple)
- Crop/rotate before decode
- Confirm before processing

B) AI Legal Decoder Output

Output must be structured into:

1) Summary tab

- **5–7 lines**
- Simple language
- No legal jargon

2) Key Points tab

- **6–10 bullets**
- Clear structure

3) Action Plan tab

- Only: **“Consult a lawyer.”**
- No extra steps, no legal advice, no DIY suggestions

C) Extracted Key Details

Displayed above tabs:

- Sender
- Receiver
- Date
- Deadline
- Subject
- Demands

Deadline display must include:

- “Reply by: ”
- Urgency label: **Low / Medium / High**

D) Category Tag (Broad)

Show 1 category tag:

1. Money / Payment Dispute
2. Property / Rent / Land
3. Employment / Workplace
4. Family / Personal Dispute
5. Business / Contract Dispute
   Fallback: **Other / Unclear**

E) Export/Share

- Export as **single PDF report**
- Contains original + decoded results
- No watermark
- No disclaimer inside PDF (disclaimer appears only in UI)

F) Minimal UX + Theme

- Minimal, Apple-like UI
- Dark mode toggle supported (manual toggle)
- No onboarding screens

---

4) Core Components

Screens

1. Login (Phone OTP)
2. Homepage (CTA + recent docs)
3. Upload (PDF/images)
4. Preview & Edit (crop/rotate)
5. Language Selection (every time)
6. Confirm Upload
7. Decoding Loader
8. Results

UI Components

- Phone input + OTP input
- Primary CTA button: **Scan & Decode**
- Recent docs list (max 3)
- Upload selector (PDF / images)
- Preview viewer + crop + rotate
- Language dropdown selector
- Spinner loader (“Decoding…”)
- Results tabs: Summary / Key Points / Action Plan
- Details card (key fields)
- Document type tag + category tag
- Urgency label (Low/Medium/High)
- Warning banner (“This may be inaccurate.”)
- Collapsible Original Extracted Text
- Rename modal
- Delete confirm modal
- Undo toast (5s)
- Export PDF button
- Logout dropdown/menu

---

5) App / User Flow (Forward-Only)

1) Login

- Enter phone → OTP → Verify → session set (30 days)

2) Homepage

- Title: **NYAYA SETU**
- Tagline: **“Legal clarity for everyone.”**
- CTA: **Scan & Decode**
- Dark mode toggle (homepage only)
- Logout (homepage header menu)
- Recent docs (max 3)

3) Upload

- Choose: PDF OR Images (multi-image allowed)
- Validate: type + size <= 10MB

4) Preview & Edit

- Crop / rotate
- Continue

5) Language Selection (every decode)

- User selects one language (8 options)
- Continue

6) Confirmation

- Show minimal confirmation
- Show warning: **“Only your latest 3 documents are kept.”**
- Proceed to Decode

7) Decoding

- Spinner + “Decoding…”

8) Results

Always show:

- Warning banner: “This may be inaccurate.”
- Document type label (auto-detected)
- Category tag

- Details card (sender/receiver/date/deadline/subject/demands)
- Reply by date + urgency label
- Document preview (first page only, modal viewer)

Tabs:

- Summary (5–7 lines)
- Key Points (6–10 bullets)
- Action Plan (“Consult a lawyer.”)

Bottom:

- Collapsible Original Extracted Text

Actions:

- Rename
- Delete (confirm + undo)
- Export PDF

Notes

- No back-navigation between steps
- No “Done” button
- Results stay open until user exits manually

---

6) Techstack

Frontend

- **Next.js** (React)
- **TypeScript**
- **TailwindCSS**
- PWA-ready structure optional (but offline not required)

Auth (OTP)

- **Firebase Auth (Phone OTP)** *(fastest MVP path)*

OCR + Document Extraction

- **Google Cloud Vision OCR** OR **Tesseract (server-side)**
  (hackathon-friendly recommendation: Google Vision for accuracy)

AI Simplification + Vernacular Generation

- LLM layer (one of):
    - OpenAI API (GPT models) for structured output
    - Alternative: Gemini / Claude (depending on availability)

Recommended output enforcement:

- JSON structured response schema for:
    - summary
    - keyPoints[]
    - extractedFields
    - detectedDocType
    - categoryTag
    - urgency

Local Storage (No Cloud)

- Browser storage approach:
    - **IndexedDB** for storing files + decoded output
    - Wrapper library like **Dexie.js** (recommended)

Export PDF

- Client-side PDF generator:
    - **pdf-lib** or **React-PDF**
- Include original document preview + decoded results

Hosting

- **Vercel** (ideal for Next.js)

---

7) Implementation Plan

Phase 0 — PRD Locked (Done)

- Confirm scope: Legal Notice decoder MVP
- Lock languages (8)
- Lock flows and UI constraints

Phase 1 — Core App Skeleton (Day 1)

- Next.js + Tailwind setup
- Routing for screens:
    - /login
    - /home
    - /upload
    - /preview
    - /language
    - /confirm
    - /loading
    - /results
- Forward-only flow enforcement
- Dark mode toggle (homepage only)
- Logout action

Deliverable:

✅ app navigation + UI shell complete

Phase 2 — Auth + Local Storage (Day 1–2)

- Phone OTP using Firebase Auth
- Session persistence 30 days
- IndexedDB setup via Dexie
- Store only last 3 docs:
    - on insert: if count > 3 → delete oldest

Deliverable:

✅ login + saved recent docs working

Phase 3 — Upload → Preview Tooling (Day 2)

- PDF + image upload validation
- Multi-image support
- Crop + rotate UI
- “Proceed to Decode” confirmation screen + warning

Deliverable:

✅ upload pipeline ready for decode

Phase 4 — OCR + LLM Decoder Pipeline (Day 2–3)

- OCR everything (even PDFs)
- Feed extracted text to LLM with strict schema:
    - detectedDocType
    - extractedFields (sender/receiver/date/deadline/subject/demands)
    - urgency label
    - category tag
    - summary (5–7 lines)
    - keyPoints (6–10)
    - actionPlan fixed string

Deliverable:

✅ decode output stable + structured

Phase 5 — Results Screen + Export (Day 3)

- Details card + urgency + tags
- Tabs: Summary / Key Points / Action Plan
- Collapsible Original Extracted Text
- Document preview modal (first page only)
- Rename + delete + undo toast
- Export single PDF report

Deliverable:

✅ MVP usable end-to-end

Phase 6 — Polish + Hackathon Readiness (Final)

- Minimal UI tuning
- Error states for OCR/LLM failures:
    - re-upload screen
    - scan tips
- Rating (1–5) + optional feedback capture
- Basic analytics events (optional)

Deliverable:

✅ demo-ready MVP with clean UX