Dynamic PDF Bill Translation System

Project Summary & End-to-End Workflow

This document describes the architecture, workflow, and internal processing of the Dynamic PDF Bill Translation System. The system converts selected English text fields (such as Party Name and Address) into Hindi while preserving the original PDF layout.

1. Project Overview

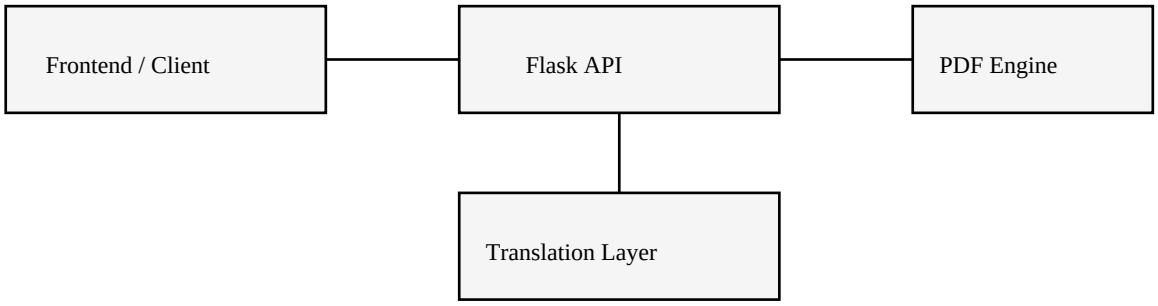
The goal of this system is to translate specific, dynamic fields inside a structured, software-generated PDF invoice from English to Hindi. The solution is configuration-driven, template-aware, and layout-safe.

Instead of regenerating the PDF or relying on OCR, the system works directly at the PDF coordinate level. Exact bounding boxes are defined in a configuration file and reused for extraction, translation, and replacement.

Feature	Description
Config-driven layout	All coordinates are defined in config.json
Dynamic text handling	Party name and address are extracted dynamically
Hindi language support	Unicode Devanagari font rendering
Layout preservation	Tables, numbers, and formatting remain unchanged
Production-ready backend	Flask-based API deployable on Render

2. System Architecture

The system follows a clear separation of concerns. Each stage of the pipeline is isolated into its own module, making the solution easy to debug, extend, and maintain.



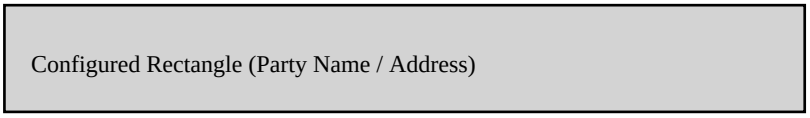
3. End-to-End Workflow

1. User uploads a structured PDF invoice via the API.
2. Backend clears previous uploads and stores the new PDF.
3. Coordinates are loaded from config.json.
4. Text is extracted only from the defined rectangular regions.
5. Extracted English text is dynamically translated to Hindi.
6. Original English text is hidden using a white overlay.
7. Hindi text is written back into the same region, centered horizontally and vertically.
8. The modified PDF is returned to the user.

4. Coordinate-Based Processing

The core of the system relies on precise PDF coordinates. Each translatable field is represented by a rectangle defined in config.json. These rectangles act as the single source of truth throughout the pipeline.

The same coordinates are used for extraction, white overlay placement, and Hindi text insertion.



5. Deployment & Hosting

The backend is designed to be easily deployed on platforms such as Render. The API exposes lightweight health-check endpoints, making it compatible with uptime monitoring services.

For production usage, an always-on hosting plan is recommended to avoid cold starts. The frontend can be deployed independently on platforms such as Vercel.

6. Conclusion

This system provides a robust, scalable, and layout-safe solution for localized PDF document generation. By combining configuration-driven coordinates, dynamic translation, and precise PDF manipulation, it avoids the pitfalls of OCR and template regeneration.

The final output preserves the original invoice structure while seamlessly presenting translated content, making it suitable for real-world billing and compliance use cases.