**📦 Stage 1: Backend Trigger + API Extraction**

**Trigger:**  
When documents land in the **base directory / blob / staging location**, a **trigger** (Azure Function, Logic App, or Batch Job) picks them up.

**✅ Steps:**

1. **Fetch metadata from T220 table**
   * Includes document type, person\_id, filename, etc.
2. **Call Document Extract API**
   * API performs:
     + Classification (e.g., Pay Stub)
     + Splitting (if multi-doc PDF)
     + Field extraction (returns structured JSON per sub-document)
3. **Store in Temporary SQL Table**
   * Each row contains:
     + Document metadata (e.g., doc ID, person\_id)
     + status = extracted / failed
     + reason\_for\_failure (if failed)
     + json\_data field with all extracted fields
     + ✅ **Optional**: Store **base64 or path of split files** (per sub-doc) for UI preview

**🧑‍💻 Stage 2: Heights UI User Review (React)**

**User Action:**  
User clicks **"Extract"** in the Heights UI to review extracted data.

**✅ Steps:**

1. **Fetch relevant rows** from the **temporary SQL table**
   * All rows for that uploaded document or person\_id are fetched.
2. **Display in modal table**
   * Each row represents one **split sub-document**
   * Show key info: pay date, net pay, etc.
3. **User clicks on a row** → opens **editable detail view**
   * Show:
     + **Extracted fields (editable form)**
     + **Document page preview** (from split PDF or base64 image stored earlier)
4. **User edits + clicks Save**
   * Save updates:
     + To Heights API or directly into Heights DB
     + Marks document row as completed or verified
5. **UI Refresh**
   * Heights screen is refreshed with final submitted values