**⚡ Core Frameworks & Tools**

**🔹 1. Electron (Main Shell)**

* **Electron** → the base framework for building your corporate dashboard desktop app.
  + Handles windows, menus, system integration.
  + Already what you’re using now.

**🔹 2. UI Framework (Frontend)**

Since Electron uses HTML/CSS/JS for UI, you can pick one of these:

* **React + TailwindCSS** → modern, component-based, fast.
* **Vue.js + Vuetify/Tailwind** → simpler learning curve, still powerful.
* **Plain Bootstrap 5 / TailwindCSS** → if you don’t want a full JS framework.

👉 For a **corporate dashboard**, I recommend **React + TailwindCSS** for a clean, modern UI.

**🔹 3. Backend Layer (Inside Electron)**

Electron apps often include a backend service to handle file operations. Options:

* **Node.js (built-in with Electron)** → use fs, path, and express for APIs.
* **Express.js** → create a lightweight REST API inside your Electron app.

**🔹 4. File Upload / Storage**

* **Multer** (if using Express.js) → handles file uploads.
* **Node.js fs module** → move/copy/delete files in your intranet directories.

**🔹 5. Database (Metadata Storage)**

To store file metadata (filename, uploader, department, permissions):

* **SQLite** → lightweight, perfect for Electron apps.
* **PostgreSQL / MySQL** → if you want a central DB server for all departments.
* **NeDB or LowDB** → simple JSON-based DB for very small setups.

**🔹 6. PDF Handling (Read-Only Copies)**

* **pdf-lib** → add watermarks, flatten pages (easy to use).
* [**HummusJS**](https://github.com/galkahana/HummusJS?utm_source=chatgpt.com) → PDF editing, encryption (powerful but older).
* **qpdf** (CLI tool you can call from Node) → set permissions (disable edit/print).
* **PDF.js** → embed PDF viewer inside Electron (for view-only mode).

**🔹 7. Authentication / Permissions**

* **Keytar** → store encrypted credentials in the OS keychain.
* **JWT (jsonwebtoken)** → if you want role-based access per department.
* **Active Directory / LDAP integration** → if you want to tie access to company logins.

**✅ Example Tech Stack for Your Use Case**

If I were building your intranet FMS module in Electron:

* **Frontend:** React + TailwindCSS
* **Backend (inside Electron):** Express.js (API) + fs for file ops
* **Database:** SQLite (per department)
* **PDF Processing:** pdf-lib (for watermark/read-only copies) + PDF.js (viewer)
* **Auth:** JWT or company Active Directory