Batch PDF Watermark Tool

Project Abstract — projectech.in (built to order) · K-1629

Introduction / Background

Stamping DRAFT or CONFIDENTIAL across a folder of PDFs is one of the most common document chores in offices and colleges, and most free tools do it one file at a time — or on someone else's server. This project builds a batch PDF watermark tool as a single-file web app: queue many PDFs, design a text watermark with full typographic control, preview it on the real pages, and stamp the whole batch with per-file downloads, all client-side.

Problem Statement

Single-file watermarking wastes time at batch scale, and server-side tools force users to upload potentially sensitive documents. A tool that parses and rebuilds PDFs locally — stamping text into the content stream rather than flattening pages to images — is both faster to use and honest about privacy.

Objectives

- Queue and process multiple PDFs in one batch run.
- Offer full watermark design: text, font, size, color, opacity, rotation, placement, page scope.
- Render a live preview on the document's real pages.
- Stamp watermarks into the PDF content stream, preserving original pages.
- Persist named presets for one-click reuse.

Proposed Methodology

Dropped files are parsed in-browser; page counts and sizes populate the queue table. The designer controls map to watermark parameters; the preview renderer draws the current design over the first page. The batch engine iterates the queue, opening each page in scope and drawing the watermark text into its content stream (tiled or single placement), then serializes each file for download. Presets serialize the full parameter set to LocalStorage.

System Architecture / Working

Single-file web app: queue layer (file intake, PDF parsing, per-file status), design layer (controls + live preview via pdf.js rendering), engine layer (pdf-lib stamping loop with progress events), and preset layer (named, persisted designs). Embedded pdf.js and pdf-lib mean zero network dependencies; three sample PDFs ship for immediate testing.

Hardware / Software Requirements

- Hardware: any laptop/desktop with a modern browser (2 GB RAM minimum).
- Software: current Chrome, Edge, Firefox or Safari; no installation, no server.
- Input: standard unencrypted PDF files.

Expected Outcomes

- A working tool that watermarks real PDF batches fully offline.
- A clear implementation of PDF content-stream text stamping.
- A preset workflow that makes repeat jobs one-click.
- Complete viva kit: report, PPT and Q&A.

Applications

- CS/IT coursework on file formats and document processing.
- College admin offices stamping drafts and circulars.
- Small businesses marking invoices and quotes.

Future Scope

- Image/logo watermarks with transparency.
- Password-protected PDF support (user-supplied password).
- Page-range scoping beyond all/first-page.
- Command-line batch mode for automation.

This is a built-to-order project. The abstract above summarizes the project as delivered: working implementation, documentation, and viva preparation included.